

# Dasar-dasar MATLAB

by Jusak Irawan, STIKOM Surabaya

## Perintah-Perintah Dasar

Note:

MATLAB akan memberikan respons secara langsung terhadap ekspresi apapun yang diketikkan pada editor MATLAB. Sebagai contoh:

```
>> 3 + 2
```

```
ans =  
      5
```

```
>> 2^5
```

```
ans =  
     32
```

```
>> sin(pi/2)
```

```
ans =  
      1
```

```
>> exp(1*pi)
```

```
ans =  
-1.0000 + 0.00001i
```

Penugasan nilai ke dalam variabel dapat dilakukan seperti contoh di bawah ini:

```
>> x = sqrt(3)
```

```
x =
```

```
1.7321
```

```
>> atan(x)
```

```
ans =
```

```
1.0472
```

```
>> pi/ans
```

```
ans =
```

```
3
```

```
>> y = exp(log(10))-10
```

```
y =
```

```
1.7764e-15
```

### **m-File**

m-file merupakan editor MATLAB yang berguna sebagai tempat menuliskan script dari kode-kode MATLAB. File ini disimpan dalam bentuk file dengan ekstensi .m. Script yang tersimpan dalam bentuk m-file dapat dieksekusi secara langsung melalui MATLAB command window.

*Note:*

### **Vektor**

Membuat vektor dengan MATLAB, sebagai contoh:

```
>> a = [1 2 3 4 5 6 7]
```

*Note:*

Table 1: Fungsi matematika dalam MATLAB

Simbol	Operasi
pi	$\pi$
exp(n)	Bilangan natural berpangkat- $n$ , $e^n$
log	Logaritma natural
log2	Logaritma dengan basis-2
log10	Logaritma dengan basis-10 (desimal)
sin	Sinus
cos	Cosinus
tan	Tangent
asin	Inverse sinus
acos	Inverse cosinus
atan	Inverse tangent

a =

1 2 3 4 5 6 7

Membuat vektor dengan bilangan ascending dengan kenaikan 2:

```
>> t = 0:2:20
```

t =

0 2 4 6 8 10 12 14 16 18 20

Manipulasi vektor dapat dilakukan seperti contoh di bawah ini:

```
>> b = a + 5
```

y =

6 7 8 9 10 11 12

```
>> c = a + b
```

```
c =  
    7 9 11 13 15 17 19
```

```
>> x = [1 2 3];
```

```
>> y = [3 4 5];
```

```
>> z = x.*y
```

```
z =  
    3 8 15
```

Operator dalam MATLAB didefinisikan dalam Tabel 2

Table 2: Operator dalam MATLAB

Simbol	Operasi
*	Perkalian
/	Pembagian
+	Penjumlahan
-	Pengurangan
^	Pangkat
.*	Perkalian setiap elemen di dalam array
./	Pembagian setiap elemen di dalam array
.^	Pemangkatan setiap elemen di dalam array

### Matrik (Array)

Membuat array dengan MATLAB, sebagai contoh:

```
>> B = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
B =
```

Note:

```
1 2 3 4
5 6 7 8
9 10 11 12
```

```
>> size(B)
```

```
ans =
```

```
3 3
```

```
>> C = ones(1,3)
```

```
ans =
```

```
1 1 1
```

```
>> C'
```

```
ans =
```

```
1
```

```
1
```

```
1
```

Perintah-perintah yang dapat digunakan untuk membangun matriks dalam MATLAB didefinisikan dalam Tabel 3

### **Operasi Matrik**

Operasi perkalian matrik mengikuti aturan perkalian sebuah matrik. Sebagai contoh:

```
>> c = [ 1 10 20 30 ];
```

```
>> B*c
```

*Note:*

Table 3: Perintah membangun matrik

Perintah	Fungsi
<code>eye</code>	matrik identitas
<code>zeros</code>	matrik dengan semua elemen bilangan 0
<code>ones</code>	matrik dengan semua elemen bilangan 1
<code>diag</code>	diagonal matrik
<code>triu</code>	matrik upper-triangular
<code>tril</code>	matrik lower-triangular
<code>repmat</code>	duplikasi matrik
<code>inv</code>	inverse matrik
<code>det</code>	determinan matrik
<code>sum</code>	menjumlah setiap elemen kolom dari matrik
<code>eig</code>	eigenvalue dan eigenvektor dari matrik

```
ans =
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

Pesan error ini disebabkan oleh dimensi kedua matrik untuk operasi perkalian tidak sesuai, matrik B berdimensi  $3 \times 4$  sedangkan matrik c berdimensi  $1 \times 4$ . Agar dimensi kedua matrik bersesuaian, operasi transpose harus dilakukan pada matrik c sebagai berikut:

```
>> c = [ 1 10 20 30 ];
>> B*c'

ans =
    201
    445
    689
```

```
>> help sum
>> sum(B)

ans =
    15    18    21    24
```

### Control Flow

Note:

MATLAB mengenal dua macam cara untuk melakukan proses looping atau iterasi, yaitu: *for loop* dan *while loop* dan dua macam cara untuk melakukan seleksi, yaitu: *if-else* dan *switch case*.

*For loop* memungkinkan sekelompok perintah diulang sebanyak suatu jumlah yang tetap. Contoh:

```
>> for j=1:4
j end

j =
    1

j =
    2

j =
    3

j =
    4
```

```
>> for j=1:4
v(j)=j; end

v =
    1    2    3    4
```

Bagaimana hasil dari contoh script di bawah ini?

