BAB III

ASPEK PROGRAMASI METODE REP DENGAN UI-FEAP

3.1 UMUM

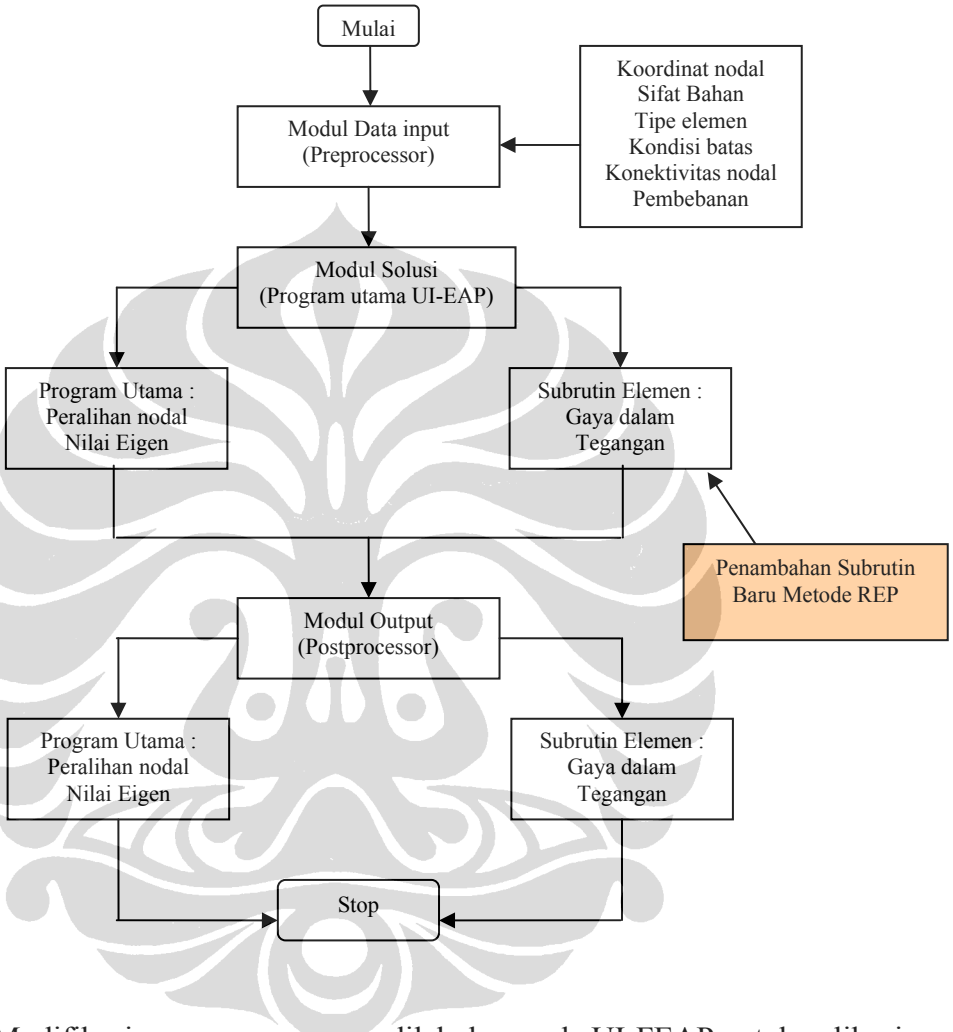
Implementasi teknik pemulihan gaya dalam REP pada UI-FEAP akan dijelaskan dalam bab ini. Implementasi ini dilakukan dengan memodifikasi subroutin yang ada didalam UI-FEAP. Penjelasan dalam bab ini sebagian besar disampaikan dalam bentuk bahasa pemograman Fortran dengan disertai gambaran singkat.

3.2 MODIFIKASI SUBRUTINE UI-FEAP

Implementasi metode REP 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 REP ini, dilakukan modifikasi pada UI-FEAP pada bagian modul solusi. Oleh karena dalam penelitian kali ini metode REP digunakan pada elemen DKMQ [K2] maka modifikasi program yang dilakukan masih terbatas pada elemen tersebut atau elemen quadrilateral lainnya yang sederajat. ***Selain itu, karena prosedur REP 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 REP

3.2.1 Daftar Subrutin Terkait Dengan Error Estimator dan Metode REP

Pada Tabel dibawah ini akan ditampilkan nama-nama subrutin beserta fungsinya yang berkaitan dengan implementasi metode REP pada elemen DKMQ [K2], formulasi estimasi error, beserta fungsinya.

Nama subrutin	Fungsi	Keterangan
ELMT04	Subrutin elemen pelat DKMQ [K2]	-
LUMP	Menghitung matriks massa tergumpal dengan bilinear shape function $\langle N_i \rangle$	-
LUMPQU	Menghitung matriks massa tergumpal dengan quadratik shape function $\langle N_w \rangle$	-
LUMPCU	Menghitung matriks massa tergumpal dengan cubic shape function $\langle N_w \rangle_\xi$	-
GEOME	Menghitung koefisien matriks Jacobian	-
HOOKE	Menghitung matriks Hooke bending $[H_b]$ dan shear $[H_s]$	-
ANDKMQ	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	-
PIQUAD	Menghitung turunan pertama quadratic shape function $\langle P \rangle$ terhadap sumbu koordinat natural ξ,η	-
JACOBQ	Menghitung koefisian matriks invers Jacobian $[j]$	-
JACOBN	Menghitung determinan matriks Jacobian $\lvert J \rvert$	
STFVKB	Menghitung matriks kekakuan lentur $[k_b]$	-
BBDKMQ	Menghitung matriks $[B_b]$	-
BBBDKMQ	Menghitung matriks $[B_{b\beta}]$	-
BBADKMQ	Menghitung matriks $[B_{b\Delta\beta}]$	-
STFVKS	Menghitung matriks kekakuan geser $[k_s]$	-
BSDKMQ	Menghitung matriks $[B_s]$	-
BSADKMQ	Menghitung matriks $[B_{s\Delta\beta}]$	-
DEFCG	Output nilai kurvatur pada titik tengah elemen	-
DEFMID	Output nilai kurvatur pada titik tengah sisi elemen	-
DEFNO	Output nilai kurvatur pada titik nodal elemen	-
EFFCG	Output nilai gaya dalam pada titik tengah elemen	-
EFFMID	Output nilai gaya dalam pada titik tengah sisi elemen	-

