

# **PENGGUNAAN APLIKASI MATLAB UNTUK ANALISA KOMPUTASI**



**PROGRAM STUDI PENDIDIKAN FISIKA  
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN  
UNIVERSITAS KRISTEN INDONESIA  
JAKARTA, Indonesia  
2019**

## **KATA PENGANTAR**

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT, atas limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan penyusunan buku penggunaan aplikasi Matlab untuk analisa komputasi.

Modul ini diperuntukkan bagi mahasiswa Universitas Kristen Indonesia khususnya prodi Pendidikan Fisika Fakultas Keguruan dan Ilmu Pendidikan yang mengikuti matakuliah ICT II. Modul ini difungsikan sebagai petunjuk dalam menyelesaikan persoalan-persoalan matematika dengan menggunakan metode numerik. Modul ini dihususkan untuk dasar dalam pemrograman computer menggunakan Matlab.

Penulis menyadari sepenuhnya bahwa modul ini tentu memiliki banyak kekurangan. Untuk itu penulis menerima masukan dan kritikan konstruktif dari berbagai pihak demi kesempurnaan dimasa yang akan datang. Semoga modul ini dapat bermanfaat bagi pembaca.

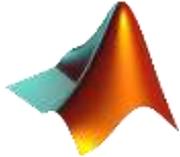
Jakarta, Oktober 2019

Penulis,

Faradiba

# DAFTAR ISI

<b>Chapter 1</b>	Pemrograman M-File	1
<b>Chapter 2</b>	Visualisasi Data	5
<b>Chapter 3</b>	Instruksi Utama	21
<b>Chapter 4</b>	Persamaan Non Linear	35
<b>Chapter 5</b>	Persamaan Linear	38
<b>Chapter 6</b>	Interpolasi	45
<b>Chapter 7</b>	Regresi Linear	48

	<b>MATLAB</b>		
	<b>Chapter 1</b>	<b>M-File</b>	

## 1.1 Pemrograman M-File

### Kemampuan yang diharapkan

- b. Mahasiswa menyelesaikan fungsi dan membuat script secara sederhana

### Uraian Materi

#### A. Konsep Dasar

Pada dasarnya, semua tools yang disediakan oleh MATLAB dibuat dalam format fungsi dan dikelompokkan ke dalam folder-folder toolbox. Selain menggunakan fungsi-fungsi yang telah ada tersebut, kita juga dapat membuat fungsi-fungsi sendiri sesuai kebutuhan. Keuntungan membuat program dalam format fungsi adalah kemudahannya untuk digunakan lagi pada program lainnya. Berikut ini pola untuk menuliskan fungsi pada MATLAB :

```
Function [out1,out2,...] = Nama (in1,in2,...) → bagian deklarasi fungsi
% penjelasan                               → bagian penjelasan fungsi (opsional)
- perintah -                               → bagian program utama
- perintah -
```

Agar lebih memahami cara penulisan fungsi, perhatikan contoh program berikut :

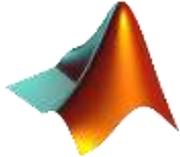
```
function y = pangkat(a,b)
%-----
%fungsi untuk menghitung perpangkatan a^b
%cara menggunakan :
%y = pangkat(2,10)
%-----
hasil = 1;
for i=1:b
    hasil = hasil * a;
end
y = hasil;
```

Simpan program di atas dengan nama sama dengan nama fungsinya yaitu pangkat.m.

Untuk menggunakan fungsi tersebut, pada *Command Window* ketikkan perintah sebagai berikut :

```
>> y = pangkat(2,6)
```

sehingga akan muncul hasil perhitungan pangkatnya. Kemudian apabila ingin melihat penjelasan fungsi, ketikkan perintah sebagai berikut :

	<b>MATLAB</b>		
	<b>Chapter 1</b>	<b>M-File</b>	

```
>> help pangkat
```

### Pemanggilan Fungsi

MATLAB menyediakan banyak sekali fungsi-fungsi yang dapat menyelesaikan berbagai permasalahan komputasi. Misalkan diberikan variabel A dan B sebagai berikut :

```
>>A = [1 3 5];
>>B = [10 6 4];
```

Untuk memanggil fungsi, parameter input dituliskan di dalam tanda kurung :

```
>>max (A) ;
```

Jika terdapat beberapa parameter input, maka penulisannya dipisahkan dengan tanda koma :

```
>>max (A,B) ;
```

Hasil output dari suatu fungsi dapat disimpan ke dalam sebuah variabel :

```
>>maxA = max (A) ;
```

Jika terdapat beberapa parameter output, maka penulisannya menggunakan tanda kurung siku :

```
>>[maxA,location] = max (A) ;
```

Untuk memanggil sebuah fungsi yang tidak memerlukan input dan tidak mengembalikan suatu output maka tuliskan nama fungsinya saja :

```
>>clc
```

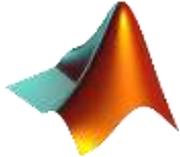
Fungsi clc akan melakukan *clear* pada *Command Window*.

Contoh :

Agar lebih memahami penulisan dan penggunaan fungsi dalam MATLAB perhatikan program berikut ini :

```
function [luas,volume] = balok (p,l,t)
% -----
%fungsi : untuk menghitung luas dan volume pada balok
%cara menggunakan :
%[luas,volume] = balok (12,10,5)
%-----
luas = 2 * (p*l + p*t + l*t);
volume = p*l*t;
```

Simpan program tersebut dengan nama **balok.m**. Selanjutnya, tugas Anda, buatlah fungsi untuk menghitung luas permukaan dan volume pada tabung dan pada kerucut. Simpan

	<b>MATLAB</b>		
	<b>Chapter 1</b>	<b>M-File</b>	

fungsi tersebut dengan nama **tabung.m** dan **kerucut.m**. Setelah Anda selesai membuat fungsi-fungsi tersebut, buatlah sebuah *script* yang meminta inputan dari user, bangun ruang apakah yang akan dihitung luas dan volumenya.

Contoh tampilan program :

```
>> Program Hitung Luas dan Volume <<
Pilihan bangun ruang :
1. Balok
2. Tabung
3. Kerucut
Masukkan pilihan Anda : 1
>> Hitung Luas dan Volume Balok <<
Masukkan panjang balok : 10
Masukkan lebar balok : 8
Masukkan tinggi balok : 5
Luas permukaan balok = 340
Volume balok = 400
```

Contoh script sederhana darimatlab:

```
1 %menghitung akar dari
2 %persamaan kuadrat y=ax^2 + bx + c
3
4 - clc %untuk membersihkan tulisan di command window
5 - clear % menghapus variabel sebelumnya
6
7 %masukkan input/kontanta
8 - a=input('masukkan kontanta a=')
9 - b=input('masukkan kontanta b=')
10 - c=input('masukkan kontanta c=')
11
12 %perhitungan akar
13 - x1=(-b + sqrt(b^2-4*a*c))/(2*a)
14 - x2=(-b - sqrt(b^2-4*a*c))/(2*a)
```

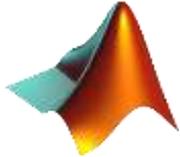
**Gambar 5.1** Contoh script sederhana

Jika dieksekusi, hasilnya akan terlihat di *Command Window* seperti berikut:



```
Command Window
masukkan kontanta a=1
a =
    1
masukkan kontanta b=4
b =
    4
masukkan kontanta c=-5
c =
   -5
x1 =
    1
x2 =
   -5
```

**Gambar 5.2** Contoh hasil running script

	<b>MATLAB</b>		
	<b>Chapter 1</b>	<b>M-File</b>	

## **B. Penugasan Kelas**

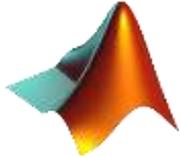
1. Buatlah script sederhana untuk menghitung luas dan keliling lingkaran!
2. Buatlah script sederhana untuk menghitung mean 46, 60, 87, 50, 86, 70, 90, 100!

## **C. Rangkuman**

1. M-file merupakan sederetan perintah matlab yang dituliskan secara berurutan sebagai sebuah file. Nama file yang tersimpan akan memiliki ekstensi .m yang menandakan bahwa file yang dibuat adalah file matlab. M-file dapat ditulis sebagai sebuah script atau dapat pula ditulis sebagai sebuah fungsi yang menerima argument atau masukan yang menghasilkan output.

## **D. Evaluasi Formatif 3**

1. Diketahui sebuah kubus dengan panjang sisi 25 cm. Buatlah script sederhana untuk menghitung volume kubus!
2. Buatlah script sederhana untuk mengenai identitas pribadi anda!

	<b>MATLAB</b>		
	<b>Chapter 2</b>	<b>Visualisasi Data</b>	

## 2.1 Visualisasi Data

Visualisasi data merupakan pembuatan dan kajian dari representasi visual dari data. Tujuan utama dari visualisasi data adalah untuk mengkomunikasikan informasi secara jelas dan efisien kepada pengguna lewat grafik informasi yang dipilih, seperti tabel dan grafik. Visualisasi yang efektif membantu pengguna dalam menganalisa dan penalaran tentang data dan bukti. Ia membuat data yang kompleks bisa diakses, dipahami dan berguna. Pengguna bisa melakukan pekerjaan analisis tertentu, seperti melakukan perbandingan atau memahami kausalitas, dan prinsip perancangan dari grafik. Grafik dari berbagai tipe digunakan untuk melihat pola atau keterkaitan dalam data untuk satu atau lebih variabel. Grafik pada umumnya, memungkinkan untuk memetakan dimensi pada data menjadi properti visual berbentuk geometris.

MATLAB menyediakan berbagai fungsi untuk menampilkan data secara dua dimensi maupun 3 dimensi. Pada kasus dimana anda membuat grafik dalam tiga dimensi, agar dapat menggambar permukaan dan menempatkan bingkai pada grafik tersebut. Warna digunakan untuk mewakili dimensi keempat.

### Kegiatan Pembelajaran 1: Membuat Grafik Data

#### Kemampuan yang diharapkan

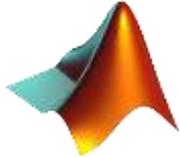
- a. Mahasiswa mampu membuat plot 2D dengan berbagai jenis plot
- b. Mahasiswa mampu membuat plot 3D.

#### Uraian Materi

##### A. Konsep Materi

Perintah plot sering digunakan untuk menggambarkan grafik 2 dimensi, perintah plot menggambarkan data dalam array pada sumbu yang bersesuaian dan menghubungkan titik-titik tersebut dengan garis lurus. Untuk keterangan lebih jelas mengenai fungsi plot ketik **help plot** di command window.

Grafik, pada umumnya, memungkinkan untuk memetakan dimensi pada data menjadi properti visual berbentuk geometris. Dalam suatu “scatterplot” (grafik x-y dengan bintik

	<b>MATLAB</b>		
	<b>Chapter 2</b>	<b>Visualisasi Data</b>	

bintik tersebar di bidangnya) misalnya, dua dimensi dipetakan dalam bidang x- dan y-. Dapat pula ditunjukkan dimensi ketiga dengan warna atau ukuran berbeda dari simbol-simbol yang telah terpampang. Grafik garis khususnya cocok untuk menunjukkan evolusi temporal, sementara grafik batang bagus sekali untuk membanding data dengan kategori tertentu. Visualisasi data dapat ditumpuk menjadi beberapa elemen grafik. Untuk membandingkan sejumlah kecil kelompok pada data, beragam sajian grafik dapat menjadi alternatif untuk menganalisis data dengan lebih mudah. Dalam semua grafik dapat menggunakan skala yang berbeda untuk mempelajari aspek-aspek yang berbeda pada data (linear atau skala logaritma).

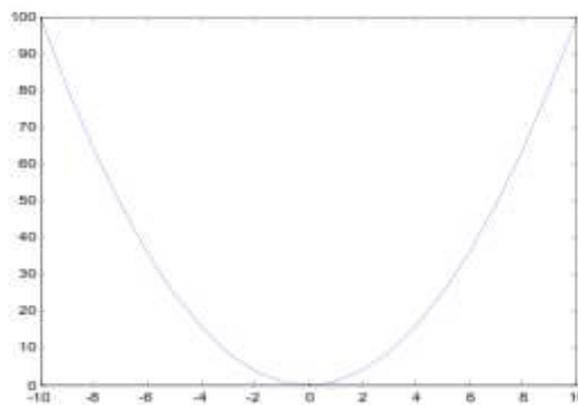
### Plot

Fungsi plot digunakan untuk menggambar grafik 2D dengan skala linear pada kedua sumbunya

Contoh:

```
>> x=-10:10;
>> y=x.^2;
>> plot(x,y)
```

Hasilnya akan tampak sebagai berikut:



Gambar 6.1 Operator Relasional

Contoh :

Data reaksi berikut telah diperoleh dari reaksi peluruhan sederhana

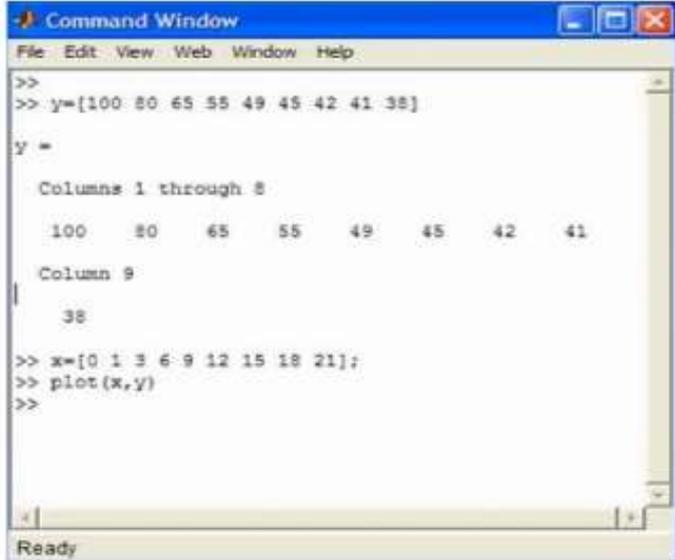


Menggunakan MATLAB untuk memplot konsentrasi A dalam mol/L terhadap waktu reaksi, dalam menit.

