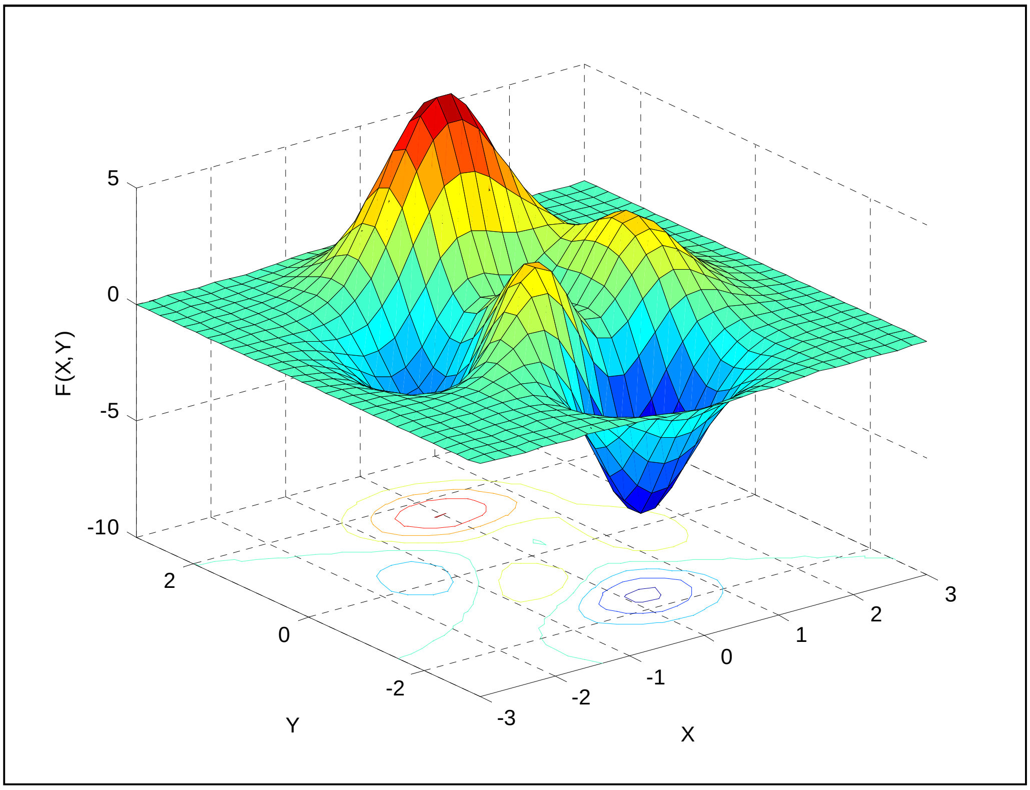
Pengantar Aplikasi MATLAB dalam Teknik Kimia



Heri Rustamaji

JURUSAN TEKNIK KIMIA FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG

2010

BAB 1 Memulai Menggunakan MATLAB

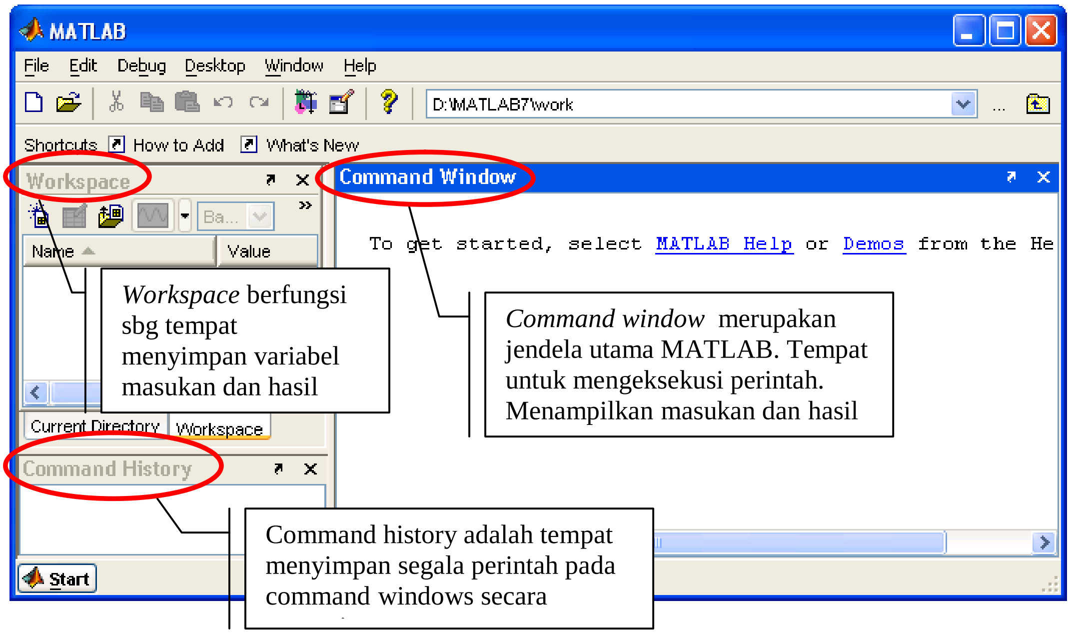
Matlab merupakan bahasa canggih untuk komputansi teknik. Matlab merupakan integrasi dari komputansi, visualisasi dan pemograman dalam suatu lingkungan yang mudah digunakan, karena permasalahan dan pemecahannya dinyatakan dalam notasi matematika biasa. Kegunaan Matlab secara umum adalah untuk :

- Matematika dan Komputansi
- Pengembangan dan Algoritma
- Pemodelan,simulasi dan pembuatan prototype
- Analisa Data,eksplorasi dan visualisasi
- Pembuatan apilikasi termasuk pembuatan *graphical user interface*

Matlab adalah sistem interaktif dengan elemen dasar array yang merupakan basis datanya. Array tersebut tidak perlu dinyatakan khusus seperti di bahasa pemograman yang ada sekarang. Hal ini memungkinkan anda untuk memecahkan banyak masalah perhitungan teknik, khususnya yang melibatkan matriks dan vektor dengan waktu yang lebih singkat dari waktu yang dibutuhkan untuk menulis program dalam bahasa C atau Fortran. Untuk memahami matlab, terlebih dahulu anda harus sudah paham mengenai matematika terutama operasi vektor dan matriks, karena operasi matriks merupakan inti utama dari matlab. Pada intinya matlab merupakan sekumpulan fungsi-fungsi yang dapat dipanggil dan dieksekusi. Fungsi-fungsi tersebut dibagi-bagi berdasarkan kegunaannya yang dikelompokan didalam toolbox yang ada pada matlab. Untuk mengetahui lebih jauh mengenai toolbox yang ada di matlab dan fungsinya anda dapat mencarinya di website http://www.mathworks.com,

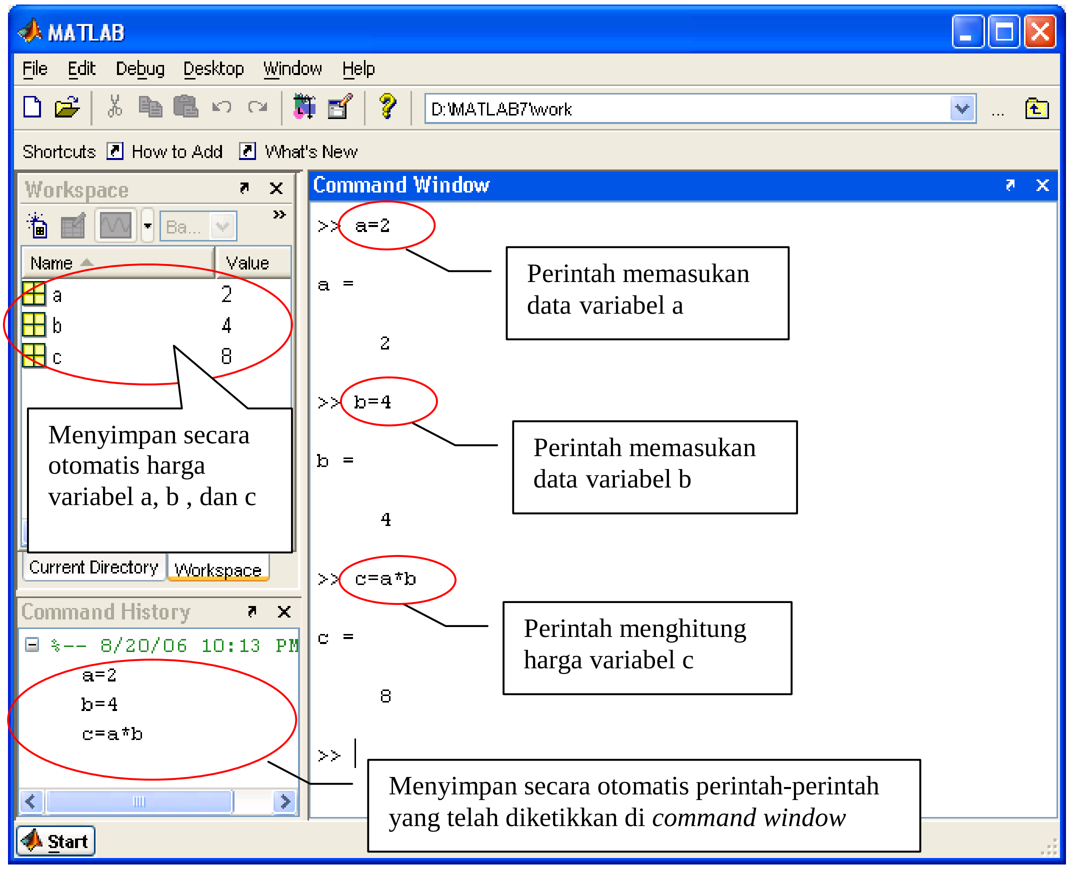
1.1 Desktop Matlab

Ketika anda mulai membuka program Matlab, akan muncul desktop Matlab yang berisi tools (*Graphical User Interface*) untuk mengatur file, variabel dan aplikasi yang berhubungan dengan Matlab.
Sebagai ilustrasi dibawah ini digambarkan *desktop* yang pertama muncul di Matlab.
Gambar 1 menunjukkan desktop MATLAB.



Gambar 1. Desktop Matlab

Untuk lebih jelas mengenai lingkungan kerja MATLAB perhatikan contoh berikut ini.



Gambar 2. Perintah dalam Command Window

1.2 Ruang Kerja Matlab

Saat anda bekerja di command window semua perintah, variabel dan data yang disimpan berada di dalam ruang kerja Matlab. Ruang kerja “default” dari Matlab yaitu di folder work di dalam folder Matlab. Apabila kita menginstal Matlab versi 7.1 di C maka folder work akan berada di C:/Matlab7p 1/work. Untuk merubah ruang kerja lakukan di Command Window, seperti anda merubah direktori di DOS.

Coba anda ketik **tes = 2** pada command window, maka akan keluar output sebagai berikut :

```
tes =
     2
```

Ini berarti variabel tes telah tersimpan di dalam ruang kerja kita.

Untuk melihat data yang telah tersimpan coba anda ketik **tes** pada commands window.

Jika anda tidak dapat mengingat nama setiap variable, maka anda dapat meminta Matlab untuk menampilkan namanya, menggunakan perintah who atau whos.

```
whos

  Name        Size                  Bytes Class

  tes         1x1                       8 double array
  tes2        1x1                       8 double array

Grand total is 2 elements using 16 bytes
```

Untuk mengetahui isi variabel tersebut anda harus memasukkan nama variabelnya dalam command window.

Untuk memanggil perintah sebelumnya, di Matlab menggunakan tombol panah pada keyboard anda (←↑↓→) .

