

Pengantar  
**Komputasi Teknik Kimia**

**DENGAN**

**MATLAB** **DAN**  
**SIMULINK**

Hak cipta pada penulis  
Hak penerbitan pada penerbit  
Tidak boleh diproduksi sebagian atau seluruhnya dalam bentuk apapun  
Tanpa izin tertulis dari pengarang dan/atau penerbit

**Kutipan Pasal 72 :**

Sanksi pelanggaran Undang-undang Hak Cipta (UU No. 10 Tahun 2012)

1. Barang siapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal (49) ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp. 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan atau denda paling banyak Rp. 5.000.000.000,00 (lima miliar rupiah)
2. Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu Ciptaan atau hasil barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah)

**HERI RUSTAMAJI, S.T., M.Eng.**

Pengantar  
**Komputasi Teknik Kimia**  
**DENGAN**  
**MATLAB** **DAN**  
**SIMULINK**

**AURA**  
ANUGRAH UTAMA RAHARJA

Perpustakaan Nasional RI:  
Katalog Dalam Terbitan (KDT)

**PENGANTAR KOMPUTASI TEKNIK KIMIA  
DENGAN MATLAB DAN SIMULINK**

**Penulis:**

HERI RUSTAMAJI, S.T., M.Eng.

**Desain Cover & Layout**

Team Aura Creative

Penerbit

**AURA**

**CV. Anugrah Utama Raharja**

**Anggota IKAPI**

**No.003/LPU/2013**

x + 280 hal : 15,5 x 23 cm

Cetakan, Juli 2017

**ISBN: 978-602-6739-14-8**

**Alamat**

Jl. Prof. Dr. Soemantri Brojonegoro, Komplek Unila

Gedongmeneng Bandar Lampung

HP. 081281430268

E-mail : [redaksiaura@gmail.com](mailto:redaksiaura@gmail.com)

Website : [www.aura-publishing.com](http://www.aura-publishing.com)

Hak Cipta dilindungi Undang-undang

# KATA PENGANTAR

---

Tujuan buku ini adalah untuk mengilustrasikan tentang (1) persoalan - persoalan dalam teknik kimia yang harus diselesaikan, (2) penyusunan model matematis persoalan teknik kimia, (3) program komputer yang digunakan untuk menyelesaikan persoalan teknik kimia secara numeris, dan (4) bagaimana mengoreksi untuk memastikan bahwa persoalan-persoalan tersebut sudah diselesaikan secara benar. Hal ini dilakukan dalam konteks bagaimana mahasiswa belajar dengan banyak menggunakan komputer. Program yang dicontohkan di sini adalah MATLAB dan Simulink. Metode penyelesaian numeris yang dijabarkan dalam buku ini adalah dengan menggunakan *built in function* yang sudah disusun dalam MATLAB.

Penjelasan lebih detail tentang metode numeris untuk menyelesaikan berbagai bentuk persamaan yang muncul dalam teknik kimia adalah di luar ruang lingkup buku ini.

Bab-bab dalam buku ini disusun berdasarkan model persamaan yang terbentuk dari persoalan-persoalan yang muncul dalam teknik kimia. Buku ini juga dapat digunakan dalam kuliah lain karena beberapa bab disusun untuk mengakomodasi kurikulum teknik kimia.

Sebagai contoh, ketika kesetimbangan reaksi kimia telah dibahas pada mata kuliah Termodinamika Teknik Kimia, pengajar dapat melaksanakan sesi latihan yang mengajarkan aplikasi komputer, menggunakan bab pada kesetimbangan reaksi kimia. Materi pada pilihan model termodinamika juga dapat ditambahkan sedikit lebih realistis terhadap kuliah termodinamika.

Bab lainnya dapat digunakan pada mata kuliah lain. Dengan cara ini mahasiswa akan menggunakan buku ini selama masa pembelajaran, dalam kuliah atau setelah kuliah Neraca Massa dan Energi, Transfer Momentum, Panas, dan Massa, Perancangan Reaktor, Optimasi Proses dan Pengendalian Proses.

Harapannya adalah bahwa mahasiswa akan mampu lebih konsentrasi pada prinsip fundamental teknik kimia dan mampu menggunakan MATLAB dan Simulink sebagai suatu *tool* untuk menyelesaikan persoalan dalam teknik kimia.

Akhir kata, Penulis mengucapkan banyak terima kasih kepada Ir. Sufriadi Burhanuddin, M.Eng. dan Ir. Azhar. M.T., yang pertama kali mengenalkan dan mengajarkan beberapa topik materi yang terkandung dalam buku ini.

HERI RUSTAMAJI

# DAFTAR ISI

---

## **BAB 1 MEMULAI MENGGUNAKAN MATLAB**

- 1.1 PENDAHULUAN\_\_\_\_1
- 1.2 MENGENAL MATLAB\_\_\_\_2
  - 1.2.1 Dekstop MATLAB\_\_\_\_3
  - 1.2.2 Ruang Kerja MATLAB\_\_\_\_3
  - 1.2.3 M-files\_\_\_\_6
    - 1.2.3.1 Aturan dan Sifat-Sifat\_\_\_\_7
    - 1.2.3.2 Syntax Untuk Membuat Fungsi\_\_\_\_8
    - 1.2.3.3 Perintah Global\_\_\_\_9
  - 1.2.4 Pemrograman MATLAB\_\_\_\_10
    - 1.2.4.1 Membuat File Program\_\_\_\_11
    - 1.2.4.2 Script : Urutan Perintah MATLAB\_\_\_\_11
    - 1.2.4.3 Fungsi : Subrutin Matlab\_\_\_\_12
    - 1.2.4.4 Format Penampilan Angka\_\_\_\_13
- 1.3 RANGKUMAN\_\_\_\_15
- 1.4 LATIHAN\_\_\_\_15
- 1.5 RUJUKAN\_\_\_\_15

## **BAB 2 KONTROL ALUR PROGRAM**

- 2.1 PENDAHULUAN\_\_\_\_16
- 2.2 PEMROGRAMAN MATLAB\_\_\_\_17
  - 2.2.1 Konstruksi if else end\_\_\_\_17

2.2.2	Loop for	18
2.2.3	Loop while	19
2.2.4	Konstruksi switch case –other wise	19
2.2.5	Operasi Vektor dan Matriks	20
2.2.6	Operasi Vektor dan Matriks pada Fungsi	24
2.3	RANGKUMAN	26
2.4	LATIHAN	27
2.5	RUJUKAN	28

### **BAB 3 VISUALISASI DATA DALAM MATLAB**

3.1	PENDAHULUAN	29
3.2	VISUALISASI GRAFIK	29
3.2.1	Pembuatan Grafik Sederhana	30
3.2.2	Grafik Multiple Plot	33
3.2.3	Grafik Tiga Dimensi	34
3.2.4	Analisis Data Percobaan	37
3.2.5	Menentukan Model dari Data Percobaan	40
3.2.6	Membuat Polinom dan Mencari Akar Polinom	43
3.3	RANGKUMAN	44
3.4	RUJUKAN	44

### **BAB 4 PERSAMAAN LINIER SIMULTAN**

4.1	PENDAHULUAN	45
4.2	MODEL MATEMATIS PERSAMAAN LINIER SIMULTAN	46
4.3	RANGKUMAN	66
4.4	RUJUKAN	66



## **BAB 5 PERSAMAAN NON LINIER TUNGGAL**

- 5.1 PENDAHULUAN\_\_\_\_68
- 5.2 MODEL MATEMATIS PERSAMAAN NON LINIER \_\_\_\_69
- 5.3 RANGKUMAN\_\_\_\_87
- 5.4 RUJUKAN\_\_\_\_88

## **BAB 6 PERSAMAAN NON LINIER SIMULTAN**

- 6.1 PENDAHULUAN\_\_\_\_89
- 6.2 PERSAMAAN NON LINIER MULTI VARIABEL \_\_\_\_90
- 6.3 RANGKUMAN\_\_\_\_107
- 6.4 LATIHAN\_\_\_\_108
- 6.5 RUJUKAN\_\_\_\_110

## **BAB 7 PERSAMAAN DIFERENSIAL BIASA**

- 7.1 PENDAHULUAN\_\_\_\_112
- 7.2 MODEL PERSAMAAN DIFERENSIAL BIASA\_\_\_\_112
- 7.3 RANGKUMAN\_\_\_\_175
- 7.4 RUJUKAN\_\_\_\_176

## **BAB 8 PERSAMAAN DIFERENSIAL PARSIAL**

- 8.1 PENDAHULUAN\_\_\_\_177
- 8.2 MODEL MATEMATIS \_\_\_\_177
- 8.3 RANGKUMAN\_\_\_\_192
- 8.4 RUJUKAN\_\_\_\_192

## **BAB 9 OPTIMASI**

- 9.1 PENDAHULUAN\_\_\_\_193
- 9.2 TOOLBOX OPTIMASI DALAM MATLAB \_\_\_\_195
- 9.3 OPTIMASI TANPA KONSTRAIN \_\_\_\_197
  - 9.3.1 Optimasi satu variabel\_\_\_\_197

9.3.2	Optimasi multivariabel	198
9.4	OPTIMASI DENGAN KONSTRAIN	203
9.5	PEMROGRAMAN LINIER	211
9.6	APLIKASI MATLAB DALAM OPTIMASI TEKNIK KIMIA	217
9.7	RUJUKAN	243

## **BAB 10 SIMULINK**

10.1	PENDAHULUAN	245
10.2	<i>SIMULINK</i> LIBRARY	245
10.3	RANGKUMAN	280
10.4	RUJUKAN	280

# **MEMULAI MENGGUNAKAN MATLAB**

## **1.1 PENDAHULUAN**

Apa itu MATLAB? Apa yang dapat dilakukan oleh MATLAB? Kemampuan apa yang dimilikinya? Bagaimana kita menggunakan MATLAB untuk menyelesaikan permasalahan-permasalahan komputasi? Mungkin kalimat tersebut adalah beberapa pertanyaan awal bagi mahasiswa yang baru pertama mengenal MATLAB. Namun bagi Anda yang sudah mengenal MATLAB pertanyaan di atas bukanlah hal yang baru dan tentu saja Anda dapat menjawabnya dengan mudah.

Pada bab ini akan dibahas mengenai MATLAB dan beberapa hal yang harus dipahami untuk menjalankan MATLAB. Setelah mengikuti kuliah ini mahasiswa diharapkan akan mampu membuka dan menjalankan MATLAB serta mampu menjelaskan fungsi dan tampilan pada MATLAB. Menguasai bab ini merupakan prasyarat untuk menguasai bab-bab berikutnya pada bahan ajar ini. Bagi mahasiswa yang ingin cepat memahami dan menguasai MATLAB maka cara yang paling mudah adalah mem-

praktekan bahan ajar ini langsung dalam lembar kerja MATLAB. Penjelasan yang lebih luas dan mendalam tentang MATLAB dapat Anda peroleh dengan membuka menu **help** pada toolbar MATLAB.

## 1.2 MENGENAL MATLAB

MATLAB (*MATrix LABoratory*) merupakan bahasa canggih untuk komputasi teknik. MATLAB merupakan integrasi dari komputasi, visualisasi dan pemrograman dalam suatu lingkungan yang mudah digunakan, karena permasalahan dan pemecahannya dinyatakan dalam notasi matematika biasa. Kegunaan MATLAB secara umum adalah untuk :

- Matematika dan komputasi
- Pengembangan dan algoritma
- Pemodelan, simulasi dan pembuatan *prototype*
- Analisa data, eksplorasi dan visualisasi
- Pembuatan aplikasi 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 pemrograman 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 dikelompokkan di dalam 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>. Untuk memulai bekerja dengan MATLAB, maka bagi mahasiswa yang baru mengenal MATLAB perlu dijelaskan tentang beberapa hal yang terkait dengan tampilan dan beberapa fungsi dalam MATLAB sebagaimana diuraikan di bawah.

### 1.2.1 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 1.1 menunjukkan desktop MATLAB.

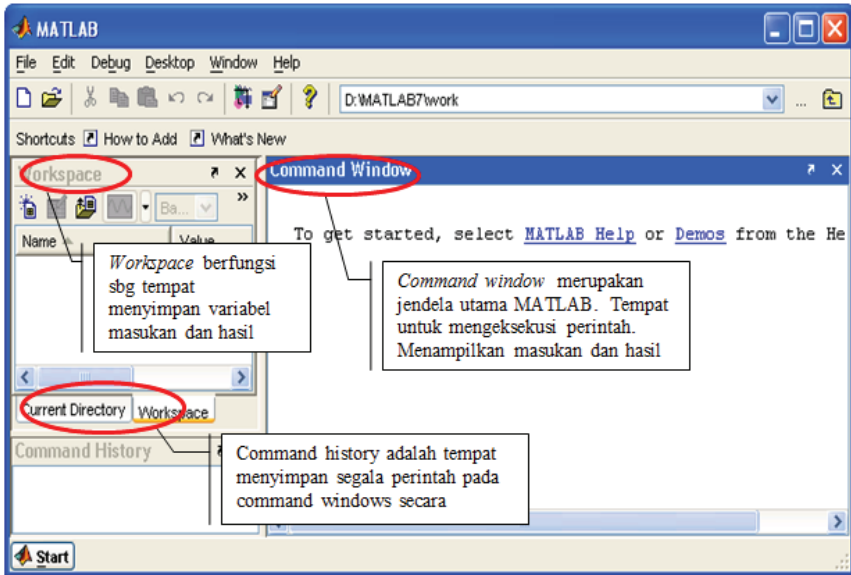
Untuk lebih jelas mengenai lingkungan kerja MATLAB perhatikan contoh pada Gambar 1.2.

### 1.2.2 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
```



**Gambar 1.1** Desktop MATLAB

Ini berarti variabel `tes` telah tersimpan di dalam ruang kerja kita. Untuk melihat data yang telah tersimpan coba anda ketik `tes` pada command window.

Jika Anda tidak dapat mengingat nama setiap variabel, 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.

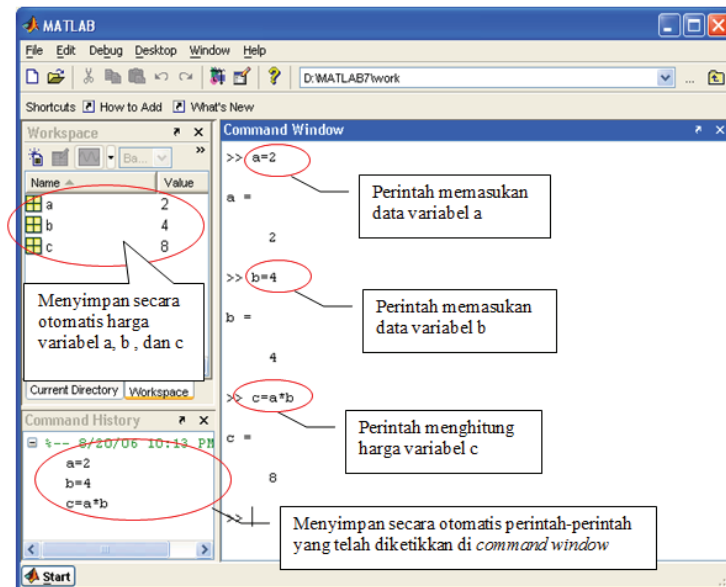
Untuk memanggil perintah sebelumnya, di MATLAB menggunakan tombol panah pada keyboard anda (  $\leftarrow$   $\uparrow$   $\downarrow$   $\rightarrow$  ).

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

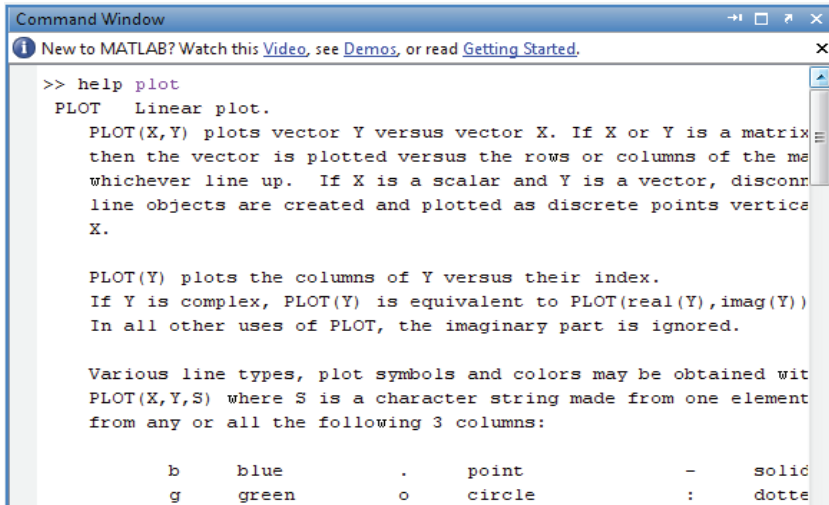
Untuk melihat keterangan dari function di MATLAB atau program yang kita buat digunakan perintah : ' **help function** ', sebagai contoh :

```
>> help plot
```

Maka akan muncul tampilan seperti Gambar 1.3.



**Gambar 1.2** Perintah dalam Command Window



**Gambar 1.3** Tampilan di command window dari help plot

### 1.2.3 M-files

Ketika Anda menggunakan fungsi-fungsi di matlab seperti *inv*, *abs*, *sqrt*, *exp*, matlab menerima variabel yang anda berikan, menghitung hasil berdasarkan input dan mengembalikan hasil akhirnya pada Anda sendiri. Bila kita ibaratkan fungsi adalah sebuah kotak hitam dimana yang kita lihat hanyalah apa yang masuk dan apa yang keluar. Sifat-sifat demikian membuat fungsi dapat menjadi suatu alat yang baik untuk mengevaluasi perintah-perintah yang menggabungkan fungsi-fungsi matematika atau deretan perintah yang sering digunakan untuk memecahkan suatu masalah besar.

Untuk itu matlab menyediakan suatu struktur untuk membuat fungsi Anda sendiri dalam bentuk m-file yang disimpan dalam komputer. 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.

### 1.2. 3.1 Aturan dan Sifat-Sifat

Fungsi m-file harus mengikuti beberapa aturan dan sejumlah sifat penting. Aturan-aturan dan sifat-sifat tersebut meliputi :

- Nama fungsi dan nama file harus identik misalnya anda membuat fungsi dengan nama pangkat maka anda memberi nama m-file anda pangkat juga.
- Baris komentar sampai dengan baris bukan komentar yang pertama adalah teks help yang ditampilkan jika anda meminta help dari fungsi yang anda buat.
- Setiap fungsi mempunyai ruang kerjanya sendiri yang berbeda dengan ruang kerja MATLAB. Satu-satunya hubungan antara ruang kerja matlab dengan variabel-variabel dalam fungsi adalah variabel-variabel input dan output fungsi. Jika suatu fungsi mengubah nilai dalam suatu variabel input, perubahan itu hanya tampak dalam fungsi dan tidak mempengaruhi variabel yang ada dalam ruang kerja MATLAB. Variabel yang dibuat oleh suatu fungsi tinggal hanya dalam ruang kerja fungsi.
- Jumlah dari argumen input dan output yang digunakan jika suatu fungsi dipanggil hanya ada dalam fungsi tersebut. Fungsi ruang kerja memuat jumlah argument input. Fungsi kerja nargout memuat jumlah argumen output. Dalam praktek, variabel-variabel nargout dan nargin biasanya digunakan untuk mengatur variabel input standar dan menentukan variabel output yang diperlukan user.

### 1.2.3.2 Syntax untuk membuat fungsi

Syntax untuk membuat fungsi adalah sebagai berikut :

```
function y = nama_fungsi(x)
```

y adalah keluaran fungsi, keluaran ini dapat satu variabel atau lebih dari satu variabel, jika keluaran lebih dari satu variabel maka mempunyai bentuk sebagai berikut :

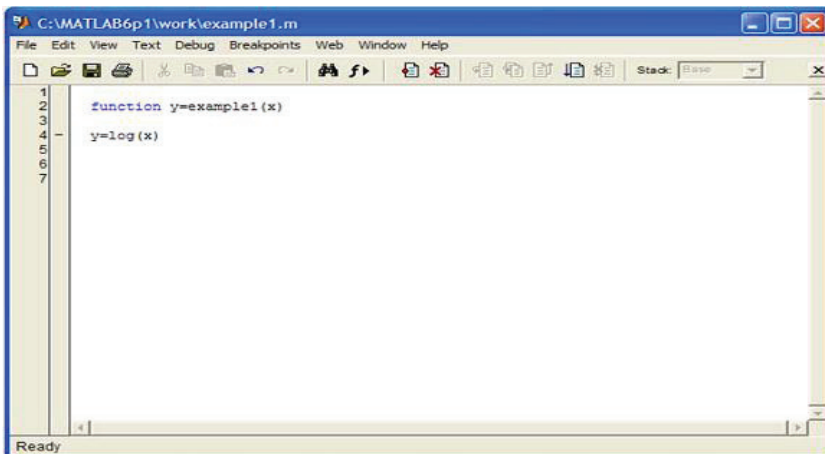
```
function [y,z,a,b] = nama_fungsi(x)
```

x adalah masukan dari fungsi, masukan ini dapat satu variabel atau lebih dari satu variabel, jika masukan lebih dari satu variabel maka mempunyai bentuk sebagai berikut :

```
function y = nama_fungsi(a,b,c,d)
```

‘nama\_fungsi’ 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 variabel yang diperlukan.

Gambar 1.4 adalah contoh suatu M-file yang digunakan untuk mengplot fungsi logaritma bilangan natural.



**Gambar 1.4** Penulisan dalam M-files

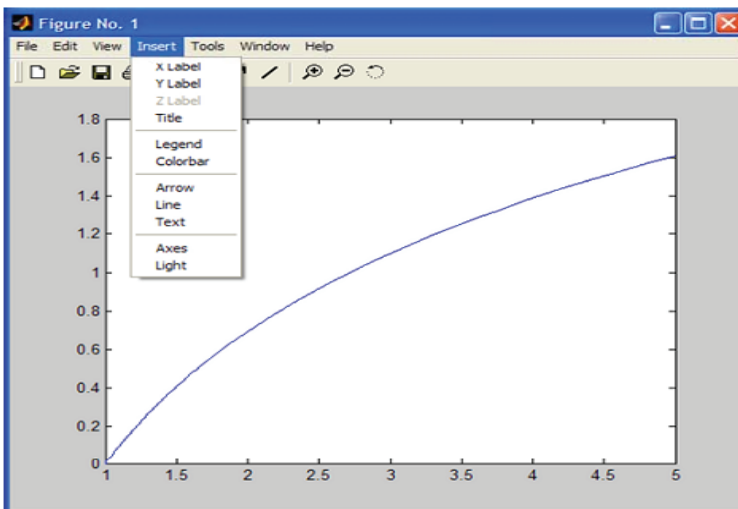
Untuk menghasilkan plot fungsi ini , berikut ini kode yang dimasukkan ke dalam command window:

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

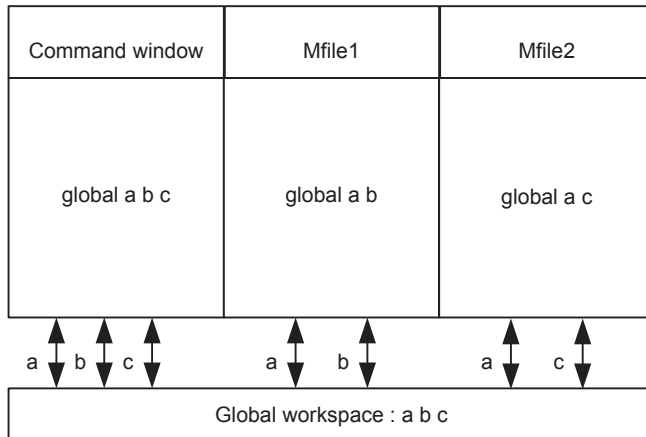
Ini akan mnghasilkan plot  $\ln(x)$  antara  $x = 1$  dan  $x = 5$ . Hasil plot ditunjukkan Gambar 1.5. Dengan menggunakan menu 'insert' kita dapat menambah judul, nama sumbu x dan y dan jika perlu *legend*.

### 1.2.3.3 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 1.6).



**Gambar 1.5** Grafik hasil fungsi example1



**Gambar 1.6** Variabel global dan lokal.

#### 1.2.4 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, menemukannya dalam lingkungan operasi MATLAB, mengeksekusinya, menangani variabel-variabel yang terlibat di dalamnya, melewati variabel melalui fungsi, dan mengatur aliran program.

### 1.2.4.1 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.

### 1.2.4.2 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.

### 1.2.4.3 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
```

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.

#### 1.2.4.4. Format Penampilan Angka

Saat MATLAB menampilkan hasil proses numerik, MATLAB mengikuti beberapa aturan. Dalam keadaan biasa, jika hasilnya berupa bilangan bulat, MATLAB menampilkannya sebagai bilangan bulat. Jika hasilnya suatu bilangan real, MATLAB akan menampilkannya sebagai bilangan dengan empat digit desimal. Jika angka penting hasilnya di luar kisaran ini, MATLAB akan menampilkannya sebagai notasi lain seperti pada kalkulator sains. Anda dapat mengatur format penampilan ini dari item *preference* pada menu File jika ada atau dengan mengetikkan perintah MATLAB pada prompt *command*. Menggunakan variabel *average\_cost*, format numerik yang tersedia adalah:

<b>MATLAB Command</b>	<b>Average_cost</b>	<b>Comments</b>
format short	50.83333	5 digits
format long	50.83333333333334	16 digits
format short e	5.0833e+01	5 digits plus eksponen
format long e	50.83333333333334e+01	16 digits plus eksponen
format short g	50.833	Better of format short or Format short e
format long g	50.83333333333333	hexadecimal
format hex	40496aaaaaaaaaab	hexadecimal
format bank	50.83	2 decimals digits
format +	50.83	Positive, negative, or zero
format rat	305/6	Rational approximation

Penting untuk diperhatikan bahwa MATLAB tidak mengubah representasi internal dari suatu bilangan sekalipun berbagai format penampilan dipilih untuk bilangan itu; hanya tampilannya saja yang berubah.



## 1.3 RANGKUMAN

- MATLAB merupakan bahasa canggih untuk komputasi teknik kegunaannya secara umum adalah untuk matematika dan komputasi, pengembangan dan algoritma, pemodelan, simulasi dan pembuatan prototype, analisa data, eksplorasi dan visualisasi, pembuatan aplikasi termasuk pembuatan *graphical user interface*.
- Perintah MATLAB dapat dituliskan langsung pada prompt command window atau dalam bentuk M-file. Penulisan pada command window dilakukan hanya untuk operasi komputasi yang sederhana. Sementara untuk operasi komputasi yang panjang dan kompleks lebih sering dituliskan dalam M-file dimana file ini dapat diedit dan dipanggil setiap saat oleh MATLAB.

## 1.4 LATIHAN

### Latihan 1.

Buatlah script penulisan dengan M-file dan pada prompt command window untuk perhitungan berikut :

$$y = 2x^2 + 5x + 8$$

Tentukan nilai  $y$  untuk setiap nilai  $x$  berikut:

$$x = 5, 6, 7 \text{ dan } 3$$

## 1.5 RUJUKAN

- 1) Dan Hanselman dan Bruce Littlefield, 1997. *MATLAB: Bahasa Komputasi dan Teknis*. hal 6-26. ANDI. Yogyakarta.
- 2) <http://www.mathworks.com>

# **KONTROL ALUR PROGRAM**

## **2.1 PENDAHULUAN**

Setelah kita membahas sekilas tentang cara kerja dan kemampuan MATLAB pada Bab 1, selanjutnya pada bab ini akan dijelaskan tentang kemampuan pemrograman MATLAB yang dijelaskan melalui beberapa konstruksi. Setelah menyelesaikan perkuliahan ini diharapkan mahasiswa akan mampu membuat alur program sederhana untuk menyelesaikan suatu permasalahan dalam bidang teknik kimia baik menggunakan konstruksi logika pemrograman maupun dengan pendekatan vector.

Bahasan ini sangat penting mengingat penyelesaian permasalahan dalam teknik kimia yang kompleks membutuhkan suatu program penyelesaian yang terkadang juga kompleks. Sehingga pemahaman terhadap logika terhadap pemrograman menjadi prasyarat untuk mampu menyelesaikan berbagai permasalahan dalam bidang teknik kimia.

Pada bagian bab ini mahasiswa diharapkan banyak berlatih membuat alur program untuk menyelesaikan soal-soal yang ada pada buku rujukan.

## 2.2 PEMROGRAMAN MATLAB

Sebagaimana bahasa pemrograman pada umumnya, MATLAB juga memiliki perintah-perintah untuk mengatur atau mengontrol alur program. Kontrol program sangat berguna karena memungkinkan komputasi-komputasi yang lalu mempengaruhi komputasi berikutnya. MATLAB menyediakan empat struktur pengambilan keputusan atau struktur kontrol program. Keempatnya adalah konstruksi if-else-end, loop for, loop while, dan konstruksi switch-case-otherwise. Konstruksi-konstruksi tersebut seringkali melibatkan banyak perintah MATLAB, yang oleh karenanya konstruksi lebih sering terdapat dalam M-file daripada diberikan langsung pada prompt MATLAB. Penjelasan singkat tentang konstruksi-konstruksi tersebut akan dijelaskan di bawah ini.

### 2.2.1 Kontruksi if – else - end

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

```

### 2.2.2 Loop 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 :

```

% program : test.m
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 ;  $x(1) = 1$ ,  $x(3) = 5.4$  dan seterusnya.

### 2.2.3 Loop while

Perintah ini akan mengulang perintah-perintah yang diapitnya selama kondisi pengujinya benar. Di dalam loop harus ada perintah-perintah yang membuat kondisi pengujian 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
```

### 2.2.4 Kontruksi switch - case - otherwise

Perintah ini mengarahkan alur program melalui sejumlah pilihan. Perintah ini untuk menggantikan perintah **if...elseif...else** yang bertumpuk.

```
% program :Pilihan.m
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

```

## 2.2.5 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 yang bersangkutan.

Secara matematik dapat dinyatakan sebagai berikut. :

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

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]  
  
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 :

$$C = AB \quad c_{ik} = a_{ij} \cdot b_{jk}$$

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 :  $x = A \setminus B$  diartikan sebagai penyelesaian

$$A * x = B$$

Pembagian kanan :  $x = A / B$  diartikan sebagai  $x =$

$$A * \text{inv}(B)$$

- **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.

- **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])

```

### 2.2.6 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 dan lainnya.

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
end
```

```

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

```

## 2.3 RANGKUMAN

- MATLAB memiliki empat konstruksi untuk mengatur alur program yang lebih sering terdapat dituliskan atau dikodekan dalam M-file daripada diberikan langsung pada prompt MATLAB.
- 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.

## 2.4 LATIHAN

### Latihan1: Menghitung Entalpi Campuran Gas

Melalui cara yang telah dibahas di atas, hitunglah entalpi suatu campuran gas CH<sub>4</sub>, H<sub>2</sub>O, H<sub>2</sub>, CO<sub>2</sub>, 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 sebagai berikut :

Gas	H <sub>f</sub>	A	B	C	D	E
CH <sub>4</sub>	-17.89	38.3870	-7.3664E-2	2.9098E-4	-2.6385E-7	8.0068E-11
H <sub>2</sub> O	-57.80	34.0471	-9.6506E-3	3.2998E-5	-2.0447E-8	4.3022E-12
H <sub>2</sub>	0.0	17.6386	6.7005E-2	-1.3148E-4	1.0588E-7	-2.9180E-11
CO <sub>2</sub>	-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 : H<sub>f</sub> dalam kkal/mol

$C_p = A + BT + CT^2 + DT^3 + ET^4$  [J/mol.K] dan T dalam K

Entalpi campuran gas dapat dihitung berdasarkan persamaan :

$$H = \sum_{i=1}^n \left( y(i) \left( H_{f,i} + \int_{T_0}^T C_{p_i} dT \right) \right)$$

## 2.5 RUJUKAN

- 1) Dan Hanselman dan Bruce Littlefield, 1997. *MATLAB: Bahasa Komputasi dan Teknis*. ANDI. Yogyakarta.
- 2) Himmelblau, D. M. 1999. *Basic Principle and Calculation in Chemical Engineering* 6<sup>th</sup> ed. Prentice hall PTR. New York.

# **VISUALISASI DATA DALAM MATLAB**

---

## **3.1 PENDAHULUAN**

Pada bab ini akan dibahas berbagai metode visualisasi data dalam MATLAB. Setelah menyelesaikan pokok bahasan bab ini mahasiswa diharapkan mampu membuat berbagai jenis grafik. Secara khusus mampu membuat grafik sederhana dua dimensi, membuat multiplot dari grafik tiga dimensi dan menentukan bentuk persamaan polinom dari sekumpulan data.

Materi ini sangat berkaitan dengan pengolahan dan visualisasi data yang biasanya diperoleh mahasiswa dari hasil penelitian. Data yang diperoleh tersebut biasanya dibuat model dan simulasi hingga diperoleh visualisasinya, sehingga materi ini akan membantu mahasiswa khususnya yang berkaitan pengolahan data.

## **3.2 VISUALISASI GRAFIK**

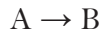
MATLAB menyediakan berbagai fungsi untuk menampilkan data secara dua dimensi maupun tiga dimensi. Pada kasus

dimana Anda membuat grafik dalam tiga dimensi, Anda dapat menggambar permukaan dan menempatkan bingkai pada grafik tersebut. Warna digunakan untuk mewakili dimensi keempat. Untuk lengkapnya akan dibahas berikut.

### 3.2.1 Pembuatan Grafik Sederhana

Perintah `plot` sering digunakan untuk menggambarkan grafik dua dimensi. Perintah `plot` menggambarkan data dalam array pada sumbu yang bersesuaian dan menghubungkan titik-titik tersebut dengan garis lurus. Untuk keterangan lebih jelas mengenai fungsi `plot` ketik **help plot** di command window.

**Contoh 2.1.** Data reaksi berikut telah diperoleh dari reaksi peluruhan sederhana:



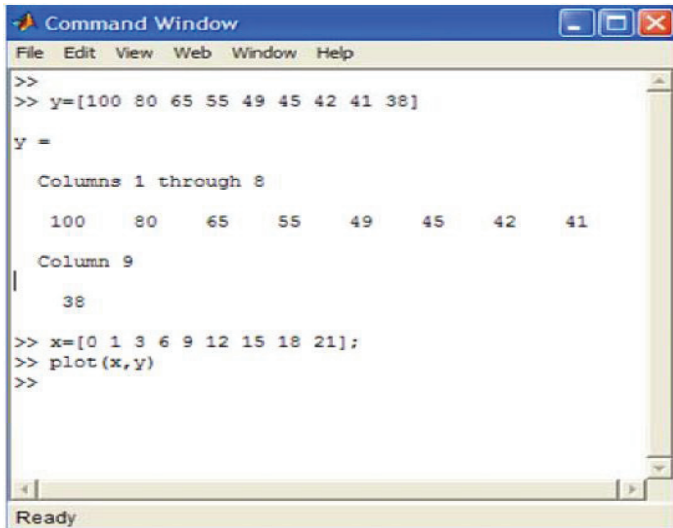
Menggunakan MATLAB untuk mengplot konsentrasi komponen A dalam mol/L terhadap waktu reaksi, dalam menit.

Waktu (Menit)	Konsentrasi (mole/L)
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  $y$  didefinisikan dalam Command Window, diikuti dengan perintah untuk mengplot data. Gambar 3.1 di bawah menunjukkan cara penulisan.





```
>>
>> y=[100 80 65 55 49 45 42 41 38]

y =

Columns 1 through 8

    100     80     65     55     49     45     42     41

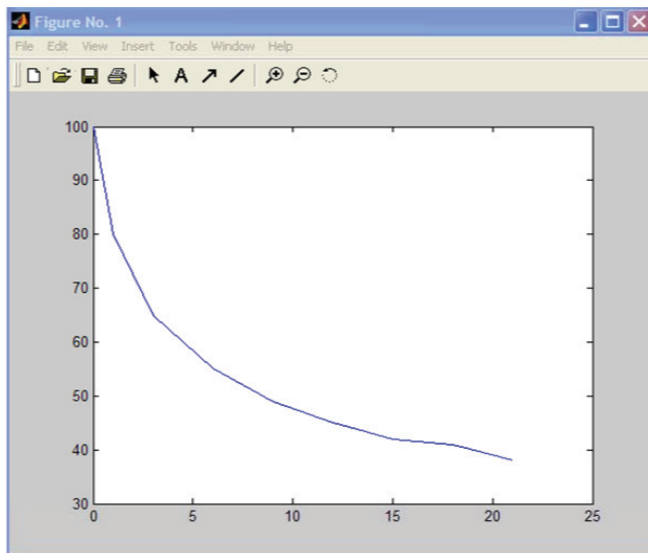
Column 9

     38

>> x=[0 1 3 6 9 12 15 18 21];
>> plot(x,y)
>>
```

**Gambar 3.1** Penulisan perintah plot

Hasil grafik ditunjukkan Gambar 3.2 di bawah.

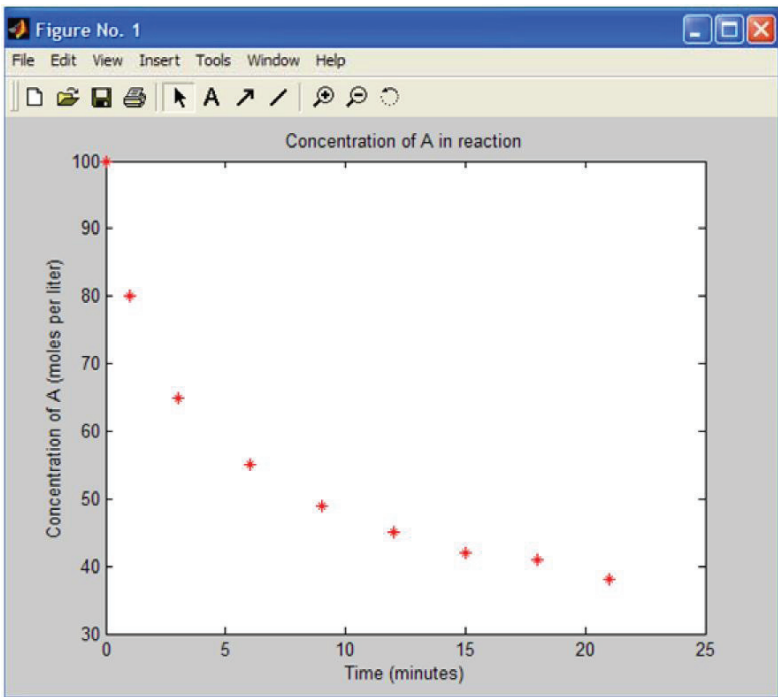


**Gambar 3.2.** Grafik hasil perintah Plot

Matrix baris 'x' (atau vektor) memiliki keluaran tampilan menggunakan ';' pada akhir baris. Syntax dapat digunakan untuk 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 klik 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 klik langsung pada garis, dan property editor garis akan muncul.
- Selanjutnya warna garis, bentuk atau bentuk marker dapat diedit. Kurva akhir ditunjukkan Gambar 3.3 di bawah.



**Gambar 3.3** Hasil grafik akhir

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:

Tabel 3.1 Karakterik dalam MATLAB

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

### 3.2.2 Grafik Multiple Plot

Function *subplot* digunakan untuk membuat suatu *figure* yang dapat memuat lebih dari satu gambar. Perintah subplot didefinisikan sebagai :

```
subplot (n, m, i)
```

Perintah tersebut membagi suatu *figure* menjadi suatu matriks  $m \times n$ , area grafik dan  $i$  berfungsi sebagai indeks penomoran gambar. Subplot dinomori dari kiri ke kanan dimulai dari baris teratas.

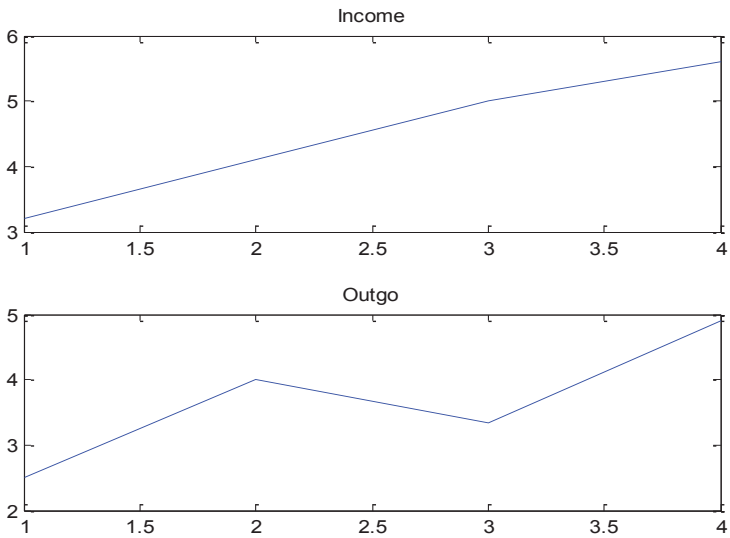
**Contoh 2.2** Untuk memplot grafik atas dan bawah

```
>> income = [3.2 4.1 5.0 5.6];
```

```

outgo = [2.5 4.0 3.35 4.9];
subplot(2,1,1); plot(income)
title('Income')
subplot(2,1,2); plot(outgo)
title('Outgo')

```



### 3.2.3 Grafik Tiga Dimensi (3D)

Perintah menggambar grafik 3D

```
surf(x,y,z)
```

Misalkan: data berikut

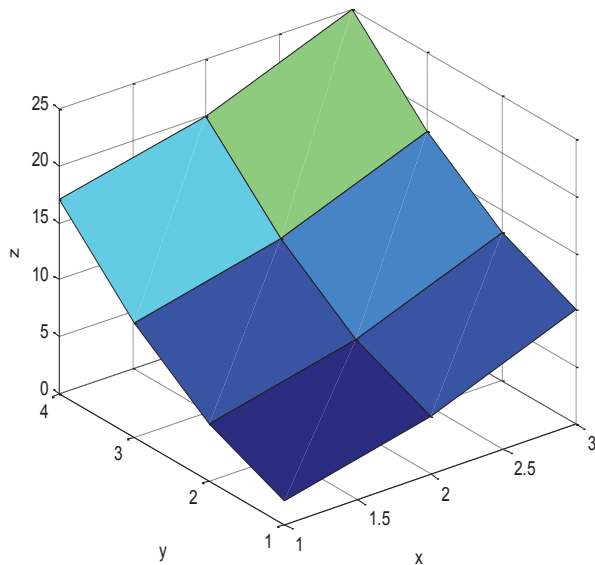
x	y	z(x=1)	z(x=2)	z(x=3)
1	1	2	5	10
2	2	5	8	13
3	3	10	13	18
	4	17	20	25

Syntax penulisan dalam MATLAB adalah:

```
>> x=[1 2 3]
x =
     1     2     3
>> y=[1 2 3 4]
y =
     1     2     3     4
>> z=[2 5 10;5 8 13;10 13 18;17 20 25]
z =
     2     5    10
     5     8    13
    10    13    18
    17    20    25
```

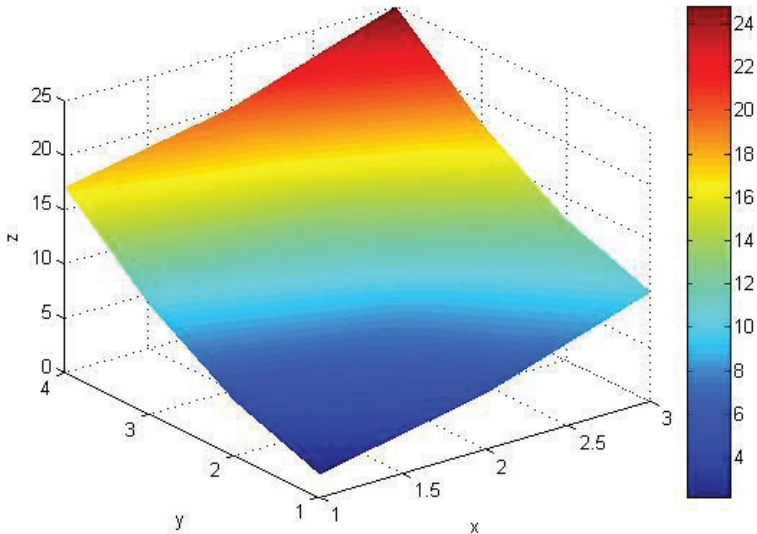
Perintah untuk menampilkan grafik tiga dimensi :

```
>> surf(x,y,z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
```



Untuk mempercantik tampilan dan mempermudah penafsiran grafik dengan menambah legenda warna ketikkan perintah berikut ini.

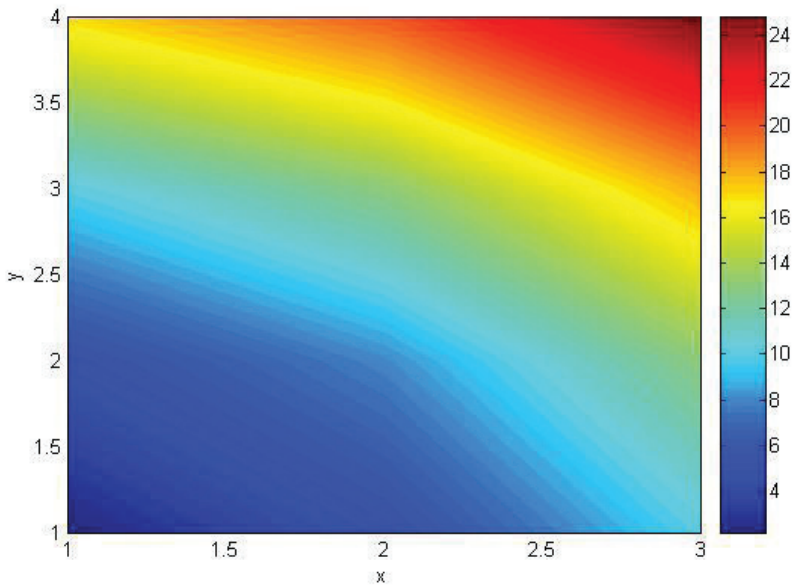
```
>> shading interp
>> colorbar
```



Apabila penafsiran grafik 3D seperti tercetak di muka masih dirasakan sulit, MATLAB telah menyediakan perintah untuk membuat grafik 3D menjadi grafik 2D.

Berikut syntax penulisan dalam MATLAB adalah:

```
>> pcolor(x, y, z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> shading interp
>> colorbar
```



### 3.2.4 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 9

    67.5000    65.7000    68.4000    65.3000    69.1000
    66.2000    68.3000    63.1000    67.3000

Columns 10 through 18

```

```

71.0000    69.9000    68.3000    66.4000    65.7000

70.1000    64.9000    69.6000    67.9000

Columns 19 through 20

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

```

### 3.2.5 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

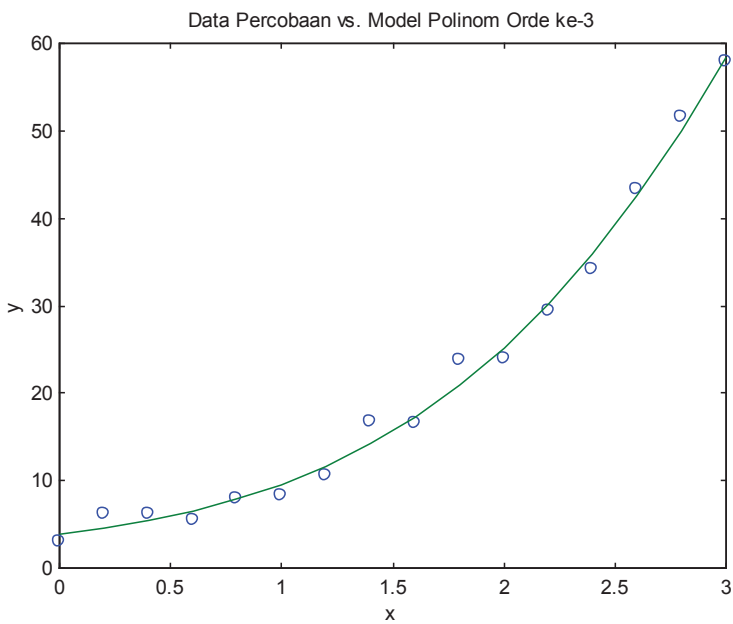
3.9039	4.6115	5.4706	6.5429	7.8901
9.5738	11.6557			

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
---------	---------



### 3.2.6 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.

$$y = x^3 + 2x^2 + 3x + 4$$

penyelesaian dengan MATLAB adal 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
```

### 3.3 RANGKUMAN

- Perintah plot menciptakan grafik suatu vector atau kolom-kolom matriks. Bentuk perintahnya adalah `plot(x1, y1, x2, y2, S2,...)`, dengan  $(x_n, y_n)$  adalah himpunan data dan  $S_n$  adalah string tambahan yang menentukan warna, symbol penandaan, dan style garis
- Grid berfungsi sebagai toggle untuk titik tanda nilai sumbu pada grafik

### 3.4 RUJUKAN

- 1) Dan Hanselman dan Bruce Littlefield, 1997. *MATLAB: Bahasa Komputasi dan Teknis*. hal 180-210 dan 214-234. Andi. Yogyakarta.
- 2) Constantinidis dan Mustoufi, 1999, *Numerical Methodes for Chemical Engineers with MATLAB Application*, Prentice-Hall: Englewood Cliffs, NJ

# **PERSAMAAN LINIER SIMULTAN**

---

## **4.1 PENDAHULUAN**

Pada bab ini akan dibahas tentang persamaan linier simulatan, sehingga diharapkan mahasiswa akan mampu menyelesaikan persamaan linier simultan yang muncul dari permasalahan teknik kimia. Mahasiswa juga diharapkan mampu menyusun suatu persamaan linier simultan dari peristiwa fisis atau kimia dalam teknik kimia dan menyelesaikan persamaan tersebut dengan MATLAB.

Topik ini sangat penting untuk dipelajari karena dalam permasalahan penyusunan neraca massa dan energi sering ditemukan persamaan yang mencerminkan perilaku dari sistem berupa seperangkat persamaan linier simultan yang kompleks.

Untuk memperdalam pemahaman mahasiswa tentang topik ini disarankan untuk membaca kembali buku rujukan yang digunakan dalam buku ini.

## 4.2 MODEL MATEMATIS PERSAMAAN LINER SIMULTAN

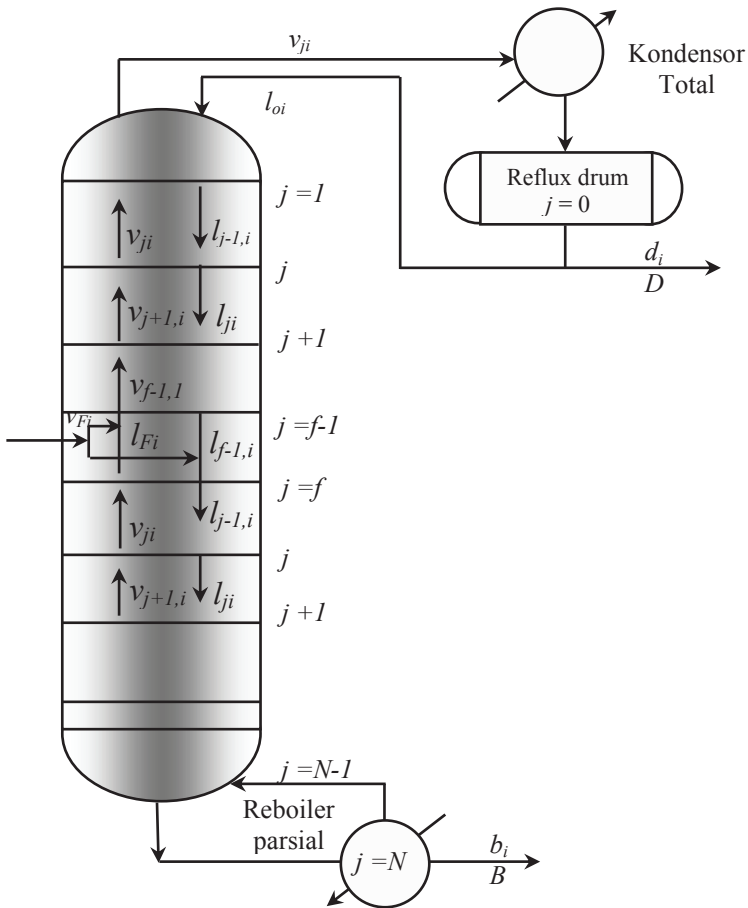
Analisis matematis dari sistem fisika-kimia yang linier sering menghasilkan model yang terdiri dari seperangkat persamaan aljabar linier. Selain itu, metode penyelesaian dari sistem persamaan nonlinier dan persamaan diferensial menggunakan teknik dari model linierisasi, sehingga membutuhkan penyelesaian berulang dari seperangkat persamaan aljabar linier. Masalah-masalah ini dapat berupa persamaan dengan kompleksitas yang tinggi yaitu dari seperangkat yang terdiri dari dua persamaan linear aljabar simultan sampai seperangkat yang melibatkan 1.000 atau bahkan 10.000 persamaan. Penyelesaian dari seperangkat dua sampai tiga persamaan aljabar linear dapat diperoleh dengan mudah dengan metode eliminasi aljabar atau dengan penerapan aturan *Cramer*.

Namun demikian, untuk sistem yang melibatkan lima atau lebih persamaan, metode eliminasi aljabar menjadi terlalu kompleks, dan aturan *Cramer* membutuhkan sejumlah operasi aritmatika tingkat tinggi, terlalu besar bahkan untuk saat ini dibutuhkan kecepatan komputer digital yang tinggi. Dalam bagian ini, diberikan beberapa contoh dari sistem yang menggambarkan aplikasi dari teknik kimia yang menghasilkan seperangkat persamaan aljabar linier simultan.

Neraca massa dan energi adalah *tools* utama bagi insinyur kimia. Neraca tersebut misalnya diterapkan pada proses distilasi bertingkat atau multikomponen yang menghasilkan seperangkat persamaan yang dapat berupa persamaan diferensial atau aljabar. Seringkali sistem yang dianalisis adalah persamaan nonlinier, sehingga menghasilkan seperangkat persamaan nonlinier. Namun, banyak prosedur telah dikembangkan dengan linearisasi persamaan dan menerapkan teknik iteratif konvergensi untuk sampai pada penyelesaian dari sistem



persamaan nonlinier. Sebuah contoh klasik dari penggunaan teknik ini adalah dalam analisis kolom distilasi, seperti yang ditunjukkan pada Gambar 4.1. Neraca massa *steady state* diterapkan pada bagian rektifikasi kolom menghasilkan persamaan berikut:



**Gambar 4.1** Kolom distilasi

Neraca massa sekitar kondensor:

$$V_1 y_{1i} = L_{01} x_{0i} + D x_{Di} \quad (4.1)$$

Neraca massa di atas tahap  $j$ :

$$V_j y_{ji} = L_{j-1} x_{j-1} + D x_{Di} \quad (4.2)$$

Asumsi bahwa tahap bearada dalam kesetimbangan dan kolom meggunakan kondensor total sehingga hubungan kesetimbangan berikut berlaku:

$$y_{ji} = K_{ji} x_{ji} \quad (4.3)$$

persamaan (4.3) disubtitusikan ke persamaan (4.1) dan (4.2) dan semuanya dibagi dengan  $D x_{Di}$  maka dihasilkan persamaan :

$$\frac{V_1 y_{1i}}{D x_{1i}} = \frac{L_0 z_{jF} F}{D x_{1i}} x_{1i} + 1 \quad (4.4)$$

$$\frac{V_j y_j}{D x_{1i}} = \left( \frac{L_{j-1}}{K_{j-1,i}} \right) \left( \frac{V_{j-1} y_{j-1,i}}{D x_{1i}} \right) + 1 \quad (4.5)$$

Laju alir molal komponen tunggal didefinisikan sebagai:

$$v_{ji} = V_1 y_{1i} \quad (4.6)$$

$$d_i = D x_{Di} \quad (4.7)$$

untuk beberapa tahap  $j$ , rasio adsorpsi didefinisikan sebagai:

$$A_{1i} = \frac{L_j}{K_j V_j} \quad (4.8)$$

dan untuk kondenser total :

$$A_{0i} = \frac{L_0}{D} \quad (4.9)$$

Subtitusi persamaan (4.6)-(4.8) ke persamaan (4.4) dan (4.5) dihasilkan:

$$\left( \frac{v_{1i}}{d_i} \right) = A_{0i} + 1 \quad (4.10)$$

$$\left( \frac{v_{1i}}{d_i} \right) = A_{j-1,i} \left( \frac{y_{j-1,i}}{d_i} \right) + 1 \quad (4.11)$$



$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (4.15)$$

$\mathbf{x}$  adalah vektor dari variabel yang tidak diketahui nilainya

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad (4.16)$$

dan  $\mathbf{c}$  adalah vektor konstanta

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_n \end{bmatrix} \quad (4.17)$$

Ketika vektor  $\mathbf{c}$  adalah vektor nol, maka seperangkat persamaan di atas disebut *homogen*.

Di dalam MATLAB, jenis persamaan-persamaan di atas dapat diselesaikan dengan menggunakan fasilitas yang sudah tersedia yaitu operasi *matrix*.

### Contoh 4.1 Mencari beberapa variabel persamaan linier simultan

Diketahui sebuah sistem persamaan linier sbb. :

$$\begin{aligned}5x_1 - 3x_2 + x_3 &= 5 \\x_1 + 2x_2 - 3x_3 + x_4 &= 1 \\3x_2 - 5x_3 + 6x_4 &= -5 \\-x_1 + 2x_3 - x_4 &= 0\end{aligned}\tag{4.1.1}$$

Sistem persamaan linier ini dapat dinyatakan dalam bentuk persamaan matriks :

$$\begin{bmatrix} 5 & -3 & 1 & 0 \\ 1 & 2 & -3 & 1 \\ 0 & 3 & -5 & 6 \\ -1 & 0 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ -5 \\ 0 \end{bmatrix}\tag{4.1.2}$$

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

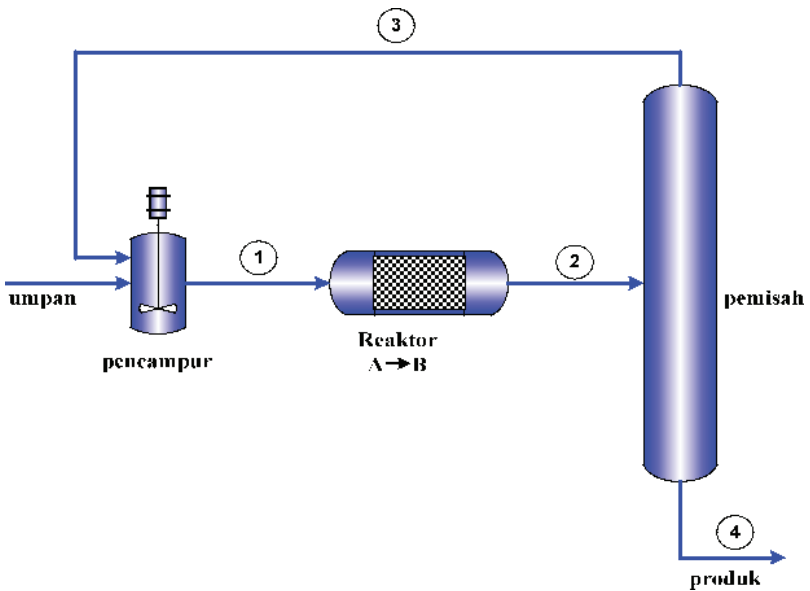
» x = A\b'
x =
     1.3165
     0.5823
     0.1646
    -0.9873

» A*x
ans =
     5.0000
```

1.0000  
-5.0000  
0

#### Contoh 4.2 : 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. Skema proses tersebut ditunjukkan pada Gambar 4.2 di bawah.



Gambar 4.2 Skema Rangkaian Proses

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 didaur-ulang.
- (2) Perbandingan mol  $A$  terhadap  $B$  di dalam alur 1 adalah 5 : 1.

