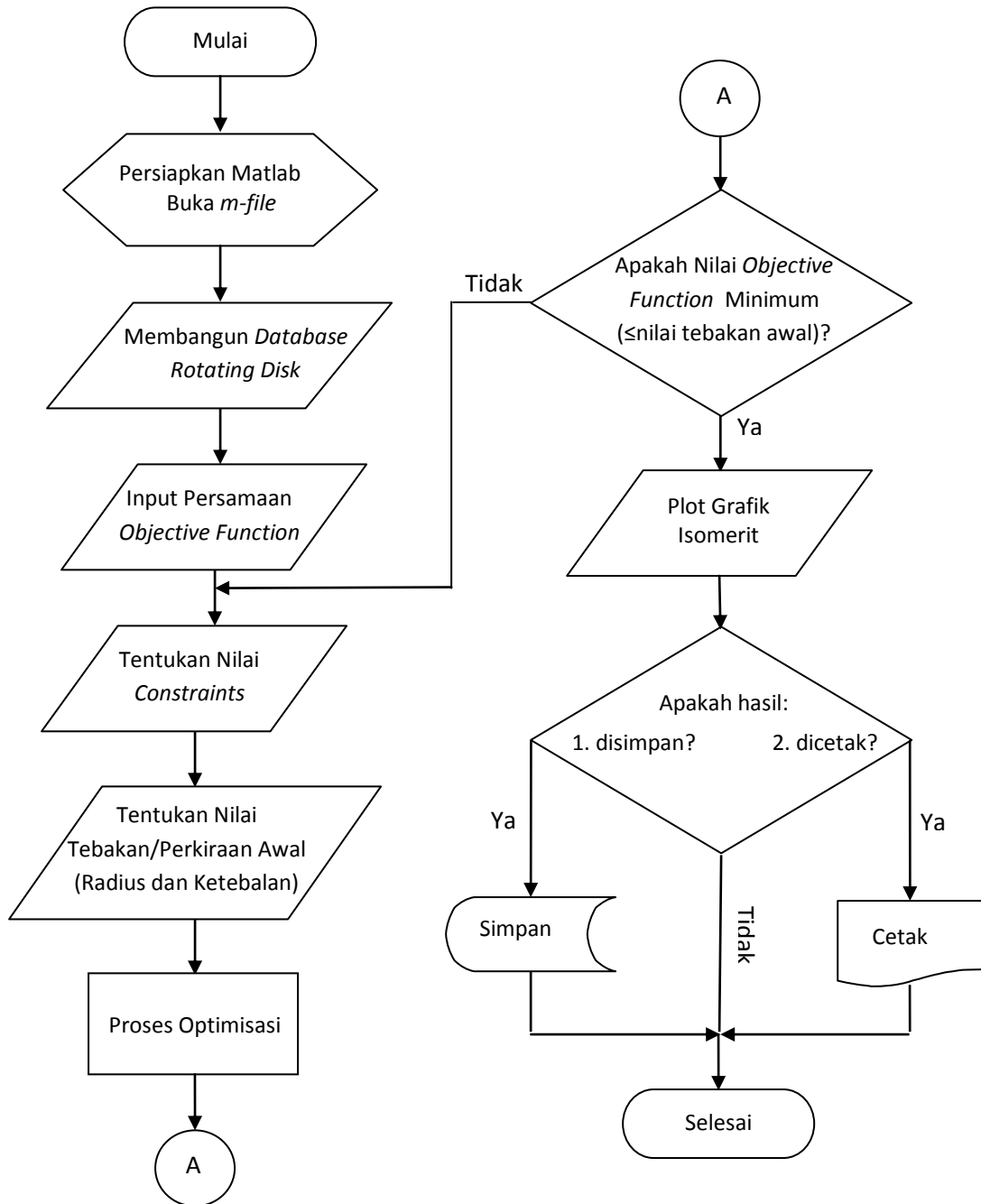


BAB III
PERANGKAT OPTIMISASI DALAM MATLAB

3.1. Langkah Analisis Optimisasi



Gambar 3.1 Diagram alir optimisasi dengan pemrograman Matlab.

Untuk mendapatkan nilai distribusi tegangan tangensial yang akan dioptimumkan, kita harus melakukan proses optimisasi dalam Matlab. Untuk lebih jelasnya dapat dilihat pada diagram alir pada Gambar 3.1.