Untuk menghapus semua semua variabel yang sudah kita masukkan digunakan perintah **clear all**

Untuk melihat keterangan dari function di Matlab atau program yang kita buat digunakan perintah : ' **help function** ' , sebagai contoh :

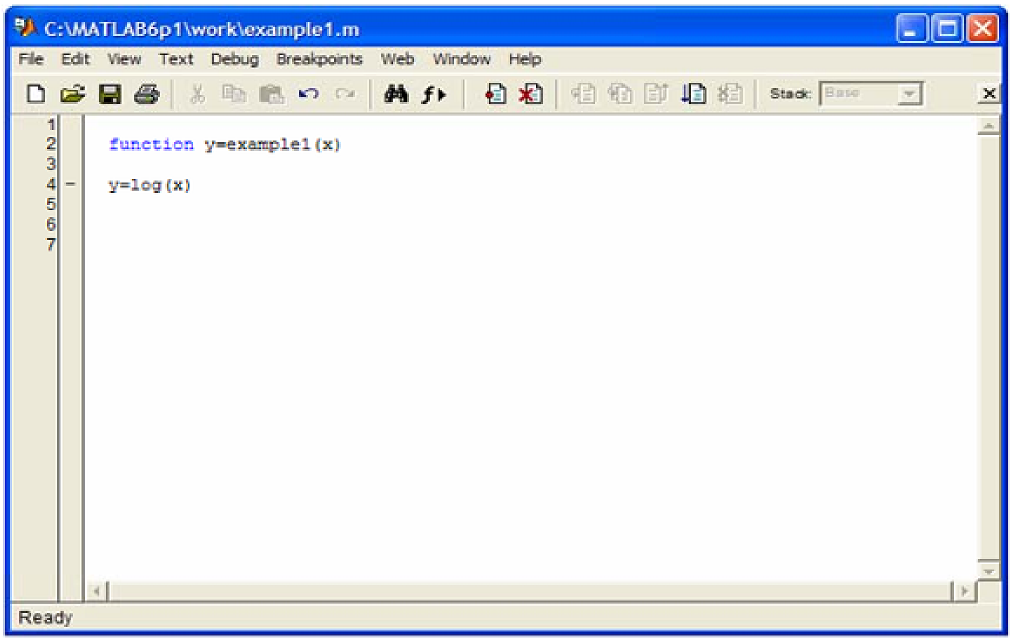
```
help plot
```

1.3 M-files

M-files dapat berisi program, scripts, persamaan atau data yang dipanggil selama eksekusi. Jika m-files adalah definisi fungsi, selanjutnya bagian yang terpenting dari jenis m-file ini adalah baris pertama. Baris pertama **harus** mengandung definisi fungsi sehingga MATLAB dapat menemukan m-files yang dipanggil. M-files tipe ini disebut dengan fungsi m-files atau fungsi file. Kode yang digunakan untuk mendefinisikan file adalah sebagai berikut:

```
function  z = file_name(x,y)
```

'file_name' adalah nama sederhana m-file (nama file harus sama dalam definisi dan nama file. Baris script selanjutnya dalam m-file dapat mendefinisikan fungsi atau fungsi dan label beberapa variable yang diperlukan. Berikut ini contoh suatu m-file yang digunakan untuk mengeplot fungsi logaritma bilangan natural.



Gambar 3. Penulisan dalam m-files

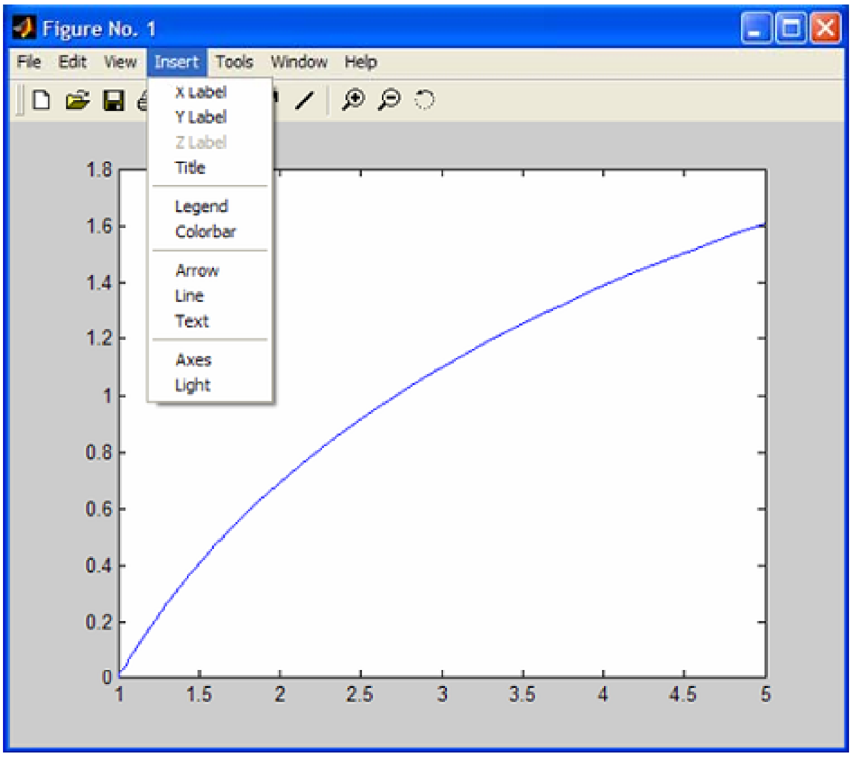
Untuk menghasilkan plot fungsi ini , berikut ini kode yang damasukkan ke dalam command window:

```
fplot ('example1',[1:5])
```

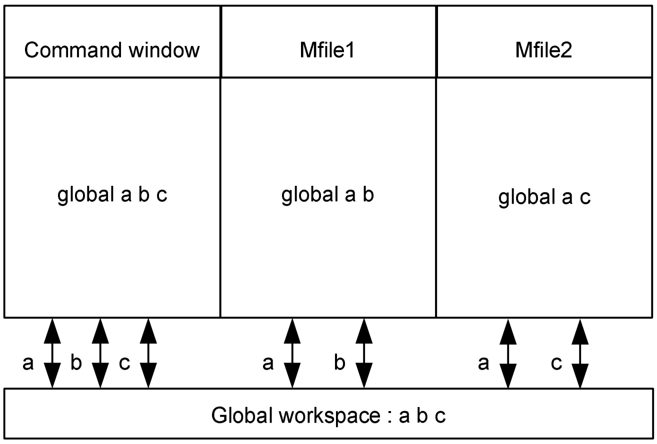
Ini akan mnghasilkan plot ln (x) antara x =1 dan x = 5. Hasil plot ditunjukkan Gambar 4. Dengan menggunakan menu'insert' kita dapat menambah judul, nama sumbu x dan y dan jika perlu *legend*

1.4 Perintah 'global'

Command workspace adalah area di dalam memori komputer dimana tersimpan berbagai parameter dan variabel yang telah didefinisikan pada *command line* (baik pada *command window* ataupun pada m-file) MATLAB. Masing – masing m-file memiliki workspace khusus yang masing – masing tidak dapat berkomunikasi sampai kita membuatnya saling berkomunikasi. Salah satu cara untuk mengkomunikasikan masing – masing workspace tersebut adalah dengan perintah ’global’ (Lihat Gambar 5).



Gambar 4. Grafik hasil fugsi example1



Gambar 5. Variabel global dan lokal**.**

1.5 Pemrograman pada Matlab

Hampir seluruh operasi Matlab dilaksanakan melalui script, yaitu urutan perintah-perintah yang dituliskan dalam bentuk teks. Seperti halnya script-script lainnya (misalnya file-file berekstensi .bat di sistem operasi DOS), script Matlab juga harus dijalankan dalam program Matlab sebagai lingkungannya.

Script ini dapat berupa urutan perintah seperti layaknya script-script lainnya, tetapi dapat

juga merupakan sebuah fungsi atau subrutin. Sebagai fungsi, script matlab dapat menerima variabel dan menghasilkan suatu besaran. Dengan demikian, pekerjaan dapat dipecah-pecah menjadi beberapa pekerjaan kecil dan dilaksanakan dalam fungsi-fungsi tersebut sehingga penyelesaiannya menjadi lebih mudah.
Berikut akan dipaparkan bagaimana membangun script dan fungsi, menempatkannya dalam lingkungan operasi Matlab, mengeksekusinya, menangani variabel-variabel yang terlibat di dalamnya, melewatkan variabel melalui fungsi, dan mengatur aliran program.

1.6 Membuat File Program yang Dapat Dieksekusi pada Matlab

File script adalah sebuah file teks biasa dan bisa dibuat menggunakan editor teks biasa seperti notepad pada windows atau editor edit.exe pada DOS. Tetapi pada Matlab versi 7.x yang baru, editor khusus telah disediakan. Editor ini dilengkapi dengan pustaka kata-kata kunci Matlab yang berwarna lain sehingga memudahkan penyuntingan program. Selain itu, editor dilengkapi juga dengan pemandu pasangan kurung (*bracket*), fungsinya menunjukkan pasangan kurung buka dan kurung tutup yang bersesuaian. Ini penting saat penyuntingan persamaan yang melibatkan kurung berlapis-lapis.
Untuk mengeksekusi script atau fungsi, tuliskanlah nama fungsi tersebut pada prompt (»). Jangan gunakan huruf kapital karena nama fungsi di Matlab diharuskan menggunakan huruf non-kapital.

1.6.1 Script : Urutan Perintah Matlab

Untuk membuat script biasa, tuliskan perintah-perintah Matlab dengan urutan yang benar. Perhatikan contoh berikut :

```
% contoh1.m : Menghitung perkalian matriks

A = [1 2 3; 4 5 6]
B = [2 3; 4 5; 6 7]
C = A*B
```

```
» contoh1

A =

     1     2     3
     4     5     6


B =

     2     3
     4     5
     6     7


C =

    28    34
    64    79
```

Script dapat menerima masukan melalui input dari keyboard, tetapi tidak dapat menerima masukan berupa argumen. Karena itu script hanya digunakan untuk program-program singkat, atau program induk.

1.6.2 Fungsi : Subrutin Matlab

Penggunaan fungsi lebih fleksibel dibandingkan script biasa. Fungsi dapat menerima masukan berupa argumen. Walaupun fungsi juga dapat menerima masukan dari keyboard, tetapi pemrogram biasanya tidak menempatkan pekerjaan ini dalam fungsi kecuali ada tujuan khusus untuk itu.

Sebuah fungsi harus memiliki header yang ditandai dengan kata-kunci **function**. Pada header tersebut terdapat variabel output, nama fungsi dan variabel input. Nama fungsi disini tidak selalu mencerminkan nama fungsi yang sebenarnya. Nama fungsi sebenarnya adalah nama file .m yang berisi fungsi tersebut. Walaupun demikian, nama fungsi pada header sebaiknya sama dengan nama file agar tidak membingungkan.
Perhatikan contoh berikut :

```
function R = contoh2(P,Q)
% contoh2.m : Menghitung perkalian matriks

R = P*Q
```

```
» A = [1 2 3; 4 5 6];
» B = [2 3; 4 5; 6 7];
» C = contoh2(A,B)

C =

    28    34
    64    79


C =

    28    34
    64    79
```

Perhatikan bahwa nama variabel di dalam fungsi dan variabel pemanggil fungsi tidak harus sama. Matlab membedakan variabel di luar fungsi dan variabel di dalam fungsi. Variabel di dalam fungsi bersifat lokal dan hanya berlaku di dalam fungsi. Hal ini akan dibahas pada sub-bagian lain.

