



# CHROME.M

Character Recognition  
with Matlab

## Dasar Matlab

Artificial Intelligence Laboratory



## Daftar Isi

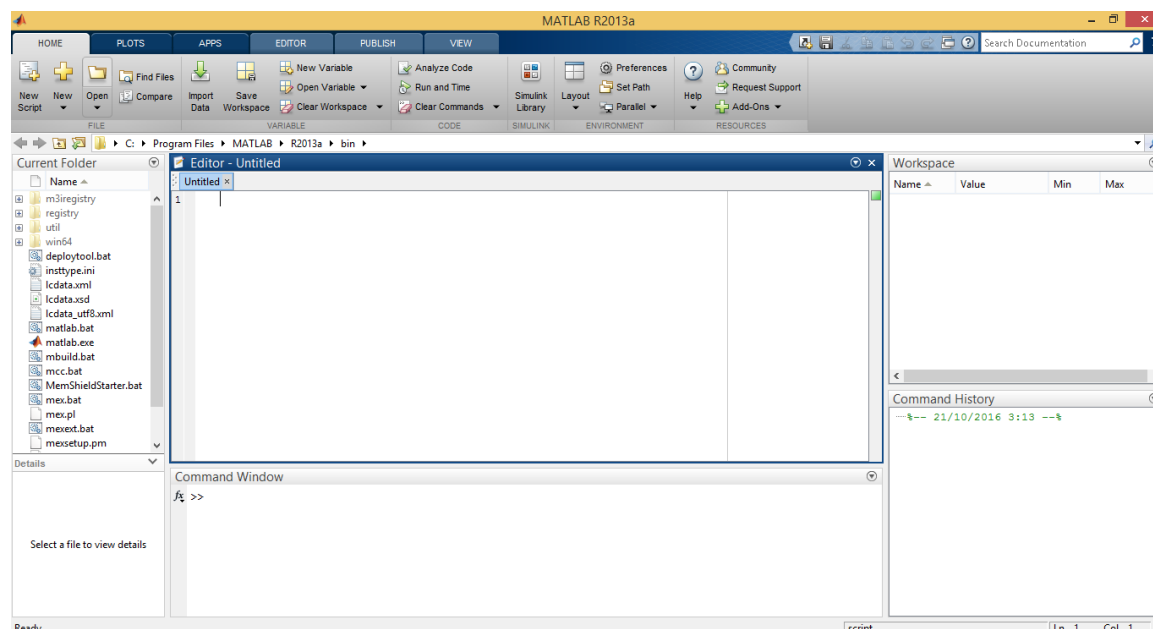
1. Pengenalan MATLAB .....	1
2. Variabel pada MATLAB .....	2
3. Operator .....	5
4. <i>Control Flow</i> .....	5
5. Pembangunan Matriks.....	8
6. Seleksi Data .....	10
7. Data Input Output File .....	12
8. Fungsi dalam Matlab-File .....	15
9. Plots.....	15
10. Input Output Image .....	18

### 1. Pengenalan MATLAB

MATLAB (*Matrix Laboratory*) merupakan bahasa pemrograman sekaligus sebuah *interface* yang dapat digunakan untuk merancang sebuah sistem ataupun produk. MATLAB lebih memfokuskan pemrosesan data yang bersifat matriks sehingga untuk pemrosesan data lain seperti data yang bersifat string tidak disarankan menggunakan MATLAB. MATLAB biasanya digunakan untuk mengimplementasikan beberapa bidang keilmuan seperti *machine learning*, *signal processing*, *image processing*, *computer vision*, *communications*, *computational finance*, *control design*, *robotics*, dan lain-lain.

MATLAB termasuk kedalam bahasa pemrograman tingkat tinggi berbasis matriks dan dapat digunakan untuk teknik komputasi numerik, sehingga MATLAB sering digunakan untuk perhitungan matematika, komputasi, modeling, simulasi, visualisasi data, eksplorasi data, pengembangan aplikasi, dan lain-lain. MATLAB juga memberikan kemudahan dengan menyediakan *tools* dan fungsi-fungsi yang dapat digunakan oleh *user*. Sebagai contoh, *plots* yang memiliki berbagai fitur untuk visualisasi data. Untuk penggunaan fungsi, dapat dilihat pada dokumentasi yang disediakan MATLAB. Dokumentasi yang disediakan cukup mudah untuk dipelajari.

Untuk orang yang baru mengenal MATLAB, ada beberapa lingkungan *interface* yang perlu diketahui seperti *Editor*, *Command Window*, *Workspace*. *Editor* digunakan sebagai sarana untuk menuliskan *code*. Layaknya *Command Prompt* pada Windows, *Command Window* digunakan untuk menuliskan perintah-perintah yang dikenal MATLAB ataupun menampilkan hasil *run program*. *Workspace* digunakan untuk menyimpan nama-nama *variable* yang telah ditulis pada *Editor* beserta *value*-nya. *Variable* yang tersimpan juga dapat dilihat secara keseluruhan.



*Note* : Perlu diketahui bahwa MATLAB tidak bersifat *open source*.