```

>> clear all
>> B = [[1 2 3]' [3 2 1]' [2 1 3]']
>> for j=2:3,
for i=j:3, B(i,:) = B(i,:) - B(j-1,:)*B(i,j-1)/B(j-1,j-1); end end

```

*While loop* akan melakukan perulangan (iterasi) secara terus menerus sampai suatu kondisi tertentu dipenuhi. Contoh:

```

>> i=0;
while i<5 disp(i); i=i+1; end

```

```

0
1
2
3
4

```

Bagaimana hasil dari contoh script di bawah ini, jika diketahui sebuah persamaan differential  $y' = x - |y|$ ,  $y(0) = 1$  diaproksimasi dengan metoda Euler?

```

>> h = 0.001;
>> x = [0:h:2];
>> y = 0*x;
>> y(1) = 1;
>> i = 1;
>> size(x)
>> max(size(x))
>> while(i<max(size(x)))
y(i+1) = y(i) + h*(x(i)-abs(y(i))); i = i + 1; end
>> plot(x,y,'go')
>> plot(x,y)

```



*Switch* melakukan pemilihan berdasarkan masing-masing case yang telah didefinisikan. Sebagai contoh, ketikkan code berikut ini ke dalam m-file dan lihatlah hasilnya.

```
bilangan=5; x=rem(bilangan,2); switch(x)
    case 1
        disp(['bilangan',num2str(bilangan),...
            'adalah bilangan ganjil'])
    case 2
        disp(['bilangan',num2str(bilangan),...
            'adalah bilangan genap'])
    otherwise
        disp('Bilangan tidak mungkin ada')
end
```

*If-Else* melakukan pemilihan berdasarkan hasil tes rasional. Sebagai contoh, ketikkan code berikut ini ke dalam m-file dan lihatlah hasilnya:

```
a = 4; b = 4; if (a<b)
    j = -1;
else if (a>b)
    j = 2;
else
    j = 3
end
```

## **Plotting**

*Note:*

Salah satu keunggulan MATLAB dibandingkan dengan bahasa pemrograman lain adalah kemampuannya untuk menghasilkan plotting grafik hasil simulasi dengan tingkat keakuratan yang cukup tinggi. Sebagai contoh, ketikkan code berikut ini ke dalam m-file dan lihatlah hasilnya:

```

h = 1/16; x = 0:h:1; y = 0*x; y(1) = 1; for i=2:max(size(y)),
    y(i) = y(i-1) + h/y(i-1);
end true = sqrt(2*x+1); figure(1); plot(x,y,'go',x,true)
figure(2); plot(x,abs(true-y),'mx')

```

Kedua gambar hasil dari plotting di atas dapat digabungkan ke dalam satu buah frame dengan perintah subplot.

```

figure(3); subplot(1,2,1); plot(x,y,'go',x,true) subplot(1,2,2);
plot(x,abs(true-y),'mx')

```

Lanjutkan dengan mengetikkan code MATLAB di bawah ini:

```

clf h = h/2; x1 = 0:h:1; y1 = 0*x1; y1(1) = 1;

for i=2:max(size(y1)),
    y1(i) = y1(i-1) + h/y1(i-1);
end true1 = sqrt(2*x1+1);

plot(x1,y1,'go',x1,true1) plot(x1,abs(true1-y1),'mx')
subplot(1,2,1); plot(x,abs(true-y),'mx') subplot(1,2,2);
plot(x1,abs(true1-y1),'mx')

title('Errors for h=1/32') xlabel('x'); ylabel('|Error|');
subplot(1,2,1); xlabel('x'); ylabel('|Error|');

title('Errors for h=1/16')

```

Tipe garis, simbol dan warna yang dapat digunakan dalam MATLAB ditunjukkan dalam Tabel 4

### Function

Sebuah *function* ditulis dalam m-file dengan nama file sama dengan nama function. Sebagai contoh:

Note:

Table 4: Tipe garis, simbol dan warna

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dash-dot
c	cyan	+	plus	-	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
b	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

```
function [x]=jumlah(a,b)
% Fungsi ini melakukan penjumlahan a dan b
x=a+b;
```

Pada MATLAB window command panggil fungsi tersebut:

```
>> help jumlah
Fungsi ini melakukan penjumlahan a dan b
>> a=5;
>> b=7;
>> hasil = jumlah(a,b)

hasil =

    12
```

## LATIHAN

1. Jika diketahui konversi dari Fahrenheit ke Celcius adalah  $C = (F - 32) * 5/9$ , tentukan nilai celcius untuk suhu  $30^{\circ}\text{F}$  sampai  $200^{\circ}\text{F}$ ! Kemudian lakukan plotting untuk kedua buah grafik Celcius dan Fahrenheit!
2. Sekelompok mahasiswa memiliki nilai angka sebagai berikut: 35, 56, 78, 97, 67, 45, 85, 77, 62, 40. Konversikan nilai angka tersebut ke nilai huruf jika diketahui:  $E < 50$ ,  $50 < D \leq 60$ ,  $60 < C \leq 70$ ,  $70 < B \leq 80$  dan  $A > 80$ .