LAMPIRAN I
TABEL NOTASI

TABEL NOTASI PADA METODE ELEMEN HINGGA

NOTASI	DEFINISI
$\tau xy,..$	Tegangan geser
$\sigma x,\sigma y,.$	Tegangan Normal
$\epsilon x,\epsilon y,.$	Regangan Normal
$\Gamma xy,..$	Regangan Geser
x,y,z	Sumbu Koordinat
u,v,w	Arah Sumbu Koordinat
σ	Vektor Tegangan
ϵ	Vektor Regangan
v	Poison's Ratio
G	Modulus Geser
E	Modulus Elastisitas
C	Operator Penghubung vektor regangan dan vektor Tegangan
E	Operator Penghubung vektor Tegangan dan vektor regangan
u	Peralihan Umum
b	Gaya Tubuh
bx,by,bz	gaya (persatuan luasan) yang bekerja pada sembarang titik,sesuai dengan sumbunya
q	Peralihan Titik Nodal
p	Gaya Titik Nodal
d	Operator deferensial linier
B	Regangan sebagai akibat peralihan nodal
U	Energi regangan semu (virtual)
W	Usaha virtual beban luar pada elemen

f	matriks operator yang menunjukkan bahwa Vektor u tergantung pada vektor q
V	Volume
K	Matriks Kekakuan
A	Luas Permukaan
L	Panjang
θ	Sudut
T	Matriks Operator penyusun ulang
β	½ (1 - poison's ratio)

TABEL PERSIAPAN DATA UNTUK PROGRAM PSCST

Data	Jumlah Baris	Data yang harus dimasukkan
Data struktur		
a. Identifikasi masalah	1	Judul yang tercantum
b. Parameter struktur	1	NN,NE,NRN,NLS,IPS, E,PR,T
C. Koordinat titik nodal	NN	J,X(J),Y(J)
d. Data elemen	NE	I,IN(I),JN(I),KN(I)
e. Daftar tahanan titik nodal	NRN	K,NRL(2K-1),NRL(2K)
Data beban		
a. Parameter beban	1	NLN,NEL,NEV,NED
b. Beban titik nodal	NLN	K,AN(2K-1),AN(2K)
c. Beban garis	NEL	I,J,BL1,BL2,BL3,BL4
g: Peralihan perletakkan	NED	I,DR(1),DR(2),....., DR(6)

Tabel Notasi Untuk Program

Variable	Definisi
AN()	Gaya Titik Nodal
AR()	Reaksi Perletakan
BL_i	Besar Beban Baris
BV_i	Besar Beban Volume
DN()	Peralihan Titik Nodal
DR(i)	Peralihan Titik Tertahan
E	Modulus Elastisitas
E_i	Konstanta Elastisitas
L,J,K	Penomoran
ID()	Penomoran Peralihan
NI(),JN()	
KN()	Indeks Titik Nodal
IR,IC	Penomoran Baris Dan Kolom
IPS,SPI	Pilihan Tegangan atau Regangan Bidang
ISN	Regangan bidang
LN	Nomor beban
NC()	Jumlah suku nol dalam kolom matrik sky-line SN
ND()	Penomoran diagonal dalam matriks SN
NDF	Jumlah derajat kebebanan
NE	Jumlah elemen
NED	Jumlah elemen peralihan yang diketahui
NEL	Jumlah elemen dengan beban garis
NEV	Jumlah elemen dengan beban volume
NLN	Jumlah elemen dengan beban titik nodal
NLS	Jumlah sistem pembebanan
NN	Jumlah titik nodal
NND	Jumlah peralihan titik nodal

NNR	Jumlah titik nodal tertahan
NRL( )	Daftar tahanan titik nodal
NRN	Jumlah titik nodal tertahan
NS	Jumlah suku dalam matriks skyline SN
PR	Poisson's Ratio
SE( )	Matriks kekakuan elemen
SN( )	Matriks kekakuan titik nodal
SX,SY,...	Tegangan

tabel kegunaan dari SubProgram :

SUBPRO.	Fungsi / Kegunaan
S DATA	Membaca dan mencetak struktur data
STIFF	Membuat skyline yang terdapat pada bagian segitiga atas dari matriks kekakuan titik nodal menjadi variabel yang bersubskrip tunggal dengan cara melakukan kontribusi terhadap matriks kekakuan, selain itu dalam SubProgram ini matriks kekakuan ini diubah untuk menghindari penyusunan kembali (akibat tekanan) sebelum dikirim ke SubProgram skyfac untuk faktorisasi.
