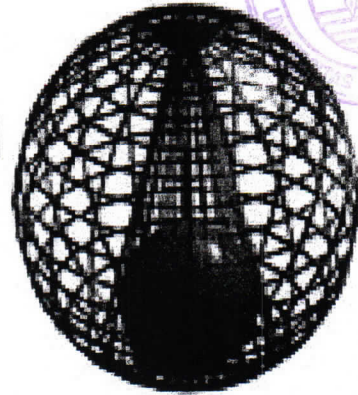


MODUL
PEMROGRAMAN
MATLAB



OLEH :

ELYAS KUSTIAWAN , S.Si., M.Si.

JURUSAN TEKNIK MESIN
FAKULTAS TEKNIK
UNIVERSITAS BANGKA BELITUNG
2014

HALAMAN PENGESAHAN

1. Judul : Modul Pemrograman MATLAB
2. Peneliti
 - a. Nama Lengkap : ELYAS KUSTIAWAN, S.Si., M.Si.
 - b. Jenis Kelamin : Laki-laki
 - c. NIDN : 0028127605
 - d. Disiplin Ilmu : Teknik Mesin
 - e. Pangkat/ Golongan : IIIC
 - f. Jabatan Fungsional : Asisten Ahli
 - g. Fakultas/Jurusan : Teknik/Teknik Mesin
 - h. Alamat : Jl. Raya Balunijuk Desa Balunijuk Kec. Merawang
 - i. No. HP /Email : 085222852988
elyaskustiawan@gmail.com
3. Biaya : Swadaya

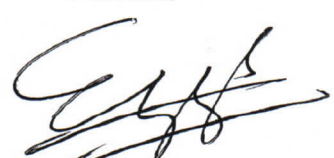
Balunijuk, Juli 2014

Mengetahui:
Pjs. Ketua Jurusan Teknik Mesin




Eka Sari Wijianti, S.Pd., M.T
NP : 308105010

Peneliti,


Elyas Kustiawan, S.Si., M.Si.
NIDN : 0028127605

Menyetujui,
Pjs. Dekan Fakultas Teknik




Fadillah Sabri, S.T., M.Eng
NP : 307103013

KATA PENGANTAR

Alhamdulillah Robbil 'alamin, hamday yuwaafii ni'amahu, wa yukaafi maziidah. Puji syukur penulis panjatkan kepada Allah SWT, *Dzat* yang memberikan rahmat dan kemakmuran kepada semua makhluk ciptaan-Nya, Perancang segala sesuatu, Pencipta seluruh benda, Pembentuk semua bentuk dan Pelukis semua wujud.

Penelitian berupa pembuatan modul pemrograman komputer yang berjudul Modul Pemrograman MATLAB dapat diselesaikan sesuai dengan waktu yang telah direncanakan.

Penulis sampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang membantu penelitian ini.

Penulis berharap, hasil penelitian berupa pembuatan modul pemrograman komputer yang berjudul Modul Pemrograman MATLAB ini dapat bermanfaat bagi seluruh civitas akademi Universitas Bangka Belitung.

Balunijuk, Juli 2014

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iii

BAB 1 PENDAHULUAN

1.1 Sekilas MATLAB.....	1
1.2 Pengenalan MATLAB	2
1.3 Window-Window dalam MATLAB.....	2
1.3.1 MATLAB Command Window/Editor.....	2
1.3.2 MATLAB Editor/Debugger.....	3
1.3.3 MATLAB Help Window dan MATLAB Demo Window	3
1.3.4 MATLAB Window Grafik.....	3
1.4 Fasilitas Help	4
1.4.1 Perintah Help.....	4
1.4.2 Perintah Helpwin.....	4
1.4.3 Command : Lookfor.....	4
1.4.4 Command : Demo	5
1.5 Contoh Sederhana Window Grafik.....	5
1.6 Tipe Data.....	6
1.6.1 Tipe Data String.....	6
1.6.1.1 Fungsi : disp.....	6
1.6.1.2 Fungsi : input.....	6
1.6.1.3 Fungsi : fprintf	7
1.6.2 Tipe Data skalar.....	7
1.6.3 Tipe Data Array.....	7

1.6.3.1 Mengassign Nilai array	8
1.6.3.2 Menambah element dari array.....	8
1.6.3.3 Mengakses sebagian element array	9
1.6.3.4 Menghapus element array.....	9
1.7 Matriks.....	9
1.7.1 Menyusun sebuah Matriks	9
1.7.2 Operator dalam Matriks.....	11
1.7.3 Fungsi-Fungsi Matriks.....	11

BAB 2 DASAR-DASAR PEMROGRAMAN FUNCTION DAN M-FILE

2.1 Function dan M-File	13
2.1.1 Membentuk M-File.....	13
2.1.2 Menjalankan suatu M-File	14
2.1.3 Menambahkan help pada user defined M-File	14
2.2 Daftar Command untuk pemrograman	15
2.3 Operator dan Karakter khusus	17
2.4 Latihan dan contoh program	18