Langkah-langkah pengerjaan dalam melakukan proses simulasi menggunakan *software*:

MATLAB

Matlab digunakan melakukan proses optimisasi dimana nantinya *output* yang dihasilkan berupa nilai *objective function* yang telah diminimumkan (sesuai batasan-batasan yang ditentukan) serta didapat geometri optimum dari *rotating disk* itu sendiri. Selain itu, dengan menggunakan bantuan Matlab akan dihasilkan grafik isomerit yang dapat memberikan informasi penting dari hasil optimisasi tersebut, yakni berupa karakteristik dari nilai *objective functions* pada *rotating disk* jika diberikan variasi dua variabel radius atau ketebalannya.

Penjelasan untuk diagram alir optimisasi dengan pemrograman Matlab di atas adalah sebagai berikut:

- a. Mulai.
- b. Mempersiapkan program bantu yang digunakan, yakni Matlab, kemudian pilih **File>New>M-file**.
- c. Membangun *database rotating disk*.
Memasukkan semua parameter yang diketahui seperti radius, ketebalan, persamaan tekanan dan persamaan untuk mencari tegangan tangensial kedalam *m-file*.
- d. Input persamaan *objective function*.
Setelah *database* awal dibangun, selanjutnya memasukkan persamaan *objective function* kedalam *m-file* tadi. Simpan atau *run m-file* tersebut.
- e. Tentukan nilai *constraints*.
Langkah selanjutnya adalah buka *m-file* baru, isikan *constraints* yang membatasi *objective functions*.
- f. Tentukan tebakan/perkiraan awal (radius dan ketebalan).

Masih dalam satu *m-file* dengan *constraint* tadi, masukkan nilai tebakan awal agar proses optimisasi dapat dijalankan.

g. Proses optimisasi.

Setelah memasukkan *constraint* dan tebakan awal, masukkan pula fungsi *fmincon* yang digunakan untuk meminimasi *objective function*. Jalankan proses optimisasi dengan menekan *run*.

h. Setelah proses iterasi optimisasi berakhir, analisa hasil tersebut apakah sudah minimum? Jika sudah minimum lanjutkan langkah 9, jika belum minimum ulangi lagi langkah 5 sampai 7 hingga diperoleh hasil yang diinginkan.

i. Setelah hasil optimasi didapat, kemudian plot hasil tersebut kedalam sebuah grafik isomerit masih dengan menggunakan program bantu Matlab.

j. Setelah plot isomerit dibuat, kita dapat menyimpan maupun mencetak hasilnya untuk kemudian dianalisis lebih lanjut.

k. Selesai.

3.2. Pemrograman Matlab

Program yang akan dibuat dalam Matlab ini berupa dua *m-file* yang terdiri dari:

- a. *m-file* satu : berupa *database* awal yang isinya mencakup *preassigned parameters*, persamaan tekanan untuk mendapatkan tekanan yang belum diketahui (P3, P4 dan P5), persamaan pembangun yang digunakan untuk mendukung perhitungan tekanan, persamaan tegangan (radial dan tangensial) dan terakhir persamaan *objective function* yang akan dicari.
- b. *m-file* dua : berisi *constraints* serta program untuk mengeksekusi optimisasi.

Format dua *m-file* ini digunakan hanya untuk mempermudah dalam penelitian ini, program optimisasi *rotating disk* tersebut dapat dibuat dalam format yang lebih praktis lagi, sehingga lebih mudah dalam pengolahan data.

3.2.1. Membangun Database Rotating Disk (*m-file* satu)

Langkah pertama yang harus dilakukan untuk penelitian ini adalah dengan membangun database dalam *m-file* matlab. Database pada matlab digunakan untuk menyimpan data semua parameter-parameter yang digunakan untuk proses optimisasi *rotating disk*.

```

1 function [objective3] = ConstructDiskObjectiveFunction3 (x)
2     L1 = 3,
3     L2 = 3,
4     L3 = 3,
5     L4 = 3,
6     L5 = 3,
7     L6 = 3,
8     R2 = 6.0; % inches
9     R3 = 5,
10    R4 = 3.5,
11    R5 = 2,
12    R6 = 1.0;
13    N = 100; % rpm
14    V = 1; % velocity, inch per sec
15    nu = 0.3; % poisson ratio (v)
16    rho = 0.284; % density
17    P2=0.0; % pressure at the outermost ring surface ,psi
18    P6=1001.0; % internal pressure at the bore, in this case P6=Pm
19    P1=0; % external pressure at the periphery (P1 = P2)
20

```

Gambar 3.2 Simulasi database pada *m-file* Matlab.

