

BAB 3

ASPEK PROGRAMASI METODE PPR DENGAN UI-FEAP

3.1 Umum

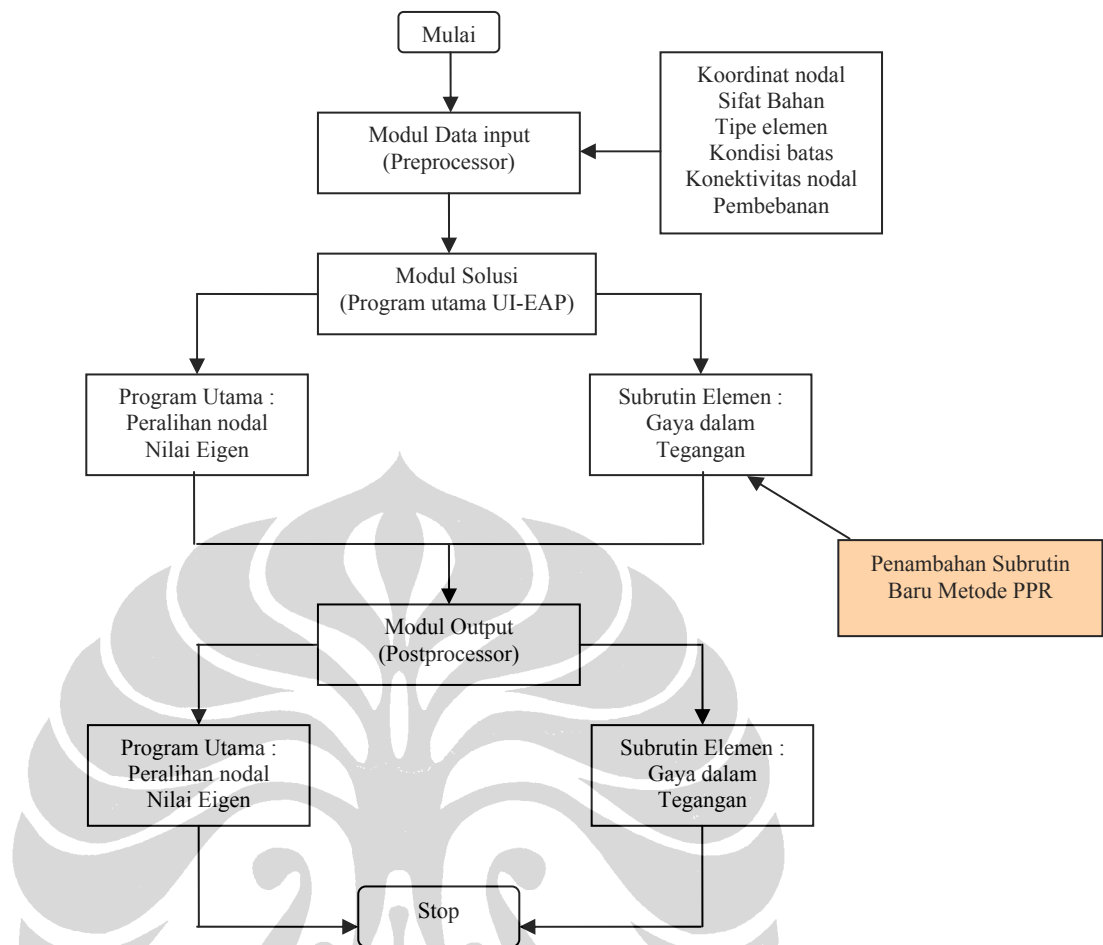
Implementasi ini akan dilakukan dengan memodifikasi subrutin yang ada didalam UI-FEAP. Penjelasan dalam bab ini sebagian besar disampaikan dalam bentuk bahasa pemrograman Fortran dengan disertai gambaran singkat.

3.2 Modifikasi Subrutin UI-FEAP

Implementasi metode PPR pada program UI-FEAP merupakan sebuah modifikasi pada paket program UI-FEAP yang perlu dilakukan untuk menunjang kegiatan penelitian kali ini. Implementasi ini bersifat *user based solution program* yang artinya implementasi tersebut tidak melibatkan modifikasi struktur baku UI-FEAP secara signifikan dan dapat dimodifikasi lebih lanjut tanpa mengganggu struktur program yang dibuat oleh *user* lain.

Seperti yang sudah dijelaskan sebelumnya bahwa UI-FEAP melibatkan tiga modul dalam melakukan proses komputasi metode elemen hingga (lihat Gambar 4.1). Dalam implementasi metode PPR ini, dilakukan modifikasi pada UI-FEAP pada bagian modul solusi. Oleh karena dalam penelitian kali ini metode PPR digunakan pada elemen MITC maka modifikasi program yang dilakukan masih terbatas pada elemen tersebut atau elemen quadrilateral lainnya yang sederajat. Selain itu, karena prosedur PPR pada dasarnya sama dengan prosedur SPR yang sudah dibuat oleh peneliti terdahulu, maka banyak subrutin-subrutin SPR yang digunakan dengan atau tanpa modifikasi.

Penjelasan mengenai subrutin-subrutin yang dimodifikasi maupun yang baru dapat dilihat pada Subbab 4.2.2.



Gambar 3.1 Modifikasi Secara Umum Yang Dilakukan Pada UI-FEAP Untuk Aplikasi Metode PPR

3.2.1 Daftar Subrutin Terkait Dengan Error Estimator dan Metode PPR

Pada Tabel dibawah ini akan ditampilkan nama-nama subrutin beserta fungsinya yang berkaitan dengan implementasi metode PPR pada elemen MITC [B10], formulasi estimasi error, beserta fungsinya.

Tabel 3.2.1 Formulasi Estimasi Error

Nama subrutin	Fungsi	Keterangan
ELMT04	Subrutin elemen pelat DKMQ [K2]	-
ELMT09	Subrutin elemen pelat MITC [B10]	Modifikasi
GEOME	Menghitung koefisien matriks Jacobian	-
HOOKE	Menghitung matriks Hooke bending $[H_b]$ dan shear $[H_s]$	-
ANMITC4	Menghitung matriks $[A_n]$	-

FEDKMQ	Menghitung vektor beban nodal ekuivalen untuk fungsi linear, kuadratik, dan kubik	-
STFVKE	Menghitung matriks kekakuan elemen $[k] = [k_b] + [k_s]$ dengan integrasi numerik 2x2 titik Gauss	-
NIQUAD	Menghitung turunan pertama bilinear shape function $\langle N \rangle$ terhadap sumbu koordinat natural ξ, η	-
JACOBQ	Menghitung koefisien matriks invers Jacobian $[j]$	-
JACOBN	Menghitung determinan matriks Jacobian $ J $	-
STFVKB	Menghitung matriks kekakuan lentur $[k_b]$	-
BBBDKMQ	Menghitung matriks $[B_b]$ elemen MITC4	-
STFVKS	Menghitung matriks kekakuan geser $[k_s]$	-
BSMITC4	Menghitung matriks $[B_s]$ elemen MITC4	-
DEF CG	Output nilai kurvatur pada titik tengah elemen	-
DEF MID	Output nilai kurvatur pada titik tengah sisi elemen	-
DEF NO	Output nilai kurvatur pada titik nodal elemen	-
EFF CG	Output nilai gaya dalam pada titik tengah elemen	-
EFF MID	Output nilai gaya dalam pada titik tengah sisi elemen	-
EFF NO	Output nilai gaya dalam pada titik nodal elemen	-
SHFUNC	Menghitung bilinear shape function $\langle N \rangle$ dan quadratik shape function $\langle P \rangle$	-
NoForDKMQ_1	Menghitung pemulihan gaya dalam dengan metode rata-rata langsung dan menghitung energi regangan FEM tingkat lokal $\ u^h\ _i^2$ dan global $\ u^h\ ^2$	-
NoForDKMQ_2	Menghitung pemulihan gaya dalam dengan metode proyeksi $[P]$, $\int_A \langle N \rangle \{M^h\} dA$, $\int_A \langle N \rangle \{T^h\} dA$, dan menghitung energi regangan FEM tingkat lokal $\ u^h\ _i^2$ dan global $\ u^h\ ^2$	-
NoForDKMQ_3	Menghitung pemulihan gaya dalam dengan metode SPR dan REP dan menghitung energi regangan FEM tingkat lokal $\ u^h\ _i^2$ dan global $\ u^h\ ^2$	-
NoForDKMQ_4	<i>Menghitung pemulihan gaya dalam dengan PPR langsung dan menghitung energi regangan FEM tingkat lokal $\ u^h\ _i^2$ dan global $\ u^h\ ^2$</i>	<i>Baru</i>
INVHB	Menghitung invers matriks $[H_b]$	-
INVHS	Menghitung invers matriks $[H_s]$	-

