

## Penggunaan Metode Elemen Hingga Pada Struktur Grid Dengan Program FreeMat

**Amir Hamzah**

Jurusan Teknik Sipil, Universitas Asahan

email: ah5211011@gmail.com

DOI: <http://dx.doi.org/10.31869/rtj.v4i1.2040>

**Abstrak:** Metode elemen hingga adalah alat yang ampuh untuk menyelesaikan masalah yang kompleks terkait dengan bidang teknik dan bidang lainnya. Metode ini paling banyak digunakan untuk memecahkan masalah analisa struktur. Di rekayasa struktural di mana analisis kritis dengan akurasi tinggi adalah parameter penting dan itu terbukti efektif. Dalam tulisan ini akan dibahas metode elemen hingga khususnya elemen grid menggunakan Program FreeMat. Pada penggunaan metode ini ada beberapa hal yang harus diketahui, yaitu modulus elastisitas E, modulus geser elastisitas G, momen inersia I, konstanta torsional J, dan panjang batang L.

### PENDAHULUAN

Metode elemen hingga adalah prosedur numerik untuk menyelesaikan masalah teknik. Metode elemen hingga adalah alat yang sangat berguna di bidang teknik sipil untuk memperkirakan secara fisik struktur fisik yang terlalu rumit untuk solusi analitis. Metode ini dalam penyelesaian masalah menggunakan pendekatan diskretisasi elemen untuk menemukan perpindahan titik simpul/joint/grid dan gaya-gaya dari struktur. Persamaan yang menggunakan elemen diskret mengacu pada metode matrik untuk analisis struktur dan hasil yang diperoleh identik dengan analisis klasik untuk struktur. Diskretisasi yang dilakukan dapat dilakukan dengan menggunakan elemen satu dimensi (elemen garis), dua dimensi (elemen bidang, ataupun tiga dimensi (elemen solid/kontinum). Pendekatan menggunakan elemen kontinum untuk menentukan pendekatan penyelesaian masalah yang lebih mendekati sebenarnya. Ada enam langkah yang digunakan untuk menyelesaikan setiap masalah menggunakan elemen hingga. Enam langkah analisis elemen hingga sebagai berikut:

#### 1. Diskretisasi

Diskretisasi adalah pernbagian suatu sistem menjadi elemen-elemen. Diskretisasi ini akan menghasilkan suatu harga pendekatan terhadap keadaan sesungguhnya. Jadi bukan merupakan solusi eksak. Massa dibagi menjadi sejumlah elemen yang kecil yang disebut "finite element". Titik potong sisi-sisi elemen disebut titik nodal dan pertemuan.

#### 2. Menulis matriks kekakuan elemen

Persamaan kekakuan elemen perlu ditulis untuk setiap elemen dalam domain menggunakan FreeMat

#### 3. Merakit matriks kekakuan global

Ini akan dilakukan dengan menggunakan pendekatan kekakuan langsung menggunakan FreeMat

#### 4. Menerapkan kondisi batas

Ini seperti penyangga dan beban serta perpindahan yang diterapkan.

#### 5. Memecahkan persamaan

Akan dilakukan dengan mempartisi matriks kekakuan global dan kemudian memecahkan persamaan yang dihasilkan menggunakan eliminasi Gaussian. Dalam tulisan ini proses partisi akan dilakukan secara manual sedangkan bagian solusi akan dilakukan menggunakan FreeMat dengan eliminasi Gaussian.

#### 6. Pasca pemrosesan - untuk mendapatkan informasi tambahan seperti reaksi dan kekuatan elemen akan dilakukan menggunakan FreeMat

Terlihat dari langkah-langkah di atas bahwa proses solusi melibatkan penggunaan kombinasi FreeMat dan beberapa operasi manual terbatas. Operasi manual dikerjakan sangat sederhana hanya berurusan dengan diskritisasi (langkah 1), menerapkan kondisi batas (langkah 4) dan mempartisi matriks kekakuan global (bagian dari langkah 5).

