

Pemrograman Dasar Numerik

Disusun oleh:

Nur Uddin, PhD.

Universitas Pembangunan Jaya

2019

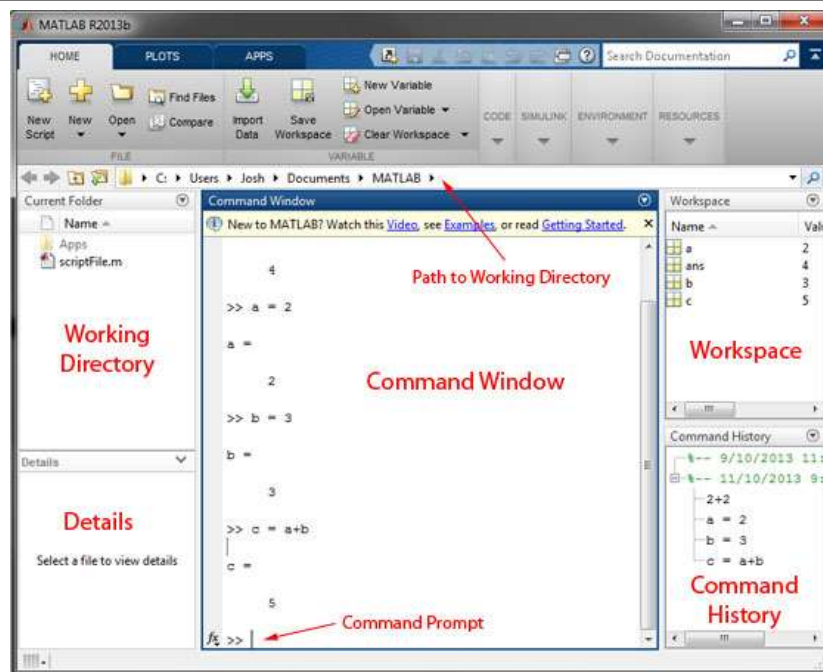
Bab I

Pengenalan Matlab

1.1 Matlab environment

Tampilan Matlab ditunjukkan pada Gambar 1.1. Dalam gambar tersebut ditunjukkan beberapa bagian sebagai berikut:

- Menubar
- Toolbar
- Current working directory
- Current directory content
- Workspace
- Command history



Gambar 1.1

1.2 Penggunaan Matlab seperti calculator

Matlab merupakan software yang cukup mudah dioperasikan bagi orang yang masih awam tentang pemrograman computer. Matlab dapat dioperasikan seperti menggunakan kalkulator sebagaimana ditunjukkan dalam contoh 1.1 sebagai berikut.

Contoh 1.1:

- a. Dalam command window ketikkan perkalian 5×10 , seperti berikut:

```
>> 5*10
```

- b. Kemudian tekan tombol Enter, maka munculah hasil perkalian tersebut:

```
>> 5*10
ans=
    50
```

1.3 Operator dasar matematika dalam Matlab

Contoh 1.1 menunjukkan operasi perkalian menggunakan Matlab. Operasi perkalian di dalam Matlab dilakukan oleh operator $*$. Tabel berikut menunjukkan operator untuk operasi dasar matematika menggunakan Matlab.

Operasi matematika	Operator		Contoh	
	Matematika	Matlab	Matematika	Matlab
Penjumlahan	+	+	$2+3$	$2+3$
Pengurangan	-	-	$2-1$	$2-1$
Perkalian	$\times$	*	3×7	$3*7$
Pembagian	:	/	$8:2$	$8/2$
Pangkat (exponensial)		$\wedge$	2^3	$2^{\wedge}3$

1.4 Operasi matematika: penjumlahan

Penjumlahan dua buah bilangan dalam Matlab diberikan dalam contoh berikut.

Contoh:

$$5 + 6 = 11$$

```
>> 5+6  
  
ans =  
  
11
```

1.5 Operasi matematika: pengurangan

Penjumlahan dua buah bilangan dalam Matlab diberikan dalam contoh berikut.

Contoh:

$$5 - 6 = -1$$

```
>> 5-6  
  
ans =  
  
-1
```

1.6 Operasi matematika: perkalian

Perkalian dua buah bilangan dalam Matlab diberikan dalam contoh berikut.

Contoh:

$$5 \times 6 = 30$$

```
>> 5*6  
  
ans =  
  
30
```

1.7 Operasi matematika: pembagian

Operasi dalam Matlab diberikan dalam contoh berikut.

Contoh:

$$12 \div 3 = 4$$

```
>> 12/3  
  
ans =  
  
4
```

Pembagian dengan nol akan menghasilkan nilai tak hingga atau *infinity* yang disimbolkan oleh ∞ . Dalam Matlab nilai tak hingga tersebut dinyatakan oleh `inf`, sebagai mana ditunjukkan dalam contoh berikut:

Contoh:

$$5 \div 0 = \infty$$

```
>> 5/0  
  
ans =  
  
Inf
```

1.8 Operasi matematika: eksponensial atau pangkat

- Operasi pemangkatan adalah suatu operasi perkalian suatu nilai dengan nilai itu sendiri dengan jumlah operasi perkalian sebanyak nilai pangkatnya.

Contoh:

Lima pangkat empat

$$5^4 = 5 \times 5 \times 5 \times 5 = 625$$

Dalam Matlab operasi pemangkatan dinyatakan oleh operator `^`. Dengan menggunakan Matlab operasi lima pangkat empat tersebut dinyatakan sebagai berikut:

1.9 Operasi matem

Dari contoh dia 625 pangkat 4 dari 625.

Operasi pencarian akar tersebut dalam matematika dinyatakan oleh:

$$\sqrt[4]{625} = (625)^{\frac{1}{4}} = 625^{0.25}$$

dalam Matlab operasi tersebut dinyatakan oleh:

```
>> 625^(1/4)
ans =
    5
```

atau

```
>> 625^0.25
ans =
    5
```

1.10 Fungsi matematika

Fungsi matematika adalah suatu proses penghitungan dari suatu input untuk menghasilkan output. Setiap fungsi memiliki nama. Secara sederhana bentuk umum sebuah fungsi adalah sebagai berikut:

$$y = namaFungsi(x)$$

Dimana:

y : output atau nilai yang dihasilkan dari proses penghitungan di dalam fungsi.

$namaFungsi$: nama sebagai identitas fungsi tersebut.

x : input atau nilai yang dimasukkan ke dalam fungsi.

Tanda kurung digunakan untuk memisahkan antara nama fungsi dan input.

Dalam pemrograman Matlab, fungsi dinyatakan dengan format yang sama seperti format diatas.

