

DASAR-DASAR MATLAB



DISUSUN OLEH :

NAMA : RIZQILLAH

NIM : 1957301020

KELAS/SEMESTER : TI 2C/3

MATA KULIAH : Prak. Metode Numerik

No. Praktikum : 01/PMetNum/IT/2020

PRODI : Teknik Informatika

LABORATORIUM INFORMATION PROCESSING

TEKNOLOGI INFORMASI DAN KOMPUTER

POLITEKNIK NEGERI LHOKSEUMAWE

TAHUN 2020

HALAMAN PENGESAHAN

Telah dilaksanakan Praktikum ke-1 Mata Kuliah Metode Numerik di Laboratorium Information Processing pada hari Jum'at, Tanggal 25 September 2020 s/d 28 September 2020 dengan Materi Praktikum :

DASAR-DASAR MATLAB

Oleh

Nama : RIZQILLAH

Nim : 1957301020

Kelas : TI 2C

Disetujui Oleh :

Dosen Pengasuh Mata Kuliah

Nilai

Mulyadi, ST.,M.Eng
Nip. 19730723 2002121 1 001

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iii
DASAR-DASAR MATLAB	1
1.1 Tujuan	1
1.2 Dasar Teori.....	1
1.2.1 Apa sih Matlab itu?	1
1.2.2 Kelengkapan pada sistem Matlab.....	2
1.3 Percobaan.....	4
1.3.1 Perangkat yang diperlukan	4
1.3.2 Langkah percobaan.....	4
1.3.3 Menentukan direktori tempat bekerja	5
1.3.4 Penggambaran grafik.....	8
1.3.5 Menyusun program sederhana.....	9
TUGAS	12
1.4 Analisa	17
1.5 Kesimpulan	21
DAFTAR PUSTAKA	22

DAFTAR GAMBAR

Gambar 1.1 Icon Matlab pada Desktop PC.....	4
Gambar 1.2 Tampilan awal Matlab.....	4
Gambar 1.3 Membuat folder baru tempat program	5
Gambar 1.4 Grafik menggunakan perintah plot.....	8
Gambar 1.5 Grafik menggunakan perintah stem	9
Gambar 1.6 Tampilan Editor.....	9
Gambar 1.7 Tampilan Save File and Run	10
Gambar 1.8 Grafik Program coba_1.m	10
Gambar 1.9 Grafik Program coba_2.m	11
Gambar 1.10 Grafik Program coba_3.m	12
Gambar 1.11 Tampilan Toolbar Matlab.....	13
Gambar 1.12 Penggunaan Grafik Mesh	13
Gambar 1.13 Penggunaan Grafik Polar	14
Gambar 1.14 Grafik 2 Sinyal Bersamaan	15
Gambar 1.15 Grafik dengan Perintah Subplot	16
Gambar 1.16 Grafik dengan Perintah Figure	17
Gambar 1.17 Grafik Sinyal Waktu kontinyu	18
Gambar 1.18 Fungsi Waktu Diskrit	18
Gambar 1.19 Penggabungan Antara 2 Sinyal Yang Berbeda	19
Gambar 1.20 Satu Figure Dalam Dua Grafik.....	19
Gambar 1.21 Memisahkan Grafik Menjadi 2 Figure	20

PRAKTIKUM 1

DASAR-DASAR MATLAB

1.1. Tujuan Praktikum

Mampu mengoperasikan matlab dan memanfaatkannya sebagai perangkat simulasi untuk praktikum sinyal dan sistem

1.2. Dasar Teori

1.2.1. Apa sih Matlab itu?

MATLAB adalah sebuah bahasa dengan (high-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan Matlab meliputi bidang–bidang:

- Matematika dan Komputasi
- Pembentukan Algorithm
- Akusisi Data
- Pemodelan, simulasi, dan pembuatan prototipe
- Analisa data, explorasi, dan visualisasi
- Grafik Keilmuan dan bidang Rekayasa

MATLAB merupakan suatu sistem interaktif yang memiliki elemen data dalam suatu array sehingga tidak lagi kita dipusingkan dengan masalah dimensi. Hal ini memungkinkan kita untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, khususnya yang berhubungan dengan matrix dan formulasi vektor, yang mana masalah tersebut merupakan momok apabila kita harus menyelesaikannya dengan menggunakan bahasa level rendah seperti Pascall, C dan Basic. Nama MATLAB merupakan singkatan dari matrix laboratory. MATLAB pada awalnya ditulis untuk memudahkan akses perangkat lunak matrik yang telah dibentuk oleh LINPACK dan EISPACK. Saat ini perangkat MATLAB telah menggabung dengan LAPACK dan BLAS library, yang merupakan satu kesatuan dari sebuah seni tersendiri dalam perangkat lunak untuk komputasi matrix. Dalam lingkungan perguruan tinggi teknik, Matlab

merupakan perangkat 2 standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan. Di industri, MATLAB merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tinggi, pengembangan dan analisisnya.

Fitur-fitur MATLAB sudah banyak dikembangkan, dan lebih kita kenal dengan nama toolbox. Sangat penting bagi seorang pengguna Matlab, toolbox mana yang mendukung untuk learn dan apply teknologi yang sedang dipelajarinya. Toolbox toolbox ini merupakan kumpulan dari fungsi-fungsi MATLAB (M-files) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah dalam kelas particular. Area-area yang sudah bisa dipecahkan dengan toolbox saat ini meliputi pengolahan sinyal, system kontrol, neural networks, fuzzy logic, wavelets, dan lain-lain.

1.2.2. Kelengkapan pada sistem Matlab

Sebagai sebuah system, MATLAB tersusun dari 5 bagian utama:

1. **Development Environment.** Merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah graphical user interfaces (GUI). Termasuk didalamnya adalah MATLAB desktop dan Command Window, command history, sebuah editor dan debugger, dan browsers untuk melihat help, workspace, files, dan search path.
2. **MATLAB Mathematical Function Library.** Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: sum, sin, cos, dan complex arithmetic, sampai dengan fungsi-fungsi yang lebih komplek seperti matrix inverse, matrix eigenvalues, Bessel functions, dan fast Fourier transforms.
3. **MATLAB Language.** Merupakan suatu high-level matrix/array language dengan control flow statements, functions, data structures, input/output, dan fitur-fitur object-oriented programming. Ini memungkinkan bagi kita untuk melakukan kedua hal baik

"pemrograman dalam lingkup sederhana " untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasi yang kompleks.