### TINJAUAN PUSTAKA

Persamaan Dasar Elemen grid adalah elemen terbatas dua dimensi dengan koordinat lokal dan global. Elemen grid memiliki modulus elastisitas E, modulus geser elastisitas G, momen inersia I, konstanta

torsional  $J$ , dan panjang  $L$ . Setiap elemen grid memiliki dua node dan cenderung dengan sudut  $\theta$  diukur berlawanan arah jarum jam dari sumbu  $X$  global positif seperti yang ditunjukkan pada gambar 1.1 dan  $C = \cos\theta$  dan  $S = \sin\theta$ . Dalam hal ini, matriks kekakuan elemen diberikan oleh matriks berikut,

$$[k]=[R]^T [k] [R] \quad (1)$$

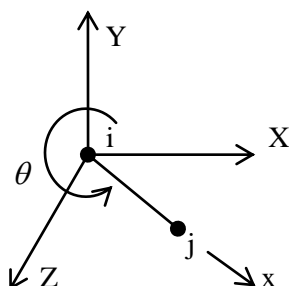
di mana matriks rotasi  $[R]$  dan matriks kekakuan  $[k]$  diberikan oleh:

$$[k]= \begin{bmatrix} \frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} \\ 0 & \frac{GJ}{L} & 0 & 0 & -\frac{GJ}{L} & 0 \\ \frac{6EI}{L^2} & 0 & \frac{4EI}{L} & -\frac{6EI}{L^2} & 0 & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} \\ 0 & -\frac{GJ}{L} & 0 & 0 & \frac{GJ}{L} & 0 \\ \frac{6EI}{L^2} & 0 & \frac{2EI}{L} & -\frac{6EI}{L^2} & 0 & \frac{4EI}{L} \end{bmatrix}$$

(2)

$$[R]= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & C & S & 0 & 0 & 0 \\ 0 & -S & C & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & C & S \\ 0 & 0 & 0 & 0 & -S & C \end{bmatrix}$$

(3)



Gambar 1.1 Grid Elemen

Jelas bahwa elemen grid memiliki enam derajat kebebasan - tiga di setiap node (satu perpindahan dan dua rotasi). Akibatnya untuk struktur dengan  $n$  node, matriks kekakuan global  $K$  akan berukuran  $3n \times 3n$  (karena kita memiliki tiga derajat kebebasan di setiap node). Matriks kekakuan global  $K$  dirakit dengan melakukan panggilan ke fungsi FreeMat GridAssemble yang ditulis khusus untuk Program ini.

Proses ini akan diilustrasikan secara rinci dalam contoh. Setelah matriks kekakuan global diperoleh, kita memiliki persamaan struktur berikut:

$$[K]\{U\} = \{F\} \quad (4)$$

di mana  $U$  adalah vektor perpindahan nodal global dan  $F$  adalah vektor gaya nodal global. Pada langkah ini kondisi batas diterapkan secara manual ke vektor  $U$  dan  $F$ . Kemudian matriks di atas diselesaikan dengan partisi dan eliminasi Gaussian. Akhirnya begitu perpindahan dan reaksi yang tidak diketahui ditemukan, vektor gaya nodal adalah diperoleh untuk setiap elemen sebagai berikut:

$$\{f\} = [k] [R] \{u\} \quad (5)$$

di mana  $\{f\}$  adalah vektor gaya nodal  $6 \times 1$  dalam elemen dan  $u$  adalah vektor perpindahan elemen  $6 \times 1$ . Elemen pertama dalam setiap vektor  $\{u\}$  adalah perpindahan melintang sedangkan elemen kedua dan ketiga adalah rotasi tentang sumbu  $X$  dan  $Z$ , masing-masing, pada simpul pertama, sedangkan elemen keempat dalam setiap vektor  $\{u\}$  adalah perpindahan melintang dan elemen kelima dan keenam adalah rotasi pada sumbu  $X$  dan  $Z$ , masing-masing, pada simpul kedua.