BAB 3 CONTROL FLOW

3.1 For Loop.....	19
3.2 While Loop.....	19
3.3 Command Switch dan Case	21
3.4 Contoh dan Latihan	22

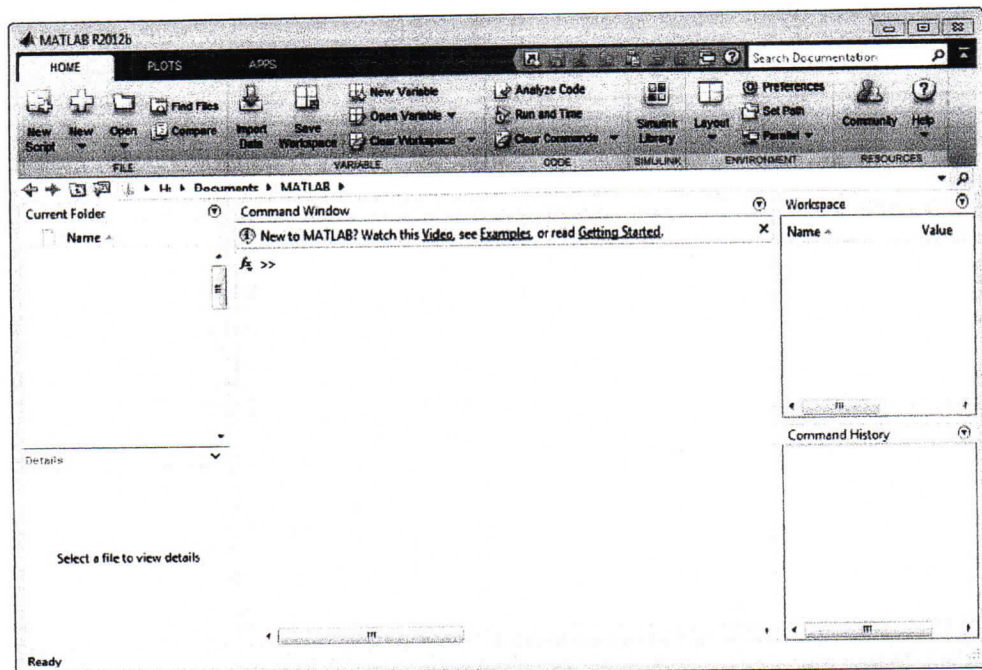
DAFTAR PUSTAKA

BAB 1 PENDAHULUAN

1.1 Sekilas MATLAB

MATLAB[®] merupakan bahasa pemrograman tingkat tinggi yang dikembangkan oleh MathWorks dan dikhususkan untuk komputasi numerik, visualisasi, dan pemrograman. Dengan memanfaatkan MATLAB, pengguna dapat melakukan analisis data, mengembangkan algoritma, dan membuat model maupun aplikasi. Bahasa, *tools*, dan fungsi-fungsi *built-in* akan memudahkan pengguna untuk mengeksplorasi berbagai pendekatan dan memperoleh solusi dengan lebih cepat dibandingkan apabila menggunakan *spreadsheets* atau bahasa pemrograman tradisional, seperti C/C++ atau Java[™]. MATLAB menggunakan konsep array/matrik sebagai standar variabel elemennya tanpa memerlukan pendeklarasian array seperti pada bahasa lainnya. Selain itu juga dapat diintegrasikan dengan aplikasi dan bahasa pemrograman eksternal seperti C, Java, .NET, dan Microsoft[®] Excel[®].

Kita dapat memulai MATLAB dengan men-double klik icon MATLAB, atau dapat mencari di Star menu Windows. Window MATLAB, disebut MATLAB Desktop, akan tampil seperti berikut :




Sebagai permulaan, kita coba ketik perintah berikut :

```
» date
ans =
10-Jun-2014
```

Perintah diatas menunjukkan tanggal pada saat kita ketik yaitu 10 Juni 2014. Selamat! Anda telah sukses mengeksekusi perintah pertama dalam MATLAB!

Command sederhana lainnya dapat kita coba yaitu command `clc` (*clear command window*). Pada command window, jika kita ingin menghapus semua teks dan kursor pindah ke atas window, maka ketiklah `clc`. Setelah terhapus, selanjutnya kita dapat mengerjakan perintah yang lainnya.

Untuk mengakhiri MATLAB, klik di File menu MATLAB command window dan pilih "Exit MATLAB". (Sebagai alternatif, kita bisa close MATLAB dengan meng-klik tanda X pada  dipojok kanan atas MATLAB.

1.2 Pengenalan MATLAB

MATLAB (*Matrix Laboratory*) adalah suatu bahasa pemrograman yang dibangun dengan dasar berpikir menggunakan sifat dan bentuk matriks. MATLAB dikembangkan oleh Mathwork.Inc. (lihat <http://www.mathwork.com>).

MATLAB merupakan *software* yang paling efisien untuk perhitungan numerik yang berbasis matriks. Dengan demikian, jika di dalam perhitungan kita mampu memformulasikan masalah kedalam format matriks, maka MATLAB merupakan *software* terbaik untuk penyelesaian numeriknya.

1.3 Window-Window dalam MATLAB

Ada beberapa macam window yang ada dalam MATLAB, yang dapat dijelaskan sebagai berikut :

1.3.1 MATLAB Command Window/Editor

Window ini merupakan window yang pertama kali dibuka setiap kita menjalankan MATLAB. Di window ini, kita dapat melakukan akses ke command-command MATLAB, mengetikkan baris-baris perintah MATLAB, mengakses window *help* dan lain-lain. Jika kita mau mengetikkan kembali perintah MATLAB yang sudah pernah ditulis pada baris di atas prompt sekarang, maka kita bisa menggunakan tanda panah ke atas.