1.7 Mengatur Alur Program

Sebagaimana bahasa pemrograman pada umumnya, Matlab juga memiliki perintah-perintah untuk mengatur alur program. Ada beberapa perintah Matlab yang dapat digunakan untuk mengatur alur program antara lain :

a. IF … ELSE

Perintah ini adalah perintah klasik pemrograman. Identik dengan jika .. maka, perintah ini dapat digunakan untuk menguji suatu kondisi tertentu. Sintaks dari perintah ini diperlihatkan pada contoh berikut :

```
% program : test.m
a = input('a = ');

if a == 0
   disp('a sama dengan nol');
elseif a < 0
   disp('a negatif');
else
   disp('a positif');
end
```

```
» test
a = 4
```

```
a positif
» test
a = -4
a negatif
» test
a = 0
a sama dengan nol
```

b. FOR

Perintah ini juga merupakan perintah klasik bahasa pemrograman. Fungsi for adalah untuk melakukan loop sejumlah urutan yang telah ditentukan. Sintaks dari perintah ini diperlihatkan pada contoh berikut :

```
for i=1:10
   for j=1:2:10
      A(i,j) = (i+j);
   end;
end;
A
```

```
» test

A =

     2     0     4     0     6     0     8     0    10
     3     0     5     0     7     0     9     0    11
     4     0     6     0     8     0    10     0    12
     5     0     7     0     9     0    11     0    13
     6     0     8     0    10     0    12     0    14
     7     0     9     0    11     0    13     0    15
     8     0    10     0    12     0    14     0    16
     9     0    11     0    13     0    15     0    17
    10     0    12     0    14     0    16     0    18
    11     0    13     0    15     0    17     0    19
```

Urutan loop dapat sebuah vektor dengan bilangan-bilangan tertentu seperti contoh di atas : untuk i urutan dari 1 sampai 10 dengan kenaikan 1, sedangkan untuk j dari 1 sampai 10 dengan kenaikan 2. Jika x = [1 1.3 5.4 2.3 5.5 7], maka **for x=x** akan memberikan harga-harga x seperti setiap elemen dalam vektor x : x(1) = 1, x(3) = 5.4 dst.

c. WHILE

Perintah ini akan mengulang perintah-perintah yang diapitnya selama kondisi pengujinya benar. Di dalam loop harus ada perintah-perintah yang membuat kondisi penguji menjadi salah, karena kalau tidak loop akan berlangsung terus.
Contoh baik dapat dilihat para perhitungan epsilon mesin Matlab.

```
% program : epsilon.m
satu = inf;
eps = 1;
while satu > 1
   eps = eps/2;                    % nilai eps semakin lama semakin kecil
   satu = 1 + eps;                 % satu suatu saat akan sama dengan 1
end;
eps = eps*2
```

```
» epsilon

eps =

  2.2204e-016
```

d. SWITCH … CASE … OTHERWISE

Perintah ini mengarahkan alur program melalui sejumlah pilihan. Perintah ini untuk menggantikan perintah IF…ELSEIF…ELSE yang bertumpuk.

```
a = input('a = ');
switch a
case {0}, disp('a nol')
case {1}, disp('a satu')
otherwise
   disp('a bukan nol atau satu');
end;
```

```
» pilihan
a = 1
a satu
» pilihan
a = 0
a nol
» pilihan
a = -3
a bukan nol atau satu
» pilihan
a = 4
a bukan nol atau satu
```

1.8 Operasi Vektor dan Matriks

Beberapa operasi vektor dan matriks yang penting antara lain adalah :

- Transposisi

  Transposisi vektor dan matriks dinyatakan dengan simbol apostrop (‘). Secara sederhana, definisi transposisi vektor dan matriks adalah mengubah posisi elemen-elemen kolom dalam vektor dan matriks menjadi elemen-elemen baris pada vektor dan matriks ybs.

  Secara matematik dapat dinyatakan sbb. :

$$A = B' \text{ jika } a_{ij} = b_{ji}$$

  Pada Matlab, perintah ini dapat dilakukan dengan :

```
» A = [1 2 3]

A =

     1     2     3

» B = A'

B =

     1
     2
     3

» P = [1 2 3; 4 5 6; 7 8 9]

P =

     1     2     3
```

```
     4     5     6
     7     8     9

» Q = P'

Q =

     1     4     7
     2     5     8
     3     6     9
```

Perlu diingat bahwa Matlab selalu mendefinisikan sebuah array (vektor) sebagai vektor baris, kecuali didefinisikan terlebih dahulu. Misalnya A = 1:5 adalah vektor [1 2 3 4 5] dan bukan vektor kolom. Hal ini berbeda dengan definisi awal (*default*) vektor pada beberapa buku referensi komputasi.

- Penjumlahan dan Pengurangan

  Penjumlahan dan pengurangan vektor dan matriks adalah penjumlahan masing-masing elemennya. Pada Matlab, operasi penjumlahan dan pengurangan ditandai dengan tanda plus (+) dan minus (-).

```
A =

     5     6     7

» B = [1 2 3]

B =

     1     2     3

» A - B

ans =

     4     4     4
```

Syarat penjumlahan dan pengurangan adalah dimensi kedua vektor atau matriks yang dijumlahkan harus sama.

- Perkalian Vektor dan Matriks

  Perkalian vektor dan matriks dilakukan menurut persamaan :

$$C = AB \qquad c_{ik} = a_{ij}.b_{jk}$$

Syarat perkalian vektor dan matriks adalah jumlah baris vektor/matriks pertama harus sama dengan jumlah kolom vektor/matriks kedua.

```
» A = [1 2 3; 4 5 6; 7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9

» B = [1 3; 2 4; 5 7]

B =

     1     3
     2     4
     5     7

» A*B

ans =

    20    32
    44    74
    68   116
```

- Pembagian Matriks
  Terdapat dua macam pembagian matriks :
  Pembagian kiri : x = A\B diartikan sebagai penyelesaian A*x = B
  Pembagian kanan : x = A/B diartikan sebagai x = A*inv(B)

- Pemangkatan Matriks
  Pemangkatan n matriks adalah perkalian matriks tersebut n kali. Syarat pemangkatan matriks tentu saja adalah bahwa matriks yang akan dipangkatkan harus bujur-sangkar sehingga dapat dikalikan berulang-ulang.

1.8.1 Operasi Elementer

Beberapa operasi khusus terhadap elemen vektor dan matriks adalah :

- Perkalian/pembagian elemen vektor dan matriks
  Perkalian elementer vektor dan matriks lain dengan perkalian vektor dan matriks biasa. Perkalian elementer ini adalah perkalian antara elemen-elemen dari dua vektor atau matriks.

```
» A = [1 2 3];
» B = [4 5 6];
» A*B'
```

```
ans =

    32                                    (perkalian vektor biasa)

» A.*B

ans =

     4     10     18                      (perkalian elementer)

» A./B

ans =

    0.2500    0.4000    0.5000            (pembagian elementer)
```

- Pemangkatan elemen vektor dan matriks
  Sama halnya dengan perkalian dan pembagian matriks, pemangkatan elementer juga berlaku untuk setiap elemen suatu vektor/matriks.

```
» A.^2

ans =

     1     4     9                   ( [1^2 2^2 3^2] )

» B.^A

ans =

     4    25   216                   ( [4^1 5^2 6^3] )
```

1.7.2 Operasi Vektor dan Matriks pada Fungsi

Operasi fungsional pada Matlab dapat diaplikasikan pada vektor dan/atau matriks. Misalnya jika x = [1 2 3] dan $y = x^2 + 2x + 4$, maka persamaan y(x) dapat dievaluasi secara vektor.

```
» x = [1 2 3];

» y = x.^2 + 2*x + 4

y =

     7     12     19
```

Beberapa fungsi bawaan dapat juga diaplikasikan secara vektor dan matriks misalnya sinus, cosinus, exponent, logatirma dls.

Pemahaman tentang operasi pada vektor dan matriks ini akan sangat membantu menyelesaikan masalah menggunakan Matlab, karena beberapa kegiatan iteratif diganti dengan hanya sebuah atau beberapa buah perintah Matlab. Sebagai ilustrasi, simaklah contoh berikut ini :

Contoh 1.1 :

Untuk menentukan harga kapasitas panas suatu campuran gas pada suatu temperatur biasanya digunakan persamaan polinom kapasitas panas zat murni. Misalkan komposisi gas adalah y_i dan $Cp_i = A_i + B_iT + C_iT^2 + D_iT^3$ maka perhitungan kapasitas panas campuran pada T = 300 K dapat dilakukan dengan cara :

- cara pertama dengan pemrograman biasa :

```
% test1.m

Cp = [1.2 0.02 0.00323 0.000003233;
   3.2 0.013 0.00466 0.000004345]

y = [0.4 0.6]

T = 300

Cpc = 0
for i=1:length(y)
   Cpi = 0
   for j=1:length(Cp)
      Cpi = Cpi + Cp(i,j)*T^(j-1)
   end;
   Cpc = Cpc + y(i)*Cpi
end;
Cpc

» test1

Cpc =

  480.3654
```

- cara kedua dengan perhitungan vektor/matriks :

```
% test2.m

Cp = [1.2 0.02 0.00323 0.000003233;
   3.2 0.013 0.00466 0.000004345];

y = [0.4 0.6];

T = 300;

P = 0:3;                    % mendefinisikan pangkat P = [0 1 2 3]
TT = T .* ones(1,4);        % menghitung vektor TT = [300 300 300 300]
TT = TT.^P                  % menghitung TT = [300^0 300^1 300^2 300^3]
Cpc = y*(Cp*TT')

» test2

Cpc =
  480.3654
```

Dari contoh ini dapat dilihat bahwa sejumlah prosedur yang biasanya dikodekan iteratif (menggunakan perintah for) dapat dihilangkan dan diganti dengan kode operasi matriks linier (tidak iteratif).

Pemahaman penggunaan operasi-operasi vektor dan matriks, selain akan mempercepat perhitungan, juga akan mempermudah pengkodean program dan dokumentasi program.

Latihan:
Menghitung Entalpi Campuran Gas
Melalui cara yang telah dibahas di atas , hitunglah entalpi suatu campuran gas CH_4, H_2O, H_2, CO_2, CO dengan komposisi masing-masing sebesar 0.22, 0.12, 0.41, 0.13, 0.12 pada temperatur 350 K. Data kapasitas panas dan entalpi pembentukan masing-masing gas murni adalah sbb. :

Gas	H_f	A	B	C	D	E
CH4	-17.89	38.3870	-7.3664E-2	2.9098E-4	-2.6385E-7	8.0068E-11
H2O	-57.80	34.0471	-9.6506E-3	3.2998E-5	-2.0447E-8	4.3022E-12
H2	0.0	17.6386	6.7005E-2	-1.3148E-4	1.0588E-7	-2.9180E-11
CO2	-94.05	19.0223	7.9629E-2	-7.3707E-5	3.7457E-8	-8.1330E-12
CO	-26.42	29.0063	2.4923E-3	-1.8644E-5	4.7989E-8	-2.8726E-11

Catatan : Hf dalam kkal/mol
$Cp = A + BT + CT^2 + DT^3 + ET^4$ [J/mol.K] dan T dalam K

Entalpi campuran gas dapat dihitung berdasarkan persamaan :

$$H = \sum_{i=1}^{n} \left(y(i) \left(H_{f,i} + \int_{To}^{T} Cp_i \, dT \right) \right)$$

BAB 2 Visualisasi Data Dalam MATLAB

Matlab menyediakan berbagai fungsi untuk menampilkan data secara dua dimensi maupun tiga dimensi. Pada kasus dimana Anda membuat grafik dalam tiga dimensi, Anda dapat menggambar permukaan dan menempatkan bingkai pada grafik tersebut. Warna digunakan untuk mewakili dimensi keempat.

