

DAFTAR ISI

	hal
Materi 1 Pendahuluan: Pengantar Matlab	1
Materi 2 Matriks	3
Materi 3 Diary, Save, Format, Fungsi Matematika, Relasi & Logika	5
Materi 4 Penyelesaian SPL & Teks	7
Materi 5 Polynomial Dan Pencocokan Kurva	10
Materi 6 Grafik 2 Dimensi	13
Materi 7 Minimum Dan Maksimum Fungsi	18
Materi 8 Grafik 3D	23
Materi 9 Fungsi Dan Integral	26
Materi 10 Ekspresi Simbolik	30
Materi 11 Script M-File	35
Materi 12 Pemrograman: For-End & If-End	37
Materi 13 If –Elseif-End & Switch Case	39
Materi 14 While-End	41

MATERI 1

PENDAHULUAN: PENGANTAR MATLAB

KOMPETENSI

Mahasiswa dapat menggunakan program komputer yang sesuai untuk menyelesaikan masalah-masalah matematika

Bahasan Aplikasi komputer

1. Pendahuluan: pengantar MATLAB
2. Matriks dan manipulasi matriks
3. Fungsi Matematika, Relasi dan Logika
4. SPL dan Teks
5. Polynomial dan pencocokan kurva
6. Grafik 2 D
7. Titik balik maksimum & minimum suatu fungsi
8. Grafik 3 D
9. Fungsi dan integral
10. Ekspresi Symbolic
11. Script m-file
12. PEMROGRAMAN: for-end & if – end
13. PEMROGRAMAN: if - elseif- end & Switch-case
14. PEMROGRAMAN: while - end

Titik berat program komputer yang digunakan :

- MATLAB (MATrix LABoratory, lihat <http://www.mathworks.com>)
- dapat mengerjakan matematika sederhana, menggambar grafik fungsi kompleks sampai dengan penyusunan program

Contoh pemakaian MATLAB untuk menyelesaikan masalah matematika sederhana:

Diketahui persoalan sbb:

Saila membeli :

- 4 buah penghapus harganya 125 perbuah,
- 6 buku halus @ 1750
- 4 pensil @1000.

Berapa banyak item yang dibeli Saila dan berapa dia harus membayar?

Pengerjaan dengan MATLAB:

Ada 2 cara :

cara 1 : menghitung secara langsung

```
>> 4+6+4
```

```
>> 4*125+6*1750+4*1000
```

cara 2 : menggunakan variabel

```
>> penghapus = 4
```

```
>> buku=6
```

```
>> pensil=4;
```

```
>> item =penghapus+buku+pensil
>> harga=penghapus*125+buku*1750+
    pensil*1000
```

dengan cara 2, dapat melakukan operasi yang lain dengan memanggil nama variabelnya

```
>> rata_rata=harga/item
```

VARIABEL



Sifat-sifat variabel dalam MATLAB, yaitu:





- Bersifat *Case sensitive*, membedakan huruf besar dan huruf kecil. Misal : Item \neq item, f \neq F.
- Maksimum 19 karakter
- Harus diawali dengan huruf, boleh diikuti angka, tetapi tidak sebaliknya. Misal **a12**, tetapi tidak boleh **12a**.
- Dua kata dihubungkan dengan underscore. Misal : buku_tulis
- Bukan variabel khusus yang sudah ditentukan dalam MATLAB. Misal: ans, pi, eps, flops, inf, NaN, i (dan) j, nargin, nargout, realmin, realmax

Untuk menghindari terjadinya kesalahan, akan lebih mudah jika SEMUA PERINTAH dalam MATLAB diketik dalam HURUF KECIL.

Operasi aritmatika dasar

Operasi	Simbol	Contoh
Penjumlahan	+	5+3
Pengurangan	-	13-12
Perkalian	*	3.14*0.85
Pembagian	/ atau \	56/8 = 8\56
Pangkat	^	5^2

 **Tip1:** Untuk memunculkan kembali perintah-perintah yang sudah diketik dalam MATLAB, tekan tombol panah atas  pada keyboard.

 **Tip2:** Jika perintah yang dimasukkan salah, edit kembali perintah tersebut dengan cara tekan tombol panah atas  pada keyboard, kemudian gunakan panah kiri  atau panah kanan  untuk menuju bagian yang salah, lalu betulkan bagian tersebut.

MATERI 2

MATRIKS

Ada 3 cara penulisan:

- cara1 :
a=[1 2 3 ; 4 5 6; 7 8 9]
- cara2 :
a=[1 2 3 (tekan enter)
 4 5 6
 7 8 9]
- Cara 3 : dengan menuliskan perbaris
a1=[1 2 3];
a2=[4 5 6];
a3=[7 8 9];
a =[a1;a2;a3]

Cara 3 bermanfaat jika matriks berukuran cukup besar dan mengandung beberapa elemen yang sama

Misal :

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$a = [a1;a2; a1;a2;a3]$$

Matriks khusus

Berikut ini beberapa perintah untuk menghasilkan matriks matriks khusus.

Perintah

HASIL

- zeros(3) matrik zero (nol) ukuran 3 x 3
- ones(2,4) matrik yang elemennya semua 1 ukuran 2 x 4
- ones(3)*pi matrik yang elemennya perkalian pi (3.14) dengan 1 berukuran 3 x 3
- rand(3,1) matrik berukuran 3 x 1 yang berelemen bilangan random (acak) positif
- randn(3,1) matrik berukuran 3 x 1 yang berelemen bilangan random positif & negatif
- magic(4) matrik bujursangkar ajaib berukuran 4 x 4
- eye(3) matrik identitas berukuran 3 x 3
- pascal(4) matrik ukuran 4 x 4 berelemen segitiga pascal

Operasi matriks

- Penjumlahan : c = a+b Ukuran matriks a dan b harus sama
 d = a+5 semua elemen a ditambah dengan 5
- Pengurangan : e = a-b ukuran matriks a dan b harus sama
 f = a-3
- Perkalian : a*b ukuran kolom a harus sama dengan ukuran baris b

	$a.*b$	perkalian antar elemen yang indeksnya sama
	$2*a$	semua elemen dikalikan 2
• Pembagian :	$c./a$	pembagian per elemen yang indeksnya sama
• Transpose :	$g=a'$	
• Penggabungan :	$h =[a b]$	menggabung ke kanan, ukuran baris a harus sama dengan ukuran baris b
	$k =[a; b]$	menggabung ke bawah, ukuran kolom a harus sama dengan ukuran kolom b

Pengaksesan elemen matriks.

Berikut ini ditampilkan beberapa perintah untuk mengakses elemen tertentu dari sebuah matriks

Elemen yang diakses	perintah
• baris ke 2 kolom ke 3	: $a(2,3)$
• kolom ke 3	: $a(:,3)$
• baris ke 2	: $a(2,:)$
• baris ke 2 kolom 1 s/d 3	: $a(2,1:3)$
• baris ke 1,4,5 semua kolom	: $a([1 4 5],:)$

MANIPULASI MATRIKS

$A = (1:10)$ menghasilkan matriks baris yang elemennya 1 sampai dengan 10

$B = (1:2:10)$ menghasilkan matriks baris yang elemennya 1 sampai dengan 10 dengan lompatan ke elemen berikutnya sebesar 2

$D = (10:-2:1)$

$E = \text{linspace}(1,10)$

$F = \text{linspace}(1,10,20)$

$G = \text{logspace}(0,3)$

$H = \text{logspace}(0,3,10)$

$A = [1 2 3 4; 5 6 7 8; 9 10 11 12]$

$A(2)$: menunjuk elemen ke 2 dari A

$A(5)$: elemen ke 5 dari A

$A(2,3)=2$: Mengubah elemen matriks baris ke 2 kolom 3 dengan 2

$A(2,6)=1$: Mengubah elemen matriks baris ke 2 kolom 6 dengan 1

$B = A(1:2,2:3)$: Membentuk matriks B dari baris 1 sd 2 dan kolom 2 sd 3 matriks A

UKURAN MATRIKS

- $S=\text{size}(A)$ Menentukan ukuran baris dan kolom matriks A
- $[b,k] = \text{size}(A)$ ket: b = baris k = kolom
- $b=\text{size}(A,1)$ banyaknya baris matriks A
- $c=\text{size}(A,2)$ banyaknya kolom matriks A
- $\text{length}(A) = \max(\text{size}(A))$

Menentukan panjang vektor baris/kolom matriks A

- $C = (0:0.1:1)*\pi$
- $\text{size}(C)$
- $\text{length}(C)$

MATERI 3

Diary, Save, Format, Fungsi Matematika, Relasi & Logika

Setelah masuk command window, Lakukan perintah berikut. Perhatikan hasilnya

```
>> diary latihan_selasa      %Menyimpan diary ke dalam ke file yang bernama latihan_selasa
>> format long
>> a=pi
>> format short
>> b=pi
>> help format
>> save lat_selasa
>> diary off                %Mengakhiri perintah diary dan menutup file
>> dir
>> clc
>> a
>> clear all
>> a
>>load lat_selasa
>> a
>> type latihan_selasa
```

Diary

```
>> diary                    % Menyimpan input dari pemakai dan hasil dari jendela command
                           dalam bentuk teks ASCII dalam direktori yang sedang aktif
>> diary nama_file        %Menyimpan diary ke dalam ke file yang bernama nama_file
>> diary off              %Mengakhiri perintah diary dan menutup file
```

Menyimpan data

Untuk menyimpan data dari file, ada 2 cara :

1. Dari menu File → Save Workspace as
2. Dari command window, ketikkan perintah

```
>> save                    %Digunakan untuk menyimpan variabel Matlab dalam format
                           biner di file matlab.mat
```

```
>> save data              % Digunakan untuk menyimpan variabel Matlab dalam format biner
                           di file data.mat
```

Format Penampilan Angka

Perintah format digunakan untuk mengatur format tampilan output.