4. **Graphics.** MATLAB memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatu grafik. Didalamnya melibatkan high-level functions (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, image processing, animation, dan presentation graphics. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi anda untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan graphical user interfaces pada aplikasi MATLAB anda.
5. **MATLAB Application Program Interface (API).** Merupakan suatu library yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinteraksi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan routines dari MATLAB (dynamic linking), pemanggilan MATLAB sebagai sebuah computational engine, dan untuk membaca dan menuliskan MAT-files.

1.3. Percobaan

1.3.1. Perangkat yang diperlukan

- PC yang dilengkapi dengan perangkat multimedia (sound card, Microphone, Speaker active, atau headset)
- Sistem Operasi Windows dan Perangkat Lunak Matlab yang dilengkapi dengan tool box DSP

1.3.2. Langkah Percobaan

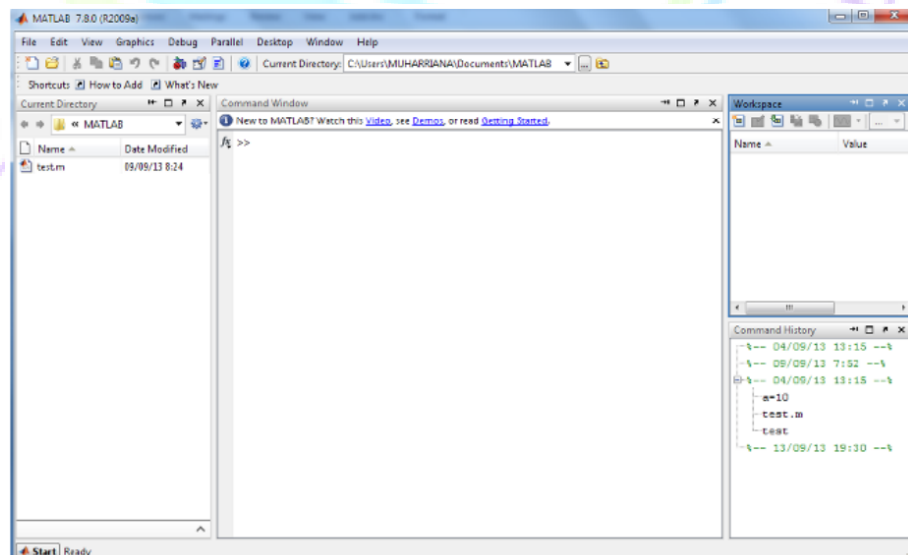
Memulai Matlab

- Memperhatikan Dekstop pada layar monitor PC, memulai MATLAB dengan melakukan double-clicking pada shortcut icon MATLAB



Gambar 1.1 Icon Matlab pada Desktop PC

- Selanjutnya akan mendapatkan tampilan seperti pada Gambar berikut ini.

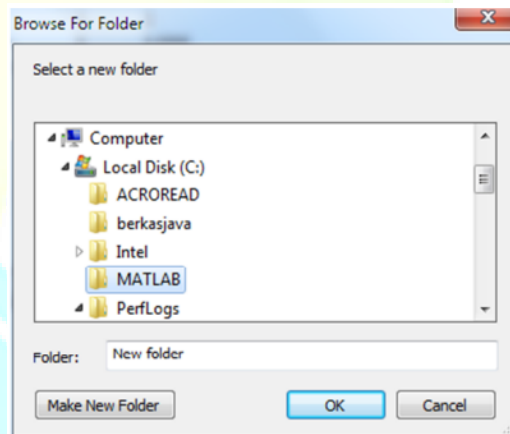


Gambar 1.2 Tampilan awal Matlab

Sedangkan untuk mengakhiri sebuah sesi MATLAB, bisa melakukan dengan dua cara, pertama pilih File -> Exit MATLAB dalam window utama MATLAB yang sedang aktif, atau cara kedua lebih mudah yaitu cukup ketikkan type quit dalam Command Window.

1.3.3. Menentukan direktori tempat bekerja

- Membuat direktori baru dengan mengarahkan pointer mouse pada kotak bertanda ... yang ada disebelah kanan tanda panah kebawah (yang menunjukkan folder yang sedang aktif). Lalu memilih new directory, selanjutnya mengetikkan “MATLAB”, dan diikuti dengan click Ok



Gambar 1.3 Membuat folder baru tempat program

Memulai Perintah Sederhana

- Menentukan variable scalar dengan cara melakukan pengetikan seperti berikut:

```
>> x=2
x =
    2
>> y=3
y =
    3
>> z=x+y
z =
    5
>> |
```

- Mendefinisikan dua buah vector, yaitu vector x dan vector y:

```
>> x=[1 2 3]
x =
     1     2     3
>> y=[4 5 6]
y =
     4     5     6
>> y(1)
ans =
     4
>> y(2)
ans =
     5
>> y(3)
ans =
     6
```

- Menjumlahkan kedua vektor

```
>> x+y
ans =
     5     7     9
>> |
```

- Menghitung inner product

```
>> x*y'
ans =
    32
```

- Memeriksa nilai y transpose

```
>> y'
```

```
ans =
```

```
4
5
6
```

- Melakukan perkalian antar elemen vektor

```
>> x.*y
```

```
ans =
```

```
4    10    18
```

- Mendefenisikan suatu matrix

```
>> A=[1 2 3
      4 5 6
      7 8 9];
```

- Mengalikan matrix A dengan transpose x

```
>> A*x'
```

```
ans =
```

```
14
32
50
```

- Mengalikan matrix A dengan matrix B

```
>> B=[1 2 3 4
      5 6 7 8
      7 6 5 4];
```

```
>> A*B
```

```
ans =
```

```
32    32    32    32
71    74    77    80
110   116   122   128
```