### 2. Variabel pada MATLAB

Dalam bahasa pemrograman Matlab, pendeklarasian variable tidak perlu dilakukan, karena semua variable dianggap sebagai array atau matriks. Variable dapat di set di dalam Editor untuk code yang sudah pasti, ataupun untuk mencoba perhitungan maupun code dapat dicoba di dalam Command Window, dan hasil akan disimpan di dalam Workspace. Command Windows juga berfungsi untuk menampilkan hasil dari perhitungan atau code yang digunakan.

Contoh variable numerik tunggal



```
Command Window
>> a = 45

a =

    45

fx >> |

Command Window
>> b = 5.67

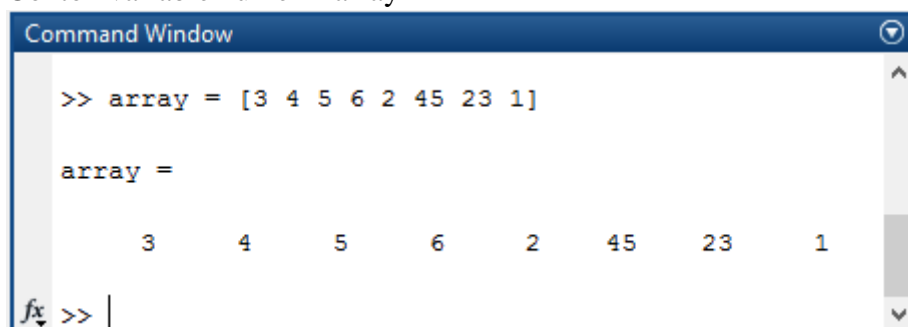
b =

    5.6700

fx >> |
```

Secara otomatis variable akan diset sesuai dengan isi dari variable.

Contoh variable numerik array



```
Command Window
>> array = [3 4 5 6 2 45 23 1]

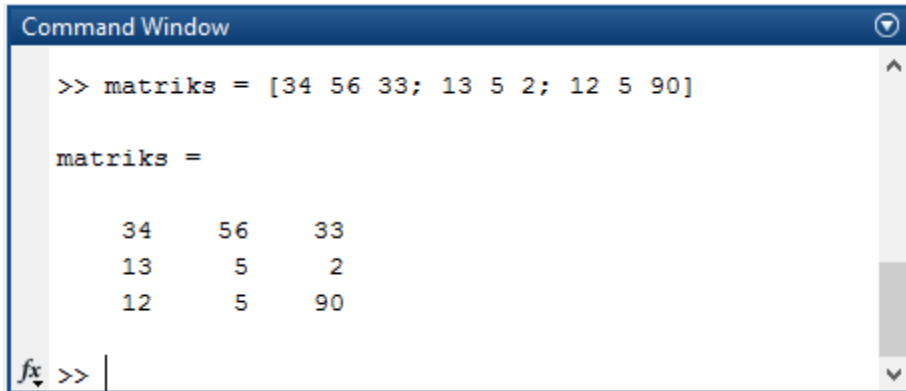
array =

     3     4     5     6     2    45    23     1

fx >> |
```

Untuk mendefinisikan variable dalam bentuk array cukup dengan memasukan data variable di dalam tanda “[ ]” dengan spasi sebagai pembeda datanya.

Contoh variable matriks



```
Command Window

>> matriks = [34 56 33; 13 5 2; 12 5 90]

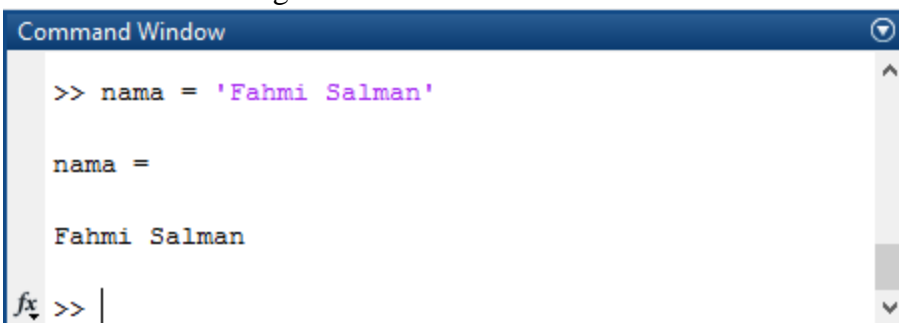
matriks =

    34    56    33
    13     5     2
    12     5    90

fx >> |
```

Untuk mendefinisikan variable dalam bentuk matriks tidak jauh berbeda dengan array, masukan data di dalam “[ ]”, dengan “;” untuk menambahkan baris baru.