Contoh 2.1

Data rekasi berikut telah diperoleh dari reaksi peluruhan sederhana:

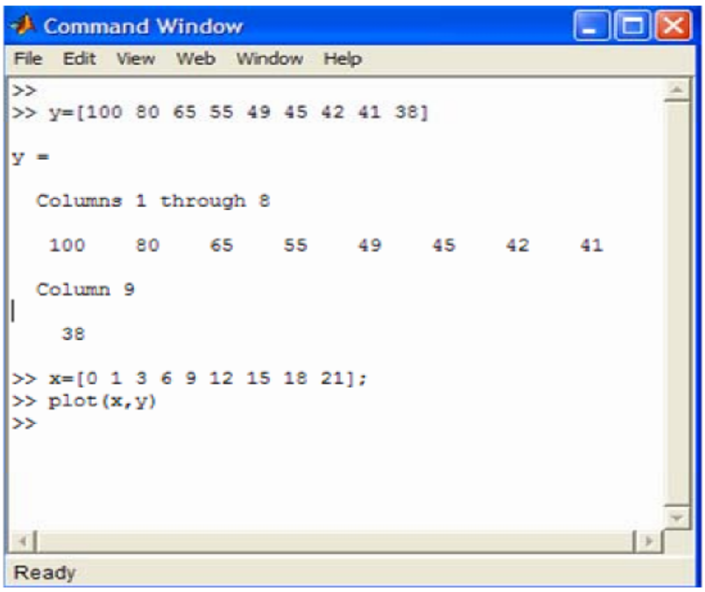
A→ B

Menggunakan MATLAB untuk memplot konsentrasi komponen A dalam mol/L terhadap waktu reaksi, dalam menit.

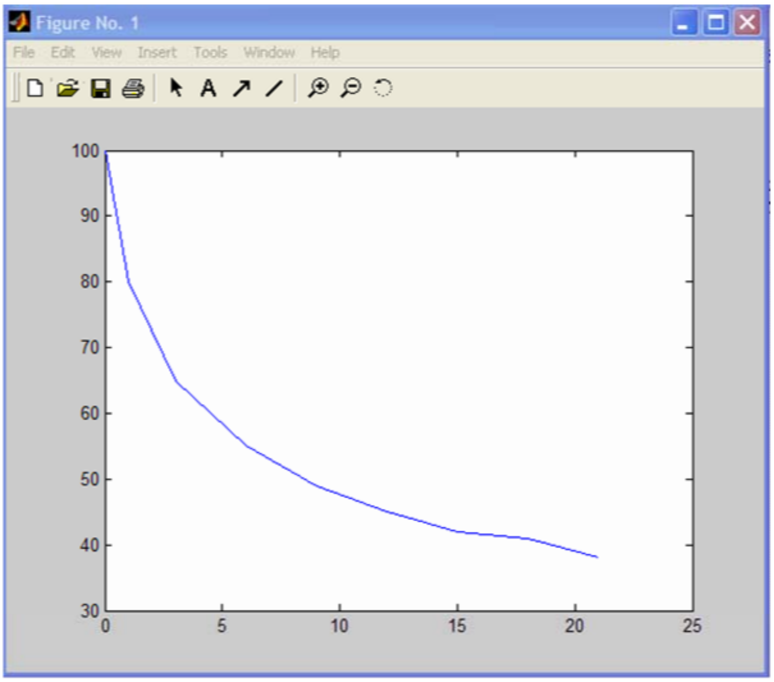
Time (Minutes)	Concentration (Mole/Liter)
0	100
1	80
3	65
6	55
9	49
12	45
15	42
18	41
21	38

Penyelesaian :

Pertama, data harus dimasukkan ke dalam MATLAB sebagai dua vektor. Vektor x dan y di definisikan dalam Command Window, diikuti dengan perintah untuk mengeplot data. Gambar di bawah menunjukkan cara penulisan.

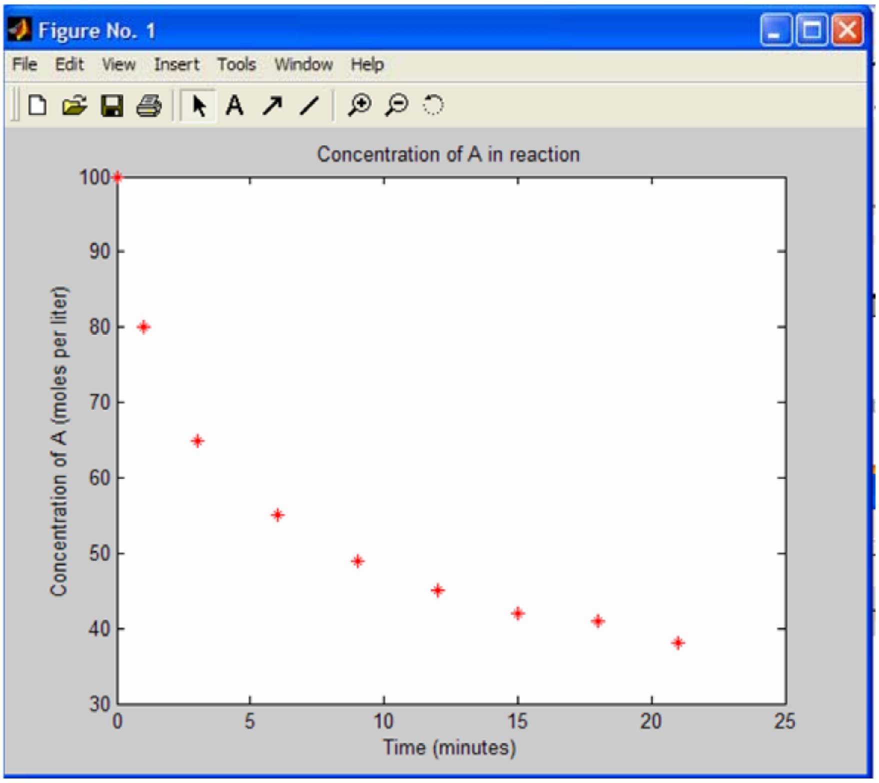


Hasil grafik ditunjukkan gambar di bawah.



Matrix baris ‘x’ (atau vektor) memiliki keluaran tampilan menggunakan ‘;’ pada akhir baris. Syntax dapat digunakan untk menentukan judul dan labels, tetapi lebih mudah menggunakan pendekatan berdasarkan GUI (Graphical User Interface) untuk mengedit gambar.

- Pilih perintah ‘Edit Plot’ pada menu ‘Tools’ atau klick kursor pada gambar
- Double click pada ruang putih dalam grafik. Hal ini memungkinkan untuk mengedit. Selanjutnya judul dan aksis dapat disisipkan di bawah perintah ‘label’
- Sekarang klick langsung pada garis, dan property editor garis akan muncul
- Selanjutnya warna garis, bentuk atau bentuk marker dapat diedit. Kurva akhir ditunjukkan di bawah.



Untuk menampilkan data statistik sederhana, ikuti petunjuk. ‘Tools Data Statistics’ dan nilai minimum, maximum, mean, median, standard deviation, and range x and y akan tampil. Di dalam kotak ini setiap statistik dapat ditambahkan ke kurva sebagai titik data/garis.

Anda dapat memilih sendiri style penandaan, warna dan bentuk garis dengan memberikan argumen ketiga pada fungsi plot untuk setiap pasangan array data. Argumen tambahan ini adalah suatu karakter string yang terdiri dari satu atau lebih karakter dari tabel di bawah ini:

Simbol	Warna	Penandaan	Simbol
b	Biru	**.**	Titik
r	Merah	**o**	Lingkaran
g	Hijau	**x**	Tanda x
c	Cyan	+	Tanda plus
m	Magenta	*	Tanda bintang
y	Kuning	**s**	Bujursangkar
k	Hitam	**d**	Diamon
w	Putih	**p**	pentagram
		h	heksagram

Menganalisis sekumpulan data percobaan

Sekumpulan data pengukuran sebuah spesimen adalah sbb. :

67.5	65.7	68.4	65.3	69.1	66.2	68.3	63.1	67.3	71.0
69.9	68.3	66.4	65.7	70.1	64.9	69.6	67.9	66.5	68.4

Analisis data menggunakan Matlab :

```
» D = [67.5 65.7 68.4 65.3 69.1 66.2  68.3 63.1 67.3 71.0 ...
       69.9 68.3 66.4 65.7 70.1 64.9 69.6 67.9 66.5 68.4]

D =

  Columns 1 through 7

   67.5000   65.7000   68.4000   65.3000   69.1000   66.2000   68.3000

  Columns 8 through 14

   63.1000   67.3000   71.0000   69.9000   68.3000   66.4000   65.7000

  Columns 15 through 20

   70.1000   64.9000   69.6000   67.9000   66.5000   68.4000

» max(D)                      % menentukan nilai maksimum

ans =

    71

» min(D)                      % menentukan nilai minimum

ans =

   63.1000

» mean(D)                     % menentukan nilai rata-rata

ans =

   67.4800

» median(D)                   % menentukan nilai tengah

ans =

   67.7000

» std(D)                      % menentukan standar deviasi

ans =

    2.0023

» sort(D)                     % mengurutkan data

ans =

  Columns 1 through 7

   63.1000   64.9000   65.3000   65.7000   65.7000   66.2000   66.4000

  Columns 8 through 14

   66.5000   67.3000   67.5000   67.9000   68.3000   68.3000   68.4000

  Columns 15 through 20

   68.4000   69.1000   69.6000   69.9000   70.1000   71.0000

» sum(D)                      % menjumlahkan data

ans =
```

```
  1.3496e+003
» prod(D)                      % mengalikan data
ans =
  3.8006e+036
» cumsum(D)                    % menjumlahkan data secara kumulatif
ans =
  1.0e+003 *
  Columns 1 through 7
    0.0675    0.1332    0.2016    0.2669    0.3360    0.4022    0.4705
  Columns 8 through 14
    0.5336    0.6009    0.6719    0.7418    0.8101    0.8765    0.9422
  Columns 15 through 20
    1.0123    1.0772    1.1468    1.2147    1.2812    1.3496
```

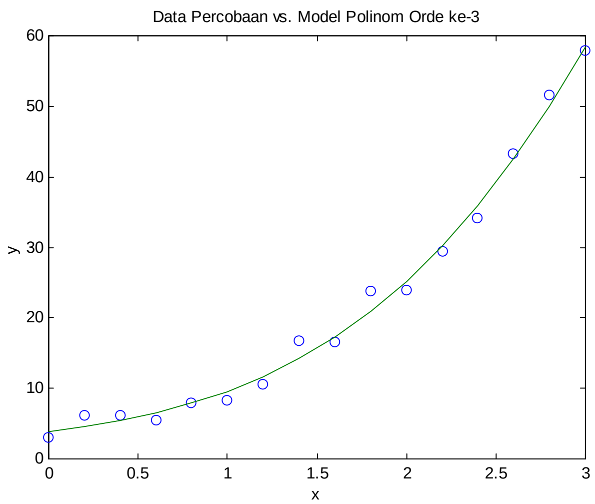
Menentukan model polinom dari sekumpulan data percobaan
Dari suatu hasil pengukuran diperoleh data sbb.

x	y
0	2.952383
0.2	6.209525
0.4	6.1624
0.6	5.465714
0.8	7.891386
1	8.181819
1.2	10.47825
1.4	16.69245
1.6	16.6094
1.8	23.76093
2	23.91732
2.2	29.41918
2.4	34.0725
2.6	43.34264
2.8	51.50684
3	57.81625