Waktu(Menit)	Konsentrasi (mol/L)
0	100
1	80
3	65
6	55
9	49
12	45
15	42
18	41
21	38

**Penyelesaian :**

Pertama, data harus dimasukkan ke dalam MATLAB sebagai dua vektor. Vektor x dan y didefinisikan dalam command window, diikuti dengan perintah untuk mengplot data.



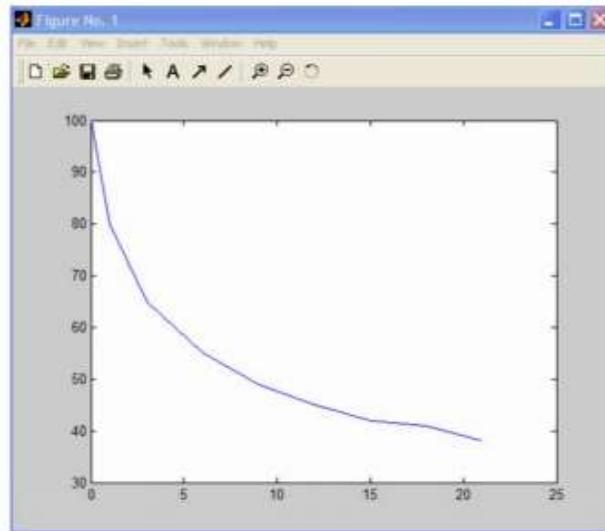
```

Command Window
File Edit View Web Window Help
>>
>> y=[100 80 65 55 49 45 42 41 38]
y =
Columns 1 through 8
    100    80    65    55    49    45    42    41
Column 9
         38
>> x=[0 1 3 6 9 12 15 18 21];
>> plot(x,y)
>>
Ready

```

**Gambar 6.2 Tampilan Input Data**

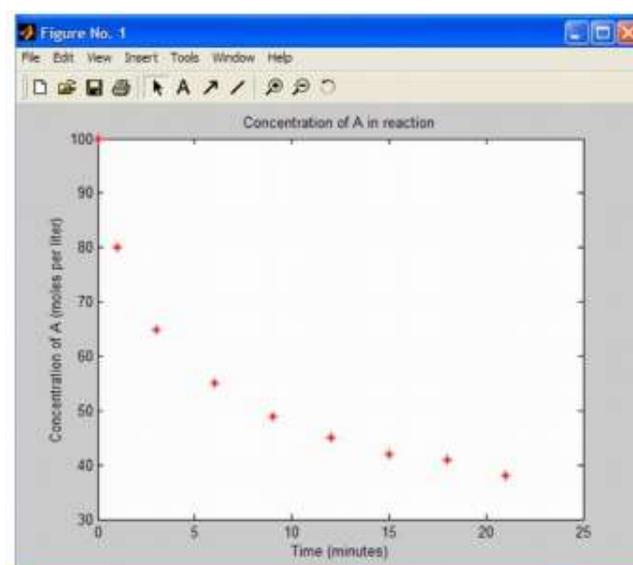
Hasil grafik ditunjukkan :



**Gambar 6.3 Tampilan Output dalam Grafik**

Matriks baris 'x' (atau vektor) memiliki keluaran tampilan menggunakan ';' pada akhir baris. Synta dapat digunakan untuk menentukan judul dan labels, tetapi lebih mudah menggunakan pendekatan berdasarkan GUI untuk mengedit gambar.

- ✓ Pilih edit plot pada menu tools atau klik kursor pada gambar.
- ✓ Double click pada ruang putih dalam grafik. Hal ini memungkinkan untuk mengedit. Selanjutnya judul dan aksis dapat disisipkan di bawah perintah label.
- ✓ Selanjutnya warna garis, atau bentuk atau bentuk marker dapat diedit, kurva khir ditunjukkan



**Gambar 6.4 Hasil Edit Output**

Untuk menampilkan data statistik sederhana, ikuti petunjuk. Tools data statistics dan nilai minimum, maximum, mean, median, standard deviation and range and y akan tampil. Di dalam kotak ini setiap statistik dapat ditambahkan ke kurva sebagai titik data/garis.

Argumen tambahan ini adalah suatu karakter string yang terdiri dari satu atau lebih karakter dari gambar di bawah ini:

**Tabel 6.1 Argumen Tambahan pada Grafik**

Simbol	Warna	Penandaan	Simbol
b	Biru	.	Titik
r	Merah	o	Lingkaran
g	Hijau	x	Tanda x
c	Cyan	+	Tanda plus
m	Magenta	*	Tanda bintang
y	Kuning	s	Bujur sangkar
k	Hitam	d	Diamon
w	putih	p	Pentagram
		h	Heksagram

### Grafik Multiple Plot

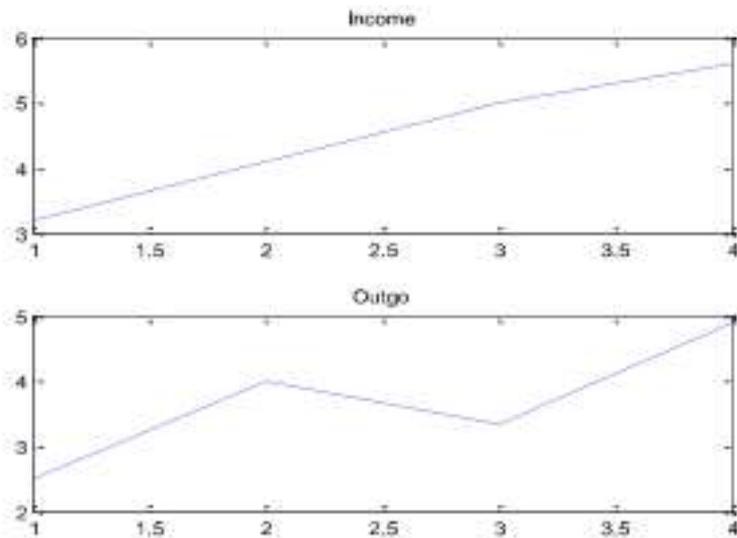
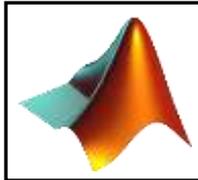
Function subplot digunakan untuk membuat suatu figure dapat memuat lebih dari satu gambar. Perintah subplot didefinisikan sebagai :

Subplot(n,m,i)

Perintah tersebut membagi suatu figure menjadi suatu matriks m x n area grafik dan i berfungsi sebagai indeks penomoran gambar. Subplot dinomori dari kiri ke kanan dimulai dari baris teratas.

**Contoh :** Untuk memplot grafik atas dan bawah

```
>>income = [3.2 4.1 5.0 5.6];
>>Outgo = [2.5 4.0 3.35 4.9];
>>Subplot(2,1,1); plot(income)
>>Title('income')
>>Subplot(2,1,2); plot(outgo)
>>Title ('outgo')
```



Gambar 6.5 Tampilan Subplot Atas Bawah

### Grafik 3 Dimensi

Perintah menggambar grafik 3D

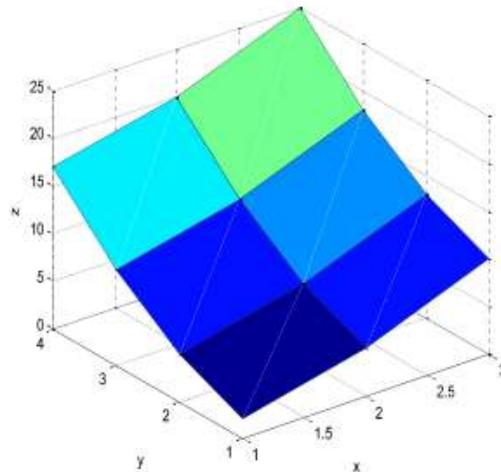
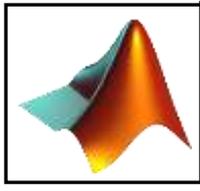
Surf(x,y,z)

Contoh : Misalkan data berikut:

X	Y	Z(x=1)	Z(x=2)	Z(x=3)
1	1	2	5	10
2	2	5	8	13
3	3	10	13	18
	4	17	20	25

Penulisan di MATLAB :

```
>> x=[1 2 3];  
>>y=[1 2 3 4];  
>>z=[2 5 10; 5 8 13; 17 20 25]  
>>surf(x,y,z)  
>>xlabel('x')  
>>ylabel('y')  
>>zlabel('z')
```

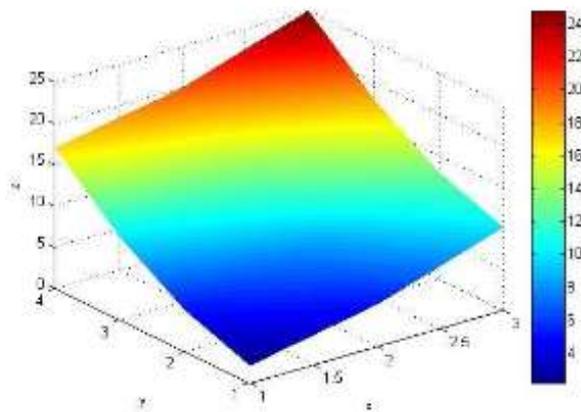


Gambar 6.6 Tampilan Grafik 3D

Untuk mempercantik tampilan dan mempermudah penafsiran grafik dengan menambah legenda warna:

```
>> shading interp
```

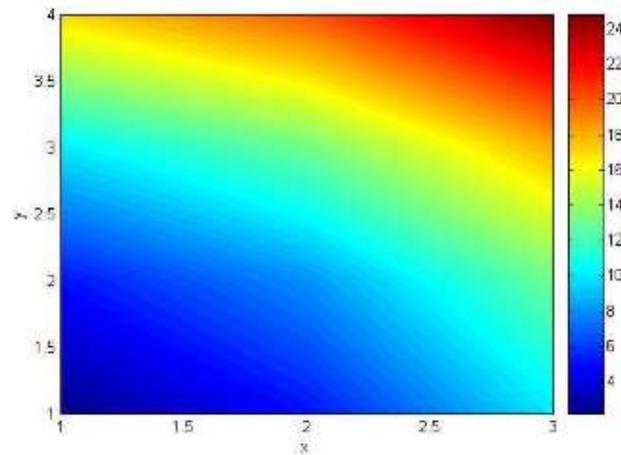
**Colorbar**



Gambar 6.7 Grafik 3D dengan Warna Yang Kontras

Apabila penafsiran grafik 3D seperti tercetak dimuka masih dirasakan sulit, MATLAB telah menyediakan perintah untuk membuat grafik 3D menjadi grafik 2D

```
>> pcolor (x, y, z)  
>> xlabel('x')  
>> ylabel('y')  
>> zlabel('z')  
>> shading interp  
>> Colorbar
```



**Gambar 6.8 Mengubah Grafik 3D Menjadi 2D**

### B. Penugasan Kelas

1. Buatlah grafik  $Y=X^2$  dan  $Y=X^3$ ?
2. Buatlah grafik  $Y=\sin(2x)$  dan  $Y=\cosinus(2x)$ ?

### C. Rangkuman

1. Fungsi *plot* adalah untuk menggambar grafik dalam bentuk yang dapat berbeda-beda, tergantung pada argumen masukan.
2. Perintah *plot* menciptakan grafik suatu vector atau kolom-kolom matriks. Bentuk perintahnya adalah `plot(x1,y1, x2,y2, S2, ...)` dengan  $(X_n, Y_n)$  adalah himpunan data dan  $S_n$  adalah string tambahan yang menentukan warna, symbol penandaan dan style garis.