ERRORESTIMATOR	Menghitung norma energi tingkat lokal $\ e^*\ _i^2$ dan global $\ e^*\ ^2$ serta perhitungan indikator penghalusan elemen ζ_k	-
FRCNO	Menghitung gaya dalam pada titik nodal elemen	-
FRCGA	Menghitung gaya dalam pada titik Gauss elemen	-
UMACR0	Subrutin <i>user command language</i> sebagai pendukung untuk perhitungan pemulihan gaya dalam	-
COUNT_THE_PATCH 1	Membentuk patch menggunakan metode <i>Nodal Based Patch</i>	-
COR	Mentransformasi koordinat riil ke dalam koordinat natural patch	-
PATCHING	Menghitung koordinat riil titik Gauss tiap elemen	-
REP_Recovery	Menghitung matriks $[D]$ dan $\{F^h\}$, menghitung matriks $\{a_n\}$, menghitung pemulihan gaya dalam pada nodal berdasarkan persamaan untuk tiap komponen gaya dalam	-
<i>PPR_Recovery</i>	<i>Menghitung matriks $[Q]$, menghitung matriks $\{a_n\}$, menghitung pemulihan gaya dalam pada nodal berdasarkan persamaan untuk tiap gaya dalam, dimana Polynomial dengan 6 term orde.</i>	<i>Baru</i>
<i>PPR_RecoveryP1</i>	<i>Menghitung matriks $[Q]$, menghitung matriks $\{a_n\}$, menghitung pemulihan gaya dalam pada nodal berdasarkan persamaan untuk tiap gaya dalam, dimana Polynomial dengan 8 term orde.</i>	<i>Baru</i>
<i>PPR_RecoveryP2</i>	<i>Menghitung matriks $[Q]$, menghitung matriks $\{a_n\}$, menghitung pemulihan gaya dalam pada nodal berdasarkan persamaan untuk tiap gaya dalam, dimana Polynomial dengan 9 term orde.</i>	<i>Baru</i>
UMACR2	Melakukan proses pemilihan teknik pemulihan gaya dalam. Menghitung error ijin elemen e_m , mencetak error norma energi elemen $\ e^*\ _i^2$ dan struktur $\ e^*\ ^2$, menghitung serta mencetak indikator error relatif struktur ϕ^*	Modifikasi
UALLOC	Mengalokasikan memori untuk user dalam menggunakan variabel yang digunakan.	-
<i>COUNT_THE_PATCH 4</i>	<i>Membentuk patch menggunakan metode <i>Nodal Based Patch</i> untuk metode PPR</i>	<i>Baru</i>
<i>getMATAq</i>	<i>Menghitung matriks $\langle P \rangle$ dan $[Q]$ dalam Patch</i>	<i>Baru</i>
<i>getMATnp</i>	<i>Menghitung matriks fungsi bentuk $[N_p]$ untuk tiap nodal dalam patch</i>	<i>Baru</i>
<i>geopatch1</i>	<i>Mendapatkan Parameter Patch, seperti determinan dan invers jacobian dari patch</i>	<i>Baru</i>
<i>COEFISIEN</i>	<i>Mendapatkan koefisien dari persamaan polynomial termasuk turunannya</i>	<i>Baru</i>

Semua subrutin yang disebutkan pada Tabel diatas, khususnya untuk jenis subrutin lama, tidak akan dibahas semuanya namun hanya subrutin-subrutin baru yang terkait dengan teknik pemulihan gaya dalam metode PPR. Untuk penjelasan

subrutin mengenai error estimasi Z^2 pada UI-FEAP serta subrutin MITC sudah tersedia banyak dokumentasi tentang itu.

Demikian juga, karena prosedur perhitungan dengan metode PPR sebenarnya sama dengan perhitungan dengan metode SPR, maka banyak subrutin-subrutin yang digunakan dalam metode SPR yang juga dipakai dalam metode PPR. Kesamaannya adalah pada pembentukan patch, serta penghitungan koordinat riil dan transformasinya ke koordinat natural patch. Subrutin-subrutin `Count_The_Patch1` atau `4_Cor`, dan `Patching` dipakai di metode PPR ini tanpa modifikasi, dan karena itu tidak akan dibahas lagi di sini.

3.2.2 Penjelasan Kode Fortran Subrutin PPR

Berikut ini adalah penjelasan singkat mengenai kode Fortran pada subrutin teknik pemulihan gaya dalam PPR. Struktur lengkap dari subrutin PPR dan contoh detail proses PPR dapat dilihat pada bagian lampiran. Penjelasan akan dititikberatkan pada ungkapan-ungkapan aritmatik, sedangkan untuk parameter array tidak akan dijelaskan semuanya. Penjelasan subrutin ini akan mengikuti langkah-langkah sebagai berikut :

- Mengalokasi memori untuk array-array yang terpakai untuk komputasi metode PPR
- Membuat database patch yaitu jumlah elemen dan nomor elemen yang tergabung pada tiap struktur, kemudian menghitung jumlah patch yang dapat dibentuk pada suatu mesh elemen yang diberikan, jumlah elemen serta nodal di dalam setiap patch yang sudah dibentuk, dan domain dari tiap patch
- Melakukan komputasi gaya dalam pada tiap nodal dengan metode PPR

3.2.2.1 Mengalokasi Memori Array-array PPR

Karena prosedur metode PPR sebagian besar sama dengan metode SPR, array-array yang diperlukan dalam penyusunan subrutin REP adalah array-array yang sama dengan yang digunakan untuk metode SPR. Karena itu, penulis hanya memodifikasi program yang sudah dibuat peneliti terdahulu untuk mengakomodasi option PPR.

Perintah-perintah makro tersebut ditulis pada subrutin berikut ini :

```
subroutine umacr2(lct,ctl,prt)
```

KODE FORTRAN :

```
c      Set command word
      if(pcomp(uct,'mac2',4)) then
        uct = 'ssrm'
      elseif (pcomp(lct,'pc',2)) then
        ERRORESTMODE = 2
      elseif (pcomp(lct,'ui',2)) then
        ERRORESTMODE = 1
      else
        Recovery_method: select case (ES)
          case (1)
            write (*,990)
            write (*,991)
990 format ('Setting to Averaging Method on Nodal Stress/Strain')
991 format ('Please run stre,node,1,last nodal again')
            return
          case (0)
            write (*,992)
            write (*,993)
992 format ('Setting to Projection Method on Nodal Stress/Strain')
993 format ('Please run stre,node,1,last nodal again')
            return
          case (2,5)
            write (*,*) 'Setting to SPR Method on Nodal Stress/Strain'
          case (3,6,7)
            write (*,*) 'Setting to REP Method on Nodal Stress/Strain'
c$      end select Recovery_method
          case (8,9,10)
            write (*,*) 'Setting to PPR Method on Nodal Stress/Strain'
            end select Recovery_method
c
          Superconvergent_method: select case (ES)
            case (2,3,5,6,7,8,9,10)
              setvar=ualloc(19,'DUM19',numnp,1)           !SUMEL
              setvar=ualloc(30,'DUM30',numnp,1)           !SUMNOD
              setvar=ualloc(2,'DUMM2',5*numnp,1)
              setvar=ualloc(20,'DUM20',numnp,2)           ! xmax
              setvar=ualloc(21,'DUM21',numnp,2)           ! xmin
              setvar=ualloc(22,'DUM22',numnp,2)           ! ymax
              setvar=ualloc(23,'DUM23',numnp,2)           ! ymin
              setvar=ualloc(25,'DUM25',numnp*numel,1)     ! elpatch
              setvar=ualloc(26,'DUM26',numnp,1)           ! cpatch
              setvar=ualloc(27,'DUM27',numnp*numnp,1)     ! nonode
              setvar=ualloc(24,'DUM24',1,1)               ! nopatch
              setvar=ualloc(28,'DUM28',numnp,2)
              setvar=ualloc(29,'DUM29',6*6,2)
c
          Patch_model: select case (patchmodel)
            case (1) ! NBP
              call Count_The_Patch1(mr(np(33)),hr(up(3)),hr(up(4))
                & ,mr(up(19)),mr(up(2)),mr(up(24)),mr(up(25)),mr(up(26))
                & ,mr(up(27)),hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)))
              case (2) ! EBP
```

PPR

```

call Count_The_Patch2(mr(np(33)),hr(up(3)),hr(up(4))
& ,mr(up(19)),mr(up(2)),mr(up(24)),mr(up(25)),mr(up(26))
& ,mr(up(27)),hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)))
case (3) ! EIBP
call Count_The_Patch3(hr(np(43)),mr(np(33)),mr(up(24)),
& mr(up(25)),mr(up(27)),hr(up(20)),hr(up(21)),hr(up(22)),
& hr(up(23)),2)
case (4) ! EBP alternatif by CWS
call Count_The_Patch3(hr(np(43)),mr(np(33)),mr(up(24)),
& mr(up(25)),mr(up(27)),hr(up(20)),hr(up(21)),hr(up(22)),
& hr(up(23)),1)
case (5) ! NBP-PPR
call Count_The_Patch4(mr(np(33)),hr(up(3)),hr(up(4))
& ,mr(up(19)),mr(up(30)),mr(up(2)),mr(up(24)),mr(up(25))
& ,mr(up(26)),mr(up(27)),hr(up(20)),hr(up(21)),hr(up(22))
& ,hr(up(23)),hr(np(43)))

```

C

```

end select Patch_model
return
end select Superconvergent_method
endif

```

PENJELASAN :

Pemilihan metode pemulihan yang digunakan dilakukan pada perintah/*statement* seperti ‘aver’, ‘proj’, ‘scpr’, ‘rep’ dan ‘ppr’, yang ditulis dalam *file* input UI-FEAP. Perintah dalam *file* input UI-FEAP di belakang *statement* Elemen yang digunakan ‘MITC’, sehingga lengkapnya menjadi MITC <metode pemulihan> <model patch>

Dengan <metode pemulihan> berupa *statement* ‘aver’, ‘proj’, ‘scpr’, ‘rep’ dan ‘ppr’ dan <model> berupa *statement* ‘nbp’, ‘ebp’, dan ‘ibp’, masing-masing untuk *nodal based patch* dan *element based patch* dan *element interface based patch*.

Dalam subrutin ini dijelaskan bahwa :

- ‘ES=1’ : melakukan perintah perhitungan gaya dalam dengan menggunakan metode rata-rata.
- ‘ES=0’ : melakukan perintah perhitungan gaya dalam dengan menggunakan metode proyeksi.
- ‘ES=2,5’ : melakukan perintah perhitungan gaya dalam dengan menggunakan metode SPR.
- ‘ES=3,6,7’ : melakukan perintah perhitungan gaya dalam dengan menggunakan

metode REP.

- ‘ES=8’ : melakukan perintah perhitungan gaya dalam dengan menggunakan metode PPR dengan Polinomial berorde-6.
- ‘ES=9’ : melakukan perintah perhitungan gaya dalam dengan menggunakan metode PPR dengan Polinomial berorde-8.
- ‘ES=10’ : melakukan perintah perhitungan gaya dalam dengan menggunakan metode PPR dengan Polinomial berorde-9.