3.2.1.1. Input Parameter-Parameter

Hal pertama yang harus diinput dalam *m-file* satu adalah parameter yang telah ditetapkan seperti berikut ini:

X = 4 segmen	L2 = 3,0 inch	N = 9552 rpm
R2 = 6,0 inch	Lm-1 = 3,0 inch	$\alpha_1 = 0,6$
R3 = 5,0 inch	P2 = 0,0	$\alpha_2 = 0,1$
R4 = 3,5 inch	Pm = 1001,0 psi	$\alpha_3 = 0,3$
R5 = 2,0 inch	Lmin = 0,6 inch	$\nu = 0,3$
Rm = 1,0 inch	Lmax = 8,0 inch	$\rho = 0,284 \text{ lb/inch}^3$

dimana,

X : jumlah segmen

N : R.P.M

R_i : Radius ke- i ($i=2,3,4,5$)

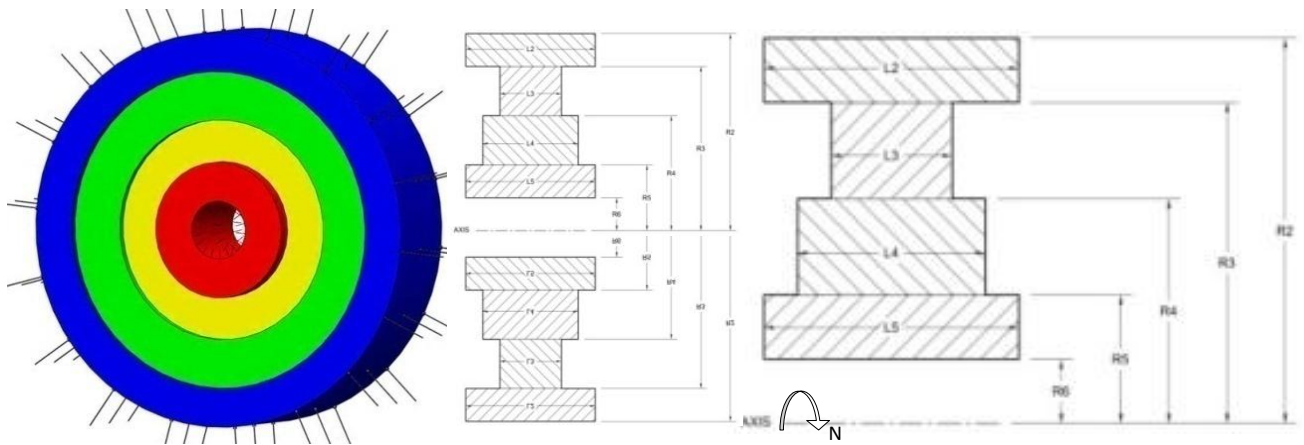
L_i : Ketebalan ke- i ($i=2,3,4,5$)

P_i : Tekanan ke- i ($i=2,3,4,5,6$)

α_i : Faktor pembebanan ke- i (untuk $i=1,2,3$)

ν : *Poisson's Ratio*

ρ : Massa jenis material *disk*



Gambar 3.3 Geometri *rotating disk* yang dimasukkan dalam *database*.

[atas seizin Wisnu Aji P. dan Ref. 1]

Jika dituliskan dalam *m-file* Matlab menjadi seperti berikut:

```
function [objective3] = ConstructDiskObjectiveFunction3 (x)
L1 = 3,
L2 = 3,
L3 = 3,
L4 = 3,
L5 = 3,
L6 = 3,
R2 = 6.0; %inches
R3 = 5,
R4 = 3.5,
R5 = 2,
R6 = 1.0;
V = 1; % velocity, inch per sec
Phi = 3.14;
Alpha_1 = 0.6; %faktor pembebanan(1)
```

