

OPTIMASI BETON BERTULANG PADA STRUKTUR PORTAL RUANG

TUGAS AKHIR SARJANA STRATA SATU

Oleh :

YOHAN NAFTALI

No. Mahasiswa : 07712/TSS

NIRM : 950051053114120051



**PROGRAM STUDI TEKNIK SIPIL
FAKULTAS TEKNIK
UNIVERSITAS ATMA JAYA YOGYAKARTA
YOGYAKARTA
AGUSTUS 1999**

PENGESAHAN

Tugas Akhir Sarjana Strata Satu

**OPTIMASI BETON BERTULANG PADA STRUKTUR
PORTAL RUANG**

Oleh :

YOHAN NAFTALI

No. Mahasiswa : 07712 / TSS

NIRM : 950051053114120051

telah diperiksa, disetujui dan diuji oleh Pembimbing

Yogyakarta, ... Agustus 1999

Pembimbing I

Pembimbing II

(Dr. Ir. FX. Nurwadi W., M.Sc.)

(Ir. Ch. Arief Sudibyo)

Disahkan oleh :

Ketua Program Studi Teknik Sipil

(Ir. Wiryawan Sardjono P., M.T.)

*Sepanjang apapun sebuah perjalanan, selalu dimulai
dengan langkah pertama*



KATA HANTAR

Terima kasih kepada Allah Bapa di surga, karena dengan rahmat dan tuntunan dari pelita Allah tugas akhir ini berhasil diselesaikan, segenap puji dan syukur dipanjatkan demi kemuliaan nama Allah Yang Maha Pengasih.

Tugas akhir yang mengambil judul Optimasi Beton Bertulang Pada Struktur Portal Ruang ini ditulis untuk memenuhi salah satu persyaratan memperoleh gelar Sarjana Teknik dalam kurikulum Strata-1 Program Studi teknik Sipil, Fakultas Teknik, Universitas Atma Jaya Yogyakarta.

Selama penulisan tugas akhir ini banyak doa, dukungan, dan bimbingan baik secara moral maupun secara material yang diberikan oleh berbagai pihak untuk memperlancar penulisan tugas akhir ini. Karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Papa dan Mama tercinta atas doa dan restu.
2. Ir. John Tri Hatmoko, M.Sc., selaku Dekan Fakultas Teknik
3. Ir. Wiryawan Sardjono P., M.T., selaku Ketua Program Studi Teknik Sipil
4. Dr. Ir. FX. Nurwadji Wibowo, M.Sc., selaku Dosen Pembimbing I yang telah memberikan bimbingan dan mengarahkan penulis, sehingga penulis berkesempatan mengenal dunia optimasi dan pemrograman.
5. Ir. Ch. Arief Sudibyo, selaku Dosen Pembimbing II yang telah memberikan bimbingan kepada penulis
6. Segenap dosen dan karyawan Program Studi Teknik Sipil Universitas Atma Jaya.
7. Mutiara atas cinta dan kasihnya untuk penulis.
8. Bu Yenny, Pak Wagianto, dan teman-teman di Laboratorium Teknik Penyehatan.
9. Yerry, Bayu, Fika, Herman, Roy, Benny, Eko, Anwar dan semua teman yang tidak dapat disebutkan satu-persatu atas bantuan dan dukungannya.

Semoga Allah Bapa memberikan imbalan atas budi baik yang telah ditanamkan mereka.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
KATA HANTAR	iii
DAFTAR ISI	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN	viii
ARTI LAMBANG DAN SINGKATAN	ix
INTISARI	xiv
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Materi Tugas Akhir	3
I.2.1. Rumusan Masalah	3
I.2.2. Batasan Masalah	6
I.3 Maksud dan Tujuan Penulisan	7
I.4. Kegunaan Penulisan	8
I.5 Metoda Penulisan	8
BAB II TINJAUAN PUSTAKA	10
II.1 Struktur Portal Ruang	10
II.1.1 Tinjauan Umum	10
II.1.2 Metoda Kekakuan	11
II.1.3 Kekakuan Batang Portal Ruang	16
II.2 Struktur Beton Bertulang	16
II.2.1 Tinjauan Umum	16
II.2.2 Balok	17
II.2.3 Kolom Biaksial	23
II.3 Metoda Optimasi Struktur	26
II.3.1 Tinjauan Umum	26
II.3.2 Metoda Optimasi Polihedron Fleksibel	35
BAB III OPTIMASI BETON BERTULANG PADA STRUKTUR	
PORTAL RUANG	38
III.1 Struktur Program	38
III.2 Sub Program Masukkan dan Keluaran	41
III.2.1 Sub Program Untuk Menampilkan Menu Utama	41
III.2.2 Sub Program Memasukkan Data	42
III.2.3 Sub Program Untuk Membaca Data Masukkan	44
III.3 Subprogram Inti	45
III.3.1 Sub Program Analisa Struktur	45
III.3.2 Sub Program Analisa Balok Persegi Panjang	46
III.3.4 Sub Program Metoda Optimasi Polihedron Fleksibel .	50
III.3.5 Subprogram Mencetak Hasil Akhir Optimasi	52
BAB IV VALIDASI DAN APLIKASI PROGRAM OPTIMASI	53

IV.1 Validasi Program Optimasi	53
IV.2 Aplikasi Program Optimasi	58
BAB V KESIMPULAN DAN SARAN	63
V.1 Kesimpulan	63
V.2 Saran	63
DAFTAR PUSTAKA	65
LAMPIRAN	

DAFTAR TABEL

2-1 Matrik Kekakuan Batang Portal Ruang	15
4-1 Beban Merata Pada Struktur Portal Bidang	54
4-2 Beban Tekanan Angin	55
4-3 Perbandingan Volume Beton dan Berat Besi	55
4-4 Perbandingan Harga Struktur	58
4-5 Dimensi Struktur Portal Ruang Hasil Optimasi	61
4-6 Rincian Harga Struktur Portal Ruang	62

DAFTAR GAMBAR

2-1 Penampang Tulangan Rangkap	19
2-2 Diagram Regangan	19
4-1 Geometri Struktur Portal Bidang	53
4-2 Penomoran Titik Kumpul dan Batang Portal Bidang	54
4-3 Geometri Struktur Portal Ruang	59
4-4 Penomoran Titik Kumpul dan Batang Portal Ruang	59
4-5 Grafik Perbandingan Nilai Fitness	60
4-6 Grafik Perbandingan Lama Proses Optimasi	61

DAFTAR LAMPIRAN

1. Program Utama	67
2. File BORLANDC.HPP	68
3. File HEADER.HPP	69
4. File PROTO.HPP	70
5. File VARIABEL.HPP	71
6. File INOUT.HPP	76
7. File STRUKTUR.HPP	86
8. File PEMBEBANAN.HPP	94
9. File SOLVER.HPP	99
10. File KOLOM.HPP	101
11. File BALOK.HPP	111
12. File ELEMEN.HPP	117
13. File POLYHEDRON.HPP	119
14. File PENORMALAN.HPP	126
15. File PENGACAKAN.HPP	128
16. File DISKRITISASI.HPP	130
17. File KENDALA.HPP	131
18. File TELUSUR.HPP	134
19. File BARU.HPP	137
20. File PENGURUTAN.HPP	140
21. File TAMPILAN.HPP	142
22. File CETAK.HPP	145
23. Konfigurasi kompilasi	152
24. Data input struktur portal gedung 2 lantai	154
25. Data diskrit struktur portal gedung 2 lantai	155
26. Data pembebanan struktur portal gedung 2 lantai	156
27. Hasil informasi struktur portal gedung 2 lantai	157
28. Hasil tiwayat proses optimasi struktur portal gedung 2 lantai	160
29. Hasil akhir struktur terbaik untuk kasus portal gedung 2 lantai	161
30. Hasil analisa struktur portal gedung 2 lantai	170
31. Laporan kendala portal gedung 2 lantai	172
32. Data input struktur portal ruang	174
33. Data diskrit struktur portal ruang	175
34. Data pembebanan struktur portal ruang	176
35. Hasil informasi struktur portal ruang	177
36. Hasil riwayat proses optimasi kasus-kasus portal ruang	179
37. Hasil akhir struktur terbaik untuk kasus portal ruang	181
38. Hasil analisa struktur portal ruang	185

ARTI LAMBANG DAN SINGKATAN

a	Tinggi blok tegangan persegi ekuivalen
A_C	Beban titik kumpul gabungan
A_E	Beban titik kumpul ekuivalen
A_{FC}	Beban titik kumpul gabungan yang selaras dengan D_F
A_{Mi}	Gaya di ujung batang i
A_{ML}	Gaya ujung batang akibat beban
A_{MSi}	Gaya jepit kedua ujung i dalam arah sumbu struktur
A_R	Reaksi di titik kumpul yang dikekang
A_{RC}	Beban titik kumpul gabungan yang selaras dengan perpindahan pada titik kumpul yang dikekang
A_S	Luas tulangan pada daerah tarik
A_S'	Luas tulangan pada daerah tekan
b	Lebar penampang beton
B	Lebar kolom
c	Jarak dari serat tekan terluar ke garis netral
C_C	Gaya internal pada daerah tekan beton
C_S	Gaya internal pada daerah tekan baja tulangan
d	Jarak dari serat tekan terluar ke pusat tulangan tarik
d'	Jarak dari serat tarik terluar ke pusat tulangan desak
dia	Diameter tulangan pada kolom
$diatldslp$	Diameter tulangan desak pada daerah lapangan balok
$diatldstm$	Diameter tulangan desak pada daerah tumpuan balok

d_{atlg}	Diameter tulangan geser
d_{atlrtp}	Diameter tulangan tarik pada daerah lapangan balok
$d_{atlrtpm}$	Diameter tulangan tarik pada daerah tumpuan balok
D_F	Perpindahan titik kumpul bebas
D_{Mi}	Perpindahan di ujung batang i
D_R	Perpindahan titik kumpul yang dikekang
E	Modulus elastisitas untuk gaya aksial
E_C	Modulus elastisitas untuk beton
E_S	Modulus elastisitas baja tulangan
f	Fungsi sasaran
f_c'	Kuat desak karakteristik beton
f_s'	Tegangan yang terjadi pada baja tarik
f_y	Kuat tarik baja tulangan
g_j	Fungsi kendala pertaksamaan ke- j
G	Modulus elastisitas untuk geser
h_j	Fungsi kendala persamaan ke- j
H	Tinggi kolom
I_x	Momen inersia pada sumbu x
I_y	Momen inersia pada sumbu y
I_z	Momen inersia pada sumbu z
$jaraktlg$	Jarak tulangan geser
JVD	Jumlah variabel desain
L	Panjang bentang batang

m	Jumlah batang
M	Momen
M_n	Momen tahanan nominal penampang
M_{ox}	Momen desain efektif pada arah x
M_{oy}	Momen desain efektif pada arah y
M_x	Momen yang terjadi pada arah sumbu x
M_y	Momen yang terjadi pada arah sumbu y
nB	Jumlah balok pada struktur
nK	Jumlah kolom pada struktur
$ntldslp$	Jumlah tulangan desak pada daerah lapangan balok
$ntldstm$	Jumlah tulangan desak pada daerah tumpuan balok
$ntltrlp$	Jumlah tulangan tarik pada daerah lapangan balok
$ntltrtm$	Jumlah tulangan tarik pada daerah tumpuan balok
N	Jumlah tulangan pada salah satu sisi kolom
N_{itik}	Jumlah titik coba
P	Gaya aksial
P_u	Gaya aksial ultimit
S_{FF}	Gaya A_F akibat satu satuan perpindahan D_F
S_J	Matrik kekakuan titik terakit
S_M	Kekakuan batang dalam arah sumbu struktur
S_{MSi}	Kekakuan batang untuk kedua ujung batang i dalam arah sumbu struktur
S_{RF}	Reaksi A_R akibat satu perpindahan D_F

T	Gaya internal pada daerah tarik baja tulangan
x	Variabel Desain
X	Ruang variabel Desain
X_B	Koordinat x titik terbaik
X_G	Koordinat x titik <i>good</i>
X_M	Koordinat x titik tengah
X_S	Koordinat x arah penelusuran
X_T	Koordinat x titik coba baru
X_W	Koordinat x titik terjelek
Y_B	Koordinat y titik terbaik
Y_G	Koordinat y titik <i>good</i>
Y_M	Koordinat y titik tengah
Y_S	Koordinat y arah penelusuran
Y_T	Koordinat y titik coba baru
Y_W	Koordinat y titik terjelek
β	Koefisien biaksial
β_I	Faktor reduksi tinggi blok tegangan
ε	Faktor konvergensi
ε_s	Regangan pada tulangan tarik
ε_s'	Regangan pada tulangan desak
ϕ	Faktor reduksi kekuatan
ρ	Rasio penulangan pada daerah tarik
ρ'	Rasio penulangan pada daerah tekan

ρ_b	Rasio imbang
ρ_{maks}	Rasio penulangan maksimum yang diijinkan

INTISARI

OPTIMASI BETON BERTULANG PADA STRUKTUR PORTAL RUANG, Yohan Naftali, tahun 1999, Peminatan Program Studi Struktur, Program Studi Teknik Sipil, Fakultas Teknik, Universitas Atma Jaya Yogyakarta.

Desain struktur merupakan salah satu bagian dari keseluruhan proses perencanaan bangunan. Semakin mahal harga material menyebabkan harga struktur bangunan menjadi semakin mahal, oleh karena itu untuk menekan biaya bangunan, perlu dicari dimensi batang sehingga memberikan harga struktur yang paling murah (optimal), tanpa melanggar kendala-kendala yang ada. Salah satu metoda untuk mendapatkan struktur yang optimal dapat dilakukan dengan proses optimasi. Kasus optimasi yang terjadi dalam bidang teknik pada umumnya berupa masalah yang tidak linier, untuk itu perlu digunakan metoda optimasi yang dapat menyelesaikan masalah tidak linier, dari dua metoda yaitu dengan bantuan kalkulus dan metoda numerik.

Beton bertulang merupakan material yang banyak digunakan untuk membuat struktur bangunan karena material pembentuknya mudah didapat. Balok utama pada struktur beton bertulang langsung ditumpu oleh kolom dan dianggap menyatu secara kaku dengan kolom, sistem ini dikatakan sebagai sistem portal. Portal ruang merupakan sistem portal yang dianalisa secara 3 dimensi. Portal ruang memiliki 6 perpindahan yang mungkin terjadi pada setiap titik kumpulnya.

Harga struktur yang murah tetapi tidak melanggar kendala yang ada merupakan fungsi sasaran yang dicari dalam tugas akhir ini. Variabel desainnya berupa dimensi penampang beton dan tulangan baja yang digunakan. Balok dan kolom berpenampang empat persegi panjang dan *uniform* sepanjang bentangnya. Hasil tugas akhir ini berupa sebuah perangkat lunak untuk optimasi beton bertulang pada struktur portal ruang. Kode program ditulis dengan bahasa C++ dan dikompilasi menggunakan Borland C++.

Penyelesaian masalah optimasi beton bertulang pada struktur portal ruang memakai metoda kalkulus terlalu rumit, oleh karena itu dalam tugas akhir ini digunakan metoda optimasi polihedron fleksibel. Dalam menganalisa struktur portal ruang digunakan metoda kekakuan. Balok dianalisa sebagai balok berpenampang empat persegi panjang dan bertulangan rangkap, kolom dianalisa sebagai kolom biaksial bertulangan simetri.

Hasil yang diperoleh dengan menggunakan program optimasi beton bertulang pada struktur portal ruang yang dikembangkan dalam tugas akhir ini dapat mengurangi harga struktur sekitar 9,9 % dibandingkan hitungan struktur yang dikerjakan tanpa dioptimasi. Dalam metoda polihedron fleksibel, semakin banyak titik coba yang digunakan, harga struktur yang didapat menjadi semakin murah, akan tetapi membutuhkan waktu proses optimasi yang semakin lama. Pengembangan optimasi beton bertulang pada struktur portal ruang merupakan hal yang masih luas dan menarik untuk dipelajari, terutama masalah pendetailan antara sambungan balok kolom dan analisis struktur dinamis yang belum dibahas dalam tugas akhir ini.

BAB I

PENDAHULUAN

I.1 Latar Belakang

Desain struktur merupakan salah satu bagian dari keseluruhan proses perencanaan bangunan. Proses desain dapat dibedakan menjadi dua tahap, tahap pertama yaitu desain umum yang merupakan peninjauan secara garis besar keputusan-keputusan desain, misalnya tata letak struktur, geometri atau bentuk bangunan, dan material, sedangkan tahap kedua adalah desain secara terinci yang antara lain meninjau tentang penentuan besar penampang balok dan kolom, luas dan penempatan tulangan yang dibutuhkan pada struktur beton bertulang dan sebagainya, untuk itu perlu dimodelkan struktur yang akan didesain sehingga dapat diperoleh besaran gaya dan informasi perpindahan yang diperlukan dalam menentukan luasan beton beserta penulangannya.

Untuk memodelkan suatu struktur bangunan yang baik dapat dimodelkan sebagai portal ruang. Portal ruang merupakan suatu struktur yang pada tiap titiknya memiliki enam perpindahan yang mungkin terjadi, sehingga bukan lagi dua atau tiga seperti pada struktur rangka atau portal bidang.

Dalam menentukan dimensi balok dan kolom pada struktur beton bertulang harus memperhatikan masalah kekuatan dan biaya, terutama pada saat ini dimana harga material semakin mahal, banyak orang yang menunda bahkan membatalkan proyeknya karena masalah biaya bahan bangunan yang mahal, oleh karena itu dalam merencanakan struktur bangunan sangat perlu diperhatikan

masalah tersebut. Kekuatan yang dibutuhkan oleh suatu struktur beton bertulang dapat dicapai dengan memberikan luasan penampang beton dan tulangan yang cukup. Analisis perhitungan yang dilakukan diharapkan dapat memperoleh hasil yang aman dan ekonomis. Untuk mendapatkan hasil yang paling murah (optimal) dapat dicapai dengan menggunakan proses optimasi. Hasil yang didapat merupakan hasil yang mempunyai harga struktur yang paling murah tetapi tetap masih mampu mendukung beban struktur dengan aman.

Metoda optimasi semakin berkembang seiring dengan perkembangan komputer. Dalam bidang teknik sipil, formulasi masalah optimasinya berupa formula tidak linear, oleh karena itu metoda optimasi yang berkembang dalam bidang teknik adalah metoda optimasi non-linear.

Dalam menyelesaikan masalah optimasi non-linear, dapat diselesaikan dengan bantuan kalkulus yang memerlukan informasi turunan, hal ini dapat dilakukan apabila masalah yang ada belum begitu kompleks, untuk masalah yang sangat kompleks dikembangkan metoda yang idenya berasal dari alam, seperti metoda algoritma genetik yang berasal dari teori Darwin "*Survival to the fittest*", metoda *simulated annealling* yang didasarkan pada proses mengerasnya logam, metoda *flexible polyhedron* yang dalam prosesnya variabel desain akan membentuk suatu segi banyak yang makin lama semakin mengecil. Dari banyak metoda di atas, sampai saat ini belum ada metoda yang dapat digunakan untuk menyelesaikan segala kasus dengan memuaskan, karena masing-masing metoda memiliki keunggulan dan kerugiannya sendiri-sendiri.

Oleh karena itu dalam tugas akhir ini, untuk melakukan optimasi beton

bertulang pada struktur portal ruang digunakan metoda polihedron fleksibel.

I.2 Materi Tugas Akhir

I.2.1 Rumusan Masalah

Proses optimasi adalah proses mencari nilai variabel desain yang memberikan nilai optimal yang berupa nilai maksimum ataupun nilai minimum dari fungsi sasaran tanpa melanggar kendala-kendala yang ada.

Secara umum masalah optimasi dapat dirumuskan sebagai :

$$\text{minimumkan } f(x), x \in X \quad (1-1)$$

yang memenuhi kendala :

$$g_j(x) \leq 0 \quad j=1,2,\dots,n \quad (1-2)$$

$$h_j(x) = 0 \quad j=1,2,\dots,m \quad (1-3)$$

pada persamaan di atas x menyatakan variabel desain, X menyatakan ruang variabel desain, $f(x)$ adalah fungsi sasaran, $g(x)$ fungsi kendala pertaksamaan, $h(x)$ fungsi kendala persamaan, n jumlah kendala pertaksamaan, m jumlah kendala persamaan.

Harga struktur yang minimum merupakan fungsi sasaran dalam optimasi ini, dengan variabel disainnya terdiri dari volume beton dan berat tulangan, dalam hal ini tulangan terdiri dari tulangan tarik dan desak untuk balok, tulangan geser untuk balok, tulangan kolom, tulangan geser untuk kolom. Kendala pertaksamaan yang harus dipenuhi berupa momen yang terjadi pada struktur harus lebih kecil atau sama dengan momen yang dapat didukung oleh masing-masing batang balok maupun kolom, gaya geser yang terjadi pada struktur harus lebih kecil atau sama

dengan gaya geser yang didukung oleh beton dan tulangan geser pada masing-masing batang balok maupun kolom, gaya tekan maupun tarik yang terjadi harus lebih kecil atau sama dengan gaya tekan atau tarik yang didukung oleh kolom, serta lendutan yang terjadi harus lebih kecil atau sama dengan lendutan ijin.

Optimasi beton bertulang pada struktur portal ruang ini dapat dirumuskan sebagai :

untuk balok :

minimumkan :

$$\begin{aligned}
 f1(\text{lebar, tinggi, diatltrtm, ntltrtm, diatldstm, ntldstm, diatltrlp,} \\
 \text{ntltrlp, diatldslp, ntldslp, diatlgs, jaraktlgs}) = (\text{lebar} \times \text{tinggi} \times \\
 \text{panjang balok} \times \text{Rp(harga beton)/m}^3) + (\frac{1}{4} \times \pi \times \text{diatltrtm}^2 \times \text{ntltrtm} \\
 \times 2 \times 0.25 \times \text{panjang balok} \times (\text{berat tulangan})\text{kg/m}^3 \times \text{Rp(harga} \\
 \text{tulangan)/kg}) + (\frac{1}{4} \times \pi \times \text{diatldstm}^2 \times \text{ntldstm} \times 2 \times 0.25 \times \text{panjang} \\
 \text{balok} \times (\text{berat tulangan})\text{kg/m}^3 \times \text{Rp(harga tulangan)/kg}) + (\frac{1}{4} \times \pi \times \\
 \text{diatltrlp}^2 \times \text{ntltrlp} \times 0.5 \times \text{panjang balok} \times (\text{berat tulangan})\text{kg/m}^3 \times \\
 \text{Rp(harga tulangan)/kg}) + (\frac{1}{4} \times \pi \times \text{diatldslp}^2 \times \text{ntldslp} \times 0.5 \times \\
 \text{panjang balok} \times (\text{berat tulangan})\text{kg/m}^3 \times \text{Rp(harga tulangan)/kg}) + \\
 (\frac{1}{4} \times \pi \times \text{diatlgs}^2 \times (2 \times (\text{lebar} - 2 \times \text{selimut beton}) + 2 \times (\text{tinggi} - 2 \\
 \text{selimut beton})) \times (\text{panjang balok/jaraktlgs}) \times (\text{berat tulangan})\text{kg/m}^3 \\
 \times \text{Rp(harga tulangan)/kg})
 \end{aligned} \tag{1-4}$$

yang memenuhi kendala pertaksamaan :

$$M_{\text{intern}} \geq M_{\text{extern}} \tag{1-5}$$

$$G_{\text{intern}} \geq G_{\text{extern}} \tag{1-6}$$

$$\text{Lendutan} \leq \text{Lendutan ijin} \quad (1-7)$$

pada perasamaan di atas f_l menyatakan harga balok pada struktur, $lebar$ dan $tinggi$ menyatakan ukuran penampang balok. Variabel desainnya terdiri atas $diatltrtm$ menyatakan diameter tulangan tarik pada daerah tumpuan, $ntltrtm$ menyatakan jumlah tulangan tarik pada daerah tumpuan, $diatldstm$ menyatakan diameter tulangan desak pada daerah tumpuan, $ntltrtm$ menyatakan jumlah tulangan desak pada daerah tumpuan, $diatltrlp$ menyatakan diameter tulangan tarik pada daerah lapangan, $ntltrlp$ menyatakan jumlah tulangan tarik pada daerah lapangan, $diatldslp$ menyatakan diameter tulangan desak pada daerah lapangan, $ntldslp$ menyatakan jumlah tulangan desak pada daerah lapangan, $diatlgs$ menyatakan diameter tulangan geser, $jaraktlgs$ menyatakan jarak antar tulangan geser. M adalah momen dan G adalah gaya geser.

untuk kolom :

minimumkan :

$$\begin{aligned} f2(B,H,dia,N,diatlgskl,jaraktlgskl) = & (B \times H \times \text{tinggi kolom} \times \\ & Rp(\text{harga beton})/m^3) + (1/4 \times \pi \times dia^2 \times (4N-4) \times \text{tinggi kolom} \times \\ & (\text{berat tulangan})kg/m^3 \times Rp(\text{harga tulangan})/kg) + (1/4 \times \pi \times \\ & diatlgskl^2 \times (2 \times (B - \text{selimut beton}) + 2 \times (H - \text{selimut beton})) \times \\ & (\text{tinggi kolom} / jaraktlgskl) \times (\text{berat tulangan})kg/m^3 \times Rp(\text{harga} \\ & \text{tulangan})/kg) \end{aligned} \quad (1-8)$$

yang memenuhi dengan kendala persamaan :

$$M_{\text{intern}} \geq M_{\text{extern}} \quad (1-9)$$

$$G_{\text{intern}} \geq G_{\text{extern}} \quad (1-10)$$

$$P_{\text{intern}} \geq P_{\text{extern}} \quad (1-11)$$

pada persamaan di atas f_2 menyatakan harga kolom pada struktur, dengan variabel desain B dan H menyatakan ukuran penampang beton kolom, d_{ia} menyatakan diameter tulangan, N merupakan jumlah tulangan yang digunakan, $d_{iatlgskl}$ merupakan diameter tulangan geser kolom, $jaraktlgskl$ menyatakan jarak antara tulangan geser pada kolom. M adalah momen, G adalah gaya geser, P adalah gaya aksial pada kolom.

Harga struktur portal seluruhnya dapat dihitung dengan

$$f = \sum_{b=1}^{b \leq nB} f1_b + \sum_{k=1}^{k \leq nK} f2_k \quad (1-12)$$

pada persamaan di atas f merupakan harga portal total, $f1_b$ menyatakan harga balok b , $f2_k$ menyatakan harga kolom k , nB menyatakan jumlah balok, nK menyatakan jumlah kolom.

Untuk menyelesaikan masalah tersebut akan dibuat program komputer yang ditulis dalam bahasa pemrograman yang sudah banyak digunakan yaitu bahasa C++ dan menggunakan data diskrit serta menggunakan metoda optimasi polihedron fleksibel yang dapat digunakan untuk menangani masalah non-linear.

1.2.2 Batasan Masalah

Dalam melakukan optimasi beton bertulang pada struktur portal ruang ini perhitungan beton bertulang didasarkan pada peraturan yang berlaku di Indonesia yaitu Tata Cara Perhitungan Struktur Beton Untuk Bangunan Gedung SK SNI T-15-1991-03 (PU, 1991), sedangkan untuk melakukan perhitungan gaya pada struktur digunakan metoda kekakuan (Weaver, 1980). Mengingat luasnya

permasalahan dan keterbatasan waktu, maka dalam tugas akhir ini diperlukan pembatasan masalah, yaitu :

- a. Analisa struktur portal ruang memakai metoda kekakuan
- b. *Rigid-end offset* diabaikan
- c. Beban yang bekerja pada struktur adalah beban gravitasi menerus
- d. Balok dan kolom berpenampang empat empat persegi panjang dan *uniform* sepanjang bentangnya
- e. Tidak ada gelincir antara beton dengan tulangan baja
- f. Perencanaan kolom biaksial menggunakan metoda kontur beban
- g. Gaya puntir diabaikan
- h. Perencanaan beton bertulang menggunakan analisis ultimit

I.3 Maksud dan Tujuan Penulisan

Penulisan tugas akhir ini dimaksudkan untuk memenuhi syarat yudisium dalam mencapai tingkat kesarjanaan Strata 1 (S1) pada Program Studi Teknik Sipil, Fakultas Teknik, Universitas Atma Jaya Yogyakarta.

Selain itu penulisan tugas akhir ini juga bertujuan untuk mengetahui lebih dalam tentang metoda-metoda optimasi yang sekarang ini mulai dirasakan manfaatnya dan mulai dikembangkan, terutama metoda yang dapat digunakan untuk melakukan optimasi biaya portal beton bertulang, yang memiliki permasalahan yang kompleks.

Hasil dari penulisan tugas akhir ini adalah berupa program komputer yang ditulis dengan bahasa C++ yang dapat dikompilasi menggunakan Borland C++

5.0 atau versi yang lebih tinggi pada sistem operasi Windows 32 bit menjadi sebuah perangkat lunak.

I.4 Kegunaan Penulisan

Perangkat lunak yang dihasilkan dalam penulisan ini berguna untuk merencanakan ukuran penampang balok dan kolom beserta penulangannya dengan memasukkan informasi tentang bentuk geometri portal ruang yang akan direncanakan.

Selain itu program yang dibuat ini dapat dijadikan dasar pengembangan perangkat lunak untuk menyelesaikan masalah optimasi sehingga dapat tercipta sebuah program yang lebih baik lagi, sehingga akan didapatkan hasil yang lebih optimum lagi.

I.5 Metoda Penulisan

Dalam melakukan penulisan ini banyak langkah-langkah yang diambil sebelum mendapatkan sebuah perangkat lunak yang siap digunakan untuk melakukan optimasi.

Tahap pertama, dilakukan studi pustaka mengenai optimasi beton bertulang pada portal ruang, metoda yang digunakan untuk menganalisis struktur digunakan metoda kekakuan, beton dianalisis dengan analisis ultimit, sedangkan kolom biaksial dianalisis menggunakan metoda kontur beban.

Tahap kedua adalah membuat suatu *listing program* untuk melakukan analisa struktur menggunakan metoda kekakuan, juga membuat *listing program*

untuk melakukan analisa balok menggunakan analisis ultimit, serta *listing program* untuk melakukan analisis kolom biaksial menggunakan metoda kontur beban.

Tahap ketiga adalah menggabungkan *listing program* yang telah diuji validasinya masing-masing sebagai *subrutin* yang dipanggil oleh sebuah fungsi utama, dalam fungsi utama itu dilakukan proses optimasi.

Tahap keempat adalah melakukan uji validasi dari perangkat lunak yang digunakan dengan hasil yang pernah dihitung tanpa menggunakan proses optimasi atau dengan perangkat lunak lain yang sudah jadi.

Tahap terakhir adalah mencoba perangkat lunak untuk memecahkan kasus-kasus yang ada untuk melakukan penyusunan laporan tugas akhir secara lengkap.

BAB II

TINJAUAN PUSTAKA

II.1 Struktur Portal Ruang

II.1.1 Tinjauan Umum

Langkah pertama dalam mendesain struktur beton bertulang adalah memperoleh informasi tentang momen, gaya geser, dan gaya aksial yang akan ditahan. Hal ini biasanya melibatkan perhitungan numerik yang rumit, oleh karena itu langkah terbaik adalah memprosesnya dengan program analisa struktur yang telah dibuat dalam komputer (Hulse dan Mosley, 1986, halaman 17).

Dengan perkembangan komputer yang cukup pesat dan harganya relatif murah, sekarang telah beredar program-program struktur, baik hasil buatan sendiri maupun perangkat lunak yang diperdagangkan. Dengan perangkat ini suatu analisis struktur dapat diselesaikan jauh lebih cepat, dan respon sistem struktur dapat diperoleh. Saat ini analisis tiga dimensi telah umum digunakan (Wahyudi dan Rahim, 1997, halaman 17).

Dalam menganalisa struktur rangka (*Framed structure*) dengan bantuan komputer dapat digunakan metoda matriks, baik dengan metoda gaya (*flexibility*) maupun metoda kekakuan (*stiffness*). Metoda kekakuan atau lebih dikenal sebagai metoda perpindahan lebih sesuai untuk diprogram dalam komputer, karena setelah model analisis struktur ditentukan, pertimbangan teknis yang lebih lanjut tidak lagi diperlukan, di sini terdapat perbedaan dengan metoda gaya, walaupun kedua metoda ini sama bentuk matematisnya, pada metoda gaya besaran yang tidak

diketahui adalah gaya kelebihan (*redundant*) yang dipilih secara sembarang, sebaliknya dalam metoda kekakuan, yang tidak diketahui adalah perpindahan titik kumpul yang tertentu secara otomatis. Jadi, jumlah yang tak diketahui dalam metoda kekakuan sama dengan derajat ketidaktentuan (*indeterminacy*) kinematis struktur (Weaver dan Gere, 1980, halaman 1).

Struktur rangka dibagi atas enam kategori : balok, rangka batang bidang, rangka batang ruang, portal bidang, balok silang (*grid*) dan portal ruang, setiap struktur rangka terdiri dari batang-batang yang panjangnya jauh lebih besar dibandingkan ukuran penampang lintangnya. Titik kumpul struktur rangka adalah titik pertemuan batang-batang, termasuk tumpuan dan ujung bebas suatu batang. Tumpuan bisa merupakan jepitan, sendi, atau tumpuan rol, dalam kasus tertentu sambungan antara batang atau batang dengan tumpuan bisa bersifat elastis (atau semi-tegar). Beban pada struktur rangka bisa berupa gaya terpusat, beban tersebar, atau kopel (Weaver dan Gere, 1980, halaman 2).

II.1.2. Metoda Kekakuan

Metoda kekakuan pertama dikembangkan dari superposisi gaya untuk koordinat perpindahan bebas, kemudian metoda ini diformalisasi dan diperluas dengan memakai matriks kesepadanan (*compatibility*) dan konsep kerja maya (*virtual*). Pada pendekatan kerja maya, matriks kekakuan titik kumpul (*joint stiffness*) yang lengkap dirakit dengan perkalian tiga matriks, walaupun versi formal tersebut menarik dan beraturan, tetapi membutuhkan matriks kesepadanan yang besar dan jarang (*sparse*). Selain itu, elemen matriks ini tidak mudah dinilai dengan tepat. Baik cara pembentukan matriks di atas ataupun perakitan dengan

proses perkalian tidak sesuai untuk diprogram pada komputer. Metodologi yang baik adalah menarik ide dari kedua pendekatan tersebut dan menambahkan beberapa teknik yang berorientasi pada komputer, sehingga didapatkan suatu cara yang disebut metoda kekakuan langsung (*direct stiffness method*) yang lebih mudah untuk diprogram pada komputer (Weaver dan Gere, 1980, halaman 148).

Wahyudi dan Rahim (1997, halaman 17) menyebutkan prosedur umum metoda kekakuan ataupun metoda elemen hingga dapat dibagi menjadi lima bagian yaitu :

- a. Penyusunan vektor beban dan matriks kekakuan elemen.
- b. Penyusunan vektor beban dan matriks kekakuan global.
- c. Penyelesaian persamaan kekakuan yang menghasilkan perpindahan global.
- d. Perhitungan deformasi elemen dengan menggunakan transformasi koordinat.
- e. Penentuan tegangan atau gaya-gaya elemen.

Kunci penyederhanaan proses perakitan matriks kekakuan titik S_J ialah pemakaian matriks kekakuan batang untuk aksi dan perpindahan di kedua ujung setiap batang. Jika perpindahan batang didasarkan pada koordinat struktur global, maka ia akan berimpit dengan perpindahan titik kumpul. Dalam hal ini semua komplikasi geometris harus ditangani secara setempat, dan transfer informasi batang ke *array* struktur bersifat langsung, yaitu matriks kekakuan dan vektor beban ekuivalen dapat dirakit dengan penjumlahan langsung sebagai ganti perkalian matriks. Dengan demikian, perakitan matriks kekakuan titik untuk m batang dapat dituliskan sebagai :

$$S_J = \sum_{i=1}^m S_{MSi} \quad (2-1)$$

dalam persamaan ini, simbol S_{MSi} menyatakan matriks kekakuan batang ke- i dengan gaya ujung dan perpindahan (untuk kedua ujung) yang dalam arah sumbu struktur. Agar dapat dijumlahkan, semua matriks kekakuan batang harus diperluas ke ukuran yang sama seperti S_J dengan menambahkan baris dan kolom nol. Akan tetapi, operasi ini dapat dihindari dalam pembuatan program dengan meletakan elemen dari S_{MSi} ke lokasi yang sesuai pada S_J . Demikian juga halnya, vektor beban ekivalen A_E dapat dibentuk dari kontribusi batang sebagai berikut :

$$A_E = -\sum_{i=1}^m A_{MSi} \quad (2-2)$$

dalam hal ini A_{MSi} adalah vektor gaya (aksi) jepit ujung dalam arah sumbu struktur di kedua ujung batang i . Seperti pada S_{MSi} , secara teoritis A_{MSi} perlu diperbesar dengan bilangan nol agar dapat dijumlahkan secara matriks, tetapi langkah ini bisa dihindari dalam program. Beban titik tumpul ekivalen dalam persamaan (2-2) kemudian dijumlahkan dengan beban sebenarnya yang diberikan di titik kumpul untuk membentuk vektor beban total atau gabungan.

Untuk memisahkan suku yang berkaitan dengan perpindahan bebas struktur dari perpindahan pengekang tumpuan (*support restraint*), matriks kekakuan dan beban harus ditata ulang (*rearranged*) dan disekat (*partitioned*). Penataan ulang suku-suku ini dapat dilakukan setelah perakitan selesai. Dalam program komputer, penataan ulang ini lebih mudah dilakukan bila informasi ditransfer dari array batang yang kecil ke array struktur yang besar.

Setelah matriks kekakuan dan vektor beban ditata ulang (*rearranged*) dan disekat-sekat (*partitioned*), maka penyelesaian dasar untuk perpindahan pada titik

kumpul yang bebas akibat beban adalah :

$$\mathbf{D}_F = \mathbf{S}_{FF}^{-1} \cdot \mathbf{A}_{FC} \quad (2-3)$$

dalam persamaan ini \mathbf{A}_{FC} adalah vektor beban titikumpul gabungan (sebenarnya dan ekuivalen) yang selaras (*correspond*) dengan \mathbf{D}_F . walaupun secara simbolis penyelesaian dalam persamaan (2-3) lebih mudah dituliskan sebagai hasil inversi matriks kekakuan \mathbf{S}_{FF} , tetapi perhitungan \mathbf{S}_{FF}^{-1} dalam program komputer hakekatnya tidak efisien. Sebagai gantinya, matriks koefisien difaktorisasi dan penyelesaian untuk \mathbf{D}_F diperoleh dari sapuan (*sweep*) kemuka dan belakang.

Bagian lain yang hendak dicari adalah reaksi tumpuan dan gaya jepit ujung. Bila hanya pengaruh beban yang diperhitungkan, persamaan untuk reaksi adalah :

$$\mathbf{A}_R = -\mathbf{A}_{RC} + \mathbf{S}_{RF} \cdot \mathbf{D}_F \quad (2-4)$$

Selain itu gaya ujung batang dituliskan sebagai :

$$\mathbf{A}_{Mi} = \mathbf{A}_{MLi} + \mathbf{S}_{Mi} \cdot \mathbf{D}_{Mi} \quad (2-5)$$

Banyak variasi dalam mengorganisasi metoda kekakuan bisa dilakukan untuk pemrograman. Tahapan analisa di atas merupakan pendekatan teratur yang memiliki beberapa segi menguntungkan bila berhadapan dengan masalah yang besar dan rumit (Weaver dan Gere, 1980, halaman 151).

Untuk menganalisa struktur portal ruang digunakan matrik kekakuan batang yang terdapat pada tabel 2-1 dan dijelaskan pada bagian II.1.3.

Tabel 2-1
Matriks Kekakuan Batang Portal Ruang

$$S_{M1} = \left[\begin{array}{cc|cccccc} \frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GI_x}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_x}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ \hline -\frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GI_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_x}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L^2} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L^2} \end{array} \right]$$

II.1.3 Kekakuan Batang Portal Ruang

Untuk membentuk matriks kekakuan titik dengan proses penjumlahan yang ditunjukkan oleh persamaan (2-1), matriks kekakuan batang yang lengkap dalam sumbu arah struktur perlu diturunkan dahulu. Selain itu, matriks kekakuan batang S_{Mi} untuk sumbu arah batang diperlukan dalam menghitung gaya ujung batang dengan persamaan (2-5). Kekakuan batang terhadap sumbu batang selalu dapat diperoleh dan ditransformasi ke sumbu struktur.

Portal ruang merupakan kasus yang paling umum untuk struktur rangka, batang dapat mengalami sembarang kedua belas perpindahan, dengan demikian, matriks kekakuan batang (S_{Mi}) untuk struktur portal ruang berordo 12 x 12 yang dapat dilihat dalam tabel 2-1, dan setiap kolom pada matriks menyatakan aksi akibat salah satu perpindahan satuan.

Matriks kekakuan batang portal ruang diperlihatkan pada tabel 2-1, matriks ini disekat untuk memisahkan bagian yang berkaitan dengan kedua ujung batang, secara umum bentuknya adalah :

$$S_{Mi} = \begin{bmatrix} S_{Mij} & S_{Mjk} \\ S_{Mkj} & S_{Mkk} \end{bmatrix}_i \quad (2-6)$$

pada persamaan di atas subskrip j dan k pada submatriks di atas menunjukan ujung batang (Weaver dan Gere, 1980, halaman 154).

II.2 Struktur Beton Bertulang

II.2.1 Tinjauan Umum

Beton bertulang telah digunakan sebagai material bangunan di setiap

negara. Pada banyak negara seperti Amerika dan Kanada, beton bertulang merupakan material struktur yang dominan dalam konstruksi bangunan. Keuntungan dari penggunaan beton bertulang adalah tulangan baja dan bahan pembentuk beton tersedia di banyak tempat, hanya dibutuhkan kemampuan yang relatif rendah dalam konstruksi beton, dan dari segi ekonomi beton bertulang lebih menguntungkan dibandingkan bahan konstruksi lainnya (MacGregor, 1997, halaman 1).

Pada struktur beton bertulang, balok utama yang langsung ditumpu oleh kolom dianggap menyatu secara kaku dengan kolom. Sistem kolom dan balok induk seperti ini dikatakan sebagai sistem portal. Sistem portal telah lama digunakan sebagai sistem bangunan yang dapat menahan beban vertikal gravitasi dan lateral akibat gempa. Sistem ini memanfaatkan kekakuan balok-balok utama dan kolom. Dengan demikian integritas antara balok utama dan kolom harus mendapat perhatian dan pendetailan tersendiri, karena di sekitar daerah ini timbul gaya geser dan momen yang besar yang dapat menyebabkan retak atau patahnya penampang (Wahyudi dan Rahim, 1997, halaman 14).

II.2.2 Balok

Balok adalah anggota struktur yang paling utama mendukung beban luar serta berat sendirinya oleh momen dan gaya geser (MacGregor, 1997, halaman 85).

Dua hal utama yang dialami oleh suatu balok adalah kondisi tekan dan tarik, yang antara lain karena adanya pengaruh lentur ataupun gaya lateral. Padahal dari data percobaan diketahui bahwa kuat tarik beton sangatlah kecil,

kira-kira 10%, dibandingkan kekuatan tekannya. Bahkan dalam problema lentur, kuat tarik ini sering tidak diperhitungkan, sehingga, timbul usaha untuk memasang baja tulangan pada bagian tarik guna mengatasi kelemahan beton tersebut, menghasilkan beton bertulang (Wahyudi dan Rahim, 1997, halaman 39).

Apabila penampang tersebut dikehendaki untuk menopang beban yang lebih besar dari kapasitasnya, sedangkan di lain pihak seringkali pertimbangan teknis pelaksanaan dan arsitektural membatasi dimensi balok, maka diperlukan usaha-usaha lain untuk memperbesar kuat momen penampang balok yang sudah tertentu dimensinya tersebut. Apabila hal demikian yang dihadapi, SK-SNI T-15-1991-03 pasal 3.3.3 ayat 4 memperbolehkan penambahan tulangan baja tarik lebih dari batas nilai ρ maksimum bersamaan dengan penambahan tulangan baja di daerah tekan penampang balok. Hasilnya adalah balok dengan penulangan rangkap dengan tulangan baja tarik dipasang di daerah tarik dan tulangan tekan di daerah tekan. Pada keadaan demikian berarti tulangan baja tekan bermanfaat untuk memperbesar kekuatan balok (Dipohusodo, 1994, halaman 85).

Dalam praktek, sistem tulangan tunggal hampir tidak pernah dimanfaatkan untuk balok, karena pemasangan batang tulangan tambahan di daerah tekan, misalnya di tepi atas penampang tengah lapangan, akan mempermudah pengaitan sengkang (*stirrup*) (Wahyudi dan Rahim, 1997, halaman 60).

Pada gambar 2-1 tampak suatu penampang dengan tulangan tekan yang ditempatkan dalam jarak d' dari serat tekan terluar dan tulangan tarik dalam jarak tinggi efektif d . Diagram regangan yang terjadi dilukiskan dalam gambar 2-2, dengan regangan serat tekan beton dianggap telah mencapai regangan maksimum

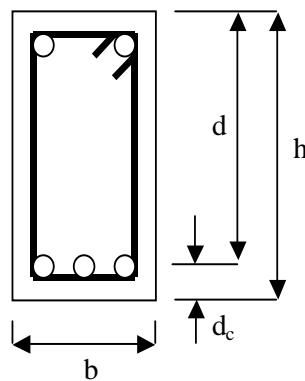
0,003. Garis netral terletak pada jarak c yang belum diketahui nilainya. Melalui perbandingan segitiga dalam diagram ini, dapat ditentukan besar regangan masing-masing tulangan, yaitu :

$$\varepsilon'_s = 0,003 \frac{c - d}{c} = 0,003 \frac{a - \beta_1 d'}{a} \quad (2-7)$$

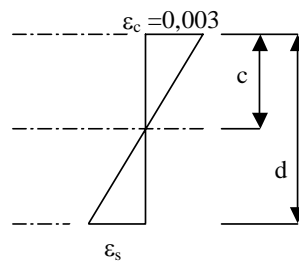
dan

$$\varepsilon_s = 0,003 \frac{d - c}{c} = 0,003 \frac{\beta_1 - a}{a} \quad (2-8)$$

pada persamaan di atas $a = \beta_1 c$



Gambar 2-1 Penampang Tulangan Rangkap



Gambar 2-2 Diagram Regangan

Pertama-tama, kedua tulangan A_s dan A'_s diasumsikan telah mencapai tegangan leleh f_y maka regangannya adalah :

$$\varepsilon'_s = 0,003 \frac{a - \beta_1 d'}{a} \geq \frac{f_y}{E_s} \quad (2-9)$$

dan

$$\varepsilon_s = 0,003 \frac{\beta_1 - a}{a} \geq \frac{f_y}{E_s} \quad (2-10)$$

Bila kedua kondisi persamaan regangan tersebut terpenuhi, tegangan pada baja tulangan menjadi $f_s = f_s' = f_y$ dengan f_s adalah tegangan pada baja tulangan tarik, f_s' adalah tegangan pada baja tulangan tekan, dan f_y adalah tegangan lelehnya.

Diagram tegangan internalnya dapat dianggap menjadi tiga bagian yaitu :

a. pada daerah tekan beton :

$$C_C = 0,85 f_c' a b \quad (2-11)$$

b. Pada daerah tekan baja tulangan

$$C_S = A_s' f_y \quad (2-12)$$

c. Pada daerah tarik baja tulangan

$$T = A_s f_y \quad (2-13)$$

dengan A_s' dan A_s berturut-turut menyatakan luas baja tulangan tekan dan tarik.

Berdasarkan keseimbangan horisontal gaya internal $C_C + C_S = T$, dihasilkan persamaan :

$$0,85 f_c' a b + A_s' f_y = A_s f_y \quad (2-14)$$

dengan tinggi blok tegangan adalah :

$$a = \frac{(A_s + A_s')}{0,85 f_c' b} \quad (2-15)$$

momen tahanan nominal penampang adalah :

$$M_n = 0,85fc'ab(d - a/2) + A_s'fy(d-d') \quad (2-16)$$

Bila regangan dalam persamaan (2-7) dan (2-8) tidak dipenuhi, mungkin tegangan tulangan tekan ataupun tulangan tarik tidak mencapai tegangan leleh materialnya, sehingga dalam hal ini persamaan tersebut tidak berlaku lagi. Untuk dapat menentukan momen ketahanan penampang, regangan aktual harus dihitung, yang dapat dilakukan dengan persamaan (2-7). Selanjutnya diperoleh :

$$f_s' = \varepsilon_s' E_s = 0,003 E_s \left(\frac{a - \beta_1 d'}{a} \right) \quad (2-17a)$$

melalui substitusi nilai a dari persamaan (2-15) dan dibagi (bd), persamaan di atas dapat dirumuskan menjadi :

$$f_s' = 600 \left(1 - \frac{\beta_1 d'}{a} \right) \quad (2-17b)$$

dan

$$f_s' = \varepsilon_s' E_s = 0,003 E_s \left(\frac{\beta_1 d - a}{a} \right) \quad (2-18a)$$

dengan cara yang sama,

$$f_s' = 600 \left(1 - \frac{\beta_1 d'}{a} \right) \quad (2-18b)$$

dengan mensubstitusi persamaan (2-17a) dan (2-18a) ke dalam persamaan (2-14), diperoleh :

$$a = \frac{A_s fy - A_s' f_s'}{0,85 fc' b} \quad (2-19)$$

Sehingga, momen tahanan penampang menjadi :

$$M_n = 0,85f_c'ab(d - a/2) + A_s'fy(d-d') \quad (2-20)$$

Momen disain dapat diperoleh dengan memberikan faktor reduksi tertentu, yang disarankan dalam SK SNI T-15-1991-03, ke dalam persamaan (2-20) (Wahyudi dan Rahim, 1997, halaman 61-63).

Standar SK SNI T-15-1991-03 pasal 2.2.3 ayat 2 memberikan faktor reduksi kekuatan ϕ untuk berbagai mekanisme, antara lain sebagai berikut :

- a. Lentur tanpa beban aksial = 0,80
- b. Geser dan puntir = 0,60
- c. Tarik aksial, tanpa dan dengan lentur (sengkang) = 0,65
- d. Tekan aksial, tanpa dan dengan lentur (spiral) = 0,7
- e. Tumpuan pada beton = 0,70

Dengan demikian dapat dinyatakan bahwa kuat momen M_R (kapasitas momen) sama dengan kuat momen ideal M_n dikalikan faktor ϕ (Dipohusodo, 1994, halaman 41).

Seperti pada tulangan tunggal, keruntuhan tarik atau tekan dapat pula terjadi pada penampang tulangan rangkap. Bila hal tersebut terjadi, keruntuhan tarik dengan pelelehan tulangan lebih disukai daripada keruntuhan tekan dengan kehancuran beton yang mendadak. Keadaan ini dapat dikendalikan dengan memberikan batasan tertentu pada tulangan terpasang. SK SNI T-15-1991-03 pasal 3.3.3 menetapkan batasan :

$$\rho_{maks} = 0,75\rho_b + \rho' \quad (2-21)$$

ρ_b adalah rasio imbang (*balance ratio*) baja tulangan yang bersesuaian dengan penampang tulangan tunggal (Wahyudi dan Rahim, 1997, halaman 64).

Untuk menentukan rasio penulangan keadaan seimbang (ρ_b) digunakan persamaan :

$$\rho_b = \frac{(0,85 f_c' \beta_1)}{f_y} \frac{600}{600 + f_y} \quad (2-22)$$

dengan f_c dan f_y dalam Mpa, β_1 adalah konstanta yang merupakan fungsi dari kelas kuat beton (Dipohusodo, 1994, halaman 37).

Standar SK SNI T-15-1991-03 menetapkan nilai β_1 diambil 0,85 untuk $f_c' \leq 30$ Mpa, berkurang 0,008 untuk setiap kenaikan 1 Mpa kuat beton, dan nilai tersebut tidak boleh kurang dari 0,65 (Dipohusodo, 1994, halaman 31).

II.2.3 Kolom Biaksial

Kolom adalah batang tekan vertikal dari rangka (*frame*) struktur yang memikul beban dari balok. Kolom meneruskan beban-beban dari elevasi atas ke elevasi yang lebih bawah hingga akhirnya sampai ke tanah melalui fondasi. Karena kolom merupakan komponen tekan, maka keruntuhan pada suatu kolom merupakan lokasi kritis yang dapat menyebabkan keruntuhan (*collapse*) lantai yang bersangkutan, dan juga merupakan batas runtuh total (*ultimate total collapse*) seluruh strukturnya. Oleh karena itu dalam merencanakan kolom perlu lebih teliti, yaitu dengan memberikan kekuatan cadangan yang lebih tinggi daripada yang dilakukan pada balok dan elemen struktur horisontal lainnya, terlebih lagi keruntuhan tekan tidak memberikan peringatan awal yang cukup jelas (Nawy, 1990).

SK SNI T-15-1991-03 (PU 1991) memberikan definisi, kolom adalah komponen struktur bangunan yang tugas utamanya menyangga beban tekan aksial dengan bagian tinggi yang tidak ditopang paling tidak tiga kali dimensi lateral terkecil. Sedangkan komponen struktur yang menahan beban aksial vertikal dengan rasio bagian tinggi dengan dimensi terkecil kurang dari tiga disebut pedestal (Dipohusodo, 1994, halaman 287).

Bila eksentrisitas beban mempunyai harga kecil sehingga gaya aksial tekan menjadi penentu, dan juga bila dikehendaki suatu kolom beton dengan penampang lintang yang lebih kecil, maka umumnya distribusi tulangan lebih baik dibuat merata di sekeliling sisi penampang tersebut. Untuk distribusi tulangan semacam ini, baja tulangan yang terletak di bagian tengah penampang akan menerima tegangan yang lebih kecil dibandingkan tulangan lainnya. Ketika kapasitas ultimit kolom tersebut telah tercapai, tegangan pada baja tulangan belum tentu mencapai tegangan lelehnya, sedangkan baja tulangan yang berada di tepi kemungkinan besar sudah leleh (Wahyudi dan Rahim, 1997, halaman 215).

Menurut Winter dan Nilson pada tahun 1994, mekanisme gaya aksial yang bekerja bersamaan dengan lentur pada kedua arah dari sumbu utama penampang terjadi pada kolom-kolom sudut bangunan, pada balok-balok yang membentuk rangka dengan kolom dan menyalurkan momen-momen ujungnya ke kolom dalam dua bidang yang tegak lurus. Keadaan yang sama juga dapat terjadi pada kolom-kolom sebelah dalam, khususnya pada tata letak kolom yang tidak teratur.

Kolom-kolom seperti pada sudut bangunan pada struktur rangka memikul gaya aksial dan sekaligus momen dari dua sumbu aksis yang disebut kolom

biaksial. Untuk menyelesaikan masalah kolom biaksial pada kolom persegi, dapat digunakan prosedur yang umum dipakai yaitu eksentrisitas biaksial, e_x dan e_y , digantikan dengan eksentrisitas uniaksial ekivalen e_{0x} atau e_{0y} , dan kolom didesain sebagai kolom uniaksial (MacGregor, 1997, halaman 466).

Apabila eksentrisitas pada arah x (e_x) dibagi dengan sisi pada arah x lebih besar daripada eksentrisitas pada arah y (e_y) dibagi sisi pada arah y maka momen desain efektif dapat dihitung dengan :

$$M_{ox} = M_x + \beta \frac{h}{b} M_y \quad (2-23)$$

Dengan cara yang sama apabila eksentrisitas pada arah y dibagi dengan sisi pada arah y lebih besar daripada eksentrisitas arah x dibagi sisi arah x, maka momen desain efektifnya adalah :

$$M_{oy} = M_y + \beta \frac{b}{h} M_x \quad (2-24)$$

Dengan M_x adalah momen yang terjadi pada arah x dan M_y adalah momen pada arah y, b adalah sisi efektif pada arah y, h adalah sisi efektif pada arah x, sedangkan β adalah koefisien biaksial yang dapat dihitung persamaan :

$$\beta = 0.3 + \frac{0.7}{0.6} \left(0.6 - \frac{P_u}{bhfc'} \right) \quad (2-25)$$

Dengan β tidak boleh lebih kecil daripada 0,3, P_u adalah gaya aksial (Hulse dan Mosley, 1986, halaman 163).

II.3 Metoda Optimasi Struktur

II.3.1 Tinjauan Umum

Penggunaan metoda optimasi dalam perencanaan struktur sebenarnya bukanlah merupakan hal yang baru dan sudah banyak dikembangkan karena manfaatnya yang banyak dirasakan. Pada tahun 1890 Maxwell mengemukakan beberapa teori tentang desain yang rasional pada suatu struktur yang kemudian dikembangkan lebih lanjut oleh Michell pada tahun 1904 (Wu, 1986, halaman 1). Beberapa penelitian tentang optimasi struktur yang ditujukan untuk penggunaan praktis telah dilakukan sekitar tahun 1940 dan 1950. Pada tahun 1960 Schmit mendemonstrasikan penggunaan teknik pemrograman non-linier untuk desain struktur dan menyebutnya dengan istilah “sintesa struktur” (Wu, 1986, halaman 1). Komputer digital yang kemudian dibuat dan mampu untuk memecahkan masalah numeris dalam skala besar telah memberikan momentum yang besar untuk penelitian. Pada awal tahun 1970 optimasi struktur telah menjadi sesuatu yang penting dalam berbagai aspek perancangan suatu struktur (Wu, 1986, halaman 1).

Ada dua pendekatan utama dalam optimasi struktur. Pendekatan yang satu menggunakan pemrograman matematika dan pendekatan yang lain menggunakan metoda kriteria optimal. Kedua pendekatan ini masing-masing mempunyai kelebihan dan kekurangan.

Setiap struktur rangka memiliki empat hal pokok dengan empat hal ini dapat merupakan suatu variabel yang dapat diubah-ubah untuk mengoptimasi struktur tersebut (variabel desain). Empat hal ini adalah ukuran elemen, geometri

struktur (posisi titik-titik kumpul), gambaran struktur (bagaimana titik-titik kumpul tersebut dihubungkan oleh elemen-elemen), dan material bangunan. Material bangunan biasanya ditentukan terlebih dahulu. Persyaratan-persyaratan yang harus dipenuhi oleh struktur disebut kendala. Dalam sebagian besar kasus, kendala berhubungan dengan kekuatan dan defleksi struktur.

Dalam banyak kasus optimasi struktur yang telah dipecahkan sejauh ini, variabel desain diasumsikan dapat berubah secara kontinyu. Akan tetapi dalam desain yang sesungguhnya kadang-kadang dijumpai variabel desain diskrit yang terbatas. Balok baja dan balok beton bertulang hanya ada dalam ukuran standar. Untuk mengatasi masalah diskrit ini, beberapa usaha telah dilakukan untuk memecahkan kasus-kasus tertentu, antara lain menggunakan pendekatan *branch-and-bound* (Wu, 1986, halaman 4).

Keberhasilan optimasi struktur sejauh ini masih terbatas. Pada satu sisi, optimasi telah berhasil digunakan dalam perancangan berbagai jenis struktur seperti overhead cranes, bangunan-bangunan standard, menara transmisi, girder dengan bentang pendek dan medium, dan bermacam-macam kendaraan termasuk pesawat terbang, mobil, dan kapal. Pada sisi lain, teknik-teknik optimasi masih kurang dapat diterapkan jika struktur sangat besar atau jika kendala-kendalanya terlalu kompleks, belum lagi jika ada pertimbangan-pertimbangan lain seperti standar produksi, pengaruh estetika struktur, dan praktek-praktek konvensional dalam industri (Wu, 1986, halaman 5).

Kesulitan utama dari penggunaan praktis optimasi struktur adalah lamanya proses komputasi sehingga biaya perhitungan menjadi relatif tinggi. Akan tetapi

belakangan ini biaya perhitungan telah turun secara drastis karena komputer pribadi mekin cepat dan makin murah harganya. Dengan demikian optimasi struktur yang sangat tergantung pada komputer akan semakin luas aplikasinya.

Formulasi masalah untuk optimasi struktur dengan variabel desain yang dapat berubah secara kontinyu dan menggunakan analisa struktur elastik adalah program non-linier. Pendekatan untuk pemecahannya menggunakan program linier sekuensial dan program non-linier (Wu, 1986, halaman 6).

Pendekatan linier sekuensialnya adalah melinierkan fungsi kendala non-linier dan fungsi sasaran kemudian menyelesaikan masalah menggunakan program linier berulang-ulang. Romstad dan Wang pada tahun 1967 dan 1968 memberikan penjelasan fisik tentang formulasi ini yaitu : yang tidak diketahui adalah penambahan variabel desain dari suatu titik dalam daerah layak yang diperoleh dari iterasi sebelumnya, dan kendala dari pendekatan linier ini adalah perpindahan ijin dari titik kumpul dan gaya-gaya yang terjadi pada elemen tidak boleh melebihi yang diijinkan. Pendekatan non-liniernya adalah menyelesaikan masalah optimasi dengan menggunakan program non-linier secara langsung. Akan tetapi persamaan analisa elastik menunjukkan masalah yang besar sebab tidak ada algoritma pada program non-linier yang dapat menangani kendala persamaan secara efisien. Teknik yang biasa digunakan untuk menangani masalah ini adalah menggunakan matriks untuk analisa struktur dalam algoritmanya (Wu, 1986, halaman 7).

Formulasi masalah untuk optimasi struktur dengan variabel desain yang dapat berubah secara kontinyu dan menggunakan analisa struktur plastis (analisa

batas) adalah program linier (Wu, 1986, halaman 8). Masalah ini dapat dipecahkan dengan mudah bahkan untuk struktur yang besar. Akan tetapi analisa batas belum dipakai secara luas dalam desain struktur. Di samping itu banyak kendala lain seperti defleksi tidak dapat dipenuhi jika perancangan didasarkan pada analisa batas.

Optimasi struktur dengan sebagian atau seluruh variabel desain merupakan nilai diskrit yang terbatas dan menggunakan analisa struktur elastis memberikan formulasi masalahnya berupa masalah non-linier diskrit. Beberapa pendekatan untuk memecahkan masalah ini telah diusulkan termasuk program integer sekuensial, algoritma penelusuran arah variabel diskrit, dan lain-lain (Wu, 1986, halaman 8).

Program integer sekuensial adalah ekstrapolasi dari program linier sekuensial ke program diskrit. Dalam setiap iterasi, program linier integer digunakan untuk menjamin penelusuran hanya berkisar dari satu titik diskrit ke yang lain. Toakley pada tahun 1968 menemukan bahwa pendekatan ini tidak efisien dan hasilnya tidak bisa diramalkan. Reinschmidt pada tahun 1971 juga mempunyai kesimpulan yang sama meskipun dia telah menunjukkan beberapa contoh kasus numerik yang dipecahkan dengan cara ini, termasuk desain elastis rangka batang yang terdiri atas 9 elemen. Saglam pada tahun 1976 menggunakan pendekatan yang mirip dan telah memecahkan beberapa contoh rangka batang.

Algoritma penelusuran arah variabel diskrit dikemukakan oleh Lai dan Achenbach (1973, halaman 119-131). Struktur portal, kantilever, dan pelat lantai dua lapis dengan kendala dinamis telah dioptimasi dengan metoda ini.

Optimasi struktur dengan sebagian atau seluruh variabel desain merupakan harga diskrit yang terbatas dan menggunakan analisa struktur plastis, formulasi masalahnya adalah program integer atau program integer campuran. Toakley (1968, halaman 1219) merumuskan masalah desain plastis dengan variabel diskrit sebagai program integer dan telah dipecahkan. Fu dan You pada tahun 1976 menggunakan metoda *complex*, tetapi mereka menemukan metoda ini konvergen prematur, kadang-kadang jauh dari nilai optimum. Levey dan Fu (1979, halaman 363-368) menggunakan metoda *complex-simplex*. Lev (1977, halaman 365-371) mengusulkan algoritma branch-and-bound untuk memecahkan masalah diskrit dengan kendala linier. Algoritma ini mengatasi keterbatasan dari beberapa algoritma program integer yang lain.

Selama perkembangan optimasi struktur, para ilmuwan secara bertahap mengembangkan banyak cara yang lain untuk merumuskan dan memecahkan berbagai masalah sehingga banyak algoritma-algoritma yang diusulkan. Sementara situasi ini dikenal secara umum sebagai sesuatu yang tidak dapat dihindari dan harus dialami selama tahap awal perkembangan bidang teknologi, ada juga saran untuk optimasi struktur dengan pendekatan yang disatukan dan sistematis.

Dalam pendekatan sistematis yang digambarkan oleh Morris dan kawan-kawan pada tahun 1982, algoritma dibagi dalam tiga bagian, tiap bagian mempunyai fungsi yang terdefinisi secara baik, dan program komputer untuk tiap bagian dapat dikembangkan dan diuji secara terpisah. Pendekatan ini memperbolehkan penggunaan program secara berdiri sendiri yang kemudian

dapat dikumpulkan dengan cepat untuk persoalan desain yang spesifik. Pendekatan ini dapat juga digunakan sebagai bahan perbandingan untuk perkembangan langkah-langkah optimasi yang baru. Di samping itu, pendekatan sistematis ini dapat dengan mudah diterapkan pada masalah yang lebih kompleks di bidang teknik yang lain (Wu, 1986, halaman 10).

Menurut Kirsch pada tahun 1981 biasanya dalam suatu perencanaan terdiri atas empat langkah yaitu :

1. Perumusan syarat-syarat fungsional, yaitu mencari dan merumuskan syarat-syarat fungsional yang dalam beberapa kasus tidak terlihat secara nyata.
2. Perencanaan dasar, misalnya pemilihan topologi, tipe struktur dan material.
3. Proses optimasi, yaitu untuk memperoleh kemungkinan perencanaan terbaik dengan kriteria, pertimbangan dan batas-batas yang ada.
4. Pendetailan, setelah seluruh penyajian optimasi, hasil yang didapat harus diperiksa dan dimodifikasi jika perlu.

