

# ALGORITMA MODULAR UNTUK SISTEM PROSES DENGAN *RECYCLE* - KEADAAN TUNAK PADA PERANGKAT LUNAK BEBAS BIAYA LISENSI

Setia Budi Sasongko

Jurusan Teknik Kimia Fakultas Teknik UNDIP  
Jl. Prof. Sudarto, SH, Tembalang-Semarang 50239; Telp.(024) 7460058  
E-mail: [budisas@undip.ac.id](mailto:budisas@undip.ac.id)

## Abstract

*Computer simulation can be developed to comprehend a system process characteristic and can be used to train the operator of process. The first problem of programming is high price of the licence. Using the FOC (free of charge) software solve of the previous problem such as Scilab and Open Office which comparable with Matlab and Excel. The objectives of this article develop process system programming which has recycle system and user friendly programming. Modular algorithm programming is introduced to solve recycle process system. Graphical User Interface (GUI) also introduces to make user friendly programming. Execution of the program performs stable and correct results.*

**Key words** : *free of charge software, modular algorithm, open office-calc, recycle, scilab*

## Abstrak

*Simulasi komputer pada sistem proses dikembangkan sebagai bagian untuk memahami karakteristik dari sistem dan diharapkan dapat digunakan sebagai training untuk operator sistem proses. Permasalahan yang mendasar dalam mengembangkan simulasi sistem proses adalah perangkat lunak yang relatif mahal. Saat ini, masalah tersebut dapat diatasi dengan menggunakan perangkat lunak yang bersifat bebas biaya, dalam hal ini digunakan Scilab dan open Office Calc yang ekuivalen dengan Matlab dan Excel. Pengembangan pemrograman dengan algoritma modular untuk sistem dengan recycle serta program yang bersifat user friendly merupakan tujuan dan sebagai bagian dari metodologi penyelesaian dari sistem proses. Eksekusi dari program beserta tahap penyelesaian menunjukkan hasil yang stabil dan cukup akurat.*

**Kata kunci** : *perangkat lunak gratis, algoritma modular, open office-calc, recycle, scilab*

## PENDAHULUAN

Salah satu tugas bidang enjinering kimia adalah merancang sistem proses. Perancangan merupakan suatu aktivitas yang cukup kompleks dan bervariasi. Saat ini perancangan dengan menggunakan bantuan komputer atau perangkat lunak siap pakai sudah cukup banyak dan berkembang, seperti Aspen-Hysys, ChemCAD dan lain sebagainya. Dengan menggunakan perangkat lunak tersebut, proses analisis untuk suatu sistem akan berlangsung dengan cepat. Permasalahan yang ada dengan menggunakan perangkat lunak tersebut adalah dibutuhkan biaya yang relatif mahal. Selain itu, dengan menggunakan perangkat lunak siap pakai terkadang filosofi dari perancangan alat menjadi berkurang. Hal ini kurang menguntungkan khususnya dalam bidang pendidikan.

Oleh karenanya, artikel ini bertujuan untuk mengembangkan ide simulasi sistem proses yang sesuai

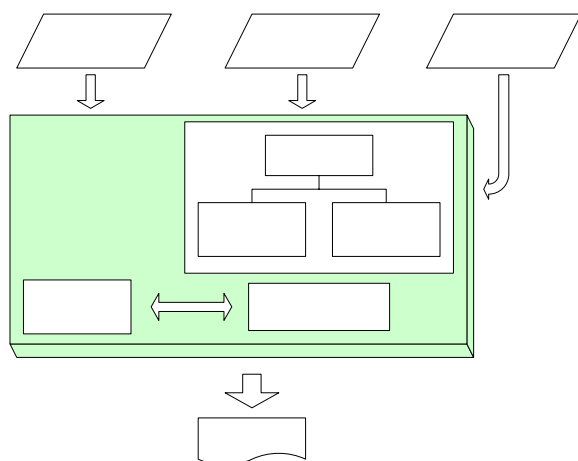
dengan kebutuhan dari pengguna akhir (*user friendly*) dengan menggunakan perangkat lunak berlisensi bebas biaya (*free of charge, FOC*), yaitu Scilab dan open Office Calc (oO-Calc). Sebagai pembatas masalah pada artikel ini, hanya akan dibahas penyusunan sistem proses berdasarkan neraca bahan (massa) saja.