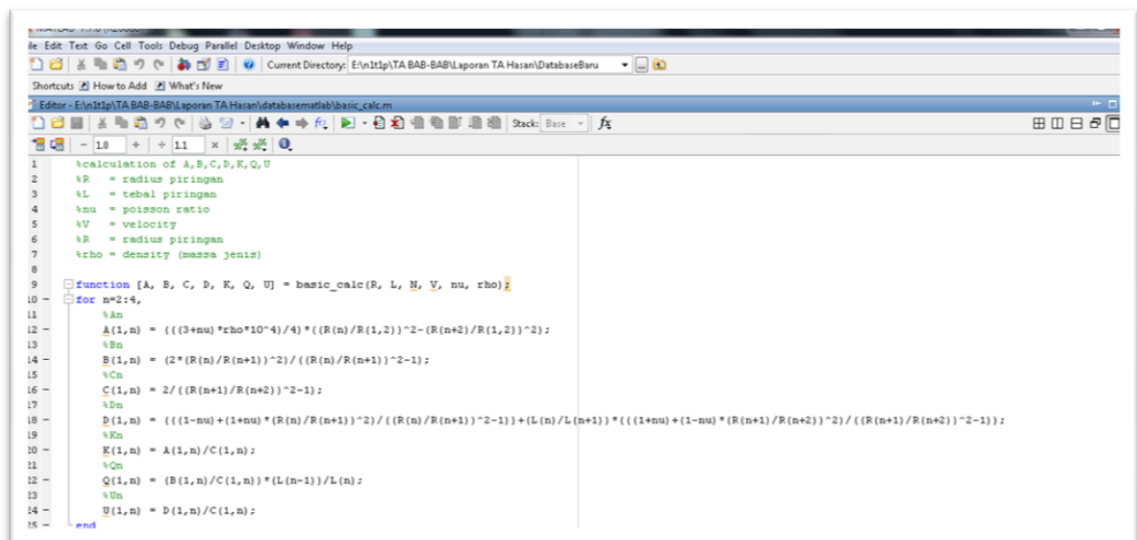
```

Alpha_2 = 0.1; %faktor pembebanan(2)
Alpha_3 = 0.3; %faktor pembebanan(3)
%nilai faktor pembebanan (Alpha) didapat dengan syarat, (0 <=
alpha <= 1) dan (Alpha 1 + Alpha 2 + Alpha 3 = 1)
nu = 0.3; % poisson ratio (v)
rho = 0.284; % density
P2=0.0; % pressure at the outermost ring surface ,psi
P6=1001.0; % internal pressure at the bore, in this case P6=Pm
P1=0; % external pressure at the periphery (P1 = P2)

```

3.2.1.2. Input Database Pendukung

Database pendukung ini berupa persamaan-persamaan yang mendukung pada perhitungan tekanan. Persamaan yang diperlukan adalah dapat dilihat pada persamaan (5.b) sampai persamaan (5.h) pada BAB II.



```

1 %calculation of A,B,C,D,K,Q,U
2 %R = radius piringan
3 %L = tebal piringan
4 %nu = poisson ratio
5 %V = velocity
6 %R = radius piringan
7 %rho = density (massa jenis)
8
9 function [A, B, C, D, K, Q, U] = basic_calc(R, L, N, V, nu, rho)
10 for n=2:4,
11     %An
12     A(1,n) = (((3+nu)*rho*10^4)/4) * ((R(n)/R(1,2))^2 - (R(n+2)/R(1,2))^2);
13     %Bn
14     B(1,n) = (2*(R(n)/R(n+1))^2) / ((R(n)/R(n+1))^2 - 1);
15     %Cn
16     C(1,n) = 2 / ((R(n+1)/R(n+2))^2 - 1);
17     %Dn
18     D(1,n) = (((1-nu)+(1+nu)*(R(n)/R(n+1))^2) / ((R(n)/R(n+1))^2 - 1) + (L(n)/L(n+1)) * (((1+nu)+(1-nu)*(R(n+1)/R(n+2))^2) / ((R(n+1)/R(n+2))^2 - 1));
19     %Kn
20     K(1,n) = A(1,n)/C(1,n);
21     %Qn
22     Q(1,n) = (B(1,n)/C(1,n)) * (L(n-1)/L(n));
23     %Un
24     U(1,n) = D(1,n)/C(1,n);
25 end

```

Gambar 3.4 Input *database* pendukung pada *m-file* Matlab.

Berikut adalah penulisan persamaannya pada matlab:

```

%An
A2 = (((3+nu)*rho*10^4)/4) * ((R2/R2)^2 - (R4/R2)^2);
A3 = (((3+nu)*rho*10^4)/4) * ((R3/R2)^2 - (R5/R2)^2);

