

# PENGEMBANGAN PIRANTI LUNAK SISTEM PROSES BERSIFAT *USER FRIENDLY* DENGAN SCILAB

(*Open Source Software*)

Setia Budi Sasongko \*)

## Abstract

*Requirement of software is necessary for the process system-computer simulation. In reality, costly of software is one of restriction for the software development. However, the costly constraint can be solved by using open source software such as Scilab and OpenOffice. The goal of this article formulates user friendly program of the process system Scilab program by Graphical User Interface (GUI) and interaction with OpenOffice-Calc for the data-base physical property. Scilab program has been listed in this article and algorithm of the program as the methodology has been presented in this article. Results of the simulation program indicate good finding for the process system.*

**Kata kunci :** *Scilab; OpenOffice, simulasi komputer; OSS (Open Source Software), Antar Muka Pengguna - Graphical User Interface (GUI).*

## Pendahuluan

Penggunaan piranti lunak dalam bidang teknik kimia sudah merupakan suatu hal yang tidak dapat dihindari. Salah satu permasalahan yang timbul pada pengembangan piranti lunak adalah masalah harga. Akan tetapi, problema tersebut dapat diatasi dengan adanya piranti lunak yang bersifat lisensi bebas biaya (*FOC*). Salah satu piranti lunak dengan lisensi bebas biaya adalah *Scilab*, yang dapat diunduh (*download*) di alamat [www.scilab.org](http://www.scilab.org) (Arief, 2008). Pada tulisan ini, penulis selain menggunakan *scilab* juga menggunakan piranti lunak yang bersifat bebas biaya lainnya, yaitu *OpenOffice.org Calc*, yang diperoleh pada alamat [www.openoffice.org](http://www.openoffice.org). Piranti lunak tersebut setipe dengan piranti lunak lembar kerja (*spreadsheet*) Excel.

Piranti lunak digunakan dalam berbagai macam kategori, mulai dari pengguna akhir (*end user*) pada piranti lunak aplikasi, sampai dengan pembuat program. Tulisan ini ditujukan untuk pembuat program pada piranti lunak aplikasi yang lebih interaktif atau lebih akrab (*user friendly*) dengan pengguna akhir. Sebagai tujuan akhir dari tulisan ini adalah mengembangkan piranti lunak pada bidang teknik kimia dengan model antar muka pengguna grafikal yang dilengkapi dengan data base karakteristik numerik material yang tersimpan pada Excel.

## Tinjauan Pustaka (Fundamental)

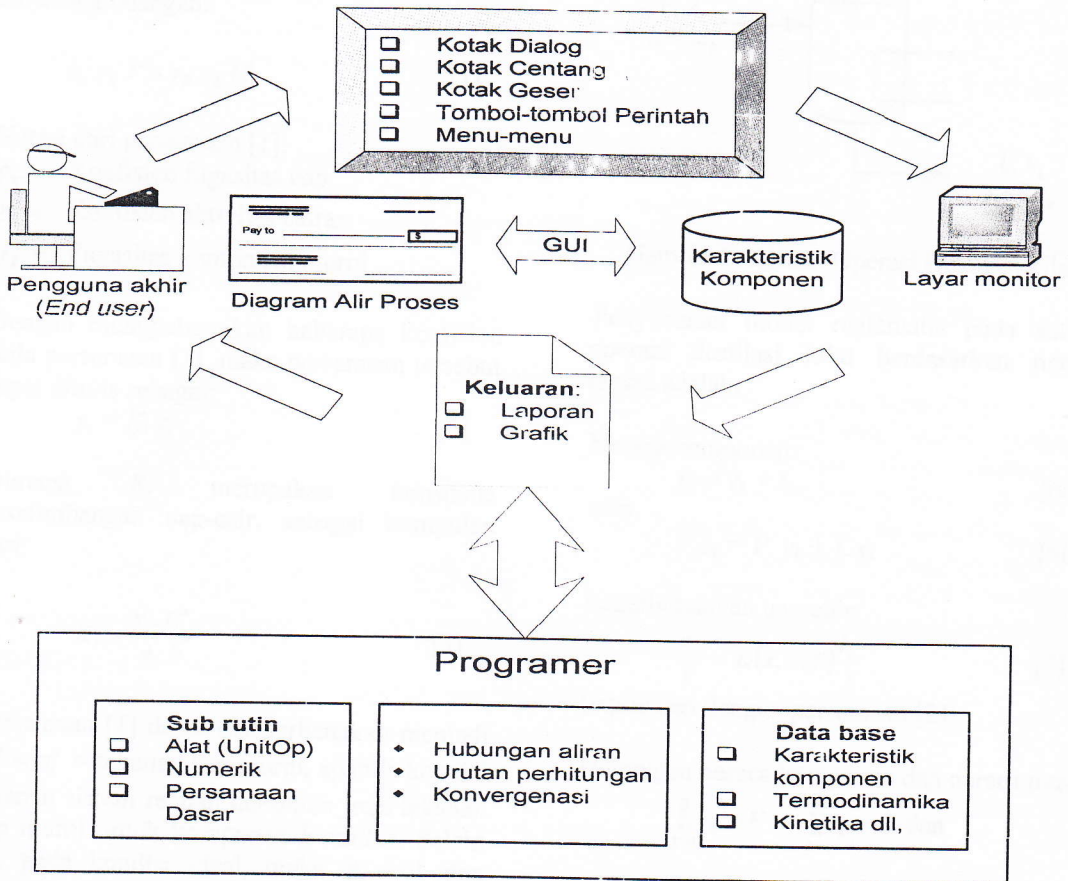
Bentuk grafis pada piranti lunak aplikasi merupakan salah satu daya tarik bagi pengguna akhir, sebagaimana yang telah dikembangkan oleh perusahaan piranti lunak seperti *Macintos*, *Microsoft*. Kemudahan pengguna akhir agar tidak dibebani dengan perintah yang harus dihafalkan digantikan dengan simbol atau ikon-ikon. Demikian pula dengan keluaran sebagai hasil dari perhitungan ataupun simulasi tidak hanya ditambahkan dalam bentuk data numerik akan tetapi dapat langsung dimunculkan dalam bentuk grafik sehingga akan memudahkan bagi pengguna akhir dalam menganalisis atau mengamati lebih detail permasalahan yang sedang dipelajari.