Berdasarkan berbagai kemajuan ilmu dan teknologi, perancangan struktur bangunan harus direncanakan secara optimal yaitu struktur yang paling ekonomis serta memenuhi segala persyaratan yang diinginkan. Oleh karena itu perlu dikembangkan suatu sistem yang mampu menangani berbagai masalah optimasi.

Secara Umum masalah optimasi ada empat jenis, yaitu :

1. Optimasi bentuk.
2. Optimasi topologi.
3. Optimasi geometri.
4. Optimasi ukuran penampang.

Dalam metoda optimasi terdapat tiga besaran utama, yaitu:

1. Variabel desain.

Besaran yang tidak berubah nilainya disebut parameter tetap, sedangkan yang nilai berubah selama proses optimasi disebut variabel desain. Variabel desain merupakan variabel yang dicari dalam masalah optimasi. Contohnya adalah ukuran komponen struktur dan geometri struktur. Data variabel desain ada dua macam, yaitu data diskrit dan data kontinu. Dalam beberapa kasus, khususnya optimasi bentuk dan geometri, variabel desain lebih sesuai dinyatakan sebagai variabel desain kontinu dibandingkan variabel diskrit.

2. Fungsi kendala.

Fungsi kendala merupakan suatu fungsi yang memberikan batasan daerah layak dan daerah tak layak. Dalam bidang teknik terdapat dua macam kendala yaitu :

- a Kendala rencana, yaitu kendala yang menentukan variabel desain selain yang memberikan batasan berdasarkan sifat. Kendala ini biasanya dapat dilihat secara nyata, misalnya batasan karena masalah fungsional, fabrikasi atau keindahan. Contoh kendala rencana adalah ketebalan plat, kemiringan atap.
- b Kendala sifat, yaitu kendala yang didapat dari persyaratan sifat. Biasanya kendala ini tidak dapat terlihat secara nyata karena berhubungan dengan analisis struktur. Contoh kendala sifat adalah batas tegangan maksimum, perpindahan (*displacements*) yang diijinkan, kekuatan tekuk.

3. Fungsi sasaran

Fungsi sasaran adalah suatu fungsi yang mengandung kriteria dari struktur yang diinginkan, misalnya struktur dengan berat paling ringan, dengan harga termurah, paling aman atau paling efisien. Pemilihan fungsi sasaran merupakan hal yang terpenting dalam proses optimasi agar dapat mencapai sasaran yang sebenarnya sedekat mungkin. Dalam beberapa situasi fungsi sasaran dapat terlihat jelas. Misalnya jika ingin mencari harga yang termurah maka fungsi sasarannya dapat diasumsikan ke dalam berat strukturnya. Namun terkadang sulit juga untuk menentukan harga yang sebenarnya dari sebuah konstruksi, misalnya struktur dengan berat paling ringan belum tentu yang termurah, karena biasanya masalah harga minimum akan termasuk juga harga bahan, fabrikasi, transportasi dan lain-lain. Masalah dalam optimasi dapat dibedakan menjadi dua, yaitu :

- a Masalah optimasi linier merupakan dasar dari metoda optimasi secara matematis. Dalam masalah ini fungsi kendala dan fungsi sasarannya semuanya dinyatakan dalam fungsi linier. Fungsi kendala dapat berupa persamaan maupun pertidaksamaan, dan fungsi sasarannya berupa meminimumkan dan memaksimumkan.
- b Masalah optimasi tak linier, yaitu bila fungsi kendala dan fungsi sasarannya tak linier.

Masalah optimasi dalam bidang teknik, pada umumnya berupa masalah optimasi tak linier. Masalah yang tak linier ini juga dapat dilinierkan, tetapi memberikan hasil yang kurang akurat ditinjau dari segi teknik. Oleh karena itu

terpaksa diselesaikan memakai program tak linier yang lebih sulit dipelajari dibandingkan program linier, karena memerlukan matematika yang kompleks.

Penyelesaian masalah optimasi dapat dipakai dua cara yaitu :

1. Metoda analisa

Metoda ini menggunakan dasar teori matematika yang dibuat oleh Maxwell pada tahun 1890 dan Michell tahun pada 1904 dan memberikan hasil eksak namun hanya dapat digunakan untuk masalah optimasi yang sederhana saja karena pada beberapa masalah yang lebih kompleks pengolahan matematikanya sangat tidak sederhana (Wibowo, 1996, halaman 2).

2. Metoda numerik

Metoda optimasi numerik berkembang sejak ditemukannya komputer sebagai alat bantu hitung. *Dynamic programming, integer programming, steepest descent, sequential unconstraint minimization technique, gradient projection, dan penalty function* merupakan metoda optimasi numerik yang sering dipakai untuk menyelesaikan masalah optimasi di bidang sipil (Wibowo, 1996, halaman 3). Dalam metoda ini nilai yang akan dicari didekati dengan cara iterasi dan proses iterasi dihentikan apabila nilai yang dicari sudah cukup dekat dengan titik optimal yang sesungguhnya (Kirsch, 1981, halaman 5).

Metoda-metoda tersebut di atas mempunyai kelemahan, yaitu mempunyai peluang relatif besar untuk konvergen ke titik optimum lokal, bila dipakai untuk menyelesaikan masalah optimasi dengan jumlah titik optimum lebih dari satu. Hal ini dapat dimaklumi, karena penurunan perumusannya berdasarkan asumsi fungsi konveks. Selain itu juga memerlukan analisis struktur yang banyak, keharusan

mengikutsertakan semua kendala dalam model matematikanya dan modifikasi variabel yang kurang efisien (Wibowo, 1996, halaman 3).

II.3.2 Metoda Optimasi Polihedron Fleksibel

Metoda polihedron fleksibel merupakan metoda pengembangan dari metoda simplex, yang dikembangkan oleh Nelder-Mead (Haftka, 1991, halaman 64), dasar pemikiran metoda simplex adalah menurunkan nilai fungsi sasaran secara kontinu dimulai dari suatu nilai fungsi awal sampai mencapai nilai fungsi minimum terpenuhi (Haftka, 1991, halaman 64).

Metoda ini bermanfaat untuk mencari harga-harga *extrem* suatu fungsi dengan banyak variabel, dengan turunan dari fungsi tersebut sulit untuk dicari. Metoda polihedron fleksibel menggunakan pencerminan (*reflection*), ekspansi (*expansion*) dan penyusutan (*contraction*) dalam melakukan penelusuran.

Polihedron Fleksibel menggunakan banyak titik coba, dengan jumlah titik coba N_{titik} minimum sama dengan jumlah variabel desain JVD ditambah satu. $N_{titik} = JVD + 1$ dipakai oleh Harb dan Matthews pada tahun 1987. Box mengusulkan $N_{titik} = 2 * JVD$, sedangkan $N_{titik} = JVD + 2$ diusulkan oleh Biles pada tahun 1983. Jumlah titik coba makin banyak dimaksudkan untuk mengurangi terjadinya konvergen prematur, tetapi memperlambat konvergensi.

Masalah meminimumkan fungsi sasaran $f = f(x,y)$, dengan dua variabel desain x dan y , dan memakai jumlah titik coba $N_{titik} = JVD + 1 = 3$, prosedurnya adalah sebagai berikut :

1. Tentukan atau acak tiga buah titik di dalam ruang variabel disain, kemudian hitung nilai fitnessnya. Titik terbaik disebut titik B (*best*) dengan koordinat

(X_B, Y_B) , titik terjelek disebut titik W (*worst*) dengan koordinat (X_W, Y_W) , sedang titik lainnya disebut titik G (*good*) dengan koordinat (X_G, Y_G)

2. Hitung titik pusat M dengan koordinat (X_M, Y_M) yang didapat dengan merata – rata titik coba selain titik W, sehingga :

$$X_M = 0.5 * (X_B + X_G) \quad (2-26)$$

$$Y_M = 0.5 * (Y_B + Y_G) \quad (2-27)$$

3. Tentukan arah penelusuran, yaitu garis yang menghubungkan titik W dan titik M dan dinyatakan sebagai :

$$X_S = X_M - X_W \quad (2-28)$$

$$Y_S = Y_M - Y_W \quad (2-29)$$

4. Cari titik coba baru dalam arah S (*search*) yang memberikan nilai fitness lebih baik daripada fitness titik W, titik coba ini tidak boleh identik dengan titik M. Titik coba baru T (*try*) dengan koordinat X_T, Y_T adalah :

$$X_T = X_W + \lambda \cdot X_S \quad (2-30)$$

$$Y_T = Y_W + \lambda \cdot Y_S \quad (2-31)$$

λ adalah koefisien refleksi. Kalau tidak ditemukan titik coba baru yang lebih baik dari titik W maka dilakukan penyusutan menuju B, besarnya penyusutan sama dengan setengah jaraknya terhadap titik b, sehingga titik B baru adalah :

$$X_{W \text{ baru}} = 0.5 * (X_W + X_B) \quad (2-32)$$

$$Y_{W \text{ baru}} = 0.5 * (Y_W + Y_B) \quad (2-33)$$

dan titik G baru adalah

$$X_{G \text{ baru}} = 0.5 * (X_W + X_G) \quad (2-34)$$

$$Y_{G \text{ baru}} = 0.5 * (Y_W + Y_G) \quad (2-35)$$

Kemudian diperiksa apakah sudah konvergen atau belum, kalau sudah maka berhenti kalau belum ulangi ke langkah dua lagi.

Pemeriksaan konvergensi dapat menggunakan persamaan-persamaan yang ada di bawah ini :

$$|X_B - X_G| \leq \varepsilon \quad (2-36a)$$

$$|X_B - X_W| \leq \varepsilon \quad (2-36b)$$

$$|X_W - X_G| \leq \varepsilon \quad (2-36c)$$

$$|Y_B - Y_G| \leq \varepsilon \quad (2-36d)$$

$$|Y_B - Y_W| \leq \varepsilon \quad (2-36e)$$

$$|Y_W - Y_G| \leq \varepsilon \quad (2-36f)$$

dengan ε adalah suatu nilai konvergensi dan diambil sekecil mungkin, misalnya 0.0003.

Secara umum metoda ini membuat sebuah segi banyak dalam ruang variabelnya yang terus diiterasi sehingga segi banyak itu makin lama makin mengecil, dan akan didapatkan hasil yang optimum begitu segi banyak itu menjadi sangat kecil sekali, yang ditentukan nilainya sebagai suatu nilai konvergensi.

BAB III

OPTIMASI BETON BERTULANG PADA STRUKTUR PORTAL RUANG

III.1 Struktur Program

Program optimasi beton bertulang pada struktur portal ruang ini ditulis dengan menggunakan bahasa pemrograman C++. Kode Program ini ditulis pada beberapa file yaitu file utama yang berekstensi cpp, dan file pembantu berekstensi hpp. Di dalam file yang berekstensi cpp terdapat suatu fungsi bernama `main()`, fungsi ini akan dijalankan pertama kali oleh program, selanjutnya dari dalam fungsi ini akan dipanggil fungsi-fungsi lainnya sehingga membentuk suatu program yang utuh.

Struktur program optimasi beton bertulang pada struktur portal ruang ini merupakan gabungan dari pemrograman terstruktur dan pemrograman berorientasi obyek.

Pemrograman terstruktur memiliki kelebihan dalam pengorganisasian dalam membuat kode-kode program sehingga mudah dibaca dan mudah untuk dipahami, gagasan pemrograman terstruktur pertama kali dikemukakan oleh Profesor Edsger W. Dijkstra dari University of Eindhoven Nederland pada tahun 1965 (Sutedjo dan Michael, 1997, halaman 1). Dalam tulisannya, beliau menyatakan bahwa pernyataan GOTO seharusnya jangan dipergunakan, namun pernyataan ini ditanggapi oleh HD. Mills yang beranggapan bahwa pemrograman terstruktur tidak hanya dihubungkan dengan pernyataan GOTO saja, tetapi pada

struktur program itu sendiri yang menentukan apakah program tersebut terstruktur atau tidak, baik menggunakan pernyataan GOTO maupun tidak.

Pemrograman berorientasi obyek diciptakan karena masih dirasakan adanya keterbatasan pada bahasa pemrograman tradisional, konsep pemrograman berorientasi obyek yaitu membagi masalah-masalah ke dalam obyek, sehingga data dan fungsi-fungsi yang akan dioperasikan digabung menjadi satu kesatuan disebut pengkapsulan (*encapsulated*).

Dalam bidang ilmu teknik, pemrograman terstruktur lebih banyak digunakan karena rumitnya masalah yang dikerjakan dan memerlukan kemudahan pemeriksaan (*debuging*), dalam program ini struktur program utama menggunakan konsep pemrograman terstruktur, sedangkan konsep pemrograman berorientasi obyek digunakan pada analisis balok dan analisis kolom. Kelas balok dan kelas kolom akan dibangkitkan apabila diperlukan analisa balok atau analisa kolom dan apabila tidak diperlukan lagi kelas-kelas itu dihancurkan, sehingga penggunaan kelas ini dapat menghemat memori komputer.

File yang terdapat pada program optimasi beton bertulang pada struktur portal ruang adalah:

- a. ORCHISF.CPP merupakan file yang memuat fungsi main().
- b. BORLANDC.HPP file yang memuat *header* pustaka yang dibuat oleh Borland.
- c. HEADER.HPP berisi kumpulan *header*.
- d. PROTO.HPP berisi prototipe program.
- e. VARIABEL.HPP berisi deklarasi variabel global.

- f. INOUT.HPP berisi kumpulan sub program penanganan masalah masukan dan keluaran.
- g. STRUKTUR.HPP berisi kumpulan sub program analisa struktur dengan metoda kekakuan.
- h. SOLVER.HPP berisi sub program pembantu penanganan matrik.
- i. KOLOM.HPP berisi kelas kolom untuk analisa kolom biaksial bertulangan simetri.
- j. BALOK.HPP berisi kelas balok untuk analisa balok persegi bertulangan rangkap.
- k. ELEMEN.HPP berisi kumpulan sub program yang menangani elemen pada balok dan kolom.
- l. POLYHEDRON.HPP berisi sub program untuk optimasi dengan menggunakan metoda polihedron fleksibel.
- m. PENORMALAN.HPP berisi kumpulan sub program yang menangani perubahan variabel balok dan kolom menjadi variabel normal.
- n. PENGACAKAN.HPP berisi sub program yang mengacak nilai variabel.
- o. DISKRITISASI.HPP berisi kumpulan sub program yang merubah nilai *continue* menjadi diskrit.
- p. KENDALA.HPP berisi sub program untuk menghitung besarnya kendala yang terjadi pada struktur.
- q. TELUSUR.HPP berisi sub program untuk melakukan penelusuran variabel.
- r. BARU.HPP berisi sub program untuk menentukan struktur baru dalam metoda optimasi.

- s. PENGURUTAN.HPP berisi sub program untuk mengurutkan fitness struktur.
- t. TAMPILAN.HPP berisi sub program yang menangani masalah tampilan.
- u. CETAK.HPP berisi sub program yang menangani masalah pencetakan hasil akhir.

III.2 Sub Program Masukkan dan Keluaran

III.2.1 Sub Program Untuk Menampilkan Menu Utama

Pada saat program dijalankan pertama kali akan muncul sebuah menu yang berisi pilihan untuk menjalankan program, menu utama program optimasi beton bertulang pada struktur portal ruang seperti diperlihatkan pada gambar 3-1.

```

*****
      Program Optimasi Beton Bertulang Pada Struktur Portal Ruang
            Oleh : Yohan Naftali
                  7712/TS
*****
Time : 23:09:48.10

1. Input data awal ke file
2. Input data beban ke file
3. Melihat isi file input
4. Mengoptimasi Struktur
5. Keluar
Pilihan (1-5) = _

```

Gambar 3-1 Menu Utama Program

Pilihan yang terdapat pada menu utama terdiri dari :

1. Input data awal ke file, untuk memasukkan data tentang informasi struktur.
2. Input data beban ke file, untuk memasukkan data beban yang bekerja pada struktur.
3. Melihat isi file input, untuk menampilkan data yang telah dimasukkan.
4. Mengoptimasi struktur, untuk memulai proses optimasi beton bertulang pada struktur portal ruang.

5. Keluar, untuk menghentikan program.

Fungsi-fungsi yang digunakan untuk menampilkan menu utama yang terdapat pada file TAMPILAN.HPP terdiri dari :

1. **void** menu_utama (), fungsi ini dipanggil dari fungsi **void** main (), selanjutnya dari fungsi ini dipanggil fungsi-fungsi lainnya.
2. **void** about (), fungsi ini dipanggil untuk menampilkan informasi program pada layar.

III.2.2 Sub Program Memasukkan Data

Untuk melakukan proses pemasukkan data (*input*), dapat dilakukan dengan dua cara yaitu :

1. Mengikuti panduan yang disediakan oleh program.
2. Mengedit teks ASCII dengan bantuan MS-DOS Editor atau Notepad.

Memasukkan data melalui panduan program, maka pertama kali akan diminta nama file generik (tanpa ekstensi), nama file ini digunakan program untuk memberi nama file-file bentukan lainnya. Selanjutnya program meminta data-data struktur, mengedit teks ASCII dengan bantuan ASCII editor lebih mudah digunakan untuk mengedit data masukkan yang telah ada.

Sebelum memulai pemasukkan data ke komputer, maka data-data struktur harus kita persiapkan terlebih dahulu supaya mempermudah proses pemasukkan data, data-data struktur yang perlu kita persiapkan adalah :

1. Nama struktur
2. Jumlah batang pada struktur
3. Jumlah titik kumpul

4. Jumlah pengeang tumpuan pada struktur
5. Jumlah titik kumpul yang dikekang
6. Koordinat titik kumpul
7. Alokasi elemen batang pada struktur
8. Pengeang titik kumpul
9. Kuat desak beton karakteristik
10. Kuat tarik tulangan utama
11. Kuat tarik tulangan sengkang
12. Beban yang mengenai struktur
13. Data lebar dan tinggi balok
14. Data sisi penampang kolom
15. Data diameter tulangan utama
16. Data diameter tulangan sengkang
17. Data jumlah tulangan
18. Data jarak antara tulangan sengkang

Setelah data kita masukkan ke dalam komputer, selanjutnya diperiksa lagi data yang telah kita masukkan untuk memastikan bahwa data yang telah dimasukkan sudah benar, Apabila ada kesalahan, maka data masukkan dapat dikoreksi dengan mengedit file masukkan.

Struktur file masukkan pada program optimasi beton bertulang pada struktur rangka ini adalah :

1. Data umum struktur terdapat pada file generik.inp
2. Data beban yang bekerja pada struktur terdapat pada file generik.bbn

3. Data diskrit mengenai ukuran penampang batang terdapat pada file generik.isd
4. Data diskrit mengenai diameter tulangan utama terdapat pada file generik.idl
5. Data diskrit mengenai diameter tulangan sengkang terdapat pada file generik.ids
6. Data diskrit mengenai jumlah tulangan utama pada salah satu penampang terdapat pada file generik.ijl
7. Data diskrit mengenai jarak antara tulangan sengkang terdapat pada file generik.ijs

Fungsi-fungsi yang digunakan dalam proses masukan adalah :

1. **void** input_data (), terdapat pada file INOUT.HPP, untuk menampilkan menu masukan
2. **void** input_data_umum (), terdapat pada file INOUT.HPP, untuk memasukkan data umum struktur ke file
3. **void** input_data_diskrit (), terdapat pada file INOUT.HPP, untuk memasukkan data elemen batang
4. **void** load_data (), terdapat pada file PEMBEBANAN.HPP, untuk memasukkan data beban yang bekerja pada struktur

III.2.3 Sub Program Untuk Membaca Data Masukan

Setelah data masukan diproses oleh komputer, dan diletakkan ke dalam file-file masukan, maka untuk dapat dilakukan proses optimasi maka data masukan tersebut perlu dibaca ulang, fungsi-fungsi yang digunakan untuk melakukan proses ini adalah :

1. **void** baca_data (), terdapat pada file INOUT.HPP, untuk membaca data

masukkan

2. **void** baca_beban (), terdapat pada file PEMBEBANAN.HPP, untuk membaca data pembebanan.

III.3 Sub Program Inti

III.3.1 Sub Program Analisa Struktur

Untuk menganalisa struktur portal ruang digunakan metoda kekakuan yang dikembangkan oleh Weaver dan Gere (1980). Secara umum proses dalam analisa struktur adalah sebagai berikut :

1. Pembentukan matrik rotasi
2. Pembentukan matrik kekakuan batang
3. Pembentukan vektor beban
4. Perhitungan perpindahan pada titik kumpul
5. Perhitungan gaya dan reaksi pada struktur

Sub program analisa struktur terdapat pada file STRUKTUR.HPP, PEMBEBANAN.HPP, dan SOLVER.HPP. File STRUKTUR.HPP berisi fungsi-fungsi untuk membentuk matrik rotasi, pembentukan matrik kekakuan batang, perhitungan perpindahan, dan perhitungan gaya dan reaksi pada struktur. Fungsi untuk membentuk vektor beban terdapat pada file PEMBEBANAN.HPP, sedangkan file SOLVER.HPP terdapat fungsi **void** banfac (), yang digunakan untuk melakukan faktorisasi matrik simetris berjalur dengan pendekatan Choleski yang dimodifikasi, dan terdapat pula fungsi **void** bansol (), yang digunakan untuk mengolah matrik berjalur.

III.3.2 Sub Program Analisa Balok Persegi Panjang

Sub program analisa balok persegi panjang terdapat pada file BALOK.HPP. Dalam file BALOK.HPP terdapat sebuah kelas yang bernama **class** balok. Kelas ini dibangkitkan untuk menghitung kendala yang terdapat pada setiap balok pada struktur.

Persamaan perhitungan kendala-kendala balok yang terdapat pada kelas balok adalah :

1. Kendala rasio tulangan maksimum

$$kendala_rho_b = \left(\frac{RHO}{RHB} \right) - 1 \quad (3-1)$$

pada persamaan di atas RHO adalah rasio penulangan, kendala_rho_b adalah kendala rasio tulangan maksimum yang dibatasi oleh RHB yaitu rasio penulangan pada keadaan luluh dikalikan 0,75

2. Kendala rasio penulangan minimum

$$kendala_rho_m = \left(\frac{RMIN}{RHO} \right) - 1 \quad (3-2)$$

pada persamaan di atas kendala_rho_m adalah kendala rasio penulangan minimum, RMIN adalah rasio penulangan minimum yang didapat dari persamaan.

$$RMIN = \frac{1,4}{fy} \quad (3-3)$$

3. Kendala momen lentur

$$kendala_M = \left(\frac{MU}{FMU} \right) - 1 \quad (3-4)$$

pada Persamaan di atas kendala_M merupakan kendala momen lentur, MU adalah momen yang bekerja pada struktur, FMU adalah momen yang dapat ditahan oleh balok.

4. Kendala lendutan

$$kendala_lendutan = \left(\frac{LENDUTAN}{LENDUTAN_IJIN} \right) - 1 \quad (3-5)$$

pada persamaan di atas kendala_lendutan adalah kendala lendutan, LENDUTAN adalah lendutan yang terjadi, LENDUTAN_IJIN adalah lendutan ijin maksimum.

5. Kendala tulangan geser

$$kendala_sb = \left(\frac{Jarak_S}{SmakS} \right) - 1 \quad (3-6)$$

pada persamaan di atas kendala_sb adalah kendala tulangan geser, Jarak_S adalah jarak sengkang, SmakS adalah jarak antar sengkang maksimum.

III.3.3 Sub Program Analisa Kolom Biaksial

Sub program analisa kolom biaksial terdapat pada file KOLOM.HPP. Dalam file KOLOM.HPP terdapat sebuah kelas yang bernama **class** kolom. Kelas ini dibangkitkan untuk menghitung kendala yang terdapat pada setiap kolom pada struktur.

Persamaan perhitungan kendala-kendala kolom yang terdapat pada kelas kolom adalah :

1. Kendala rasio tulangan maksimum

$$kendala_r_mak = \left(\frac{RHO}{0,08} \right) - 1 \quad (3-7)$$

pada persamaan di atas RHO adalah rasio penulangan, $kendala_r_mak$ adalah kendala rasio tulangan maksimum yang dibatasi 0,08 (8 %).

2. Kendala rasio penulangan minimum

$$kendala_r_min = \left(\frac{0,01}{RHO} \right) - 1 \quad (3-8)$$

pada persamaan di atas $kendala_rho_m$ adalah kendala rasio penulangan minimum yang dibatasi sebesar 0,01 (1 %).

3. Kendala gaya aksial minimum

$$kendala_po = \left(\frac{PN}{PO} \right) - 1 \quad (3-9)$$

pada persamaan di atas $kendala_po$ merupakan kendala gaya aksial minimum atau disebut juga kendala eksentrisitas minimum, PN merupakan gaya aksial nominal dan PO merupakan batas gaya aksial yang dapat dipikul oleh kolom dikalikan faktor reduksi kekuatan untuk kolom berpengikat sengkang yaitu 0,8

4. Kendala gaya aksial

$$kendala_pn = \left(\frac{PN}{PNcoba} \right) - 1 \quad (3-10)$$

pada persamaan di atas $kendala_pn$ adalah kendala gaya aksial nominal, PN merupakan gaya aksial nominal yang terjadi pada struktur, PNcoba adalah gaya aksial yang mampu didukung oleh kolom, PNoba dicari bersama-sama MNcoba berdasarkan eksentrisitas yang terjadi pada kolom dengan metoda *false posisi* untuk menentukan letak garis netral.

5. Kendala momen ekivalen biaksial

Untuk $MNX > MNY$ maka digunakan persamaan :

$$kendala_{mn} = \left(\frac{MOX}{MNcoba} \right) - 1 \quad (3-11a)$$

untuk $MNX \leq MNY$ maka digunakan persamaan :

$$kendala_{mn} = \left(\frac{MOY}{MNcoba} \right) - 1 \quad (3-11b)$$

pada persamaan di atas $kendala_{mn}$ merupakan kendala momen ekivalen biaksial, MOX merupakan momen ekivalen biaksial pada sumbu X, MOY merupakan momen ekivalen biaksial pada sumbu Y, $MNcoba$ merupakan momen nominal pada arah sumbu biaksial yang ditinjau.

6. Kendala jarak tulangan minimum

$$kendala_{tul} = \left(\frac{jarak_min}{jarak_antar_tulangan} \right) - 1 \quad (3-12)$$

pada persamaan di atas $kendala_{tul}$ adalah kendala jarak antara tulangan minimum, $jarak_min$ adalah jarak minimum yang diijinkan, $jarak_antar_tulangan$ adalah jarak antara tulangan utama.

7. Kendala kelangsingan kolom

$$kendala_{kelangsingan} = \left(\frac{rasio_kelang_sin\ gan}{22} \right) - 1 \quad (3-13)$$

Pada persamaan di atas $kendala_{kelangsingan}$ merupakan kendala kelangsingan pada kolom, $rasio_kelangsingan$ merupakan rasio kelangsingan pada kolom, yang dibatasi minimal 22.

III.3.4 Sub Program Metoda Optimasi Polihedron Fleksibel

Untuk melakukan proses optimasi digunakan metoda polihedron fleksibel, prosedur umum yang digunakan dalam program optimasi beton bertulang pada struktur portal ruang dengan metoda polihedron fleksibel adalah :

1. Penentuan variabel untuk struktur generasi pertama, yang dilakukan dengan pengacakan, dalam program optimasi beton bertulang pada struktur portal ruang ini penentuan variabel terdapat pada file PENGACAKAN.HPP dalam melakukan pengacakan, sebuah struktur dipilih supaya menggunakan elemen yang paling besar nilainya supaya dalam salah satu struktur tersebut ada yang tidak memiliki kendala, hal ini dimaksudkan supaya dalam proses optimasi tidak terjebak kepada optimal lokal yang berkendala.
2. Pembangkitan generasi pertama, sesudah seluruh struktur dibangkitkan, nilai fitness, kendala, harga, dan variabel desain disimpan dalam *array*.
3. Seluruh struktur pada generasi pertama diurutkan berdasarkan fitnessnya, pengurutan ini terdapat pada file PENGURUTAN.HPP, metoda pengurutan yang digunakan adalah metoda *Bubble Sort*, metoda ini berasal dari sifat gelembung air yang semakin ringan akan menuju ke permukaan air.
4. Penentuan arah baru untuk melakukan penelusuran, terdapat pada file TELUSUR.HPP.
5. Menggantikan struktur terjelek dengan struktur baru hasil penelusuran yang memiliki fitness lebih baik, apabila dalam penelusuran tidak ditemukan struktur yang lebih baik dari pada struktur terjelek, maka dilakukan penyusutan menuju struktur terbaik.

6. Selanjutnya dilakukan lagi pengurutan fitness, proses ini dilakukan berulang-ulang sampai pada suatu batas konvergensi.

Formulasi untuk fungsi penalti yang digunakan dalam program optimasi beton bertulang pada struktur portal ruang adalah *exterior penalty function*, proses penelusuran dimulai dari daerah tidak layak, yang kemudian secara bertahap menuju daerah layak. Formulasi fungsi penalti untuk menyelesaikan masalah meminimumkan $f(x)$ yang memenuhi kendala $g_j(x) \leq 0$, $j = 1, 2, 3, \dots, n_g$ adalah

$$\phi = \frac{r}{\left(f(x) + r \sum_{j=1}^{n_g} g_j(x) \right)} \quad (3-14)$$

dalam persamaan di atas ϕ merupakan fungsi penalti, $f(x)$ merupakan fungsi sasaran yang berupa harga struktur yang formulasinya terdapat pada persamaan (1-12), r parameter penalti, n_g jumlah kendala, dan $g_j(x)$ persamaan kendala yang telah dihitung dalam kelas balok dan kelas kolom.

Nilai r harus cukup besar agar mampu mengarahkan proses penelusuran dekat dengan daerah layak, tetapi cukup kecil agar proses meminimuman tidak mengalami kesulitan.

Dalam program optimasi beton bertulang pada struktur portal ruang konvergensi ditentukan sebagai berikut :

1. Jumlah iterasi melebihi batas yang telah ditentukan
2. Terjadi penyusutan berturut yang menghasilkan fitness yang sama sebanyak jumlah struktur desain

3. Terdapat struktur berjumlah sama dengan jumlah variabel desain yang memiliki nilai fitness sama.

III.3.5 Sub Program Mencetak Hasil Akhir Optimasi

Setelah suatu proses optimasi selesai, hasil optimasi berupa nilai-nilai variabel desain terbaik, harga, kendala dan hasil perhitungan struktur dicetak, sub program untuk mencetak hasil akhir tersebut ke dalam file adalah **void cetak_akhir()** yang terdapat pada file CETAK.HPP.

File-file hasil keluaran yang dicetak oleh program optimasi beton bertulang pada struktur portal ruang adalah :

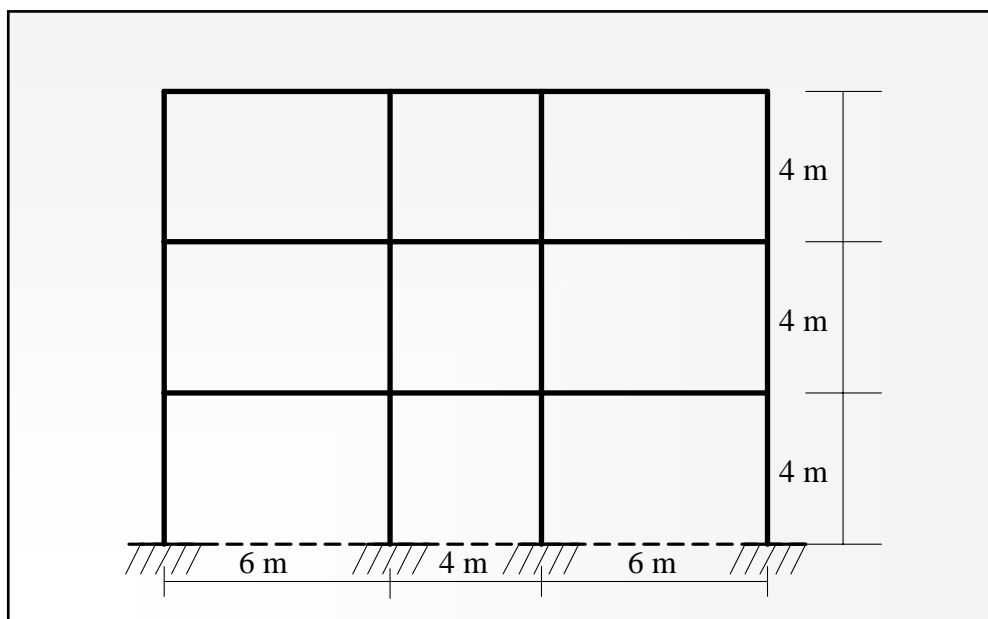
1. generik.opt berisi dimensi balok dan kolom pada struktur yang paling optimal.
2. generik.str berisi hasil perhitungan gaya batang dengan metoda kekakuan.
3. generik.kdl berisi kendala yang terdapat pada struktur.
4. generik.inf berisi informasi data masukan.
5. generik.his berisi riwayat optimasi.

BAB IV

VALIDASI DAN APLIKASI PROGRAM OPTIMASI

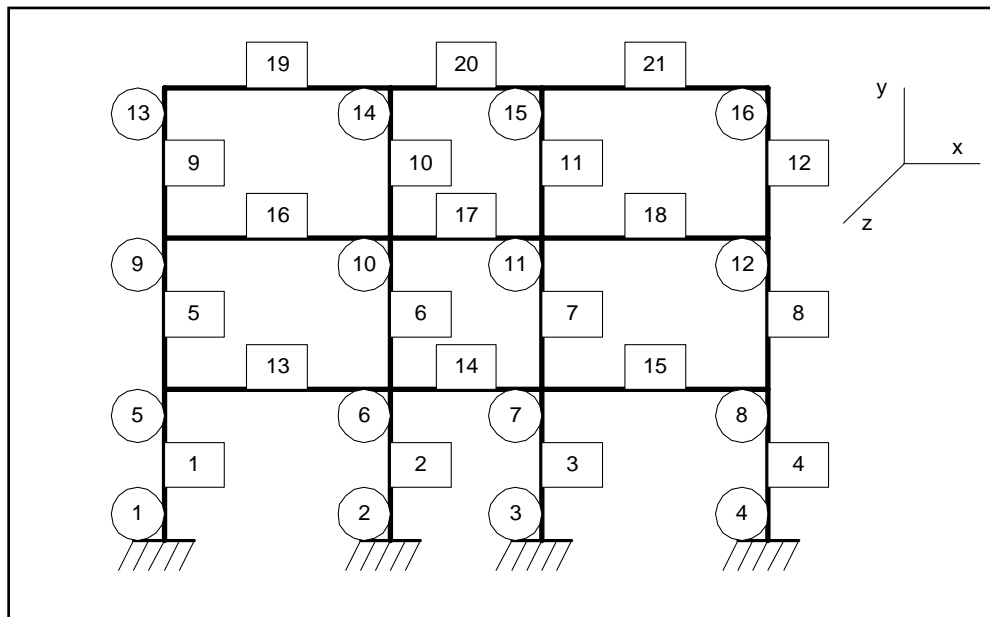
IV.1 Validasi Program Optimasi

Dalam melakukan validasi program optimasi, digunakan hasil perhitungan beton bertulang pada struktur portal yang dikerjakan oleh Ir. Harsoyo dalam bukunya berjudul gedung bertingkat II (Harsoyo, 1976), seri bina bangunan konstruksi beton bertulang. Geometri bangunan yang dimodelkan sebagai struktur portal bidang digambarkan dalam gambar 4-1.



Gambar 4-1 Geometri Struktur Portal Bidang

Langkah pertama dalam analisa struktur setelah memodelkan struktur adalah melakukan penomoran titik dan batang, dalam kasus ini terdapat 16 titik kumpul dan 21 batang yang terdiri dari 12 kolom dan 9 balok. Penomoran titik dan batang dapat dilihat dalam gambar 4-2.



Gambar 4-2 Penomoran Titik Kumpul dan Batang Portal Bidang

Beban yang bekerja pada struktur portal bidang tersebut berupa beban merata pada balok dan beban akibat tekanan angin pada titik 5, 9, 13 ke arah $-x$, besarnya beban merata pada batang diperlihatkan dalam tabel 4-1. Sedangkan besarnya beban akibat tekanan angin diperlihatkan dalam tabel 4-2.

Nomor Batang	Beban Merata Total (t/m)	Berat Sendiri (t/m)	Beban Luar (t/m)
13	4,100	0,360	3,740
14	3,491	0,360	3,131
15	4,100	0,360	3,740
16	4,100	0,360	3,740
17	3,491	0,360	3,131
18	4,100	0,360	3,740
19	1,762	0,300	1,462
20	1,447	0,300	1,147
21	1,762	0,300	1,462

Tabel 4-1 Beban Merata Pada Struktur Portal Bidang

Nomor titik	Beban tekanan angin (kg)
5	400
9	800
13	800

Tabel 4-2 Beban Tekanan Angin

Dalam tugas akhir ini struktur portal bidang pada gambar 4-1 dioptimasi dengan program optimasi beton bertulang pada struktur portal ruang, dan hasil optimasinya dibandingkan dengan hasil yang telah dikerjakan oleh Ir. Harsoyo D. (1976), semua data masukkan dapat dilihat dalam lampiran 24, dengan menggunakan mutu beton 17,5 MPa dan mutu baja tulangan 240 MPa, tekuk pada kolom dianalisa. Perbandingan hasil akhir variabel desain antara hasil yang dikerjakan oleh Ir. Harsoyo dan hasil optimasi program optimasi beton bertulang pada struktur portal ruang disajikan pada tabel 4-3.

Batang		Harsoyo (1976)		Naftali (1999)	
No.	Variabel Desain				
1	Lebar	0,30 m	0,9 m ³	0,41 m	0,6724 m ³
	Tinggi	0,75 m		0,41 m	
	Tulangan Utama	10 ϕ 25 / 4 ϕ 16	187,929 kg	8 ϕ 25	136,8158 kg
	Tulangan Sengkang	ϕ 8-150		ϕ 12-300	
2	Lebar	0,30 m	0,9 m ³	0,41 m	0,6724 m ³
	Tinggi	0,75 m		0,41 m	
	Tulangan Utama	10 ϕ 25 / 4 ϕ 16	187,929 kg	8 ϕ 22	109,0117 kg
	Tulangan Sengkang	ϕ 8-150		ϕ 12-300	
3	Lebar	0,30 m	0,9 m ³	0,40 m	0,64 m ³
	Tinggi	0,75 m		0,40 m	
	Tulangan Utama	10 ϕ 25 / 4 ϕ 16	187,929 kg	8 ϕ 22	101,2778 kg
	Tulangan Sengkang	ϕ 8-150		ϕ 8-300	
4	Lebar	0,30 m	0,9 m ³	0,43 m	0,7396 m ³
	Tinggi	0,75 m		0,43 m	
	Tulangan Utama	10 ϕ 25 / 4 ϕ 16	187,929 kg	8 ϕ 19	85,6326 kg
	Tulangan Sengkang	ϕ 8-150		ϕ 12-300	
5	Lebar	0,30 m	0,72 m ³	0,44 m	0,7744 m ³
	Tinggi	0,60 m		0,44 m	
	Tulangan Utama	10 ϕ 22 / 4 ϕ 16	137,3734 kg	8 ϕ 20	93,7609 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 12-300	

6	Lebar	0,30 m	0,72 m ³	0,44 m	0,7744 m ³
	Tinggi	0,60 m		0,44 m	
	Tulangan Utama	8 ϕ 22 / 4 ϕ 16	113,0698 kg	8 ϕ 19	77,8014 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 8-300	
7	Lebar	0,30 m	0,72 m ³	0,42 m	0,7056 m ³
	Tinggi	0,60 m		0,42 m	
	Tulangan Utama	8 ϕ 22 / 4 ϕ 16	113,0698 kg	8 ϕ 19	85,1948 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 12-300	
8	Lebar	0,30 m	0,72 m ³	0,42 m	0,7056 m ³
	Tinggi	0,60 m		0,42 m	
	Tulangan Utama	10 ϕ 22 / 4 ϕ 16	137,3734 kg	8 ϕ 22	109,4494 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 12-300	
9	Lebar	0,30 m	0,72 m ³	0,41 m	0,6724 m ³
	Tinggi	0,60 m		0,41 m	
	Tulangan Utama	6 ϕ 22 / 4 ϕ 16	88,73483 kg	8 ϕ 20	92,4476 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 12-300	
10	Lebar	0,30 m	0,72 m ³	0,45 m	0,81 m ³
	Tinggi	0,60 m		0,45 m	
	Tulangan Utama	6 ϕ 22 / 4 ϕ 16	88,73483 kg	8 ϕ 29	86,5081 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 12-300	
11	Lebar	0,30 m	0,72 m ³	0,46 m	0,8464 m ³
	Tinggi	0,60 m		0,46 m	
	Tulangan Utama	6 ϕ 22 / 4 ϕ 16	88,73483 kg	8 ϕ 28	161,6033 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 8-300	
12	Lebar	0,30 m	0,72 m ³	0,44 m	0,7744 m ³
	Tinggi	0,60 m		0,44 m	
	Tulangan Utama	6 ϕ 22 / 4 ϕ 16	88,73483 kg	4 ϕ 25	76,5066 kg
	Tulangan Sengkang	ϕ 8-180		ϕ 12-300	
13	Lebar	0,25 m	0,975 m ³	0,33 m	1,1286 m ³
	Tinggi	0,65 m		0,57 m	
	Tulangan Utama	3 ϕ 25 / 3 ϕ 25 2 ϕ 25 / 4 ϕ 25 3 ϕ 25 / 3 ϕ 25	228,126 kg	2 ϕ 25 / 3 ϕ 28 3 ϕ 25 / 2 ϕ 22 2 ϕ 25 / 3 ϕ 28	212,9059 kg
	Tulangan Sengkang	ϕ 8-100		ϕ 12-130	
14	Lebar	0,25 m	0,65 m ³	0,33 m	0,6864 m ³
	Tinggi	0,65 m		0,52 m	
	Tulangan Utama	3 ϕ 25 / 3 ϕ 25 3 ϕ 25 / 2 ϕ 25 3 ϕ 25 / 3 ϕ 25	143,9139 kg	2 ϕ 22 / 3 ϕ 22 3 ϕ 25 / 3 ϕ 20 2 ϕ 22 / 3 ϕ 22	111,5333 kg
	Tulangan Sengkang	ϕ 8-100		12 - 150	
15	Lebar	0,25 m	0,975 m ³	0,34 m	1,122 m ³
	Tinggi	0,65 m		0,55 m	
	Tulangan Utama	3 ϕ 25 / 3 ϕ 25 2 ϕ 25 / 4 ϕ 25 3 ϕ 25 / 3 ϕ 25	228,126 kg	2 ϕ 22 / 3 ϕ 28 3 ϕ 22 / 2 ϕ 22 2 ϕ 22 / 3 ϕ 28	190,1989 kg
	Tulangan Sengkang	ϕ 8-100		ϕ 12-140	
16	Lebar	0,25 m	0,975 m ³	0,31 m	1,023 m ³
	Tinggi	0,65 m		0,55 m	
	Tulangan Utama	3 ϕ 25 / 3 ϕ 25 2 ϕ 25 / 4 ϕ 25 3 ϕ 25 / 3 ϕ 25	228,126 kg	2 ϕ 28 / 3 ϕ 28 3 ϕ 25 / 2 ϕ 28 2 ϕ 28 / 3 ϕ 28	225,0688 kg
	Tulangan Sengkang	ϕ 8-100		ϕ 12-130	

17	Lebar	0,25 m	0,65 m ³	0,33 m	0,6864 m ³
	Tinggi	0,65 m		0,52 m	
	Tulangan Utama	3φ25 / 3φ25 3φ25 / 2φ25 3φ25 / 3φ25	143,9139 kg	2φ25 / 3φ25 3φ20 / 2φ25 2φ25 / 3φ25	111,4915 kg
	Tulangan Sengkang	φ8-100		φ12-170	
18	Lebar	0,25 m	0,975 m ³	0,34 m	1,1016 m ³
	Tinggi	0,65 m		0,54 m	
	Tulangan Utama	3φ25 / 3φ25 2φ25 / 4φ25 3φ25 / 3φ25	228,126 kg	2φ20 / 3φ28 3φ22 / 2φ25 2φ20 / 3φ28	191,4206 kg
	Tulangan Sengkang	φ8-100		φ12-140	
19	Lebar	0,25 m	0,75 m ³	0,33 m	0,9702 m ³
	Tinggi	0,50 m		0,49 m	
	Tulangan Utama	3φ19 / 4φ19 2φ19 / 4φ19 3φ19 / 4φ19	179,4247 kg	2φ25 / 3φ28 3φ22 / 2φ19 2φ25 / 3φ28	172,2049 kg
	Tulangan Sengkang	φ8-100		φ12-160	
20	Lebar	0,25 m	0,5 m ³	0,33 m	0,6468 m ³
	Tinggi	0,50 m		0,49 m	
	Tulangan Utama	3φ19 / 2φ19 3φ19 / 2φ19 3φ19 / 2φ19	91,3503 kg	2φ22 / 3φ22 3φ25 / 2φ28 2φ22 / 3φ22	110,299 kg
	Tulangan Sengkang	φ8-100		φ12-180	
21	Lebar	0,25 m	0,75 m ³	0,33 m	1,0098 m ³
	Tinggi	0,50 m		0,51 m	
	Tulangan Utama	3φ19 / 4φ19 2φ19 / 4φ19 3φ19 / 4φ19	179,4247 kg	2φ22 / 3φ20 3φ25 / 2φ19 2φ22 / 3φ20	151,9952
	Tulangan Sengkang	φ8-100		φ12-170	
Total Volume Beton :		16,56 m ³		17,1624 m ³	
Total Berat Besi :		3113,5654 kg		2693,1281 kg	

Tabel 4-3 Perbandingan Volume Beton dan Berat Besi

Selanjutnya dengan menetapkan harga satuan beton Rp. 250.000,00 / m³ dan harga satuan besi Rp. 5.000,00 / kg dapat dihitung harga struktur secara keseluruhan seperti yang diperlihatkan dalam tabel 4-4, dalam hal ini harga beton dan harga besi tersebut sudah termasuk biaya pemasangan seperti upah pekerja, kawat pengikat dan papan acuan (*form work*).

Dari tabel 4-4 ternyata harga struktur secara keseluruhan dapat dikurangi sebesar Rp. 1.951.587,00 atau sebesar 9,9 % dengan melakukan optimasi. Proses optimasi menggunakan 168 variabel desain dan 171 struktur desain sebagai titik

coba. Waktu proses optimasi yang diperlukan adalah 117 detik menggunakan komputer dengan prosesor Pentium Celeron 333 Mhz dan RAM 64 MB.

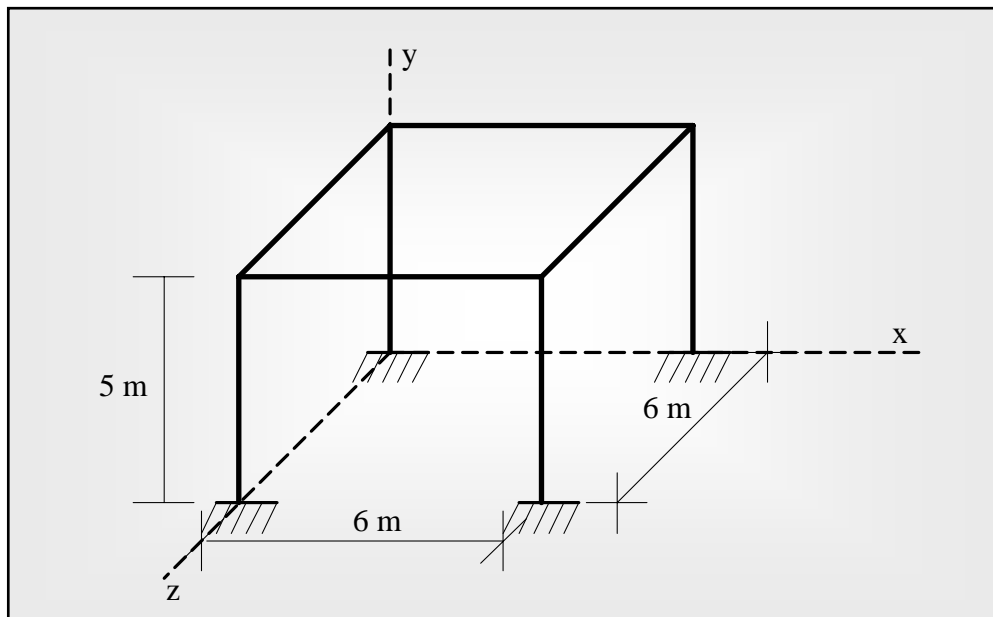
Material	Harga Satuan (Rp.)	Harsoyo (1976)		Naftali (1999)	
		Pemakaian Material	Harga (Rp.)	Pemakaian Material	Harga (Rp.)
Beton	250.000,00/m ³	16.56 m ³	4.140.000	17,1624 m ³	4.290.600
Baja	5.000,00/kg	3113,5654 kg	15.567.827	2693,1281 kg	13.465.640
Total			19.707.827		17.756.240

Tabel 4-4 Perbandingan Harga Struktur

IV.2 Aplikasi Program Optimasi

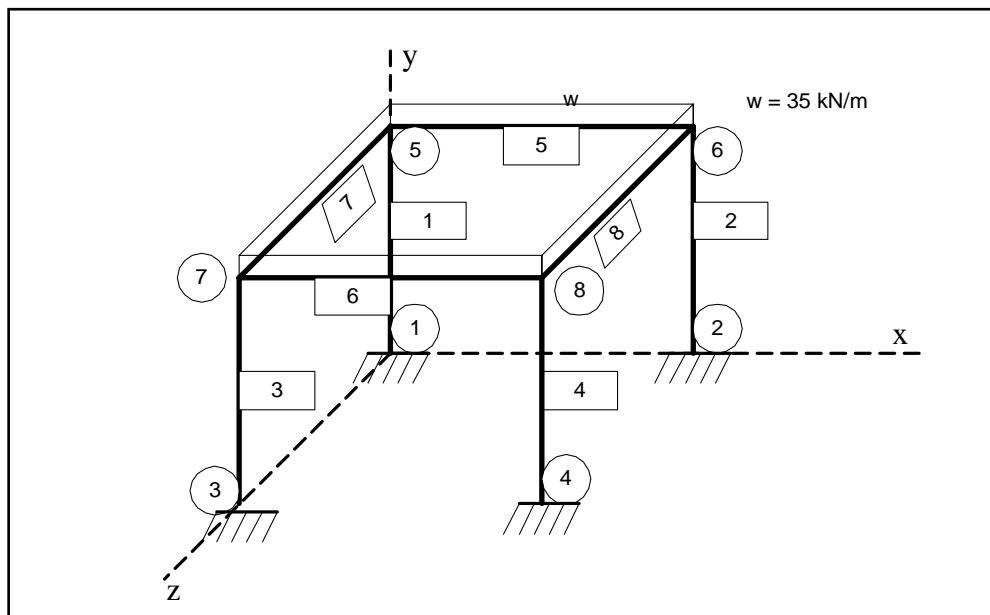
Suatu struktur portal ruang seperti pada gambar 4-3 terbuat dari beton bertulang memiliki data struktur sebagai berikut :

1. Jumlah batang 8 buah, terdiri dari 4 buah balok dan ditopang oleh 4 buah kolom.
2. Tumpuan jepit pada titik kumpul 1, 2, 3, 4.
3. Kuat desak karakteristik beton = 30 MPa.
4. Kuat tarik baja tulangan memanjang = 400 MPa.
5. Kuat tarik baja tulangan sengkang = 240 MPa.
6. Selimut beton pada balok = 50 mm.
7. Selimut beton pada kolom = 50 mm.
8. Beban Merata sebesar 35 kN/m pada semua balok.
9. Variabel desain yang digunakan terdapat pada lampiran.
10. Faktor penalti diambil 1e10.
11. Jumlah struktur desain sebagai titik coba diambil sama dengan jumlah variabel desain + 3, jumlah variabel desain x 2 dan jumlah variabel desain x 3.



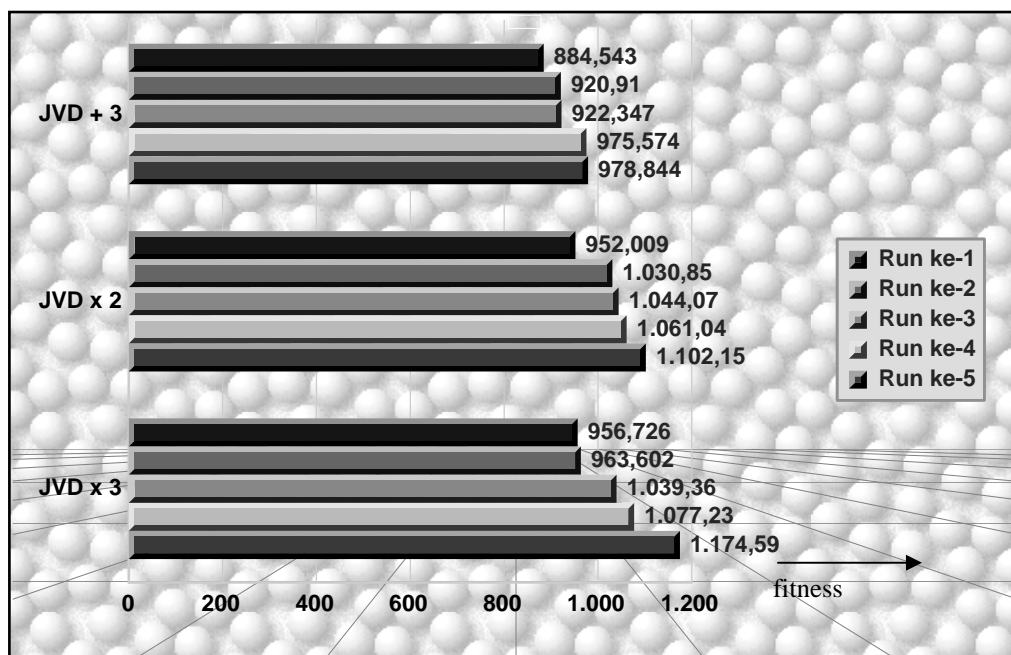
Gambar 4-3 Geometri Struktur Portal Ruang

Penomoran titik kumpul (*joint*) dan penomoran elemen batang diperlihatkan pada gambar 4-4, pada gambar 4-4 juga diperlihatkan pembebanan pada struktur.



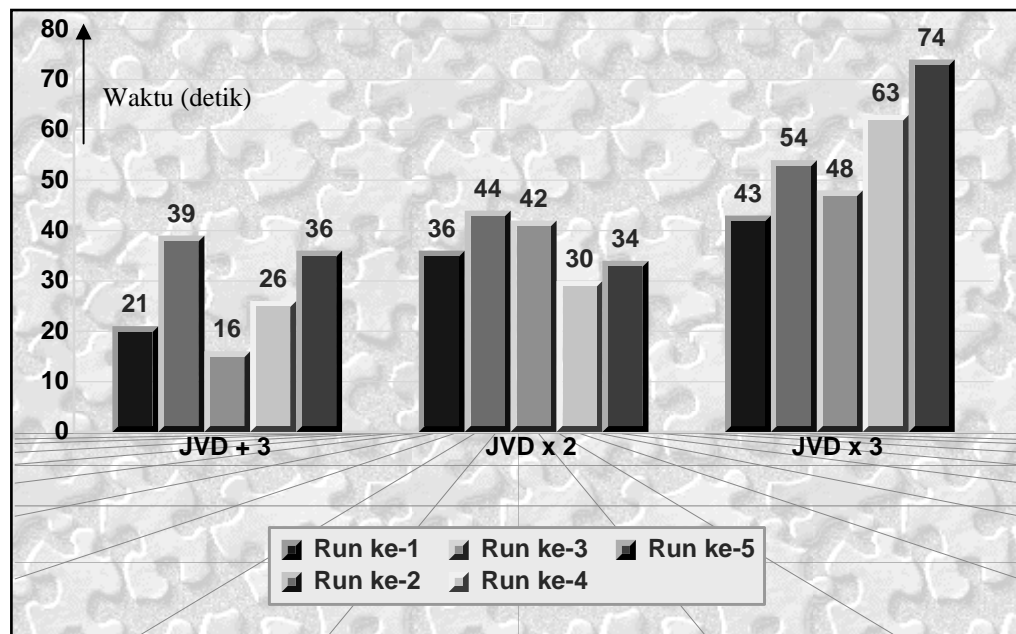
Gambar 4-4 Penomoran Titik dan Batang Portal Ruang

Program dijalankan sebanyak 15 kali dengan 3 variasi jumlah struktur desain, yaitu jumlah variabel desain + 3, jumlah variabel desain x 2, jumlah variabel desain x 3, masing masing sebanyak 5 kali, dengan menggunakan komputer dengan prosesor Pentium Celeron 333 Mhz dan RAM 64 MB, hasil optimasi disajikan dalam bentuk grafik pada gambar 4-5.



Gambar 4-5 Grafik Perbandingan Nilai Fitness

Grafik pada gambar 4-5 menunjukkan bahwa semakin banyak struktur desain yang digunakan, nilai fitness yang dicapai semakin tinggi, hal ini berarti untuk mendapatkan hasil yang makin optimal dibutuhkan jumlah struktur desain yang makin banyak, akan tetapi dengan bertambahnya jumlah struktur desain, waktu yang dibutuhkan untuk proses optimasi juga semakin banyak. Grafik perbandingan waktu proses optimasi yang dibutuhkan oleh masing-masing penambahan jumlah variabel desain ditunjukkan dalam grafik yang terdapat pada gambar 4-6.



Gambar 4-6 Grafik Perbandingan Lama Proses Optimasi

Hasil terbaik untuk kasus portal ruang pada gambar 4-3 dengan nilai fitness 1174,59 memerlukan waktu proses optimasi sebesar 74 detik. Fungsi sasarannya yaitu harga struktur, dengan mengambil harga beton Rp. 250.000,00 / m³ dan harga besi Rp. 5.000,00 / kg adalah sebesar Rp. 8.535.430,00, hasilnya ditunjukkan pada tabel 4-5, sedangkan perhitungan rincian harga terdapat pada tabel 4-6.

Batang		Penggunaan Material	
No.	Variabel Desain		
1	Lebar	0,45 m	1,0125 m ³
	Tinggi	0,45 m	
	Tulangan Utama	8φ19	102,4988 kg
	Tulangan Sengkang	φ10-300	
2	Lebar	0,50 m	1,25 m ³
	Tinggi	0,50 m	
	Tulangan Utama	8φ22	134,7477 kg
	Tulangan Sengkang	φ10-300	
3	Lebar	0,50 m	1,25 m ³
	Tinggi	0,50 m	
	Tulangan Utama	8φ28	208,6947 kg
	Tulangan Sengkang	φ10-300	

4	Lebar	0,45 m	1,0125 m ³
	Tinggi	0,45 m	
	Tulangan Utama	8 ϕ 22	127,9512 kg
	Tulangan Sengkang	ϕ 8-300	
5	Lebar	0,25 m	0,9 m ³
	Tinggi	0,60 m	
	Tulangan Utama	4 ϕ 25 / 3 ϕ 19 4 ϕ 19 / 3 ϕ 19 4 ϕ 25 / 3 ϕ 19	164,686 kg
	Tulangan Sengkang	ϕ 10-160	
6	Lebar	0,30 m	1,08 m ³
	Tinggi	0,60 m	
	Tulangan Utama	2 ϕ 25 / 4 ϕ 22 4 ϕ 22 / 3 ϕ 29 2 ϕ 25 / 4 ϕ 22	195,3126 kg
	Tulangan Sengkang	ϕ 10-160	
7	Lebar	0,35 m	1,05 m ³
	Tinggi	0,50 m	
	Tulangan Utama	2 ϕ 19 / 3 ϕ 25 4 ϕ 28 / 2 ϕ 22 2 ϕ 19 / 3 ϕ 25	171,1133 kg
	Tulangan Sengkang	ϕ 10-160	
8	Lebar	0,30 m	1,08 m ³
	Tinggi	0,60 m	
	Tulangan Utama	2 ϕ 28 / 2 ϕ 22 2 ϕ 28 / 3 ϕ 29 2 ϕ 28 / 2 ϕ 22	170,3307 kg
	Tulangan Sengkang	ϕ 10-180	
Total Volume Beton			8,635 m ³
Total Berat Besi			1275.336 kg

Tabel 4-5 Dimensi Struktur Portal Ruang Hasil Optimasi

Material	Harga Satuan (Rp.)	Pemakaian Material	Harga (Rp.)
Beton	250.000,00/m ³	8,635 m ³	2.158.750
Baja	5.000,00/kg	1275,336 kg	6.376.680
Total			8.535.430

Tabel 4-6 Rincian Harga Struktur Portal Ruang

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Dari tugas akhir ini dapat disimpulkan bahwa :

1. Program komputer optimasi beton bertulang pada struktur portal ruang yang dikembangkan dalam tugas akhir ini dengan menggunakan metoda optimasi polihedron fleksibel berhasil mengurangi harga struktur sekitar 9,9 %
2. Bertambahnya jumlah struktur desain sebagai titik coba menghasilkan struktur yang semakin murah, tetapi memerlukan waktu proses optimasi yang semakin lama
3. Untuk jumlah variabel yang relatif besar, hasil yang didapatkan oleh metoda optimasi polihedron fleksibel kurang memuaskan, metoda polihedron fleksibel cukup baik hanya untuk kasus yang memiliki jumlah variabel desain yang relatif kecil.

V.2. Saran

Untuk menyelesaikan masalah optimasi yang cukup rumit seperti optimasi beton bertulang pada struktur portal ruang, diperlukan metoda optimasi yang lebih baik lagi, ada beberapa saran yang mungkin dapat membantu dalam mengembangkan optimasi beton bertulang pada struktur portal ruang yaitu :

1. Dalam memilih bahasa pemrograman, sebaiknya digunakan bahasa yang dapat menembus memori lebih dari 64K, karena dalam pemrograman optimasi beton

bertulang pada struktur portal ruang dibutuhkan dimensi sebesar mungkin, selain itu untuk sistem operasi dapat dicoba menggunakan sistem operasi yang sudah berbasiskan 64 bit atau yang *flat memory*, misalnya menggunakan kernel linux.

2. Metoda optimasi polihedron fleksibel perlu dikembangkan lagi, terutama dalam proses penelusurannya, sehingga metoda ini dapat menjadi lebih baik lagi.

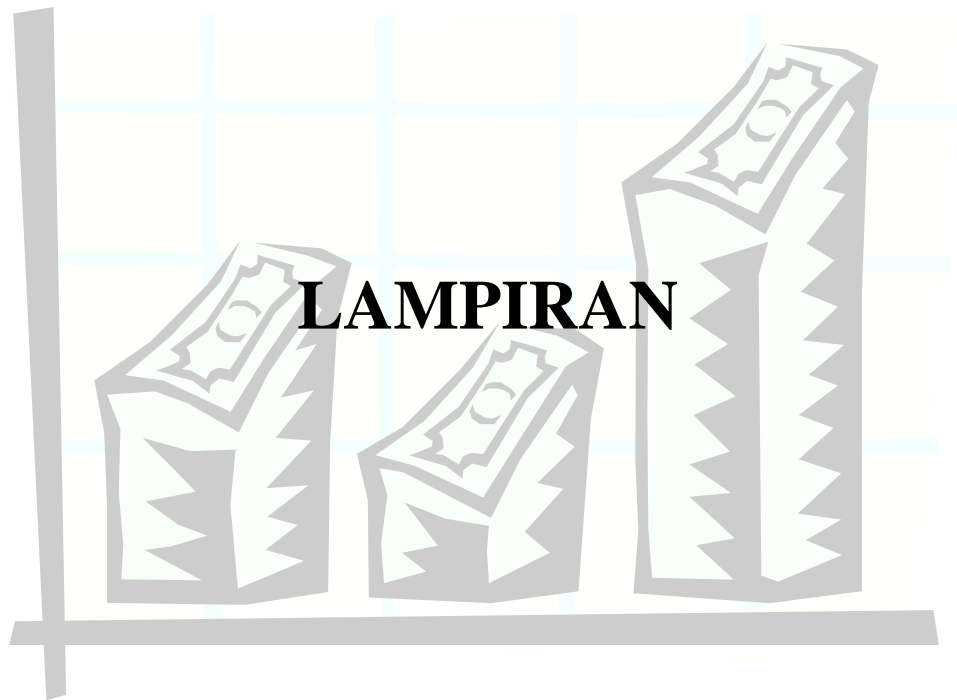
DAFTAR PUSTAKA

- Balfour, James A. D., 1986, *Computer Analysis of Structural Frameworks*, Collins, London.
- Dipohusodo, I., 1994, *Struktur Beton Bertulang*, Departemen Pekerjaan Umum RI, Gramedia Pustaka Utama, Jakarta.
- Harsoyo, 1976, *Seri Bina Bangunan, Konstruksi Beton Bertulang, Gedung Bertingkat II*, R.Sugihardjo BAE, Yogyakarta.
- Hulse, R and Mosley W.H., 1989, *Reinforced Concrete Design by Computer*, Macmillan, Singapore.
- Kirsch, U., 1981, *Optimum Structural Design*, McGraw-Hill Company.
- Lai, Y. S., and Achenbach, J. D, 1973, *Direct Search Optimization Method*, *Journal of Structural Division*, ASCE, Vol. 98, pp.119-131.
- Lev, O. E., 1977, *A Structural Optimization Solution to a Branch and Bound Problem*, *Quarterly of Applied Mathematics*, Vol. 34, pp. 365-371.
- Levey, G. C., and Fu, K.C., 1979, *A Method in Discrete Frame Optimization and Its Outlook*, *International Journal of Computers and Structures*, Vol. 10, pp. 2177-2197.
- MacGregor, James G., 1997, *International Edition, Reinforced Concrete, Mechanics and Design, Thrid Edition*, Prentice-Hall, New Jersey.
- Morris, A. J., 1982, *Foundations of Structural Optimization : A Unified Approach*, John Wiley & Sons, New York.
- Nawy, E., 1990, *Beton Bertulang Suatu Pendekatan Dasar*, Terjemahan, PT Eresco, Bandung.
- SK SNI T-15-1991-03, 1991, *Standar Tata Cara Perhitungan Struktur Beton untuk Bangunan Gedung*, Departemen Pekerjaan Umum, Yayasan Lembaga Penyelidikan Masalah Bangunan, Bandung.
- Sutedjo, B. dan Michael, 1997, *Algoritma dan Teknik Pemrograman*, Andi, Yogyakarta.
- Toakley, A. R., 1968, *Optimum Design Using Available Section*, *Journal of Structural Division*, ASCE, Vol. 34, pp. 1219-1241.