- Menjumlahkan matrix A dengan matrix B

```
>> A+B
```

```
??? Error using ==> plus
Matrix dimensions must agree.
```

- Mendefenisikan matrik nol dengan dimensi 3 baris dan 6 kolom

```
>> zeros(3, 6)
```

```
ans =
```

```
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

- Mendefenisikan matrik bernilai 1 dengan dimensi 3 baris dan 6 kolom

```
>> ones(3, 6)
```

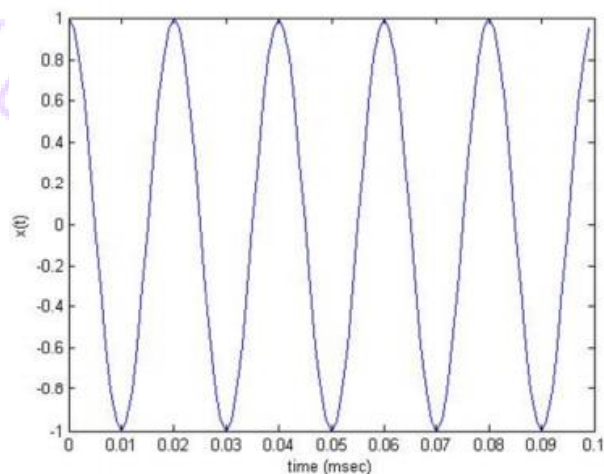
```
ans =
```

```
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
```

1.3.4. Penggambaran Grafik

- Menampilkan grafik dengan perintah plot:

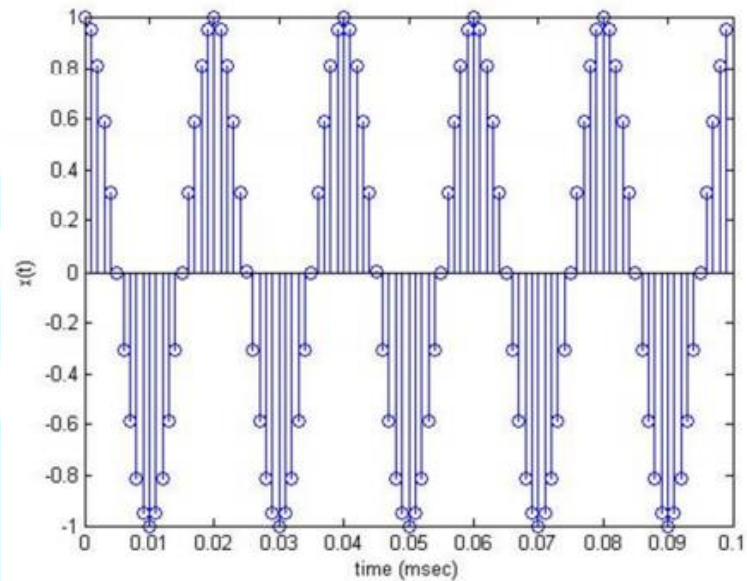
```
>> time = [0:0.001:0.099];
x = cos(0.1*pi*(0:99));
plot(time,x)
xlabel('time (msec)')
ylabel('x(t)')
```



Gambar 1.4 Grafik menggunakan perintah plot

- Menampilkan grafik dengan perintah stem

```
>> time = [0:0.001:0.099];
x = cos(0.1*pi*(0:99));
stem(time,x)
xlabel('time (msec)')
ylabel('x(t)')
```

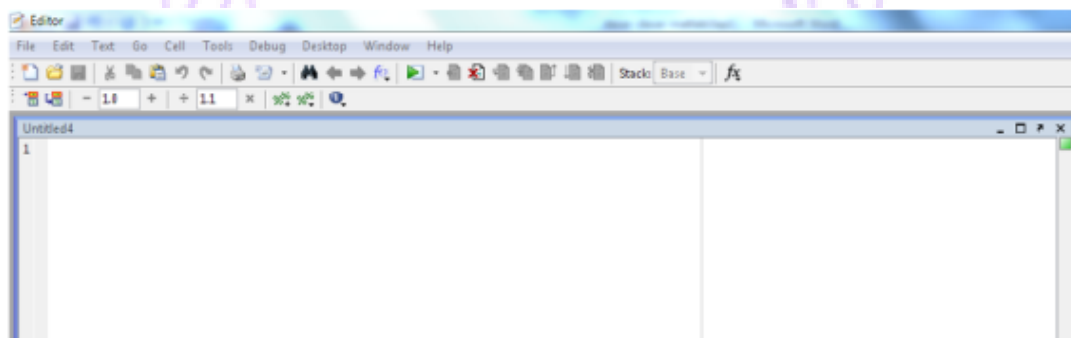


Gambar 1.5 Grafik menggunakan perintah stem

1.3.5. Menyusun program sederhana

Program Pertama

- Menekan click pada icon "New M-File" icon in the Matlab toolbar
Selanjutnya mendapatkan sebuah tampilan Matlab Editor yang masih kosong seperti ini.

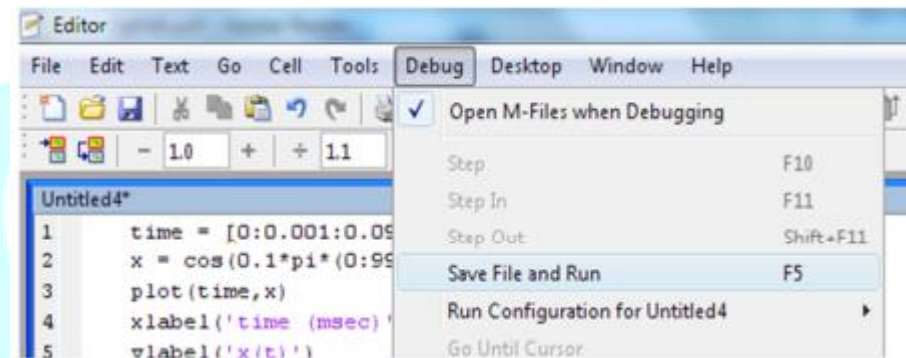


Gambar 1.6 Tampilan Editor

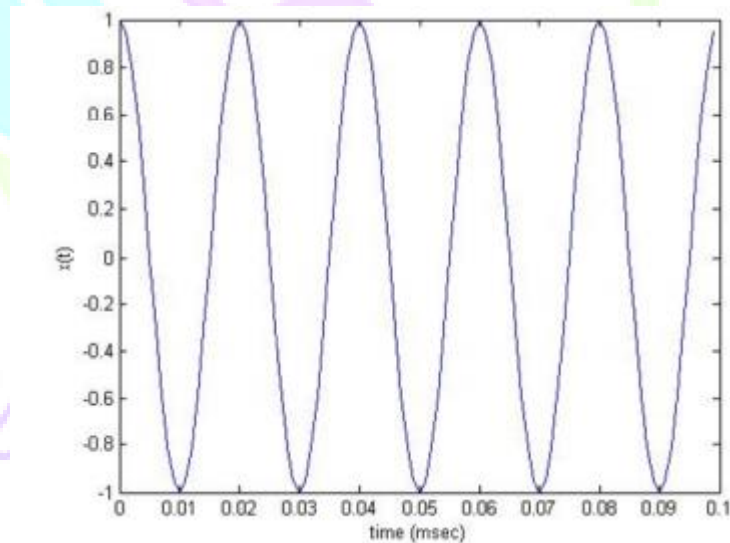
- Membuat program seperti pada contoh sebelumnya