Contoh :

>> format long Digunakan untuk mengatur format angka pi ke dalam format long (15 digit)

>> pi

Dengan cara yang sama, cobalah untuk format short, short e, long e, short g, jong g, hex, bank, +, rat.

Diskusi I:

1. Apa perbedaan pemanggilan >>a sebelum dan sesudah >>clear all dan >> load lat_selasa
2. Buat kesimpulan tentang perintah yang baru anda temui dan apa fungsinya
3. Dari bentuk bentuk format penampilan angka, jelaskan perbedaan dari format-format penulisan tersebut.

Fungsi Matematika

Ketikkan perintah berikut:

```
>> s_rad = 0:pi/3:pi      >>s1=asin(0.7071)
>> x=sin(s_rad)         >>s2=acos(0.8660)
>> y=cos(s_rad)         >>s3=atan(1.7321)
>> z=tan(s_rad)         >>sdt_rad=[s1,s2,s3]
>> s_der = s_rad*180/pi  >>sdt_der=sdt_rad*180/pi
>> tabel=[s_der;x;y;z]
```

Relasi dan Logika

Operasi Relational	Operasi Logika
<p>Sintaks</p> <p>A < B</p> <p>A > B</p> <p>A <= B</p> <p>A >= B</p> <p>A == B</p> <p>A ~= B</p> <p>Contoh :</p> <p>>> X= [1 2 3; 4 5 6; 7 8 10]</p> <p>>> X>5</p> <p>>> A = [2 7 6;9 0 5;3 0.5 6];</p> <p>>> B = [8 7 0;3 2 5;4 -1 7];</p> <p>>> A==B</p> <p>>>find(X>5)</p> <p>Cobalah juga untuk operasi A < B, A > B, A <= B, A >= B,A ~= B</p>	<ul style="list-style-type: none"> ● & : And ● : Or ● ~ : Not ● xor <p>Contoh :</p> <p>>> A = [0 1 1 0 1];</p> <p>>> B = [1 1 0 0 1];</p> <p>>> A&B</p> <p>>> A B</p> <p>>> ~A</p> <p>>>xor(A,B)</p> <p>>> find((X>5)&(X<9))</p>

MATERI 4

PENYELESAIAN SPL & TEKS

Jika diketahui sebuah SPL berikut:

$$\begin{aligned}x_1 + x_2 - x_3 &= 1 \\-2x_1 - 6x_2 + 4x_3 &= -2 \\-x_1 - 3x_2 + 3x_3 &= 1\end{aligned}$$

SPL di atas dapat dinyatakan dalam bentuk persamaan matriks $Ax=b$ dengan

$$A = \begin{pmatrix} 1 & 2 & -1 \\ -2 & -6 & 4 \\ -1 & -3 & 3 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} \quad \text{dan vector } x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Matlab hanya memberikan **satu** alternatif solusi dari SPL, akan tetapi solusi tersebut merupakan yang terbaik di antara solusi lain yang mungkin.

Ada tiga macam SPL :

1. Jumlah persamaan sama dengan jumlah variabel (Matriks bujur sangkar)

Ada dua cara penyelesaian:

- a. $x=A^{-1} * b$
- b. $x=A \setminus b$ (pembagian kiri matriks)

Selisih $A*x-b$ disebut **Residu**. Jika residu merupakan vector nol maka solusi bersifat eksak.

Contoh di atas

2. Jumlah persamaan lebih banyak daripada jumlah variable (kasus berlebihan)

$$x=A \setminus b,$$

(disebut penyelesaian kuadrat terkecil)

merupakan solusi dengan error kuadrat terkecil dari residu $A*x-b$

Contoh:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \\ 2 & 5 & 8 \end{pmatrix} \quad b = \begin{pmatrix} 366 \\ 804 \\ 351 \\ 514 \end{pmatrix}$$

3. Jumlah variabel lebih banyak daripada jumlah persamaan (kasus kekurangan)

$$x=\text{pinv}(A)*b$$

(disebut penyelesaian normal minimum)

memberikan solusi panjang atau normal x lebih kecil dari semua penyelesaian lain yang mungkin.

Memeriksa normal x dengan perintah **norm(x)**.

Contoh:

$$A = \begin{pmatrix} 1 & 4 & 7 & 2 \\ 2 & 5 & 8 & 5 \\ 3 & 6 & 0 & 8 \end{pmatrix} \quad b = \begin{pmatrix} 366 \\ 804 \\ 351 \end{pmatrix}$$

SELESAIKANLAH SPL-SPL BERIKUT:

$$\begin{aligned} 1. \quad & 3x_1 - x_2 + 2x_3 = 10 \\ & 3x_2 - x_3 = 15 \\ & 2x_1 + x_2 - 2x_3 = 0 \end{aligned}$$

$$\begin{aligned} 2. \quad & -1x + 7y + 5z = 12 \\ & 6x + 3y - 2z = 3 \\ & 8x + z = 10 \\ & 4x - 4y + 2z = -9 \end{aligned}$$

$$\begin{aligned} 3. \quad & -2x_1 + x_2 + 5x_3 = 1 \\ & 3x_2 - x_3 = 4 \\ & 8x_1 + 2x_2 = 5 \end{aligned}$$

TEKS

Karakter string adalah nilai-nilai ASCII yang ditampilkan representasinya. Penulisan string dalam MATLAB harus di antara tanda apostrof (').

Contoh :

```
>>t = 'Aplikasi komputer'
```

```
>>size(t)          melihat ukuran t
```

Untuk melihat representasi ASCII karakter string digunakan perintah sbb :

```
>>A=double(t)
```

Untuk mengembalikan ke bentuk string kembali gunakan perintah :

```
>>char(A)
```

Manipulasi Teks

Teks dalam MATLAB adalah matriks yang elemen-elemennya berupa karakter/huruf. Teks diperlakukan sama seperti matriks, sehingga perintah-perintah yang digunakan untuk memanipulasi teks sama dengan perintah untuk mengakses dan memanipulasi matriks

Contoh :

```
U=t(10:17)   mengakses/menampilkan elemen teks t mulai dari elemen ke 10 sampai ke 17
```

```
V=t(17:-1:10) menampilkan karakter secara terbalik, dimulai dari karakter ke 17 sampai 10
```

```
W=t(10 :17)'
```

```
A='matematika'
```

```
B='fisika'
```

```
C=[A B]
```

```
C=[A ' ' B 'atau Biologi.'] menggabungkan string A, B dan kata 'Biologi'
```

```
D=char(A,B)           menggabungkan string A dan B
```

```
size(D)
```

Konversi String

Matlab menyediakan beberapa perintah untuk mengkonversi string ke bilangan/numerik dan sebaliknya.

- bin2dec :mengkonversi **string** biner ke **bilangan** bulat desimal
Contoh : bin2dec('110011')
- dec2bin :mengkonversi **bilangan** bulat desimal ke **string** biner
Contoh : dec2bin(45)
- hex2dec :mengkonversi **string** heksadesimal ke **bilangan** bulat desimal

- Contoh : `hex2dec('A1F')`
- `dec2hex` : mengkonversi **bilangan** bulat desimal ke **string** heksadesimal
Contoh : `dec2hex(31)`
- `base2dec` : mengkonversi **string** berbasis x ke **bilangan** bulat desimal
Contoh :
`>> base2dec('212',3)` mengkonversi 212_3 ke bentuk desimal
- `dec2base` : mengkonversi **bilangan** desimal ke **bilangan** basis n
Contoh :
`>>dec2base(23,2)` mengkonversi 23_{10} ke basis 2
`>>dec2base(23,8)` mengkonversi 23_{10} ke basis 8
- `num2str` : (baca: numerik to string) mengkonversi bilangan numerik ke string

Contoh

```
>>Jari2=2;
>>Luas=pi*Jari2^2;
>>Y=['Lingkaran dengan jari-jari' num2str(Jari2) 'memunyai luas' num2str(Luas) '.']
>>Y=['Lingkaran dengan jari-jari' ,(Jari2), 'memunyai luas' ,(Luas) ,'.']
>>Y=['Lingkaran dengan jari-jari' ,2, 'memunyai luas' , 12.5664 ,'.']
```

- `strrep(t,'a','A')` mengkapitalkan semua huruf a/mengganti huruf a dengan A
- `lower(t)` mengkonversi string t ke huruf kecil
- `upper(t)` mengkonversi string t ke huruf besar

Menampilkan string di layar

- `disp` : menampilkan string tanpa menampilkan nama variabelnya

Contoh : `t, disp(t)`

Pencarian karakter dalam string

```
>>find(t=='p')                    menemukan karakter p dalam string t
>>findstr(t,'komputer')        menemukan kata 'komputer' dalam string t
```

Hasil pencarian string berupa nomor elemen yang menunjukkan posisi awal karakter atau kata yang dicari dalam string bersangkutan.

