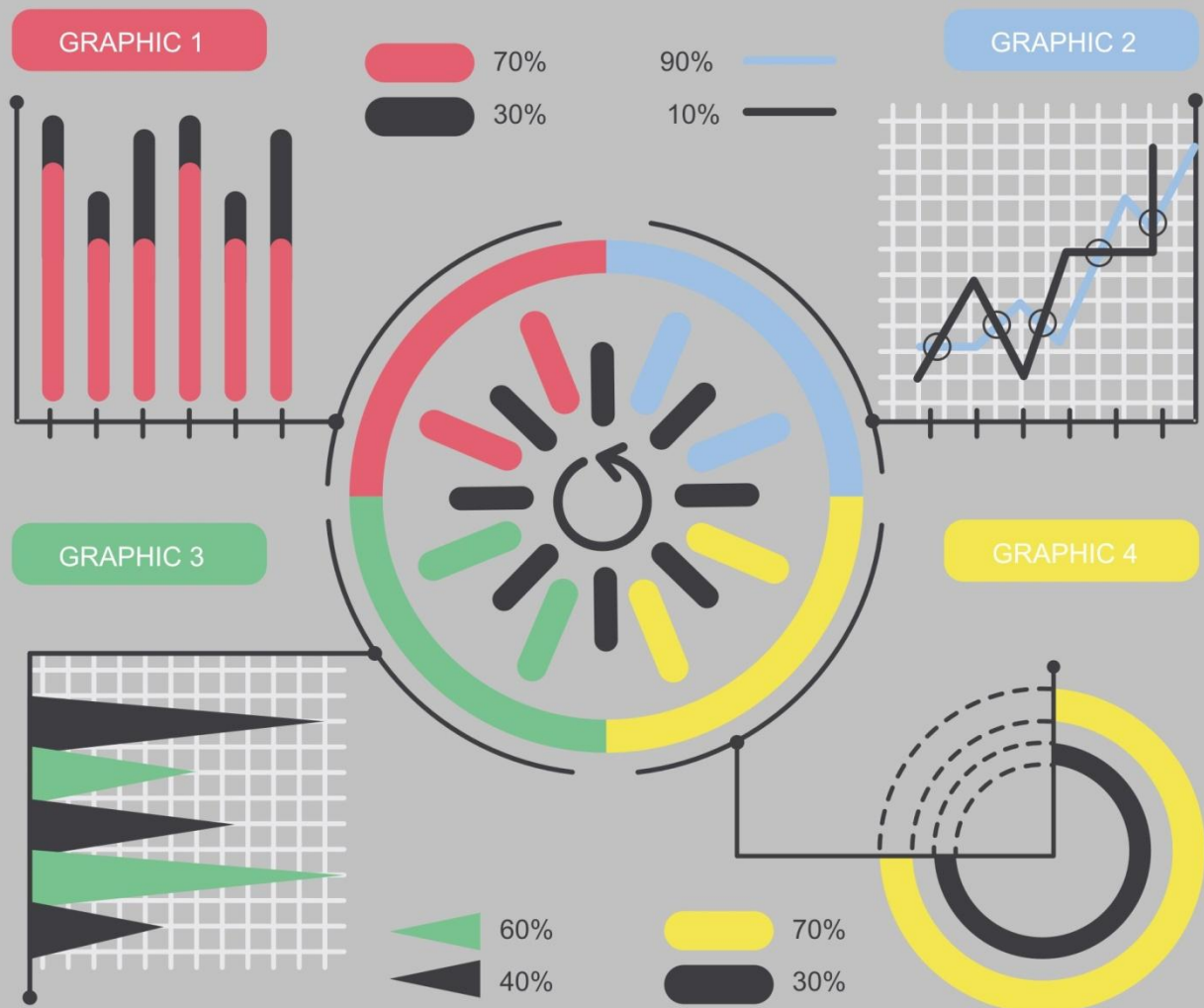


MATLAB

Untuk Pemula



Raizal Dzil Wafa

KATA PENGANTAR

Buku ini dibuat untuk memudahkan siapa saja yang ingin belajar MATLAB terutama bagi yang baru mengenal MATLAB. Buku ini sangat cocok untuk pemula terutama untuk pelajar yang sedang menempuh pendidikan di bidang teknik maupun penelitian. Materi yang ada di buku ini ditulis dengan bahasa yang ringan sehingga mudah untuk dipahami pembaca.

Materi dalam buku ini ditulis secara terurut yang dimulai dari pengenalan MATLAB sampai penulisan matematika yang kompleks dan pembuatan grafik. Semua materi yang ada dalam buku ini diujikan menggunakan MATLAB Versi 2010 yang bisa di download di internet.

Saya mengucapkan banyak terima kasih kepada semua pihak yang telah membantu saya dalam menyelesaikan ebook ini. Untuk kritik dan saran bisa Anda kirim melalui email raizal.st@gmail.com atau raizalst4@gmail.com.

Penulis

Raizal Dzil Wafa M.

COPYRIGHT

- ✚ Dilarang memperjualbelikan ebook ini tanpa seizin penulis.
- ✚ Dilarang mengcopy, menggandakan, atau pun mengedit ebook ini tanpa seizin penulis.

DAFTAR ISI

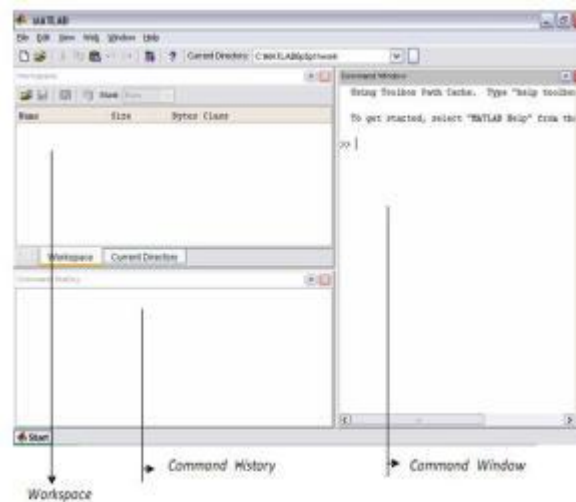
COVER	i
KATA PENGANTAR.....	ii
COPYRIGHT	iii
DAFTAR ISI	iv
Pengenalan MATLAB	1
Operasi Aritmatika.....	4
Variabel.....	9
Fungsi Matematika	17
Bilangan Kompleks	21
Vektor.....	26
Matrik	32
Grafik	36
Daftar Pustaka	38

PEMBAHASAN 1 : PENGENALAN MATLAB

MATLAB (*Matrix Laboratory*) adalah sebuah program untuk analisis dan komputasi numerik yang merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks. Bidang penggunaan MATLAB: Statistik, Matematika, Komputerisasi, Penelitian dan Pemrograman. Bagian-bagian Window pada MATLAB:

a. Command Windows

Windows ini muncul pertama kali ketika kita menjalankan program MATLAB. Command windows digunakan untuk menjalankan perintah-perintah MATLAB, memanggil tool MATLAB seperti *editor*, *fasilitas help*, *model simulink*, dan lain-lain. Ciri dari windows ini adalah adanya *prompt* (tanda lebih besar) yang menyatakan Matlab siap menerima perintah. Perintah tersebut dapat berupa fungsi-fungsi bawaan (*toolbox*) MATLAB itu sendiri.