```
time = [0:0.001:0.099];
x = cos(0.1*pi*(0:99));
plot(time,x)
xlabel('time (msec)')
ylabel('x(t)')
```

- Melanjutkan dengan menekan toolbar Debug - pilih Save and Run. - menuliskan nama program coba_1.m



Gambar 1.7 Tampilan Save File and Run



Gambar 1.8 Grafik Program coba_1.m

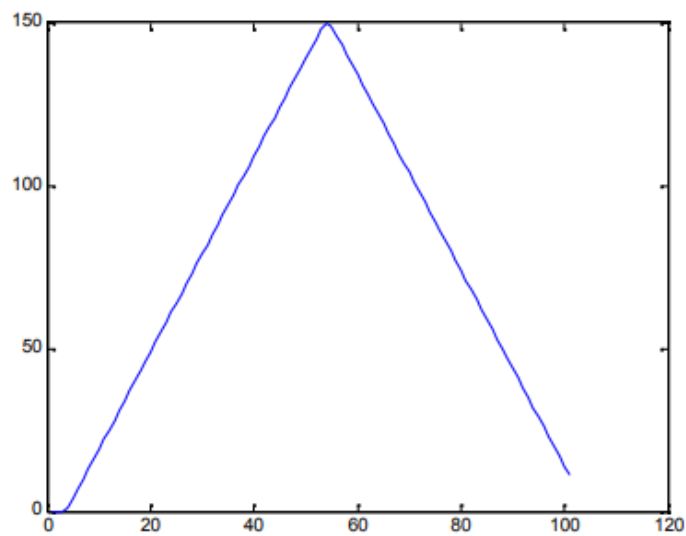
Program Kedua

- Mengetikkan program berikut pada Matlab editor dan menyimpan dengan nama coba_2.m

```

x(1:52) = [0 0 1:1:50];
x(53:102) = [50:-1:1];
h = [1 2];
for n = 3:101,
    y(n) = 0;
    for m = 1:2,
        y(n) = y(n) + h(m) * x(n-m);
    end
end
plot(y)

```



Gambar 1.9 Grafik Program coba_2.m

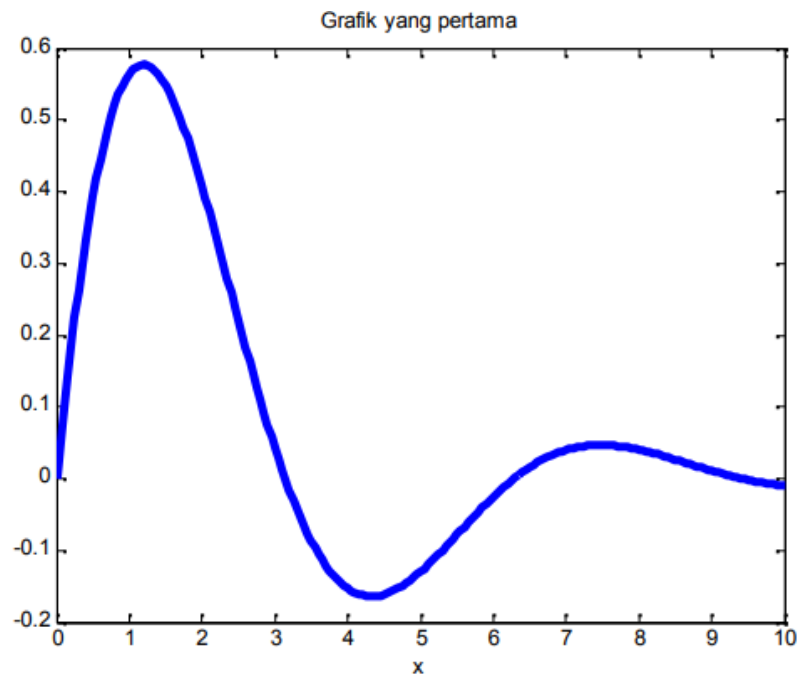
Program Ketiga

- Mengetikkan program berikut pada Matlab editor dan meyimpan file dengan nama coba_3.m

