

## CHAPTER 3 : FILE AND PROGRAMMING


Kita tidak dapat menyimpan hasil pekerjaan pada command window. Dan untuk memanggilnya kembali kita hanya dapat menggunakan history. Sekarang kita akan mempelajari cara menyimpan perintah dan hasil di Matlab dan cara untuk menjalankan hasil yang sudah sebelumnya disimpan tersebut. Selain itu kita akan mempelajari cara mengatur input dan output dari suatu program.

### JENIS FILE

Pada Matlab anda akan mengenal 3 jenis file ( arsip ), yaitu :

M – File , adalah file yang digunakan untuk menyimpan program yang dibuat dan biasanya akan digunakan kembali. M – File yang biasa digunakan adalah blank M-file dan function M-file.

Mat – File, adalah file yang secara otomatis akan dibuat oleh Matlab pada saat menyimpan hasil pekerjaan anda. Coba klik pada salah satu variabel di workspace, edit, lalu save dengan nama yang baru.

Kita kini hanya akan membahas M-File saja. Untuk membuat M-File, klik File lalu New atau klik icon . Pada layar inilah kita dapat mengetik pekerjaan dalam bentuk program/ hanya kumpulan-kumpulan perintah, seperti yang sebelumnya pernah anda lakukan pada Command Window.

#### Contoh 1 :

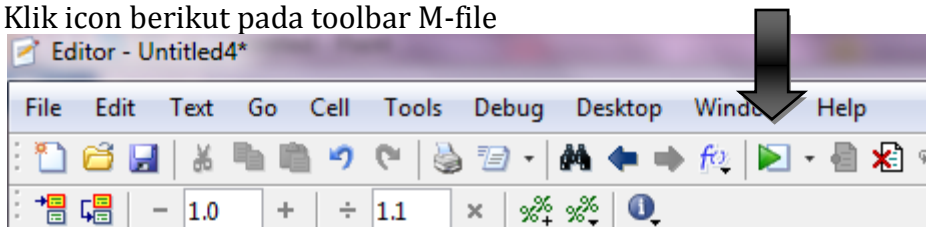
Cobalah mengetik program dibawah ini pada layar Editor, kemudian simpanlah pada folder yang sesuai (misal z:\AMT345) dengan nama file belajar31

```
% Contoh 1
% Program ini untuk menghitung perkalian matriks A dan B

A = [2 3 4;4 6 7];
B = [3 6 2;6 8 3];
C = A*B'
```

Ada banyak cara untuk menjalankan M-file, output dapat dilihat di Command Window

1. Klik icon berikut pada toolbar M-file



2. Tekan F5 pada M-file
3. ketiklah perintah berikut pada Command Window :
  - » cd z:\AMT345; %menunjukkan folder dimana M-file tsb berada
  - » belajar31 %menjalankan program yang sudah dibuat

4. pada toolbar, ubah folder dimana M-file berada lalu panggil nama file dari Command Window



» belajar31 %menjalankan program yang sudah dibuat

Ada kalanya program yang kita buat akan digunakan lagi pada program lainnya, kita dapat menggunakan function M-File. Function M-file sangat berguna jika kita ingin melakukan sejumlah perintah yang sama berulang kali.

Untuk membuat function file kita harus menggunakan sintaks dasar sbb :

```
function [ variabel output ] = nama_fungsi( variabel input )  
% Biasakan member penjelasan mengenai fungsi yang dibuat  
% Lanjutkan dengan fungsi  
  
end
```

Hal yang sangat penting dari function M-file adalah saat menyimpan file tersebut **nama file harus sesuai dengan nama fungsi yang sudah dipilih.**

### Contoh 2 :

Ketiklah perintah-perintah berikut ini pada layar editor dan simpanlah dengan nama drop

```
function [jarak, kecepatan] = drop(g,v0,t)  
% Program untuk menghitung jarak dan kecepatan benda yang  
% jatuh bebas sebagai fungsi dari  
% g (gaya gravitasi), v0 (kecepatan awal) dan t (waktu)  
  
kecepatan = g*t +v0;  
jarak = 0.5*g*t.^2 + v0*t;
```

