

MODUL 1 OPERASI DASAR MATLAB

I. Tujuan Instruksional Khusus

- Mahasiswa mampu mengoperasikan Matlab dan memanfaatkannya sebagai perangkat Simulasi untuk praktikum Sinyal dan Sistem

II. Pengenalan Perangkat Lunak 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 digunakan 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 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.

2.1. 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 & Command Window*, *Command History*, sebuah *Editor & 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*. Hal 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 komplek.
4. **Graphics.** Matlab memiliki fasilitas untuk menampilkan vektor dan matrik sebagai suatu grafik. Di dalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dikensi 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 benutk 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 Fortranmampu berinteraksi dengan Matlab. Hal ini melibatkan fasilitas untuk pemanggilan routines dari Matlab (*dynamic linking*), pemanggilan Matlab sebagai sebuah *computational engine*, dan untuk membaca dan menuliskan *MAT-files*.

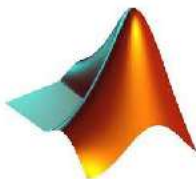
III. Perangkat Yang Diperlukan

- PC/Laptop yang dilengkapi dengan perangkat multimedia (sound card, microphone, speaker aktif, atau headset)
- Sistem Operasi Windows dan Perangkat Lunak Matlab yang dilengkapi dengan tool box DSP

IV. Langkah Percobaan

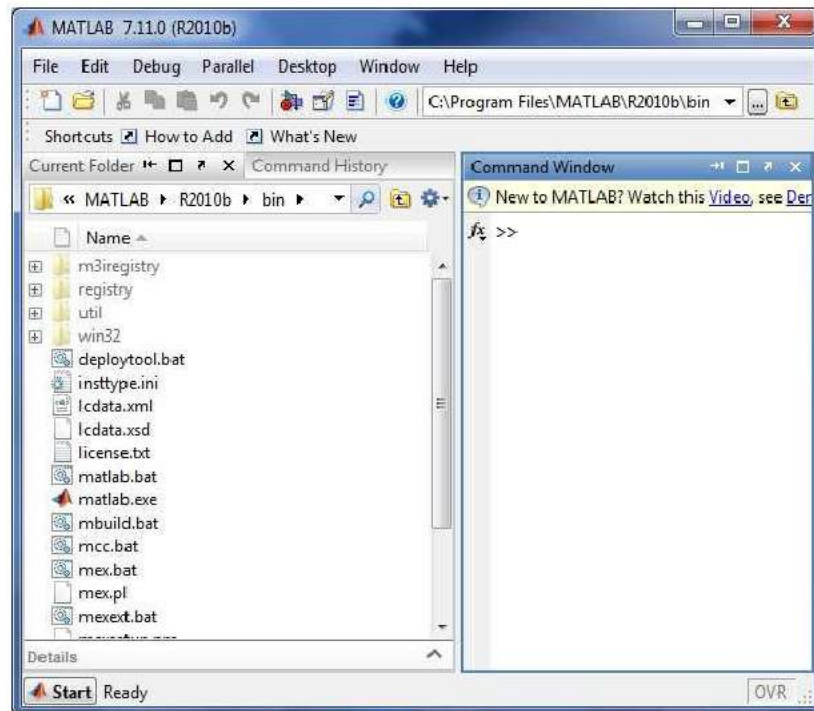
4.1. Memulai Matlab

Anda dapat memulai menjalankan Matlab dengan cara double-clicking pada shortcut icon Matlab.



Gambar 1.1. Icon Matlab pada dekstop PC/Laptop

Selanjutnya anda akan mendapatkan tampilan seperti pada Gambar berikut ini.



Gambar 1.2. Tampilan pertama Matlab

Secara umum tampilan awal Matlab akan menyajikan:

- *Command Window* yang merupakan tempat atau mengetikkan perintah yang dapat dieksekusi secara langsung.
- *Command History* yang berisi berbagai perintah yang telah dieksekusi oleh Command Window. Ini merupakan fitur untuk melakukan tracking ketika proses developing atau debugging programs atau untuk mengkonfirmasi bahwa perintah-perintah telah dieksekusi sepanjang suatu penghitungan multi-step dari command line.
- *Current Folder*, yang menyajikan informasi folder tempat bekerja saat ini dan isi yang ada di folder tersebut. Window ini bermanfaat untuk menemukan lokasi file-file dan script-script sehingga dapat diedit, dipindahkan, diganti nama, dihapus, dsb.