## Fundamental

Menurut Biegler *et al* (1997) metoda penyelesaian dari simulasi untuk sistem proses khususnya untuk neraca massa dan energi dapat dibedakan menjadi dua cara, yaitu orientasi persamaan total (*equation oriented*) dan modular. Pada penyelesaian dengan metoda orientasi persamaan total, maka semua persamaan proses (neraca massa dan energi, unjuk kerja dari peralatan, karakteristik termodinamika, transport, kinetika dan

hubungan persamaan yang lain) pada sistem yang ditinjau harus disusun secara keseluruhan kemudian diselesaikan secara simultan. Penyelesaian dengan metoda pertama, akan didapat persamaan dengan jumlah yang relatif besar dan banyak. Selain itu, penyelesaian dengan sistem berorientasi persamaan total tersebut bersifat spesifik, artinya dengan merubah sistem misalkan diagram alir proses, maka seluruh persamaan akan berubah. Sedangkan penyelesaian dengan metoda yang ke dua, yaitu sistem modular, penyelesaiannya dilakukan secara bertahap dimana tiap tahapan pada pemrograman disebut dengan modul atau dalam bahasa pemrograman sebagai sub-rutin atau bagian sub-program. Dengan demikian, pengguna akhir pada saat menyusun sistem proses, pada dasarnya merupakan penggabungan dari sub-program yang relatif kecil sehingga apabila ada kesalahan program penyelesaian atau pencarian kesalahan tersebut menjadi lebih mudah. Penggabungan dari modul-modul program pada artikel ini disebut dengan algoritma modular. Penyelesaian akan menjadi lebih rumit dan menarik, apabila dalam sistem proses terdapat aliran daur ulang (*recycle*).

Proses penyelesaian dari sistem proses secara umum, diawali dengan menentukan semua bahan yang ada dalam sistem proses baik bahan baku, produk antara, produk akhir maupun bahan inert. Dilanjutkan dengan mencari sifat-sifat kimia, fisika berdasarkan tinjauan kinetika maupun termodinamika. Kemudian menggambarkan diagram alir proses yang terdiri dari aliran bahan serta satuan operasi (*unit operation*) berupa reaktor, peralatan pemisahan, perubah ukuran bahan, perubah fase, perubah kondisi operasi, pengangkutan (pemindahan) bahan.

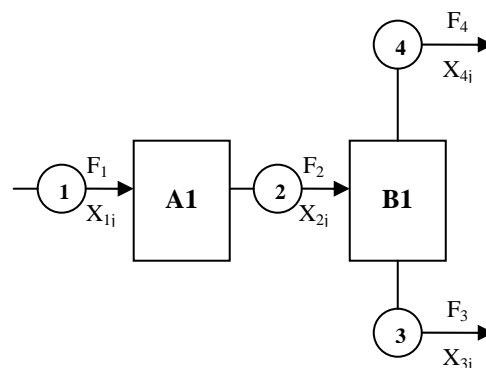


Gambar 1. Hubungan perangkat lunak dengan pengguna

Untuk mendapatkan hasil perhitungan lebih cepat dan akurat, maka dapat digunakan komputer sebagai alat bantu. Sifat-sifat dari bahan maupun perhitungan termodinamika dan kinetika bahan dapat disimpan dan dilakukan dengan bantuan komputer, demikian juga dengan sub rutin dari satuan operasi, urutan perhitungan berdasarkan diagram alir proses. Dengan demikian,

hubungan antara pengguna akhir dengan komputasi serta hasil keluaran dari komputer sebagai bagian dari perangkat lunak, dapat dilihat pada Gambar 1.

Diagram alir proses (DAP) disusun berdasarkan simbol sebagaimana yang ditunjukkan Gambar 2.



Gambar 2. Simbol pada penyusunan diagram alir proses

DAP terdiri dari peralatan atau satuan unit (*unit operation*) dan aliran (*stream*). Tiap aliran baik masuk maupun keluar peralatan ditandai dengan lingkaran yang diberi kode nomor. Pada aliran tersebut, menggambarkan karakteristik total seperti kecepatan aliran,  $F_i$ , dengan indek  $i$  menunjukkan nomor aliran. Selain karakteristik total juga terdapat karakteristik komponen yang memuat informasi berdasarkan aliran dan bahan, seperti fraksi komponen, panas jenis dan lain-lain dengan simbol,  $X_{ij}$ , dengan indek  $j$  sebagai kode nomor bahan dan  $i$  menunjukkan nomor aliran. Sedangkan sebagai satuan operasi (peralatan, *Unit Operation*) diberi kode gabungan huruf dan nomor, misalkan **A1** atau **B1**.

