UNIVERSITAS KRISTEN MARANATHA
FAKULTAS TEKNIK JURUSAN SIPIL
Drg Soeria Sumantri 65 Tlp.(022) 2012186 – 2012692 Bandung 40164

**SEMINAR ISI TUGAS AKHIR**

| | |
|---|---|
| Bidang | : Struktur |
| Judul | : **KOMPUTERISASI SAMBUNGAN LAS YANG MEMIKUL MOMEN SEBIDANG DENGAN METODE KEKUATAN BATAS BERDASARKAN SPESIFIKASI AISC – LRFD 1999.** |
| Nama | : Elga Yulius |
| NRP | : 0021042 |
| Pembimbing | : Prof. Bambang Suryoatmono .,Ph.D |
| Penguji | : 1. Ginardy Husada .,Ir.,M.T<br>2. Anang K. ,S.T ., M.T<br>3. Daud R. Wiyono , Ir.,M.Sc |
| Hari / Tanggal | : Jumat, 13 Februari 2004 |
| Waktu | : 08.00 - selesai |
| Tempat | : Ruang Sidang Fakultas Teknik Jurusan Sipil |

Bandung, Februari 2004

Menyetujui,

| | |
|---|---|
| Rini I. Rusadi,Ir | Prof. Bambang S.,Ph.D |
| Koord. Tugas Akhir | Pembimbing |

UNIVERSITAS KRISTEN MARANATHA
FAKULTAS TEKNIK JURUSAN SIPIL
Drg Soeria Sumantri 65 Tlp.(022) 2012186 – 2012692 Bandung 40164

**SEMINAR JUDUL TUGAS AKHIR**

| | |
|---|---|
| Bidang | : Struktur |
| Judul | : **KOMPUTERISASI SAMBUNGAN LAS YANG MEMIKUL MOMEN SEBIDANG DENGAN METODE KEKUATAN BATAS BERDASARKAN SPESIFIKASI AISC – LRFD 1999.** |
| Nama | : Elga Yulius |
| NRP | : 0021042 |
| Pembimbing | : Prof. Bambang Suryoatmono .,Ph.D |
| Penguji | : 1. Ginardy Husada .,Ir.,M.T<br>2. Anang K. ,S.T ., M.T<br>3. Daud R. Wiyono , Ir.,M.Sc |
| Hari / Tanggal | : Kamis, 20 November 2003 |
| Waktu | : 08.00 - selesai |
| Tempat | : Ruang Sidang Fakultas Teknik Jurusan Sipil |

Bandung, November 2003

Menyetujui,

| | |
|---|---|
| Rini I. Rusadi,Ir | Prof. Bambang S.,Ph.D |
| Koord. Tugas Akhir | Pembimbing |

# TUGAS AKHIR PEMBANDING

| | | |
|---|---|---|
| Judul | : | **KOMPUTERISASI DESAIN SAMBUNGAN ANTARA BALOK DAN KOLOM TIPE DUA PROFIL T DENGAN SIKU BADAN BERDASARKAN SPESIFIKASI AISC – LRFD 1993** |
| Nama | : | Jimmy Wijaya |
| NRP | : | 9621008 |
| Pembimbing | : | Bambang S., Ir., M.Sc., Ph.D |
| Tahun Selesai | : | 2001 |
| Masalah | : | Program komputer yang dapat mendesain suatu sambungan antara balok dan kolom tipe dua profil T dengan siku badan |

Listing Program Penampang_Las_Sudut_C

**Daftar Notasi Masuk**

```
{variabel untuk input data}
N : integer
w : real
kl : real
l : real
Fexx :integer
b : real
degree : real
teliti : real

const phi = 22 / 7
i : integer
pj_las : real
kontrol : real

{variabel untuk output data}
XIC : real
YIC :real
Rn_1 : real
Rn_2 : real
Rn_3 : real

{variabel untuk array data}
x : array[1..100] of real
y : array[1..100] of real
R : array[1..100] of real
delta_m : array[1..100] of real
Rd_ : array[1..100] of real
Ry_ : array[1..100] of real
q : array[1..100] of real
sudut : array[1..100] of real
delta : array[1..100] of real
delta_mr : array[1..100] of real
delta_u : array[1..100] of real
delta_p : array[1..100] of real
suku1_ : array[1..100] of real
```

```
suku2_1_ : array[1..100] of real
suku2_2_ : array[1..100] of real
R_ : array[1..100] of real
Sigma_momen : real
Sigma_gaya : real
Rn_3 : real
Selisih_gaya : real
z : real
q : real
x1 : real
y1 : real
a : real
c : real
x2 : array [1..100] of real
y2 : array [1..100] of real
```

