

MODUL I

PENGENALAN MATLAB

1. Apa Matlab itu?

Matlab merupakan bahasa pemrograman dengan kemampuan tinggi dalam bidang komputasi. Matlab memiliki kemampuan mengintegrasikan komputasi, visualisasi, dan pemrograman. Oleh karenanya, matlab banyak digunakan dalam bidang riset-riset yang memerlukan komputasi numerik yang kompleks. Penggunaan Matlab meliputi bidang-bidang:

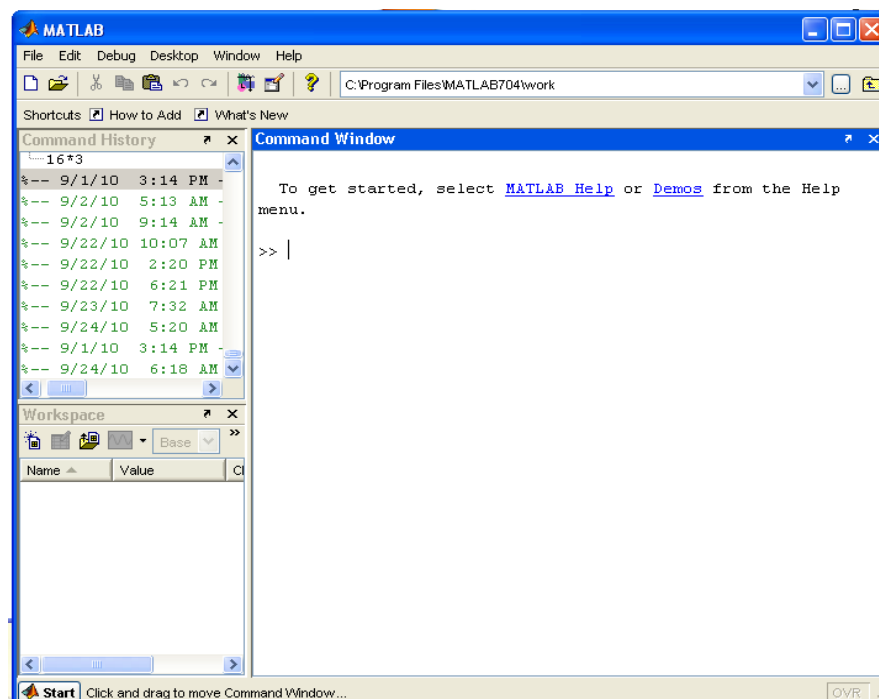
- Matematika dan Komputasi
- Pembentukan Algorithm
- Akusisi Data
- Pemodelan, simulasi, dan pembuatan prototype
- Analisa data, explorasi, dan visualisasi
- Grafik Keilmuan dan bidang Rekayasa

Matlab merupakan kepanjangan dari **Matrix Laboratory**. Sesuai dengan namanya, struktur data yang terdapat dalam Matlab menggunakan matriks atau array berdimensi dua (double). Oleh karenanya penguasaan teori matriks mutlak diperlukan bagi pengguna pemula Matlab agar mudah dalam mempelajari dan memahami operasi-operasi yang ada di Matlab.

Kita dapat belajar Matlab melalui berbagai macam cara seperti dari buku maupun internet. Banyak situs di internet yang menyediakan tutorial tentang matlab. Seperti tutorial dasar, toolboxes, simulink, dan sebagainya. Kita dapat menggunakan situs <http://www.mathworks.com>. Untuk memperoleh informasi dan pengetahuan terkini tentang matlab.

2. Memulai MATLAB

Perhatikan Dekstop pada layar monitor PC, mulailah MATLAB dengan melakukan *double-clicking* pada *shortcut icon* MATLAB. Selanjutnya akan muncul tampilan seperti pada Gambar berikut ini.



Gambar 1: Jendela Utama Matlab

Pada gambar diatas, terlihat beberapa jendela yang merupakan bagian penting di dalam Matab, antara lain:

a. Jendela perintah (Command Window)

Pada command window, semua perintah matlab dituliskan dan dieksekusi. Kita dapat menuliskan perintah perhitungan sederhana, memanggil fungsi, mencari informasi tentang sebuah fungsi dengan aturan penulisannya (help), demo program, dan sebagainya.