### Fungsi FreeMat yang digunakan.

Ada 5 fungsi FreeMat yang digunakan dalam Grid Element :

1. GridElementLength ( $x_1, y_1, x_2, y_2$ ) - Fungsi ini mengembalikan panjang elemen yang diberikan ( $x_1, y_1$ ) dan koordinat dari node kedua ( $x_2, y_2$ )

function y =

GridElementLength( $x_1, y_1, x_2, y_2$ )

$y = \text{sqrt}((x_2-x_1)*(x_2-x_1) + (y_2-y_1)*(y_2-y_1));$

2. GridElementStiffnessLokal(E,G,I,J,L).  
Fungsi ini menghitung matrik kekakuan elemen untuk setiap elemen grid dengan modulus elastisitas E, geser modulus elastisitas G, momen inersia I, konstanta torsi J, dan panjang L, mengembalikan matriks kekakuan elemen(6x 6) k.  
function y =  
GridElementStiffness(E,G,I,J,L)  
w1=12\*E\*I/(L\*L\*L);  
w2=6\*E\*I/(L\*L);  
w3=G\*I/L;  
w4=4\*E\*I/L;  
w5=2\*E\*I/L;  
y = [w1 0 w2 -w1 0 w2; 0 w3 0 0 -w3  
0;  
w2 0 w4 -w2 0 w5; -w1 0 -w2 w1 0 -  
w2;  
0 -w3 0 0 w3 0; w2 0 w5 -w2 0 w4];
3. GridElementStiffness(E,G,I,J,L,Theta).  
Fungsi ini menghitung matrik kekakuan elemen untuk setiap elemen grid dengan modulus elastisitas E, geser modulus elastisitas G, momen inersia I, konstanta torsi J, dan panjang L, mengembalikan matriks kekakuan elemen(6x 6) k.  
function y =  
GridElementStiffness(E,G,I,J,L,theta)  
x = theta\*pi/180;  
C = cos(x);  
S = sin(x);  
w1=12\*E\*I/(L\*L\*L);  
w2=6\*E\*I/(L\*L);  
w3=G\*I/L;  
w4=4\*E\*I/L;  
w5=2\*E\*I/L;  
kprime= [w1 0 w2 -w1 0 w2; 0 w3 0 0 -  
w3 0;  
w2 0 w4 -w2 0 w5; -w1 0 -w2 w1 0 -w2;  
0 -w3 0 0 w3 0; w2 0 w5 -w2 0 w4];  
R = [1 0 0 0 0 0; 0 C S 0 0 0; 0 -S C 0 0 0;  
0 0 0 1 0 0; 0 0 0 0 C S; 0 0 0 0 -S C];  
y = R'\*kprime\*R;
4. GridAssemble(K,k,i,j)  
Fungsi ini merakit matriks kekakuan elemen k dari elemen grid yang menghubungkan node i dan j ke dalam matriks kekakuan global K. Ia mengembalikan matriks kekakuan global 3n x 3n K setiap kali elemen dirakit.  
function y = GridAssemble(K,k,i,j)  
K(3\*i-2,3\*i-2)= K(3\*i-2,3\*i-2)+ k(1,1);  
K(3\*i-2,3\*i-1)= K(3\*i-2,3\*i-1)+ k(1,2);  
K(3\*i-2,3\*i)= K(3\*i-2,3\*i)+ k(1,3);