```

```

A4 = ((3+nu)*rho*10^4)/4*( (R4/R2)^2-(R6/R2)^2);
A    = [A2 A3 A4];

%Bn
B2 = (2*(R2/R3)^2)/( (R2/R3)^2-1);
B3 = (2*(R3/R4)^2)/( (R3/R4)^2-1);
B4 = (2*(R4/R5)^2)/( (R4/R5)^2-1);
B5 = (2*(R5/R6)^2)/( (R5/R6)^2-1);
B    = [B2 B3 B4 B5];

%Cn
C2 = 2/( (R3/R4)^2-1);
C3 = 2/( (R4/R5)^2-1);
C4 = 2/( (R5/R6)^2-1);
C    = [C2 C3 C4];

%Dn
D2 = ((1-nu)+(1+nu)*(R2/R3)^2)/( (R2/R3)^2-1))+
(L2/L3)*((1+nu)+(1-nu)*(R3/R4)^2)/( (R3/R4)^2-1));
D3 = ((1-nu)+(1+nu)*(R3/R4)^2)/( (R3/R4)^2-1))+
(L3/L4)*((1+nu)+(1-nu)*(R4/R5)^2)/( (R4/R5)^2-1));
D4 = ((1-nu)+(1+nu)*(R4/R5)^2)/( (R4/R5)^2-1))+
(L4/L5)*((1+nu)+(1-nu)*(R5/R6)^2)/( (R5/R6)^2-1));
D    = [D2 D3 D4];

%Kn
K2 = A2/C2;
K3 = A3/C3;
K4 = A4/C4;
K    = [K2 K3 K4];

%Un
U2 = D2/C2;
U3 = D3/C3;

```

```

U4 = D4/C4;
U    = [U2 U3 U4];

%Qn
Q2 = (B2/C2) * (L1) /L2;
Q3 = (B3/C3) * (L2) /L3;
Q4 = (B4/C4) * (L3) /L4;
Q    = [Q2 Q3 Q4];

```

3.2.1.3. Input Persamaan Tekanan (P3, P4, P5)

Setelah parameter-parameter telah dimasukkan, selanjutnya memasukkan persamaan tekanan untuk mendapatkan tekanan yang belum diketahui, yaitu P3, P4 dan P5. Untuk mendapatkan tekanan P3, P4 dan P5 adalah dengan mencari harga tekanan P4, P5 dan P6 tebakan dengan dengan terlebih dahulu menebak harga tekanan P3 secara sembarang, perhitungan tebakan P4, P5 dan P6 dilakukan sebanyak dua kali.

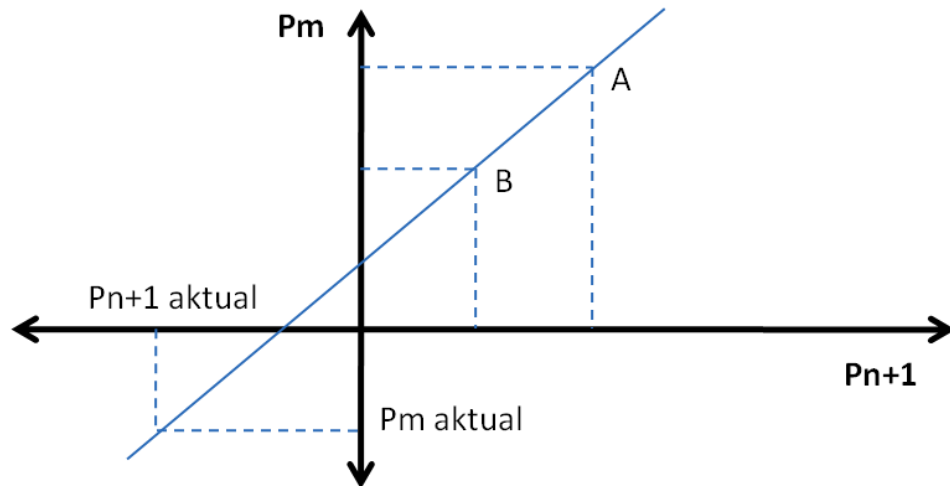
$$P_4 = K_2 V^2 - Q_2 P_2 + U_2 P_{3,tebakan}$$

$$P_5 = K_3 V^2 - Q_3 P_3 + U_3 P_{4,hasil\ tebak}$$

$$P_6 = K_4 V^2 - Q_4 P_4 + U_4 P_{5,hasil\ tebak}$$

Setelah diperoleh nilai P3, P4, P5 dan P6 tebakan, kemudian kita hitung nilai P3 aktual dengan menerapkan ekstrapolasi menggunakan persamaan garis lurus.

$$P_{3, aktual} = \frac{(P_{6, aktual} - P_{6, tebakan1}) * (P_{3, tebakan2} - P_{3, tebakan1})}{(P_{6, tebakan2} - P_{6, tebakan1})} + P_{3, tebakan1}$$



Gambar 3.5 Ekstrapolasi menggunakan persamaan garis lurus.

Keterangan dari gambar di atas:

A = hasil perhitungan tebakan pertama

B = hasil perhitungan tebakan kedua

$P_n = P_o$ = tekanan pada cincin terluar, radius terluar

$P_m = P_i$ = tekanan pada cincin terdalam (berimpit poros)

Setelah didapat P_3 aktual, maka dengan persamaan (5) pada BAB II kita dapat menghitung nilai P_4 dan P_5 aktual.