Untuk menggunakan file tersebut ketiklah perintah-perintah berikut pada Command Window (jangan lupa untuk mengubah folder dimana file di atas disimpan)

```
» g = 32.2;  
» kec_awal = 10;  
» t = 5;  
» [jarak,kecepatan]= drop(g,kec_awal,t)
```

Kemudian cobalah dengan perintah-perintah berikut :

```
» a = 32.2; b= 10; c = 5;  
» [jarak_tempuh,kecepatannya]=drop(a,b,c)
```

**Apakah yang dapat disimpulkan dari kedua percobaan tersebut?**

---

## MENGATUR INPUT DAN OUTPUT

---

Agar lebih interaktif, kita menginginkan pengguna program dapat memasukan input dan kita memunculkan hasilnya. Berikut ini adalah cara-cara untuk meminta input (masukan) dan mengatur output (keluaran/hasil).

---

### INPUT/MASUKAN

---

Masukan ini dapat kita simpan dalam variabel tertentu.

#### Contoh 3 :

Ketiklah perintah berikut pada Command Window

```
» X = input (' Hari ini tanggal berapa ?')
```

Ketiklah perintah berikut pada M-File dan simpan dengan nama latihan32 lalu jalankan.

```
Nama = input('Silahkan masukkan nama anda :','s');  
% ', 's' untuk format string  
Alamat = input ('Silakan masukkan alamat anda :','s');  
disp ('Jadi data anda adalah sbb :')  
['Nama : ', Nama]  
['Alamat : ', Alamat]
```

---

### OUTPUT/KELUARAN

---

Pada dasarnya ada 2 perintah untuk menampilkan hasil, yaitu disp dan fprintf. Tetapi dengan perintah fprintf kita dapat mengatur output dengan lebih baik.

#### Contoh 4 :

Ketiklah perintah berikut pada M-File dan simpan dengan nama latihan33 lalu jalankan.

```
NIM = [001 002 003 004 005];  
Nilai = [69 38 35 78 90];  
tabel(:,1) = NIM';  
tabel(:,2) = Nilai';  
disp(' Nilai peserta kuliah Komputasi Matematika')  
disp(' NIM            Nilai')  
disp(tabel)
```

**Perhatikan NIM, 2 angka nol di depan (00...) menjadi hilang. Bagaimana caranya agar tidak hilang?**

Ketiklah perintah berikut pada M-File dan simpan dengan nama latihan34 lalu jalankan.

```
NIM = [001 002 003 004 005];  
Nilai = [69 38 35 78 90];  
Tabel = [NIM' Nilai'];  
disp(' Nilai peserta kuliah Komputasi Matematika')  
disp ' NIM Nilai')  
fprintf('%3.0f    %3.2f\n',tabel')  
% f untuk format float, \n pindah baris baru
```

**Perhatikan penggunaan perintah %3.0f dan %3.2f. Apa kegunaannya?**

Ketiklah perintah berikut pada M-File dan simpan dengan nama latihan35 lalu jalankan.

```
disp('Belajar membuat program perhitungan sederhana')
a = input('Masukkan angka pertama :');
b = input('Masukkan angka kedua :');
fprintf('\n')
jumlah = a+b;
kurang = a-b;
kali = a*b;
bagi=a/b;
modulo = mod(a,b);
disp('jumlah      kurang      kali      bagi      modulo(a,b)')
fprintf('%2.1f\t    %2.1f\t    %2.1f\t    %2.1f\t    %2.1f\n',
jumlah,kurang,kali,bagi,modulo)
% \t untuk tabular
```

**Modifikasi tampilan output latihan35 supaya menjadi lebih rapih ?**

## PROGRAMMING

### OPERATOR RELASIONAL DAN OPERATOR LOGIKA

Operator relasional pada MATLAB berguna untuk membandingkan array berdasarkan ketidaksamaan matematika. Sedangkan operator logika berguna untuk membandingkan elemen-elemen dari array dengan menggunakan logika matematika. Perbandingan dilakukan untuk tiap-tiap elemen pada masing-masing array, oleh karena itu untuk menggunakan operator relasional/logika, array yang dibandingkan harus memiliki dimensi (ukuran) yang sama.