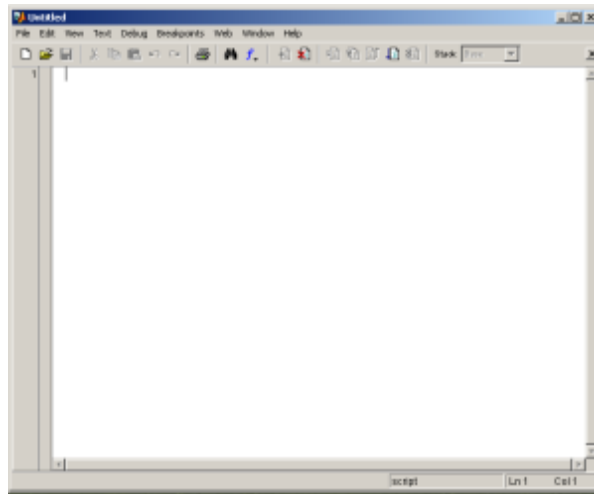
Gambar 1. Command Windows

Workspace: Menampilkan semua variabel yang pernah dibuat meliputi nama variabel, ukuran, jumlah byte dan class.

Command History : Menampilkan perintah-perintah yang telah diketikkan pada command Window.

b. Editor Windows

Windows ini merupakan tool yang disediakan oleh MATLAB yang berfungsi sebagai *editor script* MATLAB (listing perintah-perintah yang harus dilakukan oleh MATLAB). Ada dua cara untuk membuka editor ini, yaitu: Klik : *File*, lalu *New* dan kemudian *M-File*. Ketikkan “edit” pada command windows.

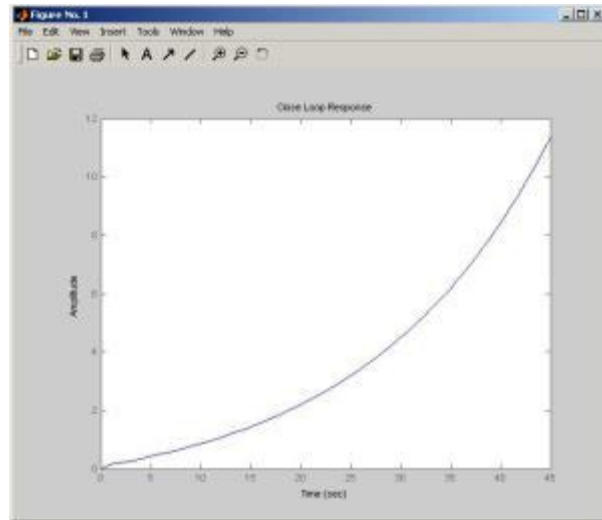


Gambar 2. Editor Windows

Secara formal suatu *script* merupakan suatu file eksternal yang berisi tulisan perintah MATLAB. Tetapi *script* tersebut bukan merupakan suatu fungsi. Ketika anda menjalankan suatu *script*, perintah di dalamnya dieksekusi seperti ketika dimasukkan langsung pada MATLAB melalui keyboard. M-file selain dipakai sebagai penamaan file juga bisa dipakai untuk menamakan fungsi, sehingga fungsi fungsi yang kita buat di jendela editor bisa di simpan dengan ekstensi .m sama dengan file yang kita panggil di jendela editor. Saat kita menggunakan fungsi MATLAB seperti *inv*, *abs*, *cos*, *sin* dan *sqrt*, MATLAB menerima variabel berdasarkan variabel yang kita berikan.

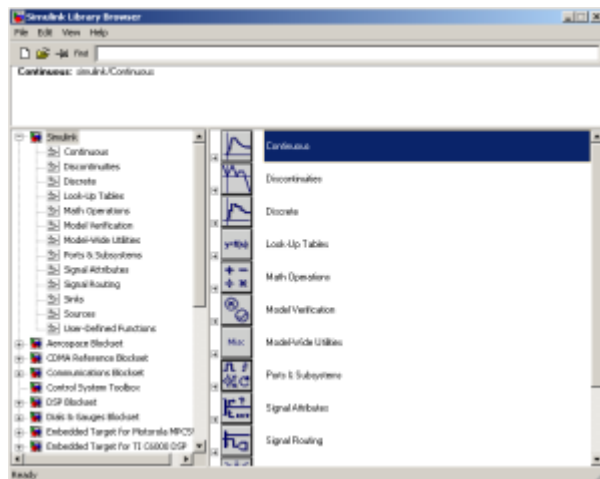
c. Figure Windows

Windows ini merupakan hasil visualisasi dari *script* MATLAB. MATLAB memberikan kemudahan bagi programmer untuk mengedit windows ini sekaligus memberikan program khusus untuk itu, sehingga selain berfungsi sebagai visualisasi output yang berupa grafik juga sekaligus menjadi media input yang interaktif.



Gambar 3. Figure Windows

Simulink windows: Windows ini umumnya digunakan untuk mensimulasikan sistem kendali berdasarkan blok diagram yang telah diketahui. Untuk mengoperasikannya ketik “*simulink*” pada command windows.