- Wang, C.K. dan Salmon C.G., 1990, *Desain Beton Bertulang*, Edisi keempat, Terjemahan, Erlangga, Jakarta.
- Wahyudi, L. dan Syahril A.R., 1997, *Struktur Beton Bertulang Standar Baru SNI T-15-1991-03*, PT Gramedia Pustaka Utama, Jakarta.
- Weaver, W.Jr. dan James M. Gere, *Analisa Matriks Untuk Struktur Rangka*, Edisi kedua, Terjemahan, Erlangga, Jakarta.
- Wibowo, F.N., 1996, *Perkembangan Metoda Optimasi pada Teknik Sipil*, diktat kuliah Metoda Optimasi Struktur, Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Wu, Peiming, 1986, *Structural Optimization with Nonlinear and Discrete Programming*, University Microfilm International, Michigan.
- Vis, W.C. dan Kusuma G.H., 1993, *Dasar-dasar Perencanaan Beton Bertulang*, Erlangga, Jakarta.
- Young, Warren C., 1989, *Roark's Formulas for Stress & Strain*, McGraw-Hill, New York.

OPTIMASI BETON BERTULANG PADA STRUKTUR PORTAL RUANG

TUGAS AKHIR SARJANA STRATA -1



[illegible]

```

/*****
/* FILE : BORLANDC.HPP
/* HEADER PUSTAKA BORLAND C++ VERSI 5.02 YANG DIPAKAI
/* PUSTAKA YANG DIPANGGIL DIBUAT OLEH BORLAND INTERNATIONAL
*****/
#include <math.h>           // Pustaka Pengolahan Matematika
#include <_defs.h>          // Definisi Umum Untuk Pointer Dan Conventions
#include <stddef.h>         // Definisi Untuk Tipe Umum Dan Null
#include <_null.h>          // Definisi Null Untuk Pointer
#include <time.h>           // Pustaka Operasi Waktu
#include <dos.h>            // Fungsi Fasilitas Disk Operating System
#include <iostream.h>       // Stream Input Output Ke Layar
#include <string.h>         // Pustaka Pembantu Operasi String
#include <fstream.h>        // Stream Untuk Menangani Masalah File
#include <stdlib.h>         // Standar Pustaka C
#include <conio.h>          // Operasi Rutin Input Output Pada Konsol
#include <iomanip.h>        // Operasi Manipulasi Input Output

```

```

/*****
/* FILE : HEADER.HPP                               */
/* KUMPULAN HEADER PROGRAM OPTIMASI BETON BERTULANG PADA STRUKTUR RUANG */
/* DENGAN METODA FLEXIBLE POLYHEDRON                */
*****/

/*****/
/* HEADER KERNEL */
/*****/
#include "Proto.hpp"
#include "Variabel.hpp"
#include "InOut.hpp"

/*****/
/* HEADER MEKANIKA REKAYASA PORTAL RUANG DENGAN METODA KEKAKUAN */
/*****/
#include "Struktur.hpp"
#include "Pembebanan.hpp"
#include "Solver.hpp"

/*****/
/* HEADER ANALISA BETON BERTULANG */
/*****/
#include "Kolom.hpp"
#include "Balok.hpp"
#include "Elemen.hpp"

/*****/
/* HEADER OPTIMASI DENGAN METODA FLEXIBLE POLYHEDRON */
/*****/
#include "Polyhedron.hpp"
#include "Penormalan.hpp"
#include "Pengacakan.hpp"
#include "Diskritisasi.hpp"
#include "Kendala.hpp"
#include "Telusur.hpp"
#include "Baru.hpp"
#include "Pengurutan.hpp"

/*****/
/* HEADER SHELL */
/*****/
#include "Tampilan.hpp"
#include "Cetak.hpp"

```

```

/*****
/* FILE : PROTO.HPP
/* PROTOTYPE OPTIMASI STRUKTUR BETON BERTULANG PADA PORTAL RUANG
*****/

/*****/
/* PROTOTYPE PROGRAM UMUM */
/*****/
void menu_utama();
void about();
void input_data();
void input_data_umum();
void input_data_diskrit();

/*****/
/* PROTOTYPE PROGRAM MEKANIKA REKAYASA STRUKTUR DENGAN METODA KEKAKUAN */
/*****/
void baca_data();
void load_data();
void kosong_beban();
void baca_beban();
void struktur();
void inersia();
void kekakuan_batang();
void beban();
void berat_sendiri();
void isi_matrik_kekakuan(int imk);
void indeks_batang(int ib);
void periksa_batang(int b);
void banfac(int N,int NB,float A_SFF[][mak]);
void bansol(int N,int NB,float U_SFF[][mak],float B_AC[],float X_DF[]);
void hasil();
void output_parameter_struktural();
void output_koordinat_titik_kumpul();
void output_informasi_batang();
void output_pengekang_titik_kumpul();
void output_beban_batang();
void output_beban_titik();

/*****/
/* PROTOTYPE PROGRAM BETON BERTULANG */
/*****/
void isi_elemen_balok(int no_el_balok);
void elemen_lapangan(int no_el_balok);
void elemen_tumpuan(int no_el_balok);
void isi_elemen_kolom(int no_el_kolom);
void lendutan(int no_batang_l);

/*****/
/* PROTOTYPE PROGRAM OPTIMASI DENGAN METODA FLEXIBLE POLYHEDRON */
/*****/
float Kendala_Harga(int var_b_nya[],int var_k_nya[]);
float isi(int no_data,float kelompok_data[]);
void optimasi();
void load_batas_atas();
void acak_variabel();
void randomisasi();
void cari_struktur_awal();
void cari_baru();
void periksa_batas();
void penelusuran();
void ganti_baru();
void penyusutan();
void normalisasi_float(float var_nor[],float var_bv[],float var_kv[]);
void normalisasi_int(int var_nor[],int var_bv[],int var_kv[]);
void unnormalisasi(float var_nor[],float var_bv[],float var_kv[]);
void cetak_akhir();
void sort(float nilai1[],float nilai2[],float nilai3[],int urut_var_b[][mak],int urut_var_k[][mak],int n_array);
int konversi(float bil_asli);

```

```

/*****/
/* FILE : VARIABEL.HPP */
/* PENDEKLARASIAN VARIABEL GLOBAL */
/*****/

/*****/
/* MAKRO VARIABEL */
/*****/
#define pi 3.14 // Phi lingkaran
#define teta 0.8 // Faktor reduksi kekuatan
#define limit_nol 1.E-3 // Penetapan angka kecil mendekati nol
#define bj_besi 7850. // berat jenis besi (kg/m^3)
#define MD 12 // Jumlah perpindahan untuk satu batang

/*****/
/* VARIABEL UMUM */
/*****/
char fi[20]; // Input nama file generik tanpa ekstensi
char finput[20]; // File *.inp untuk data umum struktur
char fsisi[20]; // File *.isd untuk data sisi penampang
char fdial[20]; // File *.idl untuk diameter tulangan utama
char fjtl[20]; // File *.ijl untuk jumlah tulangan utama
char fdias[20]; // File *.ids untuk diameter tulangan sengkang
char fjts[20]; // File *.ijs untuk jarak tulangan sengkang
char fbeban[20]; // File *.bbn untuk data beban
char fhhistory[20]; // File *.his untuk mencatat riwayat optimasi
char foptimasi[20]; // File *.opt untuk keluaran hasil optimasi
char fstruktur[20]; // File *.str untuk hasil analisa struktur
char fkendala[20]; // File *.kdl untuk keluaran kendala struktur
char finformasi[20]; // File *.inf untuk keluaran informasi masukkan
char SubName[80]; // Informasi nama sub masukkan
char ISN[80]; // Informasi nama struktur

/*****/
/* VARIABEL MEKANIKA REKAYASA UNTUK STRUKTUR PORTAL RUANG */
/*****/
int NB; // Setengah lebar jalur matrik kekakuan
int M; // Jumlah batang
int NJ; // Jumlah titik kumpul
int NR; // Jumlah Pengekang tumpuan
int NRJ; // Jumlah titik kumpul yang dikekang
int JRL[mak]; // Daftar pengekang titik kumpul
int T_K[mak]; // Daftar titik kumpul yang dikekang
int ND; // Jumlah koordinat perpindahan untuk semua titik kumpul
int N; // Jumlah derajat kebebasan
int IA[mak]; // Notasi penunjuk nol atau tidaknya sudut alfa
int JJ[mak]; // Penunjuk untuk ujung j
int JK[mak]; // penunjuk untuk ujung k
int ID[mak]; // Indeks perpindahan untuk titik kumpul
int IR,IC; // Indeks baris dan kolom
int IM[mak]; // Indeks perpindahan batang
int LML[mak]; // Tabel batang yang dibebani

float E; // Modulus elastisitas
float G; // Modulus Puntir
float X[mak]; // Koordinat struktur pada arah x ( _ ) | y
float Y[mak]; // Koordinat struktur pada arah y ( | ) | _
float Z[mak]; // Koordinat struktur pada arah z ( / ) z/ x
float b[mak]; // Lebar penampang
float h[mak]; // Tinggi penampang
float AX[mak]; // Luas Penampang
float XI[mak]; // Konstanta puntir batang
float YI[mak]; // Momen inersia terhadap sumbu y batang
float ZI[mak]; // Momen inersia terhadap sumbu z batang
float EL[mak]; // Panjang penampang

float CX; // Kosinus arah x
float Cy; // Kosinus arah y
float CZ; // Kosinus arah z
float CXZ; // Penunjuk balok atau kolom
float XP[mak]; // Koordinat x dari titik P (m)

```

```

float YP[mak]; // Koordinat y dari titik P (m)
float ZP[mak]; // Koordinat z dari titik P (m)
float XPS;      // Koordinat xs dari titik P (m)
float YPS;      // Koordinat ys dari titik P (m)
float ZPS;      // Koordinat zs dari titik P (m)
float YPG;      // Koordinat yg dari titik P (m)
float ZPG;      // Koordinat zg dari titik P (m)
float COSA;     // Cosinus sudut alfa
float SINA;     // Sinus sudut alfa

float R11[mak]; // ~
float R12[mak]; // |
float R13[mak]; // |
float R21[mak]; // |
float R22[mak]; // -- Matrik Rotasi
float R23[mak]; // |
float R31[mak]; // |
float R32[mak]; // |
float R33[mak]; // ~

float SM[13][13]; // Matrik kekakuan batang lokal
float SMRT[13][13]; // Hasil perkalian matrik SM dan matrik Rotasi
float SMS[mak][mak]; // Matrik kekakuan batang global
float SFF[mak][mak]; // Matrik kekakuan untuk perpindahan bebas
float DF[mak]; // Perpindahan titik dalam sumbu global (m)
float AJ[mak]; // Aksi beban pada titik kumpul dalam arah global
float AML[13][mak]; // Gaya ujung batang terkekang dalam arah lokal
float AE[mak]; // Beban titik kumpul ekuivalen dalam arah global
float AC[mak]; // Beban titik kumpul gabungan dalam arah global
float DJ[mak]; // Perpindahan titik kumpul dalam arah global
float AMD[mak]; // Gaya ujung batang akibat perpindahan titik
float AM[mak][mak]; // Gaya ujung batang akhir
float AR[mak]; // Reaksi tumpuan dalam arah sumbu global
float W[mak]; // Beban Merata
float W_Balok[mak]; // Beban Berat Sendiri Balok
float P_Kolom[mak]; // Gaya Terpusat Kolom

float MTUM_KI[mak]; // Momen pada tumpuan kiri
float MTUM_KA[mak]; // Momen pada tumpuan kanan
float MLAP[mak]; // Momen lapangan
float GESER_KI[mak]; // Gaya geser sebelah kiri
float GESER_KA[mak]; // Gaya geser sebelah kanan

float MKX[mak]; // Momen kolom arah x
float MKY[mak]; // Momen kolom arah y
float PK[mak]; // Gaya aksial kolom
float GK[mak]; // Gaya geser kolom

/*****/
/* VARIABEL BETON BERTULANG */
/*****/

//////////
// Data Masukkan Umum //
//////////
float FC; // Kuat desak karakteristik Beton (MPa)
float FY; // Kuat tarik baja tulangan utama (MPa)
float FYS; // Kuat tarik baja tulangan sengkang (MPa)
float BTl; // Faktor reduksi tinggi blok tegangan ekuivalen beton

//////////
// Variabel Desain Sharing //
//////////
int nsisi_B; // Jumlah data diskrit lebar balok
int nsisi_H; // Jumlah data diskrit tinggi balok
int nsisi_K; // Jumlah data diskrit sisi kolom

int nDIA; // Jumlah data diskrit diameter
int nNL; // Jumlah data diskrit jumlah diameter
int nDIAS; // Jumlah data diskrit diameter sengkang
int nJS; // Jumlah data diskrit jarak antar sengkang

```



```

float DIAS;      // Diameter sengkang (mm)
float Jarak_S;   // Jarak antar sengkang (mm)
float Sref;      // Jarak antar sengkang yang dibutuhkan (mm)
float jarak_antar_tulangan; // Jarak antar tulangan (mm)

float sisi_d_B[mak]; // Array berisi data diskrit lebar balok (mm)
float sisi_d_H[mak]; // Array berisi data diskrit tinggi balok (mm)
float sisi_d_K[mak]; // Array berisi data diskrit sisi kolom (mm)

float DIA_d[mak]; // Array berisi data diskrit diameter (mm)
float NL_d[mak];  // Array berisi data diskrit jumlah tulangan
float DIAS_d[mak]; // Array berisi data diskrit diameter sengkang (mm)
float JS_d[mak];  // Array berisi data diskrit jarak antar sengkang (mm)

//////////
// Variabel Desain Balok //
//////////
float B;      // Lebar balok (mm)
float H;      // Tinggi balok (mm)
float DIA1;   // Diameter tulangan tarik (mm)
float DIA2;   // Diameter tulangan desak (mm)
float NL1;    // Jumlah tulangan tarik (mm)
float NL2;    // Jumlah tulangan desak (mm)

//////////
// Variabel Desain Kolom //
//////////
float sisi;    // Sisi penampang kolom biaksial (mm)
float DIA;     // Diameter tulangan (mm)
float N_DIA;   // Jumlah tulangan

//////////
// Sharing Variabel Gaya //
//////////
float PU;      // Gaya aksial ultimit (N)
float MU;      // Momen ultimit (Nm)
float VU;      // Gaya geser ultimit (N)
float VC;      // Gaya geser yang disumbangkan oleh beton (N)
float VS;      // Gaya geser yang disumbangkan oleh tulangan geser (N)
float FMU;     // Momen Ultimit yang dapat ditahan(Nm)
float FPU;     // Gaya Aksial Ultimit yang dapat ditahan (N)

//////////
// Sharing Variabel Biaya //
//////////
float harga_beton; // Harga beton (Rp./m^3)
float harga_besi;  // Harga besi (Rp./kg)
float volume_beton; // Volume beton yang digunakan (m^3)
float berat_besi;   // Berat besi yang digunakan (kg)
float berat_sengkang; // Berat sengkang yang digunakan (kg)
float L;            // Panjang batang (m)

//////////
// Sharing Variabel Elemen //
//////////
float DS;      // Jarak serat tarik terluar ke tulangan tarik (mm)
float D;       // Jarak serat tekan terluar ke tulangan tarik (mm)
float AV;      // Luas tulangan geser (mm^2)
float AVmin;   // Luas tulangan geser minimum (mm^2)

//////////
// Variabel Khusus Balok //
//////////
float selimut_balok; // Tebal selimut pada balok

float DIA1lap; // Diameter tulangan tarik pada daerah lapangan
float NL1lap;  // Jumlah tulangan tarik pada daerah lapangan
float DIA2lap; // Diameter tulangan desak pada daerah lapangan
float NL2lap;  // Jumlah tulangan desak pada daerah lapangan

```

```

float DIAltum; // Diameter tulangan tarik pada daerah tumpuan
float NLltum; // Jumlah tulangan tarik pada daerah tumpuan
float DIA2tum; // Diameter tulangan desak pada daerah tumpuan
float NL2tum; // Jumlah tulangan desak pada daerah tumpuan

float AS; // Luas tulangan tarik
float ASl; // Luas tulangan desak
float HMIN; // Tinggi Minimum
float n; // Rasio Es/Ec
float fr; // Modulus Keruntuhan lentur dari beton (MPa)
float LGN; // Letak Garis Netral (mm)
float Icr; // Momen Inersia Penampang Retak (mm^4)
float Ig; // Momen Inersia Penampang Kotor (mm^4)
float Ie; // Momen Inersia Efektif (mm^4)
float Mcr; // Momen Retak (Nmm);

float LENDUTAN; // Lendutan pada tengah bentang (mm)
float LENDUTAN_IJIN; // Lendutan ijin (mm);
float Lambda; // Faktor pengali lendutan jangka panjang

//////////
// Variabel Khusus Kolom //
//////////
float MUX; // Momen arah x ultimit (Nm)
float MUY; // Momen arah y ultimit (Nm)
float selimut_kolom; // Tebal selimut pada kolom
float _K; // Faktor kelangsingan

/*****
/* VARIABEL OPTIMASI DENGAN METODA FLEXIBLE POLYHEDRON */
*****/
int jum_susut; // Jumlah penyusutan berturut
int fak_plus; // Faktor penambah (JSTD=JVD+fak_plus)
int fak_kali; // Faktor Pengali (JSTD=JVD*fak_kali)
int no_struktur; // Nomor Struktur
int JVD; // Jumlah variabel desain
int JSTD; // Jumlah struktur desain
int j_iterasi_mak; // Jumlah iterasi maksimum
int nv[mak]; // Batas atas variabel diskrit normal
int nvk[mak]; // Batas atas variabel diskrit kolom
int nvb[mak]; // Batas atas variabel diskrit balok
int iterasi_var; // Jumlah iterasi untuk melangkah
int var_b[mak][mak]; // Nomor variabel balok diskrit yang digunakan
int var_k[mak][mak]; // Nomor variabel kolom diskrit yang digunakan
int tanda_arah_b[mak]; // Tanda arah-arrah penelusuran bagian balok
int tanda_arah_k[mak]; // Tanda arah-arrah penelusuran bagian kolom
int var_b_jelek[mak]; // Koordinat variabel jelek bagian balok
int var_k_jelek[mak]; // Koordinat variabel jelek bagian kolom
int arah[mak]; // Koordinat arah penelusuran
int no_TS_terjauh; // Nomor arah yang paling jauh
int varnew_asli[mak]; // Variabel terjelek hasil penggandaan
int varnew[mak]; // Variabel baru bertipe integer
int var_b_cb[mak]; // Variabel desain balok baru
int var_k_cb[mak]; // Variabel desain kolom baru
int var_b_cb_best[mak]; // Variabel desain balok baru terbaik
int var_k_cb_best[mak]; // Variabel desain kolom baru terbaik
int no_balok[mak]; // Identifikasi nomor balok dengan nomor batang
int no_kolom[mak]; // Identifikasi nomor kolom dengan nomor batang
int jum_balok; // Jumlah balok pada struktur
int jum_kolom; // Jumlah kolom pada struktur
int js_balok; // Nomor balok (Pembantu)
int js_kolom; // Nomor kolom (Pembantu)
int lompat; // Identifikasi untuk melompat dari loop

float patok_fit; // Variabel Patokan Fitness
float finalti; // Faktor finalti untuk pelanggaran
float fitstr[mak]; // Fitness struktur
float fitcb; // Fitness baru
float fitcb_best; // Fitness baru yang terbaik

```

```

float kendalastr[mak]; // Kendala struktur
float hargastr[mak];   // Harga struktur (Rp.)
float kendala;         // Kendala
float kendalasa;       // Kendala pada saat struktur awal
float harga;           // Harga (Rp.)
float XM_b[mak];       // Koordinat titik Midle bagian balok
float XM_k[mak];       // Koordinat titik Midle bagian kolom
float XS_b[mak];       // Arah-arrah penelusuran bagian balok
float XS_k[mak];       // Arah-arrah penelusuran bagian kolom
float TM[mak];         // Koordinat titik Midle
float TS[mak];         // Koordinat titik Search
float varplus[mak];    // Variabel penambahan
float varnew_f[mak];   // Variabel baru bertipe float

////////////////////////////////////
// Variabel Kendala Kolom Biaksial //
////////////////////////////////////
float kendala_gaya;    // Kendala gaya pada material
float kendala_po;     // Gaya tekan maksimum akibat eksentrisitas
float kendala_pn;     // Kendala gaya tekan
float kendala_mn;     // Kendala momen

float kendala_r;       // Kendala rasio penulangan
float kendala_r_min;   // Rasio tulangan minimum
float kendala_r_mak;   // Rasio tulangan maksimum

float kendala_senggang; // Kendala sengkang
float kendala_tul;      // Kendala jumlah dan jarak tulangan
float kendala_kelangsingan; // Kendala kelangsingan

////////////////////////////////////
// Variabel Kendala Balok //
////////////////////////////////////
float kendala_rho;     // Besarnya kendala akibat rasio tulangan
float kendala_rho_b;   // Besarnya kendala akibat rasio tulangan balance
float kendala_rho_m;   // Besarnya kendala akibat rasio tulangan minimum
float kendala_sb;      // Besarnya kendala akibat kebutuhan sengkang
float kendala_M;       // Besarnya kendala akibat momen
float kendala_lendutan; // Besarnya kendala akibat tinggi minimum

```

```

/*****
/* FILE : INOUT.HPP
/* KUMPULAN SUBPROGRAM MASUKAN DAN KELUARAN
*****/

/*****
/* SUBPROGRAM UNTUK MEMASUKAN DATA AWAL */
*****/
void input_data()
{
    cout << " Nama file input (tanpa ekstensi) = ";
    cin >> fi;
    strcpy(fininput,fi);
    strcat(fininput, ".inp"); // input umum
    strcpy(fsis,fi);
    strcat(fsis, ".isd"); // input sisi diskrit
    strcpy(fdial,fi);
    strcat(fdial, ".idl"); // input diameter tulangan lentur
    strcpy(fjtl,fi);
    strcat(fjtl, ".ijl"); // input jumlah tulangan lentur
    strcpy(fdias,fi);
    strcat(fdias, ".ids"); // input diameter tulangan sengkang
    strcpy(fjts,fi);
    strcat(fjts, ".ijs"); // input jarak tulangan sengkang
    int pilih_input;
    do
    {
        clrscr();
        cout << " 1. Input data umum\n";
        cout << " 2. Input data diskrit elemen\n";
        cout << " 3. Keluar\n";
        cout << " Pilihan (1-3) = ";
        cin >> pilih_input;
        if(pilih_input==1)
        {
            input_data_umum();
        }
        if(pilih_input==2)
        {
            input_data_diskrit();
        }
    }while(pilih_input!=3);
}

/*****
/* SUBPROGRAM UNTUK MEMASUKAN DATA UMUM */
*****/
void input_data_umum()
{
    int iinp,jinp;
    char uji;

    ///////////////////////////////////
    // Memasukkan Data Umum //
    ///////////////////////////////////
    cout << " Input data awal\n";
    cout << " Nama file input (Tanpa Ekstensi) = " << fininput << endl;
    do
    {
        cout << " Nama Struktur (Tanpa Spasi) = ";
        cin >> ISN;
        cout << " Jumlah batang = ";
        cin >> M;
        cout << " Jumlah titik kumpul = ";
        cin >> NJ;
        cout << " Jumlah titik kumpul yang dikekang = ";
        cin >> NRJ;
        cout << " Jumlah pengekang tumpuan = ";
        cin >> NR;
        cout << " Kuat desak Karakteristik Beton (MPa) = ";
        cin >> FC;
    }
}

```

```

cout << " Kuat tarik baja tulangan lentur (MPa) = ";
cin >> FY;
cout << " Kuat tarik baja tulangan sengkang (MPa) = ";
cin >> FYS;

////////////////////////////////////
// Hitung Parameter Lainnya Secara Otomatis //
////////////////////////////////////
E=(4700.*(sqrt(FC))*1E6; // (N/mm^2) -> (N/m^2)
G=(E/2.)*(1.+0.15); // (N/m^2)
ND=6.*NJ;
N=ND-NR;

output_parameter_struktural();
cout << "\n Ingin mengulang (y/t) ";
cin >> uji;
if (uji!= 'y' )
{break;}
} while(uji== 'y' );

////////////////////////////////////
// Memasukkan Data Koordinat Titik Kumpul //
////////////////////////////////////
cout << " Inputkan koordinat titik kumpul (m),"
<< " pilih J = 0 untuk keluar \n";
do
{
    cout << " Titik kumpul J = ";
    cin >> jinp;
    if ( jinp == 0 )
    { break ; }
    cout << " Koordinat\n";
    cout << " X Y Z\n";
    cin >> X[jinp] >> Y[jinp] >> Z[jinp];
    output_koordinat_titik_kumpul();
} while(jinp != 0) ;

////////////////////////////////////
// Mengosongkan Pengekang //
////////////////////////////////////
for (int jinp=1;jinp<=ND;jinp++)
{
    JRL[jinp]=0;
}

////////////////////////////////////
// Memasukkan Data Pengekang Joint //
////////////////////////////////////
cout << endl << " Inputkan Pengekang titik kumpul,"
<< " masukan 0 untuk keluar \n";
do
{
    cout << " Kekangan ke = ";
    cin >> iinp;
    if (iinp==0)
    {
        break ;
    }
    cout << " Titik kumpul yang dikekang = ";
    cin >> T_K[iinp];
    cout << " R (1 = dikekang, 0 = tak dikekang)= ";
    cin >> JRL[6*T_K[iinp]-5]
>> JRL[6*T_K[iinp]-4]
>> JRL[6*T_K[iinp]-3]
>> JRL[6*T_K[iinp]-2]
>> JRL[6*T_K[iinp]-1]
>> JRL[6*T_K[iinp]];
    output_pengekang_titik_kumpul();
}while (iinp != 0);

```

```

////////////////////////////////////
// Memasukkan Informasi Batang //
////////////////////////////////////
cout << endl << "   Inputkan informasi batang,"
      << "   pilih I = 0 untuk keluar \n";

do
{
    cout << "   Indeks batang = ";
    cin  >> iinp;
    if (iinp == 0)
        { break ; }
    cout << "   titik j = ";
    cin  >> JJ[iinp];
    cout << "   titik k = ";
    cin  >> JK[iinp];
    cout << "   Notasi untuk menunjukan nol atau tidaknya sudut a = ";
        cin  >> IA[iinp];
    if(IA[iinp] != 0)
        {
            cout << "   Sudut alfa tidak 0, masukkan koordinat titik p
\n";

            cout << "   Untuk batang " << iinp << endl;
            cout << "   XP   YP   ZP\n";
            cin  >> XP[iinp] >> YP[iinp] >> ZP[iinp];

            cout << endl;
        }
        output_informasi_batang();
    } while ( iinp != 0 );

////////////////////////////////////
// Penulisan Ke File //
////////////////////////////////////
ofstream tulis(finput);
tulis << ISN << endl;
tulis << M << endl;
tulis << NJ << endl;
tulis << NRJ << endl;
tulis << NR << endl;
tulis << E << endl;
tulis << G << endl;
tulis << FC << endl;
tulis << FY << endl;
tulis << FYS << endl;
tulis << ND << endl;
tulis << N << endl;
tulis << "[Koordinat]" << endl;
for(int ktl=1;ktl<=NJ;ktl++)
{
    tulis << ktl << endl;
    tulis << X[ktl] << endl;
    tulis << Y[ktl] << endl;
    tulis << Z[ktl] << endl;
}
tulis << "[Pengekang]" << endl;
for (int iinp=1;iinp<=NRJ;iinp++)
{
    tulis << T_K[iinp] << endl;
    tulis << JRL[6*T_K[iinp]-5] << endl;
    tulis << JRL[6*T_K[iinp]-4] << endl;
    tulis << JRL[6*T_K[iinp]-3] << endl;
        tulis << JRL[6*T_K[iinp]-2] << endl;
    tulis << JRL[6*T_K[iinp]-1] << endl;
    tulis << JRL[6*T_K[iinp]] << endl;
}
tulis << "[InformasiBatang]" << endl;
for(int iinp=1;iinp<=M;iinp++)
{
    tulis << iinp << endl;
    tulis << JJ[iinp] << endl;
        tulis << JK[iinp] << endl;
    tulis << IA[iinp] << endl;
}

```

```

        if(IA[iinp] != 0)
        {
            tulis << XP[iinp] << endl;
            tulis << YP[iinp] << endl;
            tulis << ZP[iinp] << endl;
        }
    }
    tulis.close();
}

/*****
/* SUBPROGRAM UNTUK MEMASUKAN DATA DISKRIT VARIABEL DESAIN */
*****/
void input_data_diskrit()
{
    char ulang;
    int bsisi,ksisi;
    int bDIA,kDIA;
    int bNL,kNL;
    int bDIAS,kDIAS;
    int bJS,kJS;
    clrscr();

    //////////////////////////////////////
    // Menggenerasi Data Diskrit Dan Ditulis Ke File //
    //////////////////////////////////////
    do
    {
        //////////////////////////////////////
        // Menulis Ke File *.isd (Input Sisi Diskrit) //
        //////////////////////////////////////
        ofstream tulis1(fsisi);

        //////////////////////////////////////
        // Data Diskrit Lebar Balok (B) //
        //////////////////////////////////////
        cout << " Jumlah data sisi untuk lebar balok (B) = ";
        cin >> nsisi_B;
        cout << " Batas sisi terbawah (mm) = ";
        cin >> bsisi;
        cout << " Kenaikan sisi (mm) = ";
        cin >> ksisi;
        sisi_d_B[0]=bsisi;
        tulis1 << "[LebarBalok]" << endl;
        tulis1 << nsisi_B << endl;
        tulis1 << sisi_d_B[0] << endl;
        for(int itl1=1;itl1<nsisi_B;itl1++)
        {
            sisi_d_B[itl1]=sisi_d_B[itl1-1]+ksisi;
            tulis1 << sisi_d_B[itl1] << endl;
        }
        cout << endl;

        //////////////////////////////////////
        // Data Diskrit Tinggi Balok (H) //
        //////////////////////////////////////
        cout << " Jumlah data sisi untuk tinggi balok (H) = ";
        cin >> nsisi_H;
        cout << " Batas sisi terbawah (mm) = ";
        cin >> bsisi;
        cout << " Kenaikan sisi (mm) = ";
        cin >> ksisi;
        sisi_d_H[0]=bsisi;
        tulis1 << "[TinggiBalok]" << endl;
        tulis1 << nsisi_H << endl;
        tulis1 << sisi_d_H[0] << endl;
        for(int itl1=1;itl1<nsisi_H;itl1++)
        {
            sisi_d_H[itl1]=sisi_d_H[itl1-1]+ksisi;
            tulis1 << sisi_d_H[itl1] << endl;
        }
    }
}

```

```

cout << endl;

////////////////////////////////////
// Data Diskrit Sisi Kolom //
////////////////////////////////////
cout << " Jumlah data sisi untuk kolom = ";
cin >> nsisi_K;
cout << " Batas sisi terbawah (mm) = ";
cin >> bsisi;
cout << " Kenaikan sisi (mm) = ";
cin >> ksisi;
sisi_d_K[0]=bsisi;
tulis1 << "[SisiKolom]" << endl;
tulis1 << nsisi_K << endl;
tulis1 << sisi_d_K[0] << endl;
for(int itl1=1;itl1<nsisi_K;itl1++)
{
    sisi_d_K[itl1]=sisi_d_K[itl1-1]+ksisi;
    tulis1 << sisi_d_K[itl1] << endl;
}
cout << endl;

tulis1.close();

////////////////////////////////////
// Menulis Ke File *.idl (Input Diameter Lentur) //
////////////////////////////////////
ofstream tulis2(fdial);
cout << " Jumlah data diameter tulangan utama = ";
cin >> nDIA;
cout << " Batas diameter terbawah (mm) = ";
cin >> bDIA;
cout << " Kenaikan diameter (mm) = ";
cin >> kDIA;
DIA_d[0]=bDIA;
tulis2 << "[DiameterTulanganUtama]" << endl;
tulis2 << nDIA << endl;
tulis2 << DIA_d[0] << endl;
for(int itl2=1;itl2<nDIA;itl2++)
{
    DIA_d[itl2]=DIA_d[itl2-1]+kDIA;
    tulis2 << DIA_d[itl2] << endl;
}
cout << endl;
tulis2.close();

////////////////////////////////////
// Menulis Ke File *.ijl (Input Jumlah tulangan Lentur) //
////////////////////////////////////
ofstream tulis3(fjtl);
cout << " Jumlah data jumlah tulangan = ";
cin >> nNL;
cout << " Batas jumlah tulangan terbawah = ";
cin >> bNL;
cout << " Kenaikan jumlah tulangan = ";
cin >> kNL;
NL_d[0]=bNL;
tulis3 << "[JumlahTulanganUtama]" << endl;
tulis3 << nNL << endl;
tulis3 << NL_d[0] << endl;
for(int itl3=1;itl3<nNL;itl3++)
{
    NL_d[itl3]=NL_d[itl3-1]+kNL;
    tulis3 << NL_d[itl3] << endl;
}
cout << endl;
tulis3.close();

```



```

////////////////////////////////////
// Menulis Ke File *.ids (Input Jarak Senggang) //
////////////////////////////////////
ofstream tulis4(fdias);
cout << " Jumlah data diameter tulangan sengkang = ";
cin >> nDIAS;
cout << " Batas diameter sengkang terbawah (mm) = ";
cin >> bDIAS;
cout << " Kenaikan diameter sengkang (mm) = ";
cin >> kDIAS;
DIAS_d[0]=bDIAS;
tulis4 << "[DiameterTulanganSengkang]" << endl;
tulis4 << nDIAS << endl;
tulis4 << DIAS_d[0] << endl;
for(int itl4=1;itl4<nDIAS;itl4++)
{
    DIAS_d[itl4]=DIAS_d[itl4-1]+kDIAS;
    tulis4 << DIAS_d[itl4] << endl;
}
cout << endl;
tulis4.close();

////////////////////////////////////
// Menulis Ke File *.ijs (Input Jarak Sengkang) //
////////////////////////////////////
ofstream tulis5(fjts);
cout << " Jumlah data jarak sengkang = ";
cin >> nJS;
cout << " Batas jarak sengkang terbawah (mm) = ";
cin >> bJS;
cout << " Kenaikan jarak sengkang (mm) = ";
cin >> kJS;
JS_d[0]=bJS;
tulis5 << "[JarakAntarSengkang]" << endl;
tulis5 << nJS << endl;
tulis5 << JS_d[0] << endl;
for(int itl5=1;itl5<nJS;itl5++)
{
    JS_d[itl5]=JS_d[itl5-1]+kJS;
    tulis5 << JS_d[itl5] << endl;
}
cout << endl;
tulis5.close();

    cout << " Ulang (y/t)";
    cin >> ulang;
}while(ulang == 'y');
}

/*****
/* SUBPROGRAM UNTUK MEMBACA DATA DARI FILE MASUKKAN */
*****/
void baca_data()
{
    ifstream baca(finput);
    baca >> ISN;
    baca >> M;
    baca >> NJ;
    baca >> NRJ;
    baca >> NR;
    baca >> E;
    baca >> G;
    baca >> FC;
    baca >> FY;
    baca >> FYS;
    baca >> ND;
    baca >> N;
    baca >> SubName;
    for(int kinp=1;kinp<=NJ;kinp++)
    {

```

```

        baca >> kinp;
        baca >> X[kinp];
    baca >> Y[kinp];
    baca >> Z[kinp];
}
baca >> SubName;
for (int iinp=1;iinp<=NRJ;iinp++)
{
    baca >> T_K[iinp];
    baca >> JRL[6*T_K[iinp]-5];
    baca >> JRL[6*T_K[iinp]-4];
    baca >> JRL[6*T_K[iinp]-3];
    baca >> JRL[6*T_K[iinp]-2];
    baca >> JRL[6*T_K[iinp]-1];
    baca >> JRL[6*T_K[iinp]];
}
baca >> SubName;
for(int iinp=1;iinp<=M;iinp++)
{
    baca >> iinp;
    baca >> JJ[iinp];
    baca >> JK[iinp];
    baca >> IA[iinp];
    if(IA[iinp] != 0)
    {
        baca >> XP[iinp];
        baca >> YP[iinp];
        baca >> ZP[iinp];
    }
}
baca.close();

////////////////////////////////////
// Membaca Data Diskrit Sisi Dari File *.isd //
////////////////////////////////////
ifstream bacal(fsisi);

////////////////////////////////////
// Membaca Data Diskrit Lebar Balok (B) //
////////////////////////////////////
bacal >> SubName;
bacal >> nsisi_B;
for(int iinp=0;iinp<nsisi_B;iinp++)
{
    bacal >> sisi_d_B[iinp];
}

////////////////////////////////////
// Membaca Data Diskrit Tinggi Balok (H) //
////////////////////////////////////
bacal >> SubName;
bacal >> nsisi_H;
for(int iinp=0;iinp<nsisi_H;iinp++)
{
    bacal >> sisi_d_H[iinp];
}

////////////////////////////////////
// Membaca Data Diskrit Sisi Kolom //
////////////////////////////////////
bacal >> SubName;
bacal >> nsisi_K;
for(int iinp=0;iinp<nsisi_K;iinp++)
{
    bacal >> sisi_d_K[iinp];
}

bacal.close();

ifstream baca2(fdial);
baca2 >> SubName;

```

```

    baca2 >> nDIA;
    for(int iinp=0;iinp<nDIA;iinp++)
    {
        baca2 >> DIA_d[iinp];
    }
    baca2.close();

    ifstream baca3(fjtl);
    baca3 >> SubName;
    baca3 >> nNL;
    for(int iinp=0;iinp<nNL;iinp++)
    {
        baca3 >> NL_d[iinp];
    }
    baca3.close();

    ifstream baca4(fdias);
    baca4 >> SubName;
    baca4 >> nDIAS;
    for(int iinp=0;iinp<nDIAS;iinp++)
    {
        baca4 >> DIAS_d[iinp];
    }
    baca4.close();

    ifstream baca5(fjts);
    baca5 >> SubName;
    baca5 >> nJS;
    for(int iinp=0;iinp<nJS;iinp++)
    {
        baca5 >> JS_d[iinp];
    }
    baca5.close();
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN PARAMETER STRUKTURAL KE LAYAR */
*****/
void output_parameter_struktural()
{
    cout << " Struktur Portal Ruang " << ISN << "\n\n";
    cout << " Parameter Struktur\n";
    cout << " Jumlah batang : " << M << endl;
    cout << " DOF : " << N << endl;
    cout << " Jumlah joint : " << NJ << endl;
    cout << " Jumlah pengekang tumpuan : " << NR << endl;
    cout << " Jumlah titik kumpul yang dikekang : " << NRJ << endl;
    cout << " Modulus Elastisitas aksial : " << E << " N/m^2\n";
    cout << " Modulus Geser : " << G << " N/m^2\n";
    cout << " Properti Elemen Material\n";
    cout << " Kuat desak beton karakteristik : " << FC << " MPa\n";
    cout << " Kuat tarik baja tulangan : " << FY << " MPa\n";
    cout << " Kuat tarik tulangan sengkang : " << FYS << " MPa\n";
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN KOORDINAT STRUKTUR KE LAYAR */
*****/
void output_koordinat_titik_kumpul()
{
    cout << "\n Koordinat Titik Kumpul (m)\n";
    cout << " Titik      X          Y          Z \n";
    for(int kout=1;kout<=NJ;kout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(10) << kout
            << setw(14) << X[kout]
            << setw(14) << Y[kout]
            << setw(14) << Z[kout] << endl;
    }
}

```

```

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN INFORMASI BATANG KE LAYAR */
*****/
void output_informasi_batang()
{
    cout << "\n Informasi Batang \n";
    cout << " Batang JJ JK IA\n";
    for(int iout=1;iout<=M;iout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(8) << iout
            << setw(8) << JJ[iout]
            << setw(8) << JK[iout]
            << setw(3) << IA[iout] << endl;

        if(IA[iout] != 0)
        {
            cout << setiosflags(ios::left);
            cout << " XP = " << setw(12) << XP[iout];
            cout << " YP = " << setw(12) << YP[iout] << '\t';
            cout << " ZP = " << setw(12) << ZP[iout] << endl;
            cout << endl;
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN PENGEKANG TITIK KUMPUL KE LAYAR */
*****/
void output_pengekang_titik_kumpul()
{
    cout << "\n Pengekang Titik Kumpul \n";
    cout << " Titik JR1 JR2 JR3 JR4 JR5 JR6\n";
    for (int iout=1;iout<=NRJ;iout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(10) << T_K[iout]
            << setw(6) << JRL[6*T_K[iout]-5]
            << setw(6) << JRL[6*T_K[iout]-4]
            << setw(6) << JRL[6*T_K[iout]-3]
            << setw(6) << JRL[6*T_K[iout]-2]
            << setw(6) << JRL[6*T_K[iout]-1]
            << setw(6) << JRL[6*T_K[iout]] << endl;
    }
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN GAYA UJUNG BATANG KE LAYAR */
*****/
void output_beban_batang()
{
    cout << "\n Gaya di Ujung Batang Terkekang Akibat Beban (Nm)\n"
    << " Batang AML1 AML2 AML3 "
    << " AML4 AML5 AML6\n"
    << " AML7 AML8 AML9 "
    << " AML10 AML11 AML12\n";
    for(int kout=1;kout<=M;kout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(8) << kout
            << setw(12) << AML[1][kout]
            << setw(12) << AML[2][kout]
            << setw(12) << AML[3][kout]
            << setw(12) << AML[4][kout]
            << setw(12) << AML[5][kout]
            << setw(12) << AML[6][kout] << endl;
        cout << " "
            << setw(12) << AML[7][kout]
            << setw(12) << AML[8][kout]
            << setw(12) << AML[9][kout]
            << setw(12) << AML[10][kout]

```

```

        << setw(12) << AML[11][kout]
        << setw(12) << AML[12][kout] << endl;
    }
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN BEBAN TITIK KE LAYAR */
*****/
void output_beban_titik()
{
    cout << "\n  Beban Titik (N)\n"
    << "   Titik   Arah 1       Arah 2       Arah 3       "
    << "Arah 4       Arah 5       Arah 6\n";
    for(int kout=1;kout<=NJ;kout++)
    {
        cout << setiosflags(ios::left);
        cout << "   " << setw(8) << kout
        << setw(12) << AJ[((6*kout)-5)]
        << setw(12) << AJ[((6*kout)-4)]
        << setw(12) << AJ[((6*kout)-3)]
        << setw(12) << AJ[((6*kout)-2)]
        << setw(12) << AJ[((6*kout)-1)]
        << setw(12) << AJ[(6*kout)] << endl;
    }
}

```

```

/*****
/* FILE : STRUKTUR.HPP
/* KUMPULAN SUBPROGRAM UNTUK MENGHITUNG STRUKTUR
/* DIKEMBANGKAN DARI WEAVER & GERE
*****/

/*****
/* SUBPROGRAM MENGHITUNG GAYA DAN PERPINDAHAN PADA STRUKTUR */
*****/
void struktur()
{
    kekakuan_batang();
    banfac(N,NB,SFF);
    beban();
    bansol(N,NB,SFF,AC,DF);
    hasil();
}

/*****
/* SUBPROGRAM MENGHITUNG INERSIA */
*****/
void inersia()
{
    js_balok=0;
    js_kolom=0;
    for (int iin=1;iin<=M;iin++)
    {
        ///////////////////////////////////
        // Memeriksa Kemiringan Batang //
        ///////////////////////////////////
        periksa_batang(iin);

        ///////////////////////////////////
        // Batang Non Vertikal //
        ///////////////////////////////////
        if(CXZ>0.001)
        {
            // (mm) -> (m)
            b[iin]=isi(var_b[no_struktur][0+(12*js_balok)],sisi_d_B)/1000.;
            h[iin]=isi(var_b[no_struktur][1+(12*js_balok)],sisi_d_H)/1000.;
            js_balok++;
        }

        ///////////////////////////////////
        // Batang Vertikal //
        ///////////////////////////////////
        else
        {
            // (mm) -> (m)
            b[iin]=isi(var_k[no_struktur][0+(5*js_kolom)],sisi_d_K)/1000.;
            h[iin]=b[iin];
            js_kolom++;
        }

        ///////////////////////////////////
        // Menghitung Luas (m^2) //
        ///////////////////////////////////
        AX[iin]=b[iin]*h[iin];

        ///////////////////////////////////
        // Menghitung Konstanta Puntir //
        ///////////////////////////////////
        if(b[iin]<=h[iin])
        {
            XI[iin]=(((1./3.)-(0.21*b[iin]/h[iin]*
                ((1.-(pow(b[iin],4)/(12.*(pow(h[iin],4))))
                )))*h[iin])*(pow(b[iin],3)));
        }
        else
        {

```

```

// Formula Diambil Dari Buku Roark's Formulas for Stress & Strain //
// Warren C. Young 1989 halaman 348 //
// Formula Diambil Dari Buku Roark's Formulas for Stress & Strain //
// Warren C. Young 1989 halaman 348 //
float _a=0.5*b[iin];
float _b=0.5*h[iin];
XI[iin]=_a*(pow(_b,3))*
(
(16./3.)-(3.36*_b/_a)*(1.-(pow(_b,4))/(12.*pow(_a,4)))
);
}

// Momen Inersia //
// Formula Diambil Dari Buku Roark's Formulas for Stress & Strain //
// Warren C. Young 1989 halaman 348 //
YI[iin]=h[iin]*(pow(b[iin],3))/12.;
ZI[iin]=b[iin]*(pow(h[iin],3))/12.;
}
}

/*****
/* SUBPROGRAM MERAKIT MATRIK KEKAKUAN STRUKTUR */
*****/
void kekakuan_batang()
{
// Inisialisasi Awal //
NB=0.;
IR=0;
IC=0;
for (int ir=1;ir<=M;ir++)
{
R11[ir]=0.;R12[ir]=0.;R13[ir]=0.;
R21[ir]=0.;R22[ir]=0.;R23[ir]=0.;
R31[ir]=0.;R32[ir]=0.;R33[ir]=0.;
}

// Membentuk Matrik Rotasi //
for (int ir=1;ir<=M;ir++)
{
if ((6*(abs(JK[ir]-JJ[ir])+1))>NB)
{
NB=(6*(abs(JK[ir]-JJ[ir])+1));
}
periksa_batang(ir);
if(IA[ir]!=0)
{
XPS=XP[ir]-(X[JJ[ir]]);
YPS=YP[ir]-(Y[JJ[ir]]);
ZPS=ZP[ir]-(Z[JJ[ir]]);
}

// Membentuk Matrik Rotasi Untuk Batang Vertikal //
if(CXZ<=0.001)
{
R11[ir]=0.; R12[ir]=Cy; R13[ir]=0.;
R21[ir]=(-Cy); R22[ir]=0.; R23[ir]=0.;
R31[ir]=0.; R32[ir]=0.; R33[ir]=1.;
if(IA[ir]==0)
{
continue;
}
COSA=(-XPS*Cy)/(sqrt(XPS*XPS+ZPS*ZPS));
SINA=(-ZPS)/(sqrt(XPS*XPS+ZPS*ZPS));
R21[ir]=(-Cy*COSA);
R23[ir]=SINA;
}
}

```

[illegible]


```

        {
            for(int kr=((3*jr)-2);kr<=12;kr++)
            {
                SMS[((3*jr)-2)][kr]=R11[ir]*SMRT[((3*jr)-2)][kr]
                +R21[ir]*SMRT[((3*jr)-1)][kr]
                +R31[ir]*SMRT[(3*jr)][kr];
                SMS[((3*jr)-1)][kr]=R12[ir]*SMRT[((3*jr)-2)][kr]
                +R22[ir]*SMRT[((3*jr)-1)][kr]
                +R32[ir]*SMRT[(3*jr)][kr];
                SMS[(3*jr)][kr]=R13[ir]*SMRT[((3*jr)-2)][kr]
                +R23[ir]*SMRT[((3*jr)-1)][kr]
                +R33[ir]*SMRT[(3*jr)][kr];
            }
        }

//////////
// Mengisi Vektor Indeks Perpindahan Batang //
//////////
    indeks_batang(ir);

//////////
// Mengambil Matrik Kekakuan //
// Untuk Perpindahan Titik Kumpul Yang Bebas (SFF) //
//////////
    for(int jr=1;jr<=MD;jr++)
    {
        if(JRL[IM[jr]]==0)
        {
            for(int kr=jr;kr<=MD;kr++)
            {
                if(JRL[IM[kr]]==0)
                {
                    IR=ID[IM[jr]];
                    IC=ID[IM[kr]];
                    if(IR>=IC)
                    {
                        int ITEM=IR;
                        IR=IC;
                        IC=ITEM;
                    }
                    IC=(IC-IR+1);
                    SFF[IR][IC]=(SFF[IR][IC])+(SMS[jr][kr]);
                }
            }
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENGOLAH HASIL HITUNGAN STRUKTUR */
*****/
void hasil()
{
    //////////
    // Inisialisasi Vektor Perpindahan = 0 //
    //////////
    int ihs=N+1;
    for(int jhs=1;jhs<=ND;jhs++)
    {
        if(JRL[(ND-jhs+1)]==0)
        {
            ihs=(ihs-1);
            DJ[(ND-jhs+1)]=0.;
        }
        else
        {
            DJ[(ND-jhs+1)]=0.;
        }
    }
}

```

```

////////////////////////////////////
// Mengisi Vektor Perpindahan Semua Titik Kumpul //
// Dalam Arah Sumbu Global //
////////////////////////////////////
ihs=N+1;
for(int jhs=1;jhs<=ND;jhs++)
{
    if(JRL[(ND-jhs+1)]==0)
    {
        ihs=(ihs-1);
        DJ[(ND-jhs+1)]=DF[ihs];
    }
    else
    {
        DJ[(ND-jhs+1)]=0.;
    }
}

////////////////////////////////////
// Gaya Pada Struktur //
////////////////////////////////////
for(int ih=1;ih<=M;ih++)
{
    periksa_batang(ih);

    //////////////////////////////////////
    // Mengisi Matrik Kekakuan Batang Pada Arah Sumbu Lokal //
    //////////////////////////////////////
    isi_matrik_kekakuan(ih);

    //////////////////////////////////////
    // Mengisi Vektor Indeks Perpindahan Batang //
    //////////////////////////////////////
    indeks_batang(ih);

    //////////////////////////////////////
    // Mengisi Vektor Gaya Ujung Batang Akhir //
    //////////////////////////////////////
    for(int jh=1;jh<=MD;jh++)
    {
        //////////////////////////////////////
        // Inisialisasi Gaya Ujung Batang Akibat Perpindahan = 0 //
        //////////////////////////////////////
        AMD[jh]=0.;

        //////////////////////////////////////
        // Mengisi Vektor Gaya Di Ujung Batang //
        // Akibat Perpindahan Titik Kumpul //
        //////////////////////////////////////
        for(int kh=1;kh<=MD;kh++)
        {
            AMD[jh]=AMD[jh]+SMRT[jh][kh]*DJ[IM[kh]];
        }
        AM[ih][jh]=AML[jh][ih]+AMD[jh];
    }

    periksa_batang(ih);

    //////////////////////////////////////
    // Menyimpan Gaya Untuk Perhitungan Balok //
    //////////////////////////////////////
    if(CXZ>0.001)
    {
        //////////////////////////////////////
        // Momen di tengah bentang (Nm) //
        //////////////////////////////////////
        MLAP[ih]=(-AM[ih][6])
            +(0.125*W[ih]*pow(EL[ih],2));

        //////////////////////////////////////
        // Momen Pada Tumpuan (Nm) //
    }
}

```

```

////////////////////////////////////
MTUM_KI[ih]=(-AM[ih][6]);
MTUM_KA[ih]=AM[ih][12];

////////////////////////////////////
// Gaya Geser (N) //
////////////////////////////////////
GESER_KI[ih]=AM[ih][2];
GESER_KA[ih]=(-AM[ih][8]);
}

////////////////////////////////////
// Menyimpan Gaya Untuk Perhitungan Kolom Biaksial //
////////////////////////////////////
else
{
    //////////////////////////////////////
    // Gaya Aksial Dan Momen (N) //
    //////////////////////////////////////
    PK[ih]=fabs(AM[ih][7]);
    MKX[ih]=fabs(AM[ih][10]);
    MKY[ih]=fabs(AM[ih][12]);

    //////////////////////////////////////
    // Gaya Geser (N) //
    //////////////////////////////////////
    if(fabs(AM[ih][8])>fabs(AM[ih][9]))
    {
        GK[ih]=fabs(AM[ih][8]);
    }
    else
    {
        GK[ih]=fabs(AM[ih][9]);
    }
}

////////////////////////////////////
// Inisialisasi Vektor Reaksi Tumpuan //
////////////////////////////////////
for(int jh=1; jh<=4; jh++)
{
    if(JRL[IM[(3*jh)-2]]==1)
    {
        AR[IM[(3*jh)-2]]=0.;
    }
    if(JRL[IM[(3*jh)-1]]==1)
    {
        AR[IM[(3*jh)-1]]=0.;
    }
    if(JRL[IM[(3*jh)]]==1)
    {
        AR[IM[(3*jh)]]=0.;
    }
}

////////////////////////////////////
// Reaksi Tumpuan //
////////////////////////////////////
for(int jh=1; jh<=4; jh++)
{
    int J1=((3*jh)-2),J2=((3*jh)-1),J3=(3*jh);
    int I1=IM[J1],I2=IM[J2],I3=IM[J3];
    if(JRL[I1]==1)
    {
        AR[I1]=AR[I1]
        +R11[ih]*AMD[J1]
        +R21[ih]*AMD[J2]
        +R31[ih]*AMD[J3];
    }
    if(JRL[I2]==1)
    {

```

```

        AR[I2]=AR[I2]
        +R12[ih]*AMD[J1]
        +R22[ih]*AMD[J2]
        +R32[ih]*AMD[J3];
    }
    if(JRL[I3]==1)
    {
        AR[I3]=AR[I3]
        +R13[ih]*AMD[J1]
        +R23[ih]*AMD[J2]
        +R33[ih]*AMD[J3];
    }
}

// Menghitung Reaksi Tumpuan Untuk Titik Terkekang //
// Menghitung Reaksi Tumpuan Untuk Titik Terkekang //
for(int ih=1; ih<=ND; ih++)
{
    if(JRL[ih]==0)
    {
        continue;
    }
    AR[ih]=AR[ih]-AJ[ih]-AE[ih];
}

/*****/
/* SUBPROGRAM UNTUK MEMERIKSA KEMIRINGAN BATANG */
/*****/
void periksa_batang(int perb)
{
    EL[perb] = sqrt(
        (pow(((X[JK[perb]])-(X[JJ[perb]])),2))
        +(pow(((Y[JK[perb]])-(Y[JJ[perb]])),2))
        +(pow(((Z[JK[perb]])-(Z[JJ[perb]])),2)));
    CX=((X[JK[perb]])-(X[JJ[perb]]))/EL[perb];
    Cy=((Y[JK[perb]])-(Y[JJ[perb]]))/EL[perb];
    CZ=((Z[JK[perb]])-(Z[JJ[perb]]))/EL[perb];
    CXZ = fabs(sqrt(CX*CX + CZ*CZ));
}

/*****/
/* SUBPROGRAM UNTUK MENGHITUNG VEKTOR INDEKS PERPINDAHAN BATANG */
/*****/
void indeks_batang(int ib)
{
    IM[1] = ((6.*JJ[ib])-5.);    IM[2] = ((6.*JJ[ib])-4.);
    IM[3] = ((6.*JJ[ib])-3.);    IM[4] = ((6.*JJ[ib])-2.);
    IM[5] = ((6.*JJ[ib])-1.);    IM[6] = (6.*JJ[ib]);
    IM[7] = ((6.*JK[ib])-5.);    IM[8] = ((6.*JK[ib])-4.);
    IM[9] = ((6.*JK[ib])-3.);    IM[10] = ((6.*JK[ib])-2.);
    IM[11] = ((6.*JK[ib])-1.);    IM[12] = (6.*JK[ib]);
}

/*****/
/* SUBPROGRAM UNTUK MENGISI MATRIK KEKAKUAN BATANG PADA SUMBU LOKAL */
/*****/
void isi_matrik_kekakuan(int imk)
{
    // Mengosongkan Matrik Kekakuan Batang Lokal //
    for(int is=1; is<=12; is++)
    {
        for(int js=1; js<=12; js++)
        {
            SM[is][js]=0.;
        }
    }
}

```

```

////////////////////////////////////
// Menghitung Delapan Konstanta //
////////////////////////////////////
float SCM1A=E*AX[imk]/EL[imk];
float SCM1B=G*XI[imk]/EL[imk];
float SCM2Y=4.0*E*YI[imk]/EL[imk];
float SCM3Y=1.5*SCM2Y/EL[imk];
float SCM4Y=2.0*SCM3Y/EL[imk];
float SCM2Z=4.0*E*ZI[imk]/EL[imk];
float SCM3Z=1.5*SCM2Z/EL[imk];
float SCM4Z=2.0*SCM3Z/EL[imk];

////////////////////////////////////
// Mengisi Segitiga Atas Matrik Kekakuan Batang //
////////////////////////////////////
SM[1][1]=SCM1A; SM[1][7]=(-SCM1A);
SM[2][2]=SCM4Z; SM[2][6]=SCM3Z;
SM[2][8]=(-SCM4Z); SM[2][12]=SCM3Z;
SM[3][3]=SCM4Y; SM[3][5]=(-SCM3Y);
SM[3][9]=(-SCM4Y); SM[3][11]=(-SCM3Y);
SM[4][4]=SCM1B; SM[4][10]=(-SCM1B);
SM[5][5]=SCM2Y; SM[5][9]=SCM3Y;
SM[5][11]=SCM2Y/2.0; SM[6][6]=SCM2Z;
SM[6][8]=(-SCM3Z); SM[6][12]=SCM2Z/2.0;
SM[7][7]=SCM1A; SM[8][8]=SCM4Z;
SM[8][12]=(-SCM3Z); SM[9][9]=SCM4Y;
SM[9][11]=SCM3Y; SM[10][10]=SCM1B;
SM[11][11]=SCM2Y; SM[12][12]=SCM2Z;

////////////////////////////////////
// Mengisi Segitiga Bawah Matrik Yang Simetri //
////////////////////////////////////
for(int is=1;is<=11;is++)
{
    for(int js=is+1;js<=12;js++)
    {
        SM[js][is]=SM[is][js];
    }
}

////////////////////////////////////
// Mengalikan Matrik SM Dengan Matrik Rotasi //
////////////////////////////////////
for(int is=1;is<=4;is++)
{
    for(int js=1;js<=12;js++)
    {
        SMRT[js][((3*is)-2)]=SM[js][((3*is)-2)]*R11[imk]
        +SM[js][((3*is)-1)]*R21[imk]
        +SM[js][(3*is)]*R31[imk];
        SMRT[js][((3*is)-1)]=SM[js][((3*is)-2)]*R12[imk]
        +SM[js][((3*is)-1)]*R22[imk]
        +SM[js][(3*is)]*R32[imk];
        SMRT[js][(3*is)]=SM[js][((3*is)-2)]*R13[imk]
        +SM[js][((3*is)-1)]*R23[imk]
        +SM[js][(3*is)]*R33[imk];
    }
}
}

```

```

/*****
/* FILE : PEMBEBANAN.HPP
/* PUSTAKA FUNGSI MENGOLAH DATA PEMBEBANAN KE FILE
*****/

/*****
/* SUBPROGRAM MEMBACA DATA BEBAN */
*****/
void load_data()
{
    int ibn=0;
    char uji;
    cout << " Nama file struktur yang akan dikenai beban"
         << " (tanpa ekstensi) = ";
    cin >> fi;
    strcpy(finput,fi);
    strcat(finput, ".inp");
    strcpy(fbeban,fi);
    strcat(fbeban, ".bbn");
    baca_data();
    baca_beban();
    cout << "Ingin mengosongkan data beban yang ada (y/t) ? ";
    cin >> uji;
    if (uji!= 'y' )
    {
        kosong_beban();
    }

    ////////////
    // Beban Pada Batang //
    ////////////
    cout << " Inputkan beban merata pada batang,"
         << " pilih batang = 0 untuk keluar\n";
    do
    {
        cout << " Batang = ";
        cin >> ibn;
        if(ibn==0)
        { break; }
        periksa_batang(ibn);
        cout << " Panjang = " << EL[ibn] << endl;
        cout << " Beban merata (N/m) = "; cin >> W[ibn];
        AML[1][ibn]=0.;
        AML[2][ibn]=W[ibn]*EL[ibn]/2.;
        AML[3][ibn]=0.;
        AML[4][ibn]=0.;
        AML[5][ibn]=0.;
        AML[6][ibn]=W[ibn]*pow(EL[ibn],2)/12.;
        AML[7][ibn]=0.;
        AML[8][ibn]=W[ibn]*EL[ibn]/2.;
        AML[9][ibn]=0.;
        AML[10][ibn]=0.;
        AML[11][ibn]=0.;
        AML[12][ibn]=(-W[ibn])*pow(EL[ibn],2)/12.;
        output_beban_batang();
    } while (ibn!=0);
    ofstream tulis(fbeban);
    tulis << "[BebanBatang]" << endl;
    tulis << M << endl;
    for(int kbn=1;kbn<=M;kbn++)
    {
        tulis << kbn << endl;
        tulis << W[kbn] << endl;
        tulis << AML[1][kbn] << endl;
        tulis << AML[2][kbn] << endl;
        tulis << AML[3][kbn] << endl;
        tulis << AML[4][kbn] << endl;
        tulis << AML[5][kbn] << endl;
        tulis << AML[6][kbn] << endl;
        tulis << AML[7][kbn] << endl;
        tulis << AML[8][kbn] << endl;
    }
}

```

```

        tulis << AML[9][kbn] << endl;
        tulis << AML[10][kbn] << endl;
        tulis << AML[11][kbn] << endl;
        tulis << AML[12][kbn] << endl;
    }

    ////////////////////////////////////
    // Beban Pada Titik //
    ////////////////////////////////////
    ibn=0;
    cout << "    Inputkan beban pada titik kumpul,"
    << " pilih titik = 0 untuk keluar\n";
    do
    {
        cout << "    Titik = ";
        cin >> ibn;
        if(ibn==0)
        {
            break;
        }
        cout << "    Aksi pada arah 1 global = "; cin >> AJ[((6*ibn)-5)];
        cout << "    Aksi pada arah 2 global = "; cin >> AJ[((6*ibn)-4)];
        cout << "    Aksi pada arah 3 global = "; cin >> AJ[((6*ibn)-3)];
        cout << "    Aksi pada arah 4 global = "; cin >> AJ[((6*ibn)-2)];
        cout << "    Aksi pada arah 5 global = "; cin >> AJ[((6*ibn)-1)];
        cout << "    Aksi pada arah 6 global = "; cin >> AJ[(6*ibn)];
        output_beban_titik();
    } while (ibn!=0);
    tulis << "[BebanTitik]" << endl;
    tulis << NJ << endl;
    for(int kbn=1;kbn<=NJ;kbn++)
    {
        tulis << kbn << endl;
        tulis << AJ[((6*kbn)-5)] << endl;
        tulis << AJ[((6*kbn)-4)] << endl;
        tulis << AJ[((6*kbn)-3)] << endl;
        tulis << AJ[((6*kbn)-2)] << endl;
        tulis << AJ[((6*kbn)-1)] << endl;
        tulis << AJ[(6*kbn)] << endl;
    }
    tulis.close();
}

void kosong_beban()
{
    ////////////////////////////////////
    // Mengosongkan Beban Pada Batang //
    ////////////////////////////////////
    int ibn=0;
    do
    {
        W[ibn]=0;
        AML[1][ibn]=0.;
        AML[2][ibn]=0.;
        AML[3][ibn]=0.;
        AML[4][ibn]=0.;
        AML[5][ibn]=0.;
        AML[6][ibn]=0.;
        AML[7][ibn]=0.;
        AML[8][ibn]=0.;
        AML[9][ibn]=0.;
        AML[10][ibn]=0.;
        AML[11][ibn]=0.;
        AML[12][ibn]=0.;
        ibn++;
    } while (ibn<=M);

    ////////////////////////////////////
    // Mengosongkan Beban Pada Titik Kumpul //
    ////////////////////////////////////
    ibn=0;

```

```

do
{
    AJ[ ((6*ibn)-5) ]=0.;
    AJ[ ((6*ibn)-4) ]=0.;
    AJ[ ((6*ibn)-3) ]=0.;
    AJ[ ((6*ibn)-2) ]=0.;
    AJ[ ((6*ibn)-1) ]=0.;
    AJ[ (6*ibn) ]=0.;
    ibn++;
} while(ibn<=NJ);
}

/*****/
/* SUBPROGRAM UNTUK MEMBACA DATA PEMBEBANAN DARI FILE */
/*****/
void baca_beban()
{
    ifstream baca(fbeban);

    ///////////////////////////////////
    // Membaca Beban Batang //
    ///////////////////////////////////
    baca >> SubName;
    baca >> M;
    for(int kinp=1;kinp<=M;kinp++)
    {
        baca >> kinp;
        baca >> W[kinp];
        baca >> AML[1][kinp];
        baca >> AML[2][kinp];
        baca >> AML[3][kinp];
        baca >> AML[4][kinp];
        baca >> AML[5][kinp];
        baca >> AML[6][kinp];
        baca >> AML[7][kinp];
        baca >> AML[8][kinp];
        baca >> AML[9][kinp];
        baca >> AML[10][kinp];
        baca >> AML[11][kinp];
        baca >> AML[12][kinp];
    }

    ///////////////////////////////////
    // Membaca Beban Titik //
    ///////////////////////////////////
    baca >> SubName;
    baca >> NJ;
    for(int kinp=1;kinp<=NJ;kinp++)
    {
        baca >> kinp;
        baca >> AJ[ ((6*kinp)-5) ];
        baca >> AJ[ ((6*kinp)-4) ];
        baca >> AJ[ ((6*kinp)-3) ];
        baca >> AJ[ ((6*kinp)-2) ];
        baca >> AJ[ ((6*kinp)-1) ];
        baca >> AJ[ (6*kinp) ];
    }
    baca.close();
    berat_sendiri();
}

/*****/
/* SUBPROGRAM UNTUK MERAKIT VEKTOR BEBAN */
/*****/
void beban()
{
    ///////////////////////////////////
    // Mengosongkan Vektor Beban Titik Kumpul Gabungan //
    ///////////////////////////////////
    for(int ib=1;ib<=ND;ib++)
    {