Contoh:

Dalam Matlab sqrt adalah nama fungsi untuk menghitung akar kuadrat, maka kita dapat menggunakan fungsi tersebut untuk menghitung $\sqrt{9}$ dengan command sebagai berikut:

```
>> y=sqrt(9)
```

Terkadang variabel output y tidak digunakan jika hanya ingin melihat hasil dari fungsi tersebut, seperti ditunjukkan oleh command berikut:

```
>> sqrt(9)
```

1.11 Fungsi trigonometri

Fungsi trigonometri merupakan salah satu fungsi matematika yang sering dijumpai dalam studi engineering. Penghitungan fungsi trigonometri dari suatu sudut dalam matematika bisa menggunakan radian atau derajat. Namun dalam Matlab, penghitungan fungsi trigonometri berbasis pada radian bukan derajat, sehingga semua sudut harus dinyatakan dalam radian.

Contoh:

Gunakan Matlab untuk menghitung $\sin 30^\circ$.

Jawab:

- sudut 30° harus dikonversikan kedalam radian

$$30 \text{ derajat} = 30 \times \frac{\pi}{180} \text{ radian}$$

- Nilai π di dalam Matlab dinyatakan oleh "pi" sehingga command di Matlab untuk menghitung $\sin 30^\circ$ adalah:

```
>> sin(30*pi/180)
```

Penghitungan fungsi trigometri yang lain pada prinsipnya sama dengan contoh diatas. Command Matlab untuk fungsi trigometri diberikan di Tabel berikut.

Fungsi trigonometri		Contoh	
Matematika	Matlab	Matematika	Matlab
sin	sin()	$\sin 30^\circ$	$\sin(30*\pi/180)$
cos	cos()	$\cos 60^\circ$	$\cos(60*\pi/180)$
tan	tan()	$\tan 45^\circ$	$\tan(45*\pi/180)$

1.12 Prioritas operasi matematika

Jika dalam suatu operasi matematika terdiri dari beberapa operator yang berbeda maka ada prioritas dalam operasi matematika. Prioritas operasi diberikan dalam Tabel 1.2. Prioritas operasi matematika ini berlaku dalam pemrograman di Matlab.

Tabel 1.2: Prioritas operasi matematika

Prioritas	Operasi
1	Semua operasi di dalam tanda kurung, dari tanda kurung yang paling dalam.
2	Operasi pangkat (exponensial) dan akar pangkat, dari kiri ke kanan.
3	Operasi perkalian dan pembagian, dari kiri ke kanan.
4	Operasi penjumlahan dan pengurangan, dari kiri ke kanan.

Contoh:

1. Prioritas 1:

Operasi matematika: $3 \times (1 + 2 \times (1 + 2)) = 21$

Matlab:

```
>> 3*(1+2*(1+2))
```

2. Prioritas 2:

Operasi matematika: $5 \times 4^{0.5} \times 3^2 = 90$

Matlab:

```
>> 5*4^0.5*3^2
```

3. Prioritas 3:

Operasi matematika: $5 \times 4 \div 2 \times 4 = 40$

Matlab:

```
>> 5*4/2*4
```

4. Prioritas 4:

Operasi matematika: $1 + 4 - 3 + 2 = 4$

Matlab:

```
>> 1+4-3+2
```

1.13 Mendefinisikan variable

Variabel adalah suatu elemen yang dideklarasikan untuk merepresentasikan suatu nilai.

Contoh:

Variabel x merepresentasikan nilai 5 dan y merepresentasikan nilai 4. Secara matematik kedua variabel tersebut didefinisikan sebagai berikut:

$$x = 5$$

$$y = 4$$

Deklarasi kedua variable tersebut dalam Matlab dinyatakan sebagai berikut:

```
>> x=5
```

```
x =
```

```
5
```

1.14 Penggunaan tanda titik koma (;)

Ketika sebuah variabel dideklarasikan seperti command berikut

```
>> a=10
```

dan tombol Enter ditekan, maka Matlab akan menunjukan output bahwa a=10 seperti berikut:

```
>> a=10  
  
a=  
  
    10
```

Output seperti ini adalah contoh output yang tidak perlu ditampilkan karena variabel baru saja didefinisikan. Menampilkan output tersebut akan membuat command window menjadi penuh, dan menampilkan output dalam proses iterasi akan memperlambat komputasi.

Untuk tidak menampilkan output dalam command window tersebut digunakanlah tanda titik koma (;) untuk mengakhiri setiap command dalam program Matlab, seperti: mendefinisikan variable, fungsi, atau operasi yang lain yang menghasilkan output.

Dengan menggunakan tanda titik koma (;) sebagai penutup dalam mendefinisikan suatu variabel, misalnya

```
>> a=10;
```

maka jika tombol Enter ditekan akan menghasilkan tampilan seperti berikut

```
>> a=10;  
  
>>
```

Meskipun nilai variable a tidak ditampilkan, nilai variable a tetap disimpan.

1.15 Operasi variable

Dikarenakan variable merepresentasikan suatu nilai, maka operasi matematika dapat dilakukan pada variable tersebut.

Contoh

Dengan pendeklarasian dua variabel berikut

$$x = 5 \text{ dan } y = 4$$

maka dapat dilakukan berbagai operasi matematika, seperti berikut:

- Penjumlahan: $x + y$

```
>> x=5;
>> y=4;
>> x + y
ans =
    9
```

- Pengurangan: $x - y$

```
>> x - y
ans =
    1
```

- Perkalian: $x \times y$

```
>> x*y
ans =
   20
```

Meskipun perkalian $x \times y$ sering ditulis dalam matematika sebagai xy , namun perkalian tersebut tidak dapat dideklarasikan di dalam Matlab sebagai berikut:

```
>> xy  
Undefined function or variable 'xy'.
```

Deklarasi diatas menghasilkan notifikasi error berupa '*Undefined function or variable*' karena deklarasi diatas dianggap memanggil variable atau fungsi xy yang tidak tersedia karena tidak pernah didefinisikan sebelumnya.

- Akar dari penjumlahan dua variable: $\sqrt{x + y}$

```
>> (x+y)^0.5  
ans =  
3
```

1.16 Mendefinisikan variable baru sebagai hasil operasi variabel lain

Suatu variable baru dapat didefinisikan sebagai representasi dari hasil operasi matematik variabel yang lain.

Contoh:

Diberikan $x = 5$ dan $y = 4$, hitunglah nilai $x + y$ dan simpanlah hasil penjumlahan tersebut sebagai variable z .

Jawab:

Secara matematik penjumlahan diatas dinyatakan oleh persamaan berikut:

$$z = x + y$$

dan deklarasi dalam Matlab adalah sebagai berikut:

```
>> x = 5; y = 4;  
  
>> z = x+y  
  
z =  
  
9
```

Contoh:

Diberikan $x = 5$ dan $y = 4$, dengan menggunakan Matlab hitunglah $w = \sqrt{x + y}$.

Jawab:

```
>> x = 5; y = 4;  
  
>> w = ( x + y )^0.5  
  
z =  
  
3
```

1.17 Overwrite variable

Variable ditugaskan untuk merepresentasikan suatu nilai. Penugasan tersebut bersifat sementara atau *temporary*. Meskipun suatu variable telah ditugaskan untuk merepresentasikan suatu nilai, namun variabel tersebut bisa dipindah-tugaskan untuk merepresentasikan suatu nilai yang baru dan tidak lagi mereprentasikan nilai yang lama.