Gambar 4. Simulink Windows

PEMBAHASAN 2 : OPERASI ARITMATIKA

Pada pembahasan kali ini kita akan belajar mengenai operasi matematika sederhana seperti penjumlahan, pengurangan, perkalian, dan pembagian. Untuk melakukan operasi matematika sederhana kita dapat menggunakan simbol (+) untuk penjumlahan, simbol (-) untuk pengurangan, simbol (*) untuk perkalian, dan simbol (/) untuk pembagian. Untuk penjelasannya ikuti contoh dibawah ini:

```
>> %Operasi Matematika Sederhana
>> %PENJUMLAHAN
>> 124 + 97
```

```
ans =
```

```
221
```

```
>> %PENGURANGAN
>> 178 - 98
```

```
ans =
```

```
80
```

```
>> %PEMBAGIAN
>> 765 / 9
```

```
ans =
```

```
85
```

```
>> %PERKALIAN
>> 86 * 54
```

```
ans =
```

```
4644
```

Pada bagian selanjutnya kita akan mencoba melakukan operasi pemangkatan dengan menggunakan simbol (^) pada MATLAB. Untuk penjelasannya ikuti contoh dibawah ini:


```
>> %Operasi Pangkat pada MATLAB  
>> 2^7
```

```
ans =
```

```
128
```

```
>> 2^-6
```

```
ans =
```

```
0.0156
```

```
>> -12^4
```

```
ans =
```

```
-20736
```

```
>> -9^9
```

```
ans =
```

```
-387420489
```

```
>> |
```

Dalam MATLAB penulisan *pi* (π) dan *epsilon* (ϵ) dapat ditulis secara langsung dengan perintah “*pi*” dan juga “*eps*”. Untuk penjelasannya ikuti contoh dibawah ini:

```
>> %Penulisan  $\pi$  dan  $\epsilon$  pada MATLAB
```

```
>> %Penulisan  $\pi$ 
```

```
>> pi
```

```
ans =
```

```
3.1416
```

```
>> %Penulisan  $\epsilon$ 
```

```
>> eps
```

```
ans =
```

```
2.2204e-016
```

```
>> |
```

Hal yang perlu diperhatikan dalam menuliskan operasi matematika pada MATLAB adalah mengenai penggunaan tanda kurung () dalam menuliskan operasi matematika. Seperti prinsip matematika pada umumnya, setiap angka yang diberi tanda kurung akan dikerjakan terlebih dahulu, sama halnya pada MATLAB. Untuk penjelasannya ikuti contoh dibawah ini:

```
>> %Penulisan Tanda Kurung pada operasi MATLAB
>> %Jika ada soal 4*4-2 dan 4*(4-2) hasilnya pasti akan berbeda
>> 4*4-2

ans =

    14

>> 4*(4-2)

ans =

     8

>> |
```

Dalam MATLAB kita dapat menggunakan simbol operasi matematika dengan menggunakan fasilitas *Symbolic Math Toolbox*. Fungsi sederhana dari fasilitas *Symbolic Math Toolbox* ini adalah untuk menyimpan nilai pecahan seperti $1/2$. Karena jika kita menulis langsung $1/2$ maka hasil yang akan muncul adalah 0.5. Jika kita menginginkan hasil dalam bentuk pecahan kita bisa menggunakan perintah *sym*. Untuk penjelasannya ikuti contoh dibawah ini:

```
>> %Penggunaan Fasilitas Symbolic Arithmetic Operation pada MATLAB
>> %Contoh penulisan 1/2 dan 22/7 tanpa fasilitas Symbolic
>> 1/2

ans =

    0.5000

>> 22/7

ans =

    3.1429
```

```

>> %Contoh penulisan 1/2 dan 22/7 menggunakan fasilitas Symbolic
>> sym (1/2)

ans =

1/2

>> sym(22/7)

ans =

22/7

>> |

```

Salah satu kelebihan dari fasilitas *symbolic* ini adalah kita dapat menggunakan kembali variabel yang sudah kita masukan. Karena jika kita menggunakan fasilitas *symbolic* ini maka secara otomatis akan tersimpan dalam *history* MATLAB. Untuk lebih jelasnya ikuti contoh berikut:

```

>> %Kelebihan fasilitas Symbolic pada MATLAB
>> sym(1/2)

ans =

1/2

>> sym(7/2)

ans =

7/2

>> a = sym (1/2 + 8/2)

a =

9/2

```

SOAL LATIHAN:

Selesaikan soal ini menggunakan MATLAB!

1. $132 + 927 =$

2. $267 - 453 =$

3. $129 \times 195 =$

4. $432 \times 222 =$

5. $760 / 380 =$

6. $973 / 186 =$

7. $760 + 89 / 380 =$

8. $12^8 =$

9. $54^9 =$

10. $75 * (-9 + 5) =$

11. $(43 - 4) + 76 =$

12. $\frac{2\pi}{9} =$

13. $\frac{8\pi}{6} =$

14. Kerjakan dalam bentuk pecahan $\frac{22}{8} + \frac{32}{7} =$

15. Kerjakan dalam bentuk pecahan $\frac{17}{9} - \frac{12}{2} =$

<+++++Selamat Mengerjakan++++>

PEMBAHASAN 3 : VARIABEL

Pada pembahasan kali ini kita akan belajar mengenai penggunaan variabel pada MATLAB. Dalam melakukan operasi sebenarnya hasil dari operasi yang kita lakukan akan tersimpan dalam variabel yang bernama *ans*. Variabel *ans* sendiri merupakan singkatan dari kata answer. Variabel digunakan untuk menyimpan hasil perhitungan yang kita lakukan. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Contoh penggunaan variabel ans pada MATLAB  
>> 6 + 7
```

```
ans =  
  
13
```

```
>> 4 * 76
```

```
ans =  
  
304
```

```
>> 43 - 9
```

```
ans =  
  
34
```

Dalam MATLAB kita diizinkan untuk membuat variabel kita sendiri. Variabel ini nantinya akan menampung sebuah nilai tergantung dari input yang kita masukan pada MATLAB. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Contoh penggunaan variabel sendiri pada MATLAB  
>> x = 2
```

```
x =  
  
2
```

```
>> y=6
```

```
y =  
  
6
```

Salah satu kelebihan jika kita menggunakan variabel sendiri adalah kita dapat memanggil kembali variabel tersebut untuk melakukan operasi matematika lain. Karena variabel yang kita buat akan secara otomatis tersimpan dalam *history* MATLAB. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Kelebihan penggunaan variabel pada MATLAB
>> x=23
```

```
x =
```

```
23
```

```
>> y=31
```

```
y =
```

```
31
```

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

```
z =
```

Hal yang harus diperhatikan adalah MATLAB sangat sensitive terhadap setiap variabel yang kita tulis sehingga kita harus benar-benar teliti dalam menuliskan variabel terutama dalam penggunaan huruf kecil dan huruf capital. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Contoh perbedaan penggunaan variabel dengan huruf kecil dan kapital
>> a=32;
>> A=42;
>> r=12;
>> z= a + r
```

```
z =
```

```
44
```

```
>> y= A + r
```

```
y =
```

```
54
```

```
>> |
```