### D. Evaluasi Formatif 1

1. Buatlah grafik untuk fungsi dibawah ini!
  - a.  $Y = \sin 2x, -2\pi \leq x \leq 2\pi$
  - b.  $Y = \sin(2x + \pi/2), -2\pi \leq x \leq 2\pi$
  - c.  $Y = x^3, -10\pi \leq x \leq 10\pi$
  - d.  $Y = \{-3, -2, 4, 3, 3, 1, 2, 5\}$
  - e.  $Y = 3 \sin x + \cos x, -\pi \leq x \leq \pi$

f.  $Y = \tan x - 2 \sin x + \cos x, -\pi \leq x \leq \pi$

## Kegiatan Pembelajaran 2: Visualisasi Data

### Kemampuan yang diharapkan

- a. Mahasiswa dapat menampilkan berbagai bentuk tampilan data
- b. Mahasiswa dapat menampilkan beberapa tampilan data dalam satu gambar

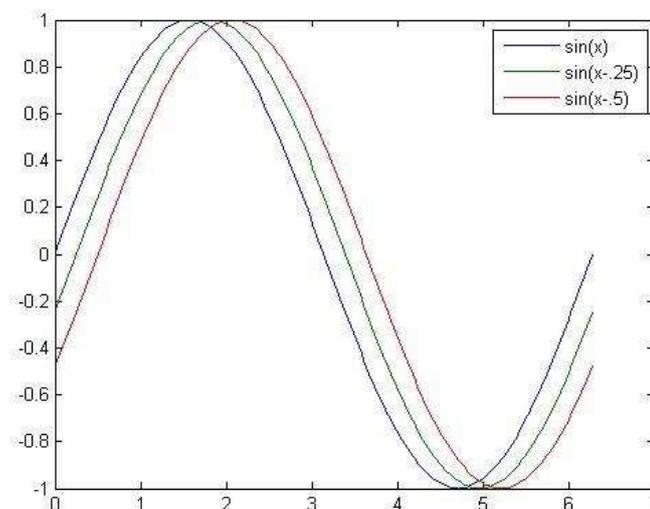
### Uraian Materi

#### A. Konsep Dasar

Beberapa pasangan argumen x-y akan menghasilkan beberapa grafik dengan menggunakan sekali pemanggilan fungsi plot. MATLAB akan menggunakan warna yang berbeda untuk masing-masing garis. Sebagai contoh, perintah berikut ini akan melakukan plot terhadap tiga buah fungsi yang berbeda :

```
x = 0:pi/100:2*pi;
y = sin(x);
y2 = sin(x-.25);
y3 = sin(x-.5);
plot(x,y,x,y2,x,y3)
legend('sin(x)', 'sin(x-.25)', 'sin(x-.5)')
```

Berikut ini hasilnya :



**Gambar 6.9** Plot untuk Tiga Buah Fungsi

### Mengubah Bentuk dan Warna Garis

Anda dapat mengubah warna, bentuk garis, dan juga marker (seperti tanda plus dan lingkaran)

ketika Anda melakukan plot data menggunakan perintah berikut ini :

```
plot(x,y,'color_style_marker')
```

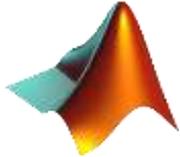
*color\_style\_marker* merupakan sebuah string yang terdiri atas satu sampai empat buah karakter yang merepresentasikan warna, bentuk garis, dan tipe marker. Sebagai contoh :

```
plot(x,y,'r:+')
```

Perintah di atas akan melakukan plot data menggunakan *dotted line* berwarna merah dan menempatkan sebuah *marker +* pada setiap titik data. Berikut ini tabel simbol yang dapat digunakan :

**Tabel 2.2 Tipe Bentuk dan Warna Garis**

Type	Values	Meanings
Color	'c' 'm' 'y' 'r' 'g' 'b' 'w' 'k'	cyan magenta yellow red green blue white black
Line style	'-' '--' '.' '.-' no character	solid dashed dotted dash-dot no line
Marker type	'+' 'o' '*' 'x' 's' 'd' '^' 'v' '>' '<' 'p' 'h' no character	plus mark unfilled circle asterisk letter x filled square filled diamond filled upward triangle filled downward triangle filled right-pointing triangle filled left-pointing triangle filled pentagram filled hexagram no marker

	<b>MATLAB</b>		
	<b>Chapter 2</b>	<b>Visualisasi Data</b>	

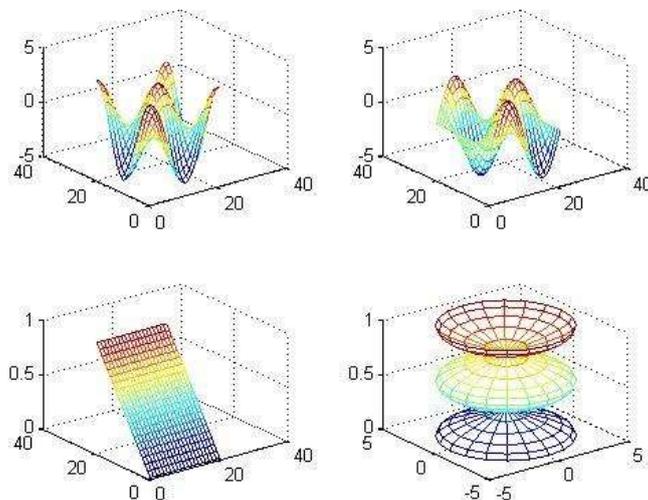
### Menampilkan Beberapa Plot dalam Satu Figure

Perintah subplot memungkinkan Anda untuk menampilkan beberapa plot pada sebuah window yang sama. Perintah berikut ini,

```
subplot(m,n,p)
```

akan membagi figure window ke dalam matrik subplot berukuran  $m \times n$  dan memilih subplot ke- $p$  untuk plot saat ini. Penomoran plot dimulai dari baris pertama pada *figure window*, kemudian baris kedua, dst. Sebagai contoh, perintah berikut ini akan melakukan plot data ke dalam empat buah bagian subregion pada figure window :

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1); mesh(X)
subplot(2,2,2); mesh(Y)
subplot(2,2,3); mesh(Z)
subplot(2,2,4); mesh(X,Y,Z)
```



**Gambar 6.10** Tampilan Empat Plot dalam Satu Gambar

### Visualisasi Data 2D dengan Fungsi Bar

Fungsi bar digunakan untuk menampilkan data dalam grafik batang. Sintak penulisannya sebagai berikut :

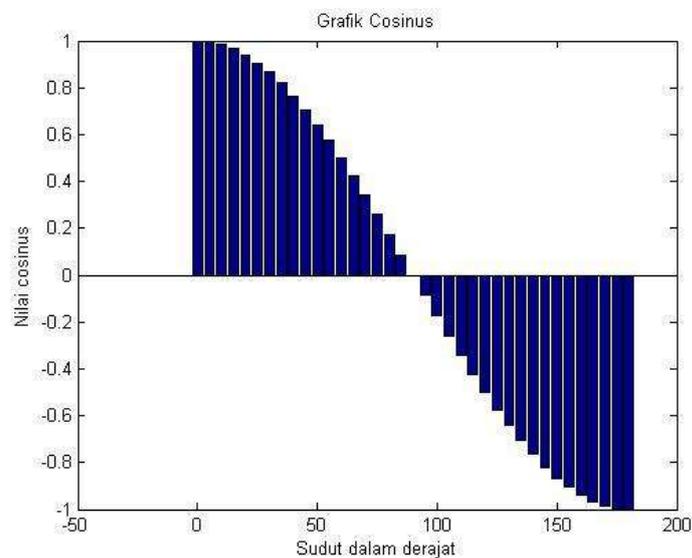
```
bar(x,y);
```

Contoh pemakaian dalam program sebagai berikut :

```

sudut = [0:5:180];
y = cos(sudut*pi/180);
bar(sudut,y)
title('Grafik Cosinus');
xlabel('Sudut dalam derajat');
ylabel('Nilai cosinus');

```



**Gambar 6.11 Tampilan Data dengan Fungsi Bar**

Selain fungsi plot dan juga fungsi bar, MATLAB juga masih menyediakan beberapa fungsi visualisasi 2D lainnya antara lain fungsi stem dan fungsi stair.

### Visualisasi Data 3D

Plot tiga dimensi biasanya digunakan untuk menampilkan suatu permukaan yang didefinisikan oleh fungsi dengan dua buah variabel,  $z = f(x,y)$ . Untuk memperoleh  $z$ , pertama-tama buatlah sekumpulan titik  $(x,y)$  pada domain fungsi menggunakan meshgrid.

```

[X,Y] = meshgrid(-2:.2:2);
Z = X .* exp(-X.^2 - Y.^2);

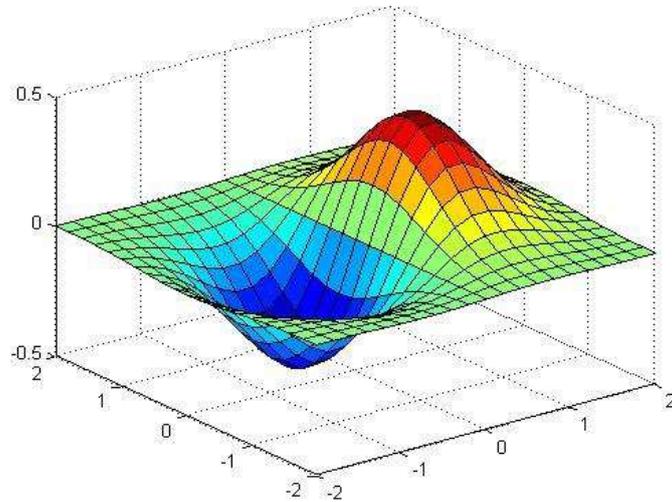
```

Kemudian buat plot permukaannya.

```

surf(X,Y,Z)

```



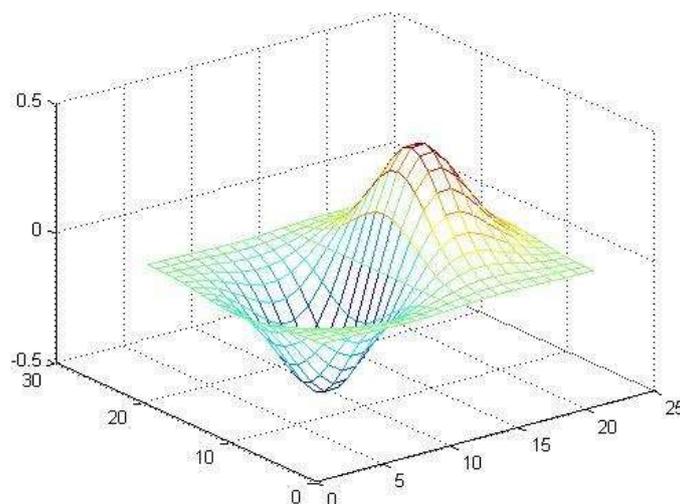
**Gambar 6.12** Tampilan plot 3D

### Fungsi mesh

Fungsi ini digunakan untuk menampilkan data dalam bentuk permukaan 3D. Cirinya berupa grid-grid yang menghubungkan 4 titik terdekat dalam ruang 3D. Sintak penulisannya sebagai berikut :

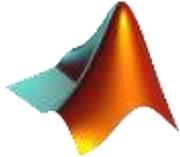
*mesh (Z)*

Jika fungsi Z di atas digambarkan menggunakan fungsi mesh, hasilnya sebagai berikut :



**Gambar 6.13** Bentuk yang Menampilkan Permukaan

### Fungsi surf

	<b>MATLAB</b>		
	<b>Chapter 2</b>	<b>Visualisasi Data</b>	

Fungsi ini digunakan untuk menampilkan data dalam bentuk permukaan 3D seperti pada mesh.

Bedanya adalah setiap grid diisi dengan warna tertentu sesuai bobot nilai. Sintak penulisannya

sebagai berikut :

```
surf(Z)
```

atau

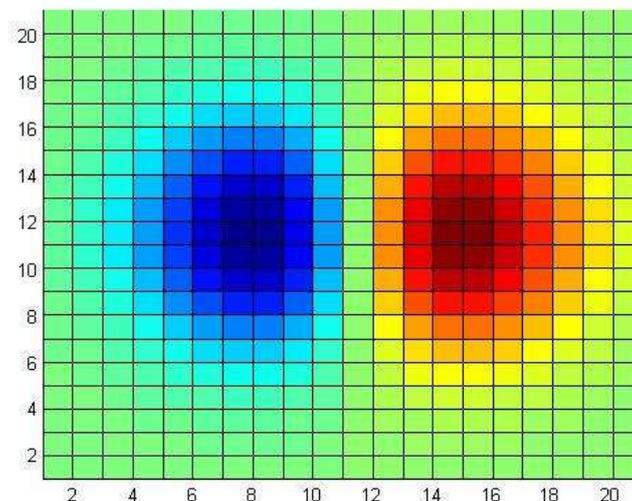
```
surf(X,Y,Z)
```

### **Fungsi pcolor**

Fungsi ini digunakan untuk menampilkan data 3D dalam bentuk permukaan 2D (tampak atas). Cirinya berupa grid berwarna yang menunjukkan bobot nilai tertentu. Sintak penulisannya sebagai berikut :

