B A B I I

TEORI PENUNJANG

II.1. Pengenalan Bahasa Pascal

Bahasa Pascal adalah salah satu diantara bahasa-bahasa pemrograman yang termasuk dalam bahasa tingkat tinggi (High Level Language) dalam pemrograman. Bahasa ini diciptakan oleh Niklaus Wirth (1970) dari Technical University di Zurich, Switzerland. Kompiler bahasa yang pertama kali diperkenalakan untuk Komputer CDC 6000 ini khusus ditujukan untuk membuat bahasa pemrograman yang mudah dipelajari dan terstruktur. Bahasa ini banyak dipakai, karena terutama bersifat Interaktif. Selain itu Turbo Pascal versi 5.0 yang penulis gunakan dalam penyusunan Tugas Akhir ini, mengikuti definisi Standart Pascal yang didefinisikan oleh K Jansen dan Niklaus Wirth di Pascal User Manual dan Report. Turbo Pascal ini adalah Copyright oleh Borland Inc. dan dapat digunakan pada Sistem operasi PC-DOS, MS-DOS, CPM-86 dan CP/M-80. Beberapa keutamaan dari bahasa Pascal antara lain adalah :

a> Proses Kompilasi dan Eksekusi yang relatif cepat.

b> Pada pemrograman yang sangat besar, maka dapat dibagi-bagi menjadi beberapa program yang lebih kecil yang disebut UNIT. Tiap Unit bisa dikompile secara terpisah, tanpa perlu mengkompile ulang program secara keseluruhan lagi. Setelah seluruh Unit terkompile, maka Unit-unit tersebut bisa digabung lagi menjadi Program yang utuh seperti semula. Selain Unit-unit yang dibuat sendiri, Turbo Pascal ini juga menyediakan Unit-unit Standard yang bisa langsung digabungkan dengan program yang kita miliki. Hal ini megakibatkan logika program menjadi lebih mudah untuk dipelajari.

c> Unit-unit yang tersedia cukup lengkap, yaitu System DOS, CRT dan Graph

c> Kesalahan-kesalahan yang terjadi dalam program mudah untuk ditelusuri. Berita kesalahan ditunjukkan baik pada saat kompilasi maupun pada saat eksekusi.

d> Mudah untuk melakukan modifikasi tanpa menimbulkan akibat terhadap blok yang lainnya. Sebab dalam program yang terstruktur dapat dibatasi effek antar blok.

e> Dapat menggunakan dan berhubungan dengan bahasa mesin.

f> Dapat menggunakan Include File sampai dengan 8 Tingkat.

g> Turbo Pascal dapat menampung SubProgram yang tersusun dari Procedure dan atau Function maksimal 512 kali untuk tiap modulnya.

h> Bisa mengkompile suatu file menjadi program yang berextension EXE hingga lebih besar dari 64 Kb.

i> Mempunyai kompiler Directive untuk kompilasi bersyarat.

Dan masih banyak lagi keistimewaan lain yang ada pada pascal, misal tipe data sederhana pada Elemen bahasa Pascal ini yang akan penulis uraikan nanti.

II.1.1. Struktur Bahasa Pascal.

struktur program bahasa Pascal terdiri atas :

a. Kepala Program (Program Heading).

b. Blok Program (Body Program)

Yang terdiri dari :

1. Bagian Deklarasi (Deklarations).

   yang bisa terdiri dari :

   - deklarasi label.
   - deklarasi konstanta.
   - deklarasi tipe.
   - deklarasi variable.
   - deklarasi procedure
   - deklarasi fungsi

2. Bagian Pernyataan (Statements).

Untuk lebih jelasnya dapat kami gambarkan sebagai berikut :

```
BAGIAN PROGRAM

    BAGIAN DEKLARASI

    BAGIAN PERNYATAAN
    begin

    end
```

II.1.2. Tipe Data

Salah satu dari elemen dasar pada bahasa Pascal tipe data Sederhana, yang mempunyai keistimewaan sendiri, jika dibandingkan dengan bahasa pemrograman lain. Berikut ini akan penulis uraikan tipe-tipe tersebut dan perbandingannya dengan bahasa Basic dan Fortran.

Pascal :

a. Tipe Integer (Integer Types).

