BAB 1
GRAFIK 2 DIMENSI

Salah satu keunggulan MATLAB ialah kemampuannya dalam menampilkan/mengolah grafik dengan command yang sederhana dan fleksibel. Pada bab ini ini kita akan belajar mengenai visualisasi data plot grafik 2-dimensi. Plot grafik adalah alat yang sangat umum untuk menggambarkan hasil dalam sains. MATLAB memiliki banyak perintah yang dapat digunakan untuk membuat berbagai macam plot.

A. EASY PLOTS

Fungsi **buit-in** paling sederhana untuk mermplotkan secara eksplisit dan implisit menggunakan sebuah fungsi perintah ezplot.

No.	Fungsi	
	Eksplisit	**Implisit**
1.	`ezplot(f)` Menggambar fungsi f = f(x) dalam domain −2π< x < 2π	`ezplot(f)` Menggambar f(x,y) = 0 dalam domain -2 π < x < 2π dan -2π < y < 2π
2.	`ezplot(f,[a,b])` Menggambar f = f(x) dalam interval a < x < b	`ezplot(f,[a,b])` Menggambar f(x,y) = 0 dalam interval a < x < b and a < y < b
3.	`ezplot(f,[xmin,xmax, ymin,ymax])` Menggambar f = f(x) dalam interval xmin	`ezplot(f,[xmin,xmax, ymin,ymax])` Menggambar f(x,y)=0 dalam interval xmin

	< x < xmax dan ymin<y<ymax	< x < xmax dan ymin<y<ymax

Tabel 1.1 Perbedaan Fungsi Perintah Ezplot Secara Eksplisit dan Implisit

B. PLOT 2 DIMENSI

Untuk memvisualisasi data secara 2 dimensi ataupun 3 dimensi, kita menggunakan berbagai *command plotting*, dimana *command* yang paling dasar ialah plot. Untuk menggambar grafik dua dimensi dapat menggunakan perintah plot. Terdapat empat bentuk penulisan fungsi ini, yaitu:

- plot(x,y), menampilkan vektor *y* (sumbu vertikal) terhadap vektor *x* (sumbu horizontal).
- plot(y)menampilkan vektor *y* terhadap indeksnya.
- plot(x,y,'string')menampilkan vektor *y* terhadap vektor *x*, dengan format menurut string.
- plot(x1,y1,'string1',x2,y2,'string2',x3,y3,'string3',...)
  menciptakan sejumlah plot sekaligus dalam satu grafik.
  String menyatakan warna, bentuk penanda dan bentuk garis antar nilai. Berikut adalah beberapa nilai yang dapat digunakan pada string.

Warna		Jenis Garis		Jenis Poin	
b	Biru	.	Titik	-	Garis lurus
g	Hijau	o	Lingkaran	:	Garis titik-titik
r	Merah	x	Tanda silang	-.	Garis putus dan titik
c	Cyan	+	Tanda plus	--	Garis putus-putus
m	Magenta	*	Tanda Bintang		
y	Kuning	s	Bujursangkar		
k	Hitam	d	Diamond/permata		
w	Putih	^	Segitiga ke atas		
		v	Segitiga ke bawah		
		<	Segitiga ke kiri		
		>	Segitiga ke kanan		
		p	Segilima		
		h	Segienam		

Tabel 1.2 String Menyatakan Warna, Bentuk Penanda Dan Bentuk Garis Antar Nilai

Misalkan:

plot(x,y,'r-') memplot x versus y dengan garis lurus warna merah

plot(x,y,'k*') menempatkan tanda * warna hitam untuk setiap titik x versus y.

plot(x,y,'g--s') memplot dengan garis putus-putus warna hijau dan menempatkan tanda bujur sangkar di setiap titik x versus y.

Perlu diingat bahwa 'string' dalam plot bersifat opsional. Apabila tidak dituliskan, maka digunakan garis lurus warna biru. Jika anda tidak memilih warna dan anda menggunakan skema standar, MATLAB akan memulainya dengan warna biru dan berputar berurutan ke tujuh warna pertama pada tabel diatas untuk setiap penambahan garis. Standar *style* garis adalah garis lurus kecuali jika anda memberikan *style* garis yang lain.

Ketika Anda menggunakan command plot, gambar sebelumnya di *figure window* akan terhapus. Lalu bagaimana jika

kita ingin memplot beberapa fungsi dalam satu figure sekaligus? Dalam hal ini kita bisa gunakan command hold.

No	Fungsi	Keterangan
1.	`hold on`	Untuk 'menahan' gambar sebelumnya supaya tak terhapus ketika ditimpa gambar baru
2.	`hold off`	Untuk menonaktifkan command hold

Tabel 1.3 Fungsi Hold On Dan Hold Off

C. MENGGAMBAR GRID, KOTAK KETERANGAN, LABEL DAN LEGENDA

Beberapa fungsi untuk memberikan keterangan pada grafik, yaitu:

No	Fungsi	Keterangan
1.	`box off`	Menghilangkan kotak pada grafik
2.	`box on`	Menampilkan kotak pada grafik
3.	`box`	Menghidupkan dan mematikan kotak pada grafik
4.	`xlabel(string)`	Menambahkan teks di sumbu x pada grafik yang aktif
5.	`ylabel(string)`	Menambahkan teks di sumbu y pada grafik yang aktif
6.	`title(string)`	Menambahkan judul pada bagian atas grafik yang aktif
7.	`grid on`	Menambahkan grid pada grafik
8.	`grid off`	Menghilangkan grid pada grafik
9.	`grid`	Menghidupkan dan mematikan grid pada grafik
10.	`text(x,y,string)`	Menambahkan teks pada lokasi (x,y) di grafik yang aktif
11.	`gtext(string)`	Menempatkan teks dengan mouse
12.	`legend('ket1','ket2','ket3',…)`	Menambahkan keterangan pada grafik dengan ket1, ket2,... sebagai label

13.	`legend off`	Menghilangkan keterangan dari grafik yang aktif
14.	`legend(...,pos)`	Memberikan keterangan dan meletakannya pada posisi tetentu sesuai pos: 0 : diletakan pada posisi terbaik 1 : kanan atas (*default*) 2 : kiri atas 3 : kiri bawah 4 : kanan bawah 5 : kanan atas di luar grafik

Tabel 1.4 Fungsi Untuk Memberikan Keterangan Pada Grafik

Berbagai fungsi yang berkaitan dengan plot di atas, berlaku pula untuk plot diskrit, plot logaritmik dan plot dalam koordinat polar. Sebagai berikut:

No	Operasi	Keterangan
1.	`stem(...)`	Sama dengan plot(...), tetapi menampilkan y sebagai data diskrit
2.	`semilogy(...)`	Sama dengan plot(...), kecuali sumbu-y menggunakan skala logaritmik (basis 10)
3.	`semilogx(...)`	Sama dengan plot(...) , kecuali sumbu-x menggunakan skala logaritmik
4.	`loglog(...)`	Sama dengan plot(...), tetapi sumbu-x dan sumbu-y menggunakan skala logaritmik
5.	`polar(theta,rho,'string')`	Membuat plot dalam koordinat polar dari sudut theta (satuan radian) versus radius rho, dengan string ditentukan

Tabel 1.5 Fungsi Plot Diskrit, Plot Logaritmik Dan Plot Dalam Koordinat Polar

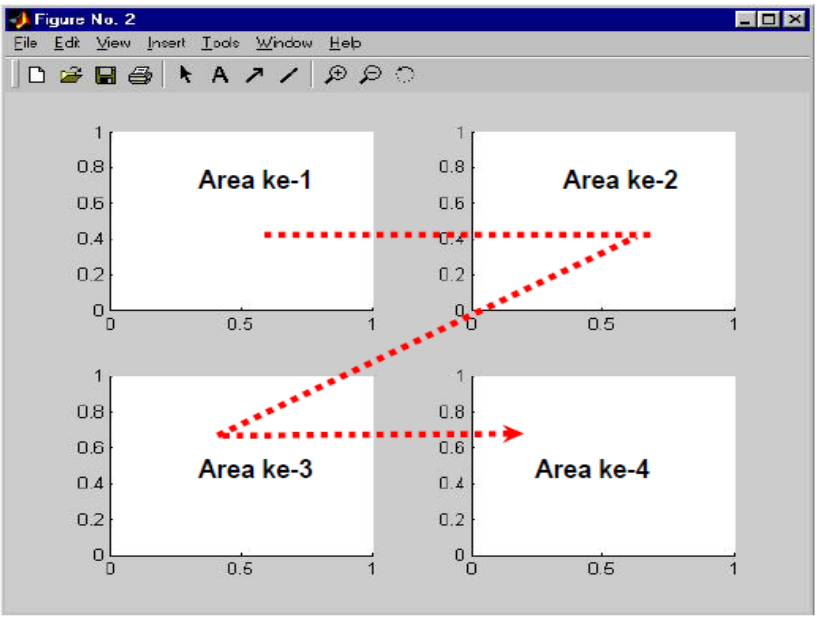
D. FIGURE DAN SUBPLOT

Figure adalah jendela untuk menampilkan gambar grafik yang anda buat seperti perintah plotdan sejenisnya. Suatu figure dapat terdiri lebih dari satu grafik. Anda mungkin ingin memplot beberapa fungsi dalam beberapa figure window yang terpisah, atau membagi satu window menjadi sejumlah grafik. Beberapa command di bawah ini bisa digunakan untuk tujuan tersebut.

No	Operasi	Keterangan
1.	`figure`	Menciptakan *figure window* baru yang kosong dan siap untuk diplot
2.	`figure(k)`	Untuk 'menduduki' *figure window* nomor-k
3.	`subplot(m,n,k)`	Membagi *figure window* menjadi m-baris × n-kolom grafik yang terpisah, dan menduduki area ke-k
4.	`clf`	"*clear figure*", mengosongkan *figure window* yang sedang 'diduduki'

Tabel 1.6 Fungsi Figure dan Subplot

Misalkan figure window berikut dibagi menjadi 2-baris × 2-kolom dengan subplot. Perhatikan urutan nomor area dari kiri-atas ke kanan-bawah.



Gambar 1.1 Pembagian Grafik Dengan "subplot"

E. GRAFIK KHUSUS 2 DIMENSI

MATLAB menyediakan beberapa fungsi yang berguna untuk membuat grafik 2-D khusus. Beberapa fungsi tersebut adalah fill, polar, bar, stairs, hist, pie, stem, compass, feather, dan errorbar. Sebagai berikut:

No	Operasi	Keterangan
1.	`fill`	Fungsi fill berguna untuk mengisi (memberi warna) ke dalam suatu kurva tertutup dalam sistem koordinat cartesius
2.	`polar`	Fungsi polar berguna untuk menggambar grafik dalam sistem koordinat polar
3.	`bar`	Fungsi bar berguna untuk membuat *bar chart* dengan kata lain fungsi bar berguna untuk

		membuat ‘persegi panjang’ di bawah kurva dengan sumbu horizontal.
4.	`stairs`	Fungsi stairs berguna untuk membuat diagram tangga, hal ini hampir sama dengan diagram bar tetapi tidak dilukiskan garis-garis di dalam.
5.	`hist`	Fungsi hist berguna untuk menggambar histogram dari distribusi statistika. Dalam hal ini banyaknya data cukup besar, suatu histogram akan mempunyai bentuk mendekati kurva normal.
6.	`pie`	Fungsi pie berguna untuk menampilkan data secara persentase, dimana setiap elemen data akan dibandingkan dengan penjumlahan seluruh data, selain itu data digambarkan dalam diagram lingkaran
7.	`stem`	Fungsi stem berguna untuk menggambar plot data diskrit
8.	`errorbar`	Fungsi errorbar berguna untuk menggambar plot grafik dengan errorbar

Tabel 1.7 Grafik Fungsi Khusus 2 Dimensi

Untuk memvisualisasi data secara 2 dimensi ataupun 3 dimensi, kita menggunakan berbagai *command plotting*, dimana *command* yang paling dasar ialah plot. Untuk menggambar grafik dua dimensi dapat menggunakan perintah plot. Terdapat empat bentuk penulisan fungsi ini, yaitu:

- plot(x,y), menampilkan vektor *y* (sumbu vertikal) terhadap vektor *x* (sumbu horizontal).
- plot(y)menampilkan vektor *y* terhadap indeksnya.
- plot(x,y,’string’)menampilkan vektor *y* terhadap vektor *x*, dengan format menurut string.

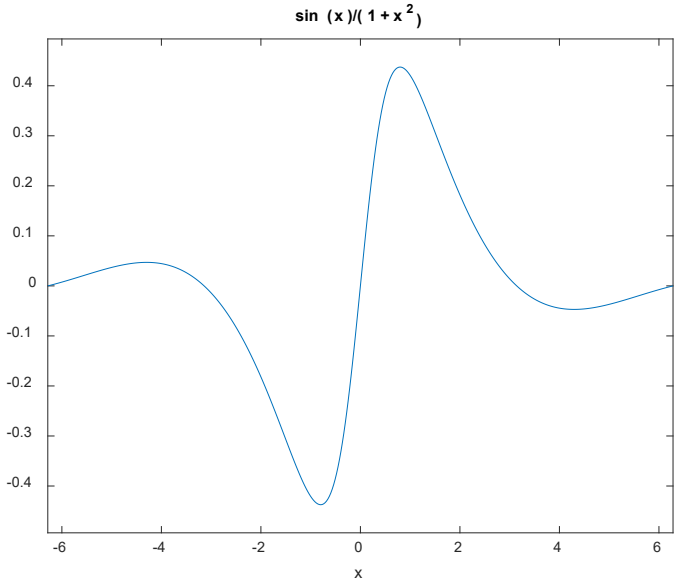
- plot(x1,y1,'string1',x2,y2,'string2',x3,y3,'string3',...)
  menciptakan sejumlah plot sekaligus dalam satu grafik.

F. EASY PLOTS MATLAB

Contoh 1.1:

Misalkan Anda ingin menggambar kurva $y = \frac{\sin(x)}{1+x^2}$. Tuliskan perintah:

```
%simpan kurva1.m
      ezplot('sin(x)/(1+x.^2)')
```



Gambar 1.1 Grafik Plot fungsi sin(x)/(1+x²)

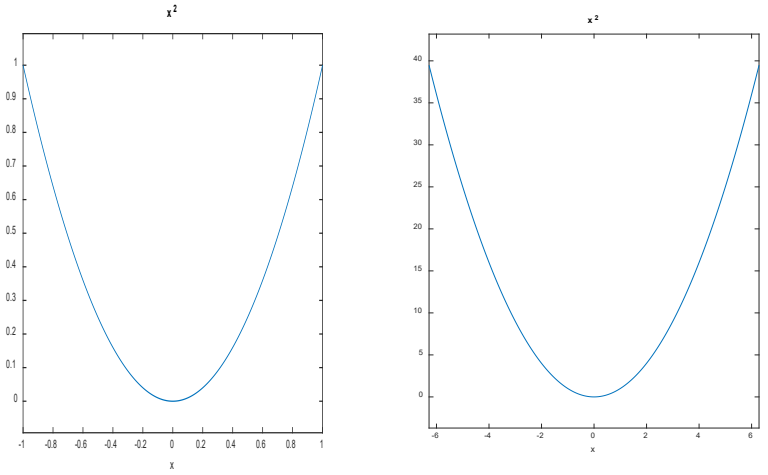
Contoh 1.2:

Misalkan kita ingin memplotkan fungsi parabola $f(x) = x^2$ pada interval $[-1;1]$. Kita dapat menggunakan perintah:

```
%simpan kurva2.m
      ezplot('x^2',[-1 1])
```

Jika interval tidak eksplisit maka digunakan interval default yaitu $[-2\pi;2\pi]$.

```
      >> ezplot('x^2')
```



Grafik 1.2 (a) interval [−1, 1]; (b) interval [−2π, 2π]

Contoh 1.3:

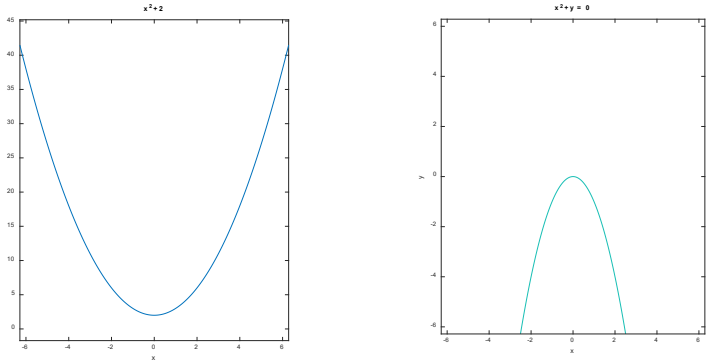
Perhatikan dua plot fungsi berikut:

```
%simpan kurva3.m
      (a) >> ezplot('x^2+2')
      (b) >> y=2; ezplot('x^2+y')
```

Pada bagian (a) fungsi memiliki konstanta yaitu 2 dan ditulis secara eksplisit di dalam fungsi. Sedangkan pada bagian (b) MATLAB akan menterjemahkan y sebagai variabel (bukan y

bernilai 2). Jadi MATLAB akan menghasilkan plot contour dari fungsi $x^2 + y = 0$.

```
>> ezplot('x^2')
```



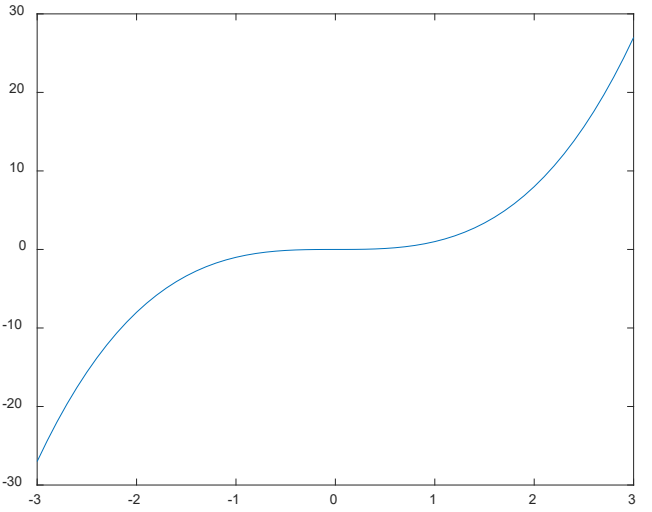
Grafik 1.3 (a) fungsi x2 + 2; (b) fungsi y = 2; x2 + y = 0

G. PLOT 2 DIMENSI MATLAB

Cobalah contoh berikut, perhatikan hasilnya, catat dan analisalah !. Kita akan memplot kurva $y = x^3$, pada rentang x = -3 hingga x = 3.

