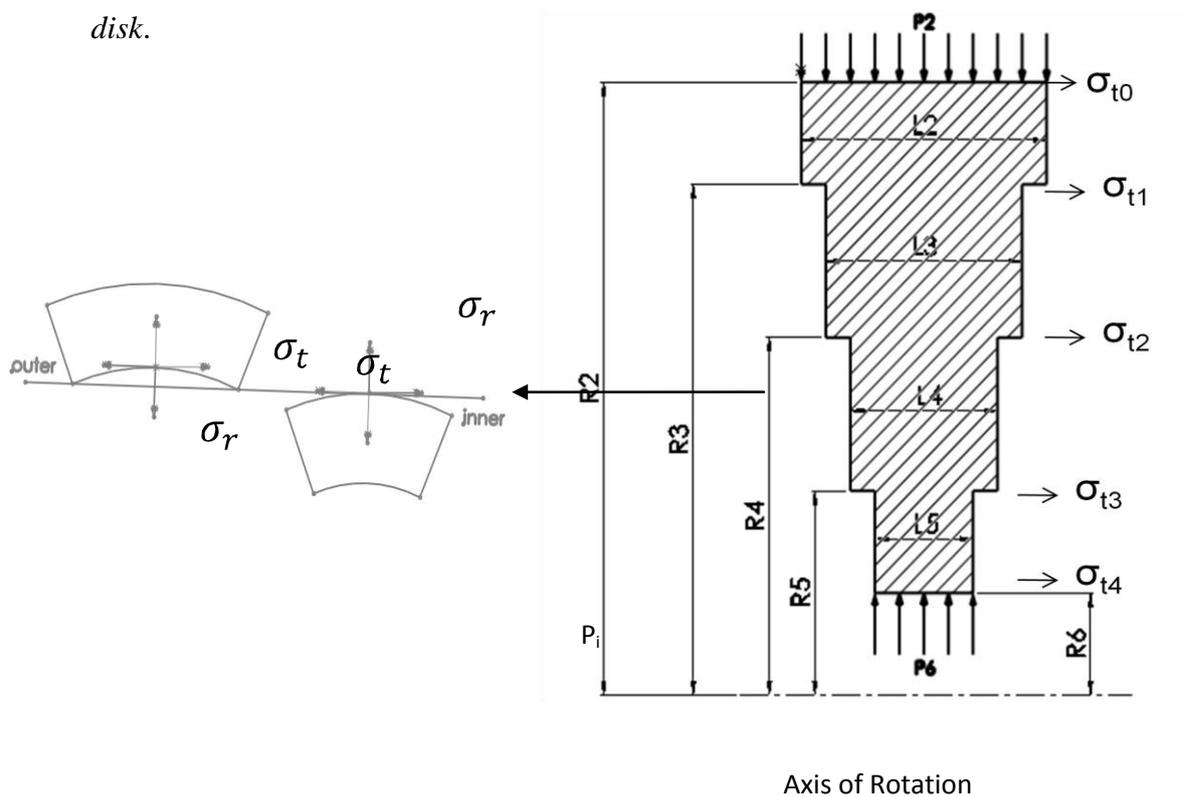


BAB III

METODE OPTIMASI MATLAB

3.1 Langkah Optimasi

Dalam membuat desain optimasi digunakan program MATLAB, suatu bahasa pemrograman perhitungan yang melibatkan operasi matematika elemen, matrik, optimasi, aproksimasi, dan lain-lain. Hal penting pertama yang harus dilakukan sebelum optimasi adalah mengetahui distribusi tegangan pada *rotating disk*.