Pada metoda algoritma modular ini, susunan dari karakteristik komponen ditandai dengan indek  $i, j$  dapat disusun dalam bentuk matrik (dua dimensi), dengan baris  $i$  menyatakan nomor aliran dan kolom  $j$  sebagai nomor bahan. Oleh karenanya akan lebih baik apabila digunakan perangkat lunak yang berbasis matrik, karena proses penyelesaiannya akan lebih cepat. Berdasarkan hal tersebut, perangkat lunak Scilab lebih tepat digunakan dalam penyusunan program ini dengan pertimbangan Scilab merupakan program berbasis matrik sebagaimana perangkat lunak matlab®(Rietsch, 2002), selain itu Scilab merupakan perangkat lunak dengan bebas biaya lisensi (Chandler and Roberts, 2002).

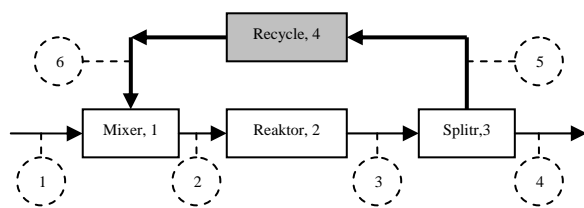
**METODE PENELITIAN**

Kerangka pikir untuk membuat simulasi sistem proses berdasarkan pada Gambar 1. Secara garis besar dapat dibagi menjadi tiga bagian:

1. Masukan dari pengguna yang merupakan informasi dasar dari simulasi:

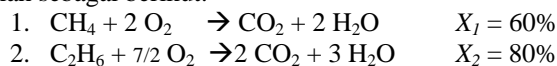
- a. Semua bahan pada sistem proses serta kondisi awal sistem proses
  - b. Susunan dari Diagram Alir Proses (DAP).
2. "Algoritma modular" sebagai bagian dari penyelesaian simulasi sistem proses.
  3. Keluaran sebagai hasil simulasi dalam bentuk tabel atau grafik.

Sebagai studi kasus, disusun Sistem Proses dengan daur ulang yang meliputi sebuah peralatan utama yaitu reaktor stokhiometri dan dua buah peralatan sebagai bagian untuk melakukan proses daur ulang (*recycle*) yaitu pencampur (*mixer*) dan pemisah (*splitter*). Pada proses daur ulang ditambah satu buah alat yang bersifat 'maya' sebagai bagian dari proses perhitungan, sebagaimana Gambar 3.



Gambar 3. Contoh Sistem Proses dengan daur ulang

Deskripsi proses sebagai gambar dari studi kasus ini adalah reaksi pembakaran metana ( $\text{CH}_4$ ) dan etana ( $\text{C}_2\text{H}_6$ ) dengan udara. Diasumsikan reaksi berlangsung dengan konversi ( $X$ ) per *pass* tetap, pada Reaktor Stokhiometri. Reaksi yang terjadi pada reaktor tersebut adalah sebagai berikut:



Sebagian dari hasil keluaran reaktor (aliran 3) dipisahkan dengan *splitter* (aliran 5) dikembalikan (*recycle*) bercampur dengan umpan segar dan dimasukkan ke dalam reaktor kembali. Sedangkan aliran keluaran lain dari *splitter* (aliran 4) sebagai *output* dari sistem. Perbandingan keluaran *splitter* yang di kembalikan dengan yang di keluarkan sebagai produk sebesar  $R_s$  dengan nilai yang diasumsikan tetap. Akan dihitung neraca massa dari tiap bahan (komponen) pada semua aliran dari sistem.

Tahapan pertama, yang perlu dilakukan oleh pengguna akhir dalam menggunakan aplikasi sistem proses ini, adalah:

1. Memilih semua bahan (komponen) pada sistem proses.
2. Memilih peralatan sistem proses  
Dalam pemilihan peralatan dimulai dari peralatan untuk bahan baku masuk, dengan penomoran awal pada peralatan. Pada contoh, mixer nomor alat 1.
3. Menentukan penomoran keluar masuk aliran pada peralatan (pada Gambar 3, nomor dalam lingkaran garis putus-putus).

Pada tahap pertama, program ini menyediakan bank data dari sifat fisis bahan yang disimpan pada perangkat lunak *spread sheet* semacam excel, akan

tetapi disini digunakan *OpenOffice.org Calc* (oO-Calc) bersifat bebas biaya, yang disimpan ekstensi xls. Sebagai contoh, Gambar 4 merupakan tampilan bank data dari bahan (kolom 2) dari Berat Molekul (kolom 3), titik leleh (kolom 4) dan seterusnya, yang disimpan dengan nama file: DataFisbs.xls.