EFFNO	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 $\|u^h\|^2$	***Modifikasi***
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 *user command language* sebagai pendukung untuk perhitungan pemulihan gaya dalam	-
COUNT_L_AT_NODES	Menghitung jumlah elemen yang tergabung pada tiap nodal struktur	-
EL_NEIGH	Mencatat nomer-nomer elemen yang tergabung pada tiap nodal struktur	-
COUNT_THE_PATCH 1	Membentuk patch menggunakan metode *Nodal Based Patch*	-
COUNT_THE_PATCH 2	Membentuk patch menggunakan metode *Element Based Patch*	-
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*	*Baru*
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	***Modifikasi***

	mencetak indikator error relatif struktur ϕ^*	
UALLOC	Mengalokasikan memori untuk user dalam menggunakan variabel yang digunakan.	-
getP	*Menghitung matriks* $\langle P \rangle$	*Baru*
Geohookean1	*Mendapatkan koodinat x dan y titik nodal dalam satu elemen, menghitung koefisien matriks Jacobian, menghitung matriks Hooke, menghitung matriks* $[A_n]$ *pada elemen dalam patch*	*Baru*
getB	*Menghitung matriks* $[B_b]$ *dan* $[B_s]$ *pada titik Gauss dalam patch*	*Baru*
getMih	*Mendapatkan gaya dalam hasil solusi*	*Baru*
getMih	*Mendapatkan gaya dalam hasil solusi*	*Baru*
COUNT_THE_PATCH 3	*Membentuk patch menggunakan metode Element Interface Based Patch*	*Baru*

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 REP. Untuk penjelasan subrutin mengenai error estimasi Z^2 pada UI-FEAP serta subrutin DKMQ [K2] sudah tersedia banyak dokumentasi tentang itu dan salah satunya dapat dilihat pada daftar referensi [H3],[M2].

Demikian juga, karena prosedur perhitungan dengan metode REP sebenarnya sama dengan perhitungan dengan metode SPR, maka banyak subrutin-subrutin yang digunakan dalam metode SPR yang juga dipakai dalam metode REP. Kesamaannya adalah pada pembentukan patch, serta penghitungan koordinat riil dan transformasinya ke koordinat natural patch. Subrutin-subrutin `Count_The_Patch1`, `Count_The_Patch2`, `Cor`, dan `Patching` dipakai di metode REP ini tanpa modifikasi, dan karena itu tidak akan dibahas lagi di sini.

3.2.2 Penjelasan Kode Fortran Subrutin REP

Berikut ini adalah penjelasan singkat mengenai kode Fortran pada subrutin teknik pemulihan gaya dalam REP. Struktur lengkap dari subrutin REP dan contoh detail proses REP 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 REP
- 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 REP

3.2.2.1 Mengalokasi Memori Array-array REP

Karena prosedur metode REP 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 REP.

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,'aver',4)) then
        ES = 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
      elseif (pcomp(lct,'proj',4)) then
        ES = 0
        write (*,992)
        write (*,993)
992     format ('Setting to Projection Method on Nodal
Stress/Strain')
```

SPR atau REP

```
993     format ('Please run stre,node,1,last nodal again')
        return
      elseif (pcomp(lct,'scpr',4) .or. pcomp(lct,'rep',4)) then
 write(*,*)'=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*'
!       write (*,994)
        if (pcomp(lct,'scpr',4)) then
          ES = 2
          write (*,*) 'Setting to SPR Method on Nodal Stress/Strain'
        else
          ES = 3
          write (*,*) 'Setting to REP Method on Nodal Stress/Strain'
        endif
write(*,*)'=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*'
        setvar=ualloc(19,'DUM19',numnp,1)
        setvar=ualloc(2,'DUMM2',5*numnp,1)
        setvar=ualloc(20,'DUM20',numnp,2)
        setvar=ualloc(21,'DUM21',numnp,2)
        setvar=ualloc(22,'DUM22',numnp,2)
        setvar=ualloc(23,'DUM23',numnp,2)
        setvar=ualloc(25,'DUM25',numnp*numel,1)
        setvar=ualloc(26,'DUM26',numnp,1)
        setvar=ualloc(27,'DUM27',numnp*numnp,1)
        setvar=ualloc(24,'DUM24',1,1)
        setvar=ualloc(28,'DUM28',numnp,2)
        setvar=ualloc(29,'DUM29',6*6,2)
89      write(*,*) 'ELEMENT BASED PATCH OR NODAL BASED PATCH
 (EBP/NBP) ? '
        read(*,*)k
        if (k .eq. 'nbp' .or. k .eq. 'NBP') then
          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)))
        elseif(k .eq. 'ebp' .or. k .eq. 'EBP') then
99        write(*,*) 'JUST USING THE INTERFACE ELEMENT OR THE WHOLE
     1SURROUNDING ELEMENT OR BACK TO PREVIOUS OPTION (IE/E/B) ? '
          read(*,*)z
          if(z .eq. 'IE' .or. z .eq. 'ie') then
            write(*,*) 'THE METHOD USING ONLY INTERFACE OF ELEMENT
     1 HAS NOT BEEN DEVELOPED YET'
            write(*,*) 'SORRY FOR THE INCONVENIENCE'
            write(*,*)
            goto 99
          elseif(z .eq. 'e' .or. z .eq. 'E') then
            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)))
          elseif (z .eq. 'b' .or. z .eq. 'B') then
            goto 89
          else
            write(*,*) 'PLEASE CHOOSE THE OPTION GIVEN'
            goto 99
          endif
        else
          write(*,*) 'PLEASE CHOOSE THE OPTION GIVEN'
          goto 89
        endif
```

```
!994    format(1x,'Setting to SPR Method on Nodal Stress/Strain')
        return
      endif
```

PENJELASAN :

Bahasa pemograman di atas terdiri dari rangkaian logika untuk mengaktifkan beberapa perintah makro dalam UI-FEAP. Perintah makro tersebut diawali dengan kata 'ssrm' yang merupakan kependekan dari *smoothed stress recovery method*, kemudian diikuti dengan kata yang sesuai untuk melakukan pekerjaan tertentu. Daftar perintah makro tersebut adalah sebagai berikut :

- 'ssrm,aver' : melakukan perintah perhitungan gaya dalam dengan menggunakan metode rata-rata.
- 'ssrm,proj' : melakukan perintah perhitungan gaya dalam dengan menggunakan metode proyeksi.
- 'ssrm,scpr' : melakukan perintah perhitungan gaya dalam dengan menggunakan metode SPR.
- ***'ssrm,rep' : melakukan perintah perhitungan gaya dalam dengan menggunakan metode REP.***
- Kemudian ditambahkan perintah logika untuk pemilihan metode *Nodal Based Patch* dan *Element Based Patc., InterFaced Based Patch*

Selanjutnya, fungsi makro 'ssrm,rep', sama seperti 'ssrm,scpr' untuk metode SPR, 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*.

Kode program `umacr2` di atas sebenarnya sudah berfungsi dengan baik, tetapi penulis berpendapat akan lebih baik jika tidak ada screen input untuk memasukkan opsi model elemen karena hal itu akan memperlambat proses. Di samping itu, penulis juga memandang perlu struktur program yang lebih terstruktur sehingga lebih memudahkan pengembangannya di masa mendatang, misalnya untuk menambah subrutin interface based patch atau metode pemulihan gaya dalam yang lain. Karena itu, penulis membuat kode program baru yang memungkinkan input opsi metode pemulihan gaya dalam dan input opsi model patch dari file input UI-FEAP. Dalam hal ini, perintah makro 'ssrm' tetap dipakai, tetapi tidak harus diikuti statement metode pemulihan seperti 'aver', 'proj', dan sebagainya, karena perintah-perintah tersebut sekarang cukup ditulis dalam file input UI-FEAP. Perintah dalam file input UI-FEAP ditambahkan di belakang statement 'DKMQ', sehingga lengkapnya menjadi

DKMQ <metode pemulihan> <model patch>

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

Modifikasi yang penulis lakukan untuk hal tersebut tidak hanya pada subrutin `umacr2`, tetapi juga sedikit pada subrutin `elmt04` pada bagian pembacaan file input, yaitu seperti dalam kode program berikut:

```
...
if (pcomp(texta(ii),'Aver',4)) ES = 1
if (pcomp(texta(ii),'Proj',4)) ES = 0
if (pcomp(texta(ii),'SCPR',4)) ES = 2
if (pcomp(texta(ii),'REP',3)) ES = 3
if (pcomp(texta(ii),'NBP',3)) patchmodel = 1
if (pcomp(texta(ii),'EBP',3)) patchmodel = 2
...
```

Sedangkan dalam umacr2 sendiri, modifikasinya adalah sebagai berikut:

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
        ERROREST MODE = 2
      elseif (pcomp(lct,'ui',2)) then
        ERROREST MODE = 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)
          write (*,*) 'Setting to REP Method on Nodal Stress/Strain'
        end select Recovery_method
        Superconvergent_method: select case (ES)
        case (2,3,5)
          setvar=ualloc(19,'DUM19',numnp,1)
          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)
          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
            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) ! IBP
           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)
```

```
        end select Patch_model
        return
      end select Superconvergent_method
    endif
```

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, Subrutin yang berkaitan dengan penelitian yang penulis lakukan yang menggunakan metode element interface based patch, subrutin yang digunakan adalah Count_The_Patch3 yang akan dijelaskan berikut*

```
subroutine Count_The_Patch3(x,ix,nopatch,elpatch,nonode,xmax,xmin,
     & ymax,ymin,jnapp)
      include  'eldata.h'
      include  'cdata.h'
      include  'iofile.h'
      include  'sdata.h'

      integer
i,j,ii,jj,nopatch,jnapp,jumnapp,nelmt(numnp),nnode(numnp)
      integer
ix(nen1,numel),elpatch(numnp,numel),nonode(numnp,numnp),cknode
      real*8
x(ndm,numnp),xmax(numnp),xmin(numnp),ymax(numnp),ymin(numnp)

      nopatch=numel
      xmax=-1e20 ; xmin=1e20 ; ymax=-1e20 ; ymin=1e20
      do i=1,numel      ! each patch center
        nelmt(i)=0 ; nnode(i)=0
        do j=1,numel    ! element under investigation
            jumnapp=0
```

```
      do jj=1,4
        do ii=1,4
          if(ix(jj,j) == ix(ii,i)) jumnapp=jumnapp+1
        end do
      end do
      if(jumnapp >= jnapp)then
        nelmt(i)=nelmt(i)+1
        elpatch(i,nelmt(i))=j
        do jj=1,4
          cknode=0
              do ii=1,nnode(i)
            if(nonode(ii,i) == ix(jj,j))then
              cknode=1 ; exit
               endif
          end do
              if(cknode == 0)then
                nnode(i)=nnode(i)+1
            nonode(nnode(i),i)=ix(jj,j)
                xmax(i)=max(xmax(i),x(1,ix(jj,j)))
                xmin(i)=min(xmin(i),x(1,ix(jj,j)))
                ymax(i)=max(ymax(i),x(2,ix(jj,j)))
                ymin(i)=min(ymin(i),x(2,ix(jj,j)))
              endif
        end do
      endif
    end do
  end do
  call write_patch_db(nopatch,elpatch,nonode,nelmt,nnode,
 & xmax,xmin,ymax,ymin)
  return

  end

  subroutine write_patch_db(nopatch,elpatch,nonode,nelmt,nnode,
 & xmax,xmin,ymax,ymin)
    include  'cdata.h'
    include  'iofile.h'
```

```
      integer i,j,nopatch,elpatch(numnp,numel),nonode(numnp,numnp)
      integer nelmt(numnp),nnode(numnp)
      real*8  xmax(numnp),xmin(numnp),ymax(numnp),ymin(numnp)

      write(iow,'(//,A,//,A,//)')'P A T C H    P R O P E R T I E
S',
     &  'Patch  E l e m e n t s  o n  p a t c h'
      do i=1,nopatch
        write(iow,'(i5,2x,12(i5))')i,(elpatch(i,j),j=1,nelmt(i))
      end do
      write(iow,'(//,A,//)')'Patch  N o d e s  o n  p a t c h'
      do i=1,nopatch
        write(iow,'(i5,2x,24(i5))')i,(nonode(j,i),j=1,nnode(i))
      end do
      write(iow,'(//,A,//,A,//)')'P a t c h  D o m a i n',
     &  'Patch     Xmax    Xmin    Ymax    Ymin'
      do i=1,nopatch

write(iow,'(i5,2x,4(f8.3))')i,xmax(i),xmin(i),ymax(i),ymin(i)
      end do

      end
```

3.2.2.3 Komputasi Gaya Dalam Nodal dengan Metode REP

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

`subroutine Patching(x,kj,ox,oy,su,jdet,xx,yy)`

- - > pendukung

Subrutin ini adalah subrutin yang sama dengan yang digunakan dalam metode SPR yang berfungsi untuk menghitung koordinat riil titik Gauss yang terdapat pada setiap elemen.

$$x(\xi,\eta) = N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4$$
$$y(\xi,\eta) = N_1 y_1 + N_2 y_2 + N_3 y_3 + N_4 y_4$$

Koordinat riil titik Gauss tersebut diperlukan dalam penghitungan matriks $[B_b]$ dan $[B_s]$.

`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. Tranformasi tersebut dilakukan berdasarkan persamaan berikut :

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

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

subroutine getP (P,xw,yw) - - > pendukung

Subrutin ini berfungsi untuk menyusun matriks <*P*> sesuai persamaan 2.17. Penulis memilih membuatnya dalam subrutin tersendiri mengingat bahwa subrutin ini bisa dimanfaatkan untuk program pemulihan solusi yang lain di masa mendatang.

KODE FORTRAN :

```
P(1,1)=1
P(1,2)=xw
P(1,3)=yw
P(1,4)=xw*xw
P(1,5)=xw*yw
P(1,6)=yw*yw
P(1,7)=xw*xw*yw
P(1,8)=xw*yw*yw
Return
```

subroutine geohookean1(x,d,ix,el,xl) - - > pendukung

Subrutin ini berfungsi untuk mendapatkan koordinat x dan y dari 4 nodal dalam suatu elemen. Koordinat tersebut diperlukan untuk input bagi subrutin-subrutin yang dipanggil kemudian, yaitu subrutin GEOME2, HOOKE2 dan ANDKMQ yang merupakan subrutin-subrutin yang diperlukan dalam penyusunan matriks [B_b] dan [B_s].

.