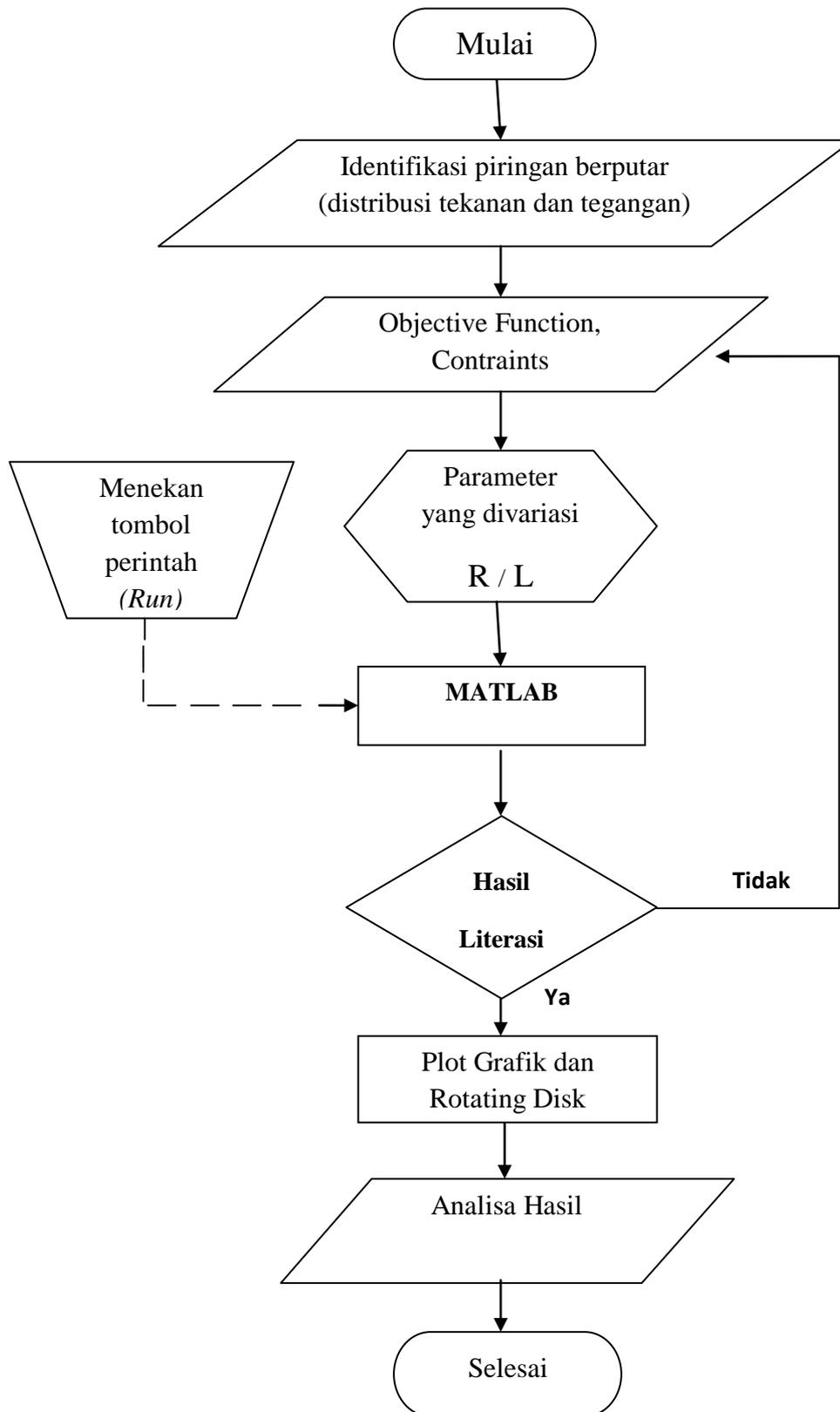


Gambar 3.1 Distribusi tegangan pada *rotating disk*

Langkah yang dilakukan pra proses optimalisasi adalah:

1. Menentukan objek yang akan dioptimalkan.
2. Menentukan *objective function* yang akan digunakan untuk proses optimalisasi.
3. Memasukkan *constraints* yang diinginkan.
4. Menentukan persamaan *objective function*.
5. Mengambil nilai parameter dan variabel yang diperlukan untuk proses optimalisasi.
6. Memasukkan dan mengolah data ke dalam MATLAB dari persamaan *objective function* dan variabel yang diperlukan untuk menghasilkan suatu nilai optimal dari suatu titik.

Pengecekan untuk tiap langkah harus dilakukan karena MATLAB bisa saja *error* karena adanya input data yang tidak sesuai atau kesalahan pada penulisan perintah dalam program.

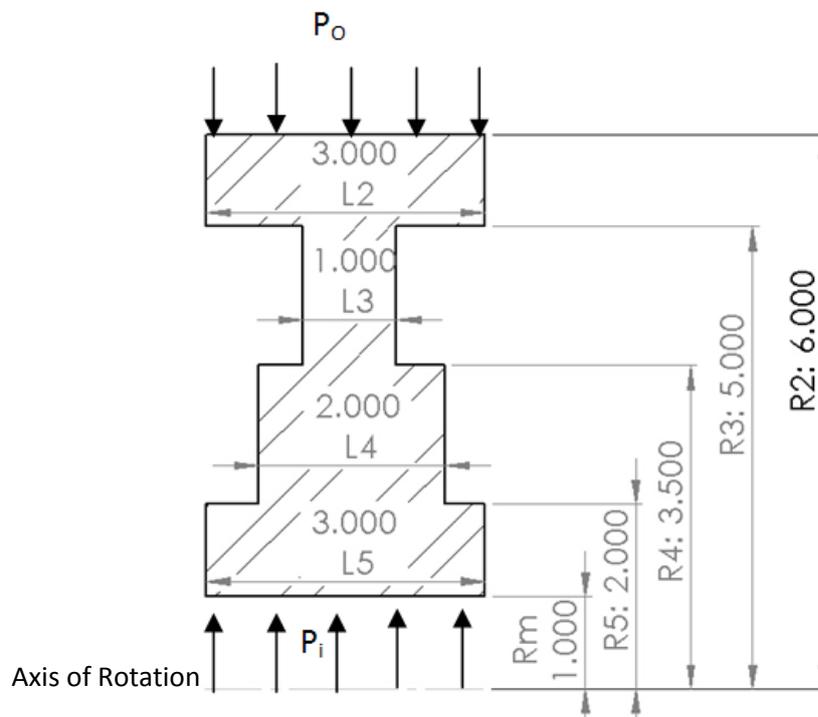


Gambar 3.2 Flow chart optimasi piringan berputar

3.2 Spesifikasi *Rotating Disk*

Sebelum melakukan optimasi *rotating disk* maka hal yang perlu dilakukan adalah mengidentifikasi distribusi tekanan pada setiap segmen. Sedangkan untuk parameter yang akan dicari yaitu berupa jari-jari dan ketebalan piringan pada tiap tingkat.