L Data	Membaca, mencetak dan memproses beban dan data tambahan yang terdaftar pada tabel data.
SKYFAC	Faktorisasi matriks kekakuan
SKYSOL	Untuk memecahkan persamaan keseimbangan titik nodal sehingga diperoleh peralihan yang dicari
RESUL	Menghitung dan mencetak hasilnya: peralihan titik nodal, tegangan elemen, reaksi titik tertahan.

LAMPIRAN II

PROGRAM PSCST

```
program pscst;
uses crt;
type aa = array[byte]         of real;
     bb = array[byte]         of integer;
     cc = array[1..50,1..50] of real;
     dd = array[1..50,1..50] of integer;
var
an,ar,dn,dr1,dr2,dr3,dr4,dr5,
dr6,x,y,z,sn,dr,jt,tsv,Xni,
Yni,Xjn,Yjn,Xkn,Ykn                  : aa;
id,ni,jn,kn,nc,nd,nrl,jp,psv1,
psv2,psv3,psv4                       : bb;
se                                   : cc;
e,e1,e2,e3,e4,e5,e6,e7,e8,e9,
bl1,bl2,bl3,bl4,bs1,bs2,bs3,
bs4,Xcl,Ycl,bv1,bv2,bv3,bv4,
Slij,pr,sx,sy,sz,sxy,t,ts,Xij,
Yij,Xki,Yki,Xjk,Yjk,Aijk,Xik,Yik   : real;
ndf,ne,ned,nel,nev,nln,nls,nn,
nnd,nnr,nrn,ns,isn,ln,i1,i2,
j1,j2,a,b,k1,k2,spi,n1,n2           : integer;
IPS,XXX                              : CHAR;
judul                                : string[50];
procedure kotak_kecil;
var i : integer;
begin
     gotoxy(11,10);write(chr(201));
     for i:=11 to 69 do write(chr(205));
     write(chr(187));
     For i:=11 to 15 do
     begin{2}
          gotoxy(11,i);write(chr(186));
          gotoxy(71,i);write(chr(186));
     end;{2}
     gotoxy(11,16);write(chr(200));
     for i:=11 to 69 do write(chr(205));
     write(chr(188));TEXTCOLOR(YELLOW+BLINK);
     gotoxy(39,16);write('FAN');TEXTCOLOR(WHITE)
end;
procedure judul1;
begin{1}
     kotak_kecil;
     gotoxy(20,12);
     writeln('ANDA AKAN BEKERJA DENGAN PROGRAM PSCST');
     gotoxy(20,14);
     write('APAKAH ANDA SIAP DENGAN DATA NYA (Y/T)   ');
     XXX :=READKEY;
end;{1}
procedure judul2;
var
i          : integer;
begin{1}
     kotak_kecil;
     gotoxy(33,12);writeln('POKOK PERMASALAHAN');
     gotoxy(20,14);readln(judul);
end;{1}
procedure kotak(x1,y1,x2,y2 :byte);
var i,l,panjang : integer;
```

```
begin{1}
     clrscr;
     panjang:=length(judul);
     l:= 40-panjang div 2;
     gotoxy(x1,y1);write(chr(201));
     for i:=x1+1 to x2-1 do write(chr(205));
     write(chr(187));
     For i:=y1+1 to 6 do
     begin{2}
          gotoxy(x1,i);write(chr(186));
          gotoxy(x2,i);write(chr(186));
     END;{2}
     GOTOXY(X1,Y1+4);WRITE(CHR(204));
     for i:=x1+1 to x2-1 do write(chr(205));
     WRITE(CHR(185));
     for i:=y1+5 to y2-1 do
     begin{3}
          gotoxy(x1,i);write(chr(186));
          gotoxy(x2,i);write(chr(186));
     end;{3}
     gotoxy(x1,y2);write(chr(200));
     for i:=x1+1 to x2-1 do write(chr(205));