```

n=201;
delx=10/(n-1);
for k=1:n
    x(k)=(k-1)*delx;
    y(k)=sin(x(k))*exp(-0.4*x(k));
end
%plot(x,y)
plot(x,y,'linewidth',4)
title('Grafik yang pertama')
xlabel('x');ylabel('y');

```



Gambar 1.10 Grafik Program coba_3.m

TUGAS

1. Dari contoh-contoh program yang sudah anda jalankan, coba berikan penjelasan arti setiap perintah terhadap output yang dihasilkannya.

Jawaban :

- Penulisan $y(n)$ menunjukkan bahwa kita ingin mengetahui *elemen* ke n dari suatu matrik
- Simbol titik koma (;) digunakan untuk menunjukkan matrik kolom.
- `x=ones(1,10)` memiliki arti nilai satu sebanyak 1 baris dan 10 kolom
- `x=zeros(3,1)` memiliki arti nilai nol sebanyak 3 baris dan 1 kolom
- `rand` untuk bilangan acak terdistribusi *uniform*
- `randn` untuk bilangan acak terdistribusi *Gaussian*
- `plot` digunakan untuk memunculkan grafik
- `hold on` digunakan untuk membuka *slot* sehingga dalam satu grafik bisa terdapat banyak sinyal dan bila sudah selesai diakhiri dengan `hold off`
- `load` untuk memanggil *file*
- `clc` perintah untuk menyimpan program di memory agar bisa dipanggil ulang
- `exit/quit` perintah untuk mengakhiri program ketika selesai
- `sqrt` perintah menghitung akar 2 dari suatu bilangan
- `clr` perintah untuk menghapus program yang diketik sehingga tidak muncul pada command window

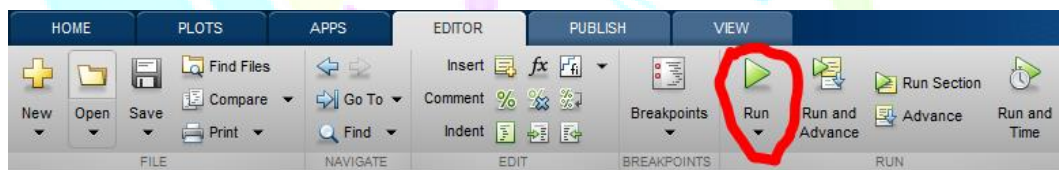
2. Coba anda cari bagaimana cara menampilkan grafik untuk tampilan tiga dimensi dan grafik polar.

Jawaban :

- a. Untuk menampilkan Grafik, terlebih dahulu anda membuat syntax untuk grafik tersebut, contohnya seperti syntax untuk grafik polar sebagai berikut :

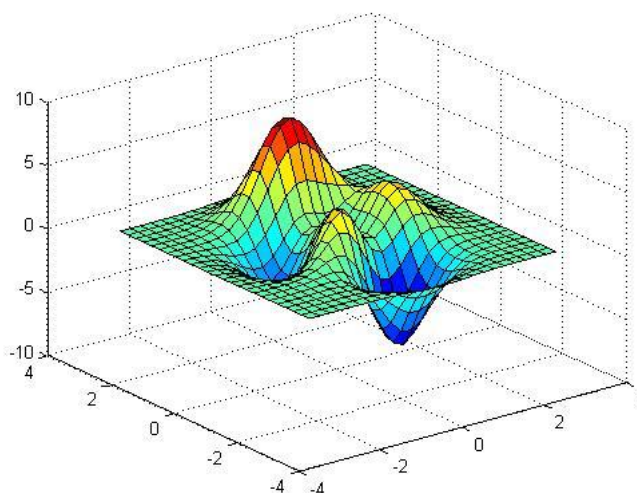
```
% Contoh plot polar
t=0:.01:2*pi;
subplot(1,2,1)
polar(t,abs(sin(2*t)).*cos(2*t)), 'b');
title('abs(sin(2*t)).*cos(2*t)')
subplot(1,2,2)
polar(t,(sin(2*t)), 'r');
title('sin(2*t)')
```

- b. Kemudian anda cukup menyimpan file tersebut dengan ekstensi (.m) kemudian anda tinggal menekan pada bagian run untuk menampilkan grafik dari syntax tersebut.



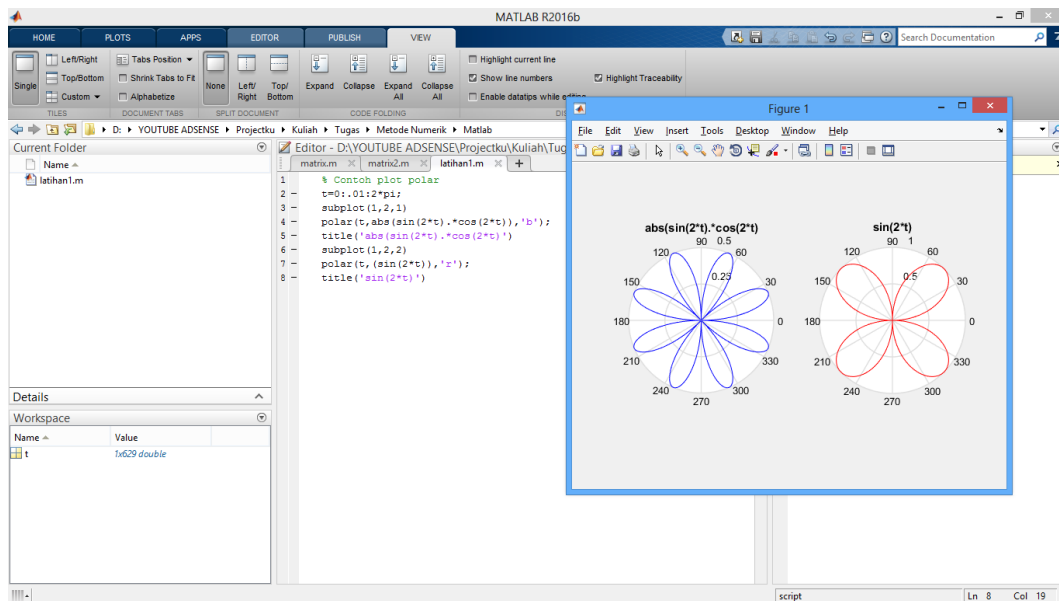
Gambar 1.11 Tampilan Toolbar Matlab

- c. Membuat grafik tampilan 3 dimensi dapat menggunakan perintah *mesh*, *surface*, *contour*, *quiver*, dan lain sebagainya



Gambar 1.12 Penggunaan Grafik Mesh

d. Membuat grafik *polar* dapat menggunakan perintah *polar*



Gambar 1.13 Penggunaan Grafik Polar

3. 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.

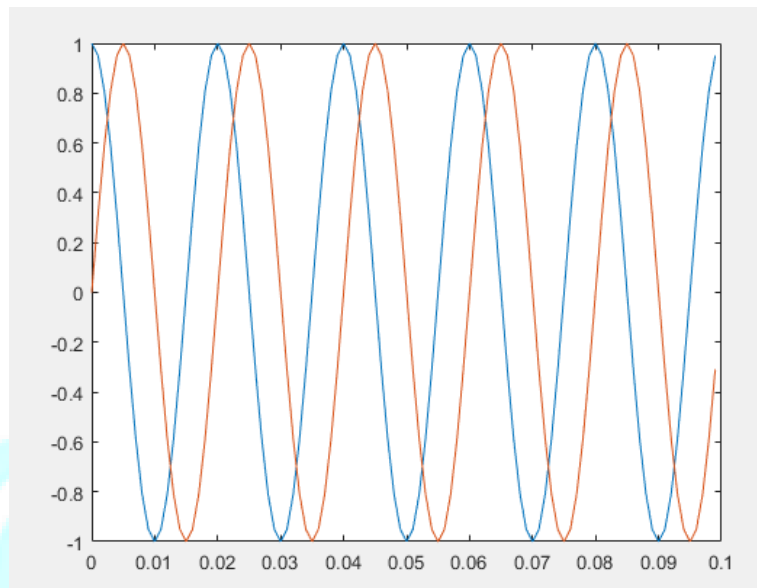
Jawaban :

Untuk menggabungkan 2 grafik sinyal menggunakan perintah *hold on* dan *hold off* kemudian masukan bentuk grafik sinusoidal.