1.3.2 MATLAB Editor/Debugger

Window ini diperuntukkan bagi pembuatan m-file. M-file adalah file yang berisi kumpulan perintah MATLAB untuk suatu keperluan tertentu. File ini mempunyai ekstention *.m yang dapat dipanggil di command window.

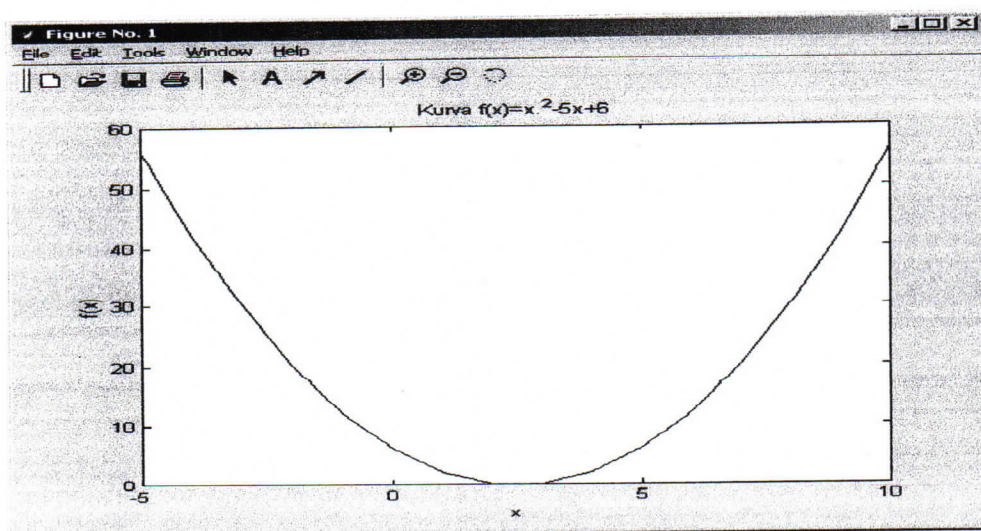
Window editor ini dapat diakses dengan menggunakan menu *pull-down*, dengan memilih menu File kemudian pilih New→M-File, maka MATLAB akan membuka window Editor/Debugger tersebut.

1.3.3 MATLAB Help Window dan MATLAB Demo Window

Window ini berisikan petunjuk dari setiap perintah MATLAB. Perintah-perintah tersebut bisa merupakan perintah MATLAB yang standar ataupun toolbox yang sudah diinstallkan dalam MATLAB di komputer anda. Toolbox adalah suatu kumpulan m-file yang sudah dibuat untuk keperluan tertentu, seperti halnya sekelompok m-file yang dibuat untuk suatu analisis statistika tertentu, seperti wavelet dan lain-lain.

1.3.4 MATLAB Window Grafik

Window ini secara otomatis dibuka untuk menampilkan suatu grafik yang dibuat dengan MATLAB, dimana hanya ada satu window grafik saja yang dapat dibuka pada suatu waktu. Sebagai ilustrasi, misalkan plot dari kurva fungsi kuadrat $f(x) = x^2 - 5x + 6$ dengan batas $-5 \leq x \leq 10$. Tampilan grafik dalam MATLAB adalah sebagai berikut :



1.4 Fasilitas Help

Salah satu yang penting dalam mempelajari *software* adalah pemahaman kita terhadap fasilitas *help* yang tersedia pada *software* tersebut. Tujuan dari fasilitas *help* adalah untuk membantu kita dalam mempelajari sintaks suatu bahasa pemrograman, disamping juga sebagai informasi mengenai kemampuan dari *software* tersebut. Sudah barang tentu sangat tidak efisien bagi kita untuk menghafal semua perintah-perintah yang ada dalam *software* tersebut, oleh karenanya kita cukup melihat dari fasilitas *help* yang ada.

1.4.1 Perintah help

Perintah *help* bertujuan untuk menampilkan informasi *help* dari suatu fungsi atau perintah *MATLAB*. Misalnya kita ingin mengetahui informasi dari perintah *disp*, tapi kita belum tahu sintaks penggunaan perintah ini. Untuk itu kita dapat mengakses fasilitas *help* dengan mengetikkan di window *command/editor* setelah prompt `>>`.

```
>>help log
```

Setelah menekan *enter*, maka di layar akan muncul keterangan sebagai berikut :


```
LOG(X) is the natural logarithm of the elements of X.  
Complex results are produced if is not positive.
```

LOG(X) adalah logaritma natural dari *X*. Hasil bilangan kompleks diperoleh jika *X* adalah bilangan negatif.

1.4.2 Perintah Helpwin

Cara lain untuk mengakses *help* adalah dengan membrowsing atau mencari sendiri pada window *help*. Ada tiga cara untuk menampilkan window *help* ini :

1. Mengakses *pull-down* menu Help, kemudian pilih Window.
2. Mengetikkan perintah *helpwin* di window *command/editor*.

```
>>helpwin
```
3. Meng-klik shortcut  di menu bar window *command/editor*.

1.4.3 Command : lookfor

Command *lookfor* bertujuan untuk mencari *command-command* yang berhubungan dengan suatu topik tertentu atau memuat suatu kata tertentu. Sebagai contoh, kita ingin melihat *command-command* yang berhubungan dengan menampilkan suatu output di layar, maka coba untuk mengetikkan :

```
>>lookfor display
```

dan kemudian tekan *enter*, maka *MATLAB* akan mencari *command-command* yang memuat kata *display* dalam keterangannya.