Penyusunan program yang bersifat akrab pengguna (*user friendly*) pada piranti lunak dengan media grafis dikenal dengan nama *Graphical User Interface (GUI)* atau *Antar Muka Pengguna Grafikal (AMP)*. Komunikasi antara pengguna akhir dengan piranti lunak aplikasi dilakukan melalui penekanan tombol riil piranti keras *mouse* atau *keyboard* pada ikon piranti lunak yang divisualisasikan dengan (*button*) tombol, *pull down menus*, *slider controls*, *check boxes* (kotak centang) maupun kotak dialog lainnya sebagaimana yang ditampilkan pada layar monitor. Jadi dengan kata lain interface input-

output antara pengguna dengan piranti lunak dilakukan melalui ikon-ikon. Penampilan dari ikon-ikon dan tampilan hasil akhir dari perhitungan atau simulasi pada layar merupakan bagian seni dari penyusun program.

Scilab sebagai suatu piranti lunak yang bersifat *open source* sudah dilengkapi dengan perintah-perintah AMP (GUI) tersebut. Selain

itu Piranti lunak Scilab ini dilengkapi pula dengan kemampuan untuk membaca data dari program lembar kerja (*spreadsheet*) pada Excel (Chandler and Roberts, 2002). Kedua kemampuan tersebut dapat dikembangkan untuk aplikasi teknik kimia yang memerlukan data base yang cukup banyak untuk karakteristik fisik dan kimia dari suatu bahan.



Gambar 1: Hubungan antara Pengguna Akhir dengan piranti lunak bersifat *User Friendly Metodologi*

Metodologi untuk menyusun perangkat lunak pada tingkat pemrograman lebih bersifat pada algoritma atau kerangka pikir untuk mencapai tujuan yang diharapkan. Algoritma pengembangan (aplikasi GUI) adalah:

1. Membuka data base di Excel
2. Pengguna akhir memilih komponen pada sistem secara interaktif, bentuk aplikasi GUI, sebagai interaksi antara Excel (aplikasi database) dengan pengguna
3. Menyimpan data komponen yang dipilih beserta karakteristik komponen.

4. Membuka sub rutin Peralatan (UnitOp), dan pengguna akhir mengisi spesifikasi alat yang diinginkan, aplikasi GUI pada peralatan (UnitOp) dengan pengguna.
5. Simulasi – perhitungan
6. Menampilkan hasil perhitungan untuk pengguna akhir, aplikasi GUI sebagai bagian dari *art*.

Pada bagian ini digunakan peralatan (UnitOp) destilasi kilat (*flash destillation*), sebagai simulasi atau studi kasus dari pemrograman.