Jika dituliskan dalam *m-file* Matlab menjadi seperti berikut:

```
P3_g = (200 - rand(1,2) * (200 - 100)); %initial guess for P3, 2 random
numbers between 100-200

for n=1:2,
    P4_g = (K2 * (V^2)) - (Q2 * P2) + U2 * P3_g(1, :); %calculation for
P6
    P5_g = (K3 * (V^2)) - (Q3 * P3_g(1, :)) + (U3 * P4_g); %using initial
guess of P3
```

```

P6_g=(K4*(V^2))-(Q4*P4_g)+(U4*P5_g);           %to obtain linear
equation for interpolation
end

%interpolation, using actual value of P6 to obtain correct value
of P3
P3 = (((P6-P6_g(1,1))*(P3_g(1,2)-P3_g(1,1)))/(P6_g(1,2)-
P6_g(1,1)))+P3_g(1,1);

%calculation of pressure for each interface
P4 = (K2*V^2)-(Q2*P2)+(U2*P3);
P5 = K3*V^2-Q3*P3+U3*P4;
P6 = K4*V^2-Q4*P4+U4*P5;
P = [P2 P3 P4 P5 P6];

```

3.2.1.4. Input Persamaan Tegangan Tangensial dan Radial

Dari nilai parameter dan tekanan yang telah dimasukkan, maka selanjutnya adalah menghitung nilai tegangan tangensial dan radial menggunakan persamaan (6.a) dan persamaan (7) pada BAB II. Jika dituliskan dalam *m-file* Matlab menjadi seperti berikut:

```

%radial stress calculation for each interface
radial_stress3 = -((1+(L2/L3))*P3)/2;
radial_stress4 = -((1+(L3/L4))*P4)/2;
radial_stress5 = -((1+(L4/L5))*P5)/2;
radial_stress6 = -((1+(L5/L6))*P6)/2;
radial_stress = [radial_stress3 radial_stress4 radial_stress5
radial_stress6];

%En
E2 = 1/((R2/R3)^2-1);
E3 = 1/((R3/R4)^2-1);
E4 = 1/((R4/R5)^2-1);
E5 = 1/((R5/R6)^2-1);

```

```

E = [E2 E3 E4 E5];

%Fn
F2 = (((((3+nu)*rho)*10^4)/4)*(R2/R2)^2)+(((1-
nu)*rho*10^4)/4)*(R3/R2)^2;
F3 = (((((3+nu)*rho)*10^4)/4)*(R3/R2)^2)+(((1-
nu)*rho*10^4)/4)*(R4/R2)^2;
F4 = (((((3+nu)*rho)*10^4)/4)*(R4/R2)^2)+(((1-
nu)*rho*10^4)/4)*(R5/R2)^2;
F5 = (((((3+nu)*rho)*10^4)/4)*(R5/R2)^2)+(((1-
nu)*rho*10^4)/4)*(R6/R2)^2;
F = [F2 F3 F4 F5];

%tangential stress (n)
tangential_stress3=- (B2*(L3/L2)*P2)+((E2+(B2/2)-
nu/2*(L2/L3))*P3)+F2*V^2;
tangential_stress4=- (B3*(L4/L3)*P3)+((E3+(B3/2)-
nu/2*(L3/L4))*P4)+F3*V^2;
tangential_stress5=- (B4*(L5/L4)*P4)+((E4+(B4/2)-
nu/2*(L4/L5))*P5)+F4*V^2;
tangential_stress6=- (B5*(L6/L5)*P5)+((E5+(B5/2)-
nu/2*(L5/L6))*P6)+F5*V^2;
tangential_stress = [abs(tangential_stress3)
abs(tangential_stress4) abs(tangential_stress5)
abs(tangential_stress6)];

```

3.2.1.5. Input Routine Objective Function

Berikut adalah *Objective function* yang harus dibuat dalam program matlab:

- Min $F(x) = (\sigma_t)_{\max} - (\sigma_t)_{\min}$
- Min $F(x) = \alpha_1 \{\text{Volume}\} + \alpha_2 \{(\sigma_t)_{\max}\} + \alpha_3 \{(\sigma_t)_{\text{average}}\}$

Properties lain yang harus dibuat untuk mendukung *objective function* yakni tegangan tangensial maksimum ($\sigma_{t \max}$), tegangan tangensial minimum ($\sigma_{t \min}$), tegangan tangensial rata-rata ($\sigma_{t \text{average}}$) dan volume total dari 4 segmen.