No.	KOMPONEN	BM	MP (K)	BP (K)	Tcr (K)
1	ACETALDEHYDE	44.054	150.2	293.6	461.0
2	ACETIC ACID	60.052	289.8	391.1	594.4
3	ACETIC ANHYDRIDE	102.089	199.0	412.0	569.0
4	ACETONE	58.080	178.2	329.4	508.1
5	ACETONITRILE	41.053	229.3	354.8	548.0
6	ACETYL CHLORIDE	78.498	160.2	323.9	500.0
7	ACETYLENE	26.038	192.4	189.2	308.9
8	ACROLEIN	56.064	186.0	326.0	506.0
9	ACRYLIC ACID	72.064	265.0	414.0	615.0
10	ACRYLONITRILE	53.064	189.5	350.5	536.0
11	ALLYL ALCOHOL	58.080	144.0	370.0	545.0
12	ALLYL CHLORIDE	76.526	138.7	318.3	514.0
13	ALLYL CYANIDE	67.091	186.7	392.0	585.0
14	ALPHA-METHYL TYRENE	118.179	0.0	438.5	654.0
15	AMMONIA	17.031	195.4	239.7	405.6
16	ANILINE	93.129	267.0	457.5	699.0
17	ANTHRACENE	178.234	489.7	614.4	883.0

Gambar 4. Bank data dari bahan menggunakan oO-Calc

Pada Gambar 4, untuk memudahkan dalam pengelolaan data-data, maka data dibedakan dalam beberapa *sheet* atau lembar kerja. Untuk membuka data yang dibagi dalam beberapa *sheet* tersebut dilakukan dengan perintah *readxls*. Agar program bersifat lebih interaktif dengan pengguna akhir, maka digunakan perintah "antar muka pengguna (GUI)", dalam hal ini digunakan perintah *x\_mdialog*, *x\_choose*, dengan hasil sebagaimana ditunjukkan pada Gambar 5, 6 dan 7.

Tabel 1. Program masukan bersifat "GUI"

```
//Simulasi Sistem Proses:

txt_awal=['DATA DIAGRAM ALIR PROSES';...
'MOHON PFD DIGAMBAR MANUAL';...
'Pindah tekan TAB';...
'Akhiri dng Finish'];

txt=['Jumlah Komponen= ';
'Jumlah Alat= ';
'Jumlah Aliran = '];

dawal=x_mdialog(txt_awal,txt,{'';''});
Jkomp=evstr(dawal(1))
JAlat=evstr(dawal(2))
JAlir=evstr(dawal(3))
//Pilih Komponen, Buka Bank Data dng ekstensi xls:

LbrKrj=readxls('C:\scijur\DataFisbs.xls');
dtlb2=LbrKrj(2);
dnamakom=dtlb2(:,2);

for ii=1:Jkomp
tulibs1=['BANK DATA EXCEL'; ...
'Pilih Komponen ke: '+ string(ii)+
'dari'+ string(Jkomp)];
nokomp(ii)=x_choose(dnamakom,tulibs1,'selesai');
namkom(ii)=dtlb2(nokomp(ii),2);
BMkomp(ii)=dtlb2(nokomp(ii),3);
end

//Pilih Alat: =====
dAlat=['Mixer'; 'Splitter'; 'Recycle'; ...
'Reaktor Stokhio'; 'Reaktor Kinetik'; ...
```

```
'Destilasi Kilat'; 'Absorber'];

txtalir={'No Aliran Masuk = '; 'No Aliran Kluar = '};
txtdal={'PENOMORAN ALIRAN MASUK DAN KELUAR'};

for ak=1:JAlat

tulisbs2=['BANK PERALATAN';...
'Pilih Alat ke: '+string(ak)+' dari '+string(JAlat)];
pilalt=x_choose(dfalat,tulisbs2,'selesai');

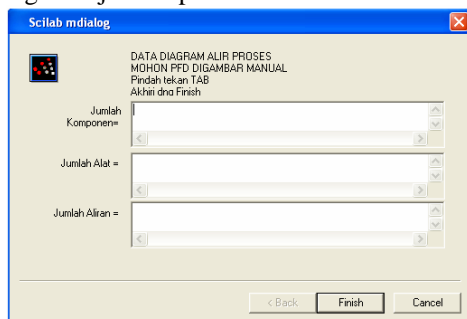
select pilalt
case 1 then getf('C:/scijur/mixer.sci'); UnOp(ak)=pilalt;
case 2 then getf('C:/scijur/spliter.sci');
        UnOp(ak)=pilalt;
case 3 then getf('C:/scijur/reakt.sci'); UnOp(ak)=pilalt;
case 4 then getf('C:/scijur/reaksto.sci'); UnOp(ak)=pilalt;
case 5 then getf('C:/scijur/reakin.sci'); UnOp(ak)=pilalt;
case 6 then getf('C:/scijur/deskit.sci'); UnOp(ak)=pilalt;
        else getf('C:/scijur/abs.sci'); UnOp(ak)=pilalt;
end // akhir pemilihan alat

//Penentuan aliran: =====
dalir=x_mdialog(txtdal,txtalir,['1 6';'2 0']);
msk=evstr(dalir(1));
klr=evstr(dalir(2));
inalir(ak,:)=msk;
otalit(ak,:)=klr;

end
```