Penggantian tugas variable untuk merepresentasikan nilai yang baru dan tidak lagi merepresentasikan nilai yang lama ini disebut sebagai *overwriting*.

Contoh:

- Variabel u pada awalnya digunakan untuk merepresentasikan nilai 5, namun kemudian variabel u tersebut diganti untuk merepresentasikan nilai 6.

$u = 5$ diubah menjadi $u = 6$

```
>> u = 5
u =
    5
>> u=6
u =
    6
>> u
u =
    6
```

- Variabel s digunakan untuk merepresentasikan nilai 25, namun kemudian variable s tersebut diganti untuk merepresentasikan $\sqrt{s}$.

$s = 25$ diubah menjadi $s = \sqrt{s}$

```
>> s = 25
s =
    25
>> s = s^0.5
s =
    5
>> s
s =
    5
```

- Variabel t digunakan untuk merepresentasikan nilai 2, namun kemudian variable t tersebut diganti untuk merepresentasikan $t + 1$.

$t = 2$ diubah menjadi $t = t + 1$

```
>> t = 2  
t =  
    2  
>> t = t + 1  
t =  
    3  
>> t  
t =  
    3
```

1.18 Menampilkan nama variabel yang telah didefinisikan (>> who)

Untuk mengetahui variable yang telah didefinisikan dalam command window, digunakanlah command *who*.

Contoh:

Misal kita sudah mendefinisikan variabel a, b, dan c dalam command window, maka nama-nama variabel yang telah didefinisikan tersebut bisa cek dengan command *who* dan jika tombol Enter ditekan akan muncul nama-nama variable tersebut seperti berikut:

```
>> who  
  
Your variables are:  
  
a b c
```

1.19 Membersihkan layar command window (>> clc)

Diatas kita telah mendefinisikan beberapa variable, melakukan operasi matematika, dan menjalankan fungsi di dalam command window, tentunya command window menjadi penuh. Untuk membersihkan command window digunakan command *clc*.

```
>> clc
```

Command *clc* hanyalah menghapus tulisan command window saja, variabel yang tersimpan di dalam command window masih tetap tersimpan.

1.20 Menghapus variabel (>> clear nama_variabel)

Untuk menghapus variable yang tersimpan dalam command window, digunakan command *clear*.

Contoh:

Variable a,b, dan c telah tersimpan dalam command window.

- Gunakan command *who* untuk melihat variable yang tersimpan.

```
>> who  
  
Your variables are:  
  
a b c
```

- Hapuslah variabel a dengan command *clear a* seperti berikut:

```
>> clear a
```

- Sekarang cek variabel yang tersimpan dengan command *who*

```
>> who  
  
Your variables are:  
  
b c
```

1.21 Menghapus semua variabel yang tersimpan (>> clear all)

Untuk menghapus semua variabel yang tersimpan dalam command window, gunakan command *clear all*.

```
>> clear all
```

Sekarang cek variabel yang tersimpan dalam command window dengan command *who* seperti berikut, maka tidak akan muncul nama variabel yang tersimpan karena sudah tidak ada lagi variabel yang tersimpan dalam command window.

```
>> who  
  
>>
```

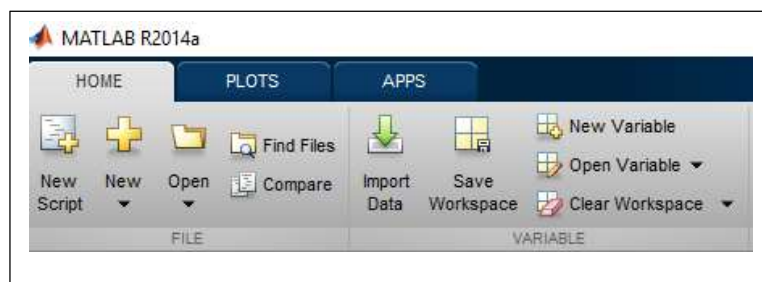
Bab 2

Pemrograman dalam M-file dan Function

2.1 Pengenalan M-file

Pada pembahasan sebelumnya di Bab 1 telah didemonstrasikan beberapa contoh operasi Matlab seperti operasi aritmatika, trigonometri, pendefinisian variable, dan penghapusan variabel. Sejauh ini operasi Matlab tersebut langsung dituliskan pada command window. Command window merupakan tempat dimana semua operasi Matlab dijalankan, namun apa yang dituliskan di command window tersebut tidak tersimpan sehingga kita harus menuliskan lagi setiap kali melakukan operasi Matlab. Jika operasi Matlab ini hanya terdiri dari satu atau dua baris tentunya tidak ada masalah untuk menuliskannya setiap kali melakukan operasi, namun jika operasi tersebut terdiri dari banyak baris tentunya akan sangat tidak efisien untuk menuliskannya setiap kali melakukan operasi Matlab.

Akan menjadi lebih efisien jika operasi yang terdiri dari banyak baris tersebut disimpan menjadi suatu program dan untuk menjalankan operasi tersebut cukup memanggil nama program tersebut di command window. Matlab mengantisipasi hal tersebut dengan menyediakan fasilitas berupa sebuah text editor untuk menuliskan suatu program. Text editor ini dapat dibuka dengan menekan ikon **new script** seperti ditunjukkan oleh gambar berikut.



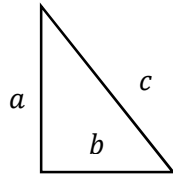
Program yang ditulis dalam text editor ini akan disimpan dalam suatu file dengan ekstensi *m* (bentuk umum nama filenya adalah **nama_file.m**). File tersebut dikenal dengan nama M-file. Program yang tersimpan dalam M-file tersebut dapat dijalankan secara langsung di Matlab command window dengan mengetikkan nama filenya (**nama_file**).

2.2 Pembuatan program m-file sederhana

Berikut diberikan contoh pembuatan program m-file yang sederhana.

Contoh 2.1:

Membuat program m-file untuk menghitung panjang sisi miring dari suatu segita siku-siku dengan rumus Pythagoras.



$$c^2 = a^2 + b^2$$

Misal diketahui panjang $a = 4$ dan $b = 3$, tentukan panjang c !

Penghitungan panjang c dapat diselesaikan dengan sebuah program m-file dengan langkah-langkah sebagai berikut:

- i. Buka m-file editor dan tuliskan code berikut:

```
a = 4;  
b=3;  
c= (a^2+b^2)^0.5
```

- ii. Simpan code tersebut, misal dengan nama **pytha1.m**
- iii. Jalankan program tersebut dengan mengetikan nama file di command window dan tekan enter. Pastikan Matlab bekerja pada folder dimana m-file tersebut disimpan. M-file tersebut juga dapat dijalankan dengan menekan tombol Run di menu Editor.