KODE FORTRAN :

```
do ii=1,2
  do jj=1,4
    xl(ii,jj)=x(ii,ix(jj,el))
  end do
end do
call GEOME2(xl)
call HOOKE2(d)
call ANDKMQ
return
```

subroutine getB (qsi,eta,bb,bs,B) --> pendukung

Subrutin ini berfungsi untuk menyusun matriks $[B_b]$ dan $[B_s]$ dalam persamaan 2.32 dan 2.33. Matriks $[B_b]$ dan $[B_s]$ ini tidak lain adalah matriks dengan persamaan yang sama dengan matriks $[B_b]$ dan $[B_s]$ dalam elemen DKMQ. Subrutin ini memanggil subrutin-subrutin DKMQ, yaitu NIQUAD, PIQUAD, JACOBQ2, BBDKMQ dan BSDKMQ yang juga sudah banyak dibahas.

KODE FORTRAN :

```
call NIQUAD(qsi,eta)
call PIQUAD(qsi,eta)
call JACOBQ2(qsi,eta)
call BBDKMQ(bb)
call BSDKMQ(bs)
do j=1,12
  do i=1,3
     B(i,j)=bb(i,j)
   end do
  do i=1,2
     B(i+3,j)=bs(i,j)
   end do
end do
return
```

subroutine getMih(vmx,vmy,vmxy,Txz,Tyz,i,Mih)

--> pendukung

Subrutin ini adalah subrutin untuk memilih gaya dalam yang aktif M_i^h dari lima gaya dalam M_x, M_y, M_{xy}, T_{xz}, dan T_{yz}.

KODE FORTRAN :

```
select case (i)
  case (1) ; Mih=vmx
  case (2) ; Mih=vmy
  case (3) ; Mih=vmxy
  case (4) ; Mih=Txz
  case (5) ; Mih=Tyz
end select
```

Subroutine REP_Recovery(x,d,vmx,vmy,vmxy,Txz,Tyz,kj, nopatch,elpatch,nonode,xmax,xmin,ymax,ymin,scm,sct)

Subrutin ini berfungsi untuk membentuk matriks $[B_b]$ dan $[B_s]$ yang untuk selanjutnya digunakan untuk membentuk matriks $[D]$ dan matriks $\{F^h\}$. Selanjutnya dari matriks-matriks tersebut dihitung vektor $\{a_n\}$ (pers. 2.33), yang kemudian digunakan untuk menghitung nilai gaya dalam pada nodal tiap elemen. Subrutin ini akan terus dipanggil oleh subrutin `NoForDKMQ_3` 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_3` ke subrutin `REP_Recovery` melalui variabel `kj`.

Subrutin `REP_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 :

```
qkj=0 ; Mm=0
do j=1,nopatch          ! tiap patch
  chknode=0 ; k=1
  do while (nonode(k,j) .ne. 0)     ! tiap node dalam patch
    if (nonode(k,j) .eq. kj) then
      chknode=1 ; exit
    endif
    k=k+1
  end do
  if (chknode .eq. 1) then     ! jika node ada dalam patch
    ...
    ...
  endif
end do      ! j
```

PENJELASAN :

Kode fortran diatas berfungsi untuk melakukan proses looping terhadap seluruh patch yang sudah dibentuk oleh subrutin `Count_the_patch 3`.

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 penghitungan pemulihan solusi. Jika tidak ada, looping dilanjutkan ke patch berikutnya.

KODE FORTRAN :

```
...
do j=1,nopatch          ! tiap patch
  ...
  if (chknode .eq. 1) then     ! jika node ada dalam patch
    qkj=qkj+1
    call Cor(x(1,kj),x(2,kj),xmax(j),xmin(j),ymax(j),ymin(j),xw,yw)
    call getP(P,xw,yw)
    do i=1,5 ! tiap komponen gaya dalam (Mx, My, Mxy, Txz, Tyz)
      ...
      ...
    end do  ! i
  endif
end do      ! j
```

PENJELASAN :

Jika nodal yang ditinjau ada dalam patch, langkah selanjutnya adalah menghitung koordinat natural patch dari nodal yang diinjau. Koordinat natural tersebut digunakan untuk membentuk matriks $<P>$ yang nantinya, misalnya sebagai contoh, digunakan untuk menghitung persamaan $M_x=<P>\{a_n\}$.

KODE FORTRAN :

```
...
do j=1,nopatch          ! tiap patch
  ...
  if (chknode .eq. 1) then     ! jika node ada dalam patch
    ...
    do i=1,5 ! tiap komponen gaya dalam (Mx, My, Mxy, Txz, Tyz)
      Hi=0 ; Fi=0 ; HtHe=0 ; HtFe=0 ; k=1 ; i1=0
      do while (elpatch(j,k) .ne. 0) ! tiap elemen anggota patch
        call geohookean1(x,d,mr(np(33)),elpatch(j,k),xl)
        call Patching(xl,kj,hr(up(3)),hr(up(4)),elpatch(j,k)
&        ,hr(up(17)),xx,yy)   ! hitung koord titik Gauss
        He=0 ; Fe=0
        do m=1,4  ! tiap titik Gauss
          ...
          ...
        end do
        ...
```

```
          ...
          k=k+1
        end do      ! elemen anggota patch
        ...
      end do  ! i
    endif
  end do      ! j
```

PENJELASAN :

Selanjutnya dilakukan looping terhadap semua komponen gaya dalam. Setelah inisialisasi variabel-variabel, dilakukan looping terhadap semua elemen yang menjadi anggota patch. Di sini koordinat keempat nodal dalam elemen didapat dengan memanggil subrutin `geohookean1`. Subrutin tersebut juga menghitung parameter-parameter yang diperlukan dalam penyusunan matriks $[B_b]$ dan $[B_s]$ seperti yang sudah dijelaskan di depan. Output lain subrutin tersebut adalah variabel array `xl` yang kemudian menjadi input dalam menghitung koordinat titik Gauss dengan subrutin `Patching`. Koordinat titik Gauss tersebut diperlukan untuk menyusun matriks $[B_b]$ dan $[B_s]$ dan matriks $<P(\xi_k,\eta_k)>$ yang merupakan komponen matriks $[D_e]$.

KODE FORTRAN :

```
...
do j=1,nopatch          ! tiap patch
  ...
  if (chknode .eq. 1) then     ! jika node ada dalam patch
    ...
    do i=1,5 ! tiap komponen gaya dalam (Mx, My, Mxy, Txz, Tyz)
      ...
      do while (elpatch(j,k) .ne. 0) ! tiap elemen anggota patch
        ...
        do m=1,4  ! tiap titik Gauss
          call
Cor(xx(m),yy(m),xmax(j),xmin(j),ymax(j),ymin(j),xw,yw)
          call getP(Pe,xw,yw)
          call getB(xw,yw,bb,bs,B)
          i1(i,1)=detj  ! 1i detj
          He=He+matmul(matmul(transpose(B),i1),Pe)
           ! He=sum(Bt 1i P detj)
          call getMih(vmx(m,elpatch(j,k)),vmy(m,elpatch(j,k))
     &           ,vmxy(m,elpatch(j,k)),Txz(m,elpatch(j,k))
     &           ,Tyz(m,elpatch(j,k)),i,Mih)
          Fe=Fe+matmul(transpose(B),i1)*Mih ! Fe=sum(B 1i Mih detj)
        end do
        Hi=Hi+He  ! Hi=sum(He)
```

```
        Fi=Fi+Fe  ! Fi=sum(Fe)
        HtHe=HtHe+matmul(transpose(He),He)      ! sum(HtHe)
        HtFe=HtFe+matmul(transpose(He),Fe)      ! sum(HtFe)
        k=k+1
      end do      ! elemen anggota patch
      HtHi=matmul(transpose(Hi),Hi)
      HtFi=matmul(transpose(Hi),Fi)
      HtHi=HtHi+HtHe    ! HtHi + sum(HtHe)
      HtFi=HtFi+HtFe    ! HtFi + sum(HtFe)
      call dlinrg(8,HtHi,8,iHtH,8)
      a=matmul(iHtH,HtFi)
      mmm=matmul(P,a)
      Mm(i)=Mm(i)+mmm(1,1)
    end do  ! i
  endif
end do      ! j
do i=1,3 ; scm(i)=Mm(i)/qkj
end do
do i=1,2 ; sct(i)=Mm(i+3)/qkj
end do
return
```