```

time=[0:0.001:0.099];
x=cos(0.1*pi*(0:99));
y=sin(0.1*pi*(0:99));
plot(time,x)
hold on
plot(time,y)
hold off

```



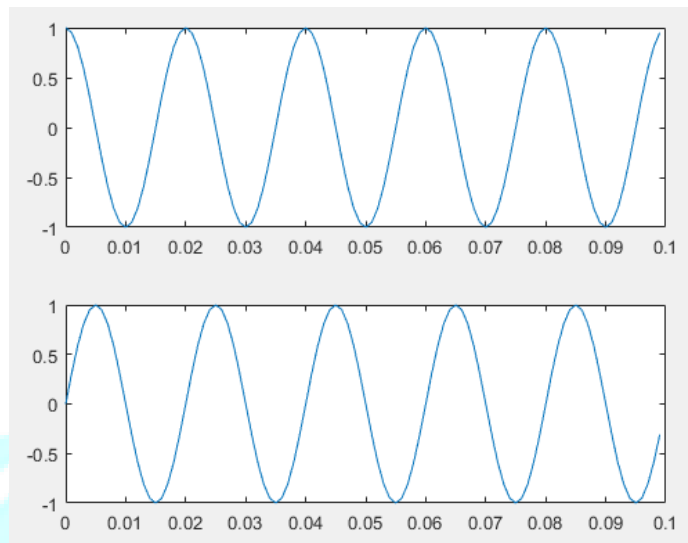
Gambar 1.14 Grafik 2 Sinyal Bersamaan

4. Bagaimana cara menampilkan lebih dari satu grafik dalam satu tampilan? Misalnya anda gunakan fungsi pada soal ke-3, satu fungsi ditampilkan diatas dan fungsi lainnya di bagian bawah.

Jawaban :

Untuk menjalankan program yang membuat 2 grafik dalam satu *figure* dapat menggunakan perintah *subplot* yang memiliki fungsi untuk membagi bagian dari beberapa grafik yang di tentukan.

```
time=[0:0.001:0.099];
x=cos(0.1*pi*(0:99));
y=sin(0.1*pi*(0:99));
plot(time,x)
hold on
plot(time,y)
hold off
subplot(211);plot(time,x)
subplot(212);plot(time,y)
```



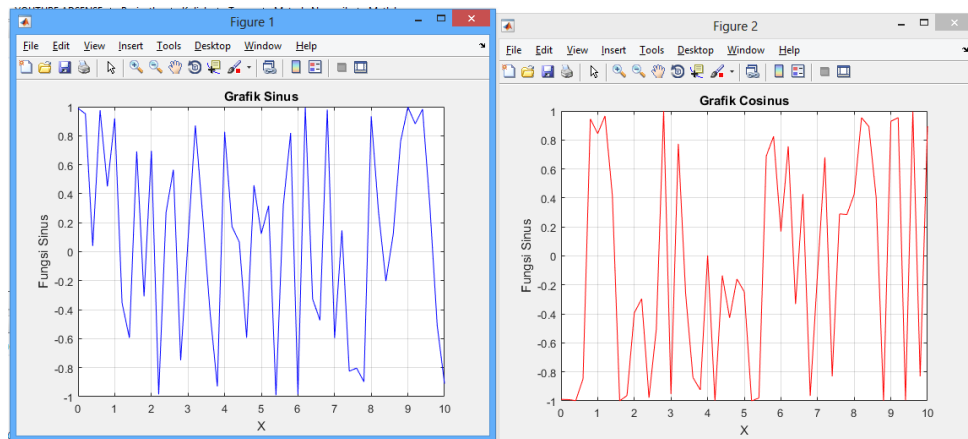
Gambar 1.15 Grafik dengan Perintah Subplot

5. Bagaimana cara menampilkan dua fungsi dimana masing-masing fungsi disajikan dalam grafik berbeda. Misalnya anda gunakan contoh kasus pada soal ke-3, fungsi pertama anda tampilkan pada figure(1), sementara fungsi kedua anda tampilkan pada figure(2).

Jawaban :

Untuk memisahkan grafik menjadi dua *figure* dapat menggunakan perintah *figure(n)* kemudian berikan perintah *plot*.