```
pcolor(Z)
```

Jika fungsi *Z* di atas digambarkan menggunakan fungsi *pcolor*, hasilnya sebagai berikut :



**Gambar 6.14 Tampilan Permukaan Tampak Atas**

### **Fungsi contourf**

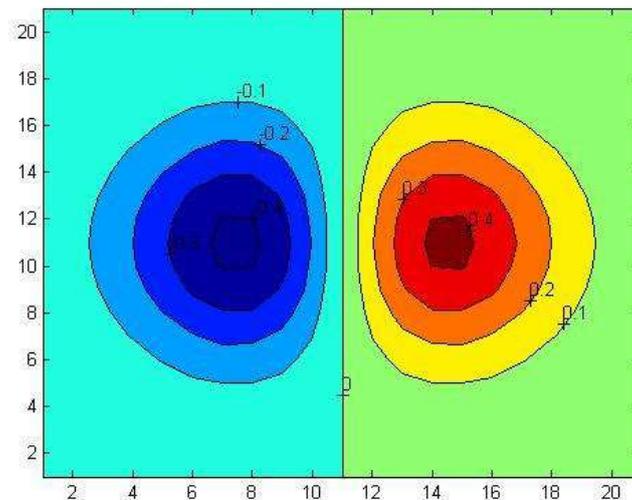
Fungsi ini digunakan untuk membuat garis kontur dari data 3D. Garis-garis kontur ini dibuat dengan teknik interpolasi dari titik-titik terdekat dan pada setiap level kontur diberikan warna sesuai bobot garis konturnya. Sintak penulisannya sebagai berikut :

```
obj_handle = contourf(Z)
```

`clabel(obj_handle)`

`obj_handle` adalah object handle yang dibutuhkan untuk menampilkan label pada garis kontur.

Hasilnya sebagai berikut :



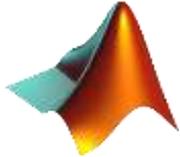
**Gambar 6.15 Menampilkan Label dari Garis Kontur**

## B. Penugasan Kelas

1. Dari contoh-contoh program yang sudah anda jalankan, berikan penjelasan arti setiap perintah terhadap output yang dihasilkannya.
2. Bagaimana cara menampilkan grafik untuk tampilan tiga dimensi dan grafik polar.

## B. Rangkuman

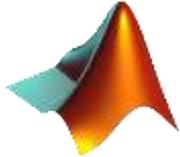
1. Beberapa argumen pasangan xy dapat dibuat beberapa kurva dengan panggilan tunggal untuk plot. Gambar secara otomatis akan menampilkan kurva dengan urutan standar warna yang berbeda-beda antara kumpulan data.
2. Perintah `subplot` memungkinkan kita untuk menampilkan plot rangkap dalam jendela yang sama atau mencetaknya pada selebar kertas yang sama.

	<b>MATLAB</b>		
	<b>Chapter 2</b>	<b>Visualisasi Data</b>	

3. Perintah [xlabel](#), [ylabel](#), dan [zlabel](#) akan menambahkan label pada sumbu  $x$ -,  $y$ -, dan  $z$ . Sedangkan perintah [title](#) akan menambahkan judul pada posisi atas gambar dan fungsi [text](#) akan menyisipkan teks dimanapun pada gambar.
4. Fungsi [meshgrid](#) melakukan transformasi domain yang ditentukan oleh vektor tunggal atau dua vektor  $x$  dan  $y$  kedalam matriks  $X$  dan  $Y$  yang digunakan untuk mengevaluasi fungsi dua variabel. Baris  $X$  merupakan pemindahan dari vektor  $x$  dan kolom  $Y$  adalah pindahan dari vektor  $y$ .

#### **D. Evaluasi Formatif 2**

1. Bagaimana cara menampilkan lebih dari satu persamaan dalam satu grafik? Misalnya anda memiliki dua fungsi sinus yang berbeda fase. Fungsi pertama anda tampilkan, lalu anda lanjutkan menampilkan fungsi kedua, dengan catatan tampilan pada fungsi pertama tidak boleh hilang.
2. Bagaimana cara menampilkan lebih dari satu grafik dalam satu tampilan?, satu fungsi ditampilkan diatas dan fungsi lainnya di bagian bawah.
3. Bagaimana cara menampilkan dua fungsi dimana masing-masing fungsi disajikan dalam grafik berbeda. Misalnya anda gunakan contoh kasus, fungsi pertama anda tampilkan pada figure(1), sementara fungsi kedua anda tampilkan pada figure(2).

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

### 3.1 Instruksi Utama

Instruksi utama terbagi atas tiga jenis yaitu: Instruksi runtunan (*sequence*), instruksi pemilihan (*selection*) dan instruksi perulangan (*repetition*). Instruksi runtunan adalah Instruksi yang dikerjakan secara beruntun, atau berurutan, baris perbaris, mulai dari pertama hingga baris terakhir, tanpa ada loncatan atau perulangan. Instruksi pemilihan adalah Instruksi yang dipakai untuk memilih satu aksi dari beberapa kemungkinan aksi berdasarkan suatu persyaratan. Sedangkan instruksi perulangan adalah instruksi yang dapat mengulangi pelaksanaan sederetan instruksi lain berulang kali sesuai dengan persyaratan yang ditetapkan. Ketiga jenis instruksi tersebut membentuk konstruksi suatu algoritma / program. Jadi, sebuah algoritma / program dapat dibangun dari tiga buah instruksi dasar tersebut.

## Kegiatan Pembelajaran 1: Instruksi Runtunan

### Kemampuan yang diharapkan

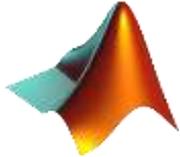
- a. Mahasiswa mampu membuat *script* dengan menggunakan instruksi runtunan
- b. Mahasiswa mampu menyelesaikan persoalan fisika dengan menggunakan instruksi runtunan

### Uraian Materi

#### A. Konsep Teori

Instruksi yang dikerjakan secara beruntun, atau berurutan, baris perbaris, mulai dari pertama hingga baris terakhir, tanpa ada loncatan atau perulangan.

- a. Tiap instruksi dikerjakan sekali, satu persatu
- b. Urutan pelaksanaan instruksi sama dengan urutan penulisan algoritma
- c. Instruksi terakhir merupakan akhir dari algoritma
- d. Urutan penulisan instruksi bisa menjadi penting bila diubah dapat menyebabkan hasil yang berbeda.

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

### Contoh Program :

%Sebuah batu dilempar ke dalam sumur dengan  $V_0=3\text{m/s}$ . Batu terdengar mengenai air setelah 2 detik tentukan kecepatan saat mencapai air dan kedalaman sumur

```
Vo=-3;
a=-9.8;
t=0:0.5:2;
Vt=Vo+a*t;
s=Vo*t+0.5*a*t.^2;
table=[t;Vt;s]'
```

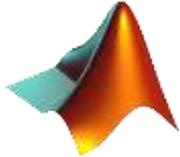
### B. Penugasan Kelas

1. Sebuah batu dilempar dengan kecepatan 40 m/s dengan sudut  $50^\circ$ . Tentukan kedudukan batu setelah 1,2,3,4,dan 5 detik. Buatlah *script* di MATLAB beserta penyelesaiannya!

### C. Rangkuman

1. Struktur Runtunan atau *Sequential* merupakan struktur dasar algoritma yang terdiri dari satu atau lebih instruksi, yang tiap tiap instruksinya dikerjakan secara berurutan sesuai dengan penulisan algoritmanya.
2. Hasil dari algoritma berstruktur runtunan ini ditentukan oleh urutan instruksinya. jika terjadi perubahan instruksi, maka hasil akhirnya kemungkinan juga berubah. Instruksi algoritma pada struktur runtunan ini akan dikerjakan semua dan tidak ada yang diabaikan.
3. Semua program dibuat berdasarkan struktur algoritma runtunan.

### D. Evaluasi Formatif 1

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

1. Diketahui susunan muatan listrik seperti pada gambar disamping. Dimana  $q_1 = +1.5 \times 10^{-3} \text{C}$ ,  $q_2 = -0.5 \times 10^{-3} \text{C}$ ,  $q_3 = +0.2 \times 10^{-3} \text{C}$ ,  $r_1 = 1.2 \text{m}$ , dan  $r_2 = 0.5 \text{m}$ . Besarnya gaya resultan pada muatan  $q$ !

## Kegiatan Pembelajaran 2: Instruksi Pemilihan

### Kemampuan yang diharapkan

- a. Mahasiswa mampu membuat *script* dengan menggunakan instruksi `if/else`
- b. Mahasiswa mampu menyelesaikan persoalan fisika dengan menggunakan instruksi `if/else`

### Uraian Materi

#### A. Konsep Dasar

Instruksi pemilihan adalah Instruksi yang dipakai untuk memilih satu aksi dari beberapa kemungkinan aksi berdasarkan suatu persyaratan Instruksi pemilihan ada dua bentuk yaitu:

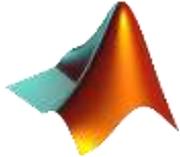
- a. Instruksi `if/then else`
- b. Instruksi `case`

Instruksi `if/then/else` digunakan untuk memilih alternatif apabila suatu syarat atau kondisi dipenuhi, atau memilih satu alternatif dari 2 kemungkinan berdasarkan apakah syarat terpenuhi atau tidak

#### 1. INSTRUKSI `if/then/else`

##### 1.1 Bentuk 1 Kasus

```
if (syarat)
    then aksi
Endif
```

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

Apabila syarat dipenuhi maka aksi dijalankan. Contoh: nilai x akan ditambah 5 apabila nilainya lebih besar dari 100.

```

if (x > 100)
    then x = x+5
endif

```

### 1.2 Bentuk 2 Kasus

```

if (syarat)
    then aksi-1
    else aksi-2
Endif

```

Apabila syarat dipenuhi maka “aksi-1” dilaksanakan, tetapi bila syarat tidak dipenuhi maka “aksi-2” yang dilaksanakan. Contoh: Bila a lebih besar dari 0 maka akan ditampilkan “bilangan ini positif” selain itu akan ditampilkan “bilangan ini negatif”.

```

if (a > 0)
    Then write ("bilangan ini positif")
    else write ("bilangan ini negatif")
endif

```

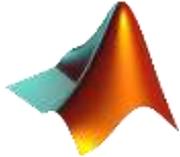
### 1.3 Bentuk bersusun (lebih dari 1 syarat)

```

if (syarat-1)
    then aksi-1
    else if (syarat-2)
        then aksi-2
        else aksi-3
    endif
endif

```

Pada bentuk ini “aksi-1” dilaksanakan segera ketika syarat-1 dipenuhi, “aksi-2” dilakukan apabila syarat-1 tidak dipenuhi tetapi syarat-2 dipenuhi, dan “aksi-3” hanya dilakukan apabila kedua syarat tidak terpenuhi.

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

```

if (syarat-1)
    then if (syarat-2)
        then aksi-1
        else aksi-2
    endif
    else aksi-3
endif

```

### Contoh Program :

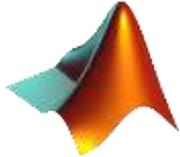
```

%gaya listrik
disp('gaya listrik');
Q1=input('muatan 1=');
Q2=input('muatan 2=');
r=input('jarak muatan=');
k=9*10^9;
F=k*(Q1*Q2)/r^2;
fprintf('F= %6.2f\n',F);
    if F<0
        fprintf('terjadi tarik menarik');
    else if F>0
        fprintf('terjadi tolak meolak');
    else F=0
        fprintf('muatan diam');
    end;
end;

```

## 2. INSTRUKSI case

**Instruksi case** digunakan sebagai instruksi pemilihan di mana aksi yang akan dilakukan bergantung pada nilai dari satu macam variabel saja. Dengan kata lain, kasus yang berdiri sendiri.

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

```

case (variabel)
    nilai-1 : aksi-1;
    nilai-2 : aksi-2;
    nilai-3 : aksi-3;
    .....
    default : aksi-n;
endcase

```

Bentuk case tersebut di terjemahkan :

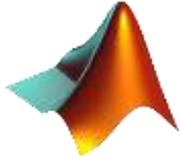
1. Dimungkinkan ada n-buah aksi, mulai dari aksi-1 hingga aksi-n
2. Setiap aksi hanya dilakukan apabila suatu nilai variabel dicapai sesuai dengan persyaratan. Misalnya, aksi-1 dilakukan hanya bila variabel bernilai nilai-1, aksi-2 dilaksanakan hanya apabila variabel bernilai nilai-2 dan seterusnya.
3. Apabila tidak ada satupun nilai variabel yang cocok maka aksi-n dikerjakan sebagai aksi “default” (hanya dikerjakan apabila tidak ada yg memenuhi syarat).

### Contoh Program :