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 '*exit*' dalam Command Window.

4.2. Memulai Perintah Sederhana

Penjumlahan dan Perkalian

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

```
>> x = 2 <Enter>
```

```
x =
```

```
2
```

```
>> y = 3
```

```
y =
```

```
3
```

```
>> z = x + y
```

```
z =
```

```
5
```

Untuk operasi perkalian anda bisa melakukan seperti berikut

```
>> z = x * y
```

```
z =
```

```
6
```

Saya percaya anda tidak mengalami kesulitan.

Operasi Vektor dan Matrik

Sebuah vektor bisa saja didefinisikan sebagai matrik yang memiliki ukuran $1 \times N$, dengan kata lain sebuah vektor adalah matrik yang hanya memiliki baris sebanyak 1, dan kolom N. Misalnya vektor x merupakan matrik yang berukuran 1×3 dengan nilai-nilai 1, 2 dan 3.

Maka kita bisa mendefinisikan vector x sbb.

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

```
x =
```

```
1    2    3
```

Sedangkan untuk vektor y yang memiliki jumlah elemen sama, tetapi dengan nilai berbeda bisa dituliskan sebagai

```
>> y = [4 5 6]
```

```
y =
```

```
4    5    6
```

Jika anda ingin mengetahui elemen ke 1 dari vektor y, and anda bisa menuliskannya sebagai berikut.

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

```
ans =
```

```
4
```

Sekarang kita jumlahkan keduanya:

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

Coba anda rubah vektor y dengan perintah

```
>> y'  
ans =  
4  
5  
6
```

Artinya anda melakukan transpose pada vektor y, kalau belum paham coba anda buka buku matematika anda tentang operasi matrik. Pasti anda menemukan jawabannya....

Sekarang hitung inner product

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

Hasil ini diperoleh dari perhitungan seperti ini $1*4 + 2*5 + 3*6 = 32$. Dimana y' adalah transpose pada y dan merupakan suatu vector kolom.

Jika anda ingin melakukan operasi perkalian sebagai dua vektor baris, coba lakukan dengan perintah `perkalianelement-demi-element`:

```
>> x.*y  
ans =  
4    10    18
```

Suatu saat anda harus merubah vektor menjadi sebuah matrik dengan ukuran tiga kali satu (3x1). Untuk itu anda harus memodifikasi penulisan vektor menjadi matrik dengan memasukkan tanda semicolon (;) diantara angka-angka tersebut.

```
>> x=[1;2;3]  
x =  
1  
2  
3
```

Matlab memberikan kemudahan anda untuk melakukan cara cepat dalam menyusun vektor/matrik tertentu, misalnya

```
>> x = ones(1, 10)
x =
    1    1    1    1    1    1    1    1    1    1
```

Atau

```
>> x = zeros(3, 1)
x =
     0
     0
     0
```

Bilangan Acak

Anda bisa melakukan pembangkitan bilangan acak dengan mudah, misalnya anda akan membangkitkan sebuah vektor yang tersusun dari 10 bilangan acak terdistribusi uniform.

```
>> rand(1,10)
ans =
    0.8147    0.9058    0.1270    0.9134    0.6324    0.0975    0.2785    0.5469    0.9575    0.9649
```

Atau anda ingin membangkitkan bilangan acak terdistribusi Gaussian (normal)

```
>> randn(1,10)
ans =
   -1.3499    3.0349    0.7254   -0.0631    0.7147   -0.2050   -0.1241    1.4897    1.4090
    1.4172
```

Untuk bilangan binary anda bisa melakukannya seperti berikut

```
>> randint(1,10)
ans =
     1     0     1     1     1     1     1     0     1     0
```

Coba anda lakukan pembangkitan sebuah vektor dengan cara seperti berikut

```
>> randint(1,10,4)
```