Dalam MATLAB kita diizinkan untuk menuliskan karakter string untuk menunjukan sebuah variabel. Sebagai contoh sekarang kita akan menggunakan variabel "result" dan "the_answer_is". Untuk lebih lengkapnya ikuti contoh berikut:

```
>> %Penggunaan karakter STRING untuk menunjukan variabel
>> a= 8

a =

     8

>> b= 19

b =

    19

>> result= a*b

result =

    152

>> the_answer_is= a * b/5

the_answer_is =

    30.4000

>> |
```

Untuk memudahkan setiap perintah yang kita masukan kita dapat menggunakan komentar pada setiap perintah yang kita masukan. Komentar ini berfungsi sebagai keterangan tambahan yang bisa kita masukan. Komentar ini tidak akan mempengaruhi terhadap perhitungan yang akan kita masukan. Untuk menuliskan komentar kita dapat menuliskan simbol persen (%) yang diikuti dengan komentar yang kita masukan. Untuk lebih jelasnya ikuti contoh berikut.

```
>> %Penggunaan komentar pada MATLAB
>> a= 4; %a merupakan inputan pertama
>> b= 7; %b merupakan inputan kedua
>> c= a + b %c merupakan hasil penjumlahan a + b

c =

    11
```

Untuk menghindari kesalahan dalam penulisan variabel, kita perlu mengetahui persyaratan dalam menuliskan sebuah variabel. Berikut ini ketentuan dalam menuliskan variabel pada MATLAB:

- ✓ MATLAB sangat sensitive terhadap penulisan huruf kecil dan kapital.

```
>> %Contoh perbedaan penggunaan variabel dengan huruf kecil dan kapital
>> a=32;
>> A=42;
>> r=12;
>> z= a + r

z =

    44

>> y= A + r

y =

    54

>> |
```

- ✓ Semua variabel yang akan dimasukan harus diawali dengan huruf (tidak boleh angka). Perhatikan contoh berikut:

```
>> %Penulisan angka dalam variabel
>> 2c=a
??? 2c=a
|
Error: Unexpected MATLAB expression.

>> |
```

- ✓ Tidak diizinkan untuk menggunakan simbol dalam menuliskan variabel.

```
>> %Penulisan simbol pada variabel
>> +a= 4
??? +a= 4
|
Error: The expression to the left of the equals sign is not a valid target for
an assignment.

>> |
```


- ✓ Underscore (_) bisa digunakan dalam menuliskan variabel.

```
>> %Penulisan underscore dalam variabel
>> R_2=4

R_2 =

     4

>> |
```

Dalam MATLAB setiap variabel yang akan kita gunakan harus terdefinisi dengan jelas. Sehingga jika kita memasukan variabel tanpa mendefinisikannya terlebih dahulu maka akan terjadi kesalahan atau error. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Melakukan perhitungan tanpa mendefinisikan variabel
>> a= 8

a =

     8

>> b=9

b =

     9

>> R= a + b / c
??? Undefined function or variable 'c'.

>> |
```

Untuk menghindari kesalahan variabel yang kita gunakan, kita bisa menggunakan perintah “*who*” untuk melihat variabel apa saja yang telah kita masukan atau definisikan kedalam MATLAB. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Menampilkan variabel yang sudah terdefinisi
>> who

Your variables are:

a  b

>> |
```

Untuk menghapus semua variabel yang telah kita definisikan kita dapat menggunakan perintah “*clear*”. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Menghapus variabel yang sudah terdefinisi sebelumnya
>> a= 5

a =

    5

>> b=6

b =

    6

>> who

Your variables are:

a  b

>> clear
>> %Cek kembali
>> who
>> %Hasilnya tidak ada karena sudah terhapus|
```

Dalam MATLAB hasil perhitungan yang dihasilkan akan menampilkan angka dengan format 4 digit dibelakang koma. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Format digit MATLAB
>> 19/4

ans =

    4.7500

>> |
```

Jika kita menginginkan format yang lebih detail kita bisa menggunakan perintah “*format long*”. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Format Long pada MATLAB
>> format long
>> 19/4

ans =

    4.7500000000000000
```

Jika kita ingin mengembalikan format sebelumnya kita bisa menggunakan perintah “*format short*”. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Format SHORT pada MATLAB
>> format short
>> 19/4
```

```
ans =
    4.7500
```

```
>> |
```

Pada pembahasan sebelumnya kita telah mempelajari penggunaan *symbolic* pada MATLAB. Dalam MATLAB kita diizinkan untuk menuliskan variabel yang disertai dengan perintah *symbolic*. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penulisan symbolic pada variabel MATLAB
>> a= 1/2
```

```
a =
    0.5000
```

```
>> b= 3/4
```

```
b =
    0.7500
```

```
>> c = sym (a +b)
```

```
c =
```

```
5/4
```

```
>> |
```

SOAL LATIHAN:

Selesaikan soal ini menggunakan MATLAB!

1. Operasikan $132 + 927 =$
2. Operasikan $123 * 7 / 8$ yang disimpan dalam variabel R!
3. Definikan variabel a,b,c dengan nilai 2;45;90!
4. Jika diketahui $x=5$; dan $y= 7$; kerjakan:
 - a. $12x + 12y =$
 - b. $78x * 6y =$
 - c. $43x - 7y =$
 - d. $43z / 6y =$
5. Operasikan $54/6*9$ dengan menggunakan format 8 angka dibelakang koma!
6. Operasikan $125 / 87$ yang disertai dengan komentar "CONTOH"!
7. Tulis variabel yang sudah terdefinisi pada MATLAB!

<+++++Selamat Mengerjakan++++>

PEMBAHASAN 4 : FUNGSI MATEMATIKA

Pada pembahasan kali ini kita akan belajar mengenai fungsi matematika pada MATLAB. Fungsi matematika sederhana pada MATLAB misalnya adalah fungsi akar kuadrat, fungsi eksponensial, fungsi trigonometri, fungsi faktorial, dll. Fungsi pertama kita awali dengan fungsi akar kuadrat. Untuk menuliskan fungsi akar kuadrat kita bisa menggunakan perintah “sqrt”. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Fungsi akar kuadrat pada MATLAB
>> sqrt(25)

ans =

     5

>> a = sqrt(25+64)

a =

    9.4340

>> |
```

Selain fungsi akar kuadrat, MATLAB juga dapat digunakan untuk menyelesaikan fungsi faktorial. Faktorial sendiri merupakan hasil perkalian antara bilangan bulat positif yang kurang dari atau sama dengan n. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penggunaan fungsi faktorial pada MATLAB
>> factorial(5)

ans =

    120

>> %Hasilnya akan sama dengan
>> 5*4*3*2*1

ans =

    120

>> |
```

Penggunaan fungsi matematika lainnya pada MATLAB adalah penggunaan fungsi trigonometri seperti sin, cos, dan tan. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penggunaan fungsi trigonometri pada MATLAB  
>> sin (30)
```

```
ans =  
  
-0.9880
```