Contoh 1.4:

```
%simpan kurva4.m
      x=-3:0.1:3;    %inkremen=0.1    agar    kurva
terlihat mulus
      y=x.^3;
      plot(x,y)
```

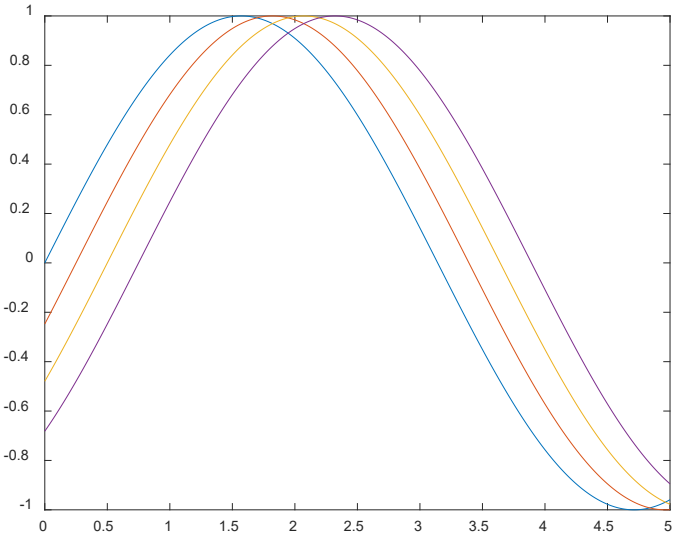


Grafik 1.4 Grafik Plot 2 Dimensi

Apa hasilnya ? Mengapa demikian ?. Cobalah contoh berikut, perhatikan hasilnya, catat dan analisalah !

Contoh 1.5:

```
%simpan kurva5.m
x=linspace(0,5,500);
y1=sin(x);
y2=sin(x-0.25);
y3=sin(x-0.5);
y4=sin(x-0.75);
plot(x,y1,x,y2,x,y3,x,y4)
```



Grafik 1.5 Beberapa Plot Dalam Satu Grafik

Sampai di sini, pahami hasil yang diberikan!. Apa yang anda temukan, catat dan analisalah!. Kemudian lanjutkan dengan perintah berikut:

```
x=linspace(0,5,500);
y1=sin(x);plot(x,y1);
hold on
y2=sin(x-0.25);plot(x,y2);
y3=sin(x-0.5);plot(x,y3);
y4=sin(x-0.75);plot(x,y4);
```

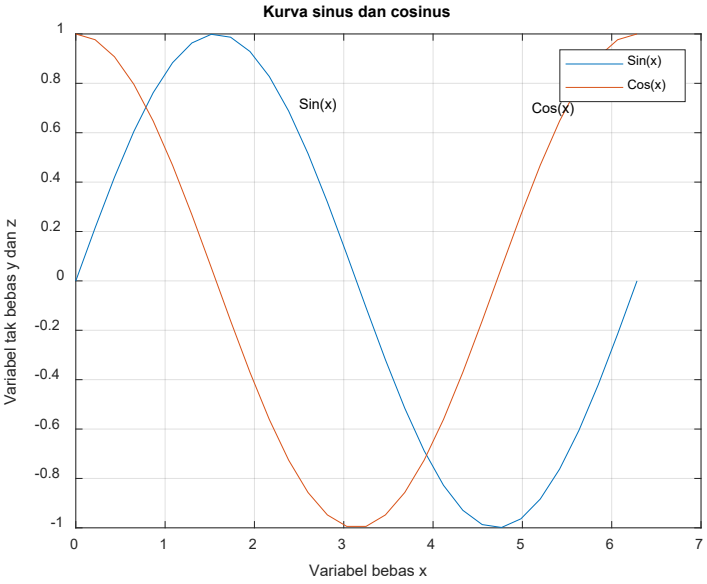
Perhatikan hasilnya, catat perbedaan-perbedaan yang anda temukan, dan analisalah!.

H. MENGGAMBAR GRID, KOTAK KETERANGAN, LABEL DAN LEGENDA MATLAB

Ketikan perintah berikut, amati perubahan yang terjadi pada grafik.

Contoh 1.6:

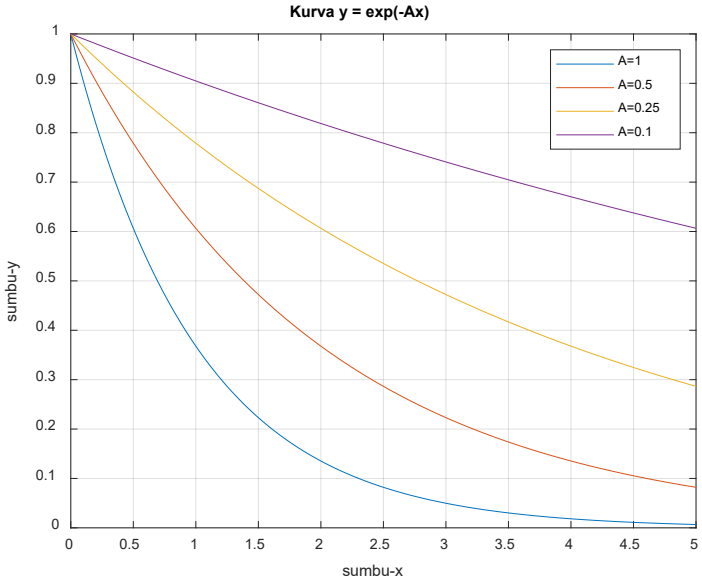
```
%simpan kurva6.m
x = linspace(0,2*pi,30);
z = cos(x);
y = sin(x);
plot(x,y,x,z)
box off
xlabel('Variabel bebas x')
ylabel('Variabel tak bebas y dan z')
title('Kurva sinus dan cosinus')
grid on, box on
text(2.5,0.7,'Sin(x)')
gtext('Cos(x)')
legend('Sin(x)','Cos(x)')
legend off
legend('Sin(x)','Cos(x)','northeast')
```



Grafik 1.6 Kurva sinus dan kosinus

Contoh 1.7:

```
      %simpan kurva7.m
clear
x=linspace(0,5,500);
y1=exp(-x); y2=exp(-0.5*x); y3=exp(-0.25*x);
y4=exp(-0.1*x);
plot(x,y1,x,y2,x,y3,x,y4)
grid on
xlabel('sumbu-x'), ylabel('sumbu-y')
title('Kurva y = exp(-Ax)')
legend('A=1','A=0.5','A=0.25','A=0.1')
```



Grafik 1.7 Kurva y=exp(-Ax)

Sampai di sini, pahami hasil yang diberikan ! Apa yang anda temukan, catat dan analisalah!. Kemudian lanjutkan dengan perintah berikut, kita coba memplot kurva tersebut dalam skala semilogaritmik

```
figure
semilogy(x,y1,x,y2,x,y3,x,y4)
grid on
xlabel('sumbu-x'), ylabel('sumbu-y')
title('Kurva y = exp(-Ax)')
legend('A=1','A=0.5','A=0.25','A=0.1')
```

Contoh 1.8:

```
%simpan kurva8.m
theta=linspace(0,2*pi,500);
```

```
rho=(cos(theta.*3)).^2;
polar(theta,rho)
```

I. FIGURE DAN SUBPLOT MATLAB

Berikut menampilkan tiga grafik yang berbeda dalam satu *figure.*

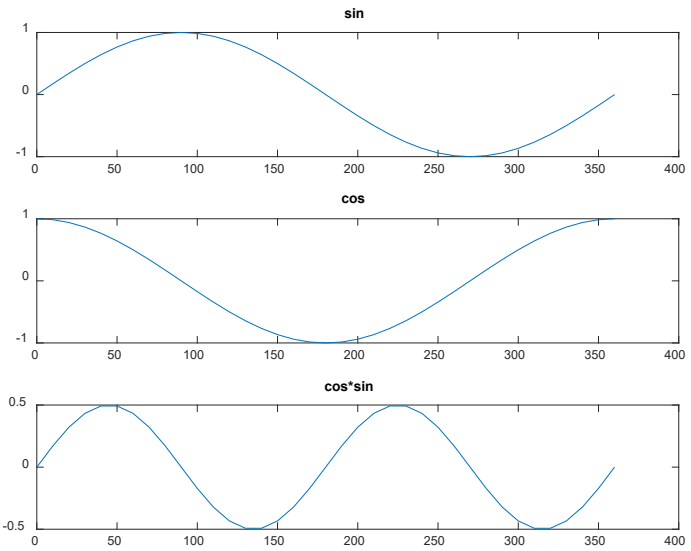
Contoh 1.9:

```
%simpan kurva9.m
clear
figure
t_deg = (0:10:360);
t_rad = t_deg*pi/180;
subplot(3,1,1); plot(t_deg,sin(t_rad));
subplot(3,1,2); plot(t_deg,cos(t_rad));
subplot(3,1,3);
plot(t_deg,cos(t_rad).*sin(t_rad));
```

Pemberian keterangan diberikan per *subplot*.

```
subplot(3,1,1); title('sin');
subplot(3,1,2); title('cos');
subplot(3,1,3); title('cos*sin');
```

Menggambar beberapa grafik dalam windows yang sama

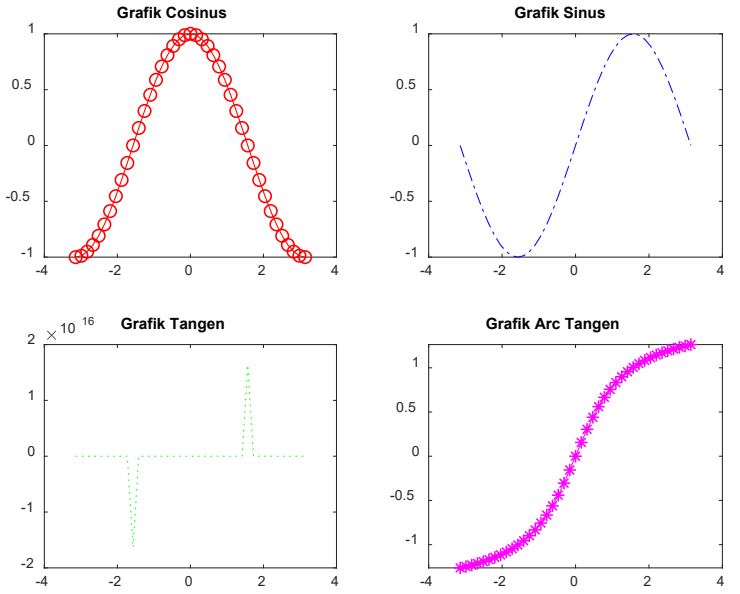


Grafik 1.8 Menggambar beberapa grafik dalam windows yang sama

Contoh 1.10:

```
%simpan kurva10.m
figure
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')
```



Grafik 1.9 Menggambar berbagai grafik dalam windows yang sama

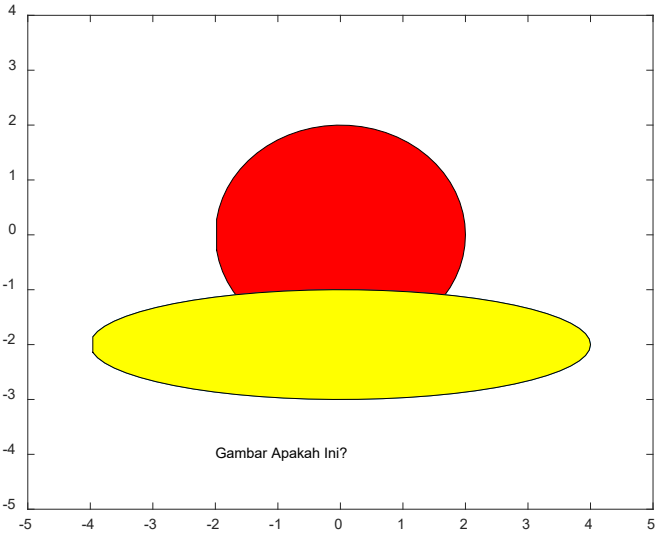
J. GRAFIK KHUSUS 2 DIMENSI MATLAB

Perhatikan hasilnya, catat perbedaan-perbedaan yang anda temukan, dan analisalah!.

Contoh 1.11:

```
%simpan kurva11.m %Grafik fungsi fill
clear
t=-3:0.1:3;
%data lingkaran
x=2*cos(t);
```

```
y=2*sin(t);
plot(x,y)
fill(x,y,'r')
%data ellips
x1=4*cos(t);
y1=-2+sin(t);
hold on
plot(x1,y1)
fill(x1,y1,'y')
axis([-5 5 -5 4])
text(-2,-4,'Gambar Apakah Ini?')
```

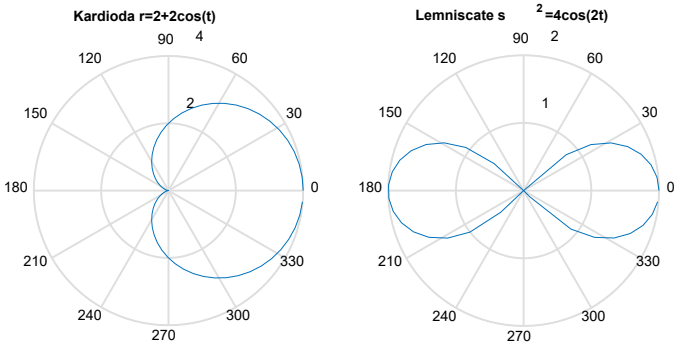


Grafik 1.10 Gambar Apakah ini

perhatikan hasil yang diperoleh!. Lanjutkan perintah berikut:

Contoh 1.12:

```
%simpan kurva12.m
%Grafik dalam koordinat polar
%Grafik kardioda dan lemniscate
clear
t=0:0.1:2*pi;
r=2+2*cos(t);              %Kardioda
s=sqrt(4*cos(2*t));        %Lemniscate
subplot(1,2,1);            %Mengaktifkan
subjendela 1
polar(t,r);                %Mengaktifkan Kardioda
title('Kardioda r=2+2cos(t)')
subplot(1,2,2)             %Mengaktifkan
subjendela 2
polar(t,s)                 %Mengaktifkan
Lemniscate
title('Lemniscate s^2=4cos(2t)')
```



Grafik 1.11 Grafik Kardioda dan Lemniscate

Apa hasilnya ? Mengapa demikian ?. Kemudian lanjutkan dengan perintah berikut:

Dari sistem koordinat polar, MATLAB mempunyai fungsi yang berguna untuk mengkonversi (mentransformasi) sistem polar ke sistem cartesius, yaitu pol2cart. Jika t menyatakan sudut, r menyatakan radius dari koordinat polar, kemudian x dan y adalah absis dan ordinat dari sistem cartesius, maka perintah :

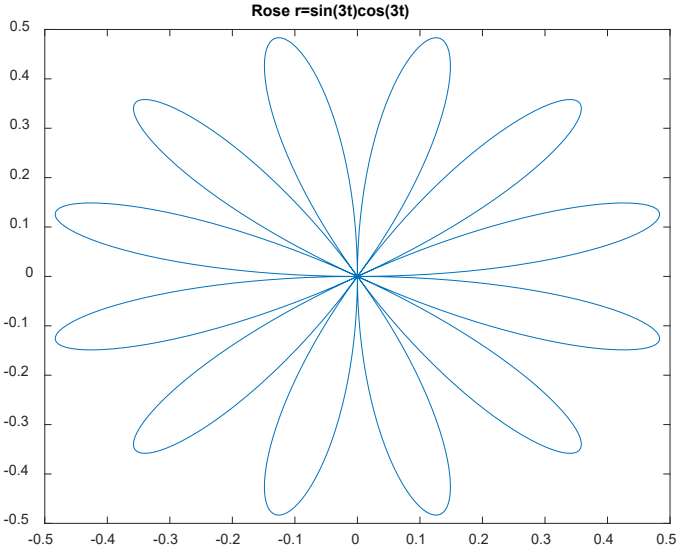
```
[x,y] = pol2cart(t,r)
plot(x,y)
```

akan menggambar grafik dengan data-data dalam sistem koordinat polar ke dalam kooordinat cartesius. Akan digambar sebuah rose (mawar) dengan persamaan r=sin(3t) *cos(3t)pada interval [0,2π]. Grafik rose pada koordinat polar akan ditunjukkan

pada perintah berikut:

Contoh 1.13:

```
%simpan kurva13.m
%Grafik rose
t=0:0.01:2*pi;
r=sin(3*t).*cos(3*t);
polar(t,r)
title('Rose r=sin(3t)cos(3t)')
[x,y]=pol2cart(t,r);
plot(x,y)
title('Rose r=sin(3t)cos(3t)')
```



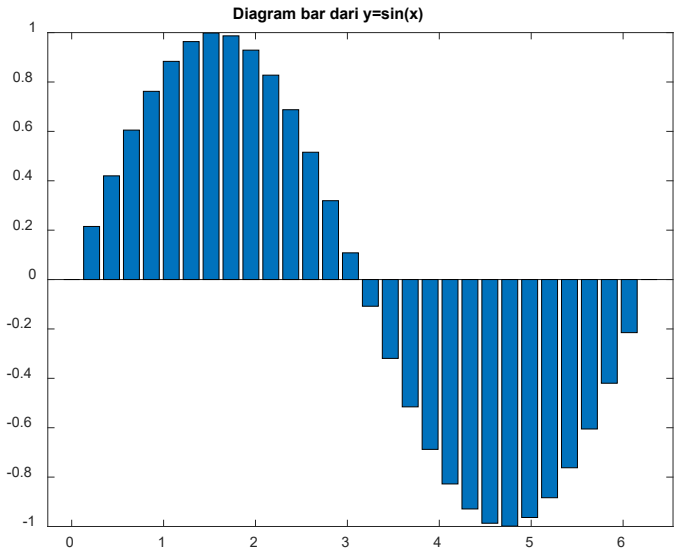
Grafik 1.12 Grafik Rose

Sampai di sini, pahami hasil yang diberikan ! Apa yang anda temukan, catat dan analisalah!. Kemudian lanjutkan dengan perintah berikut:

Contoh 1.14:

```
%simpan kurva14.m
%Contoh fungsi bar
x=linspace(0,2*pi,30);
```

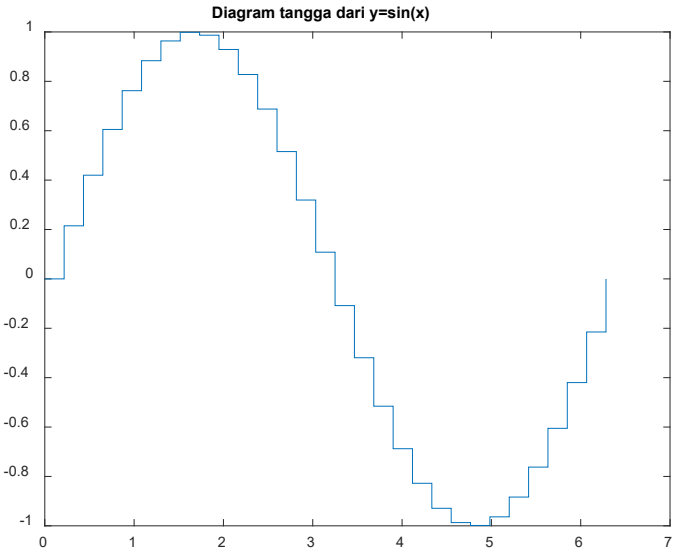
```
y=sin(x);
bar(x,y)%menggambar diagram bar
title('Diagram bar dari y=sin(x)')
```