### Penyelesaian:

Neraca massa Pencampur :

$$N_{A1} - N_{A3} = 100 \quad (4.2.1)$$

$$N_{B1} - N_{B3} = 0 \quad (4.2.2)$$

Neraca massa reaktor :

$$-N_{A1} + N_{A2} + r = 0 \quad (4.2.3)$$

$$-N_{B1} - N_{B2} - r = 0 \quad (4.2.4)$$

( $r$  = laju reaksi)

Neraca massa pemisah :

$$-N_{A2} + N_{A3} + N_{A4} = 0 \quad (4.2.5)$$

$$-N_{B2} + N_{B3} + N_{B4} = 0 \quad (4.2.6)$$

Kendala-kendala (*constraints*) :

- Porsi cabang :

$$0.8 N_{A2} + N_{A3} = 0 \quad (4.2.7)$$

$$-0.4 N_{B2} + N_{B3} = 0 \quad (4.2.8)$$

Hubungan komposisi alur :

$$N_{A1} - 5 N_{B1} = 0 \quad (4.2.9)$$

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 sebagai berikut :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & -0.8 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4 & 0 & 1 & 0 & 0 & 0 \\ 1 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} N_{A1} \\ N_{A2} \\ N_{A3} \\ N_{A4} \\ N_{A5} \\ N_{A6} \\ N_{A7} \\ N_{A8} \\ N_{A9} \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.2.10)$$

Script MATLAB dalam command window untuk menyelesaikan sistem persamaan ini adalah sebagai berikut:

```

%9x9 matriks A
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];

%9x1 matriks B
b=[100 0 0 0 0 0 0 0 0]';

%penyelesaian neraca massa dengan eliminasi
Gauss
x = A\b

x =
  227.2727
   45.4545
  159.0909
  113.6364
  127.2727
   45.4545
   31.8182
   68.1818
   68.1818

```

### Contoh 4.3 Sistem Distribusi Steam Pabrik Kimia

Gambar 4.3 di bawah merepresentasikan sistem distribusi steam dalam suatu pabrik kimia (Himmelblau et al, 2012). Neraca massa dan energi untuk sistem ini diberikan sebagai berikut:

$$181,60-x_3-132,57-x_4-x_5 = -y_1 -y_2+y_5+y_4 = 5,1 \quad (4.3.1 a)$$

$$1,17x_3-x_6 = 0 \quad (4.3.2 a)$$

$$132,57- 0,745x_7 = 61,2 \quad (4.3.3 a)$$

$$x_5+x_7-x_8- x_9-x_{10}+x_{15} = y_7 +y_8-y_3 = 99,1 \quad (4.3.4 a)$$



$$x_8 + x_9 + x_{10} + x_{11} - x_{12} - x_{13} = -y_7 = -8,4 \quad (4.3.5 a)$$

$$x_6 - x_{15} = y_6 - y_5 = 24,2 \quad (4.3.6 a)$$

$$-1,15(181,60) + x_3 - x_6 + x_{12} + x_{16} = 1,15y_1 - y_9 + 0,4 = -19,7 \quad (4.3.7 a)$$

$$181,60 - 4,594x_{12} - 0,11x_{16} = -y_1 + 1,0235y_9 + 2,45 = 35,05 \quad (4.3.8 a)$$

$$-0,0423(181,60) + x_{11} = 0,0423y_1 = 2,88 \quad (4.3.9 a)$$

$$-0,016(181,60) + x_4 = 0 \quad (4.3.10 a)$$

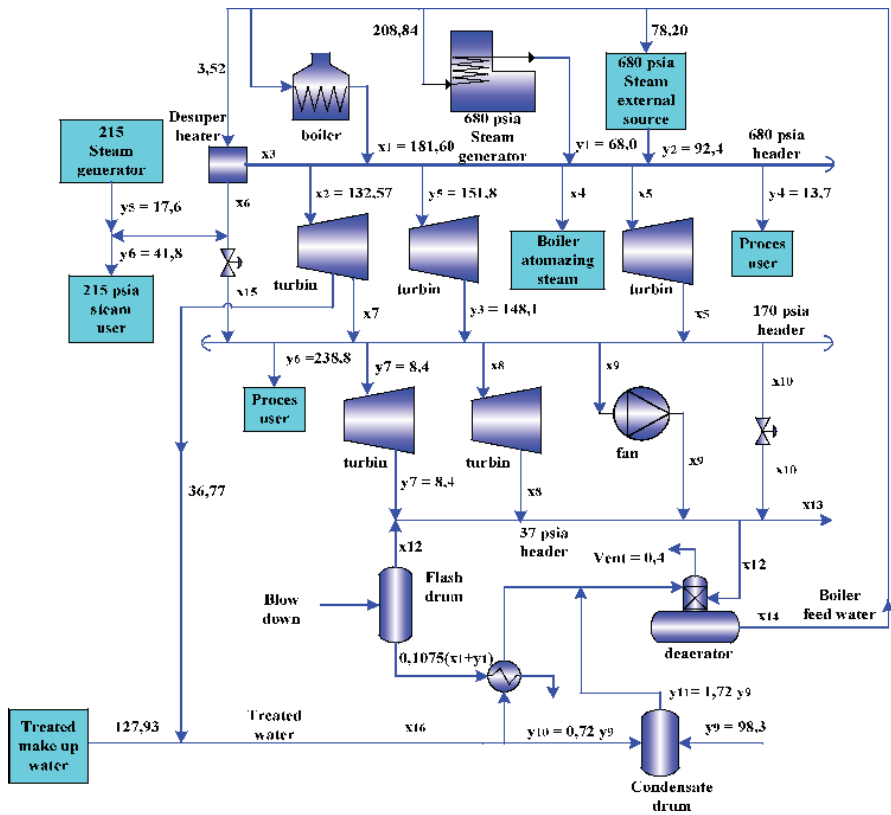
$$x_8 - 0,147x_{16} = 0 \quad (4.3.11 a)$$

$$x_5 - 0,07x_{14} = 0 \quad (4.3.12 a)$$

$$-0,0805(181,60) + x_9 = 0 \quad (4.3.13 a)$$

$$x_{12} - x_{14} + x_{16} = 0,4 - y_9 = -97,9 \quad (4.3.14 a)$$

Terdapat empat level steam pada pabrik ini yaitu: 680, 215, 170, and 37 psia. Ke- 14  $x_i$ ,  $i = 3, \dots, i = 16$  tidak diketahui nilainya dan  $y_i$  adalah parameter yang diketahui untuk sistem ini. Kedua  $x_i$  dan  $y_i$  memiliki satuan 1000 lb/jam. Tentukan 14 parameter  $x_i$  yang tidak diketahui dari  $i = 3, \dots, i = 16$ .



Gambar 4.3 Sistem distribusi steam

**Penyelesaian:**

14 persamaan ini menunjukkan neraca disekitar 14 unit operasi, yaitu:

- |                              |                                   |
|------------------------------|-----------------------------------|
| Pers. (1) 680 psia header    | Pers. (8) Condensate quench drum  |
| Pers. (2) Desuperheater      | Pers. (9) Blow down flash drum    |
| Pers. (3) Alternator turbine | Pers. (10) Boiler atomizing       |
| Pers. (4) 170 psia header    | Pers. (11) Treated feedwater pump |
| Pers. (5) 37 psia header     | Pers. (12) Boiler feedwater pump  |
| Pers. (6) 215 psia steam     | Pers. (13) Boiler fan             |
| Pers. (7) BFW balance        | Pers. (14) Deaerator- quench.     |

Seperangkat persamaan yang dinyatakan dalam soal disederhanakan menjadi seperangkat persamaan yang mengandung 14 variabel yang tidak diketahui ( $x_3 - x_{16}$ ).

$$x_3 + x_4 + x_5 = 43,93 \quad (4.3.1b)$$

$$1,17x_3 - x_6 = 0 \quad (4.3.2b)$$

$$x_7 = 95,798 \quad (4.3.3b)$$

$$x_5 + x_7 - x_8 - x_9 - x_{10} + x_{15} = 99,1 \quad (4.3.4b)$$

$$x_8 + x_9 + x_{10} + x_{11} - x_{12} - x_{13} = -8,4 \quad (4.3.5b)$$

$$x_6 - x_{15} = 24,2 \quad (4.3.6b)$$

$$x_3 - x_6 + x_{12} + x_{16} = 189,14 \quad (4.3.7b)$$

$$4,594x_{12} + 0,11x_{16} = 146,55 \quad (4.3.8b)$$

$$x_{11} = 10,56 \quad (4.3.9b)$$

$$x_4 = 2,9056 \quad (4.3.10b)$$

$$x_8 - 0,147x_{16} = 0 \quad (4.3.11b)$$

$$x_5 - 0,07x_{14} = 0 \quad (4.3.12b)$$

$$x_9 = 14,6188 \quad (4.3.13b)$$

$$x_{12} - x_{14} + x_{16} = -97,9 \quad (4.3.14b)$$

Untuk kemudahan dalam pemrograman, kita rubah urutan angka variabel sehingga  $x_3$  menjadi  $x_1$ ,  $x_4$  menjadi  $x_2$ , dan seterusnya.

$$x_1 + x_2 + x_3 = 43,93 \quad (4.3.1c)$$

$$1,17x_1 - x_4 = 0 \quad (4.3.2c)$$

$$x_5 = 95,798 \quad (4.3.3c)$$

$$x_3 + x_5 - x_6 - x_7 - x_8 + x_{13} = 99,1 \quad (4.3.4c)$$

$$x_6 + x_7 + x_8 + x_9 - x_{10} - x_{11} = -8,4 \quad (4.3.5c)$$

$$x_4 - x_{13} = 24,2 \quad (4.3.6c)$$

$$x_1 - x_4 + x_{10} + x_{14} = 189,14 \quad (4.3.7c)$$

$$4,594x_{10} + 0,11x_{14} = 146,55 \quad (4.3.8c)$$

$$x_9 = 10,56 \quad (4.3.9c)$$

$$x_2 = 2,9056 \quad (4.3.10c)$$

$$x_6 - 0,0147x_{14} = 0 \quad (4.3.11c)$$

$$x_3 - 0,07x_{12} = 0 \quad (4.3.12c)$$

$$x_7 = 14,6188 \quad (4.3.13c)$$

$$x_{10} - x_{12} + x_{14} = -97,9 \quad (4.3.14c)$$

Persamaan aljabar linier di atas dapat dinyatakan dengan:

$$Ax = c \quad (4.3.15)$$

Dimana  $A$  adalah mariks bujur sangkar yang banyak mengandung nilai nol. Script penyelesaian dalam MATLAB dpat dituliskan sebagai berikut:

$$\begin{bmatrix}
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1.7 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.594 & 0 & 0 & 0 & 0.11 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0147 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.07 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6 \\
 x_7 \\
 x_8 \\
 x_9 \\
 x_{10} \\
 x_{11} \\
 x_{12} \\
 x_{13} \\
 x_{14}
 \end{bmatrix}
 =
 \begin{bmatrix}
 43.93 \\
 0 \\
 95.798 \\
 99.1 \\
 -8.4 \\
 24.2 \\
 189.14 \\
 146.55 \\
 10.56 \\
 2.9056 \\
 0 \\
 0 \\
 14.6188 \\
 97.9
 \end{bmatrix}
 \quad (4.3.15)$$

```

P = [1 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
1.17 0 0 -1 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
0 0 1 0 1 -1 -1 -1 0 0 0 0 1 0;
0 0 0 0 0 1 1 1 1 -1 -1 0 0 0;
0 0 0 1 0 0 0 0 0 0 0 0 -1 0;
1 0 0 -1 0 0 0 0 0 1 0 0 0 1;
0 0 0 0 0 0 0 0 0 4.594 0 0 0 0.11;
0 0 0 0 0 0 0 0 1 0 0 0 0 0;
0 1 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 1 0 0 0 0 0 0 0 -0.0147;
0 0 1 0 0 0 0 0 0 0 -0.07 0 0 0;
0 0 0 0 0 0 1 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 -1 0 1];

```

```

0 0 1 0 0 0 0 0 0 0 0 -0.07 0 0;
0 0 0 0 0 0 1 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 1 0 -1 0 1];
Q =[43.93 0 95.798 99.1 -8.4 24.2 189.14
146.55 10.56 2.9056 0 0 14.6188 -97.9]';
S=P\Q
S=

20.6854
2.9056
20.3390
24.2020
95.7980
2.4211
14.6188
-0.0010
10.5600
27.957
8.0422
290.556
0.0020
164.6998

```

Berdasarkan hasil perhitungan di atas maka dapat dituliskan nilai penyelesaian untuk  $x_1$  sampai  $x_{14}$  sebagai berikut :

$x$	Nilai
$x_1$	20.3390
$x_2$	24.2020
$x_3$	95.7980
$x_4$	2.4211
$x_5$	14.6188
$x_6$	2.4211
$x_7$	14.6188

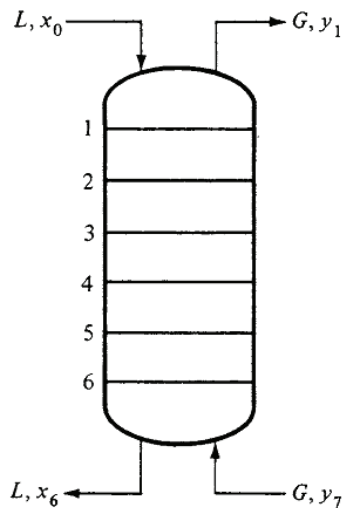
$x$	Nilai
$x_8$	0
$x_9$	10.5600
$x_{10}$	27.9567
$x_{11}$	8.0422
$x_{12}$	290.5565
$x_{13}$	0.0020
$x_{14}$	164.6998

#### Contoh 4.4 Kolom Absorpsi Enam Plat

Sebuah kolom absorpsi enam plat/tray dioperasikan pada keadaan steady state (Raramirez, 1998). Hubungan kesetimbangan linier yang terjadi antara cairan ( $x_m$ ) dan uap ( $y_m$ ) setiap plat dapat diasumsikan dengan persamaan:

$$y_m = ax_m + b \quad (4.4.1)$$

Komposisi umpan masuk ke kolom,  $x_0$  dan  $y_7$  ditentukan bersama laju alir (mol/waktu) fasa cairan ( $L$ ) dan gas ( $G$ ). Sistem ditunjukkan gambar di bawah.



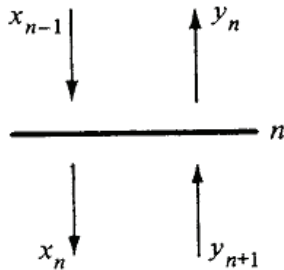
Gambar 4.4. Kolom Absorpsi

Tentukan komposisi tray, jika diketahui:

- Cairan umpan adalah cairan murni ( $x_0 = 0$ ) dan umpan gas mengandung 0.2 kmol gas terlarut/kmol gas inert ( $y_7 = 0.2$ ).
- Cairan umpan adalah cairan murni ( $x_0 = 0$ ) dan  $y_7 = 0.3$ .  
 Nilai :  $a = 0,72$  ;  $L = 66,7$  kgmol/menit  
 $b = 0$  ;  $G = 40,8$  kgmol/menit

**Penyelesaian:**

Untuk menyelesaikan soal di atas, neraca massa ditulis pada tray representative, ke- $n$ , yang ditunjukkan gambar di bawah.



**Gambar 4.5.** Tray tipikal

Penerapan neraca massa makroskopis diperoleh:

$$\text{Rate of mass in} = \text{Rate of mass out} \tag{4.4.2}$$

$$L x_{n-1} + G y_{n+1} = L x_n + G y_n$$

Menggunakan hubungan kesetimbangan linier pada neraca massa memberikan :

$$L x_{n-1} + G (a x_{n+1} + b) = L x_n + G (a x_n + b) \tag{4.4.3}$$

atau

$$L x_{n-1} - (L + Ga)x_n + Gax_{n+1} = 0 \tag{4.4.4}$$

Berikut seperangkat persamaan untuk kolom enam plat:

$$\begin{aligned}
-(L + Ga)x_1 + Gax_2 &= Lx_0 \\
Lx_1 - (L + Ga)x_2 + Gax_3 &= 0 \\
Lx_2 - (L + Ga)x_3 + Gax_4 &= 0 \\
Lx_3 - (L + Ga)x_4 + Gax_5 &= 0 \\
Lx_4 - (L + Ga)x_5 + Gax_6 &= 0 \\
Lx_5 - (L + Ga)x_6 &= -Ga \frac{y_7 - b}{a}
\end{aligned}
\tag{4.4.5}$$

$$\begin{bmatrix}
-(L+Ga) & Ga & & & & & \\
L & -(L+Ga) & Ga & & & & \\
& L & -(L+Ga) & Ga & & & \\
& & L & -(L+Ga) & Ga & & \\
& & & L & -(L+Ga) & Ga & \\
& & & & L & -(L+Ga) & Ga \\
& & & & & L & -(L+Ga)
\end{bmatrix}
=
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix}
\begin{bmatrix}
-Lx_0 \\
0 \\
0 \\
0 \\
0 \\
-Ga \frac{y_7 - b}{a}
\end{bmatrix}
\tag{4.4.5}$$

Persamaan matriks adalah :

$$\mathbf{Ax} = \mathbf{b}
\tag{4.4.6}$$

Berikut script program penyelesaian soal di atas yang ditulis dalam *command window*.

```

%baca parameter proses yang diketahui
a=0.72;
b=0;
L=66.7;%kgmol/menit
G=40.88;%kgmol/menit

%baca kondisi awal
x0=0;y7(1)=0.2; y7(2)=0.3;
%baca matriks A dan B
A=zeros(6);%6x6 matriks berelemen nol
A(1,1)=- (L+G*a);%elemen matriks pada baris ke-1
kolom ke-1
A(2,1)=L;
A(1,2)=G*a;
A(2,2)=- (L+G*a);
A(3,2)=L;
A(2,3)=G*a;
A(3,3)=- (L+G*a);

```



```

A(4,3)=L;
A(3,4)=G*a;
A(4,4)=-(L+G*a);
A(5,4)=L;
A(4,5)=G*a;
A(5,5)=-(L+G*a);
A(6,5)=L;
A(5,6)=G*a;
A(6,6)=-(L+G*a);

B1=zeros(6,1);%6x1 matriks berelemen nol
B1(1,1)=-L*x0;%elemen matriks pada baris ke-1
                    kolom ke-1
B1(6,1)=-G*a*(y7(1)-b)/a;

B2=zeros(6,1);%6x1 matriks berelemen nol
B2(1,1)=-L*x0;%elemen matriks pada baris ke-1
                    kolom ke-1
B2(6,1)=-G*a*(y7(2)-b)/a;

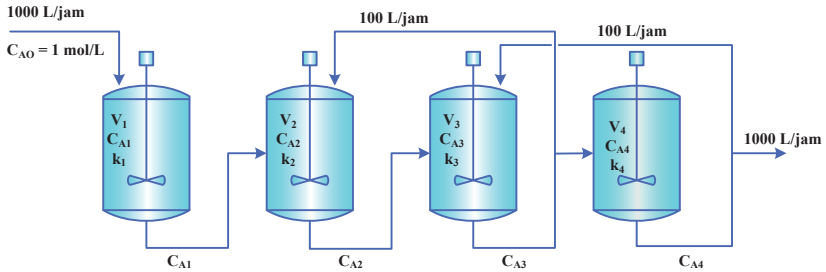
%penyelesaian komposisi tray dengan eliminasi
    Gauss
x1=A\B1
x1 =
    0.0011
    0.0038
    0.0097
    0.0230
    0.0534
    0.1221

x2=A\B2
x2 =
    0.0017
    0.0056
    0.0145
    0.0346
    0.0800
    0.1831

```

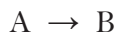
### Contoh 4.5 Penyelesaian Persamaan Reaksi Kimia dan Neraca Massa

Suatu reaksi kimia dijalankan dalam serangkaian empat reaktor alir tangki berpengaduk (*CSTR*) yang disusun seperti gambar di bawah.



**Gambar 4.6.** Rangkain Reaktor CSTR

Reaksi kimia adalah reaksi *irreversible* orde satu dengan persamaan reaksi:



Kondisi temperatur dalam setiap reaktor sebagaimana nilai konstanta laju reaksi  $k_i$  berbeda di setiap reaktor. Volume setiap reaktor,  $V_i$  juga berbeda. Nilai  $k_i$  dan  $V_i$  diberikan pada tabel di bawah. Berikut asumsi yang dapat digunakan untuk sistem tersebut.

- Sistem adalah *steady state*
- Reaksi terjadi di fasa cair
- Tidak ada perubahan volume atau densitas cairan
- Laju pengurangan komponen A dalam setiap reaktor dinyatakan dengan :

$$R_i = V_i k_i C_{Ai} \quad (\text{mol/jam}) \quad (4.5.1)$$

Reaktor	$V_i$ (L)	$k_i$ (jam <sup>-1</sup> )
1	1000	0,1
2	1500	0,2
3	100	0,4
4	500	0,3

Susunlah persamaan neraca massa untuk keempat reaktor dan tentukan konsentrasi ( $C_{Ai}$ ) yang keluar dari tiap reaktor.

### Penyelesaian:

Neraca massa unsteady state untuk setiap reaktor adalah:

$$\text{Rate of input} - \text{Rate of output} - \text{Rate of Reaction} = \text{Rate of Accumulation} \quad (4.5.2)$$

Karena sistem adalah steady state, maka laju akumulasi adalah nol, neraca massa dapat disederhanakan menjadi:

$$\text{Rate of input} - \text{Rate of output} - \text{Rate of Reaction} = 0 \quad (5.4.3)$$

Penyusunan neraca massa untuk setiap reaktor menghasilkan persamaan:

$$\begin{aligned} 1000C_{A0} - 1000C_{A1} - V_1 k_1 C_{A1} &= 0 \\ 1000C_{A1} + 100C_{A3} - 1100C_{A2} - V_2 k_2 C_{A2} &= 0 \\ 1100C_{A2} + 100C_{A4} - 1200C_{A3} - V_3 k_3 C_{A3} &= 0 \\ 1100C_{A3} - 1100C_{A4} - V_4 k_4 C_{A4} &= 0 \end{aligned} \quad (5.4.4)$$

Substitusi nilai  $C_{A0}$ ,  $k_i$  dan  $V_i$  diperoleh persamaan:

$$\begin{aligned} 1100C_{A1} &= 1000 \\ 1000C_{A1} - 1400C_{A2} + 100C_{A3} &= 0 \\ 1100C_{A2} - 1240C_{A3} + 100C_{A4} &= 0 \\ 1100C_{A3} - 1250C_{A4} &= 0 \end{aligned} \quad (4.4.5)$$

Persamaan diatas disusun dalam matriks :

$$\begin{bmatrix} 1100 & & & & \\ 1000 & -1400 & 100 & & \\ & 1100 & -1240 & 100 & \\ & & & 1100 & -1250 \end{bmatrix} \begin{bmatrix} C_{A1} \\ C_{A2} \\ C_{A3} \\ C_{A4} \end{bmatrix} = \begin{bmatrix} 1000 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.4.6)$$

Penyelesaian persamaan dengan MATLAB adalah berikut:

```
%matriks A dan B
A=[1100 0 0 0;1000 -1400 100 0;0 1100 -1240
  100;0 0 1100 -1250];
B=[1000;0;0;0];
x=A\B
x =
    0.9091
    0.6969
    0.6654
    0.5856
```

### 4.3 RANGKUMAN

- Persamaan aljabar linier simultan dapat dinyatakan dalam vektor matriks
- Untuk dapat dinyatakan dalam bentuk vector/matrik jumlah variabel yang tidak diketahui harus sama dengan jumlah persamaan
- Matriks yang terbentuk dari persamaan linier simultan adalah matrik bujur sangkar
- Persaman aljabar linier simultan dapat diselesaikan dengan MATLAB menggunakan operasi matriks

### 4.4 RUJUKAN

- 1) Constantinidis dan Mustoufi. 1999. *Numerical Methodes for Chemical Engineers with MATLAB Application*. hal 63-69. Prentice-Hall: Englewood Cliffs, NJ

- 2) Himmelblau, M. David and James B Riggs. 2012. *Basic Principles and Calculations in Chemical Engineering* 8th Ed. Prentice-Hall: NJ
- 3) Cutlip, M. B., Shacham, M. 1999. *Problem Solving in Chemical Engineering with Numerical Methods*. Hal 15-16. Prentice-Hall: Englewood Cliffs, NJ.
- 4) Ramirez, W.F. 1998. *Computational Methods in Process Simulation*. Hal: 54-56. Elsevier Science & Technology Books.

# PERSAMAAN NON-LINIER TUNGGAL

## 5.1 PENDAHULUAN

Materi ini membahas tentang persamaan non-linier satu variabel. Setelah menyelesaikan bab ini diharapkan mahasiswa akan mampu menyelesaikan persoalan dalam teknik kimia yang dapat dinyatakan dengan persamaan nonlinier. Secara khusus mahasiswa diharapkan mampu menyusun peristiwa fisis dan kimia dalam teknik kimia kedalam persamaan matematis yang berbentuk nonlinier dan mampu menyelesaikannya dengan fungsi MATLAB yaitu *fzero*.

Materi ini sangat penting mengingat banyak peristiwa dalam bidang teknik kimia yang berkaitan dengan termodinamika memiliki bentuk persamaan nonlinier yang harus dipecahkan untuk proses evaluasi, optimasi dan perancangan. Sehingga memahami dan menguasai materi dari bab ini akan membantu mahasiswa dalam memecahkan persoalan perhitungan tentang termodinamika dan lainnya yang berkaitan dengan persamaan non linier.

Pendalaman materi tentang topik yang dibahas pada bab ini dapat dibaca buku yang menjadi rujukan sebagaimana dituliskan halamannya.

## 5.2 MODEL MATEMATIS PERSAMAAN NON LINIER

Banyak persoalan dalam bidang teknik dan sains memerlukan penyelesaian persamaan nonlinier. Beberapa contoh masalah yang menggambarkan dari bidang teknik kimia dan dari aplikasi bidang lain akan dibahas pada bagian ini. Metode penyelesaian akan yang dikembangkan dalam bagian bab ini, dan contoh-contoh spesifik penyelesaian akan didemonstrasikan menggunakan perangkat lunak MATLAB.

Dalam termodinamika, hubungan tekanan-volume-suhu gas nyata dinyatakan dengan persamaan keadaan (*equation of state*). Ada beberapa persamaan semi teoritis atau empiris, seperti Redlich-Kwong, Soave-Redlich-Kwong, dan persamaan Benedicts-Webb-Rubin, yang telah digunakan secara luas dalam teknik kimia. Sebagai contoh, persamaan keadaan Soave-Redlich-Kwong yang memiliki bentuk:

$$P = \frac{RT}{v-b} - \frac{a\alpha}{v(v+b)} \quad (5.1)$$

Dimana  $P$ ,  $v$  dan  $T$  adalah tekanan, volume molar, dan suhu.  $R$  adalah konstanta gas,  $\alpha$  merupakan fungsi temperatur,  $a$  dan  $b$  adalah konstanta yang spesifik untuk masing-masing gas.

Pers (5.1) adalah polinomial derajat tiga dalam  $V$  dan dapat dengan mudah diatur kembali menjadi bentuk kanonik dari polinomial, yaitu:

$$Z^3 - Z^2 + (A - B - B^2)Z - AB = 0 \quad (5.2)$$

dimana  $Z = P v/RT$  adalah faktor kompresibilitas,  $A = aP/R^2T^2$  dan  $B = bP/RT$ .

Oleh karena itu, masalahnya adalah mencari volume spesifik gas pada temperatur dan tekanan yang diberikan untuk menemukan akar yang tepat dari persamaan polinomial.

Kasus lain, misalnya di dalam perhitungan pemisahan multikomponen, sering diperlukan untuk memperkirakan rasio

refluks minimum kolom distilasi multistage. Suatu metode telah dikembangkan oleh Underwood (1948) dan dijelaskan secara detil oleh Treybal (1980), yakni persamaan berikut:

$$\sum_{j=1}^n \frac{\alpha_j z_{jF} F}{\alpha_j - \phi} - F(1 - q) = 0 \quad (5.3)$$

dimana  $F$  adalah alju alir molar umpan,  $n$  adalah jumlah komponen pada umpan,  $z_{jF}$  adalah mol fraksi setiap komponen dalam umpan,  $q$  adalah kualitas umpan  $\alpha_j$  adalah volatilitas relative setiap komponen pada kondisi rata-rata kolom, dan  $\phi$  adalah akar persamaan. Laju alir umpan, komposisi umpan, dan kualitas biasanya diketahui, kondisi kolom rata-rata dapat diperkirakan. Sehingga hanya  $\phi$  yang tidak diketahui. Karena persamaan ini adalah persamaan polynomial dalam  $\phi$  dengan derajat  $n$ , maka ada beberapa kemungkinan nilai  $\phi$  (akar) yang cocok menyelesaikan persamaan (5.3).

Semua persamaan non linier dapat dituliskan dalam bentuk umum :

$$f(x) = 0 \quad (5.4)$$

dimana  $x$  adalah variabel tunggal dapat yang dapat memiliki akar banyak yang sesuai dengan persamaan ini. Fungsi  $f(x)$  mungkin menganggap berbagai fungsi nonlinier mulai dari persamaan polinomial yang kanonik berbentuk

$$f(x) = a_n x^n - a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0 \quad (5.5)$$

ke persamaan transendental, yang melibatkan bentuk trigonometri, eksponensial, dan logaritmik. Akar fungsi ini dapat berupa :

- 1) real dan jelas
- 2) real dan diulang
- 3) kompleks konjugat
- 4) kombinasi dari beberapa atau semua di atas

Bagian real dari akar bisa positif, negatif, atau nol.



MATLAB memiliki dua rutin yang dapat menyelesaikan fungsi pengenalan untuk satu variabel. *fzero* digunakan untuk persamaan nonlinear umum, sementara *roots* dapat digunakan jika persamaan nonlinear adalah polinomial.

### 5.1 *fzero*

Routine pertama yang kita gunakan adalah *fzero*. *fzero* menggunakan kombinasi metode numeris *interval bisection* dan *regula falsi*.

Syntax yang digunakan untuk menuliskan *fzero* adalah

```
z = fzero('function', initial guess)
```

Untuk menggunakan *fzero* Anda harus terlebih dahulu menulis m-file MATLAB untuk menghasilkan fungsi yang sedang dievaluasi. Anggap suatu fungsi:  $f(x) = x^2 - 2x - 3 = 0$ . m-file MATLAB berikut untuk mengevaluasi fungsi ini (m-file dinamakan *fncl.m*):

```
function y=fncl(x)
y=x^2-2*x-3;
```

Setelah menghasilkan m-file *fncl.m*, Anda harus menyediakan tebakan awal untuk menyelesaikan rutin *fzero*. Perintah berikut memberikan tebakan awal  $x = 0$ .

```
y = fzero('fncl', 0)
```

MATLAB akan memberikan jawabannya:

```
y = 1
```

Untuk tebakan awal  $x = 2$ , Anda memasukkan:

```
z = fzero('fncl', 2)
```

dan MATLAB kembali memberikan jawaban:

```
z = 3
```

Kita temukan bahwa ada dua solusi untuk masalah yang sama (kita dapat menggunakan rumus kuadratik untuk memperolehnya). Solusi yang diperoleh tergantung pada tebakan awal.

## 5.2 roots

Kita juga bisa menggunakan routine MATLAB *roots* untuk menemukan pengenalan suatu polynomial. Misal fungsi polinomial:

$$x^2-2x-3 = 0$$

Anda harus membuat vector koefisien polynomial dalam orde yang berurutan

$$c = [1 \ -2 \ -3]$$

kemudian Anda dapat menuliskan perintah seperti berikut:

$$\text{roots}(c)$$

dan MATLAB akan menjawab:

$$\begin{array}{l} \text{ans:} \\ 3 \\ -1 \end{array}$$

Ini adalah penyelesaian yang kita harapkan.

Berikut di bawah ini akan dijelaskan beberapa contoh penyelesaian persamaan non linier tunggal secara umum dan aplikasinya dalam teknik kimia.

### Contoh 5.1 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 5.2 Mencari temperatur untuk suatu harga $C_p$ tertentu

Diketahui sebuah persamaan kapasitas panas sebagai berikut :

$$C_p = 0.716 - 4.257 \cdot 10^{-6} T - \frac{15.04}{\sqrt{T}} \left[ \frac{kJ}{kg.K} \right] \text{ dan } T \text{ dalam K} \quad (5.2.1)$$

Akan ditentukan temperatur pada saat  $C_p = 1 \text{ kJ/kg.K}$ . Untuk itu, ubahlah persamaan di atas menjadi :

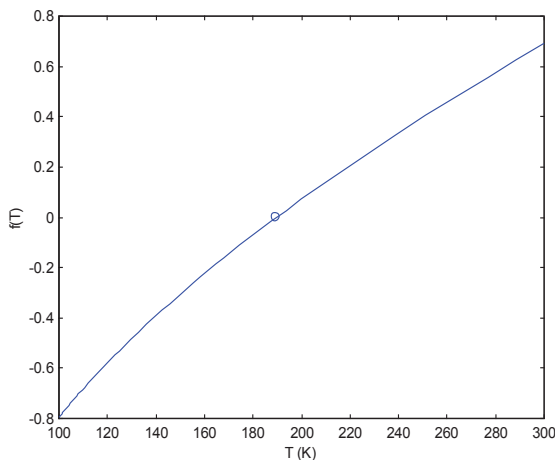
$$f(T) = 1 - 0.716 + 4.257 \cdot 10^{-6} T + \frac{15.04}{\sqrt{T}} = 0 \quad (5.2.2)$$

#### Penyelesaian :

**Tahap 1 :** membuat fungsi yang dapat mengevaluasi persamaan (5.2.2)

```
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 sebagai berikut:



**Gambar 5.1.** Grafik  $C_p$  terhadap suhu

Untuk mendapatkan harga pembuat nol 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.

### Contoh 5.3. Mencari volume spesifik gas n-butana

Tentukan volume spesifik butana pada 500 K dan 18 atm menggunakan persamaan keadaan Redlich – Kwong :

$$p = \frac{RT}{v-b} - \frac{a}{v(v+b)} \quad (5.3.1)$$

$$a = 0,42748 \left( \frac{R^2 T_c^2}{p_c} \right) \alpha, b = 0,08664 \left( \frac{RT}{p_c} \right)$$

$$\text{dan } T_r = \frac{T_r}{T_c}, \alpha = \frac{1}{T_r^{0,5}} \quad (5.3.2)$$

#### Penyelesaian :

Persamaan (5.3.1) dapat dituliskan :

$$v^3(p) - v^2(RT) + v(a - pb^2 - RTb) - ab = 0 \quad (5.3.3)$$

Untuk mendapatkan nilai  $v$  maka:

$$F(v) = 0 \quad (5.3.4)$$

Langkah penyelesaian soal di atas adalah diatas adalah :

**Tahap 1:** Pertama kita perlu menyiapkan m-file yang akan menghitung  $f(v)$ , dengan tekanan, temperatur, dan properti termodinamika yang diberikan. File ditunjukkan di bawah:

```
% menghitung vol gas butana
function y=specvol(v)
% in K atm l/gmol
```

```

% parameters for n-butane
Tc=425.2
pc=37.5
T=500 (2:27)
p=18
R=0.08206
Tr=T/Tc
alphaRK=1/(Tr)^0.5
aRK=(0.42748*(R*Tc)^2/pc)*alphaRK
bRK=0.08664*(R*Tc/pc)
y=p*v^3-R*T*v^2+(aRK-p*(bRK)^2-R*T*bRK)*v-RK*bRK;

```

Fungsi ini dinamakan 'specvol', mendefinisikan masalah yang akan diselesaikan.

**Tahap 2.** Untuk menguji fungsi 'specvol' kita keluarkan juga perintah di commands window:

```

feval('specvol',0.2)
ans=specvol(0.2)

```

Fungsi feval menyebabkan MATLAB menghitung nilai  $y$  menggunakan m-file yang diberi nama specvol ketika  $v = 0.2$ .

Output yang kita peroleh adalah:

```

Tc=425.2000
pc=37.5000
T=500
p=18 (2:29)
R=0.08206
aRK=13.8782
aRK=12.7981
bRK=0.0806
y=-0.6542

```

Kita harus memeriksa hasil ini baris demi baris, khususnya perhitungan untuk  $aRK$ ,  $bRK$ , and  $y$ .

**Tahap 3.** Ketika kita menggunakan *fzero*, fungsi *specvol* akan dievaluasi untuk variasi  $v$ .

Oleh karena itu, terlihat tidak nyaman jika konstanta ditampilkan pada layar untuk setiap iterasi. Untuk menghindari hal tersebut, kita rubah fungsi specvol dengan menambahkan

titik koma ‘ ; ’ pada setiap baris. Lakukan hal ini dan simpan save m-file, *specvol*.

**Tahap 4.** Selanjutnya, kita keluarkan perintah pada command window:

```
v = fzero('specvol', 0.2)
v = 2.0377
```

Dalam *feval*, 0.2 adalah nilai *v* yang digunakan dalam perhitungan, dimana dengan *fzero*, *v* adalah nilai tebakan awal. Untuk mengeceknya, kita dapat mengevaluasi fungsi untuk menemukan seberapa dekat dengan nol  $f(v)$ .

```
>> ans = specvol(v)
ans = -2.2204e-15
```

Tentu kita berharap ini adalah nol (atau sangat dekat dengan nol) karena kita berharap MATLAB bekerja dengan baik. Jika MATLAB tidak menemukan jawaban, MATLAB akan memberitahu kita.

**Contoh 5.4. Pemisahan Campuran dengan Flash Drum**

Pada Tabel 5.1 di bawah diberikan data untuk empat komponen hidrokarbon. Di dalamnya adalah daftar konstanta A, B, dan C untuk persamaan Antoine.

$$\log_{10} P_i^* = A_i - \frac{B_i}{C_i + T} \tag{5.4.1}$$

Tabel 5.1 Konstanta Antoine

	Etilen (1)	Etana (2)	Propana (3)	n-butana (4)
A	3,86690	3,93264	3,97721	3,84431
B	584,146	659,739	819,296	909,65
C	-18,307	-16,719	-24,417	-36,146
z	0,10	0,30	0,40	0,20

dimana:

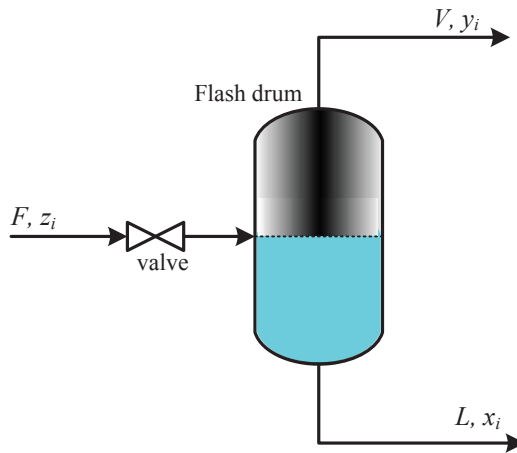
$P_i^*$  = tekanan uap komponen i (atm)

$T$  = temperatur (K)

$A, B, C$  = konstanta Antoine

$F$  adalah laju umpan molar, dan  $z$  adalah mol fraksi masing-masing komponen dalam umpan,  $V$  adalah laju alir molar uap, dan  $y$  adalah mol fraksi setiap komponen dalam uap,  $L$  adalah laju alir molar cairan dan  $x$  adalah fraksi mol setiap komponen dalam cairan.

Ingin dicari fraksi campuran dalam fasa uap ketika campuran umpan di-flash pada suhu  $60^{\circ}\text{C}$  dan tekanan  $40\text{ atm}$ .



**Gambar 5.2.** Diagram skematik flash drum

**Penyelesaian :**

Asumsi campuran ideal (dimana berlaku hukum Raoult) , berlaku persamaan kesetimbangan berikut:

$$K_i = \frac{y_i}{x_i} = \frac{P_i^*}{P}; i = 1, 2, \dots, n_c \tag{5.4.2}$$

Dimana  $k_i$  adalah konstanta kesetimbangan untuk komponen  $i$ ,  $P$  adalah tekanan total dan  $n_c$  jumlah komponen.

Persamaan neraca massa untuk flash drum selanjutnya dapat dituliskan berikut:





```

' num2str(z3) ' ' num2str(z4) ]);
disp([' Vapor ' num2str(y1) ' ' num2str(y2) ' '
num2str(y3) ' ' num2str(y4) ]);
disp([' Liquid ' num2str(x1) ' ' num2str(x2) ' '
num2str(x3) ' ' num2str(x4) ]);
disp('-----')
%
function falpha = flash(alpha)
global z1 z2 z3 z4
Pt=40;% dalam satuan atm
T=60+273;% dalam satuan K
P1 = 10^(3.86690-(584.146/(-18.307+T)));
P2 = 10^(3.93264-(659.739/(-16.719+T)));
P3 = 10^(3.97721-(819.296/(-24.417+T)));
P4 = 10^(3.84431-(909.65/(-36.146+T)));
K1 = P1/Pt; K2=P2/Pt; K3=P3/Pt; K4=P4/Pt;
x1 = (z1/(1+alpha*(K1-1)));
x2 = (z2/(1+alpha*(K2-1)));
x3 = (z3/(1+alpha*(K3-1)));
x4 = (z4/(1+alpha*(K4-1)));
y1 = K1*x1; y2 = K2*x2; y3 = K3*x3; y4 = K4*x4;
falpha = x1*(1-K1)+x2*(1-K2)+x3*(1-K3)+x4*(1-K4);

```

**Hasil Running:**

```

Nilai variabel pada tebakan awal
Nilai alpha tebakan: 0.5, Fungsi nilai : 0.2921
Nilai variabel pada penyelesaian
Nilai alpha: 0.036447, Fungsi nilai alpha: -2.7756e-017
-----
Component      Ethylene      Ethane      Propane      n-Butane
-----
Feed            0.1           0.3         0.4          0.2
Vapor           0.24241      0.51282     0.21368      0.031091
Liquid         0.094613     0.29195     0.40705      0.20639
-----

```

**Contoh 5.5. Mencari volume spesifik gas dengan persamaan Van der Waal**

Diketahui persamaan Van Der Waals sebagai berikut:

$$\left(P + \frac{a}{v^2}\right)(v - b) = RT \tag{5.5.1}$$

$$a = \frac{27}{64} \left( \frac{R^2 T_c^2}{P_c} \right) \tag{5.5.2}$$

$$b = \frac{RT_c}{8P_c} \tag{5.5.3}$$

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

Diketahui : Tekanan reduksi

$$P_r = \frac{P}{P_c}$$

(5.5.4)

Faktor kompresibilitas

$$Z = \frac{Pv}{RT}$$

(5.5.5)

di samping

- Hitung volum molar dan faktor kompresibilitas untuk gas amonia pada  $P = 56$  atm dan suhu = 450 K dengan menggunakan persamaan keadaan Van Der Waals?
- Ulangi perhitungan untuk tekanan reduksi berikut :  $P_r = 1, 2, 4, 10,$  dan  $20$  !
- Bagaimana hubungan antara faktor kompresibilitas dan tekanan reduksi (gambarakan dalam sebuah grafik) ?

### Penyelesaian :

Persamaan (5.4.1) perlu disusun kembali sehingga menjadi bentuk :

Untuk mendapatkan nilai  $v$  maka:

$$F(v) = 0 \tag{5.5.6}$$

Berikut penulisan program untuk penyelesaian soal di atas.

```
% Program Utama : Contoh_55.m
clear all % Membersihkan memory
global P a b R T % Mendefinisikan variabel2 ini sebagai
% variabel global sehingga dapat terbaca oleh
```

```

% Sub_ProgramVol_Mol.m
% Memasukkan konstanta2 & parameter keadaan
    Pc = 111.3;           % Tekanan kritis dlm atm
    Tc = 405.5;         % Suhu kritis dlm K
    R = 0.08206;        % Konstanta gas dlm atm.L/mol.K
    T = 450;            % dlm K
    P = 56;             % dlm atm

% Konstanta Van Der Waals
    a = 27/64*R^2*Tc^2/Pc;
    b = R*Tc/(8*Pc);

volguess = R*T/P;      % Memperkirakan nilai volum molar dengan
                       % persamaan gas ideal

vol = fzero ('Vol_Mol', volguess); % Menghitung volum molar
Z = P*vol/(R*T);      % Menghitung faktor
                       % kompresibilitas

disp('Pada suhu (K)           = ');T
disp('dan tekanan (atm)       = ');P
disp('Diperoleh volum molar (L) = ');vol
disp('Dan faktor kompresibilitas = ');Z

% Nilai dari berbagai tekanan reduksi disimpan dalam bentuk vektor
Pred = [1 2 4 10 20];
% Perulangan perhitungan dengan mem-variasi tekanan
for k = 1 : 5
    P = Pc*Pred(k);
    volguess = R*T/P;
    vol = fzero ('Vol_Mol',volguess);
    Z = P*vol/(R*T);
    result(k,1) = Pred(k);
    result(k,2) = vol;
    result(k,3) = P*vol/(R*T);
end
disp('Diperoleh vol molar dan faktor Z utk berbagai Pred sbb :')
disp('-----')
disp('=====')
disp('    Pred    Molar Vol    faktor Z    ')
disp('=====')
disp(result)
disp('=====')

plot(result(:,1),result(:,3),'or-')
title('Grafik Hubungan Faktor Kompresibilitas Terhadap Tekanan

```

```

Reduksi')
xlabel('Tekanan Reduksi, Pr')
ylabel('Faktor Kompresibilitas, Z')
-----

% Sub-Program : Vol_Mol.m
function y = Vol_Mol(vol)
global P a b R T
y = P*vol^3-(P*b+R*T)*vol^2+a*vol-a*b;
-----

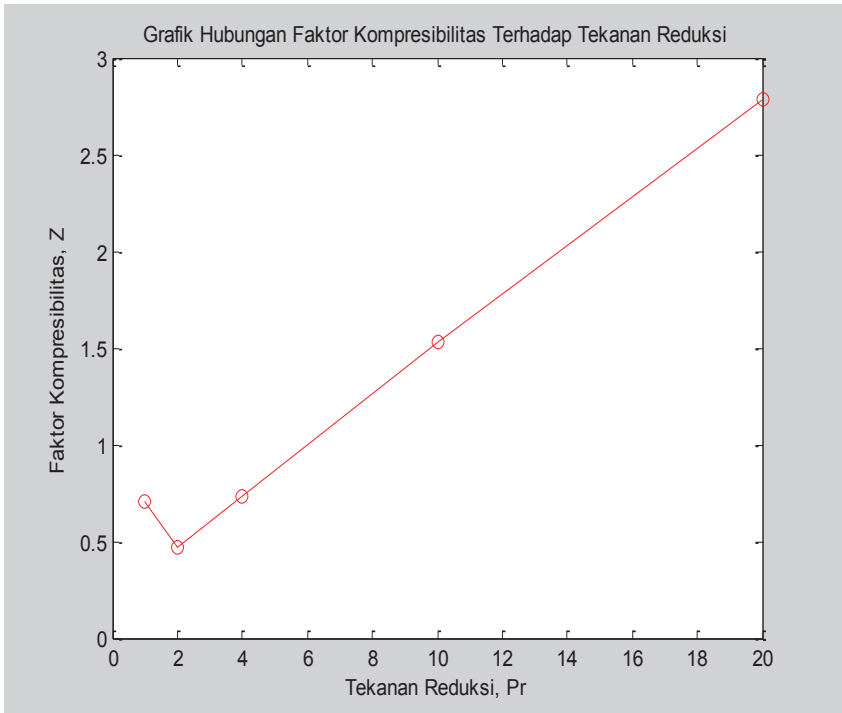
```

## Hasil Program

```

-----
>> Contoh_55
Pada suhu (K) =
T =
    450
dan tekanan (atm) =
P =
    56
Diperoleh volum molar (L) =
vol =
    0.5749
Dan faktor kompresibilitas =
Z =
    0.8718
Diperoleh vol molar dan faktor Z utk berbagai Pred sbb :
-----
=====
      Pred      Molar Vol      faktor Z
=====
      1.0000      0.2335      0.7038
      2.0000      0.0773      0.4658
      4.0000      0.0607      0.7313
     10.0000      0.0509      1.5334
     20.0000      0.0462      2.7835
=====

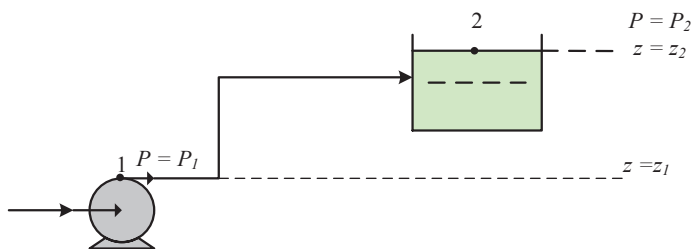
```



**Gambar 5.2.** Grafik kompresibilitas dan tekanan reduksi

### Contoh 5.6. Perhitungan laju Alir (*Flow Rate*) dalam suatu Pipa

Gambar di bawah menunjukkan suatu pipa yang megiriman air pada suhu konstan  $T = 60^\circ\text{C}$  dari titik 1 dimana tekanan  $P_1 = 150$  psig dan elevasi adalah  $z_1 = 0$  ft ke titik 2 dimana tekanan adalah atmosferik dan elevasi  $z_2 = 300$  ft.



**Gambar 5.3.** Grafik sistem perpipaan fluida

Hitung kecepatan aliran dan laju alir untuk pipa dengan panjang efektif  $L = 500, 1000, 10000$  ft dan dibuat diameter nominal 4,5,6 dan 8 in dengan schedule 40 pipa baja komersial. Buatlah grafik kecepatan aliran,  $v$  vs  $D$  dan  $L$  juga grafik laju alir,  $q$  vs  $D$  dan  $L$ .

**Penyelesaian :**

Persamaan neraca energi mekanis pada cairan *incompressible* (tak mampu mampat) yang dapat diterapkan untuk kasus ini adalah:

$$-\frac{1}{2}v^2 + g \Delta z + \frac{g_c \Delta P}{\rho} + 2 \frac{f_F L v^2}{D} = 0 \tag{5.6.1}$$

Dimana  $v$  adalah kecepatan aliran dalam ft/s,  $g$  adalah percepatan gravitasi diberikan dengan  $g = 32,174 \text{ ft/s}^2$ ,  $\Delta z = z_2 - z_1$  adalah perbedaan elevasi (ft),  $g_c$  adalah faktor konversi (dalam satuan British  $g_c = 32,174 \text{ ft.lbm/lbf.s}^2$ ),  $\Delta P = P_2 - P_1$  adalah perbedaan tekanan (lbm/ft<sup>2</sup>),  $\rho$  adalah densitas fluida (untuk air pada  $T = 60^\circ\text{F}$ ,  $\rho = 62,3 \text{ lbm/ft}^3$ ),  $f_F$  adalah faktor gesekan *Fanning*,  $L$  adalah panjang pipa (ft) dan  $D$  adalah diameter dalam pipa (ft). Untuk menggunakan metode *Successive Substitution* persamaan di atas harus dituliskan kembali:

$$v = \sqrt{\frac{g \Delta z + \frac{g_c \Delta P}{\rho}}{0,5 - 2 \frac{f_F L}{D}}} \tag{5.6.2}$$

Persamaan yang digunakan untuk menghitung faktor gesekan *Fanning* tergantung pada regim aliran. Regim aliran ditentukan oleh bilangan Reynold,  $Re$ . Bilangan Reynold adalah bilangan tak berdimensi  $Re = \rho v D / \mu$  dimana  $\mu$  adalah viskositas (untuk air pada  $T = 60^\circ\text{F}$ ,  $\mu = 0.76 \times 10^{-3} \text{ lb}_m/\text{ft.s}$ ). Untuk aliran laminar ( $Re < 2100$ ) faktor gesekan *Fanning* dapat dihitung dari persamaan :

$$f_F = \frac{16}{Re} \tag{5.6.3}$$

Untuk liran turbulen ( $Re > 2100$ ) dapat digunakan persamaan

:

$$f_F = \frac{1}{16 \left\{ \log \left[ \frac{\varepsilon/D}{3,7} - \frac{5,02}{Re} \log \left( \frac{\varepsilon/D}{3,7} - \frac{5,02}{Re} \right) \right] \right\}^2} \quad (5.6.4)$$

Berikut program MATLAB untuk kasus di atas.

File `flowvelocity.m` untuk mengeksekusi persamaan utama.

```
function flowvelocity
clear, clc, format short g, format compact
D_list=[4.026/12 5.047/12 6.065/12 7.981/12];% Inside diameter of pipe
(ft)
T = 60; %Temperature (deg. F)
for i = 1:4
D = D_list(i);
j=0;
for L=500:500:10000
j = j+1;
L_list(j)=L; % Effective length of pipe (ft)
[v(j,i), fval]=fzero(@NLEfun, [1 20], [], D, L, T);
if abs(fval)>1e-10
disp([' No Convergence for L = ' num2str(L) ' and D = '
num2str(D)]);
end
q(j,i) = v(j,i) * pi * D ^ 2 / 4* 7.481 * 60; %Flow rate (gpm)
end
end
disp(' Flow Velocity (ft/s) versus Pipe Length and
Diameter');
disp(' Tabular Results');
disp('');
disp(' L\D D=4" D=5" D=6" D=8"');
Res=[L_list v];
disp(Res);
plot(L_list,v(:,1), 'L_list,v(:,2), '+' ,L_list,v(:,3), '*', L_list,v(:,4), 'x');
legend(' D=4"', ' D=5"', ' D=6"', ' D=8"');
title(' Flow Velocity')
xlabel('Pipe Length (ft)');
ylabel('Velocity (ft/s)');
pause
disp(' Flow Rate (gpm) versus Pipe Length and Diameter');
disp(' Tabular Results');
disp('');
disp(' L\D D=4" D=5" D=6" D=8"');
Res=[L_list q(:,1) q(:,2) q(:,3) q(:,4)];
disp(Res);
plot(L_list,q(:,1), '-
', L_list,q(:,2), '+' ,L_list,q(:,3), '*', L_list,q(:,4), 'x');
legend(' D=4"', ' D=5"', ' D=6"', ' D=8"');
title(' Flow rate')
xlabel('Pipe Length (ft)');
ylabel('Flow rate (gpm)');
```

File *NLEfun* sebagai persamaan utama.

```
function fv = NLEfun(v,D,L,T)
epsilon = 0.00015; %Surface rougness of the pipe (ft)
rho = 62.122 + T * (0.0122 + T * (-0.000154 + T * (0.000000265 - (T *
0.000000000224))))); %Fluid density (lb/cu. ft.)
deltaz = 300; %Elevation difference (ft)
deltaP = -150; %Pressure difference (psi)
vis = exp(-11.0318 + 1057.51 / (T + 214.624)); %Fluid viscosity
(lbm/ft-s)
pi = 3.1416; %The constant pi
eoD = epsilon / D; %Pipe roughness to diameter ratio (dimensionless)
Re = D * v * rho / vis; %Reynolds number (dimesionless)
if (Re < 2100) %Fanning friction factor (dimensionless)
    fF = 16 / Re;
else
    fF = 1 / (16 * log10(eoD / 3.7 - (5.02 * log10(eoD / 3.7 + 14.5 /
Re) / Re)) ^ 2);
end
fv = v - sqrt((32.174 * deltaz + deltaP * 144 * 32.174 / rho) / (0.5
- (2 * fF * L / D))); %Flow velocity (ft/s)
```

Hasil perhitungan pada *command window*:

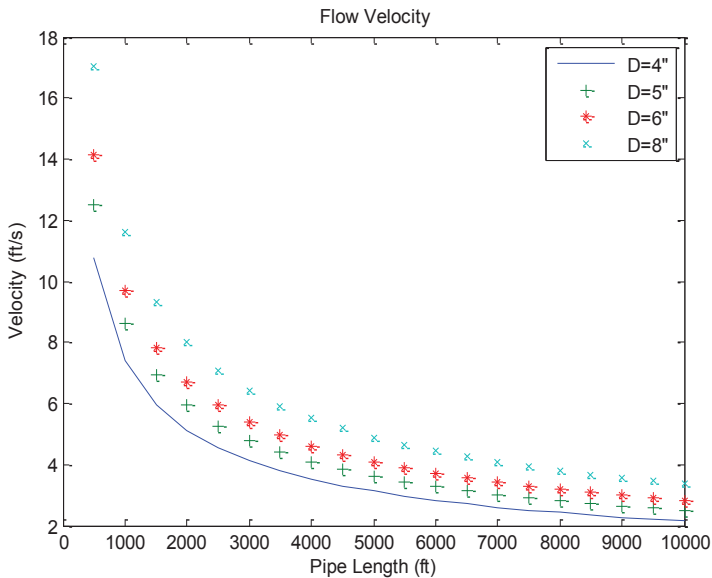
Flow Velocity (ft/s) versus Pipe Length and Diameter  
Tabular Results

L\D	D=4"	D=5"	D=6"	D=8"
500	10.773	12.516	14.15	17.035
1000	7.4207	8.6048	9.7032	11.613
1500	5.9721	6.9243	7.8051	9.3295
2000	5.1188	5.9361	6.6912	7.9953
2500	4.5409	5.2674	5.9382	7.0953
3000	4.1168	4.7769	5.3861	6.4362
3500	3.7888	4.3975	4.9592	5.927
4000	3.5255	4.093	4.6166	5.5185
4500	3.3082	3.8416	4.3338	5.1815
5000	3.1249	3.6297	4.0953	4.8973
5500	2.9677	3.4478	3.8907	4.6535
6000	2.8309	3.2896	3.7128	4.4415
6500	2.7106	3.1504	3.5561	4.2548
7000	2.6036	3.0266	3.4169	4.0889
7500	2.5077	2.9156	3.292	3.9402



8000	2.4211	2.8154	3.1793	3.8059
8500	2.3424	2.7244	3.0769	3.6838
9000	2.2706	2.6412	2.9832	3.5723
9500	2.2046	2.5648	2.8972	3.4698
10000	2.1437	2.4943	2.8179	3.3752

Grafik yang dihasilkan:



Gambar 5.5 Grafik hasil perhitungan

### 5.3 RANGKUMAN

- Banyak peristiwa fisis dan kimia yang memiliki model persamaan berupa persamaan nonlinier
- Persamaan nonlinier polynomial akan memiliki akar persamaan untuk memenuhi penyelesaian persamaan yang sesuai

- Penyelesaian persamaan non linier tunggal dengan MATLAB menggunakan fungsi *fzero* akan diperoleh satu nilai yang memberikan penyelesaian yang tepat.
- Penyelesaian persamaan non linier tunggal dengan MATLAB menggunakan fungsi *roots* akan diperoleh akar-akar yang memberikan penyelesaian yang tepat.

## 5.4 RUJUKAN

- 1) Constantinidis and Mustoufi, 1999, *Numerical Methodes for Chemical Engineers with MATLAB Application*, hal 1-4. Prentice-Hall: Englewood Cliffs, NJ.
- 2) Finlayson, B.A., 2006, *Introduction to Chemical Engineering Computing*, hal 5-15. John Wiley & Sons Inc., New Jersey
- 3) Ramirez, W.F. 1998. *Computational Methods in Process Simulation*. Hal: 54-56. Elsevier Science & Technology Books.
- 4) Sandler, S. I., 2006. *Chemical, Biochemical, and Engineering Thermodynamics*, 4<sup>th</sup>.ed. John Wiley. New York.

# PERSAMAAN NON-LINIER SIMULTAN

## 6.1 PENDAHULUAN

Materi ini membahas tentang persamaan non-linier multivariabel. Setelah menyelesaikan bab ini diharapkan mahasiswa akan mampu menyelesaikan persoalan dalam teknik kimia yang dapat dinyatakan dengan persamaan non-linier multivariabel. Secara khusus mahasiswa diharapkan mampu menyusun peristiwa fisis dan kimia dalam teknik kimia ke dalam persamaan matematis yang berbentuk nonlinier dan mampu menyelesaikannya dengan fungsi MATLAB yaitu *fsolve*,

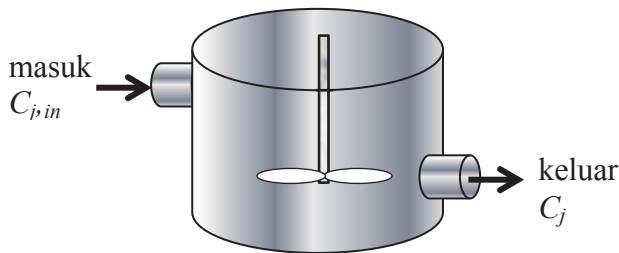
Materi ini sangat penting, mengingat banyak peristiwa dalam bidang teknik kimia yang berkaitan dengan termodinamika dan bidang lainnya memiliki bentuk persamaan non-linier yang harus dipecahkan untuk keperluan proses evaluasi, optimasi dan perancangan. Oleh karena itu, memahami dan menguasai materi dari bab ini akan membantu mahasiswa dalam memecahkan persoalan perhitungan tentang termodinamika (*bubble point*, *dew point* dan perhitungan *isothermal flash* untuk campuran multikomponen non-ideal, perhitungan *adiabatic flash* untuk campuran multikomponen),

distribusi aliran dalam jaringan pipa, perhitungan kesetimbangan kimia kompleks dan kasus lainnya yang berkaitan dengan persamaan non linier. Pendalaman materi tentang topik yang dibahas pada bab ini dapat dibaca buku yang menjadi rujukan sebagaimana dituliskan halamannya.

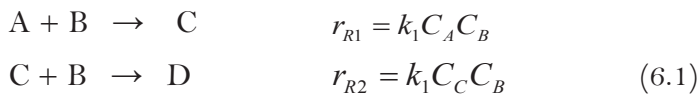
## 6.2 PERSAMAAN NON LINIER MULTIVARIABEL

Pada bab 5 telah dibahas tentang penyelesaian persamaan non-linier variabel tunggal. Selanjutnya pada bab ini akan dibahas persamaan nonlinier multivariabel atau simultan. Banyak persoalan dalam bidang teknik kimia memerlukan penyelesaian persamaan non-linier.

Sebagai contoh, misalkan suatu reaktor *continuous stirred-tank reactor* (CSTR), dioperasikan secara isothermal dimana perubahan volum akibat reaksi diabaikan, dalam mode *overflow* dengan volume fluida konstan, dan dengan dua reaksi kimia (diasumsikan elementer) seperti gambar 6.1 di bawah.



**Gambar 6.1.** CSTR dengan dua reaksi kimia



Dalam CSTR, kita berasumsi bahwa reaktor bercampur secara sempurna sehingga konsentrasi komponen setiap bagian dalam reaktor dianggap seragam. Artinya, setiap titik dalam reaktor memiliki konsentrasi yang sama untuk setiap komponen, maka disusun seperangkat persamaan neraca massa:

$$\begin{aligned} \frac{d}{dt}(V C_A) &= v(C_{A,in} - C_A) + V(-k_1 C_A C_B) \\ \frac{d}{dt}(V C_B) &= v(C_{B,in} - C_B) + V(-k_1 C_A C_B - k_2 C_C C_B) \\ \frac{d}{dt}(V C_C) &= v(C_{C,in} - C_C) + V(-k_1 C_A C_B - k_2 C_C C_B) \\ \frac{d}{dt}(V C_D) &= v(C_{D,in} - C_D) + V(k_2 C_C C_B) \end{aligned} \quad (6.2)$$

$v$  adalah laju alir volumetrik aliran umpan dan keluaran,  $V$  adalah volume tetap reaktor,  $C_j$  adalah konsentrasi spesies  $j$  dalam reaktor (dan pada aliran keluaran),  $C_{j,in}$  adalah konsentrasi spesies  $j$  di aliran masuk dan  $k_1, k_2$  adalah konstanta laju reaksi kimia dari masing-masing reaksi. Pada keadaan *steady state*, derivatif waktu di sebelah kiri pada persamaan (6.2) adalah nol, dan konsentrasi setiap spesies dalam reaktor memenuhi seperangkat dari empat persamaan aljabar nonlinier. Untuk mendapatkan persamaan ini dalam bentuk standar, kita definisikan variabel yang tidak diketahui sebagai:

$$x_1 = C_A \quad x_2 = C_B \quad x_3 = C_C \quad x_4 = C_D \quad (6.3)$$

untuk memperoleh sistem persamaan aljabar.

$$\begin{aligned} v(C_{A,in} - x_1) + V(-k_1 x_1 x_2) &= 0 \\ v(C_{B,in} - x_2) + V(-k_1 x_1 x_2 - k_2 x_3 x_2) &= 0 \\ v(C_{C,in} - x_3) + V(-k_1 x_1 x_2 - k_2 x_3 x_2) &= 0 \\ v(C_{D,in} - x_4) + V(k_2 x_3 x_2) &= 0 \end{aligned} \quad (6.4)$$

Bentuk umum persamaan non-linier simultan adalah:

$$\begin{aligned} f(1) &= f_1(x_1, x_2, x_3, \dots, x_n) = 0 \\ f(2) &= f_2(x_1, x_2, x_3, \dots, x_n) = 0 \\ f(3) &= f_3(x_1, x_2, x_3, \dots, x_n) = 0 \\ &\dots\dots\dots \\ f(n) &= f_n(x_1, x_2, x_3, \dots, x_n) = 0 \end{aligned}$$

Dalam hal ini akan dicari harga  $x_1, x_2, \dots, x_n$ .

Fungsi *fsolve* pada *MATLAB Optimization Toolbox* digunakan untuk menyelesaikan sistem persamaan non-linier simultan. *fsolve* digunakan untuk menyelesaikan problem tertentu dengan bentuk persamaan

$$F(x) = 0$$

Untuk  $x$ , dimana  $x$  adalah suatu vektor dan  $F(x)$  adalah fungsi yang mengembalikan nilai suatu vektor. Fungsi ini dapat dijalankan dengan beberapa pengaturan parameter input dan output yang berbeda.

Syntax untuk *fsolve* adalah (bisa digunakan salah satunya):

```
x = fsolve(fun,x0)
x = fsolve(fun,x0,options)
x = fsolve(problem)
```

Penjelasan:

*fsolve* menemukan akar suatu sistem persamaan nonlinier simultan.

$x = \text{fsolve}(\text{fun},x_0)$ , mulai pada  $x_0$  dan mencoba untuk menyelesaikan persamaan yang dideskripsikan dalam *fun*

$x = \text{fsolve}(\text{fun},x_0,\text{options})$ , menyelesaikan persamaan dengan pilihan optimasi tertentu dalam struktur *options*. Menggunakan *optimset* untuk mengatur pilihan ini.

$x = \text{fsolve}(\text{problem})$ , menyelesaikan *problem*, dimana *problem* adalah struktur yang dinyatakan dalam *Input Arguments*.

### ***Input Arguments***

**Fun** Sistem persamaan non linier untuk menyelesaikan *fun* adalah fungsi yang menerima suatu vektor  $x$  dan mengembalikan suatu vektor  $F$ , persamaan nonlinier dievaluasi pada  $x$ . Fungsi *fun* dapat dispesifikasi sebagai fungsi yang menangani file  $x = \text{fsolve}(@\text{myfun},x_0)$  dimana *myfun* suatu fungsi MATLAB seperti fungsi

```
F = myfun(x)
F = ... %menghitung nilai fungsi pada x
fun dapat juga sebagai fungsi yang menangani suatu
anonymous function.
```

```
x = fsolve(@ (x) sin(x.*x), x0);
```

Jika pengguna mendefinisikan nilai untuk  $x$  dan  $F$  adalah matriks, mereka dikonversi ke suatu vektor menggunakan linear indexing.

### Contoh 6.1 Persamaan Non-Linier Dua Variabel

Contoh berikut menyelesaikan sistem dua persamaan dengan dua variabel yang tidak diketahui:

$$\begin{aligned} 2x_1 - x_2 &= e^{-x_1} \\ -x_1 + 2x_2 &= e^{-x_2} \end{aligned}$$

Mulai pencarian penyelesaian pada  $x_0 = [-5 \ -5]$

#### Penyelesaian:

Persamaan di atas dituliskan kembali dalam bentuk  $F(x) = 0$  :

$$\begin{aligned} 2x_1 - x_2 - e^{-x_1} &= 0 \\ -x_1 + 2x_2 - e^{-x_2} &= 0 \end{aligned}$$

Pertama, tuliskan suatu file untuk menghitung  $F$ , nilai persamaan pada  $x$ .