     write(chr(188));
     TEXTCOLOR(YELLOW);
     gotoxy(l,Y1+2);writeln(judul);
     gotoxy(39,Y2);write('FAN');
     textcolor(white);
end;{1}
 procedure pisah(X1X,Y1Y,LL : BYTE);
 var i : integer;
 begin
 gotoxy(X1X,Y1Y);writeln(chr(209));
     for i:= 1 to LL do
     begin
     gotoxy(X1X,Y1Y+i);write(chr(179));
     end;
 end;
{ SUB PROGRAM MATRIKS KEKAKUAN STRUKTUR }
procedure matriks_k(e1,e3,e4,xij,yij,
                    xki,yki,xjk,yjk:real;
                    var kaku : cc);
begin
kaku[1,1] :=  e4*(e1*Yjk*Yjk+e3*Xjk*Xjk);
kaku[1,2] := -e4*(pr*Xjk*Yjk+e3*Xjk*Yjk);
kaku[1,3] :=  e4*(e1*Yki*Yjk+e3*Xjk*Xki);
kaku[1,4] := -e4*(pr*Xki*Yjk+e3*Xjk*Yki);
kaku[1,5] :=  e4*(e1*Yij*Yjk+e3*Xjk*Xij);
kaku[1,6] := -e4*(pr*Xij*Yjk+e3*Xjk*Yij);
kaku[2,2] :=  e4*(e1*Xjk*Xjk+e3*Yjk*Yjk);
kaku[2,3] := -e4*(pr*Yki*Xjk+e3*Yjk*Xki);
kaku[2,4] :=  e4*(e1*Xki*Xjk+e3*Yjk*Yki);
kaku[2,5] := -e4*(pr*Yij*Xjk+e3*Yjk*Xij);
kaku[2,6] :=  e4*(e1*Xij*Xjk+e3*Yjk*Yij);
kaku[3,3] :=  e4*(e1*Yki*Yki+e3*Xki*Xki);
kaku[3,4] := -e4*(pr*Xki*Yki+e3*Xki*Yki);
kaku[3,5] :=  e4*(e1*Yij*Yki+e3*Xki*Xij);
kaku[3,6] := -e4*(pr*Xij*Yki+e3*Xki*Yij);
kaku[4,4] :=  e4*(e1*Xki*Xki+e3*Yki*Yki);
kaku[4,5] := -e4*(pr*Yij*Xki+e3*Yki*Xij);
kaku[4,6] :=  e4*(e1*Xij*Xki+e3*Yki*Yij);
kaku[5,5] :=  e4*(e1*Yij*Yij+e3*Xij*Xij);
```

```
kaku[5,6] := -e4*(pr*Xij*Yij+e3*Xij*Yij);
kaku[6,6] :=  e4*(e1*Xij*Xij+e3*Yij*Yij);
end;
 procedure index(ab .        : integer;
                 bc,cd,de    : bb;
                 var  ef     : bb);
 begin
     ef[2]:=2*bc[ab];
     ef[1]:=2*bc[ab]-1;
     ef[4]:=2*cd[ab];
     ef[3]:=2*cd[ab]-1;
     ef[6]:=2*de[ab];
     ef[5]:=2*de[ab]-1;
 end;
procedure SDATA;
procedure parameter_struktur;
var i : integer;
begin{1}
     kotak(5,3,75,24);pisah(40,7,16);
     gotoxy(7,8);writeln('STRUKTUR PARAMETER   :');
     GOTOXY(7,10);WRITE('JUMLAH TITIK NODAL     : ');
     READLN(NN);
     GOTOXY(7,11);WRITE('JUMLAH ELEMEN          : ');
     READLN(NE);
     GOTOXY(7,12);WRITE('JML NODAL TERTAHAN     : ');
     READLN(NRN);
     GOTOXY(7,13);WRITE('JML PEMBEBANAN         : ');
     READLN(NLS);
     GOTOXY(7,14);WRITE('TEGANGAN BIDANG (Y/T)? : ');
     IPS := READKEY;WRITE(IPS);
     GOTOXY(7,15);WRITE('MODULUS ELASTISITAS    : ');
     READLN(E);
     GOTOXY(7,16);WRITE('POISON"S RATIO         : ');
     READLN(PR);
     GOTOXY(7,17);WRITE('KETEBALAN              : ');
     READLN(T);
     if (ips = 'y') or (ips = 'Y') then spi := 0
     else spi := 1;
END;{1}
procedure koordinat_titik_nodal;
var i,j : integer;
begin{1}
     J:=0;
     gotoxy(45,8);writeln('KOORDINAT TITIK NODAL :');
     gotoxy(45,10);writeln('NODAL        X        Y');
     for i:= 1 to NN do
     BEGIN{2}