Setiap penulisan perintah selalu diawali dengan prompt '>>'. Misal, mencari nilai sin 75°, maka pada command window kita dapat mengetikkan:

```
>> sin(75)
ans =
    -0.38778
```

b. Jendela ruang kerja (Workspace)

Jendela ini berisi informasi pemakaian variabel di dalam memori matlab. Misalkan kita akan menjumlahkan dua buah bilangan, maka pada command window kita dapat mengetikkan:

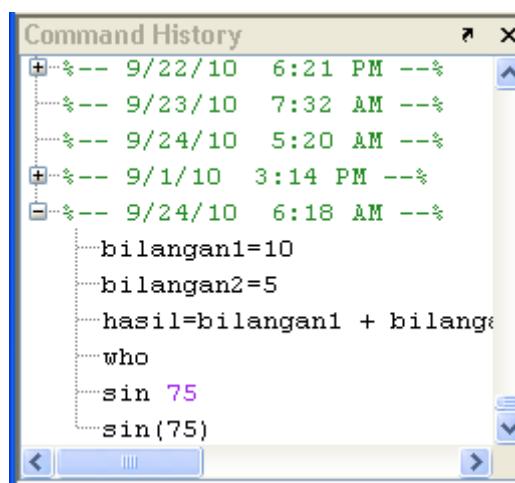
```
>> bilangan1 = 10
bilangan1=10
>> bilangan2 = 5
bilangan1=10
>> hasil= bilangan1 + bilangan2
hasil=15
```

Untuk melihat variabel yang aktif saat ini, kita dapat menggunakan perintah who.

```
>> who
Your variables are:
bilangan1    bilangan2    hasil
```

c. Jendela histori (Command History)

Jendela ini berisi informasi tentang perintah yang pernah dituliskan sebelumnya. Kita dapat mengambil kembali perintah dengan menekan tombol panah ke atas atau mengklik perintah pada jendela histori, kemudian melakukan copy-paste ke command window.



Gambar: Command History

3. Variabel dan Operator

3.1 Variabel

Seperti bahasa pemrograman lainnya, matlab pun memiliki variabel, tetapi dalam penulisannya, variabel di dalam matlab tidak perlu dideklarasikan, karena matlab mampu mengenali tipe data dari variabel dari isi variabel tersebut. Aturan penulisan variabel pada matlab sama dengan aturan pada bahasa pemrograman lainnya, yaitu bersifat case sensitive, diawali dengan huruf dan selanjutnya boleh menggunakan gabungan huruf-angka atau tanda garis bawah. Matlab mampu mengenali sampai 31 karakter pertama, selanjutnya diabaikan.

Contoh:

```
>> var1=6.7
```

```
var1 =
```

```
6.7
```

```
>> var_2=[2 3 4]
```

```
Var_2 =
```

```
2 3 4
```

Semua tipe data di matlab memiliki bentuk yang sama, yaitu array. Array minimal berukuran 0x0 dan dapat bertambah menjadi array n x m dimensi dengan sebarang ukuran. Matlab mempunyai beberapa tipe data dasar (atau class), yaitu: logical, char, numeric, cell, structure, java classes, function handles.

3.2 Operator

Di dalam matlab, operator diklasifikasikan menjadi tiga bagian, yaitu:

a) Operator Arimatika

Operator aritmatika digunakan untuk mengerjakan komputasi numeric.

Operator	Arti
+	Penjumlahan
-	Pengurangan
*	Perkalian (aturan matriks)
.*	Perkalian masing-masing elemen yang bersesuaian (aturan array)
/	Pembagian kanan (matriks)
./	Pembagian kanan (array)
\	Pembagian kiri (matriks)
.\	Pembagian kiri (array)
^	Perpangkatan (matriks)
.^	Perpangkatan (array)
:	langkah

b) Operator Relasional

Operator relasional digunakan untuk membandingkan operand-operand secara kuantitatif

Operator	Arti
==	Sama dengan
~=	Tidak sama dengan
<	Kurang dari
>	Lebih dari
<=	Kurang dari sama dengan
>=	Lebih dari sama dengan

c) Operator Logika

Operator	Arti
&	Akan menghasilkan nilai 1 jika kedua elemen yang bersesuaian memiliki nilai true dan 0 untuk lainnya
	Akan bernilai 1 jika salah satu elemennya true
~	Komplen dari elemen yang diinputkan
xor	Akan bernilai 1 jika salah satu dari kedua elemen memiliki nilai berbeda dan bernilai nol jika sama

4. Matriks

Matlab menggunakan matriks sebagai dasar komputasinya, maka pengetahuan tentang matriks sangatlah diperlukan bagi pengguna matlab. Secara garis besar matlab membagi matriks menjadi dua bagian.