Kemudian perintah untuk pemilihan patch model seperti *Nodal Based Patch* dan *Element Based Patch*. Untuk Patch model 5, menggunakan *Nodal base patch* dengan subrutin *Count_The_Patch4*. *Input* yang digunakan adalah NBP2 atau nbp2.

Selanjutnya, fungsi makro tersebut untuk dalam metode PPR, adalah membuat database patch, yaitu :

- Menghitung jumlah elemen yang terbentuk dalam suatu patch. Array yang dialokasikan adalah *sumel*.
- Mencatat nomor elemen yang tergabung dalam suatu patch. Array yang dialokasikan adalah *nelneigh*.
- Menentukan berapa jumlah patch yang dapat dibentuk, jumlah elemen dan nodal pada tiap patch, serta domain dari tiap patch. Array yang dialokasikan adalah *elel, noel, cpatch, elpatch, nonode, xmin, xmax, ymin, ymax*.

3.2.2.2 Database Patch

Database Patch terdiri dari:

- Perhitungan jumlah elemen yang tergabung dalam suatu patch.
- Pencatatan nomor-nomor elemen yang tergabung dalam suatu patch.
- Menghitung jumlah patch yang dapat dibentuk pada suatu mesh struktur.
- Menentukan domain dari tiap2 patch
- Menentukan nomer2 nodal yang tergabung dalam tiap patch

Pembuatan database patch ini menggunakan subrutin yang sama dengan subrutin untuk SPR dan REP. Dalam hal ini, berkaitan dengan penelitian yang penulis

lakukan yang menggunakan metode *nodal based patch*, subrutin yang digunakan adalah *Count_The_Patch1* dan *Count_The_Patch4*.

3.2.2.3 Komputasi Gaya Dalam Nodal dengan Metode PPR

Dalam subrutin komputasi gaya dalam nodal dengan metode PPR banyak dilakukan pemanggilan pada subrutin lainnya untuk mendukung proses komputasi. Di sini akan dijelaskan lebih dulu subrutin-subrutin pendukung tersebut.

```
subroutine Cor(a,b,xmax,xmin,ymax,ymin,xw,yw)
--> pendukung
```

Subrutin ini juga merupakan subrutin yang telah dibuat untuk metode SPR dan digunakan untuk melakukan transformasi koordinat riil ke dalam sistem koordinat natural patch. Transformasi tersebut dilakukan berdasarkan persamaan berikut :

$$x = \frac{1}{2}(1-\xi)x_{min} + \frac{1}{2}(1+\xi)x_{max} \rightarrow \xi = \frac{2x - x_{min} - x_{max}}{(x_{max} - x_{min})}$$

$$y = \frac{1}{2}(1-\eta)y_{min} + \frac{1}{2}(1+\eta)y_{max} \rightarrow \eta = \frac{2y - y_{min} - y_{max}}{(y_{max} - y_{min})}$$

dimana kita mencari koordinat natural (ξ, η) berdasarkan koordinat riil (x, y) yang diketahui.

```
subroutine getMATaq(app,ij,xkj,ykj,ipoly)
--> pendukung
```

Subrutin ini juga merupakan subrutin pendukung untuk metode PPR dan digunakan untuk mendapatkan matriks $\langle P \rangle$ dalam patch yang dihitung pada setiap titik nodal patch dalam koordinat natural (x_{kj}, y_{kj}) ,

contoh untuk polinomial 6 orde :

KODE FORTRAN :

```
app(1)=1.0d0
app(2)=xkj
app(3)=ykj
app(4)=xkj*xkj
app(5)=xkj*ykj
```

```
app(6)=ykj*ykj
return
```

contoh untuk polinomial 8 orde :

KODE FORTRAN :

```
app1(1)=1.0d0
app1(2)=xkj
app1(3)=ykj
app1(4)=xkj*xkj
app1(5)=xkj*ykj
app1(6)=ykj*ykj
app1(7)=xkj*xkj*ykj
app1(8)=xkj*ykj*ykj
```

contoh untuk polinomial 9 orde :

KODE FORTRAN :

```
app2(1)=1.0d0
app2(2)=xkj
app2(3)=ykj
app2(4)=xkj*xkj
app2(5)=xkj*ykj
app2(6)=ykj*ykj
app2(7)=xkj*xkj*ykj
app2(8)=xkj*ykj*ykj
app2(9)=xkj*xkj*ykj*ykj
return
```

Matriks $[Q]$ diperoleh dari perhitungan matriks $\langle P \rangle$ pada seluruh nodal patch. Sehingga Matriks $[Q]$ berukuran $n \times 6$, dengan n adalah jumlah nodal dalam patch dan polinomial berorde 6

```
subroutine getMATnp(matn,aq,in,ipoly)
--> pendukung
```

Subrutin ini digunakan untuk mendapatkan matriks fungsi bentuk NP dalam patch yang dihitung dari matriks $[Q]$ dengan $[[Q]^T \cdot [Q]]^{-1} \cdot [Q]^T$, contoh untuk polinomial 6 orde :

KODE FORTRAN :

```

do ij=1,13
do ji=1,6
qtrans(ji,ij)=aq(ij,ji)
enddo
enddo
c Calculated Qtranspose x Q
do ij=1,6
do ji=1,6
ptpo = 0.0d0
do jk=1,13
ptpo=ptpo+qtrans(ji,jk)*aq(jk,ij)
enddo
ptp(ji,ij)=ptpo
enddo
enddo
c Invers matriks(Qtranspose x Q)
call dlinrg(6,ptp,6,iptp,6)
c Calculated matNP = invers (Qtranspose x Q) x Qtrans
do ij=1,13
do ji=1,6
ptpq = 0.0d0
do jk=1,6
ptpq=ptpq+iptp(ji,jk)*qtrans(jk,ij)
enddo
matn(ji,ij)=ptpq
enddo
enddo
return
end

subroutine geopatch1(xmax,xmin,ymax,ymin,qsi,eta,vjip,detjp,in)
--> pendukung

```

Subrutin ini digunakan untuk mendapatkan matriks invers jacobi dan determinan jacobi pada patch *in* yang ditinjau dengan mengambil domain patch *xmin*, *xmax*, *ymin* dan *ymax* pada patch *in* tersebut. Matriks invers jacobi *vjip* digunakan untuk menghitung turunan fungsi dari koordinat natural ke koordinat kartesian,

KODE FORTRAN :

```

x1=xmin(in)
x2=xmax(in)
x3=xmax(in)
x4=xmin(in)
y1=ymin(in)

```

```

y2=ymin(in)
y3=ymax(in)
y4=ymax(in)
a1=fourth*(-x1+x2+x3-x4)
a2=fourth*(-x1-x2+x3+x4)
a3=fourth*(x1-x2+x3-x4)
b1=fourth*(-y1+y2+y3-y4)
b2=fourth*(-y1-y2+y3+y4)
b3=fourth*(y1-y2+y3-y4)
c
detjp=abs(a1*b2-b1*a2+(a1*b3-b1*a3)*qsi+(b2*a3-a2*b3)*eta)
vj11=a1+a3*eta
vj12=b1+b3*eta
vj21=a2+a3*qsi
vj22=b2+b3*qsi
vjip(1,1)=+vj22/detjp
vjip(1,2)=-vj12/detjp
vjip(2,1)=-vj21/detjp
vjip(2,2)=+vj11/detjp
return

```

Subroutine COEFISIEN (xxkj,yykj,coeff,dxcoeff,dycoeff)
 --> pendukung

Subrutin ini digunakan untuk mendapatkan nilai koefisien suku-suku dari polinomial dan turunannya terhadap koordinat natural pada nodal (xxkj,yykj) dalam koordinat natural. Contoh untuk polinomial orde 6,

KODE FORTRAN :

```

coeff(1)=1.0d0
coeff(2)=xxkj
coeff(3)=yykj
coeff(4)=xxkj*xxkj
coeff(5)=xxkj*yykj
coeff(6)=yykj*yykj
c
dxcoeff(1)=0.0d0
dxcoeff(2)=1.0d0
dxcoeff(3)=0.0d0
dxcoeff(4)=2.0d0*xxkj
dxcoeff(5)=yykj
dxcoeff(6)=0.0d0
c
dycoeff(1)=0.0d0
dycoeff(2)=0.0d0
dycoeff(3)=1.0d0

```

```

dycoeff(4)=0.0d0
dycoeff(5)=1.0d0*xxkj
dycoeff(6)=2.0d0*yykj
return

```

untuk polinomial ber-orde 8 dan 9 digunakan subrutin COEFISIENp1 dan COEFISIENp2.

```

subroutine PPR_Recovery(x,ix,ox,oy,u,d,
& nopatch,elpatch,cpatch,kj,sumnod,nonode,xmax,xmin,ymax,ymin,
& scm,sct,flag,vmx,vmy,vmxy,Txz,Tyz)

```

KODE FORTRAN :

```

do 500 i=1,nopatch ! seluruh patch
xo=0.0d0 ; yo=0.0d0
do j=1,13 ! cek node pada patch
c
if((kj.eq.nonode(j,i)).and.(nonode(j,i).ne.0)) then
qkj=qkj+1.0d0
sumelp=0
do ii=1,5
if(elpatch(i,ii).ne.0) then
sumelp=sumelp+1
endif
enddo
if(sumelp.ne.3) then
sxmax=xmax(i) ; sxmin=xmin(i)
symax=ymax(i) ; symin=ymin(i)
c calculate matriks q
aq=0.0d0 ;
do ij=1,13
if(nonode(ij,i) .ne. 0) then
sx=x(1,nonode(ij,i)) ; sy=x(2,nonode(ij,i))
if(nonode(ij,i).eq.cpatch(i)) then
xkj=0.0d0 ; ykj=0.0d0
else
call Cor(sx,sy,sxmax,sxmin,symax,symin,xw,yw)
xkj=xw ; ykj=yw
endif
call getMATaq(app,ij,xkj,ykj,ipoly) !ipoly utk poly n
do ik=1,6 ; aq(ij,ik)=app(ik) ; enddo
endif
enddo ! ij matriks p tiap node pada patch
c calculate matriks Npatch
call getMATnp(matn,aq,i,ipoly)
ghbx= 0.0d0 ; ghby= 0.0d0 ; ghw=0.0d0