         J:=J+1;
         gotoxy(46,10+J);write(i:2);
         gotoxy(57,10+J);read(x[i]);
         gotoxy(65,10+J);read(y[i]);
         IF J=13 THEN J:=0;
     END;{2}
end;{1}
procedure data_elemen;
var i,j : integer;
begin{1}
     kotak(5,3,75,24);pisah(40,7,16);
     J:=0;
     gotoxy(7,8);writeln('      DATA ELEMEN    :');
     gotoxy(7,9);writeln('SEGMEN      I      J     K');
     for i:= 1 to NE do
     BEGIN{2}
         J:=J+1;
         gotoxy(9,10+J);write(i:2);
         gotoxy(19,10+J);read(ni[i]);
         gotoxy(26,10+J);read(jn[i]);
         gotoxy(34,10+j);read(kn[i]);
         IF J=13 THEN J:=0;
     END;{2}
end;{1}
procedure tahanan_pada_titik_nodal;
var i,j,k,M : integer;
begin{1}
     gotoxy(42,8);
     writeln('DAFTAR TAHANAN TITIK NODAL  :');
     gotoxy(42,9);writeln('NODAL     R1      R2');
     for k:=1 to nnd do nrl[k]:=0;
     j:=1;
     for i:= 1 to nRn do
     BEGIN{2}
         J:=J+1;
         gotoxy(44,9+J) ;READ(K);
         gotoxy(52,9+J);read(nrl[2*K-1]);
         gotoxy(60,9+J);read(nrl[2*K]);
         IF J=10 THEN J:=0;
     END;{2}
      nnr :=0;
      for j:=1 to nnd do nnr:=nnr+nrl[j];
      ndf:=nnd-nnr;
end;{1}
procedure indeks_perpindahan;
var a,b,i,j,k,l : integer;
begin{1}
     kotak(5,3,75,24);
     for j:= 1 to nnd do nc[j]:=1;
     for i:=1 to ne do
     begin{2}
     index(i,ni,jn,kn,id);
          n1 := nnd;
          for k := 1 to 6 do
          begin{3}
               a:=id[k];
               if (nrl[a] = 0) then
               if (a < n1) then n1:=a;
          end;{3}
          for l := 1 to 6 do
          begin{4}
               b:=id[l];
               if nrl[b] = 0 then
               begin{5}
                    n2:=b-n1+1;
                    if nc[b]<n2 then nc[b]:=n2;
               end;{5}
           end;{4}
     end;{2}
     nd[1] :=1;
     for i := 2 to nnd do nd[i] := nd[i-1] + nc[i];
     ns := nd[nnd];
     gotoxy(20,11);
     write('JUMLAH TITIK NODAL TERTAHAN  = ',NDF:5);
     gotoxy(20,12);
     write('JUMLAH DERAJAD KEBEBASAN      = ',NNR:5);
     gotoxy(20,13);
     write('JML SUKU DLM MATRIKS SKYLINE = ',NS:5);
REPEAT UNTIL KEYPRESSED;
end;{1}
begin  { PROGRAM UTAMA DARI SDATA }
      clrscr;
      judul2;
      PARAMETER_STRUKTUR;
      NND:=2*NN;
      koordinat_titik_nodal;
      data_elemen;
      tahanan_pada_titik_nodal;
      indeks_perpindahan;
end;{akhir program utama prosedure Sdata}
procedure STIFF;{ MATRIKS KEKAKUAN ELEMEN }
label  100;
var i,j,jj,k,ir,ic,item,irc : integer;
    id                      : bb;
begin{1}
for i := 1 to ns do sn[i]:=0.0;
e1 := 1.0;
e2 := e1 - pr;
e3 := e2 / 2.0;
if spi = 1 then
   begin{2}
   e1 := e1 - pr;
   e2 := e1 - pr;
