

BAB I

DASAR-DASAR OPERASI MATLAB

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 dipakai 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 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.

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 kompleks 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.

LANGKAH PERCOBAAN

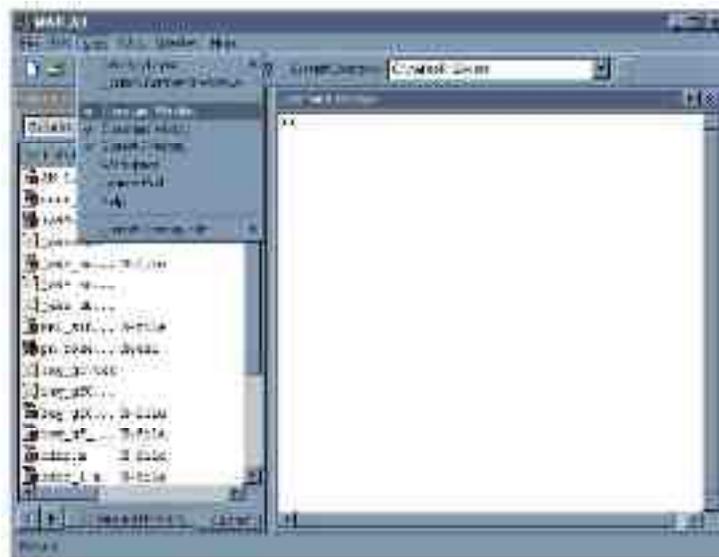
I. Memulai Matlab

Perhatikan Dekstop pada layar monitor PC, anda mulai MATLAB dengan melakukan *double-clicking* pada *shortcut icon* MATLAB



Gambar 1. Icon MATLAB pada desktop PC

Selanjutnya anda akan mendapatkan tampilan seperti pada Gambar berikut ini.

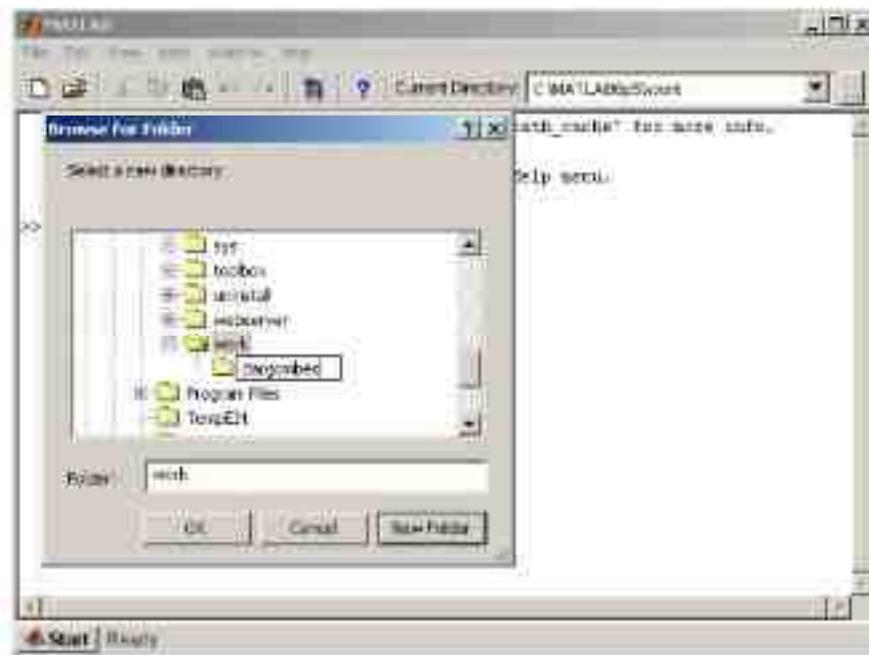


Gambar 2. Tampilan awal Matlab

Sedangkan untuk mengakhiri sebuah sesi MATLAB, anda 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.