1.4.4 Command : demo

Command `demo` berfungsi untuk membuka layar `demo` dari *software* MATLAB, utamanya mendemonstrasikan kemampuan yang dimiliki oleh MATLAB. Coba ketikkan :

```
>> demo
```

Maka MATLAB akan membuka window `demo` dan kemudian silahkan browsing menu-menu yang ada di `demo` window untuk melihat contoh penggunaan dan kemampuan dari MATLAB.

1.5 Contoh Sederhana Window Grafik

Sebelum melangkah lebih jauh dengan kemampuan MATLAB untuk pemrograman maupun penyelesaian masalah-masalah numerik, kita akan melakukan beberapa test yang sederhana. Ketikkan :

```
>> x = 4;  
>> y = 5;  
>> z = x * y;
```

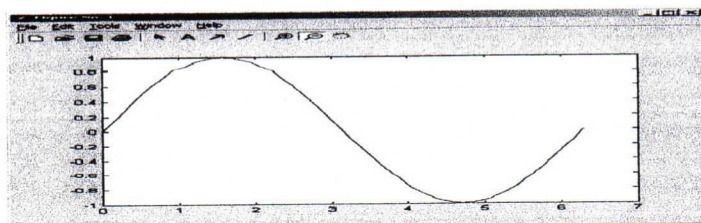
Lihat bedanya dengan mengetikkan :

```
>> z = x * y; atau  
>> x * y;
```

Pada keadaan terakhir, MATLAB mengassign hasil output ke variabel *ans* (yakni *answer*). Sekarang akan memberikan contoh sederhana penggunaan fasilitas grafik dari MATLAB. Ketikkan empat baris perintah berikut :

```
>> degree = [0:10:360];  
>> radian = degree * pi / 180;  
>> y = sin(radian);  
>> plot(radian, y);
```

Disini `degree` diassign ke suatu array `[0 10 20 ... 360]`. Kemudian `radian` menyatakan array `radian` dari angka-angka dalam array `degree` dengan rumus $\text{radian} = (\text{degree} \times \pi) / 180$. Kemudian `y = sin(radian)` akan mengassign `y` ke array `[sin(0) sin(10 π /180) ... sin(360 π /180)]` Lihat hasil dari barisan perintah di atas pada gambar dibawah ini.



1.6 Tipe Data

MATLAB mengenal tiga macam tipe data :

1. Tipe data String
2. Tipe data Skalar
3. Tipe data matriks

Array adalah matriks yang hanya terdiri atas satu baris. MATLAB memiliki banyak fungsi *built-in* untuk bekerja dan memanipulasi tipe data tersebut. Beberapa contoh dapat dilihat dalam penjelasan berikut.

1.6.1 Tipe Data String

Tipe data string merupakan tipe data yang terdiri atas huruf-huruf dan atau karakter ASCII lainnya, yang dibatasi oleh tanda ' ' (bukan tanda petik ganda). Contoh yang benar dapat diberikan sebagai berikut :

```
>>myname=' SuperHearth-Love' ;  
>>vartest='test aja ...' ;
```

Dan berikut contoh yang salah (perhatikan tanda petik !):

```
>> myname="SuperHearth-Love";  
>> vartest="test aja ...";
```

Beberapa fungsi *built-in* yang dapat digunakan untuk memanipulasi data tipe string dapat diberikan dibawah ini:

1.6.1.1 Fungsi : disp

Contoh penggunaan fungsi : disp

```
>>disp('Belajar MATLAB')  
>>x=1:5;  
>> disp(x);
```

1.6.1.2 Fungsi : input

Contoh penggunaan fungsi : input

```
>>Nama=input('Masukkan Nama l','s');  
>>disp('Nama yang anda masukkan :');  
>>disp(Nama);
```

1.6.1.3 Fungsi : fprintf

Command ini berfungsi untuk menampilkan output dilayar atau menyimpan output disuatu *external file*. Untuk lebih jelasnya silahkan melihat `help` untuk `fprintf`. Berikut diberikan contoh penggunaannya.

```
>>z='Percobaan Command fprintf';
>>Y='Ini test kedua';
>>fprintf('%s\n',z);
>>fprintf('%s\n %s\n',z,y);
>>fprintf('%s %s \n',z,y);
```



Perintah ini mirip dengan `disp`, namun lebih memiliki fleksibilitas dalam format output yang diinginkan.

1.6.2 Tipe Data Skalar

Tipe data scalar adalah nama lain dari numerik. Di dalam MATLAB kita memanipulasi data scalar menggunakan beberapa fungsi, seperti halnya untuk tipe data string.