```



```

        AC[ID[ib]]=0.;
    }

    //////////////////////////////////////
    // Mengosongkan Vektor Beban Titik Kumpul Ekvivalen //
    //////////////////////////////////////
    for (int ib=1;ib<=M;ib++)
    {
        periksa_batang(ib);
        if(CXZ<=0.001)
        {
            continue;
        }
        indeks_batang(ib);
        for(int jb=1;jb<=4;jb++)
        {
            AE[IM[(3*jb-2)]] = 0.;
            AE[IM[(3*jb-1)]] = 0.;
            AE[IM[(3*jb)]] = 0.;
        }
    }

    //////////////////////////////////////
    // Beban Titik Kumpul Ekvivalen //
    //////////////////////////////////////
    for (int ib=1;ib<=M;ib++)
    {
        periksa_batang(ib);
        if(CXZ<=0.001)
        {
            continue;
        }
        indeks_batang(ib);
        for(int jb=1;jb<=4;jb++)
        {
            AE[IM[(3*jb-2)]] = AE[IM[(3*jb-2)]]
                -R11[ib]*AML[(3*jb-2)][ib]
                -R21[ib]*AML[(3*jb-1)][ib]
                -R31[ib]*AML[(3*jb)][ib];
            AE[IM[(3*jb-1)]] = AE[IM[(3*jb-1)]]
                -R12[ib]*AML[(3*jb-2)][ib]
                -R22[ib]*AML[(3*jb-1)][ib]
                -R32[ib]*AML[(3*jb)][ib];
            AE[IM[(3*jb)]] = AE[IM[(3*jb)]]
                -R13[ib]*AML[(3*jb-2)][ib]
                -R23[ib]*AML[(3*jb-1)][ib]
                -R33[ib]*AML[(3*jb)][ib];
        }
    }

    //////////////////////////////////////
    // Beban Titik Kumpul Gabungan //
    //////////////////////////////////////
    for(int ib=1;ib<=ND;ib++)
    {
        AC[ID[ib]] = AJ[ib] + AE[ib];
    }
}

/*****
/* SUBPRROGRAM MENGHITUNG BERAT SENDIRI BALOK */
*****/
void berat_sendiri()
{
    js_balok=0;
    js_kolom=0;
    for(int ibs=1;ibs<=M;ibs++)
    {
        //////////////////////////////////////
        // Memeriksa Kemiringan Batang //
        //////////////////////////////////////
    }
}

```

```

periksa_batang(ibs);

//////////
// Batang Non Vertikal //
//////////
if(CXZ>0.001)
{
    periksa_batang(ibs);
    // (mm) -> (m)
    b[ibs]=isi(var_b[no_struktur][0+(12*js_balok)],sisi_d_B);
    h[ibs]=isi(var_b[no_struktur][1+(12*js_balok)],sisi_d_H);
    W_Balok[ibs]=24000.*b[ibs]*h[ibs]*1.E-6; // (N/m)
    W[ibs]+=W_Balok[ibs]; // (N/m)
    AML[2][ibs]+=W_Balok[ibs]*EL[ibs]/2.;
    AML[6][ibs]+=W_Balok[ibs]*pow(EL[ibs],2)/12.;
    AML[8][ibs]+=W_Balok[ibs]*EL[ibs]/2.;
    AML[12][ibs]+=(-W_Balok[ibs])*pow(EL[ibs],2)/12.;
    js_balok++;
}

//////////
// Batang Vertikal //
//////////
else
{
    periksa_batang(ibs);
    b[ibs]=isi(var_k[no_struktur][0+(5*js_kolom)],sisi_d_K);
    h[ibs]=b[ibs];
    P_Kolom[ibs]=(-24000.)*b[ibs]*h[ibs]*EL[ibs]*1.E-6; // (N)

    AJ[((6*JJ[ibs])-4)]+=P_Kolom[ibs];

    js_kolom++;
}
}
}

```

```

/*****
/* FILE : SOLVER.HPP
/* SUBPROGRAM UNTUK MENYELESAIKAN PERSAMAAN
/* DENGAN METODA CHOLESKI YANG DIMODIFIKASI
*****/

/*****
/* SUBPROGRAM UNTUK FAKTORISASI MATRIK SIMETRIS BERJALUR */
/* DENGAN PENDEKATAN CHOLESKI YANG DIMODIFIKASI
*****/
void banfac(int N,int NB,float A_SFF[][mak])
{
    int J2;
    float SUM;
    float TEMP;
    if((A_SFF[1][1])>0)
    {
        for(int jbf=2;jbf<=N;jbf++)
        {
            J2=(jbf-NB+1);
            if(J2<1)
            {
                J2=1;
            }
            if((jbf-1)!=1)
            {
                for(int ibf=2;ibf<=(jbf-1);ibf++)
                {
                    if((ibf-1)>=J2)
                    {
                        SUM=A_SFF[ibf][jbf-ibf+1];
                        for(int kbf=J2;kbf<=(ibf-1);kbf++)
                        {
                            SUM=SUM-A_SFF[kbf][ibf-kbf+1]
                                *A_SFF[kbf][jbf-kbf+1];
                        }
                        A_SFF[ibf][jbf-ibf+1]=SUM;
                    }
                    else
                    {
                        continue;
                    }
                }
            }
            SUM=A_SFF[jbf][1];
            for(int kbf=J2;kbf<=(jbf-1);kbf++)
            {
                TEMP=A_SFF[kbf][jbf-kbf+1]/A_SFF[kbf][1];
                SUM=SUM-TEMP*A_SFF[kbf][jbf-kbf+1];
                A_SFF[kbf][jbf-kbf+1]=TEMP;
            }
            if(SUM<=0)
            {
                break;
            }
            A_SFF[jbf][1]=SUM;
        }
    }
    else
    {
        clrscr();
        cout << "Tidak positif tentu !!!\n";
        cout << "Struktur tak stabil - GAGAL dibangkitkan !!\n\n";
        cout << "Periksa data struktur anda \n";
        cout << "Tekan sembarang tombol, program akan dihentikan \n";
        getch();
        exit(1);
    }
}

```

```

/*****
/* SUBPROGRAM MENGOLAH MATRIK BERJALUR DARI SUBPROGRAM BANFAC() */
*****/
void bansol(int N,int NB,float U_SFF[][mak]
            ,float B_AC[mak],float X_DF[mak])
{
    float SUM;

    //////////////////////////////////////
    // Inisialisasi Perpindahan Titik Kumpul Bebas = 0 //
    //////////////////////////////////////
    for(int ins=1;ins<=N;ins++)
    {
        X_DF[ins]=0.;
    }

    for(int ibsl=1;ibsl<=N;ibsl++)
    {
        int jbns=(ibsl-NB+1);
        if (ibsl<=NB)
        {
            jbns=1;
        }
        SUM=B_AC[ibsl];
        if(jbns<=(ibsl-1))
        {
            for(int kbns=jbns;kbns<=(ibsl-1);kbns++)
            {
                SUM=(SUM-((U_SFF[kbns][ibsl-kbns+1])*(X_DF[kbns])));
            }
            X_DF[ibsl] = SUM;
        }

        for(int ibsl=1;ibsl<=N;ibsl++)
        {
            X_DF[ibsl]=((X_DF[ibsl])/(U_SFF[ibsl][1]));
        }

        for(int I1=1;I1<=N;I1++)
        {
            int ibsl=(N-I1+1);
            int jbns=(ibsl+NB-1);
            if(jbns>N)
            {
                jbns=N;
            }
            SUM=X_DF[ibsl];

            if((ibsl+1)<=jbns)
            {
                for(int kbns=(ibsl+1);kbns<=jbns;kbns++)
                {
                    SUM=(SUM-(U_SFF[ibsl][kbns-ibsl+1]*X_DF[kbns]));
                }
            }
            X_DF[ibsl] = SUM;
        }
    }
}

```

```

/*****
/* FILE : KOLOM.HPP
/* KELAS UNTUK MENGHITUNG HARGA DAN KENDALA PADA KOLOM
*****/
class kolom
{
private:

    /*****/
    /* PENDEKLARASIAN VARIABEL PRIVATE PADA KELAS KOLOM */
    /*****/

    ////////////
    // Perhitungan Kolom //
    ////////////
    float RHO; // Rasio penulangan
    float PN; // Gaya aksial nominal
    float MNX; // Momen nominal arah x
    float MNY; // Momen nominal arah y
    float MOX; // Momen ekuivalen arah x
    float MOY; // Momen ekuivalen arah y
    float PO; // Batas gaya aksial karena eksentrisitas minimum (N)
    float PNCoba; // Gaya aksial nominal coba
    float PNB; // Gaya aksial nominal pada kondisi balance
    float MNCoba; // Momen nominal coba
    float beta; // Faktor bentuk biaksial
    float ASTOT; // Luas besi total pada penampang (mm^2)
    float CB; // Jarak serat tekan terluar ke garis netral
    // pada keadaan balance (mm)
    float FS; // Tegangan dalam tulangan pada beban kerja
    float eks; // Eksentrisitas (mm)
    float eksCoba; // Eksentrisitas maksimum (mm)
    float eksb; // Eksentrisitas pada keadaan balance (mm)
    float epsb; // Regangan baja pada keadaan balance (mm)
    float epsy; // Regangan baja pada keadaan luluh (mm)

    float rasio_kelangsingan; // Rasio kelangsingan

    float fs1b; // Tegangan baja tulangan pada keadaan balance (MPa)
    float fs1; // Tegangan baja (MPa)
    float di; // Jarak tulangan ke garis netral (mm)
    float Fsi; // Gaya Yang Disumbangkan Oleh Tulangan (N)
    float ASdi; // Luas tulangan pada jarak di
    float fsf1;
    float dll;

    ////////////
    // Variabel Pembantu Untuk Menyelesaikan Persamaan Kuadrat //
    ////////////
    float asol;
    float bsol;
    float csol;
    float dsol;

    ////////////
    // Variabel Pada Metoda False Posisi //
    ////////////
    int iterasi_fp; // Jumlah iterasi dengan metode false posisi

    float trialKiri; // Titik coba pertama
    float trialKanan; // Titik coba kedua
    float trialBaru; // Titik coba baru
    float deki; // Nilai titik coba pertama
    float deka; // Nilai titik coba kedua
    float deba; // Nilai titik coba baru

    /*****/
    /* PROTOTYPE FUNGSI PADA KELAS KOLOM */
    /*****/
    void rho();
    void jarak_tulangan();

```

```

void kelangsingan();
void analisa();
void sengkang_kolom();
float hitung_kolom (float Ccoba);

public:

/*****
/* CONSTRUCTOR CLASS KOLOM */
*****/
    kolom()
    {
        ////////////////////////////////////
        // Inisialisasi Nilai Awal //
        ////////////////////////////////////
        kendala=0.;
        harga=0.;

        ////////////////////////////////////
        // Memanggil Fungsi-Fungsi Kendala //
        ////////////////////////////////////
        rho();
        jarak_tulangan();
        kelangsingan();
        analisa();
        sengkang_kolom();

        ////////////////////////////////////
        // Menghitung Kendala Total //
        ////////////////////////////////////
        kendala=kendala_sengkang
                +kendala_r
                +kendala_tul
                +kendala_gaya
                +kendala_kelangsingan;

        ////////////////////////////////////
        // Perhitungan Berat Dan Volume //
        ////////////////////////////////////
        volume_beton=sisi*sisi*L/1.E6;
        berat_besi=ASTOT*L*bj_besi/1.E6;
        berat_sengkang=(fabs(L/(Jarak_S/1000.))-1.)
                        *4.*((sisi-(2.*selimut_kolom))/1000.)
                        *(pi/4.)*pow((DIAS/1000.),2.)
                        *bj_besi;

        ////////////////////////////////////
        // Menghitung Harga //
        ////////////////////////////////////
        harga=(volume_beton*harga_beton)
              +(berat_besi*harga_besi)
              +(berat_sengkang*harga_besi);
    }
};

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA RASIO PENULANGAN */
*****/
void kolom::rho()
{
    ASTOT=((4.*N_DIA-4.)*(pi/4.)*(pow(DIA,2)));
    RHO=(ASTOT/(pow(sisi,2)));

    ////////////////////////////////////
    // Menghitung Kendala Rho Minimum //
    ////////////////////////////////////
    kendala_r_min=((0.01/RHO)-1.);
    if(kendala_r_min<0.)
    {
        kendala_r_min=0.;
    }
}

```

```

////////////////////////////////////
// Menghitung Kendala Rho Maksimum //
////////////////////////////////////
kendala_r_mak=((RHO/0.08)-1.);
if(kendala_r_mak<0.)
{
    kendala_r_mak=0.;
}

////////////////////////////////////
// Menjumlahkan Kendala Rho //
////////////////////////////////////
kendala_r=kendala_r_min+kendala_r_mak;
}

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA JARAK ANTARA TULANGAN */
*****/
void kolom::jarak_tulangan()
{
    //////////////////////////////////////
    // Mengecek Jarak Minimum Antara Tulangan //
    //////////////////////////////////////
    float min1=(1.5*DIA);
    float min2=40.;
    float sisa=sisi-(2.*selimut_kolom)-N_DIA*DIA;

    jarak_antar_tulangan=sisa/(N_DIA-1.);
    if(jarak_antar_tulangan==0)
    {
        jarak_antar_tulangan=limit_nol;
    }

    float jarak_min=min1;
    if(min2<min1)
    {
        jarak_min=min2;
    }

    //////////////////////////////////////
    // Menghitung Kendala Total Akibat Kendala Jarak Antar Tulangan //
    //////////////////////////////////////
    kendala_tul=((jarak_min/jarak_antar_tulangan)-1.);
    if(kendala_tul<0.)
    {
        kendala_tul=0.;
    }
}

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA KELANGSINGAN KOLOM */
*****/
void kolom::kelangsingan()
{
    //////////////////////////////////////
    // Syarat Kelangsingan Kolom Menurut SK SNI T-15-1991-03 //
    // KL/r <= 22 //
    //////////////////////////////////////
    _K=0.5;
    rasio_kelangsingan=(_K*L*1000./((sisi*(sqrt((1./12.))))));

    //////////////////////////////////////
    // Menghitung Kendala Kelangsingan //
    //////////////////////////////////////
    kendala_kelangsingan=(rasio_kelangsingan/22.)-1.;
    if(kendala_kelangsingan<0)
    {
        kendala_kelangsingan = 0.;
    }
}

```



```

{
    beta=0.3;
}

////////////////////////////////////
// Mencari Momen Ekuivalen Biaksial //
////////////////////////////////////
if(MNX > MNY)
{
    MOX=MNX+(MNY*((1.-beta)/beta)); // Momen ekuivalen (Nmm)
    eks=(fabs(MOX/PN)); // (mm)
}
else
{
    MOY=MNY+(MNX*((1.-beta)/beta)); // Momen ekuivalen (Nmm)
    eks=(fabs(MOY/PN)); // (mm)
}

////////////////////////////////////
// Menghitung Letak Garis Netral Pada Keadaan Balance //
////////////////////////////////////
CB=600.*D/(FY+600.); // (mm)

////////////////////////////////////
// Menghitung Tegangan Pada Keadaan Balance //
////////////////////////////////////
deki=hitung_kolom(CB);
eksb=ekscoba; // (mm)
epsb=0.003*(CB-DS)/CB;
epsy=FY/2.E5;

if(epsb>epsy)
{
    fslb=FY; // (Mpa)
}
if(epsb<epsy)
{
    fslb=epsb*2.E5; // (MPa)
}

//-----//
// Proses Mencari Letak Garis Netral Dengan Metode False Posisi //
//-----//

////////////////////////////////////
// Pembagian Daerah Pencarian //
////////////////////////////////////
trialkiri=40.;
trialkanan=(sisi-40.);
trialbaru=(CB);

////////////////////////////////////
// Menghitung Nilai Awal //
////////////////////////////////////
deki=hitung_kolom(trialkiri);
deka=hitung_kolom(trialkanan);
deba=hitung_kolom(trialbaru);

////////////////////////////////////
// Periksa Batas Range //
////////////////////////////////////
if((deki*deka>0)&&(deki!=0)&&(deka!=0))
{
    do
    {
        //////////////////////////////////////
        // Bila Nilai Yang Kita Cari Tidak Ada Dalam Range //
        //////////////////////////////////////
        if(trialkiri>(limit_nol))
    {

```

```

        trial kiri=((trial kiri)/2.);
        deki=hitung_kolom(trial kiri);
    }
    else
    {
        trial kanan+=10.;
        deka=hitung_kolom(trial kanan);
        if(trial kanan>(sisi/0.85))
        {
            break;
        }
    }
}while(deki*deka>0&&(deki!=0)&&(deka!=0));
}

////////////////////////////////////
// Akselerator False Posisi //
////////////////////////////////////
iterasi_fp=1;
do
{
    //////////////////////////////////
    // Periksa Konvergensi //
    // Diambil E = 1 mm //
    //////////////////////////////////
    if(deba<1.&&deba>(-1.))
    {
        if(PNcoba<=0)
        {
            PNcoba=limit_nol;
        }
        MNcoba=fabs(MNcoba);
        break;
    }

    //////////////////////////////////
    // Periksa Apakah deki Dan deba Sama Tanda //
    //////////////////////////////////
    if((deki*deba)<0.)
    {
        //////////////////////////////////
        // Geser trial kanan menuju trial baru //
        //////////////////////////////////
        trial kanan=trial baru;
        deka=hitung_kolom(trial kanan);
    }
    else
    {
        //////////////////////////////////
        // Geser trial kiri menuju trial baru //
        //////////////////////////////////
        trial kiri=trial baru;
        deki=hitung_kolom(trial kiri);
    }

    //////////////////////////////////
    // Persempit Lagi Daerah Pencarian //
    //////////////////////////////////
    trial baru=((trial kiri+trial kanan)/2.);

    deba=hitung_kolom(trial baru);

    iterasi_fp++;

    if(trial kiri==trial kanan)
    {
        break;
    }

}while(iterasi_fp<=100);

```

```

////////////////////////////////////
// Insialisasi Jumlah Iterasi //
////////////////////////////////////
iterasi_fp=1;

/*-----*/
/* Proses Pencarian Dengan Metoda False Posisi Dimulai */
/*-----*/
do
{
    if(trialkiri==trialkanan)
    {
        break;
    }

    //////////////////////////////////
// Menentukan Titik Coba Baru //
    //////////////////////////////////
    trialbaru=trialkanan-((deka)*(trialkanan-trialkiri)
                        /((deka)-(deki)));

    deba=hitung_kolom(trialbaru);

    //////////////////////////////////
// Periksa Konvergensi //
// Diambil E = 10 mm //
    //////////////////////////////////
    if(deba<10.&&deba>(-10.))
    {
        if(PNcoba<=0)
        {
            PNcoba=limit_nol;
        }
        MNCoba=fabs(MNCoba);
        break;
    }

    //////////////////////////////////
// Periksa Apakah deki Dan deba Sama Tanda //
    //////////////////////////////////
    if((deki*deba)<0.)
    {
        //////////////////////////////////
        // Geser trialkanan menuju trialbaru //
        //////////////////////////////////
        trialkanan=trialbaru;
        deka=hitung_kolom(trialkanan);
    }
else
{
    //////////////////////////////////
    // Geser trial kiri menuju trialbaru //
    //////////////////////////////////
    trialkiri=trialbaru;
    deki=hitung_kolom(trialkiri);
}

    iterasi_fp++;
    if(iterasi_fp>=100)
    {
        //////////////////////////////////
        // Dihitung Dengan Metoda Pendekatan //
        //////////////////////////////////
        deka=hitung_kolom((trialkiri+trialkanan)/2.);
        break;
    }
}while(deki*deka!=0.);
FPU=fabs(PNcoba);
FMU=fabs(MNCoba)/1000.;
// Dalam N
// Hasil perhitungan dalam Nmm -> Nm

```

```

// Menghitung Kendala Akibat Gaya Aksial //
// Menghitung Kendala Akibat Gaya Lentur //
// Eksentrisitas Pada Kolom Berpengikat Sengkan //
else
{
    FPU=PO;
    FMU=0.;
    kendala_pn=0.;
    kendala_mn=0.;
}

// Menghitung Kendala Total Akibat Gaya Yang Terjadi //
kendala_gaya=kendala_po+kendala_pn+kendala_mn;
}

/*****
/* FUNGSI UNTUK MENGHITUNG TEGANGAN PADA KOLOM */
*****/
float kolom::hitung_kolom(float Ccoba)
{
    // Menghitung Gaya Yang Ditahan Beton //
    PNCoba=(0.85*BT1*FC*Ccoba*sisi);
    MNCoba=(0.85*BT1*FC*Ccoba*sisi)*((sisi/2.)-(BT1*Ccoba/2.));

    // Iterasi Untuk Kekuatan Yang Didukung Tulangan //
    for(int ikl=1; ikl<=N_DIA; ikl++)
    {
        di=(DS+((N_DIA-ikl)*(sisi-(2.*DS))/(N_DIA-1.)));

        if(di==(0.5*sisi))
        {
            continue;
        }
    }
}

```

```

////////////////////////////////////
// Luas Tulangan Pada Baris Luar //
////////////////////////////////////
if((ikl==1)||(ikl==N_DIA))
{
    ASdi=(N_DIA*(pi/4.)*pow(DIA,2));
}

////////////////////////////////////
// Luas Tulangan Pada Baris Dalam //
////////////////////////////////////
else
{
    ASdi=(2.*(pi/4.)*pow(DIA,2));
}

////////////////////////////////////
// Antisipasi Nilai Ccoba Dengan Limit Nol //
////////////////////////////////////
if(Ccoba==0)
{
    Ccoba=limit_nol;
}

////////////////////////////////////
// Hitung Tegangan Tahanan Tulangan //
// fsi = Regangan * Modulus Elastisitas //
// Dengan Regangan = 0.003[(c-di)/c] //
// Modulus Elastisitas = 2E5 MPa (Baja) //
////////////////////////////////////
fsi=((600.*(Ccoba-di))/Ccoba);

////////////////////////////////////
// Periksa Range Tegangan //
////////////////////////////////////
if(fsi>FY)
{
    fsi=FY;
}
else if(fsi<(-FY))
{
    fsi=(0.-FY);
}

////////////////////////////////////
// Reduksi Tegangan Baja Pada Daerah Desak //
////////////////////////////////////
if(di<(BT1*Ccoba))
{
    fsi=(fsi-(0.85*FC));
}

////////////////////////////////////
// Hitung Gaya Yang Disumbangkan Oleh Tulangan //
////////////////////////////////////
Fsi=fsi*ASdi;

////////////////////////////////////
// Hitung Gaya Aksial Dan Momen Tahanan Maksimal //
////////////////////////////////////
PNcoba=(PNcoba+Fsi);
MNcoba=(MNcoba+(Fsi*((sisi/2.)-di)));
}

////////////////////////////////////
// Antisipasi Divide By Zero Error //
////////////////////////////////////
if (PNcoba<=0)
{
    PNcoba=limit_nol;
}

```

```

////////////////////////////////////
// Hitung Momen Dan Gaya Aksial Total //
////////////////////////////////////
ekscoba=(MNCoba/PNCoba);

////////////////////////////////////
// Kembalikan Nilai Fungsi //
////////////////////////////////////
return (eks-ekscoba);
}

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA GAYA GESER PADA KOLOM */
*****/
void kolom::senggang_kolom()
{
    //////////////////////////////////////
    // Menghitung Antar Jarak Senggang Maksimal //
    // Berdasarkan SK SNI T-15-1991-03 pasal 3.16.10 ayat 5 //
    //////////////////////////////////////
    float Smak[3];
    Smak[0]=sisi;
    Smak[1]=16.*DIA;
    Smak[2]=48.*DIAS;
    float SmakS=Smak[0];

    //////////////////////////////////////
    // Cari Yang Terkecil //
    //////////////////////////////////////
    for (int sk=1;sk<3;sk++)
    {
        if(SmakS>Smak[sk])
        {
            SmakS=Smak[sk];
        }
    }

    Sref=SmakS;
    //////////////////////////////////////
    // Menghitung Kendala Jarak Antar Senggang //
    //////////////////////////////////////
    kendala_senggang=((Jarak_S/SmakS)-1.);
    if(kendala_senggang<0.)
    {
        kendala_senggang=0.;
    }
}

```

```

/*****
/* FILE : BALOK.HPP
/* KELAS UNTUK MENGHITUNG HARGA DAN KENDALA PADA BALOK
*****/
class balok
{
    private:

        /*****
        /* PENDEKLARASIAN VARIABEL PRIVATE PADA KELAS BALOK */
        *****/

        ////////////
        // Perhitungan Balok //
        ////////////
        float FS;      // Tegangan pada tulangan tarik (N/mm^2)
        float FS1;     // Tegangan pada tulangan desak (N/mm^2)
        float ARM;     // Lengan pusat tekan ke garis Netral (mm)
        float ARMS;    // Lengan pusat tulangan ke garis Netral (mm)
        float EPS;     // Regangan pada tulangan tarik (mm)
        float EPS1;    // Regangan pada tulangan tekan (mm)
        float EPSY;    // Regangan ijin baja pada keadaan luluh (mm)
        float RHB;     // Rasio tulangan balance
        float RHO;     // Rasio tulangan tarik
        float RH1;     // Rasio tulangan desak
        float RMIN;    // Rasio tulangan minimum
        float SmakS;   // Jarak antar sengkang maksimum yang diijinkan (mm)

        ////////////
        // Variabel Pembantu Persamaan Kuadrat //
        ////////////
        float ASOL;
        float BSOL;
        float CSOL;
        float DSOL;

        /*****
        /* PROTOTYPE FUNGSI PADA KELAS BALOK */
        *****/
        void analisa();
        void sengkang_balok();
        public:

        /*****
        /* CONSTRUCTOR CLASS BALOK */
        *****/
        balok()
        {
            ////////////
            // Inisialisai Kendala Dan Harga = 0 //
            ////////////
            kendala=0.;
            harga=0.;

            ////////////
            // Menghitung Luas Tulangan Tarik Dan Tekan //
            ////////////
            AS=(0.25*pi*(pow(DIA1,2))*NL1);
            AS1=(0.25*pi*(pow(DIA2,2))*NL2);

            ////////////
            // Memanggil Fungsi Fungsi Kendala //
            ////////////
            analisa();
            sengkang_balok();
            kendala=kendala_sb+kendala_rho+kendala_M;

            ////////////
            // Perhitungan Berat Dan Volume //
            ////////////
            volume_beton=B*H*0.5*L/1.E6;

```

```

        berat_besi=(AS+AS1)*0.5*L*bj_besi/1.E6;
        berat_sengkang=0.5*(fabs(L/(Jarak_S/1000.))-1.)*
            (
                2.*
                (
                    ( (B-(2.*selimut_balok)) /1000.)+
                    2.*(H-(2.*selimut_balok))/1000.)
                )
            )*
            (pi/4.)*pow((DIAS/1000.),2)*bj_besi;

        ////////////
        // Menghitung Harga //
        ////////////
        harga=volume_beton*harga_beton
            +berat_besi*harga_besi
            +berat_sengkang*harga_besi;
    }
};

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA TEGANGAN MATERIAL */
*****/
void balok::analisa()
{
    ////////////
    // Menghitung Faktor Pengali Tinggi Blok Tegangan Segiempat //
    // Ekuivalen (B1) //
    // Sumber : SK SNI T-15-1991-03 Pasal 3.3.2 butir 7.(3) //
    ////////////
    if(FC<=30.)
    {
        BT1=0.85;
    }
    else
    {
        BT1=0.85-0.008*(FC-30.);
    }
    if(BT1<0.65)
    {
        BT1=0.65;
    }

    DS=selimut_balok+(0.5*DIA1);

    D=H-DS;
    FS=FY;

    EPSY=FY/200000.;

    ////////////
    //Asumsi Pertama //
    ////////////
    FS1=FS;

    ////////////
    // Hitung Besarnya Lengan Ke Garis Netral (mm) //
    ////////////
    ARM=((AS-AS1)*FY)/(0.85*FC*B);
    if(ARM==0)
    {
        ARM=limit_nol;
    }

    ////////////
    // Periksa Regangan Pada Tulangan //
    ////////////
    EPS1=0.003*(ARM-BT1*DS)/ARM;
    EPS=0.003*(BT1*D-ARM)/ARM;

```



```

////////////////////////////////////
// Hitung Ulang Regangan Pada Baja Apabila Tulangan Tidak Luluh //
////////////////////////////////////
if((EPS<EPSY)|| (EPS1<EPSY))
{
    //////////////////////////////////////
    // Hitung Koefisien Persamaan Kuadrat Dan Menyelesaikannya //
    //////////////////////////////////////
    ASOL=0.85*FC*B;
    BSOL=600.*AS1-AS*FY;
    CSOL=(-600.)*BT1*AS1*DS;
    DSOL=pow(BSOL,2)-(4.*(ASOL*CSOL));
    if(DSOL<=0)
    {
        DSOL=0.;
    }
    ARMS=(-BSOL)+sqrt(DSOL);
    if(ARMS<=0&&DSOL>=0)
    {
        ARMS=(-BSOL)-sqrt(DSOL);
    }
    ARM=0.5*(ARMS/ASOL);
    if(ARM<=0)
    {
        ARM=limit_nol;
    }

    //////////////////////////////////////
    // Hitung Regangan Pada Tulangan //
    //////////////////////////////////////
    EPS1=0.003*(ARM-BT1*DS)/ARM;
    EPS=0.003*(BT1*D-ARM)/ARM;
}

if(EPS>EPSY)
{
    FS=FY;
}

FS1=EPS1*200000.;
if(FS1>FY)
{
    FS1=FY;
}
if(FS1<0)
{
    FS1=0;
}

////////////////////////////////////
// Perhitungan Rasio Penulangan //
////////////////////////////////////
RH1=AS1/(B*D);
RHB=(0.75*(0.85*FC*BT1/FY)*(600./(600.+FS)))+RH1*FS1*FS1/FY;
RHO = AS/(B*D);
RMIN=1.4/FY;

////////////////////////////////////
// Menghitung Kendala Rasio Tulangan Balance //
////////////////////////////////////
kendala_rho_b=(RHO/RHB)-1.;
if(kendala_rho_b<0)
{
    kendala_rho_b=0.;
}

////////////////////////////////////
// Menghitung Kendala Rasio Tulangan Minimum //
////////////////////////////////////
kendala_rho_m=(RMIN/RHO)-1.;
if(kendala_rho_m<0)

```

```

{
    kendala_rho_m=0;
}

////////////////////////////////////
// Menghitung Kendala Rho Total //
////////////////////////////////////
kendala_rho=kendala_rho_b+kendala_rho_m;

////////////////////////////////////
// Momen Pada Balok Nmm -> Nm //
////////////////////////////////////
FMU=(teta*((0.85*FC*ARM*B)*(D-ARM/2.)+(ASl*FSl)*(D-DS)))/1000.;

////////////////////////////////////
// Menghitung Kendala Momen Lentur //
////////////////////////////////////
if(FMU>0)
{
    kendala_M=((MU)/FMU)-1.;
}
else
{
    kendala_M=0.;
}
if(kendala_M<0)
{
    kendala_M=0.;
}
}

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA PENULANGAN GESER */
*****/
void balok::sengkang_balok()
{
    DS=selimut_balok+(0.5*DIA1);
    D=H-DS;

    AV=0.25*pi*(pow(DIAS,2));

    //////////////////////////////////////
    // Kuat Geser Nominal Yang Disumbangkan Tulangan Geser //
    //////////////////////////////////////
    VC=(1./6.)*sqrt(FC)*B*D;
    VS=((VU/teta)-VC);

    //////////////////////////////////////
    // Menghitung Jarak Sengkang Maksimal //
    //////////////////////////////////////
    float Smak[4];
    if(VS<=0)
    {
        VS=limit_nol;
    }
    Smak[0]=3.*AV*FYS/B;
    Smak[1]=(AV*FYS*D)/(VS);
    Smak[2]=0.5*D;
    Smak[3]=600.;
    if(VS>((1./3.)*sqrt(FC)*B*D))
    {
        Smak[2]=0.25*D;
        Smak[3]=300.;
    }
    SmakS=Smak[0];

    //////////////////////////////////////
    // Cari Yang Terkecil //
    //////////////////////////////////////
    for (int sk=1;sk<4;sk++)
    {

```

```

        if(SmakS>Smak[sk])
        {
            SmakS=Smak[sk];
        }
    }

    Sref=SmakS;
    //////////////////////////////////////
    // Menghitung Kendala Jarak Antar Sengkang Maksimum //
    //////////////////////////////////////
    kendala_sb=((Jarak_S/SmakS)-1.);
    if(kendala_sb<0.)
    {
        kendala_sb=0.;
    }
}

/*****
/* FUNGSI UNTUK MENGHITUNG LENDUTAN MAKSIMUM */
/* BERDASARKAN SK SNI-T-15-1991-03 AYAT 3.2.5 */
*****/
void lendutan(int no_batang_1)
{
    //////////////////////////////////////
    // Hitung Besaran Dasar //
    //////////////////////////////////////
    n=2.E5/(E*1.E-6);

    fr=0.7*sqrt(FC); // (MPa atau N/mm^2)

    //////////////////////////////////////
    // Letak Garis Netral //
    //////////////////////////////////////
    AS=(0.25*pi*(pow(DIAL,2))*NL1);
    AS1=(0.25*pi*(pow(DIA2,2))*NL2);

    LGN=(n*AS/B)*(sqrt(1.+((2.*B*(H-selimut_balok))/(n*AS)))-1.);

    //////////////////////////////////////
    // Menentukan Momen Inersia Penampang Retak Transformasi (mm^4) //
    //////////////////////////////////////
    Icr=(1./3.)*B*pow(LGN,3)+n*AS*pow((H-selimut_balok)-LGN,2);

    //////////////////////////////////////
    // Momen Inersia Penampang Kotor (mm^4) //
    //////////////////////////////////////
    Ig=(1./12.)*B*pow(H,3);

    //////////////////////////////////////
    // Momen Pada Saat Timbul Retak Pertama Kali (Nmm) //
    //////////////////////////////////////
    Mcr=fr*Ig/(0.5*H);

    //////////////////////////////////////
    // Momen Inersia Efektif Untuk Perhitungan Lendutan (mm^4) //
    //////////////////////////////////////
    Ie=pow((Mcr/(MLAP[no_batang_1]*1.e3)),3)*Ig
        +(1.-pow((Mcr/(MLAP[no_batang_1]*1.e3)),3))*Icr;

    //////////////////////////////////////
    // Menghitung Lendutan Seketika //
    //////////////////////////////////////
    // Lendutan Pada Balok (mm) //
    // Dari Buku : Reinforced Concrete Mechanics And Design //
    // Halaman : 355 //
    //////////////////////////////////////
    LENDUTAN=(
        (
            5.*(pow(L*1000.,2))

```

```

        /(48.*(E*1.E-6)*Ie)
    )*
    (
        fabs(MLAP[no_batang_1])*1000.
    )
);

////////////////////////////////////////
// Menghitung Lendutan Jangka Panjang //
// Menurut SK SNI-T-15-1991-03 pasal 3.2.5(2.5) //
////////////////////////////////////////
Lambda=2./(1.+(50.*AS1/(B*H)));
LENDUTAN=(1.+Lambda)*LENDUTAN;

LENDUTAN_IJIN=(L*1000.)/180.;

kendala_lendutan=((LENDUTAN/LENDUTAN_IJIN)-1.);
if(kendala_lendutan<0)
{
    kendala_lendutan=0.;
}
}

```

```

/*****
/* FILE : ELEMEN.HPP
/* PENGISIAN ELEMEN BALOK DAN KOLOM
*****/

/*****
/* PENGISIAN DATA ELEMEN BALOK */
*****/
void isi_elemen_balok(int no_el_balok)
{
    B=isi(var_b[no_struktur][0+(12*no_el_balok)],sisi_d_B); // (mm)
    H=isi(var_b[no_struktur][1+(12*no_el_balok)],sisi_d_H); // (mm)

    DIA1lap=isi(var_b[no_struktur][2+(12*no_el_balok)],DIA_d); // (mm)
    NL1lap=isi(var_b[no_struktur][3+(12*no_el_balok)],NL_d);
    DIA2lap=isi(var_b[no_struktur][4+(12*no_el_balok)],DIA_d); // (mm)
    NL2lap=isi(var_b[no_struktur][5+(12*no_el_balok)],NL_d);

    DIA1tum=isi(var_b[no_struktur][6+(12*no_el_balok)],DIA_d); // (mm)
    NL1tum=isi(var_b[no_struktur][7+(12*no_el_balok)],NL_d);
    DIA2tum=isi(var_b[no_struktur][8+(12*no_el_balok)],DIA_d); // (mm)
    NL2tum=isi(var_b[no_struktur][9+(12*no_el_balok)],NL_d);

    DIAS=isi(var_b[no_struktur][10+(12*no_el_balok)],DIAS_d); // (mm)
    Jarak_S=isi(var_b[no_struktur][11+(12*no_el_balok)],JS_d); // (mm)
    L=EL[no_balok[no_el_balok]]; // (m)

    if(fabs(GESER_KI[no_balok[no_el_balok]])
    >fabs(GESER_KA[no_balok[no_el_balok]]))
    {
        VU = fabs(GESER_KI[no_balok[no_el_balok]]/teta); // (N)
    }
    else
    {
        VU = fabs(GESER_KA[no_balok[no_el_balok]]/teta); // (N)
    }
}

/*****
/* ELEMEN DAERAH LAPANGAN */
*****/
void elemen_lapangan(int no_el_balok)
{
    DIA1=DIA1lap; // Tulangan tarik (mm)
    NL1=NL1lap;
    DIA2=DIA2lap; // Tulangan desak (mm)
    NL2=NL2lap;
    MU=fabs(MLAP[no_balok[no_el_balok]]/(teta)); // momen ditahan (Nm)
}

/*****
/* ELEMEN DAERAH TUMPUAN */
*****/
void elemen_tumpuan(int no_el_balok)
{
    DIA1=DIA1tum; // Tulangan tarik (mm)
    NL1=NL1tum;
    DIA2=DIA2tum; // Tulangan desak (mm)
    NL2=NL2tum;

    if(fabs(MTUM_KI[no_balok[no_el_balok]])
    >fabs(MTUM_KA[no_balok[no_el_balok]]))
    {
        MU=fabs(MTUM_KI[no_balok[no_el_balok]]/teta); // (Nm)
    }
    else
    {
        MU=fabs(MTUM_KA[no_balok[no_el_balok]]/teta); // (Nm)
    }
}

```

```

/*****/
/* PENGISIAN DATA ELEMEN KOLOM */
/*****/
void isi_elemen_kolom(int no_el_kolom)
{
    sisi   =isi(var_k[no_struktur][0+(5*no_el_kolom)],sisi_d_K);// (mm)
    DIA    =isi(var_k[no_struktur][1+(5*no_el_kolom)],DIA_d);    // (mm)
    N_DIA  =isi(var_k[no_struktur][2+(5*no_el_kolom)],NL_d);
    DIAS   =isi(var_k[no_struktur][3+(5*no_el_kolom)],DIAS_d);    // (mm)
    Jarak_S=isi(var_k[no_struktur][4+(5*no_el_kolom)],JS_d);    // (mm)

    PU     =(-(PK[no_kolom[no_el_kolom]])/(teta));                // (N)
    MUX    =(MKX[no_kolom[no_el_kolom]])/(teta);                // (Nm)
    MUY    =(MKY[no_kolom[no_el_kolom]])/(teta);                // (Nm)
    VU     =(fabs((GK[no_kolom[no_el_kolom]])/(teta)));
    // (N)
    L=EL[no_kolom[no_el_kolom]];
    // (m)
}

/*****/
/* SUBPROGRAM UNTUK MEMANGGIL ISI DATA ELEMEN */
/*****/
float isi(int no_data,float kelompok_data[])
{
    return kelompok_data[no_data];
}

```

[illegible]

```

// Penormalan Batas Variabel Balok Kolom //
// normalisasi_int(nvm,nvb,nvk);

clrscr();
about();
cout << "Generasi ke - 1\n";
cout << "\nJumlah Struktur desain = \n";
cout << "\nWaktu yang telah berjalan : 0 detik\n";
cout << "\nEstimasi waktu yang diperlukan : ?\n";

// Pembangkitan Generasi Pertama //
// Struktur Dibangkitkan Sebanyak Jumlah Struktur Desain //
//

for(int iop=0;iop<JSTD;iop++)
{
    // Inisialisasi Variabel Struktur //
    no_struktur=iop;
    kendalastr[iop]=0.;
    hargastr[iop]=0.;;
    fitstr[iop]=0.;

    // Pencatatan Jumlah Struktur Desain Yang Dibangkitkan //
    gotoxy(26,9);
    cout << (iop+1) << " ";

    // Pembacaan Data Umum Struktur //
    baca_data();

    // Membaca Beban Yang Bekerja Pada Struktur //
    baca_beban();

    // Menghitung Gaya-Gaya Struktur //
    inersia();
    struktur();

    // Menghitung Kendala Dan Harga Balok //
    for(int jop=0;jop<jum_balok;jop++)
    {
        // Periksa Kemiringan Batang //
        periksa_batang(no_balok[jop]);

        // Mengisi Properti Elemen Balok //
        isi_elemen_balok(jop);

        // Membangkitkan Balok Pada Lapangan //
        elemen_lapangan(jop);
        balok_lapangan;
    }
}

```



```

////////////////////////////////////
// Mengganti jarak sengkang //
////////////////////////////////////
    for(int cari_S=(nvb[11+(12*jop)]-1);
        cari_S>0;cari_S--)
    {
        if(isi(cari_S,JS_d)<=Sref)
        {
            var_b[no_struktur][11+(12*jop)]=cari_S;
            Jarak_S=isi(cari_S,JS_d);
            break;
        }
    }
    balok raise_lap;

    kendalastr[iop]+=kendala;
    hargastr[iop]+=harga;

    //////////////////////////////////
    // Menghitung Lendutan Pada Tengah Bentang //
    //////////////////////////////////
    lendutan(no_balok[jop]);
    kendalastr[iop]+=kendala_lendutan;

    //////////////////////////////////
    // Membangkitkan Balok Pada Tumpuan //
    //////////////////////////////////
    elemen_tumpuan(jop);
    balok tumpuan;
    kendalastr[iop]+=kendala;
    hargastr[iop]+=harga;
    }

    //////////////////////////////////
    // Menghitung Kendala Dan Harga Kolom //
    //////////////////////////////////
    for(int jop=0;jop<jum_kolom;jop++)
    {
        //////////////////////////////////
        // Periksa Kemiringan Batang //
        //////////////////////////////////
        periksa_batang(no_kolom[jop]);

        //////////////////////////////////
        // Mengisi Properti Elemen Kolom //
        //////////////////////////////////
        isi_elemen_kolom(jop);

        //////////////////////////////////
        // Membangkitkan Kolom //
        //////////////////////////////////
        kolom bangkit;

        //////////////////////////////////
        // Mengganti jarak sengkang //
        //////////////////////////////////
        for(int cari_S=(nvk[4+(5*jop)]-1);
            cari_S>0;cari_S--)
        {
            if(isi(cari_S,JS_d)<=Sref)
            {
                var_k[no_struktur][4+(5*jop)]=cari_S;
                Jarak_S=isi(cari_S,JS_d);
                break;
            }
        }
        kolom raise;
    }

```

```

////////////////////////////////////
// Bila Ada Kendala Kelangsingan Ubah Nilai Sisi //
////////////////////////////////////
if(kendala_kelangsingan>0)
{
    L=EL[no_kolom[jop]];
    float sisi_baru=((0.75*L*1000.)/(22.*sqrt((1./12.))));
    for(int cari_sisi=var_k[no_struktur][0+(5*jop)];
        cari_sisi<nvk[0+(5*jop)];cari_sisi++)
    {
        float sisi_cari=isi(cari_sisi,sisi_d_K);
        if(sisi_cari>sisi_baru)
        {
            var_k[no_struktur][0+(5*jop)]=cari_sisi;
            sisi=isi(cari_sisi,sisi_d_K);
            break;
        }
    }
    //////////////////////////////////
    // Hitung Kendala Baru //
    //////////////////////////////////
    kolom_bangkit;
}

kendalastr[iop]+=kendala;
hargastr[iop]+=harga;
}

////////////////////////////////////
// Menghitung Fitness Struktur //
////////////////////////////////////
fitstr[iop]=(finalti/(hargastr[iop]+(finalti*kendalastr[iop])));
}

////////////////////////////////////
// Lakukan Pengurutan Fitness //
////////////////////////////////////
sort(fitstr,kendalastr,hargastr,var_b,var_k,JSTD);

////////////////////////////////////
// Pengambilan waktu sistem //
////////////////////////////////////
time_t t_awal, t_sekarang, t_akhir;
time(&t_awal);

////////////////////////////////////
// Proses Pencarian Hasil Optimal Dimulai //
////////////////////////////////////
int generasi=1;
jum_susut=0;
do
{
    //////////////////////////////////
    // Menulis Nomor Generasi //
    //////////////////////////////////
    gotoxy(15,7);
    cout << (generasi+1) << " ";

    //////////////////////////////////
    // Menbangkitkan Kelas Timer //
    //////////////////////////////////
    time(&t_sekarang);

    //////////////////////////////////
    // Menulis Lama Waktu Yang Sedang Berjalan //
    //////////////////////////////////
    gotoxy(29,11);
    cout << difftime(t_sekarang,t_awal) << " detik ";
}

```



```

////////////////////////////////////
// Hentikan Iterasi Apabila Setelah JVD Kali Menyusut //
// Dan Fitness Terbaik Tetap //
////////////////////////////////////
if(patok_fit==fitstr[JSTD-1])
{
    opti << "Konvergen Pada Generasi : " << generasi
    << " karena jumlah penyusutan melebihi batas" << endl;
    break;
}

////////////////////////////////////
// Jumlah Penyusutan DiReset Lagi //
////////////////////////////////////
jum_susut=0;
}

////////////////////////////////////
// Increator //
////////////////////////////////////
generasi++;
}while(generasi<j_iterasi_mak);

////////////////////////////////////
// Mengambil Waktu Akhir //
////////////////////////////////////
time(& t_akhir);

////////////////////////////////////
// Cetak Ke Layar //
////////////////////////////////////
gotoxy(1,17);
cout << "Waktu optimasi : " << difftime(t_akhir,t_awal)
    << " detik \n\a";
cout << "\nFitness Terbaik : " << fitstr[JSTD-1]
    << " " << endl;
cout << "Harga : " << hargastr[JSTD-1]
    << " " << endl;
cout << "Kendala : " << kendalastr[JSTD-1]
    << " " << endl;
gotoxy(55,24);
cout << "Tekan <CR> untuk keluar";
getch();

////////////////////////////////////
// Cetak Ke File //
////////////////////////////////////
if (generasi==j_iterasi_mak)
{
    opti << "Selesai pada generasi " << (generasi)
    << " karena jumlah iterasi mencapai jumlah iterasi maksimum\n";
}
    opti << "Fitness = " << fitstr[JSTD-1] << endl;
    opti << "Harga = " << hargastr[JSTD-1] << endl;
    opti << "Kendala = " << kendalastr[JSTD-1] << endl;
    opti << "Waktu optimasi : " << difftime(t_akhir,t_awal)
    << " detik ";
    opti.close();
    cetak_akhir();
}

```

```

/*****
/* FILE : PENORMALAN.HPP
/* PENANGANAN MASALAH VARIABEL BALOK DAN KOLOM
/* DENGAN VARIABEL GABUNGAN BALOK KOLOM
*****/

/*****
/* SUBPROGRAM MENGEMBALIKAN VARIABEL BALOK DAN KOLOM */
*****/
void unnormalisasi(int var_nor[mak],int var_bv[mak],int var_kv[mak])
{
    int novar=0;
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            var_bv[jsum+(12*isum)]=var_nor[novar];
            novar++;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)
        {
            var_kv[jsum+(5*isum)]=var_nor[novar];
            novar++;
        }
    }
}

/*****
/* SUBPROGRAM MENORMALISASIKAN VARIABEL BALOK DAN KOLOM TIPE FLOAT */
*****/
void normalisasi_float(float var_nor[],float var_bv[],float var_kv[])
{
    int novar=0;
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            var_nor[novar]=var_bv[jsum+(12*isum)];
            novar++;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)
        {
            var_nor[novar]=var_kv[jsum+(5*isum)];
            novar++;
        }
    }
}

/*****
/* SUBPROGRAM MENORMALISASIKAN VARIABEL BALOK DAN KOLOM TIPE INTEGER */
*****/
void normalisasi_int(int var_nor[],int var_bv[],int var_kv[])
{
    int novar=0;
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            var_nor[novar]=var_bv[jsum+(12*isum)];
            novar++;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)

```

```
        {
            var_nor[novar]=var_kv[jsum+(5*isum)];
            novar++;
        }
    }
```

```

/*****
/* FILE : PENGACAKAN.HPP
/* PUSTAKA URUSAN VARIABEL DESAIN ACAK
*****/

/*****
/* SUBPROGRAM PENGENDALI LOOP RANDOMISASI */
*****/
void acak_variabel()
{
    //////////////////////////////////////
    // Pengacakan Seed Untuk Random Number Generator //
    //////////////////////////////////////
    randomize();

    load_batas_atas();

    for(int iav=0;iav<JSTD;iav++)
    {
        no_struktur=iav;

        if(iav==1)
        {
            cari_struktur_awal();
            continue;
        }

        //////////////////////////////////
        // Random Struktur Selanjutnya //
        //////////////////////////////////
        randomisasi();
    }
}

/*****
/* SUBPROGRAM UNTUK MELAKUKAN RANDOMISASI */
*****/
void randomisasi()
{
    //////////////////////////////////
    // Randomisasi Variabel Desain Balok //
    //////////////////////////////////
    for(int iran=0;iran<jum_balok;iran++)
    {
        for(int nv_b=0;nv_b<12;nv_b++)
        {
            var_b[no_struktur][nv_b+(12*iran)]=random(nvb[nv_b]);
        }
    }

    //////////////////////////////////
    // Randomisasi Variabel Desain Kolom //
    //////////////////////////////////
    for(int iran=0;iran<jum_kolom;iran++)
    {
        for(int nv_k=0;nv_k<5;nv_k++)
        {
            var_k[no_struktur][nv_k+(5*iran)]=random(nvk[nv_k]);
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENENTUKAN BATAS ATAS VARIABEL DESAIN */
*****/
void load_batas_atas()
{
    for(int isinv=0;isinv<jum_balok;isinv++)
    {
        nvb[0+(12*isinv)]=nsisi_B;// B
        nvb[1+(12*isinv)]=nsisi_H;// H
    }
}

```



```

        nvb[2+(12*isinv)]=nDIA;    // Diameter tulangan tarik lapangan
        nvb[3+(12*isinv)]=nNL;     // jumlah tulangan tarik lapangan
        nvb[4+(12*isinv)]=nDIA;    // Diameter tulangan desak lapangan
            nvb[5+(12*isinv)]=nNL;    // jumlah tulangan desak lapangan
        nvb[6+(12*isinv)]=nDIA;    // Diameter tulangan tarik tumpuan
        nvb[7+(12*isinv)]=nNL;     // jumlah tulangan tarik tumpuan
        nvb[8+(12*isinv)]=nDIA;    // Diameter tulangan desak tumpuan
        nvb[9+(12*isinv)]=nNL;     // jumlah tulangan desak tumpuan
            nvb[10+(12*isinv)]=nDIAS; // Diameter tulangan sengkang
        nvb[11+(12*isinv)]=nJS;    // jarak antar tulangan sengkang
    }
    for(int isinv=0;isinv<jum_kolom;isinv++)
    {
        nvk[0+(5*isinv)]=nsisi_K; // Sisi B=H
        nvk[1+(5*isinv)]=nDIA;    // Diameter tulangan bawah lapangan
        nvk[2+(5*isinv)]=nNL;     // jumlah tulangan pada satu sisi
        nvk[3+(5*isinv)]=nDIAS;   // Diameter tulangan sengkang
        nvk[4+(5*isinv)]=nJS;     // jarak antar tulangan sengkang
    }
}

void cari_struktur_awal()
{
    for(int iran=0;iran<jum_balok;iran++)
    {
        for(int nv_b=0;nv_b<12;nv_b++)
        {
            var_b[no_struktur][nv_b+(12*iran)]=(nvb[nv_b]-1);
            if(nv_b==5|nv_b==9|nv_b==11)
            {
                var_b[no_struktur][nv_b+(12*iran)]=0;
            }
        }
    }

    for(int iran=0;iran<jum_kolom;iran++)
    {
        for(int nv_k=0;nv_k<5;nv_k++)
        {
            var_k[no_struktur][nv_k+(5*iran)]=(nvk[nv_k]-1);
        }
    }
}

```

```

/*****/
/* FILE : DISKRITISASI.HPP
*/
/* SUBPROGRAM UNTUK MELAKUKAN KONVERSI MENJADI BILANGAN DISKRIT */
/*****/
int konversi(float bil_asli)
{
    if((fabs(bil_asli)-(abs(bil_asli)))<=0.5)
    {
        bil_asli=floor(bil_asli);
    }
    else
    {
        bil_asli=ceil(bil_asli);
    }
    return bil_asli;
}

```

```

/*****
/* FILE : KENDALA.HPP
/* SUBPROGRAM UNTUK MENGHITUNG KENDALA
*****/
float Kendala_Harga(int var_b_nya[],int var_k_nya[])
{
    ////////////////////////////////////
    // Inisialisasi Kendala Dan Harga //
    ////////////////////////////////////
    float kendalanya=0.;
    float harganya=0.;

    ////////////////////////////////////
    // Pembacaan Data Umum Struktur //
    ////////////////////////////////////
    baca_data();

    ////////////////////////////////////
    // Membaca Beban Yang Bekerja Pada Struktur //
    ////////////////////////////////////
    baca_beban();

    ////////////////////////////////////
    // Menghitung Gaya-Gaya Struktur //
    ////////////////////////////////////
    inersia();
    struktur();

    ////////////////////////////////////
    // Menghitung Kendala Dan Harga Balok //
    ////////////////////////////////////
    for(int iKH=0;iKH<jum_balok;iKH++)
    {
        ////////////////////////////////////
        // Memeriksa Kemiringan Batang //
        ////////////////////////////////////
        periksa_batang(no_balok[iKH]);

        ////////////////////////////////////
        // Pengisian Elemen Balok //
        ////////////////////////////////////
        B=isi(var_b_nya[0+(12*iKH)],sisi_d_B);
        H=isi(var_b_nya[1+(12*iKH)],sisi_d_H);

        DIA1lap=isi(var_b_nya[2+(12*iKH)],DIA_d);
        NL1lap=isi(var_b_nya[3+(12*iKH)],NL_d);
        DIA2lap=isi(var_b_nya[4+(12*iKH)],DIA_d);
        NL2lap=isi(var_b_nya[5+(12*iKH)],NL_d);

        DIA1tum=isi(var_b_nya[6+(12*iKH)],DIA_d);
        NL1tum=isi(var_b_nya[7+(12*iKH)],NL_d);
        DIA2tum=isi(var_b_nya[8+(12*iKH)],DIA_d);
        NL2tum=isi(var_b_nya[9+(12*iKH)],NL_d);

        DIAS=isi(var_b_nya[10+(12*iKH)],DIAS_d);
        Jarak_S=isi(var_b_nya[11+(12*iKH)],JS_d);
        L=EL[no_balok[iKH]];

        ////////////////////////////////////
        // Menentukan Gaya Geser Yang Terjadi //
        ////////////////////////////////////
        if(fabs(GESER_KI[no_balok[iKH]])>GESER_KA[no_balok[iKH]])
        {
            VU = fabs(GESER_KI[no_balok[iKH]]/teta);
        }
        else
        {
            VU = fabs(GESER_KA[no_balok[iKH]]/teta);
        }
    }
}

```

```

////////////////////////////////////
// Daerah Lapangan //
////////////////////////////////////
DIA1=DIA1lap;
NL1=NL1lap;
DIA2=DIA2lap;
NL2=NL2lap;
MU=fabs(MLAP[no_balok[iKH]]/teta);

////////////////////////////////////
// Membangkitkan Balok Pada Lapangan //
////////////////////////////////////
balok_lapangan;
kendalanya+=kendala;
harganya+=harga;

lendutan(no_balok[iKH]);
kendalanya+=kendala_lendutan;

////////////////////////////////////
// Daerah Tumpuan //
////////////////////////////////////
DIA1=DIA1tum;
NL1=NL1tum;
DIA2=DIA2tum;
NL2=NL2tum;

if(fabs(MTUM_KI[no_balok[iKH]])>fabs(MTUM_KA[no_balok[iKH]]))
{
    MU=fabs(MTUM_KI[no_balok[iKH]]/teta);
}
else
{
    MU=fabs(MTUM_KA[no_balok[iKH]]/teta);
}

////////////////////////////////////
// Membangkitkan Balok Pada Tumpuan //
////////////////////////////////////
balok_tumpuan;
kendalanya+=kendala;
harganya+=harga;
}

////////////////////////////////////
// Menghitung Kendala Dan Harga Kolom Biaksial //
////////////////////////////////////
for(int iKH=0;iKH<jum_kolom;iKH++)
{
    //////////////////////////////////////
    // Periksa Kemiringan Batang //
    //////////////////////////////////////
    periksa_batang(no_kolom[iKH]);

    //////////////////////////////////////
    // Pengisian Data Elemen //
    //////////////////////////////////////
    sisi    =isi(var_k_nya[0+(5*iKH)],sisi_d_K); // sisi kolom (mm)
    DIA     =isi(var_k_nya[1+(5*iKH)],DIA_d);    // Diameter tulangan
                                                // utama (mm)

    N_DIA   =isi(var_k_nya[2+(5*iKH)],NL_d);    // Jumlah tulangan
    DIAS    =isi(var_k_nya[3+(5*iKH)],DIAS_d);  // Diameter sengkang (mm)
    Jarak_S =isi(var_k_nya[4+(5*iKH)],JS_d);    // Jarak antara
                                                // sengkang (mm)

    //////////////////////////////////////
    // Gaya Aksial Yang Harus Ditahan //
    //////////////////////////////////////
    PU      =fabs((PK[no_kolom[iKH]]/(teta))); // (N)

```

```

////////////////////////////////////
// Momen Yang Harus Ditahan //
////////////////////////////////////
MUX =(MKX[no_kolom[iKH]])/(teta);      // (Nm)
MUY =(MKY[no_kolom[iKH]])/(teta);      // (Nm)

////////////////////////////////////
// Gaya Geser Yang Harus Ditahan //
////////////////////////////////////
VU =fabs((GK[no_kolom[iKH]])/(teta)); // (N)

L=EL[no_kolom[iKH]]; // Panjang kolom (m)

////////////////////////////////////
// Membangkitkan Kolom //
////////////////////////////////////
kolom bangkit;

for(int cari_S=(nvk[4+(5*iKH)]-1);
    cari_S>0;cari_S--)
{
    if(isi(cari_S,JS_d)<=Sref)
    {
        var_k_nya[4+(5*iKH)]=cari_S;
        Jarak_S=isi(cari_S,JS_d);
        break;
    }
}
kolom raise;

////////////////////////////////////
// Bila Ada Kendala Kelangsingan Ubah Nilai Sisi //
////////////////////////////////////
if(kendala_kelangsingan>0)
{
    L=EL[no_kolom[iKH]]; // Panjang kolom (m)
    float sisi_baru=((0.75*L*1000.)/(22.*sqrt((1./12.))));
    for(int cari_sisi=var_k_nya[0+(5*iKH)];
        cari_sisi<nvk[0+(5*iKH)];cari_sisi++)
    {
        float sisi_cari=isi(cari_sisi,sisi_d_K);
        if(sisi_cari>sisi_baru)
        {
            var_k_nya[0+(5*iKH)]=cari_sisi;
            sisi=isi(cari_sisi,sisi_d_K);
            break;
        }
    }
    // Hitung Kendala Baru //
    kolom bangkit;
}

kendalanya+=kendala;
harganya+=harga;
}
return(finalti/(harganya+(finalti*kendalanya)));
}

```

```

/*****
/* FILE : TELUSUR.HPP
/* SUBPROGRAM UNTUK MELAKUKAN PENELUSURAN MENUJU TITIK BARU
*****/
void penelusuran()
{
    ////////////////////////////////////////////////////
    // Mencari Letak Titik Midle //
    ////////////////////////////////////////////////////

    ////////////////////////////////////////////////////
    // Inisialisasi Awal //
    ////////////////////////////////////////////////////
    float sum_bestgood_b[mak];
    float sum_bestgood_k[mak];
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            sum_bestgood_b[jsum+(12*isum)]=0.;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)
        {
            sum_bestgood_k[jsum+(5*isum)]=0.;
        }
    }

    ////////////////////////////////////////////////////
    // Jumlah Kordinat = 12*jum_balok_p + 5*jum_kolom_p //
    // Menjumlahkan Variabel Best Dan Good Pada Balok //
    ////////////////////////////////////////////////////
    for(int nbgw=1;nbgw<JSTD;nbgw++)
    {
        ////////////////////////////////////////////////////
        // 12*jum_balok_p Titik M //
        ////////////////////////////////////////////////////
        for(int isum=0;isum<jum_balok;isum++)
        {
            for(int jsum=0;jsum<12;jsum++)
            {
                sum_bestgood_b[jsum+(12*isum)]+=var_b[nbgw][jsum+(12*isum)];
            }
        }

        ////////////////////////////////////////////////////
        // 5*jum_kolom_p Titik M //
        ////////////////////////////////////////////////////
        for(int isum=0;isum<jum_kolom;isum++)
        {
            for(int jsum=0;jsum<5;jsum++)
            {
                sum_bestgood_k[jsum+(5*isum)]+=var_k[nbgw][jsum+(5*isum)];
            }
        }
    }

    ////////////////////////////////////////////////////
    // Dari Sini Kita Mempunyai jum_kolom_p Koordinat //
    // Dan jum_balok_p Koordinat //
    // Menentukan Koordinat Titik M //
    ////////////////////////////////////////////////////
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            XM_b[jsum+(12*isum)]=sum_bestgood_b[jsum+(12*isum)]/(JSTD-1.);
        }
    }
}

```

```

for(int isum=0;isum<jum_kolom;isum++)
{
    for(int jsum=0;jsum<5;jsum++)
    {
        XM_k[jsum+(5*isum)]=sum_bestgood_k[jsum+(5*isum)]/(JSTD-1.);
    }
}

////////////////////////////////////
// Mencari Arah Penelusuran Pada Balok //
////////////////////////////////////
for(int isum=0;isum<jum_balok;isum++)
{
    for(int jsum=0;jsum<12;jsum++)
    {
        XS_b[jsum+(12*isum)]=(XM_b[jsum+(12*isum)]
                                -var_b[0][jsum+(12*isum)]);
        if(XS_b[jsum+(12*isum)]>0.)
        {
            tanda_arah_b[jsum+(12*isum)]=1.;
        }
        else
        {
            if(XS_b[jsum+(12*isum)]<0)
            {
                tanda_arah_b[jsum+(12*isum)]=(-1.);
            }
            if(XS_b[jsum+(12*isum)]==0)
            {
                tanda_arah_b[jsum+(12*isum)]=(0.);
            }
        }
    }
}

////////////////////////////////////
// Mencari Arah Penelusuran Pada Kolom //
////////////////////////////////////
for(int isum=0;isum<jum_kolom;isum++)
{
    for(int jsum=0;jsum<5;jsum++)
    {
        XS_k[jsum+(5*isum)]=(XM_k[jsum+(5*isum)]
                                -var_k[0][jsum+(5*isum)]);
        if(XS_k[jsum+(5*isum)]>0.)
        {
            tanda_arah_k[jsum+(5*isum)]=1.;
        }
        else
        {
            if(XS_k[jsum+(5*isum)]<0)
            {
                tanda_arah_k[jsum+(5*isum)]=(-1.);
            }
            if(XS_k[jsum+(5*isum)]==0)
            {
                tanda_arah_k[jsum+(5*isum)]=(0.);
            }
        }
    }
}

////////////////////////////////////
// Penggandaan Variabel Terjelek //
////////////////////////////////////

////////////////////////////////////
// 12*jum_balok_p Koordinat Titik Terjelek //
////////////////////////////////////
for(int isum=0;isum<jum_balok;isum++)

```

```

        {
            for(int jsum=0;jsum<12;jsum++)
            {
                var_b_jelek[jsum+(12*isum)]=var_b[0][jsum+(12*isum)];
            }
        }

//////////
// 5*jum_kolom_p Koordinat Titik Terjelek //
//////////
        for(int isum=0;isum<jum_kolom;isum++)
        {
            for(int jsum=0;jsum<5;jsum++)
            {
                var_k_jelek[jsum+(5*isum)]=var_k[0][jsum+(5*isum)];
            }
        }

//////////
// Inisialisasi Awal = 0 //
//////////
        for(int isum=0;isum<JVD;isum++)
        {
            TM[isum]=0.;
            TS[isum]=0.;
            varnew_asli[isum]=0;
            arah[isum]=0;
        }
        normalisasi_float(TM,XM_b,XM_k);
        normalisasi_float(TS,XS_b,XS_k);
        normalisasi_int(arah,tanda_arah_b,tanda_arah_k);

//////////
// Cari nomor Arah Terjauh //
//////////
        no_TS_terjauh=0;
        for(int cari_ts=1;cari_ts<JVD;cari_ts++)
        {
            if(fabs(TS[cari_ts])>fabs(TS[no_TS_terjauh]))
            {
                no_TS_terjauh=cari_ts;
            }
        }
    }
}