**ALGORITMA**

```
Input (N, w, kl, l, Fexx, b, arah beban, teliti)

z = Cos (degree)
a = Sin (degree)
pj_las_horizontal ← (kl) / (0.5 x N)
pj_las_vertikal ← (l) / (0.5 x N)

YIC ← l
XIC ← 0
Do
      XIC ← XIC + teliti
      q = (XIC + b + kl)
      x1 = q * (1-z)
      y1 = q * a

      x(1) ← XIC + kl – (pj_las_horizontal / 2)
      for i ← 0 to N do
            x(i+1) ← x(i) – pj_las_horizontal
            if x(i+1) < XIC then
                  x(i+1) ← XIC
            endif
      endfor
```

```
      y(1) ← l
      for i ← 0 to N do
            if x(i+1) > XIC then
                  y(i+1) ← y(i)
                  else if x(i-1) > (XIC + pj_las_horizontal / 2)
then
                  y(i+1) ← y(i) – pj_las_vertikal / 2
                  else
                  y(i+1) ← y(i) – pj_las_vertikal
            endif
      endfor

      for i ← 1 to N do
            c = q – X(i)-(q-X(i))*z
            x2(i) = c + X(i) + Y(i)*a
            y2(i) = Y(i)*z + (q-X(i))*a
      endif

      for i ← 1 to N do
      R(i) ← sqr((x2(i) – x1)^2 + (y2(i) – y1)^2)
            if x(i) > XIC then
            sudut(i) ← Atn(x2(i) / y2(i)) * 180 / phi
            else
            sudut(i) ← Atn(y2(i) / x2(i)) * 180 / phi
            endif
      endfor

      for i ← 1 to N do
            delta_m(i) ← 0.209 * (2 + sudut(i)) ^ (-0.32) * w
            delta_mr(i) ← delta_m(i) / R(i)
            delta_u(i) ← 1.087 * (6 + sudut(i)) ^ (-0.65) * w
            kontrol ← 0.17 * w

            if delta_u(i) > kontrol then
                  delta_u(i) ← kontrol
            endif

            delta(i) ← R(i) * delta_u(1) / R(1)
```

```
            delta_p(i) ← delta(i) / delta_m(i)
            suku1_(i) ← 1 + 0.5 * (sin(phi / 180 * sudut(i)))
            suku2_1_(i) ← Abs(delta_p(i) * (1.9 – 0.9 *
            delta_p(i)))
            suku2_2_(i) ← suku2_1_(i) ^ (0.3)
            if x(i) > XIC
                  R_(i) ← 0.6 * Fexx * suku1_(i) * suku2_2_(i) *
                  (0.707 * pj_las_horizontal * w)
                  Else
                  R_(i) ← 0.6 * Fexx * suku1_(i) * suku2_2_(i) *
                  (0.707 * pj_las_vertikal * w)
            Rd_(i) ← R_(i) * R(i)
                  if x(i) > XIC then
                  Ry_(i) ← R_(i) * Sin(phi / 180 * sudut(i))
                        Else
                        Ry_(i) ← R_(i) * Cos(phi / 180 * sudut(i))
                  endif
      endfor

      Sigma_momen ← 0
      Sigma_gaya ← 0

      for i ← 1 to N do
            Sigma_momen ← Sigma_momen + Rd_(i)
            Sigma_gaya ← Sigma_gaya + Ry_(i)
      endfor

      Rn_1 ← 2 * Sigma_gaya
      Rn_2 ← (2 * Sigma_momen) / (XIC + b + kl)
      Selisih_Gaya ← Rn_1 - Rn_2

Loop Until (Selisih_gaya < 1) And (Selisih_gaya > 0)


{menampilkan nilai-nilai yang sudah diproses}
Output (Rn_1, XIC, YIC)

{menampilkan nilai-nilai yang sudah diproses dalam bentuk array}
for i ← 1 to N do
```

```
    Output (i)
    Output (X(i))
    Output (Y(i))
    Output (R(i))
    Output (Sudut(i))
    Output (delta_m(i))
    Output (delta_mr(i))
    Output (delta_u(i))
    Output (delta(i))
    Output (delta_p(i))
    Output (R_(i))
    Output (Rd_(i))
    Output (Ry_(i))
endfor
```