```
n=5
x=0 : 1/n :10
A=(2*x+2).^3
B=(9*x.^5)+3
C=sin(A)
D=cos(B)
figure(1)
plot(x,C,'b-')
hold on
title('Grafik Sinus')
xlabel('X'),ylabel('Fungsi Sinus')
grid
figure(2)
plot(x,D,'r-')
hold on
title('Grafik Cosinus')
xlabel('X'),ylabel('Fungsi Sinus')
grid
hold off
```



Gambar 1.16 Grafik dengan Perintah Figure

1.4. Analisa

Matlab adalah bahasa pemrograman yang berbeda dengan bahasa pemrograman tingkat tinggi seperti visual basic, Delphi dan lain-lain, matlab memiliki kelebihan sendiri yaitu dapat menyelesaikan persoalan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan.

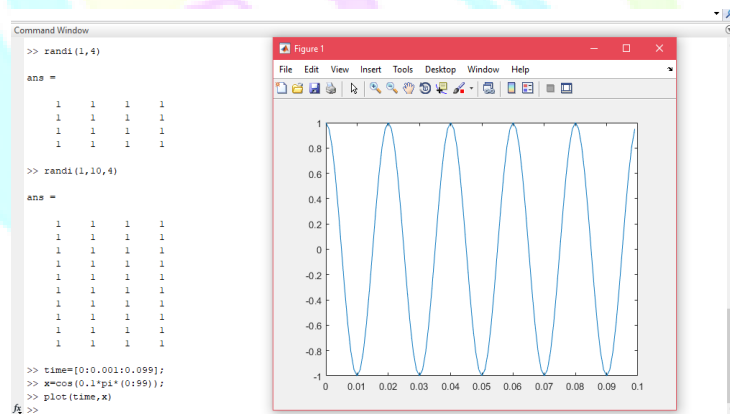
Matlab memiliki aturan/kaidah sendiri dalam penulisan perintah programnya misalnya kita ingin membuat persamaan matriks maka kita harus mengawalinya dengan tanda [dan diakhiri tanda] serta tanda ; sebagai pemisah baris satu dan baris selanjutnya. Dan untuk mengetahui invers atau determinan dari matriks tersebut kita tinggal berikan perintah inv atau det. Bukan hanya matriks, bisa juga turunan, integral, laplace, laplace balik dan lain-lain.

Selain dapat menyelesaikan perhitungan matematik, MatLab juga dapat menyajikan data input dalam tampilan Grafik. Grafik yang disajikan dapat berupa 3D atau 2D berupa plot, stem, bar atau stair. Tergantung bagaimana input yang kita berikan. Bahkan matlab juga dapat bermanfaat untuk pengoperasian fungsi laplace dan membuat diagram blok. PID controller yang merupakan salah satu jenis pengaturan yang dilakukan dapat disimulasikan dengan metode Simulink pada matlab. Setelah melakukan banyak percobaan diatas saya baru mengerti bahwa dalam penulisan perintah program pada

matlab diperlukan ketelitian yang sangat tinggi karena salah sedikit saja maka pesan error akan ditampilkan , tapi disini matlab lagi-lagi memiliki kelebihan ketika mengalami kesalahan atau error maka matlab akan memberitahu kita baris mana yang mengalami kesalahan penulisan program tersebut, sehingga kita tidak susah lagi untuk mencari kesalahan dimana yang terjadi.

Membuat Grafik

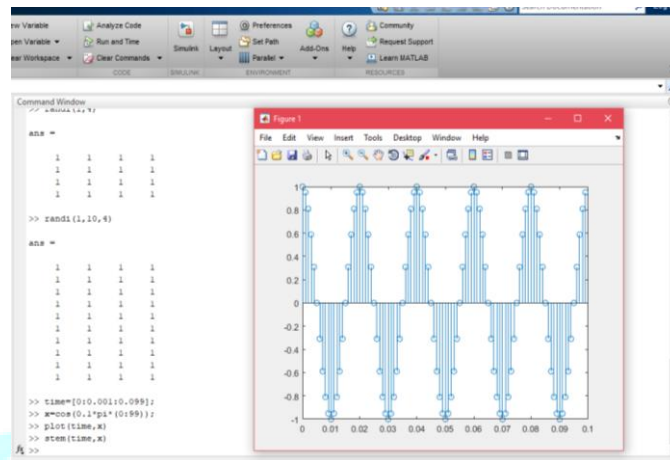
Matlab bukan hanya digunakan untuk pengoperasian-pengoperasian dasar dari matematika, namun matlab juga bisa digunakan untuk menampilkan bentuk grafik sinyal dari suatu bentuk sinyal yang kompelek, dapat dilihat pada Gambar 1.17 Grafik Sinyal Waktu kontinyu adalah salah satu contoh penggunaan matlab untuk melihat sinyal.



Gambar 1.17 Grafik Sinyal Waktu kontinyu

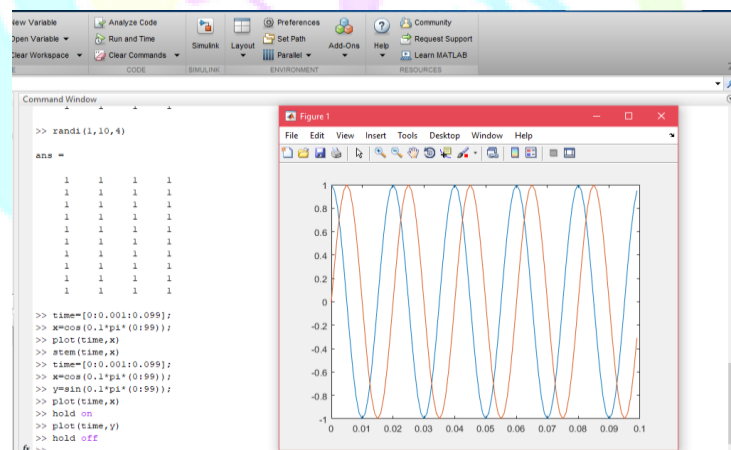
Percobaan kali ini untuk mengetahui bentuk sinyal dari suatu rumus fungsi atau *variable* fungsi *time* pada Gambar 1.18 digunakan untuk menampilkan grafik yang berbentuk waktu, dan *plot* digunakan untuk menampilkan hasil grafik pada program yang di masukan.