Adapun spesifikasi dari piringan berputar adalah :



Gambar 3.3 Penampang Rotating Disk

Keterangan :

- P_o = Tekanan pada sisi luar disk
- P_i = Tekanan pada sisi dalam disk
- L_2 = Ketebalan segmen terluar disk
- L_5 = Ketebalan segmen terdalam disk
- R_2 = Radius pada segmen terluar
- R_6 = Radius pada segmen terdalam

Data disk :

$$n = 4 \text{ segmen/tingkat}$$

$$R_2 = 6 \text{ in}$$

$$R_5 = 2 \text{ in}$$

$$R_3 = 5 \text{ in}$$

$$R_m = 1 \text{ in}$$

$$R_4 = 3.5 \text{ in}$$

$$L_2 = 3 \text{ in}$$

$$N = 10000 \text{ rpm}$$

$$L_m = 3 \text{ in}$$

$$P_2 = 0 \text{ psi}$$

$$L_{\min} = 0.6 \text{ in}$$

$$P_m = 1001,0 \text{ psi}$$

$$L_{\max} = 3 \text{ in}$$

a) Sistem Constrain :

$$L_{\min} \leq L_n \leq L_{\max}$$

$$R_{n-1} \geq R_n \geq R_{n+1}$$

b) Objective Function :

1. Meminimalkan tegangan geser maksimum.
2. Meminimalkan tegangan *Von-Misses*.

Pada kasus ini digunakan jenis material dari *High Strength Steel Low alloy* dengan properties sebagai berikut (Reff. 9 hal 893-904) :

- a. *Poison ratio* = $\nu = 0.3$.
- b. *Yield Strength* = $S_y = 36 \text{ kpsi}$.
- c. Berat Jenis = $\gamma = 0.284 \text{ lb/inch}^3$
- d. *Modulus Elastisitas* = $E = 30.10^6 \text{ psi}$.
- e. *Ultimate Strength* = $S_u = 72.5 \text{ kpsi}$.

3.3 Optimasi dengan MATLAB

Optimasi yang dilakukan dalam penelitian ini menggunakan fungsi *fminimax*. Percobaan dengan metode *fminimax* berfungsi untuk meminimalkan nilai terburuk atau kritis (terbesar) dari suatu fungsi multivariabel, yang dimulai dari nilai tebakan awal yang dimasukkan.

Persamaan matematis dalam metode ini adalah sebagai berikut:

$$\min_x \max_i f_i(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

dimana x , b , beq , lb , dan ub adalah vektor, A dan Aeq adalah matriks, dan $c(x)$, $ceq(x)$, dan $f(x)$ adalah fungsi yang mengembalikan vektor. $F(x)$, $c(x)$, dan $ceq(x)$ adalah *constrains*.

Beberapa contoh *Syntax Fminimax* :

$x = \text{fminimax}(\text{fun}, x0)$

$x = \text{fminimax}(\text{fun}, x0, A, b)$

$x = \text{fminimax}(\text{fun}, x, A, b, Aeq, beq)$

$x = \text{fminimax}(\text{fun}, x, A, b, Aeq, beq, lb, ub)$

$x = \text{fminimax}(\text{fun}, x0, A, b, Aeq, beq, lb, ub, \text{nonlcon})$

$x = \text{fminimax}(\text{fun}, x0, A, b, Aeq, beq, lb, ub, \text{nonlcon}, \text{options})$

$x = \text{fminimax}(\text{problem})$

$[x, \text{fval}] = \text{fminimax}(\dots)$

$[x, \text{fval}, \text{maxfval}] = \text{fminimax}(\dots)$

$[x, \text{fval}, \text{maxfval}, \text{exitflag}] = \text{fminimax}(\dots)$

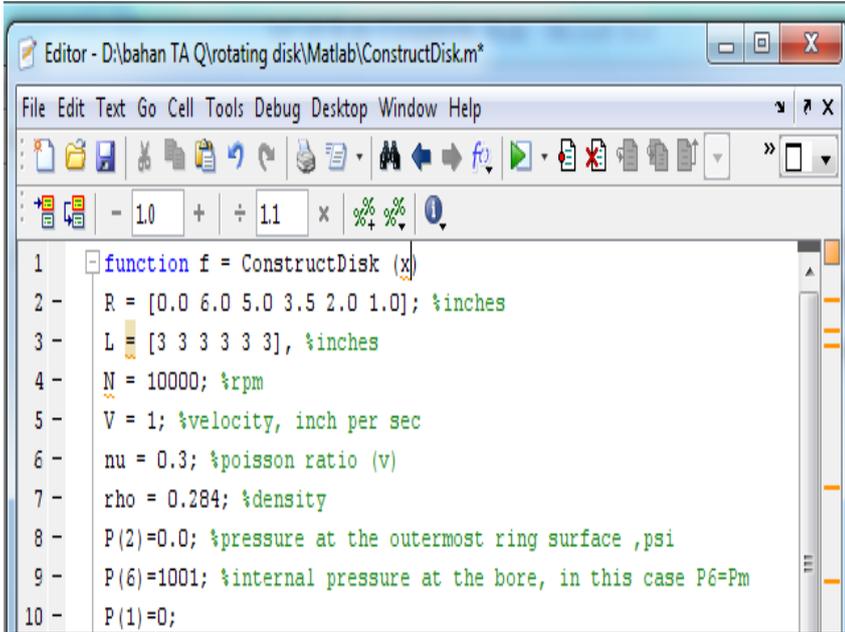
$[x, \text{fval}, \text{maxfval}, \text{exitflag}, \text{output}] = \text{fminimax}(\dots)$