Grafik 1.13 Diagram Bar

Contoh 1.15:

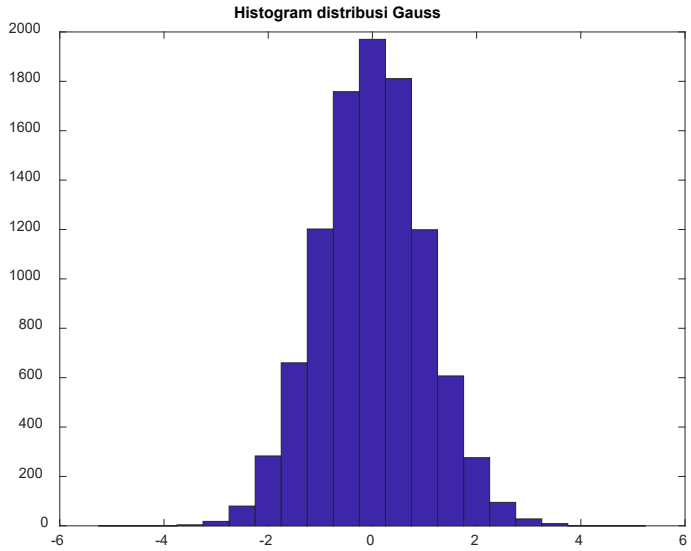
```
%simpan kurva15.m
%Contoh fungsi stairs
x=linspace(0,2*pi,30);
y=sin(x);
stairs(x,y)%menggambar diagram tangga
title('Diagram tangga dari y=sin(x)')
```



Grafik 1.14 Diagram Tangga

Contoh 1.16:

```
%simpan kurva16.m
%Contoh fungsi hist
x=-5:0.5:5;
y=randn(10000,1);%10000 data secara random
hist(y,x)%menggambar histogram
title('Histogram distribusi Gauss')
```



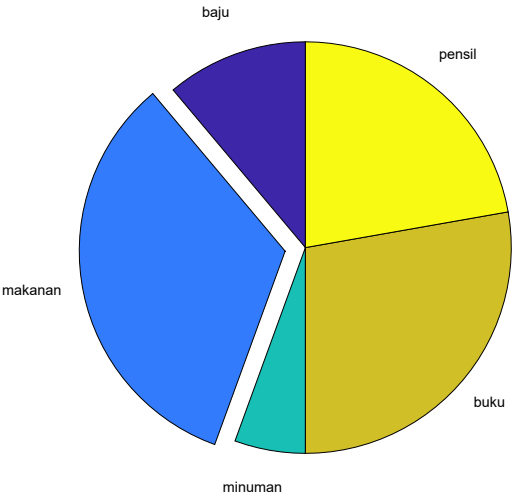
Grafik 1.15 Diagram Histogram

Sampai di sini, pahami hasil yang diberikan ! Apa yang anda temukan, catat dan analisalah!. Kemudian lanjutkan dengan perintah berikut:

Contoh 1.17:

```
%simpan kurva17.m
%Contoh fungsi pie
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'})
```

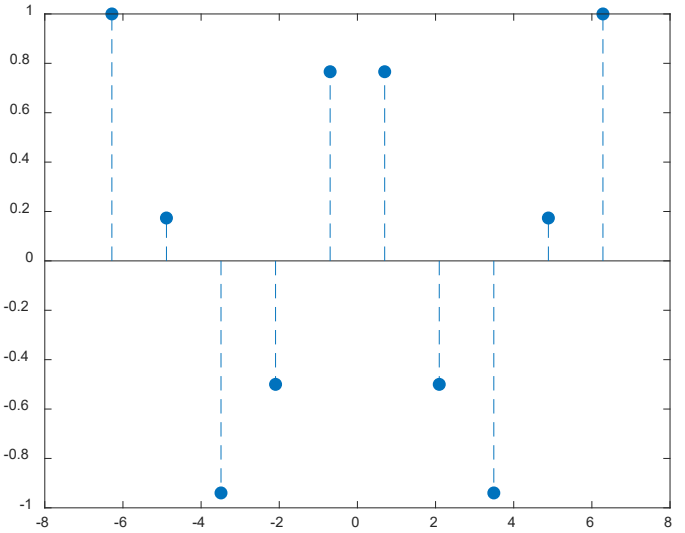


Grafik 1.16 Pie Chart

Apa hasilnya ? Mengapa demikian ?. Kemudian lanjutkan dengan perintah berikut:

Contoh 1.18:

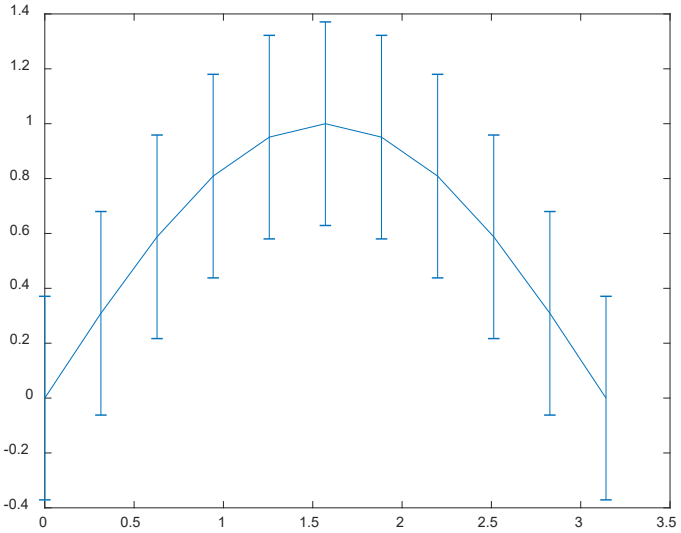
```
%simpan kurva18.m
%Contoh fungsi stem
t = linspace(-2*pi,2*pi,10);
h = stem(t,cos(t),'fill','--');
```



Grafik 1.17 Steam

Contoh 8.19:

```
%simpan kurva19.m
%Contoh fungsi errorbar
x = 0:pi/10:pi;
y = sin(x);
e = std(y)*ones(size(x));
errorbar(x,y,e)
```



Grafik 1.18 Error Bar

LATIHAN 1 MATLAB

1. Buatlah grafik dari fungsi berikut dengan menggunakan perintah ezplot

a) $f(x) = x^2 + 3\sin x - 2$ pada interval $-5 \le x \le 5$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

b) $f(x) = \dfrac{(x+5)^2}{4+x^2}$ pada interval $-3 \le x \le 5$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Buatlah kurva $y = x^4 - 9x^2$ pada interval $-6 \le x \le 6$ dengan menggunakan perintah plot

.....

.....

.....

.....

.....

.....

.....

.....

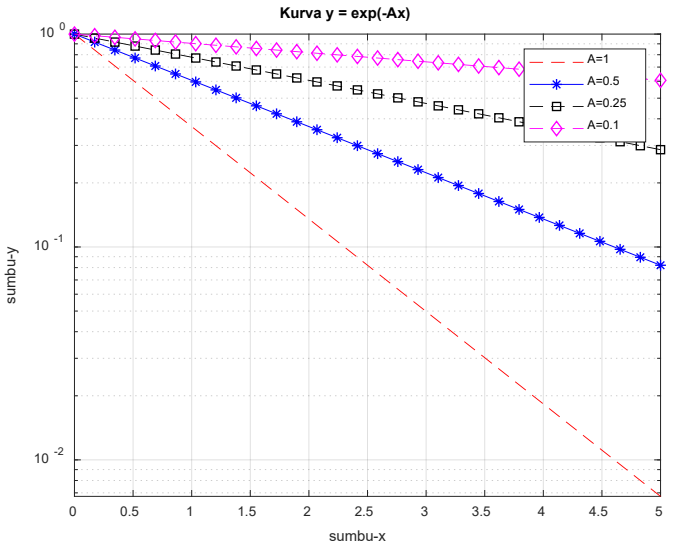
.....

.....

.....

.....

3. Perhatikan kurva berikut:



Lengkapilah sintaks berikut untuk memperoleh grafik di atas

```
clear
x=linspace(0,5,30);
y1=exp(-x); y2=exp(-0.5*x); y3=...........;
y4=...........;
figure
semilogy(x,y1,'.....',x,y2,'b*-
',x,y3,'......',x,y4,'md--')
grid on
xlabel('sumbu-x'), ylabel('sumbu-y')
title('Kurva y = exp(-Ax)')
legend('A=1','.......','A=0.25','........')
```

4. Diketahui suatu fungsi persamaan berikut dengan interval waktu $(0 < t < 20)$.

$X_1 = 2t^2 + t$

$X_2 = t^3 + 2t + 1$

$X_3 = 10t^4 - 5t^3 + t^2 + 8t + 1$

$X_4 = 5t^5 - t^4 + 10t^3 + 12t^2 + 2t + 1$

Buatlah grafik hubungan antara posisi (cm) terhadap waktu (detik) dengan menggunakan fungsi ***plot*** dan ***subplot.***

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

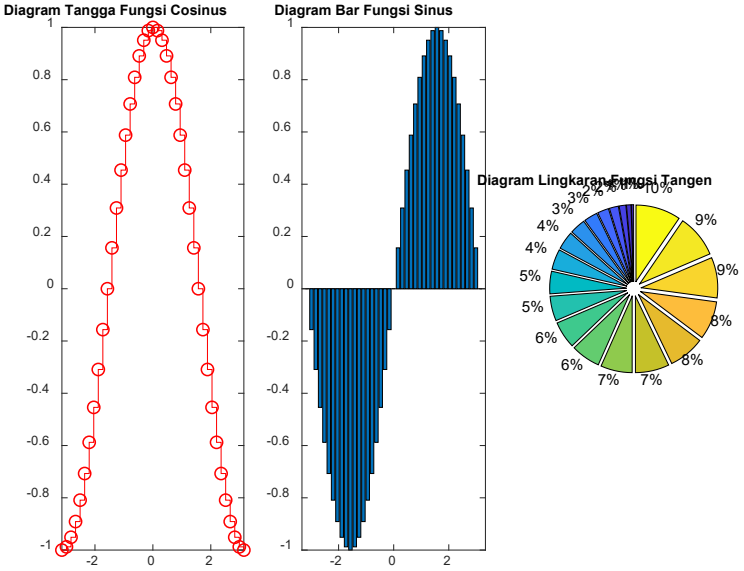
.....

.....

.....

.....

5. Perhatikan kurva berikut:



Lengkapilah sintaks berikut untuk memperoleh grafik di atas

```
figure
x = -pi:pi/20:pi;
y1=cos(x);
y2=sin(x);
.........................
subplot(1,3,1);
stairs.....................
title('Diagram Tangga Fungsi Cosinus')
.........................
bar(x,y2,'grouped')
title('Diagram Bar Fungsi Sinus')
subplot...................
pie(x,y3)
.........................
```

BAB 2
GRAFIK 3 DIMENSI

MATLAB memiliki fungsi atau perintah yang dapat menampilkan plot 3 dimensi (3D). Sebagian fungsi memiliki nama yang sama dengan fungsi untuk plot 2D dan hanya dengan menambah 3 sebagai penanda fungsi untuk 3D. Misalnya, fungsi plot garis 3D ditulis plot3, demikian juga untuk fungsi-fungsi bar3, bar3h, pie3, comet3, stem3, dan lainnya. Fungsi untuk memplot 3D antara lain mesh dan surf. MATLAB memiliki beberapa fungsi untuk membuat matriks yang digunakan untuk koordinat (x, y, z) untuk bentuk tertentu, misalnya, bola dan silinder. Dalam subbab ini akan dibahas tiga macam plot 3-dimensi: plot garis, plot permukaan (surface), dan plot kontur.

A. PLOT GARIS

Mirip dengan plot 2-dimensi, tetapi kali ini kita gunakan *command* plot3(...), dan dibutuhkan vektor **z**, untuk dimensi ketiga. Perintah plot3(x,y,z), bila x, y and z adalah tiga vektor dengan panjang sama, menggambar lintasan garis dalam ruang 3D yang melalui koordinat-koordinat x, y dan z. Untuk menggunakan fungsi plot3, kita harus membuat tiga array x, y, dan z berukuran yang sama. Vektor x dan y mengandung nilai-nilai x dan nilai y, dan vektor z mengandung nilai-nilai fungsi yang terkait dengan setiap titik x dan y. Secara umum untuk menvisualisasikan fungsi

dari dua variabel, $z = f(x, y)$:

1) Bangkitkan matriks X dan Y yang terdiri dari baris dan kolom berulang, masing-masing atas domain dari fungsi.
2) Gunakan X dan Y untuk mengevaluasi dan gambarkan grafik fungsi. Perintah/fungsi plot3 untuk menampilkan plot tiga-dimensi dari satu array titik-titik data. Perintah plot3 memiliki syntax:

No	Syntax	Keterangan
1.	`plot3(X1,Y1,Z1,...)`	X1,Y1,Z1 adalah vektor atau matriks, memplot satu atau lebih garis dalam ruang tiga dimensi melalui titik-titik yang koordinatnya adalah elemen-elemen dari X1,Y1,Z1.
2.	`plot3(X1,Y1,Z1,LineSpec,...)`	Untuk membuat dan menampilkan semua garis yang didefinisikan oleh Xn,Yn,Zn,LineSpec, dimana LineSpec adalah spesifikasi garis yang menentukan tipe garis, tanda simbol, dan warna garis plot
3.	`plot3(...,'PropertyName',PropertyValue,...)`	Himpunan properti untuk nilai properti yang ditentukan untuk semua objek garis yang dibuat oleh plot3.
4.	`h = plot3(...)`	Memberikan vektor kolom dari chart obyek garis.

Tabel 2.1 Fungsi Plot3

B. PLOT PERMUKAAN

Untuk plot permukaan (surface) dalam ruang 3 dimensi digunakan command mesh, surf, dan waterfall.

1) Fungsi Mesh

Fungsi mesh menggambarkan kerangka kawat melalui

titik-titik parametrik permukaan ditentukan oleh X, Y, dan Z, dengan warna yang ditentukan oleh C. Seperti halnya fungsi plot, fungsi mesh juga memiliki bentuk sederhana yaitu fungsi ezmesh dimana akan menggambarkan mesh atas daerah $-2\pi \le x \le 2\pi, -2\pi \le y \le 2\pi$. Perintah mesh memiliki syntax sebagai berikut:

No	Syntax	Keterangan
1.	`mesh(X,Y,Z)`	Menggambar kerangka kawat mesh dengan warna yang ditentukan oleh Z, sehingga warna proporsional untuk ketinggian permukaan. Jika X dan Y adalah vektor, panjang (X) = n dan panjang (Y) = m, di mana [m, n] = size (Z). Dalam kasus ini, (X (j), Y (i), Z (i, j)) adalah persimpangan dari garis kerangka kawat; X dan Y masing-masing sesuai dengan kolom dan baris Z. Jika X dan Y adalah matriks, (X (i, j), Y (i, j), Z (i, j)) adalah persimpangan dari garis kerangka kawat.
2.	`mesh(Z)`	Menggambar kerangka kawat mesh menggunakan X = 1: n dan Y = 1: m, dimana [m, n] = size (Z). Ketinggian, Z, adalah fungsi nilai-tunggal didefinisikan atas kotak persegi panjang.Warna sebanding dengan ketinggian permukaan.
3.	`mesh(...,C)`	Menggambar kerangka kawat mesh dengan warna yang ditentukan oleh matriks C. MATLAB melakukan transformasi linear pada data di C untuk menghasilkan warna dari colormap saat ini. Jika X, Y, dan Z adalah matriks, mereka harus memiliki ukuran yang sama seperti C.
4.	`mesh(...,'PropertyName',PropertyValue,...)`	Memberikan properti untuk nilai properti yang ditentukan untuk semua objek grafis yang dibuat oleh mesh.
5.	`meshc(...)`	Menggambar plot kontur di bidang bawah mesh.

6.	`meshz(...)`	Menggambar plot tirai sekitar mesh.

Tabel 2.2 Fungsi Mesh

2) Fungsi Surf

Fungsi surf untuk memvisualisasi fungsi matematika atas wilayah persegi panjang. Fungsi surf dan surfc membuat permukaan berwarna dengan warna yang ditentukan oleh Z atau C. Fungsi surf juga memiliki bentuk sederhana ezsurf ezmesh dimana akan menggambarkan permukaan surf atas daerah $-2\pi \le x \le 2\pi, -2\pi \le y \le 2\pi$. Perintah surf memiliki syntax sebagai berikut:

No	Syntax	Keterangan
1.	`surf(Z)`	Membuat permukaan tiga dimensi yang diarsir dari komponen z dalam matriks Z, dengan menggunakan x = 1: n dan y = 1: m, di mana [m, n] = size (Z). Ketinggian, Z, adalah fungsi nilai tunggal didefinisikan atas persegi panjang. Z menentukan data warna serta ketinggian permukaan, sehingga warna sebanding dengan ketinggian permukaan.
2.	`surf(Z,C)`	Plot ketinggian Z, fungsi nilai tunggal didefinisikan atas persegi panjang, dan menggunakan matriks C, diasumsikan memiliki ukuran yang sama Z, untuk mewarnai permukaan.
3.	`surf(X,Y,Z)`	Membuat permukaan berbayang menggunakan Z untuk data warna serta ketinggian permukaan. X dan Y adalah vektor atau matriks mendefinisikan komponen x dan y dari permukaan. Jika X dan Y adalah vektor, panjang (X) = n dan

		panjang (Y) = m, di mana [m, n] = size (Z).
4.	`surf(X,Y,Z,C)`	Menggambar permukaan berbayang, dengan warna yang didefinisikan oleh C. MATLAB melakukan transformasi linear pada data ini untuk memperoleh warna dari colormap saat ini.
5.	`surf(...,'PropertyName',PropertyValue)`	Menentukan property permukaan sesuai dengan data.
6.	`surfc(...)`	Menggambar plot kontur di bidang bawah surf.