   e3 := e2/2.0;
   end;{2}
i := 0;
repeat
  begin{3}
  i := i+1;
  Xij := X[jn[i]] - X[ni[i]];
  Xik := X[kn[i]] - X[ni[i]];
  Xjk := X[kn[i]] - X[jn[i]];
  Yij := Y[jn[i]] - y[ni[i]];
  Yik := Y[kn[i]] - y[ni[i]];
  Yjk := Y[kn[i]] - y[jn[i]];
  Aijk := ( Xij * Yik - Xik*Yij)/2.0;
  e4 := e * t/(4.0*Aijk *(1.0 + pr)*e2);
  Xki := -Xik ; Yki := -Yik;
  matriks_k(e1,e3,e4,xij,yij,xki,yki,xjk,yjk,se);
  index(i,ni,jn,kn,id);
  for j := 1 to 6 do
     for k:= j to 6 do
        begin{4}
        ir := id[j];
        if nrl[ir] = 1 then goto 100;
        ic := id[k];
        if nrl[ic] = 1 then goto 100;
             if ir > ic then
             begin{6}
                  item := ir;
                  ir   := ic;
                  ic   := item;
             end;{6}
                  irc := nd[ic] - ( ic - ir );
                  sn[irc] := sn[irc] + se[j,k];
 100 :  end;{4}
```

```
    end;{3}
until i >= ne;
for j:= 1 to nnd do
    if nrl[j] = 1 then
    begin{8}
         jj := nd[j];
         sn[jj] := 1.0;
    end;{8}
end;{1}{akhir STIFF}
PROCEDURE SKYFAC;
label 100,200,300;
var j,nj,jj,ncj,nc,i,ncj1,
    ji,jmi,ini,ii,ij,nci,
    ncji,ncd,k,ki,kj,kk        : integer;
    sum,temp                   : real;
begin{1}
for j := 2 to nnd do
begin{2}
j1 := j - 1;
nj := nd[j1];
jj := nd[j];
ncj:= jj - nj;
if ncj = 1 then goto 100;
if j1 = 1 then goto 200;
for i := 2 to j1 do
    begin{3}
    jmi := j - i;
    if ncj <= jmi + 1 then goto 300;
      i1 := i - 1;
      ini := nd[i1];
      nj := nd[j1];
      ii := nd[i];
      ij := jj - jmi;
      nci := ii - ini;
      ncji := ncj - jmi;
      ncd := ncji - nci;
      if ncd < 0 then
        begin{5}
        k1 := ncji - 1;
        ini := ini - ncd;
        end{5}
        else
        begin{6}
        k1 := nci - 1;
        nj := nj + ncd;
        end;{6}
        sum := sn[ij];
        if k1 = 0 then goto 300;
        for k := 1 to k1 do
            begin{7}
            ki := ini + k;
            kj := nj + k;
            sum := sum - sn[ki] * sn[kj];
            end;{7}
        sn[ij] := sum;
300:
end;{3}
200:
sum :=  sn[jj];
ncj1 := ncj - 1;
for k:= 1 to ncj1 do
```

```
    begin{8}
    kj := nd[j1] + k;
    kk := nd[j-ncj+k];
    temp := sn[kj]/sn[kk];
    sum := sum - temp * sn[kj];
    sn[kj] := temp;
    end;{8}
if sum <= 0 then exit;
sn[jj] := sum;
100:
end;{2}
end;{1}
procedure LDATA;
procedure parameter_beban;
var h : integer;
begin{1}
     kotak(5,1,75,25);
     gotoxy(30,5);textcolor(cyan+blink);
     writeln('nomor pembebanan ',ln:2);
     textcolor(white);
     gotoxy(30,6);writeln('PARAMETER BEBAN    :');
     GOTOXY(10,7);
     WRITE('JUMLAH ELEMEN DENGAN BEBAN TITIK NODAL  : ');