Shortint	-128 s/d 127
Integer	-32768 s/d 32767
Longint	-2147483648 s/d 2147483647
Byte	0 s/d 255
Word	0 s/d 65535

b. Tipe Real (Real Types)

Real	$2.9 * 10^{-39}$	s/d $1.7 * 10^{38}$
Singgle	$1.5 * 10^{-45}$	s/d $3.4 * 10^{38}$
Double	$5.0 * 10^{-324}$	s/d $1.7 * 10^{324}$
Extended	$3.4 * 10^{-4932}$	s/d $1.1 * 10^{4932}$

Untuk tingkat ketelitiannya :

Real	11 s/d 12 digit
Singgle	7 s/d 8 digit
Double	15 s/d 16 digit
Extended	19 s/d 20 digit

Basic :

a. Tipe Integer

Hanya ada satu ukuran yaitu :

Integer	-32768 s/d 32767

b. Tipe Real

Singgle Precision -1.7 * E+38 s/d 1.7 * E+38

Double Precision -1.7 * E+324 s/d 1.7 * E+324

Untuk Tingkat ketelitiannya :

Singgle 7 s/d 8 digit

Double15 s/d 16 digit

Fortran :

a. Tipe Integer

Terdefinisi -32768 s/d 32767

Tak didefinisikan -2147483648 s/d 2147483647

b. Tipe Real

Singgle Precision -1.7 * E+38 s/d 1.7 * E+38

Double Precision -1.7 * E+324 s/d 1.7 * E+324

Untuk Tingkat ketelitiannya :

Singgle 7 s/d 8 digit

Double15 s/d 16 digit

II.1.3. SUB PROGRAM

II.1.3.1. Procedure

Didalam pascal terdapat dua maca subroutine (subprogram) yang dinamakan procedure dan function. Perbedaan antara keduanya adalah untuk function akan mempunyai nilai sehingga function harus mempunyai type, sedang procedure tidak. Dengan struktur **procedure** dan **function** inilah sebetulnya letak kemudahan dan keluwesan Pascal.Sebab dengan procedure atau function ini dapat dibuat subprogram yang boleh dikatakan bebas sama sekali dengan subprogram lainnya, sehingga pada pembuatan program yang besar dapat dibagi-bagi menjadi procedure-procedure yang dapat dikerjakan secara terpisah dan jika terdapat kesalahan atau akan dilakukan modifikasi cukup pada subprogram yang dikehendaki.

Parameter adalh identifier yang digunakan untuk melewatkan nilai (jika ada) dari luar subprogram kedalam subprogram atau sebaliknya. Parameter ada 3 jenis :

Value Parameter (parameter nilai)

Variable parameter (parameter variabel bertipe)

Untype variable parameter

Ditinjau dari letaknya parameter dibedakan : **parameter Formal** yaitu parameter yang terdapat dalam tanda kurung setelah nama procedure pada deklarasi procedure, **parameter Aktual** adalah paramter yang berada didalam tanda kurung dibelakang nama procedure atau function pada saat procedure atau function dipanggil. Deklarasi procedure persis seperti pada deklarasi program. Selain itu perlu juga dipahami tentang variabel global dan variabel lokal.

Variabel Global : variabel yang dapat berlaku diseluruh program.

Variabel Lokal: variabel yang hanya berlaku pada tempat dia dideklarasikan.

Deklarasi Procedure :

```
procedure nama (daftar parameter formal);
   (bagian deklarasi)
   begin
     .
   end;
```

II.1.3.2. Function

Struktur suatu fungsi hampir sama dengan procedure, tetapi karena fungsi harus menghasilkan suatu nilai maka nama fungsi harus memiliki tipe.

Bentuk umum seperti pada diagram sintaks :

```
Function [nama] ([parameter normal]):[tipe hasil];
     Begin
          <bagian pernyataan>
     End;
```

II.1.4. Rekursi

Salah satu keunggulan yang dimiliki oleh bahasa pemrograman Pascal adalah adanya proses yang disebut Rekursi. Rekursi ini dimaksudkan adalah proses yang memanggil dirinya sendiri. Dalam rekursi sebenarnya terkandung pengertian fungsi dan procedure, tetapi procedure dan fungsi harus dipanggil melalui pemanggil sedangkan rekursi dapat memanggil dirinya sendiri.