2. Menentukan Direktori Tempat Bekerja

Anda dapat bekerja dengan MATLAB secara default pada directory Work ada di dalam Folder MATLAB. Tetapi akan lebih bagus dan rapi jika anda membuat satu directory khusus dengan nama yang sudah anda kususkan, “dargombes” atau nama yang lain yang mudah untuk diingat. Hal ini akan lebih baik bagi anda untuk membiasakan bekerja secara rapi dan tidak mencampur program yang anda buat dengan program orang lain. Untuk itu Arahkan pointer mouse anda pada kotak bertanda ... yang ada disebelah kanan tanda panah kebawah (yang menunjukkan folder yang sedang aktif). Pilih new directory, selanjutnya ketikkan “dargombes”, dan diikuti dengan click Ok.



Gambar 3. Membuat Folder baru tempat program

3. Memulai Perintah Sederhana

Langkah kita yang pertama adalah dengan menentukan variable scalar dengan cara melakukan pengetikan seperti berikut:

```
» x = 2 (selanjutnya tekan "Enter")
```

```
x =
```

```
2
```

```
» y = 3
```

```
y =
```

```
3
```

```
» z = x + y
```

```
z =
```

```
5
```

Tidak terlalu menjadi masalah bagi anda? Saya percaya anda tidak mengalami kesulitan, sebab anda adalah orang yang sangat cerdas.

Nah bagaimana dengan yang satu berikutnya ini? Disini kita mulai dengan 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
```

Selanjutnya ketik:

```
>> y(1)
```

```
ans =
```

```
4
```

dan ulangi untuk y(2) and y(3).

Matlab menggunakan integer positif untuk index array. Elemen pertama adalah **y(1)**, elemen kedua adalah **y(2)**, dan seterusnya. Nol atau bilangan negatif tidak diperbolehkan untuk indek array. Sekarang kita jumlahkan keduanya:

```
» x+y
```

```
ans =
```

```
5 7 9
```

dan sekarang hitung inner product:

```
» x*y'
```

```
ans =
```

```
32
```

Jawabannya adalah $1*4 + 2*5 + 3*6 = 32$! Catat, bahwa y' adalah transpose pada y dan merupakan suatu vector kolom. Untuk memeriksanya, ketikkan perintah berikut:

```
>> y'
```

```
ans =
```

```
4
```

```
5
```

```
6
```

Cara lain pada pengkombinasian dua vector adalah dilakukan melalui perkalian element demi element:

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

```
ans =
```

```
4 10 18
```

Catat periode sebelum perkalian simbol. Sekarang kita dapat mendefinisikan suatu matrix:

```
» A = [1 2 3
```

```
4 5 6
```

```
7 8 9];
```

Catat bahwa matrik tidak diulang kalau kita menggunakan semi colon. Kita sekarang kalikan A dengan transpose dari x:

```
» A*x'
```

```
ans =
```

```
14
```

```
32
```

```
50
```

Sekarang kita harus mentranspose x untuk memenuhi perkalian suatu matrik dan suatu vector kolom. Matrik-matrik ini dapat juga dikalikan satu sama lain diantara mereka:

```
» 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
```

Sekarang coba anda lakukan penjumlahan antara A dan B:

```
» A+B
```

```
??? Error using ==> +
```

Matrix dimensions must agree.

Baiklah, kita tidak dapat menambah suatu matrik 3 kali 3 dengan matrix 3 kali 4 , dan Matlab akan mendeteksi dimensi yang mismatch dan selanjutnya memeberikan pesan error. Sekarang kita cari cara lain untuk mendefinisikan matrik dan vektor. Sebagai contoh suatu matrik nol dengan dimensi 3 baris dan 6 kolom dapat dinyatakan sebagai:

```
ans =
```

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

tentu saja jika anda tambahkan suatu ";" setelah zeros(3,6), jawabannya tidak akan ditampilkan di layar monitor anda. Angka pertama, 3 menunjukkan jumlah baris, sedangkan angka kedua, 6, adalah jumlah kolom. Kita dapat pula melakukan hal yang sama untuk menampilkan angka satu seperti berikut:

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

```
ans =
```

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

Pendefinisian Vektor-vektor Besar

Suatu vektor 1 kali 100 yang menyusun sample pada sinyal cosinus dapat dibangkitkan dengan

```
>> x = cos(0.1*pi*(0:99));
```

Untuk membangkitkan suatu "ramp" dari 1 sampai 50 coba:

```
>> x = [1:1:50];
```

bilangan kedua mengindikasikan step kenaikan dari 1 sampai 50. Untuk membangkitkan suatu fungsi "ramp" dari 1 sampai 50 coba berikut ini:

```
>> x = [1:1:50];
```

Ketika anda tidak memasukkan angka kedua pada perintah diatas, maka secara otomatis (default) step kenaikan ditetapkan bernilai "1":

```
>> x = [1:50];
```

Anda bisa juga secara khusus mendefinisikan suatu rentang nilai pada x sebagai berikut::

```
>> x(51:100) = [50:-1:1]
```

Ini merupakan metode yang sangat bermanfaat untuk mensepsifikasi nilai "waktu" untuk penggambaran. Sebagai contoh, ditetapkan interval sampling dalam contoh diatas adalah 1 detik. Selanjutnya anda dapat mendefinisikan seperti berikut:

```
>> time = [0:0.001:0.099];
```

4. Penggambaran Grafik

Salah satu kelebihan dari Matlab adalah kemudahan dalam mengolah grafik. Sehingga anda tidak perlu kesulitan untuk melihat suatu respon system, misalnya pada kasus melihat bentuk sinyal dalam domain waktu anda cukup mengikuti langkah berikut.

Sekarang ketikkan:

```
>> time = [0:0.001:0.099];
```

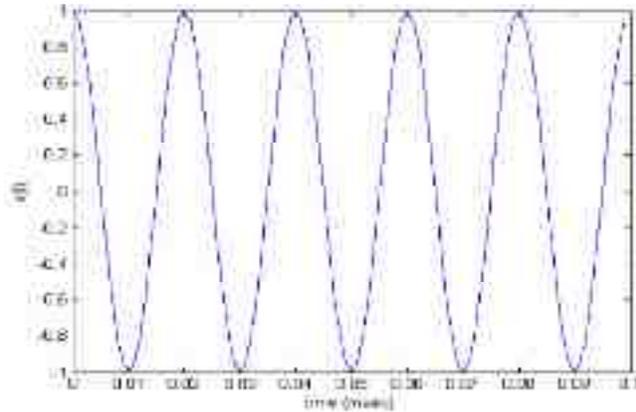
```
>> x = cos(0.1*pi*(0:99));
```

```
>> plot(time,x)
```

```
>> xlabel('time (msec)')
```

```
>> ylabel('x(t)')
```

ini akan menghasilkan gambar seperti berikut:



Gambar 4. Contoh tampilan grafik sederhana dengan perintah plot

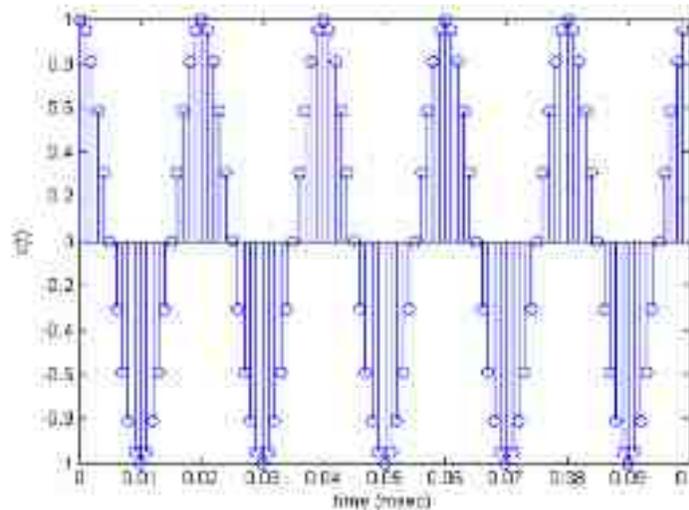
Sedangkan cara untuk menampilkan sederetan nilai fungsi waktu diskrit adalah dengan menggunakan perintah "stem". Dari contoh deretan perintah coba anda rubah beberapa bagian dengan perintah berikut