```



```

/*****
/* FILE : BARU.HPP
/* PUSTAKA PENANGANAN TITIK BARU
/*****

/*****
/* SUBPROGRAM UNTUK MENCARI TITIK BARU */
/*****
void cari_baru()
{
    ////////////////////////////////////////////////////
    // Inisialisasi Awal //
    ////////////////////////////////////////////////////
    iterasi_var=0;

    normalisasi_int(varnew_asli,var_b_jelek,var_k_jelek);
    fitch_best=0.;
    do
    {
        gotoxy(1,15);
        cprintf ("Pencarian arah baru ke - %d",(iterasi_var+1));
        cprintf ("          ");

        ////////////////////////////////////////////////////
        // Menentukan Titik X Pertama Baru //
        // Diambil Yang Arahnya Paling Jauh //
        ////////////////////////////////////////////////////
        varplus[no_TS_terjauh]=(iterasi_var+1.);

        ////////////////////////////////////////////////////
        // Variabel Baru Ditambahkan Sesuai Arahnya //
        ////////////////////////////////////////////////////
        varnew_f[no_TS_terjauh]=varnew_asli[no_TS_terjauh]
                                +varplus[no_TS_terjauh]
                                *arah[no_TS_terjauh];

        ////////////////////////////////////////////////////
        // Menentukan X Baru Lain //
        ////////////////////////////////////////////////////
        for (int icb=0;icb<JVD;icb++)
        {
            ////////////////////////////////////////////////////
            // Skip Bila icb Adalah Nomor arah Pencarian Terjauh //
            ////////////////////////////////////////////////////
            if(icb==no_TS_terjauh)
            {
                continue;
            }

            if(TS[no_TS_terjauh]!=0)
            {
                ////////////////////////////////////////////////////
                // Menghitung Penambahannya //
                ////////////////////////////////////////////////////
                varplus[icb]=fabs((TS[icb]/TS[no_TS_terjauh])
                                *varplus[no_TS_terjauh]);
                varnew_f[icb]=varnew_asli[icb]
                                +varplus[icb]*arah[icb];
            }
            else
            {
                varnew_f[icb]=varnew_asli[icb]
                                +((iterasi_var+1.)*arah[icb]);
            }
        }

        for(int icb=0;icb<JVD;icb++)
        {
            varnew[icb]=konversi(varnew_f[icb]);
        }
    }
}

```

```

////////////////////////////////////
// Memeriksa Apakah Titik Baru Tersebut Identik Dengan Titik M //
////////////////////////////////////
for(int ilp=0;ilp<JVD;ilp++)
{
    if(varnew[ilp]==TM[ilp])
    {
        if(ilp==(JVD-1))
        {
            lompat=1;
        }
    }
    else
    {
        lompat=0;
        break;
    }
}

////////////////////////////////////
// Lompat Dari Loop Apabila Indikator Pelompat bernilai 1 //
////////////////////////////////////
if(lompat==1)
{
    lompat=0;
    iterasi_var++;
    continue;
}

periksa_batas();

////////////////////////////////////
// Perubahan Variabel Umum Ke Variabel Balok Kolom //
////////////////////////////////////
unnormalisasi(varnew,var_b_cb,var_k_cb);

////////////////////////////////////
// Menghitung Kendala //
////////////////////////////////////
fitchb=Kendala_Harga(var_b_cb,var_k_cb);

if(fitchb>fitchb_best)
{
    fitchb_best=fitchb;
    for(int icb=0;icb<jum_balok;icb++)
    {
        for(int jcb=0;jcb<12;jcb++)
        {
            var_b_cb_best[jcb+(12*icb)]=var_b_cb[jcb+(12*icb)];
        }
    }
    for(int icb=0;icb<jum_kolom;icb++)
    {
        for(int jcb=0;jcb<5;jcb++)
        {
            var_k_cb_best[jcb+(5*icb)]=var_k_cb[jcb+(5*icb)];
        }
    }
    iterasi_var++;
}while(iterasi_var<(fabs(TS[no_TS_terjauh])*3));
}

/*****/
/* SUBPROGRAM UNTUK MEMERIKSA BATAS ATAS */
/* DAN BAWAH VARIABEL DESAIN BARU */
/* DAN BILA TERJADI PELANGGARAN MAKA */
/* VARIABEL ITU AKAN DICERMINKAN */
/*****/
void periksa_batas()

```

```

{
    for(int ipb=0;ipb<JVD;ipb++)
    {
        if((varnew[ipb])>(nvm[ipb]-1))
        {
            varnew[ipb]=(nvm[ipb]-1);
        }
        if(varnew[ipb]<0)
        {
            varnew[ipb]=0;
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENGGANTI VARIABEL TERJELEK DENGAN VARIABEL BARU */
*****/
void ganti_baru()
{
    for(int igbar=0;igbar<jum_balok;igbar++)
    {
        for(int jgbar=0;jgbar<12;jgbar++)
        {
            var_b[0][jgbar+(12*igbar)]=var_b_cb_best[jgbar+(12*igbar)];
        }
    }
    for(int igbar=0;igbar<jum_kolom;igbar++)
    {
        for(int jgbar=0;jgbar<5;jgbar++)
        {
            var_k[0][jgbar+(5*igbar)]=var_k_cb_best[jgbar+(5*igbar)];
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENYUSUTKAN VARIABEL DESAIN */
*****/
void penyusutan()
{
    for(int nkon=0;nkon<(JSTD-1);nkon++)
    {
        for(int igbar=0;igbar<jum_balok;igbar++)
        {
            for(int jgbar=0;jgbar<12;jgbar++)
            {
                var_b[nkon][jgbar+(12*igbar)]=
                konversi(0.5*(var_b[nkon][jgbar+(12*igbar)]
                +var_b[JSTD-1][jgbar+(12*igbar)]));
            }
        }
        for(int igbar=0;igbar<jum_kolom;igbar++)
        {
            for(int jgbar=0;jgbar<5;jgbar++)
            {
                var_k[nkon][jgbar+(5*igbar)]=
                konversi(0.5*(var_k[nkon][jgbar+(5*igbar)]
                +var_k[JSTD-1][jgbar+(5*igbar)]));
            }
        }
    }
}

```

```

/*****/
/* FILE : PENGURUTAN.HPP */
/* SUBPROGRAM UNTUK MENGURUTKAN DATA BERDASARKAN FITNESS */
/* DENGAN BANTUAN METODA BUBBLE SORT */
/*****/
void sort(float nilai1[],float nilai2[],float nilai3[]
          ,int urut_var_b[][mak],int urut_var_k[][mak]
          ,int n_array)
{
    ///////////////////////////////////
    // Metode Bubble Sort //
    ///////////////////////////////////

    ///////////////////////////////////
    // Nilai 1 Untuk Fitness //
    // Nilai 2 Untuk Kendala //
    // Nilai 3 Untuk Harga //
    // urut_var_b Untuk Variabel Balok //
    // urut_var_k Untuk Variabel Kolom //
    // n_array Adalah Jumlah Titik Yang Akan Diurutkan Fitnessnya //
    ///////////////////////////////////
    double dummy1=0.;
    float dummy2=0.;
    float dummy3=0.;
    int dummyvarb[mak];
    int dummyvark[mak];
    for (int dum1=0;dum1<JSTD;dum1++)
    {
        dummyvarb[dum1]=0;
        dummyvark[dum1]=0;
    }

    ///////////////////////////////////
    // Pemeriksaan Dimulai Dari Nilai Terkecil+1 //
    ///////////////////////////////////
    for(int isort=1;isort<n_array;++isort)
    {
        ///////////////////////////////////
        // Perbandingan Kepada Nilai-Nilai Sebelumnya //
        ///////////////////////////////////
        for(int jsort=0;jsort<=isort;++jsort)
        {
            ///////////////////////////////////
            // Bila Nilai Sekarang Lebih Kecil Daripada Nilai Sebelumnya //
            // Maka Dilakukan Penukaran Posisi Ke Tempat Nilai Sebelumnya //
            ///////////////////////////////////
            if(nilai1[isort]<nilai1[jsort])
            {
                ///////////////////////////////////
                // Pengisian Nilai Sekarang Ke Variabel Dummy //
                ///////////////////////////////////
                dummy1=nilai1[isort];
                dummy2=nilai2[isort];
                dummy3=nilai3[isort];
                for(int dvb=0;dvb<jum_balok;dvb++)
                {
                    for(int dnv_b=0;dnv_b<12;dnv_b++)
                    {
                        dummyvarb[dnv_b+(12*dvb)]
                        =urut_var_b[isort][dnv_b+(12*dvb)];
                    }
                }
                for(int dvk=0;dvk<jum_kolom;dvk++)
                {
                    for(int dnv_k=0;dnv_k<5;dnv_k++)
                    {
                        dummyvark[dnv_k+(5*dvk)]
                        =urut_var_k[isort][dnv_k+(5*dvk)];
                    }
                }
            }
        }
    }
}

```

```

////////////////////////////////////
// Pemindahan Nilai Yang Akan Digantikan Posisinya //
// Ke Posisi Sekarang //
////////////////////////////////////
nilai1[isort]=nilai1[jsort];
nilai2[isort]=nilai2[jsort];
nilai3[isort]=nilai3[jsort];
for(int dvb=0;dvb<jum_balok;dvb++)
{
    for(int dnv_b=0;dnv_b<12;dnv_b++)
    {
        urut_var_b[isort][dnv_b+(12*dvb)]
        =urut_var_b[jsort][dnv_b+(12*dvb)];
    }
}
for(int dvk=0;dvk<jum_kolom;dvk++)
{
    for(int dnv_k=0;dnv_k<5;dnv_k++)
    {
        urut_var_k[isort][dnv_k+(5*dvk)]
        =urut_var_k[jsort][dnv_k+(5*dvk)];
    }
}

////////////////////////////////////
// Pemindahan Nilai Sekarang Ke Posisi Yang Digantikan //
////////////////////////////////////
nilai1[jsort]=dummy1;
nilai2[jsort]=dummy2;
nilai3[jsort]=dummy3;
for(int dvb=0;dvb<jum_balok;dvb++)
{
    for(int dnv_b=0;dnv_b<12;dnv_b++)
    {
        urut_var_b[jsort][dnv_b+(12*dvb)]
        =dummyvarb[dnv_b+(12*dvb)];
    }
}
for(int dvk=0;dvk<jum_kolom;dvk++)
{
    for(int dnv_k=0;dnv_k<5;dnv_k++)
    {
        urut_var_k[jsort][dnv_k+(5*dvk)]
        =dummyvark[dnv_k+(5*dvk)];
    }
}

////////////////////////////////////
// Proses Penukaran Selesai //
// Melanjutkan Nilai Selanjutnya //
////////////////////////////////////
}
}

////////////////////////////////////
// Sesudah Diurutkan Nilai Terjelek Terdapat Pada Array 0 //
// Nilai Terbaik Terdapat Pada (n_array-1) //
////////////////////////////////////
}

```

```

/*****
/* FILE : TAMPILAN.HPP
*/
/* KUMPULAN SUBPROGRAM PENANGANAN TAMPILAN KE LAYAR
*/
*****/

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN MENU UTAMA */
*****/
void menu_utama()
{
    int pilih;
    do
    {
        textbackground(1);
        textcolor(LIGHTGREEN);
        clrscr();
        about();
        cout << " 1. Input data awal ke file\n";
        cout << " 2. Input data beban ke file\n";
        cout << " 3. Melihat isi file input\n";
        cout << " 4. Mengoptimasi struktur\n";
        cout << " 5. Keluar\n";
        cout << " Pilihan (1-5) = ";
        struct time t;

        gettime(&t);
        textcolor(YELLOW);
        gotoxy(61,6);
        cprintf("Time : %2d:%02d:%02d.%02d\n",
            t.ti_hour, t.ti_min, t.ti_sec, t.ti_hund);
        gotoxy(19,12);
        cin >> pilih;

        if(pilih==1)
        {
            input_data();
        }
        if(pilih==2)
        {
            load_data(); // Memasukan data beban
            cout << " Data beban sudah diisi...\n";
        }
        if(pilih==3)
        {
            cout << " Nama file input yang akan dibaca = ";
            cin >> fi;
            strcpy(finput,fi);
            strcat(finput, ".inp");
            strcpy(fbeban,fi);
            strcat(fbeban, ".bbn");
            baca_data();
            baca_beban();
            do
            {
                clrscr();
                about();
                cout << " 1. Data umum\n";
                cout << " 2. Koordinat titik kumpul\n";
                cout << " 3. Informasi batang\n";
                cout << " 4. Pengekang titik kumpul\n";
                cout << " 5. Gaya ujung batang terkekang\n";
                cout << " 6. Beban pada joint\n";
                cout << " 7. Keluar ke menu utama\n";
                cout << " Pilihan (1-7) = ";
                cin >> pilih;
                clrscr();
                if(pilih==1)
                {
                    output_parameter_struktural();
                }
            }
        }
    }
}

```

```

        if(pilih==2)
        {
            output_koordinat_titik_kumpul();
        }
        if(pilih==3)
        {
            output_informasi_batang();
        }
        if(pilih==4)
        {
            output_pengekang_titik_kumpul();
        }
        if(pilih==5)
        {
            output_beban_batang();
        }
        if(pilih==6)
        {
            output_beban_titik();
        }
        if(pilih==7)
        {
            cout << "  Anda keluar ke Menu Utama\n";
        }
        cout << "\n  Tekan Sembarang Tombol\n";
        getch();
    }while(pilih!=7);
}
if(pilih==4)
{
    cout << "  Nama file stuktur (tanpa ekstensi) = ";
    cin >> fi;
    strcpy(fininput,fi);
    strcat(fininput, ".inp"); // Nama file generik
    strcpy(fsisi,fi);
        strcat(fsisi, ".isd"); // input sisi diskrit
        strcpy(fdial,fi);
        strcat(fdial, ".idl"); // input diameter tulangan lentur
        strcpy(fjtl,fi);
        strcat(fjtl, ".ijl"); // input jumlah tulangan lentur
        strcpy(fdias,fi);
        strcat(fdias, ".ids"); // input diameter tulangan sengkang
        strcpy(fjts,fi);
        strcat(fjts, ".ijs"); // input jarak tulangan sengkang
    strcpy(fbeban,fi);
    strcat(fbeban, ".bbn"); // input beban pada batang
    strcpy(fhistory,fi);
    strcat(fhistory, ".his"); // Keluaran dari riwayat optimasi
    strcpy(foptimasi,fi);
    strcat(foptimasi, ".opt"); // Keluaran dari hasil optimasi
    strcpy(fstruktur,fi);
    strcat(fstruktur, ".str"); // Keluaran dari hasil analisa struktur
    strcpy(fkendala,fi);
    strcat(fkendala, ".kdl"); // Keluaran kendala pada struktur
    strcpy(fininformasi,fi);
    strcat(fininformasi, ".inf"); // Keluaran informasi masukan

    cout << "  file input = " << fininput << endl;
    cout << "  file beban = " << fbeban << endl;
    cout << "  Harga Beton (Rp./m^3) = ";
        cin >> harga_beton;
        cout << "  Harga Besi (Rp./kg) = ";
        cin >> harga_besi;
        cout << "  Tebal selimut kolom (minimum 40 mm) = ";
        cin >> selimut_kolom;
    cout << "  Tebal selimut balok (minimum 40 mm) = ";
        cin >> selimut_balok;
    cout << "  Faktor finalti = ";
    cin >> finalti;
    cout << "  Iterasi maksimum = ";

```

```

    cin >> j_iterasi_mak;
    cout << "    Faktor penambah jumlah variabel desain = ";
    cin >> fak_plus;
    do
    {
        cout << "    Faktor pengali jumlah variabel desain = ";
        cin >> fak_kali;
        if(fak_kali<=0)
        {
            "    Faktor pengali minimal diisi 1 !\n";
        }
    }while(fak_kali<=0);
    optimasi();
}
if(pilih==5)
{
    cout << "\n                                Anda keluar dari program\n";
}
} while(pilih!=5);
}

/*****/
/* SUBPROGRAM UNTUK MENAMPILKAN INFORMASI PROGRAM */
/*****/
void about()
{
    cout << "*****";
    cout << "*****";
    cout << "
"
    << "Program Optimasi Beton Bertulang Pada Struktur Portal Ruang\n";
    cout << "
"
    << "Oleh : Yohan Naftali\n";
    cout << "
"
    << "7712/TS\n";
    cout << "*****";
    cout << "*****\n";
}

```



```

/*****
/* FILE : CETAK.HPP
*/
/* SUBPROGRAM UNTUK MENCETAK HASIL AKHIR OPTIMASI
/* DITULIS OLEH YOHAN NAFTALI (JULI 1999)
*****/
void cetak_akhir()
{
    ofstream hopt(foptimasi);
    ofstream hstr(fstruktur);
    ofstream hkdl(fkendala);
    ofstream hinf(finformasi);

    ////////////////////////////////////
    // Pembacaan Data Umum Struktur //
    ////////////////////////////////////
    baca_data();

    ////////////////////////////////////
    // Membaca Beban Yang Bekerja Pada Struktur //
    ////////////////////////////////////
    baca_beban();

    ////////////////////////////////////
    // Mencetak Informasi Masukkan Ke File //
    ////////////////////////////////////
    hinf << " Struktur Portal Ruang " << ISN << "\n\n";
    hinf << " Parameter Struktur\n";
    hinf << " Jumlah batang : " << M << endl;
    hinf << " DOF : " << N << endl;
    hinf << " Jumlah joint : " << NJ << endl;
    hinf << " Jumlah pengeang tumpuan : " << NR << endl;
    hinf << " Jumlah titik kumpul yang dikeang : " << NRJ << endl;
    hinf << " Modulus Elastisitas aksial : " << E << " N/m^2\n";
    hinf << " Modulus Geser : " << G << " N/m^2\n\n";
    hinf << " Properti Elemen Material\n";
    hinf << " Kuat desak beton karakteristik : " << FC << " MPa\n";
    hinf << " Kuat tarik baja tulangan : " << FY << " MPa\n";
    hinf << " Kuat tarik tulangan sengkang : " << FYS << " MPa\n";

    hinf << "\n Koordinat Titik Kumpul (m)\n";
    hinf << " Titik X Y Z \n";
    for(int kout=1;kout<=NJ;kout++)
    {
        hinf << setiosflags(ios::left);
        hinf << " " << setw(10) << kout
            << setw(14) << X[kout]
            << setw(14) << Y[kout]
            << setw(14) << Z[kout] << endl;
    }

    hinf << "\n Informasi Batang \n";
    hinf << " Batang JJ JK IA\n";
    for(int iout=1;iout<=M;iout++)
    {
        hinf << setiosflags(ios::left);
        hinf << " " << setw(8) << iout
            << setw(8) << JJ[iout]
            << setw(8) << JK[iout]
            << setw(3) << IA[iout] << endl;
        if(IA[iout] != 0)
        {
            hinf << setiosflags(ios::left);
            hinf << " XP = " << setw(12) << XP[iout];
            hinf << " YP = " << setw(12) << YP[iout] << '\t';
            hinf << " ZP = " << setw(12) << ZP[iout] << endl;
            hinf << endl;
        }
    }

    hinf << "\n Pengeang Titik Kumpul \n";

```

```

hinf << " Titik JR1 JR2 JR3 JR4 JR5 JR6\n";
for (int iout=1;iout<=NRJ;iout++)
{
    hinf << setiosflags(ios::left);
    hinf << " " << setw(10)<<T_K[iout]
        << setw(6) << JRL[6*T_K[iout]-5]
        << setw(6) << JRL[6*T_K[iout]-4]
        << setw(6) << JRL[6*T_K[iout]-3]
        << setw(6) << JRL[6*T_K[iout]-2]
        << setw(6) << JRL[6*T_K[iout]-1]
        << setw(6) << JRL[6*T_K[iout]] << endl;
}

hinf << "\n Gaya di Ujung Batang Terkekang Akibat Beban (Nm)\n"
<< " Batang AML1 AML2 AML3 "
<< " << "AML4 AML5 AML6\n"
<< " AML7 AML8 AML9 "
<< "AML10 AML11 AML12\n";
for(int kout=1;kout<=M;kout++)
{
    hinf << setiosflags(ios::left);
    hinf << " " << setw(8) << kout
        << setw(12) << AML[1][kout]
        << setw(12) << AML[2][kout]
        << setw(12) << AML[3][kout]
        << setw(12) << AML[4][kout]
        << setw(12) << AML[5][kout]
        << setw(12) << AML[6][kout] << endl;
    hinf << " "
        << setw(12) << AML[7][kout]
        << setw(12) << AML[8][kout]
        << setw(12) << AML[9][kout]
        << setw(12) << AML[10][kout]
        << setw(12) << AML[11][kout]
        << setw(12) << AML[12][kout] << endl;
}

hinf << "\n Beban Titik (N)\n"
<< " Titik Arah 1 Arah 2 Arah 3 "
<< "Arah 4 Arah 5 Arah 6\n";
for(int kout=1;kout<=NJ;kout++)
{
    hinf << setiosflags(ios::left);
    hinf << " " << setw(8) << kout
        << setw(12) << AJ[((6*kout)-5)]
        << setw(12) << AJ[((6*kout)-4)]
        << setw(12) << AJ[((6*kout)-3)]
        << setw(12) << AJ[((6*kout)-2)]
        << setw(12) << AJ[((6*kout)-1)]
        << setw(12) << AJ[(6*kout)] << endl;
}

//////////
// Menghitung Gaya-Gaya Struktur //
//////////
inersia();
struktur();

//////////
// Cetak Hasil Analisa Struktur //
//////////
hstr << "Hasil Analisa Struktur Dengan Metoda Kekakuan\n";
hstr << "Dikembangkan dari Weaver & Gere\n";
hstr << "Oleh Yohan Naftali 1999\n\n";
hstr << "Nama File Generik : " << fi << endl;
hopt << "Nama Struktur : " << ISN << endl;
hstr << "Jumlah batang : " << M << endl;
hstr << "Jumlah titik kumpul : " << NJ << endl;
hstr << "Modulus elastisitas tarik/tekan : "
<< (E*1.E-6) << " MPa" << endl;
hstr << "Modulus elastisitas geser G : "

```

```

        << (G*1.E-6) << " MPa" << endl;
hstr << "\nPerpindahan Titik Kumpul\n";
hstr << "Titik  DJ1      DJ2      DJ3  ";
hstr << "      DJ4      DJ5      DJ6\n";
for(int cst=1; cst<=NJ; cst++)
{
    hstr << setiosflags(ios::left);
    hstr << setw(7) << cst;
    hstr << setiosflags( ios::fixed | ios:: showpos);
    hstr << setw(12) << DJ[6*cst-5] << setw(12);
    hstr << DJ[6*cst-4] << setw(12) << DJ[6*cst-3] << setw(12);
    hstr << DJ[6*cst-2] << setw(12) << DJ[6*cst-1] << setw(12);
    hstr << DJ[6*cst] << endl;
hstr << resetiosflags( ios::fixed | ios:: showpos);
}

hstr << "\nGaya Ujung Batang\n";
hstr << "Batang  AM1      AM2      AM3      ";
hstr << "AM4      AM5      AM6\n";
hstr << "      AM7      AM8      AM9      ";
hstr << "AM10     AM11     AM12\n";

for(int ih=1; ih<=M; ih++)
{
    periksa_batang(ih);
    isi_matrik_kekakuan(ih);
    indeks_batang(ih);
    for(int jh=1; jh<=MD; jh++)
    {
        AMD[jh]=0.;
        for(int kh=1; kh<=MD; kh++)
        {
            AMD[jh]=AMD[jh]+SMRT[jh][kh]*DJ[ IM[kh]];
        }
        AM[ih][jh]=AML[jh][ih]+AMD[jh];
    }
hstr << setiosflags(ios::left);
hstr << setw(8) << ih;
hstr << setw(12) << AM[ih][1]
    << setw(12) << AM[ih][2]
    << setw(12) << AM[ih][3]
    << setw(12) << AM[ih][4]
    << setw(12) << AM[ih][5]
    << setw(12) << AM[ih][6] << '\n';
    hstr << "
    << setw(12) << AM[ih][7]
    << setw(12) << AM[ih][8]
    << setw(12) << AM[ih][9]
    << setw(12) << AM[ih][10]
    << setw(12) << AM[ih][11]
    << setw(12) << AM[ih][12] << '\n';
}

hstr << "\nReaksi Tumpuan\n";
hstr << "Titik  AR1      AR2      AR3";
hstr << "      AR4      AR5      AR6 \n";

for(int ih=1; ih<=NJ; ih++)
{
    int J1=(6*ih-5),J2=(6*ih-4),J3=(6*ih-3),
    J4=(6*ih-2),J5=(6*ih-1),J6=(6*ih);
    int N1=JRL[J1]+JRL[J2]+JRL[J3]+JRL[J4]+JRL[J5]+JRL[J6];
    if(N1!=0)
    {
        hstr << setiosflags(ios::left);
        hstr << setw(8) << ih;

hstr << setw(12) << AR[J1]
    << setw(12) << AR[J2]
    << setw(12) << AR[J3];
        hstr << setw(12) << AR[J4]

```

```

        << setw(12) << AR[J5]
    << setw(12) << AR[J6] << '\n';
    }
}
hstr.close();

////////////////////
// Cetak Hasil Optimasi //
////////////////////
hopt << "Hasil Optimasi Beton Bertulang "
    << "Pada Struktur Portal Ruang\n";
hopt << "Metoda Optimasi : Flexible Polyhedron\n";
hopt << "Jumlah Variabel Desain : " << JVD << endl;
hopt << "Jumlah Struktur Desain : " << JSTD << endl;
hopt << "Oleh Johan Naftali 1999\n\n";
hopt << "Nama File Generik : " << fi << endl;
hopt << "Nama Struktur : " << ISN << endl;
hopt << "Jumlah batang : " << M << endl;
hopt << "Jumlah titik kumpul : " << NJ << endl;
hopt << "Modulus elastisitas tarik/tekan : "
    << (E*1.E-6) << " MPa" << endl;
hopt << "Modulus elastisitas geser G : "
    << (G*1.E-6) << " MPa" << endl;
hopt << "Kuat desak Karakteristik Beton : " << FC << " MPa\n";
hopt << "Kuat tarik baja tulangan lentur : " << FY << " MPa\n";
hopt << "Kuat tarik baja tulangan sengkang : " << FYS << " MPa\n";

////////////////////
// Cetak Header Kendala //
////////////////////
hkdl << "Kendala Pada Struktur\n";
hkdl << "Oleh Johan Naftali 1999\n\n";
hkdl << "Nama File Generik : " << fi << endl;
hkdl << "Nama Struktur : " << ISN << endl;

no_struktur=(JSTD-1);
for(int oio=0;oio<jum_balok;oio++)
{
    //////////////////////
    // Memeriksa Kemiringan Batang //
    //////////////////////
    periksa_balok(no_balok[oio]);

    //////////////////////
    // Pengisian Elemen Balok //
    //////////////////////
    isi_element_balok(oio);

    hopt << "-----\n";
    hopt << "Balok " << (oio+1) << " " << B << " x " << H << endl;
    hopt << "Nomor Batang : " << no_balok[oio] << endl;
    hopt << "Panjang Bentang : " << L << " m" << endl;
    hopt << "Beban Total : " << W[no_balok[oio]] << " N/m\n";
    hopt << "Berat Sendiri : " << W_Balok[no_balok[oio]] << " N/m\n";
    hopt << "Momen kiri : "
        << ((MTUM_KI[no_balok[oio]]/(0.8)) << " Nmm" << endl;
    hopt << "Momen tengah : "
        << ((MLAP[no_balok[oio]]/(0.8)) << " Nmm" << endl;
    hopt << "Momen kanan : "
        << ((MTUM_KA[no_balok[oio]]/(0.8)) << " Nmm" << endl;
    hopt << "Geser kiri : "
        << (GESER_KI[no_balok[oio]]/0.8) << " N" << endl;
    hopt << "Geser kanan : "
        << (GESER_KA[no_balok[oio]]/0.8) << " N" << endl;

    //////////////////////
    // Daerah Lapangan //
    //////////////////////
    elemen_lapangan(oio);
    balok_lapangan;
}

```

```

lendutan(no_balok[oio]);

////////////////////
// Cetak Hasil Optimasi //
////////////////////
hopt << "Lendutan Tengah Bentang : " << LENDUTAN
      << " mm" << endl;
hopt << "Lendutan ijin " << LENDUTAN_IJIN << " mm" << endl;
hopt << "Sengkang " << DIAS << " - " << Jarak_S << endl;
hopt << "Jarak sengkang maksimum : " << Sref << " mm\n";

hopt << "\nDaerah Lapangan\n";
hopt << "Tulangan tarik " << NL1 << " D " << DIA1 << endl;
hopt << "Tulangan desak " << NL2 << " D " << DIA2 << endl;
hopt << "Volume beton : " << volume_beton << " m^3" << endl;
hopt << "Berat Tulangan utama : " << berat_besi << " kg\n";
hopt << "Berat Tulangan geser : " << berat_sengkang << " kg\n";
hopt << "Harga balok daerah lapangan : " << harga << endl;
hopt << "Momen yang membebani : " << MU << " Nmm" << endl;
hopt << "Momen yang dapat ditahan : " << FMU << " Nmm" << endl;
hopt << endl;

////////////////////
// Cetak Kendala //
////////////////////
hkdl << "\nBalok " << (oio+1) << endl;
hkdl << "Nomor Batang : " << no_balok[oio] << endl;
hkdl << "Kendala akibat lendutan : "
      << kendala_lendutan << endl;
hkdl << "\nDaerah Lapangan\n";
hkdl << "Kendala rasio penulangan "
      << kendala_rho << endl;
hkdl << "Kendala sengkang : "
      << kendala_sb << endl;
      hkdl << "Kendala momen lentur : "
      << kendala_M << endl;

////////////////////
// Daerah Tumpuan //
////////////////////
elemen_tumpuan(oio);
balok tumpuan;

////////////////////
// Cetak Hasil Optimasi //
////////////////////
hopt << "Daerah Tumpuan\n";
hopt << "Tulangan desak " << NL2 << " D " << DIA2 << endl;
hopt << "Tulangan tarik " << NL1 << " D " << DIA1 << endl;
hopt << "Volume beton : " << volume_beton << " m^3" << endl;
hopt << "Berat Tulangan utama : " << berat_besi << " kg\n";
hopt << "Berat Tulangan geser : " << berat_sengkang << " kg\n";
hopt << "Harga balok pada tumpuan kiri + kanan : " << harga << endl;
hopt << "Momen yang membebani : " << MU << " Nmm" << endl;
hopt << "Momen yang dapat ditahan : " << FMU << " Nmm" << endl;
hopt << endl;

////////////////////
// Cetak Kendala //
////////////////////
hkdl << "\nDaerah Tumpuan\n";
hkdl << "Kendala rasio penulangan "
      << kendala_rho << endl;
hkdl << "Kendala sengkang : "
      << kendala_sb << endl;
      hkdl << "Kendala momen lentur : "
      << kendala_M << endl;
}

```

```

////////////////////////////////////
// Menghitung Kendala Dan Harga Kolom //
////////////////////////////////////
for(int oio=0;oio<jum_kolom;oio++)
{
////////////////////////////////////
// Memeriksa Kemiringan Batang //
////////////////////////////////////
periksa_batang(no_kolom[oio]);

////////////////////////////////////
// Pengisian Elemen Balok //
////////////////////////////////////
isi_elemen_kolom(oio);

////////////////////////////////////
// Membangkitkan Kolom //
////////////////////////////////////
kolom_bangkit;

////////////////////////////////////
// Cetak Hasil Optimasi //
////////////////////////////////////
hopt << "-----\n";
hopt << "Kolom " << (oio+1) << " "
    << sisi << " x " << sisi << endl;
hopt << "Nomor Batang : " << no_kolom[oio] << endl;
hopt << "Tulangan utama "
    << (4*N_DIA-4) << " D " << DIA << endl;
hopt << "Jarak antar tulangan : " << jarak_antar_tulangan << " mm\n";
hopt << "Sengkang " << DIAS << " - " << Jarak_S << endl;
hopt << "Jarak sengkang maksimum : " << Sref << " mm\n";

hopt << "Volume beton : " << volume_beton << " m^3" << endl;
hopt << "Berat Tulangan utama : " << berat_besi << " kg\n";
hopt << "Berat Tulangan geser : " << berat_sengkang << " kg\n";
hopt << "Harga kolom : " << harga << endl;

hopt << "Berat Kolom : " << (-P_Kolom[no_kolom[oio]]) << " N\n";
hopt << "Gaya aksial : "
    << (PK[no_kolom[oio]]/0.8) << " N" << endl;
hopt << "Gaya aksial yang dapat ditahan = "
    << FPU << " N\n";
hopt << "Momen arah X : "
    << (MKX[no_kolom[oio]]/0.8) << " Nmm" << endl;
hopt << "Momen arah Y : "
    << (MKY[no_kolom[oio]]/0.8) << " Nmm" << endl;
hopt << "Momen yang dapat ditahan : "
    << FMU << " Nmm\n";
hopt << "Geser pada kolom : "
    << (GK[no_kolom[oio]]/0.8) << " N" << endl;
hopt << endl;

////////////////////////////////////
// Cetak Kendala //
////////////////////////////////////
hkdl << "\nKolom " << (oio+1) << endl;
hkdl << "Nomor Batang : " << no_kolom[oio] << endl;
hkdl << "\nKendala gaya : "
    << kendala_gaya << endl;
hkdl << "Kendala rasio penulangan : "
    << kendala_r << endl;
hkdl << "Kendala jarak tulangan : "
    << kendala_tul << endl;
hkdl << "Kendala sengkang : "
    << kendala_sengkang << endl;
hkdl << "Kendala kelangsingan kolom : "
    << kendala_kelangsingan << endl;
}
hopt << "\nHarga Beton Rp." << harga_beton << " /m^3\n";

```

```
        hopt << "Harga Besi Rp." << harga_besi << " /kg\n";
        hopt << "Tebal selimut kolom " << selimut_kolom << " mm\n";
    hopt << "Tebal selimut balok " << selimut_balok << " mm\n";
    hopt << "Faktor finalti : " << finalti << endl;
    hopt << "JSTD = (JVD*" << fak_kali << ")+ " << fak_plus << endl;
}
```

```

# File ORCISF.MAK
# Konfigurasi pada saat pembuatan executable file
#
.AUTODEPEND

#
# Borland C++ tools
#
IMPLIB = Implib
BCC32 = Bcc32 +BccW32.cfg
BCC32I = Bcc32i +BccW32.cfg
TLINK32 = TLink32
ILINK32 = Ilink32
TLIB = TLib
BRC32 = Brc32
TASM32 = Tasm32
#
# IDE macros
#

#
# Options
#
IDE_LinkFLAGS32 = -LC:\BC5\LIB
LinkerLocalOptsAtC32_orcisfdexe = -Tpe -ap -c
ResLocalOptsAtC32_orcisfdexe =
BLocalOptsAtC32_orcisfdexe =
CompInheritOptsAt_orcisfdexe = -IC:\BC5\INCLUDE -D_RTLDLL;
LinkerInheritOptsAt_orcisfdexe = -x
LinkerOptsAt_orcisfdexe = $(LinkerLocalOptsAtC32_orcisfdexe)
ResOptsAt_orcisfdexe = $(ResLocalOptsAtC32_orcisfdexe)
BOptsAt_orcisfdexe = $(BLocalOptsAtC32_orcisfdexe)

#
# Dependency List
#
Dep_ORCISF = \
    orcisf.exe

ORCISF : BccW32.cfg $(Dep_ORCISF)
    echo MakeNode

Dep_orcisfdexe = \
    orcisf.obj

orcisf.exe : $(Dep_orcisfdexe)
    $(ILINK32) @&&|
    /v $(IDE_LinkFLAGS32) $(LinkerOptsAt_orcisfdexe)
    $(LinkerInheritOptsAt_orcisfdexe) +
    C:\BC5\LIB\c0x32.obj+
    orcisf.obj
    $<,$*
    C:\BC5\LIB\import32.lib+
    C:\BC5\LIB\cw32i.lib

|
orcisf.obj : orcisf.cpp
    $(BCC32) -c @&&|
    $(CompOptsAt_orcisfdexe) $(CompInheritOptsAt_orcisfdexe) -o$@ orcisf.cpp
|

# Compiler configuration file
BccW32.cfg :
    Copy &&|
-w
-R
-v

```



```
-WM-  
-vi  
-H  
-H=ORCISF.csm  
-WC  
-O2  
-OS  
-5  
-a8  
| $@
```

LAMPIRAN 2

```

/*****
/* FILE : BORLANDC.HPP
/* HEADER PUSTAKA BORLAND C++ VERSI 5.02 YANG DIPAKAI
/* PUSTAKA YANG DIPANGGIL DIBUAT OLEH BORLAND INTERNATIONAL
*****/

#include <math.h>           // Pustaka Pengolahan Matematika
#include <_defs.h>          // Definisi Umum Untuk Pointer Dan Conventions
#include <stddef.h>         // Definisi Untuk Tipe Umum Dan Null
#include <_null.h>          // Definisi Null Untuk Pointer
#include <time.h>           // Pustaka Operasi Waktu
#include <dos.h>            // Fungsi Fasilitas Disk Operating System
#include <iostream.h>       // Stream Input Output Ke Layar
#include <string.h>         // Pustaka Pembantu Operasi String
#include <fstream.h>        // Stream Untuk Menangani Masalah File
#include <stdlib.h>         // Standar Pustaka C
#include <conio.h>          // Operasi Rutin Input Output Pada Konsol
#include <iomanip.h>        // Operasi Manipulasi Input Output

```

LAMPIRAN 3

```

/*****
/* FILE : HEADER.HPP                                     */
/* KUMPULAN HEADER PROGRAM OPTIMASI BETON BERTULANG PADA STRUKTUR RUANG */
/* DENGAN METODA FLEXIBLE POLYHEDRON                               */
*****/

/*****/
/* HEADER KERNEL */
/*****/
#include "Proto.hpp"
#include "Variabel.hpp"
#include "InOut.hpp"

/*****/
/* HEADER MEKANIKA REKAYASA PORTAL RUANG DENGAN METODA KEKAKUAN */
/*****/
#include "Struktur.hpp"
#include "Pembebanan.hpp"
#include "Solver.hpp"

/*****/
/* HEADER ANALISA BETON BERTULANG */
/*****/
#include "Kolom.hpp"
#include "Balok.hpp"
#include "Elemen.hpp"

/*****/
/* HEADER OPTIMASI DENGAN METODA FLEXIBLE POLYHEDRON */
/*****/
#include "Polyhedron.hpp"
#include "Penormalan.hpp"
#include "Pengacakan.hpp"
#include "Diskritisasi.hpp"
#include "Kendala.hpp"
#include "Telusur.hpp"
#include "Baru.hpp"
#include "Pengurutan.hpp"

/*****/
/* HEADER SHELL */
/*****/
#include "Tampilan.hpp"
#include "Cetak.hpp"

```

LAMPIRAN 4

```

/*****
/* FILE : PROTO.HPP
/* PROTOTYPE OPTIMASI STRUKTUR BETON BERTULANG PADA PORTAL RUANG
*****/

/*****/
/* PROTOTYPE PROGRAM UMUM */
/*****/
void menu_utama();
void about();
void input_data();
void input_data_umum();
void input_data_diskrit();

/*****/
/* PROTOTYPE PROGRAM MEKANIKA REKAYASA STRUKTUR DENGAN METODA KEKAKUAN */
/*****/
void baca_data();
void load_data();
void kosong_beban();
void baca_beban();
void struktur();
void inersia();
void kekakuan_batang();
void beban();
void berat_sendiri();
void isi_matrik_kekakuan(int imk);
void indeks_batang(int ib);
void periksa_batang(int b);
void banfac(int N,int NB,float A_SFF[][mak]);
void bansol(int N,int NB,float U_SFF[][mak],float B_AC[],float X_DF[]);
void hasil();
void output_parameter_struktural();
void output_koordinat_titik_kumpul();
void output_informasi_batang();
void output_pengekang_titik_kumpul();
void output_beban_batang();
void output_beban_titik();

/*****/
/* PROTOTYPE PROGRAM BETON BERTULANG */
/*****/
void isi_elemen_balok(int no_el_balok);
void elemen_lapangan(int no_el_balok);
void elemen_tumpuan(int no_el_balok);
void isi_elemen_kolom(int no_el_kolom);
void lendutan(int no_batang_l);

/*****/
/* PROTOTYPE PROGRAM OPTIMASI DENGAN METODA FLEXIBLE POLYHEDRON */
/*****/
float Kendala_Harga(int var_b_nya[],int var_k_nya[]);
float isi(int no_data,float kelompok_data[]);
void optimasi();
void load_batas_atas();
void acak_variabel();
void randomisasi();
void cari_struktur_awal();
void cari_baru();
void periksa_batas();
void penelusuran();
void ganti_baru();
void penyusutan();
void normalisasi_float(float var_nor[],float var_bv[],float var_kv[]);
void normalisasi_int(int var_nor[],int var_bv[],int var_kv[]);
void unnormalisasi(float var_nor[],float var_bv[],float var_kv[]);
void cetak_akhir();
void sort(float nilai1[],float nilai2[],float nilai3[],int urut_var_b[][mak],int urut_var_k[][mak],int n_array);
int konversi(float bil_asli);
```

LAMPIRAN 5

```

/*****
/* FILE : VARIABEL.HPP
/* PENDEKLARASIAN VARIABEL GLOBAL
*****/

/*****/
/* MAKRO VARIABEL */
/*****/
#define pi 3.14 // Phi lingkaran
#define teta 0.8 // Faktor reduksi kekuatan
#define limit_nol 1.E-3 // Penetapan angka kecil mendekati nol
#define bj_besi 7850. // berat jenis besi (kg/m^3)
#define MD 12 // Jumlah perpindahan untuk satu batang

/*****/
/* VARIABEL UMUM */
/*****/
char fi[20]; // Input nama file generik tanpa ekstensi
char finput[20]; // File *.inp untuk data umum struktur
char fsisi[20]; // File *.isd untuk data sisi penampang
char fdial[20]; // File *.idl untuk diameter tulangan utama
char fjtl[20]; // File *.ijl untuk jumlah tulangan utama
char fdias[20]; // File *.ids untuk diameter tulangan sengkang
char fjts[20]; // File *.ijs untuk jarak tulangan sengkang
char fbeban[20]; // File *.bbn untuk data beban
char fhstory[20]; // File *.his untuk mencatat riwayat optimasi
char foptimasi[20]; // File *.opt untuk keluaran hasil optimasi
char fstruktur[20]; // File *.str untuk hasil analisa struktur
char fkendala[20]; // File *.kdl untuk keluaran kendala struktur
char finformasi[20]; // File *.inf untuk keluaran informasi masukkan
char SubName[80]; // Informasi nama sub masukkan
char ISN[80]; // Informasi nama struktur

/*****/
/* VARIABEL MEKANIKA REKAYASA UNTUK STRUKTUR PORTAL RUANG */
/*****/
int NB; // Setengah lebar jalur matrik kekakuan
int M; // Jumlah batang
int NJ; // Jumlah titik kumpul
int NR; // Jumlah Pengekang tumpuan
int NRJ; // Jumlah titik kumpul yang dikekang
int JRL[mak]; // Daftar pengekang titik kumpul
int T_K[mak]; // Daftar titik kumpul yang dikekang
int ND; // Jumlah koordinat perpindahan untuk semua titik kumpul
int N; // Jumlah derajat kebebasan
int IA[mak]; // Notasi penunjuk nol atau tidaknya sudut alfa
int JJ[mak]; // Penunjuk untuk ujung j
int JK[mak]; // penunjuk untuk ujung k
int ID[mak]; // Indeks perpindahan untuk titik kumpul
int IR,IC; // Indeks baris dan kolom
int IM[mak]; // Indeks perpindahan batang
int LML[mak]; // Tabel batang yang dibebani

float E; // Modulus elastisitas
float G; // Modulus Puntir
float X[mak]; // Koordinat struktur pada arah x ( _ ) | y
float Y[mak]; // Koordinat struktur pada arah y ( | ) | _
float Z[mak]; // Koordinat struktur pada arah z ( / ) z/ x
float b[mak]; // Lebar penampang
float h[mak]; // Tinggi penampang
float AX[mak]; // Luas Penampang
float XI[mak]; // Konstanta puntir batang
float YI[mak]; // Momen inersia terhadap sumbu y batang
float ZI[mak]; // Momen inersia terhadap sumbu z batang
float EL[mak]; // Panjang penampang

float CX; // Kosinus arah x
float Cy; // Kosinus arah y
float CZ; // Kosinus arah z
float CXZ; // Penunjuk balok atau kolom
float XP[mak]; // Koordinat x dari titik P (m)

```

LAMPIRAN 5

```
float YP[mak]; // Koordinat y dari titik P (m)
float ZP[mak]; // Koordinat z dari titik P (m)
float XPS;      // Koordinat xs dari titik P (m)
float YPS;      // Koordinat ys dari titik P (m)
float ZPS;      // Koordinat zs dari titik P (m)
float YPG;      // Koordinat yg dari titik P (m)
float ZPG;      // Koordinat zg dari titik P (m)
float COSA;     // Cosinus sudut alfa
float SINA;     // Sinus sudut alfa

float R11[mak]; // ~
float R12[mak]; // |
float R13[mak]; // |
float R21[mak]; // |
float R22[mak]; // -- Matrik Rotasi
float R23[mak]; // |
float R31[mak]; // |
float R32[mak]; // |
float R33[mak]; // ~

float SM[13][13]; // Matrik kekakuan batang lokal
float SMRT[13][13]; // Hasil perkalian matrik SM dan matrik Rotasi
float SMS[mak][mak]; // Matrik kekakuan batang global
float SFF[mak][mak]; // Matrik kekakuan untuk perpindahan bebas
float DF[mak]; // Perpindahan titik dalam sumbu global (m)
float AJ[mak]; // Aksi beban pada titik kumpul dalam arah global
float AML[13][mak]; // Gaya ujung batang terkekang dalam arah lokal
float AE[mak]; // Beban titik kumpul ekuivalen dalam arah global
float AC[mak]; // Beban titik kumpul gabungan dalam arah global
float DJ[mak]; // Perpindahan titik kumpul dalam arah global
float AMD[mak]; // Gaya ujung batang akibat perpindahan titik
float AM[mak][mak]; // Gaya ujung batang akhir
float AR[mak]; // Reaksi tumpuan dalam arah sumbu global
float W[mak]; // Beban Merata
float W_Balok[mak]; // Beban Berat Sendiri Balok
float P_Kolom[mak]; // Gaya Terpusat Kolom

float MTUM_KI[mak]; // Momen pada tumpuan kiri
float MTUM_KA[mak]; // Momen pada tumpuan kanan
float MLAP[mak]; // Momen lapangan
float GESER_KI[mak]; // Gaya geser sebelah kiri
float GESER_KA[mak]; // Gaya geser sebelah kanan

float MKX[mak]; // Momen kolom arah x
float MKY[mak]; // Momen kolom arah y
float PK[mak]; // Gaya aksial kolom
float GK[mak]; // Gaya geser kolom

/*****/
/* VARIABEL BETON BERTULANG */
/*****/

//////////
// Data Masukkan Umum //
//////////
float FC; // Kuat desak karakteristik Beton (MPa)
float FY; // Kuat tarik baja tulangan utama (MPa)
float FYS; // Kuat tarik baja tulangan sengkang (MPa)
float BTl; // Faktor reduksi tinggi blok tegangan ekuivalen beton

//////////
// Variabel Desain Sharing //
//////////
int nsisi_B; // Jumlah data diskrit lebar balok
int nsisi_H; // Jumlah data diskrit tinggi balok
int nsisi_K; // Jumlah data diskrit sisi kolom

int nDIA; // Jumlah data diskrit diameter
int nNL; // Jumlah data diskrit jumlah diameter
int nDIAS; // Jumlah data diskrit diameter sengkang
int nJS; // Jumlah data diskrit jarak antar sengkang
```

LAMPIRAN 5

```
float DIAS; // Diameter sengkang (mm)
float Jarak_S; // Jarak antar sengkang (mm)
float Sref; // Jarak antar sengkang yang dibutuhkan (mm)
float jarak_antar_tulangan; // Jarak antar tulangan (mm)

float sisi_d_B[mak]; // Array berisi data diskrit lebar balok (mm)
float sisi_d_H[mak]; // Array berisi data diskrit tinggi balok (mm)
float sisi_d_K[mak]; // Array berisi data diskrit sisi kolom (mm)

float DIA_d[mak]; // Array berisi data diskrit diameter (mm)
float NL_d[mak]; // Array berisi data diskrit jumlah tulangan
float DIAS_d[mak]; // Array berisi data diskrit diameter sengkang (mm)
float JS_d[mak]; // Array berisi data diskrit jarak antar sengkang (mm)

//////////
// Variabel Desain Balok //
//////////
float B; // Lebar balok (mm)
float H; // Tinggi balok (mm)
float DIA1; // Diameter tulangan tarik (mm)
float DIA2; // Diameter tulangan desak (mm)
float NL1; // Jumlah tulangan tarik (mm)
float NL2; // Jumlah tulangan desak (mm)

//////////
// Variabel Desain Kolom //
//////////
float sisi; // Sisi penampang kolom biaksial (mm)
float DIA; // Diameter tulangan (mm)
float N_DIA; // Jumlah tulangan

//////////
// Sharing Variabel Gaya //
//////////
float PU; // Gaya aksial ultimit (N)
float MU; // Momen ultimit (Nm)
float VU; // Gaya geser ultimit (N)
float VC; // Gaya geser yang disumbangkan oleh beton (N)
float VS; // Gaya geser yang disumbangkan oleh tulangan geser (N)
float FMU; // Momen Ultimit yang dapat ditahan(Nm)
float FPU; // Gaya Aksial Ultimit yang dapat ditahan (N)

//////////
// Sharing Variabel Biaya //
//////////
float harga_beton; // Harga beton (Rp./m^3)
float harga_besi; // Harga besi (Rp./kg)
float volume_beton; // Volume beton yang digunakan (m^3)
float berat_besi; // Berat besi yang digunakan (kg)
float berat_sengkang; // Berat sengkang yang digunakan (kg)
float L; // Panjang batang (m)

//////////
// Sharing Variabel Elemen //
//////////
float DS; // Jarak serat tarik terluar ke tulangan tarik (mm)
float D; // Jarak serat tekan terluar ke tulangan tarik (mm)
float AV; // Luas tulangan geser (mm^2)
float AVmin; // Luas tulangan geser minimum (mm^2)

//////////
// Variabel Khusus Balok //
//////////
float selimut_balok; // Tebal selimut pada balok

float DIA1lap; // Diameter tulangan tarik pada daerah lapangan
float NL1lap; // Jumlah tulangan tarik pada daerah lapangan
float DIA2lap; // Diameter tulangan desak pada daerah lapangan
float NL2lap; // Jumlah tulangan desak pada daerah lapangan
```

LAMPIRAN 5

```
float DIAltum; // Diameter tulangan tarik pada daerah tumpuan
float NLltum; // Jumlah tulangan tarik pada daerah tumpuan
float DIA2tum; // Diameter tulangan desak pada daerah tumpuan
float NL2tum; // Jumlah tulangan desak pada daerah tumpuan

float AS; // Luas tulangan tarik
float ASl; // Luas tulangan desak
float HMIN; // Tinggi Minimum
float n; // Rasio Es/Ec
float fr; // Modulus Keruntuhan lentur dari beton (MPa)
float LGN; // Letak Garis Netral (mm)
float Icr; // Momen Inersia Penampang Retak (mm^4)
float Ig; // Momen Inersia Penampang Kotor (mm^4)
float Ie; // Momen Inersia Efektif (mm^4)
float Mcr; // Momen Retak (Nmm);

float LENDUTAN; // Lendutan pada tengah bentang (mm)
float LENDUTAN_IJIN; // Lendutan ijin (mm);
float Lambda; // Faktor pengali lendutan jangka panjang

//////////
// Variabel Khusus Kolom //
//////////
float MUX; // Momen arah x ultimit (Nm)
float MUY; // Momen arah y ultimit (Nm)
float selimut_kolom; // Tebal selimut pada kolom
float _K; // Faktor kelangsingan

/*****
/* VARIABEL OPTIMASI DENGAN METODA FLEXIBLE POLYHEDRON */
*****/
int jum_susut; // Jumlah penyusutan berturut
int fak_plus; // Faktor penambah (JSTD=JVD+fak_plus)
int fak_kali; // Faktor Pengali (JSTD=JVD*fak_kali)
int no_struktur; // Nomor Struktur
int JVD; // Jumlah variabel desain
int JSTD; // Jumlah struktur desain
int j_iterasi_mak; // Jumlah iterasi maksimum
int nv[mak]; // Batas atas variabel diskrit normal
int nvk[mak]; // Batas atas variabel diskrit kolom
int nvb[mak]; // Batas atas variabel diskrit balok
int iterasi_var; // Jumlah iterasi untuk melangkah
int var_b[mak][mak]; // Nomor variabel balok diskrit yang digunakan
int var_k[mak][mak]; // Nomor variabel kolom diskrit yang digunakan
int tanda_arah_b[mak]; // Tanda arah-arrah penelusuran bagian balok
int tanda_arah_k[mak]; // Tanda arah-arrah penelusuran bagian kolom
int var_b_jelek[mak]; // Koordinat variabel jelek bagian balok
int var_k_jelek[mak]; // Koordinat variabel jelek bagian kolom
int arah[mak]; // Koordinat arah penelusuran
int no_TS_terjauh; // Nomor arah yang paling jauh
int varnew_asli[mak]; // Variabel terjelek hasil penggandaan
int varnew[mak]; // Variabel baru bertipe integer
int var_b_cb[mak]; // Variabel desain balok baru
int var_k_cb[mak]; // Variabel desain kolom baru
int var_b_cb_best[mak]; // Variabel desain balok baru terbaik
int var_k_cb_best[mak]; // Variabel desain kolom baru terbaik
int no_balok[mak]; // Identifikasi nomor balok dengan nomor batang
int no_kolom[mak]; // Identifikasi nomor kolom dengan nomor batang
int jum_balok; // Jumlah balok pada struktur
int jum_kolom; // Jumlah kolom pada struktur
int js_balok; // Nomor balok (Pembantu)
int js_kolom; // Nomor kolom (Pembantu)
int lompat; // Identifikasi untuk melompat dari loop

float patok_fit; // Variabel Patokan Fitness
float finalti; // Faktor finalti untuk pelanggaran
float fitstr[mak]; // Fitness struktur
float fitcb; // Fitness baru
float fitcb_best; // Fitness baru yang terbaik
```


LAMPIRAN 5

```
float kendalastr[mak]; // Kendala struktur
float hargastr[mak];   // Harga struktur (Rp.)
float kendala;         // Kendala
float kendalasa;       // Kendala pada saat struktur awal
float harga;           // Harga (Rp.)
float XM_b[mak];       // Koordinat titik Midle bagian balok
float XM_k[mak];       // Koordinat titik Midle bagian kolom
float XS_b[mak];       // Arah-arrah penelusuran bagian balok
float XS_k[mak];       // Arah-arrah penelusuran bagian kolom
float TM[mak];         // Koordinat titik Midle
float TS[mak];         // Koordinat titik Search
float varplus[mak];    // Variabel penambahan
float varnew_f[mak];   // Variabel baru bertipe float

////////////////////////////////////
// Variabel Kendala Kolom Biaksial //
////////////////////////////////////
float kendala_gaya;    // Kendala gaya pada material
float kendala_po;     // Gaya tekan maksimum akibat eksentrisitas
float kendala_pn;     // Kendala gaya tekan
float kendala_mn;     // Kendala momen

float kendala_r;       // Kendala rasio penulangan
float kendala_r_min;   // Rasio tulangan minimum
float kendala_r_mak;   // Rasio tulangan maksimum

float kendala_senggang; // Kendala sengkang
float kendala_tul;      // Kendala jumlah dan jarak tulangan
float kendala_kelangsingan; // Kendala kelangsingan

////////////////////////////////////
// Variabel Kendala Balok //
////////////////////////////////////
float kendala_rho;     // Besarnya kendala akibat rasio tulangan
float kendala_rho_b;   // Besarnya kendala akibat rasio tulangan balance
float kendala_rho_m;   // Besarnya kendala akibat rasio tulangan minimum
float kendala_sb;      // Besarnya kendala akibat kebutuhan sengkang
float kendala_M;       // Besarnya kendala akibat momen
float kendala_lendutan; // Besarnya kendala akibat tinggi minimum
```

LAMPIRAN 6

```

/*****
/* FILE : INOUT.HPP
/* KUMPULAN SUBPROGRAM MASUKAN DAN KELUARAN
*****/

/*****
/* SUBPROGRAM UNTUK MEMASUKAN DATA AWAL */
*****/
void input_data()
{
    cout << " Nama file input (tanpa ekstensi) = ";
    cin >> fi;
    strcpy(finput,fi);
    strcat(finput, ".inp"); // input umum
    strcpy(fsisi,fi);
    strcat(fsisi, ".isd"); // input sisi diskrit
    strcpy(fdial,fi);
    strcat(fdial, ".idl"); // input diameter tulangan lentur
    strcpy(fjtl,fi);
    strcat(fjtl, ".ijl"); // input jumlah tulangan lentur
    strcpy(fdias,fi);
    strcat(fdias, ".ids"); // input diameter tulangan sengkang
    strcpy(fjts,fi);
    strcat(fjts, ".ijs"); // input jarak tulangan sengkang
    int pilih_input;
    do
    {
        clrscr();
        cout << " 1. Input data umum\n";
        cout << " 2. Input data diskrit elemen\n";
        cout << " 3. Keluar\n";
        cout << " Pilihan (1-3) = ";
        cin >> pilih_input;
        if(pilih_input==1)
        {
            input_data_umum();
        }
        if(pilih_input==2)
        {
            input_data_diskrit();
        }
    }while(pilih_input!=3);
}

/*****
/* SUBPROGRAM UNTUK MEMASUKAN DATA UMUM */
*****/
void input_data_umum()
{
    int iinp,jinp;
    char uji;

    ///////////////////////////////////
    // Memasukkan Data Umum //
    ///////////////////////////////////
    cout << " Input data awal\n";
    cout << " Nama file input (Tanpa Ekstensi) = " << finput << endl;
    do
    {
        cout << " Nama Struktur (Tanpa Spasi) = ";
        cin >> ISN;
        cout << " Jumlah batang = ";
        cin >> M;
        cout << " Jumlah titik kumpul = ";
        cin >> NJ;
        cout << " Jumlah titik kumpul yang dikekang = ";
        cin >> NRJ;
        cout << " Jumlah pengekang tumpuan = ";
        cin >> NR;
        cout << " Kuat desak Karakteristik Beton (MPa) = ";
        cin >> FC;
    }
}
```

LAMPIRAN 6

```
cout << " Kuat tarik baja tulangan lentur (MPa) = ";
cin >> FY;
cout << " Kuat tarik baja tulangan sengkang (MPa) = ";
cin >> FYS;

////////////////////////////////////
// Hitung Parameter Lainnya Secara Otomatis //
////////////////////////////////////
E=(4700.*(sqrt(FC))*1E6; // (N/mm^2) -> (N/m^2)
G=(E/2.)*(1.+0.15); // (N/m^2)
ND=6.*NJ;
N=ND-NR;

output_parameter_struktural();
cout << "\n Ingin mengulang (y/t) ";
cin >> uji;
if (uji!= 'y' )
{break;}
} while(uji== 'y' );

////////////////////////////////////
// Memasukkan Data Koordinat Titik Kumpul //
////////////////////////////////////
cout << " Inputkan koordinat titik kumpul (m),"
<< " pilih J = 0 untuk keluar \n";
do
{
    cout << " Titik kumpul J = ";
    cin >> jinp;
    if ( jinp == 0 )
    { break ; }
    cout << " Koordinat\n";
    cout << " X Y Z\n";
    cin >> X[jinp] >> Y[jinp] >> Z[jinp];
    output_koordinat_titik_kumpul();
} while(jinp != 0) ;

////////////////////////////////////
// Mengosongkan Pengekang //
////////////////////////////////////
for (int jinp=1;jinp<=ND;jinp++)
{
    JRL[jinp]=0;
}

////////////////////////////////////
// Memasukkan Data Pengekang Joint //
////////////////////////////////////
cout << endl << " Inputkan Pengekang titik kumpul,"
<< " masukan 0 untuk keluar \n";
do
{
    cout << " Kekangan ke = ";
    cin >> iinp;
    if (iinp==0)
    {
        break ;
    }
    cout << " Titik kumpul yang dikekang = ";
    cin >> T_K[iinp];
    cout << " R (1 = dikekang, 0 = tak dikekang)= ";
    cin >> JRL[6*T_K[iinp]-5]
>> JRL[6*T_K[iinp]-4]
>> JRL[6*T_K[iinp]-3]
>> JRL[6*T_K[iinp]-2]
>> JRL[6*T_K[iinp]-1]
>> JRL[6*T_K[iinp]];
    output_pengekang_titik_kumpul();
}while (iinp != 0);
```

LAMPIRAN 6

```
////////////////////////////////////
// Memasukkan Informasi Batang //
////////////////////////////////////
cout << endl << "   Inputkan informasi batang,"
      << "   pilih I = 0 untuk keluar \n";

do
{
    cout << "   Indeks batang = ";
    cin  >> iinp;
    if (iinp == 0)
    { break ; }
    cout << "   titik j = ";
    cin  >> JJ[iinp];
    cout << "   titik k = ";
    cin  >> JK[iinp];
    cout << "   Notasi untuk menunjukan nol atau tidaknya sudut a = ";
    cin  >> IA[iinp];
    if(IA[iinp] != 0)
    {
        cout << "   Sudut alfa tidak 0, masukkan koordinat titik p \n";
        cout << "   Untuk batang " << iinp << endl;
        cout << "   XP   YP   ZP\n";
        cin  >> XP[iinp] >> YP[iinp] >> ZP[iinp];
        cout << endl;
    }
    output_informasi_batang();
} while ( iinp != 0 );

////////////////////////////////////
// Penulisan Ke File //
////////////////////////////////////
ofstream tulis(finput);
tulis << ISN << endl;
tulis << M << endl;
tulis << NJ << endl;
tulis << NRJ << endl;
tulis << NR << endl;
tulis << E << endl;
tulis << G << endl;
tulis << FC << endl;
tulis << FY << endl;
tulis << FYS << endl;
tulis << ND << endl;
tulis << N << endl;
tulis << "[Koordinat]" << endl;
for(int ktl=1;ktl<=NJ;ktl++)
{
    tulis << ktl << endl;
    tulis << X[ktl] << endl;
    tulis << Y[ktl] << endl;
    tulis << Z[ktl] << endl;
}
tulis << "[Pengekang]" << endl;
for (int iinp=1;iinp<=NRJ;iinp++)
{
    tulis << T_K[iinp] << endl;
    tulis << JRL[6*T_K[iinp]-5] << endl;
    tulis << JRL[6*T_K[iinp]-4] << endl;
    tulis << JRL[6*T_K[iinp]-3] << endl;
    tulis << JRL[6*T_K[iinp]-2] << endl;
    tulis << JRL[6*T_K[iinp]-1] << endl;
    tulis << JRL[6*T_K[iinp]] << endl;
}
tulis << "[InformasiBatang]" << endl;
for(int iinp=1;iinp<=M;iinp++)
{
    tulis << iinp << endl;
    tulis << JJ[iinp] << endl;
    tulis << JK[iinp] << endl;
    tulis << IA[iinp] << endl;
    if(IA[iinp] != 0)
```

LAMPIRAN 6

```
{
    tulis << XP[iinp] << endl;
    tulis << YP[iinp] << endl;
    tulis << ZP[iinp] << endl;
}
}
tulis.close();
}

/*****
/* SUBPROGRAM UNTUK MEMASUKAN DATA DISKRIT VARIABEL DESAIN */
*****/
void input_data_diskrit()
{
    char ulang;
    int bsisi,ksisi;
    int bDIA,kDIA;
    int bNL,kNL;
    int bDIAS,kDIAS;
    int bJS,kJS;
    clrscr();

    //////////////////////////////////////
    // Menggenerasi Data Diskrit Dan Ditulis Ke File //
    //////////////////////////////////////
    do
    {
        //////////////////////////////////////
        // Menulis Ke File *.isd (Input Sisi Diskrit) //
        //////////////////////////////////////
        ofstream tulis1(fsisi);

        //////////////////////////////////////
        // Data Diskrit Lebar Balok (B) //
        //////////////////////////////////////
        cout << " Jumlah data sisi untuk lebar balok (B) = ";
        cin >> nsisi_B;
        cout << " Batas sisi terbawah (mm) = ";
        cin >> bsisi;
        cout << " Kenaikan sisi (mm) = ";
        cin >> ksis;
        sisi_d_B[0]=bsisi;
        tulis1 << "[LebarBalok]" << endl;
        tulis1 << nsisi_B << endl;
        tulis1 << sisi_d_B[0] << endl;
        for(int itl1=1;itl1<nsisi_B;itl1++)
        {
            sisi_d_B[itl1]=sisi_d_B[itl1-1]+ksisi;
            tulis1 << sisi_d_B[itl1] << endl;
        }
        cout << endl;

        //////////////////////////////////////
        // Data Diskrit Tinggi Balok (H) //
        //////////////////////////////////////
        cout << " Jumlah data sisi untuk tinggi balok (H) = ";
        cin >> nsisi_H;
        cout << " Batas sisi terbawah (mm) = ";
        cin >> bsisi;
        cout << " Kenaikan sisi (mm) = ";
        cin >> ksis;
        sisi_d_H[0]=bsisi;
        tulis1 << "[TinggiBalok]" << endl;
        tulis1 << nsisi_H << endl;
        tulis1 << sisi_d_H[0] << endl;
        for(int itl1=1;itl1<nsisi_H;itl1++)
        {
            sisi_d_H[itl1]=sisi_d_H[itl1-1]+ksisi;
            tulis1 << sisi_d_H[itl1] << endl;
        }
        cout << endl;
    }
}
```