Contoh variable string



```
Command Window

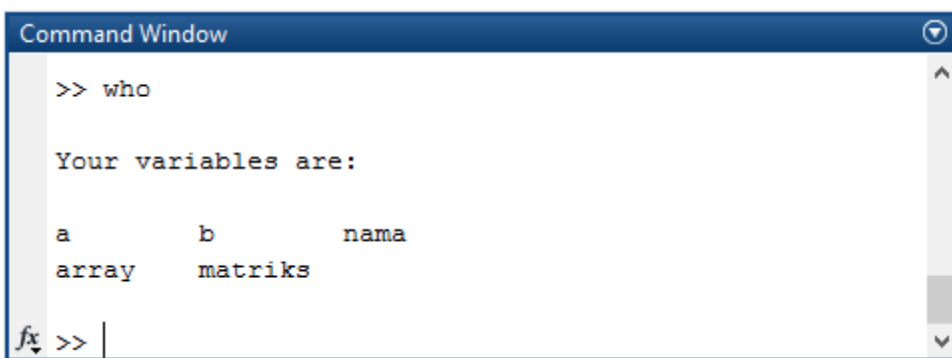
>> nama = 'Fahmi Salman'

nama =

Fahmi Salman

fx >> |
```

Untuk mendefinisikan variable string cukup dengan memasukan data string ke dalam variable, dan variable tersebut otomatis menjadi variable string.



```
Command Window

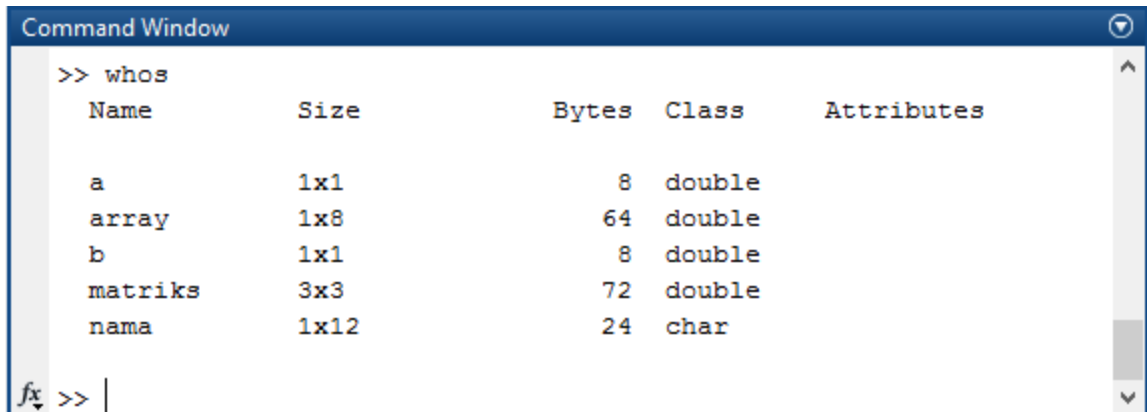
>> who

Your variables are:

a          b          nama
array      matriks

fx >> |
```

Untuk menampilkan variable yang digunakan dan tersimpan dalam workspace, cukup dengan mengetikan “who” dalam Comand Windows. Maka variable yang disimpan di workspace akan ditampilkan di command windows.