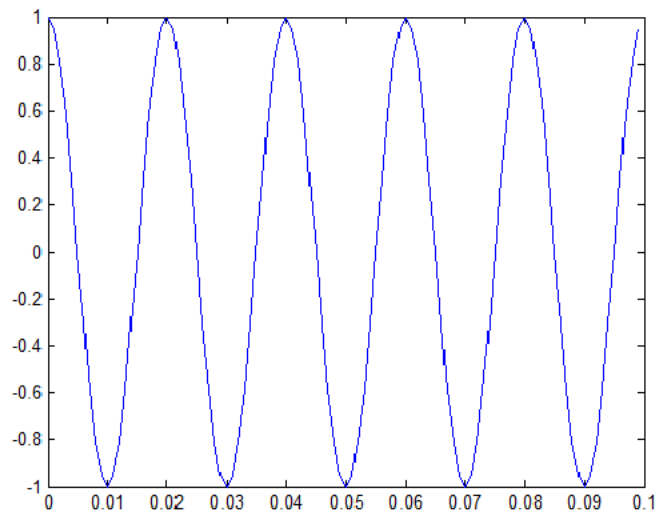
Jelaskan apa yang anda lihat pada layar monitor. Kemudian anda coba bangkitkan sebuah matrik berukuran 10x 10 yang tersusun dari bilangan acak berdistribusi Gaussian.

Membuat 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.

```
>> time = [0:0.001:0.099];  
>> x = cos(0.1*pi*(0:99));  
>> plot(time,x)
```

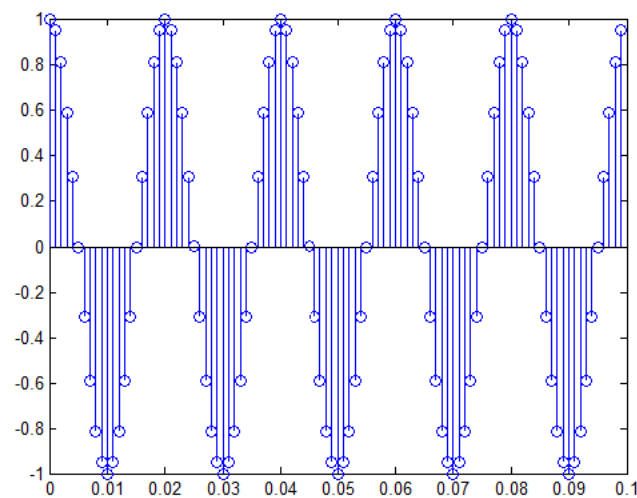
Dan akan tampil sebuah grafik sinuoida seperti berikut. Hal ini merepresentasikan sebuah sinyal dalam domain waktu konitnyu.



Gambar 1.3. Grafik sinyal waktu kontinyu

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

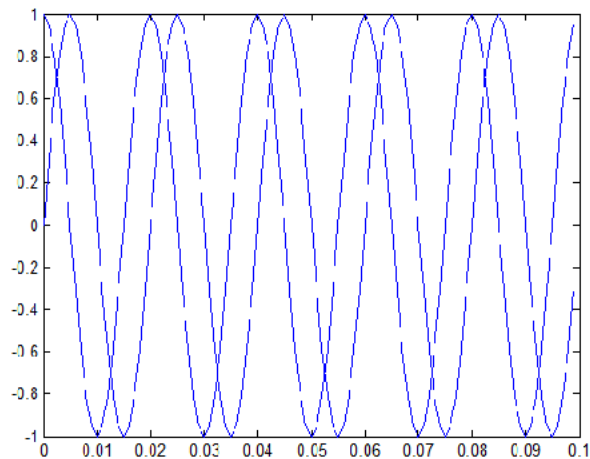
```
>> stem(time,x)
```



Gambar 1.4. Grafik sinyal waktu diskrit

Anda bisa melakukan penggabungan lebih dari satu grafik pada sebuah tampilan. Dengan perintah *hold on* dan *hold off*.

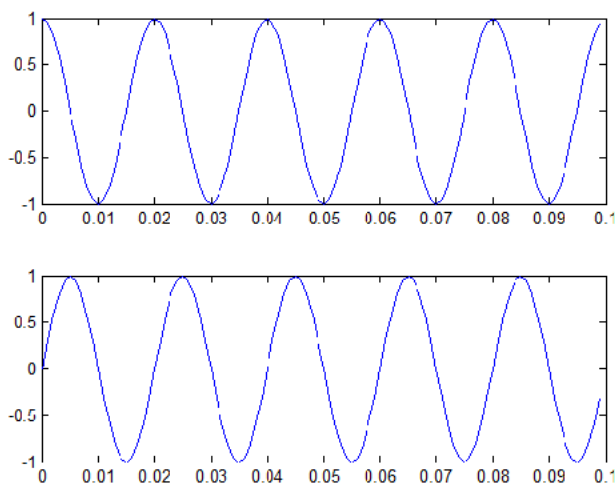
```
>> time = [0:0.001:0.099];  
>> x = cos(0.1*pi*(0:99));  
>> y = sin(0.1*pi*(0:99));  
>> plot(time,x)  
>> hold on  
>> plot(time,y)  
>> hold off
```