MATERI 5

Polynomial dan Pencocokan Kurva

1. Mencari akar
2. Mengevaluasi Nilai
3. Menggambar grafik
4. Penjumlahan & Pengurangan
5. Perkalian & Pembagian
6. Turunan & Integral
7. Polynomial rasional

- Polynomial dalam MATLAB dinyatakan sebagai vektor baris yang memuat koefisien dari variabel dengan derajat yang makin menurun.

Misal : $p(x) = x^3 - 2x - 5$

dalam MATLAB dinyatakan dengan `>>p = [1 0 -2 -5]`

1. Mencari akar

- `r = roots(p)` mencari akar-akar p
- Sebaliknya, jika diketahui akar-akar suatu polinomial, untuk mengetahui koefisien-koefisien dari polinomialnya, digunakan perintah `>>p1=poly(r)`

2. Mengevaluasi Nilai

- Untuk mengevaluasi (menghitung nilai) polynomial pada suatu bilangan tertentu, digunakan perintah `polyval`.
- Contoh :
mengetahui nilai polynomial p di $x = 5$, perintahnya:
`>>polyval(p,5)`

Misal $p(x) = x^3 + 4x^2 - 7x - 10$

Mengevaluasi nilai $p(x)$ pada interval $[-1,3]$ dilakukan dengan cara sbb:

```
>>p=[1 4 -7 -10];
```

```
>>x=linspace(-1,3); % membuat interval -1,3 yang memuat 100 titik
```

```
>>V=polyval(p,x); % mengevaluasi nilai p(x) di x
```

3. Menggambar grafik

Misal akan digambar grafik polinomial $p(x) = x^3 + 4x^2 - 7x - 10$ pada interval $[-1,3]$

Langkah-langkah menggambar grafik polinomial adalah sbb:

1. Mendefinisikan polinomial
`>>p=[1 4 -7 -10];`
2. Mengevaluasi nilai polinomial pada interval yang ditentukan
`>>x=linspace(-1,3); V=polyval(p,x);`
3. Menggambar grafik dengan perintah **plot**

```
>>plot(x,V), title('x^3-4x^2-7x-10'),xlabel('x')
```

Perintah **title** untuk memberi judul grafik dan **xlabel** untuk memberi label sumbu x

4. Penjumlahan & Pengurangan

- Misal

$$a(x) = x^2 + 2x + 3$$

$$b(x) = 5x + 6$$

derajat polynomial a dan b tidak sama

Dalam matlab:

```
>>a=[1 2 3]; b=[0 5 6];
```

Dalam polynomial b, tidak ada x^2 , sehingga agar kedua polynomial dapat dijumlahkan, maka koefisiennya diisi dengan 0

```
>>a+b % mencari a(x) + b(x),
```

```
>>a-b % mencari a(x) - b(x)
```

5. Perkalian & Pembagian

- Misal

$$a(x) = x^2 + 2x + 3 \rightarrow \text{>>a}=[1 \ 2 \ 3]$$

$$b(x) = 4x^2 + 5x + 6 \rightarrow \text{>>b}=[4 \ 5 \ 6]$$

Dalam matlab:

- $a(x) * b(x)$, dicari dengan fungsi *conv* (singkatan dari convolution), yaitu

```
>>conv(a,b)
```

- $a(x) / b(x)$, dicari dengan fungsi *deconv* (singkatan dari deconvolution), yaitu

```
>>deconv(a,b)
```

6. Turunan & Integral

$$a(x) = x^2 + 2x + 3$$

```
>>a=[1 2 3]
```

```
>>polyder(a) % Turunan dari a(x)
```

```
>>polyint(a) % Integral dari a(x)
```

7. Polynomial Rasional

$$\frac{a(x)}{b(x)} = \frac{a_1x^m + a_2x^{m-1} + \dots + a_{m+1}}{b_1x^n + b_2x^{n-1} + \dots + b_{n+1}}$$

Contoh

$$\frac{a(x)}{b(x)} = \frac{2x^2 + x + 4}{x^3 - 3x - 1}$$

Turunan dari polynomial tersebut dicari dengan cara

- $a=[2 \ 1 \ 4]$

- $b=[1 \ 0 \ -3 \ -1]$

- $[p,q]=\text{polyder}(a,b)$

p adalah pembilang & q penyebut dari turunan

Pencocokan Kurva

Pencocokan kurva berfungsi untuk menemukan suatu kurva halus yang paling mendekati data (paling cocok) tetapi tidak harus melewati setiap point data

Dalam matlab, perintah *polyfit(x,y,n)*, digunakan untuk menemukan koefisien dari polynomial kurva derajat n yang paling cocok dengan data yang diberikan.

Contoh:

Diketahui pasangan data x dan y sbb:

```
>>x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]
```

```
>>y=[-0.447 1.978 3.28 6.16 7.08 7.34 7.66 9.56 9.48 9.30 11.2]
```

```
>>a=polyfit(x,y,n)
```

- Jika dipilih $n=1$ → dinamakan *regresi linear (pendekatan garis lurus terbaik)*
- Jika dipilih $n=2$ → dinamakan *polynomial kuadratis (pencocokan kurva kuadrat terkecil)*
- Jika dipilih $n=3$ → *pencocokan kurva derajat 3*
- Dst

Misal $n = 2$, maka diperoleh

$$a = \begin{matrix} -9.8108 & 20.1293 & -0.0317 \end{matrix}$$

Artinya kurva pencocokan yang diperoleh adalah

$$y = -9.8108x^2 + 20.1293x - 0.0317$$

Setelah diperoleh persamaan kurva yang dicari, selanjutnya kurva tersebut akan dibandingkan dengan point data aslinya dengan cara menggambar grafik kurva beserta data asli dalam satu layar.

- `xi=linspace (x(1),x(end));` %membuat interval mulai dari x(1) sampai x(terakhir)
- `z=polyval(a,xi);` % mengevaluasi nilai polynomial kurva hasil pencocokan pada interval yang dibuat
- `plot(x,y,'-o',xi,z,':')` % menggambar titik-titik data asli dengan simbol 'o' dan menggambar kurva pencocokan

MATERI 6

GRAFIK 2 DIMENSI

Plot

- `plot(Y)` → menggambar garis yang didefinisikan oleh titik Y vs indeksnya (Y bil real)
 Contoh :

```
>>Y = [1 3 5 4 7 0 9 8 2];
>>plot(Y)
```
- `plot(X,Y)` → menggambar garis yang didefinisikan oleh pasangan X vs Y
 Contoh :

```
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
plot(x,y)
```

Menggambar beberapa grafik dalam 1 layar (figure)

Contoh :

- `t = 0:pi/100:2*pi;`
- `y = sin(t);`
- `y2 = sin(t-0.25);`
- `y3 = sin(t-0.5);`
- `plot(t,y,t,y2,t,y3)`

Mengatur Grafik

Warna garis, jenis garis, dan bentuk penandaan grafik dapat diatur dengan format yang telah ditentukan.

- Contoh:

```
>>plot(t,y,'r-')
```

Mengatur bentuk garis (linestyle)

- `LineStyle`(bentuk garis) pada grafik, dapat diatur dengan memilih bentuk seperti pada Tabel 6.1.

Tabel 6.1. macam-macam pengaturan bentuk garis

Symbol	Line Style
' - '	Solid line (default)
' -- '	Dashed line
' : '	Dotted line
' - . '	Dash-dot line
' none '	No line

Mengatur tebal garis (LineWidth)

Default `LineWidth` adalah 0.5 points (1 point = 1/72 inch).

Contoh;

- `x = -pi:pi/10:pi;`
- `y = tan(sin(x)) - sin(tan(x));`
- `plot(x,y,'-','LineWidth',2)`

Mengatur bentuk(mark) titik data

Pasangan titik data (x,y) dapat ditandai dengan bentuk-bentuk khusus, seperti dalam tabel sbb:

Marker Specifier	Description
'+'	Plus sign
'o'	Circle
'*'	Asterisk
'.'	Point
'x'	Cross
'square' or 's'	Square
'diamond' or 'd'	Diamond
'^'	Upward-pointing triangle
'v'	Downward-pointing triangle
'>'	Right-pointing triangle
'<'	Left-pointing triangle
'pentagram' or 'p'	Five-pointed star (pentagram)
'hexagram' or 'h'	Six-pointed star (hexagram)
'none'	No marker (default)

Contoh :

```
t = 0:pi/10:2*pi;
y = sin(t);
y2 = sin(t-0.25);
y3 = sin(t-0.5);
>> plot(t,y,'<-',t,y2,'-',t,y3,':')
>> plot(t,y,'p-',t,y2,'-',t,y3,':')
>> plot(t,y,'p',t,y2,'-',t,y3,':')
```

WARNA GARIS

Warna garis dapat diatur dengan menambahkan pilihan warna, yaitu :

- r : merah
- g : hijau
- b : biru
- y : kuning
- k : hitam
- w : putih
- c : cyan
- m : magenta