Menurut teori, data ini bersesuaian dengan model polinom orde ketiga. Berikut langkah-langkah untuk mendapatkan model tersebut

```
» x
x =
  Columns 1 through 7
         0    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000
  Columns 8 through 14
    1.4000    1.6000    1.8000    2.0000    2.2000    2.4000    2.6000
```

```
  Columns 15 through 16

    2.8000    3.0000

» y

y =

  Columns 1 through 7

    2.9524    6.2095    6.1624    5.4657    7.8914    8.1818   10.4782

  Columns 8 through 14

   16.6925   16.6094   23.7609   23.9173   29.4192   34.0725   43.3426

  Columns 15 through 16

   51.5068   57.8163

» polyfit(x,y,3)

P =

    1.2851    1.1226    3.2621    3.9039     % koefisien polinom orde-3

» yc = polyval(P,x)

yc =

  Columns 1 through 7

    3.9039    4.6115    5.4706    6.5429    7.8901    9.5738   11.6557

  Columns 8 through 14

   14.1975   17.2610   20.9077   25.1994   30.1978   35.9645   42.5613

  Columns 15 through 16

   50.0497   58.4916
```



Membuat polinom dan mencari akar polinom

Matlab memiliki fungsi-fungsi untuk membentuk polinom dari akar-akarnya, dan sebaliknya, menentukan akar-akar polinom dari sebuah polinom. Simaklah contoh berikut.

```
» P = [1 2 3 4]
```

```
P =

     1     2     3     4

» poly(P)

ans =

     1   -10    35   -50    24

» roots(ans)

ans =

    4.0000
    3.0000
    2.0000
    1.0000
```

Matlab juga dapat bekerja dengan bilangan kompleks. Untuk akar-akar yang mengandung bilangan kompleks, fungsi roots akan mengeluarkan jawaban berupa bilangan kompleks.

```
» roots(P)

ans =

  -1.6506
  -0.1747 + 1.5469i
  -0.1747 - 1.5469i
```

BAB 3 Penyelesaian Persamaan Linier Simultan

Bentuk umum persamaan linier simultan adalah:

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = {}_{12} \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3 \\
\quad . \qquad . \qquad . \qquad . \qquad . \qquad . \\
\quad . \qquad . \qquad . \qquad . \qquad . \qquad . \\
\quad . \qquad . \qquad . \qquad . \qquad . \qquad . \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n
\end{array}
$$

Dalam hal ini akan dicari harga x_1, x_2,…, x_n. Dalam MATLAB, jenis persamaan di atas dapat diselesaikan dengan menggunakan fasilitas yang sudah tersedia yaitu operasi matrix.

Contoh 3.1 Mencari beberapa variable persaaan linier simultan.
Diketahui sebuah sistem persamaan linier sbb. :

$$
\begin{array}{l}
5x_1 - 3x_2 + x_3 = 5 \\
x_1 + 2x_2 - 3x_3 + x_4 = 1 \\
3x_2 - 5x_3 + 6x_4 = -5 \\
-x_1 + 2x_3 - x_4 = 0
\end{array}
$$

Sistem persamaan linier ini dapat dinyatakan dalam bentuk persamaan matriks :

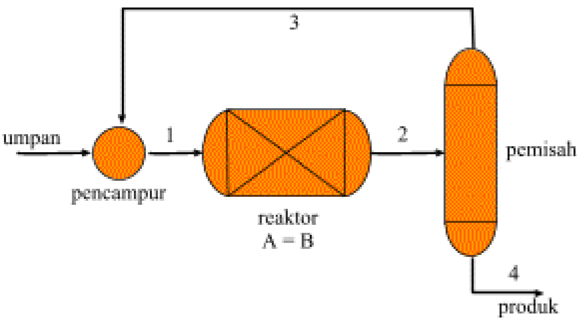
$$
\begin{bmatrix} 5 & -3 & 1 & 0 \\ 1 & 2 & -3 & 1 \\ 0 & 3 & -5 & 6 \\ -1 & 0 & 2 & -1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\begin{bmatrix} 5 \\ 1 \\ -5 \\ 0 \end{bmatrix}
$$

Penyelesaian pada Matlab :

```
» A = [5 -3 1 0; 1 2 -3 1; 0 3 -5 6; -1 0 2 -1]

A =
     5    -3     1     0
     1     2    -3     1
     0     3    -5     6
    -1     0     2    -1

» b = [5 1 -5 0]
```

```
b =
     5     1    -5     0

» x = A\b'
x =
    1.3165
    0.5823
    0.1646
   -0.9873

» A*x
ans =
    5.0000
    1.0000
   -5.0000
         0
```

Contoh 3.2 : Neraca Massa Linier Rangkaian Proses.
Suatu bahan A akan dikonversikan menjadi B dalam sebuah reaktor. Produk B keluar bersama reaktan A yang tidak bereaksi menuju pemisah sehingga reaktan A dapat dikembalikan ke reaktor. Gambar skema proses tesebut ditunjukkan pada gambar di bawah.



Produk berupa zat A murni dengan laju 100 kmol/jam. Kendala proses adalah :
1. 80 % dari A dan 40 % dari B di dalam alur 2 di daur-ulang.
2. Perbandingan mol A terhadap B di dalam alur 1 adalah 5 : 1.

Neraca massa Pencampur :

$N_{A1} – N_{A3} = 100$

$N_{B1} – N_{B3} = 0$

Neraca massa reaktor :

$- N_{A1} + N_{A2} + r = 0$

$- N_{B1} – N_{B2} – r = 0$

(r = laju reaksi)

Neraca massa pemisah :

$- N_{A2} + N_{A3} + N_{A4} = 0$
$- N_{B2} + N_{B3} + N_{B4} = 0$

Kendala-kendala :
Porsi cabang : $0.8\ N_{A2} + N_{A3} = 0$
$-0.4\ N_{B2} + N_{B3} = 0$
Hubungan komposisi alur : $N_{A1} – 5\ N_{B1} = 0$

Ada 9 persamaan linier dengan 9 variabel yang tak diketahui :
N_{A1}, N_{B1}, N_{A2}, N_{B2}, N_{A3}, N_{B3}, N_{A4}, N_{B4}, dan r.

Persamaan-persamaan di atas dapat dituliskan kembali dalam bentuk matriks sbb. :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & -0.8 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4 & 0 & 1 & 0 & 0 & 0 \\ 1 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} N_{A1} \\ N_{A2} \\ N_{A3} \\ N_{A4} \\ N_{A5} \\ N_{A6} \\ N_{A7} \\ N_{A8} \\ N_{A9} \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Script Matlab untuk menyelesaikan sistem persamaan ini adalah sbb. :

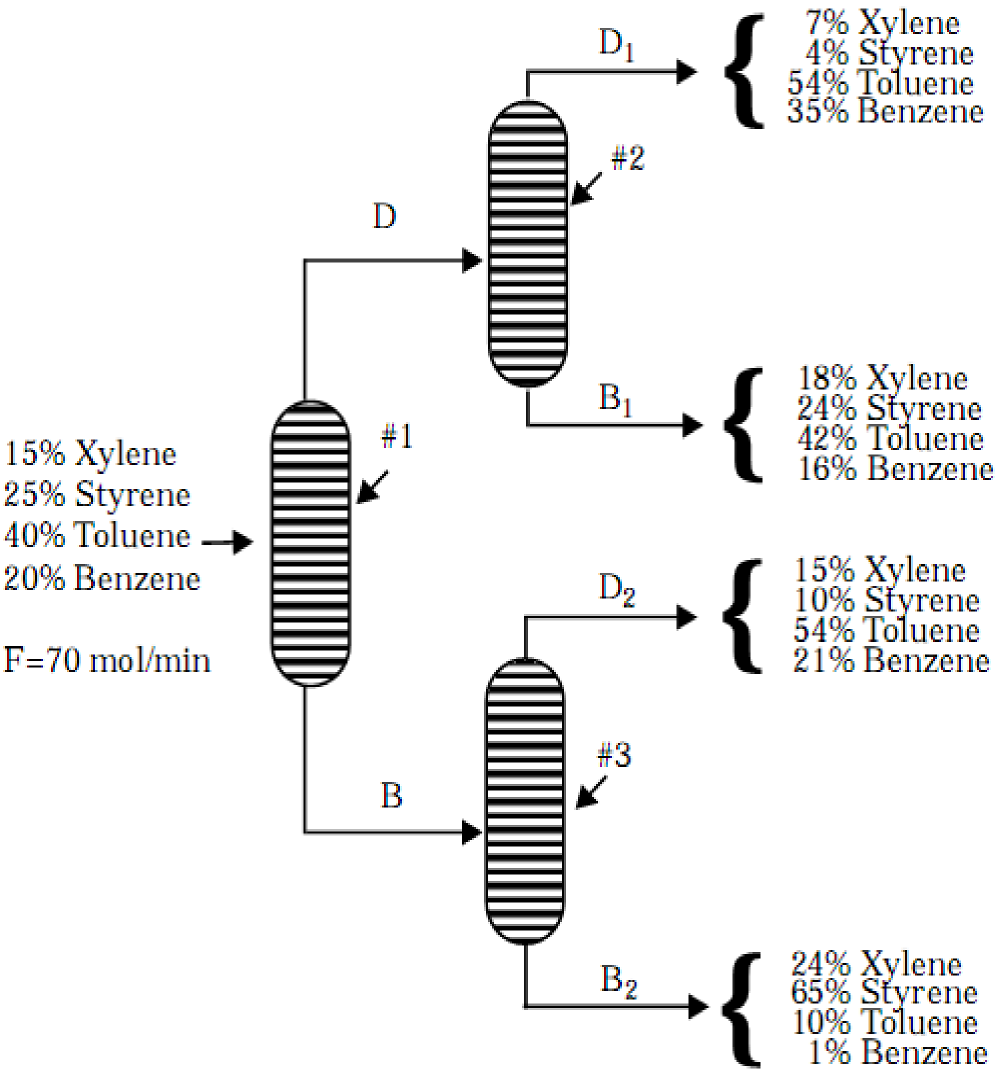
```
% Penyelesaian persamaan neraca massa : neraca.m

A = [1  0  0  0 -1  0  0  0  0;
    0  1  0  0  0 -1  0  0  0;
    -1  0  1  0  0  0  0  0  1;
    0  -1  0  1  0  0  0  0  -1;
    0  0  -1  0  1  0  1  0  0;
    0  0  0  -1  0  1  0  1  0;
    0  0  -0.8  0  1  0  0  0  0;
    0  0  0  -0.4  0  1  0  0  0;
    1  -5  0  0  0  0  0  0  0]

 b = [100 0 0 0 0 0 0 0 0]'

 x = A\b
```

```
» neraca
A =
  Columns 1 through 7

    1.0000         0         0         0   -1.0000         0
0
```

```
         0    1.0000         0         0         0   -1.0000
0
   -1.0000         0    1.0000         0         0         0
0
         0   -1.0000         0    1.0000         0         0
0
         0         0   -1.0000         0    1.0000         0
1.0000
         0         0         0   -1.0000         0    1.0000
0
         0         0   -0.8000         0    1.0000         0
0
         0         0         0   -0.4000         0    1.0000
0
    1.0000   -5.0000         0         0         0         0
0

  Columns 8 through 9

         0         0
         0         0
         0    1.0000
         0   -1.0000
         0         0
    1.0000         0
         0         0
         0         0
         0         0

b =
   100
     0
     0
     0
     0
     0
     0
     0
     0
x =
  227.2727
   45.4545
  159.0909
  113.6364
  127.2727
   45.4545
   31.8182
   68.1818
   68.1818
```

Latihan:

Xylene (1), styrene (2), toluene (3) dan benzene (4) akan dipisahkan menggunakan 3 buah menara distilasi sebagaimana ditunjukkan oleh gambar di bawah. Masing-masing yaitu : F, D, B, D1, B1, D2, dan B2 adalah laju alir molar dalam mol/menit.