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

```
ans =  
  
0.5000
```

```
>> |
```

Hal yang perlu diperhatikan adalah penggunaan satuan radian pada penggunaan fungsi trigonometri. Sehingga untuk merubahnya menjadi satuan radian kita bisa mengalikannya dengan $\frac{\pi}{180}$. Perhatikan contoh lainnya dibawah ini:

```
>> %Penggunaan fungsi trigonometri pada MATLAB  
>> cos (45)
```

```
ans =  
  
0.5253
```

```
>> cos (45*pi/180)
```

```
ans =  
  
0.7071
```

```
>> |
```

Fungsi matematika lain yang tersedia pada MATLAB adalah fungsi logaritma. Untuk menentukan logaritma kita dapat menuliskan perintah “log”. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penggunaan fungsi logaritma pada MATLAB
>> log (9)
```

```
ans =

    2.1972
```

```
>> log10 (9)
```

```
ans =

    0.9542
```

```
>> |
```

Fungsi lainnya yang dapat kita gunakan adalah fungsi *absolut*. Nilai absolute adalah nilai suatu bilangan riil tanpa tanda plus atau minus. Untuk menuliskannya kita bisa menggunakan perintah “*abs*”. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penggunaan fungsi absolut pada MATLAB
>> abs (-8)
```

```
ans =

     8
```

```
>> |
```

Penggunaan fungsi matematika pada MATLAB dapat digabungkan dengan penggunaan fasilitas *symbolic*. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penggunaan symbolic pada fungsi matematika MATLAB
>> sym(sqrt(40))
```

```
ans =

 2*10^(1/2)
```

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

```
ans =

 1/2
```

SOAL LATIHAN:

Selesaikan soal ini menggunakan MATLAB!

1. Tentukan nilai *square root* dari:
 - a. 45
 - b. 67
 - c. 32
 - d. 64
2. Tentukan factorial dari:
 - a. 12
 - b. 43
 - c. 14
 - d. 25
3. Tentukan fungsi trigonometri berikut dalam satuan radian!
 - a. Sin 30
 - b. Cos 45
 - c. Tan 90
 - d. Sin 180
4. Tentukan logaritma dari:
 - a. log 45
 - b. log 100
 - c. 10 log 255
 - d. 10 log 144

<+++++Selamat Mengerjakan++++>

PEMBAHASAN 5 : BILANGAN KOMPLEKS

Pada pembahasan kali ini kita akan mempelajari penggunaan bilangan kompleks pada MATLAB. Bilangan kompleks sendiri merupakan bilangan yang terdiri dari dua bagian, yaitu *real* dan *imaginer*. Untuk menuliskan bilangan kompleks kita bisa menggunakan dua cara, untuk lebih jelasnya ikuti contoh berikut:

➤ Penulisan cara pertama

```
>> %Penulisan bilangan kompleks pertama
>> a = 3 + j*5

a =

    3.0000 + 5.0000i

>> b = 4 - j*6

b =

    4.0000 - 6.0000i

>>
```

➤ Penulisan cara kedua

```
>> %Penulisan bilangan kompleks kedua
>> complex(2,3)

ans =

    2.0000 + 3.0000i

>> complex(6,-3)

ans =

    6.0000 - 3.0000i

>> |
```

Selain mendefinisikan bilangan kompleks, kita juga bisa melakukan operasi terhadap bilangan kompleks seperti penjumlahan, pengurangan, perkalian, dan pembagian. untuk contoh lebih jelasnya ikuti contoh berikut:

```

>> %Operasi matematika dengan bilangan kompleks
>> a = 3 + j*5;
>> b = 2 - j*2;
>> penjumlahan = a + b

penjumlahan =

    5.0000 + 3.0000i

>> pengurangan = a - b

pengurangan =

    1.0000 + 7.0000i

>> perkalian = a * b

perkalian =

   16.0000 + 4.0000i

>> pembagian = a / b

pembagian =

   -0.5000 + 2.0000i

>> |

```

Selain operasi matematika sederhana kita juga bisa mengoperasikan operasi pemangkatan. Untuk lebih jelasnya ikuti contoh berikut:

```

>> %Operasi pangkat dengan bilangan kompleks
>> a = 2 + j*3

a =

    2.0000 + 3.0000i

>> x = a^2

x =

   -5.0000 +12.0000i

```

Penggunaan operasi lain dengan menggunakan bilangan kompleks pada MATLAB

```
>> %Penggunaan operasi bilangan kompleks lainnya  
>> abs ( 3 + j*5)
```

```
ans =  
  
5.8310
```

```
>> angle ( 3 - j*2)
```

```
ans =  
  
-0.5880
```

```
>> |
```

Selain menuliskan bilangan kompleks pada MATLAB, kita juga bisa mengekstrak bilangan kompleks yang sebelumnya telah kita masukan. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Mengekstrak bilangan kompleks  
>> a = 9 + j*6
```

```
a =  
  
9.0000 + 6.0000i
```

```
>> %Perintah untuk mengekstrak bilangan real  
>> real(a)
```

```
ans =  
  
9
```

```
>> %Perintah untuk mengekstrak bilangan imajiner  
>> imag(a)
```

```
ans =  
  
6
```

```
>> |
```

Pengoperasian trigonometri menggunakan bilangan kompleks. Perhatikan contoh berikut ini:

```
>> %Penggunaan trigonometri dengan bilangan kompleks
>> sin (2-j*4)

ans =

    24.8313 +11.3566i

>> cos (3-j*2)

ans =

   -3.7245 + 0.5118i

>> tan (6+j*5)

ans =

   -0.0000 + 0.9999i

>> |
```

Pengoperasian eksponensial dan logaritma menggunakan bilangan kompleks. Perhatikan contoh berikut ini:

```
>> %Penggunaan eksponensial dan logaritma dengan bilangan kompleks
>> exp (2+j*3)

ans =

   -7.3151 + 1.0427i

>> log (2+j*3)

ans =

    1.2825 + 0.9828i

>> |
```

SOAL LATIHAN:

Selesaikan soal ini menggunakan MATLAB!