$[x, \text{fval}, \text{maxfval}, \text{exitflag}, \text{output}, \text{lambda}] = \text{fminimax}(\dots)$

Setelah mengetahui persamaan matematis yang akan digunakan untuk proses optimasi nantinya. Langkah selanjutnya adalah membuat database pada *MATLAB* untuk mempermudah melakukan *input* data maupun mengetahui *output* data dari setiap analisa tegangan pada *rotating disk*.

3.3.1 Pembuatan Database

1. Penulisan M-File



```

1 function f = ConstructDisk (x)
2 R = [0.0 6.0 5.0 3.5 2.0 1.0]; %inches
3 L = [3 3 3 3 3], %inches
4 N = 10000; %rpm
5 V = 1; %velocity, inch per sec
6 nu = 0.3; %poisson ratio (v)
7 rho = 0.284; %density
8 P(2)=0.0; %pressure at the outermost ring surface ,psi
9 P(6)=1001; %internal pressure at the bore, in this case P6=Pm
10 P(1)=0;

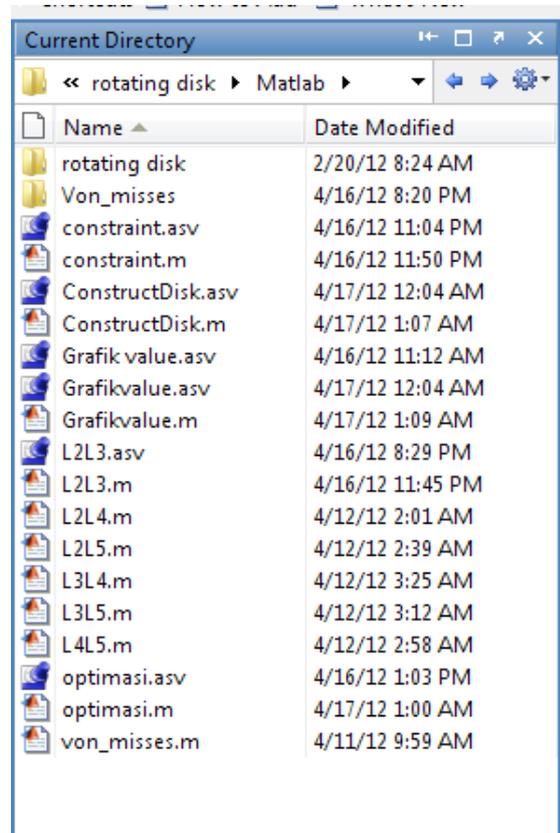
```

Gambar 3.4 Penulisan *database* pada M-File

M-File merupakan tempat dimana kita dapat membuat fungsi dan variabel sendiri. Pada gambar 3.4 bisa kita lihat bahwa nilai parameter yang ada pada spesifikasi (R, L, N, V, dll) *rotating disk*, kita masukkan ke dalam M-File.

Selain itu untuk memudahkan kita saat mengolah data dan perhitungan yang sama tanpa harus membuat fungsi baru. Fungsi M-file ini tidak dimasukkan ke dalam command window, melainkan diletakkan pada

suatu file tersendiri yang dibuat dalam editor teks. Dan hal yang sangat penting adalah untuk selalu menyimpan fungsi yang kita buat sebelum kita menjalankan perintah (*run*).



Gambar 3.5 Tempat Penyimpanan M-File

3.3.2 Perhitungan Tekanan

Langkah berikutnya adalah mencari nilai (P_3, P_4, P_5) dengan cara menebak bebas nilai dari P_3 sebanyak 2x kemudian diperoleh persamaan garis linear antara P_3 dengan P_m . Karena P_m diketahui nilainya maka dengan persamaan garis tersebut kita dapat menginterpolasi untuk mengetahui nilai absolut dari P_3 dan tekanan disemua segmennya.

```

33
34 -   P3_g = (200-rand(1,2) * (200-100)); %initial guess for P3, 2 random
35
36 -   for n=1:2,
37 -       P4_g=K(2) *V^2-Q(2) *P(2) +U(2) *P3_g(1, :);           %calculation f
38 -       P5_g=K(3) *V^2-Q(3) *P3_g(1, :) +U(3) *P4_g;           %using initial g
39 -       P6_g=K(4) *V^2-Q(4) *P4_g+U(4) *P5_g;                 %to obtain linea
40 -   end
41
42   %interpolation, using actual value of P6 to obtain correct value
43 -   P(3) = ((P(6) -P6_g(1,1)) * (P3_g(1,2) -P3_g(1,1))) / (P6_g(1,2) -P6_g
44   %P(3) = interp1(P6_g,P3_g,P(6), 'linear', 'extrap');
45
46
47   %calculation of pressure for each interface
48 -   P(4) = K(1,2) *V^2-Q(1,2) *P(2) +U(1,2) *P(3);
49 -   P(5) = K(1,3) *V^2-Q(1,3) *P(3) +U(1,3) *P(4);
50 -   P(6) = K(1,4) *V^2-Q(1,4) *P(4) +U(1,4) *P(5);
51