Diketahui :	F	= 70 mol/menit		
	$x_{f,1}$ = 0.15	; $x_{f,2}$ = 0.25	; $x_{f,3}$ = 0.40	; $x_{f,4}$ = 0.20
	$x_{d1,1}$ = 0.07	; $x_{d1,2}$ = 0.04	; $x_{d1,3}$ = 0.54	; $x_{d1,4}$ = 0.35
	$x_{b1,1}$ = 0.18	; $x_{b1,2}$ = 0.24	; $x_{b1,3}$ = 0.42	; $x_{b1,4}$ = 0.16
	$x_{d2,1}$ = 0.15	; $x_{d2,2}$ = 0.10	; $x_{d2,3}$ = 0.54	; $x_{d2,4}$ = 0.21
	$x_{b2,1}$ = 0.24	; $x_{b2,2}$ = 0.65	; $x_{b2,3}$ = 0.10	; $x_{b2,4}$ = 0.01

(a) Hitung laju alir molar untuk aliran D1, D2, B1 dan B2 ?

(b) Hitung laju alir molar dan komposisi aliran D dan B ?

Petunjuk Penyelesaian :

- Neraca massa komponen untuk keseluruhan rangkaian MD

$$x_{d1,1}D_1 + x_{b1,1}B_1 + x_{d2,1}D_2 + x_{b2,1}B_2 = x_{f,1}F$$

$$x_{d1,2}D_1 + x_{b1,2}B_1 + x_{d2,2}D_2 + x_{b2,2}B_2 = x_{f,2}F$$

$$x_{d1,3}D_1 + x_{b1,3}B_1 + x_{d2,3}D_2 + x_{b2,3}B_2 = x_{f,3}F$$

$$x_{d1,4}D_1 + x_{b1,4}B_1 + x_{d2,4}D_2 + x_{b2,4}B_2 = x_{f,4}F$$

- Neraca massa overall dan komponen untuk MD-02

$$D = D_1 + B_1$$

$$x_{d,1}D = x_{d1,1}D_1 + x_{b1,1}B_1$$

$$x_{d,2}D = x_{d1,2}D_1 + x_{b1,2}B_1$$

$$x_{d,3}D = x_{d1,3}D_1 + x_{b1,3}B_1$$

$$x_{d,4}D = x_{d1,4}D_1 + x_{b1,4}B_1$$

- Neraca massa overall dan komponen untuk MD-03

$$B = D_2 + B_2$$

$$x_{b,1}B = x_{d2,1}D_2 + x_{b2,1}B_2$$

$$x_{b,2}B = x_{d2,2}D_2 + x_{b2,2}B_2$$

$$x_{b,3}B = x_{d2,3}D_2 + x_{b2,3}B_2$$

$$x_{b,4}B = x_{d2,4}D_2 + x_{b2,4}B_2$$

BAB 4 Penyelesaian Persamaan Non-Linier

Dalam bidang teknik Kimia sering dijumpai persoalan mencari akar persamaan non linier

f(x) = 0

yang sulit diselesaikan dengan manipulasi matematika analitis. Contoh Persamaan-Persamaan Tak Linier

Jenis Pers. Tak Linier	**Contoh**
Persamaan Kuadrat	$\boxed{x^2 - 4x + 3 = 0}$
Persamaan Polinomial	$\boxed{x^4 + 6x^3 + 7x^2 - 6x - 8 = 0}$
Persamaan Transenden	$\boxed{\sin x - 2\exp(-x^2) = 0}$
Persamaan Logaritmik	$\boxed{\ln(1 + x^2) - 2\exp(-x^2) = 0}$

Matlab menyediakan fasilitas untuk menyelesaikan jenis persamaan-persamaan di atas yang telah tersusun dalam fungsi yaitu ‘fzero’.

Syntax yang digunakan untuk menuliskan fzero adalah

```
z = (‘fzero’,initial guess)
```

Contoh 4.1 Mencari akar persamaan.
Diketahui persamaan :

$$f(x) = x^3 - 2x - 5,$$

akan dicari nilai x yang menyebabkan fungsi f(x) sama dengan nol.

Penyelesaian:
tulis dalam M-file

```
function y = f(x)
y = x.^3-2*x-5;
```

Untuk mendapatkan nol mendekati 2, tuliskan :

```
z = fzero(‘f’,2)
z =
      2.0946
```

Contoh 4.2 Mencari temperatur untuk suatu harga Cp tertentu (diambil dari *Computational Methods for Process Simulation* ", Ramirez, Butterworths, 1989)
Diketahui sebuah persamaan kapasitas panas sbb. :

$$Cp = 0.716 - 4.257E^{-6}T - \frac{15.04}{\sqrt{T}} \quad \left[\frac{kJ}{kg.K}\right] \quad \text{dan T dalam K} \qquad \text{(i)}$$

Akan ditentukan temperatur pada saat Cp = 1 kJ/kg.K. Untuk itu, ubahlah persamaan di atas menjadi :

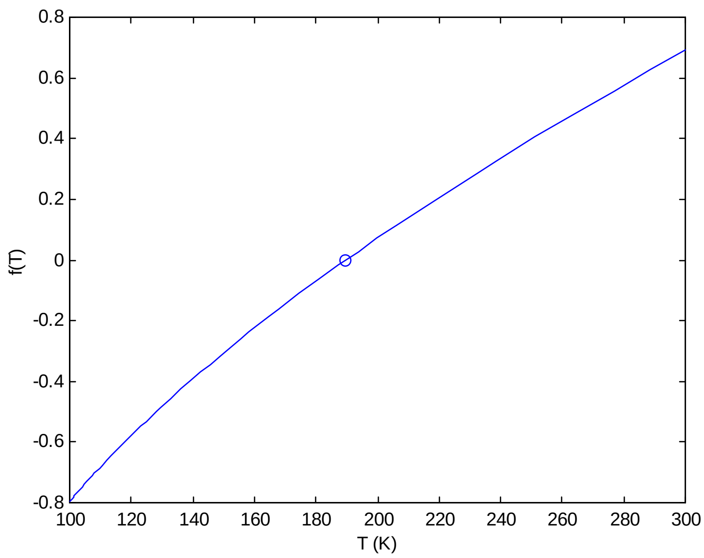
$$f(T) = 1 - 0.716 + 4.257E^{-6}T + \frac{15.04}{\sqrt{T}} = 0 \qquad \text{(ii)}$$

Penyelesaian :

Tahap 1 : membuat fungsi yang dapat mengevaluasi persamaan (ii)

```
function f = fungsi(T)
% fungsi yang akan di-nol-kan.

f = 1 - 0.716 + 4257E-6*T - 15.04/sqrt(T);
```

Apabila fungsi ini diplot (fplot('fungsi',[100 300]) akan diperoleh grafik sbb. :



Untuk mendapatkan harga penol dari fungsi tersebut digunakan fungsi **fzero** dengan tebakan awal 100 :

```
» fzero('fungsi',100)

ans =
  189.7597
```

Diperoleh T = 189.7597 K pada saat Cp = 1 kJ/kg.K.

Latihan

Diketahui persamaan Van Der Waals sebagai berikut:

$$\left(P + \frac{a}{\upsilon^2}\right)(\upsilon - b) = RT \qquad (1A.1)$$

$$a = \frac{27}{64}\left(\frac{R^2T_c^2}{P_c}\right) \qquad (1A.2)$$

$$b = \frac{RT_c}{8P_c} \qquad (1A.3)$$

Dimana :	v	= volum molar, L/mol
	T	= suhu, K
	R	= konstanta gas universal = 0,08206 atm.L/mol.K
	Tc	= suhu kritis, K (405,5 K untuk Amonia)
	Pc	= tekanan kritis, atm (111,3 atm untuk Amonia)

Diketahui : Tekanan reduksi

$$P_r = \frac{P}{P_c} \qquad (1A.4)$$

Faktor kompresibilitas

$$Z = \frac{P\upsilon}{RT} \qquad (1A.5)$$

(a) Hitung volum molar dan faktor kompresibilitas untuk gas amonia pada P = 56 atm dan suhu = 450 K dengan menggunakan persamaan keadaan Van Der Waals?

(b) Ulangi perhitungan untuk tekanan reduksi berikut : Pr = 1, 2, 4, 10, dan 20 !

(c) Bagaimana hubungan antara faktor kompresibilitas dan tekanan reduksi (gambarkan dalam sebuah grafik) ?

Petunjuk Penyelesaian

Persamaan (1A.1) perlu disusun kembali sehingga menjadi bentuk :

$$F(\upsilon) = P\upsilon^3 - (Pb + RT)\upsilon^2 + a\upsilon - ab \quad (1A.6)$$

Untuk mendapatkan nilai v maka:

F(v) = 0 (1A.7)

BAB 5 Penyelesaian Persamaan Non-Linier Simultan

Bentuk umum persamaan non-linier simultan adalah:

$f(1) = f_1(x_1, x_2, x_3, ..., x_n) = 0$
$f(2) = f_2(x_1, x_2, x_3, ..., x_n) = 0$
$f(3) = f_3(x_1, x_2, x_3, ..., x_n) = 0$
....
$f(n) = f_1(x_1, x_2, x_3, ..., x_n) = 0$

Dalam hal ini akan dicari harga $x_1, x_2, \ldots, x_n$. Dalam MATLAB, jenis persamaan di atas dapat diselesaikan dengan menggunakan fasilitas yang sudah tersedia yaitu fungsi *fsolve.*

Contoh 5.1

$$x^3 - 3xy^2 = 1/2$$
$$3x^2y - y^3 = \sqrt{3}/2$$

Langkah 1 Buat terlebih dahulu fungsi sistem persamaan taklinier dalam m-file.
Langkah 2 Buat program pengeksekusi menggunakan fsolve pada m-file yang berbeda atau dapat juga langsung di command window.
Langkah 3 Jalankan program pengeksekusi.

```
function f = sistem(x)
f=[x(1)^3-3*x(1)*x(2)^2-0.5
    3*x(1)^2*x(2)-x(2)^3-sqrt(3)/2]
```

```
>>[x,fval] = fsolve('sistem',[1 2])
Optimization terminated: first-order
optimality is less than options.TolFun.

x =

    2.5198    1.5874

fval =

  1.0e-010 *

    0.1930
    0.0966
```

Contoh 5.2

Reaksi reformasi kukus berlangsung menurut rangkaian reaksi kesetimbangan berikut:

$CH_{4(g)} + H_2O_{(g)} \rightleftharpoons CO_{(g)} + H_{2\,(g)}$ R-1

$CO_{(g)} + H_2O_{(g)} \rightleftharpoons CO_{2(g)} + H_{2\,(g)}$ R-2

Pada suhu 2000 K harga konstanta kesetimbangan untuk masing-masing reaksi adalah 1,930x10^{-4} dan 5,528. Tentukan komposisi kesetimbangan komponen-komponen apabila Gas umpan berkomposisi 20% $CH_{4(g)}$ dan 80% $H_2O_{(g)}$ berada pada kondisi suhu 2000 K dan tekanan 1 atm.