Listing Program penampang_las_sudut_L

**Daftar Notasi Masuk**

```
{variabel untuk input data}
N : integer
w : real
kl : real
l : real
Fexx : integer
b : real
degree : real
teliti : real

const phi = 22 / 7
i : integer
pj_las_horizontal : real
pj_las_vertikal : real
kontrol : real

{variabel untuk output data}
XIC : real
YIC : real
Rn_1 : real
Rn_2 : real

{variabel untuk data array}
x : array[1..100] of real
y : array[1..100] of real
R : array[1..100] of real
delta_m : array[1..100] of real
Rd_ : array[1..100] of real
Ry_ : array[1..100] of real
Rx_ : array[1..100] of real
sudut : array[1..100] of real
delta : array[1..100] of real
delta_mr : array[1..100] of real
delta_u : array[1..100] of real
delta_u_kontrol : array[1..100] of real
delta_p : array[1..100] of real
```

```
suku1_ : array[1..100] of real
suku2_1_ : array[1..100] of real
suku2_2_ : array[1..100] of real
R_ : array[1..100] of real
Sigma_Momen : real
Sigma_Gaya_Vertikal : real
Sigma_Gaya_Horisontal : real
Selisih_Gaya As : real
z : real
q : real
x1 : real
y1 : real
a : real
c : real
x2 : array [1..100] of real
y2 : array [1..100] of real
```