```

mbar 3.6 Mencari Nilai Tekanan Disemua Segmen

3.3.3 Membuat *Routine Objective Function*

Dari *database* yang telah dibuat, selanjutnya adalah membuat persamaan *objektive function* yang akan dioptimasi. *Output* yang dihasilkan dalam simulasi ini berupa nilai *objective functions* dari *rotating disk* tersebut. *Objective function* yang harus dibuat dalam program MATLAB adalah sebagai berikut:

1. $\text{Min Max } F_n(x) = (\sigma_t)_{(1, n)} - (\sigma_r)_{(1, n)} / 2$
2. $\text{Min Max } F_n(x) = \text{sqrt}((\sigma_t)_{(1, n)}^2 + (\sigma_r)_{(1, n)}^2 - (\sigma_t)_{(1, n)} \cdot (\sigma_r)_{(1, n)})$

Tegangan yang dibutuhkan untuk membuat *objective function* yaitu tegangan tangensial (σ_t), tegangan radial (σ_r) pada 4 segmen dalam *rotating disk* tersebut.

```

64      %tangential stress (n)
65      tangential_stress(1,n-1)=- (B(1,n) * (L(1,n+1)/L(1,n)) *P(1,n) +
66      end
67      %Radial stress (n)
68      for n=2:5,
69          radial_stress(1,n-1) = -((1+(L(1,n)/L(1,n+1))) *P(1,n+1))/2;
70      end
71      %Objective_function to shear_stress (n)
72      for n=1:4,
73          shear_stress(1,n)= (tangential_stress(1,n)-radial_stress(1,n)
74      end
75      %Objective_function to von-mises_stress (n)
76      for n=1:4,
77          von_mises(1,n)= sqrt((tangential_stress(1,n))^2+ (radial_stress
78      end
79      %fmax(tangential_stress);

```

Gambar 3.7 Objective Function pada *m-file* MATLAB.

3.3.4 Menentukan Nilai *Constrain*

Nilai *constrain* pada optimasi ini merupakan nilai dari batas bawah dan batas atas dari ketebalan dan radius geometri *rotating disk* itu sendiri. Berikut merupakan sistem *constrain* yang akan digunakan dalam mendisain variabel *rotating disk*:

- $L_{min} \leq L_n \leq L_{max}$.
- $R_3 \leq (R_2 - 0,1)$.
- $R_4 \leq (R_3 - 0,1)$.
- $(R_5 + 0,1) \leq R_4$.

Beberapa macam type *constrains* :

- ✓ **Bound Constrains** : batas atas dan bawah pada tiap komponen dimana $x \geq l$ dan $x \leq u$
- ✓ **Linear Inequality Constraints** : Pertidaksamaan linear dari bentuk $A \cdot x \leq b$. A adalah matriks m dengan n , yang merupakan batasan m pada variabel x vektor dengan komponen m . Dan b merupakan komponen vektor m .

Contoh dari *linear inequality constraints* :

$$x_1 + x_3 \leq 4,$$

$$2x_2 - x_3 \geq -2,$$

$$x_1 - x_2 + x_3 - x_4 \geq 9.$$

Dimana $m = 3$ dan $n = 4$

Penulisan untuk matriks A dan vektor b menjadi :

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & -2 & 1 & 0 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 2 \\ -9 \end{bmatrix}$$

Perhatikan bahwa pada pertidaksamaan diatas pada notasi matriks A “Lebih dari” vektor b dikalikan -1 guna notasi menjadi “kurang dari” untuk memudahkan MATLAB menjalankan perintah *linear inequality constraints* diatas.