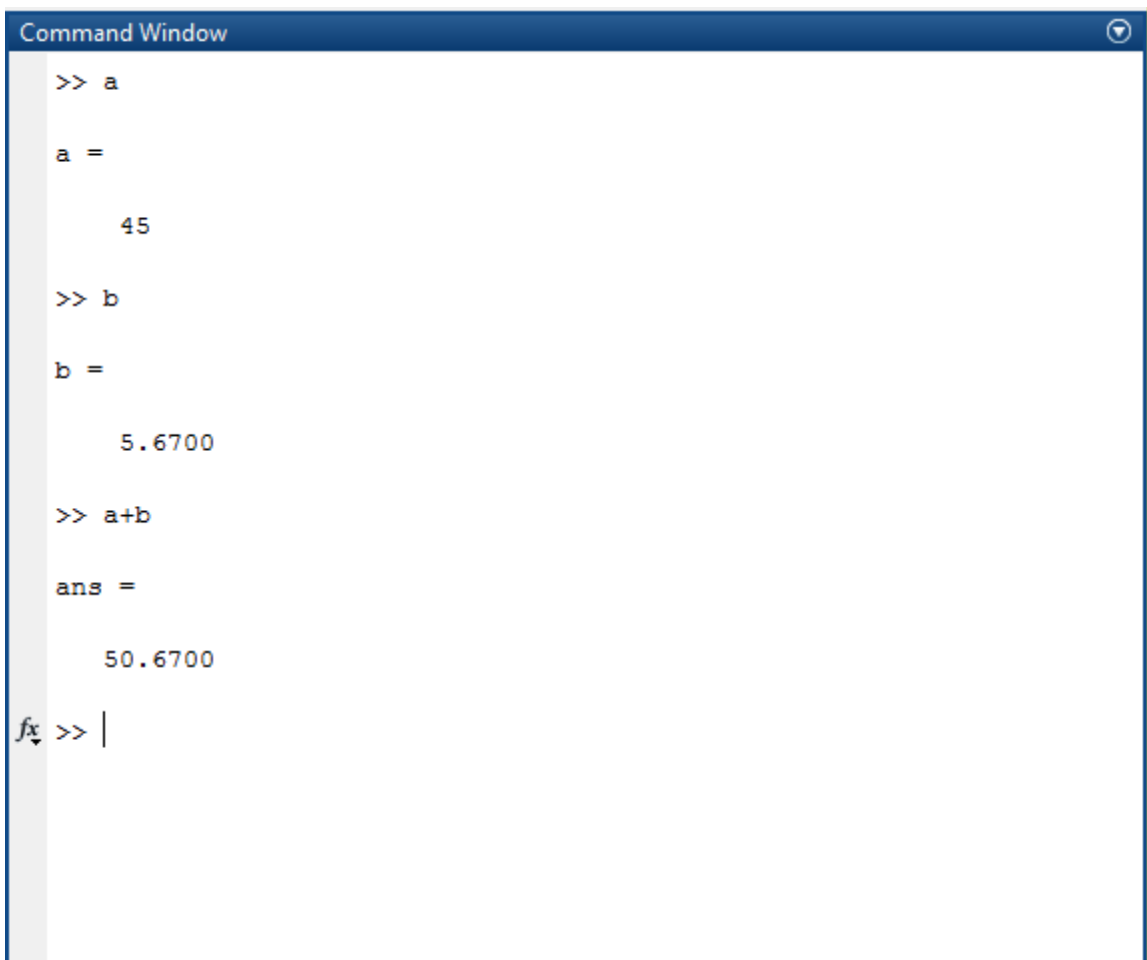
```
>> whos
```

Name	Size	Bytes	Class	Attributes
a	1x1	8	double	
array	1x8	64	double	
b	1x1	8	double	
matriks	3x3	72	double	
nama	1x12	24	char	

```
f> >> |
```

Untuk menampilkan rincian dari variable yang ada di dalam Workspace.

Tanda “;” pada akhir comand atau di editor maka hasil tidak akan ditampilkan di comand window.



```
>> a

a =

    45

>> b

b =

    5.6700

>> a+b

ans =

    50.6700

f> >> |
```

Contoh hasil yang ditampilkan di comand window

```

Command Window

>> a

a =

    45

>> b

b =

    5.6700

>> a+b;
fx >> |
  
```

Contoh hasil yang tidak ditampilkan dalam comand window

Untuk membersihkan comand window cukup dengan mengetikan “clc”, dan untuk menghapus semua variable cukup dengan mengetikan “clear”, dan “clear all” untuk membersihkan comand window dan juga variable di workspace.

### 3. Operator

#### 3.1. Operator Aritmatika

No	Operasi	Bentuk Aljabar	Bentukan Matlab	contoh
1	Penjumlahan	$A + B$	$A + B$	$3 + 4$
2	Pengurangan	$A - B$	$A - B$	$4 - 2$
3	Perkalian	$A \times B$	$A * B$	$6 * 3$
4	Pembagian	$A / B$	$A / B$	$5 / 3$
5	Exponensial	$A^B$	$A^B$	$2^2$

#### 3.2. Operator Ekspresi

Operator	Keterangan
$A < B$	A lebih kecil daripada B
$A > B$	A lebih besar daripada B
$A \leq B$	A lebih kecil atau sama dengan B
$A \geq B$	A lebih besar atau sama dengan B
$A == B$	A sama dengan B
$A \sim B$	A tidak sama dengan B

### 4. Control Flow

Seperti bahasa pemrograman lain, MATLAB juga memiliki *control flow*. *Control flow* terbagi menjadi 2 :

- a. *Loop Control Statements*

*Loop* atau perulangan biasa digunakan untuk melakukan proses yang berulang-ulang sesuai dengan kondisi yang telah ditentukan. Untuk melakukan proses *looping* dapat menggunakan beberapa syntax seperti :

- For

MATLAB Code	<pre>for variable = startval:endval     codehere end</pre>
Contoh	<pre>for i = 1:3     i end</pre>
Hasil	<pre>i = 1 i = 2 i = 3</pre>

- While

MATLAB Code	<pre>while (kondisi)     codehere end</pre>
Contoh	<pre>i = 0; while (i&lt;3)     i = i + 1 end</pre>
Hasil	<pre>i = 1 i = 2 i = 3</pre>

### b. Conditional Statements



*Conditional statements* merupakan suatu kondisi yang akan mengeksekusi *codes* tertentu jika memenuhi syarat yang telah ditentukan. Untuk memanfaatkan fitur ini dapat menggunakan beberapa syntax seperti :

- If

MATLAB Code	<pre>if (kondisi)     codehere end</pre>
Contoh	<pre>cek = true; if (cek == true)     i = 1 end</pre>
Hasil	i = 1

- If, else

MATLAB Code	<pre>if (kondisi1)     codehere else     codehere end</pre>
Contoh	<pre>cek = false; if (cek == true)     i = 1 else     i = 2 end</pre>
Hasil	i = 2

- If, elseif, else

Banyaknya *elseif* dapat disesuaikan dengan kebutuhan.

MATLAB Code	<pre> if (kondisi1)     codehere elseif (kondisi2)     codehere else     codehere end </pre>
Contoh	<pre> cek = 3; if (cek == 1)     i = 1 elseif (cek == 2)     i = 2 elseif (cek == 3)     i = 3 else     i = 4 end </pre>
Hasil	i = 3

### 5. Pembangunan Matriks

Terdapat beberapa cara untuk membangun atau pembuatan matriks dalam matlab, yang memiliki ketentuan dalam pembuatannya yaitu:

- Antara satu elemen dengan elemen yang lainnya dalam satu kolom dipisahkan dengan spasi atau koma (,).
- Memisahkan baris dengan menggunakan titik koma(;).
- Penulisan elemen berada dalam kurung siku ([ ]).