Contoh :

```
plot(t,y,'r-',t,y2,'g-*',t,y3,'c:')
• x = -pi:pi/10:pi;
• y = tan(sin(x)) - sin(tan(x));
• plot(x,y,'--rp','LineWidth',2,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','g',...
'MarkerSize',10)
```

Mengatur Nilai pada sumbu

- x = -pi:1:pi;
- y = sin(x);
- plot(x,y)
- set(gca,'XTick',-pi:pi/2:pi)
- set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})

Menambahkan Label & Judul grafik

- xlabel → menambahkan label untuk sumbu x
- ylabel → menambahkan label untuk sumbu y
- title → menambahkan judul
- Contoh :
xlabel('Sudut')
ylabel('Nilai')
title('Grafik Fungsi Sinus')

Menambahkan legend pada grafik

- Contoh :
- $x = -\pi:\pi/20:\pi$;
- `plot(x,cos(x),'-ro',x,sin(x),'-.b',x,tan(x))`
- `h = legend('cos','sin','tangen');`

Menggambar beberapa grafik dalam windows yang berbeda

Contoh :

```
x = -pi:pi/20:pi;
plot(x,cos(x),'-ro')
title('Grafik Cosinus')
figure
plot(x,sin(x),'-.b')
title('Grafik Sinus')
```

Menggambar beberapa grafik dalam windows yang sama

```
>> x = -pi:pi/20:pi;
>> subplot(2,2,1);
>> plot(x,cos(x),'-ro')
>> title('Grafik Cosinus')
>> subplot(2,2,2);
>> plot(x,sin(x),'-.b')
>> title('Grafik Sinus')
>> subplot(2,2,3);
>> plot(x,tan(x),'g')
>> title('Grafik Tangen')
>> subplot(2,2,4);
>> plot(x,atan(x),'-*m')
>> title('Grafik Arc Tangen')
```

- grid → memunculkan grid line pada gambar (on dan off)
- grid on → memunculkan grid line pada gambar
- grid off → menghilangkan grid line pada gambar

Menambahkan teks pada grafik

Sintaks : `text(x,y,'string')`

Contoh :

```
plot(0:pi/20:2*pi,sin(0:pi/20:2*pi))
```



```
text(0,0,' sin(\pi)','FontSize',10)
text(pi,0,'\leftarrow sin(\pi)','FontSize',18)
```

Penulisan karakter simbol di matlab

Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
\alpha	α	\upsilon	υ	\sim	\sim
\beta	β	\phi	ϕ	\leq	\leq
\gamma	γ	\chi	χ	\infty	∞
\delta	δ	\psi	ψ	\clubsuit	\clubsuit
\epsilon	ϵ	\omega	ω	\diamondsuit	\diamondsuit
\zeta	ζ	\Gamma	Γ	\heartsuit	\heartsuit
\eta	η	\Delta	Δ	\spadesuit	\spadesuit
\theta	θ	\Theta	Θ	\leftrightarrow	\leftrightarrow
\vartheta	ϑ	\Lambda	Λ	\leftarrow	\leftarrow
\iota	ι	\Xi	Ξ	\uparrow	\uparrow
\kappa	κ	\Pi	Π	\rightarrow	\rightarrow
\lambda	λ	\Sigma	Σ	\downarrow	\downarrow
\mu	μ	\Upsilon	Υ	\circ	\circ
\nu	ν	\Phi	Φ	\pm	\pm
\xi	ξ	\Psi	Ψ	\geq	\geq
\pi	π	\Omega	Ω	\propto	\propto
\rho	ρ	\forall	\forall	\partial	∂
\sigma	σ	\exists	\exists	\bullet	\bullet
\varsigma	ς	\ni	\ni	\div	\div
\tau	τ	\cong	\cong	\neq	\neq
\equiv	\equiv	\approx	\approx	\aleph	\aleph
\Im	\Im	\Re	\Re	\wp	\wp
\otimes	\otimes	\oplus	\oplus	\oslash	\oslash
\cap	\cap	\cup	\cup	\supseteq	\supseteq
\supset	\supset	\subset	\subset	\subset	\subset
\int	\int	\in	\in	\circ	\circ
\rfloor	\rfloor	\lceil	\lceil	\nabla	∇
\lfloor	\lfloor	\cdot	\cdot	\ldots	\dots
\perp	\perp	\neg	\neg	\prime	\prime
\wedge	\wedge	\times	\times	\emptyset	\emptyset
\rceil	\rceil	\surd	\surd	\mid	\mid
\vee	\vee	\varpi	ϖ	\copyright	\copyright
\langle	\langle	\rangle	\rangle		

- Contoh :

```
text(0.4,5,'e^{i\omega\tau} = cos(\omega\tau) + i sin(\omega\tau)')
```

Mengatur teks dengan mouse

- Sintaks : gtext('string1', 'string2', ...)

Contoh :

```
x = -pi:1:pi;
y = sin(x);
```

```
plot(x,y)
gtext('sin(\pi)')
fill → memberi isian warna (fill warna) pada grafik
fill(x,y,'c')
```

Stairs

Untuk menggambar grafik tangga

Contoh :

```
x = linspace(-2*pi,2*pi,40);
stairs(x,sin(x))
```

bar, barh

Untuk menggambar grafik batang

Sintaks :

- bar(Y)
- bar(x,Y)
- bar(...,width)
- bar(...,'style') → 'stack', 'group'
- bar(...,'bar_color') → 'r','g', ...

contoh

```
Y = round(rand(5,3)*10);
subplot(2,2,1)
bar(Y,'group')
title('Group')
subplot(2,2,2)
bar(Y,'stack')
title('Stack')
subplot(2,2,3)
barh(Y,'stack')
title('Stack')
subplot(2,2,4)
bar(Y,1.5)
title('Width = 1.5')
```

Pie

Menggambar pie chart

Sintaks :

```
pie(X)
pie(X,explode)
pie(...,labels)
```

Contoh

- x = [1 3 0.5 2.5 2]; pie(x)
- x = [1 3 0.5 2.5 2];
- explode=[0 1 0 0 0];
- pie(x, explode)
- x = [1 3 0.5 2.5 2];
- explode=[0 1 0 0 0];
- pie(x, explode, {'baju','makanan','minuman','buku','pensil'})

MATERI 7

Minimum dan maksimum fungsi

ezplot

1. fungsi eksplisit $f=f(x)$

Contoh fungsi eksplisit :

$$y=\cos(x) \qquad f(x)=\frac{\sin(x)}{1+x^2}$$

A. ezplot(f):

menggambar fungsi $f = f(x)$ dalam domain $-2*\pi < x < 2*\pi$

contoh:

```
>>ezplot('cos(x)')
>>ezplot('sin(x)/(1+x^2)')
```

B. ezplot(f, [a,b]):

menggambar $f = f(x)$ dalam interval $a < x < b$

contoh: >> ezplot('cos(x)', [0, pi])

C. ezplot(f, [xmin, xmax,ymin,ymax]):

menggambar $f = f(x)$ dalam interval $xmin < x < xmax$ dan $ymin < y < ymax$

contoh:

```
>> ezplot('sin(x)/(1+x^2)', [-4, 4,-0.5,0.5])
```

2. fungsi implisit $f=f(x,y)$

Contoh fungsi implisit:

$$x^2 - y^2 = 1 \qquad \frac{1}{y} - \log(y) + \log(-1 + y) + x = 1$$

A. ezplot(f):

menggambar $f(x,y) = 0$ dalam domain $-2*\pi < x < 2*\pi$ dan $-2*\pi < y < 2*\pi$ contoh:

```
>> ezplot('x^2-y^2-1')
```

B. ezplot(f, [a,b]):

menggambar $f(x,y) = 0$ dalam interval

$a < x < b$ and $a < y < b$

contoh:

```
>> ezplot('x^3 + y^3 - 5*x*y + 1/5',[-3,3])
```

C. ezplot(f, [xmin, xmax,ymin,ymax]):

menggambar $f(x,y)=0$ dalam interval

$xmin < x < xmax$ dan $ymin < y < ymax$

contoh:

```
>> ezplot('x^3 + y^3 - 5*x*y + 1/5',[-3,3,-5,5])
```

3. fungsi parametrik

Contoh fungsi parametrik

$$x = \sin(t)$$

$$y = \cos(t)$$

$$x^2 + y^2 = \sin^2 t + \cos^2 t = 1$$

A. ezplot(x,y):

menggambar kurva parametric $x = x(t)$ dan $y = y(t)$ dalam domain $0 < t < 2\pi$ contoh:

```
>> ezplot('sin(t)', 'cos(t)')
```

B. ezplot(x,y):

menggambar kurva parametric $x = x(t)$ dan $y = y(t)$ dalam domain $0 < t < 2\pi$ contoh:

```
>> ezplot('sin(t)', 'cos(t)')
```

Fplot

1. fplot(f,[a,b]) :

menggambar grafik f dalam selang interval $a < x < b$

Contoh: `fplot('sin(1 ./ x)', [0.01 0.1])`

2. fplot(f,[xmin,xmax,ymin,ymax]) :

menggambar grafik f dalam selang interval $x_{min} < x < x_{max}$ dan $y_{min} < y < y_{max}$

Contoh:

```
fplot(['tan(x),sin(x),cos(x)'],2*pi*[-1 1 -1 1])
```

Contoh di atas juga untuk menggambar lebih dari 1 grafik fungsi dalam 1 perintah

Minimum fungsi

Sebuah fungsi suku banyak mempunyai titik-titik ekstrim, di antaranya adalah titik balik maksimum dan titik balik minimum.