LAMPIRAN 6

```
////////////////////////////////////
// Data Diskrit Sisi Kolom //
////////////////////////////////////
cout << " Jumlah data sisi untuk kolom = ";
cin >> nsisi_K;
cout << " Batas sisi terbawah (mm) = ";
cin >> bsisi;
cout << " Kenaikan sisi (mm) = ";
cin >> ksisi;
sisi_d_K[0]=bsisi;
tulis1 << "[SisiKolom]" << endl;
tulis1 << nsisi_K << endl;
tulis1 << sisi_d_K[0] << endl;
for(int itl1=1;itl1<nsisi_K;itl1++)
{
    sisi_d_K[itl1]=sisi_d_K[itl1-1]+ksisi;
    tulis1 << sisi_d_K[itl1] << endl;
}
cout << endl;

tulis1.close();

////////////////////////////////////
// Menulis Ke File *.idl (Input Diameter Lentur) //
////////////////////////////////////
ofstream tulis2(fdial);
cout << " Jumlah data diameter tulangan utama = ";
cin >> nDIA;
cout << " Batas diameter terbawah (mm) = ";
cin >> bDIA;
cout << " Kenaikan diameter (mm) = ";
cin >> kDIA;
DIA_d[0]=bDIA;
tulis2 << "[DiameterTulanganUtama]" << endl;
tulis2 << nDIA << endl;
tulis2 << DIA_d[0] << endl;
for(int itl2=1;itl2<nDIA;itl2++)
{
    DIA_d[itl2]=DIA_d[itl2-1]+kDIA;
    tulis2 << DIA_d[itl2] << endl;
}
cout << endl;
tulis2.close();

////////////////////////////////////
// Menulis Ke File *.ijl (Input Jumlah tulangan Lentur) //
////////////////////////////////////
ofstream tulis3(fjtl);
cout << " Jumlah data jumlah tulangan = ";
cin >> nNL;
cout << " Batas jumlah tulangan terbawah = ";
cin >> bNL;
cout << " Kenaikan jumlah tulangan = ";
cin >> kNL;
NL_d[0]=bNL;
tulis3 << "[JumlahTulanganUtama]" << endl;
tulis3 << nNL << endl;
tulis3 << NL_d[0] << endl;
for(int itl3=1;itl3<nNL;itl3++)
{
    NL_d[itl3]=NL_d[itl3-1]+kNL;
    tulis3 << NL_d[itl3] << endl;
}
cout << endl;
tulis3.close();
```

LAMPIRAN 6

```
////////////////////////////////////////
// Menulis Ke File *.ids (Input Jarak Sengkang) //
////////////////////////////////////////
ofstream tulis4(fdias);
cout << " Jumlah data diameter tulangan sengkang = ";
cin >> nDIAS;
cout << " Batas diameter sengkang terbawah (mm) = ";
cin >> bDIAS;
cout << " Kenaikan diameter sengkang (mm) = ";
cin >> kDIAS;
DIAS_d[0]=bDIAS;
tulis4 << "[DiameterTulanganSengkang]" << endl;
tulis4 << nDIAS << endl;
tulis4 << DIAS_d[0] << endl;
for(int itl4=1;itl4<nDIAS;itl4++)
{
    DIAS_d[itl4]=DIAS_d[itl4-1]+kDIAS;
    tulis4 << DIAS_d[itl4] << endl;
}
cout << endl;
tulis4.close();

////////////////////////////////////////
// Menulis Ke File *.ijs (Input Jarak Sengkang) //
////////////////////////////////////////
ofstream tulis5(fjts);
cout << " Jumlah data jarak sengkang = ";
cin >> nJS;
cout << " Batas jarak sengkang terbawah (mm) = ";
cin >> bJS;
cout << " Kenaikan jarak sengkang (mm) = ";
cin >> kJS;
JS_d[0]=bJS;
tulis5 << "[JarakAntarSengkang]" << endl;
tulis5 << nJS << endl;
tulis5 << JS_d[0] << endl;
for(int itl5=1;itl5<nJS;itl5++)
{
    JS_d[itl5]=JS_d[itl5-1]+kJS;
    tulis5 << JS_d[itl5] << endl;
}
cout << endl;
tulis5.close();

    cout << " Ulang (y/t)";
    cin >> ulang;
}while(ulang == 'y');
}

/*****
/* SUBPROGRAM UNTUK MEMBACA DATA DARI FILE MASUKKAN */
*****/
void baca_data()
{
    ifstream baca(fininput);
    baca >> ISN;
    baca >> M;
    baca >> NJ;
    baca >> NRJ;
    baca >> NR;
    baca >> E;
    baca >> G;
    baca >> FC;
    baca >> FY;
    baca >> FYS;
    baca >> ND;
    baca >> N;
    baca >> SubName;
    for(int kinp=1;kinp<=NJ;kinp++)
    {
        baca >> kinp;
```

LAMPIRAN 6

```
        baca >> X[kinp];
        baca >> Y[kinp];
        baca >> Z[kinp];
    }
    baca >> SubName;
    for (int iinp=1;iinp<=NRJ;iinp++)
    {
        baca >> T_K[iinp];
        baca >> JRL[6*T_K[iinp]-5];
        baca >> JRL[6*T_K[iinp]-4];
        baca >> JRL[6*T_K[iinp]-3];
        baca >> JRL[6*T_K[iinp]-2];
        baca >> JRL[6*T_K[iinp]-1];
        baca >> JRL[6*T_K[iinp]];
    }
    baca >> SubName;
    for(int iinp=1;iinp<=M;iinp++)
    {
        baca >> iinp;
        baca >> JJ[iinp];
        baca >> JK[iinp];
        baca >> IA[iinp];
        if(IA[iinp] != 0)
        {
            baca >> XP[iinp];
            baca >> YP[iinp];
            baca >> ZP[iinp];
        }
    }
    baca.close();

    //////////////////////////////////////
    // Membaca Data Diskrit Sisi Dari File *.isd //
    //////////////////////////////////////
    ifstream bacal(fsisi);

    //////////////////////////////////////
    // Membaca Data Diskrit Lebar Balok (B) //
    //////////////////////////////////////
    bacal >> SubName;
    bacal >> nsisi_B;
    for(int iinp=0;iinp<nsisi_B;iinp++)
    {
        bacal >> sisi_d_B[iinp];
    }

    //////////////////////////////////////
    // Membaca Data Diskrit Tinggi Balok (H) //
    //////////////////////////////////////
    bacal >> SubName;
    bacal >> nsisi_H;
    for(int iinp=0;iinp<nsisi_H;iinp++)
    {
        bacal >> sisi_d_H[iinp];
    }

    //////////////////////////////////////
    // Membaca Data Diskrit Sisi Kolom //
    //////////////////////////////////////
    bacal >> SubName;
    bacal >> nsisi_K;
    for(int iinp=0;iinp<nsisi_K;iinp++)
    {
        bacal >> sisi_d_K[iinp];
    }

    bacal.close();

    ifstream baca2(fdial);
    baca2 >> SubName;
    baca2 >> nDIA;
```


LAMPIRAN 6

```
for(int iinp=0;iinp<nDIA;iinp++)
{
    baca2 >> DIA_d[iinp];
}
baca2.close();

ifstream baca3(fjtl);
baca3 >> SubName;
baca3 >> nNL;
for(int iinp=0;iinp<nNL;iinp++)
{
    baca3 >> NL_d[iinp];
}
baca3.close();

ifstream baca4(fdias);
baca4 >> SubName;
baca4 >> nDIAS;
for(int iinp=0;iinp<nDIAS;iinp++)
{
    baca4 >> DIAS_d[iinp];
}
baca4.close();

ifstream baca5(fjts);
baca5 >> SubName;
baca5 >> nJS;
for(int iinp=0;iinp<nJS;iinp++)
{
    baca5 >> JS_d[iinp];
}
baca5.close();
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN PARAMETER STRUKTURAL KE LAYAR */
*****/
void output_parameter_struktural()
{
    cout << " Struktur Portal Ruang " << ISN << "\n\n";
    cout << " Parameter Struktur\n";
    cout << " Jumlah batang : " << M << endl;
    cout << " DOF : " << N << endl;
    cout << " Jumlah joint : " << NJ << endl;
    cout << " Jumlah pengekang tumpuan : " << NR << endl;
    cout << " Jumlah titik kumpul yang dikekang : " << NRJ << endl;
    cout << " Modulus Elastisitas aksial : " << E << " N/m^2\n";
    cout << " Modulus Geser : " << G << " N/m^2\n\n";
    cout << " Properti Elemen Material\n";
    cout << " Kuat desak beton karakteristik : " << FC << " MPa\n";
    cout << " Kuat tarik baja tulangan : " << FY << " MPa\n";
    cout << " Kuat tarik tulangan sengkang : " << FYS << " MPa\n";
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN KOORDINAT STRUKTUR KE LAYAR */
*****/
void output_koordinat_titik_kumpul()
{
    cout << "\n Koordinat Titik Kumpul (m)\n";
    cout << " Titik      X          Y          Z \n";
    for(int kout=1;kout<=NJ;kout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(10) << kout
            << setw(14) << X[kout]
            << setw(14) << Y[kout]
            << setw(14) << Z[kout] << endl;
    }
}
```

LAMPIRAN 6

```

/*****/
/* SUBPROGRAM UNTUK MENAMPILKAN INFORMASI BATANG KE LAYAR */
/*****/
void output_informasi_batang()
{
    cout << "\n Informasi Batang \n";
    cout << " Batang JJ      JK      IA\n";
    for(int iout=1;iout<=M;iout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(8) << iout
                << setw(8) << JJ[iout]
                << setw(8) << JK[iout]
                << setw(3) << IA[iout] << endl;
        if(IA[iout] != 0)
        {
            cout << setiosflags(ios::left);
            cout << "  XP = " << setw(12) << XP[iout];
            cout << "  YP = " << setw(12) << YP[iout] << '\t';
            cout << "  ZP = " << setw(12) << ZP[iout] << endl;
            cout << endl;
        }
    }
}

/*****/
/* SUBPROGRAM UNTUK MENAMPILKAN PENGEKANG TITIK KUMPUL KE LAYAR */
/*****/
void output_pengekang_titik_kumpul()
{
    cout << "\n Pengekang Titik Kumpul \n";
    cout << " Titik JR1 JR2 JR3 JR4 JR5 JR6\n";
    for (int iout=1;iout<=NRJ;iout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(10) << T_K[iout]
                << setw(6) << JRL[6*T_K[iout]-5]
                << setw(6) << JRL[6*T_K[iout]-4]
                << setw(6) << JRL[6*T_K[iout]-3]
                << setw(6) << JRL[6*T_K[iout]-2]
                << setw(6) << JRL[6*T_K[iout]-1]
                << setw(6) << JRL[6*T_K[iout]] << endl;
    }
}

/*****/
/* SUBPROGRAM UNTUK MENAMPILKAN GAYA UJUNG BATANG KE LAYAR */
/*****/
void output_beban_batang()
{
    cout << "\n Gaya di Ujung Batang Terkekang Akibat Beban (Nm)\n";
    cout << " Batang AML1      AML2      AML3      "
            << "      AML4      AML5      AML6\n"
            << "      AML7      AML8      AML9      "
            << "      AML10     AML11     AML12\n";
    for(int kout=1;kout<=M;kout++)
    {
        cout << setiosflags(ios::left);
        cout << " " << setw(8) << kout
                << setw(12) << AML[1][kout]
                << setw(12) << AML[2][kout]
                << setw(12) << AML[3][kout]
                << setw(12) << AML[4][kout]
                << setw(12) << AML[5][kout]
                << setw(12) << AML[6][kout] << endl;
        cout << " "
                << setw(12) << AML[7][kout]
                << setw(12) << AML[8][kout]
                << setw(12) << AML[9][kout]
                << setw(12) << AML[10][kout]
                << setw(12) << AML[11][kout]
    }
}
```

LAMPIRAN 6

```
        << setw(12) << AML[12][kout] << endl;
    }
}

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN BEBAN TITIK KE LAYAR */
*****/
void output_beban_titik()
{
    cout << "\n  Beban Titik (N)\n"
        << "   Titik   Arah 1      Arah 2      Arah 3      "
        << "Arah 4      Arah 5      Arah 6\n";
    for(int kout=1;kout<=NJ;kout++)
    {
        cout << setiosflags(ios::left);
        cout << "   " << setw(8) << kout
            << setw(12) << AJ[(6*kout)-5]
            << setw(12) << AJ[(6*kout)-4]
            << setw(12) << AJ[(6*kout)-3]
            << setw(12) << AJ[(6*kout)-2]
            << setw(12) << AJ[(6*kout)-1]
            << setw(12) << AJ[(6*kout)] << endl;
    }
}
```

LAMPIRAN 7

```

/*****
/* FILE : STRUKTUR.HPP
/* KUMPULAN SUBPROGRAM UNTUK MENGHITUNG STRUKTUR
/* DIKEMBANGKAN DARI WEAVER & GERE
*****/

/*****
/* SUBPROGRAM MENGHITUNG GAYA DAN PERPINDAHAN PADA STRUKTUR */
*****/
void struktur()
{
    kekakuan_batang();
    banfac(N,NB,SFF);
    beban();
    bansol(N,NB,SFF,AC,DF);
    hasil();
}

/*****
/* SUBPROGRAM MENGHITUNG INERSIA */
*****/
void inersia()
{
    js_balok=0;
    js_kolom=0;
    for (int iin=1;iin<=M;iin++)
    {
        //////////////////////////////////
        // Memeriksa Kemiringan Batang //
        //////////////////////////////////
        periksa_batang(iin);

        //////////////////////////////////
        // Batang Non Vertikal //
        //////////////////////////////////
        if(CXZ>0.001)
        {
            // (mm) -> (m)
            b[iin]=isi(var_b[no_struktur][0+(12*js_balok)],sisi_d_B)/1000.;
            h[iin]=isi(var_b[no_struktur][1+(12*js_balok)],sisi_d_H)/1000.;
            js_balok++;
        }

        //////////////////////////////////
        // Batang Vertikal //
        //////////////////////////////////
        else
        {
            // (mm) -> (m)
            b[iin]=isi(var_k[no_struktur][0+(5*js_kolom)],sisi_d_K)/1000.;
            h[iin]=b[iin];
            js_kolom++;
        }

        //////////////////////////////////
        // Menghitung Luas (m^2) //
        //////////////////////////////////
        AX[iin]=b[iin]*h[iin];

        //////////////////////////////////
        // Menghitung Konstanta Puntir //
        //////////////////////////////////
        if(b[iin]<=h[iin])
        {
            XI[iin]=((((1./3.)-(0.21*b[iin]/h[iin]*
                ((1.-(pow(b[iin],4)/(12.*(pow(h[iin],4))))
                )))*h[iin])*(pow(b[iin],3))));
        }
        else
        {

```

LAMPIRAN 7

```

// Formula Diambil Dari Buku Roark's Formulas for Stress & Strain //
// Warren C. Young 1989 halaman 348 //
// Formula Diambil Dari Buku Roark's Formulas for Stress & Strain //
// Warren C. Young 1989 halaman 348 //
float _a=0.5*b[iin];
float _b=0.5*h[iin];
XI[iin]=_a*(pow(_b,3))*
(
(16./3.)-(3.36*_b/_a)*(1.-(pow(_b,4))/(12.*pow(_a,4)))
);
}

// Momen Inersia //
// Formula Diambil Dari Buku Roark's Formulas for Stress & Strain //
// Warren C. Young 1989 halaman 348 //
YI[iin]=h[iin]*(pow(b[iin],3))/12.;
ZI[iin]=b[iin]*(pow(h[iin],3))/12.;
}
}

/*****
/* SUBPROGRAM MERAKIT MATRIK KEKAKUAN STRUKTUR */
*****/
void kekakuan_batang()
{
// Inisialisasi Awal //
NB=0.;
IR=0;
IC=0;
for (int ir=1;ir<=M;ir++)
{
R11[ir]=0.;R12[ir]=0.;R13[ir]=0.;
R21[ir]=0.;R22[ir]=0.;R23[ir]=0.;
R31[ir]=0.;R32[ir]=0.;R33[ir]=0.;
}

// Membentuk Matrik Rotasi //
for (int ir=1;ir<=M;ir++)
{
if ((6*(abs(JK[ir]-JJ[ir])+1))>NB)
{
NB=(6*(abs(JK[ir]-JJ[ir])+1));
}
periksa_batang(ir);
if(IA[ir]!=0)
{
XPS=XP[ir]-(X[JJ[ir]]);
YPS=YP[ir]-(Y[JJ[ir]]);
ZPS=ZP[ir]-(Z[JJ[ir]]);
}

// Membentuk Matrik Rotasi Untuk Batang Vertikal //
if(CXZ<=0.001)
{
R11[ir]=0.; R12[ir]=Cy; R13[ir]=0.;
R21[ir]=(-Cy); R22[ir]=0.; R23[ir]=0.;
R31[ir]=0.; R32[ir]=0.; R33[ir]=1.;
if(IA[ir]==0)
{
continue;
}
COSA=(-XPS*Cy)/(sqrt(XPS*XPS+ZPS*ZPS));
SINA=(-ZPS)/(sqrt(XPS*XPS+ZPS*ZPS));
R21[ir]=(-Cy*COSA);
R23[ir]=SINA;
}
}
}
```

LAMPIRAN 7

[illegible]

LAMPIRAN 7

[illegible]

LAMPIRAN 7

```
////////////////////////////////////
// Mengisi Vektor Perpindahan Semua Titik Kumpul //
// Dalam Arah Sumbu Global //
////////////////////////////////////
ihs=N+1;
for(int jhs=1;jhs<=ND;jhs++)
{
    if(JRL[(ND-jhs+1)]==0)
    {
        ihs=(ihs-1);
        DJ[(ND-jhs+1)]=DF[ihs];
    }
    else
    {
        DJ[(ND-jhs+1)]=0.;
    }
}

////////////////////////////////////
// Gaya Pada Struktur //
////////////////////////////////////
for(int ih=1;ih<=M;ih++)
{
    periksa_batang(ih);

    //////////////////////////////////////
    // Mengisi Matrik Kekakuan Batang Pada Arah Sumbu Lokal //
    //////////////////////////////////////
    isi_matrik_kekakuan(ih);

    //////////////////////////////////////
    // Mengisi Vektor Indeks Perpindahan Batang //
    //////////////////////////////////////
    indeks_batang(ih);

    //////////////////////////////////////
    // Mengisi Vektor Gaya Ujung Batang Akhir //
    //////////////////////////////////////
    for(int jh=1;jh<=MD;jh++)
    {
        //////////////////////////////////////
        // Inisialisasi Gaya Ujung Batang Akibat Perpindahan = 0 //
        //////////////////////////////////////
        AMD[jh]=0.;

        //////////////////////////////////////
        // Mengisi Vektor Gaya Di Ujung Batang //
        // Akibat Perpindahan Titik Kumpul //
        //////////////////////////////////////
        for(int kh=1;kh<=MD;kh++)
        {
            AMD[jh]=AMD[jh]+SMRT[jh][kh]*DJ[IM[kh]];
        }
        AM[ih][jh]=AML[jh][ih]+AMD[jh];
    }

    periksa_batang(ih);

    //////////////////////////////////////
    // Menyimpan Gaya Untuk Perhitungan Balok //
    //////////////////////////////////////
    if(CXZ>0.001)
    {
        //////////////////////////////////////
        // Momen di tengah bentang (Nm) //
        //////////////////////////////////////
        MLAP[ih]=(-AM[ih][6])
            +(0.125*W[ih]*pow(EL[ih],2));

        //////////////////////////////////////
        // Momen Pada Tumpuan (Nm) //
    }
}
```


LAMPIRAN 7

```
////////////////////////////////////
MTUM_KI[ih]=(-AM[ih][6]);
MTUM_KA[ih]=AM[ih][12];

////////////////////////////////////
// Gaya Geser (N) //
////////////////////////////////////
GESER_KI[ih]=AM[ih][2];
GESER_KA[ih]=(-AM[ih][8]);
}

////////////////////////////////////
// Menyimpan Gaya Untuk Perhitungan Kolom Biaksial //
////////////////////////////////////
else
{
    //////////////////////////////////////
    // Gaya Aksial Dan Momen (N) //
    //////////////////////////////////////
    PK[ih]=fabs(AM[ih][7]);
    MKX[ih]=fabs(AM[ih][10]);
    MKY[ih]=fabs(AM[ih][12]);

    //////////////////////////////////////
    // Gaya Geser (N) //
    //////////////////////////////////////
    if(fabs(AM[ih][8])>fabs(AM[ih][9]))
    {
        GK[ih]=fabs(AM[ih][8]);
    }
    else
    {
        GK[ih]=fabs(AM[ih][9]);
    }
}

////////////////////////////////////
// Inisialisasi Vektor Reaksi Tumpuan //
////////////////////////////////////
for(int jh=1; jh<=4; jh++)
{
    if(JRL[IM[(3*jh)-2]]==1)
    {
        AR[IM[(3*jh)-2]]=0.;
    }
    if(JRL[IM[(3*jh)-1]]==1)
    {
        AR[IM[(3*jh)-1]]=0.;
    }
    if(JRL[IM[(3*jh)]]==1)
    {
        AR[IM[(3*jh)]]=0.;
    }
}

////////////////////////////////////
// Reaksi Tumpuan //
////////////////////////////////////
for(int jh=1; jh<=4; jh++)
{
    int J1=((3*jh)-2),J2=((3*jh)-1),J3=(3*jh);
    int I1=IM[J1],I2=IM[J2],I3=IM[J3];
    if(JRL[I1]==1)
    {
        AR[I1]=AR[I1]
            +R11[ih]*AMD[J1]
            +R21[ih]*AMD[J2]
            +R31[ih]*AMD[J3];
    }
    if(JRL[I2]==1)
    {

```

LAMPIRAN 7

```
        AR[I2]=AR[I2]
            +R12[ih]*AMD[J1]
            +R22[ih]*AMD[J2]
            +R32[ih]*AMD[J3];
    }
    if(JRL[I3]==1)
    {
        AR[I3]=AR[I3]
            +R13[ih]*AMD[J1]
            +R23[ih]*AMD[J2]
            +R33[ih]*AMD[J3];
    }
}

////////////////////////////////////
// Menghitung Reaksi Tumpuan Untuk Titik Terkekang //
////////////////////////////////////
for(int ih=1;ih<=ND;ih++)
{
    if(JRL[ih]==0)
    {
        continue;
    }
    AR[ih]=AR[ih]-AJ[ih]-AE[ih];
}

/*****
/* SUBPROGRAM UNTUK MEMERIKSA KEMIRINGAN BATANG */
*****/
void periksa_batang(int perb)
{
    EL[perb] = sqrt(
        (pow(((X[JK[perb]])-(X[JJ[perb]])),2))
        +(pow(((Y[JK[perb]])-(Y[JJ[perb]])),2))
        +(pow(((Z[JK[perb]])-(Z[JJ[perb]])),2)));
    CX=((X[JK[perb]])-(X[JJ[perb]]))/EL[perb];
    Cy=((Y[JK[perb]])-(Y[JJ[perb]]))/EL[perb];
    CZ=((Z[JK[perb]])-(Z[JJ[perb]]))/EL[perb];
    CXZ = fabs(sqrt(CX*CX + CZ*CZ));
}

/*****
/* SUBPROGRAM UNTUK MENGHITUNG VEKTOR INDEKS PERPINDAHAN BATANG */
*****/
void indeks_batang(int ib)
{
    IM[1] = ((6.*JJ[ib])-5.);    IM[2] = ((6.*JJ[ib])-4.);
    IM[3] = ((6.*JJ[ib])-3.);    IM[4] = ((6.*JJ[ib])-2.);
    IM[5] = ((6.*JJ[ib])-1.);    IM[6] = (6.*JJ[ib]);
    IM[7] = ((6.*JK[ib])-5.);    IM[8] = ((6.*JK[ib])-4.);
    IM[9] = ((6.*JK[ib])-3.);    IM[10] = ((6.*JK[ib])-2.);
    IM[11] = ((6.*JK[ib])-1.);    IM[12] = (6.*JK[ib]);
}

/*****
/* SUBPROGRAM UNTUK MENGISI MATRIK KEKAKUAN BATANG PADA SUMBU LOKAL */
*****/
void isi_matrik_kekakuan(int imk)
{
    //////////////////////////////////////
    // Mengosongkan Matrik Kekakuan Batang Lokal //
    //////////////////////////////////////
    for(int is=1;is<=12;is++)
    {
        for(int js=1;js<=12;js++)
        {
            SM[is][js]=0.;
        }
    }
}
```

```

////////////////////////////////////
// Menghitung Delapan Konstanta //
////////////////////////////////////
float SCM1A=E*AX[imk]/EL[imk];
float SCM1B=G*XI[imk]/EL[imk];
float SCM2Y=4.0*E*YI[imk]/EL[imk];
float SCM3Y=1.5*SCM2Y/EL[imk];
float SCM4Y=2.0*SCM3Y/EL[imk];
float SCM2Z=4.0*E*ZI[imk]/EL[imk];
float SCM3Z=1.5*SCM2Z/EL[imk];
float SCM4Z=2.0*SCM3Z/EL[imk];

////////////////////////////////////
// Mengisi Segitiga Atas Matrik Kekakuan Batang //
////////////////////////////////////
SM[1][1]=SCM1A; SM[1][7]=(-SCM1A);
SM[2][2]=SCM4Z; SM[2][6]=SCM3Z;
SM[2][8]=(-SCM4Z); SM[2][12]=SCM3Z;
SM[3][3]=SCM4Y; SM[3][5]=(-SCM3Y);
SM[3][9]=(-SCM4Y); SM[3][11]=(-SCM3Y);
SM[4][4]=SCM1B; SM[4][10]=(-SCM1B);
SM[5][5]=SCM2Y; SM[5][9]=SCM3Y;
SM[5][11]=SCM2Y/2.0; SM[6][6]=SCM2Z;
SM[6][8]=(-SCM3Z); SM[6][12]=SCM2Z/2.0;
SM[7][7]=SCM1A; SM[8][8]=SCM4Z;
SM[8][12]=(-SCM3Z); SM[9][9]=SCM4Y;
SM[9][11]=SCM3Y; SM[10][10]=SCM1B;
SM[11][11]=SCM2Y; SM[12][12]=SCM2Z;

////////////////////////////////////
// Mengisi Segitiga Bawah Matrik Yang Simetri //
////////////////////////////////////
for(int is=1;is<=11;is++)
{
    for(int js=is+1;js<=12;js++)
    {
        SM[js][is]=SM[is][js];
    }
}

////////////////////////////////////
// Mengalikan Matrik SM Dengan Matrik Rotasi //
////////////////////////////////////
for(int is=1;is<=4;is++)
{
    for(int js=1;js<=12;js++)
    {
        SMRT[js][((3*is)-2)]=SM[js][((3*is)-2)]*R11[imk]
        +SM[js][((3*is)-1)]*R21[imk]
        +SM[js][((3*is))*R31[imk];
        SMRT[js][((3*is)-1)]=SM[js][((3*is)-2)]*R12[imk]
        +SM[js][((3*is)-1)]*R22[imk]
        +SM[js][((3*is))*R32[imk];
        SMRT[js][((3*is))]=SM[js][((3*is)-2)]*R13[imk]
        +SM[js][((3*is)-1)]*R23[imk]
        +SM[js][((3*is))*R33[imk];
    }
}
}

```

LAMPIRAN 8

```

/*****
/* FILE : PEMBEBANAN.HPP
/* PUSTAKA FUNGSI MENGOLAH DATA PEMBEBANAN KE FILE
*****/

/*****
/* SUBPROGRAM MEMBACA DATA BEBAN */
*****/
void load_data()
{
    int ibn=0;
    char uji;
    cout << " Nama file struktur yang akan dikenai beban"
         << " (tanpa ekstensi) = ";
    cin >> fi;
    strcpy(finput,fi);
    strcat(finput, ".inp");
    strcpy(fbeban,fi);
    strcat(fbeban, ".bbn");
    baca_data();
    baca_beban();
    cout << "Ingin mengosongkan data beban yang ada (y/t) ? ";
    cin >> uji;
    if (uji!= 'y' )
    {
        kosong_beban();
    }

    ////////////
    // Beban Pada Batang //
    ////////////
    cout << " Inputkan beban merata pada batang,"
         << " pilih batang = 0 untuk keluar\n";
    do
    {
        cout << " Batang = ";
        cin >> ibn;
        if(ibn==0)
        { break; }
        periksa_batang(ibn);
        cout << " Panjang = " << EL[ibn] << endl;
        cout << " Beban merata (N/m) = "; cin >> W[ibn];
        AML[1][ibn]=0.;
        AML[2][ibn]=W[ibn]*EL[ibn]/2.;
        AML[3][ibn]=0.;
        AML[4][ibn]=0.;
        AML[5][ibn]=0.;
        AML[6][ibn]=W[ibn]*pow(EL[ibn],2)/12.;
        AML[7][ibn]=0.;
        AML[8][ibn]=W[ibn]*EL[ibn]/2.;
        AML[9][ibn]=0.;
        AML[10][ibn]=0.;
        AML[11][ibn]=0.;
        AML[12][ibn]=(-W[ibn])*pow(EL[ibn],2)/12.;
        output_beban_batang();
    } while (ibn!=0);
    ofstream tulis(fbeban);
    tulis << "[BebanBatang]" << endl;
    tulis << M << endl;
    for(int kbn=1;kbn<=M;kbn++)
    {
        tulis << kbn << endl;
        tulis << W[kbn] << endl;
        tulis << AML[1][kbn] << endl;
        tulis << AML[2][kbn] << endl;
        tulis << AML[3][kbn] << endl;
        tulis << AML[4][kbn] << endl;
        tulis << AML[5][kbn] << endl;
        tulis << AML[6][kbn] << endl;
        tulis << AML[7][kbn] << endl;
        tulis << AML[8][kbn] << endl;
    }
}
```

LAMPIRAN 8

```
tulis << AML[9][kbn] << endl;
tulis << AML[10][kbn] << endl;
tulis << AML[11][kbn] << endl;
tulis << AML[12][kbn] << endl;
}

////////////////////
// Beban Pada Titik //
////////////////////
ibn=0;
cout << "   Inputkan beban pada titik kumpul,"
      << " pilih titik = 0 untuk keluar\n";
do
{
    cout << "   Titik = ";
    cin >> ibn;
    if(ibn==0)
    {
        break;
    }
    cout << "   Aksi pada arah 1 global = "; cin >> AJ[((6*ibn)-5)];
    cout << "   Aksi pada arah 2 global = "; cin >> AJ[((6*ibn)-4)];
    cout << "   Aksi pada arah 3 global = "; cin >> AJ[((6*ibn)-3)];
    cout << "   Aksi pada arah 4 global = "; cin >> AJ[((6*ibn)-2)];
    cout << "   Aksi pada arah 5 global = "; cin >> AJ[((6*ibn)-1)];
    cout << "   Aksi pada arah 6 global = "; cin >> AJ[(6*ibn)];
    output_beban_titik();
} while (ibn!=0);
tulis << "[BebanTitik]" << endl;
tulis << NJ << endl;
for(int kbn=1;kbn<=NJ;kbn++)
{
    tulis << kbn << endl;
    tulis << AJ[((6*kbn)-5)] << endl;
    tulis << AJ[((6*kbn)-4)] << endl;
    tulis << AJ[((6*kbn)-3)] << endl;
    tulis << AJ[((6*kbn)-2)] << endl;
    tulis << AJ[((6*kbn)-1)] << endl;
    tulis << AJ[(6*kbn)] << endl;
}
tulis.close();
}

void kosong_beban()
{
    //////////////////////
    // Mengosongkan Beban Pada Batang //
    //////////////////////
    int ibn=0;
    do
    {
        W[ibn]=0;
        AML[1][ibn]=0.;
        AML[2][ibn]=0.;
        AML[3][ibn]=0.;
        AML[4][ibn]=0.;
        AML[5][ibn]=0.;
        AML[6][ibn]=0.;
        AML[7][ibn]=0.;
        AML[8][ibn]=0.;
        AML[9][ibn]=0.;
        AML[10][ibn]=0.;
        AML[11][ibn]=0.;
        AML[12][ibn]=0.;
        ibn++;
    } while (ibn<=M);

    //////////////////////
    // Mengosongkan Beban Pada Titik Kumpul //
    //////////////////////
    ibn=0;
```

LAMPIRAN 8

```
do
{
    AJ[(6*ibn)-5]=0.;
    AJ[(6*ibn)-4]=0.;
    AJ[(6*ibn)-3]=0.;
    AJ[(6*ibn)-2]=0.;
    AJ[(6*ibn)-1]=0.;
    AJ[(6*ibn)]=0.;
    ibn++;
} while(ibn<=NJ);
}

/*****/
/* SUBPROGRAM UNTUK MEMBACA DATA PEMBEBANAN DARI FILE */
/*****/
void baca_beban()
{
    ifstream baca(fbeban);

    ///////////////////////////////////
    // Membaca Beban Batang //
    ///////////////////////////////////
    baca >> SubName;
    baca >> M;
    for(int kinp=1;kinp<=M;kinp++)
    {
        baca >> kinp;
        baca >> W[kinp];
        baca >> AML[1][kinp];
        baca >> AML[2][kinp];
        baca >> AML[3][kinp];
        baca >> AML[4][kinp];
        baca >> AML[5][kinp];
        baca >> AML[6][kinp];
        baca >> AML[7][kinp];
        baca >> AML[8][kinp];
        baca >> AML[9][kinp];
        baca >> AML[10][kinp];
        baca >> AML[11][kinp];
        baca >> AML[12][kinp];
    }

    ///////////////////////////////////
    // Membaca Beban Titik //
    ///////////////////////////////////
    baca >> SubName;
    baca >> NJ;
    for(int kinp=1;kinp<=NJ;kinp++)
    {
        baca >> kinp;
        baca >> AJ[(6*kinp)-5];
        baca >> AJ[(6*kinp)-4];
        baca >> AJ[(6*kinp)-3];
        baca >> AJ[(6*kinp)-2];
        baca >> AJ[(6*kinp)-1];
        baca >> AJ[(6*kinp)];
    }
    baca.close();
    berat_sendiri();
}

/*****/
/* SUBPROGRAM UNTUK MERAKIT VEKTOR BEBAN */
/*****/
void beban()
{
    ///////////////////////////////////
    // Mengosongkan Vektor Beban Titik Kumpul Gabungan //
    ///////////////////////////////////
    for(int ib=1;ib<=ND;ib++)
    {
```

LAMPIRAN 8

```
    AC[ID[ib]]=0.;
}

////////////////////////////////////
// Mengosongkan Vektor Beban Titik Kumpul Ekvivalen //
////////////////////////////////////
for (int ib=1;ib<=M;ib++)
{
    periksa_batang(ib);
    if(CXZ<=0.001)
    {
        continue;
    }
    indeks_batang(ib);
    for(int jb=1;jb<=4;jb++)
    {
        AE[IM[(3*jb-2)]] = 0.;
        AE[IM[(3*jb-1)]] = 0.;
        AE[IM[(3*jb)]] = 0.;
    }
}

////////////////////////////////////
// Beban Titik Kumpul Ekvivalen //
////////////////////////////////////
for (int ib=1;ib<=M;ib++)
{
    periksa_batang(ib);
    if(CXZ<=0.001)
    {
        continue;
    }
    indeks_batang(ib);
    for(int jb=1;jb<=4;jb++)
    {
        AE[IM[(3*jb-2)]] = AE[IM[(3*jb-2)]]
            -R11[ib]*AML[(3*jb-2)][ib]
            -R21[ib]*AML[(3*jb-1)][ib]
            -R31[ib]*AML[(3*jb)][ib];
        AE[IM[(3*jb-1)]] = AE[IM[(3*jb-1)]]
            -R12[ib]*AML[(3*jb-2)][ib]
            -R22[ib]*AML[(3*jb-1)][ib]
            -R32[ib]*AML[(3*jb)][ib];
        AE[IM[(3*jb)]] = AE[IM[(3*jb)]]
            -R13[ib]*AML[(3*jb-2)][ib]
            -R23[ib]*AML[(3*jb-1)][ib]
            -R33[ib]*AML[(3*jb)][ib];
    }
}

////////////////////////////////////
// Beban Titik Kumpul Gabungan //
////////////////////////////////////
for(int ib=1;ib<=ND;ib++)
{
    AC[ID[ib]] = AJ[ib] + AE[ib];
}

}

/*****
/* SUBPRROGRAM MENGHITUNG BERAT SENDIRI BALOK */
*****/
void berat_sendiri()
{
    js_balok=0;
    js_kolom=0;
    for(int ibs=1;ibs<=M;ibs++)
    {
        //////////////////////////////////////
        // Memeriksa Kemiringan Batang //
        //////////////////////////////////////
    }
}
```

LAMPIRAN 8

```
periksa_batang(ibs);

//////////
// Batang Non Vertikal //
//////////
if(CXZ>0.001)
{
    periksa_batang(ibs);
    // (mm) -> (m)
    b[ibs]=isi(var_b[no_struktur][0+(12*js_balok)],sisi_d_B);
    h[ibs]=isi(var_b[no_struktur][1+(12*js_balok)],sisi_d_H);
    W_Balok[ibs]=24000.*b[ibs]*h[ibs]*1.E-6; // (N/m)
    W[ibs]+=W_Balok[ibs]; // (N/m)
    AML[2][ibs]+=W_Balok[ibs]*EL[ibs]/2.;
    AML[6][ibs]+=W_Balok[ibs]*pow(EL[ibs],2)/12.;
    AML[8][ibs]+=W_Balok[ibs]*EL[ibs]/2.;
    AML[12][ibs]+=(-W_Balok[ibs])*pow(EL[ibs],2)/12.;
    js_balok++;
}

//////////
// Batang Vertikal //
//////////
else
{
    periksa_batang(ibs);
    b[ibs]=isi(var_k[no_struktur][0+(5*js_kolom)],sisi_d_K);
    h[ibs]=b[ibs];
    P_Kolom[ibs]=(-24000.)*b[ibs]*h[ibs]*EL[ibs]*1.E-6; // (N)

    AJ[((6*JJ[ibs])-4)]+=P_Kolom[ibs];

    js_kolom++;
}
}
```


LAMPIRAN 9

```

/*****
/* FILE : SOLVER.HPP
/* SUBPROGRAM UNTUK MENYELESAIKAN PERSAMAAN
/* DENGAN METODA CHOLESKI YANG DIMODIFIKASI
*****/

/*****
/* SUBPROGRAM UNTUK FAKTORISASI MATRIK SIMETRIS BERJALUR */
/* DENGAN PENDEKATAN CHOLESKI YANG DIMODIFIKASI
*****/
void banfac(int N,int NB,float A_SFF[][mak])
{
    int J2;
    float SUM;
    float TEMP;
    if((A_SFF[1][1])>0)
    {
        for(int jbf=2;jbf<=N;jbf++)
        {
            J2=(jbf-NB+1);
            if(J2<1)
            {
                J2=1;
            }
            if((jbf-1)!=1)
            {
                for(int ibf=2;ibf<=(jbf-1);ibf++)
                {
                    if((ibf-1)>=J2)
                    {
                        SUM=A_SFF[ibf][jbf-ibf+1];
                        for(int kbf=J2;kbf<=(ibf-1);kbf++)
                        {
                            SUM=SUM-A_SFF[kbf][ibf-kbf+1]
                                *A_SFF[kbf][jbf-kbf+1];
                        }
                        A_SFF[ibf][jbf-ibf+1]=SUM;
                    }
                    else
                    {
                        continue;
                    }
                }
            }
            SUM=A_SFF[jbf][1];
            for(int kbf=J2;kbf<=(jbf-1);kbf++)
            {
                TEMP=A_SFF[kbf][jbf-kbf+1]/A_SFF[kbf][1];
                SUM=SUM-TEMP*A_SFF[kbf][jbf-kbf+1];
                A_SFF[kbf][jbf-kbf+1]=TEMP;
            }
            if(SUM<=0)
            {
                break;
            }
            A_SFF[jbf][1]=SUM;
        }
    }
    else
    {
        clrscr();
        cout << "Tidak positif tentu !!!\n";
        cout << "Struktur tak stabil - GAGAL dibangkitkan !!\n\a";
        cout << "Periksa data struktur anda \n";
        cout << "Tekan sembarang tombol, program akan dihentikan \n";
        getch();
        exit(1);
    }
}

```

LAMPIRAN 9

```

/*****
/* SUBPROGRAM MENGOLAH MATRIK BERJALUR DARI SUBPROGRAM BANFAC() */
*****/
void bansol(int N,int NB,float U_SFF[][mak]
            ,float B_AC[mak],float X_DF[mak])
{
    float SUM;

    //////////////////////////////////////
    // Inisialisasi Perpindahan Titik Kumpul Bebas = 0 //
    //////////////////////////////////////
    for(int ins=1;ins<=N;ins++)
    {
        X_DF[ins]=0.;
    }

    for(int ibsl=1;ibsl<=N;ibsl++)
    {
        int jbns=(ibsl-NB+1);
        if (ibsl<=NB)
        {
            jbns=1;
        }
        SUM=B_AC[ibsl];
        if(jbns<=(ibsl-1))
        {
            for(int kbns=jbns;kbns<=(ibsl-1);kbns++)
            {
                SUM=(SUM-((U_SFF[kbns][ibsl-kbns+1])*(X_DF[kbns])));
            }
        }
        X_DF[ibsl] = SUM;
    }

    for(int ibsl=1;ibsl<=N;ibsl++)
    {
        X_DF[ibsl]=((X_DF[ibsl])/(U_SFF[ibsl][1]));
    }

    for(int I1=1;I1<=N;I1++)
    {
        int ibsl=(N-I1+1);
        int jbns=(ibsl+NB-1);
        if(jbns>N)
        {
            jbns=N;
        }
        SUM=X_DF[ibsl];

        if((ibsl+1)<=jbns)
        {
            for(int kbns=(ibsl+1);kbns<=jbns;kbns++)
            {
                SUM=(SUM-(U_SFF[ibsl][kbns-ibsl+1]*X_DF[kbns]));
            }
        }
        X_DF[ibsl] = SUM;
    }
}

```

LAMPIRAN 10

```

/*****
/* FILE : KOLOM.HPP
/* KELAS UNTUK MENGHITUNG HARGA DAN KENDALA PADA KOLOM
*****/
class kolom
{
private:

    /***/
    /* PENDEKLARASIAN VARIABEL PRIVATE PADA KELAS KOLOM */
    /***/

    ////////////////
    // Perhitungan Kolom //
    ////////////////
    float RHO; // Rasio penulangan
    float PN; // Gaya aksial nominal
    float MNX; // Momen nominal arah x
    float MNY; // Momen nominal arah y
    float MOX; // Momen ekuivalen arah x
    float MOY; // Momen ekuivalen arah y
    float PO; // Batas gaya aksial karena eksentrisitas minimum (N)
    float PNCoba; // Gaya aksial nominal coba
    float PNB; // Gaya aksial nominal pada kondisi balance
    float MNCoba; // Momen nominal coba
    float beta; // Faktor bentuk biaksial
    float ASTOT; // Luas besi total pada penampang (mm^2)
    float CB; // Jarak serat tekan terluar ke garis netral
    // pada keadaan balance (mm)
    float FS; // Tegangan dalam tulangan pada beban kerja
    float eks; // Eksentrisitas (mm)
    float eksCoba; // Eksentrisitas maksimum (mm)
    float eksb; // Eksentrisitas pada keadaan balance (mm)
    float epsb; // Regangan baja pada keadaan balance (mm)
    float epsy; // Regangan baja pada keadaan luluh (mm)

    float rasio_kelangsingan; // Rasio kelangsingan

    float fs1b; // Tegangan baja tulangan pada keadaan balance (MPa)
    float fs1; // Tegangan baja (MPa)
    float di; // Jarak tulangan ke garis netral (mm)
    float Fsi; // Gaya Yang Disumbangkan Oleh Tulangan (N)
    float ASdi; // Luas tulangan pada jarak di
    float fsf1;
    float dll;

    ////////////////
    // Variabel Pembantu Untuk Menyelesaikan Persamaan Kuadrat //
    ////////////////
    float asol;
    float bsol;
    float csol;
    float dsol;

    ////////////////
    // Variabel Pada Metoda False Posisi //
    ////////////////
    int iterasi_fp; // Jumlah iterasi dengan metode false posisi

    float trialKiri; // Titik coba pertama
    float trialKanan; // Titik coba kedua
    float trialBaru; // Titik coba baru
    float deki; // Nilai titik coba pertama
    float deka; // Nilai titik coba kedua
    float deba; // Nilai titik coba baru

    /***/
    /* PROTOTYPE FUNGSI PADA KELAS KOLOM */
    /***/
    void rho();
    void jarak_tulangan();

```

LAMPIRAN 10

```
void kelangsingan();
void analisa();
void sengkang_kolom();
float hitung_kolom (float Ccoba);

public:

/*****
/* CONSTRUCTOR CLASS KOLOM */
*****/
kolom()
{
    ////////////////////////////////////
    // Inisialisasi Nilai Awal //
    ////////////////////////////////////
    kendala=0.;
    harga=0.;

    ////////////////////////////////////
    // Memanggil Fungsi-Fungsi Kendala //
    ////////////////////////////////////
    rho();
    jarak_tulangan();
    kelangsingan();
    analisa();
    sengkang_kolom();

    ////////////////////////////////////
    // Menghitung Kendala Total //
    ////////////////////////////////////
    kendala=kendala_sengkang
            +kendala_r
            +kendala_tul
            +kendala_gaya
            +kendala_kelangsingan;

    ////////////////////////////////////
    // Perhitungan Berat Dan Volume //
    ////////////////////////////////////
    volume_beton=sisi*sisi*L/1.E6;
    berat_besi=ASTOT*L*bj_besi/1.E6;
    berat_sengkang=(fabs(L/(Jarak_S/1000.))-1.)
                   *4.*((sisi-(2.*selimut_kolom))/1000.)
                   *(pi/4.)*pow((DIAS/1000.),2.)
                   *bj_besi;

    ////////////////////////////////////
    // Menghitung Harga //
    ////////////////////////////////////
    harga=(volume_beton*harga_beton)
          +(berat_besi*harga_besi)
          +(berat_sengkang*harga_besi);
}

};

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA RASIO PENULANGAN */
*****/
void kolom::rho()
{
    ASTOT=((4.*N_DIA-4.)*(pi/4.)*(pow(DIA,2)));
    RHO=(ASTOT/(pow(sisi,2)));

    ////////////////////////////////////
    // Menghitung Kendala Rho Minimum //
    ////////////////////////////////////
    kendala_r_min=((0.01/RHO)-1.);
    if(kendala_r_min<0.)
    {
        kendala_r_min=0.;
    }
}
```

LAMPIRAN 10

```
////////////////////////////////////
// Menghitung Kendala Rho Maksimum //
////////////////////////////////////
kendala_r_mak=((RHO/0.08)-1.);
if(kendala_r_mak<0.)
{
    kendala_r_mak=0.;
}

////////////////////////////////////
// Menjumlahkan Kendala Rho //
////////////////////////////////////
kendala_r=kendala_r_min+kendala_r_mak;
}

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA JARAK ANTARA TULANGAN */
*****/
void kolom::jarak_tulangan()
{
    //////////////////////////////////////
    // Mengecek Jarak Minimum Antara Tulangan //
    //////////////////////////////////////
    float min1=(1.5*DIA);
    float min2=40.;
    float sisa=sisi-(2.*selimut_kolom)-N_DIA*DIA;

    jarak_antar_tulangan=sisa/(N_DIA-1.);
    if(jarak_antar_tulangan==0)
    {
        jarak_antar_tulangan=limit_nol;
    }

    float jarak_min=min1;
    if(min2<min1)
    {
        jarak_min=min2;
    }

    //////////////////////////////////////
    // Menghitung Kendala Total Akibat Kendala Jarak Antar Tulangan //
    //////////////////////////////////////
    kendala_tul=((jarak_min/jarak_antar_tulangan)-1.);
    if(kendala_tul<0.)
    {
        kendala_tul=0.;
    }
}

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA KELANGSINGAN KOLOM */
*****/
void kolom::kelangsingan()
{
    //////////////////////////////////////
    // Syarat Kelangsingan Kolom Menurut SK SNI T-15-1991-03 //
    // KL/r <= 22 //
    //////////////////////////////////////
    _K=0.5;
    rasio_kelangsingan=(_K*L*1000./((sisi*(sqrt((1./12.))))));

    //////////////////////////////////////
    // Menghitung Kendala Kelangsingan //
    //////////////////////////////////////
    kendala_kelangsingan=(rasio_kelangsingan/22.)-1.;
    if(kendala_kelangsingan<0)
    {
        kendala_kelangsingan = 0.;
    }
}
```

LAMPIRAN 10

[illegible]

LAMPIRAN 10

```
{
    beta=0.3;
}

////////////////////////////////////
// Mencari Momen Ekuivalen Biaksial //
////////////////////////////////////
if(MNX > MNY)
{
    MOX=MNX+(MNY*((1.-beta)/beta)); // Momen ekuivalen (Nmm)
    eks=(fabs(MOX/PN)); // (mm)
}
else
{
    MOY=MNY+(MNX*((1.-beta)/beta)); // Momen ekuivalen (Nmm)
    eks=(fabs(MOY/PN)); // (mm)
}

////////////////////////////////////
// Menghitung Letak Garis Netral Pada Keadaan Balance //
////////////////////////////////////
CB=600.*D/(FY+600.); // (mm)

////////////////////////////////////
// Menghitung Tegangan Pada Keadaan Balance //
////////////////////////////////////
deki=hitung_kolom(CB);
eksb=ekscoba; // (mm)
epsb=0.003*(CB-DS)/CB;
epsy=FY/2.E5;

if(epsb>epsy)
{
    fslb=FY; // (Mpa)
}
if(epsb<epsy)
{
    fslb=epsb*2.E5; // (MPa)
}

//-----//
// Proses Mencari Letak Garis Netral Dengan Metode False Posisi //
//-----//

////////////////////////////////////
// Pembagian Daerah Pencarian //
////////////////////////////////////
trialkiri=40.;
trialkanan=(sisi-40.);
trialbaru=(CB);

////////////////////////////////////
// Menghitung Nilai Awal //
////////////////////////////////////
deki=hitung_kolom(trialkiri);
deka=hitung_kolom(trialkanan);
deba=hitung_kolom(trialbaru);

////////////////////////////////////
// Periksa Batas Range //
////////////////////////////////////
if((deki*deka>0)&&(deki!=0)&&(deka!=0))
{
    do
    {
        //////////////////////////////////////
        // Bila Nilai Yang Kita Cari Tidak Ada Dalam Range //
        //////////////////////////////////////
        if(trialkiri>(limit_nol))
        {
```

LAMPIRAN 10

```
        trial kiri=((trial kiri)/2.);
        deki=hitung_kolom(trial kiri);
    }
    else
    {
        trial kanan+=10.;
        deka=hitung_kolom(trial kanan);
        if(trial kanan>(sisi/0.85))
        {
            break;
        }
    }
}while(deki*deka>0&&(deki!=0)&&(deka!=0));
}

////////////////////////////////////
// Akselerator False Posisi //
////////////////////////////////////
iterasi_fp=1;
do
{
    //////////////////////////////////
    // Periksa Konvergensi //
    // Diambil E = 1 mm //
    //////////////////////////////////
    if(deba<1.&&deba>(-1.))
    {
        if(PNcoba<=0)
        {
            PNcoba=limit_nol;
        }
        MNcoba=fabs(MNcoba);
        break;
    }

    //////////////////////////////////
    // Periksa Apakah deki Dan deba Sama Tanda //
    //////////////////////////////////
    if((deki*deba)<0.)
    {
        //////////////////////////////////
        // Geser trial kanan menuju trial baru //
        //////////////////////////////////
        trial kanan=trial baru;
        deka=hitung_kolom(trial kanan);
    }
    else
    {
        //////////////////////////////////
        // Geser trial kiri menuju trial baru //
        //////////////////////////////////
        trial kiri=trial baru;
        deki=hitung_kolom(trial kiri);
    }

    //////////////////////////////////
    // Persempit Lagi Daerah Pencarian //
    //////////////////////////////////
    trial baru=((trial kiri+trial kanan)/2.);

    deba=hitung_kolom(trial baru);

    iterasi_fp++;

    if(trial kiri==trial kanan)
    {
        break;
    }
}while(iterasi_fp<=100);
```


LAMPIRAN 10

```
// Insialisasi Jumlah Iterasi //
////////////////////////////////////
iterasi_fp=1;

/*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-**/
/* Proses Pencarian Dengan Metoda False Posisi Dimulai */
/*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-**/
do
{
    if(trialkiri==trialkanan)
    {
        break;
    }
    ////////////////////////////////////
    // Menentukan Titik Coba Baru //
    ////////////////////////////////////
    trialbaru=trialkanan-((deka)*(trialkanan-trialkiri)/((deka)-(deki)));

    deba=hitung_kolom(trialbaru);

    ////////////////////////////////////
    // Periksa Konvergensi //
    // Diambil E = 10 mm //
    ////////////////////////////////////
    if(deba<10.&&deba>(-10.))
    {
        if(PNcoba<=0)
        {
            PNcoba=limit_nol;
        }
        MNCoba=fabs(MNCoba);
        break;
    }

    ////////////////////////////////////
    // Periksa Apakah deki Dan deba Sama Tanda //
    ////////////////////////////////////
    if((deki*deba)<0.)
    {
        ////////////////////////////////////
        // Geser trialkanan menuju trialbaru //
        ////////////////////////////////////
        trialkanan=trialbaru;
        deka=hitung_kolom(trialkanan);
    }
    else
    {
        ////////////////////////////////////
        // Geser trial kiri menuju trialbaru //
        ////////////////////////////////////
        trialkiri=trialbaru;
        deki=hitung_kolom(trialkiri);
    }

    iterasi_fp++;
    if(iterasi_fp>=100)
    {
        ////////////////////////////////////
        // Dihitung Dengan Metoda Pendekatan //
        ////////////////////////////////////
        deka=hitung_kolom((trialkiri+trialkanan)/2.);
        break;
    }
}while(deki*deka!=0.);
FPU=fabs(PNcoba);          // Dalam N
FMU=fabs(MNCoba)/1000.;   // Hasil perhitungan dalam Nmm -> N
```

LAMPIRAN 10

```
////////////////////////////////////
// Menghitung Kendala Akibat Gaya Aksial //
////////////////////////////////////
kendala_pn=((PN/PNcoba)-1.);
if(kendala_pn<0)
{
    kendala_pn=0.;
}

////////////////////////////////////
// Menghitung Kendala Akibat Gaya Lentur //
////////////////////////////////////
if(MNX > MNX)
{
    kendala_mn=((MOX/MNcoba)-1.);
    if(kendala_mn<0)
    {
        kendala_mn=0.;
    }
}
else
{
    kendala_mn=((MOY/MNcoba)-1.);
    if(kendala_mn<0)
    {
        kendala_mn=0.;
    }
}
}

////////////////////////////////////
// Bila Gaya Aksial Melanggar Gaya Aksial Batas Minimum //
// Eksentrisitas Pada Kolom Berpengikat Senggang //
////////////////////////////////////
else
{
    FPU=PO;
    FMU=0.;
    kendala_pn=0.;
    kendala_mn=0.;
}

////////////////////////////////////
// Menghitung Kendala Total Akibat Gaya Yang Terjadi //
////////////////////////////////////
kendala_gaya=kendala_po+kendala_pn+kendala_mn;
}

/*****
/* FUNGSI UNTUK MENGHITUNG TEGANGAN PADA KOLOM */
*****/
float kolom::hitung_kolom(float Ccoba)
{
    //////////////////////////////////////
    // Menghitung Gaya Yang Ditahan Beton //
    //////////////////////////////////////
    PNcoba=(0.85*BT1*FC*Ccoba*sisi);
    MNcoba=(0.85*BT1*FC*Ccoba*sisi)*((sisi/2.)-(BT1*Ccoba/2.));

    //////////////////////////////////////
    // Iterasi Untuk Kekuatan Yang Didukung Tulangan //
    //////////////////////////////////////
    for(int ikl=1; ikl<=N_DIA; ikl++)
    {
        di=(DS+((N_DIA-ikl)*(sisi-(2.*DS))/(N_DIA-1.)));

        if(di==(0.5*sisi))
        {
            continue;
        }
    }
}
```

LAMPIRAN 10

```
////////////////////////////////////
// Luas Tulangan Pada Baris Luar //
////////////////////////////////////
if((ikl==1)||(ikl==N_DIA))
{
    ASdi=(N_DIA*(pi/4.)*pow(DIA,2));
}

////////////////////////////////////
// Luas Tulangan Pada Baris Dalam //
////////////////////////////////////
else
{
    ASdi=(2.*(pi/4.)*pow(DIA,2));
}

////////////////////////////////////
// Antisipasi Nilai Ccoba Dengan Limit Nol //
////////////////////////////////////
if(Ccoba==0)
{
    Ccoba=limit_nol;
}

////////////////////////////////////
// Hitung Tegangan Tahanan Tulangan //
// fsi = Regangan * Modulus Elastisitas //
// Dengan Regangan = 0.003[(c-di)/c] //
// Modulus Elastisitas = 2E5 MPa (Baja) //
////////////////////////////////////
fsi=((600.*(Ccoba-di))/Ccoba);

////////////////////////////////////
// Periksa Range Tegangan //
////////////////////////////////////
if(fsi>FY)
{
    fsi=FY;
}
else if(fsi<(-FY))
{
    fsi=(0.-FY);
}

////////////////////////////////////
// Reduksi Tegangan Baja Pada Daerah Desak //
////////////////////////////////////
if(di<(BT1*Ccoba))
{
    fsi=(fsi-(0.85*FC));
}

////////////////////////////////////
// Hitung Gaya Yang Disumbangkan Oleh Tulangan //
////////////////////////////////////
Fsi=fsi*ASdi;

////////////////////////////////////
// Hitung Gaya Aksial Dan Momen Tahanan Maksimal //
////////////////////////////////////
PNcoba=(PNcoba+Fsi);
MNcoba=(MNcoba+(Fsi*((sisi/2.)-di)));
}

////////////////////////////////////
// Antisipasi Divide By Zero Error //
////////////////////////////////////
if (PNcoba<=0)
{
    PNcoba=limit_nol;
}
```

LAMPIRAN 10

```
////////////////////////////////////
// Hitung Momen Dan Gaya Aksial Total //
////////////////////////////////////
ekscoba=(MNCoba/PNCoba);

////////////////////////////////////
// Kembalikan Nilai Fungsi //
////////////////////////////////////
return (eks-ekscoba);
}

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA GAYA GESER PADA KOLOM */
*****/
void kolom::senggang_kolom()
{
    //////////////////////////////////////
    // Menghitung Antar Jarak Senggang Maksimal //
    // Berdasarkan SK SNI T-15-1991-03 pasal 3.16.10 ayat 5 //
    //////////////////////////////////////
    float Smak[3];
    Smak[0]=sisi;
    Smak[1]=16.*DIA;
    Smak[2]=48.*DIAS;
    float SmakS=Smak[0];

    //////////////////////////////////////
    // Cari Yang Terkecil //
    //////////////////////////////////////
    for (int sk=1;sk<3;sk++)
    {
        if(SmakS>Smak[sk])
        {
            SmakS=Smak[sk];
        }
    }

    Sref=SmakS;
    //////////////////////////////////////
    // Menghitung Kendala Jarak Antar Senggang //
    //////////////////////////////////////
    kendala_senggang=((Jarak_S/SmakS)-1.);
    if(kendala_senggang<0.)
    {
        kendala_senggang=0.;
    }
}
```

LAMPIRAN 11

```

/*****
/* FILE : BALOK.HPP
/* KELAS UNTUK MENGHITUNG HARGA DAN KENDALA PADA BALOK
*****/
class balok
{
private:

    /***/
    /* PENDEKLARASIAN VARIABEL PRIVATE PADA KELAS BALOK */
    /***/

    ////////////////
    // Perhitungan Balok //
    ////////////////
    float FS;      // Tegangan pada tulangan tarik (N/mm^2)
    float FS1;     // Tegangan pada tulangan desak (N/mm^2)
    float ARM;     // Lengan pusat tekan ke garis Netral (mm)
    float ARMS;    // Lengan pusat tulangan ke garis Netral (mm)
    float EPS;     // Regangan pada tulangan tarik (mm)
    float EPS1;    // Regangan pada tulangan tekan (mm)
    float EPSY;    // Regangan ijin baja pada keadaan luluh (mm)
    float RHB;     // Rasio tulangan balance
    float RHO;     // Rasio tulangan tarik
    float RH1;     // Rasio tulangan desak
    float RMIN;    // Rasio tulangan minimum
    float SmakS;   // Jarak antar sengkang maksimum yang diijinkan (mm)

    ////////////////
    // Variabel Pembantu Persamaan Kuadrat //
    ////////////////
    float ASOL;
    float BSOL;
    float CSOL;
    float DSOL;

    /***/
    /* PROTOTYPE FUNGSI PADA KELAS BALOK */
    /***/
    void analisa();
    void sengkang_balok();
public:

    /***/
    /* CONSTRUCTOR CLASS BALOK */
    /***/
    balok()
    {
        ////////////////
        // Inisialisai Kendala Dan Harga = 0 //
        ////////////////
        kendala=0.;
        harga=0.;

        ////////////////
        // Menghitung Luas Tulangan Tarik Dan Tekan //
        ////////////////
        AS=(0.25*pi*(pow(DIA1,2))*NL1);
        AS1=(0.25*pi*(pow(DIA2,2))*NL2);

        ////////////////
        // Memanggil Fungsi Fungsi Kendala //
        ////////////////
        analisa();
        sengkang_balok();
        kendala=kendala_sb+kendala_rho+kendala_M;

        ////////////////
        // Perhitungan Berat Dan Volume //
        ////////////////
        volume_beton=B*H*0.5*L/1.E6;
    }
};

```

LAMPIRAN 11

```
berat_besi=(AS+AS1)*0.5*L*bj_besi/1.E6;
berat_sengkang=0.5*(fabs(L/(Jarak_S/1000.))-1.)*
(
    2.*
    (
        (B-(2.*selimut_balok))/1000.)+
        2.*(H-(2.*selimut_balok))/1000.)
    )
)*
(pi/4.)*pow((DIAS/1000.),2)*bj_besi;

//////////
// Menghitung Harga //
//////////
harga=volume_beton*harga_beton
      +berat_besi*harga_besi
      +berat_sengkang*harga_besi;
}
};

/*****
/* FUNGSI UNTUK MENGHITUNG KENDALA TEGANGAN MATERIAL */
*****/
void balok::analisa()
{
    ////////////
    // Menghitung Faktor Pengali Tinggi Blok Tegangan Segiempat //
    // Ekuivalen (B1) //
    // Sumber : SK SNI T-15-1991-03 Pasal 3.3.2 butir 7.(3) //
    ////////////
    if(FC<=30.)
    {
        BT1=0.85;
    }
    else
    {
        BT1=0.85-0.008*(FC-30.);
    }
    if(BT1<0.65)
    {
        BT1=0.65;
    }

    DS=selimut_balok+(0.5*DIA1);

    D=H-DS;
    FS=FY;

    EPSY=FY/200000.;

    ////////////
    //Asumsi Pertama //
    ////////////
    FS1=FS;

    ////////////
    // Hitung Besarnya Lengan Ke Garis Netral (mm) //
    ////////////
    ARM=((AS-AS1)*FY)/(0.85*FC*B);
    if(ARM==0)
    {
        ARM=limit_nol;
    }

    ////////////
    // Periksa Regangan Pada Tulangan //
    ////////////
    EPS1=0.003*(ARM-BT1*DS)/ARM;
    EPS=0.003*(BT1*D-ARM)/ARM;
```

LAMPIRAN 11

```
////////////////////////////////////
// Hitung Ulang Regangan Pada Baja Apabila Tulangan Tidak Luluh //
////////////////////////////////////
if((EPS<EPSY)|| (EPS1<EPSY))
{
    //////////////////////////////////////
    // Hitung Koefisien Persamaan Kuadrat Dan Menyelesaikannya //
    //////////////////////////////////////
    ASOL=0.85*FC*B;
    BSOL=600.*AS1-AS*FY;
    CSOL=(-600.)*BT1*AS1*DS;
    DSOL=pow(BSOL,2)-(4.*(ASOL*CSOL));
    if(DSOL<=0)
    {
        DSOL=0.;
    }
    ARMS=(-BSOL)+sqrt(DSOL);
    if(ARMS<=0&&DSOL>=0)
    {
        ARMS=(-BSOL)-sqrt(DSOL);
    }
    ARM=0.5*(ARMS/ASOL);
    if(ARM<=0)
    {
        ARM=limit_nol;
    }

    //////////////////////////////////////
    // Hitung Regangan Pada Tulangan //
    //////////////////////////////////////
    EPS1=0.003*(ARM-BT1*DS)/ARM;
    EPS=0.003*(BT1*D-ARM)/ARM;
}

if(EPS>EPSY)
{
    FS=FY;
}

FS1=EPS1*200000.;
if(FS1>FY)
{
    FS1=FY;
}
if(FS1<0)
{
    FS1=0;
}

////////////////////////////////////
// Perhitungan Rasio Penulangan //
////////////////////////////////////
RH1=AS1/(B*D);
RHB=(0.75*(0.85*FC*BT1/FY)*(600./(600.+FS)))+RH1*FS1*FS1/FY;
RHO = AS/(B*D);
RMIN=1.4/FY;

////////////////////////////////////
// Menghitung Kendala Rasio Tulangan Balance //
////////////////////////////////////
kendala_rho_b=(RHO/RHB)-1.;
if(kendala_rho_b<0)
{
    kendala_rho_b=0.;
}

////////////////////////////////////
// Menghitung Kendala Rasio Tulangan Minimum //
////////////////////////////////////
kendala_rho_m=(RMIN/RHO)-1.;
if(kendala_rho_m<0)
```