**MATLAB akan memberikan output 1 jika pernyataan bernilai benar dan 0 jika pernyataan bernilai salah.**

Operator Relasional	Arti
<	Lebih kecil dari
<=	Lebih kecil dari atau sama dengan
>	Lebih besar dari
>=	Lebih besar dari atau sama dengan
== (tidak pakai spasi)	Sama dengan
~=	Tidak sama dengan

#### Contoh 5 :

```
» x =[6 3 9];y=[14 2 9];
» z = (x<y)
» z= (x<=y)
» z=(x~=y)
» z = (x == y)
```

#### Contoh 6 :

```
» z = x(x<y) % nilai x yang memenuhi x<y
» z = y(x==y)
» z = x(x>=y)
```

Operator	Nama Operator	Hasil untuk A = [1 0]; B = [1 1 0 0]; C = [1 0 1 0]
~	NOT	~A = [0 1]
&	AND	B&C = [1 0 0 0]
atau	OR	B C = [1 1 1 0]
xor	Eksklusive or	xor(B,C) = [0 1 1 0]

#### Contoh 7 :

```

» x = [2,-4,0,7]; y=[8,-4,7,0];
» a = x >= y | x == y
» b = x >= y & x == y

```

#### Contoh 8 :

Vektor berikut merupakan catatan hasil penjualan selama 10 hari.

[19000 18000 22000 21000 25000 19000 17000 21000 27000 29000]

Tentukanlah :

- Hari ke berapa saja penjualan di atas 20000
- Jumlah total penjualan dari hasil a.

```

» x = [19000 18000 22000 21000 25000 19000 17000 21000 27000
29000];
» a = (x>20000)
a = 0      0      1      1      1      0      0      1      1      1
» b = x*a'
b = 145000

```

### PERNYATAAN KONDISIONAL IF

Pada MATLAB terdapat beberapa struktur pernyataan kondisional if , yaitu :

#### 1. Pernyataan if

Contoh : Jika IPS (IP semester) minimal 2.5, maka boleh mengambil 21 sks pada semester yang akan datang. Kalimat ini dapat dituliskan pada M-File sbb :

```

if ips >= 2.5
    sks = 21
end

```

Simpan dengan nama if1 lalu coba berikan input pada Command window

```

» ips = 2.5
» if1
» ips = 1.5
» if1

```

#### 2. Pernyataan else

Contoh : Jika IPS minimal 2.5, maka boleh mengambil 21 sks pada semester yang akan datang. Jika tidak, hanya boleh mengambil 18 sks. Kalimat ini dapat dituliskan pada M-File sbb :

```

if ips >= 2.5
    sks = 21
else
    sks = 18
end

```

```

else
    sks = 18
end

```

Simpan dengan nama if2 lalu coba berikan beberapa input pada Command window.

### 3. Pernyataan elseif

Contoh : Jika IPS minimal 3, maka boleh mengambil 24 sks pada semester yang akan datang. Jika IPS antara 2.50 sampai 2.99, maka boleh mengambil 21 sks. Jika IPS di bawah 2.5, hanya boleh mengambil 18 sks. Keterangan tambahan bahwa IPS tidak mungkin lebih besar dari 4. Kalimat ini dapat dituliskan pada M-File sbb :

```

if ips >= 4
    display('Input IPS salah')
elseif ips >= 3
    sks = 24
elseif ips >= 2.5
    sks = 21
else
    sks = 18
end

```

Simpan dengan nama if3 lalu coba berikan beberapa input pada Command window.

**Coba modifikasi ketiga contoh di atas, menjadi lebih interaktif dengan pengguna. Simpan dengan nama latihan36. Contoh modifikasi :**

**Input :** Berapa mata kuliah yang kamu ambil semester ini ?  
 Tuliskan nilai yang diperoleh (A/B/C/D/E diselingi dengan spasi)  
 Tuliskan sks dari MK di atas (perhatikan urutannya)

**Output :** Indeks Prestasi Semester kamu adalah ...  
 Semester yang akan datang kamu boleh mengambil ... sks

## LOOPING

Struktur Loop digunakan pada program untuk melakukan pernyataan yang sama sebanyak beberapa kali. Ada 2 macam loop pada MATLAB, yaitu for dan while. Struktur for digunakan jika kita telah mengetahui berapa kali pernyataan akan dilakukan, sedangkan while digunakan jika kita tidak tahu sebelumnya berapa kali pernyataan akan dilakukan, tetapi proses akan berhenti jika suatu kondisi terpenuhi.