MATLAB mempunyai perintah **fminbnd** untuk mencari absis dari titik paling minimum dari sebuah grafik fungsi.

Berikut ini adalah langkah-langkah untuk mencari titik balik minimum dari sebuah fungsi suku banyak.

Contoh:

Tentukan titik balik minimum dari fungsi $f(x) = \frac{1}{3}x^3 - 2x^2 + 3x + 4$

Langkah 1: mendefinisikan fungsi

Tanpa menggunakan ekspresi simbolik, sebuah fungsi dapat didefinisikan dengan menggunakan perintah **inline**.

```
>>f= inline('1/3*x^3-2*x^2+3*x+4')
```

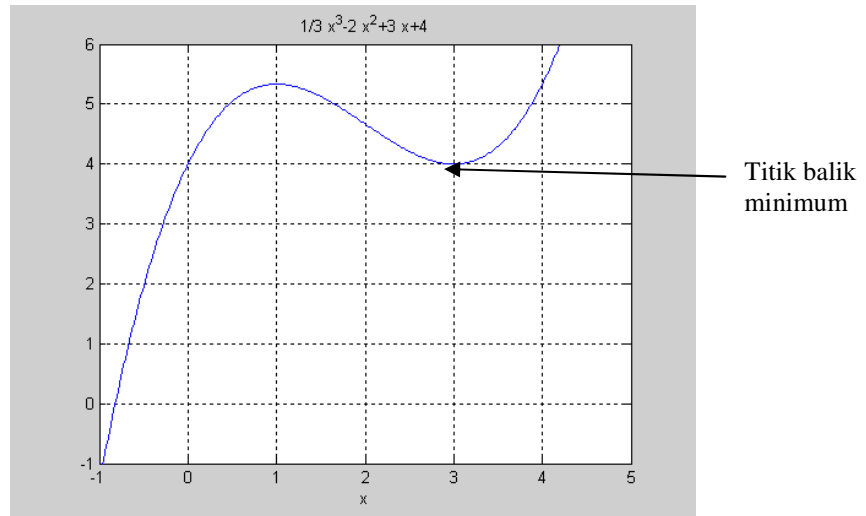
Langkah 2: menentukan interval letak titik balik minimum

Untuk menentukan interval titik balik minimum, terlebih dahulu perlu digambar grafiknya.

```
>>ezplot(f),grid
```

Jika grafik kurang jelas, intervalnya dapat diatur sbb

```
>>ezplot(f,[-1,5,-1,6])
```



Dari grafik tersebut tampak bahwa interval titik balik minimum berada pada interval (2,4).

Langkah 3: menentukan absis titik balik minimum dengan perintah `fminbnd`

Sintaks perintah: `fminbnd(f,x1,x2)`

dengan :

`f` adalah nama fungsi yang didefinisikan dengan perintah **inline** di langkah 1

`x1,x2` adalah nilai pada interval $(x1,x2)$, letak titik balik minimum terjadi.

Maka perintahnya adalah:

```
>> xmin =fminbnd(f,2,4)
```

```
xmin =
```

```
3.0000
```

Langkah 4: menentukan ordinat titik balik minimum

Ordinat titik balik minimum ditentukan dengan cara memasukkan nilai absis pada langkah 3 (`xmin`) ke dalam fungsi.

```
>> ymin =f(xmin)
```

```
ymin =
```

```
4.0000
```

Berdasarkan hasil pada langkah 3 dan 4, maka diperoleh titik balik minimum fungsi $f(x) = \frac{1}{3}x^3 - 2x^2 + 3x + 4$ adalah di (3,4).

Maksimum fungsi

Perintah **fminbnd** juga dapat dimanfaatkan untuk mencari titik balik maksimum. Berikut ini contoh langkah-langkah mencari titik balik maksimum suatu fungsi.

Contoh:

Tentukan titik balik maksimum dari fungsi $f(x) = \frac{1}{3}x^3 - 2x^2 + 3x + 4$

Langkah 1: mendefinisikan fungsi cerminan dari fungsi yang diketahui

Karena perintah **fminbnd** hanya berfungsi untuk mencari titik minimum, maka fungsi yang ada harus dicerminkan terhadap sumbu x (dibalik), sehingga titik balik maksimum akan menjadi titik balik minimum.

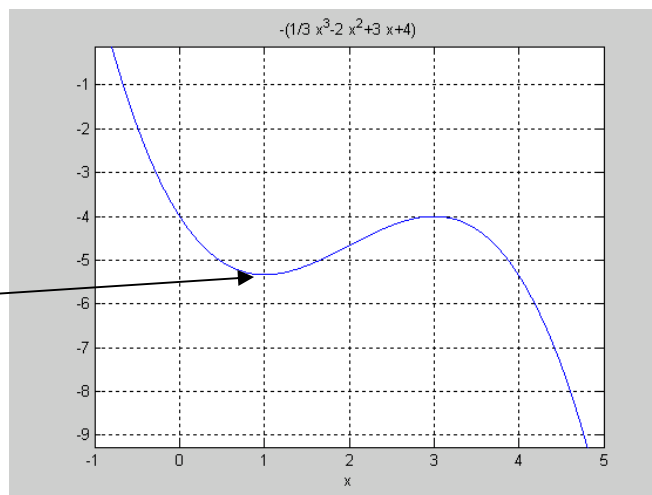
Fungsi cerminan didefinisikan dengan cara memberi tanda – (**minus**) di depan fungsi asal.

```
>>f1= inline('-(1/3*x^3-2*x^2+3*x+4)')
```

Langkah 2: menentukan interval titik balik maksimum

```
>>ezplot(f1,[-1,5])
```

Titik balik maksimum yang dicerminkan sehingga menjadi titik balik minimum



Titik balik maksimum pada grafik tersebut berada pada interval (0,2).

Langkah 3: menentukan absis titik balik maksimum dengan perintah **fminbnd**

```
>> xmax =fminbnd(f1,0,2)
```

```
xmax =
1.0000
```

Langkah 4: menentukan ordinat titik balik maksimum

Ordinat titik balik minimum ditentukan dengan cara memasukkan nilai xmax ke dalam fungsi semula (f), **bukan ke dalam fungsi cerminan (f1)**.

```
>> ymax=f(xmax)
ymax =
5.3333
```

Berdasarkan hasil pada langkah 3 dan 4, maka diperoleh titik balik maksimum fungsi $f(x) = \frac{1}{3}x^3 - 2x^2 + 3x + 4$ adalah di (1, 5.33).

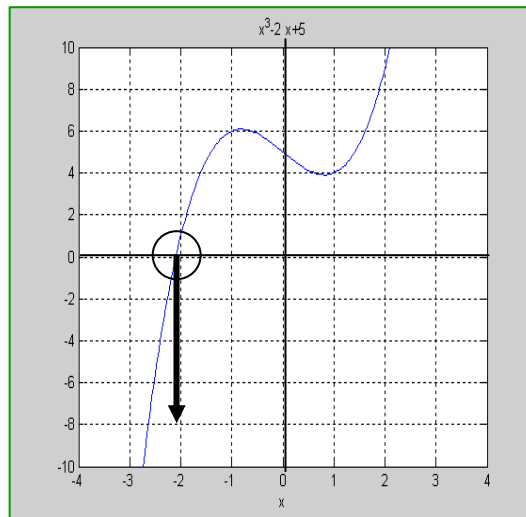
Latihan:

Carilah titik balik minimum dan maksimum dari fungsi $f(x) = 4x - x^3$.

Gunakan interval (-3 3) untuk menggambar grafik, agar diperoleh hasil yang jelas.

Titik nol

```
>>f= inline('x^3-2*x+5')
>>ezplot(f)
```



Dengan melihat figure diketahui grafik memotong sumbu x di sekitar $x=-2$, maka:

```
>> tnol=fzero(f,-2)
>> ynol = f(tnol) % mengecek nilai tnol
```

MATERI 8

Grafik 3D

Plot3

- Plot3 berfungsi untuk menggambar grafik garis 3 Dimensi

- Sintak umum

```
Plot3(x1,y1,z1,S1,x2,y2,z2,S2,...)
```

- Keterangan:

x_n, y_n, z_n : vector atau matriks

S_n : karakter string, bersifat opsional, untuk mengatur warna, symbol, style garis

- Contoh :

```
»t=0:pi/50:10*pi;
```

```
» plot3(sin(t),cos(t),t,'-o')
```

```
» title('Helix'),xlabel('sin(t)'),ylabel('cos(t)'),zlabel('t')
```

```
» text(0,0,0,'titik asal')
```

sintak perintah menambahkan text : `text(x,y,z,'string')`

mengatur axis : `axis(xmin,xmax,ymin,ymax,zmin,zmax).`

Merubah sudut pandang:

Grafik 3D mempunyai sudut pandang, yang disebut dgn azimuth dan elevasi. Nilai default azimuth=-37.5o dan elevasi= 30o.