1. Tuliskan bilangan berikut dalam bilangan kompleks!
 - a. $3 + j5$;
 - b. $7 - j9$;
 - c. $8 * j3$;
 - d. $12 / j4$;
2. Jika diketahui $x = 4 + j6$ dan $y = 4 - j2$; Tentukan:
 - a. $x + y$;
 - b. $x - y$;
 - c. $x * y$;
 - d. x / y ;
3. Jika diketahui $r = 8 + j4$; Tentukan:
 - a. $\sin(r)$;
 - b. $\cos(r)$;
 - c. $\tan(r)$;
 - d. $\cot(r)$;

<+++++Selamat Mengerjakan++++>

PEMBAHASAN 6 : VEKTOR

Pada pembahasan kali ini kita akan mempelajari vektor pada MATLAB. Pada MATLAB vektor akan disimpan dalam array satu dimensi. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penggunaan vektor pada MATLAB  
>> x = [1 2 3 4 5]
```

```
x =
```

```
1     2     3     4     5
```

```
>> |
```

Selain menginput vektor, kita juga bisa mengekstrak setiap elemen yang tersimpan pada MATLAB. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Mengekstrak elemen pada vektor  
>> x = [1 2 3 4 5]
```

```
x =
```

```
1     2     3     4     5
```

```
>> %Misalkan kita akan mengekstrak elemen ke 3  
>> x(3)
```

```
ans =
```

```
3
```

```
>> %Misalkan kita akan mengekstrak elemen ke 5  
>> x(5)
```

```
ans =
```

```
5
```

```
>> |
```

Selain mengekstrak elemen satu per satu, kita juga bisa mengekstrak sekaligus setiap elemen yang kita inginkan. Untuk lebih jelasnya ikuti contoh berikut:

```

>> %Mengekstrak elemen sekaligus pada vektor
>> x = [1 2 3 4 5]

x =

     1     2     3     4     5

>> %Misalkan kita ingin mengekstrak elemen 1 sampai 3
>> x(1:3)

ans =

     1     2     3

>> |

```

Untuk menghitung panjang elemen kita bisa menggunakan perintah *length*. Untuk lebih jelasnya ikuti contoh berikut:

```

>> %Menghitung panjang elemen
>> x = [2 3 4 5 6]

x =

     2     3     4     5     6

>> length(x)

ans =

     5

```

Untuk menghitung jumlah dari keseluruhan elemen kita bisa menggunakan perintah *sum*.

Untuk lebih jelasnya ikuti contoh berikut:

```

>> %Menghitung jumlah keseluruhan elemen
>> x = [2 3 4 5 6]

x =

     2     3     4     5     6

>> sum(x)

ans =

    20

```

Untuk mencari nilai terkecil kita bisa menggunakan perintah *min*. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Mencari nilai minimum pada elemen vektor
>> x = [12 23 4 35 6]

x =

    12    23     4    35     6

>> min(x)

ans =

     4

>> |
```

Untuk mencari nilai terbesar kita bisa menggunakan perintah *max*. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Mencari nilai max pada elemen vektor
>> x = [12 23 4 35 6]

x =

    12    23     4    35     6

>> max(x)

ans =

    35

>> |
```

Dalam mengoperasikan pemangkatan dalam vektor kita tidak bisa langsung menuliskannya menggunakan cara biasa. Kita harus menambahkan titik (.) pada angka yang akan kita pangkatkan. Untuk lebih jelasnya perhatikan contoh berikut:


```

>> %Penulisan pangkat pada operasi vektor
>> a = [ 2 3 4 5 ]

a =

     2     3     4     5

>> %Jika kita menggunakan cara biasa
>> a^2
??? Error using ==> mpower
Inputs must be a scalar and a square matrix.

>> %Penulisan yang benar pada vektor
>> a.^2

ans =

     4     9    16    25

>> |

```

Untuk mengurutkan elemen pada vektor kita bisa menggunakan perintah *sort*. Untuk lebih jelasnya perhatikan contoh berikut:

```

>> %Pengurutan elemen pada vektor
>> a = [12 3 1 54 23 53 90 6]

a =

    12     3     1    54    23    53    90     6

>> sort(a)

ans =

     1     3     6    12    23    53    54    90

>> |

```

Untuk mencari nilai rata-rata kita bisa menggunakan perintah *mean*.. Untuk lebih jelasnya perhatikan contoh berikut:

```
>> %Mencari mean
>> a = [ 2 2 3 3]
```

```
a =
```

```
     2     2     3     3
```

```
>> mean(a)
```

```
ans =
```

```
     2.5000
```

```
>> |
```

Sedangkan untuk mencari nilai tengah pada elemen vektor kita bisa menggunakan perintah *median*. Untuk lebih jelasnya perhatikan contoh berikut:

```
>> %Mencari median
>> a = [ 3 5 6 3 1 8]
```

```
a =
```

```
     3     5     6     3     1     8
```

```
>> median(a)
```

```
ans =
```

```
     4
```

```
>> |
```

SOAL LATIHAN:

Selesaikan soal ini menggunakan MATLAB!

1. Buat variabel a pada vektor dengan nilai elemen 2,3,12,3,5,53,23,12!
2. Lakukan pengekstrakan vektor a untuk elemen 3 sampai 7!
3. Lakukan pengurutan terhadap vektor a !
4. Cari nilai terkecil dari vektor a !
5. Cari nilai terbesar dari vektor a !
6. Cari nilai tengah dari vektor a !
7. Cari nilai rata-rata dari vektor a !

<+++++Selamat Mengerjakan++++>

PEMBAHASAN 7 : MATRIK

Pada pembahasan kali ini kita akan mempelajari matrik pada MATLAB. Pada MATLAB matrik akan disimpan dalam array dua dimensi yang mempunyai baris dan kolom. Untuk lebih jelasnya perhatikan contoh berikut:

```
>> %Penulisan matrik pada MATLAB  
>> x = [ 2 3 4; 1 2 3; 4 5 3]
```

```
x =
```

2	3	4
1	2	3
4	5	3

```
>> |
```

Selain menginput matrik, kita juga bisa mengekstrak setiap elemen yang tersimpan pada matrik. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Penulisan matrik pada MATLAB  
>> x = [ 2 3 4; 1 2 3; 4 5 3]
```

```
x =
```

2	3	4
1	2	3
4	5	3

```
>> %Pengekstrakan elemen pada matrik  
>> x(3,2) %3 adalah baris dan 2 adalah kolom
```

```
ans =
```

```
5
```

```
>> |
```

Selain menginput dan mengekstrak kita juga bisa mengoperasikan matrik menggunakan operasi matematika sederhana seperti penjumlahan, pengurangan, perkalian, dan pembagian. utnuak lebih jelasnya perhatikan contoh berikut:

```
>> %Operasi matematika pada matrik
>> a = [1 2 3; 2 3 4]
```

```
a =
```

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

```
>> b= [1 1 1; 2 1 1]
```

```
b =
```

```
    1    1    1
    2    1    1
```

```
>> %Penjumlahan
```

```
>> a+b
```

```
ans =
```

```
    2    3    4
    4    4    5
```

```
>> %Pengurangan
```

```
>> a - b
```

```
ans =
```

```
    0    1    2
    0    2    3
```

```
>> |
```

Untuk menghitung panjang elemen kita bisa menggunakan perintah *length*. Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Menghitung panjang elemen matrik
```

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

```
a =
```

```
    2    1    3
    2    3    1
```

```
>> length(a)
```

```
ans =
```

```
    3
```

Untuk menghitung jumlah dari keseluruhan elemen kita bisa menggunakan perintah *sum*.