Jawaban

Misal ditetapkan basis perhitungan 10 mol gas umpan

$e_1 =$ derajat reaksi dari reaksi pertama

$e_2 =$ derajat reaksi dari reaksi kedua

Fraksi mol kesetimbangan setiap komponen dapat dinyatakan sebagai berikut:

$$Y_{CO} = \frac{e_1 - e_2}{10 + 2e_1} \qquad Y_{H_2} = \frac{3e_1 + e_2}{10 + 2e_1} \qquad Y_{H_2O} = \frac{8 - e_1 - e_2}{10 + 2e_1}$$

$$Y_{CO_2} = \frac{e_2}{10 + 2e_1} \quad Y_{CH_4} = \frac{2 - e_1}{10 + 2e_1}$$

Persamaan konstanta kesetimbangan dinyatakan sebagai berikut:

$$K_1 = \frac{Y_{CO} Y_{H_2}^3 P^2}{Y_{CH_4} Y_{H_2O}} \qquad K_2 = \frac{Y_{CO_2} Y_{H_2}}{Y_{CO} Y_{H_2O}}$$

$$\left.\begin{array}{r} \boxed{\dfrac{(e_1 - e_2)(3e_1 - e_2)^3}{(2 - e_1)(8 - e_1 - e_2)(10 + 2e_1)^2} = K_1} \\[2ex] \boxed{\dfrac{e_2 (3e_1 + e_2)}{(e_1 - e_2)(8 - e_1 - e_2)} = K_2} \end{array}\right\} \text{Sistem persamaan tak linier}$$

Berikut ini pemrograman MATLAB-nya.

```
function y = KsT(e,K1,K2)
%Sistem Pers.tak linier yang akan dinolkan
y = [(e(1)-e(2))*(3*e(1)-e(2))^3 /((2-e(1))*(8-e(1)…
       - e(2))*(10+2*e(1))^2) - K1
        e(2)*(3*e(1)+e(2)) / ((e(1)-e(2))*(8-e(1)-e(2))) - K2];
```

```
clear
clc
K1 = input(‘Masukan konstanta kst. reaksi 1 = ');
K2 = input(‘Masukan konstanta kst. reaksi 2 = ');

%Pencari nol fungsi KsT.m
e  = fsolve(@(e) KsT(e,K1,K2),[1 0.5])
```

Eksekusi di MATLAB *command window*

```
>>contoh2
Masukan harga konstanta kst. reaksi 1 = 1.93e-4
Masukan harga konstanta kst. reaksi 2 = 5.528
Optimization terminated: first-order optimality is less than
     options.TolFun.

e =
      0.7480    0.6920
```

Latihan

Beberapa reaksi setimbang berlangsung dalam reaktor batch (volume konstan) dengan persamaan reaksi sebagai berikut :

$$A + B \Leftrightarrow C + D$$

$$B + C \Leftrightarrow X + Y$$

$$A + X \Leftrightarrow Z$$

Tentukan konsentrasi masing-masing komponen pada saat setimbang jika diketahui C_{A0} = C_{B0} = 1.5, K_1 = 1.06, K_2 = 2.63 dan K_3 = 5. Lakukan perhitungan dengan estimasi nilai awal sebagai berikut :

(a) $C_D = C_X = C_Z = 0$
(b) $C_D = C_X = C_Z = 1$
(c) $C_D = C_X = C_Z = 10$

Petunjuk Penyelesaian :

Sistem persamaan aljabar menggambarkan kesetimbangan rekasi di atas.
Hubungan kesetimbangan non-linier menggunakan kesetimbangan termodinamika dan hubungan linier diperoleh dari stoikiometri reaksi.

$$K_{C1} = \frac{C_C C_D}{C_A C_B} \qquad K_{C2} = \frac{C_C C_D}{C_A C_B} \qquad K_{C3} = \frac{C_C C_D}{C_A C_B}$$

$C_A = C_{AO} – C_D$ - C_Z, $C_B = C_{BO} – C_D – C_Y$

$C_C = C_D – C_Y$, $C_Y = C_X + C_Z$

Pada sperangkat persamaan ini, C_A, C_B, C_C, C_D, C_X, C_Y dan C_Z adalah konsentrasi berbagai komponen pada kesetimbngan yang dihasilkan dari konsentrasi awal hanya C_{AO} dan C_{BO}. Konstanta kesetimbangan K_{C1}, K_{C2} dan K_{C3} telah diketahui nilainya.

BAB 6 Penyelesaian Persamaan Diferensial Biasa

Definisi PDB

Persamaan diferensial biasa adalah persamaan diferensial yang terdiri atas fungsi turunan satu buah variabel bebas.

Contoh:

Persamaan gaya geser (*shear stress*) pada aliran fluida dirumuskan sbb.

$$\boxed{\frac{d\tau_{xz}}{dx} = \rho g}$$

Perhatikan PDB hanya memiliki satu buah variabel bebas yaitu x dan satu variabel terikat yaitu τ_{xz}.

Aplikasi PDB

PDB banyak ditemukan pada pemodelan-pemodelan teknik reaktor, kinetika reaksi kimia, peristiwa-peristiwa perpindahan dll.

Klasifikasi PDB

Berdasarkan ordenya PDB terdiri atas tiga jenis (paling umum ditemukan dalam permasalahan teknik kimia).

Orde 1 $\quad \frac{dy}{dx} + y = kx$

Orde 2 $\quad \frac{d^2y}{dx^2} + y\frac{dy}{dx} = kx$

Orde 3 $\quad \frac{d^3y}{dx^3} + a\frac{d^2y}{dx^2} + b\left(\frac{dy}{dx}\right)^2 = kx$

Berdasarkan ordenya PDB terdiri atas dua jenis.

1. Linier

   Persamaan umum PDB linier dirumuskan sbb:

$$b_o(x)\frac{d^ny}{dx^n} + b_1(x)\frac{d^{n-1}y}{dx^{n-1}} + \ldots + b_{n-1}(x)\frac{dy}{dx} + b_n(x)y = R(x)$$

2. Taklinier

   PDB yang tidak memenuhi persamaan umum PDB linier di muka dikelompokan ke dalam PDB tak linier.

Salah satu kegunaan MATLAB dalam teknik adalah aplikasinya untuk meneyelesaikan persamaan secara numeris persamaan diferensial biasa. MATLAB memiliki penyelesaian ode yang berbeda yang memungkinkan ode menyelesaikan secara akurat dan efisien tergantung pada tingkat kesulitan (stiffness) ode. Stiffness adalah perubahan relative pada penyelesaian satu persamaan diferensial.

Terdapat cara berbeda untuk menyusun dan mengeksekusi penyelesaian ode, namun untuk kali ini suatu system yang menggunkanan m-files banyak untuk setap penyelesaian ode akan diberikan. Dua m-files utama yang diperlukan adalah file eksekusi (run) dan ile fungsi. Untuk penyelesaian sutu ode dlam MATLAB semua ode harus didefinisikan dalam suatu fungsi m-file. Ketika memasukkan kedalam file fungsi, persamaan diferensial harus memiliki order satu berbentuk *dy/dx = f(y,x).* File fungsi harus berisi:

1. Definisi fungsi seperti `function dmdt=nama_file(t,m)`, dimana t adalah variable bebas dan m adalah variable tak bebas order satu.
2. Jika variabel global digunakan,perintah global harus disisipkan setelah definisi fungsi
3. Persamaan diferensial harus dalam bentuk deskripsi di atas, misal: `dmdt=f(m,t)`

Nama file, variable (m dan t), dan dmdt dapat berubah-ubah.

Contoh 6.1

Suatu fluida dengan densitas tetap mengalir ke dalam tangki besar yang kosong dan tak tentu pada 8 L/s. Sebuah kran dipasang untuk mengatur aliran keluar pada laju tetap 4 L/s. Turunkan dan selesaikan persamaan diferensial yang menggambarkan proses ini, di atas interval 100 detik.

Penyelesaian :

Neraca massa:

Laju Akumulasi = input – Output

$$\frac{d(\rho V)}{dt} = (8-4)\rho$$

Karena densitas konstan, sehingga

$$\frac{d(V)}{dt} = (8-4)$$

dalam liter per detik.

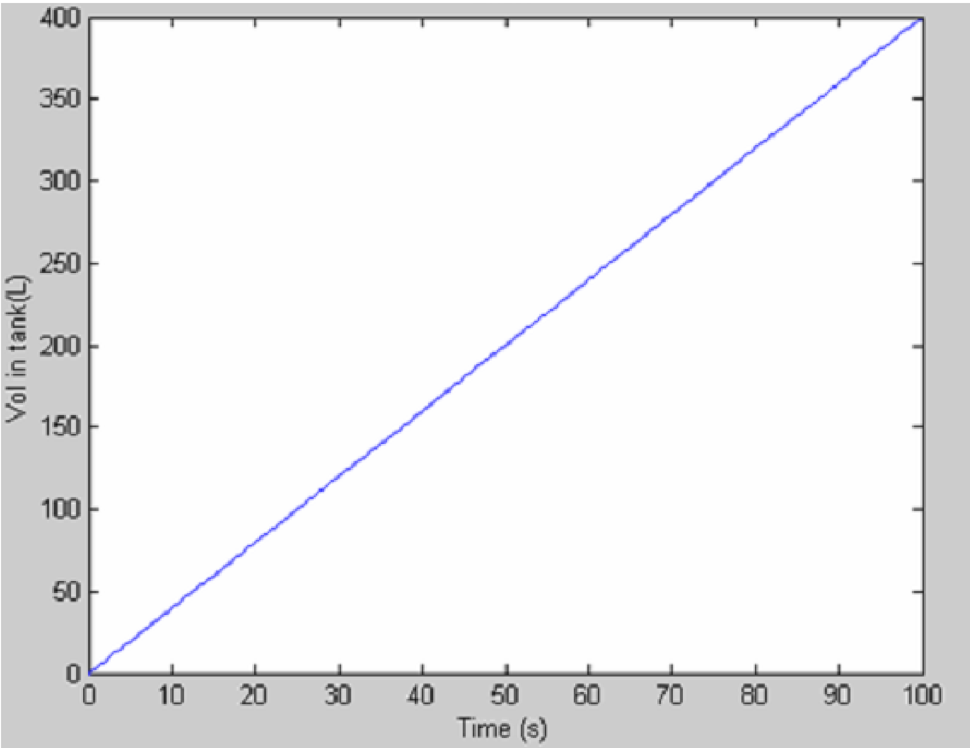
Kondisi awal pada waktu t = 0, volume dalam tangki = 0. Berikut penyelesaan persamaan di atas.

```
function dvdt = fluida(t,v)
dvdt=4
```

File fluida run digunakan untuk mengeksekusi penyelesaian. List penulisan program adalah

```
to=0;
tf=100;
tspan=[to tf];  %interval integrasi
v0=0            %kondisi awal
[t,v]=ode45('fluida'tspan,v0)
Plot(t,v(:,1))
Xlabel('Time (s)')
Ylabel ('vol in tank(L)')
Title('fluida')
```

Grafik yang dihasilkan adalah:



Contoh 6.2 Simulasi Reaktor Batch

Reaktor batch adalah reaktor yang digunakan secara sekali tempuh. Artinya umpan dimasukkan satu kali di awal reaksi dan produk dikeluarkan pada akhir reaksi. Selama reaksi tidak ada umpan yang masuk ataupun produk yang keluar.

Kecepatan proses biasanya diukur dari kecepatan pengurangan umpan :

$$-\frac{dC_A}{dt} = -r_A$$

Apabila kecepatan reaksi dapat didefinisikan sebagai :

$$-r_A = \frac{kC_A}{K + C_A}$$

maka persamaan diferensial di atas menjadi :

$$-\frac{dC_A}{dt} = \frac{kC_A}{K + C_A}$$

Jika harga k = 0.01, K = 1.03 dan C_A pada t = 0 (awal reaksi) adalah 0.5 mol/liter, maka konsentrasi A setiap waktu dapat ditentukan dengan menyelesaikan persamaan diferensial di atas.
Catatan : persamaan tersebut tidak dapat diselesaikan secara analitik karena bersifat tak linier, sehingga harus diselesaikan secara numerik.