```

%pemilihan rumus perhitungan
disp('pilih rumus perhitungan');
disp('1.volume balok');
disp('2.tegangan listrik');
x=input('pilihan anda (1-2)->');
switch x
case 1
disp('volume & luas balok');
p=input('panjang=');
l=input('lebar=');
t=input('tinggi=');
V= p*l*t
L= 2*((p*l)+(l*t)+(p*t))
disp(['volume balok= 'num2str(V)]);
disp(['luas balok= 'num2str(L)])
case 2
disp('tegangan & daya listrik');

```

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

```

i=input('arus=');
R=input('hambatan=');
V = i*R
P = V*i
disp(['tegangan listrik= 'num2str(v)]);
disp(['daya listrik= 'num2str(p)]);
otherwise
disp('tidak ada dalam pillihan')
end;

```

### Tampilan Output :

```

>>Pilih rumus perhitungan
>>Pilihan anda (1-2)> 1
>>Volume & luas balok
>>Panjang = 2
>>Lebar= 2
>>Tinggi= 2
>>V = 8
>>L = 4

```

**Gambar 7.1 Tampilan Output**

## B. Penugasan Kelas

- Gunakan instruksi pemilihan tersebut (if/else) pada soal optik geometri dengan keterangan :

$$\text{Titik fokus : } f = \frac{(s_1 \times s)}{(s + s_1)}$$

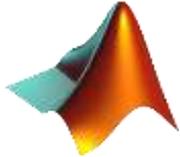
Bila  $f < 0$  maka cermin cembung

Bila  $f > 0$  maka cermin cekung

Bila  $f = 0$  maka cermin datar

## C. Rangkuman

- Percabangan if/else digunakan untuk pengujian sebuah kondisi. Jika kondisi yang diuji tersebut terpenuhi, maka program akan menjalankan pernyataan-

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

pernyataan tertentu. Jika kondisi yang diuji salah, program akan menjalankan pernyataan yang lain.

2. Percabangan case digunakan untuk memilih beberapa kondisi yang ada. Jika kondisi tidak sesuai dengan variabel maka tidak dijalankan ke kondisi tersebut melainkan dianggap sebagai “*default*” (tidak dijalankan).

#### D. Evaluasi Formatif 2

1. Gunakan instruksi pemilihan tersebut (`if/else`) pada soal percepatan dengan keterangan:

$$\text{percepatan : } a = \frac{(Vt - V_0)}{t}$$

Bila  $a < 0$  maka benda diperlambat

Bila  $a > 0$  maka benda dipercepat

Bila  $a = 0$  maka benda diam

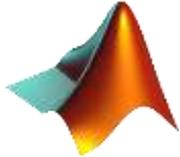
2. Gunakan instruksi pemilihan tersebut (`if/else`) pada soal efisiensi mesin. dengan keterangan :

$$\text{efisiensi: } e = 1 - \left(1 - \frac{T_2}{T_1}\right) \times 100\%$$

Bila  $0 \leq e \leq 60\%$  maka mesin boros

Bila  $60\% \leq e \leq 80\%$  maka mesin irit

Bila  $80\% \leq e \leq 100\%$  maka mesin ideal

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

## Kegiatan Pembelajaran 3: Instruksi Perulangan

### Kemampuan yang diharapkan

- a. Mahasiswa mampu membuat script perulangan dengan menggunakan instruksi `for`
- b. Mahasiswa mampu membuat script perulangan dengan menggunakan instruksi `while`

### Uraian Materi

#### A. Konsep Dasar

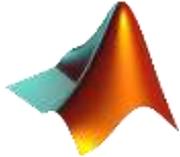
Instruksi perulangan (*repetition*) adalah instruksi yang dapat mengulangi pelaksanaan sederetan instruksi lain berulang kali sesuai dengan persyaratan yang ditetapkan

Struktur Instruksi perulangan :

1. Kondisi perulangan : suatu kondisi yang harus dipenuhi agar perulangan dapat terjadi
2. Badan (*body*) perulangan : deretan instruksi yang akan diulang-ulang pelaksanaannya.
3. Pencacah (*counter*) perulangan : suatu variabel yang nilainya harus berubah agar perulangan dapat terjadi dan pada akhirnya membatasi jumlah perulangan yang dapat dilaksanakan.

Ada 3 macam bentuk instruksi perulangan yang biasa ditemukan dalam program, yaitu:

- a. Perulangan : `while` .....
- b. Perulangan : `repeat - until`

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

c. Perulangan : for

### 1. PERULANGAN **while** .....

Instruksi **while** digunakan untuk melakukan pengulangan proses tanpa diketahui jumlah pengulangan. Iterasi ini akan berhenti bila mencapai syarat tertentu. Sintaksnya

```
while syarat
    Perintah-perintah
end
```

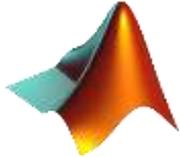
Makna dari bentuk instruksi tersebut adalah : ulangi ... instruksi-instruksi ... selama kondisi yang diberikan masih terpenuhi.

Ada instruksi yang berkaitan dengan kondisi sebelum masuk ke **while** sehingga kondisi ini benar (terpenuhi) dan pengulangan bisa dilaksanakan. Bila tidak, kemungkinan instruksi **while** tidak bisa dijalankan. Ada instruksi diantara instruksi-instruksi yang diulang yang mengubah nilai variabel perulangan bisa berhenti, atau jumlah perulangan bisa dibatasi. Bila tidak ada kemungkinan perulangan berlangsung terus tak berhingga.

Harus ada instruksi diantara instruksi yang diulang sehingga pada akhirnya dapat mengubah kondisi menjadi terpenuhi dan perulangan berhenti. Apabila diawal pelaksanaan kondisi sudah terpenuhi maka instruksi-instruksi paling tidak dikerjakan satu kali.

#### **Contoh Program :**

```
%contoh1
fak=1;
i=1;
while (i<5)
    fak=fak*i;
    i=i+1;
end
disp('faktorial dari 5 adalah = ')
disp(fak)
```

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

```
%contoh2
clear all
clc
a=1;
b=1;
while (b<=10);
    a=a*b;
    b=b+1;
end
disp('faktorial dari 10 (10!) adalah : ')
disp(a);
```

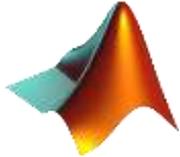
## 2. PERULANGAN `repeat - until`

Bentuk umum

```
repeat
    .....
    instruksi-instruksi
    .....
Until (kondisi)
```

Makna dari bentuk instruksi tersebut adalah : ulangi pelaksanaan instruksi instruksi hingga kondisi terpenuhi. Instruksi instruksi akan diulang hanya apabila kondisi tidak terpenuhi, dan ketika kondisi terpenuhi maka perulangan terhenti. Instruksi instruksi di kerjakan terlebih dahulu sebelum kondisi diperiksa. Harus ada instruksi yang mendahului `repeat/until` agar kondisi tidak terpenuhi sehingga perulangan terjadi. Harus ada instruksi diantara instruksi yang diulang sehingga pada akhirnya akan mengubah kondisi menjadi terpenuhi dan perulangan berhenti. Apabila diawal pelaksanaan kondisi sudah terpenuhi maka instruksi instruksi paling tidak di kerjakan satu kali

## 3. PERULANGAN `for`

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

Instruksi `for` digunakan untuk melakukan perulangan proses yang telah diketahui jumlahnya. Sintaksnya :

```
for variabel = mulai : interval : akhir
Perintah-perintah
end
```

Maknanya, ulangi instruksi-instruksi tersebut berdasarkan variabel perulangan mulai dari nilai awal hingga nilai akhir dengan perubahan nilai sebesar  $n$ . Variabel perulangan (`var`) harus bertipe dasar (*integer*, *real* atau *char*). Nilai awal harus lebih kecil dari akhir bila  $n > 0$  (positif). Nilai awal harus lebih besar dari akhir bila  $n < 0$  (negatif).

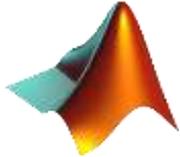
Mula-mula variabel `var` bernilai awal , kemudian setiap satu kali putaran maka nilai variabel bertambah sebesar  $n$ . Perulangan akan berhenti apabila nilai variabel sudah mencapai akhir.

#### Contoh Program :

```
%contoh1
disp ('program contoh instruksi for')
ndat=input('batas iterasi=');
    for n=1:ndat
        for m=ndat:-1:1
            a(n,m)=n^2+(5*m)
        end;
    end;
```

```
%contoh2
for x = -3:1
Y=x^2+3*x-1
Z=2*x+1
End
```

```
%contoh3
```

	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

```

for x = 1:3
A=2*x-3
For y=3:-1:1
B(x,y)=x^2-x*y+y^2
end
end
%contoh
clc
clear all
a=0;
for i=-50:100
    a=a+i;
end
sprintf('Jumlah bilangan dari -50 sampai 100 adalah = %d' , a)

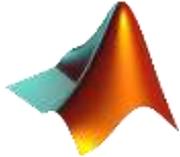
```

## B. Penugasan Kelas

1. Buatlah *script* program MATLAB yang menampilkan Nama anda sebanyak 20 kali!
2. Buatlah *script* program MATLAB menggunakan instruksi `for` atau `while` untuk menampilkan deret angka bilangan genap!

## C. Rangkuman

1. Perulangan adalah proses mengulang-ulang eksekusi blok kode tanpa henti, selama kondisi yang dijadikan acuan terpenuhi. Biasanya disiapkan variabel untuk iterasi atau variabel penanda kapan perulangan akan diberhentikan
2. Perbedaan instruksi `for` dan `while` yaitu instruksi `while` mengulang suatu proses yang belum diketahui jumlahnya. Sedangkan instruksi `for` digunakan untuk mengulang suatu proses telah diketahui jumlahnya. Pengecekan kondisi akan

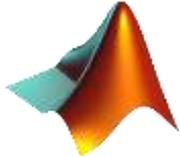
	<b>MATLAB</b>		
	<b>Chapter 3</b>	<b>Instruksi Utama</b>	

dilakukan terlebih dahulu. Jika kondisi masih bernilai *true*, maka *looping* akan terus berlanjut.

#### D. Evaluasi Formatif 3

1. Buatlah *script* program MATLAB dengan menggunakan instruksi `for` atau `while` yang menampilkan tiap huruf nama anda seperti contoh tampilan output dibawah ini!