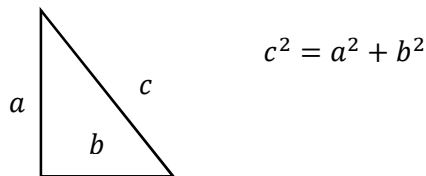
```
>> pytha1  
  
c =  
  
5
```

2.3 Pembuatan program m-file sederhana yang memiliki input

Dalam Contoh 2.1, program m-file hanya memberikan output penghitungan berdasarkan nilai variable yang didefinisikan di dalam program m-file. Bila program **pytha1.m** tersebut digunakan untuk menghitung segitiga yang berbeda, maka pajang sisi segitiga tersebut harus didefinisikan di dalam program m-file. Hal ini membuat program tersebut tidak fleksibel. Akan lebih praktis jika data segitiga dapat dimasukan tanpa harus mendefinisikannya di m-file. Matlab menyediakan fasilitas tersebut dengan command input sebagai suatu cara agar program berinteraksi dengan pengguna untuk memasukan data.

Contoh 2.2:

Membuat program m-file untuk menghitung panjang sisi miring dari suatu segita siku-siku dengan rumus Pythagoras, dimana nilai a dan b dinyatakan sebagai input.



- i. Buka m-file editor dan tuliskan code berikut:

```
a=input('Masukan panjang a: ');  
b=input('Masukan panjang b: ');  
c=(a^2+b^2)^0.5
```

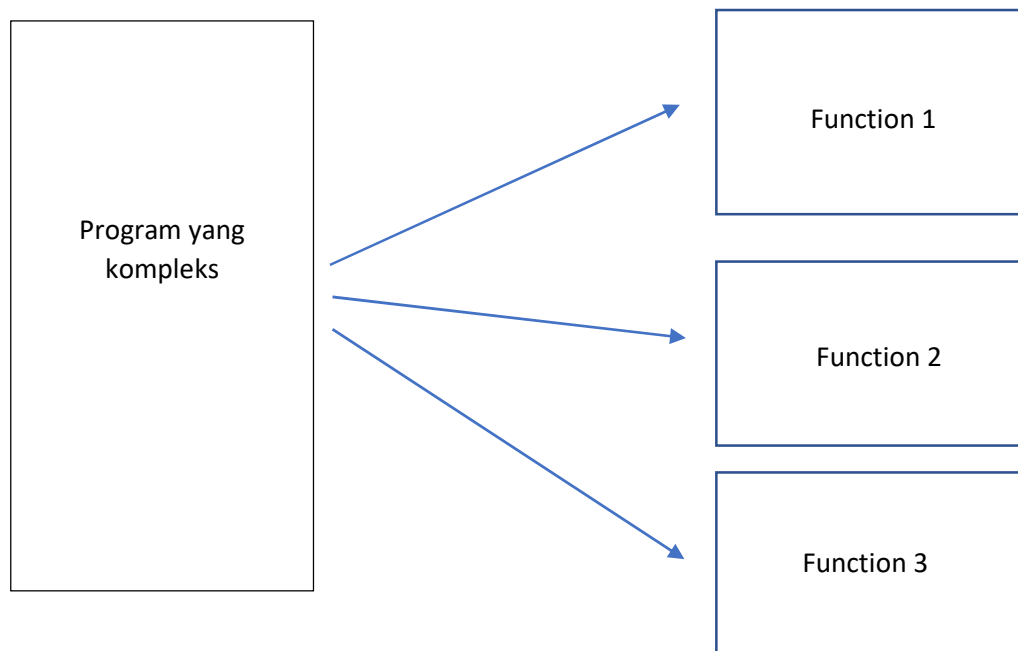
- ii. Simpan code tersebut, misal dengan nama **pytha2.m**
- iii. Jalankan program tersebut dengan mengetikan nama file di command window dan tekan enter.

```
>> pytha2  
Masukan panjang a: 6  
Masukan panjang b: 8  
  
c =  
  
10
```

2.4 Function

Di pembahasan sebelumnya diberikan dua contoh program, *pytha1.m* dan *phyta2.m*. Kedua program tersebut merupakan program yang sederhana dimana didalam program tersebut hanya terdapat satu proses perhitungan. Program sederhana hanya digunakan dalam latihan ketika memulai belajar membuat program. Dalam aplikasi nyata, jarang sekali ditemui program yang sederhana, sebuah program biasanya terdiri dari banyak proses perhitungan.

Sebuah program yang kompleks yang terdiri dari banyak proses perhitungan menyebabkan program tersebut bisa terdiri dari banyak baris sehingga menyulitkan dalam pembuatan program, tracing error, maintenance, dan modifikasi. Untuk mempermudah hal tersebut, sebuah program yang kompleks dipecah menjadi beberapa proses komputasi yg lebih sederhana yang disebut **function**. Gambar 2.1 memberikan ilustrasi sebuah program yang kompleks dipartisi kedalam tiga buah function.



Gambar 2.1: Ilustrasi sebuah program m-file yang kompleks dipartisi menjadi tiga function.

Function merupakan sebuah modul komputasi. Sebuah function terdiri dari tiga elemen dasar: input, proses, dan output. Input adalah data yang diberikan kepada function untuk diolah melalui sebuah proses komputasi. Proses komputasi tersebut menghasilkan suatu data yang disebut sebagai output dari function tersebut. Function dalam Matlab dibuat di

dalam Matlab Editor dengan format sebagai berikut. **Function yang dibuat harus disimpan dengan nama yang sama dengan namaFunction.**

1. Function dengan satu input satu output:

function y=namaFunction(x)

Contoh:

Membuat function untuk menghitung sinus suatu sudut dalam derajat.

- i. Buka m-file editor baru (Control+N) dan tuliskan code berikut:

```
function y = sinDerajat(x)
a=x*pi/180; % konversi dari derajat ke radian
y=sin(a);
```

- ii. Simpan m-file dengan nama **sinDerajat.m**
- iii. Jalankan function tersebut di command window. Pemanggilan function harus sama persis dengan nama function termasuk huruf besar dan huruf kecil.

```
>> sinDerajat(30)

ans =

    0.5000

>>
```

2. Function dengan banyak input satu output:

function y=namaFunction(x1,x2,...,xn)

Contoh:

Membuat function untuk menghitung energy kinetic sebuah benda yang bermassa m bergerak dengan kecepatan v.

$$E_k = \frac{1}{2}mv^2$$

- i. Buka m-file editor baru dan ketikan code berikut:

```
function Ek = energiKinetik(m,v)
Ek = 0.5*m*v^2;
```

- ii. Simpan dengan nama **energiKinetik.m**
- iii. Jalankan function tersebut di command window, misal untuk menghitung energi kinetic dari benda bermassa 5 kg dan bergerak dengan kecepatan 4 m/s.

```
>> energiKinetik(5,4)

ans =

    40
```