PENJELASAN :

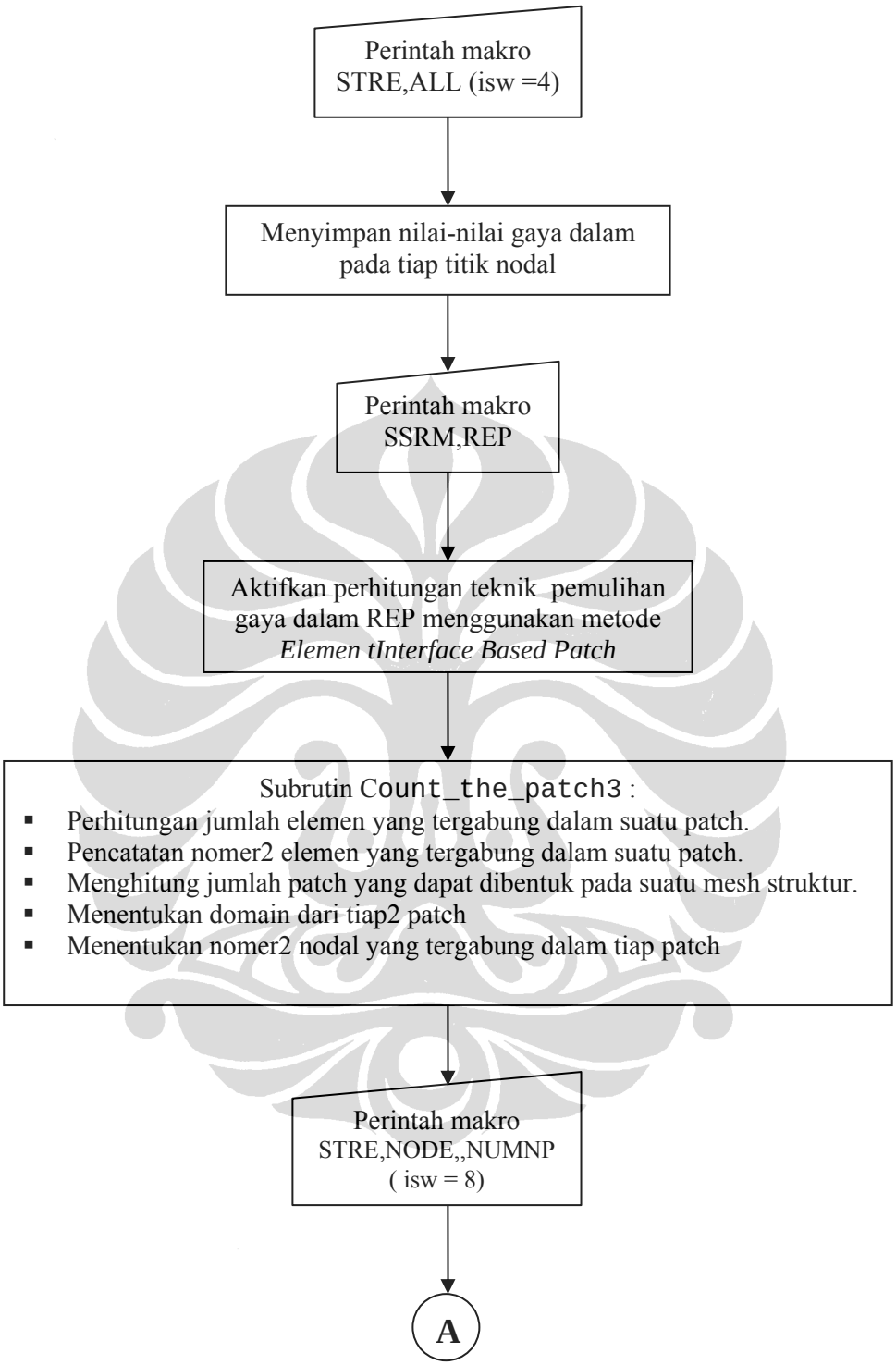
Langkah berikutnya adalah looping pada semua titik Gauss. Dalam looping ini, pertama dihitung koordinat natural patch dari tiap titik Gauss. Koordinat natural tersebut merupakan input bagi penyusunan matriks $<P(\xi_k, \eta_k)>$ dan matriks $[B]$. Dengan diperolehnya matriks $<P(\xi_k, \eta_k)>$ dan matriks $[B_b]$, selanjutnya dapat dihitung matriks $[D_e]$ dan matriks $\{F_e^h\}$. Matriks $[D_e]$ dan $\{F_e^h\}$ tersebut diakumulasikan untuk semua titik Gauss dalam satu elemen. Kemudian, matriks tersebut juga diakumulasikan untuk semua elemen dalam patch untuk membentuk matriks $[D]$ dan $\{F^h\}$. Setelah lengkap looping dalam semua elemen patch, matriks $\{a_n\}$ bisa dihitung, dan selanjutnya matriks $\{M\}$ dan $\{T\}$ bisa dihitung. Matriks $\{M\}$ dan $\{T\}$ tersebut diakumulasi untuk semua patch dan elemen yang mengandung nodal yang sedang ditinjau, untuk kemudian dibagi dengan jumlah overlapping yang terjadi.