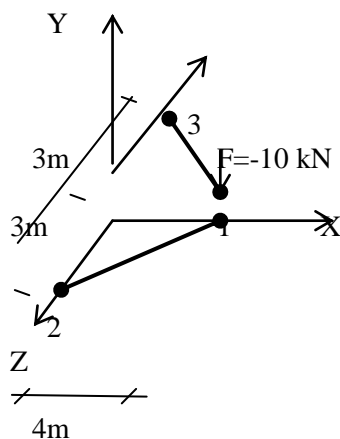
K(3\*i-2,3\*j-2)= K(3\*i-2,3\*j-2)+ k(1,4);  
K(3\*i-2,3\*j-1)= K(3\*i-2,3\*j-1)+ k(1,5);  
K(3\*i-2,3\*j)= K(3\*i-2,3\*j)+ k(1,6);  
K(3\*i-1,3\*i-2)= K(3\*i-1,3\*i-2)+ k(2,1);  
K(3\*i-1,3\*i-1)= K(3\*i-1,3\*i-1)+ k(2,2);  
K(3\*i-1,3\*i)= K(3\*i-1,3\*i)+ k(2,3);  
K(3\*i-1,3\*j-2)= K(3\*i-1,3\*j-2)+ k(2,4);  
K(3\*i-1,3\*j-1)= K(3\*i-1,3\*j-1)+ k(2,5);  
K(3\*i-1,3\*j)= K(3\*i-1,3\*j)+ k(2,6);  
K(3\*i,3\*i-2)= K(3\*i,3\*i-2)+ k(3,1);  
K(3\*i,3\*i-1)= K(3\*i,3\*i-1)+ k(3,2);  
K(3\*i,3\*i)= K(3\*i,3\*i)+ k(3,3);  
K(3\*i,3\*j-2)= K(3\*i,3\*j-2)+ k(3,4);  
K(3\*i,3\*j-1)= K(3\*i,3\*j-1)+ k(3,5);  
K(3\*i,3\*j)= K(3\*i,3\*j)+ k(3,6);  
K(3\*j-2,3\*i-2)= K(3\*j-2,3\*i-2)+ k(4,1);  
K(3\*j-2,3\*i-1)= K(3\*j-2,3\*i-1)+ k(4,2);  
K(3\*j-2,3\*i)= K(3\*j-2,3\*i)+ k(4,3);  
K(3\*j-2,3\*j-2)= K(3\*j-2,3\*j-2)+ k(4,4);  
K(3\*j-2,3\*j-1)= K(3\*j-2,3\*j-1)+ k(4,5);  
K(3\*j-2,3\*j)= K(3\*j-2,3\*j)+ k(4,6);  
K(3\*j-1,3\*i-2)= K(3\*j-1,3\*i-2)+ k(5,1);  
K(3\*j-1,3\*i-1)= K(3\*j-1,3\*i-1)+ k(5,2);  
K(3\*j-1,3\*i)= K(3\*j-1,3\*i)+ k(5,3);  
K(3\*j-1,3\*j-2)= K(3\*j-1,3\*j-2)+ k(5,4);  
K(3\*j-1,3\*j-1)= K(3\*j-1,3\*j-1)+ k(5,5);  
K(3\*j-1,3\*j)= K(3\*j-1,3\*j)+ k(5,6);  
K(3\*j,3\*i-2)= K(3\*j,3\*i-2)+ k(6,1);  
K(3\*j,3\*i-1)= K(3\*j,3\*i-1)+ k(6,2);  
K(3\*j,3\*i)= K(3\*j,3\*i)+ k(6,3);  
K(3\*j,3\*j-2)= K(3\*j,3\*j-2)+ k(6,4);  
K(3\*j,3\*j-1)= K(3\*j,3\*j-1)+ k(6,5);  
K(3\*j,3\*j)= K(3\*j,3\*j)+ k(6,6);

- y = K;
5. GridElementForce(E,G,I,J,L,Theta,u)  
Fungsi ini menghitung vektor gaya elemen menggunakan modulus elastisitas E, geser modulus elastisitas G, momen inersia I, konstanta torsi J, dan panjang L dan vektor perpindahan elemen u. mengembalikan 6 x 1 elemen gaya vektor f.  
function y =  
GridElementForce(E,G,I,J,L,theta,u)  
x = theta\*pi/180;  
C = cos(x);  
S = sin(x);  
w1=12\*E\*I/(L\*L\*L);  
w2=6\*E\*I/(L\*L);  
w3=G\*I/L;  
w4=4\*E\*I/L;  
w5=2\*E\*I/L;  
kprime=[w1 0 w2 -w1 0 w2; 0 w3 0 0 -  
w3 0;  
w2 0 w4 -w2 0 w5; -w1 0 -w2 w1 0 -w2;  
0 -w3 0 0 w3 0; w2 0 w5 -w2 0 w4];

```
w2 0 w4 -w2 0 w5; -w1 0 -w2 w1 0 -
w2;
0 -w3 0 0 w3 0; w2 0 w5 -w2 0
w4];
R = [1 0 0 0 0; 0 C S 0 0 0; 0 -S C 0 0 0;
0 0 0 1 0 0; 0 0 0 0 C S; 0 0 0 0 -S C];
y = kprime*R*u;
```