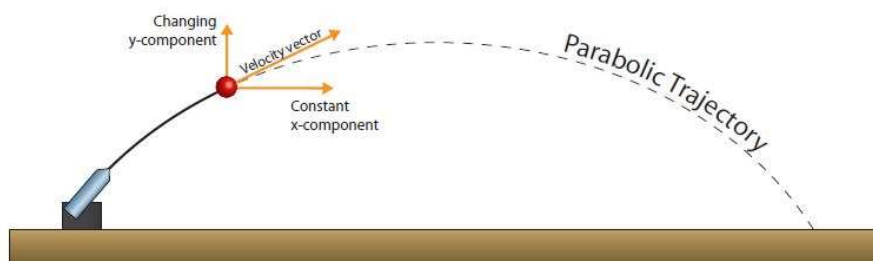
3. Function dengan satu input banyak output:

function [y1,y2,...,yn] = namaFunction(x)

Contoh:

Sebuah meriam terpasang dengan sudut tetap 30 derajat. Sebuah peluru ditembakkan dari meriam tersebut dengan kecepatan awal v_0 m/s. Jika percepatan gravitasi 9.8 m/s^2 , tentukan:

- Jarak horizontal terjauh dari meriam tersebut.
- Tentukan tinggi peluru maksimal
- Lama peluru terbang



Jawab:

Rumus untuk menghitung jarak horizontal terjauh, tinggi maksimum, dan lama peluru terbang diberikan oleh rumus berikut:

$$\text{Jarak horizontal terjauh : } x_{max} = \frac{v_0^2 \sin 2\theta}{g}$$

$$\text{Tinggi maksimum : } y_{max} = \frac{v_0^2 \sin^2 \theta}{2g}$$

$$\text{Lama peluru terbang : } t_f = \frac{2v_0 \sin \theta}{g}$$

Sebuah Matlab function dapat dibangun untuk perhitungan tersebut berdasarkan rumus-
rumus diatas dengan langkah-langkah sebagai berikut:

- i. Buka m-file editor baru dan ketikan code berikut:

```
function [xm, ym, tf] = fixedParabolic(v0)
g=9.8;
theta=30*pi/180;
xm=v0^2*sin(2*theta);
ym=v0^2*(sin(theta))^2/(2*g);
tf=2*v0*sin(theta)/g;
```

- ii. Simpan dengan nama **fixedParabolic.m**
- iii. Jalankan function tersebut di command window, jika diberikan kecepatan awal peluru 30 m/s.

```
>> [xm,ym,tf] = fixedParabolic(30)

xm =

    779.4229

ym =

    11.4796

tf =

    3.0612
```

4. Function dengan banyak input banyak output:

function [y1,y2,...,yn] =namaFunction(x1,x2,...,xn)

Contoh:

Sebuah meriam terpasang dengan sudut θ dalam derajat yang nilainya bervariasi (dapat ubah-ubah). Sebuah peluru ditembakkan dari meriam tersebut dengan kecepatan awal v_0 m/s. Jika percepatan gravitasi 9.8 m/s^2 , buatlah sebuah function untuk menghitung ketiga data berikut:

- Jarak horizontal terjauh dari meriam tersebut.
- Tentukan tinggi peluru maksimal
- Lama peluru terbang

Jawab:

- Rumus penghitungan:

$$\text{Jarak horizontal terjauh : } x_{\max} = \frac{v_0^2 \sin 2\theta}{g}$$

$$\text{Tinggi maksimum : } y_{\max} = \frac{v_0^2 \sin^2 \theta}{2g}$$

$$\text{Lama peluru terbang : } t_f = \frac{2v_0 \sin \theta}{g}$$

- Membuat Matlab function:

Buka m-file editor baru dan ketikkan kode program berikut.

```
function [xm, ym, tf] = varyingParabolic(v0, theta)
g=9.8;
thetaR=theta*pi/180; % mengubah derajat ke radian
xm=v0^2*sin(2*thetaR);
ym=v0^2*(sin(thetaR))^2/(2*g);
tf=2*v0*sin(thetaR)/g;
```

Simpan kode program tersebut dengan nama **varyingParabolic.m**

iii. Mencoba menggunakan function yang dibuat untuk penghitungan untuk dua kasus berikut:

- Sudut meriam $\theta = 45$ derajat, kecepatan awal peluru $v_0 = 50$ m/s

```
>> [xmax,ymax,tf]=varyingParabolic(50,45)

xmax =

    2500

ymax =

    63.7755

tf =

    7.2154
```

- Sudut meriam $\theta = 60$ derajat, kecepatan awal peluru $v_0 = 40$ m/s

```
>> [xm,ym,tf]=varyingParabolic(50,45)

xm =

    2500

ym =

    63.7755

tf =

    7.2154
```

Bab 3

Vektor dan Operasi Vektor

Salah satu objek yang sangat penting dalam analisa numeric adalah matriks. Matrik adalah susunan angka atau variable dengan ukuran tertentu ($m \times n$). Vektor merupakan bentuk special dari matrik, dimana vektor adalah matriks yang berukuran $m \times 1$ atau $1 \times n$. Vektor yang berukuran $m \times 1$ disebut sebagai column vector dan vektor yang berukuran $1 \times n$ disebut row vector.

3.1 Vektor

Dalam fisika, vektor didefinisikan sebagai sesuatu yang mempunyai besar dan arah. Dalam matematika, vektor didefinisikan sebagai sesuatu yang dapat dijumlahkan atau dikalikan dengan suatu skalar. Vektor dapat memiliki satu, dua, hingga n . Vektor biasanya dinyatakan sebagai susunan angka atau variable di dalam tanda kurung kotak. Terdapat dua jenis vector: column vector dan row vector.

a. Column vector

Column vector adalah vector yang hanya memiliki satu column. Berikut contoh *column vector* dan ekspresi dalam Matlab.

Contoh 1:

Column vector dan ekspresi dalam Matlab

<i>Column vector</i>	Contoh	Expresi di Matlab*
Dimensi satu	$[3]$	<code>>> [3]</code>
Dimensi dua	$\begin{bmatrix} 4 \\ 5 \end{bmatrix}$	<code>>> [4;5]</code>
Dimensi tiga	$a = \begin{bmatrix} 2 \\ 7 \\ 8 \end{bmatrix}$	<code>>> a= [2;7;8]</code>

* elemen dalam column vector dipisahkan dengan tanda titik koma (;) untuk ganti baris

b. Row vector

Row vector merupakan vector yang hanya terdiri dari satu baris. Berikut diberikan contoh row vector dan ekspresinya dalam Matlab.

Contoh 2: Row vector dan ekspresi dalam Matlab

Row vector	Contoh	Expresi di Matlab*
Dimensi satu	[3]	>> [3]
Dimensi dua	[3 4]	>> [3 4]
Dimensi tiga	$b = [2 \ 7 \ 8]$	>> b=[2,7,8]

* elemen dalam row vector dipisahkan oleh spasi atau tanda koma (,).

3.2 Ukuran vector

Ukuran vector sangat diperlukan dalam operasi vektor, sebagai contoh operasi penjumlahan dan pengurangan dua buah vector hanya dapat dilakukan jika kedua vector memiliki ukuran yang sama. Ukuran vektor dapat diperoleh dengan menggunakan Matlab melalui command: **size()**

Contoh:

Tentukan ukuran vektor berikut:

$$\text{a) } v = [3 \ 1 \ 0] \qquad \text{b) } u = \begin{bmatrix} 5 \\ 3 \\ 4 \end{bmatrix} \qquad \text{c) } w = 0$$

Jawab:

- i. Ukuran vektor v dapat diperoleh dengan code berikut:

```
>> v=[3 1 0]

v =

     3     1     0

>> size(v)

ans =

     1     3
```

Hasil tersebut menunjukkan bahwa vektor v memiliki ukuran 1×3 yang artinya vektor v terdiri dari 1 baris dan 3 kolom.