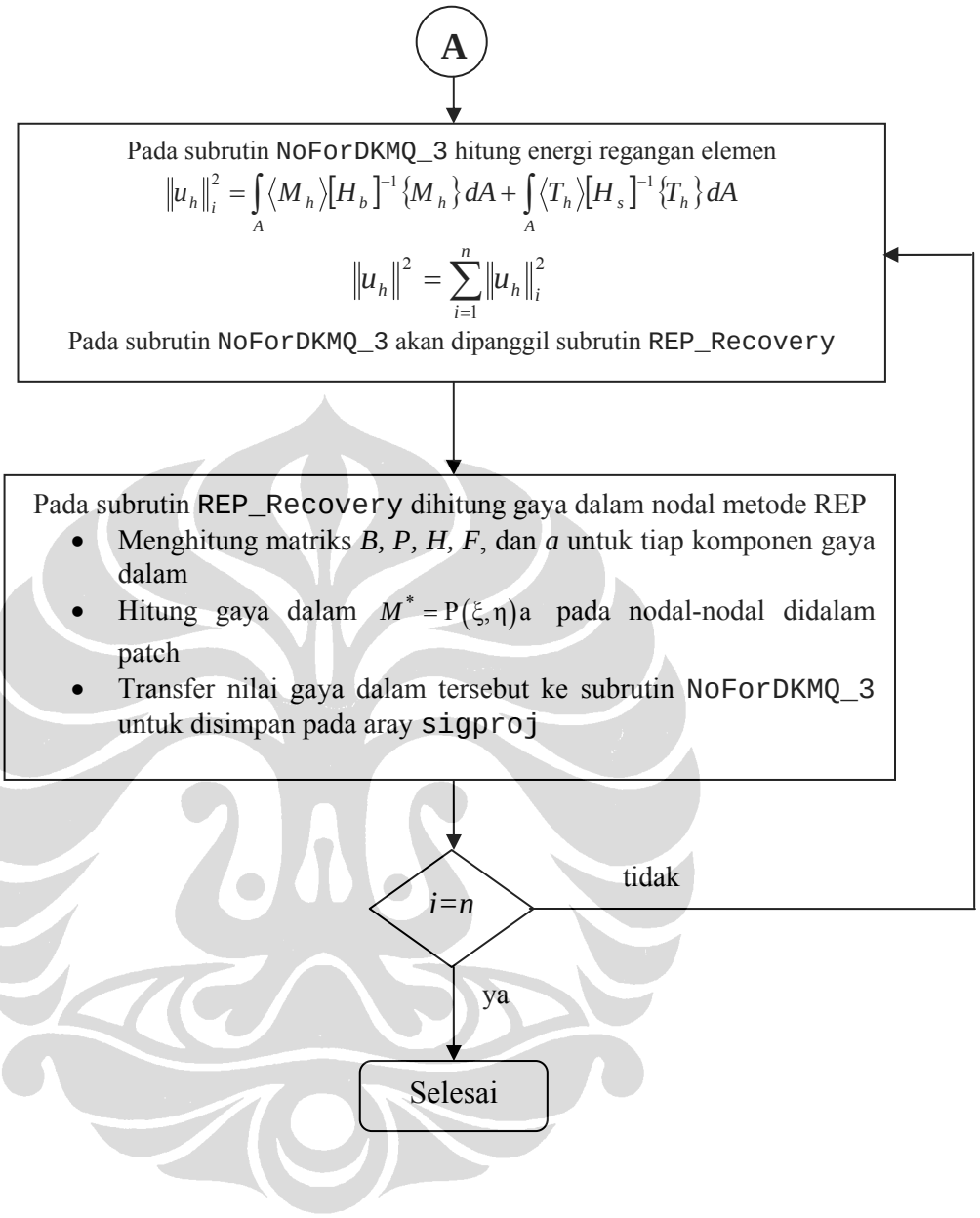
```
subrutin NoForDKMQ_3 (x,d,u,np,ix,ndm,nel,dt,
sigproj)
```

Subrutin ini termasuk di dalam subrutin elemen DKMQ [K2] yang berfungsi untuk menyimpan nilai gaya dalam nodal berdasarkan metode REP. Jadi dalam subrutin ini akan selalu dipanggil subrutin REP_Recovery. Subrutin ini juga subrutin yang dipakai dalam metode SPR oleh peneliti terdahulu. Penulis memilih untuk tidak membentuk subrutin baru (NoForDKMQ_4 misalnya) karena semua langkah dalam subrutin tersebut dipakai dalam metoda REP. Penulis hanya memodifikasi dengan menambah *select case condition* untuk menjalankan subrutin REP_Recovery.

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)
              case (2)
                call SCPR_Recovery(hr(np(43)),ix,hr(up(17)),hr(up(7))
     &          ,hr(up(8)),hr(up(9)),hr(up(10)),hr(up(11)),mr(up(19))
     &          ,mr(up(2)),hr(up(12)),k,mr(up(24)),mr(up(25)),mr(up(26))
     &          ,mr(up(27)),hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23))
     &          ,scm,sct)
              case (3)
                call
REP_Recovery(hr(np(43)),d,hr(up(7)),hr(up(8)),hr(up(9))
     &          ,hr(up(10)),hr(up(11)),k,mr(up(24)),mr(up(25)),mr(up(27))
     &          ,hr(up(20)),hr(up(21)),hr(up(22)),hr(up(23)),scm,sct)
              case (4)
c                   reserved for Recovery by Compatibility of Patches
              end select Recovery_method
              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
 8000 continue
```

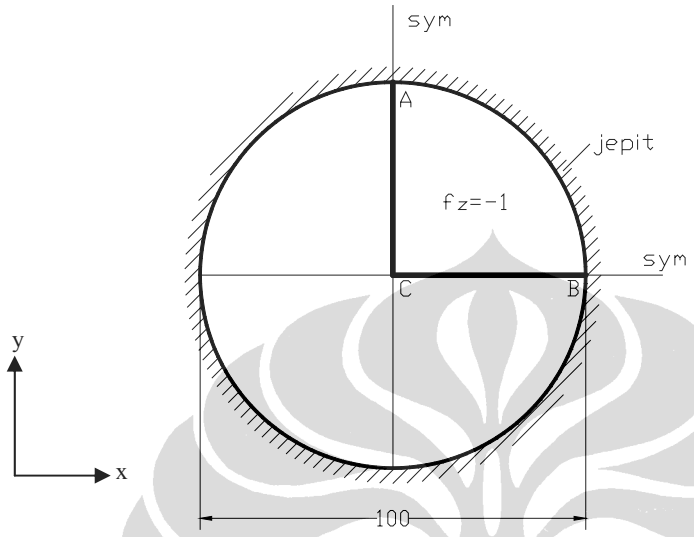

Perintah makro
STRE,ALL (isw =4)
Menyimpan nilai-nilai gaya dalam
pada tiap titik nodal
Perintah makro
SSRM,REP
Aktifkan perhitungan teknik pemulihan
gaya dalam REP menggunakan metode
Elemen tInterface Based Patch
Subrutin Count_the_patch3 :
Perhitungan jumlah elemen yang tergabung dalam suatu patch.
Pencatatan nomer2 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
Perintah makro
STRE,NODE,,NUMNP
(isw = 8)
A



Gambar 3. 2 Diagram alir perhitungan gaya dalam dan energi regangan elemen dengan menggunakan metode REP pada UI-FEAP

3.2.3 Contoh Data Input Analisa Metode REP dan Error Estimasi

Struktur pelat yang akan dianalisis :



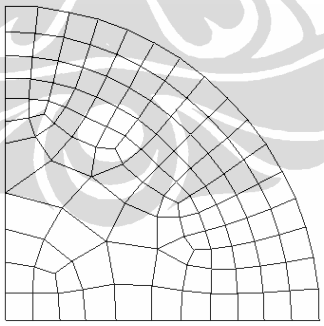
Data :
$E = 1000$; $v = 0{,}3$
$R = 50$
Beban terbagi rata : $f_z = -1$

Kondisi batas :
Jepit : $w = \beta_x = \beta_y = 0$

Kondisi simetri:
Sisi CB = $\beta_y = 0$
Sisi CA = $\beta_x = 0$

Gambar 3. 3 Contoh analisa struktur Lingkaran dengan UI-FEAP

Pelat tersebut dianalisa ¼ bagian saja dan dimodelisasi dengan mesh adaptif 110 elemen sebagai berikut :



Gambar 3. 4 Mesh Adaptif

File data input adalah sebagai berikut :