Cara – cara pembuatan matriks

- Memasukan secara langung elemen

```
Command Window

>> c = [4 5 3; 54 12 3; 6 4 2]

c =

     4     5     3
    54    12     3
     6     4     2

fx >> |
```

- Membangun data dengan batas awal dan batas akhir

```
Command Window

>> c = [20:27]

c =

    20    21    22    23    24    25    26    27

fx >> |
```

- Membangun data dengan batas awal, batas akhir dan *increment*

```
Command Window

>> c = [10:5:40]

c =

    10    15    20    25    30    35    40

fx >> |
```

- Membangun data dengan batas awal, batas akhir dan jumlah data di antara batas batas tersebut

```
Command Window

>> c = linspace(5,20,4)

c =

     5    10    15    20

fx >> |
```

Dalam matlab juga telah memberikan beberapa fungsi pembentukan matriks secara otomatis, diantaranya :

<code>zeros(n)</code>	Untuk membuat matriks yang semua datanya bernilai
<code>ones(n)</code>	Untuk membuat matriks yang semua datanya bernilai 1
<code>rand(n,m)</code>	Untuk menghasilkan matriks n x m dengan data random berdistribusi uniform
<code>randn(n,m)</code>	Untuk menghasilkan matriks n x m dengan data random berdistribusi normal
<code>eye(n)</code>	Untuk menghasilkan matriks identitas dengan ukuran n x n
<code>diag(n)</code>	Untuk menghasilkan matriks diagonal dengan elemen diagonalnya adalah vektor v
<code>magic(n)</code>	Untuk menghasilkan matriks ukuran n x n dengan isi elemen yang berbeda satu sama lain

### 6. Seleksi Data

Seleksi data biasanya digunakan untuk melakukan pengambilan sebagian data yang dibutuhkan pada data matriks.

Contoh data matriks :

Contoh code	<code>M = [1 2 3 4; 5 6 7 8; 9 10 11 12];</code>
Hasil	<pre> 1      2      3      4 5      6      7      8 9     10     11     12 </pre>

Ada bermacam-macam cara untuk mengambil sebagian data seperti :

#### a. `M(n)`

Contoh code	<code>Val = M(10)</code>
Hasil	<code>4</code>

#### b. `M(baris,kolom)`

Contoh code	Val = M(1,2)
Hasil	2

c. M(baris1:baris2,kolom1:kolom2)

Contoh code	Val = M(1:3,3:4)
Hasil	<pre> 3      4 7      8 11     12 </pre>

d. M(baris,kolom1:kolom2)

Contoh code	Val = M(2,2:4)
Hasil	<pre> 6      7      8 </pre>

e. M(baris1:baris2,kolom)

Contoh code	Val = M(1:3,1)
-------------	----------------

Hasil	<pre>1 5 9</pre>
-------	------------------

f. `M(baris,:)`

Contoh code	<code>Val = M(1,:)</code>
Hasil	<pre>1    2    3    4</pre>

g. `M(:,kolom)`

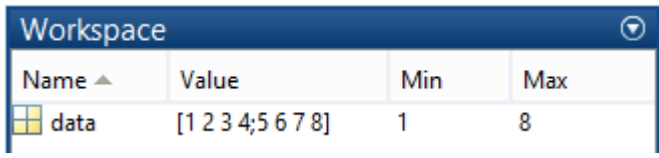
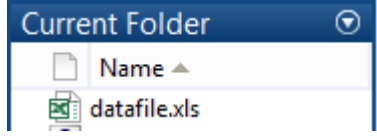
Contoh code	<code>Val = M(:,1)</code>
Hasil	<pre>1 5 9</pre>

## 7. Data Input Output File

### 7.1. File Excel

MATLAB dapat menyimpan dan membuka file Microsoft Excel (\*.xls, \*.xlsx, dan lain-lain). Berikut merupakan syntax yang dapat digunakan:

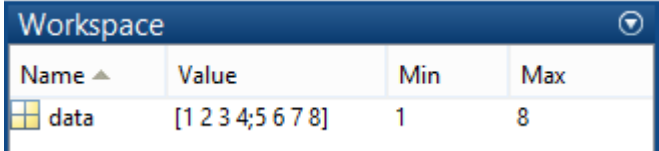
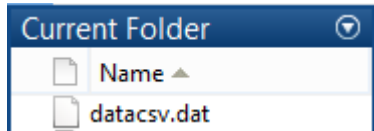
Load File MATLAB	<ul style="list-style-type: none"> <li>• <code>Data = xlsread(filename)</code></li> <li>• <code>Data = xlsread(filename,sheet)</code></li> <li>• <code>Data = xlsread(filename,xlRange)</code></li> <li>• <code>Data = xlsread(filename,sheet,xlRange)</code></li> <li>• <code>Data = xlsread(filename,sheet,xlRange,'basic')</code></li> </ul>
Contoh code	<code>data = xlsread('datafile.xls');</code>

Hasil	
Save File MATLAB	<ul style="list-style-type: none"> <li>• <code>xlswrite(filename,A)</code></li> <li>• <code>xlswrite(filename,A,sheet)</code></li> <li>• <code>xlswrite(filename,A,xlRange)</code></li> <li>• <code>xlswrite(filename,A,sheet,xlRange)</code></li> </ul>
	<pre>Data = [1 2 3 4; 5 6 7 8]; xlswrite('datafile',Data);</pre>
Hasil	