Gambar 1.5. Grafik dua sinyal bersamaan

Jika anda ingin menampilkan dua buah gambar pada frame berbeda-beda, anda bisa memanfaatkan perintah *subplot*. Coba modifikasi beberapa perintah seperti berikut.

```
>> subplot(211); plot(time,x)  
>> subplot(212); plot(time,y)
```



Gambar 1.6. Tampilan dua grafik pada dua frame

Coba anda melakukan modifikasi perintah diatas dengan cara seperti berikut.

```
>> figure(1);  
>> plot(time,x)  
>> figure(2);  
>> plot(time,y)
```

Silahkan anda berkreasi dengan berbagai model tampilan grafik, jika anda kurang puas anda bisa mengetikkan perintah help pada Matlab Command Window. Anda gali sebanyak-banyaknya tentang menampilkan grafik dengan Matlab. Selamat mencoba...

Membuka File

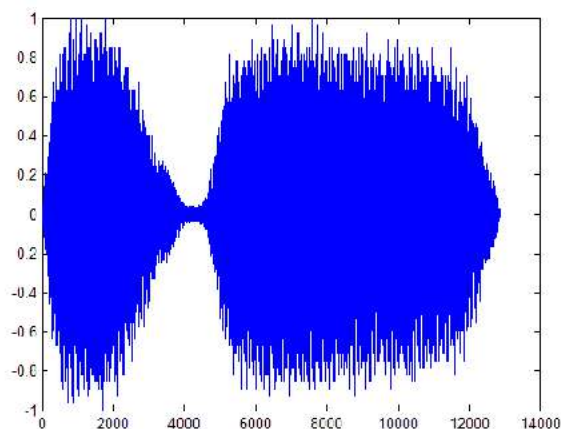
Matlab memiliki kemudahan di dalam membuka file-file tertentu yang sudah didukung oleh library-nya. Dalam hal ini anda bisa membuka file text, suara atau gambar.

```
>> clear all  
>> load train  
>> whos
```

Name	Size	Bytes	Class	Attributes
Fs	1x1	8	double	
y	12880x1	103040	double	

Dalam hal ini ditunjukkan bahwa hasil pemanggilan file 'train' dipanggil (load) secara default dengan frakuensi sampling sebesar Fs (44100 Hz), disimpan sementara pada matrik y (default) dengan ukuran 12880 x1, jumlah Byte sebesar 103040 dan merupakan tipe data double, Atribute tidak ada penjelasan. Untuk mengetahui bagaimana bunyi file tersebut, anda bisa memberikan perintah berikut.

```
>> sound(y,Fs)  
>> plot(y)
```



Gambar 1.7. Grafik dari file train

Coba anda lakukan untuk file 'handel', 'gong', dan 'chirp'.

Salah satu fungsi lain untuk membaca file suara adalah `'wavread'`, sedangkan untuk memainkan anda bisa memanfaatkan `'wavplay'`, dan untuk menyimpan anda bisa memanfaatkan `'wavwrite'`. Tentu saja anda harus banyak memanfaatkan fasilitas help untuk lebih mengenali fungsi-fungsi ini.

Anda bisa memanfaatkan perintah load untuk memanggil sebuah file gambar yang ada di dalam direktori standar Matlab dengan perintah load. Untuk membuka file gambar anda juga bisa memanfaatkan fungsi `'imread'`, sedangkan untuk menyimpannya anda bisa memanfaatkan fungsi `'imwrite'`. Format gambar yang bisa diolah dengan matlab cukup banyak, seperti tiff, jpeg, bmp dan png.

Anda masih penasaran? Coba rubah direktori tempat sesuai dengan direktori dimana anda menyimpan file gambar yang anda punya. Dalam contoh berikut ini kita pilih file actress tanah air yang cukup populer, 'Agnes Monica'. Kalau anda belum punya filenya, anda bisa download dari berbagai sumber yang ada di internet. Kemudian coba anda ketikkan perintah berikut ini. Jika anda kesulitan mencari file gambar Agness Monica, ambil foto teman sebelah anda dan beri nama `'agnes.jpg'`.

```
>> y=imread('agnes.jpg');  
>> imshow(y)
```

Anda bisa mengkonversi dari format RGB menjadi format Gray seperti berikut

```
>> yg=rgb2gray(y);  
>> imshow(yg)
```

Untuk mengetahui karakter file sebelum dan sesudah proses konversi anda bisa melakukan perintah berikut.

```
>> whos
```

Seperti sebelumnya, anda bisa melihat perbandingan jumlah array penyusunnya, dll. Dalam hal ini format rgb tersusun dari komponen pixcell x,y, r (red), g (green) dan b (blue). Sedangkan format gray tersusun dari komponen x,y, dan gray level. Selengkapnya bisa dilihat seperti berikut.