Untuk lebih jelasnya ikuti contoh berikut:

```
>> %Menghitung jumlah keseluruhan matri
```

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

```
a =
```

```
    2    1    3
    2    3    1
```

```
>> sum(a)
```

```
ans =
```

```
    4    4    4
```

Untuk mencari nilai terkecil kita bisa menggunakan perintah *min*. Untuk lebih jelasnya

ikuti contoh berikut:

```
>> %Mencari nilai terkecil pada matrik
```

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

```
a =
```

```
    2    1    3
    2    3    1
```

```
>> min(a)
```

```
ans =
```

```
    2    1    1
```

Untuk mencari nilai terbesar kita bisa menggunakan perintah *max*. Untuk lebih jelasnya

ikuti contoh berikut:

```
>> %Mencari nilai terbesar pada matrik
```

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

```
a =
```

```
    2    1    3
    2    3    1
```

```
>> max(a)
```

```
ans =
```

```
    2    3    3
```

Untuk mengurutkan elemen pada matrik kita bisa menggunakan perintah *sort*. Untuk lebih jelasnya perhatikan contoh berikut:

```
>> %Pengurutan matrik  
>> a = [ 13 1 3; 2 3 1; 12 42 5; 41 52 5]
```

```
a =
```

13	1	3
2	3	1
12	42	5
41	52	5

```
>> sort(a)
```

```
ans =
```

2	1	1
12	3	3
13	42	5
41	52	5

PEMBAHASAN 8 : GRAFIK

Pada pembahasan kali ini kita akan mempelajari cara membuat grafik pada MATLAB. Pembahasan pertama kita akan membuat plot sederhana dengan menggunakan dua vektor yaitu x dan y . untuk lebih jelasnya ikuti contoh berikut:

```
>> %Membuat grafik sederhana menggunakan vektor x dan y  
>> x = [ 1 2 3]
```

```
x =
```

```
1    2    3
```

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

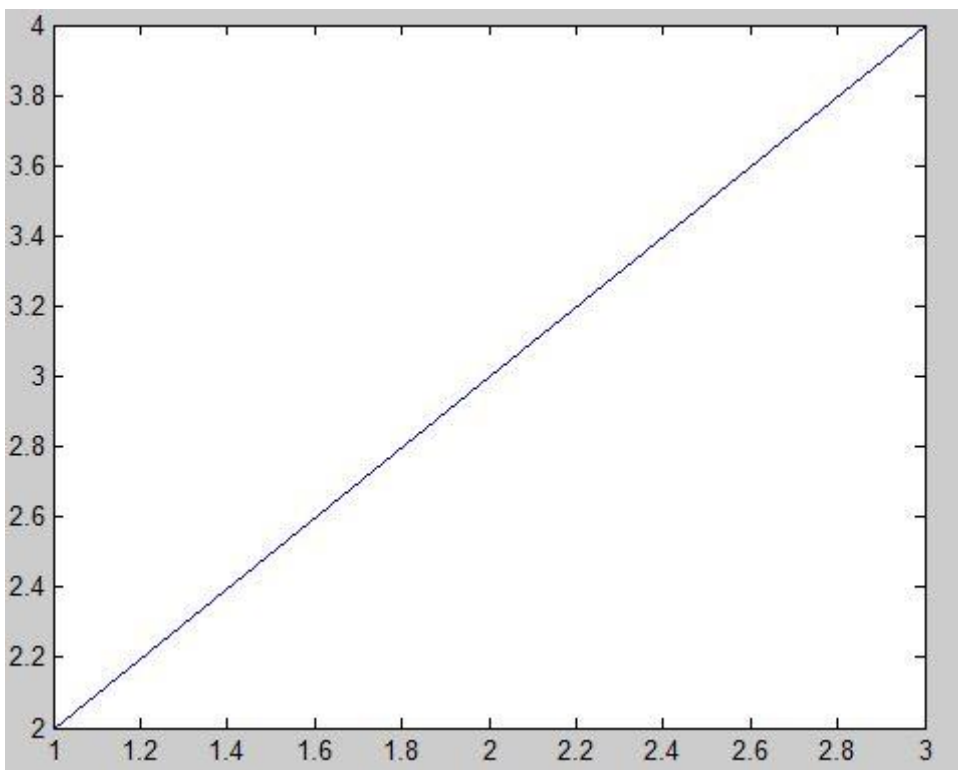
```
y =
```

```
2    3    4
```

```
>> plot(x,y)
```

```
>> |
```

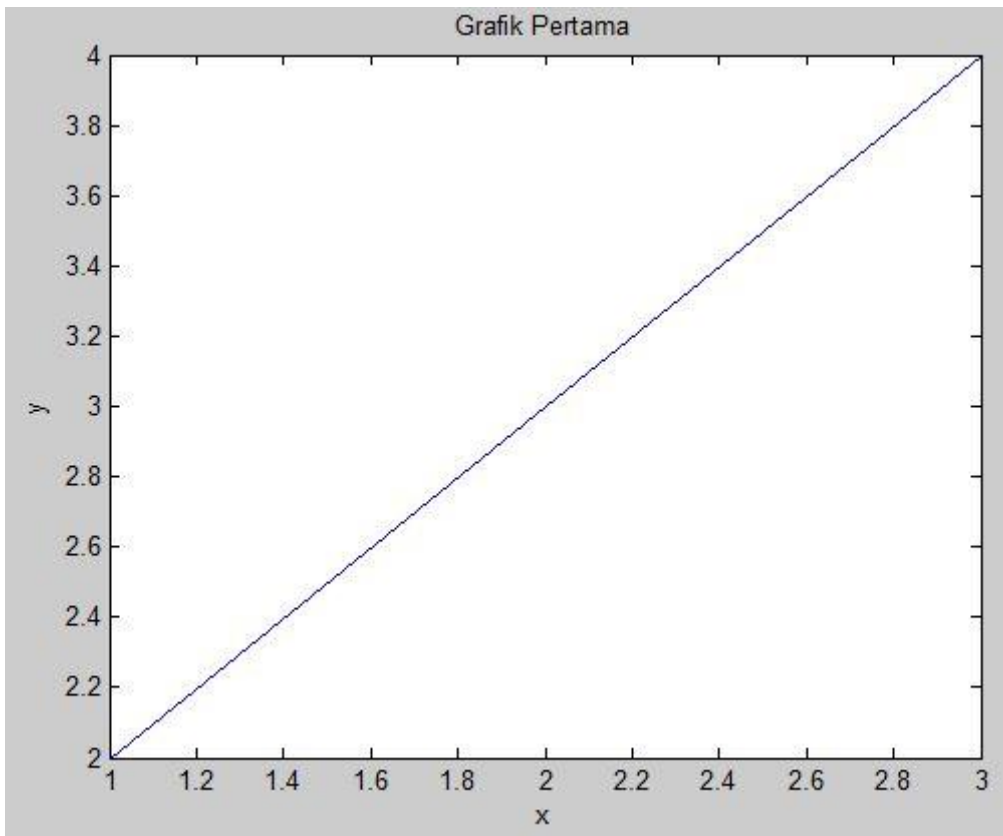
Berikut gambar grafik yang dihasilkan vektor x dan y pada contoh diatas:



Untuk menambahkan label pada vektor x dan y kita bisa menggunakan perintah *label*. Sedangkan untuk menambahkan judul kita bisa menggunakan perintah *title*('judul'). Untuk lebih jelasnya perhatikan contoh berikut:

```
>> plot(x,y)
>> title('Grafik Pertama')
>> xlabel('x')
>> ylabel('y')
>> |
```

Gambar plot yang dihasilkan setelah ditambahkan judul dan label pada sumbu x dan y :



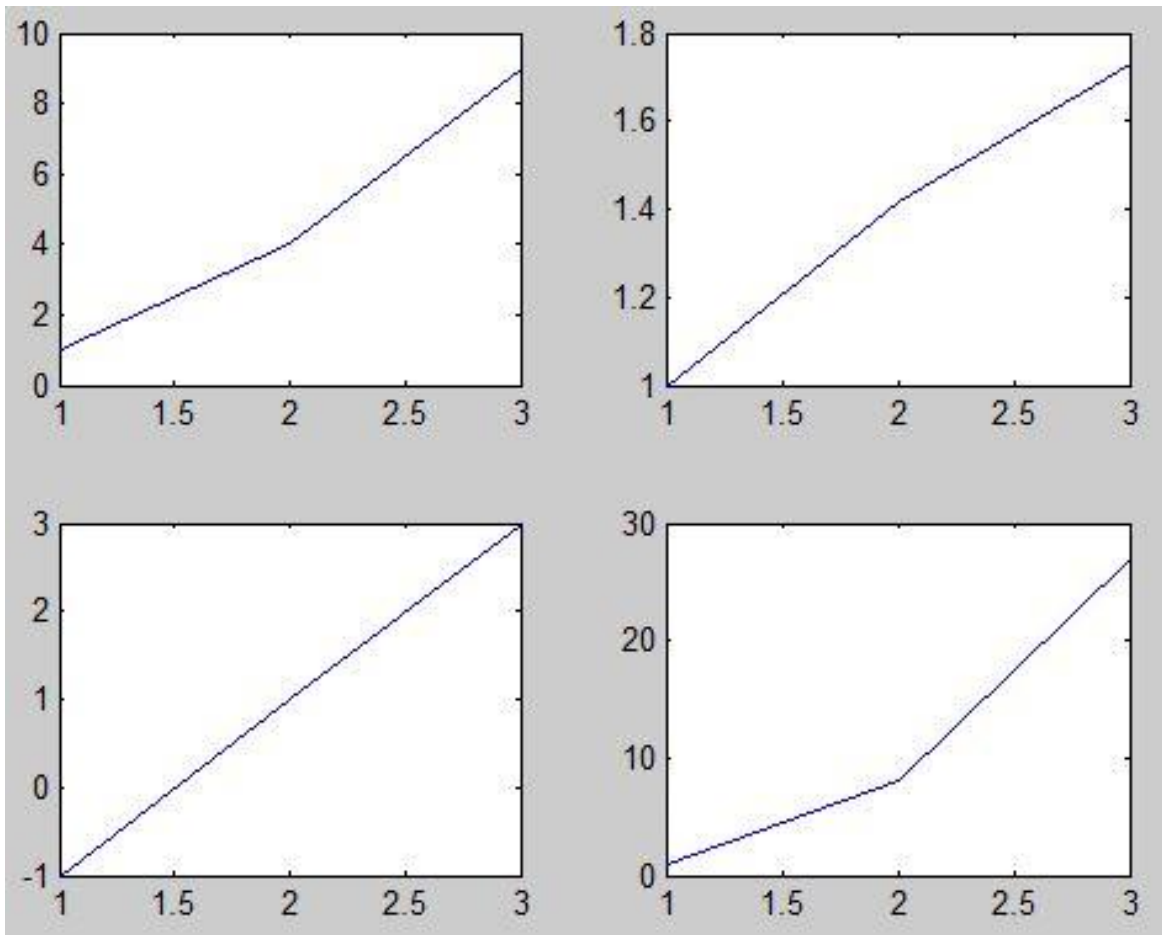
Salah satu kelebihan MATLAB adalah kita diizinkan untuk membuat 4 atau lebih grafik sekaligus dalam satu gambar sehingga akan memudahkan kita dalam membuat grafik yang kita inginkan. Untuk melakukannya kita bisa menggunakan perintah *subplot*. Untuk lebih jelasnya perhatikan contoh berikut:

```

>> %Menggunakan subplot pada MATLAB
>> x= [ 1 2 3];
>> y = x.^2;
>> z = sqrt(x);
>> w= 2*x - 3;
>> v= x.^3;
>> subplot(2,2,1)
>> plot(x,y)
>> subplot(2,2,2)
>> plot(x,z)
>> subplot(2,2,3)
>> plot(x,w)
>> subplot(2,2,4)
>> plot(x,v)
>> |

```

Gambar plot yang dihasilkan pada contoh diatas:



DAFTAR PUSTAKA

1. MATLAB for Engineer by Peter I. Katan
2. <https://www.facebook.com/groups/289294171144791/?ref=browser>
3. <https://www.facebook.com/groups/44152194130/?ref=browser>
4. <https://www.facebook.com/groups/Matlab.Simulink.for.All/?ref=browser>