1.6.3 Tipe Data Array

Tipe data array adalah suatu matriks yang hanya terdiri atas satu baris. Namun di dalam MATLAB, matriks dan array adalah dua jenis tipe data yang berbeda sifatnya. Di dalam MATLAB, array diberi index secara dinamik dimulai dari index 1. Misalnya kita melakukan assignment satu persatu seperti berikut ini :

```
>>a(1)=23;
>>a(2)=input('enter a(2)');
>>a(3)=a(1)+a(2);
```

Di sini akan terjadi *syntax error*. Jika kita melakukan assignment ke `a(0)`. Dalam MATLAB kita tidak perlu melakukan inisialisasi nilai dari komponen array, atau mendefinisikan ukurannya dimanapun di dalam program dalam MATLAB. MATLAB secara dynamic akan menaikkan komponen array jika kita menaikkan index array. Sebagai contoh, misalkan kita lakukan hal berikut :

```
>>a(1)=10;
>>a(3)=20;
```

Setelah dua perintah ini, array *a* akan memiliki tiga elemen. Namun kita belum melakukan assignment untuk nilai *a*(2). Dalam keadaan ini secara default *a*(2) akan diberi nilai 0 (jadi jika tidak di set, maka nilai default adalah 0). Namun ada baiknya dalam pemrograman, hendaknya kita selalu melakukan inisialisasi nilai suatu array ke nilai seharusnya.

Untuk menginisialisasi nilai ukuran array dapat digunakan command : `zeros`. Lihat contoh berikut :

```
>>a=zeros(100);
```

Dengan command ini, nilai *a*(1), ..., *A*(100) akan di set bernilai nol. Dengan demikian, selama ukuran array *a* tidak melebihi 100.

Berikut akan diberikan beberapa contoh assignment untuk array :

1.6.3.1 Mengassign nilai array

Misalkan :

```
>>a=[3 4 5 6];
```

Tentunya disini jelas kita mengassign nilai *a*(1)=3, *a*(2)=4, *a*(3)=5, dan *a*(4)=6.

Lihat contoh berikut :

```
>>x([1 5 10])=[0.1 0.3 0.2];
```

Maka kita mengassign *x*(1)=0.1, *x*(5)=0.3, dan *x*(10)=0.2, dan *x*(*i*) untuk *i*=2,3,4,6,7,8,9 bernilai nol.

Kita dapat juga mengisi nilai array dengan cara berikut :

```
>>x=3:1:6;
```

Yang berarti kita mengisi nilai array *x* dengan angka 3 sampai 6 dengan penambahan 1. Maka array *x* bernilai [3 4 5 6]. Disini penambahan bisa bernilai sembarang, bahkan mungkin negatif.

1.6.3.2 Menambahkan element dari array

Sebagai contoh, kita mungkin lakukan penambahan sebagai berikut :

```
>>x=[x 1 2] %menambahkan 2 element dibelakang  
>>x=[1 2 x 3] %menambahkan 2 element didepan,  
satu dibelakang.
```

1.6.3.3 Mengakses sebagian element array

Contoh : Kita hanya mengakses element ke-2 sampai ke-4 dari array x, maka ketikkan :

```
>>y=x(2:4);
```

Atau hanya element 1,3 dan 5, maka ketikkan :

```
>>y=x([1 3 5]);
```

1.6.3.4 Menghapus element array

Lihat contoh berikut :

- Menghapus element terakhir dari array x

```
>>x=x(1:length(x)-1);
```

- Menghapus element pada index tertentu
Misal x=[1 2 3 4 5 6], dan kita ingin menghapus x pada index ke-3 dan ke-5, maka ketikkan :

```
>>x([3 5])=[];
```

Maka sekarang x akan bernilai (1 2 4 6).

1.7 Matriks

Matriks merupakan bentuk data utama dari MATLAB. Juga array sebetulnya hanyalah matriks yang memiliki satu baris. Seperti halnya array, matriks didefinisikan elemen demi elemen.

1.7.1 Menyusun sebuah Matriks

Misalkan kita memiliki matriks berikut :

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Maka kita dapat memasukkan matriks ini kedalam MATLAB menggunakan salah satu dari cara berikut :

Cara 1

Ketikkan di prompt MATLAB :

```
>>A(1,1)=1;A(1,2)=2;A(2,1)=3;A(2,2)=4
```

Cara 2

```
>>A[ 1 2;3 4]
```

Cara 3

```
>>A1=[1 2];  
>>A2=[3 4];  
>>A=[A1;A2]
```

Cara 3 akan sangat bermanfaat jika matriks yang disusun berukuran cukup besar dan mengandung beberapa elemen yang sama. Seperti halnya array, ukuran matriks bertambah secara dinamis. Misalkan kita melakukan operasi berikut :

```
>>A=(1 2;3 4);  
>>A(2,3)=7;
```

Maka secara otomatis MATLAB akan mengassign nilai 0 untuk A(1,3). Jika kita tidak menyukai ide penambahan ukuran matriks sebelumnya dengan menggunakan command zeros. Sebagai contoh, untuk matriks A di atas, kita pesan memori untuk data berukuran matrik 2x3 dengan menggunakan :

```
>>A=zeros(2,3);
```

Ukuran matriks ini dapat diperiksa dengan menggunakan command size. Lihat contoh berikut :

```
>>[baris kolom]=size(A)
```

Atau jika kita hanya ingin mengetahui jumlah baris, atau kolom, gunakan perintah berikut :

```
>>baris=size(A,1)  
>>kolom=size(A,2)
```

Sehingga untuk mengetahui ukuran array, kita dapat gunakan perintah :

```
>>size(namaarray,2)
```

Namun output yang diinginkan akan keliru jika kita mengetikkan :

```
>> s=size(namaarray)
```

Karena output dari command size adalah dua bilangan yang menyatakan baris dan kolom dari matriks.