Karena rekursi dapat memanggil dirinya sendiri berarti rekursi merupakan proses berulang yang belum diketahui kapan berhentinya. Dalam pemrograman rekursi digunakan dengan cara mengekspresikannya ke dalam suku-suku dari program lain dengan menambahkan langkah-langkah sejenis.

FUNGSI STANDARD PASCAL

ABS(x)	: Harga absolut dari x, ABS(-5) = 5
Arctan(x)	: Arcustangent dari x
Cos(x)	: Cosinus dari x
Dec(x)	: Nilai x dikurangi 1,

misal x = 4, Dec(x) = 3

Exp(x)	:	e pangkat x
Frac(x)	:	Bagian fraksional dari x, misal frac(32.76) = .76
Int(x)	:	Bagian integer dari x, misal int(12.45) = 12
Inc(x)	:	Nilai x ditambah satu
Ln(x)	:	Logaritma natural dari x
Odd	:	Test jika argumen bilangan ganjil
Pi	:	Harga 3.1415926535897932385
Pred(x)	:	Predesessor dari x
Random(x)	:	Bilangan acak dari 0 sampai dengan x
Round(x)	:	Integer terdekat ke x
Sin(x)	:	sinus dari x
SQR	:	Pangkat dua, SQR(2) = 4
SQRT	:	Akar, SQRT(16) = 4
CHR	:	Mencari kode ASCII, CHR(66) = B
ORD	:	Mencari arti kode ASCII, ORD(B) = 66
TRUNCH	:	Pembulatan kebawah, TRUNCH(10/3) = 3
Upcase	:	Merubah semua argumen karakter menjadi huruf besar, Upcase(y) = Y

PROCEDURE STANDART

EXIT	:	Keluar dari suatu blok
Halt	:	Menghentikan proses program
Move	:	Menyalinkan suatu blok sebanyak count byte memory dari blok dimulai Byte pertama source dan disalinkan ke Byte pertama dan seterusnya.

Syntak : Move(Var source,dst;count :

word)

Fillchar : Mengisi sejumlah byte nilai kedalam suatu variabel.

Syntak : Fillchar(x;count: word; char)

contoh pemrograman Bahasa Pascal :

2. Determinan dengan menggunakan Metode Chio

Apabila diketahui suatu matriks **A** dengan ordo n x n.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & & a_{3n} \\ a_{41} & a_{42} & a_{43} & a_{44} & & a_{4n} \\ & & & & & \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & & a_{nn} \end{bmatrix}$$

maka | A | dengan menggunakan metode Chio adalah sebagai berikut :

akan diuraikan bagaimana metode tersebut didapat dengan gunakan sifat-sifat determinan 3 yaitu harga determinan akan k * |A| apabila salah satu baris/kolom dari |A| dikalikan dengan k dan sifat determinan yang ke 4 yaitu harga determinan |A| tidak berubah apabila baris/kolom ke-i ditambah dengan k baris/kolom ke-j

$$= \frac{1}{a_{11}^{\,n-1}} \begin{vmatrix} a_{11} & a_{11}a_{12} & a_{11}a_{13} & a_{11}a_{14} & & a_{11}a_{1n} \\ a_{21} & a_{11}a_{22} & a_{11}a_{23} & a_{11}a_{24} & & a_{11}a_{2n} \\ a_{31} & a_{11}a_{32} & a_{11}a_{33} & a_{11}a_{34} & & a_{11}a_{3n} \\ a_{41} & a_{11}a_{42} & a_{11}a_{43} & a_{11}a_{44} & & a_{11}a_{4n} \\ & & & & & \\ a_{n1} & a_{11}a_{n2} & a_{11}a_{n3} & a_{11}a_{n4} & & a_{11}a_{nn} \end{vmatrix}$$

$$= \frac{1}{a_{11}^{n-1}} \begin{vmatrix} a_{11} & 0 & 0 & 0 & 0 \\ a_{21} & a_{11}a_{22}-a_{12}a_{21} & a_{11}a_{23}-a_{13}a_{21} & a_{11}a_{24}-a_{14}a_{21} & a_{11}a_{2n}-a_{1n}a_{21} \\ a_{31} & a_{11}a_{32}-a_{12}a_{31} & a_{11}a_{33}-a_{13}a_{31} & a_{11}a_{34}-a_{14}a_{31} & a_{11}a_{3n}-a_{1n}a_{31} \\ a_{41} & a_{11}a_{42}-a_{12}a_{41} & a_{11}a_{43}-a_{13}a_{41} & a_{11}a_{44}-a_{14}a_{41} & a_{11}a_{4n}-a_{1n}a_{41} \\ a_{n1} & a_{11}a_{n2}-a_{12}a_{n1} & a_{11}a_{n3}-a_{13}a_{n1} & a_{11}a_{n4}-a_{14}a_{n1} & a_{11}a_{nn}-a_{1n}a_{n1} \end{vmatrix}$$