Name	Size	Bytes	Class	Attributes
X	200x320	512000	double	
caption	2x1	4	char	
map	81x3	1944	double	
y	214x235x3	150870	uint8	
yg	214x235	50290	uint8	



(a) Format RGB

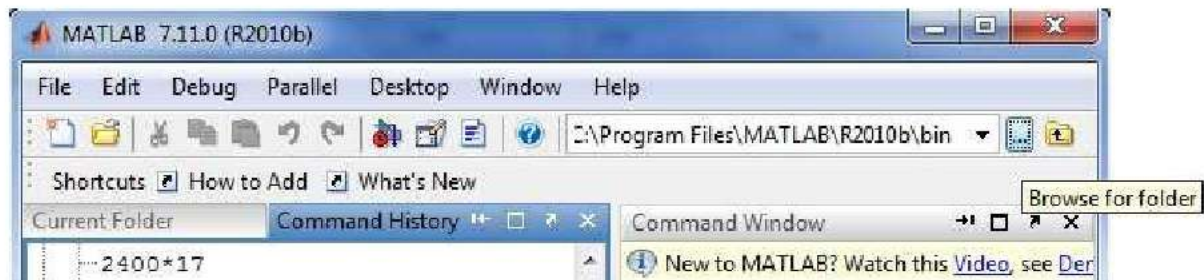


(b) Format Gray

Gambar 1.8. Tampilan file image

4.3. 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 persiapkan, “Matlab_SinyalSistem” 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).



Gambar 1.9. Proses awal membuat folder tempat bekerja

Selanjutnya akan muncul sebuah window **Browse For Folder**, anda click pada tombol **Make New Folder**, dan ketikkan nama folder untuk membuat direktori tempat kerja anda. Dalam hal ini anda bisa memberi nama ‘*Matlab_SinyalSistem*’, dan dilanjutkan dengan menekan tombol **Ok**.

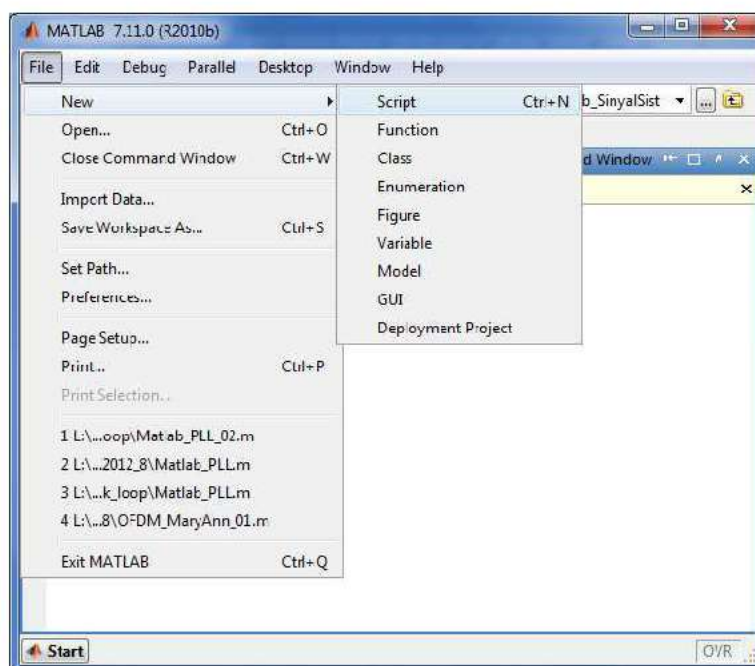
Setelah langkah ini pada tampilan Matlab Current Folder akan mengalami perubahan, bisa saja tampil kosong tanpa ada satu file apapun. Hal ini terjadi karena anda belum mebuat sebuah programatau mengkopikan file ke folder tersebut.