Pada Tabel 1 merupakan program utama dengan menggunakan Scilab sebagai tahap pertama dari simulasi sistem proses yang berisi tentang interaksi antara pengguna akhir dengan program berupa:

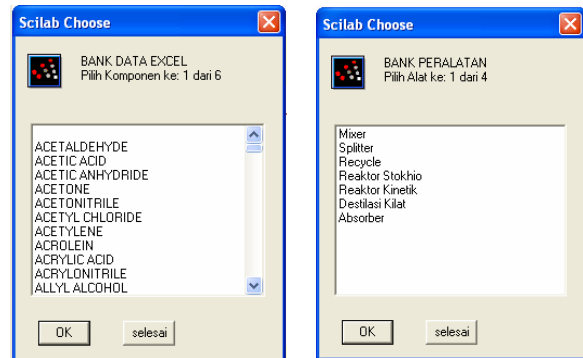
- a. Penentuan jumlah komponen, jumlah alat satuan operasi dan jumlah aliran yang ada pada sistem proses. Proses interaksi antara program dengan pengguna akhir digunakan interface sebagaimana yang ditunjukkan pada Gambar 5.



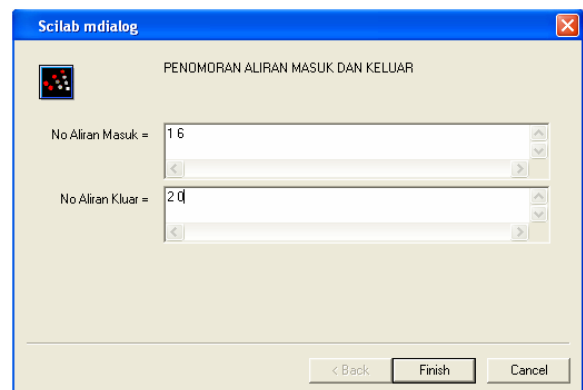
Gambar 5. Penentuan jumlah komponen, alat dan aliran dengan perintah `x_mdialog`.

- b. Pemilihan semua bahan (komponen) yang terlibat dalam sistem proses, sebagai hasil eksekusi pada tahap ini dapat dilihat pada gambar 6a. Pada program ini, pada pengguna hanya mengklik bahan (komponen) yang digunakan. Pada judul terlihat proses pemilihan komponen ke 1 dari jumlah komponen total 6 berdasarkan data yang telah dimasukkan sebelumnya oleh pengguna sebagaimana Gambar 5.
- c. Pemilihan peralatan berdasarkan urutan dari diagram alir proses, dengan hasil eksekusi seperti pada Gambar 6b. Jadi “Mixer” sebagai alat ke 1, “Reaktor Stokhiometri” alat ke 2, “Splitter” alat ke 3 dan “Recycle” yang bersifat maya sebagai alat ke 4. Pada judul terlihat hal yang sama seperti pada proses sebelumnya.

- d. Setiap kali satu alat telah dipilih, maka akan muncul jendela dialog sebagaimana yang ditunjukkan pada Gambar 7. Pada bagian ini, program meminta pengguna memasukkan nomor aliran masuk dan keluar dari alat dalam bentuk matrik dengan ukuran yang sama. Karena mixer terdiri dari dua masukkan dan satu keluaran, maka ditulis 1 6 untuk aliran masukkan dan 2 0 untuk aliran keluaran. Hal yang sama untuk selanjutnya pada reaktor masukkan 2 0 dan aliran keluaran 3 0.



a. Komponen  
b. Peralatan  
Gambar 6. Pemilihan komponen dan alat dengan perintah `x_choose`.



Gambar 7. Penentuan nomor aliran masuk dan keluar untuk tiap alat.