### 7.2. File CSV

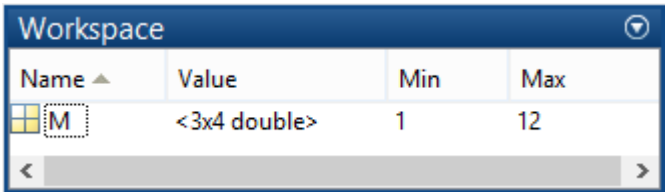
MATLAB dapat menyimpan dan membuka file berformat CSV (*Comma Separated Value*). Berikut merupakan syntax yang dapat digunakan:

Load File MATLAB	<ul style="list-style-type: none"> <li>• <code>Data = csvread(filename)</code></li> <li>• <code>Data = csvread(filename,row,col)</code></li> <li>• <code>Data = csvread(filename,row,col, csvRange)</code></li> </ul>
Contoh code	<pre>data = csvread('datacsv.dat');</pre>

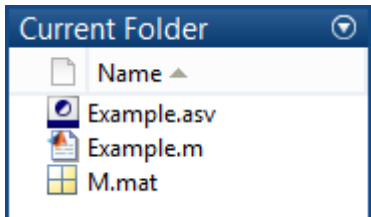
Hasil	
Save File MATLAB	<ul style="list-style-type: none"> <li>• <code>csvwrite(filename,M)</code></li> <li>• <code>csvwrite(filename,M,row,col)</code></li> </ul>
Contoh code	<pre>data = [1 2 3 4; 5 6 7 8]; csvwrite('datacsv.dat',data);</pre>
Hasil	

### 7.3. File MATLAB

File MATLAB merupakan file yang berasal dari hasil pekerjaan yang disimpan melalui MATLAB. Hasil file MATLAB berformat \*.mat. Berikut merupakan syntax yang dapat digunakan :

Load File MATLAB	<pre>load M</pre>
Hasil	



Save File MATLAB	<pre>M = [1 2 3 4; 5 6 7 8; 9 10 11 12]; save M</pre>
Hasil	

### 8. Fungsi dalam Matlab-File

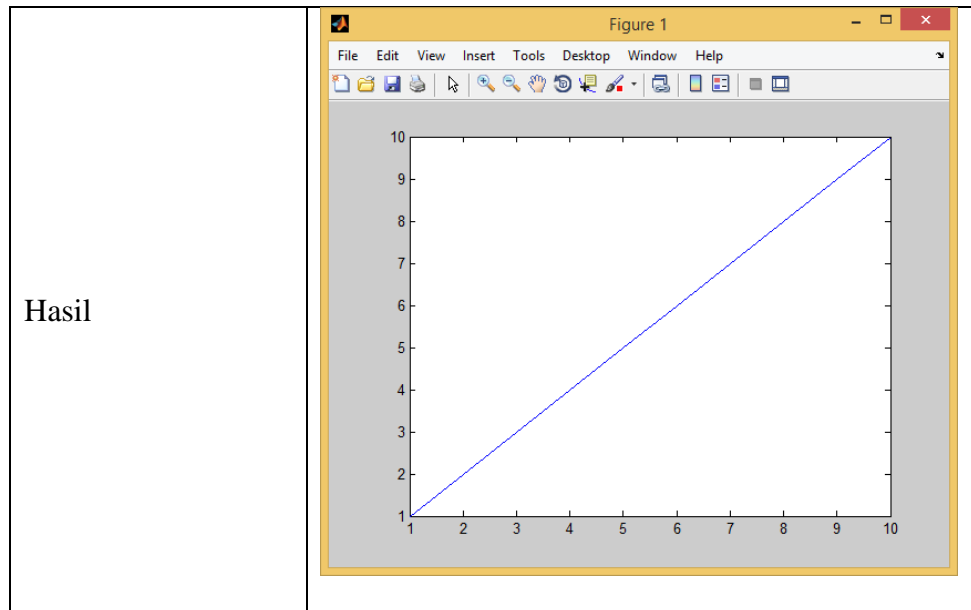
Fungsi adalah sekelompok pernyataan yang bersama-sama melakukan tugas. Dalam MATLAB, fungsi didefinisikan di file yang terpisah. Nama file dan fungsi haruslah sama. Fungsi beroperasi pada variable di workspace tersendiri dan juga biasa disebut dengan local workspace, terpisah dengan workspace yang biasa diakses di command prompt MATLAB yang bisa disebut base workspace. Fungsi dapat menerima lebih dari satu argumen input dan dapat juga mengembalikan output lebih dari satu juga.