4.1 Matriks Khusus

Matriks khusus merupakan matriks yang didefinisikan oleh matlab, sehingga kita tinggal menggunakannya. Contoh: matriks nol, matriks diagonal, matriks identitas, dan sebagainya.

a) Matriks nol

Matriks yang elemennya bilangan nol

Bentuk umum:

```
>> zeros(n,m)
```

Contoh :

```
>> zeros(2,3)
```

ans =

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

b) Matriks satu

Matriks yang elemennya bilangan nol

Bentuk umum:

```
>> ones(n,m)
```

Contoh :

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

ans =

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

c) Matriks identitas

Bentuk umum:

```
>> eye(n)
```

Contoh :

```
>> eye(3)
```

ans =

```
1    0    0
0    1    0
0    0    1
```

d) Matriks bujur sangkar ajaib

Matriks yang memiliki jumlahan yang sama pada tiap baris, kolom maupun diagonalnya

Bentuk umum:

>>magic(n)

Contoh :

>> magic(4)

ans =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

e) Matriks acak

Matriks isinya bernilai acak berdasarkan distribusi statistic.

Bentuk umum:

>>rand(n,m)

Contoh :

>> rand(4,4)

ans =

0.61543	0.17627	0.41027	0.81317
0.79194	0.40571	0.89365	0.0098613
0.92181	0.93547	0.057891	0.13889
0.73821	0.9169	0.35287	0.20277

4.2 Matriks yang didefinisikan oleh pengguna

Selain bentuk khusus, matlab juga menyediakan bentuk matriks yang disefinisikan oleh pengguna, yaitu menggunakan tanda kurung siku.

Contoh:

>> A=[1 2 3; 3 4 5]

A =

1	2	3
3	4	5

Tanda semicolon ';' digunakan untuk memisahkan baris satu dengan yang lain.

Perhatian !

Coba Anda bangkitkan dua buah matriks dengan sembarang ukuran, kemudian lakukan operasi aritmatika dan juga cari tahu determinan dan invers dari suatu matriks. Perhatikan cara kerjanya!

5. Pemrograman M-File

M-file merupakan sederetan perintah matlab yang dituliskan secara berurutan sebagai sebuah file. Nama file yang tersimpan akan memiliki ekstensi .m yang menandakan bahwa file yang dibuat adalah file matlab. M-file dapat ditulis sebagai sebuah script atau dapat pula ditulis sebagai sebuah fungsi yang menerima argument atau masukan yang menghasilkan output.

Contoh script sederhana dari matlab:

```
1      %menghitung akar dari
2      %persamaan kuadrat y=ax^2 + bx + c
3
4 -    clc %untuk membersihkan tulisan di command window
5 -    clear % menghapus variabel sebelumnya
6
7      %masukkan input/kontanta
8 -    a=input('masukkan kontanta a=')
9 -    b=input('masukkan kontanta b=')
10 -   c=input('masukkan kontanta c=')
11
12     %perhitungan akar
13 -   x1=(-b + sqrt(b^2-4*a*c))/(2*a)
14 -   x1=(-b - sqrt(b^2-4*a*c))/(2*a)
15
```

Jika di running, hasilnya akan terlihat di command window seperti berikut:

```
Command Window
masukkan kontanta a=1
a =
    1
masukkan kontanta b=4
b =
    4
masukkan kontanta c=-5
c =
   -5
x1 =
    1
x2 =
   -5
```

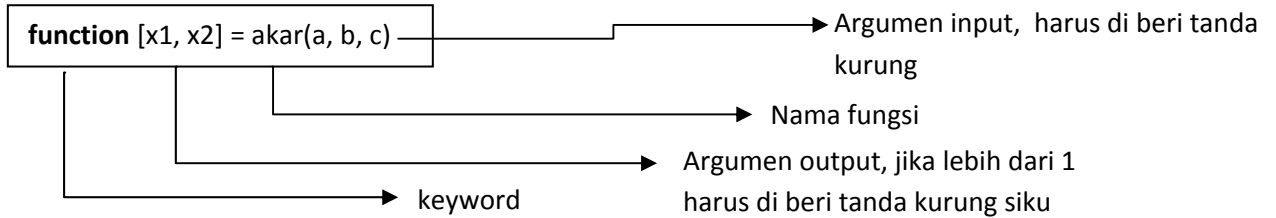
Perhatian !