Percobaan selanjutnya mengubah sinyal kontinyu menjadi sinyal diskrit, cara mengubahnya menambahkan perintah *stem* untuk mengubah bentuk sinyal menjadi distrik seperti pada Gambar 1.18 Fungsi Waktu Diskrit



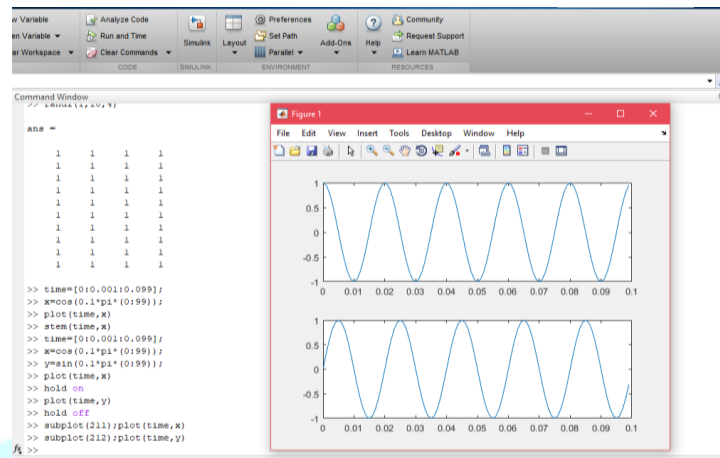
Gambar 1.18 Fungsi Waktu Diskrit

Matlab tidak hanya digunakan untuk menampilkan 1 bentuk sinyal saja namun juga bisa digunakan untuk menggabungkan dua bentuk sinyal, untuk contoh dapat dilihat pada Gambar 1.19 Penggabungan Antara 2 Sinyal Yang Berbeda.



Gambar 1.19 Penggabungan Antara 2 Sinyal Yang Berbeda

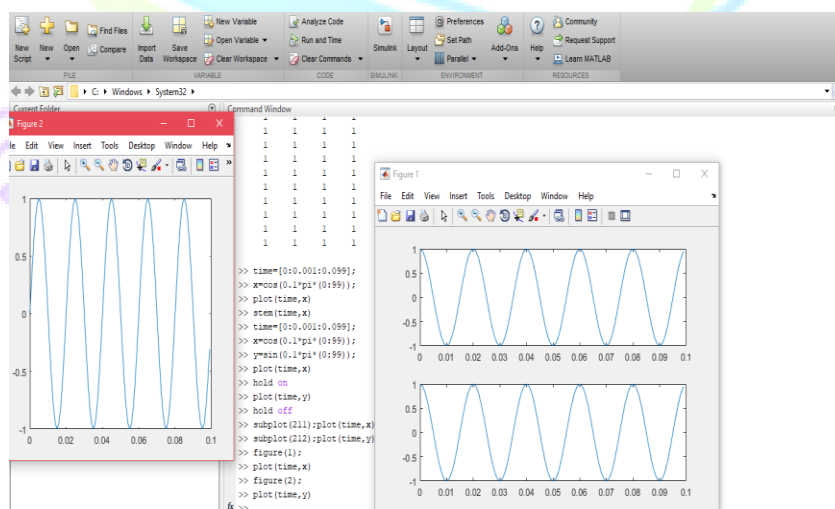
Untuk penggunaan lebih dari 1 grafik yang digabungkan hanya dengan memasukan 2 variabel berbeda menggunakan fungsi *hold on* setelah *plot* satu grafik sinyal berfungsi untuk membuat gambar sinyal 1 bisa dimasukkan gambar sinyal 2 jika sudah selesai maka bisa dimatikan dengan rumus *hold off*. Percobaan selanjutnya matlab juga dapat digunakan untuk memunculkan dua sinyal dalam satu frame. Dapat dilihat pada Gambar 1.20 Satu *Figure* Dalam 2 Grafik



Gambar 1.20 Satu *Figure* Dalam Dua Grafik

Percobaan ini menggunakan fungsi *subplot* untuk membuat dua grafik menjadi dalam satu *frame*, dan di masukan angka 211 dan 212 artinya angka 2 digunakan untuk membuat 2 grafik dalam satu *figure*, dan 1 ditengah menunjukkan jumlah baris grafik pada satu *figure*, dan angka 2 atau 1 terakhir adalah untuk urutan grafik yang di masukan pada *figure* tersebut.

Matlab tidak hanya digunakan untuk menggabungkan dua grafik namun juga dapat membuat memisahkan grafik ke bentuk *figure* sendiri seperti Gambar 1.21 Memisahkan Grafik Menjadi 2 *Figure*



Gambar 1.21 Memisahkan Grafik Menjadi 2 *Figure*

Percobaan ini menggunakan fungsi *figure* untuk membuat bingkai berbeda untuk grafik yang sebelumnya tersambung. yang fungsinya sendiri dapat dilihat pada Gambar 1.21 Memisahkan Grafik Menjadi 2 *Figure*

1.5. Kesimpulan

- Matlab dapat menampilkan grafik dari suatu gelombang sinyal serta menggabungkan dua buah gelombang sinyal dan membuat gelombang sinyal kontinyu menjadi gelombang sinyal diskrit
- Matlab (*matrix laboratory*) memudahkan akses perangkat lunak matrik yang telah dibentuk oleh LINPACK dan EISPACK
- Matlab adalah sebuah bahasa dengan (*high-performance*) kinerja tinggi untuk komputasi masalah teknik



DAFTAR PUSTAKA

- [1] Korps Asisten Dasar Sistem Kendali. 2009. Modul Praktikum Dasar Sistem Kendali. Inderalaya : Universitas Sriwijaya
- [2] Tri Budi Santoso, Miftahul Huda .“Modul 1 Praktikum Sinyal dan Sistem : Dasar-dasar Operasi Matlab”.
- [3] Firmansyah.“Dasar-Dasar Pemograman Matlab”. <http://Ilmukomputer.com>