```
>> stem(time,x)
```

```
>> xlabel('time (msec)')
```

```
>> ylabel('x(t)')
```

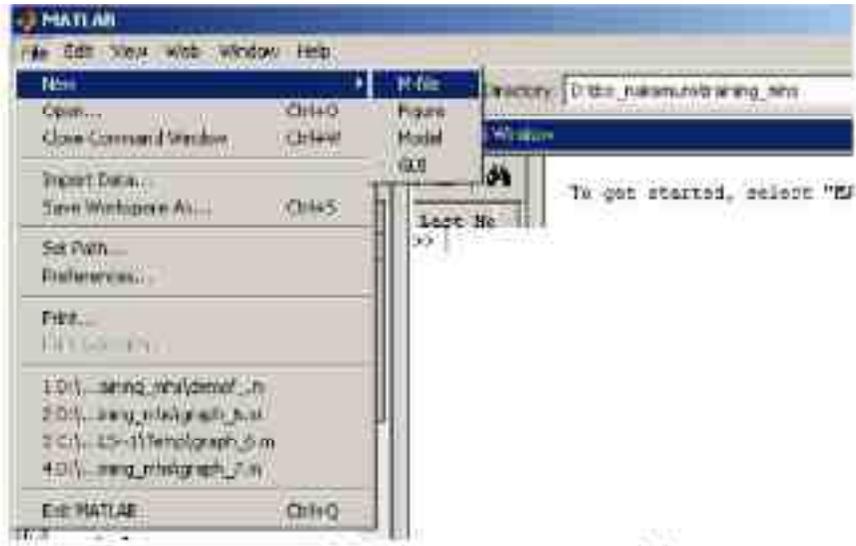
Apakah hasilnya seperti berikut ini?



Gambar 5. Contoh tampilan grafik dengan perintah stem

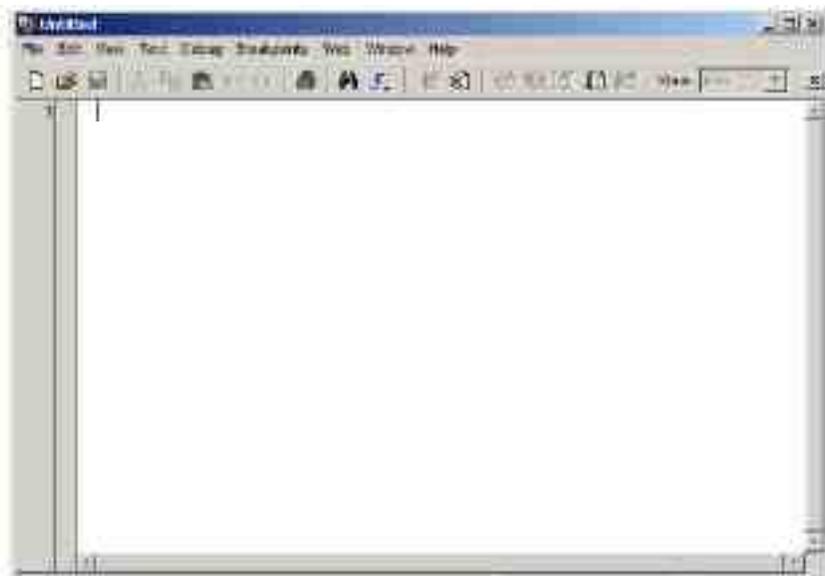
5. Menyusun Program Sederhana

Anda dapat mengedit suatu file text yang tersusun dari beberapa perintah Matlab. Ini dapat dilakukan dengan menekan double-click pada icon "New M-File" icon in the Matlab toolbar.



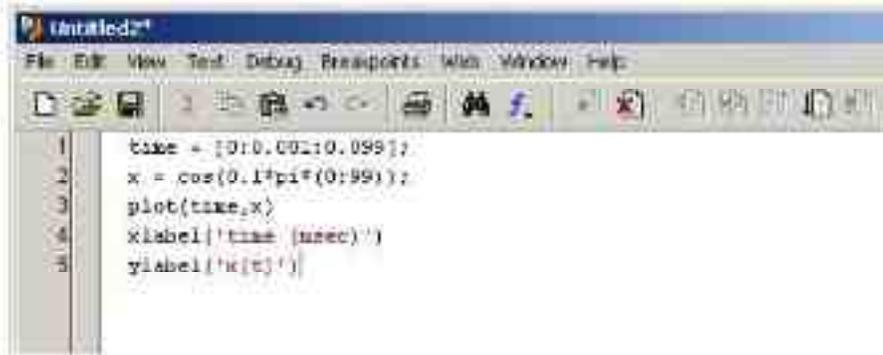
Gambar 6. Langkah awal menyusun program sederhana

Selanjutnya anda akan mendapatkan sebuah tampilan Matlab Editor yang masih kosong seperti ini.

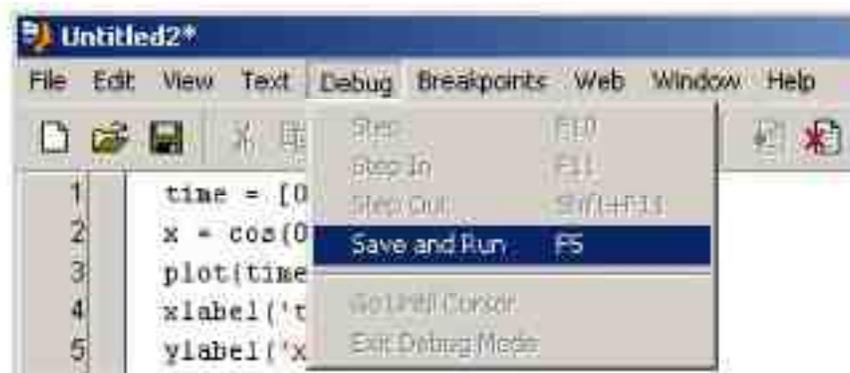


Gambar 7. Tampilan Matlab Editor tempat membuat program.

Selanjutnya anda buat program seperti pada contoh sebelumnya



Gambar 8. Contoh penulisan program pada Matlab Editor



Gambar 9. Cara menyimpan dan mengeksekusi program anda

Lanjutkan dengan menekan toolbar Debug, dan jangan lupa anda pilih Save and Run. Disitu anda harus menuliskan nama program. Anda tuliskan coba_1, secara otomatis akan menjadi file coba_1.m dan akan anda lihat tampilan hasilnya. Seperti apa ya?

Program Pertama

Cobalah untuk membuat program seperti berikut ini pada Matlab editor, dan jangan lupa anda simpan dengan nama coba_1