```
function F = myfun(x)
F = [2*x(1) - x(2) - exp(-x(1));
     -x(1) + 2*x(2) - exp(-x(2))];
```

Simpan file fungsi ini sebagai myfun.m di lokasi path MATLAB.

Selanjutnya, atur nilai awal dan options untuk memanggil fsolve

```
x0 = [-5; -5]; % tebakan awal untuk penyelesaian
options = optimset('Display', 'iter'); % pilihan
untuk menampilkan output dan iterasi
[x, fval] = fsolve(@myfun, x0, options) %
memanggil penyelesaian
```

Setelah beberapa iterasi, fsolve menemukan jawaban:

Iteration	Func-count	Norm of f(x)	First-order step	Trust-region optimality	radius
0	3	23535.6		2.29e+004	1
1	6	6001.72	1	5.75e+003	1
2	9	1573.51	1	1.47e+003	1
3	12	427.226	1	388	1
4	15	119.763	1	107	1
5	18	33.5206	1	30.8	1
6	21	8.35208	1	9.05	1
7	24	1.21394	1	2.26	1
8	27	0.016329	0.759511	0.206	2.5
9	30	3.51575e-006	0.111927	0.00294	2.5
10	33	1.64763e-013	0.00169132	6.36e-007	2.5

Equation solved.

fsolve completed because the vector of function values is near zeros as measured by the default value of the function tolerance, and the problem appears regular as measured by the gradient.

```
x =
    0.5671
    0.5671
fval =
    1.0e-006 *
   -0.4059
   -0.4059
```

### Contoh 6.2 Menentukan nilai x dan y dari persamaan

$$x^3 - 3xy^2 = 1/2$$

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

#### Langkah penyelesaian:

Langkah 1 : Buat terlebih dahulu fungsi sistem persamaan non-linier dalam m-file.

```
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];
```



Langkah 2 : Buat program pengekseski menggunakan fsolve pada m-file yang

Langkah 3 : Jalankan program pengekseski.

```
>>[x,fval] = fsolve('sistem',[1 2])

Optimization terminated: first-order optimality is less than options.TolFun.

x =

    2.5198    1.5874

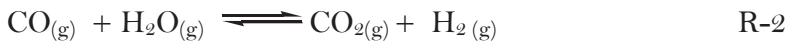
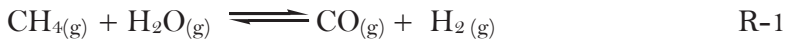
fval =

    1.0e-010 *

    0.1930
    0.0966
```

### Contoh 6.3 Stoikiometri Reaksi Kesetimbangan

Reaksi reformasi steam (*steam reforming*) berlangsung menurut serangkaian reaksi kesetimbangan berikut:



Pada suhu 2000 K harga konstanta kesetimbangan untuk masing-masing reaksi adalah  $1,930 \times 10^{-4}$  dan 5,528. Tentukan komposisi kesetimbangan komponen-komponen apabila Gas umpan berkomporsi 20%  $\text{CH}_4(\text{g})$  dan 80%  $\text{H}_2\text{O}(\text{g})$  berada pada kondisi suhu 2000 K dan tekanan 1 atm.

### Jawaban

Misal ditetapkan basis perhitungan 10 mol gas umpan .

$\varepsilon_1$  = tingkat reaksi dari reaksi pertama

$\varepsilon_2$  = tingkat reaksi dari reaksi kedua

Fraksi mol kesetimbangan setiap komponen dapat dinyatakan sebagai berikut:

$$y_{CO} = \frac{\varepsilon_1 - \varepsilon_2}{10 + 2\varepsilon_1} \quad y_{H_2} = \frac{3\varepsilon_1 + \varepsilon_2}{10 + 2\varepsilon_1}$$

$$y_{H_2O} = \frac{8 - \varepsilon_1 - \varepsilon_2}{10 + 2\varepsilon_1}$$

$$y_{CO_2} = \frac{\varepsilon_2}{10 + 2\varepsilon_1} \quad y_{CH_4} = \frac{2 - \varepsilon_1}{10 + 2\varepsilon_1}$$

Persamaan konstanta kesetimbangan dinyatakan sebagai berikut:

$$K_1 = \frac{y_O y_{H_2}^3 P^2}{y_{CH_4} y_{H_2O}}$$

$$K_2 = \frac{y_{CO_2} y_{H_2}}{y_{CO} y_{H_2O}}$$

Jika dinyatakan sebagian tingkat reaksi:

$$K_1 = \frac{(\varepsilon_1 - \varepsilon_2)(3\varepsilon_1 - \varepsilon_2)^3}{(2 - \varepsilon_1)(8 - \varepsilon_1 - \varepsilon_2)(10 + 2\varepsilon_1)^2}$$

$$K_2 = \frac{\varepsilon_2(3\varepsilon_1 + \varepsilon_2)}{(\varepsilon_1 - \varepsilon_2)(8 - \varepsilon_1 - \varepsilon_2)}$$

Kedua persamaan di atas merupakan persamaan non-linier simultan.  $\varepsilon_1$  dan  $\varepsilon_2$  merupakan variable yang nilainya akan dicari.

Berikut ini pemrograman MATLAB-nya.

```
function y = equilibrium(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(@equilibrium (e,K1,K2), [1 0.5])
```

Eksekusi di MATLAB *command window*

```
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.
```

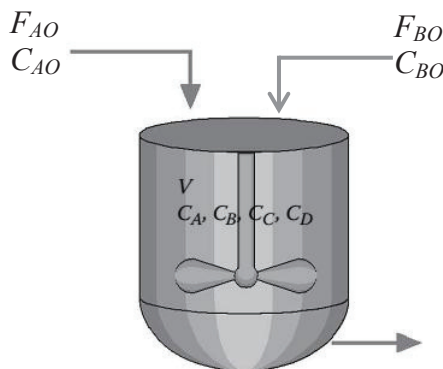
```
e =
    0.7480    0.6920
```

### Contoh 6.4 Reaksi Eksotermik Reversibel

Suatu reaksi eksotermik dapat balik dengan dua reaktan dan dua produk dinyatakan dengan persamaan :



Reaksi berlangsung dalam sebuah CSTR yang memiliki dua aliran umpan  $F_{A0}$  dan  $F_{B0}$  ( $\text{m}^3/\text{s}$ ), dengan komposisi  $C_{A0}$  dan  $C_{B0}$  ( $\text{kmol}/\text{m}^3$ ) seperti Gambar 6.2.



**Gambar 6.2.** CSTR dengan reaksi kimia *reversible*

Laju reaksi total adalah selisih laju reaksi forward dan reverse. Kita asumsikan order satu independen pada komposisi.

$$r = k_F C_A C_B - k_R C_C C_D$$

$$= k_{0F} \exp\left(\frac{-E_F}{RT}\right) C_A C_B - k_{0R} \exp\left(\frac{-E_R}{RT}\right) C_C C_D$$

Ingin dicari komposisi keluaran reaktor pada suhu 300 sampai 400 K dengan incremental 10 K, jika diketahui parameter-parameter sebagai berikut:

Pada $T = 333 \text{ K}$ , nilai $K_{eq}$	= 10
$k_{or}$	= $((29,3 \cdot 10^6)/K_{eq}) \exp(139,4 \cdot 10^6/R/333)$
$k_F$	= $4,239 \cdot 10^6 \exp(-69,71 \cdot 10^6/RT) \text{ m}^3 \text{ s}^{-1} \text{ kmol}^{-1}$
$k_R$	= $k_{or} \cdot \exp(-139,4 \cdot 10^6/RT) \text{ m}^3 \text{ s}^{-1} \text{ kmol}^{-1}$
Konstanta gas, $R$	= $8.314 \text{ J/kmol} \cdot \text{K}$
Volume reaktor, $V_r$	= $100 \text{ m}^3$
Laju alir umpan, $F_{A0}$	= $0,004377 \text{ m}^3/\text{s}$
Laju alir umpan, $F_{B0}$	= $0,004377 \text{ m}^3/\text{s}$
Komposisi umpan, $C_{A0}$	= $8,01 \text{ kmol}/\text{m}^3$
Komposisi umpan, $C_{B0}$	= $8,01 \text{ kmol}/\text{m}^3$

### Penyelesaian:

Terdapat empat komponen dalam sistem ini, sehingga empat neraca komponen dapat dituliskan:

Komponen A (kmol A/s):

$$F_{A0} C_{A0} = F C_A + V_R r$$

$$= F C_A + V_R (k_F C_A C_B - k_R C_C C_D)$$

Komponen B (kmol B/s):

$$F_{B0} C_{A0} = F C_B + V_R (k_F C_A C_B - k_R C_C C_D)$$

Komponen C (kmol C/s):

$$0 = FC_C + V_R(-k_F C_A C_B + k_R C_C C_D)$$

Komponen D (kmol D/s):

$$0 = FC_D + V_R(-k_F C_A C_B + k_R C_C C_D)$$

Asumsi bahwa kita mengetahui nilai kondisi umpan, volume reaktor, dan suhu reaksi (konstanta laju reaksi  $k_F$  dan  $k_R$ ), sehingga ada lima variabel yang tidak diketahui dari empat persamaan di atas :  $F$ ,  $C_A$ ,  $C_B$ ,  $C_C$  dan  $C_D$ . Karena itu diperlukan penambahan persamaan. Jika densitas semua komponen adalah sama, laju alir volumetrik keluaran akan sama terhadap laju alir volumetrik dua umpan.

$$F_{A0} + F_{B0} = F$$

Terdapat empat persamaan dan empat variabel yang tidak diketahui, sehingga persamaan di atas dapat diselesaikan. Namun penyelesaian persamaan aljabar non linier simultan ini sulit jika dilakukan dengan cara biasa, dan akan lebih sulit jika reaksi bukan order satu. Peyelesaian di atas mudah diselesaikan dengan *fsolve function* pada Matlab untuk menentukan empat variabel yang tidak diketahui.

Berikut script program matlab file `cstr_reversible.m` untuk menyelesaikan soal di atas.

### Program utama:

```
function cstrreverse
clc, clear
%function cstrreverse
Keq=10;
kor=((29.3e-6)/Keq)*exp(139.4e6/8314/333);
%set VR dan TR
vr=100;
tre=[300:10:400];
n_tr=size(tre,2);
```

```

for k=1:n_tr
    tr=tre(k);
    kf=4.239e6*exp(-69.7e6/tr/8314);
    kr=kor*exp(-139.4e6/8324/tr);
    fa0=0.004377;ca0=8.01;fb0=0.004377;cb0=8.01;
    flow=fa0+fb0;
    parameters=[vr,kf,kr,fa0,ca0,fb0,cb0];
    %tebakan awal untuk ca cb cc cd
    xo=[0.2 0.3 0.4 0.1];

    options=optimset('MaxFunEvals',1000,'TolFun',1e-8);
    %use fsolve to solve four nonlinier algebraic
egguations

[x,feval]=fsolve('reverse',xo,options,parameters);
    ca=x(1);
    cb=x(2);
    cc=x(3);
    cd=x(4);
    conversion=(fa0*ca0-flow*ca)/(fa0*ca0);
    result(k,1)=tr;
    result(k,2)=conversion;
end

disp('          Hasil perhitungan')
disp('          =====')
disp('          Tr (K)   konversi,x  ')
disp('          -----')
disp(result)
disp('          -----')
plot(result(:,1),result(:,2),'-o')
legend('Vr=100');
title('Konversi versus Tr ')
xlabel('Temperatur Reaksi, Tr (K)');
ylabel('Fraksi konversi, x');

%-----
function f=reverse(x,parameters)
ca=x(1);
cb=x(2);
cc=x(3);
cd=x(4);
vr=parameters(1);
kf=parameters(2);
kr=parameters(3);
fa0=parameters(4);

```

```

ca0=parameters(5);
fb0=parameters(6);
cb0=parameters(7);
flow=fa0+fb0;
ratef=kf*ca*cb;
rater=kr*cc*cd;
%empat persamaan
f(1,1)=fa0*ca0-flow*ca-vr*(ratef-rater);
f(2,1)=fb0*cb0-flow*cb-vr*(ratef-rater);
f(3,1)=-flow*cc-vr*(-ratef+rater);
f(4,1)=-flow*cd-vr*(-ratef+rater);

```

### Hasil running di Command window:

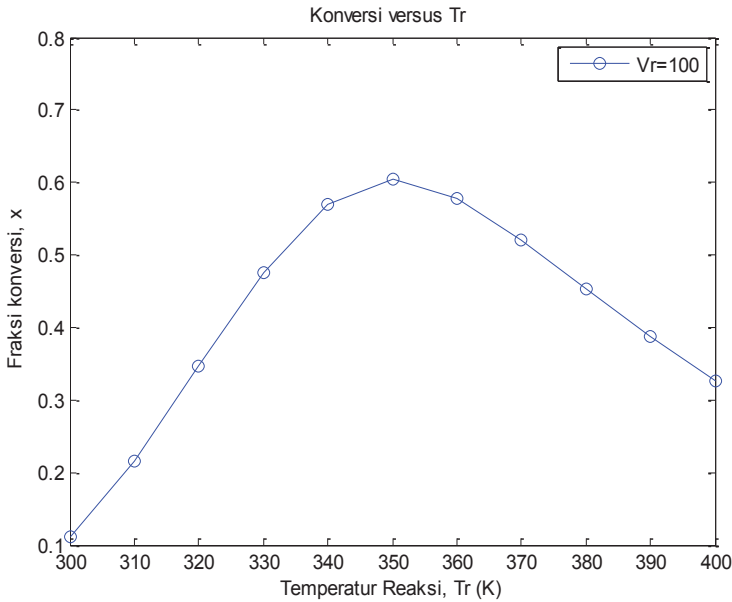
Hasil perhitungan

```

=====
Tr (K)   konversi,x
-----
300      0.11178
310      0.21495
320      0.34569
330      0.47503
340      0.57028
350      0.60423
360      0.57865
370      0.52102
380      0.45359
390      0.38723
400      0.32661
-----

```

Grafik yang dihasilkan:



**Gambar 6.3.** Grafik hasil simulasi

### **Contoh 6.5 Kesetimbangan Kimia Kompleks dengan Minimasi Energi Gibbs**

Gas etana bereaksi dengan uap air menghasilkan hidrogen dengan katalis perengkahan pada suhu  $T = 1000$  K dan tekanan  $P = 1$  atm. Umpan mengandung 4 mol  $H_2O$  per mol  $CH_4$ . Balzisher *et al.* (1972) menyatakan bahwa hanya komponen yang dicantumkan pada tabel 6.1 yang terdapat dalam kesetimbangan (asumsi bahwa tidak ada endapan carbon). Energi Gibbs pembentukan berbagai komponen pada suhu reaksi (1000 K) diberikan pada tabel 6.1. Ingin dicari komposisi kesetimbangan yang keluar dari reaktor berdasarkan data tersebut.



Tabel 6-1. Komponen yang ada pada keluaran Reaktor Perengkahan

No	Komponen	Gibbs Energy (kcal/gmol)	Umpan (gmol)	Estimasi awal keluaran
1	CH <sub>4</sub>	4,61		0,001
2	C <sub>2</sub> H <sub>4</sub>	28,249		0,001
3	C <sub>2</sub> H <sub>2</sub>	40,604		0,001
4	CO <sub>2</sub>	-94,61		0,993
5	CO	-47,942		1
6	O <sub>2</sub>	0		0,001
7	H <sub>2</sub>	0		5,992
8	H <sub>2</sub> O	-46,03	4	1
9	C <sub>2</sub> H <sub>6</sub>	26,13	1	0,001

Formulasikan kasus di atas sebagai masalah minimasi dengan konstrain. Nyatakan konstrain kedalam fungsi objective menggunakan *Lagrange multiplier* dan diferensiasikan fungsi tersebut untuk memperoleh sistem persamaan aljabar non liner simultan.

### Penyelesaian :

Fungsi objective untuk meminimalkan energi Gibbs diberikan dengan :

$$\min_{n_i} \frac{G}{RT} = \sum_{i=1}^c n_i \left( \frac{G_i^o}{RT} + \ln \frac{n_i}{\sum n_i} \right)$$

Dimana  $n_i$  adalah jumlah mol komponen  $i$ ,  $c$  adalah jumlah komponen,  $R$  adalah konstanta gas, dan  $G_i$  adalah energi Gibbs komponen murni  $i$  pada suhu  $T$ . Minimisasi persamaan di atas harus dilakukan dengan konstrain neraca atom:

$$\text{Neraca atom oksigen} \quad g_1 = 2n_4 + n_5 + 2n_6 + n_7 - 4 = 0$$

$$\text{Neraca atom hidrogen} \quad g_2 = 4n_1 + 4n_2 + 2n_3 + 2n_7 + 2n_8 + 6n_9 - 14 = 0$$

$$\text{Neraca atom karbon} \quad g_3 = n_1 + 2n_2 + 2n_3 + n_4 + n_5 + 2n_9 - 2 = 0$$

Ketiga konstrain di atas dapat dikealkan kedalam fungsi objective menggunakan *Langrange Multiplier* :  $\lambda_1, \lambda_2, \lambda_3$ . Fungsi objective yang diperluas adalah:

$$\min_{n_i, \lambda_i} F = \sum_{i=1}^c n_i \left( \frac{G_i^o}{RT} + \ln \frac{n_i}{\sum n_i} \right) + \sum_{j=1}^3 \lambda_j g_j$$

Kondisi minimum untuk fungsi di atas pada titik tertentu adalah bahwa semua turunan parsial  $F$  terhadap  $n_i$  dan  $\lambda_i$  sesuai pada titik tersebut. Turunan parsial  $F$  terhadap  $n_i$  adalah :

$$\frac{\partial F}{\partial n_1} = \frac{G_1^o}{RT} + \ln \frac{n_1}{\sum n_i} + 4\lambda_2 + \lambda_3 = 0$$

Turunan parsial yang lainnya terhadap  $n_i$  dapat diperoleh dengan cara yang sama.

Diharapkan bahwa jumlah komponen utama pada kesetimbangan mendekati nol, lebih dipilih untuk menuliskan persamaan dalam bentuk yang tidak memerlukan perhitungan logaritma angka yang sangat kecil. Penyusunan kembali persamaan di atas diperoleh :

$$n_1 - \sum n_i \exp \left( \frac{G_1^o}{RT} + 4\lambda_2 + \lambda_3 \right) = 0$$

Turunan parsial  $F$  terhadap  $\lambda_1, \lambda_2$ , dan  $\lambda_3$  masing-masing adalah  $g_1, g_2$  dan  $g_3$ .

Fungsi *fsolve* pada *MATLAB Optimization Toolbox* digunakan untuk menyelesaikan sistem persamaan nonlinier simultan.



```

C2H2 = x(11);
O2 = x(12);
R = 1.9872;
sum = H2 + O2 + H2O + CO + CO2 + CH4 + C2H6 + C2H4 +
C2H2;
fx(1,1) = 2 * CO2 + CO + 2 * O2 + H2O - 4; %Oxygen
balance
fx(2,1) = 4 * CH4 + 4 * C2H4 + 2 * C2H2 + 2 * H2 + 2 *
H2O + 6 * C2H6 - 14; %Hydrogen balance
fx(3,1) = CH4 + 2 * C2H4 + 2 * C2H2 + CO2 + CO + 2 *
C2H6 - 2; %Carbon balance
fx(4,1) = log(H2 / sum) + 2 * lamda2;
fx(5,1) = -46.03 / R + log(H2O / sum) + lamda1 + 2 *
lamda2;
fx(6,1) = -47.942 / R + log(CO / sum) + lamda1 +
lamda3;
fx(7,1) = -94.61 / R + log(CO2 / sum) + 2 * lamda1 +
lamda3;
fx(8,1) = 4.61 / R + log(CH4 / sum) + 4 * lamda2 +
lamda3;
fx(9,1) = 26.13 / R + log(C2H6 / sum) + 6 * lamda2 + 2
* lamda3;
fx(10,1) = 28.249 / R + log(C2H4 / sum) + 4 * lamda2 +
2 * lamda3;
fx(11,1) = C2H2 - exp(-(40.604 / R + 2 * lamda2 + 2 *
lamda3)) * sum;
fx(12,1) = O2 - exp(-2 * lamda1) * sum;

```

Hasil running di command window:

Variable values at the initial estimate		
Variable	Value	Function Value
x1	10	-0.0138
x2	10	0
x3	10	0
x4	5.992	19.5944
x5	1	4.6407
x6	1	-6.3214
x7	0.993	-19.8127
x8	0.001	43.2161
x9	0.001	84.0454
x10	0.001	65.1117
x11	0.001	0.001
x12	0.0001	9.9981e-005

Warning: The Gauss-Newton algorithm may be removed in a future release. Additionally, the option `NonlEqnAlgorithm` will be ignored in a future release. Running the Gauss-Newton algorithm. Set option `Algorithm` to either 'trust-region-dogleg', 'trust-region-reflective', or 'levenberg-marquardt' instead.

Maximum number of function evaluations exceeded. Increase `OPTIONS.MaxFunEvals`.

Variable values at the solution

Variable	Value	Function Value
x1	24.4197	0
x2	0.25306	0
x3	1.5598	0
x4	5.3452	-1.1102e-016
x5	1.5216	2.1094e-015
x6	1.3885	1.9984e-015
x7	0.54492	-1.5543e-015
x8	0.066564	0
x9	1.6707e-007	1.3323e-015
x10	9.5412e-008	1.3323e-015
x11	3.157e-010	-7.2378e-025
x12	5.4592e-021	-5.7219e-030

## 6.3 RANGKUMAN

- 1) Banyak peristiwa fisis dan kimia yang memiliki model persamaan berupa persamaan nonlinier tunggal maupun simultan
- 2) Persamaan nonlinear simultan akan memiliki akar-akar persamaan untuk memenuhi penyelesaian persamaan yang sesuai.
- 3) Penyelesaian persamaan nonlinier simultan dengan MATLAB menggunakan fungsi *fsolve* akan diperoleh suatu nilai yang memberikan penyelesaian yang tepat.

## 6.4 LATIHAN

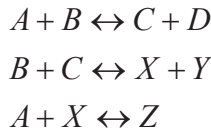
### Latihan 6.1

Selesaikan persamaan berikut:

$$\begin{aligned}(x_1 - 1)^3 + x_2^2 &= 0 \\ x_1 + x_2 &= 1\end{aligned}$$

### Latihan 6.2.

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



Tentukan konsentrasi masing-masing komponen pada saat setimbang jika diketahui  $C_{A0} = C_{B0} = 1.5$ ,  $K_{C1} = 1.06$ ,  $K_{C2} = 2.63$  dan  $K_{C3} = 5$ . 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.

$$K_{C1} = \frac{C_C C_D}{C_A C_B} \quad K_{C2} = \frac{C_X C_Y}{C_B C_C} \quad K_{C3} = \frac{C_Z}{C_A C_X}$$

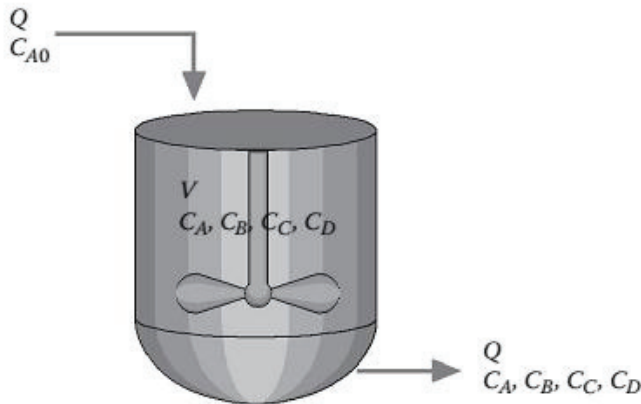
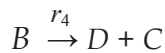
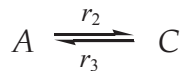
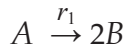
$$C_A = C_{A0} - C_D - C_Z, \quad C_B = C_{B0} - C_D - C_X$$

$$C_C = C_D - C_X, \quad C_Y = C_X + C_Z$$

Pada serangkaian persamaan ini,  $C_A$ ,  $C_B$ ,  $C_C$ ,  $C_D$ ,  $C_X$ ,  $C_Y$  dan  $C_Z$  adalah konsentrasi berbagai komponen pada kesetimbangan yang dihasilkan dari konsentrasi awal hanya  $C_{A0}$  dan  $C_{B0}$ . Konstanta kesetimbangan  $K_{C1}$ ,  $K_{C2}$  dan  $K_{C3}$  telah diketahui nilainya.

### Latihan 6.3

Perhatikan reaktor CSTR pada gambar 6.3 di bawah.  $Q$  adalah laju alir volumetrik (L/s),  $V$  adalah volume reaktor (L), dan  $C_i$  adalah konsentrasi masing-masing empat komponen (gmol/L). Misalkan reaksi hipotetik berikut berlangsung dalam CSTR :



**Gambar 6.3.** Continuous Stirred Tank Reactor

Dimana :

$$\begin{aligned}r_1 &= k_1 C_A \\r_2 &= k_2 C_A^{3/2} \\r_3 &= k_3 C_C^2 \\r_4 &= k_4 C_B^2\end{aligned}$$

$k_1, k_2, k_3$  dan  $k_4$  adalah konstanta laju reaksi dengan satuan yang sesuai. Nilai untuk konstanta tersebut adalah berikut:

$$\begin{aligned}k_1 &= 1,5 \text{ s}^{-1} \\k_2 &= 0,1 \text{ L}^{1/2} / \text{gmol}^{1/2} - \text{s} \\k_3 &= 0,1 \text{ L} / \text{gmol} - \text{s} \\k_4 &= 0,5 \text{ L} / \text{gmol} - \text{s}\end{aligned}$$

$r_i$  memiliki satuan  $\text{gmol/L/s}$ . Ingin dicari konsentrasi masing-masing komponen keluaran reaktor dengan tebakan awal konsentrasi keluaran masing-masing kompoeen adalah 0,5  $\text{gmol/L}$ .

### **Petunjuk Penyelesaian:**

Neraca massa untuk masing-masing empat komponen membentuk seperangkat persamaan nonlinier berikut:

$$\begin{aligned}C_A &= C_{A0} + V(-k_1 C_A - k_2 C_A^{3/2} + k_3 C_C^2) / Q \\C_B &= C_{B0} + V(2k_1 C_A - k_4 C_B^2) / Q \\C_C &= C_{C0} + V(k_2 C_A^{3/2} - k_3 C_C^2 + k_4 C_B^2) / Q \\C_D &= C_{D0} + V(k_4 C_B^2) / Q\end{aligned}$$

## **6.5 RUJUKAN**

- 1) Balzisher, R. E., Samuels, M. R., and Eliassen, J. D., 1972. *Chemical Engineering Thermodynamics*, Englewood Cliffs, N.J.: Prentice Hall.



- 2) Fogler, H. S. 1999. *Elements of Chemical Reaction Engineering*, 3rd ed., Englewood Cliffs, NJ: Prentice-Hall
- 3) Michael B. Cutlip and Moderchai Sacham. 2007. *Recent Advances in Chemical Engineering Problem Solving*. ASEE Mathematical Software.

## PERSAMAAN DIFERENSIAL BIASA

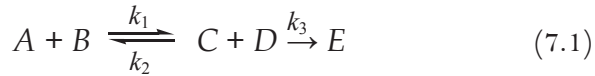
### 7.1 PENDAHULUAN

Bab ini membahas tentang persamaan diferensial biasa. Setelah menyelesaikan pokok bahasan ini mahasiswa diharapkan akan mampu menyelesaikan persamaan diferensial biasa dengan MATLAB. Secara khusus mahasiswa diharapkan akan mampu menyusun permasalahan fisis dan kimia dalam Teknik Kimia dalam bentuk persamaan diferensial biasa, menyelesaikan persamaan diferensial biasa dengan fungsi *ode45* dalam MATLAB. Persamaan diferensial biasa banyak ditemukan pada pemodelan-pemodelan teknik reaktor, kinetika reaksi kimia, peristiwa-peristiwa perpindahan dan lain-lain.

### 7.2 MODEL PERSAMAAN DIFERENSIAL BIASA

Persamaan diferensial muncul dari kajian dinamis proses fisis dan kimia yang memiliki satu variabel bebas. Variabel tersebut dapat berupa variabel jarak,  $x$  atau variabel waktu,  $t$  yang bergantung pada geometri sistem dan kondisi batasnya.

Sebagai contoh suatu reaksi kimia dengan bentuk:



Berlangsung dalam suatu reaktor, neraca massa dapat diterapkan :

$$\begin{aligned} \text{Rate of input} - \text{Rate of output} - \text{Rate of Reaction} \\ = \text{Rate of Accumulation} \end{aligned} \quad (7.2)$$

Untuk reaktor batch, aliran masuk dan keluar adalah nol, sehingga neraca massa disederhanakan menjadi:

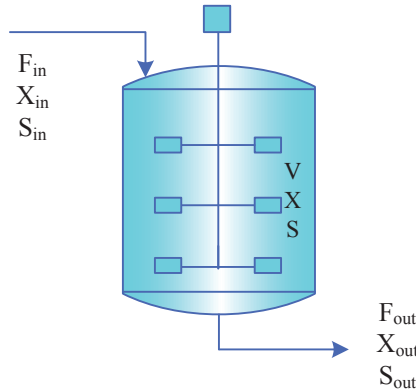
$$\text{Rate of Reaction} = \text{Rate of Accumulation} \quad (7.3)$$

Asumsi bahwa reaksi (7.1) berlangsung dalam fasa cair dengan perubahan volum diabaikan. Jika persamaan (7.3) ditulis untuk setiap komponen yang bereaksi maka akan memiliki bentuk :

$$\begin{aligned} \frac{dC_A}{dt} &= -k_1 C_A C_B + k_2 C_C C_D \\ \frac{dC_B}{dt} &= -k_1 C_A C_B + k_2 C_C C_D \\ \frac{dC_C}{dt} &= k_1 C_A C_B + k_2 C_C C_D - k_3 C_C^m C_D^n \\ \frac{dC_D}{dt} &= k_1 C_A C_B + k_2 C_C C_D - k_3 C_C^m C_D^n \\ \frac{dC_E}{dt} &= k_3 C_C^m C_D^n \end{aligned} \quad (7.4)$$

dimana  $C_A$ ,  $C_B$ ,  $C_C$ ,  $C_D$ , and  $C_E$ , menyatakan konsentrasi lima komponen pada reaksi kimia. Reaksi ini merupakan satu set persamaan diferensial non-linier order satu simultan, yang menggambarkan perilaku dinamis reaksi kimia. Dengan metode yang disusun pada bab ini, persamaan ini dengan satu set kondisi awal, dapat diintegrasikan untuk memperoleh profil waktu semua konsentrasi.

Kasus lain, pada pertumbuhan mikroorganisme, misalnya dalam fermentor kontinu dengan tipe yang ditunjukkan Gambar 7.1 di bawah:



**Gambar 7.1** Fermentor kontinu

Volume cairan dalam fermentor adalah  $V$ , laju alir nutrisi ke dalam fermentor adalah  $F_{in}$ , dan laju air produk keluar fermentor adalah  $F_{out}$ . Neraca massa sel  $X$  adalah:

$$\begin{aligned} \text{Rate of input} - \text{Rate of output} + \text{Rate of Production} \\ = \text{Rate of Accumulation} \end{aligned}$$

$$F_{in}X_{in} - F_{out}X_{out} + r_xV = \frac{d(VX)}{dt} \quad (7.5)$$

Neraca massa untuk substrat  $S$  diberikan dengan persamaan:

$$F_{in}S_{in} - F_{out}S_{out} + r_SV = \frac{d(VS)}{dt} \quad (7.6)$$

Neraca volume total adalah:

$$F_{in} - F_{out} = \frac{d(V)}{dt} \quad (7.7)$$

Jika kita buat asumsi bahwa fermentor tercampur sempurna, yaitu konsentrasi di setiap titik dalam fermentor sama, maka :

$$\begin{aligned} X_{in} &= F_{out} \\ S_{in} &= S_{out} \end{aligned} \quad (7.8)$$

Dan persamaan disederhanakan menjadi :

$$\begin{aligned} \frac{d(VX)}{dt} &= (F_{in}X_{in} - F_{out}X) + r_x V \\ \frac{d(VS)}{dt} &= (F_{in}S_{in} - F_{out}S) + r_s V \\ \frac{dV}{dt} &= F_{in} - F_{out} \end{aligned} \quad (7.9)$$

Asumsi berikutnya dibuat bahwa laju alir masuk dan keluar fermentor adalah sama, dan laju pembentukan sel dan penggunaan substrat diberikan oleh :

$$r_x = \frac{\mu_{max} SX}{K + S} \quad (7.10)$$

$$r_s = -\frac{1}{Y} \frac{\mu_{max} SX}{K + S} \quad (7.11)$$

Persamaan (7.9) disederhanakan menjadi:

$$\frac{dX}{dt} = \left( \frac{F_{out}}{V} \right) (X_{in} - X) + \frac{\mu_{max} SX}{K + S} \quad (7.12)$$

$$\frac{dS}{dt} = \left( \frac{F_{out}}{V} \right) (S_{in} - S) - \frac{1}{Y} \frac{\mu_{max} SX}{K + S} \quad (7.13)$$

Persamaan (7.12) dan (7.13) adalah seperangkat persamaan diferensial biasa simultan, yang menggambarkan dinamik fermentasi kultur kontinyu.

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

$$\text{Orde 1 : } \quad \frac{dy}{dx} + y = kx \quad (7.14)$$

$$\text{Orde 2 : } \quad \frac{d^2y}{dx^2} + y \frac{dy}{dx} = kx \quad (7.15)$$

Orde 3 :

$$\frac{d^3y}{dx^3} + a \frac{d^2y}{dx^2} + b \left( \frac{dy}{dx} \right)^2 = kx \quad (7.16)$$

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

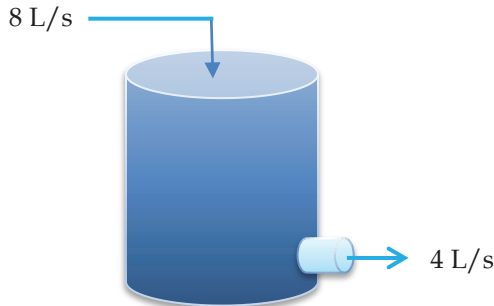
Terdapat beberapa cara berbeda untuk menyusun dan mengeksekusi penyelesaian *ode*, namun untuk kali ini suatu sistem yang menggunakan m-files untuk setiap penyelesaian *ode* akan diberikan. Dua m-files utama yang diperlukan adalah file eksekusi (run) dan file fungsi. Untuk penyelesaian suatu *ode* dalam MATLAB, semua *ode* harus didefinisikan dalam suatu fungsi m-file. Ketika memasukkan ke dalam 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 7.1 Aliran Fluida Pada Tangki

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 sampai interval 50 detik.



Gambar 7.2 Aliran fluida pada tangki

#### Penyelesaian :

Neraca massa:

$$\text{Laju Akumulasi} = \text{input} - \text{Output}$$

$$\frac{d(\rho V)}{dt} = (8 - 4)\rho \quad (7.1.1)$$

Karena densitas konstan, sehingga:

$$\frac{dV}{dt} = (8 - 4) = 4 \quad (7.1.2)$$

dalam liter per detik.

Kondisi awal pada waktu  $t = 0$ , volume dalam tangki = 0.

Berikut penyelesaian persamaan di atas dengan program MATLAB:

File *fluida* untuk fungsi utama:

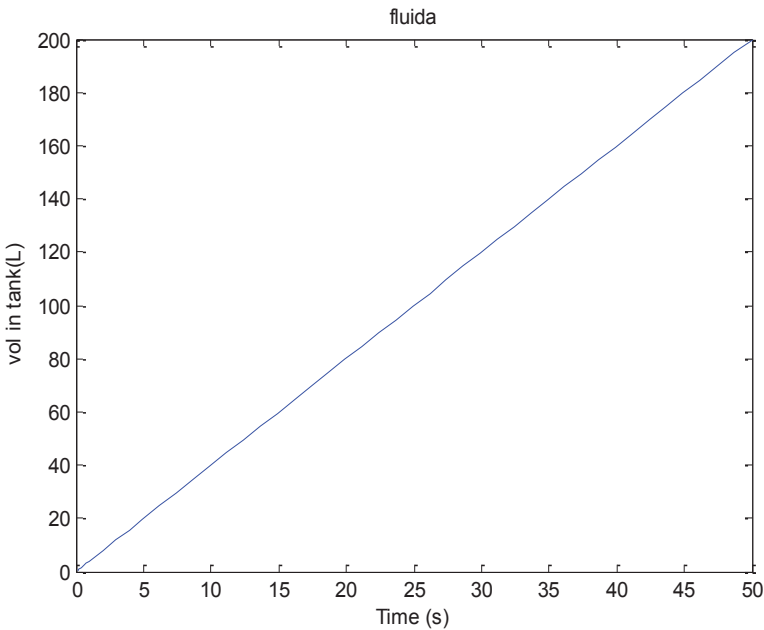
```
function dvdt = fluida(t,v)
dvdt=4;
```

File *runfluida* digunakan untuk mengeksekusi penyelesaian.

List penulisan program adalah

```
function runfluida
to=0;
tf=50;
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 seperti Gambar 7.3.



**Gambar 7.3** Plot waktu terhadap volum

### Contoh 7.2 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 :

$$-\frac{dC_A}{dt} = -r_A \quad (7.2.1)$$

Apabila kecepatan reaksi dapat didefinisikan sebagai :

$$-r_A = \frac{k C_A}{K + C_A} \quad (7.2.2)$$

maka persamaan diferensial di atas menjadi :

$$-\frac{dC_A}{dt} = \frac{k C_A}{K + C_A} \quad (7.2.3)$$

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.

File `run_rbatch1.m` sebagai penyelesaian soal di atas. fungsi utama.

```
function run_rbatch1
clc,clear
tspan=[0:1:10] %interval waktu dari 0 sampai 10
dengan increment 1
%eksekusi persamaan diferensial
[t,Ca]=ode45(@rbatch,tspan,0.5);
[t Ca]% untuk menampilkan t dan Ca dlm baris yg sama
%-----
function dcdt=rbatch(t,Ca)
% menghitung fungsi ruas kanan dari pers.neraca
massa reaktor batch
dcdt=0.1*Ca/(1.03+Ca);
dcdt=dcdt';
```

Hasil run:

```
ans =
      0      0.5000
  1.0000      0.5334
  2.0000      0.5682
```

3.0000	0.6045
4.0000	0.6422
5.0000	0.6813
6.0000	0.7218
7.0000	0.7637
8.0000	0.8070
9.0000	0.8516
10.0000	0.8975

Pada hasil diatas, kolom pertama menunjukkan waktu dan kolom kedua menunjukkan konsentrasi.

Apabila harga  $k$  bergantung pada temperatur menurut persamaan Arrhenius berikut :

$$k = 10^8 e^{\left(\frac{-95.000}{RT}\right)} \quad (7.2.4)$$

dan reaksi yang terjadi bersifat eksotermik ( $\Delta H = 432$  kJ/mol) dan dilaksanakan pada temperatur 500 K, maka persamaan-persamaan neraca massa dan energi sistem adalah sebagai berikut :

$$-\frac{dC_A}{dt} = \frac{10^8 e^{\left(\frac{-95.000}{RT}\right)} C_A}{K + C_A} \quad (7.2.5)$$

$$\rho C_p \frac{dT}{dt} = \frac{10^8 e^{\left(\frac{-95.000}{RT}\right)} C_A (\Delta H)}{K + C_A} \quad (7.2.6)$$

Jika  $\rho$  dan  $C_p$  dianggap sama dengan air (1000 kg/m<sup>3</sup> dan 4.3 kJ/kg), maka program untuk solusi kasus di atas ditunjukkan file run\_rbatch2.m.

```
function run_rbatch2
clc,clear
[t,Y]=ode23(@rbatch_2,[0 50],[0.5 500])
plot(t,Y(:,1),'-+r')
title('konsentrasi dalam reaktor')
xlabel('waktu (min)')
ylabel('Ca (mol/L)')
```

```

pause
plot(t,Y(:,2),'-ob')
title('temperatur dalam reaktor')
xlabel('waktu (min)')
ylabel('T (K)')
%-----
function dcdt=rbatch_2(t,y)
% menghitung fungsi ruas kanan dari pers.neraca
massa reaktor batch
Ca = y(1);
T = y(2);
dcdt(1)=-1E8*exp(-
95000/(8.314*T))*Ca/(1.03+Ca);
dcdt(2)=1E8*exp(-
5000/(8.314*T))*Ca/(1.03+Ca)*(432*1000)/(1000*4.3
);
dcdt=dcdt';

```

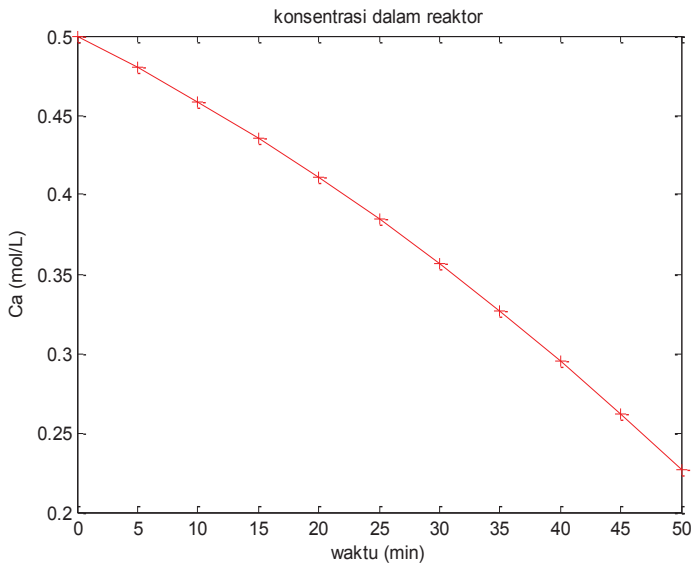
Hasil running:

```

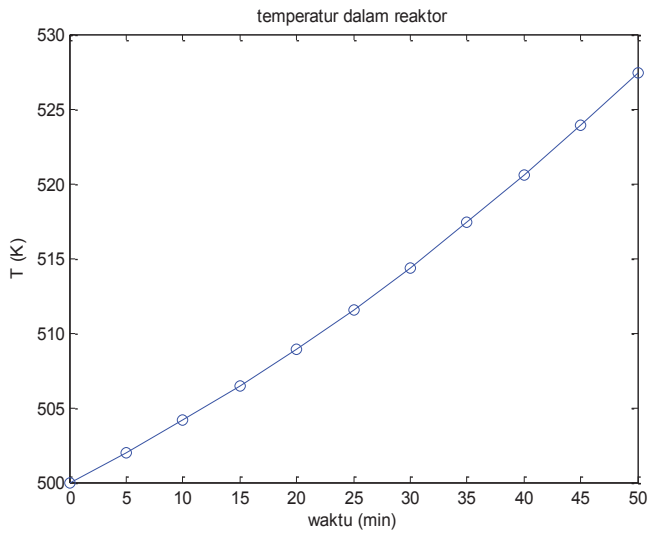
ans =
      0      0.5000    500.0000
    5.0000    0.4799    502.0147
   10.0000    0.4585    504.1645
   15.0000    0.4357    506.4621
   20.0000    0.4112    508.9204
   25.0000    0.3850    511.5520
   30.0000    0.3570    514.3673
   35.0000    0.3271    517.3723
   40.0000    0.2953    520.5644
   45.0000    0.2618    523.9267
   50.0000    0.2271    527.4201

```

Grafik yang dihasilkan:



(a)

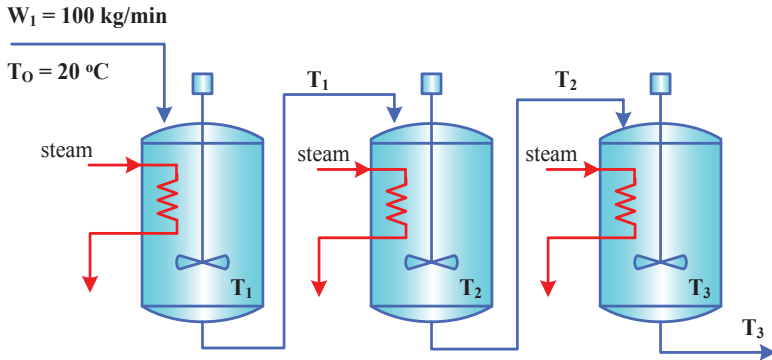


(b)

**Gambar 7.4** Perubahan (a) konsentrasi dan temperatur (b) terhadap waktu

### Contoh 7.3 Pemanasan Pada Tangki Seri

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



**Gambar 7.5** Sistem pemanasan minyak

Pada saat awal, masing-masing tangki diisi dengan 1000 kg minyak pada suhu 20 °C. Steam jenuh pada suhu 250 °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 ketiga dengan laju yang sama. Suhu minyak umpan adalah 20°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,  $C_p = 2.0$  kJ/kg. Laju perpindahan panas dari steam ke minyak tiap tangki dinyatakan dengan persamaan sebagai berikut :

$$Q = Ua(T_{steam} - T) \quad (7.3.1)$$

Dimana  $Ua = 10$  kJ/menit.°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*?

### Penyelesaian :

Asumsi :

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

Disusun neraca panas *unsteady state* untuk masing – masing tangki.

Untuk tangki 1 :

Panas Akumulasi = Panas masuk – Panas keluar

$$MC_p \frac{dT_1}{dt} = WC_p T_0 + Ua(T_{steam} - T_1) - WC_p T_1 \quad (7.3.2)$$

Persamaan di atas dapat disusun kembali sebagai berikut :

$$\frac{dT_1}{dt} = \frac{WC_p(T_0 - T_1) + Ua(T_{steam} - T_1)}{MC_p} \quad (7.3.3)$$

Analog untuk tangki 2 :

$$\frac{dT_2}{dt} = \frac{WC_p(T_1 - T_2) + Ua(T_{steam} - T_2)}{MC_p} \quad (7.3.4)$$

Untuk tangki 3 :

$$\frac{dT_3}{dt} = \frac{WC_p(T_2 - T_3) + Ua(T_{steam} - T_3)}{MC_p} \quad (7.3.5)$$

Penyelesaian untuk kasus diatas dengan MATLAB ditunjukkan file runlatihan7\_3.m berikut:

```
function runlatihan7_3
clear, clc
global W UA M Cp Tsteam To
```

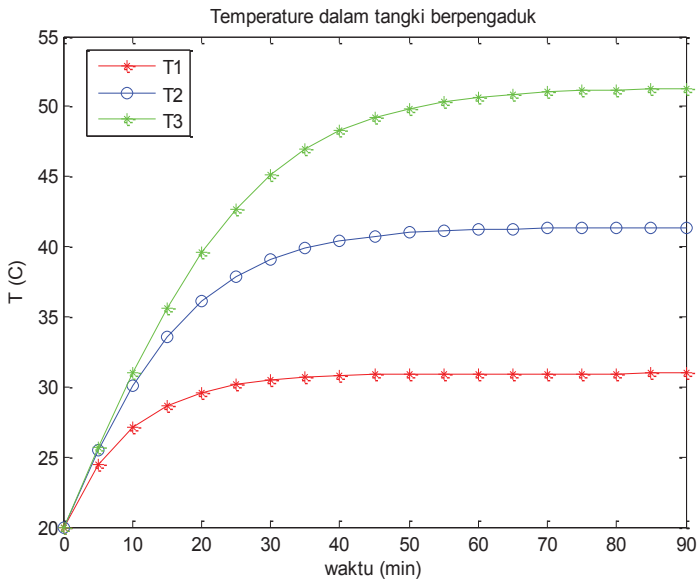
```

%nilai-nilai parameter yang diketahui
W=100; % kg/min
UA=10; % kJ/min.C
M=1000; % kg
Cp=2.0; % kJ/kg
Tsteam=250; % C
To=20; % C
%nilai awal T1, T2, dan T3
Tawal=[20 20 20];%oC
%increment waktu
tmulai=0; % min
takhir=90; % min
tspan=[tmulai:5:takhir];
%fungsi untuk menjalankan fungsi MATLAB
[t,Y]=ode45(@latihan7_3,tspan,Tawal);
%plot grafik t vs T
plot(t,Y(:,1),'*-r',t,Y(:,2),'o-b',t,Y(:,3),'*-g')
title('Temperature dalam tangki berpengaduk')
xlabel('waktu (min)')
ylabel('T (C)')
[t Y]
%-----
function dTdt=latihan7_3(t,Y)
global W UA M Cp Tsteam To
T1=Y(1);
T2=Y(2);
T3=Y(3);
%persamaan-persamaan diferensial yang terlibat
dTdt(1)=( (W*Cp*(To-T1)) +UA*(Tsteam-T1) ) / (M*Cp) ;
dTdt(2)=( (W*Cp*(T1-T2)) +UA*(Tsteam-T2) ) / (M*Cp) ;
dTdt(3)=( (W*Cp*(T2-T3)) +UA*(Tsteam-T3) ) / (M*Cp) ;
dTdt=dTdt';

```

Hasil running:

```
ans =  
      0  20.0000  20.0000  20.0000  
  5.0000  24.4744  25.4917  25.6595  
 10.0000  27.1193  30.0693  30.9567  
 15.0000  28.6863  33.5538  35.6443  
 20.0000  29.6107  36.0841  39.5612  
 25.0000  30.1592  37.8510  42.6961  
 30.0000  30.4827  39.0600  45.1140  
 35.0000  30.6744  39.8701  46.9260  
 40.0000  30.7879  40.4060  48.2515  
 45.0000  30.8550  40.7565  49.2027  
 50.0000  30.8948  40.9837  49.8738  
 55.0000  30.9183  41.1298  50.3411  
 60.0000  30.9322  41.2233  50.6624  
 65.0000  30.9404  41.2826  50.8811  
 70.0000  30.9453  41.3202  51.0286  
 75.0000  30.9482  41.3438  51.1274  
 80.0000  30.9499  41.3587  51.1930  
 85.0000  30.9509  41.3680  51.2363  
 90.0000  30.9515  41.3738  51.2648
```

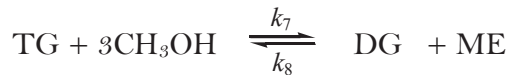
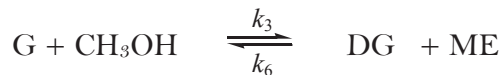
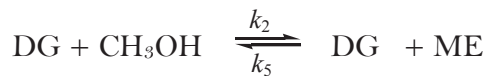
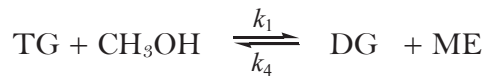


Gambar 7.6 Grafik perubahan suhu terhadap waktu



### Contoh 7.4. Reaksi Transesterifikasi Secara *Batch*

Pada proses pembuatan biodiesel dengan reaksi transesterifikasi, minyak biji bunga matahari direaksikan secara batch dengan metanol dengan rasio molar metanol terhadap minyak adalah 6:1. Reaksi dijalankan pada fasa cair dengan bantuan katalis NaOH. Menurut Klofutar (2010) reaksi didekati dengan reaksi multistap sebagai berikut:



dimana  $TG$  = trigliserida,  $CH_3OH$  = methanol,  $DG$  = digliserida,  $MG$  = monogliserida,  $G$  = gliserol, dan  $ME$  = metil ester.

Ingin dicari konsentrasi setiap komponen setelah reaksi dijalankan selama 120 menit. Nilai konsentrasi mula-mula untuk trigliserida dan methanol adalah 0,4721 dan 0,8 mol/L.

Nilai konstanta laju reaksi pada suhu 50°C adalah :  $k_1 = 0.0727$ ;  $k_2 = 0.1569$ ;  $k_3 = 0,0956$ ;  $k_4 = 0,0213$ ;  $k_5 = 0,0638$ ;  $k_6 = 0,0083$ ;  $k_7 = 0.00026$ ;  $k_8 = 0.0000022$  (L mol<sup>-1</sup> min<sup>-1</sup>).

#### Penyelesaian :

Persamaan neraca massa untuk masing-masing komponen dapat dituliskan:

$$\frac{dC_{TG}}{dt} = -k_1 C_{TG} C_{MOH} + k_4 C_{DG} C_{ME} - k_7 C_{TG} C_{MOH}^3 + k_8 C_G C_{ME}^3$$

$$\begin{aligned}\frac{dC_{DG}}{dt} &= k_1 C_{TG} C_{MOH} - k_2 C_{DG} C_{MOH} - k_4 C_{DG} C_{ME} + k_5 C_{MG} C_{ME} \\ \frac{dC_{MG}}{dt} &= k_2 C_{DG} C_{MOH} - k_3 C_{MG} C_{MOH} - k_5 C_{MG} C_{ME} + k_6 C_G C_{ME} \quad (7.4.1) \\ \frac{dC_G}{dt} &= k_3 C_{MG} C_{MOH} - k_6 C_G C_{ME} + k_7 C_{TG} C_{MOH}^3 - k_8 C_G C_{ME}^3 \\ \frac{dC_{MOH}}{dt} &= -\frac{dC_{ME}}{dt}\end{aligned}$$

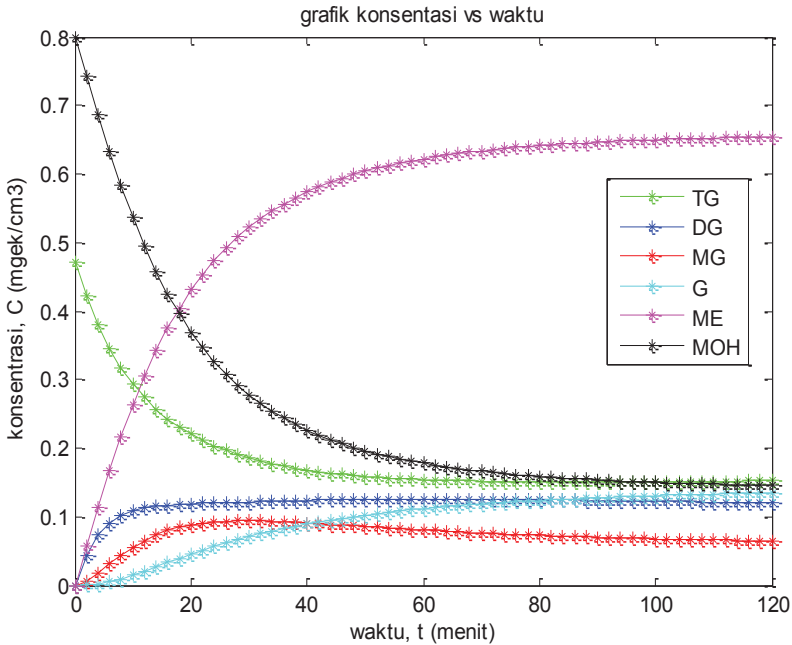
Dengan  $C_{TG}$ ,  $C_{DG}$ ,  $C_{MG}$ ,  $C_G$ ,  $C_{ME}$ ,  $C_{MOH}$  masing-masing adalah konsentrasi trigliserida, digliserida, monogliserida, gliserol, metil ester dan methanol.