### Model Matematis Destilasi Kilat

Kesetimbangan uap-cairan merupakan salah satu dasar proses pemisahan. Pada cabang termodinamika, pengembangan persamaan kesetimbangan uap-cair cukup banyak dengan hasil yang akurat (Bequette, 1998). Akan tetapi secara umum, pendekatan kesetimbangan uap-cairan adalah  $f_{komp}^{uap} = f_{komp}^{cair}$  dimana kedua hubungan tersebut dinyatakan dengan:

$$\phi_k \cdot y_k \cdot P = \gamma_k \cdot x_k \cdot f_k^o \quad [1]$$

Notasi dari persamaan [1]:

$\phi_k$  = Koefisien fugasitas uap

$\gamma_k$  = Koefisien aktivitas cairan

$f_k^o$  = Fugasitas komponen murni

Dengan menggabungkan beberapa koefisien pada persamaan [1], maka persamaan tersebut dapat ditulis sebagai:

$$y_i = K_i \cdot x_i \quad [2]$$

Dimana  $K_i$  merupakan konstanta kesetimbangan uap-cair, sebagai kumpulan dari:

$$K_i = \frac{\gamma_i \cdot f_i^o}{\phi_i \cdot P} \quad [3]$$

Persamaan [1] dapat disederhanakan menjadi  $f_i^o = p_i^o =$  tekanan uap murni, apabila kondisi tekanan sistem rendah demikian juga tekanan uap murni untuk komponen  $k$  rendah. Selain itu, pada kondisi ideal, maka  $\phi_k = 1$  dan  $\gamma_k = 1$ . Dengan demikian, pada kondisi ideal dan tekanan tidak terlalu tinggi, persamaan [2] dapat diubah menjadi:

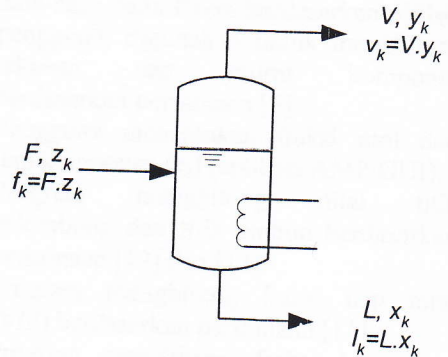
$$K_i = \frac{p_i^o}{P} = \frac{y_k}{x_k} \quad [4]$$

Tekanan uap murni dapat menggunakan persamaan Antoine dengan bentuk:

$$\ln p = A - \frac{B}{(T+C)} \quad [5]$$

Dimana untuk setiap komponen  $k$  diperlukan tiga data, yaitu  $A$ ,  $B$  dan  $C$ .

Aplikasi kesetimbangan uap-cair pada sistem pemisahan sebagaimana yang ditunjukkan pada gambar 1, satu aliran umpan  $F$  terpisah menjadi dua aliran uap-cair.



Gambar 2: Satuan Operasi Destilasi kilat

Penyusunan model matematis pada satuan operasi destilasi kilat berdasarkan neraca massa adalah:

Neraca komponen:

$$f_k = v_k + l_k \quad [6a]$$

atau

$$F \cdot z_k = V \cdot y_k + L \cdot x_k \quad [6b]$$

Kesetimbangan uap-cair:

$$\frac{v_k}{V} = K(x, P, T) \frac{l_k}{L} \quad [7]$$

(ekuivalen dengan persamaan [2])

Hubungan neraca komponen dan neraca total:

$$\sum_{k=1}^n v_k = V \quad ; \quad \sum_{k=1}^n l_k = L \quad \text{dan}$$

$$\sum_{k=1}^n f_k = F \quad [8]$$

Pengabungan dari persamaan diatas, maka:

$$x_k = \frac{z_k}{1 + \phi(K_k - 1)} \quad [9]$$

dan

$$y_k = \frac{K_k \cdot z_k}{1 + \phi(K_k - 1)} \quad [10]$$

Pada sistem kesetimbangan akan terjadi:  $\sum x_k = \sum y_k$ , maka akan didapat persamaan:

$$f(\phi) = \sum_{k=1}^n \frac{z_k(1-K_k)}{\phi(K_k-1)+1} = 0 \quad [11]$$

Dimana:  $\phi$  (psi) =  $V/F$  = fraksi uap.

Dengan menggunakan persamaan [11], maka fraksi uap  $\phi$  dapat dihitung dengan menggunakan metoda numerik misalkan Newton Raphson dengan nilai berkisar antara 0 sampai dengan 1. Nilai tersebut terpenuhi apabila suhu operasi berada diantara suhu titik gelembung (*bubble point temperature*) dan suhu titik embun (*dew point temperature*). Suhu titik gelembung merupakan suhu jenuh dari suatu campuran cairan yang akan mulai berubah fase uap-cair, sedangkan suhu titik embun adalah suhu jenuh dari campuran uap yang akan mulai berubah fase.