```

```

do ik=1,6
do jk=1,13
ghw(ik)=ghw(ik)+matn(ik,jk)*u(1,nonode(jk,i))
ghbx(ik)=ghbx(ik)+matn(ik,jk)*u(2,nonode(jk,i))
ghby(ik)=ghby(ik)+matn(ik,jk)*u(3,nonode(jk,i))
enddo
enddo
      sx=x(1,kj) ; sy=x(2,kj)
      sxmax=xmax(i) ; sxmin=xmin(i)
      symax=yax(i) ; symin=ymin(i)
      call Cor(sx,sy,sxmax,sxmin,symax,symin,xw,yw)
      xxkj=xw ; yykj=yw
c   calculate gradien displacement
      call COEFISIEN (xxkj,yykj,coeff,dxcoeff,dycoeff)
      dxbx=0.0d0 ; dybx=0.0d0 ; dxby=0.0d0 ; dyby=0.0d0
do ik=1,6
      dxbx=dxbx+ghbx(ik)*dxcoeff(ik)
      dybx=dybx+ghbx(ik)*dycoeff(ik)
      dxby=dxby+ghby(ik)*dxcoeff(ik)
      dyby=dyby+ghby(ik)*dycoeff(ik)
enddo
      call geopatch1(xmax,xmin,ymax,ymin,xxkj,yykj,vjip,i)
      dxtbx=vjip(1,1)*dxbx+vjip(1,2)*dybx
      dytbx=vjip(2,1)*dxbx+vjip(2,2)*dybx
      dxtby=vjip(1,1)*dxby+vjip(1,2)*dyby
      dytby=vjip(2,1)*dxby+vjip(2,2)*dyby
c   calculate gradient recovery PPR (Mxx,Myy,Mxy)
      momen(kj,1)=momen(kj,1)+ hb11*dxtbx+hb12*dytby
      momen(kj,2)=momen(kj,2)+ hb12*dxtbx+hb22*dytby
      momen(kj,3)=momen(kj,3)+ hb33*(dxtby+dytby)
      else
c for 3 element use Stage 2 : (Same as SPR on the Gauss Nodal every element
patch)
      if(sumelp.eq.3) then
      qkj=qkj+1.0d0
c
      call PPR_RecoveryLIN2(hr(np(43)),hr(up(7)),hr(up(8)),hr(up(9))
&      ,hr(up(10)),hr(up(11)),kj,mr(up(24)),mr(up(25)),mr(up(27))
&      ,hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)),momen,i)
      endif
      endif
      endif ! check node kj dlm patch
      enddo !j, seluruh node dlm patch
500 continue ! ii = seluruh patch
c
      scm(1)=momen(kj,1)/qkj
      scm(2)=momen(kj,2)/qkj
      scm(3)=momen(kj,3)/qkj

```

```

      sct(1)=0.0d0 ; sct(2)=0.0d0
      return
      end

```

PENJELASAN :

Subrutin ini berfungsi untuk membentuk matriks $\langle P \rangle$ dan $[Q]$ yang untuk selanjutnya digunakan untuk membentuk matriks $[A_p]$ dan $[N_p]$ (pers.....) yang kemudian digunakan menghitung kelengkungan/gradien perpindahan pada nodal tiap elemen.

Subrutin ini akan terus dipanggil oleh subrutin NoForDKMQ_4 ketika UI-FEAP melakukan *looping* seluruh elemen untuk menghitung gaya dalam nodal struktur. Ketika melakukan *loop* tiap elemen tersebut, dilakukan pula *loop* tiap nodal pada elemen tersebut. Nodal yang ditinjau diwakili dengan nomor nodal global yang ditransfer dari subrutin NoForDKMQ_4 ke subrutin PPR_Recovery melalui variabel k.j.

Subrutin PPR_Recovery ini akan dijelaskan tahap demi tahap. Dalam tiap tahapnya, struktur *looping do* keseluruhan tetap ditampilkan untuk tetap memberikan gambaran global dari keseluruhan subrutin.

KODE FORTRAN

```

      do 500 i=1,nopatch ! seluruh patch
      xo=0.0d0 ; yo=0.0d0
      do j=1,13 ! cek node pada patch
c
      if((kj.eq.nonode(j,i)).and.(nonode(j,i).ne.0)) then
          qkj=qkj+1.0d0
          sumelp=0
          do ii=1,5
              if(elpatch(i,ii).ne.0) then
                  sumelp=sumelp+1
              endif
          enddo
c Pemilihan metode gradient berdasarkan jumlah elemen dalam patch
c untuk jumlah elemen tidak sama dengan 3, menggunakan metode PPR standar
c untuk jumlah elemen 3, Metode PPR dengan menggunakan SPR pada titik Gaus
c dalam setiap elemen.

```

```

if(sumelp.ne.3) then
....
else
call PPR_RecoveryLIN2(hr(np(43)),hr(up(7)),hr(up(8)),hr(up(9))
& ,hr(up(10)),hr(up(11)),kj,mr(up(24)),mr(up(25)),mr(up(27))
& ,hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)),momen,i)

endif
endif
endif ! check node kj dlm patch
enddo !j, seluruh node dlm patch
500 continue ! ii = seluruh patch
c
scm(1)=momen(kj,1)/qkj
scm(2)=momen(kj,2)/qkj
scm(3)=momen(kj,3)/qkj
sct(1)=0.0d0 ; sct(2)=0.0d0
return
end

```

PENJELASAN

Pada setiap elemen akan dilakukan perhitungan pemulihan gaya dalam pada nodal dalam elemen tersebut yang ditandai dengan nodal kj pada elemen tersebut. Nodal kj dicek terhadap setiap nodal pembentuk patch, yang di looping pada keseluruhan patch yang ada. Proses looping terhadap seluruh patch yang sudah dibentuk oleh subrutin `Count_the_patch1` atau 4. Di sini dilakukan pengecekan pada tiap nodal dalam patch tersebut untuk mengetahui apakah nodal yang sedang ditinjau ada dalam patch tersebut. Jika nodal yang ditinjau ada dalam patch, proses dilanjutkan ke pengecekan jumlah elemen dalam patch yang ditinjau. Jika tidak ada, looping dilanjutkan ke patch berikutnya.

Pada Patch yang ditinjau dimana terdapat nodal kj sebagai nodal patchnya, dilakukan pengecekan jumlah elemen untuk pemilihan metode pemulihan gaya dalam PPR yaitu metode PPR pada patch berelemen tidak sama dengan 3 dan metode PPR pada patch berelemen 3. Metode PPR pada patch berelemen tidak sama dengan 3, akan diuraikan lebih lanjut. Metode PPR untuk patch berelemen 3 sama dengan metode SPR, dalam perhitungannya dilakukan pada subrutin `PPR_RecoveryLIN2`. Subrutin `PPR_RecoveryLIN2` ini dimodifikasi dari metode SPR yang dilakukan peneliti sebelumnya, dan dalam tesis ini tidak dijelaskan lagi.

Selanjutnya dari perhitungan pemulihan gaya dalam dengan metode PPR diperoleh Momen $\{M\}$. Momen $\{M\}$ tersebut diakumulasi untuk semua patch yang mengandung nodal yang sedang ditinjau, untuk kemudian dibagi dengan jumlah overlapping yang terjadi.

KODE FORTRAN

```

do 500 i=1,nopatch ! seluruh patch
  xo=0.0d0 ; yo=0.0d0
  do j=1,13 ! cek node pada patch
c
    if((kj.eq.nonode(j,i)).and.(nonode(j,i).ne.0)) then
      .....
      if(sumelp.ne.3) then
c
        sxmax=xmax(i) ; sxmin=xmin(i)
        symax=yax(i) ; symin=ymin(i)
c
        calculate matriks q
        aq=0.0d0 ;
        do ij=1,13
          if(nonode(ij,i) .ne. 0) then
            sx=x(1,nonode(ij,i)) ; sy=x(2,nonode(ij,i))
            call Cor(sx,sy,sxmax,sxmin,symax,symin,xw,yw)
            xkj=xw ; ykj=yw
            call getMATaq(app,ij,xkj,ykj,ipoly) !ipoly utk poly n
            do ik=1,6 ; aq(ij,ik)=app(ik) ; enddo
          endif
        enddo ! ij matriks p tiap node pada patch
c
        calculate matriks Npatch
        call getMATnp(matn,aq,i,ipoly)
        ghbx= 0.0d0 ; ghby= 0.0d0 ; ghw=0.0d0
        do ik=1,6
          do jk=1,13
            ghw(ik)=ghw(ik)+matn(ik,jk)*u(1,nonode(jk,i))
            ghbx(ik)=ghbx(ik)+matn(ik,jk)*u(2,nonode(jk,i))
            ghby(ik)=ghby(ik)+matn(ik,jk)*u(3,nonode(jk,i))
          enddo
        enddo

```

PENJELASAN

berfungsi untuk membentuk matriks $[P]$ dan $[Q]$ yang untuk selanjutnya digunakan untuk membentuk matriks $[A_p]$ yang kemudian untuk menghitung matriks $\{a\}$ untuk perpindahan arah z, rotasi x dan rotasi y yaitu ghw, ghbx dan ghby.