### 9. Plots

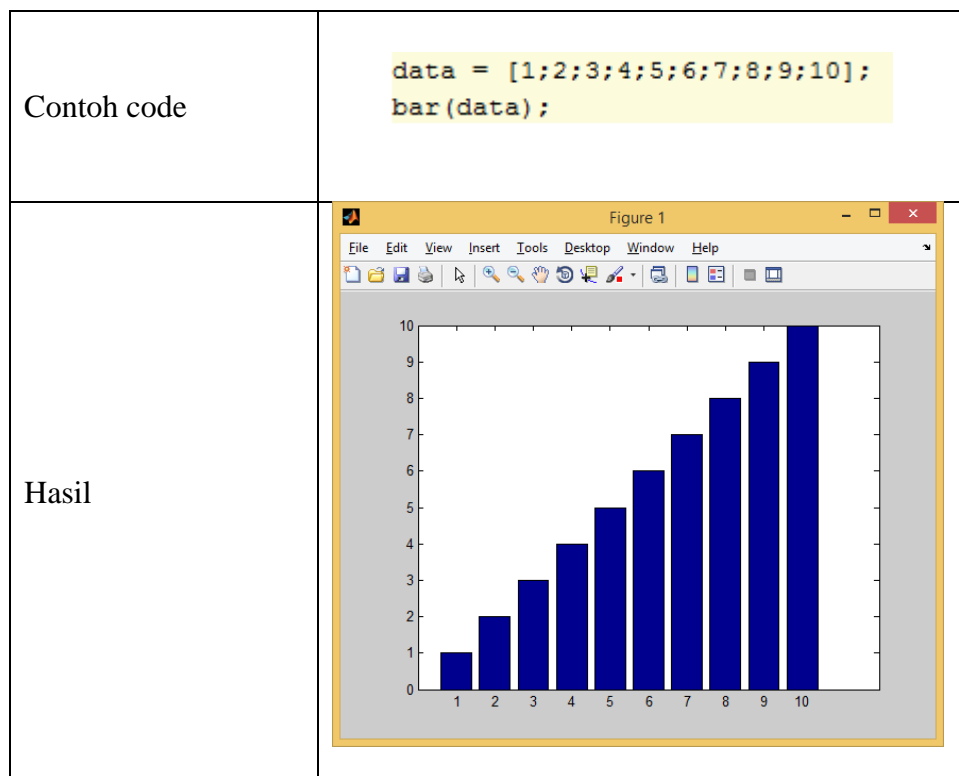
Plots merupakan salah satu fungsi yang disediakan MATLAB untuk menampilkan suatu graf pada *window screen*. Berbagai macam model plots seperti plot, bar, pie, hist, contour, dan lain-lain dapat digunakan hanya dengan memanggil *syntax* sesuai dengan penggunaan masing-masing plots. Berikut merupakan beberapa contoh penggunaan model plots:

- Plot

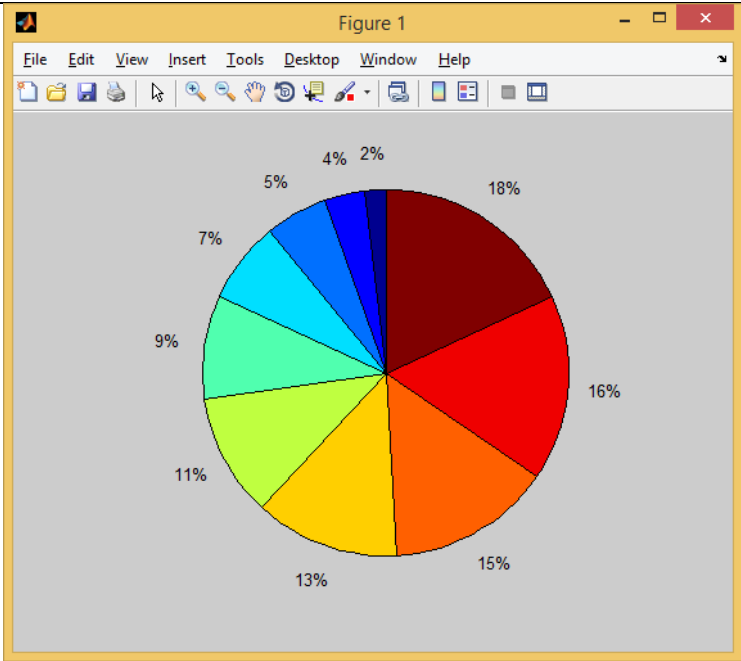
Contoh code	<pre>data = [1;2;3;4;5;6;7;8;9;10]; plot(data);</pre>
-------------	---