Tabel 2.3 Fungsi Surf

3) Waterfall

Perintah waterfall menghasilkan grafik yang sama dengan perintah **mesh**, tetapi garis-garis jala hanya tampak dari arah sumbu x. Perintah waterfall memiliki syntax sebagai berikut:

No	Syntax	Keterangan
1.	`waterfall(Z)`	Menciptakan plot waterfall menggunakan x = 1:size(Z,2) dan y = 1:size(Z,1). Z menentukan warna, jadi warna secara proporsional untuk ketinggian permukaan
2.	`waterfall(X,Y,Z)`	Menghasilkan plot waterfall menggunakan nilai-nilai di X, Y dan Z. Z juga menentukan warna, jadi secara proporsional untuk ketinggian permukaan. Jika X dan Y adalah vektor, X bersesuaian dengan kolom Z, dan Y bersesuaian dengan baris, dimana length(x) = n, length(y) = m, and [m,n] = size(Z). X dan Y adalah vektor atau matriks yang didefinisikan pada koordinat z dari plot (contohnya ketinggian di bawah plane). Jika C tidak termasuk, warna proporsional ke Z.

3.	`waterfall(...,C)`	Menggunakan skala nilai warna untuk menghasilkan warna-warna dari colormap. memperkecil skala atau memperbesar skala Warna ditentukan oleh range C, dimana ukurannya harus sama dengan Z. MATLAB melakukan transformasi linier pada C untuk menghasilkan warna-warna dari colormap.

Tabel 2.4 Fungsi Waterfall

C. PLOT KONTUR

Kontur yaitu menggambar grafik garis yang sesuai dengan ketinggian bidang. Perintah kontur memiliki syntax sebagai berikut:

No	Syntax	Keterangan
1.	`contour(X,Y,Z)`	Menggambar kontur dari nilai di Z dengan 10 level. Elemen Z diterjemahkan sebagai level-level di atas bidang (x,y).
2.	`contour3`	Plot kontur dalam ruang 3 Dimensi
3.	`C = contour(X,Y,Z)`	Menghitung matriks kontur C.
4.	`contour(X,Y,Z,n)`	Menggambar kontur dengan n level.
5.	`contour(...,'string')`	Menggambar kontur dengan property yang ditentukan oleh string.
6.	`clabel(C)`	Menuliskan angka pada garis-garis kontur untuk menunjukkan level.
7.	`meshc(X,Y,Z)`	Menggambar permukaan seperti pada command mesh, dan juga memplot kontur pada dasar grafik.

Tabel 2.5 Fungsi Kontur

D. ANIMASI

Untuk membuat animasi dari plot, dapat digunakan fungsi comet untuk animasi plot 2D dan comet3 untuk animasi 3D. Selain itu juga dapat juga menggunakan fungsi getframe dam movie. Untuk moving plot, dapat dilakukan melalui fungsi timer yang dapat menangani objek dari grafik. MATLAB memiliki fungsi atau perintah yang dapat menampilkan plot 3 dimensi (3D). Sebagian fungsi memiliki nama yang sama dengan fungsi untuk plot 2D dan hanya dengan menambah 3 sebagai penanda fungsi untuk 3D. Misalnya, fungsi plot garis 3D ditulis plot3, demikian juga untuk fungsi-fungsi bar3, bar3h, pie3, comet3, stem3, dan lainnya. Fungsi untuk memplot 3D antara lain mesh dan surf. MATLAB memiliki beberapa fungsi untuk membuat matriks yang digunakan untuk koordinat (x, y, z) untuk bentuk tertentu, misalnya, bola dan silinder.

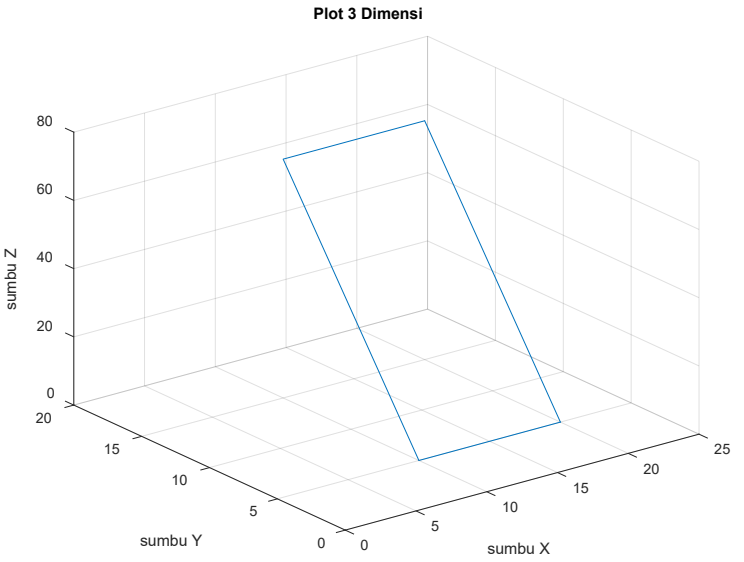
E. PLOT GARIS MATLAB

Cobalah contoh berikut, perhatikan hasilnya, catat dan analisalah !. Mari kita mulai dengan plot garis di dalam ruang 3 dimensi. Ini mirip dengan plot 2 dimensi, tetapi kali ini kita gunakan command plot3(...), dan dibutuhkan vektor z, untuk dimensi ketiga.

Contoh 2.1:

```
%simpan grafik1.m
X = [10 20 20 10 10];
Y = [5 5 15 15 5];
Z = [0 0 70 70 0];
```

```
plot3(X,Y,Z); grid on;
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
title ('Plot 3 Dimensi');
axis([0 25 0 20 0 80])
```



Grafik 2.1 Plot 3 Dimensi

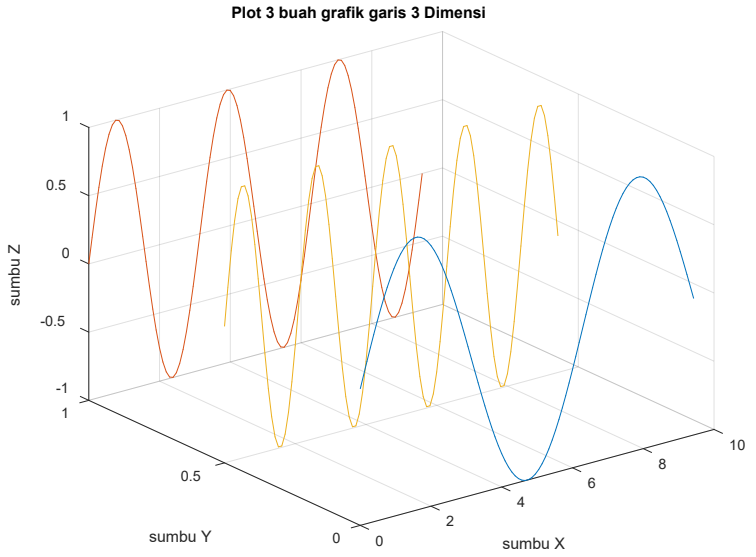
Sampai di sini, pahami hasil yang diberikan ! Apa yang anda temukan, catat dan analisalah!. Kemudian lanjutkan dengan perintah berikut:

Contoh 2.2:

Menggambar 3 buah grafik garis 3 Dimensi dalam satu figure

```
%simpan grafik2.m
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)
grid on
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
title ('Plot 3 buah grafik garis 3 Dimensi');
```



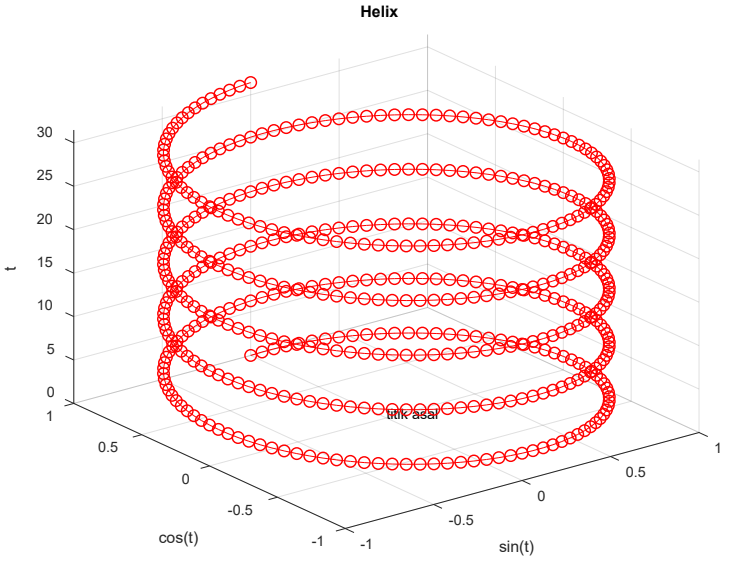
Grafik 2.2 Plot 3 Buah Garis

Apa hasilnya ? Mengapa demikian ?. Cobalah contoh berikut, perhatikan hasilnya, catat dan analisalah !

Contoh 2.3:

Misalkan kita ingin memplotkan fungsi circular helix dalam bentuk parametrik x = sin(t);y = cos(t); z = t. Kita dapat menggunakan perintah:

```
%simpan grafik3.m
t=0:pi/50:10*pi;
plot3(sin(t),cos(t),t,'-ro')
xlabel('sin(t)'),ylabel('cos(t)'),zlabel('t')
title('Helix')
text(0,0,0,'titik asal')
grid on
```

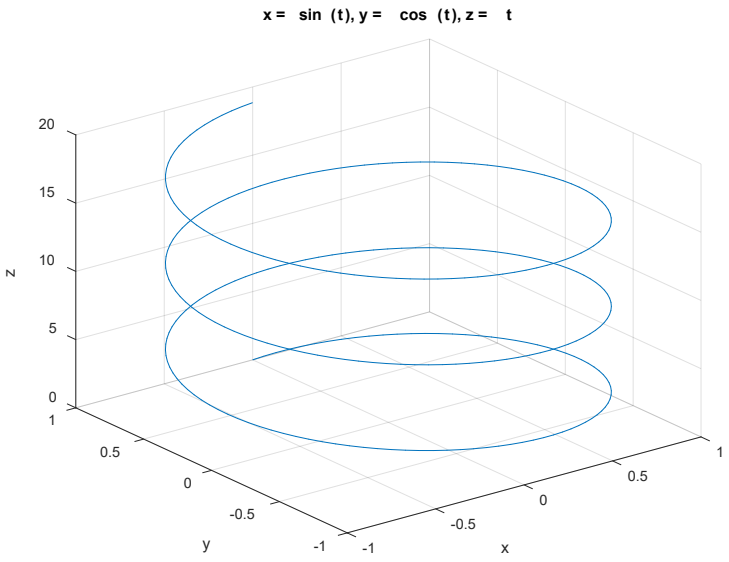


Grafik 2.3 Helix

Lanjutkan perintah berikut:

Plot kurva parametrik dengan menggunakan ezplot3.

```
ezplot3('sin(t)','cos(t)','t',[0,6*pi])
```



Grafik 2.4 x=sin(t), y=cos(t), z=t

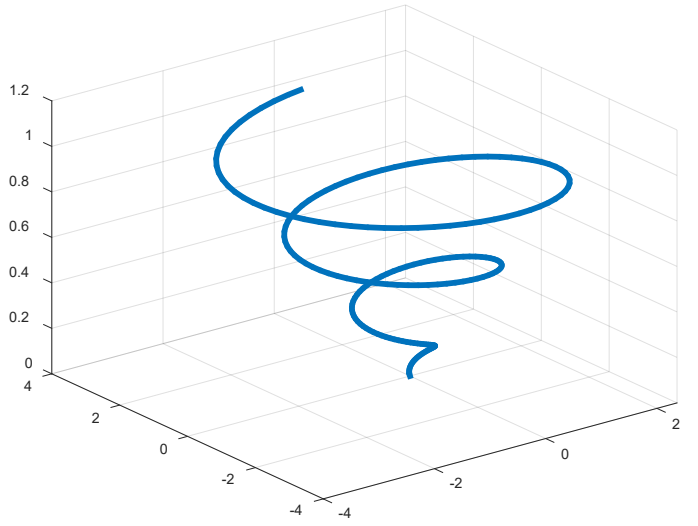
Apa hasilnya ? Mengapa demikian ?. Cobalah contoh berikut, perhatikan hasilnya, catat dan analisalah !

Lanjutkan perintah berikut:

Contoh 2.4:

Plot kurva dengan menambahkan LineSpec ketebalan garis.

```
%simpan grafik4.m
t=0:pi/200:pi;
h=plot3(t.*sin(6*t),t.*cos(6*t),t/3);
set(h,'linewidth',3);
grid on
```

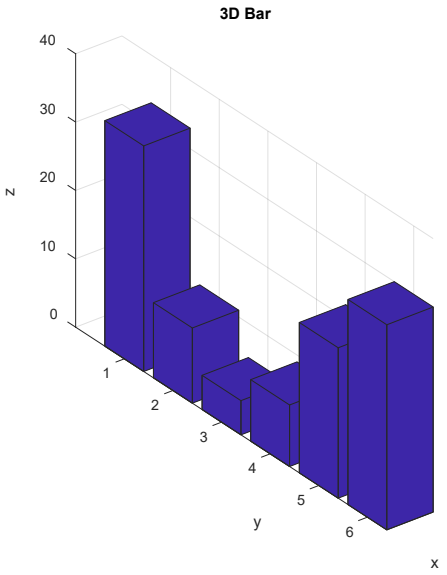


Grafik 2.5 Grafik Ketebalan

Contoh 2.5:

Plot kurva untuk menggambarkan plot diagram batang bar3 untuk berdimensi 3.

```
%simpan grafik5.m
y = 1:6;
z = [33 11 5 9 22 30];
bar3(y,z)
xlabel('x')
ylabel('y')
zlabel('z')
title('3D Bar')
```



Grafik 2.6 Grafik 3D Bar

Amatilah perbedaan hasil antara mesh, surf dan waterfall !. Perhatikan hasilnya, catat perbedaan-perbedaan yang anda temukan, dan analisalah!. Coba juga plot 3 Dimensi lainnya untuk bar3h, pie3 dan stem3!.

F. PLOT PERMUKAAN MATLAB

Untuk plot permukaan (surface) dalam ruang 3 dimensi digunakan command mesh, surf, dan waterfall. Contoh berikut ini menggambarkan fungsi dua variabel $z = x^2 + y^2$. Caranya ialah:

1) Definisikan batas-batas nilai x dan y yang akan diplot
2) Gunakan ***command*** meshgrid untuk "mengisi" bidang-XY dengan jalinan titik
3) Hitunglah fungsi 3 dimensi untuk jalinan titik tersebut

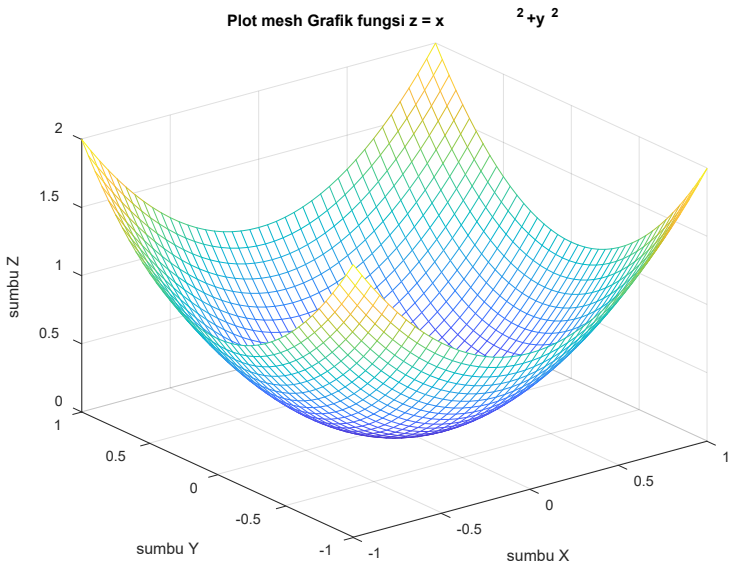
4) Buatlah plot dengan ***command*** mesh, surf, dan waterfall.

Contoh 2.6:

Cobalah contoh berikut, perhatikan hasilnya, catat dan analisalah !

Misalkan akan digambar grafik dari parabola hyperbolik $z = x^2 + y^2$ pada kotak $\{(x, y) : -1 \le x \le 1, -1 \le y \le 1\}$

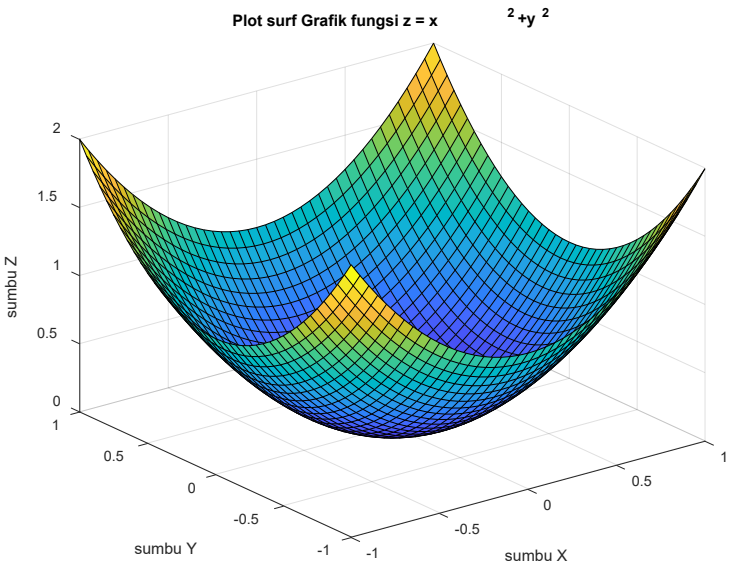
```
%simpan grafik6.m
x=-1:0.05:1;
y=x;
[X,Y]=meshgrid(x,y);
Z=Y.^2+X.^2;
mesh(X,Y,Z)
grid on
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot mesh Grafik fungsi z = x^2+y^2');
```



Grafik 2.7 Plot Mesh

Lanjutkan perintah berikut:

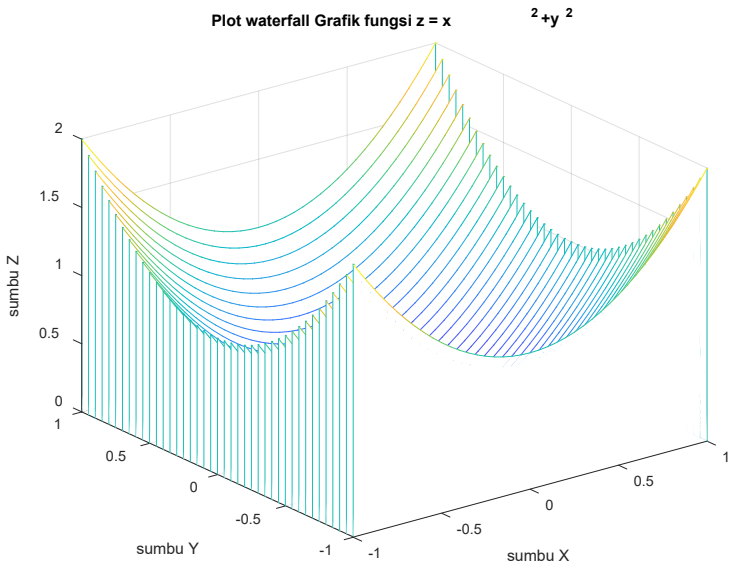
```
surf(X,Y,Z);
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot surf Grafik fungsi z = x^2+y^2');
```



Grafik 2.8 Plot Surf

Lanjutkan perintah berikut:

```
waterfall(X,Y,Z);
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot waterfall Grafik fungsi z =
x^2+y^2');
```



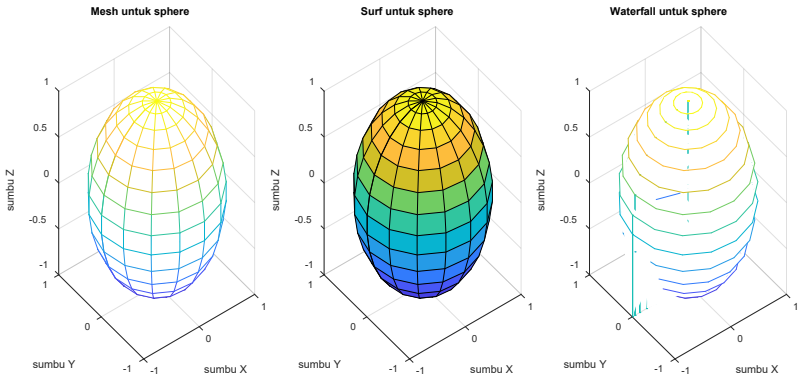
Grafik 2.9 Plot Waterfall

Amatilah perbedaan hasil antara mesh, surf dan waterfall !. Perhatikan hasilnya, catat perbedaan-perbedaan yang anda temukan, dan analisalah!.