Gambar 1.10. Membuat nama folder baru tempat bekerja

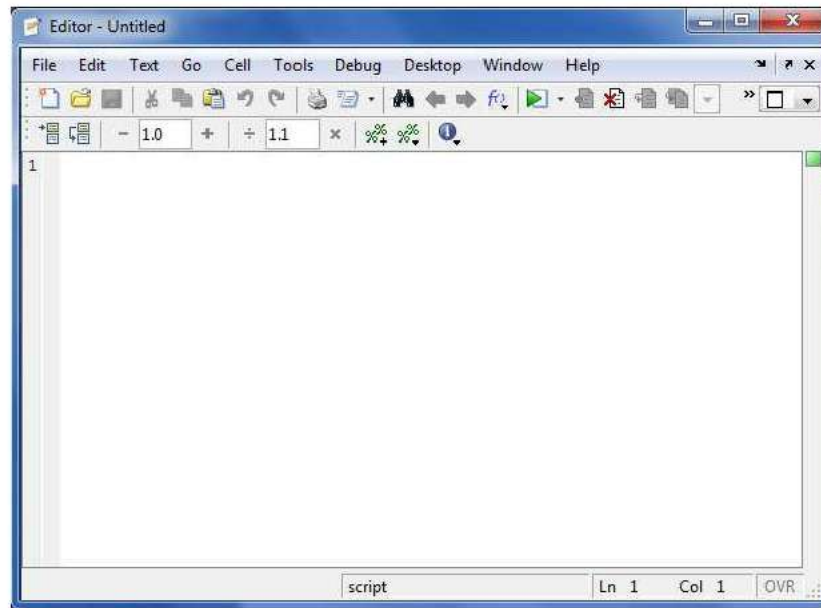
4.4. Membuat Program Baru

Anda dapat menyusun sebuah program di Matlab dengan memanfaatkan Matlab Editor, sehingga setelah anda selesai dengan sebuah proses perhitungan anda bisa menyimpan program tersebut, memanggil di waktu yang lain atau melakukan perubahan sesuai keinginan anda. Ini dapat dilakukan jika anda membuat program Matlab pada Matlab editor, caranya adalah dengan menekan Click pada **File**→**New**→ **Script** **Ctrl+N**. Pada versi Matlab berbeda, bisa saja caranya sedikit berbeda.



Gambar 1.11. Membuat program baru dengan Matlab Editor

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



Gambar 1.12. Tampilan awal Matlab Editor

Untuk membuat program anda bisa mengetikkan script berikut ini.

```
%File Name: buka_gambar.m  
clear all  
y=imread('agnes.jpg');  
figure(1);  
imshow(y)  
yg=rgb2gray(y);  
figure(2);  
imshow(yg)
```

Anda simpan dengan cara click tanda panah hijau ke arah kanan, dan beri nama '*buka_gambar.m*'. Selanjutnya secara otomatis Matlab akan melakukan eksekusi program anda. Anda bisa juga hanya menyimpan dengan cara Click gambar floppy disk pada toolbar Matlab Editor, atau bisa juga dengan Click pada File→Save Ctrl+S→ tuliskan nama file, misalnya '*buka_gambar.m*'. Tentu saja penyimpanan anda lakukan pada folder yang sudah ditetapkan dimana file image '*agnes.jpg*' berada.

4.5. Membuat Fungsi Matlab

Anda bisa membangun fungsi sendiri dengan Matlab Editor. Setelah anda membuka Matlab Editor, anda ketikkan script berikut ini.

```
function y = x2(t)  
y = t.^2;
```

Anda simpan dengan nama 'x2.m', kalau anda lupa tidak perlu khawatir, sebab Matlab akan secara otomatis menyimpan sesuai dengan nama variable di belakang perintah *function*, yaitu x2. Untuk memanfaatkan fungsi tersebut, anda bisa memanggilnya melalui perintah di Matlab Command Window,

```
>>t=0:1:10;  
>> y_2=x2(t)  
y_2 =  
    0    1    4    9   16   25   36   49   64   81  100
```

Atau bisa juga melalui sebuah program yang anda buat pada Matlab Editor.

V. TUGAS

1. Dari contoh-contoh program yang sudah anda jalankan, coba berikan penjelasan arti setiap perintah terhadap output yang dihasilkannya.
2. Coba anda cari bagaimana cara menampilkan grafik untuk tampilan tiga dimensi dan 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 tamplan pada fungsi pertama tidak boleh hilang.
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.
5. Bagaimana cara menampilkan dua fungsi diman 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).