```
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)
```

Hasil apa yang anda dapatkan ?

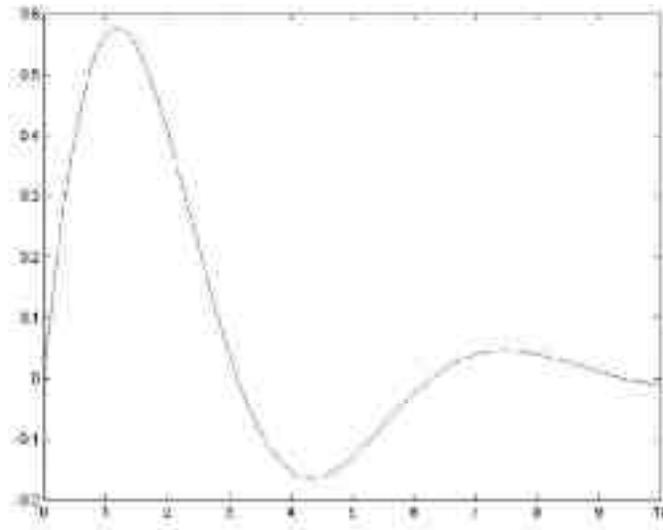
Dalam hal ini anda harus memahami arti setiap perintah yang anda tuliskan dalam Matlab, tidak ada salahnya anda bertanya kepada instruktur apa arti perintah-perintah tersebut.

Program Kedua

Satu contoh lain program untuk for adalah pembangkitan gambar seperti berikut.