```
>> F
>> FA
>> FAR
>> FARA
>> FARAH
```

	<b>MATLAB</b>		
	<b>Chapter 4</b>	<b>Persamaan Non Linear</b>	

### 4.1 Persamaan Non Linear

Berbagai teknik pendekatan numerik untuk masalah mendapatkan akar persamaan nonlinier. Cara termudah sudah kita perlihatkan secara sekilas yaitu dengan cara grafis. Teknik tersebut sebenarnya tidak termasuk ke dalam metode numerik, mengingat teknik ini tidak melewati serangkaian kaidah-kaidah analisis numerik. Beberapa metode numerik akan dibahas secara detail antara lain metode bagi dua (bisection), Newton-Raphson, posisi palsu (regula falsi/interpolasi linier), Secant dan metode iterasi langsung. Contoh soal juga akan diberikan untuk memberikan gambaran jelas terhadap metode yang dipelajari.

## Kegiatan Pembelajaran 1: Metode Bisection

### Kemampuan yang diharapkan

- a. Mahasiswa mampu menyelesaikan persamaan nonlinear dengan menggunakan metode Bisection

### Uraian Materi

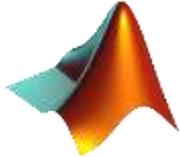
#### A. Konsep Teori

Metode bagi dua merupakan metode analisis numerik paling sederhana diantara metode-metode analisis lainnya. Metode ini termasuk metode yang robust atau tangguh. Artinya, meskipun metode ini idenya sangat sederhana namun selalu dapat menemukan akar persamaan yang dicari. Salah satu kekurangan yang dimiliki oleh metode ini adalah bahwa kita harus menentukan dua terkaan awal, yaitu  $x_a$  dan  $x_b$  yang mengurung sebuah akar persamaan yang idcari, sehingga apabila  $f_a=f(x_a)$  dan  $f_b=f(x_b)$ , maka akan dipenuhi  $f_a f_b \leq 0$  . Apabila dipenuhi  $f_a f_b = 0$  maka salah satu dari  $x_a$  dan  $x_b$  yang berada pada  $x_1$  atau keduanya merupakan akar persamaan yang dicari.

#### B. Alogaritma Metode Bisection

Langkah 1 : Pilih taksiran nilai a sebagai batas bawah interval dan taksiran nilai b sebagai batas atas interval. Jika terpenuhi kondisi :

- $f(a) \times f(b) < 0$  ; maka ada akar dalam interval, selanjutnya ke langkah 2.
- $f(a) \times f(b) > 0$  ; maka tidak ada akar dalam interval. Geser posisi interval.
- $f(a) \times f(b) = 0$  ; maka a dan b, salah satu merupakan akar.

	<b>MATLAB</b>		
	<b>Chapter 4</b>	<b>Persamaan Non Linear</b>	

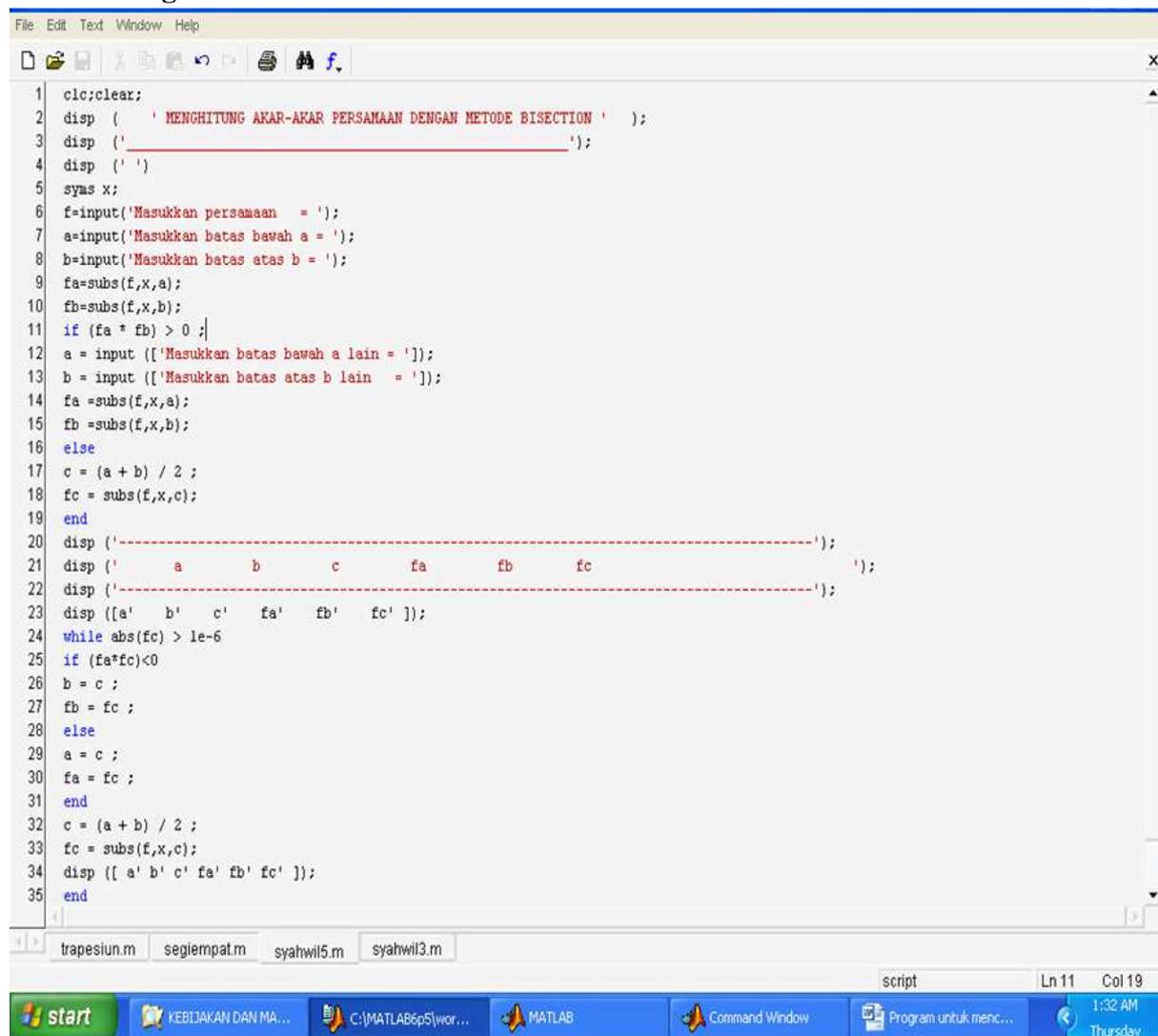
Langkah 2 : Taksiran akar yang pertama  $c$  dimana,  $c = (a + b)/2$

Langkah 3 : Evaluasi keberadaan akar, apakah dalam subinterval pertama (antara  $a$  dan  $c$ ) atau dalam subinterval kedua (antara  $c$  dan  $b$ ). Jika diperoleh :

- $f(a) \times f(c) < 0$  ; akar berada dalam subinterval pertama, maka  $b = c$ . selanjutnya ke langkah 4.
- $f(a) \times f(c) > 0$  ; akar berada dalam subinterval ke dua, maka  $a = c$ . Selanjutnya ke langkah 4.
- $f(a) \times f(c) = 0$  ;  $c$  adalah akar.

Langkah 4: Kembali ke langkah 2 dan proses hingga langkah 3

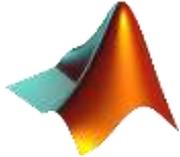
### Contoh Program :



```

1  clc;clear;
2  disp ( ' MENGHITUNG AKAR-AKAR PERSAMAAN DENGAN METODE BISECTION ' );
3  disp ( ' _____ ' );
4  disp ( ' ' )
5  syms x;
6  f=input('Masukkan persamaan = ');
7  a=input('Masukkan batas bawah a = ');
8  b=input('Masukkan batas atas b = ');
9  fa=subs(f,x,a);
10 fb=subs(f,x,b);
11 if (fa * fb) > 0 ;
12 a = input (['Masukkan batas bawah a lain = ']);
13 b = input (['Masukkan batas atas b lain = ']);
14 fa =subs(f,x,a);
15 fb =subs(f,x,b);
16 else
17 c = (a + b) / 2 ;
18 fc = subs(f,x,c);
19 end
20 disp ( '-----' );
21 disp ( '      a      b      c      fa      fb      fc      ' );
22 disp ( '-----' );
23 disp (['a'  b'  c'  fa'  fb'  fc' ]);
24 while abs(fc) > 1e-6
25 if (fa*fc)<0
26 b = c ;
27 fb = fc ;
28 else
29 a = c ;
30 fa = fc ;
31 end
32 c = (a + b) / 2 ;
33 fc = subs(f,x,c);
34 disp (['a'  b'  c'  fa'  fb'  fc' ]);
35 end

```

	<b>MATLAB</b>		
	<b>Chapter 4</b>	<b>Persamaan Non Linear</b>	

Petunjuk :

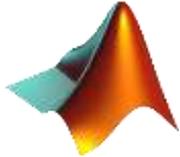
- Ketik nama file untuk program di atas pada jendela command window
- Masukkan persamaan (huruf x kecil) =  $x^2-x-6$
- Masukkan batas bawah a = -3
- Masukkan batas atas b = 2

### **B. Penugasan Kelas**

1. Buatlah program untuk metode Regula Falsi atau sekan!

### **C. Evaluasi Formatif**

1. Uji Program di atas dengan menggunakan persamaan  $f(x) = x^3 - x - 6 = 0$  masukkan batas bawah = -1 dan batas atas = 3 !

	<b>MATLAB</b>		
	<b>Chapter 5</b>	<b>Persamaan Linear</b>	

### 5.1 Persamaan Linear

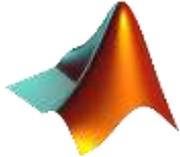
Suatu sistem persamaan linier adalah sistem persamaan yang terdiri dari sejumlah persamaan (berhingga) dan sejumlah variabel (berhingga). Mencari solusi suatu sistem persamaan linier adalah mencari nilai-nilai variabel-variabel tersebut, sehingga memenuhi semua sistem persamaan tersebut.

Aplikasi matriks yang disusun dalam bentuk matriks (SPL) meliputi aturan crammer, eliminasi gauss, invers matriks, dimana penggunaan metode-metode tersebut digunakan dalam teknik kimia untuk menyelesaikan persamaan neraca massayang penyelesaiannya sesuai dengan sifat-sifat operasi matriks. Persamaan linear digunakan untuk menyelesaikan neraca massa dan energi, tapi aplikasi terbesar dalam penyelesaian persamaan linear adalah mengimplementasikan metoda numerik lainnya.

Metode penyelesaian sistem persamaan linear dengan determinan ( aturan Cramer) tidak praktis untuk sistem yang besar. Ada beberapa metode penyelesaian praktis sistem persamaan linear antara lain:

- Metode eliminasi Gauss
- Metode eliminasi Gauss – Jordan
- Metode matriks balikan
- Metode dekomposisi LU
- Metode lelaran Jacobi
- Metode lelaran Gauss – Seidel

Walaupun metode persamaan SPL beragam, namun sebagian besar metode tersebut, terutama metode 1 sampai 4, tetap didasarkan kepada metode yang paling dasar, yaitu metode eliminasi Gauss. Metode eliminadi Gauss – Jordan, meetode matriks balikan, dan metode dekomosisi LU merupakan bentuk varisai lain dari metode eliminasi Gauss. sedangkan metode lelaran Jacobi dan metode lelaran Gauss – Seidel dikembangkan dari gagasan metode lelaran pada solusi persamaan non linear. Adapun metode-metode yang digunakan untuk penyelesaian Sistem Persamaan Linier (SPL) yang akan dibahas adalah Metode Eliminasi Gauss dan Metode Thomas.

	<b>MATLAB</b>		
	<b>Chapter 5</b>	<b>Persamaan Linear</b>	

### *Metoda Eliminasi Gauss*

**Eliminasi Gauss** adalah suatu metode untuk mengoperasikan nilai-nilai di dalam matriks sehingga menjadi matriks yang lebih sederhana lagi. Dengan melakukan operasi baris sehingga matriks tersebut menjadi matriks yang baris. Ini dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks. Caranya dengan mengubah persamaan linear tersebut ke dalam matriks teraugmentasi dan mengoperasikannya. Setelah menjadi matriks baris, lakukan substitusi balik untuk mendapatkan nilai dari variabel-variabel tersebut.

### **Kelebihan dan Kekurangan**

Metode ini digunakan dalam analisis numerik untuk meminimalkan mengisi selama eliminasi, dengan beberapa tahap

Keuntungan :

- Menentukan apakah sistem konsisten
- Menghilangkan kebutuhan untuk menulis ulang variabel setiap langkah
- Lebih mudah untuk memecahkan

kelemahan :

- memiliki masalah akurasi saat pembulatan desimal

### **Contoh**

Selesaikan sistem persamaan berikut dengan metode Eliminasi Gauss

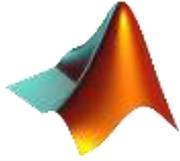
$$x + y + 2z = 9$$

$$2x + 4y - 3z = 1$$

$$3x + 6y - 5z = 0$$

### **Contoh Program**

```
clc;
clear;
disp('Solusi dari persamaan: x + y + 2z = 9')
disp(' 2x+4y - 3z = 1')
disp(' 3x+6y - 5z = 0')
disp('Menggunakan Metode Eliminasi Gauss')
A=[1 1 2 9;2 4 -3 1;3 6 -5 0]
disp('Baris 2 = -2 kali Baris 1 + Baris 2')
A(2,:)= -2*A(1,:)+A(2,:)
disp('Baris 3 = -3 kali Baris 1 + Baris 3')
A(3,:)= -3*A(1,:)+A(3,:)
disp('Baris 2 = Baris 2 bagi 2')
```



# MATLAB



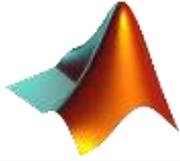
## Chapter 5

## Persamaan Linear

```
A(2,:) = A(2, :)/2  
disp('Baris 3 = -3 Baris 2 + Baris 3')  
A(3,:) = -3*A(2, :)+A(3, :)  
disp('Baris 3 = -2 kali Baris 3')  
A(3,:) = -2*A(3, :)  
z = A(3, 4)  
y = A(2, 4) - z*A(2, 3)  
x = A(1, 4) - (A(1, 2)*y + A(1, 3)*z)
```

Kemudian pilih Save and Run, Maka hasil Output nya seperti tampilan dibawah ini:

```
MATLAB  
File Edit Debug Desktop Window Help  
C:\User\nt\Doc  
Shortcuts How to Add What's New  
Solusi dari persamaan:  $x + y + 2z = 9$   
 $2x + 4y - 3z = 1$   
 $3x + 6y - 5z = 0$   
Menggunakan Metode Eliminasi Gauss  
A =  
1 1 2 9  
2 4 -3 1  
3 6 -5 0  
Baris 2 = -2 kali Baris 1 + Baris 2  
A =  
1 1 2 9  
0 2 -7 -17  
3 6 -5 0  
Baris 3 = -3 kali Baris 1 + Baris 3  
A =  
1 1 2 9  
0 2 -7 -17  
0 3 -11 -27  
Baris 2 = Baris 2 bagi 2  
A =  
1.0000 1.0000 2.0000 9.0000  
0 1.0000 -3.5000 -8.5000
```