Perintah untuk merubah sudut pandang adalah `view(az,el)`

Contoh :

menggambar 3 buah grafik garis 3D dalam satu figure

```
» x=linspace(0,3*pi);
```

```
» z1=sin(x);
```

```
» z2=sin(2*x);
```

```
» z3=sin(3*x);
```

```
» y1=zeros(size(x));
```

```
» y2=ones(size(x));
```

```
» y3=y2/2;
```

```
» plot3(x,y1,z1,x,y2,z2,x,y3,z3)
```

contoh perubahan sudut pandang:

```
» subplot(2,2,1);
```

```
» plot3(x,y1,z1,x,y2,z2,x,y3,z3)
```

```
» title('default, Az=-37.5 El=30')
```

```
» view(-37.5,30)
```

```
» subplot(2,2,2);
```

```
» plot3(x,y1,z1,x,y2,z2,x,y3,z3)
```

```
» title('dirotasi ke 52.5')
```

```
» view(-37.5+90,30)
```

```
» subplot(2,2,3);
```

```
» plot3(x,y1,z1,x,y2,z2,x,y3,z3)
```

```
» title('elevasi menjadi 60')
```



```

» view(-37.5,60)
» subplot(2,2,4)
» plot3(x,y1,z1,x,y2,z2,x,y3,z3)
» title('Az=0 El=90')
» view(0,90)
» grid

```

Beberapa fungsi view:

- view(2) mengeset sudut pandang ke default 2 dimensi, az=0, el=90
- view(3) mengeset sudut pandang ke default 3 dimensi, az=-37.5, el=30
- [az,el]=view Memperoleh nilai azimuth dan elevasi dari grafik yang ada

Mesh

Perintah Mesh berfungsi untuk membuat grafik dari fungsi dengan dua variabel $z=f(x,y)$. Hasilnya berupa grafik yang tampilannya seperti jala.

Contoh:

```

» x=-7.5:0.5:7.5;
» y=x;
» [X,Y]=meshgrid(x,y);
» R=sqrt(X.^2+Y.^2)+eps;
» Z=sin(R)./R;
» mesh(X,Y,Z)

```

Melihat bagian grafik yang tersembunyi:

- Perintah hidden off akan membuat bagian grafik yang tersembunyi menjadi terlihat.

Cobalah perintah hidden off di bawah perintah mesh(X,Y,Z). Untuk menyembunyikan lagi, ketikkan hidden on

- meshgrid berfungsi untuk membuat matriks X yang baris-barisnya duplikat dari matriks x, dan Y yang kolom-kolomnya duplikat matriks y. Lebih jelasnya, perhatikan hasil dari contoh berikut:

```

» x=2:0.5:5
» y=x*2
» [X,Y]=meshgrid(x,y)

```

Variasi perintah mesh adalah meshc dan meshz, sintaknya sama dengan sintak mesh

Waterfall

- Perintah waterfall menghasilkan grafik yang sama dengan perintah mesh, tetapi garis-garis jala hanya tampak dari arah sumbu x.
- Contoh


```
» waterfall(X,Y,Z)
```

Surf

- Perintah surf berfungsi untuk menggambar grafik bidang permukaan. Sintaknya sama dengan perintah mesh.
- Contoh :


```
surf(X,Y,Z)
```

Ada dua variasi perintah surf, yaitu surfc dan surf1

Contour, Pcolor dan Quiver

Pcolor merupakan ekuivalensi 2 dimensi dari surf.

Contour menggambar grafik garis yang sesuai dengan ketinggian bidang

Quiver memberi arah panah, banyak digunakan untuk menggambar medan magnet

Shading interp; mengubah bayangan/warna menjadi interpolasi

Contoh :

```
» [X,Y,Z]=peaks(30);
» [X,Y,Z]=peaks(30);
» surf(X,Y,Z)
» pcolor(X,Y,Z)
» shading interp
» hold on
» contour(X,Y,Z,19,'k') % gambar 19 garis contour warna 'k' (hitam)
» [dy,dx]=gradient(Z,0.5,0.5);
» quiver(X,Y,dx,dy)
» cs=contour(X,Y,Z); %mengambil angka sesuai contour/ketinggian
» clabel(cs) % memberi label angka sesuai ketinggian
```

MATERI 9

Fungsi dan Integral

Beberapa fungsi '**Fungsi**' dalam MATLAB disajikan dalam tabel berikut:

Category	Function	Description
Plotting	fplot	Plot function.
Optimization and zero finding	fminbnd	Minimize function of one variable with bound constraints.
	fminsearch	Minimize function of several variables.
	fzero	Find zero of function of one variable.
Numerical Integration	quad	Numerically evaluate integral, adaptive Simpson quadrature.
	quadl	Numerically evaluate integral, adaptive Lobatto quadrature.
	dblquad	Numerically evaluate double integral.

Penyajian Fungsi (Functions) dalam MATLAB

MATLAB dapat menyajikan fungsi matematika dengan menggunakan **inline** dan **M-files**.

a. Sebagai obyek inline

Sebagai **obyek inline**, sudah kita lakukan beberapa waktu yang lalu.

Contoh:

$$f(x) = \frac{1}{(x-0.3)^2 + 0.01} + \frac{1}{(x-0.9)^2 + 0.04} - 6$$

Dengan menggunakan inline, fungsi tersebut didefinisikan sbb:

```
f = inline('1./((x-0.3).^2 + 0.01) + 1./((x-0.9).^2 + 0.04) - 6');
```

Mengevaluasi f di 2.0

```
>>f(2.0)
ans =
    -4.8552
```

Dengan inline, anda juga dapat membuat fungsi yang lebih dari 1 argumen dengan menpesifikasikan nama-nama argumennya. Contoh, fungsi berikut mempunyai 2 buah argument input, yaitu x dan y:

```
>> f= inline('y*sin(x)+x*cos(y)', 'x', 'y')
>> f(pi,2*pi)
ans =
    3.1416
```

b. Sebagai MATLAB Functions

Fungsi ini dibuat melalui menu **File → New → M-File**.

Kemudian tulis fungsi di bawah ini di layer MATLAB editor.

```
function y = fsi(x) %nama fungsinya fsi
y = 1./((x - 0.3).^2 + 0.01) + 1./((x - 0.9).^2 + 0.04) - 6;
```

Simpan dengan nama **fsi.m**, caranya pilih menu **File→save As→beri nama fsi.m→save**

Catatan penting:

- ♪ Nama file harus sama dengan nama fungsi
- ♪ Jangan menggunakan nama fungsi dengan nama-nama fungsi bawaan yang sudah ada dalam MATLAB, misal **sin**, **log**, **exp**, **polyfit**, **fzero**, **fmin**, dll.
Hal tersebut akan merusak fungsi bawaan MATLAB, dan akan merugikan anda sendiri! (pada saat dipanggil, akan muncul pesan 'Attempt to execute SCRIPT as a function')
- ♪ Gunakan operasi elemen (menggunakan titik untuk perkalian dan pembagian, pangkat)
- ♪ Sebaiknya akhiri definisi rumus dengan tanda ; (titik koma) agar hasilnya tiak ditampilkan.

Mengevaluasi fungsi **fsi.m** pada 2.0

Cara 1	Cara 2
<pre>fh = @fsi; feval(fh,2.0) ans = -4.8552</pre>	<pre>feval('fsi',2.0) ans = -4.8552</pre>

Catatan:

untuk menggunakan fungsi yang didefinisikan dengan menggunakan M-File, maka nama fungsi harus diapit dengan tanda petik '. Lihat cara 2 di atas. Sedangkan yang didefinisikan dengan inline, langsung dipanggil nama fungsinya tanpa diapit tanda '.

Integrasi Numerik

Integral digunakan untuk mencari luas area di bawah suatu fungsi.

Misal diketahui fungsi yang telah didefinisikan sbg **fsi.m** di atas. Luas area di bawah fungsi tsb dalam interval $0 < x < 1$ dapat dicari dengan cara sbb:

```
>> x=0:0.07:1; %selisih data ditentukan sendiri, makin kecil
makin bagus
>> y=fsi(x); %memanggil fungsi fsi.m
>> area=trapz(x,y)
area =
29.4611
```

Selain itu, MATLAB menyediakan fungsi-fungsi berikut untuk mencari integral sebuah fungsi:

quad Use adaptive Simpson quadrature
quadl Use adaptive Lobatto quadrature
dblquad Numerically evaluate double integral

Contoh: Integral fungsi **fsi.m** dari 0 to 1, dicari dengan cara sbb:

```
q = quad(@fsi,0,1) %tanda @ digunakan untuk memanggil fungsi
fsi.m
q =
29.8583
```

Misal akan dicari: $\int_0^2 \frac{1}{x^3 - 2x - 5}$

Ada 3 cara yang dapat dilakukan:

Cara 1:

```
Q = quad('1./(x.^3-2*x-5)',0,2);
```

Cara 2; dengan inline:

```
F = inline('1./(x.^3-2*x-5)');
```

```
Q = quad(F, 0, 2);
Cara 3; dengan fungsi M-File
Q = quad(@myfun, 0, 2);
dengan myfun.m is an M-file.
```

```
function y = myfun(x)
y = 1./(x.^3-2*x-5);
```

Anda dapat melakukan cara yang sama untuk perintah **quadl** (bukan *quad* 'satu', tetapi *quad* 'L'), atau **quad8**.