```
UI-FEAP**
130,110,1,2,3,4

mate,1
user 4
DKMQ rep ibp
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
1      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
```

3.2.4 Contoh Data Output

File data output adalah sebagai berikut :

```
UI-FEAP**

     Solution date: Sun Feb 18 11:13:31 2007

            UNIX/PC 7.1b
            - 12/15/98 -

     Input Data Filename: IADLJPT3.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

 UI-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  4

     Degree of Freedom Assignments    Local    Global
                                     Number    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.
          ================================================
               DISCRETE KIRCHHOFF MINDLIN 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 =  3 Recovery by Equilibrium in Patches (REP)
          patch type      = Element Interface Based Patch

UI-FEAP**

    Nodal Coordinates

     node    1 Coord    2 Coord
       1  6.000E+01  1.000E+01
       2  1.000E+01  6.000E+01
       3  5.613E+01  2.928E+01
       4  4.536E+01  4.536E+01
       .
       .
       .
     127  1.000E+01  4.848E+01
     128  1.520E+01  5.973E+01
     129  1.000E+01  5.590E+01
     130  3.367E+01  5.404E+01

UI-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


 UI-FEAP**

     E l e m e n t s

   Elmt Mat Reg  1 Node  2 Node  3 Node  4 Node
      1   1   0     112     110      17      83
      2   1   0     110     111      74      17
      3   1   0     109     107      12     113
     .
     .
     .
    107   1   0      87     105      27      26
    108   1   0      35      30      74     111
    109   1   0      35      89      87      26
    110   1   0     111      88      89      35

     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   =      317   Number nodes     =     130
          Average col. height   =      127   Number elements  =     110
          Number profile terms  =    40107   Number materials =       1
          Number rigid bodies   =        0   Number joints    =       0
          Est. factor time-sec  = 5.9917E-02

          Material    Element Tag   Element Type  History Terms  Element
Terms
             1         1              4             0             12

 *Macro   1 * tang                 v:   1.00       0.00       0.00
                                                              t=     4.81
0.00
   Residual norm =     1.9193508E+02    1.0000000E+00      t=     5.10
0.00
 Condition check: D-max 0.5944E+03; D-min 0.1659E+01; Ratio 0.3582E+03
 Maximum no. diagonal digits lost:  2
   End Triangular Decomposition                           t=     5.38
0.00
   Number of operations =  668973 plus        6 Mega-ops
  Time: CPU =         5.46 , System =         0.00
--> SOLVE AT    30.31 Mflops. Time=        0.22
   Energy convergence test
    Maximum   =    6.969734190955363E+05 Current   =
6.969734190955363E+05
    Relative  =    1.000000000000000E+00 Tolerance =    1.000000000000000E-
16

 *Macro   1 * disp ALL             v:   0.00       0.00       0.00
                                                              t=    11.62
0.00

 UI-FEAP**

  N o d a l   D i s p l a c e m e n t s     Time       0.00000E+00
```

```
                                                  Prop. Ld.  1.00000E+00

   Node     1 Coord     2 Coord     1 Displ     2 Displ     3 Displ
      1  6.0000E+01  1.0000E+01  0.0000E+00  0.0000E+00  0.0000E+00
      2  1.0000E+01  6.0000E+01  0.0000E+00  0.0000E+00  0.0000E+00
      3  5.6132E+01  2.9283E+01  0.0000E+00  0.0000E+00  0.0000E+00
      4  4.5355E+01  4.5355E+01  0.0000E+00  0.0000E+00  0.0000E+00
     .
     .
     .
    126  1.0000E+01  4.1581E+01 -3.8499E+02  0.0000E+00 -3.2240E+01
    127  1.0000E+01  4.8481E+01 -1.7758E+02  0.0000E+00 -2.6669E+01
    128  1.5201E+01  5.9729E+01  0.0000E+00  0.0000E+00  0.0000E+00
    129  1.0000E+01  5.5897E+01 -2.6688E+01  0.0000E+00 -1.2307E+01
    130  3.3671E+01  5.401E+01  0.0000E+00  0.0000E+00  0.0000E+00

 *Macro   1 * stre ALL              v:   0.00       0.00       0.00
                                                              t=    25.94
0.00

 UI-FEAP**


 F O R C E S  O N  G A U S S  P O I N T S  O F  E L E M E N T
 ============================================================