KODE FORTRAN

```

sx=x(1,kj) ; sy=x(2,kj)
sxmax=xmax(i) ; sxmin=xmin(i)
symax=ymax(i) ; symin=ymin(i)
call Cor(sx,sy,sxmax,sxmin,symax,symin,xw,yw)
xxkj=xw ; yykj=yw
c calculate gradien displacement
call COEFISIEN (xxkj,yykj,coeff,dxcoeff,dycoeff)
dxbx=0.0d0 ; dybx=0.0d0 ; dxby=0.0d0 ; dyby=0.0d0
do ik=1,6
dxbx=dxbx+ghbx(ik)*dxcoeff(ik)
dybx=dybx+ghbx(ik)*dycoeff(ik)
dxby=dxby+ghby(ik)*dxcoeff(ik)
dyby=dyby+ghby(ik)*dycoeff(ik)
enddo

```

PENJELASAN

Menghitung kelengkungan/gradien perpindahan rotasi x dan rotasi y pada nodal kj yang ditinjau. Kelengkungan ini sebagai gradien perpindahan nodal kj pada koordinat natural (xxkj,yykj)

KODE FORTRAN

```

call geopatch1(xmax,xmin,ymax,ymin,xxkj,yykj,vjip,i)
dxtbx=vjip(1,1)*dxbx+vjip(1,2)*dybx
dytbx=vjip(2,1)*dxbx+vjip(2,2)*dybx
dxtby=vjip(1,1)*dxby+vjip(1,2)*dyby
dytby=vjip(2,1)*dxby+vjip(2,2)*dyby
c calculate gradient recovery PPR (Mxx,Myy,Mxy)
momen(kj,1)=momen(kj,1)+ hb11*dxtbx+hb12*dytby
momen(kj,2)=momen(kj,2)+ hb12*dxtbx+hb22*dytby
momen(kj,3)=momen(kj,3)+ hb33*(dxtby+dytbx)

```

PENJELASAN :

berfungsi untuk menghitung kelengkungan/gradien perpindahan rotasi x dan rotasi y dalam koordinat kartesian. Transformasi ini dengan menggunakan invers matriks jacobi. Gaya dalam Momen yang dipulihkan dihitung berdasarkan hubungan matriks Hook dan kelengkungan. Gaya dalam Momen ini pada nodal kj yang ditinjau pada setiap patch yang mengandung nodal kj.

subrutin NoForDKMQ_4 (x,d,u,np,ix,ndm,nel,dt, sigproj)

Subrutin ini termasuk di dalam subrutin *elem09* yang berfungsi untuk menyimpan nilai gaya dalam nodal berdasarkan metode PPR. Jadi dalam subrutin ini akan selalu dipanggil subrutin PPR_Recovery. Penulis memilih untuk membentuk subrutin baru (NoForDKMQ_4) agar berbeda dari subrutin SPR dan REP yang menggunakan subrutin NoForDKMQ_3.

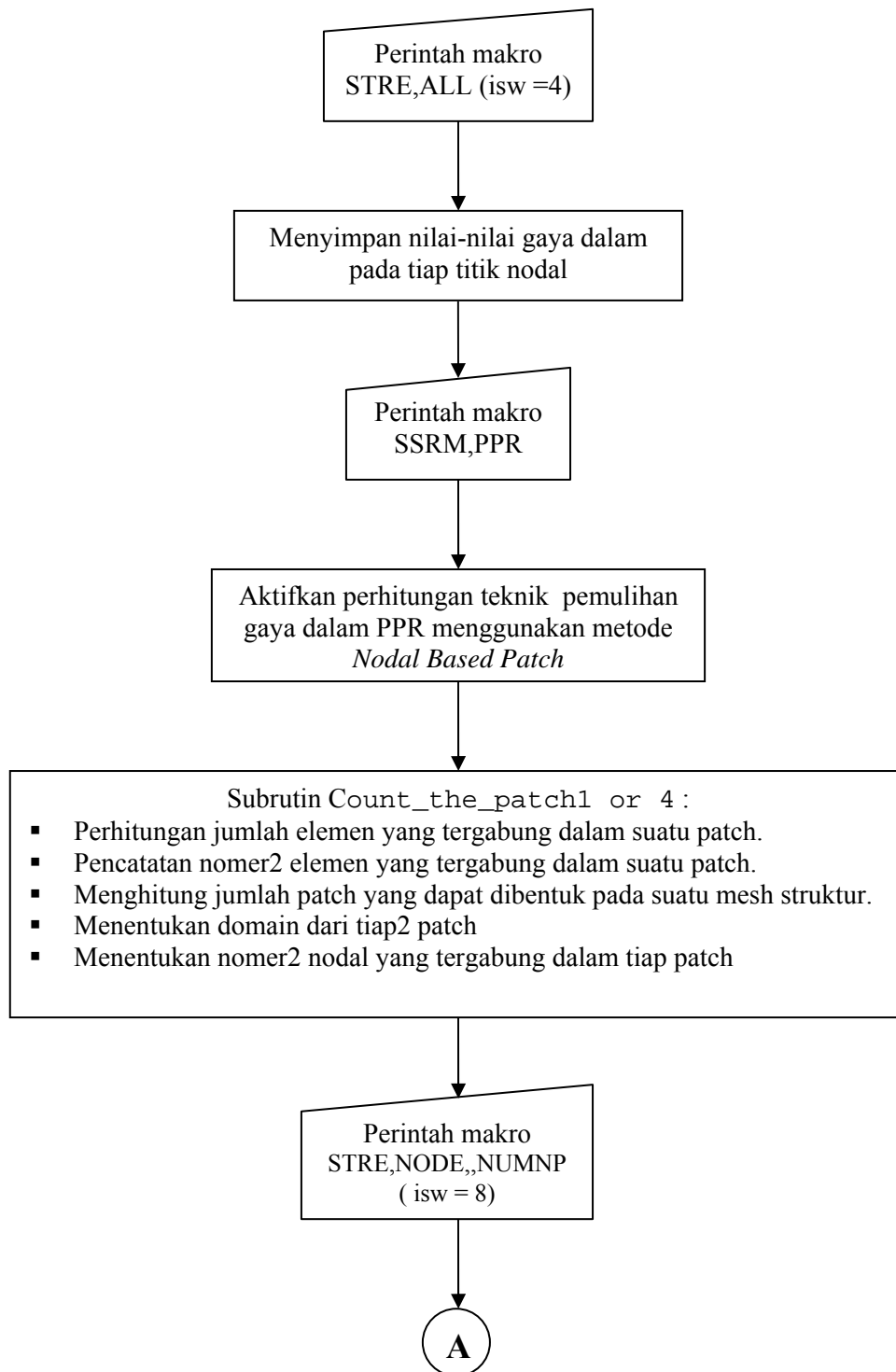
KODE FORTRAN :

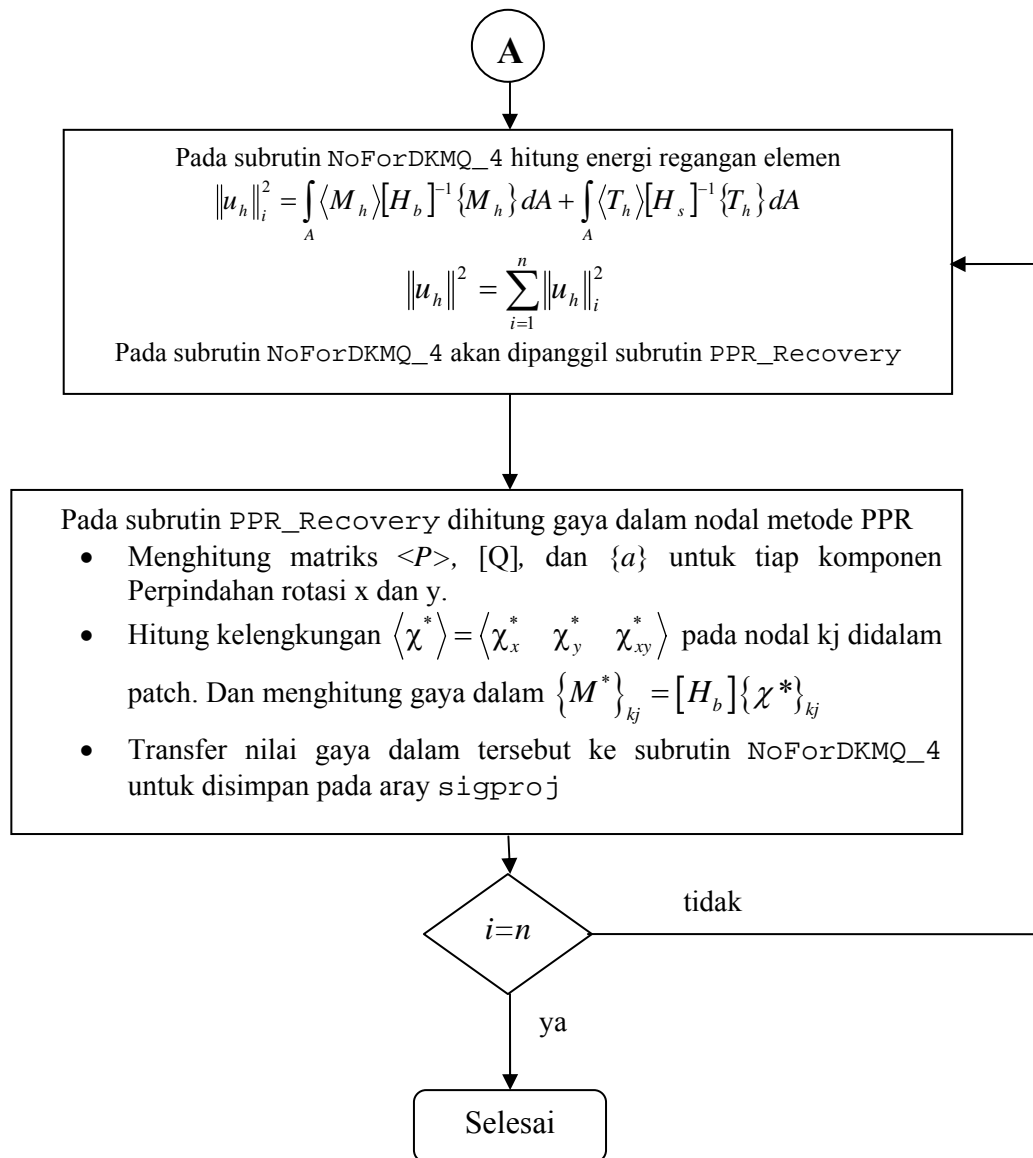
```

      do 8000 j=1,4
         k=abs(ix(j))
      if(k.gt.0) then
         if(sigproj(k,1) .ne. 0.0) then
            goto 8000
         else
            Recovery_method: select case (ES)
      C polynomial 6 orde
            case (8)
               call PPR_Recovery(hr(np(43)),hr(np(33)),hr(up(3)),
                & hr(up(4)),hr(np(40)),d,
                & mr(up(24)),mr(up(25)),mr(up(26)),k,mr(up(34)),mr(up(27)),
                & hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)),
                & scm,sct,flag,hr(up(7)),hr(up(8)),
                & hr(up(9)),hr(up(10)),hr(up(11)))
      C polynomial 8 orde
            case (9)
               call PPR_RecoveryP1(hr(np(43)),hr(np(33)),hr(up(3)),
                & hr(up(4)),hr(np(40)),d,
                & mr(up(24)),mr(up(25)),mr(up(26)),k,mr(up(30)),mr(up(27)),
                & hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)),
                & scm,sct,flag,hr(up(7)),hr(up(8)),
                & hr(up(9)),hr(up(10)),hr(up(11)))
      C polynomial 9 orde
            case (10)
               call PPR_RecoveryP2(hr(np(43)),hr(np(33)),hr(up(3)),
                & hr(up(4)),hr(np(40)),d,
                & mr(up(24)),mr(up(25)),mr(up(26)),k,mr(up(30)),mr(up(27)),
                & hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)),
                & scm,sct,flag,hr(up(7)),hr(up(8)),
                & hr(up(9)),hr(up(10)),hr(up(11)))
      c
            end select Recovery_method
      c
            dt(k)=1.0
            sigproj(k,1)=scm(1)
            sigproj(k,4)=scm(3)
            sigproj(k,3) =0.d0
            sigproj(k,2)=scm(2)
            sigproj(k,5)=sct(2)
            sigproj(k,6)=sct(1)
            endif
         endif
      8001 continue

      return
      end