Contoh 2.7:

```
%simpan grafik7.m
[x,y,z]= sphere(15);
subplot(1,3,1);mesh(x,y,z);title('Mesh untuk
sphere');
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
subplot(1,3,2);surf(x,y,z);title('Surf untuk
sphere');
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
```

```
subplot(1,3,3);
waterfall(x,y,z);title('Waterfall untuk
sphere');
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
```



Grafik 2.10 Plot Sphere

G. PLOT KONTUR MATLAB

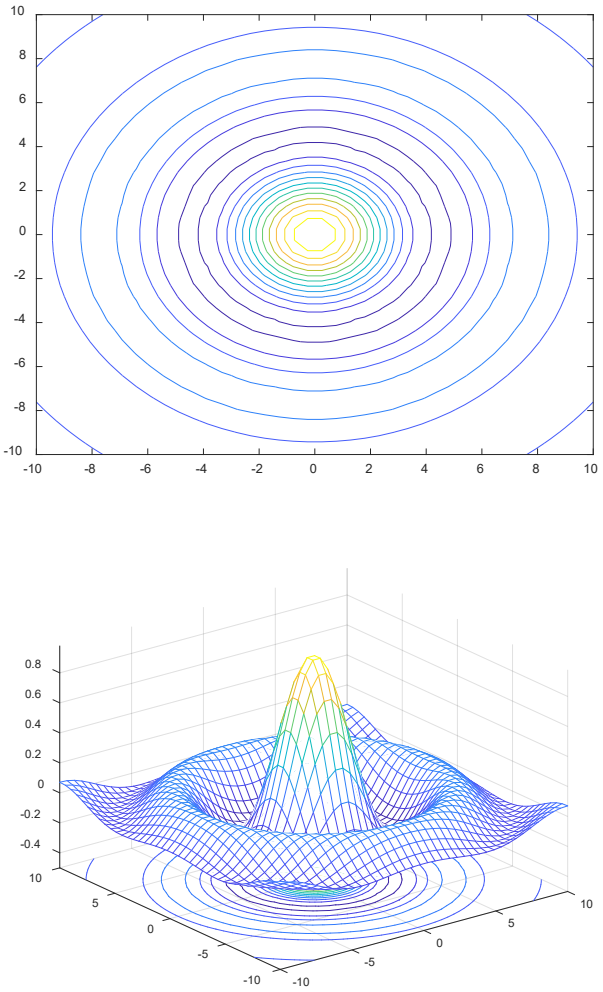
Contoh 2.8:

Akan digambarkan kontur dari fungsi $z = \frac{\sin r}{r}, r = \sqrt{x^2 + y^2}$, lalu bandingkan dengan plot permukaannya:

```
%simpan grafik8.m
x=linspace(-10,10,40);
y=x;
[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;
Z=sin(R)./R;
```

```
figure; contour(X,Y,Z);
figure; meshc(X,Y,Z);
```

di sini kita menggunakan variabel eps, untuk mencegah perhitungan $\frac{0}{0}$ ketika R = 0.



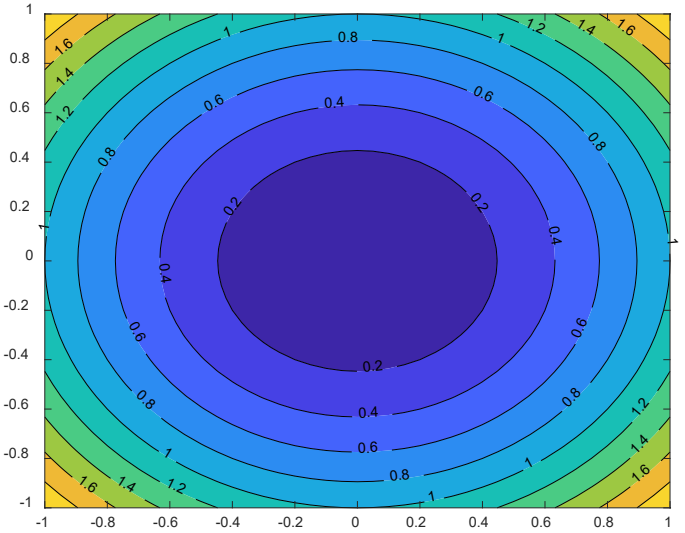
Grafik 2.10 Contour

Apa hasilnya ? Mengapa demikian ?

Sampai di sini, pahami hasil yang diberikan ! Apa yang anda temukan, catat dan analisalah!.

Dari contoh di atas, akan dibuat contourf dan clabel sebagai berikut:

```
x=-1:0.05:1;
y=x;
[X,Y]=meshgrid(x,y);
Z=Y.^2+X.^2;
mesh(X,Y,Z)
grid on
xlabel('sumbu X'); ylabel('sumbu Y');
zlabel('sumbu Z');
[c,h] =contourf(X,Y,Z,'k-');
clabel(c,h)
```

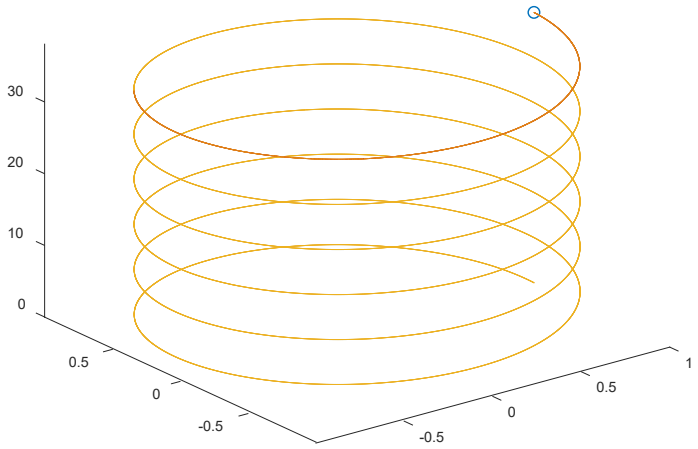


Grafik 2.11 Contouf dan Clabel

H. ANIMASI

Fungsi comet3 untuk membuat animasi dari plot.

```
t = 0:0.001:12*pi;
comet3(cos(t), sin(t), t)
```



Grafik 2.12 Plot Comet3

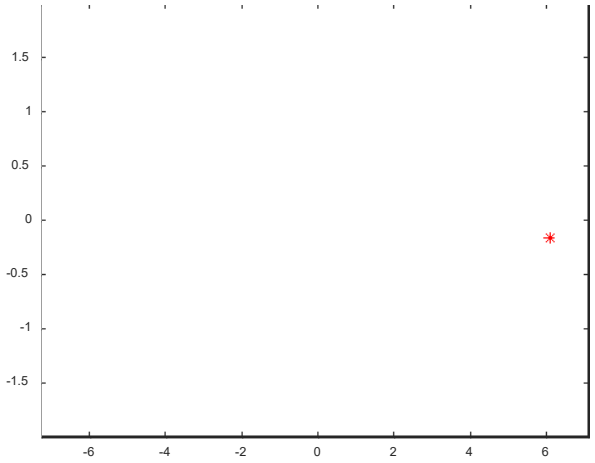
Perhatikan hasilnya, catat perbedaan-perbedaan yang anda temukan, dan analisalah!. perhatikan hasil yang diperoleh!. Lanjutkan perintah berikut:

Contoh 9.9:

Fungsi getframe dan movie untuk membuat animasi dari plot.

```
%simpan grafik10.m
x = -2*pi:1/5: 2*pi;
y = sin(x);
n =length(x);
for i = 1:n
plot(x(i),y(i),'r*')
```

```
axis([min(x)-1 max(x)+ 1 min(y)-1 max(y)+1])
M(i) = getframe;
end
movie(M)
```



Grafik 2.13 Fungsi getframe dan movie

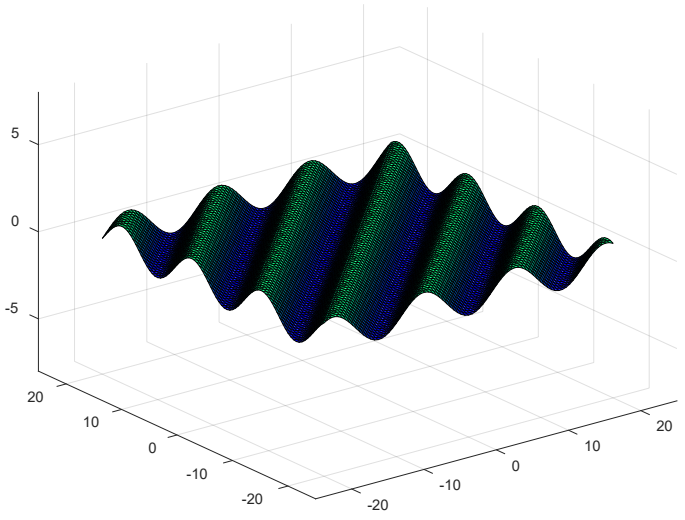
Sampai di sini, pahami hasil yang diberikan ! Apa yang anda temukan, catat dan analisalah!. Kemudian lanjutkan dengan perintah berikut:

Contoh 2.10:

Fungsi movingplot untuk membuat animasi dari plot.

```
%simpan grafik11.m
[X,Y] = meshgrid(-20:.5:20);
i=0;
while (i<100)
hold off;
R = ((X - Y) + eps +i)/2;
```

```
Z = sin(R); %figure(1)
surf(X,Y,Z)
colormap winter
axis([-25 25 -25 25 -8 8])
i=i+1;
hold on;
t = timer('StartDelay', 0.2,'TimerFcn', ...
@(x,y)disp(''));
start(t)
wait(t)
delete(t)
end
```



Grafik 9.13 Fungsi movingplot

LATIHAN 2 MATLAB

1. Gambarkan kurva berikut ini di dalam ruang 3 Dimensi

$$\left.\begin{aligned} x &= 1+\cos t \\ y &= 2+\sin t \\ z &= 1-\cos 2t \end{aligned}\right\} 0 \le t \le 2\pi$$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Misalkan akan digambar grafik dari Hiperbola $z = \frac{x^2}{64} - \frac{y^2}{36}$ pada kotak $\{(x,y) : -5 \le x \le 5, -5 \le y \le 5\}$, gunakan inkremen sebesar 0.5. Buatlah program untuk membuat grafik 3 Dimensi diatas!. Gunakan fungsi **surf, mesh,** dan **waterfall** untuk memplot fungsi diatas. Gunakan juga subplot(m,n,k) dengan, 1 baris dan 3 kolom grafik yang terpisah.

.....

.....

.....

.....

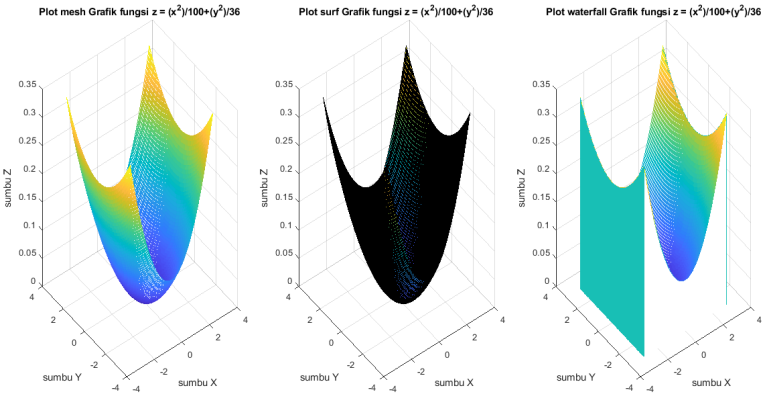
.....

.....

.....

.....

3. Perhatikan kurva berikut:

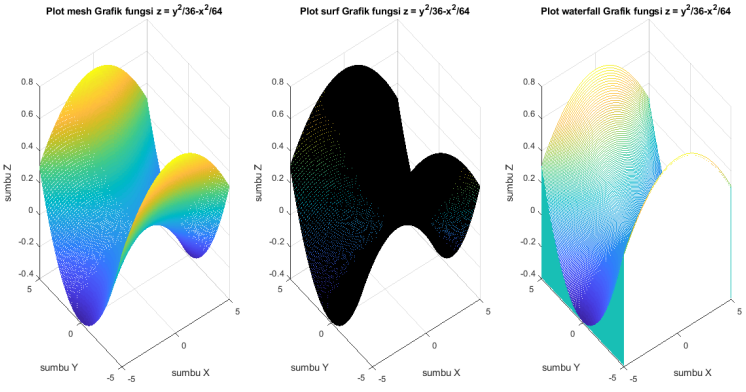


Kurva di atas merupakan skrip pemograman di bawah ini, cek pada baris ke berapa letak kesalahan skrip program?. Tuliskan skrip program yang benar (cukup tuliskan saja perbaikan skrip yang benar tanpa menulis seluruh skrip).

```
% Persamaan Ellips z=(x^2)/100+(y^2)/36
x=-3:0.05:3;%inkremen=0.05 agar kurva terlihat
mulus
y=x;
[X,Y]=meshgrid(x,y);
Z=((Y.^2)/100)+((X^2)/36);
subplot(1,3,1);
```

```
mesh(X,Y)
grid on
xlabel('sumbu X');
ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot mesh Grafik fungsi z =
(x^2)/100+(y^2)/36');
subplot(1,3,2);
surfgrid(X,Y,Z);
xlabel('sumbu X');
ylabel('sumbu Y');
zlabel('sumbu Z);
title('Plot surf Grafik fungsi z =
(x^2)/100+(y^2)/36');
subplot(1,3,3);
waterfall(X,Y,Z);
xlabel('sumbu X');
ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot waterfall Grafik fungsi z =
(x^2)/100+(y^2)/6');
```

4. Perhatikan kurva berikut:



Kurva di atas merupakan skrip pemograman di bawah ini, cek pada baris ke berapa letak kesalahan skrip program?. Tuliskan skrip program yang benar (cukup tuliskan saja perbaikan skrip yang benar tanpa menulis seluruh skrip).

```
% Persamaan Hyperbola z=y^2/36-x^2/64
x=-5:0.05:5;%inkremen=0.05 agar kurva terlihat
mulus
y=x;
[X,Y]=meshgrid(x,y);
Z=(Y^2/36)-(X^2/64);
subplot(1,3,1);
meshgrid(X,Y,Z)
grid on
xlabel('sumbu X');
ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot mesh Grafik fungsi z = y^2/36-
x^2/64);
subplot(1,3,2);
surf(X,Y,Z);
xlabel('sumbu X');
ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot surf Grafik fungsi z = y^2/36-
x^2/64');
subplot(1,3,5);
waterfall(X,Y,Z);
xlabel('sumbu X');
ylabel('sumbu Y');
zlabel('sumbu Z');
title('Plot waterfall Grafik fungsi z = y^2/36-
x^2/6');
```

5. Plot kontur dari fungsi dua variabel berikut ini:

$f(x, y) = \cos x \sin 2y, \text{untuk } 0 \le x \le 4\pi, 0 \le y \le 4\pi$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

BAB 3
FUNGSI DAN INTERPOLASI

A. Fungsi dan Interpolasi

Berbagai fungsi matematis bisa dievaluasi dan dianalisis dengan berbagai command yang ada di MATLAB. Salah satu fungsi matematis yang sering digunakan, yaitu polinomial, maksimum dan minimum dari fungsi, interpolasi, dan curve-fitting menggunakan MATLAB akan dibahas pula.

1. POLINOMIAL

Suatu polinomial, *p(x)*, berderajat *n* dinyatakan sebagai sebuah vektor baris **p** berukuran *n*+1. Elemen vektor menunjukkan koefisien dari polinomial yang diurutkan dari orde tertinggi ke terendah.

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_1 x^n + a_0 ;$$

dinyatakan sebagai:

$\mathrm{p} = \left(a_n a_{n-1} \ldots a_1 a_0 \right)$. *Command* berikut digunakan untuk menangani polinomial:

No.	Perintah Matlab	Keterangan
1.	`polyval(p,x)`	Mengevaluasi polinomial p pada nilai x. X bisa berupa skalar maupun vektor
2.	`poly(x)`	Menghitung vektor sepanjang n+1 yang mewakili suatu polinomial orde-n. Vektor x sepanjang n berisi akar-akar dari polinom tersebut
3.	`roots(p)`	Menghitung vektor berisi akar-akar dari polinomial p

4.	`conv(p,q)`	Menghitung produk (hasil perkalian) dari polinomial p dan q. Bisa juga dianggap sebagai konvolusi antara p dan q
5	`[k,r]=deconv(p,q)`	Membagi polinomial p dengan q. Hasil pembagian disimpan dalam polinom k dan sisa pembagian dalam polinom r. Bisa juga dianggap sebagai dekonvolusi antara p dan q
6.	`polyder(p)`	Menghitung vektor sepanjang n berisi turunan pertama dari polinom p

Tabel 3.1 Perintah Menghitung Polinomial

2. NOL DARI FUNGSI

Untuk mencari nol dari fungsi f(x), sama saja dengan mencari solusi dari f(x)=0. Nol dari suatu fungsi satu variabel bisa dicari dengan command fzero. Sementara untuk polinomial gunakanlah roots seperti pada subbab sebelumnya. Algoritma yang digunakan pada fzero bersifat iteratif, dan membutuhkan tebakan awal (initial guess) yang tidak terlalu jauh dari nol fungsi yang dicari.