Berikut program MATLAB untuk meyelesaian contoh kasus di atas.

```
function simulasicontoh7_4;
clc,clear
% menyelesaikan PD ordiner simultan
% nilai awal
Yo=[0.4721 0 0 0 0 0.8];
% interval waktu
tspan=[0:2:120]; % menit
[t,Y]=ode45(@sub_simulasi4,tspan,Yo);
CTG =Y(:,1);
CDG =Y(:,2);
CMG =Y(:,3);
CG =Y(:,4);
CME =Y(:,5);
CMOH =Y(:,6);
plot(tspan,CTG,'-*g',tspan,CDG,'-*b',tspan,CMG,'-*r',...
      tspan,CG,'-*c',tspan,CME,'-*m',tspan,CMOH,'-*k');
title('grafik konsentasi vs waktu');
xlabel('waktu, t (menit)');
ylabel('konsentrasi, C (mg/ml)');
legend('TG','DG','MG','G','ME','MOH')
%-----
function dydt=sub_simulasi4(t,Y);
k1=0.0727;k2=0.1569;k3=0.0956;k4=0.0213;k5=0.0638;k6=0.0083;
k7=0.00026;k8=0.0000022;
CTG=Y(1);CDG=Y(2);CMG=Y(3);CG=Y(4);CME=Y(5);CMOH=Y(6);
```

$$\begin{aligned} \text{dydt (1)} &= -k_1 \cdot \text{CTG} \cdot \text{CMOH} + k_4 \cdot \text{CDG} \cdot \text{CME} - k_7 \cdot \text{CTG} \cdot \text{CMOH}^3 + k_8 \cdot \text{CG} \cdot \text{CME}^3; \\ \text{dydt (2)} &= k_1 \cdot \text{CTG} \cdot \text{CMOH} - k_2 \cdot \text{CDG} \cdot \text{CMOH} - k_4 \cdot \text{CDG} \cdot \text{CME} + k_5 \cdot \text{CMG} \cdot \text{CME}; \\ \text{dydt (3)} &= k_2 \cdot \text{CDG} \cdot \text{CMOH} - k_3 \cdot \text{CMG} \cdot \text{CMOH} - k_5 \cdot \text{CMG} \cdot \text{CME} + k_6 \cdot \text{CG} \cdot \text{CME}; \\ \text{dydt (4)} &= k_3 \cdot \text{CMG} \cdot \text{CMOH} - k_6 \cdot \text{CG} \cdot \text{CME} + k_7 \cdot \text{CTG} \cdot \text{CMOH}^3 - k_8 \cdot \text{CG} \cdot \text{CME}^3; \\ \text{dydt (5)} &= k_1 \cdot \text{CTG} \cdot \text{CMOH} + k_2 \cdot \text{CDG} \cdot \text{CMOH} + k_3 \cdot \text{CMG} \cdot \text{CMOH} - k_4 \cdot \text{CDG} \cdot \text{CME} - \\ & k_5 \cdot \text{CMG} \cdot \text{CME} - k_6 \cdot \text{CG} \cdot \text{CME} + 3 \cdot k_7 \cdot \text{CTG} \cdot \text{CMOH}^3 - 3 \cdot k_8 \cdot \text{CG} \cdot \text{CME}^3; \\ \text{dydt (6)} &= -\text{dydt (5)}; \\ \text{dydt} &= \text{dydt '}; \end{aligned}$$

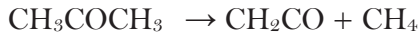
Konsentrasi setiap komponen ditampilkan dalam bentuk grafik seperti terlihat pada gambar di bawah.



**Gambar 7.7** Grafik perubahan suhu terhadap waktu

### Contoh 7.5. Produksi Asetat Anhidrid dengan PFR (Multitubular)

Dalam perancangan reaktor untuk produksi asam asetat anhidrid salah satu kunci yang menentukan adalah tahap perengkangan endotermis fasa uap aseton menjadi ketene dan metana (Fogler, 1999):



Reaksi ini adalah order satu terhadap aseton dan laju reaksi spesifik dinyatakan dengan persamaan:

$$k = 8,2 \times 10^{14} \exp\left[-\frac{34.222}{T}\right] \quad (7.5.1)$$

Dimana  $k$  memiliki satuan detik dan  $T$  dengan satuan kelvin. Dalam desain ini diperlukan umpan sebanyak 7.850 kg eseton per jam ke dalam reaktor tubular. Reaktor terdiri atas susunan tube sebanyak 1000 dengan ukuran 1 inch schedule 40.

Susunan 1000 tube 1-inch schedule 40 dengan panjang 1,79 m setara dengan 1 m<sup>3</sup> (0,001 m<sup>3</sup>/tube = 1 dm<sup>3</sup>/tube) dan memberikan koversi 20%. Ketene tidak stabil dan cenderung meledak, sehingga kondisi tersebut menjadi alasan untuk menjaga kondisi reaktor pada konversi rendah. Namun demikian, material pipa dan ukuran schedule harus diperiksa untuk mempelajari bahwa material tersebut sesuai pada temperature dan tekanan operasi.

Fluida penukar panas memiliki laju alir,  $m$ , 0,111 mol/s, dengan kapasitas panas 34,5 J/mol.K.

Kita harus mempertimbangkan empat kasus.

Temperature dan tekanan masuk adalah sama untuk semua kasus pada 1.035 K dan 162 kPa (1,6 atm) dan temperatur fluida pemanas tersedia pada 1.250 K.

- a) Kasus 1 : Reaktor dioperasikan secara adiabatik
- b) Kasus 2 : Temperatur fluida pemanas konstan,  $T_c = 1.250$  K
- c) Kasus 3 : Fluida pemanas searah (*co-current*) dengan reaktan dengan,  $T_{co} = 1.250$
- d) Kasus 4 : Fluida pemanas berlawanan arah (*counter-current*) dengan reaktan  $T_{co} = 1.250$

Informasi tambahan :

$$\text{CH}_3\text{COCH}_3 \text{ (A)} : H_A^o(T_R) = -216,67 \text{ kJ/mol}$$

$$\text{CH}_2\text{CO (B)} : H_B^o(T_R) = -61,09 \text{ kJ/mol}$$

$$\begin{aligned} \text{CH}_4 \text{ (C)} & : H_C^\circ(T_R) = -74,81 \text{ kJ/mol} \\ U & = 110 \text{ J/s.m}^2\text{.K} \end{aligned}$$

**Penyelesaian :**

Misalkan A = CH<sub>3</sub>COCH<sub>3</sub>, B = CH<sub>2</sub>CO dan C = CH<sub>4</sub>, Persamaan rekasi dapat dituliskan :



Urutan penyelesaian untuk PFR dengan Heat effect adalah :

**1. Neraca mol steady state untuk reaktor PFR :**

$$\frac{dX}{dV} = -\frac{r_A}{F_{A0}} \quad (7.5.2)$$

**2. Persamaan Laju reaksi :**

$$-r_A = k C_A \quad (7.5.3)$$

**3. Stoikiometri (Reaksi fasa gas tanpa hilang tekanan):**

$$C_A = \frac{C_{A0}(1-X)T_0}{(1+\varepsilon X)T} \quad (7.5.4)$$

$$\varepsilon = y_{A0}\delta = 1(1+1-1) = 1$$

Kombinasi persamaan 7.5.3 dan 7.5.4, dihasilkan :

$$-r_A = k \frac{C_{A0}(1-X)T_0}{(1+X)T} \quad (7.5.5)$$

**4. Neraca Energi :**

a) Neraca Reaktor

Perubahan temperatur reaktor sebagai fungsi volume reaktor:

$$\frac{dT}{dV} = \frac{Ua(T_a - T) + (-r_A)(-\Delta H_{rxn})}{F_{A0} \left( \sum \Theta_i C_{pi} + X \Delta C_p \right)} \quad (7.5.6)$$

atau,

$$\frac{dT}{dV} = \frac{Ua(T_a - T) + (-r_A)(-\Delta H_{rxn})}{F_A C_{pA} + F_B C_{pB} + F_C C_{pC}} \quad (7.5.7)$$

#### b) Heat Exchanger.

Kita akan menggunakan fluida pemanas untuk aliran *co-current* sebagai basis kasus. Selanjutnya kita dapat memodifikasi program penyelesaian ODE untuk menyelesaikan kasus lain yang berkesesuaian dengan mengalikan nol atau minus.

Untuk aliran *co-current*:

$$\frac{dT_a}{dV} = \frac{Ua(T_a - T)}{m_c C_{pc}} \quad (7.5.8)$$

### 5. Parameter untuk Neraca Energi:

Panas reaksi pada suhu tertentu terhadap suhu referensi (298 K) adalah :

$$\Delta H_{rxn} = \Delta H_{rxn}^0 + \Delta \alpha (T - T_R) + \frac{\Delta \beta}{2} (T^2 - T_R^2) + \frac{\Delta \gamma}{3} (T^3 - T_R^3) \quad (7.5.9)$$

Kapasitas Panas komponen:

$$C_{pA} = 26,63 + 0,183 T - 45,86 \times 10^{-6} T^2 \text{ (J/mol K)}$$

$$C_{pB} = 20,04 + 0,0945 T - 30,95 \times 10^{-6} T^2 \text{ (J/mol K)}$$

$$C_{pC} = 13,39 + 0,077 T - 18,71 \times 10^{-6} T^2 \text{ (J/mol K)}$$

$$\Delta \alpha = \alpha_C + \alpha_B - \alpha_A = 13,9 + 20,04 - 26,63 = 6,8 \text{ J/mol.K}$$

$$\Delta \beta = \beta_C + \beta_B - \beta_A = 0,007 + 0,0945 - 0,183 = -0,0115 \text{ J/mol.K}^2$$

$$\begin{aligned} \Delta \gamma &= \gamma_C + \gamma_B - \gamma_A = (-18,71 \times 10^{-6}) + (-30,95 \times 10^{-6}) - (-45,86 \times 10^{-6}) \\ &= -3,8 \times 10^{-6} \text{ J/mol.K}^3 \end{aligned}$$

$$\frac{\Delta\beta}{2} = 5,75 \times 10^{-3}$$

$$\frac{\Delta\gamma}{3} = -1,27 \times 10^{-6}$$

Panas reaksi standar,  $\Delta H_{rxn}^0$  :

$$\begin{aligned}\Delta H_{rxn}^o(T_R) &= H_B^o(T_R) + H_C^o(T_R) - H_A^o(T_R) \\ &= (-61,09) + (-74,81) - (-216,67) \text{ kJ/kmol} \\ &= 80,77 \text{ kJ/mol}\end{aligned}$$

Heat Exchange. Luas area transfer panas per satuan volume pipa adalah :

$$a = \frac{\pi DL}{(\pi D^2 / 4)L} = \frac{4}{D} = \frac{4}{0,0266 \text{ m}} = 150 \text{ m}^{-1}$$

$$U = 110 \text{ J/m}^2 \cdot \text{s} \cdot \text{K}$$

Kombinasi koefisien transfer panas dengan area diperoleh:

$$Ua = 16.500 \text{ J/m}^3 \cdot \text{s} \cdot \text{K}$$

## 6. Perhitungan Parameter Neraca Mol pada Basis Per Tube:

$$F_{A0} = \frac{7.850 \text{ kg/jam}}{58 \text{ kg/kmol}} \times \frac{1}{1000 \text{ tube}} = 0,135 \text{ kmol/jam} = 0,0376 \text{ mol/detik}$$

$$C_{A0} = \frac{P_{A0}}{RT} = \frac{162 \text{ kPa}}{8,31 \frac{\text{kPa} \cdot \text{m}^3}{\text{kmol} \cdot \text{K}} (1.035 \text{ K})} = 0,0188 \frac{\text{kmol}}{\text{m}^3} = 18,8 \frac{\text{kmol}}{\text{m}^3}$$

$$v_0 = \frac{F_{A0}}{C_{A0}} = 2 \text{ dm}^3/\text{detik}, \quad V = \frac{1 \text{ m}^3}{1000 \text{ tube}} = \frac{0,001 \text{ m}^3}{\text{tube}}$$

## Kasus 1: Adiabatis

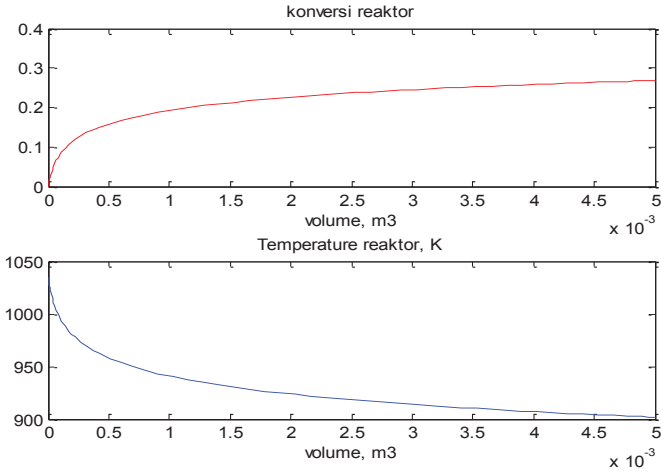
Kita mulai dengan kasus 1 adiabatis untuk menunjukkan pengaruh bagaimana reaksi hilang/terhenti pada saat temperatur drop. Pada kenyataannya kita akan menambah sampai volume 5 dm<sup>3</sup> untuk mengamati efek ini dan menunjukkan bagaimana kebutuhan penambahan suatu penukar panas. Untuk ilustrasi bagaimana reaksi endotermis secara virtual dapat terhenti secara total. Volume satu pipa ditambah dari 1 dm<sup>3</sup> sampai 5 dm<sup>3</sup>. Untuk adiabatis :

$$Ua = 16.500 \times 0$$

Program Matlab ditunjukkan di bawah dan Gambar 7.8 menunjukkan grafik output.

```
1 function pfrcase1
2 -   clc, clear
3 -   vspan=[0 0.005];
4 -   Y0=[0 1035];
5 -   [v,Y]=ode45(@reactorcase1,vspan,Y0);
6 -   subplot(2,1,1)
7 -   plot(v,Y(:,1),'-r')
8 -   title('konversi reaktor')
9 -   xlabel('volume, m3')
10 -  subplot(2,1,2)
11 -  plot(v,Y(:,2),'-b')
12 -  title('Temperature reaktor, K')
13 -  xlabel('volume, m3')
14 -  %-----
15 function dYdv=reactorcase1(v,Y)
16 -   X=Y(1);
17 -   T=Y(2);
18 -   CA0 = 18.85;% mol/m3
19 -   Da = 6.8;%delta a
20 -   Db = -(5.75E-3)*2;%delta b
21 -   Dc = -(1.27E-6)*3;%delta c
22 -   FA0 = 0.03754;%mol/s
23 -   T0 = 1035;%K
24 -   Tr = 298;%K
25 -   FA = FA0*(1-X);
26 -   FB = FA0*X;
27 -   FC = FA0*X;
28 -   DHrxn0 = 80770;%J/mol
29 -   CpA = 26.63 + 0.183*T-45.86E-6*T^2;
30 -   CpB = 20.04 + 0.0945*T-30.95E-6*T^2;
31 -   CpC = 13.39 + 0.077*T-18.71E-6*T^2;
32 -   k = 8.2E14*exp(-34222/T);
33 -   ra = -k*CA0*(1-X)*(T0/T)/(1 + X);
34 -   DHrxn = DHrxn0 + Da*(T-Tr) + (Db/2)*(T^2-Tr^2) + (Dc/3)*(T^3-Tr^3);
35 -   dYdv(1) = -ra/FA0;
36 -   dYdv(2) = -ra*(-DHrxn)/(FA*CpA + FB*CpB + FC*CpC);
37 -   dYdv=dYdv';
```





**Gambar 7.8** Grafik perubahan terhadap volume  
(a) reaktor konversi (b) temperatur

### Kasus 2: Temperatur fluida pemanas konstan, $T_a$

Program penyelesaian kasus 2 sama seperti kasus 1, dengan merubah :

$$\frac{dT_a}{dV} = \frac{Ua(T_a - T)}{m_c C_{pc}} \times 0$$

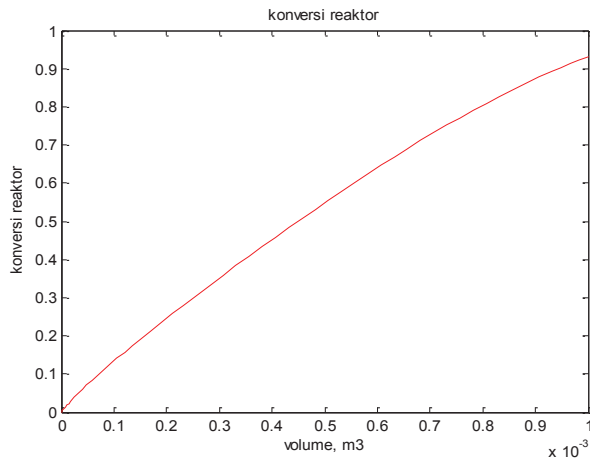
$$Ua = 16.500 \text{ J/m}^3/\text{s/K}$$

Program Matlab ditunjukkan di bawah dan Gambar 7.9 menunjukan grafik output.

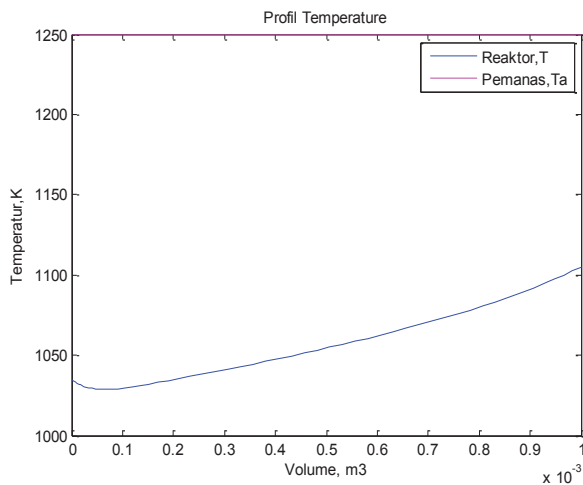
```

1 function pfrcase2
2 -   clc, clear
3 -   vspan=[0 0.001];
4 -   Y0=[0 1035 1250];
5 -   [v,Y]=ode45(@reactorcase2,vspan,Y0);
6 -   plot(v,Y(:,1),'-r')
7 -   title('konversi reaktor')
8 -   xlabel('volume, m3')
9 -   ylabel('konversi reaktor')
10 -  pause
11 -  plot(v,Y(:,2),'-b',v,Y(:,3),'-m')
12 -  title('Profil Temperature')
13 -  xlabel('Volume, m3')
14 -  ylabel('Temperatur,K ')
15 -  legend('Reaktor,T','Pemanas,Ta')
16 -  %-----
17 function dYdv = reactorcase2(v,Y)
18 -   X=Y(1);
19 -   T=Y(2);
20 -   Ta=Y(3);
21 -   CA0 = 18.85;% mol/m3
22 -   Da = 6.8;%delta a
23 -   Db = -(5.75E-3)*2;%delta b
24 -   Dc = -(1.27E-6)*3;%delta c
25 -   FA0 = 0.03754;%mol/s
26 -   T0 = 1035;%K
27 -   Tr = 298;%K
28 -   FA = FA0*(1-X);
29 -   FB = FA0*X;
30 -   FC = FA0*X;
31 -   Ua = 1.65E+4;
32 -   mc = 0.111;
33 -   Cpc = 34.5;
34 -   DHrxn0 = 80770;%J/mol
35 -   CpA = 26.63 + 0.183*T-45.86E-6*T^2;
36 -   CpB = 20.04 + 0.0945*T-30.95E-6*T^2;
37 -   CpC = 13.39 + 0.077*T-18.71E-6*T^2;
38 -   k = 8.2E14*exp(-34222/T);
39 -   ra =-k*CA0*(1-X)*(T0/T)/(1 + X);
40 -   DHrxn = DHrxn0 + Da*(T-Tr) + (Db/2)*(T^2-Tr^2) + (Dc/3)*(T^3-Tr^3);
41 -   dYdv(1)= -ra/FA0;
42 -   dYdv(2)= (Ua*(Ta-T)-ra*(-DHrxn))/(FA*CpA + FB*CpB + FC*CpC);
43 -   dYdv(3)= Ua*(Ta-T)/(mc*Cpc)*0;
44 -   dYdv=dYdv';

```



(a)



(b)

**Gambar 7.9** Grafik perubahan terhadap volume reaktor  
(a) konversi dan (b) temperatur

**Kasus 3 : Fluida pemanas searah (*co-current*) dengan reaktan**

Neraca energi untuk fluida pemanas adalah :

$$\frac{dT_a}{dV} = \frac{Ua(T_a - T)}{m_c C_{pc}}$$

dengan  $T_{a0} = 1.250$  K pada  $V = 0$ .

Program Matlab ditunjukkan di bawah dan Gambar 7.10 menunjukkan grafik output.

```

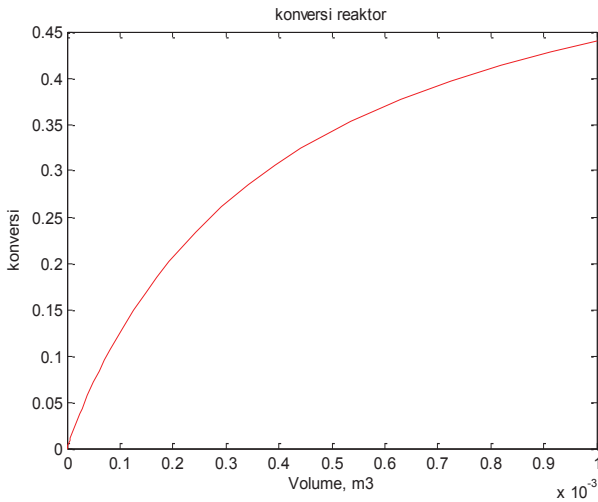
1  function pfrcase3
2  -   clc, clear
3  -   vspan=[0 0.001];
4  -   Y0=[0 1035 1250];
5  -   [v,Y]=ode15s(@reactorcase3,vspan,Y0);
6  -   %subplot(3,1,1);
7  -   plot(v,Y(:,1),'-r')
8  -   title('konversi reaktor')
9  -   xlabel('Volume, m3')
10 -   ylabel('konversi')
11 -   pause
12 -   plot(v,Y(:,2),'-b',v,Y(:,3),'-m')
13 -   title(' Profil Temperature')
14 -   xlabel('Volume, m3')
15 -   ylabel('Temperatur,K ')
16 -   legend('Reaktor, T','Pemanas, Ta')
17 -   %-----
18  function dYdv = reactorcase3(v,Y)
19 -   X=Y(1);
20 -   T=Y(2);
21 -   Ta=Y(3);
22 -   CA0 = 18.85;% mol/m3
23 -   Da = 6.8;%delta a
24 -   Db = -(5.75E-3)*2;%delta b
25 -   Dc = -(1.27E-6)*3;%delta c
26 -   FA0 = 0.03754;%mol/s
27 -   T0 = 1035;%K
28 -   Tr = 298;%K
29 -   FA = FA0*(1-X);
30 -   FB = FA0*X;
31 -   FC = FA0*X;
32 -   Ua = 1.65E+4;
33 -   mc = 0.111;
34 -   Cpc = 34.5;
35 -   DHrxn0 = 80770;%J/mol
36 -   CpA = 26.63 + 0.183*T-45.86E-6*T^2;
37 -   CpB = 20.04 + 0.0945*T-30.95E-6*T^2;

```

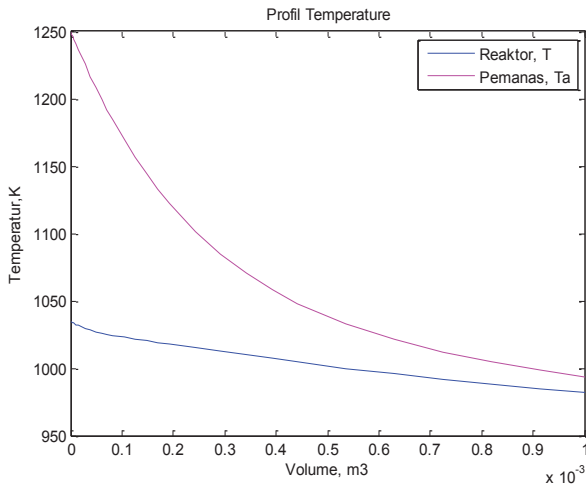
```

38 - CpC = 13.39 + 0.077*T-18.71E-6*T^2;
39 - k = 8.2E14*exp(-34222/T);
40 - ra = -k*CA0*(1-X)*(T0/T)/(1 + X);
41 - DHrxn = DHrxn0 + Da*(T-Tr) + (Db/2)*(T^2-Tr^2) + (Dc/3)*(T^3-Tr^3);
42 - dYdv(1) = -ra/FA0;
43 - dYdv(2) = (Ua*(Ta-T)+ra*(DHrxn))/(FA*CpA + FB*CpB + FC*CpC);
44 - dYdv(3) = Ua*(T-Ta)/(mc*Cpc);
45 - dYdv=dYdv';

```



(a)



(b)

**Gambar 7.10** Grafik perubahan terhadap volume  
(a) konversi dan (b) temperatur

## Kasus 4: Fluida pemanas searah dengan reaktan

Neraca energi untuk fluida pemanas sama dengan kasus 3, dengan mengalikannya dengan -1 :

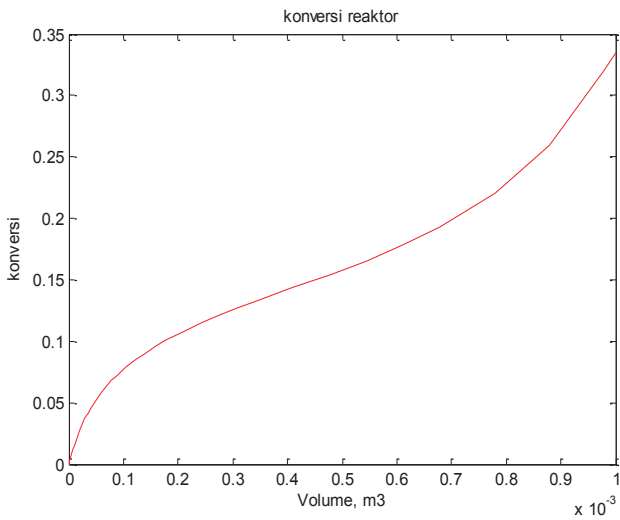
$$\frac{dT_a}{dV} = -\frac{Ua(T_a - T)}{m_c C_{pc}}$$

dengan  $T_a(V=0) = 994,6$  K untuk memperoleh  $T_{a0} = 1.250$  K.

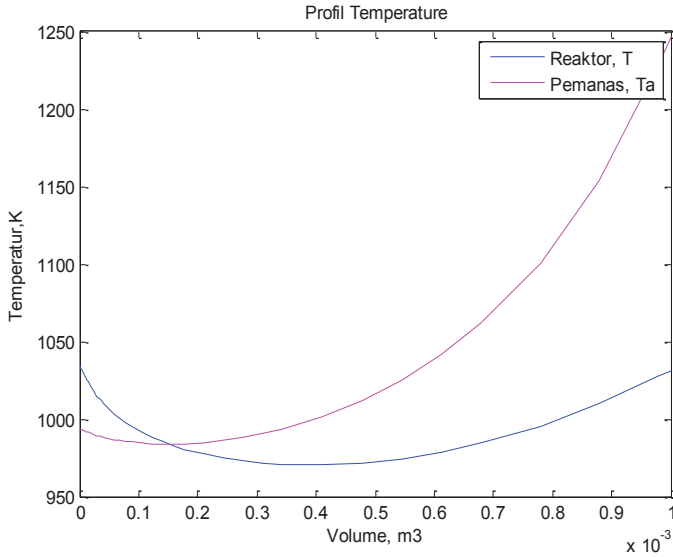
Program Matlab ditunjukkan di bawah dan Gambar 7.11 menunjukkan grafik output.

```
1 function pfrcase4
2 -   clc, clear
3 -   vspan=[0 0.001];
4 -   Y0=[0 1035 993.6];
5 -   [v,Y]=ode15s(@reactorcase4,vspan,Y0);
6 -   %subplot(3,1,1);
7 -   plot(v,Y(:,1),'-r')
8 -   title('konversi reaktor')
9 -   xlabel('Volume, m3')
10 -  ylabel('konversi')
11 -  pause
12 -  plot(v,Y(:,2),'-b',v,Y(:,3),'-m')
13 -  title(' Profil Temperature')
14 -  xlabel('Volume, m3')
15 -  ylabel('Temperatur,K ')
16 -  legend('Reaktor, T','Pemanas, Ta')
17 -  %-----
18 function dYdv = reactorcase4(v,Y)
19 -   X=Y(1);
20 -   T=Y(2);
21 -   Ta=Y(3);
22 -   CA0 = 18.85;% mol/m3
23 -   Da = 6.8;%delta a
24 -   Db = -(5.75E-3)*2;%delta b
25 -   Dc = -(1.27E-6)*3;%delta c
26 -   FA0 = 0.03754;%mol/s
27 -   T0 = 1035;%K
28 -   Tr = 298;%K
29 -   FA = FA0*(1-X);
30 -   FB = FA0*X;
31 -   FC = FA0*X;
32 -   Ua = 1.65E+4;
33 -   mc = 0.111;
34 -   Cpc = 34.5;
35 -   DHrxn0 = 80770;%J/mol
36 -   CpA = 26.63 + 0.183*T-45.86E-6*T^2;
37 -   CpB = 20.04 + 0.0945*T-30.95E-6*T^2;
38 -   CpC = 13.39 + 0.077*T-18.71E-6*T^2;
39 -   k = 8.2E14*exp(-34222/T);
40 -   ra = -k*CA0*(1-X)*(T0/T)/(1 + X);
41 -   DHrxn = DHrxn0 + Da*(T-Tr) + (Db/2)*(T^2-Tr^2) + (Dc/3)*(T^3-Tr^3);
42 -   dYdv(1)= -ra/FA0;
```

$$\begin{aligned}
 43 - & \quad dYdv(2) = (Ua*(Ta-T)+ra*(DHrxn)) / (FA*CpA + FB*CpB + FC*CpC) ; \\
 44 - & \quad dYdv(3) = -Ua*(T-Ta) / (mc*Cpc) ; \\
 45 - & \quad dYdv=dYdv' ;
 \end{aligned}$$



(a)

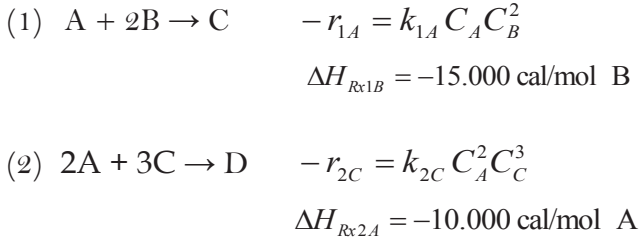


(b)

**Gambar 7.11** Grafik perubahan terhadap volume  
(a) konversi dan (b) temperatur

## Contoh 7.6 Reaksi Kompleks dengan Efek Panas dalam PFR

Berikut reaksi kompleks fasa gas mengikuti hukum laju reaksi elementer (Fogler, 1999):



Reaksi berlangsung pada reaktor PFR. Umpan adalah stoikiometri pada reaksi (1) terhadap A dan B dengan  $F_{A0} = 5$  mol/menit. Volume reaktor adalah  $10 \text{ dm}^3$  dan konsentrasi total yang masuk adalah  $C_{T0} = 0,2 \text{ mol/dm}^3$ . Tekanan masuk adalah 100 atm dan temperatur masuk adalah 300 K. Laju alir pendingin adalah 50 mol/menit dan memiliki kapasitas panas  $C_{pCO} = 10 \text{ cal/mol.K}$  serta masuk pada suhu 325 K.

$$\begin{aligned} k_{1A} &= 40 \left( \frac{\text{dm}^3}{\text{mol}} \right)^2 / \text{menit pada suhu 300 K dengan } E_1 = 8.000 \text{ cal/mol} \\ k_{2C} &= 2 \left( \frac{\text{dm}^3}{\text{mol}} \right)^4 / \text{menit pada suhu 300 K dengan } E_1 = 12.000 \text{ cal/mol} \\ C_{pA} &= 10 \text{ cal/mol/K} & Ua &= 80 \text{ cal/min/K} \\ C_{pB} &= 12 \text{ cal/mol/K} & T_{a0} &= 325 \text{ K} \\ C_{pC} &= 12 \text{ cal/mol/K} & m &= 50 \text{ mol/min} \\ C_{pD} &= 12 \text{ cal/mol/K} \end{aligned}$$

Plot  $F_A$ ,  $F_B$ ,  $F_C$ ,  $F_D$ ,  $y$ ,  $T$  dan  $T_a$  sebagai fungsi  $V$  untuk :

(a) Pertukaran panas searah (*co-current*)



- (b) Pertukaran panas lawan arah (*counter current*)
- (c)  $T_a$  konstan
- (d) Operasi Adiabatik

**Penyelesaian :**

Reaksi Fasa Gas pada PFR tanpa hilang tekan ( $y = 1$ )

**Neraca mol :**

$$\frac{dF_A}{dV} = r_A \quad (F_{A0} = 5 \text{ mol/menit}) \quad (7.6.1)$$

$$\frac{dF_B}{dV} = r_B \quad (F_{B0} = 10 \text{ mol/menit}) \quad (7.6.2)$$

$$\frac{dF_C}{dV} = r_C \quad V_i = 10 \text{ dm}^3 \quad (7.6.3)$$

$$\frac{dF_D}{dV} = r_D \quad (7.6.4)$$

**Persamaan laju :**

$$r_{1A} = -k_{1A} C_A C_B^2 \quad (7.6.5)$$

$$r_{2C} = -k_{2C} C_A^2 C_C^3 \quad (7.6.6)$$

**Laju relatif :**

$$r_{1B} = 2 r_{1A} \quad (7.6.7)$$

$$r_{1C} = -r_{1A} \quad (7.6.8)$$

$$r_{2A} = \frac{2}{3} r_{2C} = -\frac{2}{3} k_{2C} C_A^2 C_C^3 \quad (7.6.9)$$

$$r_{2D} = -\frac{1}{3} r_{2C} = \frac{1}{3} k_{2C} C_A^2 C_C^3 \quad (7.6.10)$$

**Laju total** untuk komponen  $A$ ,  $B$ ,  $C$  dan  $D$  adalah :

$$r_A = r_{1A} + r_{2A} = -k_{1A} C_A C_B^2 - \frac{2}{3} k_{2C} C_A^2 C_C^3 \quad (7.6.11)$$

$$r_B = r_{1B} = -2k_{1A} C_A C_B^2 \quad (7.6.12)$$

$$r_C = r_{1C} + r_{2C} = k_{1A} C_A C_B^2 - k_{2C} C_A^2 C_C^3 \quad (7.6.13)$$

$$r_D = r_{2D} = \frac{1}{3} k_{2C} C_A^2 C_C^3 \quad (7.6.14)$$

**Stoikiometri :**

$$C_A = C_{T0} \left( \frac{F_A}{F_T} \right) y \quad (7.6.15)$$

$$C_B = C_{T0} \left( \frac{F_B}{F_T} \right) y \quad (7.6.16)$$

$$C_C = C_{T0} \left( \frac{F_C}{F_T} \right) y \quad (7.6.17)$$

$$C_D = C_{T0} \left( \frac{F_D}{F_T} \right) y \quad (7.6.18)$$

$$y = 1 \quad (7.6.19)$$

$$F_T = F_A + F_B + F_C + F_D \quad (7.6.20)$$

**Parameter :**

$$k_{1A} = 40 \exp \left[ \frac{E_1}{R} \left( \frac{1}{300} - \frac{1}{T} \right) \right] (\text{dm}^3/\text{mol})^2 / \text{min} \quad (7.6.21)$$

$$k_{2C} = 2 \exp \left[ \frac{E_2}{R} \left( \frac{1}{300} - \frac{1}{T} \right) \right] (\text{dm}^3/\text{mol})^4 / \text{min} \quad (7.6.22)$$

$$R = 1,987 \text{ cal/mol/K}$$

**Neraca Energi :**

Perubahan suhu reaktor terhadap volume :

$$\frac{dT}{dV} = \frac{Q_g - Q_r}{\sum F_j C_{p_j}} \quad (7.6.23)$$

$$\sum F_j C_{p_j} = F_A C_{p_A} + F_B C_{p_B} + F_C C_{p_C} + F_D C_{p_D} \quad (7.6.24)$$

Panas yang diserap (*Heat Romoved*) :

$$Q_r = Ua(T - T_a) \quad (7.6.25)$$

Panas yang dibangkitkan (*Heat Generated*):

$$Q_r = \sum r_{ij} \Delta H_{R_{xij}} = r_{1B} \Delta H_{R_{x1B}} + r_{2A} \Delta H_{R_{x2A}} \quad (7.6.26)$$

(a) **Fluida pndingin searah (*co-current*) dengan reaktan**

Neraca pertukaran panas untuk aliran co-current adalah :

$$\frac{dT_a}{dV} = \frac{Ua(T - T_a)}{m_c C_{p_{cO}}} \quad (7.6.27)$$

Program Matlab ditunjukkan di bawah dan Gambar 7.12 menunjukan grafik output.

```

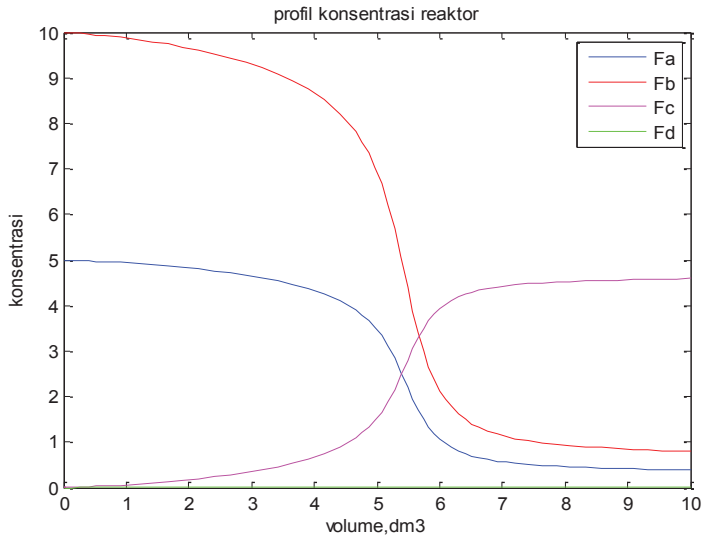
1  function reaction_complex
2  -  clc, clear
3  -  vspan=[0 10];
4  -  Y0=[5 10 0 0 300 325];
5  -  [v,Y]=ode45(@rcomplex,vspan,Y0);
6  -  plot(v,Y(:,1),'-b',v,Y(:,2),'-r',v,Y(:,3),'-m',v,Y(:,4),'-g')
7  -  title('profil konsentrasi reaktor')
8  -  xlabel('volume,dm3')
9  -  ylabel('konsentrasi')
10 -  legend('Fa','Fb','Fc','Fd')
11 -  pause
12 -  plot(v,Y(:,5),'-b',v,Y(:,6),'-r')
13 -  title('Profil Temperatur')
14 -  xlabel('volume,dm3')
15 -  ylabel('Temperatur,K')
16 -  legend('T','Ta')
17 -  Fa0=Y(1,1);Fat=Y(end,1);
18 -  X=(Fa0-Fat)/Fa0
19 -  %-----

```

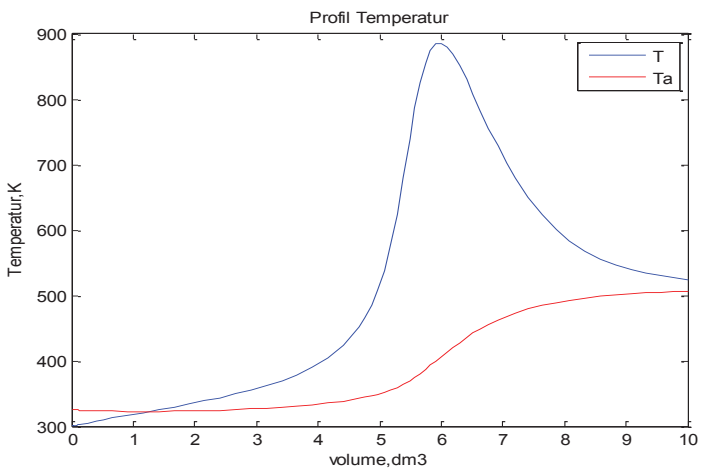
```

20 function dYdV=rcomplex(V,Y)
21 Fa=Y(1);
22 Fb=Y(2);
23 Fc=Y(3);
24 Fd=Y(4);
25 T =Y(5);
26 Ta=Y(6);
27 E2 = 12000;
28 y = 1;
29 R = 1.987;
30 Ft = Fa+Fb+Fc+Fd;
31 To = 300;
32 k2c = 2*exp((E2/R)*((1/300)-(1/T)));
33 E1 = 8000;
34 Cto = 0.2;
35 Ca = Cto*(Fa/Ft)*(To/T)*y;
36 Cc = Cto*(Fc/Ft)*(To/T)*y;
37 r2c = -k2c*Ca^2*Cc^3;
38 Cpco = 10;
39 m=50;
40 Cb = Cto*(Fb/Ft)*(To/T)*y;
41 k1a=40*exp((E1/R)*((1/300)-(1/T)));
42 r1a = -k1a*Ca*Cb^2;
43 r1b = 2*r1a;
44 rb = r1b;
45 r2a = (2/3)*r2c;
46 DH1b = -15000;
47 DH2a =-10000;
48 r1c = -r1a;
49 Ta55= 325;
50 Cpd = 16;
51 Cpa = 10;
52 Cpb = 12;
53 Cpc = 14;
54 sumFiCpi = Cpa*Fa+Cpb*Fb+Cpc*Fc+Cpd*Fd;
55 rc = r1c+r2c;
56 Ua = 80;
57 r2d = -(1/3)*r2c;
58 ra = r1a+r2a;
59 rd = r2d;
60 Og = r1b*DH1b+r2a*DH2a;
61 Or = Ua*(T-Ta);
62 dYdV(1) = ra;
63 dYdV(2) = rb;
64 dYdV(3) = rc;
65 dYdV(4) = rd;
66 dYdV(5) = (Og-Or)/sumFiCpi;
67 dYdV(6) = Ua*(T-Ta)/m/Cpco;
68 dYdV=dYdV';

```



(a)



(b)

**Gambar 7.12** Grafik perubahan terhadap volume  
(a) laju alir molar dan (b) temperatur

**(b) Fluida pendingin berlawanan arah (*counterr-current*) dengan reaktan**

Neraca pertukaran panas untuk aliran co-current adalah :

$$\frac{dT_a}{dV} = -\frac{Ua(T - T_a)}{m_c C_{pCO}} \quad (7.6.27)$$

Program Matlab ditunjukkan di bawah dan Gambar 7.13 menunjukkan grafik output.

```

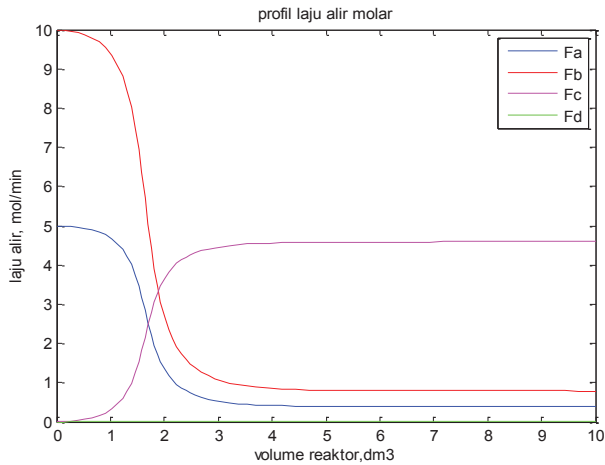
1  function reaction_complex2
2  -   clc, clear
3  -   vspan=[0 10];
4  -   Y0=[5 10 0 0 300 507];
5  -   [v,Y]=ode45(@rcomplex,vspan,Y0);
6  -   plot(v,Y(:,1),'-b',v,Y(:,2),'-r',v,Y(:,3),'-m',v,Y(:,4),'-g')
7  -   title('profil laju alir molar')
8  -   xlabel('volume reaktor,dm3')
9  -   ylabel('laju alir, mol/min')
10 -  legend('Fa','Fb','Fc','Fd')
11 -  pause
12 -  plot(v,Y(:,5),'-b',v,Y(:,6),'-r')
13 -  title('Profil Temperatur')
14 -  xlabel('volume reaktor,dm3')
15 -  ylabel('Temperatur,K')
16 -  legend('T','Ta')
17 -  Fa0=Y(1,1);Fat=Y(end,1);
18 -  X=(Fa0-Fat)/Fa0
19 -  %-----
20 function dYdV=rcomplex(v,Y)
21 -   Fa=Y(1);
22 -   Fb=Y(2);
23 -   Fc=Y(3);
24 -   Fd=Y(4);
25 -   T =Y(5);
26 -   Ta=Y(6);
27 -   E2 = 12000;
28 -   y = 1;
29 -   R = 1.987;
30 -   Ft = Fa+Fb+Fc+Fd;
31 -   To = 300;
32 -   k2c = 2*exp((E2/R)*((1/300)-(1/T)));
33 -   E1 = 8000;
34 -   Cto = 0.2;
35 -   Ca = Cto*(Fa/Ft)*(To/T)*y;
36 -   Cc = Cto*(Fc/Ft)*(To/T)*y;
37 -   r2c = -k2c*Ca^2*Cc^3;
38 -   Cpco = 10;m=50;
39 -   Cb = Cto*(Fb/Ft)*(To/T)*y;
40 -   k1a=40*exp((E1/R)*((1/300)-(1/T)));
41 -   r1a = -k1a*Ca*Cc^2;
42 -   r1b = 2*r1a;r2a = r1b;r2a = (2/3)*r2c;
43 -   DH1b = -15000;DH2a =-10000;
44 -   r1c = -r1a;
45 -   Ta55= 325;
46 -   Cpd = 16;Cpa = 10;Cpb = 12;Cpc = 14;
47 -   sumFiCpi = Cpa*Fa+Cpb*Fb+Cpc*Fc+Cpd*Fd;

```

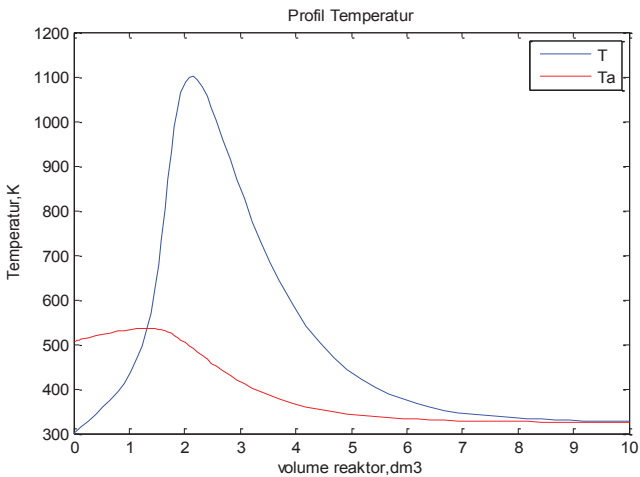
```

48 - rc = r1c+r2c;Ua = 80;
49 - r2d = -(1/3)*r2c;
50 - ra = r1a+r2a;r2d = r2d;
51 - Qg = r1b*DH1b+r2a*DH2a;
52 - Qr = Ua*(T-Ta);
53 - dYdV(1) = ra;
54 - dYdV(2) = rb;
55 - dYdV(3) = rc;
56 - dYdV(4) = r2d;
57 - dYdV(5) = (Qg-Qr)/sumFiCpi;
58 - dYdV(6) = -Ua*(T-Ta)/m/Cpco;
59 - dYdV=dYdV';

```



(a)



(b)

**Gambar 7.13** Grafik perubahan terhadap volume  
(a) laju alir molar dan (b) temperatur

### (c) Pertukaran panas dengan $T_a$ konstan

Neraca pertukaran panas untuk  $T_a$  konstan adalah :

$$\frac{dT_a}{dV} = -\frac{Ua(T - T_a)}{m_c C_{pCO}} * 0$$

Program Matlab ditunjukkan di bawah dan Gambar 7.14 menunjukkan grafik output.

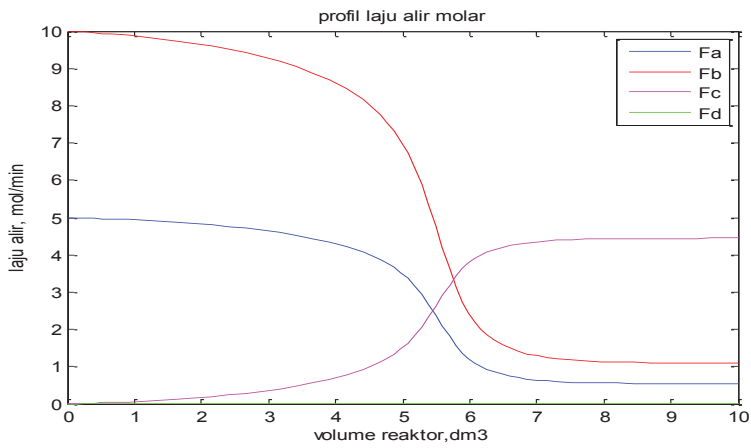
```
1 function reaction_complex3
2 clc, clear
3 vspan=[0 10];
4 Y0=[5 10 0 0 300 325];
5 [v,Y]=ode45(@rcomplex,vspan,Y0);
6 plot(v,Y(:,1),'-b',v,Y(:,2),'-r',v,Y(:,3),'-m',v,Y(:,4),'-g')
7 title('profil laju alir molar')
8 xlabel('volume reaktor, dm3')
9 ylabel('laju alir, mol/min')
10 legend('Fa', 'Fb', 'Fc', 'Fd')
11 pause
12 plot(v,Y(:,5),'-b',v,Y(:,6),'-r')
13 title('Profil Temperatur')
14 xlabel('volume reaktor, dm3')
15 ylabel('Temperatur, K')
16 legend('T', 'Ta')
17 Fa0=Y(1,1); Fat=Y(end,1);
18 X=(Fa0-Fat)/Fa0
19 %-----
20 function dYdV=rcomplex(Y)
21 Fa=Y(1);
22 Fb=Y(2);
23 Fc=Y(3);
24 Fd=Y(4);
25 T=Y(5);
26 Ta=Y(6);
27 E2 = 12000;
28 y = 1;
29 R = 1.987;
30 Ft = Fa+Fb+Fc+Fd;
31 To = 300;
32 k2c = 2*exp((E2/R)*((1/300)-(1/T)));
33 Cto = 0.2;
34 Ca = Cto*(Fa/Ft)*(To/T)*y;
35 Cc = Cto*(Fc/Ft)*(To/T)*y;
36 r2c = -k2c*Ca^2*Cc^3;
37 Cpc0 = 10;
38 m=50;
39 Cb = Cto*(Fb/Ft)*(To/T)*y;
40 k1a=40*exp((E1/R)*((1/300)-(1/T)));
41 r1a = -k1a*Ca*Cb^2;
42 r1b = 2*r1a; rb = r1b; r2a = (2/3)*r2c;
43 DH1b = -15000; DH2a = -10000;
44 r1c = -r1a;
45 Ta55= 325;
```



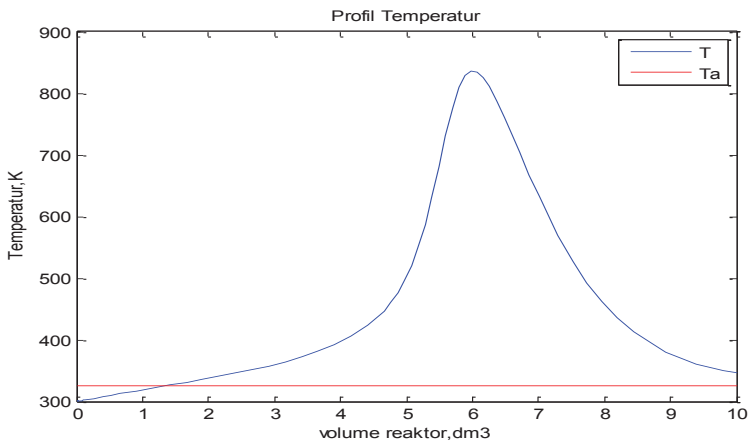
```

46 - Cpd = 16;Cpa = 10;Cpb = 12;Cpc = 14;
47 - sumFiCpi = Cpa*Fa+Cpb*Fb+Cpc*Fc+Cpd*Fd;
48 - rc = r1c+r2c;Ua = 80;
49 - r2d = -(1/3)*r2c;
50 - ra = r1a+r2a;rd = r2d;
51 - Qg = r1b*DH1b+r2a*DH2a;
52 - Qr = Ua*(T-Ta);
53 - dYdV(1) = ra;
54 - dYdV(2) = rb;
55 - dYdV(3) = rc;
56 - dYdV(4) = rd;
57 - dYdV(5) = (Qg-Qr)/sumFiCpi;
58 - dYdV(6) = (-Ua*(T-Ta)/m/Cpco)*0;
59 - -dYdV=dYdV';

```



(a)



(b)

**Gambar 7.14** Grafik perubahan terhadap volume  
(a) laju alir molar dan (b) temperatur

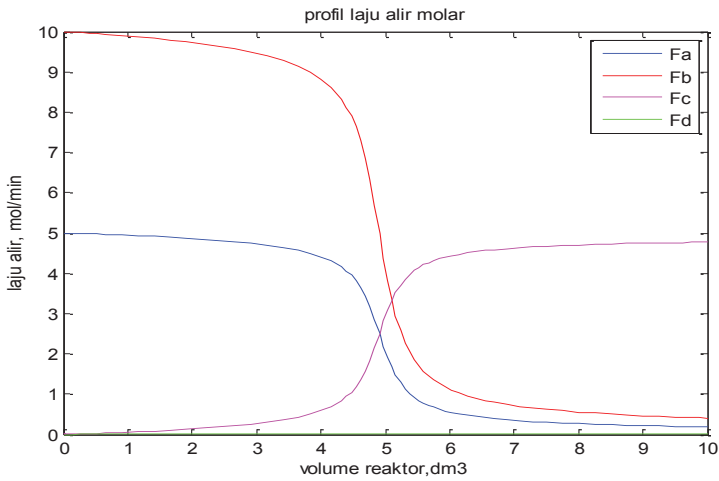
## (d) Proses Adiabatis

```
1 function reaction_complex4
2 -   clc, clear
3 -   vspan=[0 10];
4 -   Y0=[5 10 0 0 300];
5 -   [v,Y]=ode45(@rcomplex,vspan,Y0);
6 -   plot(v,Y(:,1),'-b',v,Y(:,2),'-r',v,Y(:,3),'-m',v,Y(:,4),'-g')
7 -   title('profil laju alir molar')
8 -   xlabel('volume reaktor, dm3')
9 -   ylabel('laju alir, mol/min')
10 -  legend('Fa','Fb','Fc','Fd')
11 -  pause
12 -  plot(v,Y(:,5),'-b')%,v,Y(:,6),'-r')
13 -  title('Profil Temperatur')
14 -  xlabel('volume reaktor, dm3')
15 -  ylabel('Temperatur, K')
16 -  legend('T','Ta')
17 -  Fa0=Y(1,1);Fat=Y(end,1);
18 -  X=(Fa0-Fat)/Fa0
19 -  %-----
20 function dYdV=rcomplex(v,Y)
21 -   Fa=Y(1);
22 -   Fb=Y(2);
23 -   Fc=Y(3);
24 -   Fd=Y(4);
25 -   T =Y(5);
26 -   %Ta=Y(6);
27 -   E2 = 12000;
28 -   y = 1;
29 -   R = 1.987;
30 -   Ft = Fa+Fb+Fc+Fd;
31 -   To = 300;
32 -   k2c = 2*exp((E2/R)*((1/300)-(1/T)));
33 -   E1 = 8000;
34 -   Cto = 0.2;
35 -   Ca = Cto*(Fa/Ft)*(To/T)*y;
36 -   Cc = Cto*(Fc/Ft)*(To/T)*y;
37 -   r2c = -k2c*Ca^2*Cc^3;
38 -   Cpco = 10;
39 -   m = 50;
40 -   Cb = Cto*(Fb/Ft)*(To/T)*y;
41 -   k1a=40*exp((E1/R)*((1/300)-(1/T)));|
42 -   r1a = -k1a*Ca*Cc^2;
43 -   r1b = 2*r1a;rb = r1b;r2a = (2/3)*r2c;
44 -   DH1b = -15000;DH2a =-10000;
45 -   r1c = -r1a;
46 -   Ta= 325;
47 -   Cpd = 16;Cpa = 10;Cpb = 12;Cpc = 14;
48 -   sumFiCpi = Cpa*Fa+Cpb*Fb+Cpc*Fc+Cpd*Fd;
49 -   rc = r1c+r2c;Ua = 80*0;
50 -   r2d = -(1/3)*r2c;
```

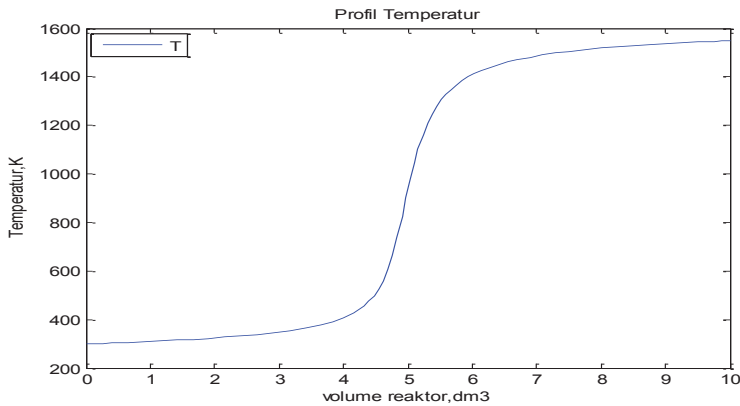
```

51 - ra = r1a+r2a;rd = r2d;
52 - Qg = r1b*DH1b+r2a*DH2a;
53 - Qr = Ua*(T-Ta);
54 - dYdV(1) = ra;
55 - dYdV(2) = rb;
56 - dYdV(3) = rc;
57 - dYdV(4) = rd;
58 - dYdV(5) = (Qg-Qr)/sumFiCpi;
59 - %dYdV(6) = (-Ua*(T-Ta)/m/Cpco);
60 - dYdV=dYdV';

```



(a)



(b)

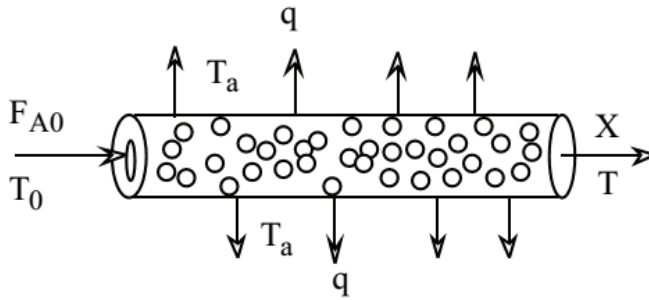
**Gambar 7.15** Grafik perubahan terhadap volume  
(a) laju alir molar dan (b) temperatur

### Contoh 7.7 Fixed Bed Reactor

Reaksi elementer fasa gas dijalankan dalam reaktor fixed bed. Terdapat penukar panas di sekeliling reaktor, dan terdapat pressure drop sepanjang reaktor.



Gas masuk reaktor pada suhu 450 K.



**Gambar 7.16** Grafik perubahan terhadap volume  
(a) laju alir molar dan (b) temperatur

Laju untuk reaksi katalisis di atas dinyatakan dengan persamaan :

$$r_A = -k \left[ C_A^2 - \frac{C_C}{K_C} \right] \quad (7.7.1)$$

Konstanta laju reaksi berdasarkan reaktan  $A$  dan mengikuti persamaan Arrhenius :

$$k = k_{T=450K} \exp \left[ \frac{E}{R} \left( \frac{1}{T_0} - \frac{1}{T} \right) \right] \quad (7.7.2)$$

Kontanta kesetimbangan berubah terhadap suhu dan ditentukan dari persamaan van't Hoff's dengan  $\Delta C_p = 0$ .

$$K_C = K_{T=450K} \exp \left[ \frac{\Delta H_R}{R} \left( \frac{1}{T_0} - \frac{1}{T} \right) \right] \quad (7.7.3)$$

Beberapa parameter lainnya diberikan di bawah :

$C_{PA} = 40 \text{ J/g-mol.K}$	$R = 8,314 \text{ J/g-mol.K}$
$C_{PC} = 80 \text{ J/g-mol.K}$	$F_{A0} = 5 \text{ g-mol/min}$
$\Delta H_R = -40.000 \text{ J/g-mol}$	$Ua/\rho_c = 0,8 \text{ J/kg cat.min.K}$
$E = 41.800 \text{ J/g-mol.K}$	$T_a = 500 \text{ K}$
$k_0 = 0.5 \text{ dm}^6/\text{kg-min-mol @ 450 K}$	$\alpha = 0,019/\text{kg katalis}$
$K_0 = 25.000 \text{ dm}^3/\text{g-mol @ 450 K}$	$P_0 = 10 \text{ atm}$
$C_{A0} = 0,271 \text{ g-mol/dm}^3$	$y_{A0} = 1,0 \text{ (Pure A feed)}$
$T_0 = 450 \text{ K}$	$W_f = 20 \text{ kg catalis}$

- (a) Buat grafik konversi ( $x$ ), tekanan reduksi ( $y$ ) dan temperatur ( $T \times 10^{-3}$ ) sepanjang reaktor dari  $W = 0 \text{ kg}$  sampai  $W = 20 \text{ kg}$  katalis.
- (b) Buat grafik profil konsentrasi reaktan A dan produk C dari  $W = 0 \text{ kg}$  sampai  $W = 20 \text{ kg}$  katalis.