- ii. Ukuran vektor u dapat diperoleh dengan code berikut:

```
>> u=[5;3;4]

u =

     5
     3
     4

>> size(u)

ans =

     3     1
```

Hasil tersebut menunjukkan bahwa vektor u memiliki ukuran 3×1 yang artinya vektor u terdiri dari 3 baris dan 1 kolom.

- iii. Ukuran vektor w dapat diperoleh dengan code berikut:

```
>> w=0

w =

     0

>> size(w)

ans =

     1     1
```

Variabel w merupakan suatu skalar. Hasil penghitungan menggunakan Matlab menunjukkan bahwa vektor w memiliki ukuran 1×1 yang artinya vektor w terdiri dari 1 baris dan 1 kolom. Dengan demikian skalar merupakan suatu vektor yang memiliki ukuran 1×1 .

3.3 Transpose

Transpose adalah sebuah operasi untuk merubah bentuk sebuah column vector menjadi row vector atau sebaliknya. Transpose dalam matematik dinyatakan dengan menuliskan superscript T dari suatu vektor.

Di dalam Matlab, transpose dinyatakan dengan tanda petik satu (') dari suatu vektor. Misal transpose dari vektor a dinyatakan sebagai a^T dan Matlab dinyatakan sebagai `>> a'`

Contoh 3:

Vektor dan transpose	Expresi di Matlab
$[3]^T = [3]$	<code>>> [3]'</code> ans 3
$[3 \ 4]^T = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$	<code>>> [3 4]'</code> ans 3 4
$b = [2 \ 7 \ 8]$ $b^T = \begin{bmatrix} 2 \\ 7 \\ 8 \end{bmatrix}$	<code>>> b=[2 7 8]</code> ans 2 7 8 <code>>>b'</code> ans 2 7 8

3.4 Operasi Dasar Vector

Operasi dasar vektor yang dibahas dalam section ini meliputi penjumlahan dan pengurangan vektor, perkalian vektor dengan skalar, dan perkalian vektor dengan vektor.

3.4.1 Penjumlahan dan pengurangan dua buah vektor

Penjumlahan atau pengurangan vektor hanya dapat dilakukan jika ukuran dari vektor tersebut sama.

Contoh:

- a. Definisikan tiga vektor berikut:

```
>> q=[3 5 5];  
>> r=[6 3 7];  
>> s=[3; 5; 6];
```

- b. Lakukan operasi penjumlahan dan pengurangan berikut:

```
>> q+r  
ans =  
    9    8   12
```

```
>> q-r  
ans =  
   -3    2   -2
```

- c. Investigasi kenapa operasi berikut tidak berhasil?

```
>> q+s  
  
Error using +  
Matrix dimensions must agree.
```

- d. Apa yang perlu dilakukan supaya operasi $q+s$ dapat dilakukan ?
[Silahkan coba untuk mencari jawabannya sendiri.]

3.4.2 Perkalian vektor dengan skalar

Vektor dapat dikalikan dengan skalar.

Contoh:

Perkalian vektor dengan skalar	Expresi di Matlab
$5 \times [3] = [15]$	<pre>>> 5* [3] ans 15</pre>
$5 \times [3 \ 4] = [15 \ 20]$	<pre>>> 5* [3 4] ans 15 20</pre>
$b = [2 \ 7 \ 8]$ $b^T = \begin{bmatrix} 2 \\ 7 \\ 8 \end{bmatrix}$ $3b^T = \begin{bmatrix} 6 \\ 21 \\ 24 \end{bmatrix}$	<pre>>> b=[2 7 8] ans 2 7 8 >> 3*b' ans 6 21 24</pre>
$c = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	<pre>>> c=[0;1;0] c = 0 1 0 >> -5*c ans = 0 -5 0</pre>

3.4.3 Perkalian vektor

Perkalian vektor mengikuti aturan matriks dimana:

- a. Perkalian dua buah: vektor a berukuran $m \times 1$ dan vektor b berukuran $1 \times n$ akan menghasilkan vektor c yang berukuran $m \times n$.

$$[a]_{m \times 1} [b]_{1 \times n} = [c]_{m \times n}$$

Contoh:

Perkalian vektor dengan vektor	Expresi di Matlab
Diberikan dua buah vektor: $a = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, b = [4 \quad 5]$ Hitunglah: $c = ab$	<pre>>> a=[1;2;3] a = 1 2 3 >> b=[4,5] b = 4 5 >> c=a*b c = 4 5 8 10 12 15 >> size(c) ans = 3 2</pre>

Vektor a berukuran 3×1 dan vektor b berukuran 1×2 , perkalian vektor a dan b menghasilkan vektor dengan ukuran 3×2 .

- b. Perkalian vektor a berukuran $m \times 1$ dengan vektor a transpose menghasilkan matrik bujursangkar d yang berukuran $m \times m$.

$$[a]_{m \times 1} [a^T]_{1 \times m} = [c]_{m \times m}$$

Contoh:

Perkalian vektor dengan vektor	Expresi di Matlab
Diberikan vektor: $a = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$ Hitunglah: $c = aa^T$	<pre>>> a=[3;5;1] a = 3 5 1 >> size(a) ans = 3 1 4 >> c=a*a' c = 9 15 3 15 25 5 3 5 1 >> size(c) ans = 3 3</pre>

- c. Perkalian vektor b yang berukuran $1 \times n$ dengan vektor b transpose menghasilkan skalar e (vektor berukuran 1×1).

$$[b]_{1 \times n} [b^T]_{n \times 1} = [d]_{1 \times 1}$$

Contoh:

Perkalian vektor dengan vektor	Expresi di Matlab
Diberikan vektor: $b = [4 \quad 5 \quad 1]$ Hitunglah: $d = bb^T$	<pre>>> b=[4 5 1] b = 4 5 1 >> size(b) ans = 1 3 >> d=b*b' d = 42 >> size(d) ans = 1 1</pre>
Diberikan vektor: $a = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$ Hitunglah: $c = aa^T$	<pre>>> a=[3;5;1] a = 3 5 1 >> size(a) ans = 3 1 >> c=a'*a c = 35 >> size(c) ans = 1 1</pre>

3.5 Operasi vektor secara element by element

Operasi vektor secara element by element adalah melakukan operasi pada element vektor. Operasi vektor secara element by element ditandai dengan adanya tanda titik sebelum operator.