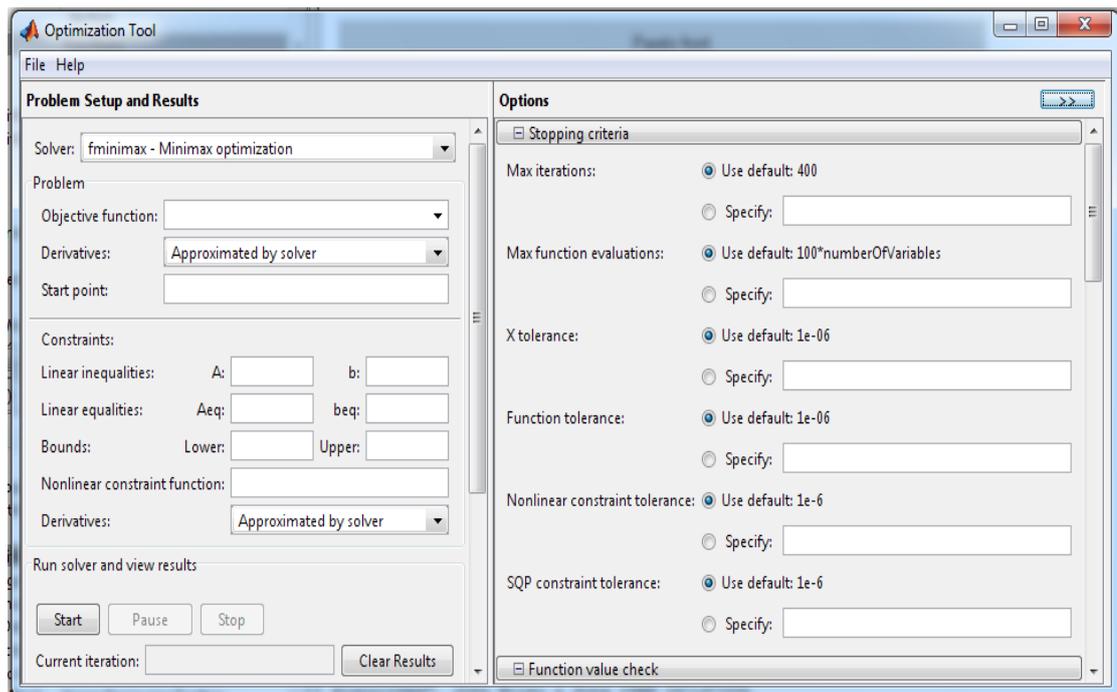
- ✓ **Linear equality Constraints** : Persamaan linier dari bentuk $A_{eq} \cdot x = b_{eq}$. Ini merupakan persamaan m dengan n komponen vektor dari x .
- ✓ **Nonlinear Constraints** : Batas ketidaksamaan nonlinear adalah bentuk dari $c(x) \leq 0$, dimana c adalah pembatas vektor, yang salah satu komponennya untuk tiap pembatas. Demikian pula dengan batas persamaan nonlinear berasal dari bentuk $c_{eq}(x) = 0$. Jika kita

memasukkan gradien untuk c dan ceq maka akan didapat hasil yang lebih akurat.

3.3.5 Menjalankan Proses Optimasi

Dalam menjalankan optimasi ada 2 cara :

A. Menggunakan *optimization tool*



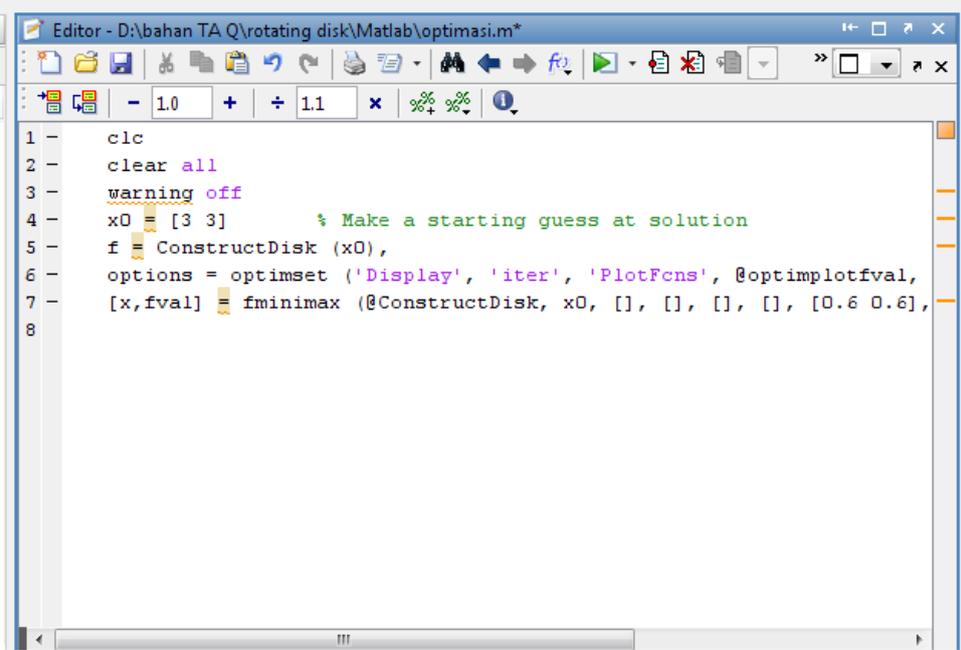
Gambar 3.8 *Optimization tool* MATLAB.

Pada gambar 3.8 diatas, disebelah kanan kolom **option** yaitu untuk mengatur segala kriteria optimasi yang sesuai dengan kehendak kita atau jika tidak, set semua pengaturan pada pilihan “ *Use default* “. Sedangkan pada kolom sebelah kiri adalah pengaturan masalah dan hasil.

- *Solver* : Memilih jenis fungsi yang digunakan untuk optimasi.
- *Objective Function* : Memasukkan fungsi nilai vektor untuk solusi minimax.

- *Derivative* : Menentukan gradien jika tersedia atau solusi pendekatan.
- *Start point* : Menentukan tebakan awal
- *Constraints* : Memasukkan segala macam *constraints* yang kita inginkan.