```
MATLAB
File Edit Debug Desktop Window Help
C:\Users\int\Doc
Shortcuts How to Add What's New

Baris 3 = -3 Baris 2 + Baris 3
A =
    1.0000    1.0000    2.0000    9.0000
         0    1.0000   -3.5000   -6.5000
         0         0   -0.5000   -1.5000

Baris 3 = -2 kali Baris 3
A =
    1.0000    1.0000    2.0000    9.0000
         0    1.0000   -3.5000   -6.5000
         0         0    1.0000    3.0000

z =
     3

y =
     2

x =
     1

>>
Start OVR
```

### **Metoda Eliminasi Gauss Jordan**

Eliminasi Gauss-Jordan (EGJ), prinsipnya: mirip sekali dengan metode Eliminasi Gauss, namun dalam metode ini jumlah operasi numerik yang dilakukan jauh lebih besar, karena matriks A. mengalami inversi terlebih dahulu untuk mendapatkan matriks identitas (I). Karena kendala tersebut, maka metode ini sangat jarang dipakai, namun sangat bermanfaat untuk menginversikan matriks.

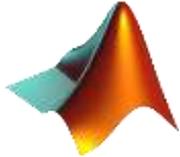
### **Contoh**

Selesaikan sistem persamaan berikut dengan metode Eliminasi Gauss

$$x + y + 2z = 9$$

$$2x + 4y - 3z = 1$$

$$3x + 6y - 5z = 0$$

	<b>MATLAB</b>		
	<b>Chapter 5</b>	<b>Persamaan Linear</b>	

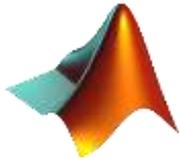
### Contoh Program

```

clc;
clear;
disp('Solusi dari persamaan:      x + y + 2z = 9')
disp('                          2x+4y - 3z = 1')
disp('                          3x+6y - 5z = 0')
disp('Menggunakan Metode Eliminasi Gauss-Jordan')
A=[1 1 2 9;2 4 -3 1;3 6 -5 0]
    disp('Baris 1 = Baris 1 bagi baris 1 kolom 1')
A(1,:)=A(1,+)/A(1,1)
disp('Baris 2 = - Baris 2 kolom 1 kali baris 1 + Baris 2')
A(2,:)=-A(2,1)*A(1,)+A(2,:)
disp('Baris 3 = - Baris 3 kolom 1 kali baris 1 + Baris 3')
A(3,:)=-A(3,1)*A(1,)+A(3,:)
disp('Baris 2 = Baris 2 bagi baris 2 kolom 2')
A(2,:)=A(2,+)/A(2,2)
disp('Baris 3 = - Baris 3 kolom 2 kali baris 2 + Baris 3')
A(3,:)=-A(3,2)*A(2,)+A(3,:)
disp('Baris 3 = Baris 3 bagi baris 3 kolom 3')
A(3,:)=A(3,+)/A(3,3)
disp('Baris 1 = - Baris 1 kolom 2 kali baris 2 + Baris 1')
A(1,:)=-A(1,2)*A(2,)+A(1,:)
disp('Baris 2 = - Baris 2 kolom 3 kali baris 3 + Baris 2')
A(2,:)=-A(2,3)*A(3,)+A(2,:)
disp('Baris 1 = - Baris 1 kolom 3 kali baris 3 + Baris 1')
A(1,:)=-A(1,3)*A(3,)+A(1,:)
disp('Dengan demikian, diperoleh:')
x1=A(1,4)
x2=A(2,4)
x3=A(3,4)

```

Kemudian pilih Save and Run, Maka hasil Output nya seperti tampilan dibawah ini:



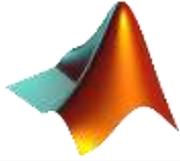
# MATLAB



## Chapter 5

## Persamaan Linear

```
MATLAB
File Edit Debug Desktop Window Help
latihan MATLAB
Shortcuts How to Add What's New
0 3 -11 -27
Baris 2 = Baris 2 bagi baris 2 kolom 2
A =
    1.0000    1.0000    2.0000    9.0000
         0    1.0000   -3.5000   -8.5000
         0    3.0000  -11.0000  -27.0000
Baris 3 = - Baris 3 kolom 2 kali baris 2 + Baris 3
A =
    1.0000    1.0000    2.0000    9.0000
         0    1.0000   -3.5000   -8.5000
         0         0   -0.5000   -1.5000
Baris 3 = Baris 3 bagi baris 3 kolom 3
A =
    1.0000    1.0000    2.0000    9.0000
         0    1.0000   -3.5000   -8.5000
         0         0    1.0000    3.0000
Baris 1 = - Baris 1 kolom 2 kali baris 2 + Baris 1
A =
    1.0000         0    5.5000   17.5000
         0    1.0000   -3.5000   -8.5000
         0         0    1.0000    3.0000
```



```
MATLAB
File Edit Debug Desktop Window Help
Shortcuts How to Add What's New
Baris 2 = - Baris 2 kolom 3 kali baris 3 + Baris 2
A =
    1.0000    0    5.5000   17.5000
    0    1.0000    0    2.0000
    0    0    1.0000    3.0000
Baris 1 = - Baris 1 kolom 3 kali baris 3 + Baris 1
A =
    1    0    0    1
    0    1    0    2
    0    0    1    3
Dengan demikian, diperoleh:
x1 =
    1
x2 =
    2
x3 =
    3
>>
Start OVR
```

### B. Penugasan Kelas

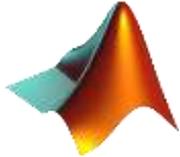
1. Buatlah program untuk metode Eliminasi Gauss!

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

### C. Evaluasi Formatif

1. Buatlah program untuk metode Eliminasi Gauss Jordan!

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

	<b>MATLAB</b>		
	<b>Chapter 6</b>	<b>Interpolasi</b>	

## 6.1 Interpolasi

Dalam bidang matematika [analisis numeris](#), **interpolasi** adalah metode menghasilkan titik-titik data baru dalam suatu jangkauan dari suatu set diskret data-data yang diketahui.

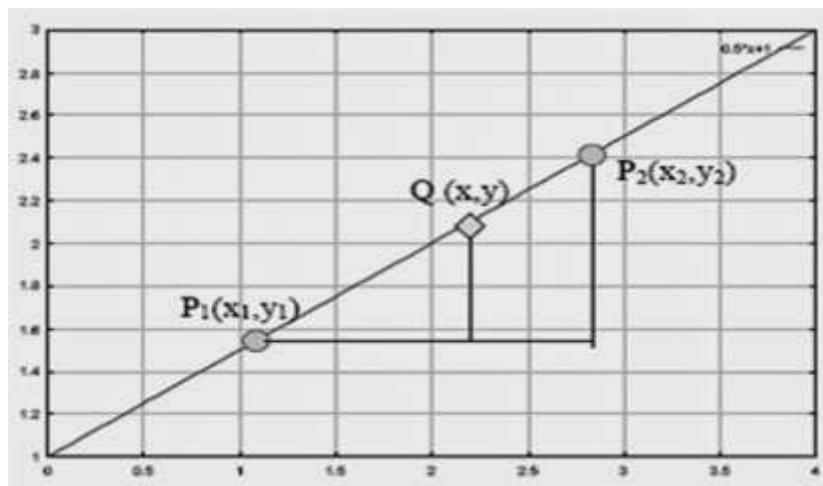
Sebuah permasalahan berbeda yang berhubungan dekat dengan interpolasi adalah pendekatan/aprosimasi suatu fungsi kompleks melalui suatu fungsi sederhana. Seandainya formula untuk suatu fungsi tertentu diketahui namun terlalu rumit untuk dinilai secara efisien, maka beberapa titik data yang diketahui dari fungsi asli tersebut dapat digunakan untuk menghasilkan suatu interpolasi berdasarkan suatu fungsi yang lebih sederhana. Tentu saja, ketika suatu fungsi yang lebih sederhana digunakan untuk memperkirakan titik data dari fungsi asli, biasanya muncul kesalahan interpolasi; namun tergantung pada domain masalahnya dan pada metode interpolasi yang digunakannya, keuntungan dari kesederhanaan/kemudahannya lebih menguntungkan daripada hasil berkurangnya keakuratan.

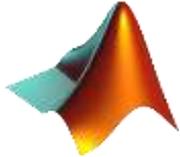
Mempelajari berbagai metode Interpolasi yang ada untuk menentukan titik-titik antara dari  $n$  buah titik dengan menggunakan suatu fungsi pendekatan tertentu. Metode Interpolasi ada beberapa macam yaitu:

- a) Interpolasi Linier
- b) Interpolasi Kuadratik
- c) Interpolasi Polinomial

### *Interpolasi Linier*

Menentukan titik-titik antara 2 buah titik dengan menggunakan pendekatan fungsi garis lurus.



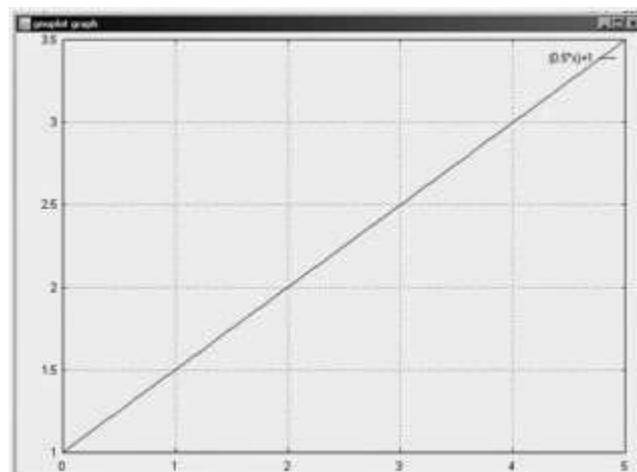
	<b>MATLAB</b>		
	<b>Chapter 6</b>	<b>Interpolasi</b>	

Persamaan garis lurus yang melalui 2 titik P1(x1,y1) dan P2(x2,y2) dapat dituliskan dengan:

$$y = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) + y_1$$

### Contoh

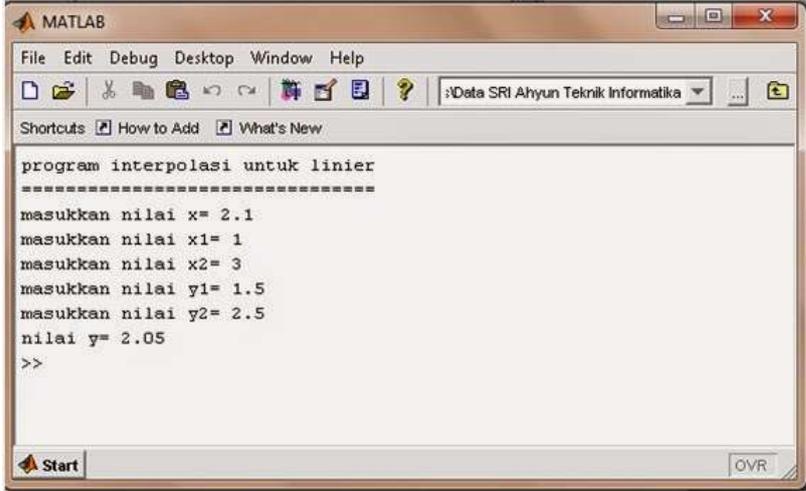
Cari nilai y untuk titik x=2.1 yang berada diantara titik(1,1.5) dan(3,2.5)



### Contoh Program

```
clear;
clc;
disp('program interpolasi untuk linier');
disp('=====');
x=input('masukkan nilai x= ');
x1=input('masukkan nilai x1= ');
x2=input('masukkan nilai x2= ');
y1=input('masukkan nilai y1= ');
y2=input('masukkan nilai y2= ');
y=((y2-y1)/(x2-x1))*(x-x1)+y1;
disp(['nilai y= ', num2str(y)]);
```

Kemudian pilih Save and Run, Maka hasil Output nya seperti tampilan dibawah ini:



```

MATLAB
File Edit Debug Desktop Window Help
:Data SRI Ahyun Teknik Informatika
Shortcuts How to Add What's New

program interpolasi untuk linier
=====
masukkan nilai x= 2.1
masukkan nilai x1= 1
masukkan nilai x2= 3
masukkan nilai y1= 1.5
masukkan nilai y2= 2.5
nilai y= 2.05
>>
  
```

## B. Penugasan Kelas

1. Diketahui suatu nilai tabel distribusi 'Student t' sebagai berikut :

$$t_{5\%} = 2,015$$

$$t_{2,5\%} = 2,571$$

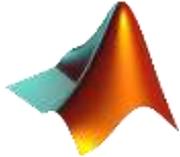
Berapa  $t_{4\%} = ?$

## C. Evaluasi Formatif

1. Jarak yang dibutuhkan sebuah kendaraan untuk berhenti adalah fungsi kecepatan. Data percobaan berikut ini menunjukkan hubungan antara kecepatan dan jarak yang dibutuhkan untuk menghentikan kendaraan.

Kecepatan (mil/jam)	10	20	30	40	50	60	70
Jarak henti (feet)	12	21	46	65	90	111	148

- Perkirakan jarak henti yang dibutuhkan bagi sebuah kendaraan yang melaju dengan kecepatan 45 mil/jam.

	<b>MATLAB</b>		
	<b>Chapter 7</b>	<b>Regresi Linear</b>	

### 7.1 Regresi Linear

Analisis regresi linier sederhana adalah hubungan secara linear antara satu variabel independen (X) dengan variabel dependen (Y). Analisis ini untuk mengetahui arah hubungan antara variabel independen dengan variabel dependen apakah positif atau negatif dan untuk memprediksi nilai dari variabel dependen apabila nilai variabel independen mengalami kenaikan atau penurunan.. Data yang digunakan biasanya berskala interval atau rasio.

Rumus regresi linear sederhana sebagai berikut:

$$Y' = a + bX$$

Keterangan:

$Y'$  = Variabel dependen (nilai yang diprediksikan)

$X$  = Variabel independen

$a$  = Konstanta (nilai  $Y'$  apabila  $X = 0$ )

$b$  = Koefisien regresi (nilai peningkatan ataupun penurunan)

Nilai-nilai  $a$  dan  $b$  dapat dihitung dengan menggunakan Rumus dibawah ini :

$$a = \frac{(\sum y) (\sum x^2) - (\sum x) (\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{n(\sum xy) - (\sum x) (\sum y)}{n(\sum x^2) - (\sum x)^2}$$

Berikut ini adalah Langkah-langkah dalam melakukan Analisis Regresi Linear Sederhana :

1. Tentukan Tujuan dari melakukan Analisis Regresi Linear Sederhana
2. Identifikasikan Variabel Faktor Penyebab (Predictor) dan Variabel Akibat (Response)
3. Lakukan Pengumpulan Data
4. Hitung  $X^2$ ,  $Y^2$ ,  $XY$  dan total dari masing-masingnya
5. Hitung  $a$  dan  $b$  berdasarkan rumus diatas.
6. Buat Model Persamaan Regresi Linear Sederhana.

7. Lakukan Prediksi atau Peramalan terhadap Variabel Faktor Penyebab atau Variabel Akibat.

**Contoh 1 (Command Window)**

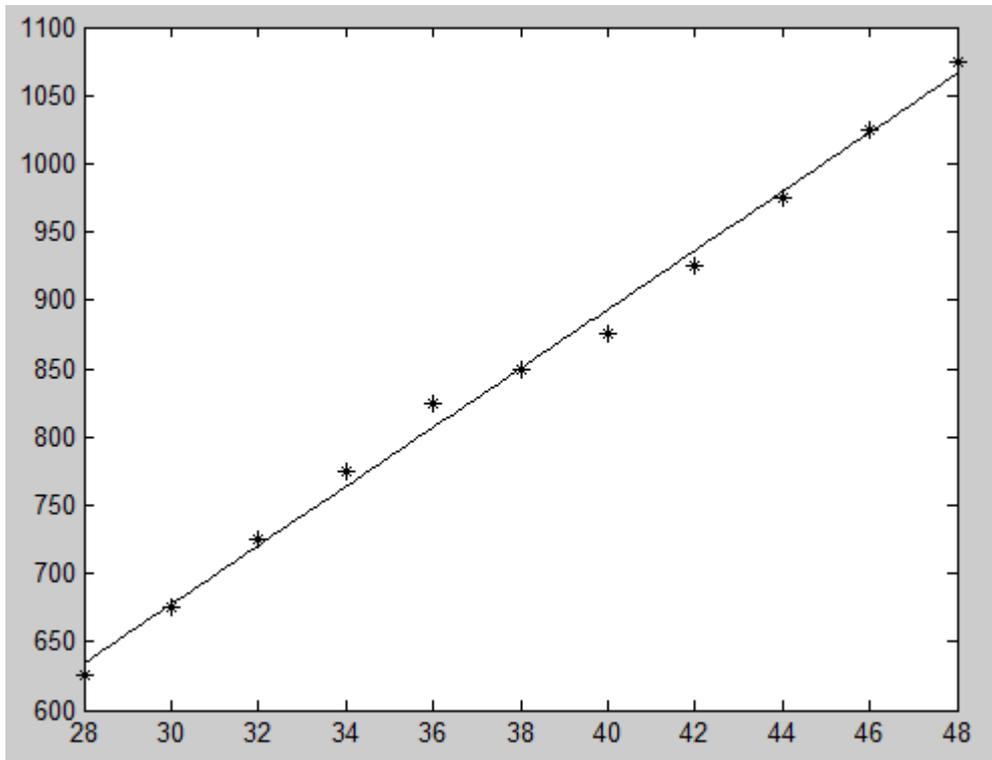
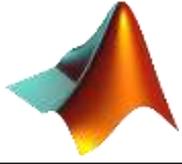
Diberikan hubungan suhu dan tegangan seperti tabel dibawah :

KONVERTER SUHU LM 35	
T (derajat C)	V (mili volt)
28	625
30	675
32	725
34	775
36	825
38	850
40	875
42	925
44	975
46	1025
48	1075

Maka langsung saja ketikkan tulisan d bawah ini pada *command window* nya matlab:

```
>> T=[28:2:48];
>> V=[625 675 725 775 825 850 875 925 975 1025 1075];
>> p=polyfit(T,V,2);
>> x=linspace(28,48,200); y=polyval(p,x);
>> plot(x,y,'k-',T,V,'k*')
>> p
```

```
p =
    -0.0000    21.5909    29.5455    {HASIL}
```



Maka, persamaan regresi liniernya adalah:  $V = 21.5909T + 29.5455$

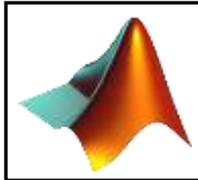
Grafik yang muncul tertampil adalah seperti di atas. Titik-titik pada gambar tersebut menunjukkan titik-titik data pada saat percobaan, sedangkan garis lurus menunjukkan grafik

**Contoh 2 (M-File)**

X (Beban Beton)	Y(Kuat Tekan Beton)
1	1/4
2	1/2
3	1
4	2
5	3/2
6	4
7	3

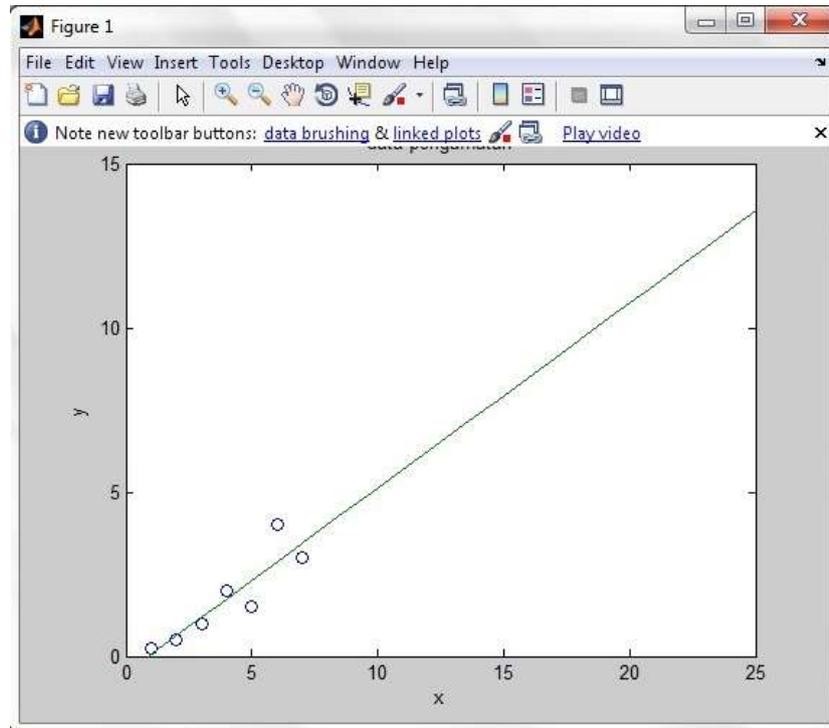
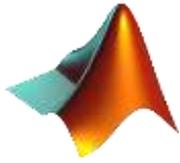
Bagaimana hubungan antara beban beton yang diberikan terhadap kuat tekan beton yang dihasilkan? Seperti apa Persamaan hubungan kedua variabel tersebut?

```
clear;clc;  
xi=[1 2 3 4 5 6 7];
```



```
yi=[1/4 1/2 1 2 3/2 4 3];  
  
Xi2=xi*xi';  
XiYi=xi*yi';  
Sx=sum(xi);  
Sy=sum(yi);  
N=7;  
a=(Xi2*Sy-Sx*XiYi)/(N*Xi2-Sx^2);  
b=(N*XiYi-Sx*Sy)/(N*Xi2-Sx^2);  
xx=0:25;  
yy=a+b*xx;  
plot(xi,yi,'o',xx,yy)  
axis([0 25 0 15])  
title('data pengamatan')  
xlabel('x')  
ylabel('y')  
fprintf('Persamaan Regresi Linear Y=[%2.4f]+[%2.4f]x\n',a,b)
```

```
Command Window  
  
a =  
    -0.5000  
  
b =  
    0.5625  
  
Persamaan Regresi Linear Y=[-0.5000]+[0.5625]x  
fx >>
```



### PEMBAHASAN PROGRAM REGRESI LINIER

```
clear;clc;
```

\*command clear untuk menghapus beberapa atau semua variabel

\*command clc untuk membersihkan command window

```
xi=[1 2 3 4 5 6 7];
```

\*Diberikan inputan sebanyak i=7, dengan dideklarasikan x=1,2,3,4,5,6,7

```
yi=[1/4 1/2 1 2 3/2 4 3];
```

\*Diberikan inputan sebanyak i=7, dengan dideklarasikan y=1/4,1/2,1,2, 3/2,4,3

```
Xi2=xi*xi';
```

```
XiYi=xi*yi';
```

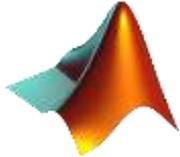
\*Rumus dari Xi2 adalah xi\*xi

\*Rumus dari XiYi adalah xi\*yi

```
Sx=sum(xi);
```

```
Sy=sum(yi);
```

\*Sx adalah jumlah dari hasil xi

	<b>MATLAB</b>		
	<b>Chapter 7</b>	<b>Regresi Linear</b>	

\* $S_y$  adalah jumlah dari hasil  $y_i$

$N=7$ ;

\*Inputan  $N$  sebanyak 7 dari data yang diketahui

$a = (X_i^2 * S_y - S_x * X_i Y_i) / (N * X_i^2 - S_x^2)$  ;

$b = (N * X_i Y_i - S_x * S_y) / (N * X_i^2 - S_x^2)$  ;

$a$

$b$

\*Untuk menampilkan hasil dari  $a$  dan  $b$

$xx=0:25$ ;

$yy=a+b*xx$ ;

\*Rumus untuk mencari suatu persamaan regresi linier adalah  $a+b*x$

`plot(xi, yi, 'o', xx, yy)`

\*Untuk menampilkan gambar dari hasil persamaan regresi linier

`axis([0 25 0 15])`

\*Sumbu untuk  $x$  diberikan sepanjang 25 dan  $y$  diberikan sepanjang 15

`title('data pengamatan')`

\*Untuk memberikan tampilan judul pada hasil plot

`xlabel('x')`

`ylabel('y')`

\*Untuk memberikan label  $x$  dan  $y$  pada hasil plot

`fprintf('Persamaan Regresi Linear Y=[%2.4f]+[%2.4f]x\n', a, b)`

\*Untuk menampilkan hasil output persamaan regresi linier  $Y=a+b*x$  pada layar command window dengan 4 angka dibelakang koma.

## B. Penugasan Kelas

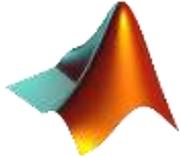
1. Diketahui suatu nilai tabel distribusi 'Student t' sebagai berikut :

$$t_5\% = 2,015$$

$$t_{2,5}\% = 2,571$$

Berapa  $t_4\% = ?$

## C. Evaluasi Formatif

	<b>MATLAB</b>		
	<b>Chapter 7</b>	<b>Regresi Linear</b>	

1. Jarak yang dibutuhkan sebuah kendaraan untuk berhenti adalah fungsi kecepatan. Data percobaan berikut ini menunjukkan hubungan antara kecepatan dan jarak yang dibutuhkan untuk menghentikan kendaraan.

Kecepatan (mil/jam)	10	20	30	40	50	60	70
Jarak henti (feet)	12	21	46	65	90	111	148

- Tentukan persamaan hubungan kedua variabel tersebut
- Perkirakan jarak henti yang dibutuhkan bagi sebuah kendaraan yang melaju dengan kecepatan 45 mil/jam.