3.5.1 Operasi ekponensial vektor element by element

Operasi vector element by element juga dapat dilakukan untuk operasi ekponensial dengan symbol (.^)

Contoh:

Operasi pembagian vektor element by element	Expresi di Matlab
Diberikan vektor: $b = [4 \quad 9 \quad 10]$ Hitunglah nilai kuadrat setiap element vektor b.	<pre>>> b = [4 9 10]; >> d = b.^2 d = 16 81 100</pre>

3.5.2 Operasi perkalian vektor element by element

Perkalian vektor element by element adalah mengkalikan element yang sama dari dua buah vector. Misal c adalah hasil perkalian element by element dari vector a dan b , perkalian ini dinyatakan sebagai berikut:

$$[c_{i1}] = [a_{i1} \times b_{i1}] \quad \text{untuk } a \text{ dan } b \text{ merupakan column vector}$$

atau

$$[c_{1i}] = [a_{1i} \times b_{1i}] \quad \text{untuk } a \text{ dan } b \text{ merupakan row vector.}$$

Perkalian vector element by element dalam Matlab dinyatakan dengan symbol (*).

Contoh:

Operasi perkalian vektor element by element	Expresi di Matlab
Diberikan vektor: $b = [4 \quad 5 \quad 1]$ $c = [2 \quad 3 \quad 2]$ Hitunglah hasil perkalian element by element vektor b dan c. $d = b \times c$	<pre>>> b = [4 5 1]; >> c = [2 3 2]; >> d = b .* c d = 8 15 2</pre>

3.5.3 Operasi pembagian vektor element by element

Pembagian vector element by element dinyatakan dengan symbol ($./$) dan prinsip operasinya sama dengan perkalian vector element by element.

Contoh:

Operasi pembagian vektor element by element	Expresi di Matlab
Diberikan vektor: $b = [4 \quad 9 \quad 10]$ $c = [2 \quad 3 \quad 5]$ Hitunglah hasil pembagian element by element vektor b oleh c. $d = b ./ c$	<pre>>> b = [4 9 10]; >> c = [2 3 5]; >> d = b ./ c d = 2 3 2</pre>

3.5.4 Operasi vektor secara *element by element* pada suatu fungsi

Vektor dapat dijadikan input bagi suatu fungsi. Function dalam Matlab melakukan operasi komputasi secara element by element.

Contoh:

```
>> x = [100 ; 4 ; 81]
x =
    100
     4
     81

>> y = sqrt(x)
y =
    10
     2
     9
```

Contoh:

```
>> theta=[0 30 60 90];
theta =
    0    30    60    90

>> sin(theta*pi/180)
ans =
    0    0.5000    0.8660    1.0000
```

Bab 4

Matriks dan Operasi Matriks

Matrik merupakan suatu variable yang memiliki elemen sebanyak satu atau lebih. Matrik memiliki struktur elemen dengan ukuran tertentu ($m \times n$), dimana m menyatakan banyaknya baris dan n menyatakan banyaknya kolom. Vektor merupakan bentuk special dari matrik dimana baris atau kolomnya berjumlah satu.

Salah satu keunggulan Matlab dari bahasa pemrograman yang lain adalah Matlab memberikan kemudahan dalam melakukan operasi matriks. Matriks dideklarasikan dengan sangat sederhana didalam pemrograman menggunakan Matlab, begitu juga dengan operasi matriks.

4.1 Deklarasi matriks

Cara mendeklarasikan matriks dilakukan secara mudah baik dalam command window atau m-file. Setiap elemen matriks dalam satu baris dipisahkan oleh spasi atau tanda koma. Tanda titik koma (;) digunakan untuk memisahkan baris dalam matriks.

Contoh:

Mendeklarasikan matriks $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 7 \end{bmatrix}$ dan matriks $B = \begin{bmatrix} 5 & 3 & 1 \\ 2 & 1 & 0 \end{bmatrix}$ dalam Matlab.

```
>> A=[1 2 3;3 4 7]
A =
     1     2     3
     3     4     7

>> B=[5,3,1;2,1,0]
B =
     5     3     1
     2     1     0
```

4.2 Ukuran matriks

Ukuran matriks dapat diketahui dengan perintah “*size*”.

Contoh:

Tentukan ukuran matriks $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 7 \end{bmatrix}$.

Jawab:

```
>> A=[1 2 3;3 4 7]
A =
     1     2     3
     3     4     7

>> size(A)
ans =
     2     3
```

Matriks A berukuran 2×3 yang berarti matriks tersebut terdiri dari dua baris dan tiga kolom.

4.3 Penjumlahan dan pengurangan matriks

Pada dua buah matriks, operasi penjumlahan dan pengurangan hanya dapat dilakukan jika dan hanya jika kedua matriks tersebut memiliki ukuran yang sama.

Contoh:

Diberikan tiga matriks sebagai berikut

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 7 \end{bmatrix}, B = \begin{bmatrix} 4 & 6 & 1 \\ 5 & 2 & 3 \end{bmatrix}, \text{ dan } C = \begin{bmatrix} 3 & 2 \\ 5 & 6 \end{bmatrix}.$$

Deklarasikan ketiga matriks tersebut dalam Matlab dan lakukan operasi penjumlahan berikut:

Operasi matriks	Code program
-----------------	--------------

$A + B$	<pre>>> A = [1 2 3;3 4 7]; >> B = [4 6 1; 5 2 3]; >> C = [3 2; 5 6]; >> A+B ans = 5 8 4 8 6 10</pre>
$A - B$	<pre>>> A = [1 2 3;3 4 7]; >> B = [4 6 1; 5 2 3]; >> C = [3 2; 5 6]; >> A-B ans = -3 -4 2 -2 2 4</pre>
$A + C$	<pre>>> A = [1 2 3;3 4 7]; >> B = [4 6 1; 5 2 3]; >> C = [3 2; 5 6]; >> A+C Error using + Matrix dimensions must agree.</pre>

Investigasi apa yang menyebabkan penjumlahan $A + C$ menghasilkan error.

4.4 Perkalian matriks dengan skalar

Perkalian matrik dengan scalar akan menghasilkan matrik dengan ukuran yang sama dengan matriks tersebut namun elemen matrik akan bernilai sebesar hasil perkalian element matrik dengan scalar tersebut.

Contoh:

Diberikan matriks

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 7 \end{bmatrix},$$

Hitunglah nilai $B = 5A$.