- Bar

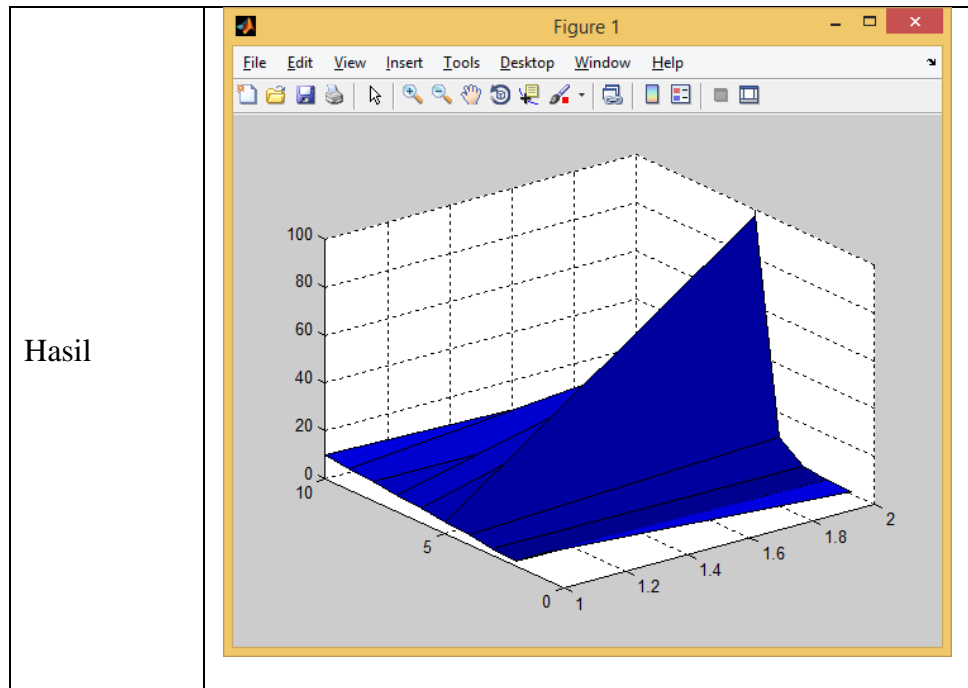


- Pie

Contoh code	<pre>data = [1;2;3;4;5;6;7;8;9;10]; pie(data);</pre>
Hasil	

- Surf

Contoh code	<pre>data = [10 1;2 1;3 2;4 10;5 98;         6 39;7 38;8 7;9 20;10 6]; surf(data);</pre>
-------------	--

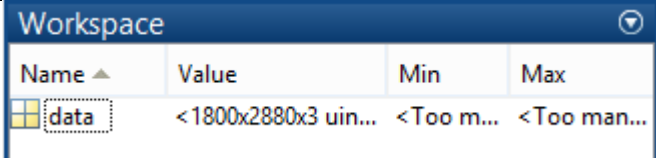


### 10. Input Output Image

MATLAB dapat digunakan untuk pemrosesan *image*. Terdapat beberapa syntax yang digunakan untuk pemrosesan *image*:

- Imread

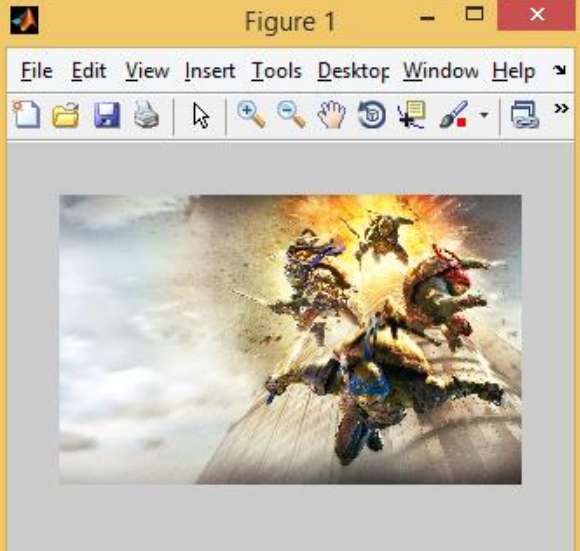
*Syntax* ini digunakan untuk melakukan penyimpanan data yang dihasilkan dari *file image* yang diproses ke suatu *variable*.

Contoh code	<pre>data = imread('sampleImage.jpg');</pre>
Hasil	

- Imshow

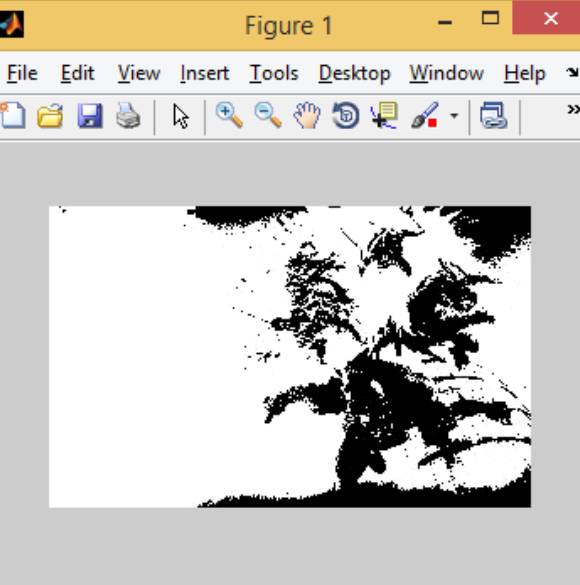
*Syntax* ini digunakan untuk menampilkan *image* yang diproses pada *windows* baru.

Contoh code	<pre>imshow('sampleImage.jpg');</pre>
-------------	---------------------------------------

Hasil	
-------	--

- Imwrite

*Syntax* ini digunakan untuk membangun *value* yang ada menjadi suatu *image*. *Syntax* ini biasanya digunakan setelah *image* sebenarnya telah melalui beberapa proses sehingga hasil menghasilkan *image* yang berbeda.

Contoh code	<pre>data = imread('sampleImage.jpg'); newdata = im2bw(data); imwrite(newdata, 'blackwhite.jpg'); imshow('blackwhite.jpg')</pre>
Hasil	

Dari hasil tersebut, *image* sebenarnya telah melalui proses perubahan warna menjadi hitam putih menggunakan *syntax im2bw*.

Selain itu masih banyak *syntax* yang dapat digunakan untuk melakukan pemrosesan *image*.