B. Membuat *Routine Optimasi* pada m-file



```

1 -   clc
2 -   clear all
3 -   warning off
4 -   x0 = [3 3] % Make a starting guess at solution
5 -   f = ConstructDisk (x0),
6 -   options = optimset ('Display', 'iter', 'PlotFcns', @optimplotfval,
7 -   [x,fval] = fminimax (@ConstructDisk, x0, [], [], [], [], [0.6 0.6],
8

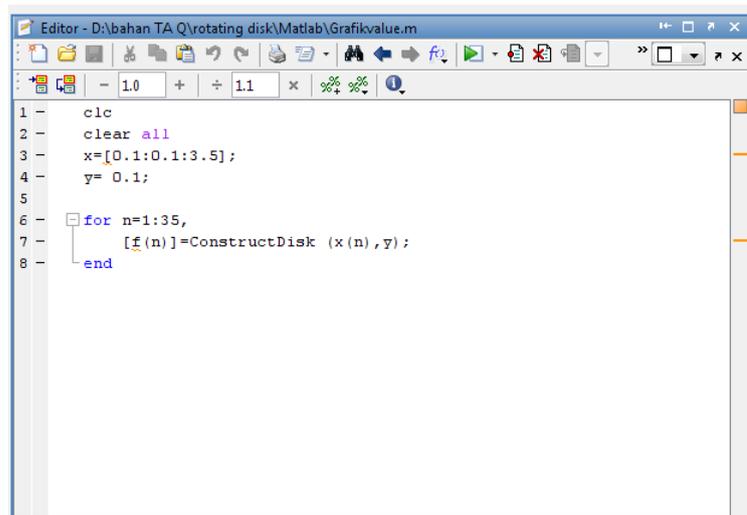
```

Gambar 3.9 *Routine fminimax solver* MATLAB.

Selain menggunakan *optimization tool* ada cara lain yaitu dengan cara manual dengan membuat *Routine* seperti pada gambar 3.9 diatas. Sama halnya pada penggunaan *optimization tool*, diperlukan tebakan awal (X_0) untuk melakukan pendekatan nilai pada proses optimasi nantinya.

Langkah-langkah dalam proses ini adalah:

1. Membuat fungsi dasar dalam proses ini diberi nama ‘ *ConstructDisk* ‘ yang didalamnya berisi parameter *spesifikasi disk*, persamaan analisa tegangan pada *rotating disk* dan *objective function*.
2. Membuat *routine* untuk mencari data fungsi dari variasi R dan L



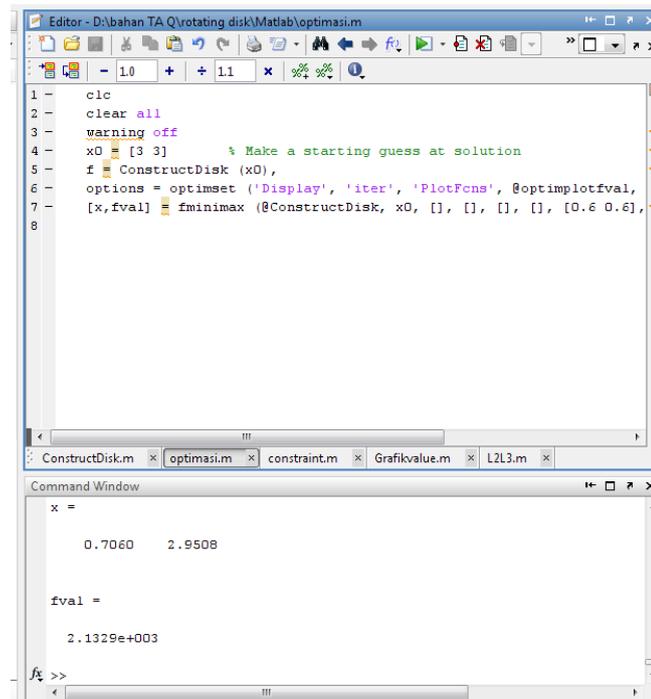
```

1 -   clear
2 -   clear all
3 -   x=[0.1:0.1:3.5];
4 -   y= 0.1;
5
6 -   for n=1:35,
7 -       [f(n)]=ConstructDisk (x(n),y);
8 -   end

```

Gambar 3.10 *Routine fungsi variasi R dan L* MATLAB.

3. Memilih jenis *solver* yang akan digunakan untuk melakukan proses optimasi. *Solver* yang digunakan untuk melakukan proses optimasi adalah *solver fminimax*.
4. Pengetikan syntax fungsi *fminimax* pada m-file.
5. Memasukkan nilai *start point* atau tebakan awal.
6. Menentukan nilai constrain berupa *upper bound* dan *lower bound* yaitu nilai pada koordinat sumbu x dan sumbu y.
7. Menjalankan proses simulasi.
Output yang dihasilkan berupa nilai variasi L atau R dan *objective function* yang paling minimum yang telah diberikan *constrain*.



```

1 - clc
2 - clear all
3 - warning off
4 - x0 = [3 3] % Make a starting guess at solution
5 - f = ConstructDisk(x0),
6 - options = optimset('Display','iter','PlotFcns',@optimplotfval,
7 - [x,fval] = fminimax(@ConstructDisk,x0,[],[],[],[],[0.6 0.6],
8