Example: Menghitung panjang Kurva

Anda dapat menggunakan *quad* atau *quadl* untuk menghitung panjang kurva. Misal diketahui kurva parameter sbb:

$$x(t) = \sin(2t), \quad y(t) = \cos(t) \quad z(t) = t$$

where $t \in [0, 3\pi]$

Grafik plot 3 kurva tsb adalah sbb:

```
t = 0:0.1:3*pi;
plot3(sin(2*t), cos(t), t)
```

Rumus panjang busur menyebutkan bahwa panjang kurva adalah integral dari akar kuadrat turunan persamaan parameternya:

Fungsi dengan nama **hcurve.m**

```
function f = hcurve(t)
f = sqrt(4*cos(2*t).^2 + sin(t).^2 + 1);
```

Integral fungsi tsb dicari dengan cara:

```
len = quad(@hcurve, 0, 3*pi)
len =
    1.7222e+01
```

Maka panjang kurva tsb adalah 17.2.

Integral ganda

Rumus umum:

$$\int_{y \min}^{y \max} \int_{x \min}^{x \max} f(x, y) dx dy$$

For this example $f(x, y) = y \sin(x) + x \cos(y)$.

$$\text{atau} \int_0^{\pi} \int_{\pi}^{2\pi} y \sin(x) + x \cos(y)$$

dicari dengan cara:

```
Q = dblquad(inline('y*sin(x)+x*cos(y)'), pi, 2*pi, 0, pi)
atau
```

```
Q = dblquad(@integrnd, pi, 2*pi, 0, pi)
```

Dengan **integrnd.m** adalah fungsi M-file.

```
function z = integrnd(x, y)
z = y*sin(x)+x*cos(y);
```

fungsi **integrnd.m** mengintegrasikan $y*\sin(x)+x*\cos(y)$ dalam $\pi \leq x \leq 2*\pi$, $0 \leq y \leq \pi$.

Tugas:

Carilah contoh-contoh soal yang menggunakan integral dan integral ganda. Kerjakan soal-soal tsb dengan MATLAB.

Kerjakan berkelompok, dengan cacah soal sebanyak 2 x anggota kelompok.

Tulis soal beserta penyelesaiannya. Sangat dianjurkan soal dalam bentuk soal cerita.

Kumpulkan saat praktikum minggu ini

MATERI 10

EKSPRESI SIMBOLIK

- Objek symbolic adalah struktur data yang menyimpan representasi string dari simbol.
- Objek symbolic digunakan untuk merepresentasikan variabel, ekspresi, dan matriks symbolic.

Matlab menyediakan Symbolic Math Toolbox untuk melakukan operasi dengan symbol-simbol matematis tanpa bilangan, dengan catatan, simbol-simbol yang akan digunakan dalam operasi harus didefinisikan terlebih dahulu.

Membentuk Variabel dan ekspresi Symbolic

Contoh :

<pre>>>syms x >>y=sin(x) >>syms a b c x >>f = sym('a*x^2 + b*x + c') >>syms x y z >>r = sqrt(x^2 + y^2 + z^2)</pre>	<pre>» syms a b c d » M=[a b;c d] %matriks M dengan elemen a b c d » size (M) » det(M)</pre>
---	--

Perintah subs

•Contoh

```
>>f = sym('a*x^2 + b*x + c') >>subs(f,a,2) %mengganti nilai a dengan 2
>>subs(f,x,3) %Mengganti nilai x dengan 3
```

Mengekstrak pembilang dan penyebut: dengan perintah numden

Misalnya diketahui persamaan $f = \frac{ax^2}{b-x}$ ekspresi simboliknya adalah : `>> f=a*x^2/(b-x)`

```
» [n,d]=numden(f) %mengetahui pembilang dan penyebut dari f
```

Operasi aljabar .

Lakukan operasi berikut untuk f1 dan g $f1 = 2x^2 + x - 5$ $g = x^2 - x + 7$

```
» f1+g
» f1-g
» f1/g
» f1*g
```

Diketahui fungsi h dan k: $h = \frac{x}{1+u^2}$ $k = \cos(x+v)$

Dapat dicari bundaran (o) fungsi

```
» compose (h,k) % mencari (h o k) (x)
» compose (h,k,u,v) % mencari (h o k) (u)
» compose (k,h,v,u) % mencari (k o h) (v)
```

Differential (Turunan) fungsi satu variabel

$$f = \sin(5x) \quad g = e^{\cos(x)}$$

```
>>syms x
>>f = sin(5*x)
>>g = exp(x)*cos(x)
>>diff(f)
>>diff(g)
```

Turunan kedua

```
>>diff(f,2)
>>diff(g,2)
```

Turunan ketiga

```
>>diff(g,3)
```

•syms a b x n t theta

f	diff(f)
x^n	x^{n-1}
$\sin(a*t+b)$	$\cos(a*t+b) * a$
$\exp(i*theta)$	$i*\exp(i*theta)$

Untuk menyederhanakan bentuk fungsinya digunakan perintah 'simplify'

•Contoh :

```
simplify(diff(x^n))
```

Turunan fungsi rasional

$$f(x) = \frac{3x^2 + 6x - 1}{x^2 + x - 3}$$

```
•syms x
•f = (3*x^2 + 6*x - 1)/(x^2 + x - 3)
•f1 = diff(f)
•f1=simplify(f1) %menyederhanakan bentuk
•pretty(f1)
```

Turunan) fungsi beberapa variabel

Contoh :

$$f = \sin(st)$$

```
>>syms s t
>>f=sin(s*t)
>>diff(f,t) >>diff(f,s) >>diff(f,t,2)
```

Contoh dalam bentuk Matriks

```
>>syms a x
>>A = [cos(a*x),sin(a*x);-sin(a*x),cos(a*x)]
>>diff(A)
```


Limit

Operasi Matematika	Perintah dalam MATLAB
$\lim_{x \rightarrow 0} f(x)$	limit(f)
$\lim_{x \rightarrow a} f(x)$	limit(f,x,a) atau limit(f,a)
$\lim_{x \rightarrow a^-} f(x)$	limit(f,x,a,'left')
$\lim_{x \rightarrow a^+} f(x)$	limit(f,x,a,'right')

Contoh :

$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$ <p>Jika diambil $f(x) = \cos(x)$</p> <pre>>>syms h n x >>limit((cos(x+h) - cos(x))/h,h,0)</pre> $\lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n$ <pre>>>limit((1 + x/n)^n,n,inf)</pre>	$\lim_{x \rightarrow 0^-} \frac{x}{ x } \longrightarrow \text{>>limit}(x/abs(x),x,0,'left')$ $\lim_{x \rightarrow 0^+} \frac{x}{ x } \longrightarrow \text{>>limit}(x/abs(x),x,0,'right')$
--	---

Integral

Mathematical Operation	Perintah dalam MATLAB
$\int x^n dx = \frac{x^{n+1}}{n+1}$	<pre>syms x n int(x^n) atau int(x^n,x)</pre>
$\int_0^{\pi/2} \sin(2x) dx = 1$	<pre>syms x int(sin(2*x),0,pi/2) Atau int(sin(2*x),x,0,pi/2)</pre>
$g = \cos(at+b)$	<pre>syms a b t g = cos(a*t + b); int(g) atau int(a,t)</pre>

Penjumlahan untuk deret Tak Hingga

Contoh

$1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$ <pre>>>syms x k >>s1 = symsum(1/k^2,1,inf)</pre>	$1 + x + x^2 + \dots$ <pre>>>s2 = symsum(x^k,k,0,inf)</pre>
--	---

Collect

Mengelompokkan variable yang berderajat sama.

Contoh :

f	collect(f)
$(x-1) * (x-2) * (x-3)$	$x^3-6*x^2+11*x-6$
$x * (x * (x-6) + 11) - 6$	$x^3-6*x^2+11*x-6$
$(1+x) * t + x * t$	$2 * x * t + t$

Expand

Menjabarkan semua perkalian dan penjumlahan

Contoh:

f	expand(f)
$a * (x + y)$	$a * x + a * y$
$(x-1) * (x-2) * (x-3)$	$x^3-6*x^2+11*x-6$
$x * (x * (x-6) + 11) - 6$	$x^3-6*x^2+11*x-6$
$\exp(a+b)$	$\exp(a) * \exp(b)$
$\cos(x+y)$	$\cos(x) * \cos(y) - \sin(x) * \sin(y)$
$\cos(3 * \arccos(x))$	$4 * x^3 - 3 * x$

Factor

Menyatakan suatu polinomial sebagai perkalian polinomial dengan derajat lebih rendah

Contoh :

f	factor(f)
$x^3-6*x^2+11*x-6$	$(x-1) * (x-2) * (x-3)$
$x^3-6*x^2+11*x-5$	$x^3-6*x^2+11*x-5$
x^6+1	$(x^2+1) * (x^4-x^2+1)$

solve

Berfungsi menyelesaikan sebuah persamaan

```
>>syms a b c x
```

```
>>S = a*x^2 + b*x + c;
```

```
>>solve(S) →mencari penyelesaian S=0
```

```
ans =
```

```
[1/2/a*(-b+(b^2-4*a*c)^(1/2))]
```

```
[1/2/a*(-b-(b^2-4*a*c)^(1/2))]
```

Jika ruas kanan tdk 0, misal $a*x^2 + b*x = c$

Maka

```
>>solve('a*x^2 + b*x =c')
```

Diketahui SPL:

$$x + y - z = 1$$

$$-2x - 6y + 4z = -2$$

$$-x - 3y + 3z = 1$$

diselesaikan dengan MATLAB

```
>>syms x y z
```

```
>>[x,y,z]=solve(x+y-z-1,-2*x-6*y+4*z+2,-x-3*y+3*z-1)
```

MATERI 11

Script M-file

Script berupa sederetan perintah dalam suatu teks file, yang ditulis dalam editor m file, sehingga nama Script akan diakhiri dengan ekstensi '.m'.