### Penyelesaian :

Pada kasus di atas diasumsikan tidak ada gradien konsentrasi dan temperatur pada arah radial di beberapa titik dalam katalis. Desain reaktor akan menggunakan konversi A yang dinyatakan dengan  $X$  dan temperatur  $T$  yang keduanya sebagai fungsi lokasi di dalam bed katalis tertentu dengan berat katalis  $W$ .

### Neraca mol :

$$\frac{dX}{dW} = \frac{-r_A}{F_{A0}} \quad (7.7.4)$$

### Persamaan laju reaksi :

$$r_A = -k \left[ C_A^2 - \frac{C_C}{K_C} \right] \quad (7.7.1)$$

Konstanta laju reaksi :

$$k = 0,5 \exp \left[ \frac{E}{R} \left( \frac{1}{T_0} - \frac{1}{T} \right) \right] \quad (7.7.5)$$

konstanta kesetimbangan reaksi :

$$K_C = 2,5 \times 10^4 \exp \left[ \frac{\Delta H_R}{R} \left( \frac{1}{T_0} - \frac{1}{T} \right) \right] \quad (7.7.6)$$

### Stoikiometri :

Konsentrasi dinyatakan sebagai fungsi konversi dan suhu yang mengalami perubahan volumetrik karena penurunan jumlah mol selama reaksi. Oleh karena itu :

$$C_A = C_{A0} \left( \frac{1-X}{1+\varepsilon X} \right) \frac{P}{P_0} \frac{T_0}{T} = C_{A0} \left( \frac{1-X}{1-0,5X} \right) y \frac{T_0}{T} \quad (7.7.7)$$

$$y = \frac{P}{T_0}$$

$$C_C = \left( \frac{0,5C_{A0}X}{1-0,5X} \right) y \frac{T_0}{T} \quad (7.7.8)$$

### Neraca Energi :

Neraca energi dapat dituliskan:

$$\frac{dT}{dW} = \frac{\frac{Ua}{\rho_c} (T_a - T) + (-r_A)(-\Delta H_{rxn})}{F_{A0} \left( \sum \Theta_i C_{pi} + X \Delta C_p \right)} \quad (7.7.9)$$

Jika hanya reaktan A maka dapat disederhanakan :

$$\frac{dT}{dW} = \frac{\frac{Ua}{\rho_c}(T_a - T) + (-r_A)(-\Delta H_{rxn})}{F_{A0}C_{pA}} \quad (7.7.10)$$

**Pressure Drop :**

Pressure drop dapat dinyatakan sebagai persamaan diferensial:

$$\frac{d\left(\frac{P}{P_0}\right)}{dW} = \frac{-\alpha(1 + \varepsilon X) P_0 T}{2 P T_0} \quad (7.7.11)$$

atau

$$\frac{dy}{dW} = \frac{-\alpha(1 - 0,5 X) T}{2y T_0} \quad (7.7.12)$$

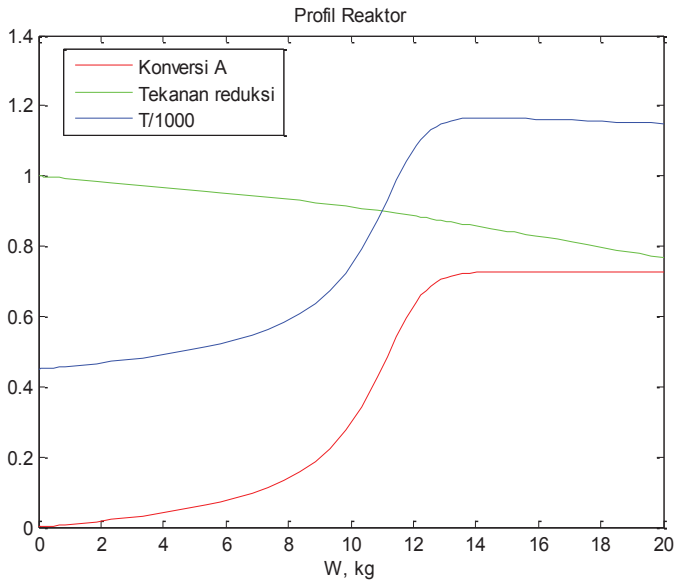
Penulisan program untuk penyelesaian kasus di atas dicantumkan di bawah dan grafik simulasi ditunjukkan pada Gambar 7.17..

```

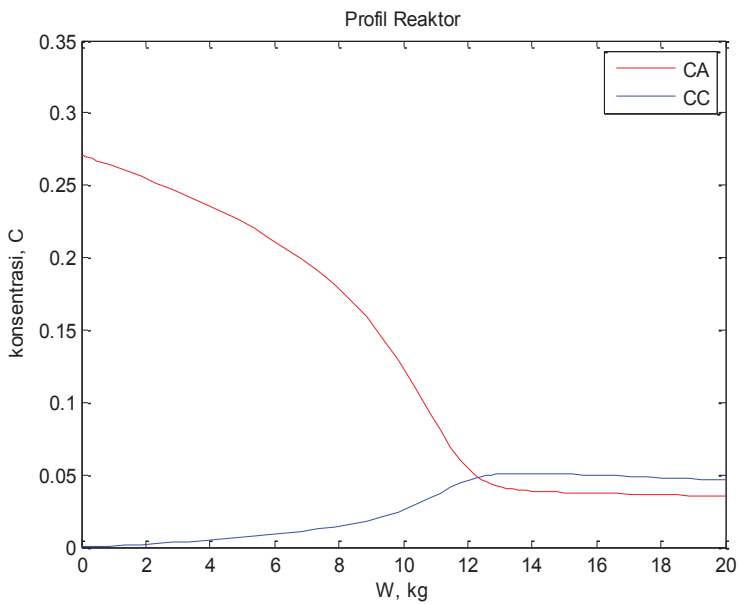
1  function fixbed_reactor
2  -   clc, clear
3  -   global Ta delH CPA FAO
4  -   %set parameters
5  -   Ta = 500;
6  -   delH = -40000;
7  -   CPA = 40;
8  -   FAO = 5;
9  -   %nilai awal
10 -   Y0=[0 450 1];
11 -   wspan=[0 20];
12 -   %integrasi persamaan
13 -   [w,Y]=ode45(@fixbr,wspan,Y0);
14 -   %plot hasil
15 -   Tt = Y(:,2)/1000;%suhu dengan pembagi 1000
16 -   plot(w,Y(:,1), 'r',w,Y(:,3), 'g',w,Tt, 'b')
17 -   title ('Reactor Model')
18 -   xlabel ('W')
19 -   legend('Konversi A','Tekanan reduksi','T/1000')
20 -   pause
21 -   x=Y(:,1);
22 -   T=Y(:,2);
23 -   y=Y(:,3);
24 -   temp = 0.271.*(450./T).*y./(1-0.5.*x);
25 -   CA = temp.*(1-x);
26 -   CC = temp.* 0.5.*x;
27 -   plot(w,CA, 'r',w,CC, 'b')
28 -   title ('konsentrasi')
29 -   xlabel ('W')
30 -   ylabel ('konsentrasi, C')
31 -   legend('CA','CC')
32 -   %-----
33  function dYdw=fixbr(w,Y)
34 -   global Ta delH CPA FAO
35 -   x = Y(1);
36 -   T = Y(2);
37 -   y = Y(3);
38 -   k = 0.5*exp(5032*(1/450 - 1/T));
39 -   temp = 0.271*(450/T)*y/(1-0.5*x);
40 -   CA = temp * (1-x);
41 -   CC = temp * 0.5 * x;
42 -   Kc = 25000*exp(delH/8.314*(1/450-1/T));
43 -   rA = -k*(CA*CA - CC/Kc);
44 -   dYdw(1) = -rA/FAO;
45 -   dYdw(2) = (0.8*(Ta-T)+rA*delH)/(CPA*FAO);
46 -   dYdw(3) = -0.015*(1-0.5*x)*(T/450)/(2*y);
47 -   dYdw=dYdw';

```





(a)

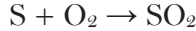


(b)

**Gambar 7.17** Grafik perubahan terhadap berat katalis (a) konversi, tekanan dan  $T/1000$ , (b) konsentrasi A dan C

### Contoh 7.8. Oksidasi Sulfur Dioksida

Dalam pembuatan asam sulfat dari belerang, tahap pertama adalah pembakaran belerang dalam suatu furnace untuk menjadi belerang dioksida (Fogler, 1999):



Selanjutnya  $SO_2$  dikonversi menjadi sulfur trioksida menggunakan katalis :



Persamaan laju reaksi untuk oksidasi  $SO_2$  dalam katalis tertentu adalah :

$$-r'_{SO_2} = k \sqrt{\frac{P_{SO_2}}{P_{SO_3}}} \left[ P_{O_2} - \left( \frac{P_{SO_3}}{K_p P_{SO_2}} \right)^2 \right] \quad (7.8.1)$$

Dimana  $P_i$  adalah tekanan parsial komponen  $i$ . Persamaan ini dapat digunakan ketika konversi lebih besar dari 5%.

Untuk konversi dibawah 5%, laju reaksi mengikuti persamaan :

$$-r'_{SO_2} = k \left( 0,848 - \frac{0,1176}{K_p^2} \right) \quad (7.8.2)$$

Umpan ke  $SO_2$  konverter adalah 7900 lbmol/h, dan terdiri 11%  $SO_2$ , 10%  $O_2$ , dan 79 % inert (sebagai  $N_2$ ). Konverter terdiri atas 4.631 tube dengan katalis padat, panjang masing-masing 20 ft. Tube memiliki 3 in O.D. dan 2,871 in I.D. Tube katalis akan didinginkan dengan cairan mendidih pada 805 °F sehingga temperatur pendingin konstan. Tekanan masuk reaktor 2 atm.

Untuk temperatur masuk 740°F dan 940°F, plot profil konversi, temperatur dan pressure drop.

Diberikan data :

$$\phi = 0,45$$

$$U = 10 \text{ Btu/jam.ft}^2.\text{oR}$$

$$\rho_0 = 0,054 \text{ lb/ft}^3$$

$$A_c = 0,0442 \text{ ft}^2$$

$$P_o = 2 \text{ atm}$$

$$T_o = 1400^\circ\text{R}$$

$$D_p = 0,015 \text{ ft}$$

$$g_c = 4,17 \times 10^8 \text{ lbf}\cdot\text{ft}/\text{lbm}\cdot\text{h}^2$$

$$\mu = 0,090 \text{ lb/ft}\cdot\text{h pada } 1400^\circ\text{R}$$

Konstanta kesetimbangan pada  $T$  adalah

$$K_p = \exp\left(\frac{42.311}{RT} - 11,24\right) \quad (K_p \text{ dalam atm}^{-1/2}, T \text{ dalam } ^\circ\text{R}) \quad (7.8.3)$$

Konstanta laju reaksi dinyatakan dengan persamaan :

$$k = \exp\left[\frac{-176.008}{RT} - (110,1 \ln T) + 912,8\right] \quad (k \text{ dalam lbmol SO}_2/\text{lbcat}\cdot\text{s}\cdot\text{atm}, T \text{ dalam } ^\circ\text{R}) \quad (7.8.4)$$

Terdapat efek difusional pada katalis pada suhu tersebut dan persamaan di atas harus dipandang sebagai persamaan empiris yang memperkirakan konstanta laju reaksi efektif dalam kisaran suhu  $814^\circ\text{--}1138^\circ\text{F}$ . Tabel JANAF digunakan untuk memberikan data berikut:

$$C_{p\text{SO}_2} = 7,208 + 5,633 \times 10^{-3} T - 1,343 \times 10^{-6} T^2 \quad (7.8.5)$$

$$C_{p\text{O}_2} = 5,731 + 2,323 \times 10^{-3} T - 4,886 \times 10^{-6} T^2 \quad (7.8.6)$$

$$C_{p\text{SO}_3} = 8,511 + 9,517 \times 10^{-3} T - 3,325 \times 10^{-6} T^2 \quad (7.8.7)$$

$$C_{p\text{N}_2} = 6,248 + 8,778 \times 10^{-4} T - 2,130 \times 10^{-8} T^2 \quad (7.8.8)$$

Dengan  $C_p$  dalam  $\text{Btu}/\text{lbmol}\cdot^\circ\text{R}$  dan  $T$  dalam  $^\circ\text{R}$ .

$$\Delta H_R(800^\circ\text{F}) = -42.471 \text{ Btu}/\text{lbmol SO}_2$$

**Penyelesaian :**

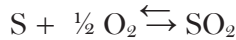
**Neraca mol:**

$$\frac{dX}{dW} = \frac{-r'_A}{F_{A0}} \quad (7.8.9)$$

### Persamaan Laju:

$$-r'_{SO_2} = k \sqrt{\frac{P_{SO_2}}{P_{SO_3}}} \left[ P_{O_2} - \left( \frac{P_{SO_3}}{K_p P_{SO_2}} \right)^2 \right] \quad (7.8.1)$$

### Hubungan Stoikiometri :



Kita misalkan A menyatakan  $SO_2$  dan  $v_i$  koefisien stoikiometri untuk komponen  $i$ :

$$P_i = C_i (RT) = C_{A0} \frac{(\Theta_i + v_i X)(RT)P}{(1 + \epsilon X)(T/T_0)P_0} = P_{A0} \frac{(\Theta_i + v_i X)(RT)P}{(1 + \epsilon X)P_0} \quad (7.8.10)$$

Substitusi untuk tekanan parsial dalam laju reaksi dan kombinasi menghasilkan :

$$\begin{aligned} \frac{dX}{dW} &= \frac{-r'_A}{F_{A0}} \\ &= \frac{k}{F_{A0}} \sqrt{\frac{1-X}{\Theta_{SO_3} + X}} \left[ \frac{P}{P_0} P_{A0} \left( \frac{\Theta_{SO_3} - \frac{1}{2} X}{1 + \epsilon X} \right) - \left( \frac{\Theta_{SO_3} + X}{1-X} \right)^2 \frac{1}{K_p^2} \right] \end{aligned} \quad (7.8.11)$$

Dimana :

$$\epsilon = -0,055, P_{A0} = 0,22 \text{ atm}; \Theta_{SO_2} = 1,0; \Theta_{O_2} = 0,91; \Theta_{SO_3} = 0,0;$$

$$\Theta_{N_2} = 7,17;$$

$$F_{T0} = 7.900 \text{ lb mol/h}; F_{A0} = 869 \text{ lb mol/h}$$

### Per tube :

Berat katalis dalam satu tube =

$$W = \rho_c \frac{\pi D^2}{4} L = 18,54 \text{ lb cat./tube}$$

$$F_{A0} = \frac{869}{4.631} = 0,188 \text{ lb mo/tube}$$

Substitusi nilai diatas diperoleh :

$$\frac{dX}{dW} = 5,329k \sqrt{\frac{1-X}{X}} \left\{ \frac{0,2 - 0,11X}{1 - 0,055X} \frac{P}{P_0} - \left[ \frac{X}{(1-X)K_p} \right]^2 \right\} \quad (7.8.12)$$

**Neraca Energi :**

$$\frac{dT}{dW} = \frac{\frac{4U}{\rho_c D} (T_a - T) + (-r'_A) (-\Delta H_R(T))}{F_{A0} (\sum \Theta_i C_{pi} + X \Delta C_p)} \quad (7.8.13)$$

$$\Delta H_R(T) = \Delta H_R^0(T) + \Delta \alpha (T - T_R) + \frac{\Delta \beta}{2} (T^2 - T_R^2) + \frac{\Delta \gamma}{3} (T^3 - T_R^3) \quad (7.8.14)$$

Untuk oksidasi  $\text{SO}_2$ ,  $\text{S} + \frac{1}{2} \text{O}_2 \rightarrow \text{SO}_2$ ,

$$\Delta \alpha = \alpha_{\text{SO}_3} - \frac{1}{2} \alpha_{\text{O}_2} - \alpha_{\text{SO}_2} = 8,511 - (0,5)(5,731) - 7,208 = -1,563$$

$$\begin{aligned} \Delta \beta &= \beta_{\text{SO}_3} - \frac{1}{2} \beta_{\text{O}_2} - \beta_{\text{SO}_2} = (9,517 \times 10^{-3}) - \frac{1}{2} (2,323 \times 10^{-3}) - (5,633 \times 10^{-3}) \\ &= -0,738 \times 10^{-6} \end{aligned}$$

$$\begin{aligned} \Delta \gamma &= \gamma_{\text{SO}_3} - \frac{1}{2} \gamma_{\text{O}_2} - \gamma_{\text{SO}_2} = (-2,325 \times 10^{-6}) - \frac{1}{2} (-4,886 \times 10^{-7}) - (-1,343 \times 10^{-6}) \\ &= -0,738 \times 10^{-6} \end{aligned}$$

Substitusi ke persamaan (7.9.9) dengan  $T_R = 1260^\circ \text{R}$  diperoleh :

$$\begin{aligned} \Delta H_R(T) &= -42,471 - (1,563)(T - 1260) + (1,36 \times 10^{-3})(T^2 - 1260^2) \\ &\quad - (2,459 \times 10^{-7})(T^3 - 1260^3) \end{aligned} \quad (7.8.15)$$

$\Delta H_R$  dalam Btu/lb mol,  $T$  dalam R

$$\sum \Theta_i C_{pi} = 57,23 + 0,014T - 1,94 \times 10^{-6} T^2 \quad (7.8.16)$$

Koefisien transfer panas :

$$\begin{aligned} \frac{U\pi D}{\rho_c A_c} &= \frac{4U}{\rho_c D} = \frac{4(10 \text{ Btu/h} \cdot \text{ft}^2 \cdot ^\circ \text{R})}{(33,8 \text{ lb/ft}^2) [(2,78/12) \text{ft}]} \\ &= 5,11 \text{ Btu/h} \cdot \text{lb} \cdot \text{cat} \cdot ^\circ \text{R} \end{aligned}$$

Persamaan (7.8.13) menjadi :

$$\frac{dT}{dW} = \frac{5,11(T_a - T) + (-r'_A)(-\Delta H_R(T))}{0,188(\sum \Theta_i C_{pi} + X \Delta C_p)} \quad (7.8.17)$$

### Pressure drop :

Pressure drop diberikan dengan persamaan :

$$\frac{dP}{dL} = - \frac{(1-\phi)G(1+\epsilon X)}{\rho_0(P/P_0)(T_0/T)g_c D_p \phi^3} \left[ \frac{150\mu(1-\phi)}{D_p} + 1,75G \right] \quad (7.8.18)$$

Dimana :  $G = \sum F_{i0} M_i / A_c = 1307,6 \text{ lb/ft}^2 \cdot \text{h}$

$A_c = \text{cross sectional area} : \pi D^2 / 4$

dengan  $W = \rho_i A_c L$  , maka diperoleh persamaan :

$$\frac{dP}{dW} = - \frac{GTP_0(1-\phi)(1+\epsilon X)}{\rho_c A_c \rho_0 T_0 P g_c D_p \phi^3} \left[ \frac{150\mu(1-\phi)}{D_p} + 1,75G \right] \quad (7.8.19)$$

Substitusi parameter-parameter ke persamaan (7.8.18) diperoleh :

$$\frac{dP}{dW} = - \frac{1,12 \times 10^{-6} (1 - 0,055X) T}{P} (5500\mu + 2288) \quad (7.8.20)$$

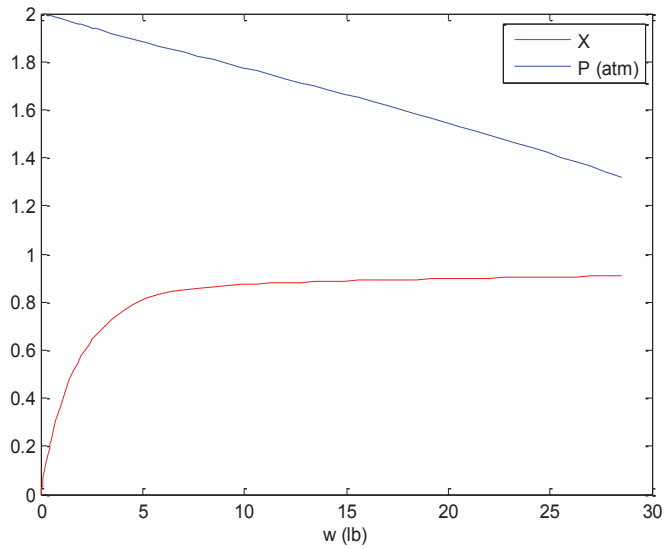
Persamaan (7.8.12), (7.8.17) dan (7.8.20) diselesaikan secara simultan.

Program MATLAB dan grafik hasil simulasi ditunjukkan di bawah.

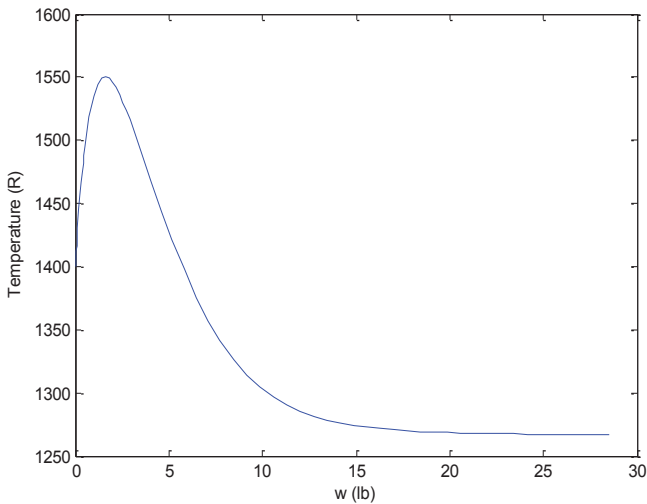
```

1      % Contoh untuk Reaktor Oksidasi SO2
2      function packedreactor_so2
3      -   clc,clear
4      -   wspan = [0 28.54]; % Range untuk berat katalis reaktor
5      -   Y0 = [0 2 1400]; % Nilai awal X, P dan T
6      -   [w Y]=ode45(@pbreactor,wspan,Y0);
7      -   plot(w,Y(:,1),'x',w,Y(:,2),'b')
8      -   legend('X','P (atm)')
9      -   xlabel('w (lb)')
10     -   pause
11     -   plot(w,Y(:,3));
12     -   ylabel('Temperature (R)')
13     -   xlabel('w (lb)')
14     -   %-----
15     -   function dYdw=pbreactor(w,Y)
16     -       x = Y(1);
17     -       P = Y(2);
18     -       T = Y(3);
19     -       % Persamaan eksplisit
20     -       fao = .188;
21     -       visc = .090;
22     -       Ta = 1264.67;
23     -       deltah = -42471-1.563*(T-1260)+.00136*(T^2-1260^2)-2.459*10^(-7)*(T^3-1260^3);
24     -       sum = 57.23+.014*T-1.94*10^(-6)*T^2;
25     -       dcp = -1.5625+2.72*10^(-3)*T-7.38*10^(-7)*T^2;
26     -       k = 3600*exp(-176008/T-(110.1*log(T))+912.8);
27     -       thetaso = 0;
28     -       Po = 2;
29     -       Pao = .22;
30     -       thetao = .91;
31     -       eps = -.055;
32     -       R = 1.987;
33     -       Kp = exp(42311/R/T-11.24);
34     -       if (x<=.05)
35     -           ra = (-k*(.848-.012/(Kp^2)));
36     -       else
37     -           ra=(-k*((1-x)/(thetaso+x))^5*(P/Po*Pao*((thetao-.5*x)/(1+eps*x))...
38     -               -((thetaso+x)/(1-x))^2/(Kp^2)));
39     -       end
40     -       % persamaan diferensial
41     -       dYdw(1) = -(ra)/fao;
42     -       dYdw(2) = (-1.12*10^(-8)*(1-.055*x)*T)*(5500*visc+2288)/P;
43     -       dYdw(3) = (5.11*(Ta-T)+(-ra)*(-deltah))/(fao*(sum+x*dcp));
44     -       dYdw=dYdw';

```



(a)



(b)

### Contoh 7.9. Distribusi Suhu Batang Logam

Suatu batang logam dengan jari-jari,  $R = 0,2$  m, panjang  $L = 10$  m, kedua ujungnya melekat pada dinding yang suhunya berbeda, yaitu  $T_a = 300$  K dan  $T_b = 400$  K (Gambar 7.8). Suhu udara diketahui,  $T_\infty = 200$  K. Diameter batang sangat kecil



dibanding dengan panjang batang dan konduktivitas panas batang besar [ $k = 200 \text{ J}/(\text{s.m.K})$ ] sehingga gradien suhu dalam batang arah radial bisa diabaikan. Koefisien perpindahan panas antara permukaan batang dan udara,  $h = 1 \text{ J}/\text{m}^2\text{K s}$ . Ingin dicari distribusi suhu batang pada keadaan steady state pada arah  $x$  untuk kondisi (a) hanya panas konveksi mengalir dari batang ke udara, (b) ada panas radiasi yang hilang dari batang ke udara dengan  $\sigma = 2,7 \cdot 10^{-9} \text{ K}^{-3}\text{m}^{-2}$ .

**Penyelesaian :**

Pada kasus di atas terjadi dua macam perpindahan panas, yaitu

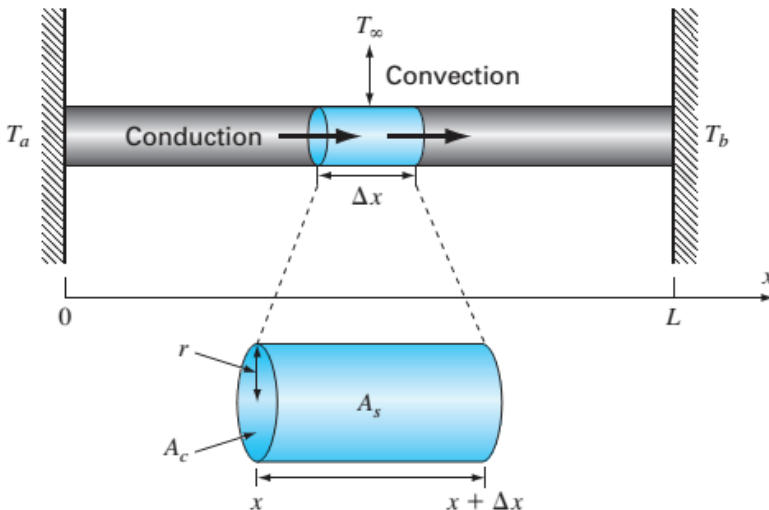
- 1) Perpindahan panas konduksi arah aksial dalam batang,

$$q = -k A \frac{dT}{dx} \tag{7.9.1}$$

- 2) Perpindahan panas antar fasa dari permukaan batang ke udara batang,

$$q = h A(T - T_\infty) \tag{7.9.2}$$

Neraca Panas pada elemen volume sepanjang  $\Delta x$  pada batang :



**Gambar 7.8** Batang logam

$$\text{Rate of input} - \text{Rate of output} = \text{Rate of Accumulation} \quad (7.9.3)$$

$$\left( -k \pi r^2 \frac{dT}{dx} \Big|_x \right) - \left( -k \pi r^2 \frac{dT}{dx} \Big|_{x+\Delta x} + h 2\pi r \Delta x (T - T_\infty) \right) = 0$$

Modifikasi persamaan (7.5.3) menghasilkan :

$$\frac{\frac{dT}{dx} \Big|_{x+\Delta x} - \frac{dT}{dx} \Big|_x}{\Delta x} + \frac{2h}{kr} (T - T_\infty) = 0 \quad (7.9.4)$$

Jika diambil  $\Delta x \rightarrow 0$ , maka menurut definisi matematika diperoleh:

$$\frac{d}{dx} \left( \frac{dT}{dx} \right) - \frac{2h}{kr} (T - T_\infty) = 0 \quad (7.9.5)$$

Persamaan matematis untuk persamaan ini adalah :

$$\boxed{\frac{d^2 T}{dx^2} - \frac{2h}{kr} (T - T_\infty) = 0} \quad (7.9.5)$$

Dengan kondisi batas pada :

$$x = 0 ; T = T_a \quad (7.9.6)$$

$$x = L ; T = T_b \quad (7.9.7)$$

Pada kenyataannya persamaan di atas secara fisik menyatakan kondisi kedua batas batang yang dikenal dengan istilah *boundary condition*.

Penyelesaian persamaan (7.9.5) dengan keadaan batas pers (7.9.6) dan pers (7.9.7) akan menghasilkan  $T = f(x)$ .

(a) Persamaan diatas dapat diselesaikan dengan MATLAB dengan cara merubah persamaan (7.5.5) menjadi dua buah persamaan oder satu simultan.

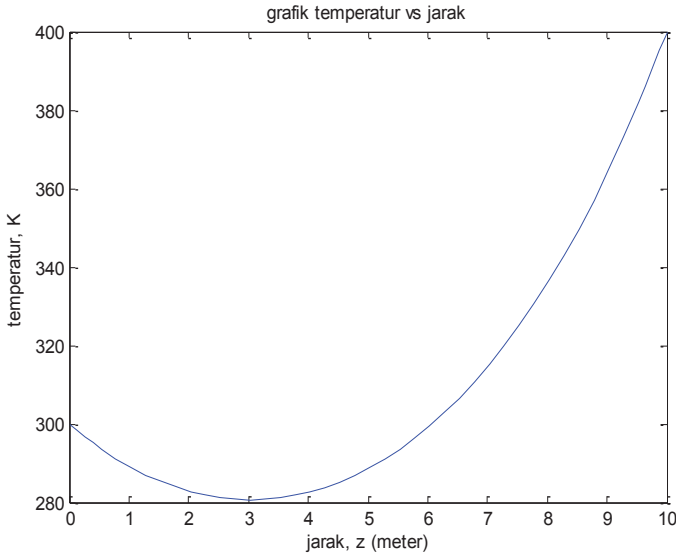
$$\frac{dT}{dx} = z \quad (7.9.5) \quad (7.9.8)$$

$$\frac{dz}{dx} = \frac{2h}{kr} (T - T_\infty) \quad (7.9.5) \quad (7.9.9)$$

Untuk harga  $T(0) = T_a$ , sementara harga  $z(0)$  tidak diketahui. Nilai  $z$  divalusi dengan dengan cara trial and error sehingga diperoleh  $T_b = f(z) = 400$  menggunakan persamaan diferensial.

```
function contoh75A
clc, clear
zaf=fzero(@subcontoh75A,-50);
%-----
function r=subcontoh75A(z0)
T0 = 300; % K
xspan = [0 10];% panjang batang
[x,y]=ode45(@subsubcontoh75A,xspan,[300 z0])
plot(x,y(:,1))
title('grafik temperatur vs jarak');
xlabel('jarak, z (meter)');
ylabel('temperatur, K ');
r = y(length(x),1)-400;
%-----
function dydx=subsubcontoh75A(x,y)
T = y(1); z = y(2);
h = 1; %J/m2K.s
Tudara = 200; %K
k = 200; %J/s m K
r = 0.2; % m
dydx(1)= z;
dydx(2)= 2*h*(T-Tudara)/(k*r);
dydx=dydx';
```

Grafik profil suhu terhadap jarak pada batang ditunjukkan Gambar di bawah



**Gambar 7.9** Profil Suhu pada batang

- (b) Untuk sistem yang melibatkan panas radiasi, persamaan disusun sebagai berikut:

$$\frac{dT}{dx} = z \quad (7.9.5) \quad (7.9.8)$$

$$\frac{dz}{dx} = \frac{2h}{kr}(T - T_{\infty}) + \sigma(T^4 - T_{\infty}^4) \quad (7.9.5) \quad (7.9.10)$$

Program penyelesaian dengan MATLAB sebagai berikut:

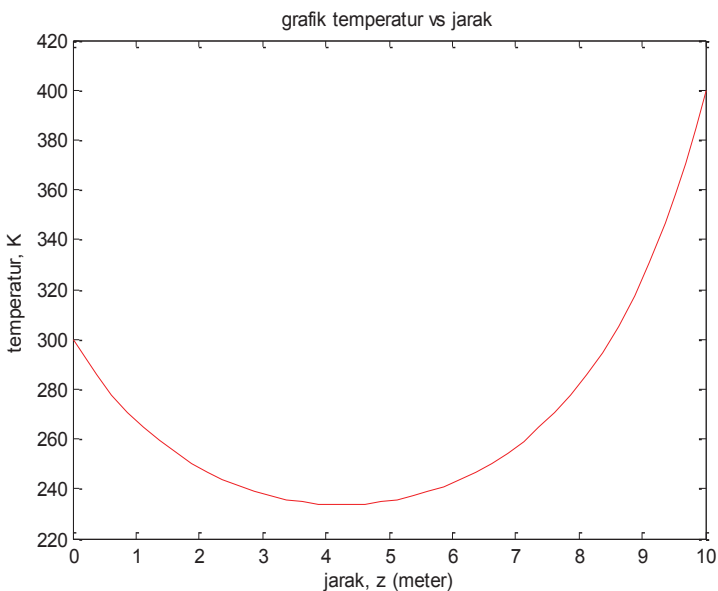
```
function contoh75B
clc, clear
zaf=fzero(@subcontoh75A,-50);
%-----
function r=subcontoh75A(z0)
T0 = 300; % K
xspan = [0 10];% panjang batang
[x,y]=ode45(@subsubcontoh75A,xspan,[300 z0])
```

```

plot(x,y(:,1))
title('grafik temperatur vs jarak');
xlabel('jarak, z (meter)');
ylabel('temperatur, K ');
r = y(length(x),1)-400;
%-----
function dydx=subsubcontoh75A(x,y)
T = y(1); z = y(2);
h = 1; %J/m2K.s
Tudara = 200; %K
k = 200; %J/s m K
r = 0.2; % m
sigma = 2.7E-9;%K/m2
dydx(1)= z;
dydx(2)= 2*h*(T-Tudara)/(k*r)+sigma*(T^4-
Tudara^4);
dydx=dydx';

```

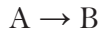
Grafik profil suhu terhadap jarak pada batang ditunjukkan Gambar 7.10 di bawah



**Gambar 7.10** Profil Suhu pada batang dengan panas radiasi

### Contoh 7.10 Difusi dan Reaksi Kimia dalam Katalisator

Reaksi fasa gas order satu :



Berlangsung dipermukaan pori katalisator padat berbentuk bola berjari-jari  $R = 2$  cm. Sistem dianggap isotermis. Konsentrasi  $A$  pada fasa gas di permukaan luar bola  $C_{As} = 5,0 \times 10^{-4}$  gmol/cm<sup>3</sup>. Gas  $A$  mendifusi ke dalam bola melalui pori katalisator sambil sebagian bereaksi di permukaan katalisator dengan konstanta kecepatan reaksi  $k$ . Luas permukaan pori tiap volum katalisator =  $a$ . Harga  $ka = 1,0 \times 10^{-3}$  1/s. Harga difusivitas efektif,  $D_e = 1,0 \times 10^{-4}$  cm<sup>2</sup>/s. Konsentrasi di pusat katalis diasumsikan  $4,5 \times 10^{-4}$  gmol/cm<sup>3</sup>. Kecepatan difusi  $A$  melalui pori dianggap mengikuti persamaan yang mirip dengan difusi gas, yaitu :

$$N_A \left( \frac{\text{gmol A}}{\text{waktu}} \right) = -D_e (\text{luas}) \frac{dC_A}{d(\text{jarak})} = z \quad (7.10.1)$$

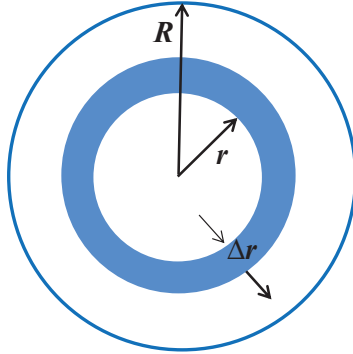
Kecepatan reaksi di permukaan pori katalisator mengikuti persamaan :

$$r_A \left( \frac{\text{gmol A}}{(\text{waktu})(\text{luas pori})} \right) = k C_A \quad (7.10.2)$$

Ingin di cari konsentrasi gas A pada berbagai posisi dalam bola pada keadaan steady.

#### Penyelesaian :

Dalam menyusun model matematis, perpindahan massa  $A$  dalam bola dianggap ke arah  $r$  positif keluar walaupun sebenarnya ke arah  $r$  negatif karena kedua keadaan tersebut akan memberikan hasil persamaan matematis yang sama.



**Gambar 7.11** Elemen volum setebal  $\Delta r$  dalam bola berpori

Neraca massa  $A$  pada elemen volum:

*Rate of input* – *Rate of output* = *Rate of Accumulation*

$$\left( -D_e 4\pi r^2 \frac{dC_A}{dr} \Big|_r \right) - \left( -D_e 4\pi (r+\Delta r)^2 \frac{dC_A}{dr} \Big|_{r+\Delta r} \right) = k C_A a 4\pi r^2 \Delta r \quad (7.10.3)$$

( $4\pi r^2 \Delta r$  menyatakan volume dari elemen volum, yaitu luasnya  $4\pi r^2$  dan tebalnya  $\Delta r$ )

Modifikasi persamaan (7.10.3) dihasilkan :

$$\frac{(r+\Delta r)^2 \frac{dC_A}{dr} \Big|_{r+\Delta r} - \frac{dC_A}{dr} \Big|_r}{\Delta r} - \frac{k a}{D_e} r^2 C_A = 0 \quad (7.10.4)$$

Jika diambil  $\Delta r \rightarrow 0$ , maka menurut definisi matematika diperoleh:

$$\frac{d}{dr} \left( r^2 \frac{dC_A}{dr} \right) - \frac{k a}{D_e} r^2 C_A = 0 \quad (7.10.5)$$

$$r^2 \frac{d^2 C_A}{dr^2} + 2r \frac{dC_A}{dr} - \frac{k a}{D_e} r^2 C_A = 0 \quad (7.10.6)$$

$$\frac{d^2 C_A}{dr^2} + \frac{2}{r} \frac{dC_A}{dr} - \frac{k a}{D_e} C_A = 0 \quad (7.10.7)$$

Dengan kondisi batas pada :

$$r = R ; C_A = C_{As} \quad (7.10.8)$$

$$r = 0 ; C_A = C_{A0} = \frac{dC_A}{dr} = 0 \quad (7.10.9)$$

Persamaan diatas dapat diselesaikan dengan MATLAB dengan cara merubah persamaan (7.10.7) menjadi dua buah persamaan oder satu simultan.

$$\frac{dC_A}{dr} = u \quad (7.10.10)$$

$$\frac{du}{dr} = -\frac{2u}{r} + \frac{ka}{D_e} C_A \quad (7.10.11)$$

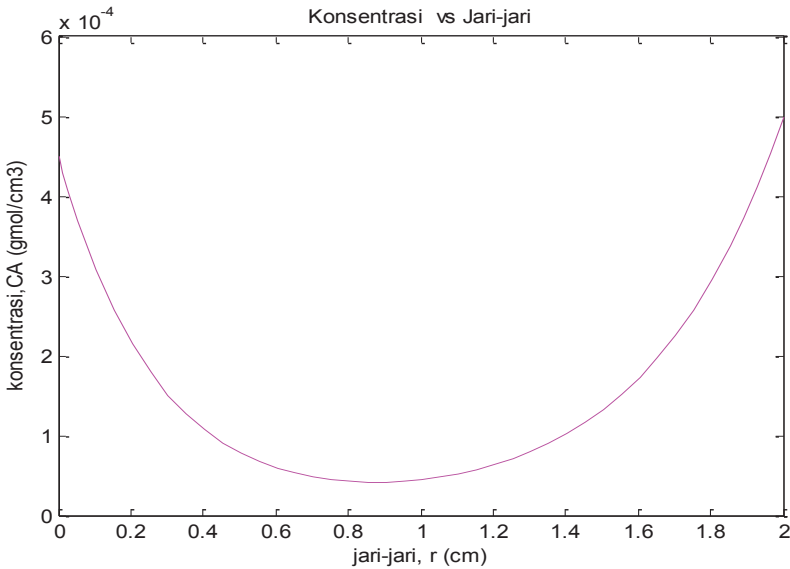
Untuk harga  $C_A(0) = C_{A0}$ , sementara harga  $u(0)$  tidak diketahui. Nilai  $u$  diveluasi dengan degan cara trial and error sehingga diperoleh  $C_A = f(r) = C_{AS}$  menggunakan persamaan diferensial.

```
function contoh76
clc, clear
cas=fzero(@subcontoh76,0.002);
%-----
function r=subcontoh76(u0)
ca0=0.00045;%gmol/cm3
r = 2;%cm
rspan = [0 r];%inteval jari-jari
[x,y]=ode45(@subsubcontoh76,rspan,[ca0 u0]);
plot(x,y(:,1),'m')
title('Konsentrasi vs Jari-jari');
xlabel('jari-jari, r (cm)');
ylabel('konsentrasi,CA (gmol/cm3) ');
%m=y(length(x),1)
r = y(end,1)-5E-4;
%-----
function dydr=subsubcontoh76(x,y)
ca = y(1);u = y(2);
r = 2;% cm
ka = 1E-3; %1/s
De = 1E-4; %cm2/s
dydr(1)= u;
```



$$dy_{dr}(2) = -(2 \cdot u/r) + (k_a \cdot c_a / De);$$

$$dy_{dr} = dy_{dr}';$$



**Gambar 7.12** Elemen volum setebal  $\Delta r$  dalam bola berpori

### 7.3 RANGKUMAN

- Persamaan Diferensial Biasa (PDB) merupakan persamaan diferensial dimana fungsi yang tidak diketahui (variabel terikat) adalah fungsi dari variabel bebas tunggal
- Dalam bidang Teknik Kimia Persamaan diferensial muncul dari kajian proses fisis dan kimia dinamis yang memiliki satu variabel bebas. Variabel tersebut dapat berupa variabel jarak,  $z$  atau variabel waktu,  $t$  yang bergantung pada geometri sistem dan kondisi batasnya.
- Penyelesaian persamaan diferensial biasa dapat diselesaikan dengan MATLAB menggunakan fungsi ode yang akan diperoleh nilai variabel terikat untuk setiap variabel bebas yang diberikan.

## 7.4 RUJUKAN

- a. Constantinidis dan Mustoufi, 1999, *Numerical Methodes for Chemical Engineers with MATLAB Application*, hal 262-264. Prentice-Hall: Englewood Cliffs, NJ.
- b. Finlayson, B.A., 2006, *Introduction to Chemical Engineering Computing*, hal 115-123. John Wiley & Sons Inc., New Jersey
- c. Dan Hanselman dan Bruce Littlefield, 1997. *MATLAB: Bahasa Komputasi dan Teknis*. hal 15 dan 154-165. Andi. Yogyakarta.
- d. Fogler, H. S. 1999. *Elements of Chemical Reaction Engineering*, 3rd ed., Englewood Cliffs, NJ: Prentice-Hall.
- e. B. Klofutar, J. Golob, B. Likozar, C. Klofutar, E.Zagar, Poljanšek D. 2010. *The transesterification of rapeseed and waste sunflower oils: Mass-transfer and kinetics in a laboratory batch reactor and in an industrial-scale reactor/separator setup*. *Bioresource Technology* 101. Pp. 3333-3344
- f. Sediawan, Wahyudi Budi. 1997. *Pemodelan Matematis dan Penyelesaian Numeris dalam Teknik Kimia*. Andi. Yogyakarta.

# PERSAMAAN DIFERENSIAL PARSIAL

## 8.1 PENDAHULUAN

Bab ini membahas tentang persamaan diferensial parsial. Setelah menyelesaikan pokok bahasan ini mahasiswa diharapkan akan mampu menyelesaikan persamaan diferensial parsial dengan MATLAB. Secara khusus mahasiswa diharapkan akan mampu menyusun permasalahan fisis dan kimia dalam Teknik Kimia dalam bentuk persamaan diferensial parsial, menyelesaikan persamaan diferensial parsial dengan *methode of line* dalam MATLAB. Persamaan diferensial parsial banyak ditemukan pada pemodelan-pemodelan teknik kimia berbasis fenomena transpor (momentum, panas dan massa).

## 8.2 MODEL MATEMATIS

Hukum konservasi massa, momentum dan energi membentuk basis penerapan fenomena transpor. Hukum ini berlaku pada aliran fluida yang menghasilkan persamaan perubahan (*equation of change*), yang menggambarkan perubahan kecepatan, temperatur dan konsentrasi terhadap waktu dan posisi. Dinamika

sistem, yang memiliki lebih dari satu variabel bebas dimodelkan dalam persamaan diferensial parsial (Constantinidis dan Mostoufi, 1999).

Sebagai contoh persamaan neraca massa:

$$\left( \begin{array}{c} \text{Rate of mass} \\ \text{accumulation} \end{array} \right) = \left( \begin{array}{c} \text{rate of} \\ \text{mass in} \end{array} \right) - \left( \begin{array}{c} \text{rate of} \\ \text{mass in} \end{array} \right) \quad (8.1)$$

diterapkan pada volume elemen stasioner  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$  melewati fluida murni yang sedang mengalir menghasilkan persamaan kontinuitas (*equation of continuity*):

$$\frac{\partial \rho}{\partial t} = - \left( \frac{\partial}{\partial x} \rho v_x + \frac{\partial}{\partial y} \rho v_y + \frac{\partial}{\partial z} \rho v_z \right) \quad (8.2)$$

Dimana  $\rho$  adalah densitas fluida,  $v_x$ ,  $v_y$ ,  $v_z$  adalah kecepatan komponen dalam tiga koordinat rektanguler.

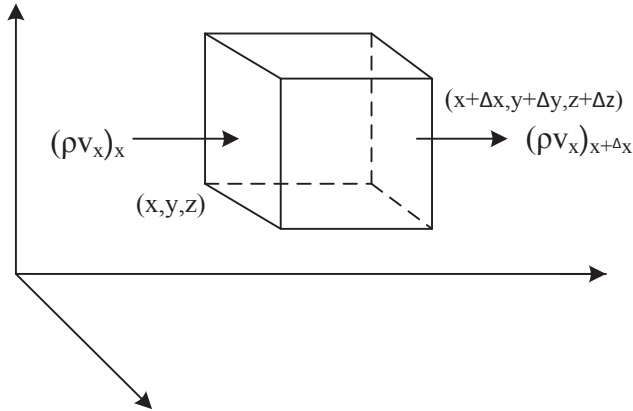
Aplikasi untuk neraca momentum:

$$\left( \begin{array}{c} \text{Rate of} \\ \text{momentum} \\ \text{accumulation} \end{array} \right) = \left( \begin{array}{c} \text{Rate of} \\ \text{momentum} \\ \text{in} \end{array} \right) - \left( \begin{array}{c} \text{Rate of} \\ \text{momentum} \\ \text{out} \end{array} \right) + \left( \begin{array}{c} \text{Sum of forces} \\ \text{action on} \\ \text{system} \end{array} \right) \quad (8.3)$$

Pada elemen volume  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ , untuk aliran fluida isotermal, menghasilkan persamaan gerak dalam tiga arah :

$$\begin{aligned} \frac{\partial}{\partial t} \rho v_j = & - \left( \frac{\partial}{\partial x} \rho v_x v_j + \frac{\partial}{\partial y} \rho v_y v_j + \frac{\partial}{\partial z} \rho v_z v_j \right) \\ & - \left( \frac{\partial}{\partial x} \tau_{xy} + \frac{\partial}{\partial y} \tau_{yy} + \frac{\partial}{\partial z} \tau_{zy} \right) - \frac{\partial p}{\partial j} + \rho g_j \quad j = x, y, \text{ atau } z \end{aligned} \quad (8.4)$$

Dimana  $\tau_{xy}$  adalah shear stress tensor,  $p$  adalah tekanan dan  $g_j$  adalah percepatan gravitasi.



**Gambar 8.1** Elemen Volume  $\Delta x \Delta y \Delta z$  untuk Aliran Fluida

Aplikasi untuk neraca energi adalah:

$$\left( \begin{array}{c} \text{Rate of} \\ \text{accumulation} \\ \text{of energy} \end{array} \right) = \left( \begin{array}{c} \text{Rate of} \\ \text{energy in} \\ \text{by convection} \end{array} \right) - \left( \begin{array}{c} \text{Rate of} \\ \text{energy out} \\ \text{by convection} \end{array} \right) \quad (8.5)$$

$$+ \left( \begin{array}{c} \text{Net Rate of} \\ \text{heat addition} \\ \text{by conduction} \end{array} \right) - \left( \begin{array}{c} \text{Net Rate of work} \\ \text{done by system} \\ \text{on surroundings} \end{array} \right)$$

Pada elemen volume  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ , untuk aliran fluida non-isotermal dihasilkan persamaan neraca energi:

$$\rho C_v \left( \frac{\partial T}{\partial t} + v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) = - \left( \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} \right) \quad (8.6)$$

$$- T \left( \frac{\partial p}{\partial T} \right)_p \left( \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} \right) - \left( \tau_{xx} \frac{\partial v_x}{\partial x} + \tau_{yy} \frac{\partial v_y}{\partial y} + \tau_{zz} \frac{\partial v_z}{\partial z} \right)$$

$$- \left[ \tau_{xy} \left( \frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) + \tau_{xz} \left( \frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) + \tau_{yz} \left( \frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \right) \right]$$

Dimana  $T$  adalah temperatur,  $C$  adalah kapasitas panas pada volume konstan, dan  $q_i$  adalah komponen fluks energi yang diberikan hukum Fourier untuk panas konduksi :

$$q_i = -k \frac{\partial T}{\partial t} \quad i = x, y, \text{ atau } z \quad (8.7)$$

dimana  $k$  adalah konduktivitas termal.

Untuk konduksi dalam padatan, dimana kecepatan adalah nol persamaan (8.6) dapat disederhanakan. Ketika dikombinasikan dengan pers. (8.7), menghasilkan persamaan konduksi panas non seteady state tiga dimensi yang terkenal :

$$\rho C_p \frac{\partial T}{\partial t} = k \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (8.8)$$

dimana  $C_p$  adalah kapasitas panas pada tekanan konstan, menggantikan  $C_v$ , dan  $k$  telah diasumsikan konstan di dalam padatan.

Persamaan kontinuitas untuk komponen  $A$  dalam suatu campuran biner (komponen  $A$  dan  $B$ ) dengan densitas fluida yang konstan,  $\rho$  dan koefisien difusi,  $D_{AB}$  yang konstan adalah :

$$\frac{\partial c_A}{\partial t} + \left( v_x \frac{\partial c_A}{\partial x} + v_y \frac{\partial c_A}{\partial y} + v_z \frac{\partial c_A}{\partial z} \right) = D_{AB} \left( \frac{\partial^2 c_A}{\partial x^2} + \frac{\partial^2 c_A}{\partial y^2} + \frac{\partial^2 c_A}{\partial z^2} \right) + R_A \quad (8.9)$$

Dimana  $c_A$  = konsentrasi molar  $A$  dan  $R_A$  = laju produksi molar komponen  $A$ . Persamaan ini mereduksi hukum kedua *difusi Fick* ketika  $R_A = 0$  dan  $v_x = v_y = v_z = 0$ .

$$\frac{\partial c_A}{\partial t} = D_{AB} \left( \frac{\partial^2 c_A}{\partial x^2} + \frac{\partial^2 c_A}{\partial y^2} + \frac{\partial^2 c_A}{\partial z^2} \right) \quad (8.10)$$

Persamaan diferensial yang paling umum muncul dalam teknik kimia adalah persamaan order satu dan order dua. Persamaan diferensial parsial diklasifikasikan berdasarkan order, linieritas dan kondisi batasnya. Order persamaan diferensial parsial ditentukan oleh order tertinggi turunan parsial dalam persamaan. Contoh persamaan diferensial parsial order satu, dua, tiga adalah :

$$\text{Order satu :} \quad \frac{\partial u}{\partial x} - \alpha \frac{\partial u}{\partial y} = 0 \quad (8.11)$$

$$\text{Order dua :} \quad \frac{\partial^2 u}{\partial x^2} + u \frac{\partial u}{\partial y} = 0 \quad (8.12)$$

$$\text{Order tiga : } \left( \frac{\partial T}{\partial t} \right)^3 + \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial u}{\partial y} = 0 \quad (8.13)$$

Persamaan diferensial dikategorikan kedalam persamaan *linier*, *kuasilinear* dan *nonlinier*. Sebagai contoh, misalkan persamaan order dua berikut:

$$a(\cdot) + \frac{\partial^2 u}{\partial y^2} + 2b(\cdot) \frac{\partial^2 u}{\partial x \partial y} + c(\cdot) \frac{\partial^2 u}{\partial x^2} = 0 \quad (8.14)$$

Jika koefisien adalah konstan atau fungsi variabel independen hanya  $[(\cdot) \equiv (x,y)]$ , maka persamaan (8.14) adalah linier. Jika koefisien adalah fungsi variabel terikat dan atau beberapa turunannya order yang lebih dari persamaan  $[(\cdot) \equiv (x, y, u, \partial u / \partial x, \partial u / \partial y)]$  maka persamaanya adalah kuasilinear. Jika koefisien adalah fungsi turunan yang ordernya sama seperti persamaan  $[(\cdot) \equiv (x, y, u, \partial^2 u / \partial x^2, \partial^2 u / \partial y^2, \partial^2 u / \partial x \partial y)]$  maka persamaanya adalah nonlinier. Berdasarkan definisi di atas, persamaan (8.11) adalah linier, persamaan (8.12) adalah kuasilinear, dan pers (8.13) adalah nonlinier.

Persamaan diferensial order dua linier dalam dua variabel bebas selanjutnya diklasifikasikan ke dalam tiga bentuk kanonik : *elliptic*, *parabolic*, dan *hiperbolic*. Bentuk umum persamaan jenis ini adalah :

$$a \frac{\partial^2 u}{\partial x^2} + 2b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial y} + e \frac{\partial u}{\partial x} + fu + g \quad (8.15)$$

Dimana koefisien adalah konstan atau hanya fungsi variabel independen. Bentuk tiga kanonik ditentukan oleh kriteria berikut:

$$b^2 - ac < 0, \quad \textit{elliptic} \quad (8.16a)$$

$$b^2 - ac = 0, \quad \textit{parabolic} \quad (8.16b)$$

$$b^2 - ac > 0, \quad \textit{hyperbolic} \quad (8.16c)$$

Jika  $g = 0$ , maka persamaan (8.15) adalah persamaan diferensial *homogen*.

Contoh klasik persamaan diferensial order dua yang

membentuk kanonik tiga adalah :

Persamaan Laplace (*elliptic*)

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (8.17)$$

Persamaan difusi atau konduksi panas :

$$\alpha \frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \quad (8.18)$$

Persamaan gelombang :

$$\alpha^2 \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2} \quad (8.18) \quad (8.19)$$

### Contoh 8.1 Perpindahan Panas *Unsteady state*

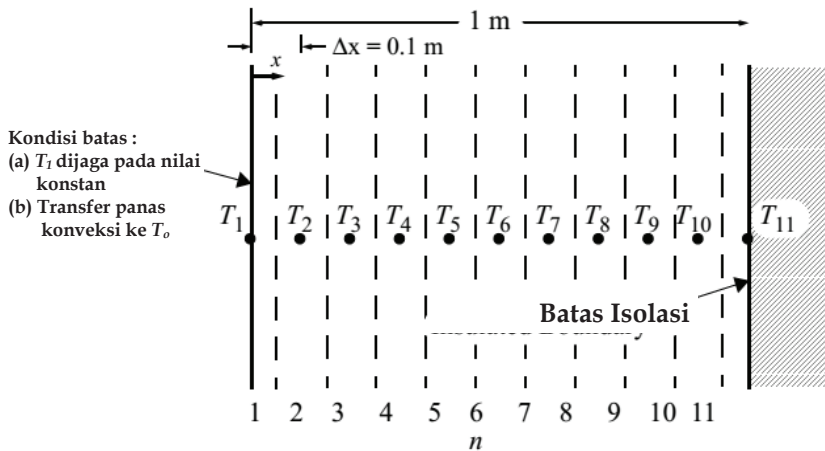
Pada suatu slab yang sangat luas, maka perpindahan panas hanya berlangsung pada arah  $x$ . Perpindahan panas *unsteady state* dalam suatu slab pada arah  $x$  dinyatakan dengan persamaan diferensial parsial :

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (8.1.1)$$

dimana  $T$  adalah temperatur (K) ,  $t$  adalah waktu (s), dan  $\alpha$  adalah difusivitas ( $m^2/s$ ) diberikan dengan  $k/\rho C_p$ . Dalam hal ini konduktivitas termal  $k$  (W/m.K) densitas,  $\rho$  ( $kg/m^3$ ) dan kapasitas panas,  $C_p$  (J/kg.K) adalah konstan (Geankoplis, 1999). Misalkan material slab dengan ketebalan 1 m dilindungi isolasi nonkonduksi. Slab ditunjukkan Gambar 8.2. Untuk penyelesaian numeris slab dibagi kedalam  $N$  bagian dengan  $N + 1$  titik ujung. Slab mula-mula memiliki temperatur seragam  $100^\circ C$ . Hal ini memberikan kondisi awal bahwa semua temperatur di titik-titik internal diketahui nilainya pada saat  $t = 0$ .

$$T_n = 100 \text{ untuk } n = 2 \dots (N + 1) \text{ pada } t = 0 \quad (8.1.2)$$





**Gambar 8.2** Perpindahan Panas Konduksi *Unsteady State* pada Slab Satu Dimensi

Jika pada waktu nol permukaan luar tiba-tiba terjadi pada temperatur konstan  $0^\circ\text{C}$ , hal ini memberikan kondisi batas pada titik 1.

$$T_1 = 0 \quad \text{untuk } t \geq 0 \quad (8.1.3)$$

Kondisi batas lain adalah bahwa batas isolasi pada  $N + 1$  tidak ada konduksi panas. Sehingga :

$$\frac{\partial T_{N+1}}{\partial x} = 0 \quad \text{untuk } t \geq 0 \quad (8.1.4)$$

Sebagai catatan bahwa masalah ini sama dengan keadaan yang memiliki suatu slab dengan dua kali ketebalan terhadap temperatur awal kedua permukaan.

Ketika konveksi dianggap hanya sebagai model transfer panas ke permukaan slab, neraca energi dapat dibuat di permukaan yang menghubungkan antara energi input dengan konveksi terhadap energi output dengan konduksi. Oleh karena itu kapanpun untuk transfer normal ke permukaan slab pada arah  $x$  adalah :

$$h(T_0 - T_1) = -k \frac{\partial T}{\partial x} \Big|_{x=0} \quad \text{untuk } t \geq 0 \quad (8.1.5)$$

dimana  $h$  adalah koefisien transfer panas konveksi dalam  $W/m^2.K$  dan  $T_o$  adalah temperatur embien.

- a) Selesaikan secara numeris persamaan (8.1.1) dengan kondisi batas awal dan akhir persamaan (8.1.2), (8.1.3) dan (8.1.4) untuk kasus dimana  $\alpha = 2 \times 10^{-5} m^2/s$  dan permukaan slab dijaga konstan pada  $T_i = 0 \text{ }^\circ C$ . Penyelesaian harus menggunakan *numerical method of lines* dengan  $N = 10$  bagian. Plot temperature  $T_2, T_3, T_4$ , and  $T_5$  sebagai fungsi waktu sampai 6000 s.
- b) Ulangi bagian (a) untuk kasus dimana panas konveksi muncul pada permukaan slab dengan  $h = 25 W/m^2.K$ , dan  $k = 10 W/m.K$ .