### Struktur for

```

for x = m:s:n
    % pernyataan
end

```

dimana x = variabel loop, m = nilai awal variabel loop, s = increment (jika tidak didefinisikan, maka MATLAB akan menggunakan increment 1) dan n = nilai akhir variabel loop.

### Contoh 9 :

Ketiklah perintah berikut pada M-File lalu jalankan.

```

% Program untuk menghitung kuadrat dari 5, 15, 25 dan 35
for k = 5:10:35

```

```

        x = k^2
end

```

#### Contoh 10 :

Program ini untuk membangun sebuah matriks berukuran 6×4 dimana elemennya adalah penjumlahan dari letak baris dan kolom. Simpan dengan nama latihan37

```

for m = 1:6 % Loop luar;

    for n = 1:4 % Loop dalam;
        A(m,n) = m+n; % elemen A tidak ditampilkan
    end % End loop dalam

end % End loop luar
A % tampilkan matriks A

```

#### Struktur While

```

while x>0 % ekspresi logika
    %pernyataan
end

```

Untuk menggunakan loop while hal yang harus diperhatikan adalah variabel yang digunakan dalam loop harus memiliki nilai awal dan pernyataan pengubah nilai tersebut.

#### Contoh 11 :

```

x = 1; % nilai awal variabel
while x~=5
    disp(x)
    x =x+1; % pengubah nilai variabel x baru adalah x lama + 1
end

```

Dan hati-hati agar tidak terjadi infinite loop (loop tidak pernah berhenti).

#### Contoh 12 :

```

x = 8;
while x~=0
    x = x-3
end

```

Perintah *break* dapat digunakan jika kita ingin “keluar” dari loop sebelum nilai variabel yang seharusnya menghentikan loop tercapai.

#### Contoh 13 :

```

for k = 1:10
    x = 50 - k^2;
    if x < 0
        break
    end
    y = sqrt(x)
end

```

Bentuk dasar yang digunakan adalah sbb

```
switch ekspresi input (skalar atau string)
    case nilai1
        pernyataan1
    case nilai2
        pernyataan2
    case nilai...
        pernyataan ...
    otherwise
        pernyataan lainnya
end
```

Input akan dibandingkan dengan masing-masing nilai1, nilai2, nilai3, dst pada masing-masing case, jika nilainya sama, maka pernyataan di bawah case yang bersesuaian yang akan dikerjakan. Jika tidak ada nilai yang sama dengan input, maka pernyataan yang akan dikerjakan adalah pernyataan otherwise. Dan jika tidak ada pernyataan otherwise maka program akan langsung berlanjut ke pernyataan setelah end.

#### Contoh 14 :

Program berikut menyatakan arah sesuai besar sudut yang dibentuk dengan arah utara.

```
angle = input('Please insert the angle ')
switch angle
    case 45
        disp('Northeast')
    case 135
        disp('Southeast')
    case 225
        disp('Southwest')
    case 315
        disp('Northwest')
    otherwise
        disp('Direction Unknown')
end
```

Bandingkan dengan program berikut

```
angle = input('Please insert the angle ')
switch angle
    case {0,360}
        disp('North')
    case {-180,180}
        disp('South')
    case {-270,90}
        disp('East')
    case {-90,270}
        disp('West')
    otherwise
        disp('Direction Unknown')
end
```



1. Buatlah program untuk menghitung luas permukaan dan volume bola dimana jari-jari bola ( $r$ ) diketahui. Tampilkan hasilnya dalam bentuk tabel dimana kolom pertama menyatakan jari-jari bola, kolom kedua luas permukaan bola ( $L_p = 4\pi r^2$ ) dan kolom ketiga menyatakan volumenya ( $V = \frac{4}{3}\pi r^3$ ). Lalu pada Command window beri input jari-jari bola  $r = 1, 1.1, 1.2, \dots, 5$ .
2. Buatlah program yang sama dengan no 1 tetapi dengan menampilkan kolom radius dengan 2 angka di belakang koma dan kolom luas dan volume 4 angka di belakang koma.