**ALGORITMA**

```
Input (N, w, kl, l, Fexx, b, arah beban, teliti)

z = Cos (degree)
a = Sin (degree)
pj_las_horizontal ← (kl) / (0.5 x N)
pj_las_vertikal ← (l) / (0.5 x N)

XIC ← 0.001
Do
      YIC ← YIC + 0.01
      x(1) ← (XIC + kl) - pj_las_horizontal / 2
      for i ← 1 to N do
                  x(i + 1) ← x(i) - pj_las_horizontal
                        if x(i + 1) < XIC then
                        x(i + 1) ← XIC
                        endif
      endfor

      y(1) ← l - YIC
```

```
for i ← 1 to N do
            if x(i + 1) > XIC then
                  y(i + 1) ← Y(i)
                  elseif (X(i - 1) > (XIC +
                  pj_las_horizontal / 2)) then
                  y(i + 1) ← y(i) - pj_las_vertikal / 2
                  else
                  y(i + 1) ← y(i) - pj_las_vertikal
            endif
endfor

for i ← 1 to N do
      R(i) ← Sqr(x(i) * x(i) + y(i) * y(i))
      if x(i) > XIC then
          sudut(i) ← Abs(180 / phi * Atn(x(i) / y(i)))
      else
          sudut(i) ← Abs(180 / phi * Atn(y(i) / x(i)))
      endif
endfor


for i ← 1 to N do
      delta_m(i) ← 0.209 * (2 + sudut(i)) ^ (-0.32) * w
      delta_mr(i) ← delta_m(i) / R(i)
      delta_u(i) ← 1.087 * (6 + sudut(i)) ^ (-0.65) * w
      kontrol ← 0.17 * w

      if delta_u(i) > kontrol then
            delta_u_kontrol(i) ← kontrol
            else
            delta_u_kontrol(i) ← delta_u(i)
      endif
endfor

for i ← 1 to N do
      delta(i) ← R(i) * delta_u_kontrol(N) / R(N)
      delta_p(i) ← delta(i) / delta_m(i)
      suku1_(i) ← 1 + 0.5 * (Sin(phi / 180 * sudut(i))) ^
      (1.5)
```

```
            suku2_1_(i) ← Abs(delta_p(i) * (1.9 - 0.9 *
            delta_p(i)))
            suku2_2_(i) ← suku2_1_(i) ^ (0.3)
            if x(i) > XIC then
                  R_(i) ← 0.6 * Fexx * suku1_(i) * suku2_2_(i) *
                  (0.707 * pj_las_horizontal * w)
                  Else
                  R_(i) ← 0.6 * Fexx * suku1_(i) * suku2_2_(i) *
                  (0.707 * pj_las_vertikal * w)
            if X(i) > XIC then
                  Rx_(i) ← R_(i) * Cos(phi / 180 * sudut(i))
                  else
                      Rx_(i) ← R_(i) * Sin(phi / 180 * sudut(i))
                  if Y(i) < 0 then
                      Rx_(i) ← -1 * R_(i) * Sin(phi / 180 *
                      sudut(i))
                  endif
            endif
      endfor

      Sigma_Gaya_Horisontal ← 0

      for i ← 1 to N do
            Sigma_Gaya_Horisontal ← Sigma_Gaya_Horisontal + Rx_(i)
      endfor

Loop Until (Sigma_Gaya_Horisontal < 1 And Sigma_Gaya_Horisontal >
0)

XIC ← 0
Do
      XIC ← XIC + teliti
      q = (XIC + b + kl)
      x1 = q * (1-z)
      y1 = q * a

      x(1) ← (XIC + kl) - pj_las / 2

      for i ← 1 to N do
```

```
            x(i + 1) ← x(i) - pj_las_horizontal
            if x(i + 1) < XIC then
            x(i + 1) ← XIC
            endif
      endfor

      y(1) ← 1 - YIC
      for i ← 1 to N do
            if x(i + 1) > XIC then
                  y(i + 1) ← y(i)
                  elseif (X(i - 1) > (XIC + pj_las_horizontal /
2)) then
                  y(i + 1) ← Y(i) - pj_las_vertikal / 2
                  else
                  y(i + 1) ← y(i) - pj_las_vertikal
            endif
      endfor

      for i ← 1 to N do
            c = q – X(i)-(q-X(i))*z
            x2(i) = c + X(i) + Y(i)*a
            y2(i) = Y(i)*z + (q-X(i))*a
      endif

      for i ← 1 to N do
      R(i) ← Sqr((x2(i) – x1)^2 + (y2(i) – y1)^2)
            if x(i) > XIC then
            sudut(i) ← abs(180 / phi * Atn(x2(i) / y2(i)))
            else
            sudut(i) ← abs(180 / phi * Atn(y2(i) / x2(i)))
            endif
      endfor

      for i ← 1 to N do
            delta_m(i) ← 0.209 * (2 + sudut(i)) ^ (-0.32) * w
            delta_mr(i) ← delta_m(i) / R(i)
            delta_u(i) ← 1.087 * (6 + sudut(i)) ^ (-0.65) * w
                  kontrol ← 0.17 * w
```

```
      if delta_u(i) > kontrol then
            delta_u_kontrol(i) ← kontrol
            else
            delta_u_kontrol(i) ← delta_u(i)
      endif
endfor

for i ← 1 to N do
      delta(i) ← R(i) * delta_u_kontrol(N) / R(N)
      delta_p(i) ← delta(i) / delta_m(i)
      suku1_(i) ← 1 + 0.5 * (Sin(phi / 180 * sudut(i))) ^
      (1.5)
      suku2_1_(i) ← Abs(delta_p(i) * (1.9 - 0.9 *
      delta_p(i)))
      suku2_2_(i) ← suku2_1_(i) ^ (0.3)
      if x(i) > XIC then
            R_(i) ← 0.6 * Fexx * suku1_(i) * suku2_2_(i) *
            (0.707 *  pj_las_horizontal * w)
            Else
            R_(i) ← 0.6 * Fexx * suku1_(i) * suku2_2_(i) *
            (0.707 *  pj_las_vertikal * w)
      Rd_(i) ← R_(i) * R(i)

      if x(i) > XIC then
            Ry_(i) ← R_(i) * Sin(phi / 180 * sudut(i))
            else
            Ry_(i) ← R_(i) * Cos(phi / 180 * sudut(i))
      endif
endfor

Sigma_Momen ← 0
Sigma_Gaya_Vertikal ← 0

for i ← 1 to N do
      Sigma_Momen ← Sigma_Momen + Rd_(i)
      Sigma_Gaya_Vertikal ← Sigma_Gaya_Vertikal + Ry_(i)
endfor

Rn_1 ← Sigma_Gaya_Vertikal
```

```
      Rn_2 ← Sigma_Momen / (XIC + kl + b)
      Selisih_Gaya ← Rn_1 - Rn_2

Loop Until (Selisih_Gaya < 1 And Selisih_Gaya > 0)

{menampilkan nilai-nilai yang sudah diproses}
Output (Rn_1, XIC, YIC)

{menampilkan nilai-nilai yang sudah diproses dalam bentuk array}
for i ← 1 to N do
  Output (i)
    Output (X(i))
    Output (Y(i))
    Output (R(i))
    Output (Sudut(i))
    Output (delta_m(i))
    Output (delta_mr(i))
    Output (delta_u(i))
    Output (delta(i))
    Output (delta_p(i))
    Output (R_(i))
    Output (Rd_(i))
    Output (Ry_(i))
    Output (Rx_(i))
endfor
```

Tabel 8-37
Electrode Strength Coeffisients

| **Electrode** | **$F_{exx}$ (ksi)** | **$C_1$** |
|---|---|---|
| E60 | 60 | 0.857 |
| E70 | 70 | 1.00 |
| E80 | 80 | 1.03 |
| E90 | 90 | 1.16 |
| E100 | 100 | 1.21 |
| E110 | 110 | 1.34 |