### Penyelesaian:

Untuk kasus (a) dengan  $N = 10$  dan dengan bagian panjang  $\Delta x = 0,1 m$ , pers (8.1.1) dapat dituliskan menggunakan rumus *central difference* untuk turunan kedua sebagai:

$$\frac{\partial T_n}{\partial t} = \frac{\alpha}{(\Delta x)^2} (T_{n+1} - 2T_n + T_{n-1}) \quad \text{untuk } (2 \leq n \leq 10) \quad (8.1.6)$$

Kondisi batas yang dinyatakan pers (8.1.4) dapat dituliskan menggunakan *backward finite difference* order dua sebagai :

$$\frac{\partial T_{11}}{\partial t} = \frac{3T_{11} - 4T_{10} + T_9}{2(\Delta x)} = 0 \quad (8.1.7)$$

yang dapat diselesaikan untuk  $T_{11}$  menghasilkan :

$$T_{11} = \frac{4T_{10} - T_9}{3} = 0 \quad (8.1.8)$$

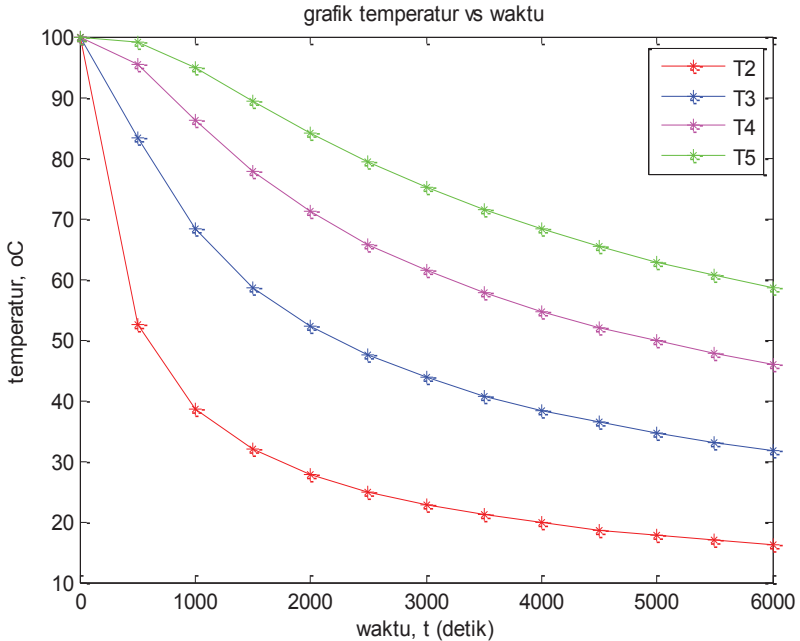
Penyelesaian kasus di atas dengan proram MATLAB disusun dan dijalankan sebagai berikut:

```
function latihan_81A
clear, clc, format short g, format compact
tspan=[0:500:6000]; % Range independent variable
Y0=[100 100 100 100 100 100 100 100 100]; % nilai awal T2-T10
[t,Y]=ode45(@ODEfun,tspan,Y0)
```

```

T2=Y(:,1);
T3=Y(:,2);
T4=Y(:,3);
T5=Y(:,4);
T6=Y(:,5);
T7=Y(:,6);
T8=Y(:,7);
T9=Y(:,8);
T10=Y(:,9);
plot(tspan,T2,'-r',tspan,T3,'-b',tspan,T4,'-m',tspan,T5,'-
*g')
title('grafik temperatur vs waktu');
xlabel('waktu, t (detik)');
ylabel('temperatur, oC ');
legend('T2','T3','T4','T5')
%- - - - -
function dYdt=ODEfun(t,Y);
T2=Y(1);
T3=Y(2);
T4=Y(3);
T5=Y(4);
T6=Y(5);
T7=Y(6);
T8=Y(7);
T9=Y(8);
T10=Y(9);
alpha = .00002;
deltax = .1;
T1= 0;
T11=(4.*T10 - T9)./ 3;
dYdt(1) = alpha / (deltax ^ 2) * (T3 - (2 * T2) + T1);
dYdt(2) = alpha / (deltax ^ 2) * (T4 - (2 * T3) + T2);
dYdt(3) = alpha / (deltax ^ 2) * (T5 - (2 * T4) + T3);
dYdt(4) = alpha / (deltax ^ 2) * (T6 - (2 * T5) + T4);
dYdt(5) = alpha / (deltax ^ 2) * (T7 - (2 * T6) + T5);
dYdt(6) = alpha / (deltax ^ 2) * (T8 - (2 * T7) + T6);
dYdt(7) = alpha / (deltax ^ 2) * (T9 - (2 * T8) + T7);
dYdt(8) = alpha / (deltax ^ 2) * (T10 - (2 * T9) + T8);
dYdt(9) = alpha / (deltax ^ 2) * (T11 - (2 * T10) + T9);
dYdt=dYdt';

```



**Gambar 8.3** Profil Temperatur Konduksi Panas *Unsteady State* pada Slab Satu Dimensi

Untuk kasus (b), neraca energi pada permukaan slab di atas dapat digunakan untuk menentukan hubungan antara temperatur permukaan slab  $T_1$ , temperatur ambien  $T_0$ , dan temperatur pada titik-titik internal. Dalam kasus ini persamaan *second-order forward difference* untuk turunan pertama dapat diterapkan pada permukaan :

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} = \frac{(-T_3 + 4T_2 - 3T_1)}{\Delta x} \quad (8.1.9)$$

dan disubstitusikan ke pers (8.1.5) menghasilkan :

$$h(T_0 - T_1) = -k \frac{(-T_3 + 4T_2 - 3T_1)}{2\Delta x} \quad (8.1.10)$$

Persamaan terdahulu dapat diselesaikan untuk memperoleh  $T_1$  :

$$T_1 = \frac{2hT_0\Delta x - kT_3 + 4kT_2}{3k + 2h\Delta x} \quad (8.1.11)$$

Penyelesaian kasus di atas dengan proram MATLAB disusun dan dijalankan sebagai berikut:

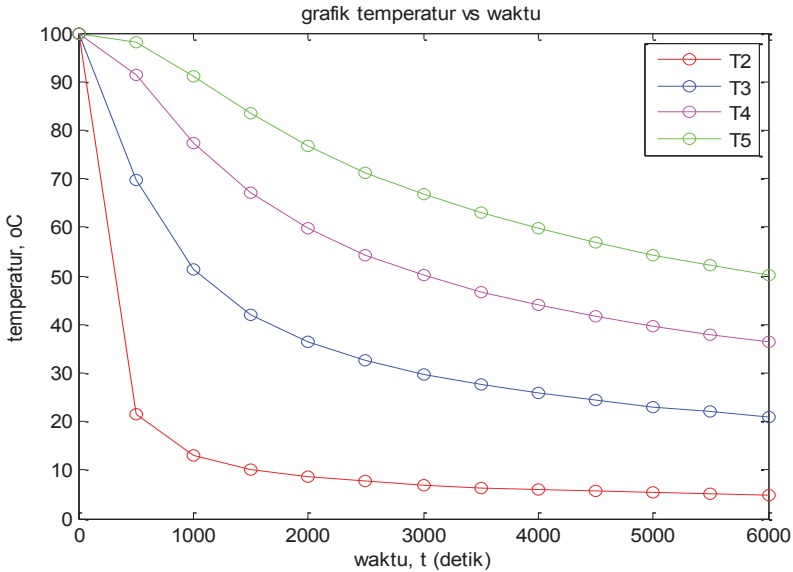
```
function latihan_81B
clear, clc, format short g, format compact
tspan=[0:500:6000]; % Range for the independent variable
Y0=[100 100 100 100 100 100 100 100 100]; % nilai awal T2-T10
[t Y] = ode45(@ODEfun,tspan,Y0)
T2 = Y(:,1);
T3 = Y(:,2);
T4 = Y(:,3);
T5 = Y(:,4);
T6 = Y(:,5);
T7 = Y(:,6);
T8 = Y(:,7);
T9 = Y(:,8);
T10 = Y(:,9);
plot(tspan,T2,'-or',tspan,T3,'-ob',tspan,T4,'-om',tspan,T5,'-og')
title('grafik temperatur vs waktu');
xlabel('waktu, t (detik)');
ylabel('temperatur, oC ');
legend('T2', 'T3', 'T4', 'T5')
%-----

function dYdt=ODEfun(t,Y);
T2 = Y(1);
T3 = Y(2);
T4 = Y(3);
T5 = Y(4);
T6 = Y(5);
T7 = Y(6);
T8 = Y(7);
T9 = Y(8);
T10 = Y(9);
alpha = .00002;
T0 = 0;
h = 25;%W/m2.K
k = 10;% W/m.K
deltax = .1;
T1 = (2*h*T0*deltax-(k*T3+4*k*T2))/(3*k+2*h*deltax);
T11 = (4*T10 - T9)/ 3;
dYdt(1) = alpha / (deltax ^ 2) * (T3 - (2 * T2) + T1);
dYdt(2) = alpha / (deltax ^ 2) * (T4 - (2 * T3) + T2);
dYdt(3) = alpha / (deltax ^ 2) * (T5 - (2 * T4) + T3);
dYdt(4) = alpha / (deltax ^ 2) * (T6 - (2 * T5) + T4);
```

```

dYdt (5) = alpha / (deltax ^ 2) * (T7 - (2 * T6) + T5);
dYdt (6) = alpha / (deltax ^ 2) * (T8 - (2 * T7) + T6);
dYdt (7) = alpha / (deltax ^ 2) * (T9 - (2 * T8) + T7);
dYdt (8) = alpha / (deltax ^ 2) * (T10 - (2 * T9) + T8);
dYdt (9) = alpha / (deltax ^ 2) * (T11 - (2 * T10) + T9);
dYdt=dYdt';

```



**Gambar 8.4** Profil Temperatur Transfer Panas *Unsteady State* dengan Konveksi dalam Slab Satu Dimensi

### Contoh 8.2 Difusi Massa *Unsteady state* dan Reaksi pada Slab

Gas  $\text{CO}_2$  pada tekanan satu atmosfer sedang diabsorpsi ke dalam larutan alkali yang mengandung katalis. Gas  $\text{CO}_2$  yang terlarut dinyatakan sebagai reaktan  $A$  yang mengalami difusi dan reaksi order satu, satu arah dalam larutan yang dinyatakan sebagai persamaan diferensial sebagai berikut.

$$\frac{\partial C_A}{\partial t} = D_{AB} \frac{\partial^2 C_A}{\partial x^2} - k C_A \quad (8.2.1)$$

dimana  $C_A$  adalah konsentrasi  $\text{CO}_2$  terlarut ( $\text{kgmol}/\text{m}^3$ ),  $t$  adalah

waktu (detik),  $D_{AB}$  adalah difusivitas  $\text{CO}_2$  dalam larutan alkali B ( $\text{m}^2/\text{s}$ ),  $x$  adalah jarak dari atas larutan ( $\text{m}$ ), dan  $k$  adalah konstanta laju reaksi order satu ( $1/\text{s}$ ). Tekanan parsial  $\text{CO}_2$  adalah  $P_{A0} = 1,0132 \times 10^5 \text{ Pa}$ , dan kelarutan  $\text{CO}_2$  adalah  $S = 2,961 \times 10^{-7} \text{ kg.mol/Pa}$ .

Kondisi awal untuk persamaan (8.2.1) diberikan dengan :

$$C_A = 0 \text{ pada awal waktu } t = 0 \text{ untuk semua } x \quad (8.2.2)$$

dan kondisi batas adalah

$$C_A = C_{As} \text{ untuk } t > 0 \text{ dan } x = 0 \quad (8.2.3)$$

dan

$$C_A = 0 \text{ untuk } t > 0 \text{ dan } x = \infty \quad (8.2.4)$$

Pada sistem tersebut, Geankoplis (1993) menyediakan data berikut:

$$\begin{aligned} C_{As} &= P_{A0}S = (1,0132 \times 10^5 \text{ Pa})(2,961 \times 10^{-7} \text{ kg.mol/Pa.}) \\ &= 0,03 \text{ kg.mol/m}^3 \end{aligned}$$

$$D_{AB} = 1,5 \times 10^{-9} \text{ m}^2/\text{s} \text{ dan } k = 35 \text{ s}^{-1}$$

- a) Hitung konsentrasi A terlarut dalam larutan setelah 0,01 detik dengan menggunakan *numerical methode of line* dengan 11 titik (10 interval dan interval antar node  $1 \times 10^{-6}$ ).
- b) Hitung jumlah total A yang terserap sampai waktu 0,01 detik.

### Penyelesaian :

Penyelesaian persoalan di atas dapat dilakukan dengan cara yang sama seperti Contoh 8.1. Untuk 10 interval permukaan slab dapat dituliskan :

$$\frac{\partial C_A}{\partial t} = \frac{D_{AB}}{(\Delta x)^2} (C_{A_{n+1}} - 2C_{A_n} + C_{A_{n-1}}) - kC_{A_n} \quad \text{untuk } (2 \leq n \leq 10) \quad (8.2.5)$$

Kondisi awal untuk persamaan (8.2.2) dan (8.2.3) dapat dituliskan :

$$C_{A1} = 0 \text{ pada } t = 0 \quad (8.2.6)$$

dan

$$C_{A1} = P_{A0}S = (1,0132 \times 10^5 \text{ Pa})(2,961 \times 10^{-7} \text{ kg.mol/Pa.}) = 0,03 \text{ kg.mol/m}^3 \quad \text{untuk } t > 0 \quad (8.2.7)$$

kondisi batas pada pers (8.2.4) dengan asumsi kondisi (a) menjadi:

$$C_{A11} = 0 \text{ untuk } t \geq 0 \quad (8.2.8)$$

Penyelesaian kasus di atas dengan proram MATLAB disusun dan dijalankan seperti pada program `function_latihan_82A`:

```
function latihan_82A
clear, clc, format short g, format compact
tspan=[0:0.0005:0.01]; % Range for the independent variable
Y0=[0 0 0 0 0 0 0 0 0]; % Nilai awal CA2 - CA10
[t Y] = ode45(@ODEfun,tspan,Y0)
CA2 = Y(:,1);
CA3 = Y(:,2);
CA4 = Y(:,3);
CA5 = Y(:,4);
CA6 = Y(:,5);
CA7 = Y(:,6);
CA8 = Y(:,7);
CA9 = Y(:,8);
CA10 = Y(:,9);
plot(tspan,CA2,'-r',tspan,CA3,'-b',tspan,CA4,'-m',tspan,CA5,'-g')
title('grafik konsentrasi A vs waktu');
xlabel('waktu, t (detik)');
ylabel('konsentrasi CA, kgm01/m3 ');
legend('CA2','CA3','CA4','CA5')
%-----
function dCAdt=ODEfun(t,Y);
CA2 = Y(1);
CA3 = Y(2);
CA4 = Y(3);
CA5 = Y(4);
CA6 = Y(5);
CA7 = Y(6);
CA8 = Y(7);
CA9 = Y(8);
CA10 = Y(9);
DAB = .0000000015;
k = 35;
```

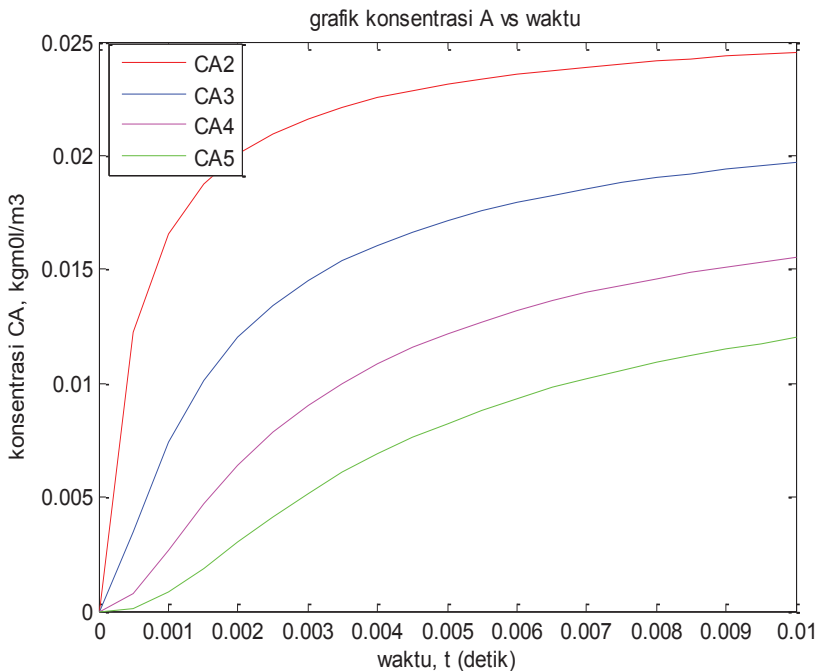


```

if (t == 0)
    CA1 = 0;
else
    CA1 = .03;
end
deltax = .000001;
CA11 = 0;
dCAdt (1) = DAB / (deltax ^ 2) * (CA3 - (2 * CA2) + CA1) - k*CA2;
dCAdt (2) = DAB / (deltax ^ 2) * (CA4 - (2 * CA3) + CA2) - k*CA3;
dCAdt (3) = DAB / (deltax ^ 2) * (CA5 - (2 * CA4) + CA3) - k*CA4;
dCAdt (4) = DAB / (deltax ^ 2) * (CA6 - (2 * CA5) + CA4) - k*CA5;
dCAdt (5) = DAB / (deltax ^ 2) * (CA7 - (2 * CA6) + CA5) - k*CA6;
dCAdt (6) = DAB / (deltax ^ 2) * (CA8 - (2 * CA7) + CA6) - k*CA7;
dCAdt (7) = DAB / (deltax ^ 2) * (CA9 - (2 * CA8) + CA7) - k*CA8;
dCAdt (8) = DAB / (deltax ^ 2) * (CA10 - (2 * CA9) + CA8) - k*CA9;
dCAdt (9) = DAB / (deltax ^ 2) * (CA11 - (2 * CA10) + CA9) - k*CA10;
dCAdt=dCAdt';

```

Grafik profil konsentrasi terhadap waktu ditunjukkan gambar berikut :



**Gambar 8.5** Profil difusi dan reaksi *unsteady state* dalam slab satu dimensi

## 8.3 RANGKUMAN

- a. Persamaan Diferensial Parsial (PDP) merupakan persamaan diferensial dimana fungsi yang tidak diketahui (variabel terikat) adalah fungsi dari lebih dari satu variabel bebas.
- b. Penyelesaian persamaan diferensial parsial dapat diselesaikan *dengan metode of line* dan MATLAB menggunakan fungsi ode yang akan diperoleh nilai variabel terikat untuk setiap variabel bebas yang diberikan.

## 8.4 RUJUKAN

- a. Constantinidis dan Mustoufi.1999. *Numerical Methodes for Chemical Engineers with MATLAB Application*. hal 262-264. Prentice-Hall: Englewood Cliffs. NJ.
- b. Finlayson, B.A. 2006. *Introduction to Chemical Engineering Computing*. hal 115-123. John Wiley & Sons Inc. New Jersey
- c. Bird, R. B., Stewart, W. E., Lightfoot, E. N., *Transport Phenomena*, 2nd ed. Wiley: New York, 2002
- d. Geankoplis, C.J., 1993. *Transport Process and Unit Operation*. Prentice-Gill. New Jersey.

# OPTIMASI

## 9.1 PENDAHULUAN

Optimasi dapat diartikan sebagai suatu proses untuk mencari kondisi yang optimum, dalam arti paling menguntungkan. Optimasi bisa berupa maksimasi atau minimasi. Bila kita berhadapan dengan masalah keuntungan, keadaan optimum adalah yang memberikan keuntungan maksimum (maksimasi) sedang bila berhadapan dengan masalah pengeluaran pengorbanan keadaan optimum adalah yang memberikan pengeluaran/pengorbanan minimum (minimasi).

Optimasi mencakup kegiatan yang dijalankan secara inovatif dan matematis. Optimasi inovatif bersifat strategis dalam arti mencari kondisi optimum dengan cara mencari proses-proses baru atau cara-cara baru atau flow diagram baru dan lain-lain, misalnya dengan mengubah proses ekstraksi dari cross current menjadi current current, atau mengubah cara pengambilan minyak atsiri dari stem distillation menjadi ekstraksi. Sebaliknya optimasi matematis bersifat taktis dalam arti prosesnya/caranya/flow diagramnya tetap sedang yang dicari adalah kondisi-kondisi operasi yang memberikan keadaan optimum secara matematis.

Dalam pelaksanaan optimasi, seringkali perhitungan alat-alat dilakukan dengan cara kasar. Setelah kondisi optimum diketahui baru dilakukan perhitungan dengan alat-alat dijalankan dengan teliti, hitungan optimasinya menjadi sangat besar.

Dalam kalkulus tersedia cara optimasi untuk satu perubah bebas maupun untuk perubah banyak. Yang diperlukan adalah turunan diferensialnya.

Untuk optimasi dengan satu variabel

$$y = f(x) \quad (9.1)$$

Maka harga  $y$  maksimum atau minimum terjadi pada :

$$\frac{\partial y}{\partial x} = 0 \quad (9.2)$$

Dimana apabila  $\frac{\partial^2 y}{\partial x^2} < 0$  terjadi  $y$  maksimum sedang bila

$\frac{\partial^2 y}{\partial x^2} > 0$  terjadi  $y$  minimum.

Untuk optimasi dengan banyak perubah:

$$y = f(x_1, x_2, x_3, \dots, x_n) \quad (9.3)$$

Maka  $y$  maksimum atau minimum terjadi pada:

$$\frac{\partial y}{\partial x_1} = 0; \frac{\partial y}{\partial x_2} = 0; \dots; \frac{\partial y}{\partial x_n} = 0 \quad (9.4)$$

Ada lagi suatu cara khusus untuk optimasi, yaitu dengan cara *Lagrange*. Jika dijumpai maksimasi atau minimasi

$$y = f(x_1, x_2, x_3, \dots, x_n) \quad (9.5)$$

sedang misal antar variabel bebas ada hubungan matematis

$$g_1 = f(x_1, x_2, x_3, \dots, x_n) \quad (9.6)$$

$$g_2 = f(x_1, x_2, x_3, \dots, x_n) \quad (9.7)$$

maka keadaan optimum terjadi pada:

$$\frac{\partial y}{\partial x_1} + \lambda \frac{\partial g_1}{\partial x_1} + \beta \frac{\partial g_2}{\partial x_1} = 0 \quad (9.8)$$

$$\frac{\partial y}{\partial x_2} + \lambda \frac{\partial g_1}{\partial x_2} + \beta \frac{\partial g_2}{\partial x_2} = 0 \quad (9.9)$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\frac{\partial y}{\partial x_n} + \lambda \frac{\partial g_1}{\partial x_n} + \beta \frac{\partial g_2}{\partial x_n} = 0 \quad (9.10)$$

dengan  $\lambda$  dan  $\beta$  berupa konstanta.

Dalam optimasi variabel yang akan dimaksimumkan atau diminimumkan ( $y$ ) disebut *objective function* sedang variabel  $x_1, x_2, \dots, x_n$  disebut variabel bebasnya atau *design variable*.

Meskipun optimasi dapat diselesaikan secara analitis, namun sebagian besar praktek optimasi memerlukan metode numeris, penyelesaian dengan komputer.

## 9.2 TOOLBOX OPTIMASI DALAM MATLAB

Toolbox optimasi harus diinstal dalam komputer sebagai tambahan MATLAB sebelum digunakan. Tabel 9.1 menunjukkan beberapa fungsi yang tersedia dalam toolbox optimasi. Sebagian besar routine optimasi ini memerlukan M-file (tersimpan dalam *current directory*) yang mengandung definisi persoalan yang akan diselesaikan, beberapa contoh file akan ditampilkan dan didiskusikan nanti. *Default* parameter optimasi digunakan secara luas, namun hal itu bisa dimodifikasi melalui suatu perintah 'options' yang tersedia dalam program. Sintaks perintah suatu fungsi optimasi secara umum dalam bentuk :

$$[x, \text{FunValue}] = \text{fminX}(\text{'ObjFun'}, \dots, \text{options})$$

Pernyataan di sisi kiri menyatakan kuantitas yang dijalankan oleh 'function'. Argumen output ini dijelaskan Table 9.2. Sisi kanan, fminX menyatakan salah satu fungsi yang diberikan Tabel 9.1. Dapat terdiri beberapa argumen untuk fungsi fminX, seperti nilai awal variabel, batas atas dan batas atas variabel, nama M-file yang mengandung fungsi dan gradien, lagoritma optimasi yang berkaitan dengan data, dan lainnya. Kegunaan fungsi ini dicontohkan dalam bagian berikutnya untuk beberapa tipe soal dan kondisi.

Tabel 9.1 *Toolbox* Fungsi Optimasi

Type of problem	Formulation	Fuction
One variable unconstrain minimization	Find $x \in [x_l, x_u]$ to minimize $f(x)$	fminbnd
Unconstrain minimization	Find $x$ to minimize $f(x)$	fminunc fminsearch
Constrain minimization	Find $x$ to minimize $f(x)$ Subject to $nx = e, Ax \leq b$ $h_j = 0, j = 1$ to $p$ $g_i = 0, i = 1$ to $m$ $x_{il} \leq x_i \leq x_{iu}$	fmincon
Linier Programming	Find $x$ to minimize $f(x)=c^T x$ Subject to $Nx = e, Ax \leq b$	linprog
Quadratic programming	Find $x$ to minimize $f(x)=c^T x + \frac{1}{2} x^T H x$ Subject to $Nx = e, Ax \leq b$	quadprog

Tabel 9.2 Argumen Fungsi Optimasi

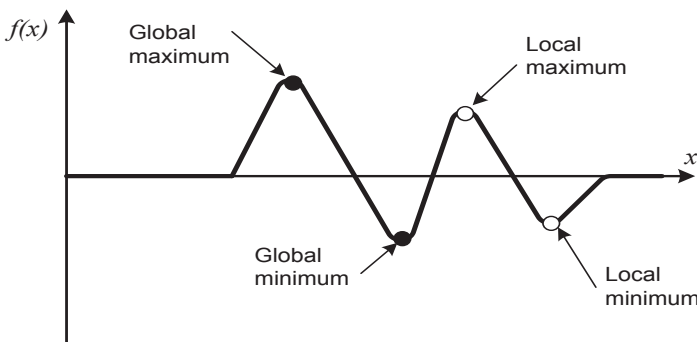
Argument	Description
x	The solution vector or matrix found by optimization function. If <code>exitFlag &gt; 0</code> then <code>x</code> is a solution, otherwise <code>x</code> is the latest value from optimization routine
FunValue	Value of the objective fundtion, <code>ObjFun</code> , at the solution <code>x</code>
Exitflag	The exit condition for optimization function. If <code>ExitFlag</code> is positive, then the optimization converged to a solution <code>x</code> . If <code>ExitFlag</code> is negative, the optimization routine did not converge to a solution
Output	The outputvector contain several pieces of information about the optimization process. It prvides the number of functions (output iteration) and the name of the algorithm used to solve the problem (output algorithm), etc.

## 9.3 OPTIMASI TANPA KONSTRAIN (UNCONSTRAINT OPTIMIZATION)

### 9.3.1 Optimasi Satu variabel

Pada bagian ini akan dijelaskan teknik untuk menemukan nilai maksimum atau minimum suatu fungsi variabel tunggal  $f(x)$ . Suatu gambar yang berguna dalam kaitan ini adalah “roller coaster” satu variabel seperti yang ditunjukkan Gambar 9.1. Sebagaimana lokasi akar persamaan yang rumit dengan fakta bahwa beberapa akar persamaan dapat terjadi untuk fungsi tunggal, hal yang sama untuk *local optimum* dan *global optimum* keduanya dapat terjadi dalam optimasi

*Global optimum* menyatakan penyelesaian yang sangat baik. *Local optimum*, meskipun tidak semua sangat baik, lebih baik daripada nilai terdekatnya (Chapra, 2012). Kasus yang termasuk dalam local optimum disebut *multimodal*. Dalam kasus seperti itu, kita hampir akan selalu tertarik dalam menemukan global optimum. Tambahan pula, kita harus difokuskan pada kesalahan hasil nilai lokal untuk global optimum.



**Gambar 9.1.** Suatu fungsi yang secara asimtot mendekati nol pada plus dan minus  $\infty$  dan memiliki dua titik maksimum dan minimum.

Pertama kita ilustrasikan penggunaan fungsi `fminbnd` untuk minimasi suatu fungsi variabel tunggal  $f(x)$  dengan batas pada  $x$  sebagai  $x_l \leq x \leq x_u$ .

### Contoh 9.1 Minimasi *unconstraint* dengan satu variabel

Tentukan nilai  $x$  untuk meminimalkan  $f(x) = 2 - 4x + e^x$ ,  $-10 \leq x \leq 10$ .

#### Penyelesaian.

Untuk menyelesaikan soal, kita tuliskan dalam m-file yang mencari nilai dari fungsi objektif.

```
function optml
clc, clear
Lb=-10;Ub=10;
[x,fval,exitflag] = fminbnd(@objfun1,Lb,Ub)
%-----
function f = objfun1(x)
f = 2-4*x+exp(x);
```

The output from the function is given as :

```
x =
    1.3863
fval =
    0.4548
exitflag =
     1
output =
  iterations: 21
  funcCount: 42
  algorithm: [1x33 char]
  message: [1x196 char]
```

### 9.3.2 Optimasi Multi variabel

Selain fungsi satu variabel, optimasi juga berlaku untuk fungsi multivariabel. Untuk kasus dua variabel ditunjukkan



Gambar 9.2. Contoh berikut menunjukkan kemampuan MATLAB untuk memvisualisasikan gambar fungsi dua variabel.

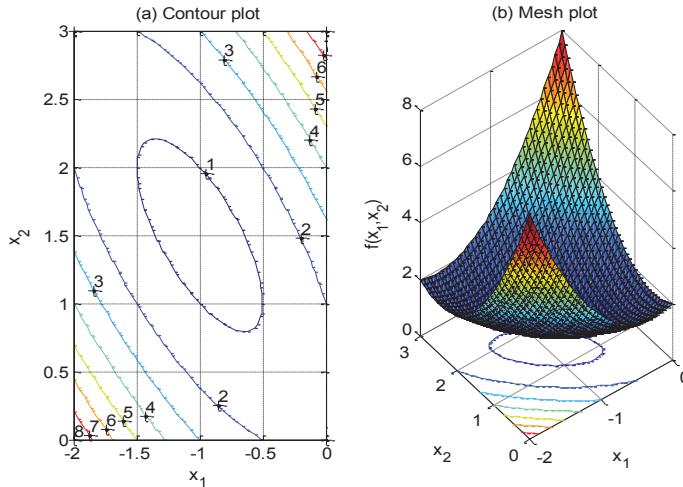
### Contoh 9.2 Minimasi *unconstraint* dengan dua variabel

Gunakan kemampuan grafik MATLAB untuk menampilkan fungsi berikut dan secara visual estimasi nilai minimumnya pada  $-2 \leq x_1 \leq 0$  and  $0 \leq x_2 \leq 3$ :

$$f(x_1, x_2) = 2 + x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

**Penyelesaian.** Berikut script untuk membuat plot *contour* dan *mesh* dari fungsi di atas.

```
x=linspace(-2,0,40);y=linspace(0,3,40);
[X,Y] = meshgrid(x,y);
Z=2+X-Y+2*X.^2+2*X.*Y+Y.^2;
subplot(1,2,1);
cs=contour(X,Y,Z);clabel(cs);
xlabel('x_1');ylabel('x_2');
title('(a) Contour plot');grid;
subplot(1,2,2);
cs=surfc(X,Y,Z);
zmin=floor(min(Z));
zmax=ceil(max(Z));
xlabel('x_1');ylabel('x_2');zlabel('f(x_1,x_2)');
title('(b) Mesh plot');
```



**Gambar 9.2.** Grafik fungsi dua variabel dengan (a) Contour dan (b) mesh

Seperti ditampilkan pada gambar di atas, kedua plot mengindikasikan bahwa fungsi memiliki nilai minimum  $f(x_1, x_2) = 0$  sampai 1 terletak pada sekitar  $x_1 = -1$  and  $x_2 = 1.5$ .

Teknik untuk optimasi multivariabel tanpa konstrain dapat diklasifikasikan dalam beberapa cara yang dibagi berdasarkan kebutuhannya untuk menggunakan penurunan. Pendekatan yang memerlukan penurunan disebut metode *gradient* atau *descent*. Sementara pendekatan yang tidak memerlukan evaluasi penurunan disebut *nongradient* atau *direct*.

MATLAB standar memiliki fungsi *fminsearch* dan *fminunc* yang dapat digunakan untuk menentukan nilai minimum fungsi multivariabel tanpa konstrain. Fungsi *fminsearch* menggunakan metode pencarian *Simplex Nelder-Mead*, yang merupakan metode *direct* yang hanya menggunakan nilai fungsi (tidak membutuhkan penurunan) atau tidak memerlukan nilai numeris atau gradien analisis fungsi obyektif.

Sementara itu, fungsi *fminunc* memerlukan informasi

dan dengan *option* 'LargeScale' diatur ke 'off', ini menggunakan metode *BGF quasi-newton* dengan prosedur pencarian kombinasi *quadratic* dan *cubik line*. Formula *DFP* yang mendekati invers matrik *Hessian* dapat dipilih dengan mengatur pilihan *Hessupdate* menjadi *steepdesc*. *Fminsearch* umumnya kurang efisien daripada *fminunc*. Namun demikian, ini bisa menjadi efektif untuk persoalan yang evaluasi gradiennya mahal atau tidak mungkin (Arora, 2012).

Untuk menyelesaikan persoalan, kita menuliskan suatu m-file yang mencari nilai fungsi obyektif. Selanjutnya, fungsi minimasi tanpa konstrain *fminsearch* atau *fminunc* dijalankan dengan mengeksekusi M-file lainnya.

Pernyataan sederhana sintaknya adalah :

```
[xmin, fval] = fminsearch(function,x0)
[xmin, fval] = fminunc(function,x0)
```

Dimana *xmin* dan *fval* adalah lokasi dan nilai minimum, *function* adalah nama fungsi yang dievaluasi, dan *x0* adalah tebakan awal. *x0* dapat berupa scalar, vector atau suatu matrik.

Berikut contoh sederhana penggunaan *fminsearch* dan *fminunc* untuk menentukan nilai minimum dari grafik yang ditunjukkan Gambar 9.2

```
function optmulti
clc, clear
[x, fval] = fminsearch(@objfun2, [-0.5, 0.5])
%-----
function f = objfun2(x)
f = 2+x(1)-x(2)+2*x(1)^2+2*x(1)*x(2)+x(2)^2;
```

Hasil eksekusi ditampilkan dibawah:

```
x =  
    -1.0000    1.5000  
  
fval =  
  
    0.7500  
  
exitflag =  
  
    1  
output =  
  
    iterations: 40  
    funcCount: 76  
    algorithm: [1x33 char]  
    message: [1x196 char]
```

Untuk script fminunc ditampilkan dibawah:

```
function optmulti2  
clc, clear  
[x,fval,exitflag] = fminunc(@objfun2,[-  
0.5,0.5])  
%-----  
function f = objfun2(x)  
f = 2+x(1)-x(2)+2*x(1)^2+2*x(1)*x(2)+x(2)^2;
```

Nilai variabel ditampilkan dibawah:

```
x =  
    -1.0000    1.5000  
  
fval =  
  
    0.7500  
  
exitflag =  
  
    1
```

```

output =
  iterations: 3
  funcCount: 15
  stepsize: 1
  firstorderopt: 1.4901e-008
  algorithm: [1x38 char]
  message: [1x438 char]

```

## 9.4 OPTIMASI DENGAN KONSTRAIN (CONSTRAINT OPTIMIZATION)

Persoalan optimasi dengan konstrain umum diselesaikan dengan fungsi *fmincon* seperti didefinisikan pada tabel 9.1. Prosedur untuk menjalankan fungsi ini sama dengan soal tanpa konstrain tetapi pada fungsi ini harus disediakan M-file yang mengandung fungsi konstrain. Jika ekspresi gradien analitis diprogram dalam fungsi objektif dan M-file fungsi konstrain, maka fungsi ini dinyatakan melalui perintah *option*, jika tidak *fmincon* akan menggunakan perhitungan gradien numeris berdasarkan metode *finite difference*.

*fmincon* menyelesaikan persoalan minimasi dalam bentuk :

$$\min_x f(x) \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

$x$ ,  $b$ ,  $beq$ ,  $lb$ , dan  $ub$  adalah vektor,  $A$  dan  $Aeq$  adalah matriks,  $c(x)$  dan  $ceq(x)$  adalah fungsi yang menjalankan vektor dan  $f(x)$  adalah fungsi yang menjalankan skalar.  $f(x)$ ,  $c(x)$ , dan  $ceq(x)$  dapat berupa fungsi nonlinier.

*fmincon* menerapkan empat algoritma yang berbeda, yaitu :

*interior point*, *SQP*, *activeset*, dan *trust region reflective*. Memilih salah satunya melalui option Algorithm: sebagai contoh memilih SQP, atur options = optimset('Algorithm','sqp'), dan selanjutnya masukan option ke fmincon.

Penulisan syntax :

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
```

Deskripsi :

f	Fungsi yang diminimasi. fun is adalah fungsi yang menerima vektor x dan skalar, f adalah fungsi objektif yang dievaluasi pada x.
x0	Nilai awal x
A	<i>matrix for linear inequality constraints</i>
b	<i>vector for linear inequality constraints</i>
Aeq	<i>matrix for linear equality constraints</i>
beq	<i>vector for linear equality constraints</i>
c(x)	<i>nonlinear inequalities constrains</i>
ceq(x)	<i>nonlinear equalities constrains</i>
lb	<i>vector of lower bounds</i>
ub	<i>vector of upper bounds</i>
nonlcon	<i>nonlinear constraint function</i>

Untuk memperjelas penggunaan fungsi ini, diberikan contoh berikut:

### Contoh 9.3 Optimasi dengan *linier inequality constraint*

Selesaikan soal minimasi berikut:

$$\min_x f(x) = -x_1 x_2 x_3 \quad (9.3.1)$$

Terhadap konstrain:

$$0 \leq x_1 + 2x_2 + 2x_3 \leq 72 \quad (9.3.2)$$

dengan nilai tebakan awal  $x$  adalah (10; 10; 10)

### Penyelesaian:

Pada soal di atas konstrain hanya berupa *linier inequality*, sementara konstrain lain tidak ada. Untuk menentukan nilai minimasi pada fungsi di atas dapat digunakan syntax yang paling sederhana.

Penulisan script:

```
function optconlin1
clc, clear
x0=[10 10 10]; %nilai tebakan awal x(1) dan x(2)
A=[-1 -2 -2;...
    1 2 2] %matriks linier inequality constrain
b=[0;72] %vektor linier inequality constrain
%file eksekusi optimasi
[x,fval] = fmincon(@objfun1,x0,A,b)
%-----
function f = objfun1(x)
%objective function
f = -x(1)*x(2)*x(3)
```

Eksekusi program menghasilkan output:

```
lower      upper      ineqlin      ineqnonlin
          2
x =
    24.0000    12.0000    12.0000
fval =
   -3.4560e+003
```

### Contoh 9.4 Optimasi dengan *nonlinier inequality constraint* dan *bound constraint*

Selesaikan soal minimasi berikut:

$$\min_x f(x) = e^{x_1} (4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1) \quad (9.4.1)$$

Terhadap konstrain:

$$x_1x_2 - x_1 - x_2 \leq -1,5 \quad (9.4.2)$$

$$x_1x_2 \geq -10 \quad (9.4.3)$$

$$0 \leq x_1 \leq 0 \quad (9.4.4)$$

#### Penyelesaian:

Karena kedua konstrainnya adalah nonlinier, kita tidak dapat melewati konstrain pada *fmincon* dalam satu file perintah. Sehingga kita dapat membuat file kedua, *confun.m* yang mencari nilai kedua konstrain pada *x* saat ini dalam vektor *c*. Optimasi konstrain *fmincon* selanjutnya dieksekusi. Karena *fmincon* meminta konstrain ditulis dalam bentuk  $c(x) \leq 0$ , konstrain harus ditulis dalam bentuk :

$$x_1x_2 - x_1 - x_2 + 1,5 \leq 0 \quad (9.4.5)$$

$$-x_1x_2 - 10 \leq 0 \quad (9.4.6)$$

Penulisan syntax M-file untuk soal di atas adalah :

```
function optimcon1
clc, clear
x0 = [-1 1]; %nilai tebakan awal x(1) dan x(2)
A=[];b=[]; %linier inequality constrain
Aeq=[];beq=[]; %linier equality constraints
lb=[0 0];ub=[]; %lower dan upper bound
options = optimset('Algorithm','active-
set'); %memilih algoritma
[x,fval]=fmincon(@objfun,x0,A,b,Aeq,beq,lb,u
b,@confun,options)
%-----
function [c, ceq] = confun(x)
```



```

% Nonlinear inequality constraints
c = [1.5 + x(1)*x(2) - x(1) - x(2);
     -x(1)*x(2) - 10];
% Nonlinear equality constraints
ceq = [];

%-----
function f = objfun(x)
%objective function
f = exp(x(1))*(4*x(1)^2 + 2*x(2)^2 +
4*x(1)*x(2) + 2*x(2)+1);

```

Ouput program adalah :

```

Active      inequalities      (to      within
options.TolCon = 1e-006):
  lower      upper      ineqlin   ineqnonlin

         1         1

x =
0    1.5000

fval =
      8.5000

```

**Contoh 9.5 Optimasi dengan *nonlinier inequality constraint, bound constraint* dan *gradients***

Selesaikan soal minimasi berikut:

$$\text{Minimasi } f(x) = (x_1 - 10)^3 + (x_2 - 20)^3 \tag{9.5.1}$$

Terhadap konstrain:

$$g_1(x) = 100 - (x_1 - 5)^3 - (x_2 - 5)^3 \leq 0 \tag{9.5.2}$$

$$g_2(x) = 100 - (x_1 - 5)^3 - (x_2 - 5)^3 \leq 0 \tag{9.5.3}$$

$$13 \leq x_1 \leq 100, 0 \leq x_2 \leq 100 \tag{9.5.4}$$

### Penyelesaian:

Penyelesaian optimum untuk soal diketahui  $x = 14,095$ ;  $0,84296$ ) dan  $f(x) = -6961,9$  (Arora, 1981). M-file untuk soal ini diberikan di bawah:

```
function optconnlingrad
clc, clear
% Set bounds for variables
Lb = [13; 0]; Ub = [100; 100];
% Set initial design
x0 = [20.1 5.84];
% Invoke fmincon; four [ ] indicate no linear
    constraints in the problem
options = optimset('Algorithm','interior-
point',...

'Display','iter','GradObj','on','GradConstr','o
n')
[x, FunVal, ExitFlag, Output] = ...
fmincon(@ObjAndGrad9_5,x0,[],[],[],[],Lb,Ub,@Co
nstAndGrad9_5)
%-----
%M-File for Objective Function and Gradient
    Evaluations
function [f, gf] = ObjAndGrad9_5(x)
% Re-name design variables x
x1 = x(1); x2 = x(2);
% Evaluate objective function
f = (x1-10)^3 + (x2-20)^3;
% Compute gradient of objective function
if nargin > 1
gf(1,1) = 3*(x1-10)^2;
gf(2,1) = 3*(x2-20)^2;
end
%-----
%M-File Constraint Functions and Their
    Gradients Evaluation
function [g, h, gg, gh] = ConstAndGrad9_5(x)
%Re-name design variables
x1 = x(1); x2 = x(2);
```

```

% Inequality constraints
g(1) = 100-(x1-5)^2-(x2-5)^2;
g(2) = -82.81+ (x1-6)^2 + (x2-5)^2;
% Equality constraints (none)
h = [];
% Gradients of constraints
if nargout > 2
gg(1,1) = -2*(x1-5);
gg(2,1) = -2*(x2-5);
gg(1,2) = 2*(x1-6);
gg(2,2) = 2*(x2-5);
gh = [];
end

```

Tabel Iterasi Output MATLAB dan hasil optimasi:

Iter	F-count	f(x)	Feasibility	First-order optimality	Norm of step
0	1	-1.808858e+003	1.167e+002	5.850e+002	
1	2	-6.941059e+003	1.401e+002	1.123e+003	5.811e+000
2	3	-7.105234e+003	1.281e+002	1.122e+003	4.446e-001
3	4	-7.575301e+003	7.439e+001	1.580e+002	2.137e+000
4	5	-7.921643e+003	1.051e+001	4.645e+001	3.233e+000
5	6	-7.949393e+003	1.218e+000	2.373e+001	5.829e-001
6	7	-7.950558e+003	8.713e-001	6.663e-001	2.270e-002
7	8	-7.950670e+003	9.018e-001	1.200e-001	1.894e-003
8	9	-7.950677e+003	9.001e-001	1.200e-001	1.104e-004
9	10	-7.950747e+003	8.748e-001	1.197e-001	1.654e-003
10	11	-7.950680e+003	8.693e-001	1.196e-001	3.268e-004
11	12	-7.877851e+003	8.050e-001	7.413e+000	6.894e-002
12	14	-7.508476e+003	5.784e-001	4.541e+001	3.543e-001
13	15	-7.074594e+003	2.500e-001	4.313e+001	4.194e-001
14	16	-6.984897e+003	2.595e-002	2.608e+000	8.441e-002
15	17	-6.962111e+003	7.145e-004	4.051e-001	2.213e-002
16	18	-6.961812e+003	1.278e-006	5.011e-003	2.828e-004
17	19	-6.961814e+003	3.422e-009	7.986e-005	1.442e-006

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in feasible directions, to within the default value of the function tolerance, and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

x =  
14.0950 0.8430

FunVal =  
-6.9618e+003

ExitFlag =

1

Output =

iterations: 17  
funcCount: 19  
constrviolation: [1x1 double]  
stepsize: [1x1 double]  
algorithm: [1x14 char]  
firstorderopt: [1x1 double]  
cgiterations: 0  
message: [1x782 char]

### Interpretasi hasil optimasi :

Berikut penjelasan dari tabel yang ditampilkan dari hasil optimasi:

- Kolom Pertama, kata *Iter*, adalah jumlah iterasi dari 0 sampai 9, *Fmincon* melakukan 17 iterasi untuk konvergen.
- Kolom kedua, kata *F-count*, melaporkan jumlah waktu kumulatif dari fungsi yang dievaluasi. Baris akhir menunjukkan *F-count* 19, menandakan bahwa *fmincon* mengevaluasi fungsi 19 kali dalam proses

untuk menemukan nilai minimum.

- Kolom ketiga, tanda  $f(x)$ , menampilkan nilai *objective function*. Nilai final,  $-6.961814e+003$  adalah nilai minimum yang dilaporkan dalam optimasi dan dimunculkan pada pesan yang keluar dalam *command window*.
- Kolom keempat, *Feasibility* menjelaskan bahwa algoritma optimasi menemukan nilai parameter yang sesuai dengan semua konstrain terhadap toleransi tertentu namun tidak meminimalkan fungsi obyektif.
- Kolom kelima, *First-order optimality* adalah ukuran seberapa dekat suatu titik  $x$  terhadap nilai optimal.
- Kolom keenam, *Norm of step*, menjelaskan ukuran langkah saat ini (*size is the Euclidean norm, or 2-norm*)

Pesan di bawah kolom menjelaskan bahwa pencarian untuk nilai optimum konstrain selesai karena turunan fungsi obyektif mendekati nol pada arah yang diizinkan oleh konstrain, dan konstrain tersebut sangat sesuai.

## 9.5 PEMROGRAMAN LINIER (*LINIER PROGRAMMING*)

Persoalan optimasi yang memiliki fungsi obyektif linier dan fungsi konstrain linier disebut program linier (*Linier Programming, LP*).

Persoalan-persoalan dalam kelompok pemrograman linier adalah yang terpenting karena tiga alasan yang berbeda :

- 1) Ini adalah salah satu ilustrasi kajian yang terpenting bagaimana ide dan algoritma berubah dari analisis klasik atau perhitungan manual menjadi program komputer modern

- 2) Pemrograman linier mencakup persoalan yang luas.
- 3) Banyak algoritma diterapkan terhadap persoalan non-linier yang mengeksploitasi penyelesaian perograman linier.

Beberapa aplikasi metode program linier adalah untuk (1) penugasan, (2) pencampuran, (3) distribusi, (4) penentuan jaringan aliran, (5) penjadwalan, (6) transportasi dan (7) jadwal perjalanan penjualan.

Aplikasi pada industri pertama kali dari program linier dibuat dalam fraksionasi minyak. Secara umum, fraksionasi minyak memiliki pilihan pembelian minyak mentah dari beberapa sumber yang berbeda dengan komposisi yang berbeda dan pada harga yang berbeda. Hal ini dapat membuat produk yang berbeda, seperti bahan bakar pesawat terbang, bahan bakar diesel, gasoline dalam jumlah yang bervariasi. Kemungkinan kendala muncul karena keterbatasan pada jumlah minyak mentah yang tersedia dari sumber utama, kapasitas fraksionasi untuk menghasilkan produk tertentu dan seterusnya. Kombinasi antara minyak mentah yang dibeli dan produk yang dihasilkan dicari yang menghasilkan keuntungan maksimum.

Masalah umum program linier dapat dinyatakan dalam bentuk standar berikut :

**Bentuk Scalar :**

$$\text{Minimasi } f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (9.4.1) \quad \mathbf{(9.11)}$$

Terhadap konstrain :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \quad (9.4.1) \quad \mathbf{(9.12)}$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \quad (9.4.1) \quad \mathbf{(9.13)}$$

$$\vdots$$

$$\begin{aligned}
 a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \\
 x_1 &\geq 0 \\
 x_2 &\geq 0 \\
 &\vdots \\
 x_n &\geq 0
 \end{aligned}
 \tag{9.4.1} \quad \mathbf{(9.14)}$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \tag{9.4.1}$$

Dimana  $c_j$ ,  $b_j$ , dan  $a_{ij}$  ( $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ ) adalah konstanta yang diketahui, dan  $x_j$  adalah variabel desain.

**Bentuk Matriks :**

$$\text{Minimasi } f(\mathbf{X}) = \mathbf{c}^T \mathbf{X} \tag{9.4.1} \quad \mathbf{(9.15)}$$

Terhadap konstrain

$$\begin{aligned}
 \mathbf{aX} &= \mathbf{b} \\
 \mathbf{X} &\geq 0
 \end{aligned}
 \tag{9.4.1} \quad \mathbf{(9.16)}$$

dimana :

$$\mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}, \quad \mathbf{b} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{Bmatrix}, \quad \mathbf{c} = \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{Bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Karakteristik persoalan program linier yang dinyatakan dalam bentuk standar adalah ;

1. Fungsi obyektif adalah **tipe minimasi**
2. Semua kontrain adalah tipe persamaan

3. Semua variabel bebas adalah non negatif

Persoalan optimasi program linier diselesaikan dengan MATLAB menggunakan fungsi *linprog*.

*linprog* menyelesaikan persoalan minimasi dalam bentuk :

$$\min_x f^T(x) \begin{cases} A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

*f*, *x*, *b*, *beq*, *lb*, dan *ub* adalah vektor, *A* dan *Aeq* adalah matriks.

Syntax program :

```
x = linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
```

Deskripsi :

<i>f</i>	Fungsi yang diminimasi. <i>f</i> un is adalah fungsi yang menerima vektor <i>x</i> dan skalar, <i>f</i> adalah fungsi objektif yang dievaluasi pada <i>x</i> .
<i>x0</i>	Nilai awal <i>x</i>
<i>A</i>	<i>matrix for linear inequality constraints</i>
<i>b</i>	<i>vector for linear inequality constraints</i>
<i>Aeq</i>	<i>matrix for linear equality constraints</i>
<i>beq</i>	<i>vector for linear equality constraints</i>
<i>lb</i>	<i>vector of lower bounds</i>
<i>ub</i>	<i>vector of upper bounds</i>
<i>options</i>	struktur option dibuat dengan <i>optimset</i>

Untuk memperjelas penggunaan fungsi ini, diberikan contoh berikut:



**Contoh 9.6.** Tentukan nilai minimum untuk persoalan program linier berikut dengan MATLAB.

$$\text{Minimasi } f = -x_1 - 2x_2 - x_3$$

Terhadap konstrain :

$$2x_1 + x_2 - x_3 \leq 2$$

$$2x_1 - x_2 + 5x_3 \leq 6$$

$$4x_1 + x_2 + x_3 \leq 6$$

$$x_3 \leq 0; \quad i = 1, 2, 3$$

**Penyelesaian :**

**Pertama,** ekspresi fungsi objektif dalam bentuk  $f^T x$  dan identifikasi vektor  $x$  dan  $f$  sebagai :

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \quad \text{dan} \quad f = \begin{Bmatrix} -1 \\ -2 \\ -1 \end{Bmatrix}$$

Eksperi konstrain dalam bentuk  $Ax \leq b$  dan identifikasi matriks  $A$  dalam bentuk vektor  $b$  sebagai :

$$c = \begin{bmatrix} 2 & 1 & -1 \\ 2 & -1 & 5 \\ 4 & 1 & 1 \end{bmatrix} \quad \text{dan} \quad b = \begin{Bmatrix} 2 \\ 6 \\ 6 \end{Bmatrix}$$

**Kedua,** Gunakan perintah untuk mengeksekusi program linier menggunakan metode *simplex* yang ditunjukkan script berikut:

```
function contoh94_1
clc
clear all
f=[-1;-2;-1];
A=[2 1 -1;2 -1 5;4 1 1];
b=[2;6;6];
lb=zeros(3,1);
Aeq=[];
beq=[];
```

```

options =
optimset('LargeScale','off','Simplex','on');
[x,fval,exitflag,output] =
linprog(f,A,b,Aeq,beq,lb,[],[],...
optimset('Display','iter'))

```

Hasil simulasi :

Residuals:	Primal	Dual	Duality	Total
	Infeas	Infeas	Gap	Rel
	A*x-b	A'*y+z-f	x'*z	Error
Iter 0:	1.03e+003	7.97e+000	1.50e+003	4.00e+002
Iter 1:	4.11e+002	2.22e-016	2.78e+002	4.72e+001
Iter 2:	1.16e-013	1.90e-015	2.85e+000	2.33e-001
Iter 3:	1.78e-015	1.80e-015	3.96e-002	3.96e-003
Iter 4:	7.48e-014	1.02e-015	1.99e-006	1.99e-007
Iter 5:	2.74e-014	9.76e-016	2.01e-012	1.98e-013

Optimization terminated.

x =

```

0.0000
4.0000
2.0000

```

fval =

```

-10.0000

```

exitflag =

```

1

```

output =

```

iterations: 5
algorithm: [1x27 char]
cgiterations: 0
message: [1x24 char]
constrviolation: 0
firstorderopt: 8.5746e-013

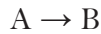
```

## 9.6 APLIKASI MATLAB DALAM OPTIMASI TEKNIK KIMIA

Pada bagian ini akan dibahas beberapa contoh penggunaan MATLAB untuk menyelesaikan persoalan dalam bidang teknik kimia.

### Contoh 9. 7. Optimasi Reaktor *Batch*

Reaktor *batch* beroperasi secara adiabatik untuk reaksi fasa cair order 2:



Dengan persamaan kecepatan reaksi :

$$r_A = k C_A^2 \quad (9.7.1)$$

Dengan

$$k = A \exp\left(-\frac{E}{RT}\right) \quad (9.7.2)$$

Perubahan entalpi reaksi ( $AH_R$ ), volum reaktor ( $V_R$ ) dan kapasitas panas larutan ( $C_p$ ) dianggap tetap. Waktu untuk bongkar pasang,  $t_p$ . Umpan masuk pada suhu  $T_f$ , dan konsentrasi  $A = C_{A0}$ . Reaksi dihentikan pada saat konversi  $X_R$ . Konversi A mula-mula  $X_{A0}$ . Ingin dicari konversi A yang memberikan kecepatan produksi B tiap waktu maksimum. Diketahui:  $V_R = 10.000$  L;  $C_{A0} = 1$  gmol/L;  $\rho = 1$  kg/L;  $C_p = 1,0$  kcal/kg/K;  $A = 1,1E-07$  L/gmol/menit;  $E = 14$  kcal/gmol;  $AH_R = -6$  kcal/gmol;  $R = 0,001987$  kcal/gmol/K ;  $T_f = 350$  K;  $t_p = 120$  menit;  $X_{R0} = 0$ .

### Penyelesaian :

Dari Neraca mol A diperoleh persamaan (Sediawan dan Prasetya, 1997):

$$t_R = \frac{1}{C_{A0}} \int_{x_{in}}^{x_{out}} \frac{1}{k(1-x_A)^2} dx \quad (9.7.3)$$

Dari neraca energi diperoleh persamaan :

$$T = T_f - \frac{C_{A0} \Delta H_R}{\rho C_p} x_A \quad (9.7.4)$$

Produksi tiap batch =

$$P = \frac{V_R C_{A0} x_R}{\text{siklus batch}} \quad (9.7.4)$$

Kecepatan produksi B adalah :

$$B = \frac{V_R C_{A0} x_R}{t_R + t_p} \quad (\text{maksimum}) \quad (9.7.4)$$

Maksimasi kecepatan produksi B = f (X<sub>R</sub>) dengan cara optimasi satu variabel.

```
function contoh9_7
clc,clear
global Cao
Cao = 1;          %gmol/L
xopt = fminunc(@optbatch,0.2,0.99);
B = latihan8(xopt);
tr = (1./Cao).*quad(@sub_optbatch,0,xopt);
disp('HASIL PERHITUNGAN ')
disp('-----')
disp(['x optimum      = ' num2str(xopt)])
disp(['B maksimum     = ' num2str(B), '
gmol/menit'])
```

```

disp(['tr opt          = ' num2str(tr), '
menit'])
disp('-----')
%-----
function B=optbatch(x)
global Cao
tp = 120;      %menit
xo = 0;       %konversi awal
x = 0.7;
Vr = 10000;   %L
tr = (1./Cao).*quad(@sub_optbatch,xo,x);
B = (Vr.*Cao.*x)./(tr+tp);
%-----
function y=sub_optbatch(x)
global Cao
Tf = 350;     %K
rho = 1;     %kg/L
Cp = 1;      %kcal/kg/K
A = 1e7;     %L/gmol/menit
delHr = -6;  %kcal/gmol
R = 0.001987; %kcal/gmol/K
E = 14;     %kcal/gmol

T = Tf - ((Cao.*delHr.*x)./(rho.*Cp));
k = A.*exp(-E./(R.*T));
y = 1./(k.*(1-x).^2);

```

HASIL PERHITUNGAN

```

-----
x optimum          = 0.2
B maksimum         = 30.4915 gmol/menit
tr opt            = 13.3254 menit
-----

```

### Contoh 9.8 Optimasi Reaksi Isotermis

Reaksi fasa cair bolak-balik eksotermis :



Dijalankan dalam reaktor *batch* adiabatik. Volume campuran dapat dianggap tetap. Suhu awal reaktor  $T_0$ . Kecepatan

reaksi mengikuti persamaan:

$$r_A = k \left( C_A - \frac{C_B}{K} \right) \quad (9.8.1)$$

Dengan :

$$k = A \exp \left( -\frac{E}{RT} \right) \quad (9.8.2)$$

$$K = \alpha \exp \left( -\frac{\beta}{T} \right) \quad (9.8.3)$$

Panas reaksi  $\lambda = -80.000$  kal/gmol, densitas campuran  $\rho = 1100$  g/L, kapasitas panas campuran  $C_p = 1,2$  kal/g/K, konsentrasi mula-mula  $C_{A0} = 1$  gmol/L, konstanta tumbukan  $A = 25$ , energi aktivasi  $E = 10.000$ , konstanta gas universal  $R = 1,987$ ; konstanta  $\alpha = 1750$  dan  $\beta = -5000$ . Koversi awal  $X_0 = 0$  dan konversi akhir yang diinginkan  $X_N = 0,3$ . Ingin dicari suhu awal yang memberikan waktu reaksi minimum.

### Penyelesaian :

Dari Neraca mol  $A$  diperoleh persamaan :

$$t = \int_{X_0}^{X_N} \frac{1}{k [(1 - x_A) - Kx]} dx \quad (9.8.4)$$

Dari neraca energi diperoleh persamaan :

$$T = T_0 - \frac{C_{A0} \Delta H_R}{\rho C_p} (x_A - x_0) \quad (9.8.5)$$

Nilai  $T_0$  dicari dengan meminimalkan  $t = f(T_0)$ , dan  $t$  dihitung dengan integrasi.