1.7.2 Operator dalam matriks

Operator	Keterangan
+ dan -	Penambahan dan pengurangan dengan syarat matriks/vector operand mempunyai ukuran yang sama
*	Multiplikasi dengan syarat bahwa orde kolom matriks ke-1 sama dengan orde baris matrik ke-2
'	Transpos
a.*b	Mengalikan elemen matriks a dengan elemen matriks b yang letaknya sama
a./b	Pembagian elemen matriks a dengan elemen matriks b yang letaknya sama
a.^b	Memangkatkan elemen matriks a dengan elemen matriks b yang letaknya sama

Cobalah anda latihan untuk melihat hasil operasi-operasi diatas dengan matriks anda.

1.7.3 Fungsi-fungsi matriks

Fungsi-fungsi aljabar linear dapat kita temui pada direktori matfun dalam *MATLAB Toolbox*. Fungsi-fungsi matriks tersebut antara lain :

Inv	Invers matriks
Det	Determinan
Eig	Nilai eigen dan vektor eigen
Rank	Rank matriks
Norm	Normal matriks atau vektor
Orth	Ortogonalisasi
Eye	Matriks identitas
Zeros	Matriks nol
Ones	Matriks satuan
Tril	Matriks segitiga bawah
Triu	Matriks segitiga atas
Hilb	Matriks hilbert

Contoh :

```
>>A=[1 2 3;4 5 6;7 8 0];
>>inv(A)
ans=
  -1.7778   0.8889  -0.1111
   1.5556  -0.7778   0.2222
  -0.1111   0.2222  -0.1111
>>det(A)
ans=
 27
```

*****&&&&&&&&&&*****

BAB 2 DASAR-DASAR PEMROGRAMAN FUNCTION DAN M-FILE

2.1 Function dan M-File

Untuk pembuatan program dalam MATLAB terdapat suatu teknik pemrograman yang dikenal sebagai pembentukan m-file (file yang berekstension m) dengan menggunakan command function. Editor yang digunakan adalah MATLAB editor/debugger yang telah dikenalkan di Bab 1. Membuka editor ini dapat dilakukan dengan cara :

Menggunakan menu File → New M-File di MATLAB command window, yang secara otomatis membuka MATLAB editor/debugger untuk file bernama untitled.m.

2.1.1 Membentuk m-file

M-file didefinisikan menggunakan command function. Sintaks standar command function adalah :

```
Function[output1,output2,...]=NamaFunction(input1,input2, ...)
```

Disini output1,output2, ... adalah barisan nama output yang akan dimunculkan di layar command window MATLAB, jika kita memanggil NamaFunction dengan input argumen adalah input1,input2, ...

Input dan output diatas adalah optional argumen, yang berarti kita mungkin membuat suatu function tanpa input argumen. Jika diberikan tanpa output argumen, maka yang ditampilkan adalah eksekusi dari statement/ekspresi terakhir.

Sebagai contoh, kita ingin menghitung mean dan variansi dari suatu data. Sebagai input adalah suatu data dan output adalah vektor mean dan variansi, maka kita dapat membentuk m-file sebagai berikut:

```
function [mean,variansi]=contohmfile(x);  
x=input('masukkan data =');  
n=length(x);  
jum1=0;  
jum2=0;  
for i=1:n  
    jum1=jum1+x(i);  
end;  
mean=jum1/n;  
for i=1:n  
    jum2=jum2+((x(i)-mean)^2);  
end;  
variansi=jum2/(n-1);
```

```
disp('mean adalah');
disp(mean);
disp('variansi adalah');
disp(variansi);
```

(Disini kita telah menggunakan control flow for. Lebih lanjut tentang control flow diberikan pada Bab control flow. Juga kita anggap bahwa fungsi/command built-in sum belum didefinisikan sehingga digunakan for loop diatas). Kemudian simpan file ini secara default nama yang digunakan oleh MATLAB adalah contohMFile.m, karenanya anda simpan nama m-file yang anda buat sesuai dengan nama function yang anda definisikan sendiri untuk m-file anda tersebut.

2.1.2 Menjalankan suatu m-file

Untuk memanggil/mengeksekusi m-file, pertama ke MATLAB command editor dengan mengaktifkan kembali window ini.

Untuk menjalankan program tekan F5.

Misalkan sekarang anda ingin menghitung mean dan variansi dari data berikut :

1,5 3 2,5 1,5 4 10

Maka di layar akan muncul

```
mean=
    3.7500
variansi=
    10.2750
```

2.1.3 Menambahkan help pada user defined m-file

Supaya maksud dan tujuan dari m-file yang kita definisikan dapat dipahami orang lain, maka kita perlu membuat help untuk m-file tersebut, sedemikian hingga jika kita ketikkan di MATLAB command window help nama_m-file_kita maka akan muncul keterangan/help dari nama m_file_kita. Hal ini dapat dilakukan dengan menggunakan command % dari MATLAB.