     READLN(NLN);
     GOTOXY(10,8);
     WRITE('JUMLAH ELEMEN DENGAN BEBAN GARIS        : ');
     READLN(NEL);
     GOTOXY(10,9);
     WRITE('JUMLAH ELEMEN DENGAN BEBAN VOLUME       : ');
     READLN(NEV);
     gotoxy(5,11);writeln(chr(199));
     for h := 1 to 69 do
     begin
     gotoxy(5+h,11);write(chr(196));
     end;gotoxy(75,11);writeln(chr(182));
END;{1}
procedure beban_titik_nodal;
var i,j,M : integer;
begin{1}
for j:=1 to nnd do an[j] := 0 ;
if nln > 0 then
begin{2}
     i := 0;
     gotoxy(32,12);writeln('BEBAN NODAL   :');
     gotoxy(29,13);writeln('TITIK    AN1    AN2');
     for j := 1 to nLN do
     begin{3}
          i := i + 1;
          gotoxy(30,13+i);READ(M);
          gotoxy(37,13+i);readln(an[2*M-1]);
          gotoxy(44,13+i);readln(an[2*M]);
          if i = 11 then i := 0;
     end;{3}
end;{2}
end;{1}
procedure beban_garis;
var i,a,b,j,h : integer;
BEGIN{1}
if nel > 0 then
   begin{2}
       h := 0;
```

```
         GOTOXY(35,12);writeln('BEBAN GARIS');
         GOTOXY(19,13);write('I      J');
         writeln('      BL1       BL2       BL3       BL4');
         for i := 1 to nel do
         begin{3}
         h:= h +1;
         GOTOXY(19,13+h);READLN(A);
         GOTOXY(24,13+h);READLN(B);
         GOTOXY(31,13+h);READLN(BL1);
         GOTOXY(40,13+h);READLN(BL2);
         GOTOXY(49,13+h);READLN(BL3);
         GOTOXY(58,13+h);READLN(BL4);
         if h =11 then h := 0;
         I1  := 2*A-1;I2:= I1+1;
         J1  := 2*B-1;J2:= J1+1;
         XCL  := X[B] - X[A];
         YCL  := Y[B] - Y[A];
         SLIJ  := SQRT(SQR(XCL)+SQR(YCL));
         AN[I1]:=AN[I1]+BL1*SLIJ/3. + BL2*SLIJ/6.;
         AN[I2]:=AN[I2]+BL3*SLIJ/3. + BL4*SLIJ/6.;
         AN[J1]:=AN[J1]+BL1*SLIJ/6. + BL2*SLIJ/3.;
         AN[J2]:=AN[J2]+BL3*SLIJ/6. + BL4*SLIJ/3.;
         END;{3}
      END;{2}
 END;{1}
 procedure beban_volume;
 var i,j,h : integer;
 begin{1}
 if nev > 0 then
 begin{2}
      h := 0;
      gotoxy(33,12);write('BEBAN VOLUME  :');
      gotoxy(20,13);write('TITIK      ');
      write('BEBAN-1      BEBAN-2');
      for i := 1 to nev do
      begin{3}
           h := h + 1;
           gotoxy(20,13+h);write(i:3);
           gotoxy(30,13+h);readln(bv1);
           gotoxy(42,13+h);readln(bv2);
           if h = 11 then h := 0;
      index(i,ni,jn,kn,id);
      Xij  := X[jn[i]]-X[ni[i]];
      Xik  := X[kn[i]]-X[ni[i]];
      Yij  := Y[jn[i]]-Y[ni[i]];
      Yik  := Y[kn[i]]-Y[ni[i]];
        Aijk  := (Xij*Yik - Xik*Yij)/2.0;
        An[id[1]]  := an[id[1]]  + bv1 * aijk * t / 3.0;
        An[id[2]]  := an[id[2]]  + bv2 * aijk * t / 3.0;
        An[id[3]]  := an[id[3]]  + bv1 * aijk * t / 3.0;
        An[id[4]]  := an[id[4]]  + bv2 * aijk * t / 3.0;