LAMPIRAN 11

[illegible]

LAMPIRAN 11

```
        if(SmakS>Smak[sk])
        {
            SmakS=Smak[sk];
        }
    }

    Sref=SmakS;
    //////////////////////////////////////
    // Menghitung Kendala Jarak Antar Senggang Maksimum //
    //////////////////////////////////////
    kendala_sb=((Jarak_S/SmakS)-1.);
    if(kendala_sb<0.)
    {
        kendala_sb=0.;
    }
}

/*****/
/* FUNGSI UNTUK MENGHITUNG LENDUTAN MAKSIMUM */
/* BERDASARKAN SK SNI-T-15-1991-03 AYAT 3.2.5 */
/*****/
void lendutan(int no_batang_1)
{
    //////////////////////////////////////
    // Hitung Besaran Dasar //
    //////////////////////////////////////
    n=2.E5/(E*1.E-6);

    fr=0.7*sqrt(FC); // (MPa atau N/mm^2)

    //////////////////////////////////////
    // Letak Garis Netral //
    //////////////////////////////////////
    AS=(0.25*pi*(pow(DIA1,2))*NL1);
    AS1=(0.25*pi*(pow(DIA2,2))*NL2);

    LGN=(n*AS/B)*(sqrt(1.+((2.*B*(H-selimut_balok))/(n*AS)))-1.);

    //////////////////////////////////////
    // Menentukan Momen Inersia Penampang Retak Transformasi (mm^4) //
    //////////////////////////////////////
    Icr=(1./3.)*B*pow(LGN,3)+n*AS*pow((H-selimut_balok)-LGN,2);

    //////////////////////////////////////
    // Momen Inersia Penampang Kotor (mm^4) //
    //////////////////////////////////////
    Ig=(1./12.)*B*pow(H,3);

    //////////////////////////////////////
    // Momen Pada Saat Timbul Retak Pertama Kali (Nmm) //
    //////////////////////////////////////
    Mcr=fr*Ig/(0.5*H);

    //////////////////////////////////////
    // Momen Inersia Efektif Untuk Perhitungan Lendutan (mm^4) //
    //////////////////////////////////////
    Ie=pow((Mcr/(MLAP[no_batang_1]*1.e3)),3)*Ig
        +(1.-pow((Mcr/(MLAP[no_batang_1]*1.e3)),3))*Icr;

    //////////////////////////////////////
    // Menghitung Lendutan Seketika //
    //////////////////////////////////////
    // Lendutan Pada Balok (mm) //
    // Dari Buku : Reinforced Concrete Mechanics And Design //
    // Halaman : 355 //
    //////////////////////////////////////
    LENDUTAN=(
        (
            5.*(pow(L*1000.,2))
        )
    )
}
```

LAMPIRAN 11

```
        /(48.*(E*1.E-6)*Ie)
    )*
    (
        fabs(MLAP[no_batang_1])*1000.
    )
);

////////////////////////////////////////
// Menghitung Lendutan Jangka Panjang //
// Menurut SK SNI-T-15-1991-03 pasal 3.2.5(2.5) //
////////////////////////////////////////
Lambda=2./(1.+(50.*AS1/(B*H)));
LENDUTAN=(1.+Lambda)*LENDUTAN;

LENDUTAN_IJIN=(L*1000.)/180.;

kendala_lendutan=((LENDUTAN/LENDUTAN_IJIN)-1.);
if(kendala_lendutan<0)
{
    kendala_lendutan=0.;
}
}
```

LAMPIRAN 12

```

/*****
/* FILE : ELEMEN.HPP
/* PENGISIAN ELEMEN BALOK DAN KOLOM
*****/

/*****
/* PENGISIAN DATA ELEMEN BALOK */
*****/
void isi_elemen_balok(int no_el_balok)
{
    B=isi(var_b[no_struktur][0+(12*no_el_balok)],sisi_d_B); // (mm)
    H=isi(var_b[no_struktur][1+(12*no_el_balok)],sisi_d_H); // (mm)

    DIA1lap=isi(var_b[no_struktur][2+(12*no_el_balok)],DIA_d); // (mm)
    NL1lap=isi(var_b[no_struktur][3+(12*no_el_balok)],NL_d);
    DIA2lap=isi(var_b[no_struktur][4+(12*no_el_balok)],DIA_d); // (mm)
    NL2lap=isi(var_b[no_struktur][5+(12*no_el_balok)],NL_d);

    DIA1tum=isi(var_b[no_struktur][6+(12*no_el_balok)],DIA_d); // (mm)
    NL1tum=isi(var_b[no_struktur][7+(12*no_el_balok)],NL_d);
    DIA2tum=isi(var_b[no_struktur][8+(12*no_el_balok)],DIA_d); // (mm)
    NL2tum=isi(var_b[no_struktur][9+(12*no_el_balok)],NL_d);

    DIAS=isi(var_b[no_struktur][10+(12*no_el_balok)],DIAS_d); // (mm)
    Jarak_S=isi(var_b[no_struktur][11+(12*no_el_balok)],JS_d); // (mm)
    L=EL[no_balok[no_el_balok]]; // (m)

    if(fabs(GESER_KI[no_balok[no_el_balok]])
        >fabs(GESER_KA[no_balok[no_el_balok]]))
    {
        VU = fabs(GESER_KI[no_balok[no_el_balok]]/teta); // (N)
    }
    else
    {
        VU = fabs(GESER_KA[no_balok[no_el_balok]]/teta); // (N)
    }
}

/*****
/* ELEMEN DAERAH LAPANGAN */
*****/
void elemen_lapangan(int no_el_balok)
{
    DIA1=DIA1lap; // Tulangan tarik (mm)
    NL1=NL1lap;
    DIA2=DIA2lap; // Tulangan desak (mm)
    NL2=NL2lap;
    MU=fabs(MLAP[no_balok[no_el_balok]]/(teta)); // momen ditahan (Nm)
}

/*****
/* ELEMEN DAERAH TUMPUAN */
*****/
void elemen_tumpuan(int no_el_balok)
{
    DIA1=DIA1tum; // Tulangan tarik (mm)
    NL1=NL1tum;
    DIA2=DIA2tum; // Tulangan desak (mm)
    NL2=NL2tum;

    if(fabs(MTUM_KI[no_balok[no_el_balok]])
        >fabs(MTUM_KA[no_balok[no_el_balok]]))
    {
        MU=fabs(MTUM_KI[no_balok[no_el_balok]]/teta); // (Nm)
    }
    else
    {
        MU=fabs(MTUM_KA[no_balok[no_el_balok]]/teta); // (Nm)
    }
}

```

LAMPIRAN 12

```

/*****/
/* PENGISIAN DATA ELEMEN KOLOM */
/*****/
void isi_elemen_kolom(int no_el_kolom)
{
    sisi    =isi(var_k[no_struktur][0+(5*no_el_kolom)],sisi_d_K); // (mm)
    DIA     =isi(var_k[no_struktur][1+(5*no_el_kolom)],DIA_d);   // (mm)
    N_DIA    =isi(var_k[no_struktur][2+(5*no_el_kolom)],NL_d);
    DIAS     =isi(var_k[no_struktur][3+(5*no_el_kolom)],DIAS_d); // (mm)
    Jarak_S=isi(var_k[no_struktur][4+(5*no_el_kolom)],JS_d);     // (mm)

    PU  =(-(PK[no_kolom[no_el_kolom]])/(teta)); // (N)
    MUX =(MKX[no_kolom[no_el_kolom]])/(teta);  // (Nm)
    MUY =(MKY[no_kolom[no_el_kolom]])/(teta);  // (Nm)
    VU  =(fabs((GK[no_kolom[no_el_kolom]])/(teta))); // (N)
    L=EL[no_kolom[no_el_kolom]];                // (m)
}

/*****/
/* SUBPROGRAM UNTUK MEMANGGIL ISI DATA ELEMEN */
/*****/
float isi(int no_data,float kelompok_data[])
{
    return kelompok_data[no_data];
}

```

LAMPIRAN 13

```

/* FILE : POLYHEDRON.HPP */
/* OPTIMASI STRUKTUR DENGAN METODA FLEXIBLE POLYHEDRON */
/* DITULIS OLEH YOHAN NAFTALI (JUNI 1999) */
/******/

/*****/
/* SUBPROGRAM UNTUK MELAKUKKAN OPTIMASI */
/*****/

void optimasi()
{
    // Pembangkitan Kelas Ofstream Untuk Penanganan Masalah File //
    ofstream opti(fhistory);

    // Membaca Data Input Dari File //
    baca_data();

    // Menghitung Jumlah Balok Kolom Dan Identifikasi Balok Kolom //
    jum_balok=0;
    jum_kolom=0;
    for(int iop=1;iop<=M;iop++)
    {
        // Memeriksa Kemiringan Batang //
        periksa_batang(iop);

        // Batang Non Vertikal -> Balok //
        if(CXZ>0.001)
        {
            no_balok[jum_balok]=iop;
            jum_balok ++;
        }

        // Batang Vertikal -> Kolom //
        else
        {
            no_kolom[jum_kolom]=iop;
            jum_kolom++;
        }
    }

    // Menghitung Jumlah Variabel Desain //
    JVD=(12*jum_balok)+(5*jum_kolom);

    // Jumlah Struktur Yang Dibangkitkan //
    JSTD=(JVD*fak_kali)+fak_plus;

    // Mengacak Variabel //
    acak_variabel();

    // Mengisi Batas Atas Variabel //
    load_batas_atas();
}

```

LAMPIRAN 13

```
////////////////////////////////////
// Penormalan Batas Variabel Balok Kolom //
////////////////////////////////////
normalisasi_int(nvm,nvb,nvk);

clrscr();
about();
cout << "Generasi ke - 1\n";
cout << "\nJumlah Struktur desain = \n";
cout << "\nWaktu yang telah berjalan : 0 detik\n";
cout << "\nEstimasi waktu yang diperlukan : ?\n";

////////////////////////////////////
// Pembangkitan Generasi Pertama //
// Struktur Dibangkitkan Sebanyak Jumlah Struktur Desain //
////////////////////////////////////

for(int iop=0;iop<JSTD;iop++)
{
    ///////////////////////////////////
    // Inisialisasi Variabel Struktur //
    ///////////////////////////////////
    no_struktur=iop;
    kendalastr[iop]=0.;
    hargastr[iop]=0.;
    fitstr[iop]=0.;

    ///////////////////////////////////
    // Pencatatan Jumlah Struktur Desain Yang Dibangkitkan //
    ///////////////////////////////////
    gotoxy(26,9);
    cout << (iop+1) << " ";

    ///////////////////////////////////
    // Pembacaan Data Umum Struktur //
    ///////////////////////////////////
    baca_data();

    ///////////////////////////////////
    // Membaca Beban Yang Bekerja Pada Struktur //
    ///////////////////////////////////
    baca_beban();

    ///////////////////////////////////
    // Menghitung Gaya-Gaya Struktur //
    ///////////////////////////////////
    inersia();
    struktur();

    ///////////////////////////////////
    // Menghitung Kendala Dan Harga Balok //
    ///////////////////////////////////
    for(int jop=0;jop<jum_balok;jop++)
    {
        ///////////////////////////////////
        // Memeriksa Kemiringan Batang //
        ///////////////////////////////////
        periksa_batang(no_balok[jop]);

        ///////////////////////////////////
        // Mengisi Properti Elemen Balok //
        ///////////////////////////////////
        isi_elemen_balok(jop);

        ///////////////////////////////////
        // Membangkitkan Balok Pada Lapangan //
        ///////////////////////////////////
        elemen_lapangan(jop);
        balok_lapangan;
```

LAMPIRAN 13

```
////////////////////////////////////
// Mengganti jarak sengkang //
////////////////////////////////////
for(int cari_S=(nvb[11+(12*jop)]-1);
    cari_S>0;cari_S--)
{
    if(isi(cari_S,JS_d)<=Sref)
    {
        var_b[no_struktur][11+(12*jop)]=cari_S;
        Jarak_S=isi(cari_S,JS_d);
        break;
    }
}
balok raise_lap;

kendalastr[iop]+=kendala;
hargastr[iop]+=harga;

////////////////////////////////////
// Menghitung Lendutan Pada Tengah Bentang //
////////////////////////////////////
lendutan(no_balok[jop]);
kendalastr[iop]+=kendala_lendutan;

////////////////////////////////////
// Membangkitkan Balok Pada Tumpuan //
////////////////////////////////////
elemen_tumpuan(jop);
balok tumpuan;
kendalastr[iop]+=kendala;
hargastr[iop]+=harga;
}

////////////////////////////////////
// Menghitung Kendala Dan Harga Kolom //
////////////////////////////////////
for(int jop=0;jop<jum_kolom;jop++)
{
    //////////////////////////////////////
    // Periksa Kemiringan Batang //
    //////////////////////////////////////
    periksa_batang(no_kolom[jop]);

    //////////////////////////////////////
    // Mengisi Properti Elemen Kolom //
    //////////////////////////////////////
    isi_elemen_kolom(jop);

    //////////////////////////////////////
    // Membangkitkan Kolom //
    //////////////////////////////////////
    kolom bangkit;

    //////////////////////////////////////
    // Mengganti jarak sengkang //
    //////////////////////////////////////
    for(int cari_S=(nvk[4+(5*jop)]-1);
        cari_S>0;cari_S--)
    {
        if(isi(cari_S,JS_d)<=Sref)
        {
            var_k[no_struktur][4+(5*jop)]=cari_S;
            Jarak_S=isi(cari_S,JS_d);
            break;
        }
    }
}
kolom raise;
```

LAMPIRAN 13

```
////////////////////////////////////////
// Bila Ada Kendala Kelangsingan Ubah Nilai Sisi //
////////////////////////////////////////
if(kendala_kelangsingan>0)
{
    L=EL[no_kolom[jop]];
    float sisi_baru=((0.75*L*1000.)/(22.*sqrt((1./12.))));
    for(int cari_sisi=var_k[no_struktur][0+(5*jop)];
        cari_sisi<nvk[0+(5*jop)];cari_sisi++)
    {
        float sisi_cari=isi(cari_sisi,sisi_d_K);
        if(sisi_cari>sisi_baru)
        {
            var_k[no_struktur][0+(5*jop)]=cari_sisi;
            sisi=isi(cari_sisi,sisi_d_K);
            break;
        }
    }
    // Hitung Kendala Baru //
    // Hitung Kendala Baru //
    kolom bangkit;
}

kendalastr[iop]+=kendala;
hargastr[iop]+=harga;
}

////////////////////////////////////////
// Menghitung Fitness Struktur //
////////////////////////////////////////
fitstr[iop]=(finalti/(hargastr[iop]+(finalti*kendalastr[iop])));
}

////////////////////////////////////////
// Lakukan Pengurutan Fitness //
////////////////////////////////////////
sort(fitstr,kendalastr,hargastr,var_b,var_k,JSTD);

////////////////////////////////////////
// Pengambilan waktu sistem //
////////////////////////////////////////
time_t t_awal, t_sekarang, t_akhir;
time(&t_awal);

////////////////////////////////////////
// Proses Pencarian Hasil Optimal Dimulai //
////////////////////////////////////////
int generasi=1;
jum_susut=0;
do
{
    //////////////////////////////////////////
    // Menulis Nomor Generasi //
    //////////////////////////////////////////
    gotoxy(15,7);
    cout << (generasi+1) << " ";

    //////////////////////////////////////////
    // Membangkitkan Kelas Timer //
    //////////////////////////////////////////
    time(&t_sekarang);

    //////////////////////////////////////////
    // Menulis Lama Waktu Yang Sedang Berjalan //
    //////////////////////////////////////////
    gotoxy(29,11);
    cout << difftime(t_sekarang,t_awal) << " detik";
}
```


LAMPIRAN 13

```
////////////////////////////////////
// Menulis Estimasi Waktu //
////////////////////////////////////
gotoxy(34,13);
cout << int((difftime(t_sekarang,t_awal)/(generasi+1))
        *(j_iterasi_mak-generasi)) << " detik          ";

////////////////////////////////////
// Menentukan Arah Penelusuran //
////////////////////////////////////
penelusuran();

////////////////////////////////////
// Mencari Titik Baru Untuk Menggantikan Titik Worst //
////////////////////////////////////
cari_baru();

////////////////////////////////////
// Membandingkan Fitness Struktur Baru //
////////////////////////////////////

if((fitcb_best)>(fitstr[0]))
{
    if(jum_susut>1)
    {
        opti << "          Penyusutan berturut berakhir setelah "
        << jum_susut << " kali\n";
        jum_susut=0;
    }

    //////////////////////////////////////
    // Ganti Variabel Terjelek Dengan Variabel Baru //
    //////////////////////////////////////
    ganti_baru();
}
else
{
    //////////////////////////////////////
    // Lakukan Penyusutan //
    //////////////////////////////////////
    penyusutan();
    jum_susut++;
    if(jum_susut==1)
    {
        patok_fit=fitstr[JSTD-1];
        opti << "Penyusutan berturut pertama pada generasi : " << generasi
        << " Fitness terbaik : " << fitstr[JSTD-1] << endl;
    }
}

////////////////////////////////////
// Inisialisasi Variabel Struktur //
////////////////////////////////////
no_struktur=0;
kendalastr[0]=0.;
hargastr[0]=0.;
fitstr[0]=0.;

////////////////////////////////////
// Pembacaan Data Umum Struktur //
////////////////////////////////////
baca_data();

////////////////////////////////////
// Membaca Beban Yang Bekerja Pada Struktur //
////////////////////////////////////
baca_beban();
```

LAMPIRAN 13

[illegible]

LAMPIRAN 13

```
//////////////////////////////////////////
// Hentikan Iterasi Apabila Setelah JVD Kali Menyusut //
// Dan Fitness Terbaik Tetap //
//////////////////////////////////////////
if(patok_fit==fitstr[JSTD-1])
{
    opti << "Konvergen Pada Generasi : " << generasi
    << " karena jumlah penyusutan melebihi batas" << endl;
    break;
}

//////////////////////////////////////////
// Jumlah Penyusutan DiReset Lagi //
//////////////////////////////////////////
jum_susut=0;
}

//////////////////////////////////////////
// Increator //
//////////////////////////////////////////
generasi++;
}while(generasi<j_iterasi_mak);

//////////////////////////////////////////
// Mengambil Waktu Akhir //
//////////////////////////////////////////
time(& t_akhir);

//////////////////////////////////////////
// Cetak Ke Layar //
//////////////////////////////////////////
gotoxy(1,17);
cout << "Waktu optimasi : " << difftime(t_akhir,t_awal)
    << " detik \n\a";
cout << "\nFitness Terbaik : " << fitstr[JSTD-1]
    << " " << endl;
cout << "Harga : " << hargastr[JSTD-1]
    << " " << endl;
cout << "Kendala : " << kendalastr[JSTD-1]
    << " " << endl;
gotoxy(55,24);
cout << "Tekan <CR> untuk keluar";
getch();

//////////////////////////////////////////
// Cetak Ke File //
//////////////////////////////////////////
if (generasi==j_iterasi_mak)
{
    opti << "Selesai pada generasi " << (generasi)
    << " karena jumlah iterasi mencapai jumlah iterasi maksimum\n";
}
opti << "Fitness = " << fitstr[JSTD-1] << endl;
opti << "Harga = " << hargastr[JSTD-1] << endl;
opti << "Kendala = " << kendalastr[JSTD-1] << endl;
opti << "Waktu optimasi : " << difftime(t_akhir,t_awal)
    << " detik ";
opti.close();
cetak_akhir();
}
```

LAMPIRAN 14

```

/*****
/* FILE : PENORMALAN.HPP
/* PENANGANAN MASALAH VARIABEL BALOK DAN KOLOM
/* DENGAN VARIABEL GABUNGAN BALOK KOLOM
*****/

/*****
/* SUBPROGRAM MENGEMBALIKAN VARIABEL BALOK DAN KOLOM */
*****/
void unnormalisasi(int var_nor[mak],int var_bv[mak],int var_kv[mak])
{
    int novar=0;
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            var_bv[jsum+(12*isum)]=var_nor[novar];
            novar++;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)
        {
            var_kv[jsum+(5*isum)]=var_nor[novar];
            novar++;
        }
    }
}

/*****
/* SUBPROGRAM MENORMALISASIKAN VARIABEL BALOK DAN KOLOM TIPE FLOAT */
*****/
void normalisasi_float(float var_nor[],float var_bv[],float var_kv[])
{
    int novar=0;
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            var_nor[novar]=var_bv[jsum+(12*isum)];
            novar++;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)
        {
            var_nor[novar]=var_kv[jsum+(5*isum)];
            novar++;
        }
    }
}

/*****
/* SUBPROGRAM MENORMALISASIKAN VARIABEL BALOK DAN KOLOM TIPE INTEGER */
*****/
void normalisasi_int(int var_nor[],int var_bv[],int var_kv[])
{
    int novar=0;
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            var_nor[novar]=var_bv[jsum+(12*isum)];
            novar++;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)

```

LAMPIRAN 14

```
        {
            var_nor[novar]=var_kv[jsum+(5*isum)];
            novar++;
        }
    }
```

LAMPIRAN 15

```

/*****
/* FILE : PENGACAKAN.HPP
/* PUSTAKA URUSAN VARIABEL DESAIN ACAK
*****/

/*****
/* SUBPROGRAM PENGENDALI LOOP RANDOMISASI */
*****/
void acak_variabel()
{
    //////////////////////////////////////
    // Pengacakan Seed Untuk Random Number Generator //
    //////////////////////////////////////
    randomize();

    load_batas_atas();

    for(int iav=0;iav<JSTD;iav++)
    {
        no_struktur=iav;

        if(iav==1)
        {
            cari_struktur_awal();
            continue;
        }

        //////////////////////////////////////
        // Random Struktur Selanjutnya //
        //////////////////////////////////////
        randomisasi();
    }
}

/*****
/* SUBPROGRAM UNTUK MELAKUKAN RANDOMISASI */
*****/
void randomisasi()
{
    //////////////////////////////////////
    // Randomisasi Variabel Desain Balok //
    //////////////////////////////////////
    for(int iran=0;iran<jum_balok;iran++)
    {
        for(int nv_b=0;nv_b<12;nv_b++)
        {
            var_b[no_struktur][nv_b+(12*iran)]=random(nvb[nv_b]);
        }
    }

    //////////////////////////////////////
    // Randomisasi Variabel Desain Kolom //
    //////////////////////////////////////
    for(int iran=0;iran<jum_kolom;iran++)
    {
        for(int nv_k=0;nv_k<5;nv_k++)
        {
            var_k[no_struktur][nv_k+(5*iran)]=random(nvk[nv_k]);
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENENTUKAN BATAS ATAS VARIABEL DESAIN */
*****/
void load_batas_atas()
{
    for(int isinv=0;isinv<jum_balok;isinv++)
    {
        nvb[0+(12*isinv)]=nsisi_B; // B
        nvb[1+(12*isinv)]=nsisi_H; // H
    }
}

```

LAMPIRAN 15

```
        nvb[2+(12*isinv)]=nDIA;    // Diameter tulangan tarik lapangan
        nvb[3+(12*isinv)]=nNL;     // jumlah tulangan tarik lapangan
        nvb[4+(12*isinv)]=nDIA;    // Diameter tulangan desak lapangan
        nvb[5+(12*isinv)]=nNL;     // jumlah tulangan desak lapangan
        nvb[6+(12*isinv)]=nDIA;    // Diameter tulangan tarik tumpuan
        nvb[7+(12*isinv)]=nNL;     // jumlah tulangan tarik tumpuan
        nvb[8+(12*isinv)]=nDIA;    // Diameter tulangan desak tumpuan
        nvb[9+(12*isinv)]=nNL;     // jumlah tulangan desak tumpuan
        nvb[10+(12*isinv)]=nDIAS;   // Diameter tulangan sengkang
        nvb[11+(12*isinv)]=nJS;     // jarak antar tulangan sengkang
    }
    for(int isinv=0;isinv<jum_kolom;isinv++)
    {
        nvk[0+(5*isinv)]=nsisi_K; // Sisi B=H
        nvk[1+(5*isinv)]=nDIA;    // Diameter tulangan bawah lapangan
        nvk[2+(5*isinv)]=nNL;     // jumlah tulangan pada satu sisi
        nvk[3+(5*isinv)]=nDIAS;   // Diameter tulangan sengkang
        nvk[4+(5*isinv)]=nJS;     // jarak antar tulangan sengkang
    }
}

void cari_struktur_awal()
{
    for(int iran=0;iran<jum_balok;iran++)
    {
        for(int nv_b=0;nv_b<12;nv_b++)
        {
            var_b[no_struktur][nv_b+(12*iran)]=(nvb[nv_b]-1);
            if(nv_b==5|nv_b==9|nv_b==11)
            {
                var_b[no_struktur][nv_b+(12*iran)]=0;
            }
        }
    }

    for(int iran=0;iran<jum_kolom;iran++)
    {
        for(int nv_k=0;nv_k<5;nv_k++)
        {
            var_k[no_struktur][nv_k+(5*iran)]=(nvk[nv_k]-1);
        }
    }
}
```

LAMPIRAN 16

```

/*****
/* FILE : DISKRITISASI.HPP
/* SUBPROGRAM UNTUK MELAKUKAN KONVERSI MENJADI BILANGAN DISKRIT
*****/
int konversi(float bil_asli)
{
    if((fabs(bil_asli)-(abs(bil_asli)))<=0.5)
    {
        bil_asli=floor(bil_asli);
    }
    else
    {
        bil_asli=ceil(bil_asli);
    }
    return bil_asli;
}

```


LAMPIRAN 17

```

/*****
/* FILE : KENDALA.HPP
/* SUBPROGRAM UNTUK MENGHITUNG KENDALA
*****/
float Kendala_Harga(int var_b_nya[],int var_k_nya[])
{
    // Inisialisasi Kendala Dan Harga //
    float kendalanya=0.;
    float harganya=0.;

    // Pembacaan Data Umum Struktur //
    baca_data();

    // Membaca Beban Yang Bekerja Pada Struktur //
    baca_beban();

    // Menghitung Gaya-Gaya Struktur //
    inersia();
    struktur();

    // Menghitung Kendala Dan Harga Balok //
    for(int iKH=0;iKH<jum_balok;iKH++)
    {
        // Memeriksa Kemiringan Batang //
        periksa_batang(no_balok[iKH]);

        // Pengisian Elemen Balok //
        B=isi(var_b_nya[0+(12*iKH)],sisi_d_B);
        H=isi(var_b_nya[1+(12*iKH)],sisi_d_H);

        DIA1lap=isi(var_b_nya[2+(12*iKH)],DIA_d);
        NL1lap=isi(var_b_nya[3+(12*iKH)],NL_d);
        DIA2lap=isi(var_b_nya[4+(12*iKH)],DIA_d);
        NL2lap=isi(var_b_nya[5+(12*iKH)],NL_d);

        DIA1tum=isi(var_b_nya[6+(12*iKH)],DIA_d);
        NL1tum=isi(var_b_nya[7+(12*iKH)],NL_d);
        DIA2tum=isi(var_b_nya[8+(12*iKH)],DIA_d);
        NL2tum=isi(var_b_nya[9+(12*iKH)],NL_d);

        DIAS=isi(var_b_nya[10+(12*iKH)],DIAS_d);
        Jarak_S=isi(var_b_nya[11+(12*iKH)],JS_d);
        L=EL[no_balok[iKH]];

        // Menentukan Gaya Geser Yang Terjadi //
        if(fabs(GESER_KI[no_balok[iKH]])>GESER_KA[no_balok[iKH]])
        {
            VU = fabs(GESER_KI[no_balok[iKH]])/teta;
        }
        else
        {
            VU = fabs(GESER_KA[no_balok[iKH]])/teta;
        }
    }
}

```

LAMPIRAN 17

```
//////////
// Daerah Lapangan //
//////////
DIA1=DIA1lap;
NL1=NL1lap;
DIA2=DIA2lap;
NL2=NL2lap;
MU=fabs(MLAP[no_balok[iKH]]/teta);

//////////
// Membangkitkan Balok Pada Lapangan //
//////////
balok_lapangan;
kendalanya+=kendala;
harganya+=harga;

lendutan(no_balok[iKH]);
kendalanya+=kendala_lendutan;

//////////
// Daerah Tumpuan //
//////////
DIA1=DIA1tum;
NL1=NL1tum;
DIA2=DIA2tum;
NL2=NL2tum;

if(fabs(MTUM_KI[no_balok[iKH]])>fabs(MTUM_KA[no_balok[iKH]]))
{
    MU=fabs(MTUM_KI[no_balok[iKH]]/teta);
}
else
{
    MU=fabs(MTUM_KA[no_balok[iKH]]/teta);
}

//////////
// Membangkitkan Balok Pada Tumpuan //
//////////
balok_tumpuan;
kendalanya+=kendala;
harganya+=harga;
}

//////////
// Menghitung Kendala Dan Harga Kolom Biaksial //
//////////
for(int iKH=0;iKH<jum_kolom;iKH++)
{
    //////////
    // Memeriksa Kemiringan Batang //
    //////////
    periksa_batang(no_kolom[iKH]);

    //////////
    // Pengisian Data Elemen //
    //////////
    sisi    =isi(var_k_nya[0+(5*iKH)],sisi_d_K); // sisi kolom (mm)
    DIA     =isi(var_k_nya[1+(5*iKH)],DIA_d);    // Diameter tulangan
                                                // utama (mm)

    N_DIA   =isi(var_k_nya[2+(5*iKH)],NL_d);    // Jumlah tulangan
    DIAS    =isi(var_k_nya[3+(5*iKH)],DIAS_d);  // Diameter sengkang (mm)
    Jarak_S =isi(var_k_nya[4+(5*iKH)],JS_d);    // Jarak antara
                                                // sengkang (mm)

    //////////
    // Gaya Aksial Yang Harus Ditahan //
    //////////
    PU      =fabs((PK[no_kolom[iKH]]/(teta))); // (N)
```

LAMPIRAN 17

```
////////////////////////////////////
// Momen Yang Harus Ditahan //
////////////////////////////////////
MUX =(MKX[no_kolom[iKH]])/(teta); // (Nm)
MUY =(MKY[no_kolom[iKH]])/(teta); // (Nm)

////////////////////////////////////
// Gaya Geser Yang Harus Ditahan //
////////////////////////////////////
VU =fabs((GK[no_kolom[iKH]])/(teta)); // (N)

L=EL[no_kolom[iKH]]; // Panjang kolom (m)

////////////////////////////////////
// Membangkitkan Kolom //
////////////////////////////////////
kolom bangkit;

for(int cari_S=(nvk[4+(5*iKH)]-1);
    cari_S>0;cari_S--)
{
    if(isi(cari_S,JS_d)<=Sref)
    {
        var_k_nya[4+(5*iKH)]=cari_S;
        Jarak_S=isi(cari_S,JS_d);
        break;
    }
}
    kolom raise;

////////////////////////////////////
// Bila Ada Kendala Kelangsingan Ubah Nilai Sisi //
////////////////////////////////////
if(kendala_kelangsingan>0)
{
    L=EL[no_kolom[iKH]]; // Panjang kolom (m)
    float sisi_baru=((0.75*L*1000.)/(22.*sqrt((1./12.))));
    for(int cari_sisi=var_k_nya[0+(5*iKH)];
        cari_sisi<nvk[0+(5*iKH)];cari_sisi++)
    {
        float sisi_cari=isi(cari_sisi,sisi_d_K);
        if(sisi_cari>sisi_baru)
        {
            var_k_nya[0+(5*iKH)]=cari_sisi;
            sisi=isi(cari_sisi,sisi_d_K);
            break;
        }
    }
    // Hitung Kendala Baru //
    kolom bangkit;
}

kendalanya+=kendala;
harganya+=harga;
}
return(finalti/(harganya+(finalti*kendalanya)));
}
```

LAMPIRAN 18

```

/*****
/* FILE : TELUSUR.HPP
/* SUBPROGRAM UNTUK MELAKUKAN PENELUSURAN MENUJU TITIK BARU
*****/
void penelusuran()
{
    ///////////////////////////////////////////////////
    // Mencari Letak Titik Midle //
    ///////////////////////////////////////////////////

    ///////////////////////////////////////////////////
    // Inisialisasi Awal //
    ///////////////////////////////////////////////////
    float sum_bestgood_b[mak];
    float sum_bestgood_k[mak];
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            sum_bestgood_b[jsum+(12*isum)]=0.;
        }
    }
    for(int isum=0;isum<jum_kolom;isum++)
    {
        for(int jsum=0;jsum<5;jsum++)
        {
            sum_bestgood_k[jsum+(5*isum)]=0.;
        }
    }

    ///////////////////////////////////////////////////
    // Jumlah Kordinat = 12*jum_balok_p + 5*jum_kolom_p //
    // Menjumlahkan Variabel Best Dan Good Pada Balok //
    ///////////////////////////////////////////////////
    for(int nbgw=1;nbgw<JSTD;nbgw++)
    {
        ///////////////////////////////////////////////////
        // 12*jum_balok_p Titik M //
        ///////////////////////////////////////////////////
        for(int isum=0;isum<jum_balok;isum++)
        {
            for(int jsum=0;jsum<12;jsum++)
            {
                sum_bestgood_b[jsum+(12*isum)]+=var_b[nbgw][jsum+(12*isum)];
            }
        }

        ///////////////////////////////////////////////////
        // 5*jum_kolom_p Titik M //
        ///////////////////////////////////////////////////
        for(int isum=0;isum<jum_kolom;isum++)
        {
            for(int jsum=0;jsum<5;jsum++)
            {
                sum_bestgood_k[jsum+(5*isum)]+=var_k[nbgw][jsum+(5*isum)];
            }
        }
    }

    ///////////////////////////////////////////////////
    // Dari Sini Kita Mempunyai jum_kolom_p Koordinat //
    // Dan jum_balok_p Koordinat //
    // Menentukan Koordinat Titik M //
    ///////////////////////////////////////////////////
    for(int isum=0;isum<jum_balok;isum++)
    {
        for(int jsum=0;jsum<12;jsum++)
        {
            XM_b[jsum+(12*isum)]=sum_bestgood_b[jsum+(12*isum)]/(JSTD-1.);
        }
    }
}

```

LAMPIRAN 18

```
for(int isum=0;isum<jum_kolom;isum++)
{
    for(int jsum=0;jsum<5;jsum++)
    {
        XM_k[jsum+(5*isum)]=sum_bestgood_k[jsum+(5*isum)]/(JSTD-1.);
    }
}

////////////////////////////////////
// Mencari Arah Penelusuran Pada Balok //
////////////////////////////////////
for(int isum=0;isum<jum_balok;isum++)
{
    for(int jsum=0;jsum<12;jsum++)
    {
        XS_b[jsum+(12*isum)]=(XM_b[jsum+(12*isum)]
                                -var_b[0][jsum+(12*isum)]);
        if(XS_b[jsum+(12*isum)]>0.)
        {
            tanda_arah_b[jsum+(12*isum)]=1.;
        }
        else
        {
            if(XS_b[jsum+(12*isum)]<0)
            {
                tanda_arah_b[jsum+(12*isum)]=(-1.);
            }
            if(XS_b[jsum+(12*isum)]==0)
            {
                tanda_arah_b[jsum+(12*isum)]=(0.);
            }
        }
    }
}

////////////////////////////////////
// Mencari Arah Penelusuran Pada Kolom //
////////////////////////////////////
for(int isum=0;isum<jum_kolom;isum++)
{
    for(int jsum=0;jsum<5;jsum++)
    {
        XS_k[jsum+(5*isum)]=(XM_k[jsum+(5*isum)]
                                -var_k[0][jsum+(5*isum)]);
        if(XS_k[jsum+(5*isum)]>0.)
        {
            tanda_arah_k[jsum+(5*isum)]=1.;
        }
        else
        {
            if(XS_k[jsum+(5*isum)]<0)
            {
                tanda_arah_k[jsum+(5*isum)]=(-1.);
            }
            if(XS_k[jsum+(5*isum)]==0)
            {
                tanda_arah_k[jsum+(5*isum)]=(0.);
            }
        }
    }
}

////////////////////////////////////
// Penggandaan Variabel Terjelek //
////////////////////////////////////

////////////////////////////////////
// 12*jum_balok_p Koordinat Titik Terjelek //
////////////////////////////////////
for(int isum=0;isum<jum_balok;isum++)
```

LAMPIRAN 18

```
{
    for(int jsum=0;jsum<12;jsum++)
    {
        var_b_jelek[jsum+(12*isum)]=var_b[0][jsum+(12*isum)];
    }
}

////////////////////////////////////
// 5*jum_kolom_p Koordinat Titik Terjelek //
////////////////////////////////////
for(int isum=0;isum<jum_kolom;isum++)
{
    for(int jsum=0;jsum<5;jsum++)
    {
        var_k_jelek[jsum+(5*isum)]=var_k[0][jsum+(5*isum)];
    }
}

////////////////////////////////////
// Inisialisasi Awal = 0 //
////////////////////////////////////
for(int isum=0;isum<JVD;isum++)
{
    TM[isum]=0.;
    TS[isum]=0.;
    varnew_asli[isum]=0;
    arah[isum]=0;
}
normalisasi_float(TM,XM_b,XM_k);
normalisasi_float(TS,XS_b,XS_k);
normalisasi_int(arah,tanda_arah_b,tanda_arah_k);

////////////////////////////////////
// Cari nomor Arah Terjauh //
////////////////////////////////////
no_TS_terjauh=0;
for(int cari_ts=1;cari_ts<JVD;cari_ts++)
{
    if(fabs(TS[cari_ts])>fabs(TS[no_TS_terjauh]))
    {
        no_TS_terjauh=cari_ts;
    }
}
}
```

LAMPIRAN 19

```

/*****
/* FILE : BARU.HPP
/* PUSTAKA PENANGANAN TITIK BARU
*****/

/*****
/* SUBPROGRAM UNTUK MENCARI TITIK BARU */
*****/
void cari_baru()
{
    ////////////////////////////////////
    // Inisialisasi Awal //
    ////////////////////////////////////
    iterasi_var=0;

    normalisasi_int(varnew_asli,var_b_jelek,var_k_jelek);
    fitch_best=0.;
    do
    {
        gotoxy(1,15);
        cprintf ("Pencarian arah baru ke - %d",(iterasi_var+1));
        cprintf ("          ");

        ////////////////////////////////////
        // Menentukan Titik X Pertama Baru //
        // Diambil Yang Arahnya Paling Jauh //
        ////////////////////////////////////
        varplus[no_TS_terjauh]=(iterasi_var+1.);

        ////////////////////////////////////
        // Variabel Baru Ditambahkan Sesuai Arahnya //
        ////////////////////////////////////
        varnew_f[no_TS_terjauh]=varnew_asli[no_TS_terjauh]
                                +varplus[no_TS_terjauh]
                                *arah[no_TS_terjauh];

        ////////////////////////////////////
        // Menentukan X Baru Lain //
        ////////////////////////////////////
        for (int icb=0;icb<JVD;icb++)
        {
            ////////////////////////////////////
            // Skip Bila icb Adalah Nomor arah Pencarian Terjauh //
            ////////////////////////////////////
            if(icb==no_TS_terjauh)
            {
                continue;
            }

            if(TS[no_TS_terjauh]!=0)
            {
                ////////////////////////////////////
                // Menghitung Penambahannya //
                ////////////////////////////////////
                varplus[icb]=fabs((TS[icb]/TS[no_TS_terjauh])
                                *varplus[no_TS_terjauh]);
                varnew_f[icb]=varnew_asli[icb]
                                +varplus[icb]*arah[icb];
            }
            else
            {
                varnew_f[icb]=varnew_asli[icb]
                                +((iterasi_var+1.)*arah[icb]);
            }
        }

        for(int icb=0;icb<JVD;icb++)
        {
            varnew[icb]=konversi(varnew_f[icb]);
        }
    }
}
```

LAMPIRAN 19

```
////////////////////////////////////
// Memeriksa Apakah Titik Baru Tersebut Identik Dengan Titik M //
////////////////////////////////////
for(int ilp=0;ilp<JVD;ilp++)
{
    if(varnew[ilp]==TM[ilp])
    {
        if(ilp==(JVD-1))
        {
            lompat=1;
        }
    }
    else
    {
        lompat=0;
        break;
    }
}

////////////////////////////////////
// Lompat Dari Loop Apabila Indikator Pelompat bernilai 1 //
////////////////////////////////////
if(lompat==1)
{
    lompat=0;
    iterasi_var++;
    continue;
}

periksa_batas();

////////////////////////////////////
// Perubahan Variabel Umum Ke Variabel Balok Kolom //
////////////////////////////////////
unnormalisasi(varnew,var_b_cb,var_k_cb);

////////////////////////////////////
// Menghitung Kendala //
////////////////////////////////////
fitchb=Kendala_Harga(var_b_cb,var_k_cb);

if(fitchb>fitchb_best)
{
    fitchb_best=fitchb;
    for(int icb=0;icb<jum_balok;icb++)
    {
        for(int jcb=0;jcb<12;jcb++)
        {
            var_b_cb_best[jcb+(12*icb)]=var_b_cb[jcb+(12*icb)];
        }
    }
    for(int icb=0;icb<jum_kolom;icb++)
    {
        for(int jcb=0;jcb<5;jcb++)
        {
            var_k_cb_best[jcb+(5*icb)]=var_k_cb[jcb+(5*icb)];
        }
    }
    iterasi_var++;
}while(iterasi_var<(fabs(TS[no_TS_terjauh])*3));
}

/*****/
/* SUBPROGRAM UNTUK MEMERIKSA BATAS ATAS */
/* DAN BAWAH VARIABEL DESAIN BARU */
/* DAN BILA TERJADI PELANGGARAN MAKA */
/* VARIABEL ITU AKAN DICERMINKAN */
/*****/
void periksa_batas()
```


LAMPIRAN 19

```
{
    for(int ipb=0;ipb<JVD;ipb++)
    {
        if((varnew[ipb])>(nvm[ipb]-1))
        {
            varnew[ipb]=(nvm[ipb]-1);
        }
        if(varnew[ipb]<0)
        {
            varnew[ipb]=0;
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENGGANTI VARIABEL TERJELEK DENGAN VARIABEL BARU */
*****/
void ganti_baru()
{
    for(int igbar=0;igbar<jum_balok;igbar++)
    {
        for(int jgbar=0;jgbar<12;jgbar++)
        {
            var_b[0][jgbar+(12*igbar)]=var_b_cb_best[jgbar+(12*igbar)];
        }
    }
    for(int igbar=0;igbar<jum_kolom;igbar++)
    {
        for(int jgbar=0;jgbar<5;jgbar++)
        {
            var_k[0][jgbar+(5*igbar)]=var_k_cb_best[jgbar+(5*igbar)];
        }
    }
}

/*****
/* SUBPROGRAM UNTUK MENYUSUTKAN VARIABEL DESAIN */
*****/
void penyusutan()
{
    for(int nkon=0;nkon<(JSTD-1);nkon++)
    {
        for(int igbar=0;igbar<jum_balok;igbar++)
        {
            for(int jgbar=0;jgbar<12;jgbar++)
            {
                var_b[nkon][jgbar+(12*igbar)]=
                konversi(0.5*(var_b[nkon][jgbar+(12*igbar)]
                +var_b[JSTD-1][jgbar+(12*igbar)]));
            }
        }
        for(int igbar=0;igbar<jum_kolom;igbar++)
        {
            for(int jgbar=0;jgbar<5;jgbar++)
            {
                var_k[nkon][jgbar+(5*igbar)]=
                konversi(0.5*(var_k[nkon][jgbar+(5*igbar)]
                +var_k[JSTD-1][jgbar+(5*igbar)]));
            }
        }
    }
}
```

LAMPIRAN 20

```

/*****
/* FILE : PENGURUTAN.HPP
/* SUBPROGRAM UNTUK MENGURUTKAN DATA BERDASARKAN FITNESS
/* DENGAN BANTUAN METODA BUBBLE SORT
*****/
void sort(float nilai1[],float nilai2[],float nilai3[]
          ,int urut_var_b[][mak],int urut_var_k[][mak]
          ,int n_array)
{
    ///////////////////////////////////////////////////////////////////
    // Metode Bubble Sort //
    ///////////////////////////////////////////////////////////////////

    ///////////////////////////////////////////////////////////////////
    // Nilai 1 Untuk Fitness
    // Nilai 2 Untuk Kendala
    // Nilai 3 Untuk Harga
    // urut_var_b Untuk Variabel Balok
    // urut_var_k Untuk Variabel Kolom
    // n_array Adalah Jumlah Titik Yang Akan Diurutkan Fitnessnya
    ///////////////////////////////////////////////////////////////////
    double dummy1=0.;
    float dummy2=0.;
    float dummy3=0.;
    int dummyvarb[mak];
    int dummyvark[mak];
    for (int dum1=0;dum1<JSTD;dum1++)
    {
        dummyvarb[dum1]=0;
        dummyvark[dum1]=0;
    }

    ///////////////////////////////////////////////////////////////////
    // Pemeriksaan Dimulai Dari Nilai Terkecil+1 //
    ///////////////////////////////////////////////////////////////////
    for(int isort=1;isort<n_array;++isort)
    {
        ///////////////////////////////////////////////////////////////////
        // Perbandingan Kepada Nilai-Nilai Sebelumnya //
        ///////////////////////////////////////////////////////////////////
        for(int jsort=0;jsort<=isort;++jsort)
        {
            ///////////////////////////////////////////////////////////////////
            // Bila Nilai Sekarang Lebih Kecil Daripada Nilai Sebelumnya //
            // Maka Dilakukan Penukaran Posisi Ke Tempat Nilai Sebelumnya //
            ///////////////////////////////////////////////////////////////////
            if(nilai1[isort]<nilai1[jsort])
            {
                ///////////////////////////////////////////////////////////////////
                // Pengisian Nilai Sekarang Ke Variabel Dummy //
                ///////////////////////////////////////////////////////////////////
                dummy1=nilai1[isort];
                dummy2=nilai2[isort];
                dummy3=nilai3[isort];
                for(int dvb=0;dvb<jum_balok;dvb++)
                {
                    for(int dnv_b=0;dnv_b<12;dnv_b++)
                    {
                        dummyvarb[dnv_b+(12*dvb)]
                            =urut_var_b[isort][dnv_b+(12*dvb)];
                    }
                }
                for(int dvk=0;dvk<jum_kolom;dvk++)
                {
                    for(int dnv_k=0;dnv_k<5;dnv_k++)
                    {
                        dummyvark[dnv_k+(5*dvk)]
                            =urut_var_k[isort][dnv_k+(5*dvk)];
                    }
                }
            }
        }
    }
}

```

LAMPIRAN 20

```
//////////////////////////////////////////
// Pemindahan Nilai Yang Akan Digantikan Posisinya //
// Ke Posisi Sekarang //
//////////////////////////////////////////
nilai1[isort]=nilai1[jsort];
nilai2[isort]=nilai2[jsort];
nilai3[isort]=nilai3[jsort];
for(int dvb=0;dvb<jum_balok;dvb++)
{
    for(int dnv_b=0;dnv_b<12;dnv_b++)
    {
        urut_var_b[isort][dnv_b+(12*dvb)]
        =urut_var_b[jsort][dnv_b+(12*dvb)];
    }
}
for(int dvk=0;dvk<jum_kolom;dvk++)
{
    for(int dnv_k=0;dnv_k<5;dnv_k++)
    {
        urut_var_k[isort][dnv_k+(5*dvk)]
        =urut_var_k[jsort][dnv_k+(5*dvk)];
    }
}

//////////////////////////////////////////
// Pemindahan Nilai Sekarang Ke Posisi Yang Digantikan //
// Ke Posisi Sekarang //
//////////////////////////////////////////
nilai1[jsort]=dummy1;
nilai2[jsort]=dummy2;
nilai3[jsort]=dummy3;
for(int dvb=0;dvb<jum_balok;dvb++)
{
    for(int dnv_b=0;dnv_b<12;dnv_b++)
    {
        urut_var_b[jsort][dnv_b+(12*dvb)]
        =dummyvarb[dnv_b+(12*dvb)];
    }
}
for(int dvk=0;dvk<jum_kolom;dvk++)
{
    for(int dnv_k=0;dnv_k<5;dnv_k++)
    {
        urut_var_k[jsort][dnv_k+(5*dvk)]
        =dummyvark[dnv_k+(5*dvk)];
    }
}

//////////////////////////////////////////
// Proses Penukaran Selesai //
// Melanjutkan Nilai Selanjutnya //
//////////////////////////////////////////
}
}

//////////////////////////////////////////
// Sesudah Diurutkan Nilai Terjelek Terdapat Pada Array 0 //
// Nilai Terbaik Terdapat Pada (n_array-1) //
//////////////////////////////////////////
}
```

LAMPIRAN 21

```

/*****
/* FILE : TAMPILAN.HPP
/* KUMPULAN SUBPROGRAM PENANGANAN TAMPILAN KE LAYAR
*****/

/*****
/* SUBPROGRAM UNTUK MENAMPILKAN MENU UTAMA */
*****/
void menu_utama()
{
    int pilih;
    do
    {
        textbackground(1);
        textcolor(LIGHTGREEN);
        clrscr();
        about();
        cout << " 1. Input data awal ke file\n";
        cout << " 2. Input data beban ke file\n";
        cout << " 3. Melihat isi file input\n";
        cout << " 4. Mengoptimasi struktur\n";
        cout << " 5. Keluar\n";
        cout << " Pilihan (1-5) = ";
        struct time t;

        gettime(&t);
        textcolor(YELLOW);
        gotoxy(61,6);
        printf("Time : %2d:%02d:%02d.%02d\n",
            t.ti_hour, t.ti_min, t.ti_sec, t.ti_hund);
        gotoxy(19,12);
        cin >> pilih;

        if(pilih==1)
        {
            input_data();
        }
        if(pilih==2)
        {
            load_data(); // Memasukan data beban
            cout << " Data beban sudah diisi...\n";
        }
        if(pilih==3)
        {
            cout << " Nama file input yang akan dibaca = ";
            cin >> fi;
            strcpy(fininput,fi);
            strcat(fininput, ".inp");
            strcpy(fbeban,fi);
            strcat(fbeban, ".bbn");
            baca_data();
            baca_beban();
            do
            {
                clrscr();
                about();
                cout << " 1. Data umum\n";
                cout << " 2. Koordinat titik kumpul\n";
                cout << " 3. Informasi batang\n";
                cout << " 4. Pengekang titik kumpul\n";
                cout << " 5. Gaya ujung batang terkekang\n";
                cout << " 6. Beban pada joint\n";
                cout << " 7. Keluar ke menu utama\n";
                cout << " Pilihan (1-7) = ";
                cin >> pilih;
                clrscr();
                if(pilih==1)
                {
                    output_parameter_struktural();
                }
            }
            if(pilih==2)

```

LAMPIRAN 21

```
{
    output_koordinat_titik_kumpul();
}
if(pilih==3)
{
    output_informasi_batang();
}
if(pilih==4)
{
    output_pengekang_titik_kumpul();
}
if(pilih==5)
{
    output_beban_batang();
}
if(pilih==6)
{
    output_beban_titik();
}
if(pilih==7)
{
    cout << "  Anda keluar ke Menu Utama\n";
}
cout << "\n  Tekan Sembarang Tombol\n";
getch();
}while(pilih!=7);
}
if(pilih==4)
{
    cout << "  Nama file stuktur (tanpa ekstensi) = ";
    cin >> fi;
    strcpy(fininput,fi);
    strcat(fininput, ".inp");      // Nama file generik
    strcpy(fsisi,fi);
    strcat(fsisi, ".isd");        // input sisi diskrit
    strcpy(fdial,fi);
    strcat(fdial, ".idl");        // input diameter tulangan lentur
    strcpy(fjtl,fi);
    strcat(fjtl, ".ijl");        // input jumlah tulangan lentur
    strcpy(fdias,fi);
    strcat(fdias, ".ids");        // input diameter tulangan sengkang
    strcpy(fjts,fi);
    strcat(fjts, ".ijs");        // input jarak tulangan sengkang
    strcpy(fbeban,fi);
    strcat(fbeban, ".bbn");       // input beban pada batang
    strcpy(fhistory,fi);
    strcat(fhistory, ".his");     // Keluaran dari riwayat optimasi
    strcpy(foptimasi,fi);
    strcat(foptimasi, ".opt");    // Keluaran dari hasil optimasi
    strcpy(fstruktur,fi);
    strcat(fstruktur, ".str");    // Keluaran dari hasil analisa struktur
    strcpy(fkendala,fi);
    strcat(fkendala, ".kdl");     // Keluaran kendala pada struktur
    strcpy(finformasi,fi);
    strcat(finformasi, ".inf");   // Keluaran informasi masukkan

    cout << "  file input = " << fininput << endl;
    cout << "  file beban = " << fbeban << endl;
    cout << "  Harga Beton (Rp./m^3) = ";
    cin >> harga_beton;
    cout << "  Harga Besi (Rp./kg) = ";
    cin >> harga_besi;
    cout << "  Tebal selimut kolom (minimum 40 mm) = ";
    cin >> selimut_kolom;
    cout << "  Tebal selimut balok (minimum 40 mm) = ";
    cin >> selimut_balok;
    cout << "  Faktor finalti = ";
    cin >> finalti;
    cout << "  Iterasi maksimum = ";
    cin >> j_iterasi_mak;
```

LAMPIRAN 21

```
        cout << "  Faktor penambah jumlah variabel desain = ";
        cin >> fak_plus;
        do
        {
            cout << "  Faktor pengali jumlah variabel desain = ";
            cin >> fak_kali;
            if(fak_kali<=0)
            {
                "  Faktor pengali minimal diisi 1 !\n";
            }
        }while(fak_kali<=0);
        optimasi();
    }
    if(pilih==5)
    {
        cout << "\n                                Anda keluar dari program\n";
    }
} while(pilih!=5);
}

/*****/
/* SUBPROGRAM UNTUK MENAMPILKAN INFORMASI PROGRAM */
/*****/
void about()
{
    cout << "*****";
    cout << "*****";
    cout << "
    << "Program Optimasi Beton Bertulang Pada Struktur Portal Ruang\n";
    cout << "
    << "Oleh : Yohan Naftali\n";
    cout << "
    << "7712/TS\n";
    cout << "*****";
    cout << "*****\n";
}
```

LAMPIRAN 22

```

/*****
/* FILE : CETAK.HPP
/* SUBPROGRAM UNTUK MENCETAK HASIL AKHIR OPTIMASI
/* DITULIS OLEH YOHAN NAFTALI (JULI 1999)
*****/

void cetak_akhir()
{
    ofstream hopt(foptimasi);
    ofstream hstr(fstruktur);
    ofstream hkdl(fkendala);
    ofstream hinf(finformasi);

    ////////////////////////////////////
    // Pembacaan Data Umum Struktur //
    ////////////////////////////////////
    baca_data();

    ////////////////////////////////////
    // Membaca Beban Yang Bekerja Pada Struktur //
    ////////////////////////////////////
    baca_beban();

    ////////////////////////////////////
    // Mencetak Informasi Masukkan Ke File //
    ////////////////////////////////////
    hinf << " Struktur Portal Ruang " << ISN << "\n\n";
    hinf << " Parameter Struktur\n";
    hinf << " Jumlah batang : " << M << endl;
    hinf << " DOF : " << N << endl;
    hinf << " Jumlah joint : " << NJ << endl;
    hinf << " Jumlah pengekang tumpuan : " << NR << endl;
    hinf << " Jumlah titik kumpul yang dikekang : " << NRJ << endl;
    hinf << " Modulus Elastisitas aksial : " << E << " N/m^2\n";
    hinf << " Modulus Geser : " << G << " N/m^2\n\n";
    hinf << " Properti Elemen Material\n";
    hinf << " Kuat desak beton karakteristik : " << FC << " MPa\n";
    hinf << " Kuat tarik baja tulangan : " << FY << " MPa\n";
    hinf << " Kuat tarik tulangan sengkang : " << FYS << " MPa\n";

    hinf << "\n Koordinat Titik Kumpul (m)\n";
    hinf << " Titik X Y Z \n";
    for(int kout=1;kout<=NJ;kout++)
    {
        hinf << setiosflags(ios::left);
        hinf << " " << setw(10) << kout
            << setw(14) << X[kout]
            << setw(14) << Y[kout]
            << setw(14) << Z[kout] << endl;
    }

    hinf << "\n Informasi Batang \n";
    hinf << " Batang JJ JK IA\n";
    for(int iout=1;iout<=M;iout++)
    {
        hinf << setiosflags(ios::left);
        hinf << " " << setw(8) << iout
            << setw(8) << JJ[iout]
            << setw(8) << JK[iout]
            << setw(3) << IA[iout] << endl;
        if(IA[iout] != 0)
        {
            hinf << setiosflags(ios::left);
            hinf << " XP = " << setw(12) << XP[iout];
            hinf << " YP = " << setw(12) << YP[iout] << '\t';
            hinf << " ZP = " << setw(12) << ZP[iout] << endl;
            hinf << endl;
        }
    }

    hinf << "\n Pengekang Titik Kumpul \n";
    hinf << " Titik JR1 JR2 JR3 JR4 JR5 JR6\n";

```

LAMPIRAN 22

```

for (int iout=1;iout<=NRJ;iout++)
{
    hinf << setiosflags(ios::left);
    hinf << " " << setw(10)<<T_K[iout]
        << setw(6) << JRL[6*T_K[iout]-5]
        << setw(6) << JRL[6*T_K[iout]-4]
        << setw(6) << JRL[6*T_K[iout]-3]
        << setw(6) << JRL[6*T_K[iout]-2]
        << setw(6) << JRL[6*T_K[iout]-1]
        << setw(6) << JRL[6*T_K[iout]] << endl;
}

hinf << "\n Gaya di Ujung Batang Terkekang Akibat Beban (Nm)\n"
    << " Batang AML1 AML2 AML3 "
        << "AML4 AML5 AML6\n"
    << " AML7 AML8 AML9 "
        << "AML10 AML11 AML12\n";
for(int kout=1;kout<=M;kout++)
{
    hinf << setiosflags(ios::left);
    hinf << " " << setw(8) << kout
        << setw(12) << AML[1][kout]
        << setw(12) << AML[2][kout]
        << setw(12) << AML[3][kout]
        << setw(12) << AML[4][kout]
        << setw(12) << AML[5][kout]
        << setw(12) << AML[6][kout] << endl;
    hinf << " "
        << setw(12) << AML[7][kout]
        << setw(12) << AML[8][kout]
        << setw(12) << AML[9][kout]
        << setw(12) << AML[10][kout]
        << setw(12) << AML[11][kout]
        << setw(12) << AML[12][kout] << endl;
}

hinf << "\n Beban Titik (N)\n"
    << " Titik Arah 1 Arah 2 Arah 3 "
        << "Arah 4 Arah 5 Arah 6\n";
for(int kout=1;kout<=NJ;kout++)
{
    hinf << setiosflags(ios::left);
    hinf << " " << setw(8) << kout
        << setw(12) << AJ[((6*kout)-5)]
        << setw(12) << AJ[((6*kout)-4)]
        << setw(12) << AJ[((6*kout)-3)]
        << setw(12) << AJ[((6*kout)-2)]
        << setw(12) << AJ[((6*kout)-1)]
        << setw(12) << AJ[(6*kout)] << endl;
}

////////////////////////////////////
// Menghitung Gaya-Gaya Struktur //
////////////////////////////////////
inersia();
struktur();

////////////////////////////////////
// Cetak Hasil Analisa Struktur //
////////////////////////////////////
hstr << "Hasil Analisa Struktur Dengan Metoda Kekakuan\n";
hstr << "Dikembangkan dari Weaver & Gere\n";
hstr << "Oleh Yohan Naftali 1999\n\n";
hstr << "Nama File Generik : " << fi << endl;
hopt << "Nama Struktur : " << ISN << endl;
hstr << "Jumlah batang : " << M << endl;
hstr << "Jumlah titik kumpul : " << NJ << endl;
hstr << "Modulus elastisitas tarik/tekan : "
    << (E*1.E-6) << " MPa" << endl;
hstr << "Modulus elastisitas geser G : "
    << (G*1.E-6) << " MPa" << endl;

```


LAMPIRAN 22

```
hstr << "\nPerpindahan Titik Kumpul\n";
hstr << "Titik   DJ1           DJ2           DJ3   ";
hstr << "           DJ4           DJ5           DJ6\n";
for(int cst=1; cst<=NJ; cst++)
{
    hstr << setiosflags(ios::left);
    hstr << setw(7) << cst;
    hstr << setiosflags( ios::fixed | ios:: showpos);
    hstr << setw(12) << DJ[6*cst-5] << setw(12);
    hstr << DJ[6*cst-4] << setw(12) << DJ[6*cst-3] << setw(12);
    hstr << DJ[6*cst-2] << setw(12) << DJ[6*cst-1] << setw(12);
    hstr << DJ[6*cst] << endl;
    hstr << resetiosflags( ios::fixed | ios:: showpos);
}

hstr << "\nGaya Ujung Batang\n";
hstr << "Batang   AM1           AM2           AM3           ";
hstr << "AM4           AM5           AM6\n";
hstr << "           AM7           AM8           AM9           ";
hstr << "AM10           AM11           AM12\n";

for(int ih=1; ih<=M; ih++)
{
    periksa_batang(ih);
    isi_matrik_kekakuan(ih);
    indeks_batang(ih);
    for(int jh=1; jh<=MD; jh++)
    {
        AMD[jh]=0.;
        for(int kh=1; kh<=MD; kh++)
        {
            AMD[jh]=AMD[jh]+SMRT[jh][kh]*DJ[IM[kh]];
        }
        AM[ih][jh]=AML[jh][ih]+AMD[jh];
    }
    hstr << setiosflags(ios::left);
    hstr << setw(8) << ih;
    hstr << setw(12) << AM[ih][1]
        << setw(12) << AM[ih][2]
        << setw(12) << AM[ih][3]
        << setw(12) << AM[ih][4]
        << setw(12) << AM[ih][5]
        << setw(12) << AM[ih][6] << '\n';
    hstr << "           "
        << setw(12) << AM[ih][7]
        << setw(12) << AM[ih][8]
        << setw(12) << AM[ih][9]
        << setw(12) << AM[ih][10]
        << setw(12) << AM[ih][11]
        << setw(12) << AM[ih][12] << '\n';
}

hstr << "\nReaksi Tummpuan\n";
hstr << "Titik   AR1           AR2           AR3";
hstr << "           AR4           AR5           AR6 \n";

for(int ih=1; ih<=NJ; ih++)
{
    int J1=(6*ih-5),J2=(6*ih-4),J3=(6*ih-3),
        J4=(6*ih-2),J5=(6*ih-1),J6=(6*ih);
    int N1=JRL[J1]+JRL[J2]+JRL[J3]+JRL[J4]+JRL[J5]+JRL[J6];
    if(N1!=0)
    {
        hstr << setiosflags(ios::left);
        hstr << setw(8) << ih;

        hstr << setw(12) << AR[J1]
            << setw(12) << AR[J2]
            << setw(12) << AR[J3];
        hstr << setw(12) << AR[J4]
            << setw(12) << AR[J5]
```

LAMPIRAN 22

```
        << setw(12) << AR[J6] << '\n';
    }
}
hstr.close();

////////////////////
// Cetak Hasil Optimasi //
////////////////////
hopt << "Hasil Optimasi Beton Bertulang "
      << "Pada Struktur Portal Ruang\n";
hopt << "Metoda Optimasi : Flexible Polyhedron\n";
hopt << "Jumlah Variabel Desain : " << JVD << endl;
hopt << "Jumlah Struktur Desain : " << JSTD << endl;
hopt << "Oleh Yohan Naftali 1999\n\n";
hopt << "Nama File Generik : " << fi << endl;
hopt << "Nama Struktur : " << ISN << endl;
hopt << "Jumlah batang : " << M << endl;
hopt << "Jumlah titik kumpul : " << NJ << endl;
hopt << "Modulus elastisitas tarik/tekan : "
      << (E*1.E-6) << " MPa" << endl;
hopt << "Modulus elastisitas geser G : "
      << (G*1.E-6) << " MPa" << endl;
hopt << "Kuat desak Karakteristik Beton : " << FC << " MPa\n";
hopt << "Kuat tarik baja tulangan lentur : " << FY << " MPa\n";
hopt << "Kuat tarik baja tulangan sengkang : " << FYS << " MPa\n";

////////////////////
// Cetak Header Kendala //
////////////////////
hkdl << "Kendala Pada Struktur\n";
hkdl << "Oleh Yohan Naftali 1999\n\n";
hkdl << "Nama File Generik : " << fi << endl;
hkdl << "Nama Struktur : " << ISN << endl;

no_struktur=(JSTD-1);
for(int oio=0;oio<jum_balok;oio++)
{
    //////////////////////
    // Memeriksa Kemiringan Batang //
    //////////////////////
    periksa_batang(no_balok[oio]);

    //////////////////////
    // Pengisian Elemen Balok //
    //////////////////////
    isi_elemen_balok(oio);

    hopt << "-----\n";
    hopt << "Balok " << (oio+1) << " " << B << " x " << H << endl;
    hopt << "Nomor Batang : " << no_balok[oio] << endl;
    hopt << "Panjang Bentang : " << L << " m" << endl;
    hopt << "Beban Total : " << W[no_balok[oio]] << " N/m\n";
    hopt << "Berat Sendiri : " << W_Balok[no_balok[oio]] << " N/m\n";
    hopt << "Momen kiri : "
          << ((MTUM_KI[no_balok[oio]]/(0.8)) << " Nmm" << endl;
    hopt << "Momen tengah : "
          << ((MLAP[no_balok[oio]]/(0.8)) << " Nmm" << endl;
    hopt << "Momen kanan : "
          << ((MTUM_KA[no_balok[oio]]/(0.8)) << " Nmm" << endl;
    hopt << "Geser kiri : "
          << (GESER_KI[no_balok[oio]]/0.8) << " N" << endl;
    hopt << "Geser kanan : "
          << (GESER_KA[no_balok[oio]]/0.8) << " N" << endl;

    //////////////////////
    // Daerah Lapangan //
    //////////////////////
    elemen_lapangan(oio);
    balok_lapangan;

    lendutan(no_balok[oio]);
```