3. Tentukanlah kapan harga barang A lebih mahal daripada harga barang B, jika diketahui data penjualan selama 10 hari sbb :

Hari ke-	1	2	3	4	5	6	7	8	9	10
Harga A (dalam ribuan)	19	8	22	21	25	19	17	21	27	29
Harga B (dalam ribuan)	22	17	20	19	24	18	16	25	28	27

4. Sebuah proyektil ditembakkan ke udara dengan kecepatan awal  $v_0 = 40$  meter/detik dan sudut dengan tanah sebesar  $\theta = 30^\circ$ . Ketinggian proyektil itu dari atas tanah dinyatakan dalam persamaan  $h(t) = v_0 t \sin(\theta) - 0,5gt^2$  sedangkan kecepatannya adalah  $v(t) = \sqrt{v_0^2 - 2v_0 g t \sin(\theta) + g^2 t^2}$ , dimana gaya gravitasi  $g = 9,81$  meter/detik<sup>2</sup>. Proyektil tersebut akan menyentuh tanah ketika  $h(t) = 0$ , waktu yang dibutuhkan untuk itu adalah  $t_s = 2\left(\frac{v_0}{g}\right)\sin(\theta)$ .

Tentukan waktu ( $t$  sampai 3 angka di belakang koma) saat :

- a) Ketinggian proyektil tidak lebih rendah dari 15 meter.
  - b) Ketinggian proyektil tidak lebih rendah dari 15 meter **dan** kecepatannya tidak lebih besar dari 36 meter/detik.
  - c) Ketinggian proyektil lebih rendah dari 5 meter **atau** kecepatannya lebih besar dari 35 meter/detik.
5. Buatlah program untuk menghitung  $y = \sqrt{x}$ , jika  $x \geq 0$  untuk setiap elemen pada array  $x$ , jika terdapat nilai negatif, maka program akan memunculkan komentar bahwa ada elemen  $x$  yang bernilai negatif.
  6. Buatlah function M-File untuk menghitung nilai  $f(x)$  dengan input suatu bilangan  $x$  dimana 
$$f(x) = \begin{cases} e^x - 1 & , x < 0 \\ \sqrt{x} & , 0 \leq x < 10 \\ \ln(x) & , x \geq 10 \end{cases}$$
  7. Buatlah program untuk membuat perhitungan hasil kali 2 buah matriks sembarang dengan menggunakan looping for.  
 Input :    Ukuran matriks A =  
           Ukuran matriks B =  
           Input elemen matriks A =  
           Input elemen matriks B =  
 Output : A×B (sebelumnya periksa terlebih dahulu apakah A dan B dapat dikalikan)

8. Buatlah program sederhana untuk mesin cashier kantin di SB. Contoh tampilan yang diinginkan (imput diberi warna merah)

Menu hari ini

```
1. Nasi kuning Rp 4.000
2. Nasi uduk   Rp 4.000
3. Tumis tempe Rp 1.000
4. Telor       Rp 2.000
5. Bihun       Rp 1.000
6. Bakwan      Rp 1.500
7. Sambel      Rp 500
```

Anda mau menu nomor berapa ? [8]

Tidak ada menu tersebut hari ini

Anda mau menu nomor berapa ? [1 3 4 6]

Total Rp 8500

Bayar ? 10000

Kembali Rp 1500

9. Buatlah program untuk mencari jenis dan akar dari suatu persamaan kuadrat. Contoh tampilan (imput diberi warna merah)

Masukkan koefisien persamaan kuadrat anda : [1 -2 3]

Persamaan tersebut tidak memiliki akar real

Masukkan koefisien persamaan kuadrat anda : [1 -3 2]

Persamaan tersebut memiliki 2 akar real beda

akar1 =

-1

akar2 =

-2

10. Buatlah program untuk menampilkan n baris segitiga pascal. Contoh untuk n = 10

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```