Jika dituliskan dalam *m-file* Matlab menjadi seperti berikut:

Untuk objective function 1: $\text{Min } F(x) = (\sigma_t)_{\max} - (\sigma_t)_{\min}$

```
%nilai tegangan tangensial maksimum
max_sigma_t = max(tangential_stress); %...(1)

%nilai tegangan tangensial minimum
min_sigma_t = min(tangential_stress); %...(2)

%objective3, max sigma(t) - min sigma(t)
%substitusi dari persamaan (1) dan (2)
objective3 = (max_sigma_t - min_sigma_t),
end
```

Untuk objective function 2: $\text{Min } F(x) = \alpha_1\{\text{Volume}\} + \alpha_2\{(\sigma_t)_{\max}\} + \alpha_3\{(\sigma_t)_{\text{average}}\}$

```
%nilai tegangan tangensial maksimum
max_sigma_t = max(tangential_stress);
%tegangan tangensial maksimum ini kemudian disubstitusikan ke
persamaan objective4

%average sigma(t)
average_sigma_t = mean(tangential_stress);
%average sigma(t) ini kemudian disubstitusikan ke persamaan
objective4

%volume rotating disk for all segment
%merupakan volume total piringan dari penjumlahan 4 segmen
volume
volume = (Phi* ((R2-R3)^2)*L2)+(Phi* ((R3-R4)^2)*L3)+(Phi* ((R4-
R5)^2)*L4)+(Phi* ((R5-R6)^2)*L5);

%objective function 4 = (alpha(1)*volume) + alpha(2)*max
sigma(t) + alpha(3)*average sigma(t))
```


Constraints untuk radius R:

$$R2 > R3 > R4 > R5 > R6$$

Karena bentuk umum penggunaan pada optimasi di Matlab seperti berikut ini:

$$A \cdot x \leq b$$

Maka bentuknya *constraints* semula perlu kita sesuaikan, perhatikan contoh penulisan *constraints* dalam Matlab berikut:

$$\left. \begin{array}{l} \text{a. } R3 \leq (R2 - 0,1) \\ \text{b. } R4 \leq (R3 - 0,1) \\ \text{c. } (R5 + 0,1) \leq R4 \end{array} \right\} \begin{array}{c} \text{A} \qquad \qquad \text{b} \\ \left[\begin{array}{cc} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{array} \right] \cdot [x] \leq \left[\begin{array}{c} 5,9 \\ -0,1 \\ -2,1 \end{array} \right] \end{array}$$

Contoh penulisan *constraints* untuk variabel R3,R4
(R2=6; R5=2) inch

$$\left. \begin{array}{l} \text{a. } R3 \leq (R2 - 0,1) \\ \text{b. } (R4 + 0,1) \leq R3 \\ \text{c. } R5 \leq (R4 - 0,1) \\ \text{d. } (R6 + 0,1) \leq R5 \end{array} \right\} \begin{array}{c} \left[\begin{array}{cc} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{array} \right] \cdot [x] \leq \left[\begin{array}{c} 5,9 \\ -3,6 \\ 3,4 \\ -1,1 \end{array} \right] \end{array}$$

Contoh penulisan *constraints* untuk variabel R3,R5
(R2=6; R4=3,5; R6=1) inch

$$\left. \begin{array}{l} \text{a. } R4 \leq (R3 - 0,1) \\ \text{b. } R5 \leq (R4 - 0,1) \\ \text{c. } (R6 + 0,1) \leq R5 \end{array} \right\} \begin{array}{c} \left[\begin{array}{cc} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{array} \right] \cdot [x] \leq \left[\begin{array}{c} 4,9 \\ -0,1 \\ -1,1 \end{array} \right] \end{array}$$