Sekarang coba anda membuat script sederhana untuk menghitung

- Luas dan keliling lingkaran
- Mean dari data nilai berikut: 46, 60, 87, 50, 86, 70, 90, 100

6. Fungsi

Fungsi adalah m-file yang menerima argument input dan menghasilkan argument output. Fungsi dapat dipanggil langsung dari command window atau dari suatu m-file yang berbeda. Aturan penulisan fungsi adalah sebagai berikut



Fungsi di dalam matlab jika disimpan secara default akan tersimpan dengan nama yang sama dengan nama fungsinya.

Contoh fungsi:

```
1 function [x1, x2]=akar(a, b, c)
2 - x1=(-b + sqrt(b^2-4*a*c))/(2*a)
3 - x2=(-b - sqrt(b^2-4*a*c))/(2*a)
4
```

Jika kita akan menggunakan fungsi tersebut, maka pada command window kita tuliskan:

```
>> akar(1, 8, -2)
```

Maka akan menghasilkan

```
x1 =
    1.1231
x2 =
   -7.1231
```

Perhatian !
Sekarang coba anda membuat fungsi dari praktik sebelumnya

7. Grafik

Matlab mempunyai bermacam-macam fungsi untuk menampilkan grafik, dimana setiap fungsi memiliki perbedaan dalam menskalakan garis sumbu. Setiap menerima inputan dalam bentuk vector atau matriks, matlab akan menskalakan secara otomatis.

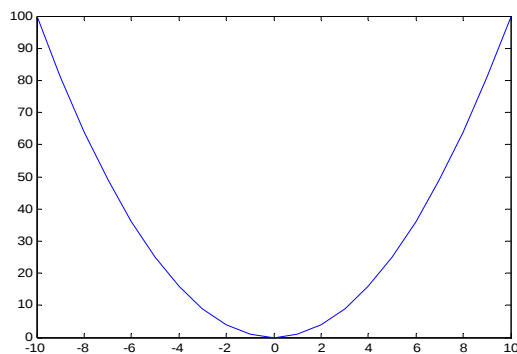
7.1 plot

Fungsi plot digunakan untuk menggambar grafik 2D dengan skala linear pada kedua sumbunya.

Contoh:

```
>> x=-10:10;  
>> y=x.^2;  
>> plot(x,y)
```

Hasilnya akan tampak sebagai berikut:



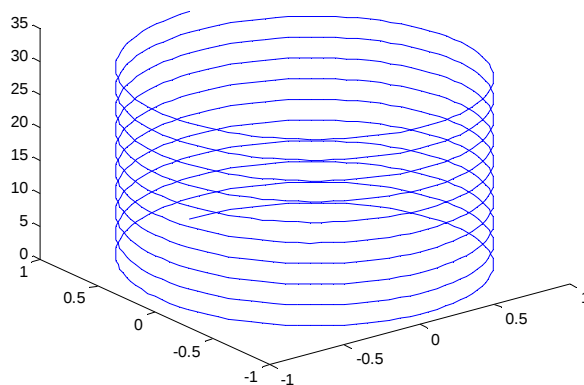
7.2 plot3

Fungsi plot3 digunakan untuk menampilkan grafik 3 dimensi. Plot3 memerlukan 3 argumen dengan bentuk plot3(x,y,z), dimana x, y, z merupakan 3 bagian vector yang sama panjang.