Langkah pertama, buatlah fungsi yang dapat mengevaluasi fungsi ruas kanan persamaan diferensial tsb.

```
function yprime = furuka(t,Ca)
% menghitung fungsi ruas kanan dari persamaan neraca massa reaktor
batch

yprime = 0.1*Ca/(1.03 + Ca);
```

Kemudian langkah berikutnya adalah menggunakan fungsi ode23 atau ode45 yang telah disediakan Matlab untuk menentukan konsentrasi A setiap waktu.

```
[t,Ca] = ode23('furuka',[0 50],0.5)

t =

         0
    1.2240
    6.2240
   11.2240
   16.2240
   21.2240
   26.0582
   30.4158
   34.4216
   38.2175
   41.8888
   45.4822
   49.0247
   50.0000
```

```
Ca =

    0.5000
    0.4611
    0.3240
    0.2203
    0.1456
    0.0941
    0.0607
    0.0405
    0.0277
    0.0193
    0.0136
    0.0096
    0.0068
    0.0062
```

Apabila harga k bergantung pada temperatur menurut persamaan Arhenius berikut :

$$k = 10^8 e^{-\frac{95.000}{RT}}$$

dan reaksi yang terjadi bersifat eksotermik (ΔH = 432 kJ/mol) dan dilaksanakan pada temperatur 500 K, maka persamaan-persamaan neraca massa dan energi sistem adalah sbb. :

$$-\frac{dC_A}{dt} = \frac{10^8 e^{-\frac{95.000}{RT}} C_A}{K + C_A}$$

$$\rho Cp \frac{dT}{dt} = \frac{10^8 e^{-\frac{95.000}{RT}} C_A}{K + C_A}(\Delta H)$$

Jika ρ dan Cp dianggap sama dengan air (1000 kg/m^3 dan 4.3 kJ/kg), maka fungsi furuka menjadi :

```
function yprime = furuka(t,y)
% menghitung fungsi ruas kanan dari persamaan neraca massa reaktor
batch
Ca = y(1);
T = y(2);
yprime = zeros(2,1);
yprime(1) = - 1E8*exp(-95000/(8.314*T))*Ca/(1.03 + Ca);
yprime(2) = 1E8*exp(-95000/(8.314*T))*Ca/(1.03 +
Ca)*(432*1000)/(1000*4.3);
```

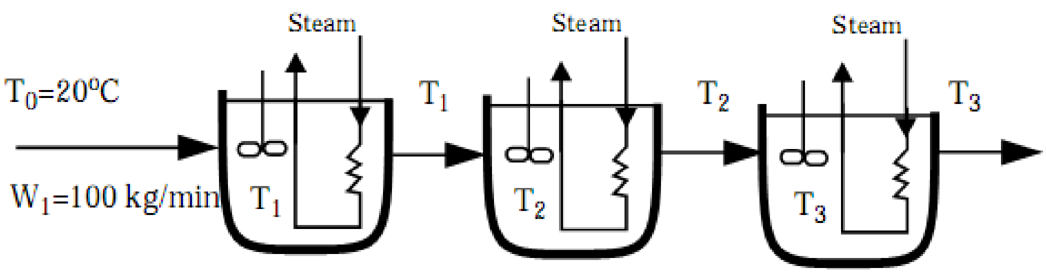
Kemudian integrasikan dengan batas yang sama :

Kemudian integrasikan dengan batas yang sama :

```
[t,Y] = ode23('furuka',[0 50],[0.5 500])

t =

     0
     5
    10
    15
    20
    25
    30
    35
    40
    45
    50


Y =

    0.5000  500.0000
    0.4799  502.0147
    0.4585  504.1645
    0.4357  506.4621
    0.4112  508.9204
    0.3850  511.5520
    0.3570  514.3673
    0.3271  517.3723
    0.2953  520.5644
    0.2618  523.9267
    0.2271  527.4201
```

Latihan

Tiga buah tangki yang disusun seri digunakan untuk memanaskan minyak mentah sebelum diumpankan ke fraksinator untuk pemisahan lanjut.



Pada saat awal, masing-masing tangki diisi dengan 1000 kg minyak pada suhu 20 ^{o}C. Steam jenuh pada suhu 250 ^{o}C dikondensasikan di dalam koil yang tercelup pada masing-masing tangki. Minyak diumpankan ke tangki pertama dengan laju 100 kg/menit dan dialirkan ke tangki kedua maupun tangki dengan laju yang sama. Suhu minyak umpan adalah 20 ^{o}C. Tangki dilengkapi pengaduk sehingga pencampuran di dalam tangki dapat dianggap sempurna, dan suhu di dalam tangki seragam. Demikian juga dengan suhu aliran keluar tangki sama dengan suhu di dalam tangki. Kapasitas panas minyak, Cp = 2.0 kJ/kg. Laju perpindahan panas dari steam ke minyak tiap tangki dinyatakan dengan persamaan sebagai berikut :

$$Q = Ua(T_{steam} - T)$$

Dimana Ua = 10 kJ/mnt.^{o}C yaitu perkalian antara koefisien transfer panas dan luas area perpindahan panas koil untuk masing-masing tangki.

Tentukan suhu steady state di tiap tangki, dan berapa interval waktu yang dibutuhkan agar T_3 mencapai 99 % kondisi steady state-nya pada saat start-up ?

Petunjuk Penyelesaian :

- Asumsi :
  i. Laju alir minyak menuju masing – masing tangki dianggap sama ($W_0 = W_1 = W_2 = W_3 = W$).
  ii. Densitas minyak konstan, sehingga jumlah (massa dan volum) minyak di dalam masing – masing tangki sama dan konstan ($M_1 = M_2 = M_3 = M$).

- Susun neraca panas unsteady state masing – masing tangki.

  Untuk tangki 1 :

  Panas Akumulasi = Panas masuk – Panas keluar

$$MC_p \frac{dT_1}{dt} = WC_pT_0 + Ua(T_{steam} - T_1) - WC_pT_1$$

  Persamaan di atas dapat disusun kembali sebagai berikut :

$$\frac{dT_1}{dt} = \frac{WC_p(T_0 - T_1) + Ua(T_{steam} - T_1)}{MC_p}$$

  Analog untuk tangki 2 :

$$\frac{dT_2}{dt} = \frac{WC_p(T_1 - T_2) + Ua(T_{steam} - T_2)}{MC_p}$$

  Untuk tangki 3 :

$$\frac{dT_3}{dt} = \frac{WC_p(T_2 - T_3) + Ua(T_{steam} - T_3)}{MC_p}$$

BAB 7 Optimisasi Fungsi

Optimisasi adalah usaha mendapatkan suatu keadaan dimana objektif dari keadaan tersebut maksimum/minimum. Metoda numerik untuk mendapatkan nilai optimum dapat digunakan untuk berbagai keperluan.Untuk mencari harga minimum dan maksimum kita dapat menggunakan perintah fminsearch. Berikut ini cara penulisannya.

```
[x,fval,exitflag] = fminsearch(fun,x0)
```

keterangan:

fun = Fungsi yang akan diminimumkan atau dimaksimumkan

xo = Tebakan awal

x = Harga x yang menyebabkan fungsi minimum atau maksimum

fval = Nilai maksimum atau minimum.

exitflag = Kriteria penghentian proses iterasi.

fminsearch mencapai kekonvergenan pada satu nilai x. Harga ini yang kita cari.

Contoh 7.1

Optimasi Variabel Tunggal

Carilah titik minimum $x^2 + 4x + 3 = 0$ dengan menggunakan subrutin fiminsearch dalam MATLAB.

Penyelesaian;

File fungsi:

```
kuadrat=x^2+4*x+3
```

File eksekusi:

```
>> [x,fval,exitflag]=fminsearch('kuadrat',2)

x =
   -2.0000
fval =
    -1
exitflag =
     1
```

Contoh 7.2

Optimasi Variabel Jamak

Carilah titik minimum dari persamaan multivariabel berikut ini.

$$y = (x_1 - 3)^2 + 0.5(x_2 - 4)^2 + 3$$

Penyelesaian:

```
%multivaribel.m
function y = multivariabel(x);

y = (x(1)-3)^2 + 0.5*(x(2)-4)^2 + 3;
```

```
%kasus12
[x,fval] = fminsearch('multivariabel',[1,16])
```

```
>> kasus12
x =
    3.0000    4.0000
```

Contoh 7.3 Regresi dengan meminimumkan kuadrat terkecil

Sekumpulan data percobaan disajikan pada tabel berikut ini (file optim.dat) :

```
0      :8.132429853
0.1    :6.740876506
0.2    :5.818350842
0.3    :4.745293322
0.4    :4.264413716
0.5    :3.752775897
0.6    :3.188875704
0.7    :2.359092343
0.8    :2.107375147
0.9    :2.097267258
1      :1.897768103
1.1    :1.67504687
1.2    :1.348539115
1.3    :1.144961449
1.4    :1.124253306
1.5    :0.906495819
1.6    :0.504671754
1.7    :0.670491443
1.8    :0.559044565
1.9    :0.726153496
2      :0.657388383
```

Data ini memiliki model matematik sbb. :

$$f(x) = \alpha_1 e^{\beta_1 x} + \alpha_2 e^{\beta_2 x}$$

Untuk menentukan parameter-parameter α_1, α_2, β_1, dan β_2, dapat dilakukan tahapan berikut :

1. Buatlah fungsi pengevaluasi f(x).

```
function f = model(x,alpha,beta);
% fungsi model dari data

f = alpha(1)*exp(beta(1)*x) + alpha(2)*exp(beta(2)*x);
```

2. Fungsi berikut ini akan dibandingkan dengan data untuk harga alpha dan beta tertentu.

```
function phi = banding(x)
% fungsi yang membandingkan data dengan model

global xd yd

alpha = x(1:2);
beta = x(3:4);
yc = model(xd,alpha,beta);
phi = norm(yc - yd);
```

3. Terakhir, buat script sebagai program utamanya.

```
%main program optim.m
clear
global xd yd
M = dlmread('optim.dat',':');
xd = M(:,1); yd = M(:,2);

% tembakan awal alpha [5 3], beta [-2 -1]
x0 = [5 3 -2 -1];
xf = fmins('banding',x0)
```

Hasil perhitungan adalah sbb. :

```
» optim
xf =
    4.4279    3.6638   -2.3275   -1.0246
```

Latihan :

Tabel berikut adalah data tekanan uap Benzen pada berbagai suhu.

Suhu T (^{o}C)	Tekanan P (mmHg)
-36.7	1
-19.6	5
-11.5	10
-2.6	20
7.6	40

15.4	60
26.1	100
42.2	200
60.6	400
80.1	760

(a) Plot data-data tersebut ke dalam persamaan polinomial order 4 !

(b) Plot data-data tersebut ke dalam persamaan Clausius-Clapeyron !

(c) Plot data-data tersebut ke dalam persamaan Antoine !

Gambarkan hubungan antara data dan nilai terhitung masing-masing dalam suatu grafik.

Referensi:

Cutlip, M. B., Shacham, M., 1999, *Problem Solving in Chemical Engineering with Numerical Methods*. Prentice-Hall: Englewood Cfiffs, NJ.

Finlayson, B.A., 2006, *Introduction to Chemical Engineering Computing*, John Wiley & Sons Inc., New Jersey

http://www.mathworks.com,