**Contoh Perhitungan**

Diketahui struktur grid seperti gambar 1.2. Ditentukan E = 210 Gpa, modulus geser, G = 84 Gpa, Inersia I = 20 X 10<sup>-5</sup> m<sup>4</sup> dan konstanta torsional J = 5 x 10<sup>-5</sup> m<sup>4</sup>.



Gambar 1.2 Struktur grid

```
% Program Grid Elemen 02
%-----;
% No.batang Titik-i Titik-j Pjg batang
%-----;
% 1 1 2 5.0 ;
% 2 1 3 5.0 ;
%-----;
```

E = 210e6 % kN/mm2 = 210 GPa = 210e6 kN/m2

G = 84e6

I = 20e-5

J = 5e-5

L1 = GridElementLength(4,0,0,-3)

L2 = GridElementLength(4,0,0,3)

theta1 = 180-atan(3/4)\*180/pi

theta2 = 180+atan(3/4)\*180/pi

% Kekakuan Lokal

k11 = GridElementStiffnessLokal(E,G,I,J,L1)

k12 = GridElementStiffnessLokal(E,G,I,J,L2)

% Kekakuan Global

k1 = GridElementStiffness(E,G,I,J,L1,theta1)

k2 = GridElementStiffness(E,G,I,J,L2,theta2)

% Merakit Matrik

K = zeros(9,9)

K = GridAssemble(K,k1,1,2)

K = GridAssemble(K,k2,1,3)

k = K(1:3,1:3)

f = [-10;0;0]

u = k\f

% Gaya btg global

U = [u;0;0;0;0;0;0]

F = K\*U

% Gaya batang Lokal

u1 = [U(1) ; U(2) ; U(3) ; U(4) ; U(5) ; U(6)]

u2 = [U(1) ; U(2) ; U(3) ; U(7) ; U(8) ; U(9)]

% Gaya batang

f1 = GridElementForce(E,G,I,J,L1,theta1,u1)

f2 = GridElementForce(E,G,I,J,L2,theta2,u2)

**Hasil keluarannya :**

E = 210000000

G = 84000000

I = 2.0000e-004

J = 5.0000e-005

L1 = 5

L2 = 5

theta1 = 143.1301

theta2 = 216.8699

**Kekakuan Lokal**

k11 =

4032 0 10080 -4032 0 10080

0 840 0 0 -840 0

10080 0 33600 -10080 0 16800

-4032 0 -10080 4032 0 -10080

0 -840 0 0 840 0

10080 0 16800 -10080 0 33600

k12 =

4032 0 10080 -4032 0 10080

0 840 0 0 -840 0

10080 0 33600 -10080 0 16800

-4032 0 -10080 4032 0 -10080

0 -840 0 0 840 0

10080 0 16800 -10080 0 33600

**Kekakuan global**

k1 =

1.0e+004 \*

0.4032 -0.6048 -0.8064 -0.4032

-0.6048 1.2634 1.5725 0.6048

-0.8064 1.5725 2.1806 0.8064

-0.4032 0.6048 0.8064 0.4032

-0.6048 0.5510 0.8467 0.6048

-0.8064 0.8467 1.0450 0.8064

-0.6048 -0.8064

0.5510 0.8467

0.8467 1.0450  
 0.6048 0.8064  
 1.2634 1.5725  
 1.5725 2.1806  
 k2 =  
 1.0e+004 \*  
 0.4032 0.6048 -0.8064 -0.4032  
 0.6048 1.2634 -1.5725 -0.6048  
 -0.8064 -1.5725 2.1806 0.8064  
 -0.4032 -0.6048 0.8064 0.4032  
 0.6048 0.5510 -0.8467 -0.6048  
 -0.8064 -0.8467 1.0450 0.8064  
 0.6048 -0.8064  
 0.5510 -0.8467  
 -0.8467 1.0450  
 -0.6048 0.8064  
 1.2634 -1.5725  
 -1.5725 2.1806