```

function soal98
    clc
    Tsol=fminunc(@latihan98,500);
    to=latihan10(Tsol);
    disp( 'HASIL PERHITUNGAN ')
    disp('-----')
    disp(['Suhu optimum,T = ',num2str(Tsol),'
    K'])
    disp(['Waktu optimum,t = ',num2str(to),'
    menit'])
    %-----
    function t=latihan98(To)
    integral =
    quad(@sub_latihan98,0,0.3,[],[],To);
    t = integral;
    %-----
    function y=sub_latihan98(x,To)
    clc
    E=10000;A=25;R=1.987;alpha=1750;beta=-5000;
    lambda =-80000;rho=1100;Cp=1.2;Cao=1;
    T = To+((lambda.*Cao.*x)./(rho.*Cp));
    k = A.*exp(-E./(R.*T));
    K = alpha.*exp(beta./T);
    y = 1./(k.*((1-x)-K.*x));

```

**HASIL PERHITUNGAN**

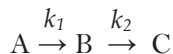
```

-----
Suhu optimum,T = 736.3458 K
Waktu optimum,t = 25.6406 menit

```

**Contoh 9.9 Optimasi Temperatur reaksi *multiple***

Suatu reaksi berurutan (*series*) fasa cair order 1.



dijalankan dalam sebuah reaktor *batch* yang beroperasi secara isothermis. Harga-harga konstanta  $k_1$  dan  $k_2$  mengikuti

persamaan :

$$k_1 = A_1 \exp\left(-\frac{E_1}{RT}\right) \quad (9.9.1)$$

$$k_2 = A_2 \exp\left(-\frac{E_2}{RT}\right) \quad (9.9.2)$$

Ingin dicari suhu optimum yang memberikan  $C_B$  yang maksimum untuk waktu  $\tau$  tertentu. Gunakan MATLAB untuk memaksimasi  $C_B = f(T)$  serta metode Runge Kutta untuk menyelesaikan persamaan-persamaan diferensial yang terbentuk. Data  $A_1 = 0,145$  (L/menit),  $A_2 = 0,05$  (L/menit),  $E_1 = 1167,8$ ;  $E_2 = 1580,9$  (kal/gmol);  $C_{A0}=2,0$  gmol/L,  $\tau = 60$  menit,  $R = 1,987$  cal/gmol/K, Kisaran suhu : 300 – 500 K.

### Penyelesaian :

Neraca mol masing komponen adalah :

$$\frac{dC_A}{dt} = -k_1 C_A \quad (9.9.3)$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B \quad (9.9.4)$$

$$\frac{dC_C}{dt} = k_2 C_B \quad (9.9.5)$$

Script program untuk soal di atas adalah sebagai berikut:

```
function soal99
clc
Topt=fminbnd(@latihan99,300,500);
disp('HASIL PERHITUNGAN ')
disp('-----')
disp(['T optimum      = ' num2str(Topt) ])
disp('-----')
```



```

%-----
function CB=latihan99(T)
Y0=[2 0 0];
tspan=[0:5:100];
[t Y]=ode45(@sub_latihan99,tspan,Y0,[],[],T)
[t Y];
plot(t,Y(:,1),t,Y(:,2),t,Y(:,3))
xlabel('waktu (menit)');
ylabel('konsentrasi (gmol/L)');
legend('CA','CB','CC')
CAt=Y(end,1);
CBt=Y(end,2);
Cct=Y(end,3);
CB=0-Y(end,2);
%-----
function dCdt=sub_latihan99(t,Y,flag,T)
CA=Y(1);CB=Y(2);CC=Y(3);
A1=0.145; %l/min
A2=0.05; %l/min
E1=1167; %cal/gmol
E2=1580; %cal/gmol %gmol/L
R=1.987; %cal/gmol/K
k1=A1.*exp(-E1./(R.*T));
k2=A2.*exp(-E2./(R.*T));
dCdt(1)=-k1.*CA;
dCdt(2)=k1.*CA-k2.*CB;
dCdt(3)=k2.*CB;
dCdt=dCdt';

```

```

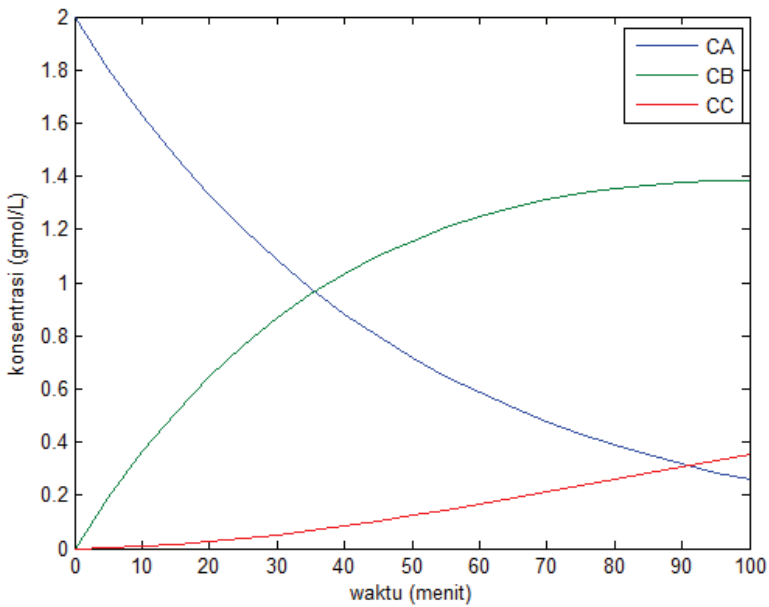
C =
    2.0000         0         0
    1.8054    0.1928    0.0017
    1.6298    0.3636    0.0067
    1.4712    0.5143    0.0145
    1.3281    0.6472    0.0247
    1.1989    0.7639    0.0372
    1.0822    0.8662    0.0516
    0.9769    0.9554    0.0677
    0.8819    1.0328    0.0853
    0.7961    1.0998    0.1041

```

0.7186	1.1573	0.1241
0.6487	1.2064	0.1449
0.5856	1.2478	0.1666
0.5286	1.2824	0.1889
0.4772	1.3110	0.2118
0.4308	1.3340	0.2352
0.3889	1.3522	0.2589
0.3510	1.3661	0.2829
0.3169	1.3760	0.3071
0.2860	1.3825	0.3315
0.2582	1.3859	0.3559

HASIL PERHITUNGAN

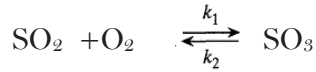
-----  
 T optimum = 300.0001  
 -----



**Gambar 9.3.** Grafik perubahan konsentrasi terhadap waktu dengan

### Contoh 9.10 Optimasi Suhu Masuk Reaktor Adiabatis Bolak-Balik

Reaksi fasa gas bolak-balik eksotermis :



dijalankan dalam reaktor tabung adiabatik dengan bantuan katalisator padat berbentuk butir-butir. Aliran gas dianggap *plug-flow*. Konversi  $\text{SO}_2$  keluar reaktor diharapkan  $X_{out}$ : 0,64. Karena reaktor bekerja adiabatik, sedangkan reaksinya endotermis, maka sambil berjalan sepanjang reaktor suhu gas akan naik. Pada reaksi bolak-balik eksotermis, suhu perlu dioptimumkan, karena jika suhu terlalu rendah maka kecepatan reaksi akan sangat lambat; sebaliknya, jika suhu terlalu tinggi, kesetimbangan reaksi akan bergeser ke kiri (prinsip kesetimbangan). Karena reaktor adiabatik, yang bisa diatur hanyalah suhu umpan reaktor ( $T_0$ ). Jadi dalam hal ini ingin ditentukan suhu optimum umpan reaktor, agar untuk mencapai konversi  $\text{SO}_2$ ,  $X_{out}$ : 0,64, diperlukan jumlah katalisator ( $W$ ) yang minimum.

Kecepatan reaksi mengikuti persamaan (Froment and Bischoff , 1990) :

$$r = \frac{k_1 P_{\text{O}_2} P_{\text{SO}_2} \left( 1 - \frac{P_{\text{SO}_3}}{P_{\text{SO}_2} \cdot P_{\text{O}_2}^{1/2} \cdot K_p} \right)}{22.414 \cdot \left( 1 + K_2 \cdot P_{\text{SO}_2} + K_3 \cdot P_{\text{SO}_3} \right)^2} \quad (9.10.1)$$

$$k_1 = \exp \left( 12,16 - \frac{5.473}{T} \right) \quad (9.10.2)$$

$$K_2 = \exp \left( -9,953 + \frac{8.619}{T} \right) \quad (9.10.3)$$

$$K_3 = \exp\left(-71,745 - \frac{52.956}{T}\right) \quad (9.10.4)$$

$$K_p = \exp\left(-10,68 - \frac{11.300}{T}\right) \quad (9.10.5)$$

Umpan reaktor berjumlah 55.000 kg/jam dengan komposisi mol  $\text{SO}_2 = 7,8\%$ ,  $\text{O}_2 = 10,8\%$  dan  $\text{N}_2 = 81,4\%$ . Tekanan reaktor dianggap tetap 1 atm.

### Penyelesaian :

Berdasarkan Sediawan dan Prasetya (1997) diperoleh beberapa persamaan berikut :

Dari neraca massa diperoleh persamaan :

$$W = 137,3 \int_{x=0}^{x=0,64} \left(\frac{1}{r}\right) dx \quad (9.10.6)$$

Tekanan parsial komponen :

$$P_{\text{SO}_2} = \frac{(1-x)}{(12,82 - 0,5x)} P_t \quad (9.10.7)$$

$$P_{\text{O}_2} = \frac{(1,3846 - 0,5x)}{(12,82 - 0,5x)} P_t \quad (9.10.8)$$

$$P_{\text{SO}_3} = \frac{x}{(12,82 - 0,5x)} P_t \quad (9.10.9)$$

Dari neraca energi diperoleh persamaan :

$$T = T_0 + 241,7x \quad (9.10.10)$$

Script program untuk penyelesaian soal di atas diberikan berikut:

```
function contoh9_9
clc,clear
[Topt,f]=fminbnd(@optimasisuhu,600,750);
disp('HASIL PERHITUNGAN')
```

```

disp('-----')
disp(['T optimum      = ' num2str(Topt),' K'])
disp(['berat katalis = ' num2str(f),' kg'])
disp('-----')
%-----
function w=optimasisuhu(To)
w=137.3*quad(@sub_optimasisuhu,0,0.64,[],[],To);
%-----
function y=sub_optimasisuhu(x,To)
T=To+241.7.*x;
PSO2=(1-x)/(12.82-0.5.*x);
PO2=(1.3846-x)/(12.82-0.5.*x);
PSO3=x/(12.82-0.5.*x);
k1=exp(12.16-(5473./T));
K2=exp(-9.953+(8619./T));
K3=exp(-71.745-(52956./T));
Kp=exp(-10.68+(11300./T));
r1=k1.*PO2.*PSO2.*(1-(PSO3./(PSO2.*PO2.^0.5.*Kp)));
r2=22.414.*(1+K2.*PSO2+K3.*PSO3).^2;
r=r1./r2;
y=1./r;

```

Hasil perhitungan program adalah :

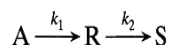
```

HASIL PERHITUNGAN
-----
T optimum      = 726.6269 K
berat katalis = 4073.9199 kg
-----

```

### Contoh 9.11 Menentukan Parameter Konstanta Laju Reaksi

Suatu reaksi seri dijalankan dalam reaktor secara *batch*. Reaksi yang terjadi dinyatakan dengan persamaan :



Neraca mol masing komponen adalah :

$$\frac{dC_A}{dt} = -k_1 C_A \quad (9.11.1)$$

$$\frac{dC_R}{dt} = k_1 C_A - k_2 C_R \quad (9.11.2)$$

$$\frac{dC_S}{dt} = k_2 C_R \quad (9.11.3)$$

Berdasarkan hasil eksperimen, perubahan konsentrasi komponen (mol/L) terhadap waktu reaksi,  $t$  (jam) ditampilkan pada tabel di bawah:

$t$	$C_A$	$C_R$	$C_S$
0	1.0000	0	0
0.50	0.3624	0.3949	0.2427
1.00	0.1313	0.3077	0.5610
1.50	0.0476	0.1801	0.7723
2.00	0.0173	0.0939	0.8889
2.50	0.0063	0.0459	0.9478
3.00	0.0023	0.0216	0.9761
3.5	0.0008	0.0099	0.9893
4.0	0.0003	0.0045	0.9952
4.5	0.0001	0.0020	0.9979
5.0	0.0000	0.0009	0.9991

Tentukan konstanta laju reaksi diatas berdasarkan data yang tersedia.

**Penyelesaian :**

Soal di atas diselesaikan dengan minimasi *Sum Square of Errors* (SSE) antara konsentrasi yang terhitung dengan konsentrasi data penelitian.

$$SSE = \sum_{i=1}^{i=n} (C_{i,hitung} - C_{i,data})^2 \quad (9.11.4)$$

Penulisan program untuk soal di atas diberikan berikut.

Script untuk program utama:

```
function contoh9_11
%Program optimasi parameter2 pada proses
clc
clear
%tebakan awal
kr = [2 1];
khit =
fminsearch('estparkinetik',kr,optimset('TolX
',1e-3))
```

Script untuk Sub program:

```
%-----
function SSE = estparkinetik(kr)
% menyelesaikan PD ordiner simultan
k1 = kr(1);
k2 = kr(2);
Yo=[1 0 0];
%interval waktu
tspan=[0:0.5:5];
[t,Y]=ode45('sub_estparkinetik',tspan,Yo,[],k1,k2)
CAhitung=Y(:,1);
CRhitung=Y(:,2);
CShitung=Y(:,3);
CAdata=[1;0.3624;0.1313;0.0476;0.0173;0.0063;0.0023;0
.0080;0.0030;0.0010;0.0000];
CRdata=[0;0.3949;0.3077;0.1801;0.0939;0.0459;0.0216;0
.0099;0.0045;0.0020;0.0009];
CSdata=[0;0.2427;0.5610;0.7723;0.8889;0.9478;0.9761;0
.9893;0.9952;0.9979;0.9991];
SSE=sum(CAhitung-CAdata).^2+sum(CRhitung-
CRdata).^2+sum(CShitung-CSdata).^2;
plot(tspan,CAdata,'ob',tspan,CAhitung,'b',tspan,CRdat
a,'or',tspan,CRhitung,'-r',...
tspan,CSdata,'og',tspan,CShitung,'-g')
```

```

title('grafik C vs waktu');
xlabel('waktu');
ylabel('C');
legend('CAdata', 'CAhit', 'CRdata', 'CRhit', 'CSdata', 'CS
hit')

```

Script untuk sub\_sub program :

```

function dCdt=sub_estparkinetik(t,Y,flag,k1,k2)
CA=Y(1);CR=Y(2);CS=Y(3);
dCdt(1) = -k1.*CA;
dCdt(2) = k1.*CA-k2.*CR;
dCdt(3) = k2.*CR;
dCdt = dCdt';

```

Hasil perhitungan :

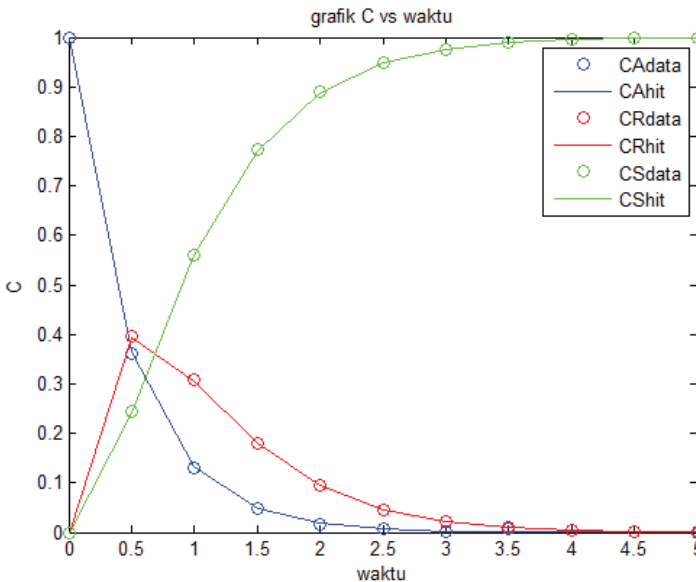
```

khit =

    2.0141    1.7559

```

Grafik simulasi :



**Gambar 9.4.** Grafik perubahan konsentrasi terhadap waktu



### Contoh 9.12 Pencocokan Data Kesetimbangan Uap-Cair dengan Regresi Non Linier

Data kesetimbangan uap-cair sering dikorelasikan menggunakan dua parameter yang dapat diatur pada campuran biner (Edgar et al., 2001). Dalam banyak kasus, kesetimbangan uap-cair multi komponen dapat diprediksi hanya dengan parameter biner. Untuk tekanan rendah, persamaan kesetimbangan adalah :

$$x_i \gamma_i P_i^{sat} = y_i P \quad (9.12.1)$$

Dengan :

- $P$  = tekanan total
- $P_i^{sat}$  = tekanan uap jenuh komponen  $i$
- $x_i$  = mol fraksi fasa cair komponen  $i$
- $\gamma_i$  = koefisien aktivitas komponen  $i$
- $y_i$  = fraksi mol fasa uap

Model van Laar untuk campuran biner adalah :

$$\ln \gamma_1 = A_{12} \left[ \frac{A_{21} x_2}{A_{12} x_1 + A_{21} x_2} \right]^2 \quad (9.12.2)$$

$$\ln \gamma_2 = A_{21} \left[ \frac{A_{12} x_2}{A_{12} x_1 + A_{21} x_2} \right]^2 \quad (9.12.3)$$

Dimana  $A_{12}$  dan  $A_{21}$  adalah konstanta biner yang dicari dengan optimasi untuk mencocokkan data perhitungan untuk  $x_i$ .

Data kesetimbangan uap-cair suatu campuran Air (1) dan 1,4 dioksan (2) diberikan Tabel 9.12.1 di bawah. Konstanta untuk persamaan Antoine juga diberikan pada tabel.

Tentukan konstanta van Laar  $A_{12}$  dan  $A_{21}$  berdasarkan data eksperimen tersebut.

**Tabel 9.12.1** Data Ekperimen sistem (1) Air dan (2) 1,4 Dioksane pada suhu 20 °C

Data eksperimen		Nilai Prediksi		
$x_1$	$p^{\text{expt}}$ (mmHg)	$p^{\text{cal}}$	$p$	$y^{\text{calc}}$
0,00	28,10	28,10	0,00	0,00
0,10	34,40	34,20	-0,20	0,2508
0,20	36,70	36,95	0,25	0,3245
0,30	36,90	36,97	0,07	0,3493
0,40	36,80	36,75	-0,05	0,3576
0,50	36,70	36,64	-0,06	0,3625
0,60	36,50	36,56	0,06	0,3725
0,70	36,40	36,36	-0,04	0,3965
0,80	32,90	32,84	-0,06	0,4503
0,90	27,70	27,72	0,02	0,5781
1,00	17,50	17,50	0,00	1,0

$$\log P_i^{\text{sat}} = a_1 - \frac{a_2}{T + a_3} \quad p^{\text{sat}} : \text{mmHg}, T : ^\circ \text{C}$$

	$a_1$	$a_2$	$a_3$	Range
(1)Air	8,07131	1730,630	233,426	(1-100°C)
(2)1,4 Dioksan	7,43155	1554,679	240,337	(20-105°C)

Note : Data reported by Hororka et al. (1936)

### Penyelesaian :

Tekanan total campuran dapat dituliskan :

$$P = y_1 P + y_2 P \quad (9.12.4)$$

Dengan menggunakan persamaan (9.12.1) - (9.12.3) diperoleh persamaan :

$$P = x_1 \exp \left( A_{12} \left[ \frac{A_{21} x_2}{A_{12} x_1 + A_{21} x_2} \right]^2 \right) P_1^{\text{sat}} + x_2 \exp \left( A_{21} \left[ \frac{A_{12} x_1}{A_{12} x_1 + A_{21} x_2} \right]^2 \right) P_2^{\text{sat}} \quad (9.12.5)$$

Tekanan jenuh komponen dapat diprediksi dengan persamaan Antoine. Untuk temperatur yang diberikan dan sistem biner  $x_2 = (1 - x_1)$  tekanan adalah :

$$P = P(x_1, A_{12}, A_{21}) \quad (9.12.6)$$

Dua koefisien biner ditentukan dari nilai eksperimen  $P$  terhadap  $x_1$  dengan regresi non linier yaitu dengan minimasi fungsi objektif :

$$f = \sum (P_j^{calc} - P_j^{exp})^2 \quad (9.12.7)$$

Program MATLAB yang digunakan untuk menyelesaikan soal 9.12 diberikan berikut:

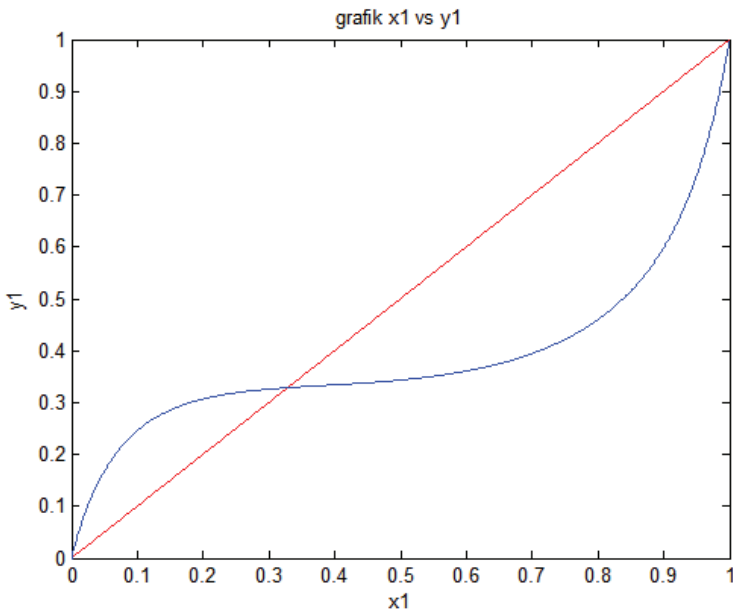
```
function contoh9_12
clc
clear
%tebakkan awal
x = [2 1];
A =
fminsearch(@activity,x,optimset('Display','iter'))
%-----
function f = activity(x)
A12=x(1);A21=x(2);
x1=[0:0.1:1]';
T=20;
x2=(1-x1);
Pdata=[28.1 34.4 36.7 36.9 36.8 36.7 36.5 35.4 32.9
27.7 17.5]';
a1w=8.07131;a1d=7.43155;a2w=1730.630;a2d=1554.679;a3w
=233.46;a3d=240.337;
p1=10.^(a1w-(a2w./(T+a3w)));p2=10.^(a1d-
(a2d./(T+a3d)));
gamma1=exp(A12.*(A21.*x2./(A12.*x1+A21.*x2)).^2);
gamma2=exp(A21.*(A12.*x1./(A12.*x1+A21.*x2)).^2);
Pcal=x1.*gamma1.*p1+x2.*gamma2.*p2;
f=sum(Pcal-Pdata).^2
y1=x1.*gamma1.*p1./Pcal;
plot(x1,x1,'-r',x1,y1,'-b')
title('grafik x1 vs y1')
```

```
xlabel('x1')
ylabel('y1')
```

Hasil perhitungan :

$$A = \begin{matrix} & 2.1405 & 1.5242 \end{matrix}$$

Setelah diperoleh nilai  $A_{12} = 2,1405$  dan  $A_{21} = 1,5242$  selanjutnya dihitung nilai  $y_1$  dan di plot terhadap  $x_1$ . Hasil plot ditunjukkan Gambar 9.5 di bawah

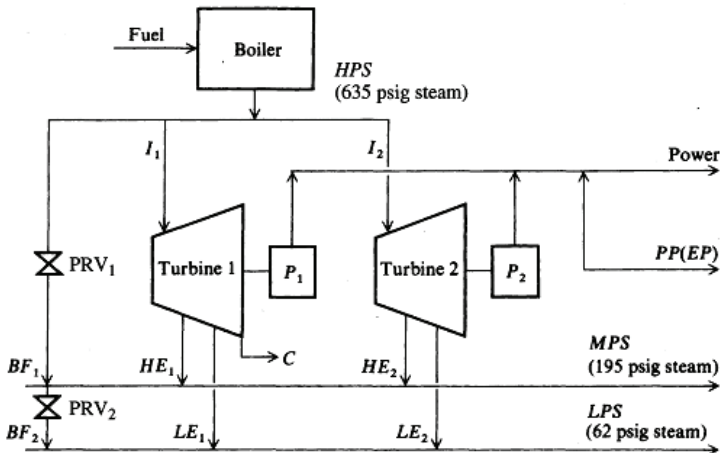


**Gambar 9.5.** Grafik perubahan fraksi fasa uap terhadap fraksi fasa cair

### Contoh 9.13 Optimasi Sistem *Boiler-Steam Turbin Generator*

Desain dan operasi *Sistem Boiler-Steam Turbin Generator* dalam industri kimia ditunjukkan gambar 9.6 (Edgar et al.,

2001). Gambar tersebut menunjukkan sistem *steam* dan daya untuk *power house* kecil dengan bahan bakar pulp kayu. Untuk memproduksi daya listrik, sistem ini memiliki dua turbin generator yang karakternya tercantum pada Tabel 9.13.1. Turbin 1 berupa turbin *double extraction* dengan dua aliran *intermediate* keluar pada 195 dan 62 psi, stage terakhir menghasilkan kondensate yang digunakan sebagai air umpan boiler. Turbin 2 adalah *single extraction* dengan satu aliran *intermediate* pada 195 psi dan aliran keluar pada 62 psi tanpa menghasilkan kondensate.



**Gambar 9.6.** Sistem *Boiler -Steam Turbine Generation*

- Key :**  $I_i$  = inlet flowrate for turbine  $i$  [lbm/h]  
 $HE_i$  = exit flow rate from turbin  $i$  to 195 header [lbm/h]  
 $LE_i$  = exit flow rate from turbin  $i$  to 62 psi header [lbm/h]  
 $C$  = condensate flow rte from turbine 1 [lbm/h]  
 $P_i$  = power generated by turbine  $i$  [kw]  
 $BF_1$  = bypass flowrate from 635 psi to 195 psi header [lbm/h]  
 $BF_2$  = bypass flowrate from 195 psi to 62 psi header [lbm/h]  
 $HPS$  = flowrate through 635 psi header [lbm/h]  
 $MPS$  = flowrate through 195 psi header [lbm/h]  
 $LPS$  = flowrate through 62 psi header [lbm/h]

- $PP$  = purchased power [kw]
- $EP$  = excess power [kw]
- $PRV$  = pressure reducing valve

Turbin 1 lebih efisien karena pelepasan energi dari kondensasi steam, namun tidak dapat menghasilkan power seperti turbin kedua. Steam berlebih dapat melewati turbin menjadi steam dua level melalui *pressure reducing valve*.

Tabel 9.13.2 berisi daftar informasi tentang perbedaan level steam dan Tabel 9.13.3 memberikan data kebutuhan steam. Untuk memenuhi kebutuhan daya listrik, kekurangan daya listrik boleh dibeli dari produsen lain dengan minimum nilai 12.000 kW. Jika daya listrik yang diperlukan memenuhi kebutuhan sistem lebih kecil dari nilai ini, daya yang tidak digunakan akan dikenakan biaya pinalti. Tabel 9.13.4. memberikan data biaya bahan bakar untuk boiler dan tambahan daya listrik untuk mengoperasikan sistem utilitas.

**Tabel 9.13.1 Data turbin**

Turbin 1			Turbin 2		
Maximum generative capacity	6.250	kW	Maximum generative capacity	9.000	kW
Minimum load	2.500	kW	Minimum load	3.000	kW
Maxium inlet flow	192.000	lb <sub>m</sub> /h	Maxium inlet flow	244.000	lb <sub>m</sub> /h
Maximum condensate flow	62.000	lb <sub>m</sub> /h	Maximum 62 psi exhaust	142.000	lb <sub>m</sub> /h
Maximum internal flow	132.000	lb <sub>m</sub> /h	High pressure extraction at	192	psig
High pressure extraction at	192	psig	Low pressure extraction at	62	psig
Low pressure extraction at	62	psig			

**Tabel 9.13.2 Data steam header**

Header	Pressure (psig)	Temperature (°F)	Enthalpy (Btu/lb <sub>m</sub> )
High pressure steam	635	720	1359,8
Medium pressure steam	195	130 superheat	1267,8
Low pressure steam	62	130 superheat	1251,4
Feedwater (codensate)			193,0

**Tabel 9.13.3 Data kebutuhan sistem**

Sumber	Kebutuhan
Medium pressure steam (195 psig)	271.536 lbm/h
Low pressure steam (62 psig)	100.623 lbm/h
Electric power	24.550 kW

**Tabel 9.13.4 Data Energi**

Sumber	Kebutuhan
Fuel Cost	271.536 lbm/h
Boiler efficiency	0,75
Steam Cost (635 psi)	\$2,24/10 <sup>6</sup> Btu \$2,24(1359,8 -193)/10 <sup>6</sup> \$0,002614/lb <sub>m</sub>
Purchased electric power	\$0,0239/kWh average
Demand penalty	\$0,009825/kWh
Base Purchased power	12.000 kW

Ingin dicari biaya minimum sistem dengan memilih laju alir steam dan daya yang dibangkitkan atau dibeli terhadap kebutuhan dan batasan sistem.

Fungsi *objective* adalah biaya operasi per jam :

$$\text{minimasi : } f = 0,00261 HPS + 0,0239 PP + 0,00983 EP \quad (9.13.1)$$

Konstrain disusun ke dalam serangkaian persamaan berikut:

**Turbin 1 :**

$$\begin{aligned} P_1 &\leq 6.250 \\ P_1 &\geq 2.500 \\ HE_1 &\leq 192.000 \\ C &\leq 62.000 \\ I_1 - HE_1 &\leq 132.000 \end{aligned} \quad (9.13.2)$$

**Turbin 2 :**

$$\begin{aligned} P_2 &\leq 9.000 \\ P_2 &\geq 3.000 \\ HE_1 &\leq 192.000 \\ I_2 &\leq 62.000 \\ LE_2 &\leq 142.000 \end{aligned} \quad (9.13.3)$$

**Neraca Massa :**

$$\begin{aligned} HPS - I_1 - I_2 - BF_1 &= 0 \\ I_1 + I_2 + BF_1 - C - MPS - LPS &= 0 \\ I_1 - HE_1 - LE_1 - C &= 0 \\ I_2 - HE_2 - LE_2 &= 0 \\ HE_1 + HE_2 + BF_1 - BF_2 - MPS &= 0 \\ LE_1 + LE_2 + BF_2 - LPS &= 0 \end{aligned} \quad (9.13.4)$$

**Daya yang dibeli (*Power purchased*):**

$$EP + PP \geq 12.000 \quad (9.13.5)$$

**Kebutuhan (*Demands*):**

$$\begin{aligned} MPS &\geq 271.536 \\ LPS &\geq 100.623 \\ P_1 + P_2 + PP &\geq 24.550 \end{aligned} \quad (9.13.6)$$

***Neraca Energi***

$$\begin{aligned} 1.359,8 I_1 - 1.267,8 HE_1 - 1.251,4 LE_1 - 192 C - 3.413 P_1 &= 0 \\ 1.359,8 I_2 - 1.267,8 I_2 - 1.251,4 LE_2 - 192C - 3.413P_2 &= 0 \end{aligned} \quad (9.13.7)$$



### Penyelesaian :

Persoalan ini diselesaikan dengan MATLAB menggunakan fungsi linprog. Berikut langkah penyelesaian untuk soal diatas.

### Kombinasi variabel dalam satu vektor

Jumlah variabel sebanyak 16 yaitu :  $I_1, I_2, HE_1, HE_2, LE_1, LE_2, C, BF_1, BF_2, HPS, MPS, LPS, P_1, P_2, PP$ , dan  $EP$ .

Buat variabel tersebut dalam satu vektor. Berikut kode membuat vektor menggunakan *cell array of strings*. Masing-masing string adalah nama variabel.

```
variables =  
{ 'I1', 'I2', 'HE1', 'HE2', 'LE1', 'LE2', 'C', 'BF1', ...  
  'BF2', 'HPS', 'MPS', 'LPS', 'P1', 'P2', 'PP', 'EP' };  
N = length(variables);  
% create variables for indexing  
for v = 1:N  
    eval([variables{v}, ' = ', num2str(v), ';' ]);  
end
```

### Konstrain batas

Terdapat empat variabel dengan batas bawah dan enam variabel dengan batas atas berdasarkan persamaan-persamaan di atas.

Batas bawah adalah :

$$\begin{aligned} P_1 &\geq 2.500 \\ P_2 &\geq 3.000 \\ MPS &\geq 271.536 \\ LPS &\geq 100.623 \end{aligned}$$

Juga semua variabel adalah positif, yang artinya memiliki batas bawah nol.

Buat vektor batas bawah sebagai vektor 0, kemudian tambahkan empat batas bawah lainnya.

```
lb = zeros(size(variables));  
lb([P1,P2,MPS,LPS]) = ...  
[2500,3000,271536,100623];
```

### ***Linear Inequality Constraints***

Terdapat tiga *linear inequalities* dalam persamaan :

$$I_1 - HE_1 \leq 132,000$$

$$EP + PP \geq 12,000$$

$$P_1 + P_2 + PP \geq 24,550.$$

Agar memiliki persamaan dalam bentuk  $A \cdot x \leq b$ , letakkan semua variabel di sisi kiri *inequality*. Semua persamaan telah memiliki bentuk tersebut. Pastikan bahwa setiap *inequality* dalam bentuk *lebih kecil* dengan mengalikannya dengan (-1).

$$I_1 - HE_1 \leq 132,000$$

$$-EP - PP \leq -12,000$$

$$-P_1 - P_2 - PP \leq -24,550.$$

Dalam lembar kerja MATLAB, buat matriks nol dengan  $3 \times 16$ , berkaitan  $3$  *linier inequalities* dalam  $16$  variabel. Buat vektor  $b$  dengan tiga komponen.

```
A = zeros(3,16);  
A(1,I1) = 1; A(1,HE1) = -1; b(1) = 132000;  
A(2,EP) = -1; A(2,PP) = -1; b(2) = -12000;  
A(3,[P1,P2,PP]) = [-1,-1,-1];  
b(3) = -24550;
```

### ***Linear Equality Constraints***

Terdapat delapan persamaan linier yaitu persamaan (9.13.4) dan persamaan (9.13.7). Bentuk persamaan adalah  $Aeq \cdot x = beq$ .

Selanjutnya tulis matrks  $Aeq$  dan vektor  $beq$  yang berhubungan dengan persamaan ini. Dalam lembar kerja MATLAB, buat matriks  $Aeq$  sebagai matriks nol dengan  $8 \times 16$ , berkaitan dengan  $8$  persamaan linier dalam  $16$  variabel. Buat vektor  $beq$  dengan  $8$  komponen semuanya nol

```
Aeq = zeros(8,16); beq = zeros(8,1);
Aeq(1,[LE2,HE2,I2]) = [1,1,-1];
Aeq(2,[LE1,LE2,BF2,LPS]) = [1,1,1,-1];
Aeq(3,[I1,I2,BF1,HPS]) = [1,1,1,-1];
Aeq(4,[C,MPS,LPS,HPS]) = [1,1,1,-1];
Aeq(5,[LE1,HE1,C,I1]) = [1,1,1,-1];
Aeq(6,[HE1,HE2,BF1,BF2,MPS]) = [1,1,1,-1,-1];
Aeq(7,[HE1,LE1,C,P1,I1]) =
[1267.8,1251.4,192,3413,-1359.8];
Aeq(8,[HE2,LE2,P2,I2]) =
[1267.8,1251.4,3413,-1359.8];
```

**Objective**

Fungsi objective adalah persamaan (9.13.1) yang dapat dituliskan :

$$f^T x = 0.002614 \cdot HPS + 0.0239 \cdot PP + 0.009825 \cdot EP.$$

Tulis ekspresi ini sebagai vektor f dikali dengan vektor x.

```
f = zeros(size(variables));
f([HPS PP EP]) = [0.002614 0.0239 0.009825];
```

**Solusi soal dengan linprog**

Anda sekarang sudah memasukkan yang dibutuhkan linprog solver. Panggil solver dan cetak keluaran dalam bentuk format :

```
[x fval] = linprog(f,A,b,Aeq,beq,lb,ub);
for d = 1:N
    fprintf('%12.2f \t%s\n',x(d),variables{d})
end
```

Berikut program lengkap dari penjelasan di atas.

```
function contoh9_13
clc,clear
%variabel
variables =
{'I1','I2','HE1','HE2','LE1','LE2','C','BF1',
, ...
```

```

'BF2','HPS','MPS','LPS','P1','P2','PP','EP'}
;
N = length(variables);
% create variables for indexing
for v = 1:N
    eval([variables{v}, ' = ',
num2str(v), ';' ]);
end
%lower and upper bound
lb = zeros(size(variables));
lb([P1,P2,MPS,LPS]) = ...
    [2500,3000,271536,100623];
ub = Inf(size(variables));
ub([P1,P2,I1,I2,C,LE2]) = ...
    [6250,9000,192000,244000,62000,142000];
%linier inequality constraint
A = zeros(3,16);
A(1,I1) = 1; A(1,HE1) = -1; b(1) = 132000;
A(2,EP) = -1; A(2,PP) = -1; b(2) = -12000;
A(3,[P1,P2,PP]) = [-1,-1,-1];
b(3) = -24550;
%linier equality constraint
Aeq = zeros(8,16); beq = zeros(8,1);
Aeq(1,[LE2,HE2,I2]) = [1,1,-1];
Aeq(2,[LE1,LE2,BF2,LPS]) = [1,1,1,-1];
Aeq(3,[I1,I2,BF1,HPS]) = [1,1,1,-1];
Aeq(4,[C,MPS,LPS,HPS]) = [1,1,1,-1];
Aeq(5,[LE1,HE1,C,I1]) = [1,1,1,-1];
Aeq(6,[HE1,HE2,BF1,BF2,MPS]) = [1,1,1,-1,-
1];
Aeq(7,[HE1,LE1,C,P1,I1]) =
[1267.8,1251.4,192,3413,-1359.8];
Aeq(8,[HE2,LE2,P2,I2]) =
[1267.8,1251.4,3413,-1359.8];
%objective function
f = zeros(size(variables));
f([HPS PP EP]) = [0.002614 0.0239 0.009825];
%solver
[x fval] = linprog(f,A,b,Aeq,beq,lb,ub);

```