apabila a_{11} dikeluarkan maka akan didapatkan :

$$= \frac{1}{a_{11}^{n-2}} \begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{1n} \\ a_{21} & a_{2n} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{1n} \\ a_{31} & a_{3n} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{1n} \\ a_{41} & a_{4n} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{n1} & a_{n2} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{n1} & a_{n3} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{n1} & a_{n4} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{1n} \\ a_{n1} & a_{nn} \end{vmatrix} \end{vmatrix}$$

Contoh soal :

$$M = \begin{bmatrix} 4 & 2 & 1 & 5 \\ 3 & 2 & 4 & 2 \\ 1 & 6 & 2 & 1 \\ 2 & 3 & 1 & 4 \end{bmatrix}$$

maka | M | adalah :

$$\begin{vmatrix} 4 & 2 & 1 & 5 \\ 3 & 2 & 4 & 2 \\ 1 & 6 & 2 & 1 \\ 2 & 3 & 1 & 4 \end{vmatrix} = \frac{1}{4^2} \begin{vmatrix} 2 & 13 & -7 \\ 22 & 7 & -1 \\ 8 & 2 & 6 \end{vmatrix}$$

$$= \frac{1}{16} * \frac{1}{2} * \begin{vmatrix} -272 & 152 \\ 100 & 68 \end{vmatrix}$$

$$= -103$$

Apabila digunakan program pascal sebagai berikut :

```
Program Chio;
uses crt;
type
x = array[1..25,1..25] of real;
Var
  A,B : x ;
  m,n,I,J,K : Integer;
  DET,P : real;
Procedure Baca_Data;
Begin
Write ('Isikan besar ordonya, max 25  ');
Readln (N);
clrscr;
k:=1;
For I:= 1 to N do
    For J:= 1 to N do
    begin
    gotoxy(3*j,i);Readln(A[I,J]);
    k:= k+1;
    End;
end;
procedure Determinan;
Begin
P:= 1;M:= N;
For I:= N downto 3 do
    Begin
    P:= P*(1/Exp((i-2)*ln(a[1,1])));
    For J:= 1 to I-1 do
        For K:= 1 to I-1 do
  B[J,K]:=(a[1,1]*A[j+1,k+1])-(A[1,k+1]*A[j+1,1]);
```

```
31        For J:= 1 to I-1 do
32            For k:= 1 to I-1 do
33        A[J,K]:=B[J,k]
34        End;
35        DET:= P*((A[1,1]*A[2,2]) - (A[1,2]*A[2,1]));
36        clrscr;
37        gotoxy(25,10);
38        Writeln('Determinan yang dicari : ',DET:10:4);
39    end;
41    Begin{program utama}
42         clrscr;
43         textbackground(1);
44         textcolor(3);
45         Baca_Data;
46         determinan;
47         repeat until keypressed;
48    End.{ akhir program utama }
```

Hasilnya :

```
Isikan besarnya ordo, max 25 : 4
4  2  1  5
3  2  4  2
1  6  2  1
2  3  1  4

Determinan Yang dicari : -103.0000
```

keterangan program :

Baris	Keterangan
1	Judul Program

2	Memanggil Unit CRT
3-8	Deklarasi Global
9-21	SubProgram procedure untuk membaca data besar-ordo dan data matriks M
22-39	SubProgram Procedure Determinan dengan rincian
27	- $1/a_{11}{}^{n-2}$
28-32	- Proses Penciutan Matriks M sehingga menjadi Matriks ordo 2
35-38	- Mencari dan Mencetak harga Determinan yang Sudah dimidifikasi
41-48	- Program Utama

II.2. Konsep Elemen Hingga

Metode Elemen Hingga merupakan salah satu metode pendekatan yang dapat digunakan dalam analisa Numerik suatu kontinum. Bila suatu kontinum dibagi bagi menjadi beberapa bagian yang lebih kecil dan berhingga, maka bagian-bagian kecil ini disebut Elemen Hingga. Bagian - bagian kecil ini biasanya memiliki bentuk geometri yang lebih sederhana dibandingkan dengan kontinumnya. Dengan analisa pada ukuran elemen yang berhingga tersebut memungkinkan metode ini dapat melakukan perubahan persoalan dari jumlah derajad kebebasan yang tak hingga menjadi berhingga.