Selanjutnya, tahap kedua merupakan tahap yang dilakukan oleh program yaitu perhitungan secara modular. Arti modular, tiap alat dibuat dalam satu modul atau sub rutin berdasarkan kriteria yang telah dimasukkan oleh pengguna dan hasil perhitungan dari modul sebelumnya. Pada kasus diatas digunakan empat modul dengan sub rutin sebagaimana yang ditunjukkan pada Tabel 2 sampai dengan 5.

Pada artikel ini, hanya akan dibahas modul berdasarkan kasus diatas saja. Gambar 8 menunjukkan diagram balok dari modul mixer beserta notasi variabel masukkan dan keluaran. Semua variabel diproses dalam susunan larik (*array*) dalam bentuk matrik. Dalam hal ini, indek *i* menunjukkan aliran dan indek *j* sebagai komponen

(bahan). Penyelesaian modul mixer berdasarkan pada persamaan berikut:

$$F_{O_i} = \sum_{i=1}^{\text{aliran\_in}} F_{in_i} \quad (1)$$

$$X_{O_{i,j}} = \frac{\sum F_{in_i} \cdot X_{i,j}}{F_{O_i}} \quad (2)$$



Gambar 8. Diagram balok modul mixer

Tabel 2. Modul mixer

```
function [F0mix,X0mix]=mixer(Fin,Xin);
//Modul pencampur
  FinT=Fin';
  F0mix=sum(FinT);
  X0mix=FinT*Xin/F0mix;
endfunction
```

Untuk modul reaktor stokhiometri diasumsikan dengan neraca bahan sistem linear, dengan penyederhanaan:

- Konversi molar dari sejumlah reaksi paralel, NR
- Tiap reaksi  $r$ , didefinisikan komponen kunci, pereaksi pembatas,  $l(r)$
- Normalisasi koefisien stoikiometri:

$$N_{r,k} = \frac{S_{r,k}}{S_{r,l}}$$

Dimana:

$S_{r,k}$  = koefisien stoikiometri untuk reaksi ke  $r$  pada komponen  $k$   
 $S_{r,l}$  = koefisien stoikiometri komponen kunci  $l$  untuk reaksi  $r$ .

- Dengan menggunakan konvensi
  - $N_{r,k} > 0$ , jika komponen merupakan produk
  - $N_{r,k} < 0$ , jika komponen merupakan reaktan
  - $N_{r,k} = 0$ , jika komponen merupakan inert
- Definisi fraksi konversi per pass didasarkan pada pereaksi pembatas

$$M_{RX,k} = M_{IN,k} + \sum_{r=1}^{NR} N_{r,k} X_r M_{IN,l} \quad (3)$$

Dimana:  $X_r$  = konversi per pass pada reaksi ke  $r$

Normalisasi koefisien stoikiometri,  $N_{r,k}$  dapat dilihat Tabel 3.

Tabel 3. Koefisien stoikiometri pada reaksi 1 dan 2 dengan kode nomor komponen

No.komp, →	1	2	3	4	5	6
$r, \downarrow$	CH <sub>4</sub>	O <sub>2</sub>	C <sub>2</sub> H <sub>6</sub>	CO <sub>2</sub>	H <sub>2</sub> O	N <sub>2</sub>
1	-1	-2	0	1	2	0
2	0	-7/2	-1	2	3	0

Persamaan untuk pereaksi pembatas dapat ditulis sebagai berikut:

$$M_{RX, CH_4} = M_{IN, CH_4} - 0,6 M_{IN, CH_4} = 0,4 M_{IN, CH_4} \quad (4)$$

$$M_{RX, C_2H_6} = M_{IN, C_2H_6} - 0,8 M_{IN, C_2H_6} = 0,2 M_{IN, C_2H_6} \quad (5)$$

Persamaan (4) dan (5) dapat ditulis dengan kode nomor komponen sebagaimana persamaan (6) dan (7).

$$M_{RX, 1} = M_{IN, 1} - 0,6 M_{IN, 1} = 0,4 M_{IN, 1} \quad (6)$$

$$M_{RX, 3} = M_{IN, 3} - 0,8 M_{IN, 3} = 0,2 M_{IN, 3} \quad (7)$$

Untuk komponen yang lain:

Komponen O<sub>2</sub> dengan kode nomor 2:

$$M_{RX, 2} = M_{IN, 2} - 2 (0,6) M_{IN, 1} - 7/2 (0,8) M_{IN, 3} \quad (8)$$

Komponen H<sub>2</sub>O dengan kode nomor 5:

$$M_{RX, 5} = M_{IN, 5} + 2 (0,6) M_{IN, 1} + 3 (0,8) M_{IN, 3} \quad (9)$$

Komponen CO<sub>2</sub> dengan kode nomor 4:

$$M_{RX, 4} = M_{IN, 4} + (0,6) M_{IN, 1} + 2 (0,8) M_{IN, 3} \quad (10)$$

Sedangkan modul untuk reaktor stokhiometri dapat dilihat pada Tabel 4 berikut.