```





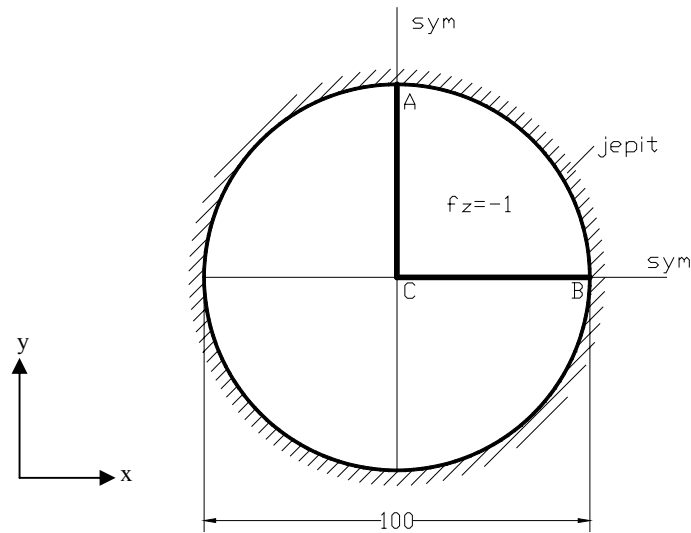
Gambar 3.2 Diagram Alir Perhitungan Gaya Dalam dan Energi Regangan Elemen dengan Menggunakan Metode PPR pada UI-FEAP

3.2.2.4 Komputasi Indikator Error Relatif Parsial Lentur

UI-FEAP sudah memiliki subrutin penghitungan indikator error relatif struktur yang dibuat oleh peneliti terdahulu. Dalam penelitian ini dibutuhkan nilai indikator error relatif parsial untuk lentur saja sesuai persamaan (2.61) dan (2.62).

3.2.3 Contoh Data Input Analisa Metode PPR dan Error Estimasi

Struktur pelat yang akan dianalisis :



Data :

$$E = 1000 ; \nu = 0,3$$

$$R = 50$$

Beban terbagi rata : $f_z = -1$

Kondisi batas :

$$\text{Jepit : } w = \beta_x = \beta_y = 0$$

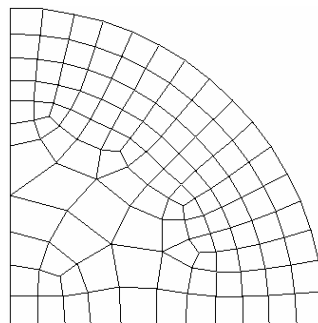
Kondisi simetri:

$$\text{Sisi } CB = \beta_y = 0$$

$$\text{Sisi } CA = \beta_x = 0$$

Gambar 3.3 Contoh Analisa Struktur Lingkaran dengan UI-FEAP

Pelat tersebut dianalisa $\frac{1}{4}$ bagian saja dan dimodelisasi dengan mesh adaptif 110 elemen sebagai berikut :



Gambar 3.4 Mesh Adaptif

File data input adalah sebagai berikut :

```
feap**
130,110,1,2,3,4

mate,1
user 9
MITC PPR NBP2
quadrature 2 2
elas isot 1000 0.3
thickness data 1
load data -1
```

```

coord
1      0      60      10
2      0      10      60
.      .      .      .
.      .      .      .
129    0      10      55.89663189
130    0      33.67090302  54.04189313

bound
1      0      1      1      1
2      0      1      1      1
3      0      1      1      1
4      0      1      1      1
5      0      1      1      1
6      0      0      1      1
7      0      0      1      0
8      0      0      1      0
9      0      0      1      0
10     0      0      0      1
11     0      0      0      1
12     0      0      0      1
37     0      1      1      1
38     0      0      0      1
39     0      1      1      1
40     0      0      0      1
41     0      0      1      0
42     0      1      1      1
43     0      0      1      0
44     0      1      1      1
113    0      0      0      1
114    0      0      0      1
115    0      1      1      1
116    0      1      1      1
117    0      0      0      1
118    0      0      0      1
119    0      0      1      0
120    0      0      1      0
121    0      1      1      1
122    0      1      1      1
123    0      1      1      1
124    0      0      0      1
125    0      1      1      1
126    0      0      1      0
127    0      0      1      0
128    0      1      1      1
129    0      0      1      0
130    0      1      1      1

elem
1      0      1      112    110    17      83
2      0      1      110    111    74      17
.      .      .      .      .      .      .
.      .      .      .      .      .      .
109    0      1      35     89     87     26
110    0      1      111    88     89     35

end
interactive
stop

```

```
feap**
```

Solution date: Mon Dec 14 18:48:48 2009

UNIX/PC 7.1b

- 12/15/98 -

Input Data Filename: iadlsnd3.txt

```

Number of Nodal Points - - - - - :      130
Number of Elements   - - - - - :      110

Spatial Dimension of Mesh - - - - - :          2
Degrees-of-Freedom/Node (Maximum) - :          3
Number Element Nodes (Maximum) - :          4

Number of Material Sets - - - - - :          1
Number Parameters/Set (Program) - :         200
Number Parameters/Set (Users ) - :          50

```

feap**

M a t e r i a l P r o p e r t i e s

Material Set 1 for User Element Type 9

Degree of Freedom Assignments	Local Number	Global Number
	1	1
	2	2
	3	3

U s e r 4 : P l a t e B e n d i n g

M e c h a n i c a l P r o p e r t i e s

Plate & Shell Analysis

Modulus E 1.00000E+03
Poisson ratio 3.00000E-01

Thickness 1.00000E+00
Loading - q -1.00000E+00
Thickness 1.00000E+00
Loading - q -1.00000E+00
Density 0.00000E+00

1-Gravity 0.00000E+00
2-gravity 0.00000E+00
3-gravity 0.00000E+00

Formulation : Small deformation.
 Plate/Shell : Kappa 8.33333E-01

Material density is zero.

```
=====
MIXED INTERPOLATED TENSORIAL COMPONENTS PLATE BENDING
=====
modulus Young E = 0.10000E+04
poisson ratio p = 0.30000
thickness h = 0.10000E+01
massa jenis = 0.00000E+00
mass option = 3
distr.load fz = -0.10000E+01
kind of fz = 1
output option = 0
smoothed forces = 8Polynomial preserving Recovery Method
Patch type =Nodal Based Patch-PPR
```

feap**

Nodal Coordinates

node	1 Coord	2 Coord
1	6.000E+01	1.000E+01
2	1.000E+01	6.000E+01
3	5.615E+01	2.925E+01
4	4.536E+01	4.536E+01
5	2.935E+01	5.611E+01
6	1.000E+01	1.000E+01
7	1.000E+01	4.289E+01
8	1.000E+01	3.231E+01
9	1.000E+01	2.178E+01
10	2.212E+01	1.000E+01
11	3.322E+01	1.000E+01
.		
.		
.		
122	4.860E+01	4.179E+01
123	5.396E+01	3.382E+01
124	5.777E+01	2.475E+01
125	5.541E+01	1.000E+01
126	5.975E+01	1.499E+01
127	1.000E+01	4.752E+01
128	1.520E+01	5.973E+01
129	1.000E+01	5.590E+01
130	3.367E+01	5.404E+01

feap**

N o d a l B . C .