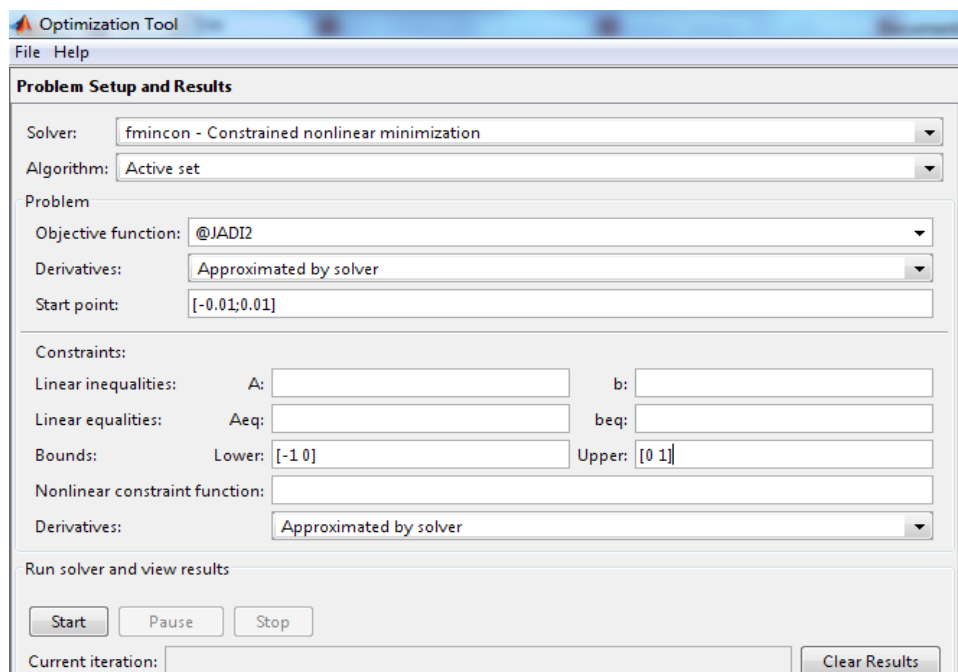
Contoh penulisan *constraints* untuk variabel R4,R5
(R3=5; R6=1) inch

3.2.2.2. Menjalankan Proses Optimisasi

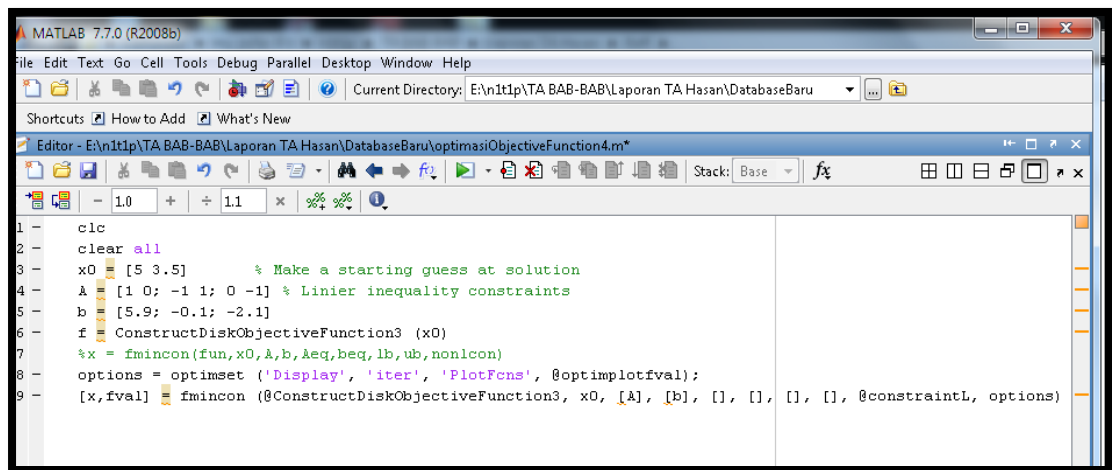
Langkah-langkah dalam proses ini adalah:

- Memilih jenis *solver* yang akan digunakan untuk melakukan proses optimisasi. *Solver* yang digunakan untuk melakukan proses optimasi adalah *solver fmincon*.
- Pengetikan *syntax* fungsi *fmincon* pada *m-file*.
- Memasukkan nilai *start point* atau tebakan awal.
- Menentukan nilai constrain berupa *upper bound* dan *lower bound* yaitu nilai pada koordinat sumbu *x* dan sumbu *y*.
- Menjalankan proses simulasi.

Output yang dihasilkan berupa nilai *objective function* yang paling minimum yang telah diberikan *constraints*.



Gambar 3.7 Toolbox optimasi pada *based-gradient method (fmincon)*.



Gambar 3.8 *M-file* eksekusi optimisasi.

Berikut adalah format program eksekusi optimisasi pada *mfile* Matlab:

```

clc
clear all
x0 = [5 3.5] % Make a starting guess at solution
A = [1 0; -1 1; 0 -1] % % Linier inequality constraints
b = [5.9; -0.1; -2.1]
f = ConstructDiskObjectiveFunction3 (x0)
%x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
options = optimset ('Display', 'iter', 'PlotFcns',
@optimplotfval);
[x,fval] = fmincon (@ConstructDiskObjectiveFunction3, x0, [A],
[b], [], [], [], @constraintL, options)

```