        An[id[5]]  := an[id[5]]  + bv1 * aijk * t / 3.0;
        An[id[6]]  := an[id[6]]  + bv2 * aijk * t / 3.0;
     end{3}
end;{2}
end;{1}
begin  { PROGRAM UTAMA DARI LDATA }
      clrscr;
      textbackground(1);
      PARAMETER_BEBAN;
      BEBAN_titik_nodal;
      BEBAN_GARIS;
      beban_volume;
      READLN;
end;{akhir program utama prosedure LDATA}
procedure SKYSOL;
label 100,200;
var i,ii,nci,ki,kr,k,ik,
    kmi,nk,nck,ncki,ini : integer;
    sum                 : real;
begin{1}
for i:= 1 to nnd do
    begin{2}
    sum := an[i];
    if i = 1 then goto 100;
    i1 := i-1 ; ini := nd[i1];
    ii := nd[i] ; nci := ii - ini;
    k1 := nci - 1 ; kr := i - nci;
    if k1 = 0 then goto 100;
    for k:= 1 to k1 do
      begin{5}
      ki := ini + k;
      kr := kr + 1;
      sum := sum - sn[ki]*dn[kr];
      end;{5}
100: dn[i] := sum;
    END;{2}
    for i := 1 to nnd do
    begin{6}
         ii := nd[i];
         dn[i] := dn[i]/sn[ii];
    end;{6}
    for i1 := 1 to nnd do
    begin{7}
          i := nnd - i1 + 1;
          k2 := i + 1;
          sum := dn[i];
          if i = nnd then goto 200;
            for k := k2 to nnd do
            begin{9}
                k1 := k - 1;
               kmi := k - i;
                nk := nd[k1];
               nck := nd[k] - nk;
              ncki := nck - kmi;
            if ncki > 0 then
                begin{10}
                  ik := nk + ncki;
                 sum := sum - sn[ik] * dn[k];
                end;{10}
            end;{9}
200:  dn[I]:= SUM;
      end;{7}
end;{1}
procedure RESUL;
procedure titik_nodal;
var i,j,M : integer;
begin{1}
     kotak(5,3,75,24);
     gotoxy(30,7);textcolor(cyan+blink);
     writeln('nomor pembebanan ',ln:2);
     textcolor(2);j:=0;
    gotoxy(30,8);writeln('PERALIHAN TITIK NODAL :');
    gotoxy(20,9);
    writeln('NODAL          DN1          DN2');
    for i:= 1 to NN do
    BEGIN{2}
       J:=J+1;
       gotoxy(20,10+J);
       write(i:3,'  ',dn[2*i-1],'  ',dn[2*i]);
       IF J=13 THEN J:=0;
    END;{2}
READLN;
end;{1}
procedure tegangan_elemen;
var i,j,k,it,ip,ir,ic,p,q,r   : integer;
    jt,jp,tsv                 : bb;
    Sx1,Sy1,Sx2,Sy2,Sxy1,Sxy2: real;
begin{1}
    clrscr;
    kotak(2,3,78,24);
    gotoxy(30,7);textcolor(cyan+blink);
    writeln('nomor pembebanan ',ln:2);
    textcolor(2);j:=0;
    gotoxy(30,8);writeln('TEGANGAN ELEMEN :');
    gotoxy(4,9);
    write('elemen     Sx              Sy');
    write('           Sxy             Sz');
    i  := 1;
    while i <= ne do
    begin
    Xij := X[jn[i]] - X[ni[i]];
    Xik := X[kn[i]] - X[ni[i]];
    Xjk := X[kn[i]] - X[jn[i]];
    Yij := Y[jn[i]] - Y[ni[i]];
    Yik := Y[kn[i]] - Y[ni[i]];