**Merakit matrik**

1.0e+004 \*  

$$\begin{pmatrix} 0.8064 & -0.0000 & -1.6128 & -0.4032 \\ -0.0000 & 2.5267 & 0.0000 & 0.6048 \\ -1.6128 & 0.0000 & 4.3613 & 0.8064 \\ -0.4032 & 0.6048 & 0.8064 & 0.4032 \\ -0.6048 & 0.5510 & 0.8467 & 0.6048 \\ -0.8064 & 0.8467 & 1.0450 & 0.8064 \\ -0.4032 & -0.6048 & 0.8064 & 0 \\ 0.6048 & 0.5510 & -0.8467 & 0 \\ -0.8064 & -0.8467 & 1.0450 & 0 \end{pmatrix}$$

-0.6048 -0.8064 -0.4032 0.6048 -0.8064  
 0.5510 0.8467 -0.6048 0.5510 -0.8467  
 0.8467 1.0450 0.8064 -0.8467 1.0450  
 0.6048 0.8064 0 0 0  
 1.2634 1.5725 0 0 0  
 1.5725 2.1806 0 0 0  
 0 0 0.4032 -0.6048 0.8064  
 0 0 -0.6048 1.2634 -1.5725  
 0 0 0.8064 -1.5725 2.1806

$$\begin{pmatrix} U_{1y} \\ \phi_{1x} \\ \phi_{1z} \\ U_{2y} \\ \phi_{2x} \\ \phi_{2z} \\ U_{3y} \\ \phi_{3x} \\ \phi_{3z} \end{pmatrix} = \begin{pmatrix} F_{1y} \\ M_{1x} \\ M_{1z} \\ F_{2y} \\ M_{2x} \\ M_{2z} \\ F_{3y} \\ M_{3x} \\ M_{3z} \end{pmatrix}$$

1.0e+004 \*  

$$\begin{pmatrix} 0.8064 & -0.0000 & -1.6128 & -0.4032 \\ -0.0000 & 2.5267 & 0.0000 & 0.6048 \\ -1.6128 & 0.0000 & 4.3613 & 0.8064 \\ -0.4032 & 0.6048 & 0.8064 & 0.4032 \\ -0.6048 & 0.5510 & 0.8467 & 0.6048 \\ -0.8064 & 0.8467 & 1.0450 & 0.8064 \\ -0.4032 & -0.6048 & 0.8064 & 0 \\ 0.6048 & 0.5510 & -0.8467 & 0 \\ -0.8064 & -0.8467 & 1.0450 & 0 \end{pmatrix}$$
  
 -0.6048 -0.8064 -0.4032 0.6048 -0.8064  
 0.5510 0.8467 -0.6048 0.5510 -0.8467  
 0.8467 1.0450 0.8064 -0.8467 1.0450  
 0.6048 0.8064 0 0 0  
 1.2634 1.5725 0 0 0  
 1.5725 2.1806 0 0 0  
 0 0 0.4032 -0.6048 0.8064  
 0 0 -0.6048 1.2634 -1.5725  
 0 0 0.8064 -1.5725 2.1806

$$\begin{pmatrix} U_{1y} \\ \phi_{1x} \\ \phi_{1z} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -10 \\ 0 \\ 0 \\ F_{2y} \\ M_{2x} \\ M_{2z} \\ F_{3y} \\ M_{3x} \\ M_{3z} \end{pmatrix}$$