Node	1-b.c.	2-b.c.	3-b.c.
1	1	1	1
2	1	1	1
3	1	1	1
.			
.			
11	0	0	1
12	0	0	1
.			
.			
119	0	1	0
120	0	1	0

feap**

E l e m e n t s

Elmt	Mat	Reg	1 Node	2 Node	3 Node	4 Node
1	1	0	114	112	17	85
2	1	0	112	113	76	17
3	1	0	80	111	115	38
4	1	0	29	77	110	109
.						
.						
104	1	0	72	70	108	25
105	1	0	109	12	115	111
106	1	0	71	72	25	27
107	1	0	26	107	71	27
108	1	0	91	89	107	26
109	1	0	40	12	109	110
110	1	0	78	76	113	30
111	1	0	77	78	30	31

P a r t i t i o n 1

E q u a t i o n / P r o b l e m S u m m a r y :

Space dimension (ndm) =	2	Number dof (ndf) =	3
Number of equations =	351	Number nodes =	130
Average col. height =	142	Number elements =	111
Number profile terms =	49766	Number materials =	1
Number rigid bodies =	0	Number joints =	0
Est. factor time-sec =	1.2740E+02		

Material	Element Tag	Element Type	History Terms	Element Terms
1	1	9	0	12

feap**

```

Macro Instructions

Macro Statement      Variable 1  Variable 2  Variable 3
  tang              1.0000E+00  0.0000E+00  0.0000E+00
  stre all          0.0000E+00  0.0000E+00  0.0000E+00
  ssrc rep          0.0000E+00  0.0000E+00  0.0000E+00
  stre node         0.0000E+00  1.3000E+02  0.0000E+00
  ssrc ui           0.0000E+00  0.0000E+00  0.0000E+00
*Macro 1 * tang      v: 1.00      0.00      0.00
                                     t= 0.11      0.00
Residual norm =      1.8394778E+02  1.0000000E+00  t= 0.12      0.00
Condition check: D-max 0.3658E+04; D-min 0.1523E+01; Ratio 0.2403E+04
Maximum no. diagonal digits lost: 3
End Triangular Decomposition          t= 0.17      0.00
Number of operations = 250275 plus      9 Mega-ops
Time: CPU = 0.17 , System = 0.00
--> SOLVE AT 185.01 Mflops. Time= 0.05
Energy convergence test
Maximum = 3.882991649582296E+06 Current = 3.882991649582296E+06
Relative = 1.000000000000000E+00 Tolerance = 1.000000000000000E-16
*Macro 2 * stre all  v: 0.00      0.00      0.00
                                     t= 0.17      0.00

feap**

```

```

FORCES ON GAUSS POINTS OF ELEMENT
=====

```

ELMT	X	Y	Mx	My	Mxy	Txz	Tyz
1	42.586	27.225	-.2718E+03	-.3357E+03	0.4432E+02	0.1670E+02	0.8411E+01
1	40.691	26.396	-.2762E+03	-.3438E+03	0.4723E+02	0.1716E+02	0.7355E+01
1	41.657	24.784	-.2725E+03	-.3466E+03	0.4577E+02	0.1773E+02	0.7696E+01
1	43.561	25.427	-.2682E+03	-.3379E+03	0.4287E+02	0.1737E+02	0.8773E+01
2	39.397	25.765	-.3164E+03	-.3695E+03	0.3548E+02	0.1697E+02	0.7032E+01
2	37.756	24.868	-.3202E+03	-.3727E+03	0.3790E+02	0.1649E+02	0.7899E+01
2	38.980	23.806	-.3156E+03	-.3736E+03	0.3930E+02	0.1583E+02	0.7132E+01
2	40.429	24.349	-.3116E+03	-.3695E+03	0.3607E+02	0.1610E+02	0.6402E+01
3	49.960	13.312	-.2006E+03	-.3272E+03	0.8806E+01	0.1913E+02	0.9814E+00
3	47.481	13.205	-.2042E+03	-.3393E+03	0.9061E+01	0.1913E+02	0.9261E+00
3	47.546	10.859	-.2041E+03	-.3395E+03	0.4974E+01	0.1922E+02	0.9287E+00
3	50.046	10.887	-.2005E+03	-.3276E+03	0.4853E+01	0.1922E+02	0.9848E+00
4	41.453	16.855	-.3180E+03	-.3926E+03	0.1719E+02	0.1482E+02	0.3039E+01
4	39.095	16.893	-.3208E+03	-.4021E+03	0.1635E+02	0.1483E+02	0.3228E+01
4	38.988	14.738	-.3227E+03	-.4023E+03	0.1348E+02	0.1488E+02	0.3226E+01
4	41.457	14.702	-.3199E+03	-.3932E+03	0.1428E+02	0.1487E+02	0.3039E+01
5	21.889	40.745	-.3683E+03	-.2929E+03	0.3351E+02	0.7340E+01	0.1685E+02
5	23.166	40.905	-.3685E+03	-.2911E+03	0.3566E+02	0.7320E+01	0.1701E+02

```

5 23.366 42.210 -.3618E+03 -.2894E+03 0.3606E+02 0.6860E+01 0.1708E+02
5 21.856 42.364 -.3629E+03 -.2912E+03 0.3416E+02 0.6836E+01 0.1684E+02
6 31.018 35.419 -.3366E+03 -.3200E+03 0.4928E+02 0.1034E+02 0.1142E+02
6 32.269 34.648 -.3345E+03 -.3224E+03 0.5031E+02 0.1065E+02 0.1192E+02
6 33.351 35.449 -.3305E+03 -.3186E+03 0.4809E+02 0.1049E+02 0.1213E+02
6 32.054 36.518 -.3328E+03 -.3171E+03 0.4750E+02 0.1009E+02 0.1165E+02
7 24.735 54.285 -.2262E+03 -.6434E+02 0.6429E+02 0.8386E+01 0.2256E+02
.
.
.
108 31.803 32.465 -.3573E+03 -.3467E+03 0.4342E+02 0.1112E+02 0.1070E+02
108 31.854 33.848 -.3518E+03 -.3451E+03 0.4343E+02 0.1049E+02 0.1072E+02
108 30.435 34.480 -.3547E+03 -.3462E+03 0.4122E+02 0.1009E+02 0.9824E+01
108 29.878 33.016 -.3587E+03 -.3474E+03 0.4170E+02 0.1079E+02 0.9559E+01
109 38.871 10.835 -.3268E+03 -.4090E+03 0.3401E+01 0.1510E+02 0.8805E+00
109 41.450 10.827 -.3239E+03 -.3995E+03 0.3808E+01 0.1510E+02 0.1145E+01
109 41.456 13.087 -.3229E+03 -.3992E+03 0.6818E+01 0.1493E+02 0.1146E+01
109 38.928 13.115 -.3258E+03 -.4089E+03 0.6375E+01 0.1492E+02 0.8849E+00
110 37.938 21.675 -.3428E+03 -.3913E+03 0.3132E+02 0.1259E+02 0.6767E+01
110 38.312 23.090 -.3370E+03 -.3901E+03 0.3154E+02 0.1405E+02 0.6382E+01
110 36.846 23.936 -.3410E+03 -.3933E+03 0.2905E+02 0.1463E+02 0.7394E+01
110 36.001 22.288 -.3450E+03 -.3938E+03 0.2925E+02 0.1305E+02 0.8207E+01
111 37.460 18.518 -.3593E+03 -.4081E+03 0.2260E+02 0.1317E+02 0.5428E+01
111 37.710 20.450 -.3542E+03 -.4073E+03 0.2448E+02 0.1226E+02 0.5547E+01
111 35.516 20.933 -.3576E+03 -.4147E+03 0.2180E+02 0.1223E+02 0.5431E+01
111 35.037 18.879 -.3621E+03 -.4148E+03 0.2027E+02 0.1314E+02 0.5218E+01
*Macro 3 * ssrc rep v: 0.00 0.00 0.00
t= 0.19 0.00

```

P A T C H P R O P E R T I E S

Patch type: Nodal Based Patch

N u m b e r o f E l e m e n t M a k i n g U p T h e P a t c h e s

node	number of patches
1	1
2	1
3	2
4	2
5	2
6	1
7	2
8	2
9	2
10	2
11	2
12	2
13	4
14	4

15	4
16	4
17	4
18	4
19	4
20	4
21	4
22	4
23	4

.

.

.

121	2
122	2
123	2
124	2
125	2
126	2
127	2
128	2
129	2
130	2

E l e m e n t M a k i n g u p T h e P a t c h e s

node	element number on each patches
1	79, 0, 0, 0, 0
2	89, 0, 0, 0, 0
3	69, 75, 0, 0, 0
4	8, 65, 0, 0, 0
5	7, 98, 0, 0, 0
6	61, 0, 0, 0, 0
7	99,100, 0, 0, 0
8	40,101, 0, 0, 0
9	12, 62, 0, 0, 0
10	11, 57, 0, 0, 0
.	
.	
.	
119	11, 61, 0, 0, 0
120	61, 62, 0, 0, 0
121	12,101, 0, 0, 0
122	65, 66, 0, 0, 0
123	68, 69, 0, 0, 0
124	75, 76, 0, 0, 0
125	18, 79, 0, 0, 0
126	78, 79, 0, 0, 0
127	85, 99, 0, 0, 0
128	24, 89, 0, 0, 0
129	88, 89, 0, 0, 0
130	27, 98, 0, 0, 0

- - > P A T C H P R O P E R T I E S < - -

N u m b e r o f P a t c h 94														
p a t c h	e	l	e	m	e	n	o	n	p	a	t	c	h	nodes as center of patch
1	9	37	38	63	0									13
2	6	58	70	90	0									14
3	28	39	40	101	0									15
4	30	41	42	102	0									16
5	1	2	71	80	0									17
6	5	81	82	94	0									18
7	22	51	95	97	0									19
8	21	50	91	93	0									20
9	25	54	59	64	0									21
10	26	55	92	96	0									22
.														
.														
82	10	43	44	60	0									102
83	10	58	59	60	0									103
84	9	41	57	102	0									104
85	8	54	55	56	0									105
86	7	51	52	53	0									106
87	6	50	107	108	0									107
88	5	49	103	104	0									108
89	4	47	105	109	0									109
90	4	30	42	109	0									110
91	3	47	48	105	0									111
92	1	2	43	46	0									112
93	2	36	46	110	0									113

```

94          1  43  44  45  0          114

- - > P A T C H   P R O P E R T I E S < - -

  D o m a i n   o f   E v e r y   P a t c h
p a t c h      x_max x_min y_max y_min
  1            30.384, 16.224, 27.506, 14.626

  2            38.951, 30.187, 39.606, 31.767

  3            22.582, 10.000, 39.275, 27.488

  4            38.286, 27.947, 19.151, 10.000

  5            45.558, 36.682, 28.211, 20.964

  6            28.050, 21.261, 46.410, 38.947

  7            31.604, 21.649, 53.768, 43.882

  8            34.050, 25.097, 43.882, 35.259

  9            45.203, 34.050, 45.309, 34.436

 10            38.931, 28.050, 50.321, 39.606
  .
  .
  .
 80            28.942, 16.104, 21.274, 10.000

 81            23.158, 10.000, 21.781, 10.000

 82            46.421, 35.878, 36.575, 26.659

 83            44.374, 34.227, 39.556, 29.595

 84            33.944, 22.125, 20.150, 10.000

 85            45.355, 33.110, 51.573, 39.556

 86            29.346, 18.506, 58.960, 48.720

 87            34.227, 25.493, 39.320, 31.767

 88            25.097, 17.905, 43.172, 35.348

 89            46.651, 37.900, 17.906, 10.000

 90            42.398, 33.218, 18.242, 10.000