Contoh:

```
>> t=0:pi/100:10*pi;  
>> plot3(sin(2*t), cos(2*t), t)
```

Hasilnya akan tampak sebagai berikut:



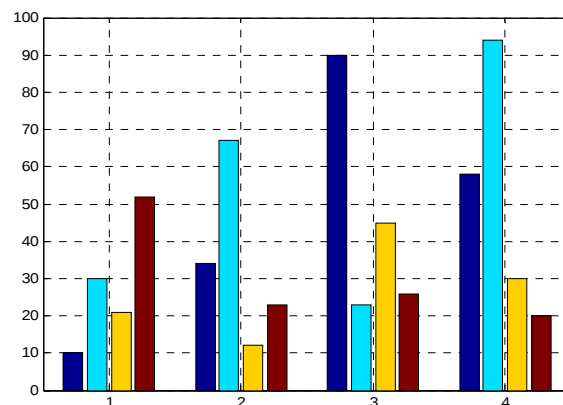
7.3 Bar

Fungsi bar digunakan untuk menampilkan data yang berbentuk vector maupun matriks. grafik bar digunakan untuk menampilkan sekumpulan data selama kurun waktu tertentu dan cocok untuk menampilkan data dalam bentuk diskrit.

Contoh:

```
>> t=[10 30 21 52; 34 67 12 23; 90, 23, 45, 26; 58 94 30 20];  
>> bar(t)  
>> grid on
```

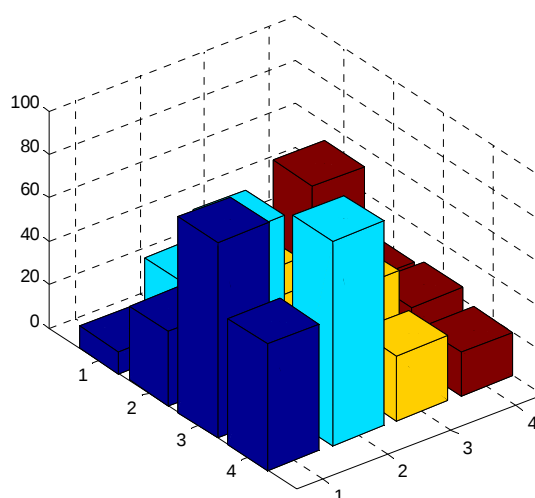
Hasilnya akan tampak sebagai berikut:



Matlab juga menyediakan dalam bentuk 3 dimensi, yaitu bar3. Misal grafik diatas disajikan dalam bar3, maka kita ketik perintah berikut:

```
>> bar3
```

Hasilnya akan tampak sebagai berikut:



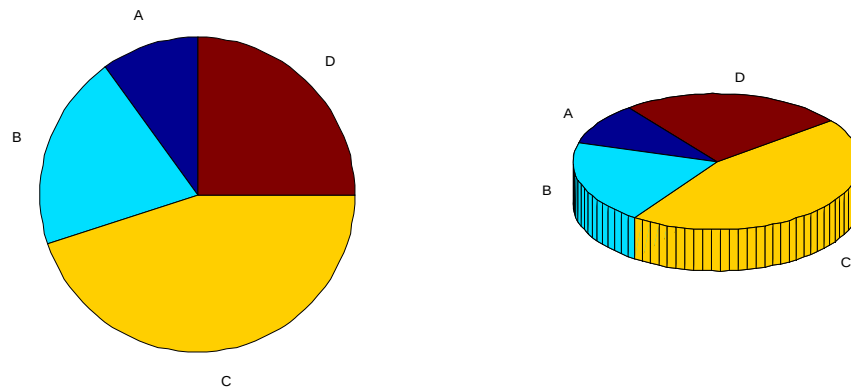
7.3 Pie

Fungsi pie digunakan untuk menampilkan data secara prosentase, dimana setiap elemen data akan dibandingkan dengan penjumlahan seluruh data yang ada. Grafik pie dapat disajikan dalam bentuk 2 dimensi maupun 3 dimensi.

Contoh:

```
>> x=[10 20 45 25];  
>> pie(x, {'A', 'B', 'C', 'D'}) → pie 2 dimensi  
>> pie3(x, {'A', 'B', 'C', 'D'}) → pie 3 dimensi
```

Perintah diatas akan menghasilkan gambar sebagai berikut:

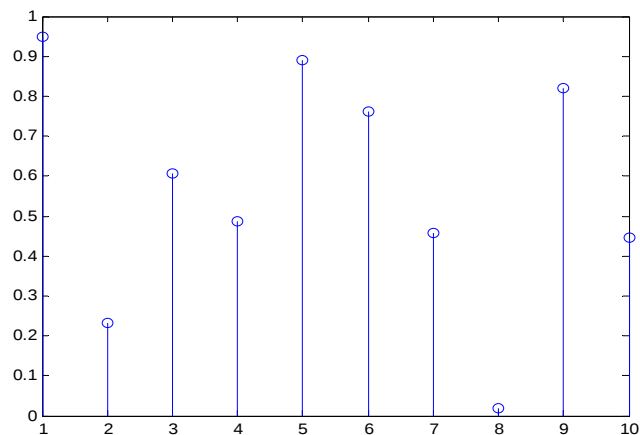


7.3 Stem

Fungsi stem cocok digunakan untuk menampilkan data dalam bentuk diskrit.

Contoh:

```
>> x=1:10;  
>> y=rand(1,10);  
>> stem(x,y)
```



Perhatian !

Sekarang coba anda membuat grafik dari masing-masing fungsi berikut:

a) $y = \sin 2x, -2\pi \leq x \leq 2\pi$

b) $y = \sin (2x + \pi/2), -2\pi \leq x \leq 2\pi$

c) $y = \sin (2x - \pi/2), -2\pi \leq x \leq 2\pi$

d) $y = x^3 - 3x^2, -10 \leq x \leq 10$

e) $y = \{-3, -2, 4, 3, 3, 1, 2, 5\}$

Kemudian tampilkan grafik a), b) dan c) dalam 1 figure

8. Statement Kontrol

Sama seperti bahasa pemrograman yang lain, matlab juga memiliki statemen kontrol. Berikut adalah beberapa pembahasan statemen kontrol yang ada di Matlab:

8.1 if, else, dan elseif

if merupakan statemen control yang digunakan untuk mengevaluasi ekspresi logika dan mengeksekusi kelompok statemen yang didasarkan pada nilai ekspresi. Penulisan kontrol if else adalah sebagai berikut:

```
if <ekspresi logika>
    statemen-statemen
else if <ekspresi logika>
    statemen-statemen
else <ekspresi logika>
    statemen-statemen
end
```

contoh:

```
1 - nilai=input('masukkan nilai:')
2 - if(nilai>60)
3 -     disp('Anda Lulus')
4 - else
5 -     disp('Anda tidak lulus')
6 - end
```

8.2 while

while digunakan untuk melakukan proses perulangan selama kondisi ekspresi terpenuhi. Begitu kondisi sudah tidak terpenuhi lagi, maka proses perulangan akan langsung dihentikan.

Bentuk umum dari while adalah:

```
While <ekspresi>
    Statemen-statemen
end
```

contoh:

```
1 - fak=1;
2 - i=1;
3 - while(i<5)
4 -     fak=fak*i;
5 -     i=i+1;
6 - end
7 - disp('faktorial dari 5 adalah:')
8 - disp(fak)
```

8.3 for

for digunakan untuk melakukan proses perulangan selama kondisi ekspresi terpenuhi. Perbedaannya dengan while adalah pada for jumlah perulangan dapat diketahui, sedangkan pada while bergantung pada nilai ekspresi.

Bentuk umum dari for adalah:

```
for indeks=nilai awal:step:nilai akhir
    Statemen-statemen
end
```

contoh:

```
1 - clc
2 - temp=0;
3 - for i=1:10
4 -     temp=temp+i;
5 - end
6 - sprintf('jumlah 10 bilangan asli pertama adalah %d', temp)
```

8.4 continu dan break

Statemen continu dan break digunakan pada statemen kontrol for atau while. Fungsi continu adalah melanjutkan ke iterasi berikutnya tanpa menjalankan statemen yang ada di bawah kondisi continu. Sedangkan break, digunakan untuk menghentikan proses iterasi tanpa melanjutkan perulangan.

Contoh:

```
1 - clc
2 - for i=1:10
3 -     if(i==5 || i==7)
4 -         continue
5 -     end
6 -     disp(i);
7 - end
```

```
1 - clc
2 - for i=1:10
3 -     if(i==5)
4 -         break
5 -     end
6 -     disp(i);
7 - end
```

Perhatian !

Sekarang coba anda membuat sembarang aplikasi dari masing-masing kontrol di atas

TUGAS PRAKTIKUM 1.

Laporkan semua perintah yang terdapat dalam kotak.