ELMT    X       Y       Mx          My       Mxy       Txz       Tyz
   1   43.330   27.373 0.6397E+02 -.941E+01 0.4905E+02 0.1311E+02 0.6977E+01
   1   41.603   26.694 0.3625E+02 -.2721E+02 0.413E+02 0.1314E+02 0.6911E+01
   1   1.248   24.970 0.3603E+02 -.3224E+02 0.196E+02 0.1303E+02 0.6871E+01
   1   44.089   25.552 0.6478E+02 -.1417E+02 0.4751E+02 0.1301E+02 0.6934E+01
   2   40.324   26.279 0.1956E+02 -.3874E+02 0.4124E+02 0.1204E+02 0.6214E+01
   2   38.558   25.822 -.6041E+01 -.5435E+02 0.381E+02 0.1221E+02 0.5546E+01
   2   39.017   24.045 -.5057E+01 -.5927E+02 0.3707E+02 0.1063E+02 0.5139E+01
   2   40.889   24.566 0.1899E+02 -.4415E+02 0.3980E+02 0.1048E+02 0.5699E+01
   3   45.998   13.278 0.6660E+02 -.4553E+02 0.8356E+01 0.1358E+02 0.1001E+01
   3   43.559   13.302 0.3032E+02 -.6608E+02 0.831E+01 0.1358E+02 0.1022E+01
   3   43.530   10.885 0.2898E+02 -.6897E+02 0.4578E+01 0.1347E+02 0.1023E+01
   3   46.017   10.878 0.6579E+02 -.4780E+02 0.4662E+01 0.1347E+02 0.1000E+01.
   .
   .
 108   34.905   25.413 -.5032E+02 -.7807E+02 0.3361E+02 0.1103E+02 0.6588E+01
 108   35.077   22.679 -.5748E+02 -.9637E+02 0.3072E+02 0.8764E+01 0.6446E+01
 108   37.459   23.539 -.2832E+02 -.7455E+02 0.3385E+02 0.9191E+01 0.5263E+01
 108   306   25.590 -.2322E+02 -.6390E+02 0.3578E+02 0.1111E+02 0.5593E+01
 109   33.299   27.561 -.5236E+02 -.7709E+02 0.3430E+02 0.1001E+02 0.7893E+01
 109   34.158   29.469 -.3811E+02 -.5492E+02 0.3993E+02 0.8656E+01 0.8505E+01
 109   32.023   30.700 -.4917E+02 -.5809E+02 0.4048E+02 0.7887E+01 0.7172E+01
 109   30.576   28.738 -.7268E+02 -.8414E+02 0.3606E+02 0.9264E+01 0.6156E+01
 110   36.972   26.862 -.2026E+02 -.6190E+02 0.4055E+02 0.1189E+02 0.8881E+01
 110   36.959   28.287 -.1449E+02 -.4792E+02 0.4326E+02 0.1127E+02 0.8875E+01
 110   35.480   28.822 -.2938E+02 -.510E+02 0.4334E+02 0.1113E+02 0.8492E+01
 110   35.013   27.059 -.4561E+02 -.7607E+02 0.3914E+02 0.1183E+02 0.8304E+01
 *Macro   1 * ssrm REP              v:   0.00       0.00       0.00
```

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

Patch type: Element Interface Based Patch 1

Patch  E l e m e n t s   o n   p a t c h

    1      1    2   38   40   68
    2      1    2   41   77  108
    3      3    4   1   44
    4      3    4   45  103.
    .
    .
  107      6  105  106  107  109
  108      2   32   33  108  110
  109     32   67  107  109  110
  110     15   41  108  109  110

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

   element      element number in patch
    1            1, 38, 39, 40,  2, 41, 68, 77, 69,
    2            1,  2, 38, 41,108,110, 33, 77, 68,
    3            3, 1, 43, 44,  4, 45,
    .
    .
    .
  107            6, 67,107,109, 47,105, 28,104,106, 32,
  108           32,108,109,110, 31, 33, 34,  2, 77, 41,
  109           32,108,109,110, 15, 67,  6,107,106,
  110            2, 41,108,110, 15, 67,109, 32,



- - > 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     110

 p a t c h      e l e m e n  o n  p a t c h
     1          1,   38,   39,   40,    2,   41,   68,   77,   69,
     2          1,    2,   38,   41,  108,  110,   33,   77,   68,
     3          3,   1,   43,   44,    4,   45,
     4          3,    4,   1,   45,   31,  103,
     .
     .
     .
   107          6,   67,  107,  109,   47,  105,   28,  104,  106,   32,
   108         32,  108,  109,  110,   31,   33,   34,    2,   77,   41,
   109         32,  108,  109,  110,   15,   67,    6,  107,  106,
   110          2,   41,  108,  110,   15,   67,  109,   32,

   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           49.727,  37.431,  33.192,  21.056
     2           46.095,  34.061,  31.298,  18.743
     3           51.228,  37.914,  18.743,  10.000
     4           46.940,  33.148,  21.272,  10.000
     .
     .
     .
```

```
   107           374,  10.000,  39.254,  21.272
   108           1.763,  26.086,  31.876,  14.586
   109           39.360,  19.067,  37.059,  21.272
   110           41.795,  26.086,  34.436,  21.272

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

 n o d e
  112,110, 17, 83,100, 34, 36, 91, 32,111, 74, 88, 73, 85, 76, 81,- - >
belong to patch  1
  112,110, 17, 83,111, 74,100, 34, 88, 35, 30, 89, 75, 76, 73, 85,- - >
belong to patch  2
  109,107, 12,113, 84, 29, 33, 78, 38,108,114, 75,- - >
belong to patch  3
  109,107, 12,113,108,114, 84, 29, 75, 30, 31, 40,- - >
belong to patch  4
    .
    .
    .
  105, 87, 14, 55, 89, 86, 27, 26, 35, 69, 20, 70, 25,  8, 15, 68, 46, 30,- - > belong
to patch107
   46, 30, 35, 26, 74,111, 89, 87, 88, 75, 31,108, 76, 16,110, 17, 73, 34,- - > belong
to patch108
   46, 30, 35, 26, 74,111, 89, 87, 88, 86, 34, 14,105, 55, 27, 15,- - >
belong to patch109
  110,111, 74, 17, 34, 88, 35, 30, 89, 86, 14, 87, 26, 46,- - >
belong to patch110

 *Macro   1 * stre NODE           v:   0.00       130.       0.00
                                                          t=   3.73
0.00

 UI-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      5 Value      6 Value

     1  3.1359E+02  9.3497E+01  0.0000E+00  7.7256E-01
        1.3569E+02  1.6100E+02  7.0625E+05
        3.1355E+02  9.3537E+01  0.0000E+00  2.9673E+00 -6.8629E-01  1.9630E+01

     2  3.1372E+02  9.3215E+01  0.0000E+00  8.9157E+01
        1.3564E+02  1.6110E+02  7.1059E+05
        9.3263E+01  3.1367E+02  0.0000E+00  3.2445E+00  2.061E+01 -7.521E-01

     3  3.1485E+02  9.3792E+01  0.0000E+00  2.2843E+01
        1.3621E+02  1.6165E+02  7.1569E+05
        2.8153E+02  1.2711E+02  0.0000E+00  7.9086E+01  7.6269E+00  1.8254E+01

     4  3.1393E+02  9.4690E+01  0.0000E+00  4.4955E+01
        1.3621E+02  1.6103E+02  6.9685E+05
        2.0449E+02  2.0414E+02  0.0000E+00  1.0962E+02  1.4070E+01  1.3683E+01.
     .
     .

   127  1.0232E+02 -2.6003E+01  0.0000E+00  8.9336E+01
        2.5439E+01  6.7838E+01  6.9759E+04
       -2.5985E+01  1.0230E+02  0.0000E+00  1.4881E+00  1.5477E+01  7.2721E-01

   128  3.1394E+02  9.115E+01  0.0000E+00  8.3848E+01
        1.3605E+02  1.6110E+02  7.0209E+05
        9.6739E+01  3.111E+02  0.0000E+00  2.3411E+01  2.0272E+01  2.1285E+00
```

```
    129  2.3188E+02  4.7103E+01  0.0000E+00  8.9254E+01
         9.2993E+01  1.2256E+02  4.1093E+05
         4.7135E+01  2.3184E+02  0.0000E+00  2.4040E+00  1.8002E+01 -8.1309E-01

    130  3.1332E+02  9.3571E+01  0.0000E+00  6.1534E+01
         1.3563E+02  1.6084E+02  7.0277E+05
         1.4349E+02  2.6339E+02  0.0000E+00  9.2075E+01  1.7083E+01  9.6283E+00
*Macro   1 * ssrm UI              v:   0.00       0.00       0.00
                                                          t=   7.55
0.00
   M e s h   R e f i n e m e n t s   f o r   5 %   E r r o r

    elmt   Energy Norm Error          psi

       1     0.77877E+00             0.22173
       2     0.69091E+00             0.20885
       3     0.87588E+01             0.74360
             .
             .
           106     0.18048E+02             1.06741
           107     0.73620E+01             0.68174
           108     0.39817E+01             0.50137
           109     0.32923E+01             0.45590
           110     0.21150E+01             0.36540

   Allowable Element Energy Norm Error             = 0.39799881E+01
   Twice of Global Finite Element Strain Energy    = 0.6969731E+06
   Predicted Global Energy Norm Error              = 0.72748499E+03
   Predicted Relative Global Energy Norm Error (%) =         3.229

*End of Macro Execution*                                t=     2.65   0.00
```