    Yjk := Y[kn[i]] - Y[jn[i]];
    Aijk := (Xij * Yik - Xik*Yij)/2.0;
    E5 := e/(2.0*Aijk*(1.0+pr)*e2);
    i1:=2*ni[i]-1;i2:=i1+1;
    j1:=2*jn[i]-1;j2:=j1+1;
    k1:=2*kn[i]-1;k2:=k1+1;
    SX1:=e1*(-Yjk*Dn[i1]+Yik*Dn[j1]-Yij*dn[k1]);
    SX2:=pr*( Xjk*dn[i2]-Xik*dn[j2]+Xij*dn[k2]);
    SX := e5 * ( sx1 + sx2 );
    SY1:=pr*(-Yjk*Dn[i1]+Yik*Dn[j1]-Yij*dn[k1]);
    SY2:=e1*( Xjk*dn[i2]-Xik*dn[j2]+Xij*dn[k2]);
    SY := e5 * ( sy1 + sy2 );
    Sxy1:=Xjk*dn[i1]-Yjk*dn[i2]-Xik*dn[j1];
    Sxy2:=Yik*dn[j2]+Xij*dn[k1]-Yij*dn[k2];
    Sxy := e5 * e3 * ( Sxy1 + Sxy2 );
    sz := pr * ( sx + sy ) * spi;
    gotoxy(4,9+i);
    writeln(i:2,'',sx,' ',sy,' ',sxy,' ',sz);
    e4 := e5 * t / 2.0;
    Xki := -Xik ; Yki := -Yik;
    matriks_k(e1,e3,e4,xij,yij,xki,yki,xjk,yjk,se);
    for p := 2 to 6 do
    begin
    q := p - 1;
    for r := 1 to q do se[p,r] := se[r,p];
    end;
    index(i,ni,jn,kn,id);
```

```
     for j:= 1 to 6 do
     begin{6}
          ir := id[j];
          if nrl[ir] = 1 then
               for k := 1 to 6 do
               begin{8}
                     ic := id[k];
                 ar[ir] := ar[ir] + se[j,k] * dn[ic];
               end{8}
     end;{6}
i := i + 1;
end;
READLN;
end;{1}
procedure reaksi_perletakan;
var j: integer;
begin{1}
     kotak(5,3,75,24);
     gotoxy(30,7);textcolor(cyan+blink);
     writeln('nomor pembebanan ',ln:2);
     textcolor(2);j:=0;
     gotoxy(30,8);writeln('REAKSI PERLETAKAN :');
     gotoxy(20,10);
     writeln('NODAL             AR1             AR2');
     for j:= 1 to nn do
     begin{2}
          j1 := 2 * j - 1;
          j2 := 2 * j;
          n1 := nrl[j1] + nrl[j2];
          if n1 > 0 then
          begin{3}
          gotoxy(20,10+J);
          write(j:3,'   ',ar[j1],'   ',ar[j2]);
          end;{3}
     end;{2}
READLN;
end;{1}
begin {program utama RESUL }
      clrscr;
      titik_nodal;
      tegangan_elemen;
      reaksi_perletakan;
      READLN;
end;
LABEL 100;
VAR I,j : INTEGER;
{PROGRAM UTAMA PSCST}
BEGIN {1}
CLRSCR;
TEXTBACKGROUND(BLUE);
TEXTCOLOR(white);
REPEAT
 BEGIN{2}
 JUDUL1;
 IF (XXX = 'Y') OR (XXX = 'y') THEN
  BEGIN{3}
SDATA;
STIFF;
readln;
IF SN[1] > 0.0  THEN
BEGIN{4}
```

```
SKYFAC;
LN := 1;
WHILE NLS >= LN DO
BEGIN{5}
      LDATA;
      SKYSOL;
      RESUL;
LN := LN + 1;
CLRSCR;
KOTAK_KECIL;
GOTOXY(20,13);
WRITE('MAU MELIHAT HASIL BERIKUTNYA ( Y/T )   ?');
XXX := READKEY;
IF (XXX = 't') OR (XXX = 'T') THEN GOTO 100;
END;{5}
END;{4}
END;{3}
100:
clrscr;
KOTAK_KECIL;
GOTOXY(30,13);WRITE('MAU MENCOBA LAGI ( Y/T ) ');
XXX := READKEY;
CLRSCR;
END;{2}
UNTIL (XXX = 't') OR (XXX = 'T') ;
KOTAK_KECIL;
GOTOXY(30,13);WRITELN('GOOD BLESS YOU AND GOD LUCK');
READLN;
END.{1}
```