Jawab:

$$B = 5A = \begin{bmatrix} 5 & 10 & 15 \\ 15 & 20 & 35 \end{bmatrix},$$

Deklarasi program Matlab untuk perkalian tersebut adalah:

```
>> A = [1 2 3;3 4 7]
```

```
A =
```

```
1   2   3
3   4   7
```

```
>> size(A)
```

```
ans =
```

```
2   3
```

```
>> B=5*A
```

```
B =
```

```
5   10   15
15   20   35
```

```
>> size(B)
```

```
ans =
```

```
2   3
```

4.5 Perkalian matriks dengan matriks

Operasi perkalian matrik A dan B dapat dilakukan jika dan hanya jika matriks A memiliki ukuran $m \times n$ dan matriks B memiliki ukuran $n \times p$. Hasil perkalian kedua matriks tersebut akan menghasilkan matriks dengan ukuran $m \times p$.

$$[A]_{m \times n} [B]_{n \times p} = [C]_{m \times p}$$

Contoh:

Diberikan matriks $A = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$ dan $B = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 7 \end{bmatrix}$, hitunglah nilai:

- $C = AB$
- $D = BA$

Jawab:

- Penjumlahan matriks $C = AB$

Operasi matriks	Code program
$C = AB$	<pre> >> A=[2 -1; 3 4] A = 2 -1 3 4 >> size(A) ans = 2 2 >> B = [1 2 3; 3 4 7] B = 1 2 3 3 4 7 >> size(B) ans = 2 3 >> C=A*B C = -1 0 -1 15 22 37 >> size(C) ans = 2 3 </pre>

b. Perkalian matriks $D = BA$

Operasi matriks	Code program
$D = BA$	<pre> >> A=[2 -1;3 4] A = 2 -1 3 4 >> size(A) ans = 2 2 >> B = [1 2 3;3 4 7] B = 1 2 3 3 4 7 >> size(B) ans = 2 3 >> D=B*A Error using * </pre>

Investigasi apa yang menyebabkan operasi perkalian $D=B*A$ terjadi error.

3.6 Transpose Matriks

Transpose adalah sebuah operasi untuk merubah posisi suatu elemen matriks yang terletak pada baris m dan kolom n menjadi baris n dan kolom m. Operasi transpose satu matriks secara matematik dinotasikan dengan tanda superscript T suatu matriks. Di dalam Matlab, transpose dinyatakan dengan memberikan tanda petik satu (') pada suatu matriks.

Sebagai contoh transpose suatu matriks A secara matematik dinyatakan oleh A^T dan dalam Matlab dinyatakan oleh A' .

Contoh:

Matriks dan transpose matriks	Expresi di Matlab
$[3]^T = [3]$	>> [3]' ans 3
$[3 \ 4]^T = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$	>> [3 4]' ans 3 4
$b = [2 \ 7 \ 8]$ $b^T = \begin{bmatrix} 2 \\ 7 \\ 8 \end{bmatrix}$	>> b=[2 7 8] ans 2 7 8 >>b' ans 2 7 8
$A = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$ $A^T = \begin{bmatrix} 2 & 3 \\ -1 & 4 \end{bmatrix}$	>> A=[2 -1; 3 4] A = 2 -1 3 4 >> A' ans = 2 3 -1 4

Bab 4

Loop Computation

Dalam bab ini akan dibahas loop computation menggunakan *for* dimana komputasi dilakukan berulang hingga jumlah perulangan tertentu yang telah didefinisikan. Loop computation menggunakan *for* sering digunakan dalam penghitungan matematika, misalnya dalam penghitungan integral. Dalam Matlab loop computation biasanya ditulis dalam m-file.

Loop computation menggunakan *for* di dalam Matlab ditulis dalam m-file dengan format sebagai berikut:

```
for k=1:N
    operasi
end
```

dimana *k* menyatakan indeks pengulangan dan *N* menyatakan banyaknya pengulangan. Dalam pemrograman menggunakan Matlab, indeks pengulangan *k* tersebut merupakan bilangan bulat positif ($k > 0$). Dalam loop tersebut, operasi dikerjakan secara berulang sebanyak *N* kali pengulangan.

Contoh 1:

Hitung jumlah bilangan bulat dari 0 sampai dengan 20

Jawab:

- a. Secara matematik penjumlahan tersebut dapat dinyatakan sebagai berikut:

$$y = 1 + 2 + \cdots + 20 = \sum_{k=1}^{20} k$$

- b. Penjumlahan tersebut dapat dinyatakan dalam program Matlab dalam m-file sebagai berikut:

```
y=0;           % initial value
for k=1:20
    y=y+k;
end
```

Dalam program diatas, diperlukan inisialisasi y untuk mendefinisikan nilai awal y. Program bekerja dengan alur sebagai berikut:

- i. Nilai y awal : $y=0$
- ii. Pada saat $k=1$, nilai $y=0$ (dari nilai awal) dan $y=y+k$ mengimplikasikan $y=0+1$ sehingga menghasilkan $y=1$.
- iii. Pada saat $k=2$, nilai $y=1$ (dari perhitungan di $k=1$) dan $y=y+k$ mengimplikasikan $y=1+1$ sehingga menghasilkan $y=2$.
- iv. Proses perhitungan tersebut diulang hingga $k=20$.

Contoh 2:

Hitung jumlah bilangan bulat dari 0 sampai dengan 10

Jawab:

- a. Nyatakan penjumlahan tersebut dalam persamaan matematika sebagai berikut:

$$y = 0 + 1 + 2 + \dots + 10 = \sum_{n=0}^{10} n$$

- b. Tuliskan Matlab programnya dalam m-file.

Penjumlahan $\sum_{n=0}^{10} n$ memiliki indeks dari 0 hingga 10. Dikarenakan indeks dalam

Matlab paling kecil adalah 1 maka penjumlahan tersebut harus diubah sehingga n dalam penjumlahan tersebut dimulai dari 1, sebagaimana ditunjukkan dalam persamaan berikut:

$$y = \sum_{n=0}^{10} n = \sum_{n=0+1}^{10+1} (n-1) = \sum_{n=1}^{11} (n-1).$$

Sehinga dapat dibuat program Matlab dalam m-file sebagai berikut:

```
y=0;           % initial value
for n=1:11
    y=y+(n-1);
end
```

- c. Program untuk penjumlahan diatas dapat juga dibuat dengan mendefinisikan suatu variable baru, misalkan x sebagai berikut:

$$y = 0 + 1 + 2 + \dots + 10 = \sum_{k=1}^{11} x_k$$

$x(1)$ $x(2)$ $x(3)$ $x(11)$

dimana $x(k) = k - 1$. Sehingga dapat dibuat program sebagai berikut:

```
y=0;           % initial value
for k=1:11
    x(k)=k-1;
    y=y+x(k);
end
```

Contoh 3:

Berikut diberikan data temperature di suatu kota untuk rentang waktu 15 hari. Hitunglah rata-rata temperature dalam selang waktu tersebut.

Date	11	12	13	14	15	16	17	18	19	20
Temperature	32	35	37	35	31	35	29	30	26	26

Jawab:

Penghitungan nilai temperature rata-rata dilakukan dengan langkah-berikut:

- a. Nyatakan temperature dalam row vector:
 - b. Hitung jumlah data
 - c. Hitung total temperature dengan loop for
 - d. Hitung nilai rata-rata = nilai total / jumlah data:
- dan dituliskan dalam m-file sebagai berikut:

```
% a. Nyatakan temperature dalam row vector
T = [32, 35, 37, 35, 31, 35, 29, 30, 26, 26];

% b. Hitung jumlah data
[Nr,Nc] = size(T);
N=Nc;

% c. Hitung total temperature (Tt)
Tt = 0;
for k=1:N
    Tt = Tt + T(k);
end

% d. Hitung temperature rata-rata (Tr)
Tr = Tt/N
```