No.	Perintah Matlab	Keterangan
1.	`fplot('fcn',lim ,'string')`	Memplot fungsi **fcn** pada interval lim dengan property yang didefinisikan oleh string, **fcn** berupa M-file yang berisi definisi fungsi. **lim** berupa vektor 2 elemen berisi batas interval **xmin** dan **xmax.**
2.	`fzero('fcn',x0)`	Menghitung nol dari fungsi **fcn** dengan nilai tebakan awal **x0.**

3.	`fzero('fcn',x0, tol)`	menghitung nol dari fungsi **fcn** dengan nilai tebakan awal **x0**. **tol** menentukan toleransi error dari perhitungan pendekatan yang diinginkan

Tabel 3.2 Perintah Menghitung Nol Fungsi

3. INTERPOLASI

Pada fungsi yang memiliki sejumlah titik terbatas, dimungkinkan untuk menentukan titik-titik perantaranya dengan interpolasi. Cara termudah untuk menghitungnya ialah dengan menggunakan interpolasi linier untuk menghubungkan dua titik yang berdekatan. Command **interp1** menggunakan algoritma khusus untuk interpolasi titik-titik data yang terpisah secara seragam. Untuk command ini, kita harus tambahkan **tanda asteris '*'** di depan nama metode yang diinginkan, misalkan **interp(x,y,xx,'*nearest').**

<table>
<tr><th>No.</th><th>Perintah Matlab</th><th>Keterangan</th></tr>
<tr><td>1.</td><td><code>yy = interp1(x,y,xx)</code></td><td>Menghitung minimum dari fungsi multi variabel fcn dengan tebakan awal berupa vektor x0.</td></tr>
<tr><td>2.</td><td><code>interp1(x,y,xx,'string')</code></td><td>Menghitung interpolasi 1-dimensi; string menunjukkan metode yang digunakan.
<table>
<tr><td>linear</td><td>interpolasi linier</td></tr>
<tr><td>nearest</td><td>interpolasi “nearest-neighbor”</td></tr>
</table>
</td></tr>
</table>

		spline	interpolasi "cubic-spline"
		cubic	interpolasi kubik, membutuhkan jarak pisah
		Apabila string tidak dituliskan, maka digunakan interpolasi linier. Untuk semua metode tersebut, x harus diurutkan ascending/descending.	
3.	`interp1q(x,y,xx)`	Bekerja seperti interp1 namun lebih cepat untuk titik-titik data yang terpisah tak seragam. x, y, dan xx harus berupa vektor kolom.	

Tabel 3.3 Perintah Menghitung Interpolasi

4. CURVE-FITTING

Pencocokkan kurva (*curve-fitting*) yang akan dibahas di sini ialah pencocokkan titik-titik data dengan suatu fungsi polinomial dengan metode pendekatan kuadrat terkecil (*least squares approximation*). Tujuannya adalah menemukan suatu kurva halus yang paling mendekati data tetapi tidak harus melewati setiap point data.

No.	Perintah Matlab	Keterangan
1.	`polyfit(x,y,n)`	Untuk menghitung vektor berisi koefisien polinomial orde-*n* yang mendekati titik-titik data di (*xi*,*yi*)

2.	`[p,E]=polyfit(x,y,n)`	Menghitung vektor polinomial **p** dan matriks **E** yang bisa digunakan oleh *command* polyval untuk mengestimasi error.

Tabel 10.4 Perintah Menghitung Curve-Fitting

5. POLINOMIAL MATLAB

Contoh 3.1:

Misalkan kita memiliki dua polinomial sebagai berikut:

$f(x) = 4x^3 - 3x + 3$ dan $g(x) = 7x^2 + 5$

Dalam MATLAB kedua polinomial ini dinyatakan dengan:

```
>> f = [4 0 -3 3];
>> g = [7 0 5];
```

Untuk mengevaluasi polinomial pada x = 10 kita tuliskan:

```
>> nilai1 = polyval(f,10), nilai2 =
polyval(g,10)
nilai1 =
        3973
nilai2 =
   705
```

Namun bisa pula x berbentuk vektor:

```
>> x = -3:3
>> nilai1 = polyval(f,x), nilai2 = polyval(g,x)
nilai1 =
   -96   -23     2     3     4    29   102
nilai2 =
```

```
    68     33     12      5     12     33     68
```

Jika kita kalikan kedua polinomial tersebut, akan diperoleh sebuah polinomial baru:

```
>> p = conv(f,g)
p =
    28      0     -1     21    -15     15
```

yang mewakili:

$$p(x) = 28x^5 - x^3 + 21x^2 - 15x + 15$$

Akar-akar dari polinomial f(x) dan g(x) bisa kita hitung:

```
>> akar_f = roots(f), akar_g = roots(g)
akar_f =
  -1.1777 + 0.0000i
   0.5888 + 0.5387i
   0.5888 - 0.5387i
akar_g =
   0.0000 + 0.8452i
   0.0000 - 0.8452i
```

Turunan pertama dan kedua dari g(x) bisa kita hitung pula:

```
>> g1=polyder(g), g2=polyder(g1)
g1 =
    14     0
g2 =
    14
```

yang masing-masing mewakili

$$g'(x) = 14x \text{ dan } g''(x) = 14$$

6. NOL DARI FUNGSI MATLAB

Fungsi matematis bisa dinyatakan dalam bentuk M-file di MATLAB. Misalkan fungsi

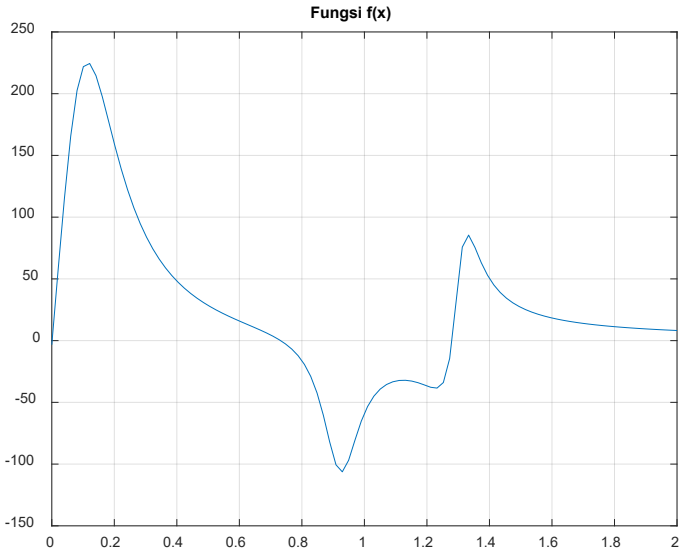
$$f(x)=\frac{5x-6,4}{(x-13)^2+0,002}+\frac{9x}{x^3+0,03}-\frac{x-0,4}{(x-0,92)^2+0,005}$$

bisa kita tuliskan pada editor M-file.

```
function y = f(x)
y = (5.*x - 6.4)./((x-1.3).^2 + 0.002) + ...
(9.*x)./(x.^3 + 0.003) - ...
(x - 0.4)./((x-0.92).^2 + 0.005);
```

Fungsi f didefinisikan menggunakan operator elemen-per-elemen .* ./ .^ + -, sehingga apabila fungsi dipanggil dengan argumen vektor maka hasilnya juga berupa vektor. Semua fungsi MATLAB pada bab ini harus didefinisikan seperti contoh tersebut. Fungsi tersebut bisa diplot dengan *command* **plot**:

```
>> x = linspace(0,2); % membuat vektor x
>> plot(x,f(x)); % memplot grafik f(x)
>> grid on;
>> title('Fungsi f(x)');
```



Grafik 3.1 Fungsi rasional f(x)

Atau menggunakan *command* **fplot**:

```
>> fplot('f',[0 2]); % memplot grafik f(x)
>> grid on;
>> title('Fungsi f(x)');
```

Untuk mencari nol dari fungsi *f(x)*, sama saja dengan mencari solusi dari *f(x)* = 0. Nol dari suatu fungsi satu variabel bisa dicari dengan *command* **fzero**. Sementara untuk polinomial gunakanlah **roots** seperti pada subbab 8.1. Algoritma yang digunakan pada **fzero** bersifat iteratif, dan membutuhkan tebakan awal (*initial guess*) yang tidak terlalu jauh dari nol fungsi yang dicari. *Command* **zerodemo** akan memberikan demonstrasi dari topik ini. Kita akan menghitung nol dari fungsi *f(x)* sebagai berikut:

```
>> x1=fzero('f',0), x2=fzero('f',0.5),
x3=fzero('f',2)
x1 =
    0.0011
x2 =
    0.7320
x3 =
    1.2805
```

Misalkan kita ingin menghitung titik potong dari dua fungsi: $\cos 2x$ dan $5x-2$; atau dengan kata lain mencari solusi dari persamaan:

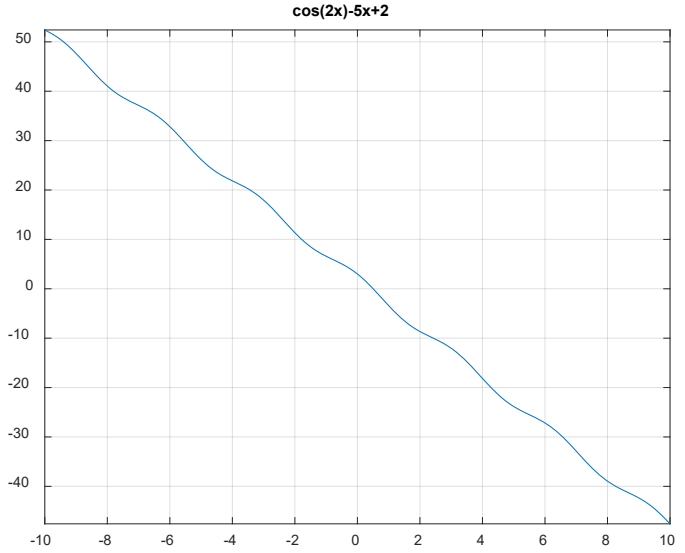
$s(x) = \cos 2x - 5x + 2 = 0$

Maka, pertama kita definisikan fungsi **cosm.m** dalam M-file.

```
function s = cosm(x)
s = cos(2*x)-5*x + 2;
```

Kemudian kita plot untuk memudahkan mendapatkan tebakan awal:

```
>> fplot('cosm',[-10 10]);
>> grid on;
>> title('cos(2x)-5x+2');
```



Grafik 3.2 Fungsi cos(2x)-5x+2

Kita lihat bahwa $x = 2$ merupakan tebakan awal yang bagus.

```
>> nol = fzero('cosm',2)

nol =

    0.5060
```

7. INTERPOLASI MATLAB

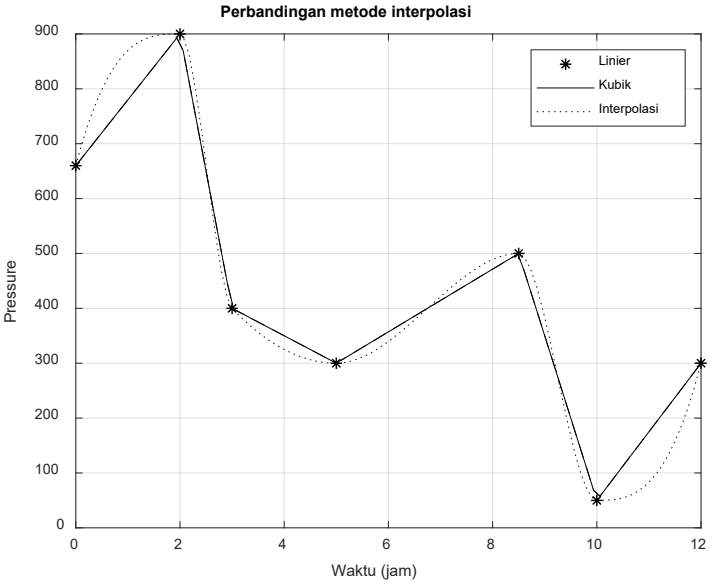
Pada fungsi yang memiliki sejumlah titik terbatas, dimungkinkan untuk menentukan titik-titik perantaranya dengan interpolasi. Cara termudah untuk menghitungnya ialah dengan menggunakan interpolasi linier untuk menghubungkan dua titik yang berdekatan. *Command* **interp1** menggunakan algoritma khusus untuk interpolasi titik-titik data yang terpisah secara seragam. Untuk *command* ini, kita harus tambahkan tanda asteris '*' di depan nama metoda yang diinginkan, misalkan

interp(x,y,xx,'*nearest'). Misalkan kita memiliki data tekanan udara dalam suatu ruang tertutup yang diukur pada jam-jam tertentu sebagai berikut:

```
>> t = [0 2 3 5 8.5 10 12];
>> pres = [660 900 400 300 500 50 300];
```

Sekarang kita interpolasi dengan beberapa metode dan kita plot pada satu gambar sekaligus:

```
>> tt = linspace(0,12,100);
>> PP1 = interp1(t,pres,tt,'*Linear');
>> PP2 = interp1(t,pres,tt,'*Cubic');
>> PP3 = interp1(t',pres,tt');
>> figure;
>> plot(t,pres,'k*',tt,PP1,'k-
',tt,PP2,'k:',tt,PP3,'k--')
>> grid on;
>> xlabel('Waktu (jam)'), ylabel('Pressure')
>> legend('Linier','Kubik','Interpolasi')
>> title('Perbandingan metode interpolasi')
```



Grafik 3.3 Perbandingan metode interpolasi

8. CURVE-FITTING

Pencocokkan kurva (*curve-fitting*) yang akan dibahas di sini ialah pencocokkan titik-titik data dengan suatu fungsi polinomial dengan metode pendekatan kuadrat terkecil (*least squares approximation*). Tujuannya adalah menemukan suatu kurva halus yang paling mendekati data tetapi tidak harus melewati setiap point data.

```
>> t = [0 2 3 5 8.5 10 12];
>> pres = [660 900 400 300 500 50 300];
>> p3 = polyfit(t,pres,3)
p3 =
    0.5857   -6.9967  -38.3200  727.0393
>> p4 = polyfit(t,pres,4)
p4 =
```

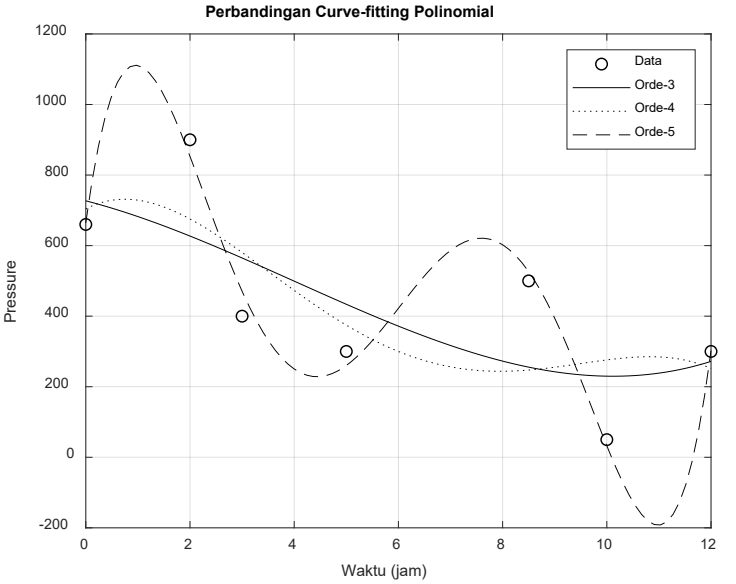
```
   -0.3022    7.8645  -60.4717   77.6181
704.1170
>> p5 = polyfit(t,pres,5)
p5 =
1.0e+003 *
0.0006 -0.0183 0.1908 -0.8055 1.0783 0.6648
```

Polinomial yang diwakili oleh **p3**, **p4**, dan **p5** ialah:

$$p(x) = 0,6x^3 - 7x^2 - 38,3x + 727$$
$$p(x) = -0,3x^4 + 7,9x^3 - 60,5x^2 + 77,6x + 704,1$$
$$p(x) = 0,6x^5 - 18,3x^4 + 190,8x^3 - 805,5x^2 + 1078,3x + 664,8$$

Berikutnya kita plot data dan ketiga kurva polinomial tersebut untuk dibandingkan.

```
>> tt = linspace(0,12,100);
>> kurva_p3 = polyval(p3,tt);
>> kurva_p4 = polyval(p4,tt);
>> kurva_p5 = polyval(p5,tt);
>> figure
>>
plot(t,pres,'ko',tt,kurva_p3,'k',tt,kurva_p4,'k
:',tt,
 kurva_p5,'k--')
>> grid on;
>> xlabel('Waktu (jam)'), ylabel('Pressure')
>> legend('Data', 'Orde-3','Orde-4','Orde-5')
>> title('Perbandingan Curve-fitting
Polinomial')
```



Grafik 3.4 Perbandingan Curve-fitting Polinomial orde 3, 4, dan 5

LATIHAN 3 MATLAB

1. Nyatakanlah polinomial berikut dalam bentuk vektor baris:

$$p(x) = x^2 - 1 \qquad q(x) = x^4 - \frac{10}{9}x^2 + \frac{1}{9}$$

$$r(x) = \left(x^2 + \frac{3}{2}x + \frac{1}{2}\right)\left(x^3 - \frac{3}{2}x^2 + \frac{1}{2}x\right)$$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Evaluasilah ketiga polinomial berikut pada nilai-nilai: $x = -1.5, -1.2, -0.9, ..., 1.2, 1.5$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Buatlah plot dari ketiga polinomial pada no.1 tersebut pada rentang: $-1,5 \le x \le 1,5$. Buatlah inkremen x cukup kecil agar kurva terlihat mulus.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

4. Diketahui tiga buah polinomial yaitu:

$p_1(x) = 2x^3 + 5x^2 + 3x + 6,\ p_2(x) = x^3 - 3x - 1,$ dan

$p_3(x) = x^2 + 3x + 2$

Dengan menggunakan perintah MATLAB, tentukan:

a) Mencari akar-akar $p_1(x)$

.....

.....

.....

.....

.....

.....

.....

b) Mengalikan $p_1(x)$ dengan $p_3(x)$

.....

.....

.....

.....

.....

.....

.....

c) Menjumlahkan $p_1(x)$ dengan $p_2(x)$.

.....

.....

.....

.....

.....

.....

.....

5. Hitunglah nol fungsi rasional berikut ini pada rentang $-10 \le x \le 10$

$$F(x) = \frac{x-1}{x^2+1}$$

.....

.....

.....

.....

.....

.....

.....

.....

6. Berikut ini data distribusi Peminjaman Buku selama 20 hari terakhir.

Distribusi Peminjaman Buku	
Peminjaman Buku hari Ke	**Frekuensi**
1	185
2	130
2	101
3	72
4	54
5	40
6	29
7	22
8	17
9	11

10	10
11	8
12	6
13	4
14	3
15	2
16	2
17	1
18	1
19	1
20	1

Plot data distribusi ini dan dekatilah dengan dua metode interpolasi!