LAMPIRAN 22

```
//////////
// Cetak Hasil Optimasi //
//////////
hopt << "Lendutan Tengah Bentang : " << LENDUTAN
      << " mm" << endl;
hopt << "Lendutan ijin " << LENDUTAN_IJIN << " mm" << endl;
hopt << "Sengkang " << DIAS << " - " << Jarak_S << endl;
hopt << "Jarak sengkang maksimum : " << Sref << " mm\n";

hopt << "\nDaerah Lapangan\n";
hopt << "Tulangan tarik " << NL1 << " D " << DIA1 << endl;
hopt << "Tulangan desak " << NL2 << " D " << DIA2 << endl;
hopt << "Volume beton : " << volume_beton << " m^3" << endl;
hopt << "Berat Tulangan utama : " << berat_besi << " kg\n";
hopt << "Berat Tulangan geser : " << berat_sengkang << " kg\n";
hopt << "Harga balok daerah lapangan : " << harga << endl;
hopt << "Momen yang membebani : " << MU << " Nmm" << endl;
hopt << "Momen yang dapat ditahan : " << FMU << " Nmm" << endl;
hopt << endl;

//////////
// Cetak Kendala //
//////////
hkdl << "\nBalok " << (oio+1) << endl;
hkdl << "Nomor Batang : " << no_balok[oio] << endl;
hkdl << "Kendala akibat lendutan : "
      << kendala_lendutan << endl;
hkdl << "\nDaerah Lapangan\n";
hkdl << "Kendala rasio penulangan "
      << kendala_rho << endl;
hkdl << "Kendala sengkang : "
      << kendala_sb << endl;
hkdl << "Kendala momen lentur : "
      << kendala_M << endl;

//////////
// Daerah Tumpuan //
//////////
elemen_tumpuan(oio);
balok tumpuan;

//////////
// Cetak Hasil Optimasi //
//////////
hopt << "Daerah Tumpuan\n";
hopt << "Tulangan desak " << NL2 << " D " << DIA2 << endl;
hopt << "Tulangan tarik " << NL1 << " D " << DIA1 << endl;
hopt << "Volume beton : " << volume_beton << " m^3" << endl;
hopt << "Berat Tulangan utama : " << berat_besi << " kg\n";
hopt << "Berat Tulangan geser : " << berat_sengkang << " kg\n";
hopt << "Harga balok pada tumpuan kiri + kanan : " << harga << endl;
hopt << "Momen yang membebani : " << MU << " Nmm" << endl;
hopt << "Momen yang dapat ditahan : " << FMU << " Nmm" << endl;
hopt << endl;

//////////
// Cetak Kendala //
//////////
hkdl << "\nDaerah Tumpuan\n";
hkdl << "Kendala rasio penulangan "
      << kendala_rho << endl;
hkdl << "Kendala sengkang : "
      << kendala_sb << endl;
hkdl << "Kendala momen lentur : "
      << kendala_M << endl;
}
```

LAMPIRAN 22

```
////////////////////////////////////
// Menghitung Kendala Dan Harga Kolom //
////////////////////////////////////
for(int oio=0;oio<jum_kolom;oio++)
{
    ///////////////////////////////////
    // Memeriksa Kemiringan Batang //
    ///////////////////////////////////
    periksa_batang(no_kolom[oio]);

    ///////////////////////////////////
    // Pengisian Elemen Balok //
    ///////////////////////////////////
    isi_element_kolom(oio);

    ///////////////////////////////////
    // Membangkitkan Kolom //
    ///////////////////////////////////
    kolom bangkit;

    ///////////////////////////////////
    // Cetak Hasil Optimasi //
    ///////////////////////////////////
    hopt << "-----\n";
    hopt << "Kolom " << (oio+1) << " "
        << sisi << " x " << sisi << endl;
    hopt << "Nomor Batang : " << no_kolom[oio] << endl;
    hopt << "Tulangan utama "
        << (4*N_DIA-4) << " D " << DIA << endl;
    hopt << "Jarak antar tulangan : " << jarak_antar_tulangan << " mm\n";
    hopt << "Sengkang " << DIAS << " - " << Jarak_S << endl;
    hopt << "Jarak sengkang maksimum : " << Sref << " mm\n";

    hopt << "Volume beton : " << volume_beton << " m^3" << endl;
    hopt << "Berat Tulangan utama : " << berat_besi << " kg\n";
    hopt << "Berat Tulangan geser : " << berat_sengkang << " kg\n";
    hopt << "Harga kolom : " << harga << endl;

    hopt << "Berat Kolom : " << (-P_Kolom[no_kolom[oio]]) << " N\n";
    hopt << "Gaya aksial : "
        << (PK[no_kolom[oio]]/0.8) << " N" << endl;
    hopt << "Gaya aksial yang dapat ditahan = "
        << FPU << " N\n";
    hopt << "Momen arah X : "
        << (MKX[no_kolom[oio]]/0.8) << " Nmm" << endl;
    hopt << "Momen arah Y : "
        << (MKY[no_kolom[oio]]/0.8) << " Nmm" << endl;
    hopt << "Momen yang dapat ditahan : "
        << FMU << " Nmm\n";
    hopt << "Geser pada kolom : "
        << (GK[no_kolom[oio]]/0.8) << " N" << endl;
    hopt << endl;

    ///////////////////////////////////
    // Cetak Kendala //
    ///////////////////////////////////
    hkdl << "\nKolom " << (oio+1) << endl;
    hkdl << "Nomor Batang : " << no_kolom[oio] << endl;
    hkdl << "\nKendala gaya : "
        << kendala_gaya << endl;
    hkdl << "Kendala rasio penulangan : "
        << kendala_r << endl;
    hkdl << "Kendala jarak tulangan : "
        << kendala_tul << endl;
    hkdl << "Kendala sengkang : "
        << kendala_sengkang << endl;
    hkdl << "Kendala kelangsingan kolom : "
        << kendala_kelangsingan << endl;
}
hopt << "\nHarga Beton Rp." << harga_beton << " /m^3\n";
hopt << "Harga Besi Rp." << harga_besi << " /kg\n";
```

LAMPIRAN 22

```
hopt << "Tebal selimut kolom " << selimut_kolom << " mm\n";
hopt << "Tebal selimut balok " << selimut_balok << " mm\n";
hopt << "Faktor finalti : " << finalti << endl;
hopt << "JSTD = (JVD*" << fak_kali << ")+ " << fak_plus << endl;
}
```

LAMPIRAN 23

```
# File ORCISF.MAK
# Konfigurasi pada saat pembuatan executable file
#
.AUTODEPEND

#
# Borland C++ tools
#
IMPLIB = Implib
BCC32 = Bcc32 +BccW32.cfg
BCC32I = Bcc32i +BccW32.cfg
TLINK32 = TLink32
ILINK32 = Ilink32
TLIB = TLib
BRC32 = Brc32
TASM32 = Tasm32
#
# IDE macros
#

#
# Options
#
IDE_LinkFLAGS32 = -LC:\BC5\LIB
LinkerLocalOptsAtC32_orcisfdexe = -Tpe -ap -c
ResLocalOptsAtC32_orcisfdexe =
BLocalOptsAtC32_orcisfdexe =
CompInheritOptsAt_orcisfdexe = -IC:\BC5\INCLUDE -D_RTLDLL;
LinkerInheritOptsAt_orcisfdexe = -x
LinkerOptsAt_orcisfdexe = $(LinkerLocalOptsAtC32_orcisfdexe)
ResOptsAt_orcisfdexe = $(ResLocalOptsAtC32_orcisfdexe)
BOptsAt_orcisfdexe = $(BLocalOptsAtC32_orcisfdexe)

#
# Dependency List
#
Dep_ORCISF = \
    orcisf.exe

ORCISF : BccW32.cfg $(Dep_ORCISF)
    echo MakeNode

Dep_orcisfdexe = \
    orcisf.obj

orcisf.exe : $(Dep_orcisfdexe)
    $(ILINK32) @&&|
    /v $(IDE_LinkFLAGS32) $(LinkerOptsAt_orcisfdexe)
    $(LinkerInheritOptsAt_orcisfdexe) +
    C:\BC5\LIB\c0x32.obj+
    orcisf.obj
    $<,$*
    C:\BC5\LIB\import32.lib+
    C:\BC5\LIB\cw32i.lib

|
orcisf.obj : orcisf.cpp
    $(BCC32) -c @&&|
    $(CompOptsAt_orcisfdexe) $(CompInheritOptsAt_orcisfdexe) -o$@ orcisf.cpp
|

# Compiler configuration file
BccW32.cfg :
    Copy &&|
-w
-R
-v
```

LAMPIRAN 23

```
-WM-  
-vi  
-H  
-H=ORCISF.csm  
-WC  
-O2  
-OS  
-5  
-a8  
| $@
```

LAMPIRAN 24

Input data struktur portal gedung 2 lantai pada gambar 4-1

File : gedung.inp

GEDUNG_2_LANTAI

21	16	16	60
1.96615e+10		1.13054e+10	
17.5	240	240	
96	36		

[Koordinat]

1	0	0	0
2	6	0	0
3	10	0	0
4	16	0	0
5	0	4	0
6	6	4	0
7	10	4	0
8	16	4	0
9	0	8	0
10	6	8	0
11	10	8	0
12	16	8	0
13	0	12	0
14	6	12	0
15	10	12	0
16	16	12	0

[Pengekang]

1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	0	0	1	1	1	0
6	0	0	1	1	1	0
7	0	0	1	1	1	0
8	0	0	1	1	1	0
9	0	0	1	1	1	0
10	0	0	1	1	1	0
11	0	0	1	1	1	0
12	0	0	1	1	1	0
13	0	0	1	1	1	0
14	0	0	1	1	1	0
15	0	0	1	1	1	0
16	0	0	1	1	1	0

[Batang]

1	1	5	0
2	2	6	0
3	3	7	0
4	4	8	0
5	5	9	0
6	6	10	0
7	7	11	0
8	8	12	0
9	9	13	0
10	10	14	0
11	11	15	0
12	12	16	0
13	5	6	0
14	6	7	0
15	7	8	0
16	9	10	0
17	10	11	0
18	11	12	0
19	13	14	0
20	14	15	0
21	15	16	0

LAMPIRAN 25

Input Data Diskrit struktur portal gedung 2 lantai pada gambar 4-1

File : gedung.isd

[LebarBalok]

21

200	210	220	230	240	250	260	270	280	290	300	310	320	330
340	350	360	370	380	390	400							

[TinggiBalok]

41

200	210	220	230	240	250	260	270	280	290	300	310	320	330
340	350	360	370	380	390	400	410	420	430	440	450	460	470
480	490	500	510	520	530	540	550	560	570	580	590	600	

[SisiKolom]

16

350	360	370	380	390	400	410	420	430	440	450	460	470	480
490	500												

File : gedung.idl

[DiameterTulanganUtama]

8

19	20	22	25	28	29	32	36
----	----	----	----	----	----	----	----

File : gedung.ijl

[JumlahTulanganUtama]

3

2	3	4
---	---	---

File : gedung.ids

[DiameterTulanganSengkang]

3

8	12	13
---	----	----

File : gedung.ijs

[JarakAntarSengkang]

23

80	90	100	110	120	130	140	150	160	170
180	190	200	210	220	230	240	250	260	270
280	290	300							

LAMPIRAN 26

Input data pembebanan pada struktur gambar 4-1

File : Gedung.bbn

[BebanBatang]

21							
1	0	0	0	0	0	0	0
		0	0	0	0	0	0
2	0	0	0	0	0	0	0
		0	0	0	0	0	0
3	0	0	0	0	0	0	0
		0	0	0	0	0	0
4	0	0	0	0	0	0	0
		0	0	0	0	0	0
5	0	0	0	0	0	0	0
		0	0	0	0	0	0
6	0	0	0	0	0	0	0
		0	0	0	0	0	0
7	0	0	0	0	0	0	0
		0	0	0	0	0	0
8	0	0	0	0	0	0	0
		0	0	0	0	0	0
9	0	0	0	0	0	0	0
		0	0	0	0	0	0
10	0	0	0	0	0	0	0
		0	0	0	0	0	0
11	0	0	0	0	0	0	0
		0	0	0	0	0	0
12	0	0	0	0	0	0	0
		0	0	0	0	0	0
13	37400	0	112200	0	0	0	112200
		0	112200	0	0	0	-112200
14	31310	0	62620	0	0	0	41746.7
		0	62620	0	0	0	-41746.7
15	37400	0	112200	0	0	0	112200
		0	112200	0	0	0	-112200
16	37400	0	112200	0	0	0	112200
		0	112200	0	0	0	-112200
17	31310	0	62620	0	0	0	41746.7
		0	62620	0	0	0	-41746.7
18	37400	0	112200	0	0	0	112200
		0	112200	0	0	0	-112200
19	14620	0	43860	0	0	0	43860
		0	43860	0	0	0	-43860
20	11470	0	22940	0	0	0	15293.3
		0	22940	0	0	0	-15293.3
21	14620	0	43860	0	0	0	43860
		0	43860	0	0	0	-43860

[BebanTitik]

16						
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	8000	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	8000	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	4000	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0

LAMPIRAN 27

Hasil berupa informasi struktur pada gambar 4-1

File : Gedung.inf

Struktur Portal Ruang GEDUNG_2_LANTAI

Parameter Struktur

Jumlah batang : 21

DOF : 36

Jumlah joint : 16

Jumlah pengekang tumpuan : 60

Jumlah titik kumpul yang dikekang : 16

Modulus Elastisitas aksial : 1.96615×10^{10} N/m²

Modulus Geser : 1.13054×10^{10} N/m²

Properti Elemen Material

Kuat desak beton karakteristik : 17.5 MPa

Kuat tarik baja tulangan : 240 MPa

Kuat tarik tulangan sengkang : 240 MPa

Koordinat Titik Kumpul (m)

Titik	X	Y	Z
1	0	0	0
2	6	0	0
3	10	0	0
4	16	0	0
5	0	4	0
6	6	4	0
7	10	4	0
8	16	4	0
9	0	8	0
10	6	8	0
11	10	8	0
12	16	8	0
13	0	12	0
14	6	12	0
15	10	12	0
16	16	12	0

Informasi Batang

Batang	JJ	JK	IA
1	1	5	0
2	2	6	0
3	3	7	0
4	4	8	0
5	5	9	0
6	6	10	0
7	7	11	0
8	8	12	0
9	9	13	0
10	10	14	0
11	11	15	0
12	12	16	0
13	5	6	0
14	6	7	0
15	7	8	0
16	9	10	0
17	10	11	0
18	11	12	0
19	13	14	0
20	14	15	0
21	15	16	0

Pengekang Titik Kumpul

Titik	JR1	JR2	JR3	JR4	JR5	JR6
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	0	0	1	1	1	0
6	0	0	1	1	1	0

LAMPIRAN 27

7	0	0	1	1	1	0
8	0	0	1	1	1	0
9	0	0	1	1	1	0
10	0	0	1	1	1	0
11	0	0	1	1	1	0
12	0	0	1	1	1	0
13	0	0	1	1	1	0
14	0	0	1	1	1	0
15	0	0	1	1	1	0
16	0	0	1	1	1	0

Gaya di Batang	Ujung Batang	Terkekang	Akibat	Beban (Nm)		
	AML1	AML2	AML3	AML4	AML5	AML6
	AML7	AML8	AML9	AML10	AML11	AML12
1	0	0	0	0	0	0
	0	0	0	0	0	0
2	0	0	0	0	0	0
	0	0	0	0	0	0
3	0	0	0	0	0	0
	0	0	0	0	0	0
4	0	0	0	0	0	0
	0	0	0	0	0	0
5	0	0	0	0	0	0
	0	0	0	0	0	0
6	0	0	0	0	0	0
	0	0	0	0	0	0
7	0	0	0	0	0	0
	0	0	0	0	0	0
8	0	0	0	0	0	0
	0	0	0	0	0	0
9	0	0	0	0	0	0
	0	0	0	0	0	0
10	0	0	0	0	0	0
	0	0	0	0	0	0
11	0	0	0	0	0	0
	0	0	0	0	0	0
12	0	0	0	0	0	0
	0	0	0	0	0	0
13	0	125743	0	0	0	125743
	0	125743	0	0	0	-125743
14	0	70856.8	0	0	0	47237.9
	0	70856.8	0	0	0	-47237.9
15	0	125419	0	0	0	125419
	0	125419	0	0	0	-125419
16	0	124476	0	0	0	124476
	0	124476	0	0	0	-124476
17	0	70856.8	0	0	0	47237.9
	0	70856.8	0	0	0	-47237.9
18	0	125419	0	0	0	125419
	0	125419	0	0	0	-125419
19	0	55502.4	0	0	0	55502.4
	0	55502.4	0	0	0	-55502.4
20	0	30701.6	0	0	0	20467.7
	0	30701.6	0	0	0	-20467.7
21	0	55977.6	0	0	0	55977.6
	0	55977.6	0	0	0	-55977.6

Beban Titik (N)						
Titik	Arah 1	Arah 2	Arah 3	Arah 4	Arah 5	Arah 6
1	0	-16137.6	0	0	0	0
2	0	-16137.6	0	0	0	0
3	0	-15360	0	0	0	0
4	0	-17750.4	0	0	0	0
5	8000	-18585.6	0	0	0	0
6	0	-18585.6	0	0	0	0
7	0	-16934.4	0	0	0	0
8	0	-16934.4	0	0	0	0
9	8000	-16137.6	0	0	0	0
10	0	-19440	0	0	0	0
11	0	-20313.6	0	0	0	0
12	0	-18585.6	0	0	0	0

LAMPIRAN 27

13	4000	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0

LAMPIRAN 28

Hasil tentang riwayat proses optimasi struktur pada gambar 4-1

File : Gedung.his

Penyusutan berturut pertama pada generasi : 274 Fitness terbaik : 225.645
 Penyusutan berturut berakhir setelah 2 kali
Penyusutan berturut pertama pada generasi : 837 Fitness terbaik : 562.289
Penyusutan berturut pertama pada generasi : 1008 Fitness terbaik : 563.087
Konvergen Pada Generasi : 1178 karena jumlah penyusutan melebihi batas
Fitness = 563.087
Harga = 1.77592e+07
Kendala = 0
Waktu optimasi : 177 detik

LAMPIRAN 29

Hasil akhir struktur terbaik untuk kasus portal pada gambar 4-1

File : Gedung.opt

Nama Struktur : GEDUNG_2_LANTAI
Hasil Optimasi Beton Bertulang Pada Struktur Portal Ruang
Metoda Optimasi : Flexible Polyhedron
Jumlah Variabel Desain : 168
Jumlah Struktur Desain : 171
Oleh Yohan Naftali 1999

Nama File Generik : data01/gedung
Nama Struktur : GEDUNG_2_LANTAI
Jumlah batang : 21
Jumlah titik kumpul : 16
Modulus elastisitas tarik/tekan : 19661.5 MPa
Modulus elastisitas geser G : 11305.4 MPa
Kuat desak Karakteristik Beton : 17.5 MPa
Kuat tarik baja tulangan lentur : 240 MPa
Kuat tarik baja tulangan sengkang : 240 MPa

Balok 1 330 x 570
Nomor Batang : 13
Panjang Bentang : 6 m
Beban Total : 41914.4 N/m
Berat Sendiri : 4514.4 N/m
Momen kiri : -106424 Nmm
Momen tengah : 129345 Nmm
Momen kanan : -158332 Nmm
Geser kiri : 148528 N
Geser kanan : -165830 N
Lendutan Tengah Bentang : 19.305 mm
Lendutan ijin 33.3333 mm
Sengkang 12 - 130
Jarak sengkang maksimum : 152.099 mm

Daerah Lapangan
Tulangan tarik 3 D 25
Tulangan desak 2 D 22
Volume beton : 0.5643 m³
Berat Tulangan utama : 52.5578 kg
Berat Tulangan geser : 46.8794 kg
Harga balok daerah lapangan : 638261
Momen yang membebani : 129345 Nmm
Momen yang dapat ditahan : 133420 Nmm

Daerah Tumpuan
Tulangan desak 2 D 25
Tulangan tarik 3 D 28
Volume beton : 0.5643 m³
Berat Tulangan utama : 66.5893 kg
Berat Tulangan geser : 46.8794 kg
Harga balok pada tumpuan kiri + kanan : 708419
Momen yang membebani : 158332 Nmm
Momen yang dapat ditahan : 164707 Nmm

Balok 2 330 x 520
Nomor Batang : 14
Panjang Bentang : 4 m
Beban Total : 35428.4 N/m
Berat Sendiri : 4118.4 N/m
Momen kiri : -68181.4 Nmm
Momen tengah : 20389.6 Nmm
Momen kanan : -91455.9 Nmm
Geser kiri : 82752.4 N
Geser kanan : -94389.6 N
Lendutan Tengah Bentang : 0.0940298 mm
Lendutan ijin 22.2222 mm
Sengkang 12 - 150

LAMPIRAN 29

Jarak sengkang maksimum : 228.75 mm

Daerah Lapangan

Tulangan tarik 3 D 25
Tulangan desak 2 D 20
Volume beton : 0.3432 m³
Berat Tulangan utama : 32.968 kg
Berat Tulangan geser : 24.37 kg
Harga balok daerah lapangan : 372490
Momen yang membebani : 20389.6 Nmm
Momen yang dapat ditahan : 119295 Nmm

Daerah Tumpuan

Tulangan desak 2 D 22
Tulangan tarik 3 D 22
Volume beton : 0.3432 m³
Berat Tulangan utama : 29.8253 kg
Berat Tulangan geser : 24.37 kg
Harga balok pada tumpuan kiri + kanan : 356776
Momen yang membebani : 91455.9 Nmm
Momen yang dapat ditahan : 94289.1 Nmm

Balok 3 340 x 550
Nomor Batang : 15
Panjang Bentang : 6 m
Beban Total : 41806.4 N/m
Berat Sendiri : 4406.4 N/m
Momen kiri : -135899 Nmm
Momen tengah : 99262 Nmm
Momen kanan : -135390 Nmm
Geser kiri : 156859 N
Geser kanan : -156689 N
Lendutan Tengah Bentang : 15.8906 mm
Lendutan ijin 33.3333 mm
Sengkang 12 - 140
Jarak sengkang maksimum : 165.51 mm

Daerah Lapangan

Tulangan tarik 3 D 22
Tulangan desak 2 D 22
Volume beton : 0.561 m³
Berat Tulangan utama : 44.7379 kg
Berat Tulangan geser : 42.3425 kg
Harga balok daerah lapangan : 575652
Momen yang membebani : 99262 Nmm
Momen yang dapat ditahan : 101053 Nmm

Daerah Tumpuan

Tulangan desak 2 D 22
Tulangan tarik 3 D 28
Volume beton : 0.561 m³
Berat Tulangan utama : 61.376 kg
Berat Tulangan geser : 42.3425 kg
Harga balok pada tumpuan kiri + kanan : 658842
Momen yang membebani : 135899 Nmm
Momen yang dapat ditahan : 157836 Nmm

Balok 4 310 x 550
Nomor Batang : 16
Panjang Bentang : 6 m
Beban Total : 41492 N/m
Berat Sendiri : 4092 N/m
Momen kiri : -120596 Nmm
Momen tengah : 112796 Nmm
Momen kanan : -156278 Nmm
Geser kiri : 149648 N
Geser kanan : -161542 N
Lendutan Tengah Bentang : 17.5832 mm
Lendutan ijin 33.3333 mm

LAMPIRAN 29

Senggang 12 - 130
Jarak senggang maksimum : 136.968 mm

Daerah Lapangan
Tulangan tarik 3 D 25
Tulangan desak 2 D 28
Volume beton : 0.5115 m³
Berat Tulangan utama : 63.6499 kg
Berat Tulangan geser : 44.4754 kg
Harga balok daerah lapangan : 668501
Momen yang membebani : 112796 Nmm
Momen yang dapat ditahan : 127293 Nmm

Daerah Tumpuan
Tulangan desak 2 D 28
Tulangan tarik 3 D 28
Volume beton : 0.5115 m³
Berat Tulangan utama : 72.4681 kg
Berat Tulangan geser : 44.4754 kg
Harga balok pada tumpuan kiri + kanan : 712592
Momen yang membebani : 156278 Nmm
Momen yang dapat ditahan : 157138 Nmm

Balok 5 330 x 520
Nomor Batang : 17
Panjang Bentang : 4 m
Beban Total : 35428.4 N/m
Berat Sendiri : 4118.4 N/m
Momen kiri : -68118.1 Nmm
Momen tengah : 20452.9 Nmm
Momen kanan : -79987.3 Nmm
Geser kiri : 85603.7 N
Geser kanan : -91538.3 N
Lendutan Tengah Bentang : 0.0724975 mm
Lendutan ijin 22.2222 mm
Senggang 12 - 170
Jarak senggang maksimum : 230 mm

Daerah Lapangan
Tulangan tarik 3 D 20
Tulangan desak 2 D 25
Volume beton : 0.3432 m³
Berat Tulangan utama : 30.195 kg
Berat Tulangan geser : 21.3912 kg
Harga balok daerah lapangan : 343731
Momen yang membebani : 20452.9 Nmm
Momen yang dapat ditahan : 84694.9 Nmm

Daerah Tumpuan
Tulangan desak 2 D 25
Tulangan tarik 3 D 25
Volume beton : 0.3432 m³
Berat Tulangan utama : 38.5141 kg
Berat Tulangan geser : 21.3912 kg
Harga balok pada tumpuan kiri + kanan : 385326
Momen yang membebani : 79987.3 Nmm
Momen yang dapat ditahan : 119278 Nmm

Balok 6 340 x 540
Nomor Batang : 18
Panjang Bentang : 6 m
Beban Total : 41806.4 N/m
Berat Sendiri : 4406.4 N/m
Momen kiri : -142863 Nmm
Momen tengah : 92298.2 Nmm
Momen kanan : -135999 Nmm
Geser kiri : 157918 N
Geser kanan : -155630 N
Lendutan Tengah Bentang : 14.5626 mm

LAMPIRAN 29

Lendutan ijin 33.3333 mm
Sengkang 12 - 140
Jarak sengkang maksimum : 154.982 mm

Daerah Lapangan
Tulangan tarik 3 D 22
Tulangan desak 2 D 25
Volume beton : 0.5508 m³
Berat Tulangan utama : 49.9512 kg
Berat Tulangan geser : 41.5996 kg
Harga balok daerah lapangan : 595454
Momen yang membebani : 92298.2 Nmm
Momen yang dapat ditahan : 98860.4 Nmm

Daerah Tumpuan
Tulangan desak 2 D 20
Tulangan tarik 3 D 28
Volume beton : 0.5508 m³
Berat Tulangan utama : 58.2702 kg
Berat Tulangan geser : 41.5996 kg
Harga balok pada tumpuan kiri + kanan : 637049
Momen yang membebani : 142863 Nmm
Momen yang dapat ditahan : 154251 Nmm

Balok 7 330 x 490
Nomor Batang : 19
Panjang Bentang : 6 m
Beban Total : 18500.8 N/m
Berat Sendiri : 3880.8 N/m
Momen kiri : -49540.2 Nmm
Momen tengah : 54526.8 Nmm
Momen kanan : -68648.5 Nmm
Geser kiri : 66193.3 N
Geser kanan : -72562.7 N
Lendutan Tengah Bentang : 8.44502 mm
Lendutan ijin 33.3333 mm
Sengkang 12 - 160
Jarak sengkang maksimum : 214.5 mm

Daerah Lapangan
Tulangan tarik 3 D 22
Tulangan desak 2 D 19
Volume beton : 0.4851 m³
Berat Tulangan utama : 40.1902 kg
Berat Tulangan geser : 32.7127 kg
Harga balok daerah lapangan : 485789
Momen yang membebani : 54526.8 Nmm
Momen yang dapat ditahan : 87732.1 Nmm

Daerah Tumpuan
Tulangan desak 2 D 25
Tulangan tarik 3 D 28
Volume beton : 0.4851 m³
Berat Tulangan utama : 66.5893 kg
Berat Tulangan geser : 32.7127 kg
Harga balok pada tumpuan kiri + kanan : 617785
Momen yang membebani : 68648.5 Nmm
Momen yang dapat ditahan : 136348 Nmm

Balok 8 330 x 490
Nomor Batang : 20
Panjang Bentang : 4 m
Beban Total : 15350.8 N/m
Berat Sendiri : 3880.8 N/m
Momen kiri : -35061.1 Nmm
Momen tengah : 3315.94 Nmm
Momen kanan : -32611.1 Nmm
Geser kiri : 38989.5 N
Geser kanan : -37764.5 N

LAMPIRAN 29

Lendutan Tengah Bentang : 0.000110305 mm
Lendutan ijin 22.2222 mm
Sengkang 12 - 180
Jarak sengkang maksimum : 213.75 mm

Daerah Lapangan
Tulangan tarik 3 D 25
Tulangan desak 2 D 28
Volume beton : 0.3234 m³
Berat Tulangan utama : 42.4333 kg
Berat Tulangan geser : 19.0202 kg
Harga balok daerah lapangan : 388117
Momen yang membebani : 3315.94 Nmm
Momen yang dapat ditahan : 110786 Nmm

Daerah Tumpuan
Tulangan desak 2 D 22
Tulangan tarik 3 D 22
Volume beton : 0.3234 m³
Berat Tulangan utama : 29.8253 kg
Berat Tulangan geser : 19.0202 kg
Harga balok pada tumpuan kiri + kanan : 325077
Momen yang membebani : 35061.1 Nmm
Momen yang dapat ditahan : 87723.8 Nmm

Balok 9 330 x 510
Nomor Batang : 21
Panjang Bentang : 6 m
Beban Total : 18659.2 N/m
Berat Sendiri : 4039.2 N/m
Momen kiri : -63125.1 Nmm
Momen tengah : 41832.9 Nmm
Momen kanan : -58046.8 Nmm
Geser kiri : 70818.4 N
Geser kanan : -69125.6 N
Lendutan Tengah Bentang : 3.18938 mm
Lendutan ijin 33.3333 mm
Sengkang 12 - 170
Jarak sengkang maksimum : 223.75 mm

Daerah Lapangan
Tulangan tarik 3 D 25
Tulangan desak 2 D 19
Volume beton : 0.5049 m³
Berat Tulangan utama : 48.0101 kg
Berat Tulangan geser : 31.9529 kg
Harga balok daerah lapangan : 526040
Momen yang membebani : 41832.9 Nmm
Momen yang dapat ditahan : 116470 Nmm

Daerah Tumpuan
Tulangan desak 2 D 22
Tulangan tarik 3 D 20
Volume beton : 0.5049 m³
Berat Tulangan utama : 40.0793 kg
Berat Tulangan geser : 31.9529 kg
Harga balok pada tumpuan kiri + kanan : 486386
Momen yang membebani : 63125.1 Nmm
Momen yang dapat ditahan : 82326.5 Nmm

Kolom 1 410 x 410
Nomor Batang : 1
Tulangan utama 8 D 25
Jarak antar tulangan : 117.5 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 400 mm
Volume beton : 0.6724 m³
Berat Tulangan utama : 123.245 kg
Berat Tulangan geser : 13.5708 kg

LAMPIRAN 29

Harga kolom : 852179
Berat Kolom : 16137.6 N
Gaya aksial : 407773 N
Gaya aksial yang dapat ditahan = $1.96661e+06$ N
Momen arah X : 0 Nmm
Momen arah Y : 34729.3 Nmm
Momen yang dapat ditahan : 167472 Nmm
Geser pada kolom : 11275.9 N

Kolom 2 410 x 410
Nomor Batang : 2
Tulangan utama 8 D 22
Jarak antar tulangan : 122 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 352 mm
Volume beton : 0.6724 m^3
Berat Tulangan utama : 95.4409 kg
Berat Tulangan geser : 13.5708 kg
Harga kolom : 713158
Berat Kolom : 16137.6 N
Gaya aksial : 654813 N
Gaya aksial yang dapat ditahan = $2.25132e+06$ N
Momen arah X : 0 Nmm
Momen arah Y : 34452.3 Nmm
Momen yang dapat ditahan : 118413 Nmm
Geser pada kolom : 14714.3 N

Kolom 3 400 x 400
Nomor Batang : 3
Tulangan utama 8 D 22
Jarak antar tulangan : 117 mm
Sengkang 8 - 300
Jarak sengkang maksimum : 352 mm
Volume beton : 0.64 m^3
Berat Tulangan utama : 95.4409 kg
Berat Tulangan geser : 5.83688 kg
Harga kolom : 666389
Berat Kolom : 15360 N
Gaya aksial : 655848 N
Gaya aksial yang dapat ditahan = $2.55049e+06$ N
Momen arah X : 0 Nmm
Momen arah Y : 14606.9 Nmm
Momen yang dapat ditahan : 56796.8 Nmm
Geser pada kolom : 3849.15 N

Kolom 4 430 x 430
Nomor Batang : 4
Tulangan utama 8 D 19
Jarak antar tulangan : 136.5 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 304 mm
Volume beton : 0.7396 m^3
Berat Tulangan utama : 71.1863 kg
Berat Tulangan geser : 14.4463 kg
Harga kolom : 613063
Berat Kolom : 17750.4 N
Gaya aksial : 425845 N
Gaya aksial yang dapat ditahan = $1.43579e+06$ N
Momen arah X : 0 Nmm
Momen arah Y : 61832.4 Nmm
Momen yang dapat ditahan : 208444 Nmm
Geser pada kolom : 25409.6 N

Kolom 5 440 x 440
Nomor Batang : 5
Tulangan utama 8 D 20
Jarak antar tulangan : 140 mm

LAMPIRAN 29

Sengkang 12 - 300
Jarak sengkang maksimum : 320 mm
Volume beton : 0.7744 m³
Berat Tulangan utama : 78.8768 kg
Berat Tulangan geser : 14.8841 kg
Harga kolom : 662404
Berat Kolom : 18585.6 N
Gaya aksial : 236013 N
Gaya aksial yang dapat ditahan = 639545 N
Momen arah X : 0 Nmm
Momen arah Y : 66895.8 Nmm
Momen yang dapat ditahan : 181255 Nmm
Geser pada kolom : 34647.6 N

Kolom 6 440 x 440
Nomor Batang : 6
Tulangan utama 8 D 19
Jarak antar tulangan : 141.5 mm
Sengkang 8 - 300
Jarak sengkang maksimum : 304 mm
Volume beton : 0.7744 m³
Berat Tulangan utama : 71.1863 kg
Berat Tulangan geser : 6.61513 kg
Harga kolom : 582607
Berat Kolom : 18585.6 N
Gaya aksial : 382998 N
Gaya aksial yang dapat ditahan = 1.5848e+06 N
Momen arah X : 0 Nmm
Momen arah Y : 52271.2 Nmm
Momen yang dapat ditahan : 216262 Nmm
Geser pada kolom : 26992.4 N

Kolom 7 420 x 420
Nomor Batang : 7
Tulangan utama 8 D 19
Jarak antar tulangan : 131.5 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 304 mm
Volume beton : 0.7056 m³
Berat Tulangan utama : 71.1863 kg
Berat Tulangan geser : 14.0085 kg
Harga kolom : 602374
Berat Kolom : 16934.4 N
Gaya aksial : 383431 N
Gaya aksial yang dapat ditahan = 2.1089e+06 N
Momen arah X : 0 Nmm
Momen arah Y : 24947.9 Nmm
Momen yang dapat ditahan : 137182 Nmm
Geser pada kolom : 13696 N

Kolom 8 420 x 420
Nomor Batang : 8
Tulangan utama 8 D 22
Jarak antar tulangan : 127 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 352 mm
Volume beton : 0.7056 m³
Berat Tulangan utama : 95.4409 kg
Berat Tulangan geser : 14.0085 kg
Harga kolom : 723647
Berat Kolom : 16934.4 N
Gaya aksial : 247988 N
Gaya aksial yang dapat ditahan = 637102 N
Momen arah X : 0 Nmm
Momen arah Y : 71843 Nmm
Momen yang dapat ditahan : 182109 Nmm
Geser pada kolom : 36350.1 N

LAMPIRAN 29

Kolom 9 410 x 410
Nomor Batang : 9
Tulangan utama 8 D 20
Jarak antar tulangan : 125 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 320 mm
Volume beton : 0.6724 m^3
Berat Tulangan utama : 78.8768 kg
Berat Tulangan geser : 13.5708 kg
Harga kolom : 630338
Berat Kolom : 16137.6 N
Gaya aksial : 66193.3 N
Gaya aksial yang dapat ditahan = 126633 N
Momen arah X : 0 Nmm
Momen arah Y : 49540.2 Nmm
Momen yang dapat ditahan : 94774 Nmm
Geser pada kolom : 25810.2 N

Kolom 10 450 x 450
Nomor Batang : 10
Tulangan utama 8 D 19
Jarak antar tulangan : 146.5 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 304 mm
Volume beton : 0.81 m^3
Berat Tulangan utama : 71.1863 kg
Berat Tulangan geser : 15.3218 kg
Harga kolom : 635041
Berat Kolom : 19440 N
Gaya aksial : 111552 N
Gaya aksial yang dapat ditahan = 567435 N
Momen arah X : 0 Nmm
Momen arah Y : 33587.4 Nmm
Momen yang dapat ditahan : 170827 Nmm
Geser pada kolom : 17369.1 N

Kolom 11 460 x 460
Nomor Batang : 11
Tulangan utama 8 D 28
Jarak antar tulangan : 138 mm
Sengkang 8 - 300
Jarak sengkang maksimum : 384 mm
Volume beton : 0.8464 m^3
Berat Tulangan utama : 154.599 kg
Berat Tulangan geser : 7.00426 kg
Harga kolom : $1.01961\text{e}+06$
Berat Kolom : 20313.6 N
Gaya aksial : 108583 N
Gaya aksial yang dapat ditahan = $1.09943\text{e}+06$ N
Momen arah X : 0 Nmm
Momen arah Y : 30514 Nmm
Momen yang dapat ditahan : 308947 Nmm
Geser pada kolom : 17110.4 N

Kolom 12 440 x 440
Nomor Batang : 12
Tulangan utama 4 D 25
Jarak antar tulangan : 290 mm
Sengkang 12 - 300
Jarak sengkang maksimum : 400 mm
Volume beton : 0.7744 m^3
Berat Tulangan utama : 61.6225 kg
Berat Tulangan geser : 14.8841 kg
Harga kolom : 576133
Berat Kolom : 18585.6 N
Gaya aksial : 69125.6 N
Gaya aksial yang dapat ditahan = 127125 N

LAMPIRAN 29

Momen arah X : 0 Nmm
Momen arah Y : 58046.8 Nmm
Momen yang dapat ditahan : 106750 Nmm
Geser pada kolom : 30550.7 N

Harga Beton Rp.250000 /m³
Harga Besi Rp.5000 /kg
Tebal selimut kolom 50 mm
Tebal selimut balok 50 mm
Faktor finalti : 1e+10
JSTD = (JVD*1)+3: 177 detik

LAMPIRAN 30

Hasil perhitungan gaya batang pada kasus portal pada gambar 4-1

File : Gedung.str

Hasil Analisa Struktur Dengan Metoda Kekakuan
Dikembangkan dari Weaver & Gere
Oleh Yohan Naftali 1999

Nama File Generik : data01/gedung
Jumlah batang : 21
Jumlah titik kumpul : 16
Modulus elastisitas tarik/tekan : 19661.5 MPa
Modulus elastisitas geser G : 11305.4 MPa

Perpindahan Titik Kumpul

Titik	DJ1	DJ2	DJ3	DJ4	DJ5	DJ6
1	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
2	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
3	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
4	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
5	+0.000644	-0.000395	+0.000000	+0.000000	+0.000000	-0.000842
6	+0.000662	-0.000634	+0.000000	+0.000000	+0.000000	+0.000347
7	+0.000663	-0.000667	+0.000000	+0.000000	+0.000000	-0.000527
8	+0.000677	-0.000375	+0.000000	+0.000000	+0.000000	+0.000629
9	+0.001354	-0.000593	+0.000000	+0.000000	+0.000000	-0.000717
10	+0.001327	-0.000956	+0.000000	+0.000000	+0.000000	+0.000258
11	+0.001318	-0.001021	+0.000000	+0.000000	+0.000000	-0.000374
12	+0.001310	-0.000604	+0.000000	+0.000000	+0.000000	+0.000575
13	+0.001554	-0.000657	+0.000000	+0.000000	+0.000000	-0.000573
14	+0.001508	-0.001046	+0.000000	+0.000000	+0.000000	+0.000203
15	+0.001494	-0.001104	+0.000000	+0.000000	+0.000000	-0.000212
16	+0.001450	-0.000662	+0.000000	+0.000000	+0.000000	+0.000416

Gaya Ujung Batang

Batang	AM1 AM7	AM2 AM8	AM3 AM9	AM4 AM10	AM5 AM11	AM6 AM12
1	326218 -326218	-9020.7 9020.7	0 0	0 0	0 0	-8299.37 -27783.4
2	523850 -523850	11771.4 -11771.4	0 0	0 0	0 0	19523.9 27561.8
3	524678 -524678	-3079.32 3079.32	0 0	0 0	0 0	-631.738 -11685.5
4	340676 -340676	20327.7 -20327.7	0 0	0 0	0 0	31844.9 49465.9
5	188811 -188811	-27718 27718	0 0	0 0	0 0	-57355.5 -53516.6
6	306398 -306398	21593.9 -21593.9	0 0	0 0	0 0	44558.7 41816.9
7	306745 -306745	-10956.8 10956.8	0 0	0 0	0 0	-23868.9 -19958.3
8	198390 -198390	29080.1 -29080.1	0 0	0 0	0 0	58845.8 57474.4
9	52954.6 -52954.6	-20648.2 20648.2	0 0	0 0	0 0	-42960.5 -39632.2
10	89241.7 -89241.7	13895.3 -13895.3	0 0	0 0	0 0	28711.3 26869.9
11	86866.2 -86866.2	-13688.3 13688.3	0 0	0 0	0 0	-30342.1 -24411.2
12	55300.5 -55300.5	24440.6 -24440.6	0 0	0 0	0 0	51324.9 46437.5
13	-10697.3 10697.3	118822 132664	0 0	0 0	0 0	85139 -126666
14	-874.811 874.811	66201.9 75511.7	0 0	0 0	0 0	54545.1 -73164.7
15	-8752.38 8752.38	125487 125351	0 0	0 0	0 0	108719 -108312
16	15069.5 -15069.5	119718 129234	0 0	0 0	0 0	96477.1 -125023
17	7370.91 -7370.91	68483 73230.6	0 0	0 0	0 0	54494.5 -63989.8
18	4639.57	126334	0	0	0	114290

LAMPIRAN 30

	-4639.57	124504	0	0	0	-108799
19	24648	52954.6	0	0	0	39632.2
	-24648	58050.2	0	0	0	-54918.8
20	10752.2	31191.6	0	0	0	28048.8
	-10752.2	30211.6	0	0	0	-26088.9
21	24440.5	56654.7	0	0	0	50500.1
	-24440.5	55300.5	0	0	0	-46437.5

Reaksi Tumpuan

Titik	AR1	AR2	AR3	AR4	AR5	AR6
1	9020.7	342356	0	0	0	-8299.37
2	-11771.4	539988	0	0	0	19523.9
3	3079.32	540038	0	0	0	-631.738
4	-20327.7	358426	0	0	0	31844.9
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0

LAMPIRAN 31

Laporan kendala yang terdapat pada struktur portal pada gambar 4-1

File : Gedung.kdl

Kendala Pada Struktur

Nama File Generik : data01/gedung
Nama Struktur : GEDUNG_2_LANTAI

Balok 1 Nomor Batang : 13
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 2 Nomor Batang : 14
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 3 Nomor Batang : 15
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 4 Nomor Batang : 16
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 5 Nomor Batang : 17
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 6 Nomor Batang : 18
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 7 Nomor Batang : 19
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 8 Nomor Batang : 20
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

Balok 9 Nomor Batang : 21
Kendala akibat lendutan : 0
Daerah Lapangan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0
Daerah Tumpuan
Kendala rasio penulangan 0 Kendala sengkang : 0 Kendala momen lentur : 0

LAMPIRAN 31

Kolom 1 Nomor Batang : 1
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 2 Nomor Batang : 2
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 3 Nomor Batang : 3
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 4 Nomor Batang : 4
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 5 Nomor Batang : 5
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 6 Nomor Batang : 6
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 7 Nomor Batang : 7
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 8 Nomor Batang : 8
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 9 Nomor Batang : 9
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 10 Nomor Batang : 10
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 11 Nomor Batang : 11
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

Kolom 12 Nomor Batang : 12
Kendala gaya : 0 Kendala rasio penulangan : 0 Kendala jarak tulangan : 0
Kendala sengkang : 0 Kendala kelangsingan kolom : 0

LAMPIRAN 32

Input data struktur portal ruang pada gambar 4-3

File : aplikasi.inp

Aplikasi_Portal_Beton_3D

8	8	4	24
2.5743e+10		1.48022e+10	
30	400	240	
48	24		

[Koordinat]

1	0	0	0
2	6	0	0
3	0	0	6
4	6	0	6
5	0	5	0
6	6	5	0
7	0	5	6
8	6	5	6

[Pengekang]

1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1

[InformasiBatang]

1	1	5	0
2	2	6	0
3	3	7	0
4	4	8	0
5	5	6	0
6	7	8	0
7	5	7	0
8	6	8	0

LAMPIRAN 33

Input data diskrit struktur portal ruang pada gambar 4-3

File : aplikasi.isd

[LebarBalok]

5
200 250 300 350 400

[TinggiBalok]

11
250 300 350 400 450 500 550 600 650 700 750

[SisiKolom]

5
400 450 500 550 600

File : aplikasi.idl

[DiameterTulanganUtama]

8
19 20 22 25 28 29 32 36

File : aplikasi.ijl

[JumlahTulanganUtama]

4
2 3 4 5

File : aplikasi.ids

[DiameterTulanganSengkang]

3
8 10 12

File : aplikasi.ijs

[JarakAntarSengkang]

21
100 110 120 130 140 150 160 170 180 190 200
210 220 230 240 250 260 270 280 290 300

LAMPIRAN 34

Input data pembebanan struktur portal ruang pada gambar 4-3

File : aplikasi.bbn

[BebanBatang]

8	1	0	0	0	0	0	0
		0	0	0	0	0	0
2	0	0	0	0	0	0	0
		0	0	0	0	0	0
3	0	0	0	0	0	0	0
		0	0	0	0	0	0
4	0	0	0	0	0	0	0
		0	0	0	0	0	0
5	35000	0	105000	0	0	0	105000
		0	105000	0	0	0	-105000
6	35000	0	105000	0	0	0	105000
		0	105000	0	0	0	-105000
7	35000	0	105000	0	0	0	105000
		0	105000	0	0	0	-105000
8	35000	0	105000	0	0	0	105000
		0	105000	0	0	0	-105000

[BebanTitik]

8						
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0

LAMPIRAN 35

Hasil berupa informasi struktur portal ruang pada gambar 4-3

File : aplikasi.inf

Struktur Portal Ruang Aplikasi_Portal_Beton_3D

Parameter Struktur

Jumlah batang : 8

DOF : 24

Jumlah joint : 8

Jumlah pengekang tumpuan : 24

Jumlah titik kumpul yang dikekang : 4

Modulus Elastisitas aksial : 2.5743×10^{10} N/m²

Modulus Geser : 1.48022×10^{10} N/m²

Properti Elemen Material

Kuat desak beton karakteristik : 30 MPa

Kuat tarik baja tulangan : 400 MPa

Kuat tarik tulangan sengkang : 240 MPa

Koordinat Titik Kumpul (m)

Titik	X	Y	Z
1	0	0	0
2	6	0	0
3	0	0	6
4	6	0	6
5	0	5	0
6	6	5	0
7	0	5	6
8	6	5	6

Informasi Batang

Batang	JJ	JK	IA
1	1	5	0
2	2	6	0
3	3	7	0
4	4	8	0
5	5	6	0
6	7	8	0
7	5	7	0
8	6	8	0

Pengekang Titik Kumpul

Titik	JR1	JR2	JR3	JR4	JR5	JR6
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1

Gaya di Ujung Batang Terkekang Akibat Beban (Nm)

Batang	AML1 AML7	AML2 AML8	AML3 AML9	AML4 AML10	AML5 AML11	AML6 AML12
1	0	0	0	0	0	0
	0	0	0	0	0	0
2	0	0	0	0	0	0
	0	0	0	0	0	0
3	0	0	0	0	0	0
	0	0	0	0	0	0
4	0	0	0	0	0	0
	0	0	0	0	0	0
5	0	115800	0	0	0	115800
	0	115800	0	0	0	-115800
6	0	117960	0	0	0	117960
	0	117960	0	0	0	-117960
7	0	117600	0	0	0	117600
	0	117600	0	0	0	-117600
8	0	117960	0	0	0	117960
	0	117960	0	0	0	-117960

Beban Titik (N)

Titik	Arah 1	Arah 2	Arah 3	Arah 4	Arah 5	Arah 6
-------	--------	--------	--------	--------	--------	--------

LAMPIRAN 35

1	0	-24300	0	0	0	0
2	0	-30000	0	0	0	0
3	0	-30000	0	0	0	0
4	0	-24300	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0

LAMPIRAN 36

Hasil berupa riwayat proses optimasi struktur portal ruang pada gambar 4-3

File : aplikasi.his

1. Jumlah Struktur Desain = JVD + 3

Run ke-1

Penyusutan berturut pertama pada generasi : 275 Fitness terbaik : 884.543
Konvergen Pada Generasi : 345 karena jumlah penyusutan melebihi batas
Fitness = 884.543
Harga = 1.13053e+07
Kendala = 0
Waktu optimasi : 21 detik

Run ke-2

Penyusutan berturut pertama pada generasi : 388 Fitness terbaik : 920.91
Konvergen Pada Generasi : 780 karena jumlah penyusutan melebihi batas
Fitness = 920.91
Harga = 1.08588e+07
Kendala = 0
Waktu optimasi : 39 detik

Run ke-3

Penyusutan berturut pertama pada generasi : 111 Fitness terbaik : 896.938
Penyusutan berturut berakhir setelah 2 kali
Penyusutan berturut pertama pada generasi : 126 Fitness terbaik : 915.185
Penyusutan berturut berakhir setelah 2 kali
Penyusutan berturut pertama pada generasi : 145 Fitness terbaik : 922.347
Konvergen Pada Generasi : 215 karena jumlah penyusutan melebihi batas
Fitness = 922.347
Harga = 1.08419e+07
Kendala = 0
Waktu optimasi : 16 detik

Run ke-4

Konvergen pada generasi : 612
Fitness = 975.574
Harga = 1.02504e+07
Kendala = 0
Waktu optimasi : 26 detik

Run ke-5

Penyusutan berturut pertama pada generasi : 445 Fitness terbaik : 882.731
Penyusutan berturut berakhir setelah 2 kali
Penyusutan berturut pertama pada generasi : 712 Fitness terbaik : 972.161
Konvergen pada generasi : 782
Fitness = 978.844
Harga = 1.02161e+07
Kendala = 0
Waktu optimasi : 36 detik

2. Jumlah Struktur Desain = JVD x 2

Run ke-1

Penyusutan berturut pertama pada generasi : 378 Fitness terbaik : 952.009
Penyusutan berturut berakhir setelah 2 kali
Penyusutan berturut pertama pada generasi : 382 Fitness terbaik : 952.009
Konvergen Pada Generasi : 517 karena jumlah penyusutan melebihi batas
Fitness = 952.009
Harga = 1.05041e+07
Kendala = 0
Waktu optimasi : 36 detik

Run ke-2

Penyusutan berturut pertama pada generasi : 189 Fitness terbaik : 929.264
Penyusutan berturut berakhir setelah 2 kali
Penyusutan berturut pertama pada generasi : 580 Fitness terbaik : 1093.41
Konvergen pada generasi : 649
Fitness = 1102.15
Harga = 9.0732e+06
Kendala = 0
Waktu optimasi : 34 detik

LAMPIRAN 36

Run ke-3

Penyusutan berturut pertama pada generasi : 213 Fitness terbaik : 956.24
Konvergen pada generasi : 848
Fitness = 1044.07
Harga = 9.57794e+06
Kendala = 0
Waktu optimasi : 42 detik

Run ke-4

Penyusutan berturut pertama pada generasi : 204 Fitness terbaik : 945.211
Penyusutan berturut berakhir setelah 3 kali
Penyusutan berturut pertama pada generasi : 208 Fitness terbaik : 1061.04
Konvergen Pada Generasi : 343 karena jumlah penyusutan melebihi batas
Fitness = 1061.04
Harga = 9.42469e+06
Kendala = 0
Waktu optimasi : 30 detik

Run ke-5

Penyusutan berturut pertama pada generasi : 709 Fitness terbaik : 951.672
Konvergen pada generasi : 778
Fitness = 1030.85
Harga = 9.7007e+06
Kendala = 0
Waktu optimasi : 44 detik

3. Jumlah Struktur Desain = JVD x 3

Run ke-1

Penyusutan berturut pertama pada generasi : 252 Fitness terbaik : 956.726
Konvergen Pada Generasi : 455 karena jumlah penyusutan melebihi batas
Fitness = 956.726
Harga = 1.04523e+07
Kendala = 0
Waktu optimasi : 43 detik

Run ke-2

Penyusutan berturut pertama pada generasi : 644 Fitness terbaik : 963.602
Konvergen Pada Generasi : 847 karena jumlah penyusutan melebihi batas
Fitness = 963.602
Harga = 1.03777e+07
Kendala = 0
Waktu optimasi : 54 detik

Run ke-3

Penyusutan berturut pertama pada generasi : 341 Fitness terbaik : 1039.36
Konvergen Pada Generasi : 545 karena jumlah penyusutan melebihi batas
Fitness = 1039.36
Harga = 9.62127e+06
Kendala = 0
Waktu optimasi : 48 detik

Run ke-4

Penyusutan berturut pertama pada generasi : 377 Fitness terbaik : 933.474
Penyusutan berturut berakhir setelah 2 kali
Konvergen pada generasi : 1150
Fitness = 1077.23
Harga = 9.28304e+06
Kendala = 0
Waktu optimasi : 63 detik

Run ke-5

Penyusutan berturut pertama pada generasi : 378 Fitness terbaik : 978.152
Konvergen pada generasi : 1272
Fitness = 1171.59
Harga = 8.53543e+06
Kendala = 0
Waktu optimasi : 74 detik

LAMPIRAN 37

Hasil akhir struktur terbaik untuk kasus portal ruang pada gambar 4-3

File : aplikasi.opt

Nama Struktur : Aplikasi_Portal_Beton_3D
Hasil Optimasi Beton Bertulang Pada Struktur Portal Ruang
Metoda Optimasi : Flexible Polyhedron
Jumlah Variabel Desain : 68
Jumlah Struktur Desain : 204
Oleh Yohan Naftali 1999

Nama File Generik : aplikasi
Nama Struktur : Aplikasi_Portal_Beton_3D
Jumlah batang : 8
Jumlah titik kumpul : 8
Modulus elastisitas tarik/tekan : 25743 MPa
Modulus elastisitas geser G : 14802.2 MPa
Kuat desak Karakteristik Beton : 30 MPa
Kuat tarik baja tulangan lentur : 400 MPa
Kuat tarik baja tulangan sengkang : 240 MPa

Balok 1 250 x 600
Nomor Batang : 5
Panjang Bentang : 6 m
Beban Total : 38600 N/m
Berat Sendiri : 3600 N/m
Momen kiri : -93297.9 Nmm
Momen tengah : 123827 Nmm
Momen kanan : -106649 Nmm
Geser kiri : 142525 N
Geser kanan : -146975 N
Lendutan Tengah Bentang : 16.5438 mm
Lendutan ijin 33.3333 mm
Sengkang 10 - 160
Jarak sengkang maksimum : 168.684 mm

Daerah Lapangan
Tulangan tarik 4 D 19
Tulangan desak 3 D 19
Volume beton : 0.45 m³
Berat Tulangan utama : 46.716 kg
Berat Tulangan geser : 25.866 kg
Harga balok daerah lapangan : 475410
Momen yang membebani : 123827 Nmm
Momen yang dapat ditahan : 183502 Nmm

Daerah Tumpuan
Tulangan desak 4 D 25
Tulangan tarik 3 D 19
Volume beton : 0.45 m³
Berat Tulangan utama : 66.238 kg
Berat Tulangan geser : 25.866 kg
Harga balok pada tumpuan kiri + kanan : 573020
Momen yang membebani : 106649 Nmm
Momen yang dapat ditahan : 139707 Nmm

Balok 2 300 x 600
Nomor Batang : 6
Panjang Bentang : 6 m
Beban Total : 39320 N/m
Berat Sendiri : 4320 N/m
Momen kiri : -102355 Nmm
Momen tengah : 118820 Nmm
Momen kanan : -88960.6 Nmm
Geser kiri : 149682 N
Geser kanan : -145218 N
Lendutan Tengah Bentang : 9.2669 mm
Lendutan ijin 33.3333 mm
Sengkang 10 - 160
Jarak sengkang maksimum : 188.4 mm

LAMPIRAN 37

Daerah Lapangan
Tulangan tarik 4 D 22
Tulangan desak 3 D 29
Volume beton : 0.54 m³
Berat Tulangan utama : 82.4324 kg
Berat Tulangan geser : 26.9907 kg
Harga balok daerah lapangan : 682115
Momen yang membebani : 118820 Nmm
Momen yang dapat ditahan : 243842 Nmm

Daerah Tumpuan
Tulangan desak 2 D 25
Tulangan tarik 4 D 22
Volume beton : 0.54 m³
Berat Tulangan utama : 58.8988 kg
Berat Tulangan geser : 26.9907 kg
Harga balok pada tumpuan kiri + kanan : 564447
Momen yang membebani : 102355 Nmm
Momen yang dapat ditahan : 243813 Nmm

Balok 3 350 x 500
Nomor Batang : 7
Panjang Bentang : 6 m
Beban Total : 39200 N/m
Berat Sendiri : 4200 N/m
Momen kiri : -101127 Nmm
Momen tengah : 119373 Nmm
Momen kanan : -114716 Nmm
Geser kiri : 144735 N
Geser kanan : -149265 N
Lendutan Tengah Bentang : 15.4015 mm
Lendutan ijin 33.3333 mm
Sengkang 10 - 160
Jarak sengkang maksimum : 161.486 mm

Daerah Lapangan
Tulangan tarik 4 D 28
Tulangan desak 2 D 22
Volume beton : 0.525 m³
Berat Tulangan utama : 75.8696 kg
Berat Tulangan geser : 23.6168 kg
Harga balok daerah lapangan : 628682
Momen yang membebani : 119373 Nmm
Momen yang dapat ditahan : 305240 Nmm

Daerah Tumpuan
Tulangan desak 2 D 19
Tulangan tarik 3 D 25
Volume beton : 0.525 m³
Berat Tulangan utama : 48.0101 kg
Berat Tulangan geser : 23.6168 kg
Harga balok pada tumpuan kiri + kanan : 489385
Momen yang membebani : 114716 Nmm
Momen yang dapat ditahan : 190563 Nmm

Balok 4 300 x 600
Nomor Batang : 8
Panjang Bentang : 6 m
Beban Total : 39320 N/m
Berat Sendiri : 4320 N/m
Momen kiri : -102269 Nmm
Momen tengah : 118906 Nmm
Momen kanan : -88770.3 Nmm
Geser kiri : 149700 N
Geser kanan : -145200 N
Lendutan Tengah Bentang : 9.87336 mm
Lendutan ijin 33.3333 mm
Sengkang 10 - 180

LAMPIRAN 37

Jarak sengkang maksimum : 188.4 mm

Daerah Lapangan

Tulangan tarik 2 D 28
Tulangan desak 3 D 29
Volume beton : 0.54 m^3
Berat Tulangan utama : 75.6293 kg
Berat Tulangan geser : 23.9095 kg
Harga balok daerah lapangan : 632694
Momen yang membebani : 118906 Nmm
Momen yang dapat ditahan : 198299 Nmm

Daerah Tumpuan

Tulangan desak 2 D 28
Tulangan tarik 2 D 22
Volume beton : 0.54 m^3
Berat Tulangan utama : 46.8824 kg
Berat Tulangan geser : 23.9095 kg
Harga balok pada tumpuan kiri + kanan : 488960
Momen yang membebani : 102269 Nmm
Momen yang dapat ditahan : 150791 Nmm

Kolom 1 450 x 450
Nomor Batang : 1
Tulangan utama 8 D 19
Jarak antar tulangan : 146.5 mm
Sengkang 10 - 300
Jarak sengkang maksimum : 304 mm
Volume beton : 1.0125 m^3
Berat Tulangan utama : 88.9829 kg
Berat Tulangan geser : 13.5159 kg
Harga kolom : 765619
Berat Kolom : 24300 N
Gaya aksial : 287260 N
Gaya aksial yang dapat ditahan = 863066 N
Momen arah X : 733.231 Nmm
Momen arah Y : 91972.2 Nmm
Momen yang dapat ditahan : 271092 Nmm
Geser pada kolom : 30668.3 N

Kolom 2 500 x 500
Nomor Batang : 2
Tulangan utama 8 D 22
Jarak antar tulangan : 167 mm
Sengkang 10 - 300
Jarak sengkang maksimum : 352 mm
Volume beton : 1.25 m^3
Berat Tulangan utama : 119.301 kg
Berat Tulangan geser : 15.4467 kg
Harga kolom : 986239
Berat Kolom : 30000 N
Gaya aksial : 296675 N
Gaya aksial yang dapat ditahan = $1.08939 \times 10^6 \text{ N}$
Momen arah X : 597.372 Nmm
Momen arah Y : 107095 Nmm
Momen yang dapat ditahan : 393148 Nmm
Geser pada kolom : 30937.6 N

Kolom 3 500 x 500
Nomor Batang : 3
Tulangan utama 8 D 28
Jarak antar tulangan : 158 mm
Sengkang 10 - 300
Jarak sengkang maksimum : 448 mm
Volume beton : 1.25 m^3
Berat Tulangan utama : 193.248 kg
Berat Tulangan geser : 15.4467 kg
Harga kolom : 1.35597×10^6

LAMPIRAN 37

Berat Kolom : 30000 N
Gaya aksial : 298947 N
Gaya aksial yang dapat ditahan = 1.65179×10^6 N
Momen arah X : 165.848 Nmm
Momen arah Y : 103680 Nmm
Momen yang dapat ditahan : 572763 Nmm
Geser pada kolom : 33103.8 N

Kolom 4 450 x 450
Nomor Batang : 4
Tulangan utama 8 D 22
Jarak antar tulangan : 142 mm
Sengkang 8 - 300
Jarak sengkang maksimum : 352 mm
Volume beton : 1.0125 m^3
Berat Tulangan utama : 119.301 kg
Berat Tulangan geser : 8.65016 kg
Harga kolom : 892882
Berat Kolom : 24300 N
Gaya aksial : 290418 N
Gaya aksial yang dapat ditahan = 1.12221×10^6 N
Momen arah X : 137.749 Nmm
Momen arah Y : 88514.9 Nmm
Momen yang dapat ditahan : 341799 Nmm
Geser pada kolom : 27419 N

Harga Beton Rp.250000 / m^3
Harga Besi Rp.5000 /kg
Tebal selimut kolom 50 mm
Tebal selimut balok 50 mm
Faktor finalti : 1×10^{10}
JSTD = (JVD*3)+0

Hasil perhitungan gaya batang pada kasus portal ruang gambar 4-3

File : aplikasi.str

Hasil Analisa Struktur Dengan Metoda Kekakuan
Dikembangkan dari Weaver & Gere
Oleh Yohan Naftali 1999

Nama File Generik : apl15/aplikasi
Jumlah batang : 8
Jumlah titik kumpul : 8
Modulus elastisitas tarik/tekan : 25743 MPa
Modulus elastisitas geser G : 14802.2 MPa

Perpindahan Titik Kumpul

Titik	DJ1	DJ2	DJ3	DJ4	DJ5	DJ6
1	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
2	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
3	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
4	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000	+0.000000
5	-0.000259	-0.000220	-0.000277	+0.001051	+0.000034	-0.000968
6	-0.000296	-0.000184	+0.000305	+0.000864	+0.000018	+0.000888
7	+0.000295	-0.000186	-0.000310	-0.000947	+0.000005	-0.000862
8	+0.000266	-0.000223	+0.000275	-0.000929	-0.000006	+0.000927

Gaya Ujung Batang

Batang	AM1	AM2	AM3	AM4	AM5	AM6
	AM7	AM8	AM9	AM10	AM11	AM12
1	229808	-22620.5	24534.7	-586.584	-42837.8	-39524.8
	-229808	22620.5	-24534.7	586.584	-79835.5	-73577.8
2	237340	24750.1	23883.5	-477.898	-36536.6	38074.5
	-237340	-24750.1	-23883.5	477.898	-82881	85675.8
3	239158	-23933.7	-26483.1	-132.679	40809	-36724.3
	-239158	23933.7	26483.1	132.679	91606.2	-82944.3
4	232334	21804.2	-21935.2	110.199	38493.5	38209.1
	-232334	-21804.2	21935.2	-110.199	71182.4	70811.9
5	23734.6	114020	-826.098	1065.98	2531.83	74638.4
	-23734.6	117580	826.098	-1065.98	2424.76	-85319.2
6	22819.6	119746	-1122.27	-166.143	3433.64	81883.8
	-22819.6	116174	1122.27	166.143	3299.98	-71168.5
7	25360.8	115788	1114.12	-1060.55	-3118.41	80901.5
	-25360.8	119412	-1114.12	1060.55	-3566.31	-91772.4
8	23057.5	119760	1015.41	-356.562	-2902.66	81815
	-23057.5	116160	-1015.41	356.562	-3189.78	-71016.3

Reaksi Tumpuan

Titik	AR1	AR2	AR3	AR4	AR5	AR6
1	22620.5	254108	24534.7	42837.8	-586.584	-39524.8
2	-24750.1	267340	23883.5	36536.6	-477.898	38074.5
3	23933.7	269158	-26483.1	-40809	-132.679	-36724.3
4	-21804.2	256634	-21935.2	-38493.5	110.199	38209.1