```
%File Name:coba_2.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');
```

Bagaimana hasilnya?



Gambar 10. Tampilan program grafik ketiga

6. Fungsi dalam Matlab

Matlab juga mampu untuk menuliskan fungsi yang didefinisikan oleh pemakainya. Buat sebuah fungsi dengan menuliskan program berikut ini:

```
function y = x2(t)
```

```
y = t^2;
```

Anda simpan dengan nama "x2.m" selanjutnya anda dapat memanfaatkan fungsi tersebut melalui Matlab line command dengan cara berikut:

```
>>t=0:1:10;
```

```
>> y_2=x2(t)
```

Hasilnya adalah seperti berikut:

```
y_2 =
```

```
    0  1  4  9 16 25 36 49 64 81 100
```

Anda bisa juga memanggil fungsi tersebut melalui program pada panggil_1.m file yang anda buat seperti berikut:

```
t=0:1:10;
```

```
y_2=x2(t)
```

Hasilnya adalah sama seperti menggunakan command line window.

Program ke tiga

Pembangkitan Sinyal Waktu Kontinyu Sinusoida

1. Disini kita mencoba membangkitkan sinyal sinusoida untuk itu coba anda buat program seperti

berikut:

```
Fs=100;  
t=(1:100)/Fs;  
s1=sin(2*pi*t*5);  
plot(t,s1)
```

2. Lakukan perubahan pada nilai s1:

```
s1=sin(2*pi*t*10);
```

Dan perhatikan apa yang terjadi, kemudian ulangi untuk mengganti angka 10 dengan 15, dan 20. Perhatikan apa yang terjadi.

3. Coba anda edit kembali program anda sehingga bentuknya persis seperti pada langkah 1, kemudian lanjutkan dengan melakukan perubahan pada nilai amplitudo, sehingga bentuk perintah pada s1 menjadi:

```
s1=2*sin(2*pi*t*5);
```

4. Kembalikan program anda sehingga menjadi seperti pada langkah pertama. Sekarang coba anda lakukan sedikit perubahan sehingga perintah pada s1 menjadi:

```
s1=2*sin(2*pi*t*5 + pi/2);
```

Coba anda perhatikan, apa yang terjadi? Apa yang baru saja anda lakukan adalah merubah nilai fase awal sebuah sinyal dalam hal ini nilai $\theta = \pi/2 = 90^\circ$. Sekarang lanjutkan langkah anda dengan merubah nilai fase awal menjadi 45° , 120° , 180° , dan 225° .

Program Ke Empat

Pembangkitan Sinyal Persegi

Disini akan kita bangkitkan sebuah sinyal persegi dengan karakteristik frekuensi dan amplitudo yang sama dengan sinyal sinus. Untuk melakukannya ikuti langkah berikut ini.

1. Buat sebuah file baru dan beri nama coba_kotak.m kemudian buat program seperti berikut ini.

```
Fs=100;
```

```

t=(1:100)/Fs;
s1=SQUARE(2*pi*5*t);
plot(t,s1,'linewidth',2)
axis([0 1 -1.2 1.2])

```

2. Coba anda lakukan satu perubahan dalam hal ini nilai frekuensinya anda rubah menjadi 10 Hz, 15 Hz, dan 20 Hz. Apa yang anda dapatkan?

3. Kembalikan bentuk program menjadi seperti pada langkah pertama, Sekarang coba anda rubah nilai fase awal menjadi menjadi 45°, 120°, 180°, dan 225°.

Program Ke Lima

Pembangkitan Sinyal Waktu Diskrit, Sekuen Konstan

Disini akan kita lakukan pembangkitan sinyal waktu diskrit. Sebagai langkah awal kita mulai dengan membangkitkan sebuah sekuen unit step. Sesuai dengan namanya, unit step berarti nilainya adalah satu satuan. Untuk itu anda ikuti langkah berikut ini.

1. Buat program baru dan anda ketikkan perintah seperti berikut:

```

%File Name: SS1_3.m
%Oleh: tri Budi 212
%Pembangkitan Unit Step Sekuen
L=input('Panjang Gelombang (>=40)=')
P=input('Panjang Sekuen =')
for n=1:L
if (n>=P)
step(n)=1;
else
step(n)=0;
end
end
x=1:L;
stem(x,step)

```

2. Anda ulangi langkah pertama dengan cara me-*run* program anda dan masukan nilai untuk panjang gelombang dan panjang sekuen yang berbeda-beda.