Contoh:

Untuk mencari luas persegi panjang, urutan perintah yang diketikkan lewat **command window** adalah sbb:

```
» p=16;  
» l=11;  
» luas=p*l
```

Sederetan perintah tersebut dapat dituliskan dalam **editor m-file**, dengan cara sbb:

1. klik menu **File – New – M-file**
2. Dalam editor m-file ketikkan sbb:

```
p=16;  
l=11;  
luas=p*l
```

3. Klik menu File – save as. Beri nama **contoh1**

Untuk menjalankannya, kembali ke command window, lalu panggil nama file contoh1.

```
» contoh1
```

Latihan :

Buatlah script m-file untuk menghitung tinggi menara, jika diketahui jarak orang ke menara adalah 500 m, tinggi orang 170cm, dan sudut elevasi = 75° . Simpan dengan nama Menara.m

Memberi komentar

Komentar dalam program diawali dengan tanda %. Buka kembali contoh1.m dan edit sehingga menjadi spt berikut:

Contoh:

```
% contoh script untuk mencari luas persegi panjang  
p=16; % panjang  
l=11; % lebar  
luas=p*l
```

Menggunakan perintah input

Pada contoh di atas, untuk memasukkan nilai p dan l yang bervariasi kita dapat menggunakan perintah **input**.

- Bukalah file **contoh1.m**
- Edit perintah-perintahnya sehingga menjadi:

```
% contoh script untuk mencari luas persegi panjang  
p=input(' Berapa panjangnya = ')  
l=input(' Berapa lebarnya = ')  
luas=p*l
```
- Simpan kembali dengan memilih menu File-save.
- Jalankan kembali lewat **command window**.

Latihan:

1. Edit file Menara.m, dan beri perintah input untuk memasukkan nilai-nilai yang diperlukan.
2. Buat script m-file untuk mencari akar-akar persamaan kuadrat, dengan masukkan nilai a,b,dan c. Gunakan perintah input dan beri komentar agar program menjadi jelas dibaca.

Menggunakan perintah echo

Perintah echo digunakan untuk menampilkan baris perintah yang sedang dikerjakan.

Contoh :

```
% contoh penggunaan perintah echo
echo on
p=input(' Berapa panjangnya = ')
l=input(' Berapa lebarnya= ')
luas=p*l
echo off
```

Dalam beberapa hal, perintah echo bermanfaat untuk mengetahui baris perintah beserta hasilnya, sehingga mudah melakukan pelacakan jika terjadi kesalahan.

Menggunakan perintah display:

Perintah display atau disp berguna untuk mencetak tulisan di layar

Contoh :

```
p=input(' Berapa panjangnya = ')
l=input(' Berapa lebarnya= ')
luas=p*l;
disp('luas persegi panjang ')
disp(luas)
```

MATERI 12

PEMROGRAMAN: FOR-END & IF-END

Dasar Membuat program

Contoh: program mencari luas segitiga

Sebuah program ada 3 hal penting:

1. harus ada input
contoh ; input: alas dan tinggi
2. proses
rumus luas dihitung : $luas=(alas*tinggi)/2$
3. output
hasil perhitungan ditampilkan dgn perintah disp

FOR-END

Sintaknya :

```
for nama variabel = nilawal:step:nilakhir  
    deret_perintah
```

End

Jika step tdk disebutkan, maka step =1.

Contoh:

```
for n=1:10  
    disp(n+1)  
end;
```

LATIHAN:

Buatlah perintah yang hasilnya sbb:

```
luas lingkaran dgn jari-jari 1 = 3.1416  
luas lingkaran dgn jari-jari 4 = 50.2655  
luas lingkaran dgn jari-jari 7 = 153.938  
luas lingkaran dgn jari-jari 10 = 314.1593
```

for end bisa berada di dalam for –end yang lain.

Contoh

```
for n=1:5  
    for m=1:4  
        A(n,m)=n+m;  
    end;  
end;  
disp(A)
```

IF- END

Sintaknya :

if *ekspresi_logika***Perintah****end**

Perintah di atas akan mengecek nilai ekspresi logika, jika benar perintah akan dikerjakan, jika tidak, maka program akan berhenti

- Contoh ekspresi logika
 $a > 0$; $b >= c$; $luas == 100$; $c \sim = 0$;
- Operator perbandingan:
 $>$, $<$, $>=$, $<=$, $\sim =$, $==$

Kadang juga digunakan or (|), and (&), not (~)

Contoh:

```
if (a>0) | (b>0)
```

Contoh :

```
% contoh script untuk mencari luas segitiga
```

```
a= input(' Berapa alasnya = ')
```

```
if a<0
```

```
disp('nilai a tidak boleh negatif')
```

```
break % keluar dari program
```

```
end
```

```
t= input(' Berapa tingginya= ')
```

```
luas=(a*t)/2
```

MATERI 13

IF –ELSEIF-END & SWITCH CASE

IF –ELSEIF-END

Sintaksnya:

```
if ekspresi_logika
    Perintah1           % Jika benar akan mengerjakan perintah1
elseif ekspresi_logika % Jika salah akan mengecek elseif berikutnya
    perintah2
.....
elseif
    perintah ke-n
end
```

Contoh

```
%script untuk menentukan kelulusan
Nilai=input('Berapa nilainya :')
if Nilai >= 60
    disp('Selamat Anda Lulus')
elseif Nilai<60
    disp('Maaf Anda Gagal')
end
```

LATIHAN :

Buat Script m-file untuk menentukan Nilai mahasiswa dengan ketentuan sbb:

Nilai A : ≥ 85

Nilai B : $70 \leq \text{nilai} < 85$

Nilai C : $60 \leq \text{nilai} < 70$

Nilai D : < 60

Switch case

Sintaknya:

```
switch ekspresi
case {tes_ekspresi1}
    deret_perintah1
case { tes_ekspresi2}
    deret_perintah2
otherwise
    deret_perintah3
end
```

Contoh:

```
disp('Masukkan pilihan anda (1-inf)')
x=input('pilihan ');
```



```
switch x
case{1}
    disp('pilihan no 1')
case {2}
    disp('pilihan no 2')
otherwise
    disp('pilihan anda ')
    disp(x)
end
```

Latihan :

Buatlah program untuk mencari volume dari silinder, kubus, kerucut.

Tampilan:

Menu volume:

=====

1.Silinder

2.Kubus

3.kerucut

Pilihan anda?

MATERI 14

WHILE-END

Sintaknya:

```
while ekspresi
perintah_ke_1
...
...
perintah_ke_n
end
```

- Perintah ke 1 s/d ke n akan terus dikerjakan selama kondisi dalam ekspresi masih dipenuhi (bernilai benar).
- Perintah dalam while end harus memuat penambahan atau pembaharuan counter (pengendali), sehingga kondisi dalam ekspresi selalu berganti.

Contoh penggunaan while end :

```
n=0;
while n < 5
    disp(n)
    n=n+1;
end
```

- Selama $n < 5$ maka $disp(n)$ dan $n=n+1$ akan terus dikerjakan.
- Pembaharuan conter dalam perintah di atas adalah : $n=n+1$

Contoh :Menghitung rata-rata 5 bilangan

```
n=1;jumlah=0;
while n <= 5
    a=input(['masukkan data ke ' num2str(n)]);
    jumlah =jumlah+a;
    n=n+1;
end
rata=jumlah/5;
disp(' ')
disp('Rata-ratanya = ')
disp(rata)
```

Hal-hal yg harus diperiksa:

- 1. apakah ekspresi dlm contoh while-end di atas?
- 2. sudah adakah nilai awal dari variabel pengendali dlm ekspresi tsb?
- 3. perintah apakah yg diulang?
- 4. sudah adakah perubahan/pembaharuan variabel pengendali?

LATIHAN

Cetaklah kalimat “latihan aplikom” selama sebuah bilangan yg dimasukkan tidak = 0

CONTOH :

```
syms y t;  
jawab = 'y';  
while jawab == 'y'  
    a=input('masukkan sebuah bilangan ');  
    t=[num2str(a) ' pangkat 3 = '];  
    disp(t)  
    b=a^3;  
    disp(b)  
    jawab=input('mau mengulang lagi? (y/t) ');  
end
```

%%%%%%%%%