```

for d = 1:N
    fprintf('%12.2f \t%s\n',x(d),variables{d})
end
fval

```

Hasil simulasi:

Optimization terminated.

136328.74	I1
244000.00	I2
128159.00	HE1
143377.00	HE2
0.00	LE1
100623.00	LE2
8169.74	C
0.00	BF1
0.00	BF2
380328.74	HPS
271536.00	MPS
100623.00	LPS
6250.00	P1
7060.71	P2
11239.29	PP
760.71	EP

fval =

1.2703e+003

Hasil di atas adalah penyelesaian optimum terhadap program linier yang disajikan persamaan 9.13.1 – 9.13.7. Biaya minimum adalah \$ 1.270/jam.

## 9.7 RUJUKAN

- a. Thomas E. Edgar, Himmelblau, D., and Lasdon, 2001. *Optimization of Chemical Processes*. Mc. Graw Hill. New York.

- b. Arora, Jasbir, 2012, *Introduction to Optimum Design*, 3<sup>rd</sup> ed. Elsevier. UK.
- c. Steven C. Chapra. 2012. *Applied Numerical Methods With MATLAB For Engineers And Scientists*. 3rd ed. Mc. Graw Hill. New York.
- d. Sediawan, Wahyudi Budi. 1997. *Pemodelan Matematis dan Penyelesaian Numeris dalam Teknik Kimia*. Andi. Yogyakarta

# SIMULINK

---

## 10.1 PENDAHULUAN

Bab ini membahas tentang Simulink dan penggunaannya untuk menyelesaikan persoalan-persoalan dalam Teknik Kimia. Setelah menyelesaikan pokok bahasan ini mahasiswa diharapkan akan mampu menggunakan Simulink untuk menyelesaikan persoalan model matematis yang terbentuk dari persoalan fisis dan kimia dalam perkuliahan maupun kejadian nyata di lapangan.

## 10.2 SIMULINK LIBRARY

**Apa itu Simulink?** Simulink adalah paket software untuk pemodelan, simulasi dan analisis sistem dinamis. Software ini mendukung sistem linier dan nonlinear, model waktu kontinu, waktu tersample (diskrit) atau gabungan keduanya. Sistem juga dapat berupa *multirate*, yaitu memiliki bagian yang berbeda antara yang tersample atau ter-*update* pada laju yang berbeda.

Pemodelan Simulink menyediakan *Graphical User Interface* (GUI) untuk membuat model seperti blok diagram, menggunakan operasi klik dan drag mouse. Dengan fasilitas ini Anda dapat menggambar model seperti Anda akan menggambar dengan pensil pada kertas. Hal ini jauh dari paket simulasi sebelumnya dimana kita perlu memformulasikan persamaan diferensial dan persamaan yang berbeda dalam suatu bahasa atau program. Simulink memiliki *block library of sinks, sources, linear and nonlinear components, and connectors* yang lengkap. Anda dapat memodifikasi atau membuat blok sendiri.

Model adalah hirarkis, sehingga Anda dapat membuat model menggunakan kedua pendekatan top-down atau bottom-up.

Setelah Anda mendefinisikan suatu model, Anda dapat mensimulasikan menggunakan suatu pilihan metode integrasi, juga dari menu Simulink atau dengan memasukkan perintah di *command window* MATLAB.

Menu-menu sangat nyaman untuk kerja interaktif, sementara pendekatan command line sangat bermanfaat/berguna untuk menjalankan simulasi batch (misal, Jika Anda sedang menjalankan simulasi Monte Carlo atau ingin menyapu parameter pada kisaran nilai tertentu). Dengan menggunakan Scope dan display blok lainnya, Anda dapat melihat hasil simulasi saat simulasi sedang berjalan.

Tool analisis model terdiri linierisasi dan perangkat cepat yang dapat diakses dari MATLAB Command line, ditambah banyak tools dalam MATLAB dan aplikasi toolbox. Karena MATLAB dan Simulink terintegrasi, anda juga dapat melakukan simulasi, analisis dan revisi model dalam beberapa cara.

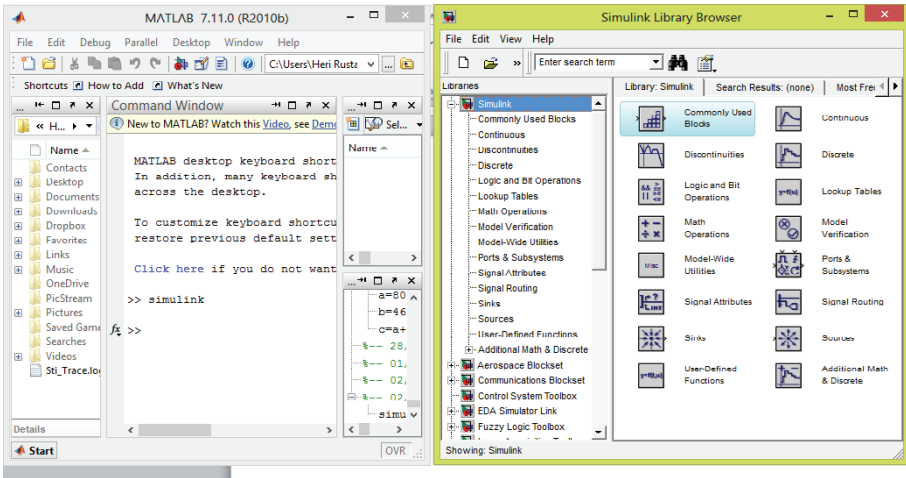
### **Memulai Matlab dan Simulink**

Untuk memulai Simulink Anda dapat melakukan langkah-langkah berikut:


- 1) Start MATLAB dan selanjutnya pada Command Window

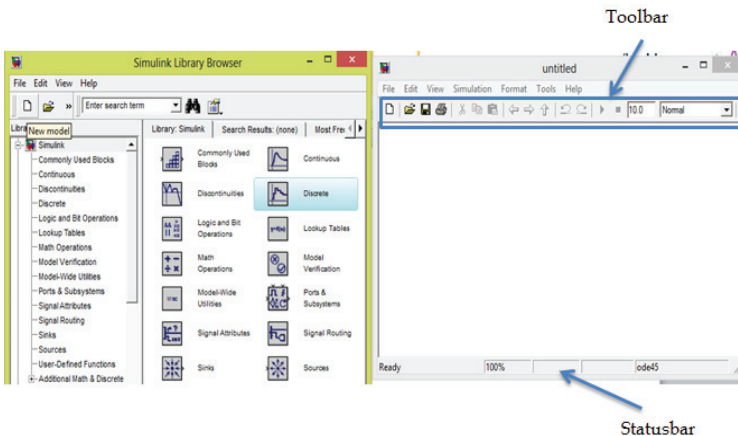


MATLAB ketik **simulink**. Maka akan muncul gambar seperti di bawah:



**Gambar 10.1** Tampilan command window MATLAB dan Simulink library

2) Buka New Simulation Window dari menu file di pojok kiri atas dengan cara klik pada tanda . Akan dihasilkan menu seperti gambar di bawah.



**Gambar 10.2** Simulink library dan new worksheet window

3) Anda dapat membuat block diagram sendiri dengan drag dan dropping block yang sesuai dari jendela Simulink utama. Beberapa block yang umumnya paling banyak dipakai :

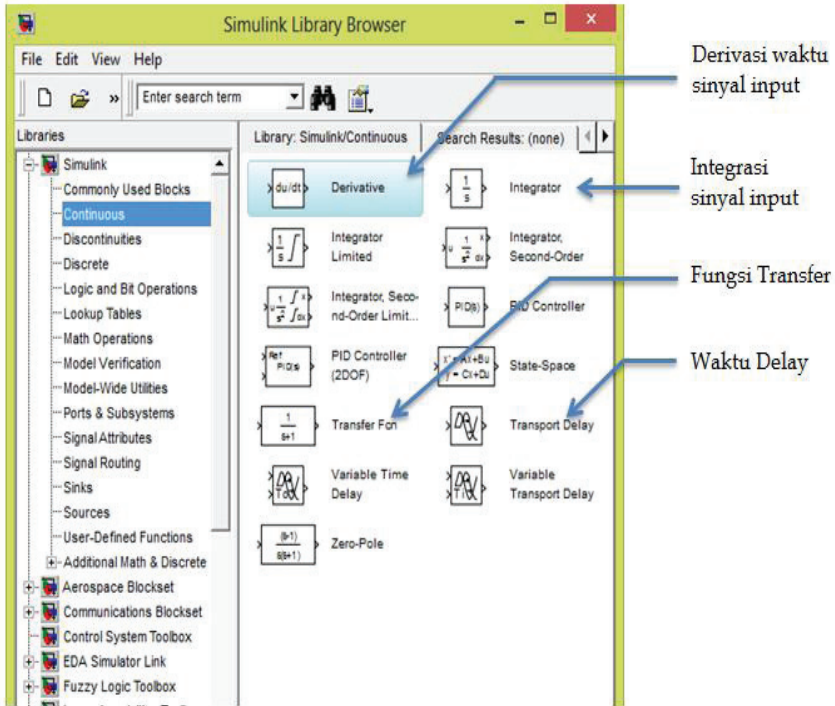
**a. Commonly Used Bloks**



**Gambar 10.3.** Tampilan Blok pada menu tree *Commonly Used Block*

Dari *Commonly Used Block* kita dapat menggunakan konstanta, perkalian dengan skalar, menampilkan beberapa sinyal keluar menjadi satu sinyal atau membagi satu sinyal input menjadi beberapa sinyal keluar dan lainnya.

## b. Continuous

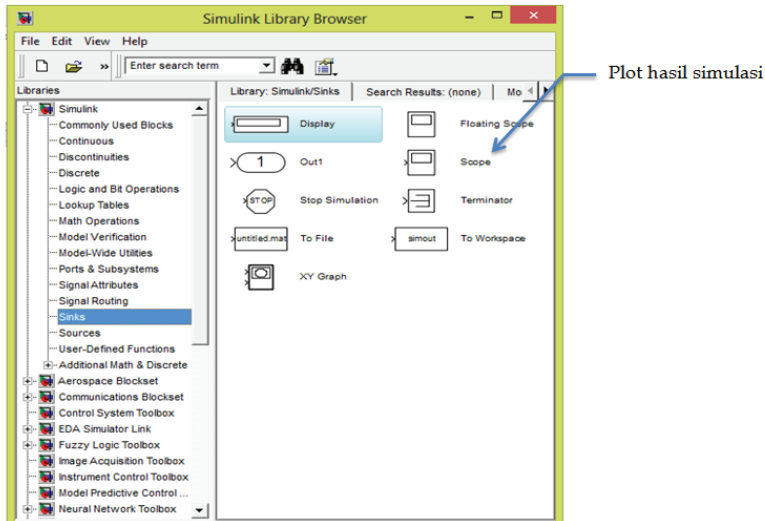


**Gambar 10.4.** Tampilan Blok pada menu tree **Continuous**

Dari blok **Continuous** (klik dua kali pada tombol **Continuous**) Anda dapat menggunakan tipe blok yang sesuai untuk membuat sistem dinamis (misal fungsi transfer, *time delay* dan lainnya).

## c. Sink

Dari **Sink** kita sering menggunakan **Scope** untuk menampilkan hasil simulasi.



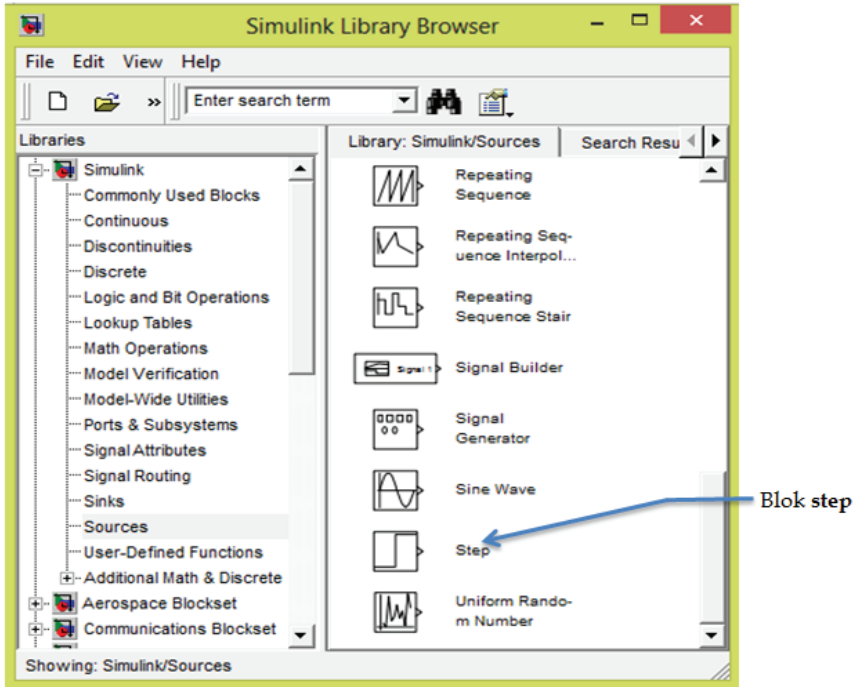
Gambar 10.5. Tampilan Blok pada menu tree Sink.

#### d. Source

Dari **Sources** suatu fungsi **Step** digunakan untuk mensimulasikan perubahan step. Untuk mengetahui parameter yang harus dimasukkan pada masing-masing blok, Anda dapat melakukan drag pada worksheet window Simulink yang baru dan klik dua kali pada blok tersebut. Tampilan blok Source di tunjukan Gambar 10.6.

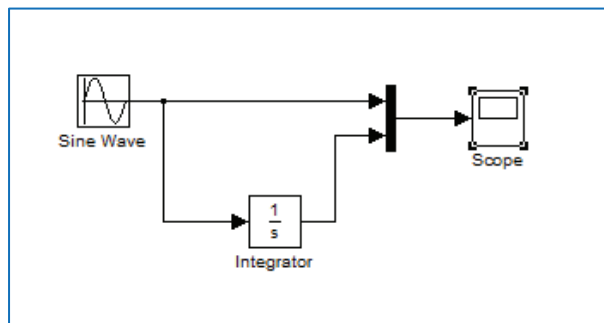
### Membuat Model Sederhana

Contoh berikut menunjukkan bagaimana membuat suatu model menggunakan banyak model perintah dak aksi yang akan Anda gunakan untuk membuat model Anda sendiri. Instruksi untuk membuat model sederhana akan dijelaskan di bawah. Untuk penjelasan detail lainnya akan diberikan pada bab berikutnya.



**Gambar 10.6.** Tampilan Blok pada menu tree “*Source*”.

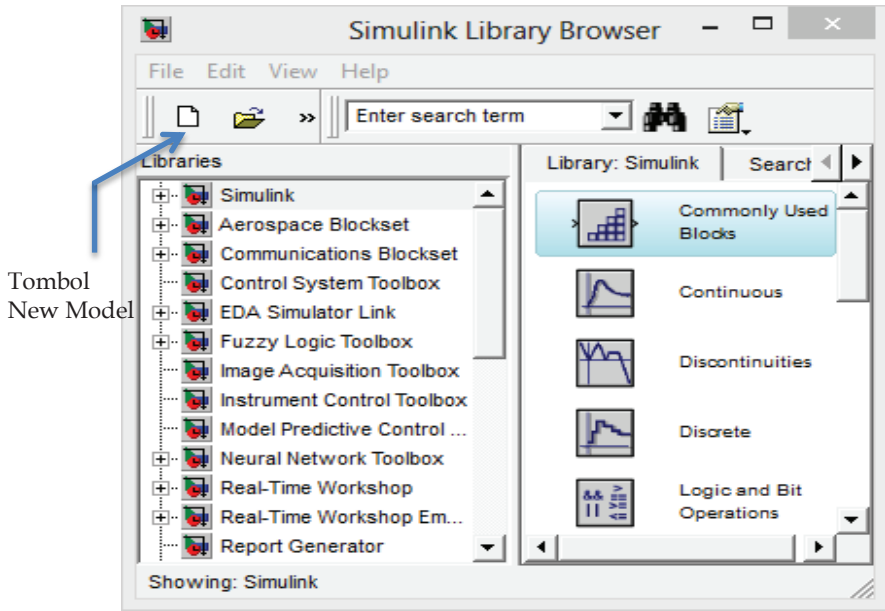
Model ini mengintegrasikan suatu gelombang sinus dan menampilkan hasil sepanjang gelombang sinus. Blok diagram untuk model seperti Gambar 10.7.



**Gambar 10.7** Blok diagram integrasi gelombang sinus

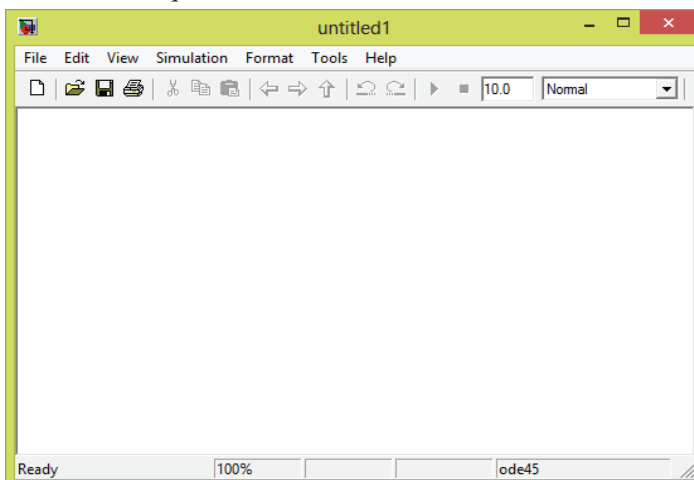
Untuk membuat model, pertama ketik simulink dalam MATLAB Command Window. Pada Microsoft Window akan

muncul **Simulink Library Browser** Seperti Gambar 10.8.



**Gambar 10.8.** Jendela Simulink Library Browser

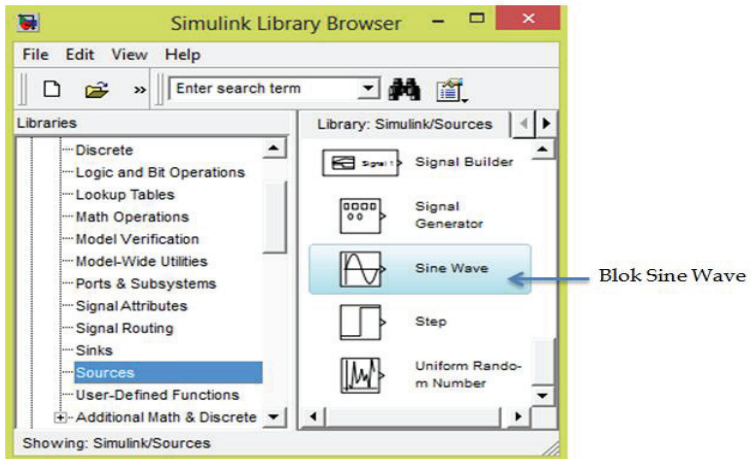
Untuk membuat model baru pada Windows, pilih tombol **New Model** pada toolbar Library Browser. Simulink membuka jendela model baru seperti Gambar 10.9.



**Gambar 10.9.** Jendela Simulink baru

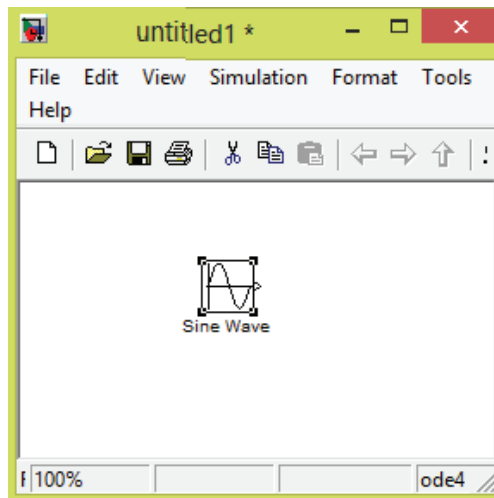
Untuk membuat model, Anda perlu mengkopir blok ke dalam model dari Simulink Bloks library berikut :

- Source library (blok Sine wave)
- Sink Library (blok Scope)
- Continuous Library (blok Integrator)
- Signal & System library (blok Mux)



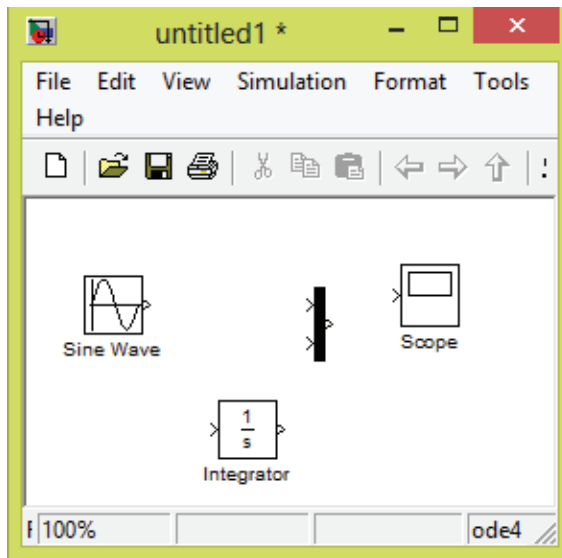
**Gambar 10.9.** Menu Source

Selanjutnya drag blok Sine Wave dari Source Window ke jedela model Anda.



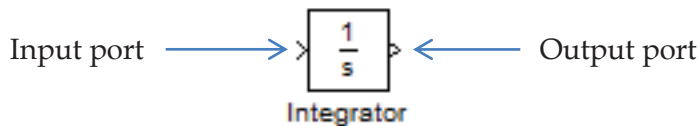
**Gambar 10.10.** Blok Sin wave

Kopi blok yang lainnya dengan cara yang sama dari masing-masing library ke model window sehingga tampak seperti gambar di bawah.



**Gambar 10.11.** Beberapa blok Simulink

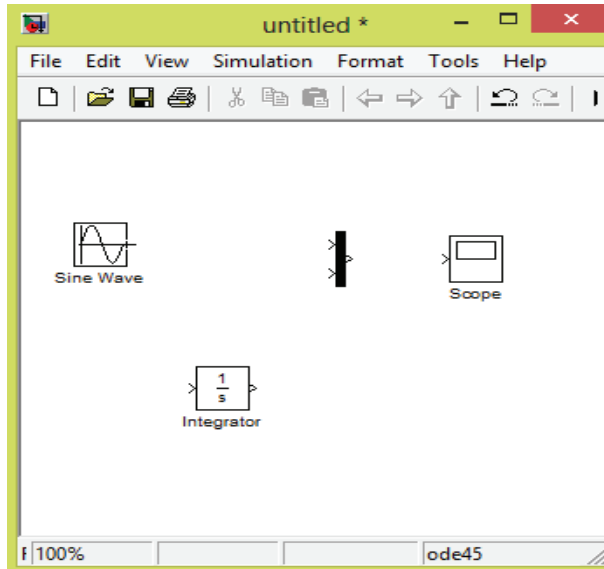
Jika Anda memperhatikan ikon blok, Anda melihat tanda kurung di sisi kanan Sine Wave dan dua di sisi kiri blok Mux. Simbol “>” yang keluar dari suatu blok adalah *output port* dan jika simbol menuju blok maka disebut *input port*. Sinyal keluar dari output port dan masuk ke input port blok lainnya melalui garis hubung. Saat blok terkoneksi, simbol tdk muncul.



**Gambar 10.12.** Input dan output port suatu blok

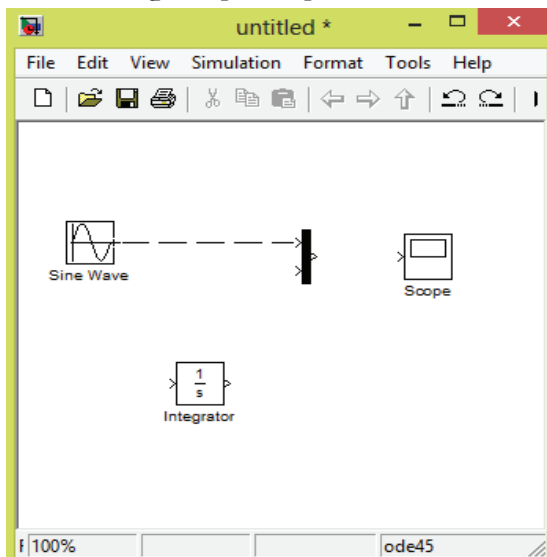
Kini saatnya menyambungkan blok. Sambungkan blok Sine Wave ke atas input port blok Mux. Posisi pointer di atas output port sisi kanan blok Sine Wave. Catatan bahwa bentuk kursor berubah melewati garis.





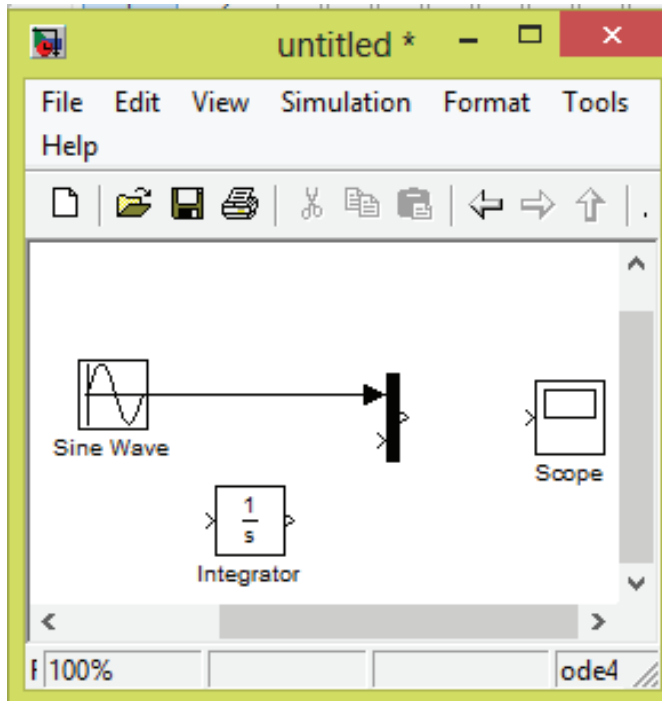
**Gambar 10.9.** Susunan blok Simulink

Tahan tombol mouse dan pindahkan kursor ke input port blok Mux, maka akan muncul garis putus-putus saat mouse di tekan.



**Gambar 10.9.** Drag koneksi blok

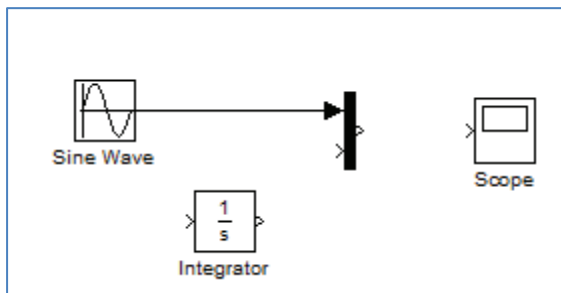
Sekarang lepaskan mouse. Maka blok akan tersambung.



**Gambar 10.9.** Koneksi blok

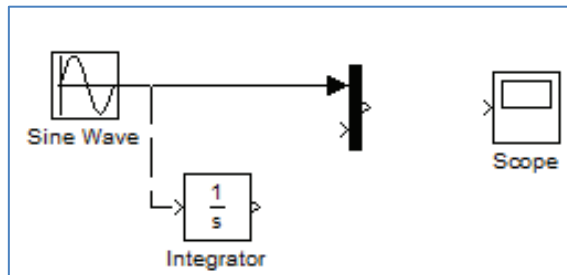
Cara membuat garis cabang berbeda dari cara membuat garis sebagaimana yang telah Anda lakukan. Untuk menyambungkan dengan garis yang ada, ikuti langkah berikut:

1. Pertama, posisi pointer pada garis antara blok Sin wave dan Mux.



**Gambar 10.9.** Koneksi blok Sin wave dan mux

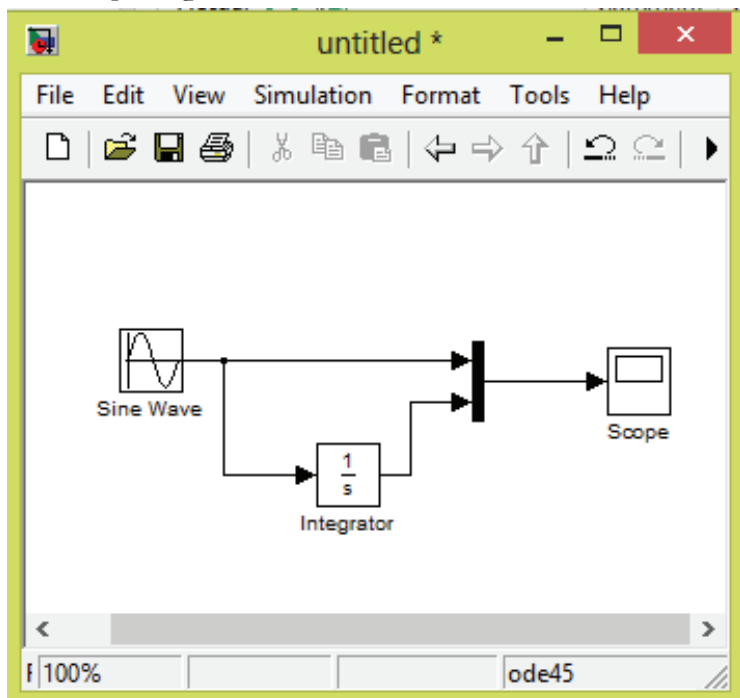
2. Tekan dan tahan tombol Ctrl. Tekan mouse dan drag pointer ke input port blok Integrator.



**Gambar 10.9.** Membuat cabang blok Simulink

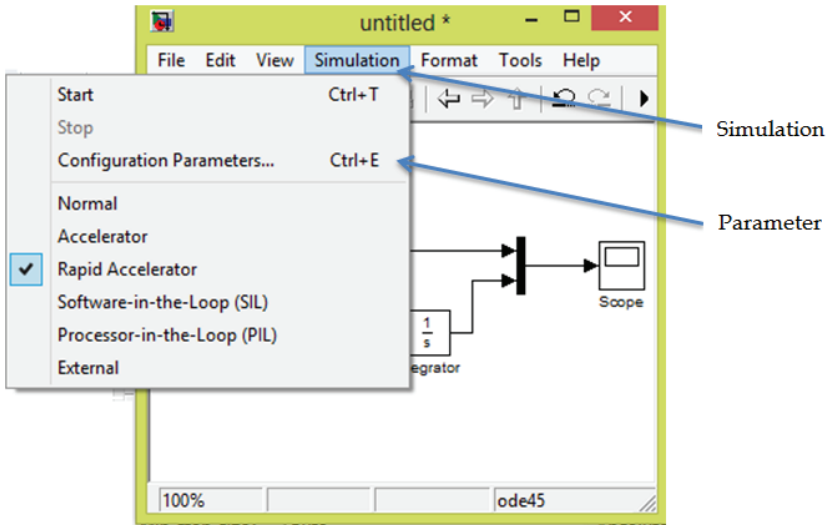
3. Lepaskan tombol mouse. Simulink menggambar suatu garis antara titik awal dan input port blok Integrator.

Selesai membuat koneksi blok. Jika sudah selesai, model Anda akan terlihat seperti gambar di bawah.

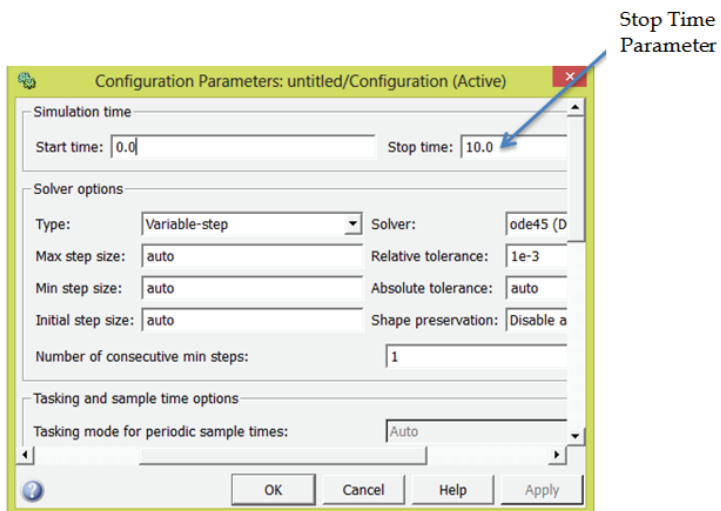


**Gambar 10.9.** Model Simulink lengkap

Selanjutnya, buka blok Scope untuk melihat output simulasi. Jaga endela Scope tetap terbuka, atur Simulink untuk berjalan selama 10 detik. Pertama, atur parameter simulasi dengan memilih **Parameters** dari menu **Simulation**. Pada kotak dialog yang muncul, tulis **Stop time** adalah 10.0.

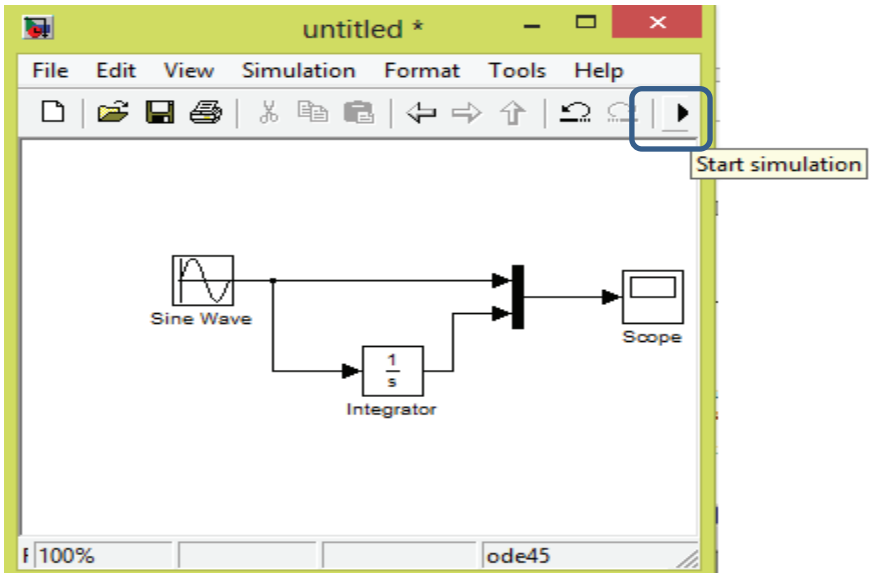


Gambar 10.9. Menu Simulasi



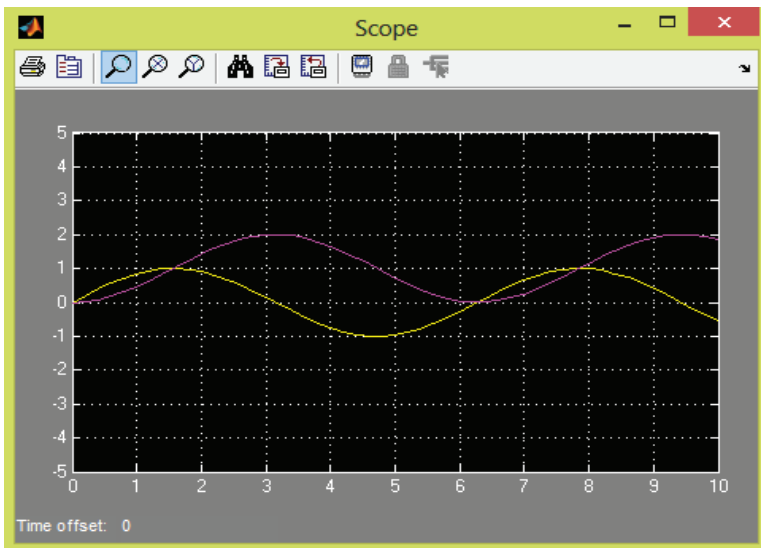
Gambar 10.9. Pengaturan menu simulasi

Keluar dari Configuration parameter dengan klik tombol OK. Pilih **Start Simulation** untuk untuk melihat output simulasi dari blok Scope.



**Gambar 10.9.** Jendela Simulink baru

Hasil simulasi terlihat seperti gambar berikut.



**Gambar 10.9.** Jendela Simulink baru

Untuk mengatur skala grafik secara otomatis klik **Autoscale**.



**Gambar 10.9.** Jendela Simulink baru

Untuk menyimpan Simulasi klik **Save as** dari menu **File** dan ketik nama file dan lokasi

Pada bagian selanjutnya akan diberikan beberapa contoh penggunaan Simulink untuk menyelesaikan beberapa kasus dalam Teknik Kimia.

### **Contoh 10.1 Persamaan Diferensial**

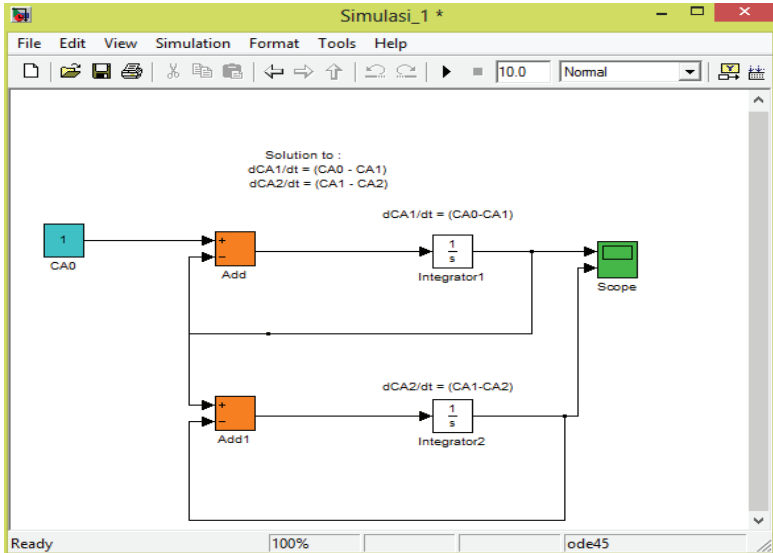
Selesaikan dua persamaan diferensial berikut dengan Simulink.

$$\frac{dC_{A1}}{dt} = C_{A0} - C_{A1} \quad (10.1.1)$$

$$\frac{dC_{A2}}{dt} = C_{A1} - C_{A2} \quad (10.1.2)$$

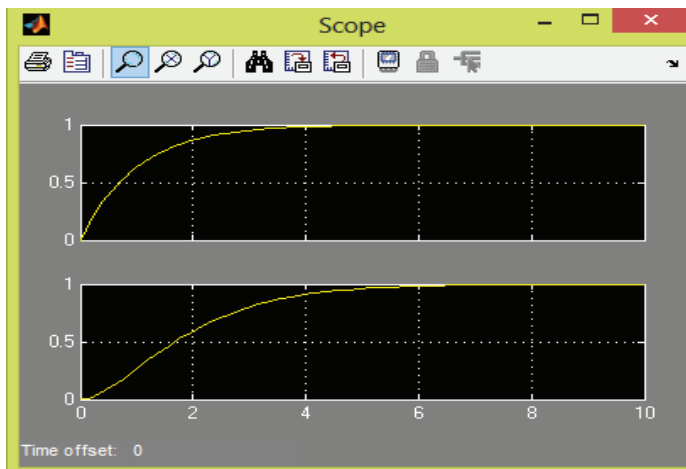
## Penyelesaian :

Untuk menyelesaikan kasus di atas, lakukan langkah yang telah dijelaskan di atas sehingga dihasilkan model blok Simulink seperti Gambar 10.1.1.



**Gambar 10.1.1.** Model blok Simulink untuk Contoh 10.1

Hasil Simulasi dapat dilihat pada Gambar di bawah.



**Gambar 10.1.2.** Hasil plot Simulink untuk Contoh 10.1

### Contoh 10.2 Persamaan Diferensial

Selesaikan seperangkat persamaan diferensial order biasa berikut dengan konsentrasi  $A$  awal 1 mol/L dan konsentrasi awal  $B$  adalah nol.

$$\frac{dC_A}{dt} = -2C_A + C_B \quad (10.2.1)$$

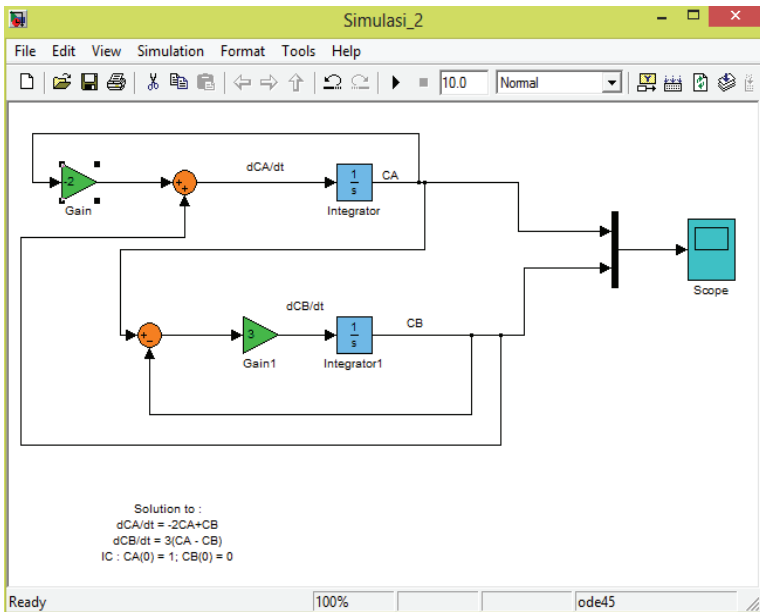
$$\frac{dC_B}{dt} = 3(C_A - C_B) \quad (10.2.1) \quad (10.2.1)$$

$$C_A(0) = 1; C_B(0) = 0$$

#### Penyelesaian :

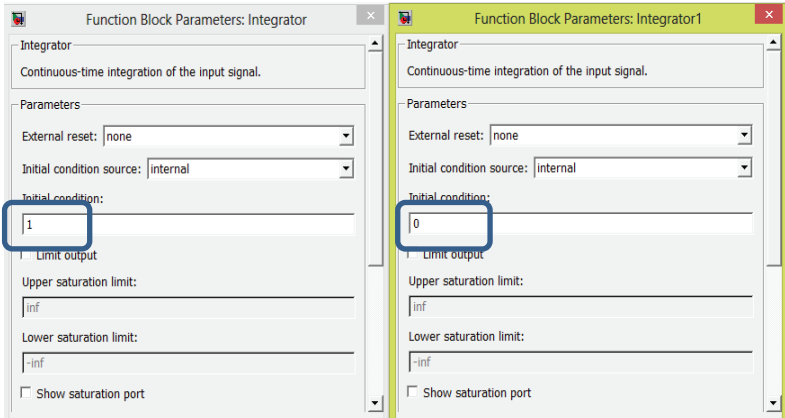
Dengan cara yang sama dengan langkah pada Contoh 10.1 dihasilkan model blok Simulink seperti Gambar 10.2.1.

Selanjutnya klik dua kali pada blok **Integrator** dan **Integrator 1** dan atur initial condition masing-masing dengan nilai 1 dan 0, seperti ditunjukkan Gambar 10.2.2



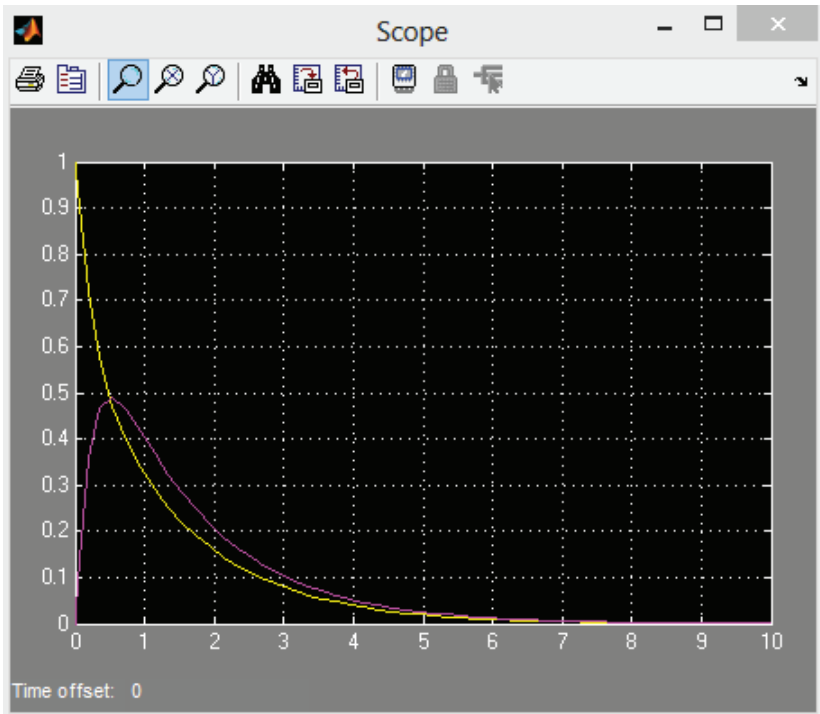
Gambar 10.2.1. Model Simulink untuk Contoh 10.2





**Gambar 10.2.2.** Pengaturan initial condition

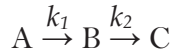
Klik Start Simulation dan selanjutnya klik blok **Scope** dua kali untuk melihat output simulasi. Grafik hasil simulasi dari Scope terlihat seperti gambar di bawah.



**Gambar 10.2.3.** Hasil plot Simulink untuk Contoh 10.2

### Contoh 10.3 Reaksi Multiple dalam Reaktor Batch

Berikut diberikan reaksi seri :



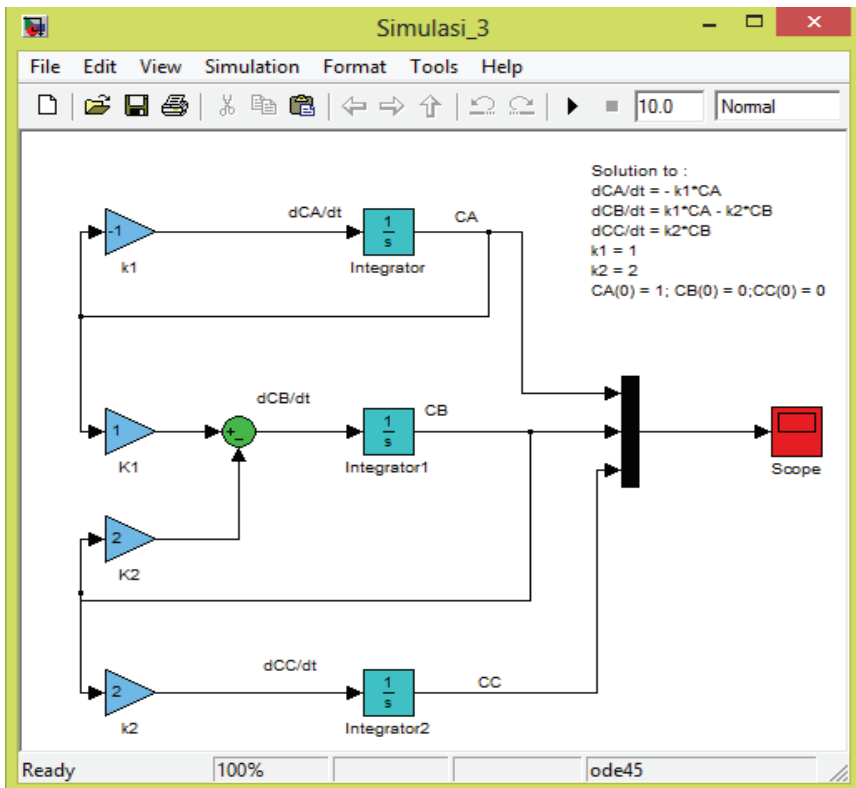
$$\frac{dC_A}{dt} = -k_1 C_A \quad (10.3.1)$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B \quad (10.3.2)$$

$$\frac{dC_C}{dt} = k_2 C_B \quad (10.3.3)$$

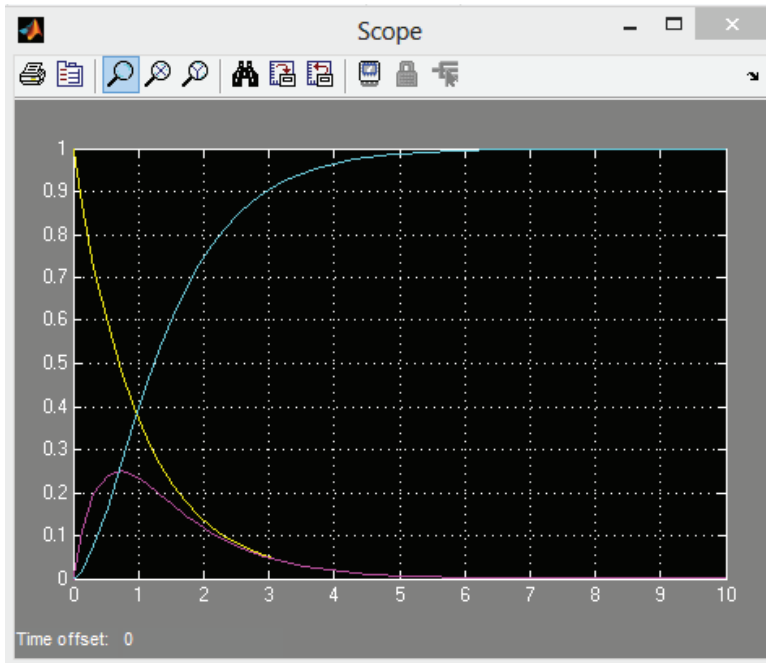
#### Penyelesaian :

Blok Simulink untuk kasus di atas ditunjukkan gambar di bawah.



Gambar 10.3.1. Model Simulink untuk Contoh 10.3

Dengan mengikuti cara sebelumnya akan diperoleh output simulasi dari blok Scope seperti gambar di bawah.



**Gambar 10.3.2.** Hasil plot Simulink untuk Contoh 10.3

### Contoh 10.4 Reaksi Katalisis Enzim

Suatu enzim  $E$ , mengkatalisis substrat  $S$  untuk membentuk produk  $P$  dengan formasi kompleks intermediat  $ES$ , seperti ditunjukkan reaksi berikut :



Gunakan neraca massa untuk menyederhanakan reaksi enzimatik untuk memperoleh persamaan diferensial yang menggambarkan dinamika reaksi. Gunakan nilai awal berikut dan konstanta laju berikut untuk mengintegrasikan persamaan diferensial dan plot profil waktu untuk semua variabel dalam model:

Kondisi awal :  $[S]_0 = 1,0 \mu\text{M}$   $[E]_0 = 0,1 \mu\text{M}$

$$[ES]_0 = 0 \quad [P]_0 = 0$$

Konstanta :  $k_1 = 0,1 (\mu\text{M})^{-1} \text{ s}^{-1}$   $k_{-1} = 0,1 \text{ s}^{-1}$   $k_2 = 0,3 \text{ s}^{-1}$

Tentukan waku (dalam detik) reaksi untuk mencapai 99,9% substrat terkonversi.

**Penyelesaian :**

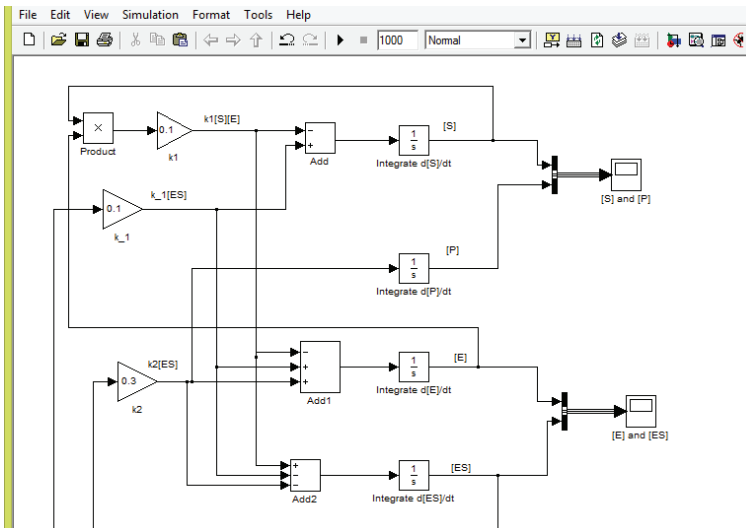
$$\frac{d[S]}{dt} = -k_1[S][E] + k_{-1}[ES] \tag{10.4.1}$$

$$\frac{d[E]}{dt} = -k_1[S][E] + k_{-1}[ES] + k_2[ES] \tag{10.4.2}$$

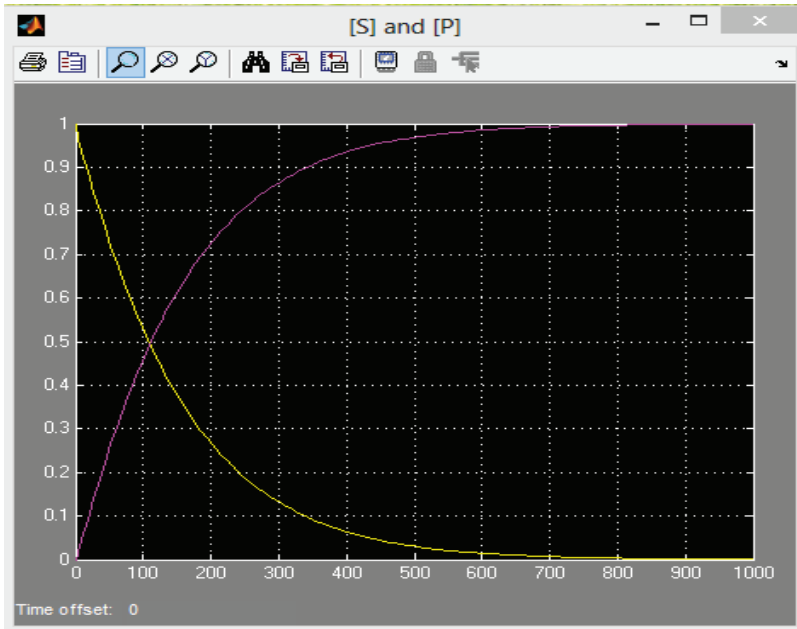
$$\frac{d[ES]}{dt} = k_1[S][E] - k_{-1}[ES] - k_2[ES] \tag{10.4.3}$$

$$\frac{d[P]}{dt} = k_2[ES] \tag{10.4.4}$$

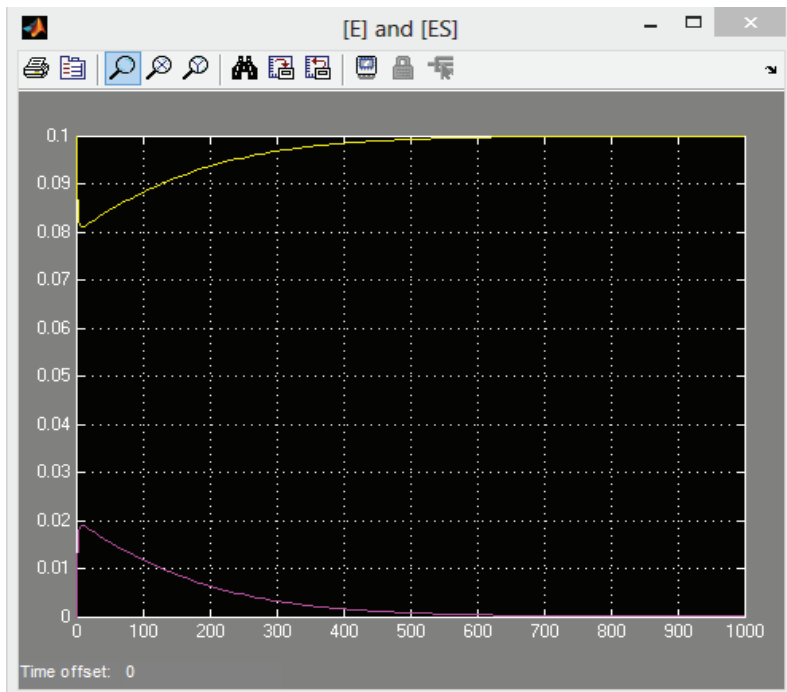
Kita integralkan dari 0 sampai 1000 detik dengan mengatur batas integrasi pada *Configuration parameter*.



**Gambar 10.4.1.** Model Simulink untuk Contoh 10.4



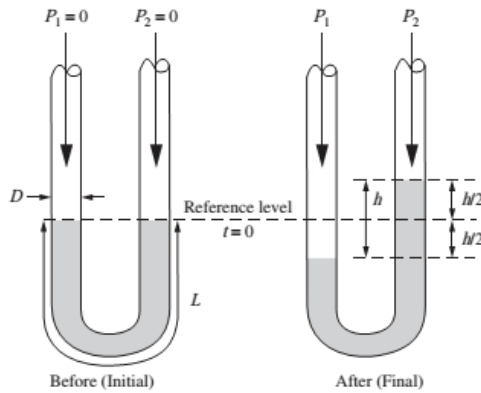
**Gambar 10.4.2.** Hasil plot Simulink untuk [S] dan [P]



**Gambar 10.4.3.** Hasil plot Simulink untuk [E] dan [S]

### Contoh 10.5. Respon Dinamis Open Loop Sistem Oder Dua

Misalkan suatu manometer seperti ditunjukkan seperti Gambar 10.5.1 digunakan untuk mengukur perbedaan tekanan antara dua instrument pada suatu garis udara. Fluida kerja dalam manometer adalah air. Tentukan respon manometer terhadap *step change* dalam tekanan yang melewati kaki manometer.



**Gambar 10.5.1.** Tekanan manometer

Data :

$$L : 200 \text{ cm}, \quad D : 0,11 \text{ cm}; 0,21 \text{ cm}; 0,32 \text{ cm}$$

$$g : 980 \text{ cm/s}^2$$

$$\mu : 1 \text{ cP} = 0,01 \text{ g/cm.s}$$

$$\rho : 1,0 \text{ g/cm}^3$$

$$\frac{\Delta P}{\rho g} = \begin{cases} 0 & \text{untuk } t < 0 \\ 10 & \text{untuk } t \geq 0 \end{cases}$$

**Penyelesaian :**

Persamaan diferensial untuk manometer adalah :

$$\frac{2L}{3g} \frac{d^2 h}{dt^2} + \frac{16\mu L}{\rho D^2 g} \frac{dh}{dt} + h = \frac{P_1 - P_2}{\rho g} = \frac{\Delta P}{\rho g} \quad (10.5.1)$$

Dalam bentuk variabel deviasi yang ditrasformasi menjadi:

$$\tau^2 s^2 Y(s) + 2\zeta \tau Y(s) + Y(s) = X(s) \quad (10.5.2)$$

dengan :

$$Y = h - h_s \text{ dan}$$

$$X = \left( \frac{\Delta P}{\rho g} - \frac{\Delta P}{\rho g} \right)_s$$

$$\tau = \sqrt{\frac{2L}{3g}}$$

$$\xi = \frac{8\mu}{\rho D^2} \sqrt{\frac{3L}{2g}}$$

*Time constant* untuk manometer :

$$\tau = \sqrt{\frac{2L}{3g}} = \sqrt{\frac{2(200 \text{ cm})}{3(980 \text{ cm}^2/\text{s})}} = 0,369 \text{ s}$$

*Damping coefficient* untuk tiga diameter tube yang berbeda :

$$\xi = \frac{8\mu}{\rho D^2} \sqrt{\frac{2L}{3g}} = \frac{8[0,001 \text{ g/cm} \cdot \text{s}]}{(1,0 \text{ g/cm}^3)(D^2)} \sqrt{\frac{3(200 \text{ cm})}{2(980 \text{ cm}^2/\text{s})}} = \frac{0,0443}{D^2}$$

Diameter (cm)	$\xi$
0,11	3,66
0,21	11,00
0,31	0,46

Dari data di atas jelas kita memiliki satu *underdamped system* ( $\xi < 1$ ), satu *critically damped system* ( $\xi = 1$ ) dan satu *overdamped system* ( $\xi > 1$ ).

Untuk memperoleh respon fungsi transfer  $Y/X$  terhadap input  $X$  digunakan MATLAB dan Simulink .

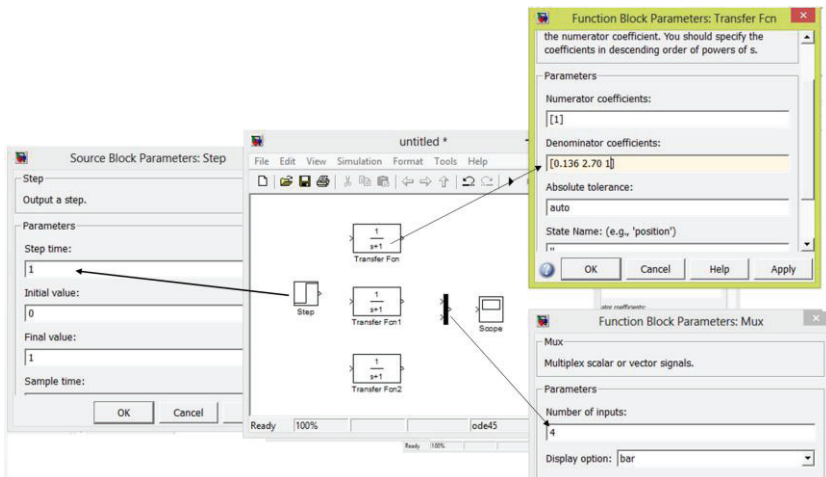
$$\frac{Y(s)}{X(s)} = \frac{1}{\tau^2 s^2 + 2\zeta\tau s + 1} \quad \text{dan} \quad X = \frac{10}{s}$$

Tiga fungsi transfer yang diperlukan sebagai berikut :

Diameter (cm)	$\tau$	$\zeta$	$\tau^2$	$2 \zeta \tau$	Fungsi transfer
0,11	0,369	3,66	0,136	3,66	$\frac{1}{0,136s^2 + 2,700s + 1}$
0,21	0,369	11,00	0,136	11,00	$\frac{1}{0,136s^2 + 0,738s + 1}$
0,31	0,369	0,46	0,136	0,46	$\frac{1}{0,136s^2 + 0,340s + 1}$

Penyelesaian fungsi transfer di atas dengan Model Simulink ditunjukkan pada langkah di bawah.

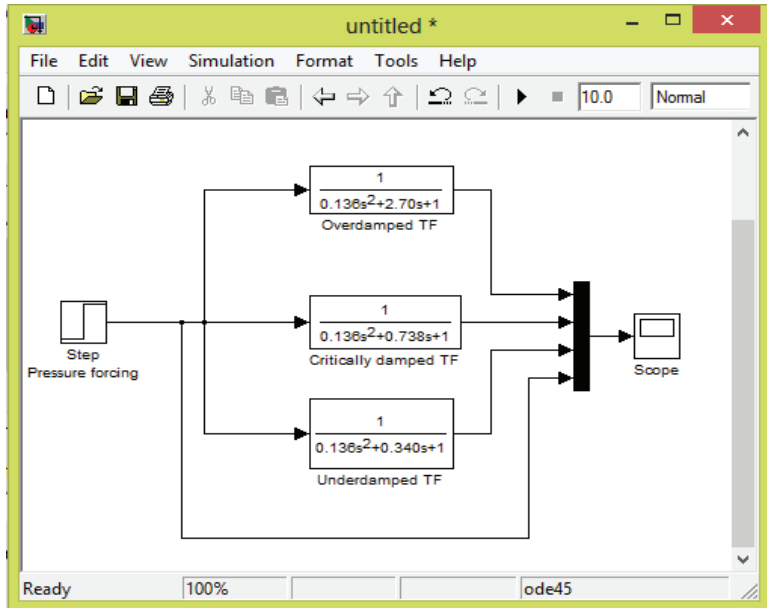
Pertama, Pada menu baru di lembar kerja drag dan drop beberapa blok. Pada masing-masing klik dua kali sehingga tampak seperti gambar di bawah.



**Gambar 10.5.2.** Pengaturan parameter blok

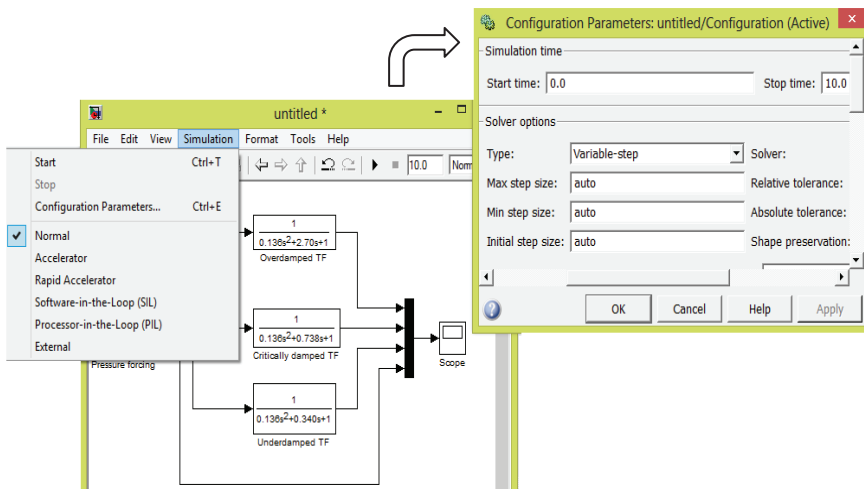
Kedua, hubungkan semua blok sehingga diperoleh sistem seperti gambar di bawah.





**Gambar 10.5.3.** Model Simulink untuk contoh soal 10.5

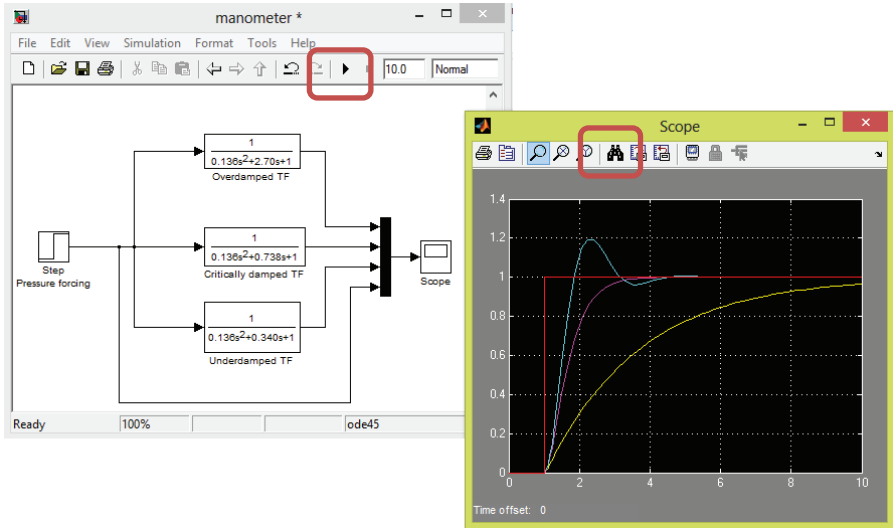
Ketiga, pada menu simulasi diatur **stop time** 10 menit.



**Gambar 10.5.4.** Pengaturan parameter simulasi

Keempat, jalankan simulasi dengan menekan menu **Start** dan klik **Autoscale** pada tampilan scope. Diperoleh output simulasi seperti

gambar di bawah.



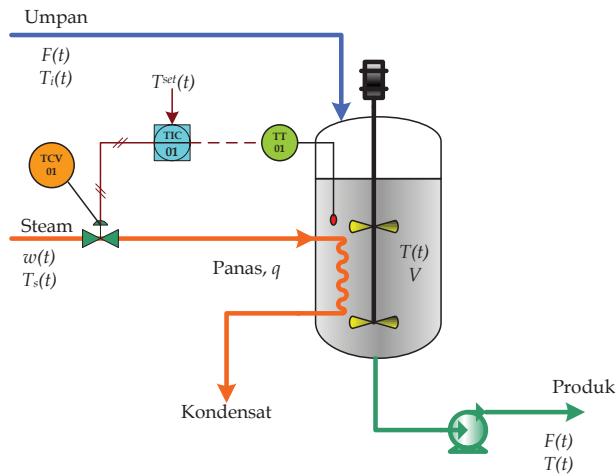
**Gambar 10.5.5.** Hasil plot simulink untuk Contoh 10.5

### Contoh 10.6 Kontrol Temperatur Pemanas Tangki Alir Berpengaduk

Tangki berpengaduk seperti Gambar 10.6.1, digunakan untuk memanaskan aliran proses sehingga pencampuran komponen mencapai seragam. Kontrol temperatur adalah penting karena temperatur tinggi cenderung mendekomposisi produk, sementara temperatur rendah menghasilkan pencampuran yang kurang sempurna. Tangki dipanaskan dengan steam di sisi koil. Digunakan *Proportional Integral Derivative (PID) Controller* untuk mengendalikan temperatur dalam tangki dengan memanipulasi posisi valve steam. Turunkan blok diagram lengkap dan fungsi transfer untuk data desain berikut:

**Proses.** Umpan memiliki densitas,  $\rho$  sebesar  $68 \text{ lb/ft}^3$  dan kapasitas panas,  $C_p$  adalah  $0,8 \text{ Btu/lb-}^\circ\text{F}$ . Volume cairan dalam tangki dijaga konstan pada  $120 \text{ ft}^3$ . Koil terdiri atas pipa dengan panjang  $205 \text{ ft}$ , ukuran  $4 \text{ inchi}$  schedule 40 yang

beratnya 10,8 lb/ft, memiliki kapasitas panas 0,12 Btu/lb-°F dan diameter luar 4,5 in. Koefisien tansfer panas keseluruhan  $U$ , berdasarkan luas area bagian luar koil, dan diperkirakan 2,1 Btu/min-ft<sup>2</sup>-°F. Steam yang tersedia adalah steam jenuh pada tekanan 30 psia, dapat diasumsikan bahwa panas laten kondensasi adalah konstan pada 966 Btu/lb. Dapat juga diasumsikan bahwa temperatur masuk  $T_i$  adalah konstan.



**Gambar 10.6.1** Kontrol temperatur pemanasan tangki berpengaduk

**Kondisi Desain.** Aliran umpan  $F$  pada kondisi desain adalah 15 ft<sup>3</sup>/min, dan temperaturnya  $T_i$  adalah 100°F. Isi tangki harus dijaga pada temperatur 150°F. Kemungkinan gangguan adalah perubahan laju alir dan temperatur umpan.

**Sensor Temperatur dan Transmitter.** Sensor temperatur memiliki kisaran kalibrasi 100-200°F dan *time constant*,  $\tau_r$  adalah 0,75 menit.

**Control Valve.** Control valve dirancang untuk 100% overkapasitas dan perubahan hilang tekanan dapat diabaikan. Valve sama dengan persentase valve dengan

parameter perubahan,  $\alpha$  adalah 50. Aktuator memiliki *time constant*,  $\tau_v$  sebesar 0,2 menit.

### Penyelesaian.

Pendekatan dilakukan dengan menurunkan persamaan yang mendeskripsikan perilaku dinamis dalam tangki, control valve, sensor/transmitter dan controller. Selanjutnya kita akan mentransformasi Laplace persamaan untuk memperoleh loop blok diagram.

**Proses.** Neraca energi untuk cairan dalam tangki, asumsi panas yang hilang diabaikan, pencampuran sempurna, volume konstan dan densitas konstan diperoleh persamaan :

$$V\rho C_p \frac{dT}{dt} = F\rho C_p (T_i - T_{ref}) + UA(T_s - T) - F\rho C_p (T - T_{ref}) \quad (10.6.1)$$

Neraca energi pada koil, asumsi bahwa koil pada temperatur sama seperti steam, dihasilkan persamaan :

$$C_M \frac{dT_s}{dt} = w\lambda - UA(T_s - T) \quad (10.6.2)$$

Untuk proses steady state:

$$0 = F\rho C_p (\bar{T}_i - T_{ref}) + UA(\bar{T}_s - \bar{T}) - F\rho C_p (\bar{T} - T_{ref}) \quad (10.6.3)$$

$$0 = \bar{w}\lambda - UA(\bar{T}_s - \bar{T}) \quad (10.6.4)$$

Persamaan (10.6.1) dikurang persamaan (10.6.3) diperoleh persamaan :

$$V\rho C_p \frac{d(T - \bar{T})}{dt} = F\rho C_p [(T_i - \bar{T}_i) - (T - \bar{T})] + UA[(T_s - \bar{T}_s) - (T - \bar{T})] \quad (10.6.5)$$

Persamaan (10.6.2) dikurang persamaan (10.6.4) diperoleh persamaan :

$$C_M \frac{d(T_s - \bar{T}_s)}{dt} = W\lambda - UA[(T_s - \bar{T}_s) - (T - \bar{T})] \quad (10.6.6)$$

Variabel deviasi :

$$T' = (T - \bar{T})$$

$$T'_s = (T_s - \bar{T}_s)$$

$$W = (w - \bar{w})$$

Persamaan (10.6.5) menjadi :

$$V\rho C_p \frac{dT'}{dt} = F\rho C_p (T'_i - T') + UA(T'_s - T') \quad (10.6.7)$$

$$V\rho C_p \frac{dT'}{dt} + (F\rho C_p + UA)T' = F\rho C_p T'_i + UAT'_s \quad (10.6.8)$$

$$\frac{V\rho C_p}{(F\rho C_p + UA)} \frac{dT'}{dt} + T' = \frac{F\rho C_p}{(F\rho C_p + UA)} T'_i + \frac{UA}{(F\rho C_p + UA)} T'_s \quad (10.6.9)$$

$$\tau \frac{dT'}{dt} + T' = K_1 T'_i + K_2 T'_s \quad (10.6.10)$$

Dimana :

$$\tau = \frac{V\rho C_p}{(F\rho C_p + UA)} \quad (10.6.11)$$

$$K_1 = \frac{F\rho C_p}{(F\rho C_p + UA)} \quad (10.6.12)$$

$$K_2 = \frac{UA}{(F\rho C_p + UA)} \quad (10.6.13)$$

Transformasi lapalace untuk persamaan (10.6.10)

$$T'(s) = \frac{K_1}{\tau s + 1} T'_i(s) + \frac{K_2}{\tau s + 1} T'_s(s) \quad (10.6.13)$$

Persamaan (10.6.6) menjadi :

$$C_M \frac{dT'_s}{dt} = W\lambda - UA(T'_s - T') \quad (10.6.14)$$

$$C_M \frac{dT'_s}{dt} + UAT'_s = W\lambda + UAT' \quad (10.6.15)$$

$$\frac{C_M}{UA} \frac{dT'_s}{dt} + T'_s = \frac{\lambda}{UA} W + T' \quad (10.6.16)$$

$$\tau_c \frac{dT'_s}{dt} + T'_s = K_w W + T' \quad (10.6.17)$$

Dimana :

$$\tau_c = \frac{C_M}{UA} \quad (10.6.18)$$

$$K_w = \frac{\lambda}{UA} \quad (10.6.19)$$

Transformasi lapalace untuk persamaan (10.6.17)

$$T'_s(s) = \frac{K_w}{\tau_c s + 1} W(s) + \frac{1}{\tau_c s + 1} T'(s) \quad (10.6.20)$$

**Sensor Temperatur Dan Transmitter.** Elemen pengukur temperatur, yang yang membaca suhu dan mengirimkan sinyal  $T_m$  ke controler, dapat menunjukkan dynamic lag. Fungsi transfer untuk sensor dan transmitter adalah:

$$\frac{T'_m(s)}{T'(s)} = G_m(s) = \frac{1}{\tau_m s + 1} \quad (10.6.21)$$

**Control valve.** Valve pneumatik selalu mempunyai dynamic lag, yang artinya posisi sistem tidak merespon secara instan terhadap suatu perubahan dalam tekanan dari controller. Hubungan antara aliran dengan tekanan valve untuk linier valve sering dinyatakan dengan fungsi transfer untuk control valve order satu adalah :

$$\frac{W(s)}{M(s)} = G_v(s) = \frac{K_v}{\tau_v s + 1} \quad (10.6.22)$$

**Proporsional Integral Deriative (PD) Controller.** Kontrol mode ini adalah kombinasi mode propotional, integral dan derivatif dan diberikan dengan persamaan :

PID Controller

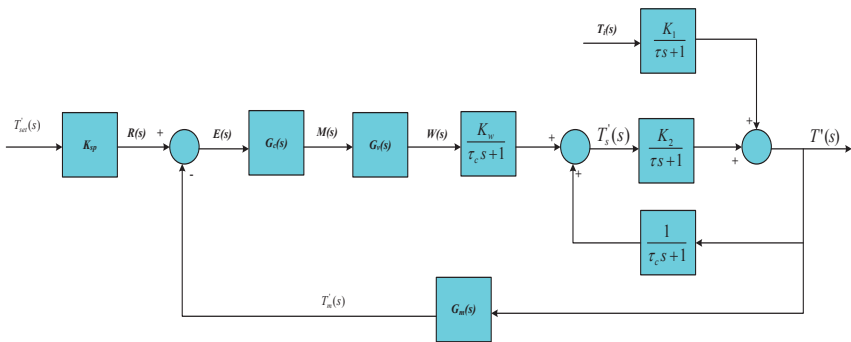
$$p = K_c \varepsilon + \frac{K_c}{\tau_I} \int_0^t \varepsilon dt + K_c \tau_D \frac{d\varepsilon}{dt} + p_s \quad (10.6.23)$$

Dalam hal ini,  $K_c$  adalah *controller gain*,  $\tau_I$  adalah *integral time* dan  $\tau_D$  adalah *derivative time*. Ketiga nilai tersebut dapat diatur di dalam controller. Fungsi transfer untuk controller ini adalah:

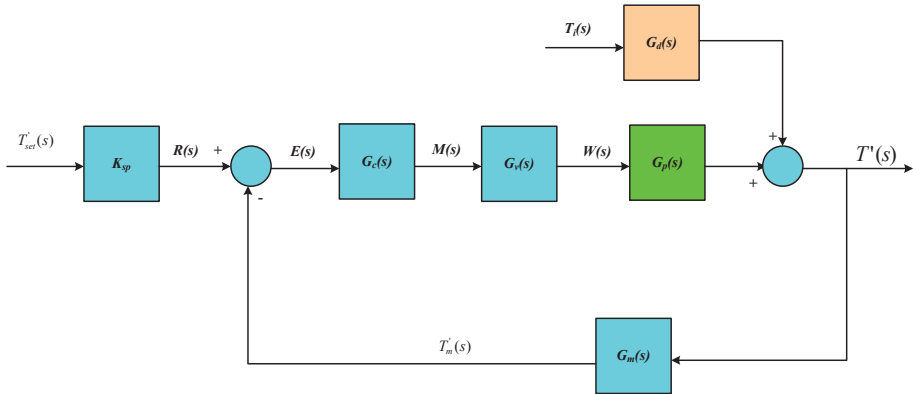
Fungsi Transfer PID controller

$$\frac{M(s)}{\varepsilon(s)} = G_c(s) = K_c \left( 1 + \frac{1}{\tau_I s} + \tau_D s \right) \quad (10.6.24)$$

**Blok Diagram Loop.** Gambar 10.6.2 menunjukkan blok diagram lengkap untuk control loop. Semua fungsi transfer dalam diagram telah diturunkan. Dengan menggunakan aturan untuk memodifikasi blok diagram, diperoleh diagram yang lebih sederhana seperti Gambar 10.6.3.



**Gambar 10.6.2** Blok diagram loop kontrol temperatur



**Gambar 10.6.3** Penyerderhanaan loop kontrol temperatur

Fungsi Transfer dalam diagram adalah:

$$G_d(s) = \frac{K_1(\tau_c s + 1)}{(\tau s + 1)(\tau_c s + 1) - K_2} \quad (10.6.25)$$

$$G_p(s) = \frac{K_w K_2}{(\tau s + 1)(\tau_c s + 1) - K_2} \quad (10.6.26)$$

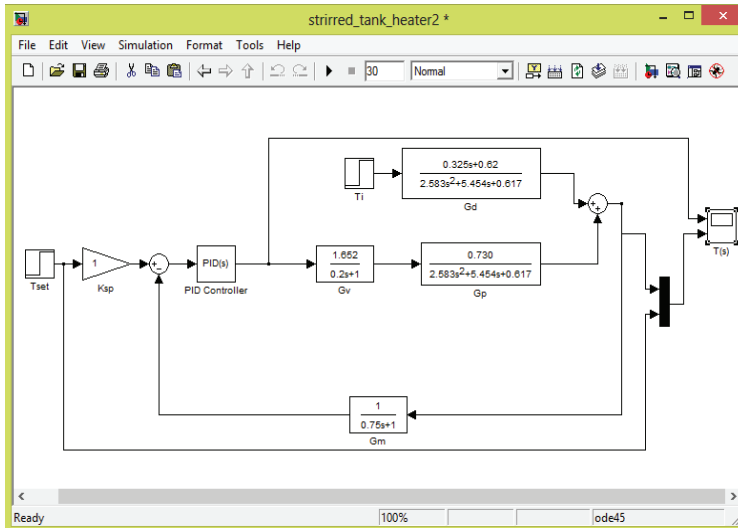
Nilai numeris untuk parameter dalam fungsi transfer diberikan pada Tabel 10.6.1 yang dihitung dari data yang diberikan dalam soal.

Tabel 10.6.1 Nilai parameter untuk Contoh 10.6.1

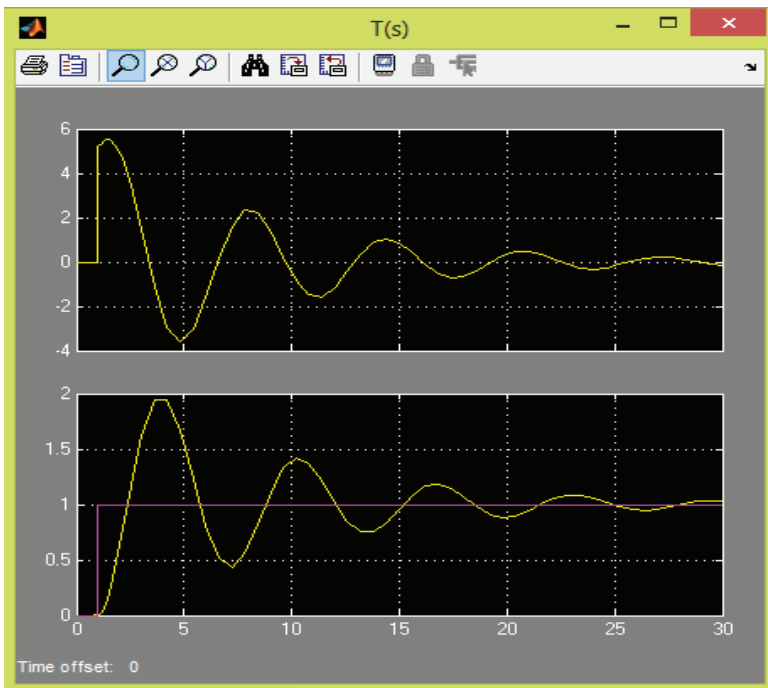
$A$	$= 241,5 \text{ ft}^2$	$\tau$	$= 4,93 \text{ menit}$
$C_M$	$= 265,7 \text{ Btu}/^\circ\text{F}$	$\tau_c$	$= 0,524 \text{ menit}$
$K_I$	$= 0,62^\circ\text{F}/(\text{ft}^3/\text{min})$	$K_w$	$= 1,905 \text{ }^\circ\text{F}/(\text{lb}/\text{min})$
$K_2$	$= 0,383 \text{ }^\circ\text{F}/^\circ\text{F}$	$K_T$	$= 1,0 \text{ \%TO}/^\circ\text{F}$
$K_v$	$= 1,652$ $(\text{lb}/\text{min})/\text{\%CO}$	$K_{sp}$	$= 1,0\% \text{TO}/^\circ\text{F}$
$\tau_m$	$= 0,75 \text{ menit}$	$\tau_T$	$= 0,20 \text{ menit}$

Blok diagram Simulink untuk menyelesaikan soal di atas dan plot hasil simulasi simulink ditunjukkan Gambar 10.6.4. dan Gambar 10.6.5.





Gambar 10.6.4 Blok diagram Simulink untuk kontrol temperatur



Gambar 10.6.5 Plot hasil simulasi Simulink untuk kontrol temperatur

## 10.3 RANGKUMAN

- a. Simulink adalah paket software untuk pemodelan, simulasi dan analisis sistem dinamis
- b. Pemodelan Simulink menyediakan *Graphical User Interface* (GUI) untuk membuat model seperti blok diagram, menggunakan operasi klik dan drag mouse.
- c. Pemodelan Simulink banyak digunakan dalam bidang Teknik Kimia untuk simulasi dan analisis sistem dinamis dan kontrol proses kimia.

## 10.4 RUJUKAN

- a. Mathworks. 1999. *Simulink-Dynamic System Simulation for MatLab Version 3*. <http://www.mathworks.com>
- b. Donald R. Coughanowr and Steven E. LeBlanc. 2008. *Process Systems Analysis And Control 3rd ed*. Mc Graw Hill. New York.
- c. Karis, Steven T. 2008. *Introduction to Simulink with Engineering Applications 3rd. ed.*, Orchard publication. [www.orchardpublications.com](http://www.orchardpublications.com)
- d. Carlos A. Smith and Amando B Corripio. 1997. *Principles and Practice of Automatic Process Control*. John Wiley . New York.