```

```

91          50.995, 42.300, 18.407, 10.000

92          44.617, 32.924, 31.369, 23.266

93          41.312, 30.384, 29.595, 20.964

94          48.183, 37.465, 33.291, 23.983

```

N o d e s o n E v e r y P a t c h

n o d e

```

13, 48, 47, 98, 99,101,100, 46,104, 0, 0, 0, 0,- - >belong to patch 1
14, 58, 57, 60,107, 89, 91,103, 88, 0, 0, 0, 0,- - >belong to patch 2
15, 28, 73, 41, 74, 8,121, 45, 99, 0, 0, 0, 0,- - >belong to patch 3
16, 77, 31,104, 46,118, 11,110, 40, 0, 0, 0, 0,- - >belong to patch 4
17, 85,114,113,112, 76, 78, 87, 75, 0, 0, 0, 0,- - >belong to patch 5
18, 56, 53, 67, 69,108, 70, 54, 72, 0, 0, 0, 0,- - >belong to patch 6
19, 52, 49, 68,106, 69, 53, 50, 56, 0, 0, 0, 0,- - >belong to patch 7
20, 60, 55, 54, 56, 72, 71, 57,107, 0, 0, 0, 0,- - >belong to patch 8
21, 95, 60, 59,105, 58,103, 97, 96, 0, 0, 0, 0,- - >belong to patch 9
22,105, 51, 50, 52, 56, 55, 59, 60, 0, 0, 0, 0,- - >belong to patch 10
.
.
.
100, 46, 13,101, 98,119, 10,104,118, 0, 0, 0, 0,- - >belong to patch 80
101, 13, 98,120, 9, 6,119,100, 10, 0, 0, 0, 0,- - >belong to patch 81
102, 36, 96, 88,103, 34,112, 93,114, 0, 0, 0, 0,- - >belong to patch 82
103, 96, 21, 14, 58, 88, 34, 36,102, 0, 0, 0, 0,- - >belong to patch 83
104, 31, 46,100, 13, 10,118, 16, 11, 0, 0, 0, 0,- - >belong to patch 84
105, 4,117, 51, 44, 22, 59, 95, 21, 0, 0, 0, 0,- - >belong to patch 85
106, 5,116, 62, 42, 24, 68, 49, 19, 0, 0, 0, 0,- - >belong to patch 86
107, 14, 57, 71, 20, 27, 26, 89, 91, 0, 0, 0, 0,- - >belong to patch 87
108, 70, 18, 23, 67, 73, 28, 72, 25, 0, 0, 0, 0,- - >belong to patch 88
109,111, 86,110, 77, 29, 40,115, 12, 0, 0, 0, 0,- - >belong to patch 89
110, 29, 77, 16, 31, 11, 40,109, 12, 0, 0, 0, 0,- - >belong to patch 90
111, 80, 33, 29, 86,109, 12, 38,115, 0, 0, 0, 0,- - >belong to patch 91
112,114,102, 90, 34,113, 76, 85, 17, 0, 0, 0, 0,- - >belong to patch 92
113,112, 34, 90, 48, 30, 17, 76, 78, 0, 0, 0, 0,- - >belong to patch 93
114, 93, 36, 34,102,112, 17, 32, 85, 0, 0, 0, 0,- - >belong to patch 94
*Macro 4 * stre node v: 0.00 130. 0.00
t= 0.22 0.00

```

feap**

F o r c e s o n N o d e s a f t e r R e c o v e r y

```

Node 1-Pr.Value 2-Pr.Value 3-Pr.Value 1-Pr.Angle
      I_1 Value  J_2 Value  J_3 Value
      1 Value   2 Value   3 Value   4 Value

```

```

1 -1.8201E+00 -2.2112E+02 0.0000E+00 4.2301E-01

```



```

-7.4312E+01  1.2714E+02 -5.4834E+05
-1.8320E+00 -2.2110E+02  0.0000E+00  1.6190E+00

2 -1.8237E+00 -2.2104E+02  0.0000E+00  8.9625E+01
-7.4288E+01  1.2710E+02 -5.4777E+05
-2.2103E+02 -1.8331E+00  0.0000E+00  1.4337E+00

3 -8.9186E-01 -2.1987E+02  0.0000E+00  2.2669E+01
-7.3588E+01  1.2669E+02 -5.4260E+05
-3.3419E+01 -1.8735E+02  0.0000E+00  7.7877E+01

4 -8.4722E-01 -2.1967E+02  0.0000E+00  4.5049E+01
-7.3507E+01  1.2659E+02 -5.4129E+05
-1.1045E+02 -1.1007E+02  0.0000E+00  1.0941E+02

5 -9.4435E-01 -2.1977E+02  0.0000E+00  6.7164E+01
-7.3571E+01  1.2661E+02 -5.4163E+05
-1.8681E+02 -3.3904E+01  0.0000E+00  7.8269E+01

6 -5.1879E+02 -5.2461E+02  0.0000E+00 -4.4487E+01
-3.4780E+02  3.0122E+02  7.2907E+06
-5.2165E+02 -5.2175E+02  0.0000E+00 -2.9083E+00

7 -2.9188E+02 -3.8560E+02  0.0000E+00  8.9996E+01
-2.2583E+02  2.0111E+02  1.6524E+06
-3.8560E+02 -2.9188E+02  0.0000E+00  5.9576E-03

8 -4.1535E+02 -4.5938E+02  0.0000E+00  8.9063E+01
-2.9158E+02  2.5347E+02  4.1990E+06
-4.5937E+02 -4.1537E+02  0.0000E+00  7.1972E-01

9 -4.9131E+02 -5.0582E+02  0.0000E+00 -8.7942E+01
-3.3238E+02  2.8794E+02  6.3528E+06
-5.0580E+02 -4.9133E+02  0.0000E+00 -5.2078E-01

10 -4.8878E+02 -5.0213E+02  0.0000E+00  2.4749E+00
-3.3030E+02  2.8613E+02  6.2363E+06
-4.8881E+02 -5.0211E+02  0.0000E+00  5.7590E-01
.
.
.

124 -8.8272E-01 -2.1957E+02  0.0000E+00  1.7100E+01
-7.3486E+01  1.2652E+02 -5.4042E+05
-1.9792E+01 -2.0066E+02  0.0000E+00  6.1463E+01

125 -8.9907E+01 -2.7153E+02  0.0000E+00  1.9367E-01
-1.2048E+02  1.3832E+02 -3.8575E+05
-8.9909E+01 -2.7153E+02  0.0000E+00  6.1391E-01

```

```

126 -3.8590E-01 -2.1868E+02 0.0000E+00 5.8977E+00
      -7.3022E+01 1.2614E+02 -5.3568E+05
      -2.6906E+00 -2.1638E+02 0.0000E+00 2.2312E+01

127 -2.2475E+02 -3.4796E+02 0.0000E+00 -8.9969E+01
      -1.9091E+02 1.7643E+02 7.0364E+05
      -3.4796E+02 -2.2475E+02 0.0000E+00 -6.6744E-02

128 -3.1975E-01 -2.1867E+02 0.0000E+00 8.3886E+01
      -7.2996E+01 1.2616E+02 -5.3583E+05
      -2.1619E+02 -2.7964E+00 0.0000E+00 2.3122E+01

129 -8.0629E+01 -2.6625E+02 0.0000E+00 8.9824E+01
      -1.1562E+02 1.3653E+02 -4.2258E+05
      -2.6624E+02 -8.0631E+01 0.0000E+00 5.7044E-01

130 -8.0188E-01 -2.1923E+02 0.0000E+00 6.1649E+01
      -7.3343E+01 1.2634E+02 -5.3815E+05
      -1.6997E+02 -5.0057E+01 0.0000E+00 9.1282E+01

```

Mesh Refinements for 5% Error

elmt	Energy Norm Error	psi
1	0.26332E+02	0.54885
2	0.99145E+01	0.33678
3	0.67011E+02	0.87557
4	0.33457E+02	0.61867
5	0.55885E+01	0.25285
.		
.		
.		
101	0.63146E+02	0.84994
102	0.38617E+02	0.66467
103	0.19293E+02	0.46980
104	0.50032E+01	0.23924
105	0.46981E+02	0.73312
106	0.10987E+02	0.35453
107	0.11812E+02	0.36759
108	0.63469E+01	0.26946
109	0.41953E+02	0.69278
110	0.93792E+01	0.32756
111	0.22950E+02	0.51240

```

Allowable Element Energy Norm Error          = 0.93494300E+01
Twice of Global Finite Element Strain Energy = 0.38810856E+07
Predicted Global Energy Norm Error           = 0.57376846E+04
Predicted Relative Global Energy Norm Error (%) =          3.842

Relative Bending Error Norm (%)              =          3.842

```

End of Macro Execution

t= 0.92 0.00

R e s t a r t O u t p u t D a t a

Time step number = 0

Time for restart = 0.00000E+00

Time increment = 0.00000E+00

Displacements output

Proportional load = 1.00000E+00

Arc-length load = 1.00000E+00

Force vector output

History data output

Saved Restart File: Radlsnd3.txt