Sebagai contoh, misalkan untuk file contohMFile.m diatas kita ingin membuat helpnya, maka ubahlah m-file tersebut dengan menambahkan beberapa keterangan, seperti di bawah ini :

```

function [mean,variansi]=contohmfile(x);
%Fungsi ini akan menghitung mean dan variansi dari vektor data
%Contoh penggunaan :
%misalkan x=1:10
%maka ketikkan [mean,variansi]=contohMFile(x);
%akan menghitung nilai mean dan variansi dari x;
n=length(x);
jum1=0;
jum2=0;
for i=1:n
    jum1=jum1+x(i);
end;
mean=jum1/n;
for i=1:n
    jum2=jum2+((x(i)-mean)^2);
end;
variansi=jum2/(n-1);
disp('mean adalah');
disp(mean);
disp('varaiansi adalah');
disp(variansi);

```

>> help contohMFile

Maka di layar akan muncul :

Fungsi ini akan menghitung mean dan variansi dari vektor data

Contoh penggunaan :

misalkan x=1:10

maka ketikkan [mean,variansi]=contohMFile(x);

akan menghitung nilai mean dan variansi dari x;

Catatan : MATLAB bersifat *case sensitif* sehingga variable atau fungsi "Namafungsi" berbeda dengan "namafungsi"

2.2 Daftar Command untuk pemrograman

Daftar dari command dasar MATLAB yang dapat digunakan untuk pemrograman dapat diakses dengan mengetikkan help lang di command editor.

Beberapa command penting dapat dilihat dibawah ini :

Control flow.

If	-Conditionally execute statements.
Else	-IF statement condition.
Elseif	-IF statement condition.
End	-Terminate scope of FOR, WHILE, SWITCH, TRY and IF statements.
For	-Repeat statements a specific number of times.
While	-Repeat statements an indefinite number of times.
Break	-Terminate execution of WHILE or FOR loop.

Switch -Switch among several cases based on expression.
Case -SWITCH statement case.
Otherwise -Default SWITCH statement case.
Try -Begin TRY block.
Catch -Begin CATCH block.
Return -Return to invoking function.

Evaluation and execution.

Eval -Execute string with MATLAB expression.
Evalc -Evaluate MATLAB expression with capture.
Feval -Execute function specified by string.
Evalin -Evaluate expression in workspace.
Builtin -Execute built-in function from overloaded method.
Assignin -Assign variable in workspace.
Run -Run script.

Scripts, functions, and variables.

Script -About MATLAB scripts and M-files.
Function -Add new function.
Global -Define global variable.
Persistent-Define persistent variable.
Mfilename -Name of currently executing M-file.
Lists -Comma separated lists.
Exis -Check if variables or functions are defined.
Isglobal -True for global variables.
Mlock -Prevent M-file from being cleared.
Munlock -Allow M-file to be cleared.
Mislocked -True if M-file cannot be cleared.
Precedence -Operator Precedence in MATLAB.

Argument handling.

Nargchk -Validate number of input arguments.
Nargin -Number of function input arguments.
Nargout -Number of function output arguments.
Varargin -Variable length input argument list.
Varargout -Variable length output argument list.
Inputname -Input argument name.

Message display.

Error -Display error message and abort function.
Warning -Display warning message.
Lasterr -Last error message.
Lastwarn -Last warning message.
Errortrap -Skip error during testing.
Disp -Display an array.
Display -Overloaded function to display an array.
Fprintf -Display formatted message.
Sprintf -Write formatted data to a string.

Interactive input.

Input -Prompt for user input.
 Keyboard -Invoke keyboard from M-file.
 pause -Wait for user response.
 Uimenu -Create user interface menu.
 Uicontrol -Create user interface control.

2.3 Operator dan karakter khusus

Macam-macam operator dan karakter khusus dapat diakses dengan menggunakan command help ops. Lihat operator berikut :

Arithmetic operators.

plus	- Plus	+
uplus	- Unary plus	+
minus	- Minus	-
uminus	- Unary minus	-
mtimes	- Matrix multiply	*
times	- Array multiply	.*
mpower	- Matrix power	^
power	- Array power	.^
mldivide	- Backslash or left matrix divide	\
mrdivide	- Slash or right matrix divide	/
ldivide	- Left array divide	.\
rdivide	- Right array divide	./
kron	- Kronecker tensor product	kron

Relational operators.

eq	- Equal	==
ne	- Not equal	~=
lt	- Less than	<
gt	- Greater than	>
le	- Less than or equal	<=
ge	- Greater than or equal	>=

Logical operators.

and	- Logical AND	&
or	- Logical OR	
not	- Logical NOT	~
xor	- Logical EXCLUSIVE OR	
any	- True if any element of vector is nonzero	
all	- True if all elements of vector are nonzero	

Special characters.

colon	- Colon	:
paren	- Parentheses and subscripting	()
paren	- Brackets	[]
paren	- Braces and subscripting	{ }
punct	- Decimal point	.
punct	- Structure field access	.
punct	- Parent directory	..
punct	- Continuation	...

punct	- Separator	'
punct	- Semicolon	;
punct	- Comment	%
punct	- Invoke operating system command	!
punct	- Assignment	=
punct	- Quote	'
transpose	- Transpose	.
ctranspose	- Complex conjugate transpose	'
horzcat	- Horizontal concatenation	[,]
vertcat	- Vertical concatenation	[:,]
subsasgn	- Subscripted assignment	(), { }, ..
subsref	- Subscripted reference	(), { }, ..
subsindex	- Subscript index	

Bitwise operators.

bitand	- Bit-wise AND.
bitcmp	- Complement bits.
bitor	- Bit-wise OR.
bitmax	- Maximum floating point integer.
bitxor	- Bit-wise XOR.
bitset	- Set bit.
bitget	- Get bit.
bitshift	- Bit-wise shift.