Tabel 4. Modul reaktor stokhiometri

```
function[Fmout,XCout, Bout]=reaksto(Fmin,XCin,Konv,Koef,LR)
//Reaktan Stokhiometri
//Fm = molar flowrate total = mol/waktu
//B = molar flowrate komponen = mol/waktu
//XC = fraksi mol komponen
//LR: Limiting reaktan, dalam bentuk baris, misal: LR=[1 3]
//Konv dalam bentuk matrix kolom

  Bin=Fmin.*XCin;
  ukr=size(LR);
  jr=ukr(1,2);
  for i=1:jr
    BLR(1,i)=Bin(1,LR(1,i));
  end
  BLR=BLR';
  Xblr=Konv.*BLR;

  Koef=Koef'; //ubah ke bentuk kolom
  Brx=Koef*Xblr; //flowrate reaksi
  Bout=Bin'+Brx;
  Fmout=sum(Bout);
  XCout=Bout./Fmout;
endfunction
```

Modul splitter hampir sama dengan modul mixer, hanya pada mixer aliran masuk lebih dari satu menjadi satu, sedangkan pada splitter aliran masuk satu menjadi lebih dari satu (banyak). Sub rutin untuk splitter dapat dilihat pada Tabel 5.

Tabel 5. Modul splitter

```
function [FSpO,XSpO]=spliter(FSpin,XSpin,RS);
//Alat pemisah 1 menjadi 2 aliran dengan rasio Rs
FSpO(2,1)=FSpin*RS; //***Aliran recycle
FSpO(1,1)=FSpin-FSpO(2,1); //***Aliran Keluar
XSpO(1,:)=XSpin;
XSpO(2,:)=XSpin;
endfunction
```

Modul recycle dianggap bagian dari alat yang bersifat abstrak atau maya. Pada bagian ini disebut juga dengan *tear*.

Tabel 6. Modul recycle

```
function [FOT,XOT]=tear(FiT,XiT)
//modul untuk recycle:
  FOT=sum(FiT)./2;
  XOT=(sum(XiT,'r'))./2;
endfunction
```

Proses perhitungan pada modul *tear* untuk proses recycle dengan metoda bagi dua, sebagaimana yang ditunjukkan pada Tabel 6.

**HASIL DAN PEMBAHASAN**

Untuk sistem proses dengan *recycle*, proses penyelesaiannya secara iterasi. Pada Tabel 7 merupakan tampilan untuk tiap alat dalam bentuk matrik ukuran 6 x 6 sebagai fraksi mol komponen (bahan). Kolom dari 1 sampai 6 merupakan komponen dari bahan dengan urutan sebagaimana yang tabel konversi diatas. Sedangkan baris menunjukkan aliran dari 1 sampai dengan aliran 6. Tabel 7 merupakan proses iterasi tahap pertama, sebagaimana pada Mixer terlihat nilai yang terisi hanya pada aliran 1 dan aliran 2, hal ini disebabkan pada saat itu nilai yang terhitung pada aliran 1 sebagai umpan (masukkan awal) dari sistem dan aliran 2 sebagai hasil penjumlahan dari Mixer. Pada aliran 6 keluaran dari *tear* atau masukkan untuk mixer masih dianggap nol.

Tabel 7. Hasil eksekusi sistem proses secara bertahap

Mixer - UnOp=1 - Fraksi mol						
0.038	0.231	0.038	0.000	0.000	0.923	
0.038	0.231	0.038	0.000	0.000	0.923	
0.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000	0.000	0.000	0.000	0.000	
Reaktor - UnOp=2 - Fraksi mol						
0.038	0.231	0.038	0.000	0.000	0.923	
0.038	0.231	0.038	0.000	0.000	0.923	
0.012	0.062	0.006	0.068	0.111	0.741	
0.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000	0.000	0.000	0.000	0.000	
Splitter - UnOp=3 - Fraksi mol						
0.038	0.231	0.038	0.000	0.000	0.923	
0.038	0.231	0.038	0.000	0.000	0.923	
0.012	0.062	0.006	0.068	0.111	0.741	
0.012	0.062	0.006	0.068	0.111	0.741	
0.000	0.000	0.000	0.000	0.000	0.000	
Tear - UnOp-maya=4 - Fraksi mol						
0.038	0.231	0.038	0.000	0.000	0.923	
0.038	0.231	0.038	0.000	0.000	0.923	
0.012	0.062	0.006	0.068	0.111	0.741	
0.012	0.062	0.006	0.068	0.111	0.741	
0.012	0.062	0.006	0.068	0.111	0.741	
0.006	0.031	0.003	0.034	0.056	0.370	