**Penyelesaian persamaan :**

$$1.0e+004 * \begin{pmatrix} 0.8064 & -0.0000 & -1.6128 & U_{1y} \\ -0.0000 & 2.5267 & 0.0000 & \phi_{1x} \\ -1.6128 & 0.0000 & 4.3613 & \phi_{1x} \end{pmatrix} \begin{pmatrix} 10 \\ 0 \\ 0 \end{pmatrix}$$

k =  
 1.0e+004 \*  
 0.8064 -0.0000 -1.6128  
 -0.0000 2.5267 0.0000  
 -1.6128 0.0000 4.3613  
 f =  
 -10  
 0  
 0

**Perpindahan Nodal 1**

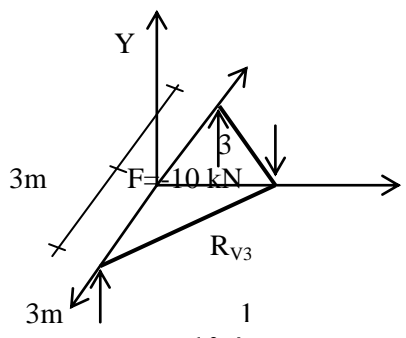
u =  
 1.0e-003 \*  
 -4.7622  
 -0.0000  
 -1.7611

U =  
 1.0e-003 \*  
 -4.7622  
 -0.0000  
 -1.7611  
 0  
 0  
 0  
 0  
 0  
 0  
 0

f2 =  
 -5.0000  
 0.8876  
 -0.6657  
 5.0000  
 -0.8876  
 -24.3343

**Vektor gaya pada Nodal:**

F =  
 -10.0000  
 0.0000  
 0  
 5.0000  
 13.8905  
 20.0000  
 5.0000  
 -13.8905  
 20.0000



$R_{v1} = -10 \text{ kN}$   
 $M_{1x} = 0$   
 $M_{1z} = 0$   
 $R_{v2} = 5 \text{ kN}$   
 $M_{2x} = 13.8905 \text{ kNm}$   
 $M_{2z} = 20.0 \text{ kNm}$   
 $R_{v3} = 5 \text{ kN}$   
 $M_{3x} = -13.8905 \text{ kNm}$   
 $M_{3z} = 20.0 \text{ kNm}$

Z 2  
 $R_{v2}$

**Deformasi batang 1**

u1 =  
 1.0e-003 \*  
 -4.7622  
 -1.7611  
 0  
 0  
 0

**Deformasi batang 2**

u2 =  
 1.0e-003 \*  
 -4.7622  
 -0.0000  
 -1.7611  
 0  
 0  
 0

**Gaya batang**

f1 =

**SIMPULAN**

Hasil perhitungan dengan FreeMat cukup akurat dan lebih mudah untuk melakukan variasi-variasi yang diperlukan. Dalam penyelesaian struktur grid menggunakan FreeMat terlebih dahulu harus dicari kekakuan lokal dan global. Hasil perhitungan diatas sudah di validasi dan sesuai dengan Program SAP2000.

**REFERENSI**

P.I. Kattan, 2002. "MATLAB Guide to Finite Elemen", Springer – Verlag.  
 Wiryanto Dewo Broto, 2007. "Aplikasi Rekayasa Konstruksi dengan SAP 2000", Elex Media Komputindo.  
 Fausett.L, 1999. "Applied Numerical Analysis Using MATLAB", Prentice Hall.  
 George R.Buchanan, 1995. "Finite Element Analysis", McGraw- Hill, INC.  
 Larry J. Segerlind, 1984."Applied Element analysis", John Wiley and Sons. Inc.  
 Amar Khennane, 2013."Introduction to Finite Element Analysis Using MATLAB and Abaqus", CRC Press Taylor and Francis Group.  
 Yoyong Arfiadi, 2016."Analisis Struktur dengan Program Matlab dan FreeMat", Cahaya Atma Pustaka.  
<https://it.wikipedia.org/wiki/FreeMat>