Set operators.

union	- Set union.
unique	- Set unique.
intersect	- Set intersection.
setdiff	- Set difference.
setxor	- Set exclusive-or.
ismember	- True for set member.

2.4 Latihan dan contoh program

Buat suatu program untuk mencari luas dan keliling lingkaran dengan input file adalah jari-jari lingkaran?

BAB 3 CONTROL FLOW

Berikut ini akan dibahas perintah *control flow* untuk mengatur jalannya eksekusi dari statement di dalam MATLAB script/m-file.

3.1 For loop

Syntax dari for loop dalam MATLAB adalah :

```
For v=matrix
    Statement1;
    Statement2;
    ...
end
```

Contoh sederhana :

```
>> k=[1 2;5 6]
k =
     1     2
     5     6
>> for v=k
y=v
x=2*v
end;
y =
     1
     5
x =
     2
    10
y =
     2
     6
x =
     4
    12
```

3.2 While loop

Syntax dasar dalam MATLAB dari while loop adalah :

```
While variabel
    Statement1;
    Statement2;
    ...
End
```


Variabel diatas menyatakan suatu ekspresi boolean. Daftar ekspresi boolean di dalam **MATLAB** dapat dilihat dibawah ini :

a==b	Betul jika a sama dengan b
a > b	Betul jika a lebih besar dari b
a < b	Betul jika a lebih kecil dari b
a >= b	Betul jika a lebih besar atau sama dengan b
a <= b	Betul jika a lebih kecil atau sama dengan b
a ~= b	Betul jika a tidak sama dengan b
a & b	Betul jika kedua ekspresi boolean a dan b betul
a b	Betul jika paling sedikit satu diantara ekspresi boolean a dan b betul
a xor b	Betul jika hanya satu diantara ekspresi boolean a dan b betul
~ a	Negasi a betul jika ekspresi boolean a bernilai salah



Contoh penggunaan while-loop :

Kita akan berikan dua contoh while loop. Pertama contoh sederhana yang akan menampilkan bilangan 0 sampai 4 di layar sebagai berikut :

```
function Contohwhileloop();
%Program sederhana ini menampilkan bilangan
%dari 0 sampai 4 di layar
i=0;
while i<5
    disp(i);
    i=i+1;
end;
```

Hasil eksekusi :

```
» Contohwhileloop
0
1
2
3
4
```

3.3 Command switch dan case

Command ini berfungsi untuk melakukan suatu pengecekan syarat dari beberapa kemungkinan berdasarkan bentuk suatu ekspresi. Bentuk umum dari loop ini adalah :

```
SWITCH switch_expr
  CASE case_expr,
    Statement,...,statement
  CASE {case_expr1,case_expr2,case_expr3,...}
    Statement,...,statement
  ...
  OTHERWISE,
    Statement,...,statement
END
```

Lihat contoh berikut. Disini dimisalkan saja satuan variabel string. Tujuan dari program ini adalah untuk mengkonversi satuan ke *sentimeter*.

```
function y=konversi(x,satuan);
% Program untuk mengkonversi satuan ke sentimeter
switch lower(satuan)
case {'inch','in'}
    y=x/2.54;
case {'feet','ft'}
    y=x/(2.54*12);
case {'mater','m'}
    y=x*100;
case {'milimeter','mm'}
    y=x/10;
case {'centimeter','cm'}
    y=x;
otherwise
    disp(['satuan tidak dikenal :'satuan]);
    y=nan;
end
```

Hasil eksekusi :

```
» a=konversi(1,'in')
a =
    0.3937
» a=konversi(1,'ft')
a =
    0.0328
» a=konversi(1,'m')
a =
    100
» a=konversi(1,'mm')
a =
    0.1000
» a=konversi(1,'cm')
a =
     1
» a=konversi(1,'max')
satuan tidak dikenal :max
a =
    NaN
```

3.4 Contoh dan Latihan

Andaikan tidak tersedia command `root` dari suatu polynomial yang direpresentasikan dalam array. Buat satu program untuk menghitung akar-akar real dari persamaan kuadrat $y = ax^2 + bx + c$ menggunakan algoritma biasa. Langkah-langkah pemrograman :

- a. Input : koefisien a , b , dan c dari persamaan kuadrat (command input).
- b. Gunakan IF-ELSE untuk mencari akar real dari persamaan kuadrat itu. Dengan D adalah diskriminan $D = b^2 - 4ac$
Jika $D > 0$, maka dimiliki dua akar real tidak sama.
Jika $D = 0$, maka dimiliki dua akar real sama (tunggal).
Jika $D < 0$, maka dimiliki akar imajiner.
Sedangkan akar persamaan ini dihitung menggunakan persamaan ABC.
- c. Print out di layar dengan command `disp`.

DAFTAR PUSTAKA

- [1] Hager, W., William, 1988, *APPLIED NUMERICAL LINEAR ALGEBRA*, New Jersey : Prentice Hall, Englewood Cliffs.
- [2] Kudus, Abdul dan Sukarsi, I. 2003. *PANDUAN PRAKTIKUM METODE NUMERIK 1 Menggunakan Program Matlab*. Laboratorium Komputasi Matematika : Jurusan Matematika F-MIPA UNISBA.
- [3] Steven, C., Chapra and Raymond, P., Canale, 2002, *Numerical Methods for Engineers With programming and Software Applications*, International Edition : Mc Graw-Hill Higher Education.
- [4] <http://www.mathwork.com>