```

Command Window

```

x =

    0.7060    2.9508

fval =

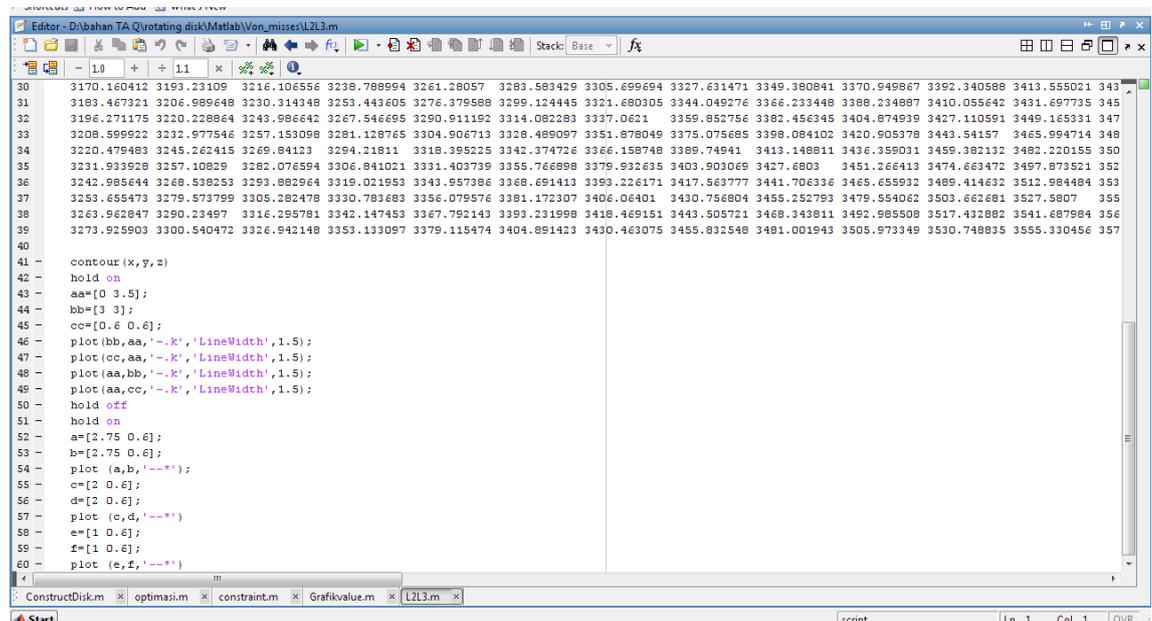
    2.1329e+003

fx >>

```

Gambar 3.11 Proses Optimasi MATLAB.

8. Memasukan nilai X optimum dan data dari point 2 ke *routine* baru untuk memperoleh plot grafik proses jalannya optimasi yang dilakukan.



```

30 3170.160412 3193.23109 3216.106556 3238.788994 3261.28057 3283.582429 3305.699694 3327.631471 3349.380841 3370.949867 3392.340588 3413.555021 3434
31 3183.467321 3206.989648 3230.314348 3253.443605 3276.379588 3299.124445 3321.680305 3344.049276 3366.233448 3388.234887 3410.055642 3431.697735 345
32 3196.271175 3220.228864 3243.886642 3267.546695 3290.611192 3314.082283 3337.0621 3359.852756 3382.456345 3404.874939 3427.110561 3449.165331 347
33 3208.599922 3232.977546 3257.153098 3281.128765 3304.606713 3328.489097 3351.878049 3375.075685 3398.094102 3420.905378 3443.54157 3465.994714 348
34 3220.479483 3245.262415 3269.84123 3294.21811 3318.395225 3342.374726 3366.158748 3389.74841 3413.148811 3436.359031 3459.382132 3482.220155 350
35 3231.933928 3257.10829 3282.076594 3306.841021 3331.403739 3355.766898 3379.932635 3403.803069 3427.6803 3451.266413 3474.663472 3497.873521 352
36 3242.985644 3268.538253 3293.882964 3319.021953 3343.857386 3368.651413 3393.226171 3417.563777 3441.706336 3465.655932 3489.414632 3512.984484 353
37 3253.655473 3279.573759 3305.282478 3330.783683 3356.079576 3381.172307 3406.06401 3430.756804 3455.252793 3479.554062 3503.662681 3527.5807 355
38 3263.962847 3290.23497 3316.295781 3342.147453 3367.792143 3393.231998 3418.469151 3443.505721 3468.343811 3492.985508 3517.432882 3541.687984 356
39 3273.925903 3300.940472 3326.842148 3353.133097 3378.115474 3404.891423 3430.463075 3455.832548 3481.001943 3505.973349 3530.748835 3555.330456 357
40
41 - contour(X,y,z)
42 - hold on
43 - aa=[0 3.5];
44 - bb=[3 3];
45 - cc=[0.6 0.6];
46 - plot(bb,aa,'-k','LineWidth',1.5);
47 - plot(cc,aa,'-k','LineWidth',1.5);
48 - plot(aa,bb,'-k','LineWidth',1.5);
49 - plot(aa,cc,'-k','LineWidth',1.5);
50 - hold off
51 - hold on
52 - a=[2.75 0.6];
53 - b=[2.75 0.6];
54 - plot(a,b,'--*');
55 - c=[2 0.6];
56 - d=[2 0.6];
57 - plot(c,d,'--*');
58 - e=[1 0.6];
59 - f=[1 0.6];
60 - plot(e,f,'--*')

```

Gambar 3.12 Routine Plot Grafik MATLAB.