Analisa dengan metode elemen hingga umumnya didasarkan pada asumsi-asumsi terhadap fungsi perpindahan (Displacement), fungsi tegangan (stress) atau keduanya. Secara umum asumsi yang pertama kali diambil adalah menentukan bentuk fungsi perpindahan terhadap geometri elemen. Untuk lebih jelasnya, langkah-langkah yang dipakai pada analisa struktur dengan menggunakan metode elemen hingga adalah sebagai berikut:

1 Membagi kontinum menjadi beberapa elemen dengan jumlah tertentu dan dengan geometri yang sederhana.

2 Memilih letak dan jumlah nodal pada elemen dari data yang sudah ada dimana kondisi kesetimbangannya akan didapat.

3 Mengasumsikan fungsi peralihan pada setiap elemen sedemikian hingga peralihan pada tiap titik sembarang dipengaruhi oleh titik nodalnya.

4 Pada setiap elemen khusus yang terpilih, harus dipenuhi persyaratan hubungan regangan-peralihan dan hubungan tegangan-regangan.

5 Menentukan kekakuan dan beban titik nodal ekivalen untuk setiap elemen dengan menggunakan prinsip usaha atau prinsip energi.

6 Menurunkan persamaan keseimbangan untuk setiap titik nodal dari diskretisasi kontinum ini sesuai dengan kontribusi elemennya.

7 Menyelesaikan persamaan keseimbangan, untuk mendapatkan perpindahan titik nodal.

8 Menghitung tegangan elemen pada titik-titik yang diinginkan.

9 Menghitung gaya reaksi pada nodal yang ditahan (jika perlu)

Langkah-langkah tersebut didasarkan pada langkah-langkah umum dalam Metode Elemen Hingga sebagai berikut:

1 Diskretisasi dan memilih konfigurasi Elemen.

2 Memilih model atau fungsi pendekatan.

3 Menentukan hubungan Regangan-Tegangan.

4 Menurunkan persamaan-persamaan Elemen.

5 Peralihan persamaan Elemen untuk mendapatkan Persamaan Global atau Persamaan Rakitan dan mengenal syarat batas.

6 Memecahkan besaran-besaran primer yang tak diketahui.

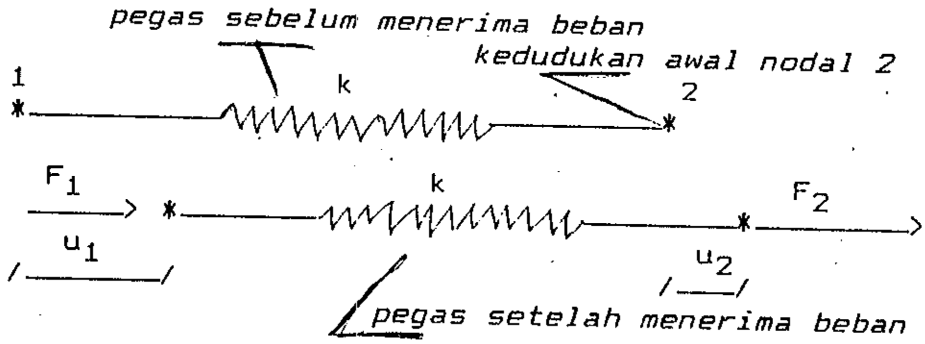
7 Memecahkan besaran-besaran penurunan (sekunder).

8 Intepretasi hasil-hasil.

II.3. Matriks Kekakuan Lokal Elemen Sederhana

Elemen rangka batang dapat dianggap sebagai pegas linier. Keduatitik ujungnya merupakan titik nodal yang dapat menerima gaya dan perpindahan (displacement) dalam arah sumbunya .