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

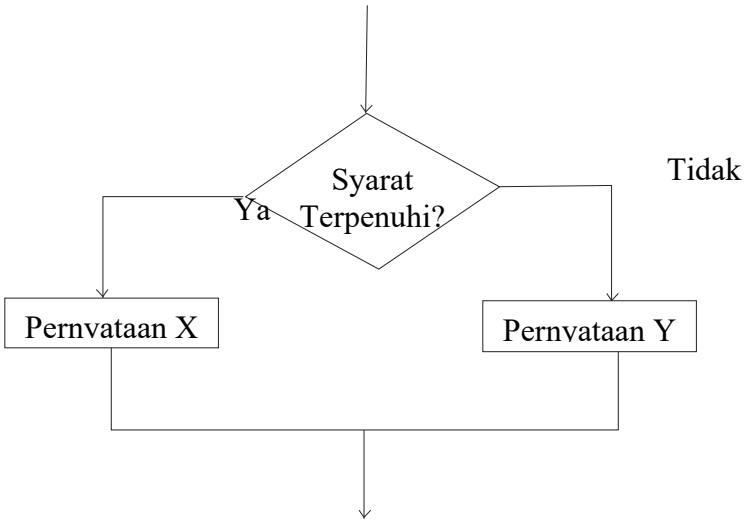
.....

BAB 4
PENCABANGAN BERSYARAT

Program-program yang dihasilkan pada **Bab-bab sebelumnya** merupakan program yang tidak mempunyai kendali logika. Pada program tersebut, pernyataan demi pernyataan dieksekusi dari baris pertama hingga baris terakhir. Pada program yang lebih kompleks, program logika tidak mengalir sesederhana itu. Terdapat kasus – kasus yang mengharuskan program logika mengalir dengan syarat tertentu. Secara umum, di semua bahasa pemrograman termasuk MATLAB, terdapat dua macam kendali aliran, yaitu percabangan bersyarat dan percabangan perulangan.

A. ALIRAN LOGIKA

Aliran logika pada kendali pencabangan bersyarat digambarkan dengan *flowchart* berikut:



Grafik 4.1 Aliran Logika Pada Kendali Pencabangan Bersyarat

Pelaksanaan eksekusi pada pernyataan X atau Y tergantung pada hasil pengujian syarat. Jika syarat terpenuhi, maka eksekusi berikutnya adalah pernyataan X, tetapi jika syarat tidak terpenuhi, maka eksekusi selanjutnya adalah pernyataan Y. Kadang – kadang pernyataan Y tidak diperlukan, yang berarti pengujian syarat dilakukan untuk menentukan apakah pernyataan X perlu dieksekusi atau tidak.

B. OPERATOR RELASI DAN LOGIKA

Untuk menyatakan syarat pemilihan biasanya digunakan perbandingan antara dua buah nilai. Perbandingan dilakukan dengan menggunakan operator relasi. Berikut operator relasi:

No.	Operator Relasi	Keterangan
1.	<	Kurang dari
2.	<=	Kurang dari atau sama dengan
3.	>	Lebih dari
4.	>=	Lebih dari atau sama dengan
5.	==	Sama dengan
6.	~=	Tidak sama dengan

Tabel 4.1 Operator Relasi

Operator logika digunakan untuk mendapatkan nilai logik dari hubungan dua buah variabel atau lebih berdasarkan hukum-hukum logika. A dan B di bawah ini dapat berupa skalar, vektor, maupun matriks, asalkan ukuran A dan B sama. Operator tersebut adalah:

No.	Operator Logika	Keterangan
1.	`and(A,B)atau A & B`	Operator logika AND antara A dan B
2.	`or(A,B) atau A|B`	Operator logika OR
3.	`xor(A,B)`	Operator logika XOR
4.	`not(A) atau ~A`	Operator logika NOT pada A

Tabel 4.2 Operator Logika

Operator logika NOT akan menghasilkan nilai logika yang berlawanan dari nilai logika sebelumnya. Jika sebelumnya bernilai 1, maka operator NOT menyebabkan bernilai 0; begitu sebaliknya jika sebelumnya bernilai 0, dengan operator NOT menjadikan bernilai 1. Sedangkan untuk mendapatkan nilai logika dari pembandingan dua buah variabel atau lebih berdasarkan hukum-hukum logika dari operator AND, OR, XOR dan NOT. Adapun

tabel kebenaran yang digunakan pada setiap operasi logika sebagai berikut:

No	A	B	A & B	A | B	xor(A,B)	~A
1.	nol	nol	0	0	0	1
2.	nol	bukan nol	0	1	1	1
3.	bukan nol	nol	0	1	1	0
4.	bukan nol	bukan nol	1	1	0	0

Tabel 4.3 Nilai Logika

Nilai benar pada MATLAB dinyatakan dengan nilai 1 dan sebaliknya nilai salah dinyatakan dengan nilai 0. Hasil operasi 0 atau 1 dapat digunakan sebagai syarat pemilihan. Sebaliknya, hasil perbandingan dapat pula digunakan dalam operasi matematis. Operator logika menyediakan cara untuk mengevaluasi ekspresi logika. Operator tersebut adalah:

No	Fungsi	Keterangan
1.	`xor(x,y)`	Operasi exclusive OR. Bernilai benar (1) jika x dan y mempunyai nilai yang berbeda; bernilai salah (0) jika x dan y mempunyai nilai yang sama.
2.	`any(x)`	Bernilai benar (1) jika terdapat elemen dalam vektor x yang tidak nol; bernilai benar (1) jika tiap-tiap kolom dalam matriks x mempunyai elemen tidak nol.
3.	`all(x)`	Bernilai benar (1) jika semua elemen dalam vektor x adalah tidak nol; bernilai benar (1) jika tiap-tiap kolom dalam matriks x semuanya mempunyai elemen tidak nol.

Tabel 4.4 Fungsi-Fungsi Logika Dan Relasional

Untuk memastikan urutan operasi, maka tiap syarat ditulis dengan menggunakan tanda kurung "( )". Hal ini terutama apabila syarat tebentuk dari beberapa syarat yang dievaluasi

menggunakan ***operator logika***. Contoh syarat yang tersusun dari beberapa syarat.

```
Syarat = syarat1 & ( syarat2 | syarat3 )
  = (a == b) & ((b ~=c) | (b > d ))
```

C. PERNYATAAN if...elseif...else...end

Ini merupakan pernyataan untuk percabangan program berdasarkan satu/beberapa kondisi tertentu. Sintaks yang digunakan dalam MATLAB dinyatakan dengan pernyataan sebagai berikut :

Satu syarat if...end	Dua syarat if...else...end	Tiga syarat atau lebih if...elseif...else...end

`if syarat`
`Pernyataan X;`
`end;`
	`if syarat`
	`pernyataan X;`
	`else`
	`pernyataan Y;`
	`end;`
		`if syarat1`
		`pernyataan X;`
		`elseif syarat 2`
		`pernyataan Y;`
		`elseif syarat 3`
		`pernyataan R;`
		`elseif...`
		`...`
		`else...`
		`pernyataan Z;`
		`end;`

Tabel 4.5 Perintah Pernyataan if...elseif...else...end

Pernyataan Z pada kasus ketiga berguna apabila tidak ada syarat pada pernyataan lain.

D. PERNYATAAN switch…case...otherwise...end

Sering kali pengujian syarat bukan nilai benar (yang berarti syarat terpenuhi) atau salah (yang berarti syarat tidak terpenuhi). Jika syarat berupa operasi matematis, maka yang dievaluasi

sebagai syarat adalah kesamaan hasil dengan konstanta yang telah didefinisikan sebelumnya. Pada kasus ini, penggunaan pernyataan *if...elseif...else...end* kurang efektif. Sebagai alternatif dari statement *if... elseif... else... end*, kita bisa menggunakan statement *switch*. Sintaksnya ialah:

Pernyataan switch…case...otherwise...end

```
switch nama_variabel
case{kondisi1,kondisi2,...}
      Dijalankan jika kondisi1 atau kondisi2
dst...
      dipenuhi
case{kondisiA,kondisiB,...}
      Dijalankan jika kondisiA atau kondisiB
dst...
      dipenuhi
case{kondisiX,kondisiY,...}
      Dijalankan jika kondisiX atau kondisiY
dst...
      dipenuhi
case{...}
      ...dst...
```

Buat program-program berikut, simpan sebagai m-file, kemudian jalankan. Catat dan amati hasilnya.

E. OPERATOR RELASI DAN LOGIKA MATLAB

Untuk menambah pemahaman, mari kita praktekkan contoh di bawah ini di *command window*:

Contoh 4.1:

```
>> A = [1 2 0 -1 -2]; B = [1 0 0 0 5];
>> C = and(A,B)
```

```
C =      1     0     0     0     1
>> D=A|B|C
D =      1     1     0     1     1
>> E = xor(~A,B)
E =      1     0     1     0     1
```

Sekarang, mari kita mencoba membuat fungsi untuk menentukan suatu tahun termasuk kabisat atau tidak. Jangkauan tahun yang bisa dihitung ialah 1900 hingga 2500. Kita ketahui bahwa tahun kabisat terjadi pada tahun-tahun berkelipatan 4, kecuali tahun akhir abad; namun untuk tahun akhir abad berkelipatan 400 termasuk kabisat pula. Ketik program berikut, simpan **iskabisat.m**

Contoh 4.2:

```
% Fungsi untuk mengetahui tahun kabisat atau
tidak
% iskabisat.m
function hasil = iskabisat(thn)
% thn: merupakan masukan bilangan bulat positif
% hasil = 1 jika kabisat, 0 jika tidak
if thn<1900|thn>2500
disp('Tahun yang valid: 1900-2500');
hasil=[];
return
end
if
rem(thn,4)==0&(rem(thn,100)~=0|rem(thn,400)==0)
hasil=1;
else
hasil=0;
```

```
end
```

Pada fungsi tersebut, terdapat dua control statement " if ":

- if thn<1900 | thn>2500

  Berarti jika variabel tahun kurang dari 1900 atau lebih dari 2500, *command* di dalam "if" tersebut akan dijalankan.

- if rem(thn,4)==0 & (rem(thn,100)~=0|rem(thn,400)==0)

  Berarti jika variabel thn habis dibagi 4, logika DAN (rem(thn, 100)~=0|rem(thn,400)==0)bernilai 1 (true), maka *command* setelah " if " akan dijalankan. Perlu diperhatikan bahwa logika (rem(thn,100)~=0|rem(thn,400)==0)akan bernilai 1 bila tahun bukan tahun abad (kelipatan 100); ataupun kalau tahun abad haruslah kelipatan 400. Sekarang kita bisa coba:

```
>> iskabisat(2020), iskabisat(1972)
ans =
1
ans =
1
```

Fungsi ini hanya bisa mengolah masukan skalar. Lalu bagaimana kalau diinginkan masukan berupa vektor atau matriks? Kita bisa ubah fungsinya menjadi berikut ini:

Contoh 4.3:

```
% Fungsi untuk mengetahui tahun kabisat atau
tidak
% iskabisat2.m
function hasil = iskabisat2(thn)
```

```
% thn : merupakan masukan bilangan bulat
positif
% hasil = 1 jika kabisat, 0 jika tidak
if sum(sum(thn<1900 | thn>2500))~= 0
disp('Tahun yang valid: 1900 - 2500');
hasil=[];
return
end
hasil=rem(thn,4)==0
&(rem(thn,100)~=0|rem(thn,400)==0);
```

Sekarang kita bisa coba untuk menentukan tahun kabisat antara 1998 hingga 2010.

```
>> iskabisat2(1998:2020)
  Columns 1 through 13
   0   0   1   0   0   0   1   0   0   0   1
0   0
  Columns 14 through 23
   0   1   0   0   0   1   0   0   0   1
```

F. PERNYATAAN If...elseif...else...end MATLAB

Ketik program berikut dan simpan dengan nama **ifend1.m** Program tersebut hanya mempunyai satu pilihan, vektor x akan ditampilkan bila masukan, yaitu variabel

Contoh 4.4:

```
x = round(rand(1,10)*20)-10;
```

```
jawab = input('apakah x akan ditampilkan
[0=tidak,1=ya]?');
if(jawab==1)
disp(x);
end;
```

Pada beberapa kasus diperlukan masukan berupa jawaban ya atau tidak. MATLAB mampu menerima karakter y atau t, tetapi harus dimasukkan sebagai 'y' atau 't' (dalam tanda petik tunggal). Hal ini kurang efektif, sehingga lebih mudah jika dipakai 0 untuk *tidak* dan 1 untuk *ya*. Program berikut simpan dengan nama **ifend2.m**, untuk menguji nilai positif atau negatif dari masukan. Program mempunyai dua pilihan.

Contoh 4.5:

```
a=input('masukan=');
if(abs(a)==a)
disp('bilangan positif');
else
disp('bilangan negatif');
end;
```

Uji untuk bilangan positif dan negatif. Bagaimana dengan bilangan 0, positif atau negatif?.

Ketik program berikut dan simpan dengan nama **abc1.m**!

Contoh 4.6:

```
%Akar persamaan kuadrat dengan
%rumus abc
clear;
a=input('Masukkan Nilai a = ');
```

```
b=input('Masukkan Nilai b = ');
c=input('Masukkan Nilai c = ');
d=b^2-(4*a*c);
if d>0
   x1=(-b-d^0.5)/(2*a);
   x2=(-b+d^0.5)/(2*a);
   fprintf('X1 = %.2f\n',x1);
    fprintf('X2 = %.2f\n',x2);
elseif d==0
      x=-b/(2*a);
      fprintf('X = %.2f\n',x);
   else
      disp('Akar Kompleks');
   end
```

Contoh 4.7:

Buatlah program berikut dan simpan dengan nama **toko.m**. Perhatikan soal berikut: Sebuah toko yang menjual buah-buahan dengan harga per kg Rp. 15.000,- sedangkan pembelian dibatasi maksimal 100 kg dan menetapkan akan memberikan potongan harga menurut tabel di bawah ini:

Daftar Potongan Harga	
10%	Pembelian lebih dari 10 kg
15%	Pembelian lebih dari 20 kg
20%	Pembelian lebih dari 50 kg

Jalankan program berikut dan amati hasilnya.

```
x=input('Masukkan jumlah pembelian (kg):');
```

```
bayar=x*15000;
if x<0
    disp('Nilai harus positif')
elseif x>=0 & x<=10
    bayar=bayar;
elseif x>10 & x<=20
    bayar=(1-10/100)*bayar;
elseif x>20 & x<=50
    bayar=(1-15/100)*bayar;
elseif x>50 & x<=100
    bayar=(1-20/100)*bayar;
else
    bayar=[];
    disp('Pembelian di batasi 100 kg')
end
disp(['Jumlah yang harus dibayar =
Rp',num2str(bayar)]);
```

G. PERNYATAAN switch…case…otherwise...end MATLAB

Pernyataan switch mengeksekusi serangkaian pernyataan yang kriterianya didasarkan pada nilai variabel atau ekspresi. Hanya kasus yang cocok pertama saja yang dieksekusi, sedangkan kasus lainnya tidak dieksekusi. Buatlah program berikut dan simpan dengan nama **tiket.m**

Contoh 4.8:

```
disp('Arah pintu masuk:');
disp('1. Tiket VIP');
disp('2. Tiket Kelas 1');
```

```
disp('3. Tiket Kelas 2');
disp('4. Tiket Tribun Atas');
disp('5. Loket Tiket');
x=input('Masukkan kelas tiket yang diambil:');
switch x
    case 1
        disp('Pintu depan 101 dan 102')
    case 2
        disp('Pintu samping kanan 103 dan 104')
    case 3
        disp('Pintu samping kiri 105 dan 106')
    case 4
        disp('Pintu belakang 107 dan 108')
    case 5
        disp('Pintu utama gedung')
    otherwise
        disp('Kelas yang dimasukkan salah')
end
```

pada **m-file** ini kita menginginkan salah satu keluaran dari lima pilihan yang ada, yaitu menampilkan dimana letak pintu masuk berdasarkan tiket kelas yang dimiliki, untuk mengatasi kemungkinan lain digunakan perintah **otherwise**.

```
>> tiket
Arah pintu masuk:
1. Tiket VIP
2. Tiket Kelas 1
3. Tiket Kelas 2
```

```
4. Tiket Tribun Atas
5. Loket Tiket
Masukkan kelas tiket yang diambil:2
Pintu samping kanan 103 dan 104
>> tiket
Arah pintu masuk:
1. Tiket VIP
2. Tiket Kelas 1
3. Tiket Kelas 2
4. Tiket Tribun Atas
5. Loket Tiket
Masukkan kelas tiket yang diambil:6
Kelas yang dimasukkan salah
```

Sesungguhnya, kendali alur **switch - case - otherwise - end** mempunyai kemiripan dengan **if-elseif-else-end**. Tetapi alur **switch** hanya bisa mengevaluasi kesamaan, sedangkan **if** bisa untuk kesamaan maupun ketidaksamaan. Di lain pihak, **switch** dapat membandingkan string dengan panjang berbeda, sedangkan **if** masih membutuhkan perintah **strcmp** untuk membandingkan string dengan panjang berbeda. File **tiket.m** di atas dapat pula ditulis dengan menggunakan if sebagai berikut. Untuk membedakan file simpan dengan nama **tiket2.m**

Contoh 4.9:

```
disp('Arah pintu masuk:');
disp('1. Tiket VIP');
disp('2. Tiket Kelas 1');
disp('3. Tiket Kelas 2');
```

```
disp('4. Tiket Tribun Atas');
disp('5. Loket Tiket');
x=input('Masukkan kelas tiket yang diambil:');
if x==1
    disp('Pintu depan 101 dan 102')
elseif x==2
    disp('Pintu samping kanan 103 dan 104')
elseif x==3
    disp('Pintu samping kiri 105 dan 106')
elseif x==4
    disp('Pintu belakang 107 dan 108')
elseif x==5
    disp('Pintu utama gedung')
else
    disp('Kelas yang dimasukkan salah')
end
```

Tetapi alur **switch** tidak bisa digunakan untuk kasus dalam contoh pemakaian if sebelumnya.

Program berikut (**sisabagi.m**) menampilkan sisa bagi.

Contoh 4.10:

```
m = input('m=');
n = mod(m,5);
if(n==0)%konstanta 0
    disp('tak bersisa');
elseif (n==1)%konstanta 1
    disp('sisa satu');
elseif (n==2)%konstanta 2
```

```
    disp('sisa dua');
elseif (n==3)%konstanta 3
    disp('sisa tiga');
elseif (n==4)%konstanta 4
    disp('sisa empat');
else
    fprintf('m bilangan pecahan,sisa %f\n',n);
end
```

Bandingkan program diatas dengan program berikut (**modulo.m**). Manakah yang lebih efektif, dari sisi pembacaan dan penulisan program.