Selanjutnya pada alat kedua, reaktor baris ke tiga terisi, yang merupakan hasil reaksi oksidasi dari gas metana dan etana sebagaimana yang ditunjukkan deskripsi proses. Pada baris ke tiga semua komponen terisi nilainya. Baris ke empat sampai ke enam masih kosong, karena perhitungan belum sampai pada modul tersebut. Pada modul splitter, baris ke empat yang berarti aliran ke empat dan ke lima sudah terisi. Hal ini disebabkan karena proses perhitungan pada splitter sudah dieksekusi pada saat itu.

Tahap terakhir pada iterasi pertama adalah alat *recycle* (proses *tear*) dengan metoda bagi dua yaitu bagi dua antara aliran (dalam hal ini baris) ke lima dan ke enam. Hasil akhir proses bagi dua adalah baris ke enam pada Tabel 6. Pada baris tersebut terlihat bahwa

nilainya masih jauh berbeda dibandingkan pada baris 3, 4 dan 5. Ketiga baris tersebut mempunyai nilai komposisi yang sama dikarenakan, peralatan splitter hanya membagi aliran tanpa merubah komposisi. Proses iterasi diatas berjalan terus hingga didapat nilai antara aliran ke lima dan ke enam sama dengan toleransi 0.0000001. Hasil akhir dari proses iterasi dapat dilihat pada Tabel 8 dengan iterasi sebanyak 49 kali. Sedangkan pada Tabel 9 merupakan hasil akhir berdasarkan neraca massa, dimana kolom pertama merupakan laju alir total (kg/jam) sedangkan kolom dua sampai dengan tujuh merupakan fraksi berat dari komponen (bahan) CH<sub>4</sub>, O<sub>2</sub>, C<sub>2</sub>H<sub>6</sub>, CO<sub>2</sub>, H<sub>2</sub>O dan N<sub>2</sub>.

Tabel 8. Hasil akhir iterasi

Hasil akhir iterasi dengan tol=0.0000001 sebesar 49 adalah						
0.038	0.231	0.038	0.000	0.000	0.923	
0.028	0.168	0.027	0.026	0.042	0.859	
0.010	0.051	0.005	0.074	0.121	0.740	
0.010	0.051	0.005	0.074	0.121	0.740	
0.010	0.051	0.005	0.074	0.121	0.740	
0.010	0.051	0.005	0.074	0.121	0.740	

Tabel 9. Hasil akhir iterasi berdasarkan neraca massa

NERACA MASSA							
Kolom-1: Laju alir (kg/jam), Kolom 2...7: Fraksi berat aliran 1...6							
Laju alir(kg/jam)	XCH4	XO2	XC2H6	XCO2	XH2O	XN2	
9100	0.018	0.211	0.033	0.000	0.000	0.738	
13000	0.014	0.165	0.025	0.035	0.023	0.738	
13000	0.006	0.058	0.005	0.115	0.077	0.738	
9100	0.006	0.058	0.005	0.115	0.077	0.738	
3900	0.006	0.058	0.005	0.115	0.077	0.738	
3900	0.006	0.058	0.005	0.115	0.077	0.738	

**KESIMPULAN**

Penerapan algoritma modular pada sistem proses dengan *recycle* menggunakan perangkat lunak bebas biaya memberikan hasil yang cukup baik. Penyusunan program dibuat bersifat *user friendly* dan umum sehingga akan memudahkan pengguna akhir.

**UCAPAN TERIMA KASIH**

Pada kesempatan ini, penulis mengucapkan terima kasih pada konsorsium Scilab serta Open Office yang telah mengijinkan penggunaan perangkat lunaknya.

**DAFTAR PUSTAKA**

Biegler, L.T., Grossmann, I.E., and Westerberg A.W., (1997), *Systematic Methods of Chemical Process Design*, Prentice Hall PTR.

Chandler G., Roberts S., (2002), *Intoduction to Scilab*, (<http://comptsci.anu.edu.au/Scilab/primer.pdf>)

Rietsch, E., (2002), *An Introduction to Scilab from a Matlab User's Point of View*, (<http://www.math.tu-berlin.de/~ehrhartd/scilab/doc/html/intro>)