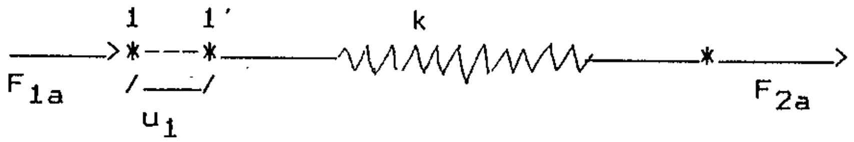
Mula-mula akan dibahas kasus yang paling sederhana, yaitu sebuah pegas linier yang dapat memerima gaya dan mengalami perpindahan dalam satu arah (yaitu arah sumbu pegas) pada kedua ujungnya (titik nodal). Yang akan dibahas adalah hubungan antara perpindahan dan gaya pada kedua ujung pegas tersebut.



Konvensi tanda :

Untuk menurunkan hubbungan perpindahan u_1 dan u_2 pada kedua ujung pegas akibat gaya F_1 dan F_2 yang menimbulkannya, dilakukan tiga langkah berurutan sebagai berikut :

I. Mula-mula dianggap $u_2 = 0$, yang digambarkan dibawah ini :



Pegas berdefleksi u_1 akibat F_{1a}

$$F_{1a} = k * u_1 \quad (1)$$

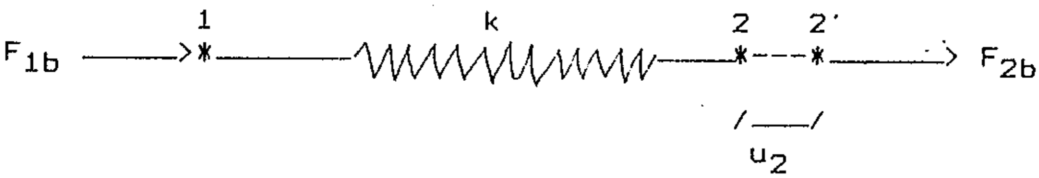
Keseimbangan pegas mensyaratkan jumlah gaya-gaya dalam arah horizontal sama dengan nol :

$$F_{1a} + F_{2a} = 0 \text{ atau } F_{1a} = - F_{2a} \quad (2)$$

dari persamaan (1) dan (2) diperoleh hubungan berikut :

$$F_{2a} = - F_{1a} = - k\ u_1 \quad (3)$$

II Langkah berikutnya adalah dengan menganggap $u_1 = 0$, yang digambarkan sebagai berikut :



Pegas berdefleksi u_2 akibat F_{2b}. Hubungan antara F_{2b} dan u_2 dinyatakan oleh hubungan berikut :

$$F_{2b} = k * u_2 \quad (4)$$

Keseimbangan pegaas mensyaratkan jumlah gaya-gaya horizontal sama dengan nol :

$$F_{1b} + F_{2b} = 0 \quad (5)$$

Dari persamaan (4) dan (5) diperoleh hubungan berikut :

$$F_{1b} = - F_{2b} = - k * u_2 \quad (6)$$

III Superposisi kasus I dan II menghasilkan kasus pegas yang kedua titik nodalnya mengalami perpindahan u_1 dan u_2 akibat gaya F_1 dan gaya F_2.

Gaya titik nodal 1 adalah :

$$F_1 = F_{1a} + F_{1b} \qquad (7)$$

Gaya titik nodal 2 adalah :

$$F_2 = F_{2a} + F_{2b} \qquad (8)$$

Dengan memasukkan persamaan (1), (3), (4), (6) kedalam persamaan (7) dan (8) diperoleh :

$$F_1 = k * u_1 - k * u_2 \qquad (9)$$

$$F_2 = -k * u_1 + k * u_2 \qquad (10)$$

atau jika dituliskan dalam bentuk matriks :

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

atau secara simbolik :

$$[k] = [u] = [F]$$

dimana [k] adalah matriks kekakuan

[u] adalah vektor perpindahan

[F] adalah vektor beban/gaya

Dalam istilah Elemen Hingga (Finite Element), maka matriks kekakuan dari satu elemen dinamakan matriks kekakuan lokal (local stiffness matrix). Yang perlu diperhatikan dari matriks ini adalah :

- matriks kekakuan lokal adalah simetrik.
- Besar matriks kekakuan lokal adalah jumlah titik nodal elemen dikalikan dengan derajad kebebasannya (degree of freedom) atau dimensi kartesiusnya, dalam hal ini dua titik nodal, masing masing dengan satu derajad kebebasan, 2 x 1 = 2 dan besar matriks [k] adalah 2 x 2.