Contoh 4.11:

```
m=input('m='); n=mod(m,5);
switch n
case 0
disp('tak bersisa');
case 1
    disp('sisa satu');
case 2
disp('sisa dua');
case 3
    disp('sisa tiga');
case 4
    disp('sisa empat');
otherwise
fprintf('m bilangan pecahan,sisa %f\n',n);
end
```

Apakah konstanta case dapat berupa bilangan pecahan ? Buat program untuk mengujinya.

LATIHAN 11 MATLAB

1. Andaikan X=5, Y=10, dan Z=15; selidiki nilai logika dari pernyataan berikut ini :

a) Y=Z-X

.....

.....

.....

.....

.....

.....

.....

b) (X<5)OR(Z>(X+Y))

.....

.....

.....

.....

.....

.....

.....

c) NOT(Y<=12)AND(X MOD 2 =0)

.....

.....

.....

.....

.....

.....

.....

2. Buatlah program MATLAB menggunakan fungsi ***If...elseif...else...end*** untuk "Program Konversi Suhu" untuk suhu Celcius (C), Fahrenheit (F), Reamur (R) dan Kelvin (K) dengan rumusan sebagai berikut:

a) Untuk konversi suhu Celcius (C)

$$R = \frac{4}{5}C$$

$$F = \frac{9}{5}C + 32$$

$$K = C + 273$$

.....

.....

.....

.....

.....

.....

.....

b) Untuk konversi suhu Fahrenheit (F)

$$R = \frac{4}{9}F - 32$$

$$C = \frac{5}{9}F - 32$$

$$K = C + 273$$

.....

.....

.....

.....

.....

.....

.....

c) Untuk konversi suhu Reamur (R)

$$C = \frac{5}{4}R$$

$$F = \frac{9}{5}R - 32$$

$$K = C + 273$$

.....

.....

.....

.....

.....

.....

.....

d) Untuk konversi suhu Kelvin (K)

$$C = K - 273$$

$$R = \frac{4}{5}C$$

$$K = \frac{9}{5}C + 32$$

.....

.....

.....

.....

.....

.....

.....

3. Buatlah program MATLAB menggunakan fungsi ***If...elseif...else...end*** untuk menghitung nilai akhir mahasiswa dari nilai tugas1, tugas2 dan tugas3, UTS serta UAS dengan, Rumus: Rata-rata tugas (tugas1,tugas2, dan tugas3) bobot 30%, UTS bobot 30% dan UAS bobot 40%. Kriteria penilaian:

Standar Skor	**Nilai**	**Keterangan**
80 ≤ Nilai Total ≤ 100	A	Istimewa
75 ≤ Nilai Total < 80	B+	Baik Sekali
70 ≤ Nilai Total < 75	B	Baik
60 ≤ Nilai Total < 70	C+	Cukup Baik
50 ≤ Nilai Total < 60	C	Cukup
25 ≤ Nilai Total < 50	D	Kurang
Nilai Total < 25	E	Kurang Sekali

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

4. Buatlah program MATLAB menggunakan fungsi ***switch…case…otherwise...end*** untuk menghitung volume dan luas permukaan bangun ruang: kubus, balok, prisma, limas, kerucut, tabung dan bola.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

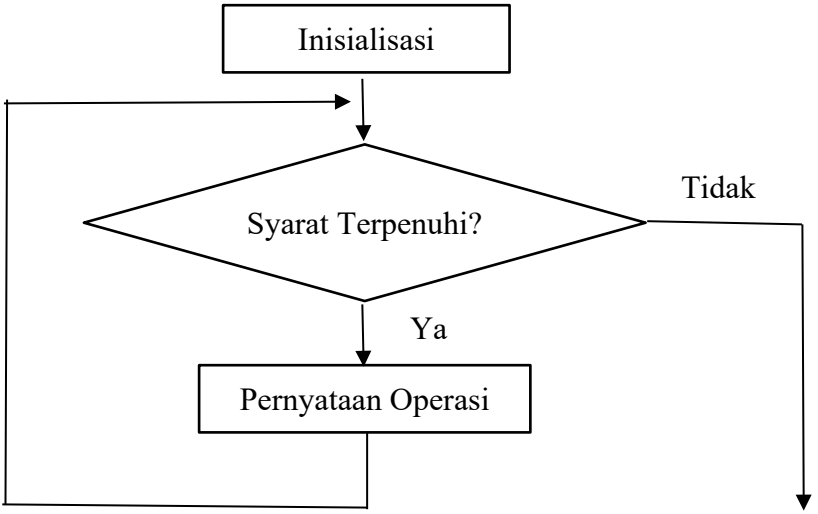
.....

BAB 5
KENDALI PERULANGAN

Inisialisasi biasanya merupakan penetapan nilai awal dari pencacah. Nilai tersebut akan diuji pada syarat perulangan. Syarat perulangan sama sebagaimana pada pencabangan bersyarat. Padanya berlaku pula operator relasi dan logika. Aliran logika perulangan sebagai berikut:

A. ALIRAN LOGIKA

Aliran logika pada kendali perulangan digambarkan dengan *flowchart* berikut:



Grafik 5.1 Aliran Logika Pada Kendali Perulangan

Dalam pemrograman, dimungkinkan suatu input atau proses dilakukan berkali-kali atau berulang-ulang sesuai

keperluan. Untuk itu diperlukan perintah yang berhubungan dengan perulangan (looping). Dalam MATLAB dikenal dua buah jenis perulangan, yaitu **for** dan **while.** Perulangan for dipakai untuk mengulangi perintah (sekelompok perintah) secara pasti (tertentu), dimana banyaknya perulangan ini biasanya sudah ditentukan terlebih dahulu.

B. PERULANGAN for

Pernyataan loop for digunakan ketika diperlukan untuk mengulang atau mengiterasi pernyataan dalam suatu skrip atau fungsi. Dalam hal ini telah diketahui terlebih dahulu berapa banyak pengulangan yang akan diulang. Loop for membutuhkan nilai-nilai eksplisit agar dapat berfungsi. Nilai-nilai ini harus dinyatakan di dalam loop. Sintak MATLAB untuk loop for adalah:

Satu perulangan for Satu blok_perintah	Dua perulangan for atau lebih Satu blok perintah	Dua perulangan for atau lebih Dua perulangan blok perintah atau lebih
for x=array blok_perintah end	for x=array for y=array ... for n=array blok_perintah1 end end ... end	for x=array for y=array ... for n=array blok_perintah1 end end ... end for x=array for y=array ... for n=array blok_perintah2 end end ... end ... for x=array for y=array ... for n=array blok_perintah**n** end end ... end

Tabel 5.1 Perintah Pernyataan for...end

Dengan sintak tersebut, blok_perintah di antara for dan end akan dieksekusi satu kali setiap satu nilai dari array x.

Setiap satu nilai dari array x, blok_perintah2 akan diproses

sebanyak anggota array y. Begitu seterusnya sampai semua anggota array x terpakai. Ilustrasi, jika array x mempunyai *m* anggota dan array y mempunyai *n* anggota, maka banyaknya perulangan yang terjadi adalah *mn* kali. Dalam sintak tersebut blok_perintah1 dan blok_perintah3 bisa ada bisa tidak ada. Baik blok_perintah1 maupun blok_perintah3 hanya akan diulang berdasarkan banyaknya anggota array x.

C. PERULANGAN while

Pernyataan loop while digunakan ketika diperlukan untuk mengulang atau mengiterasi pernyataan dalam suatu skrip atau fungsi dimana banyak pengulangan belum diketahui secara pasti.

Satu perulangan while Satu blok_perintah	Dua perulangan while atau lebih Satu blok perintah	Dua perulangan while atau lebih Dua perulangan blok perintah atau lebih
inisialisasi while ekspresi blok_perintah end	inisialisasi while ekspresi1 while ekspresi2 ... while ekspresin blok_perintah1 end end ... end	inisialisasi while ekspresi1 while ekspresi2 ... while ekspresin blok_perintah1 end end ... end while ekspresi1 while ekspresi2 ... while ekspresin blok_perintah2 end end ... end ... while ekspresi1 while ekspresi2 ... while ekspresin blok_perintah**n** end end ... end

Tabel 5.2 Perintah Pernyataan while...end

Dalam sintak tersebut blok_perintah akan dieksekusi jika ekspresi bernilai benar (1). Jadi selama ekspresi masih bernilai benar, blok_perintah akan selalu diproses. Sebaliknya jika

ekspresi bernilai salah (0), maka proses perulangan dihentikan, atau blok_perintah tidak dieksekusi lagi.

Buatlah program – program berikut, simpan sebagai *m-file*, kemudian jalankan. Catat dan amati hasilnya.

D. PERULANGAN for MATLAB

Contoh 5.1:

Program berikut (**for1.m**) menunjukkan perulangan naik, tanpa menyebutkan nilai perubahan

```
for i=1:10
fprintf('%d\n',i);
end;
```

Contoh 5.2:

Program berikut (**for2.m**) menunjukkan perulangan turun dengan penurunan sebesar 2.

```
for i =20:-2:0
fprintf ('%d \n', i);
end;
```

Apa yang terjadi jika nilai akhir lebih besar dari nilai awal, tetapi perubahan negatif (perulangan turun)?.

Contoh 5.3:

```
for i=1:10
x(i)=sin(i*pi/10);
end
x
```

```
x =
0.3090 0.5878 0.8090 0.9511 1.0000 0.9511
0.8090
0.5878 0.3090 0.0000
```

Statement diatas dapat dijelaskan sebagai berikut:

Untuk i sama dengan satu sampai sepuluh, kerjakan seluruh statement sampai statement end. Pertama kali dieksekusi perulangan for untuk i=1, lalu i=2 dan seterusnya sampai i=10. Setelah i=10, perulangan for berhenti dan setiap perintah yang ada setelah statement end akan dikerjakan yang dalam contoh diatas menampilkan elemen-elemen x. Perulangan for tidak dapat dihentikan dengan mengubah nilai i di tengah-tengah perulangan,

Contoh 5.4:

```
for i=1:10
x(i)=sin(i*pi/10);
i=5;
end
x
x =
0.3090 0.5878 0.8090 0.9511 1.0000 0.9511
0.8090
0.5878 0.3090 0.0000
```

Statement 1:10 adalah statement standar pembuatan array. Setiap array **Matlab** yang valid dapat digunakan untuk perulangan for.

Contoh 5.5:

```
data=[1 2 3 4;5 6 7 8];
for n=data
x=n(1)-n(2)
end

x =
-4
x =
-4
x =
-4
x =
-4
```

Biasanya perulangan *for* dapat dibuat, sehingga berada dalam perulangan *for* yang lain sebanyak yang diinginkan.

Contoh 5.6:

```
for i=1:10
   for j=1:5
U(i,j)=i^2+j^2;
   end
end
U
U =
2 5 10 17 26
5 8 13 20 29
10 13 18 25 34
```

```
17  20  25  32  41
26  29  34  41  50
37  40  45  52  61
50  53  58  65  74
65  68  73  80  89
82  85  90  97  106
101  104  109  116  125
```

Dengan cara diatas kita membuat suatu matriks dimulai dengan membuat elemen matriks baris 1 elemen kolom 1 sampai 5, dilanjutkan dengan membuat elemen matriks baris 2 kolom 1 sampai 5 dan seterusnya sampai baris ke 10.

E. PERULANGAN PADA MATRIKS

Pada perulangan diatas, perulangan diterapkan pada vektor. Perulangan dapat diterapkan pada matriks. Operasi perulangan biasanya dilakukan perbaris. Pada tiap baris, dilakukan operasi perulangan per kolom. Program berikut melakukan operasi perulangan pada matriks. Pada program menampilkan elemen – elemen matriks beserta posisinya dalam matriks.

Contoh 5.7:

```
disp('Program matrik 1(mat1.m)')
% Buat matriks
p = input('cacah baris = ');
q = input('cacah kolom = ');
a = 10 * rand (p, q);
%tampilan elemen matriks
```

```
for i = 1:p
for j = 1:q
fprintf ('[%d,%d]= %.3f\n', i, j, a(i, j));
end
end
```

Contoh 5.8:

```
Program matrik 2(mat2.m)
%Metode left division dengan input matrik tdk
langsung
%Elemen matrik diinputkan satu per satu
clc;
clear;
v=input('Berapa var = ');
disp('Inputkan elemen matrik a ');
disp('==========================');
for i=1:v
   for j=1:v
      fprintf('a %d%d ',i,j);
      a(i,j)=input(' = ');
   end
end
disp('Inputkan matrik b ')
disp('=================');
for i=1:v
   fprintf('b %d%d ',i,j);
   b(i,1)=input('= ');
end
```

```
clc;
disp('Matrik a = ');disp(a);
disp('Matrik b = ');disp(b);
c=a\b;
disp('Penyelesaiannya adalah =');
disp(c);
```

F. PERULANGAN while MATLAB

Contoh 5.9:

```
%Program perulangan while
i=6;
while i>5
i=i+1;
    disp(i);
end;
```

Contoh 5.10:

```
%Program perulangan while
limit = 20;
sum = 0;
i = 1;
while i <= limit
i=i+1;
sum = sum+i ;
end
fprintf('sum = %d\n',sum);
```

Contoh 5.11:

Program berikut menunjukkan perulangan bersyarat untuk menampilkan bilangan 1 hingga 10.

```
i = 1    %inisialisasi
while ( i <= 10) %syarat perulangan
fprintf ('%d \n',i);
i = i+1; %pertambahan positif
end
```

Pada program di atas, syarat terhadap pencacah berlaku sebagaimana batas akhir pada pernyataan for.

Contoh 5.12:

Program berikut menjumlahkan akar kuadrat dari bilangan genap, hingga diperoleh jumlah total tertentu. Padanya, syarat perulangan tidak berkenaan dengan nilai pencacah, tapi berkenaan dengan hasil operasi.

```
n=0; %inisialisasi
jumlah = 0;
while (jumlah <= 100) % syarat perulangan
jumlah=jumlah + sqrt(n);
n = n+2; % pertambahan positif
end;
fprintf('total = %.3f dengan n = %d \n',
jumlah, n);
```

LATIHAN 5 MATLAB

1. Kapan sebaiknya menggunakan perulangan for dan kapan menggunakan perulangan while?

   .....

   .....

   .....

   .....

   .....

2. Buatlah program MATLAB menggunakan fungsi ***Perulangan Berbatas (for)*** untuk menampilkan n suku pertama dan jumlah suku ke-n dengan suku ke-1 adalah 3 dan suku ke-2 adalah 9. Untuk

   a) Barisan dan deret aritmatika

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

.....

b) Barisan dan deret geometri

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Diketahui suatu program sebagai berikut :

```
for x=5:-2:1
for y=1:4
B(x,y)=x^2-x*y+y^2
end
A=3*x-4
end
```

a) Apakah penulisan program tersebut sudah benar?.Jika ada kesalahan perbaikilah program tersebut!.

.....

.....

.....

.....

.....

.....

.....

.....

b) Apakah hasil dari program tersebut? Jelaskan!

.....

.....

.....

.....

.....

.....

.....

.....

4. Diketahui sebuah program sebagai berikut:

```
awal=2;
beda=5;
urutan=3;
if urutan<0
sukunya=awal+(urutan-1)*beda
else
for urutan=0:0.5:3
kakinya=awal+(urutan-1)*beda
end
end
```

a) Apakah penulisan program tersebut sudah benar?.Jika ada kesalahan perbaikilah program tersebut!.

.....

.....

.....

.....

.....

.....

.....

.....

.....

b) Apakah hasil dari program tersebut? Jelaskan!

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

5. Tentukan hasil dari m.file berikut, dengan menjelaskan prosesnya:

a)
```
total = 0;
for i = 1:6;
i = i + 2;
```

```
total = total + i
end
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

b)
```
x= [-2:3:10];
n = 0;
while n <= length(x) - 1
n = n + 1;
if x(n) > 0
    fprintf('%f \n', x(n))
end
end
```

.....

.....

.....

.....

.....

.....

DAFTAR PUSTAKA

Boyce, W. E., & DiPrima, R. . (2000). Elementary Differential Equation and Boundary Value Problems. New York: John Willey & Sons.

Edwards, C., & Penney, D. (2008). Differential Equation and Computer Modeling. New Jersey: Prentice Hall.

Hanselman, D., & Littlefield, B. (2000). MATLAB Bahasa Komputasi Teknis. Yogyakarta: Penerbit Andi.

Howard, P. (2007). Solving ODE in MATLAB. Retrieved March 13, 2017, from http://www.math.tamu.edu/reu/comp/matode.pdf

Irawan, Feriza A. 2012. Buku Pintar Pemrograman MATLAB. Yogyakarta: MediaKom.

Martono, K. (1999). Kalkulus. Jakarta: Erlangga.

Optimal Control. New Jersey: Prentice Hall.Team Labkomputer UMM. 2011. Modul Praktikum MATLAB. Malang: Universitas Muhammadiyah Malang.

Penney, D. ., & Edward C, H. (2000). Differential Equations and Boudary Value Problems. New Jersey: Prentice Hall.

Purcell, E. J., & Varberg, D. (1994). Kalkulus dan Geometri Analitis (5th ed.). Jakarta: Erlangga.

Rachmatin, D. (2009). Petunjuk Praktikum Program Aplikasi Matematika. Bandung: UPI.

Teguh Widiarsono. 2005. Tutorial Praktis Belajar Matlab. Jakarta.

Widiarsono, T. (2005). Tutorial Praktis Belajar Matlab. Jakarta.

Zaelani, A., Cunayah, C., & Irawan, E. I. (2008). 1700 Bank Soal Bimbingan Pemantapan Matematika. Bandung: Yrama Widya.

Zahnur. 2013. Modul Praktikum Komputasi Matematika. Aceh: Universitas Syiah Kuala

Zhou, K., Doyle, J. C., & Glover, K. (2000). Robust and

PROFIL PENYUSUN

Malim Muhammad adalah Dosen Pendidikan Matematika Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Purwokerto yang lahir di Medan, 07 Februari 1986. Bidang Ilmu yang dikuasai adalah Data Science, Big Data, Komputasi Statistik, Aplikasi Komputer Matematika, Statistika Deskriptif, Statistika Inferensia, dan Pemodelan Matematika. Beliau sudah mengeluarkan Buku Ajar Statistika Inferensia Tahun 2017. Beliau aktif mengikuti seminar baik Nasional dan Internasional serta menghasilkan banyak artikel ilmiah terindeks Scopus, WoS, Sinta dan Lainnya. Beliau pernah mendapatkan presetasi sebagai Pemakalah Terbaik (Best Paper) pada Seminar Nasional Pendidikan Matematika Universitas Ahmad Dahlan (SENDIKMAD) 2018.