Berdasarkan hal tersebut, maka suhu titik gelembung dapat dicari dengan menggunakan persamaan [12] sedangkan untuk suhu titik embun dengan persamaan [13] pada tekanan total konstan ( $P$ , diketahui).

$$f(T_{bp}) = \sum_{k=1}^n \frac{P_k^o(T_{bp}) \cdot x_k}{P} - 1 = 0 \quad [12]$$

$$f(T_{dp}) = \sum_{k=1}^n \frac{P \cdot y_k}{P_k^o(T_{dp})} - 1 = 0 \quad [13]$$

### Deskripsi dari program simulasi

Tujuan dari simulasi ini adalah menentukan perbandingan Uap terhadap Umpan ( $V/F$ ), komposisi fraksi uap komponen ( $y_k$ ) dan juga komposisi fraksi cair komponen ( $x_k$ ) pada suhu operasi diantara titik embun dan titik gelembung. Pada simulasi ini pengguna akhir dapat memilih sendiri komponen yang akan disimulasikan melalui interaktif secara grafis (GUI), demikian juga dengan penentuan komposisi umpan. Selanjutnya pengguna dapat dapat menentukan suhu operasi pemisahan secara interaktif grafis dimana nilainya berada antara titik embun dan titik gelembung, yang kedua sudah dihitung oleh program. Hasil akhir perhitungan perbandingan  $V/F$ , serta komposisi komponen keluar dari destilasi kilat ditampilkan pada jendela (*window*) hasil.

Algoritma perhitungan untuk destilasi kilat adalah sebagai berikut:

1. Pengguna memilih komponen ( $k$ ) yang akan masuk pada destilasi kilat (aplikasi AMP/GUI).
2. Program akan menyimpan data konstanta Antoine ( $A_k, B_k, C_k$ ) tiap komponen dari data-base pada Excel berdasarkan pilihan pengguna, digunakan untuk menghitung tekanan uap murni komponen berdasarkan persamaan [5].
3. Pengguna menentukan fraksi mol dari tiap komponen ( $z_k$ ) (aplikasi AMP/GUI).
4. Program menghitung nilai titik gelembung dan titik embun berdasarkan persamaan [12] dan [13].
5. Program menghitung fraksi uap total ( $V/F$ ) berdasarkan persamaan [11].
6. Program menghitung fraksi komponen keluar berdasarkan neraca komponen persamaan [6].
7. Menampilkan hasil perhitungan.

### Listing Program dan Hasil simulasi

Program yang baik, apabila bersifat sederhana dan umum. Oleh karenanya, program disusun dalam bentuk sub-rutin, sehingga dapat digunakan untuk aplikasi lain sebagai bagian dari sistem. Penggunaan sub-rutin pada Scilab dalam bentuk *function*. Berikut program utama dari simulasi destilasi kilat disajikan pada tabel 1.

Tabel 1: Program utama destilasi kilat

```
//Buka data di excel - tahap 1
lmor=readxls('c:\scijur\DataFisbs.xls')
;
dtlb1=lubr(3);
dfnamakom=dtlb1(:,2);

//memilih komponen: - tahap 2
tulisbs=['BANK DATA - SETIA BUDI
SASONGKO'; 'Pilih komponen'];

nokomp=1; k=0;
while nokomp>0
    nokomp=x_choose(dfnamakom,tulisbs,...
    'Selesai');
    if nokomp > 0 then
        k=k+1;
        txttdia=['MASUKKAN FRAKSI ANTARA 0-
1'; 'DILANJUTKAN KLIK Finish'
        'SETIA BUDI SASONGKO'];
        frk(k)=x_mdialog(txttdia,'Fraksi=
','');
        Idkom(k)=dtlb1(nokomp,1);
```



```

namakom(k)=dtlbl(nokomp,2);
Ai(k)=dtlbl(nokomp,3);
Bi(k)=dtlbl(nokomp,4);
Ci(k)=dtlbl(nokomp,5);
end
end

//merubah posisi kolom menjadi baris
zi=evstr(frk');
Ai=Ai';
Bi=Bi';
Ci=Ci';

// tahap 3
Pt=760; //Tekanan total 1 atm

tdp=dewpt1(zi,Ai,Bi,Ci,Pt)
tbp=bubpt1(zi,Ai,Bi,Ci,Pt)

// tahap 4
tulis1=['Nilai dewpoint-ttk embun = \
+string(tdp),'Nilai bubblept-ttk uap = \
+string(tbp)];

T=x_mdialog(tulis1,'Suhu operasi=
\,');
T=evstr(T);
// tahap 5
if T<tdp & T>tbp then
[psi,xi,yi]=flash1(zi,Ai,Bi,Ci,Pt,T)
x_message_modeless(['Fr.uap (V/F):',

```

```

+string(psi),'', 'Fraksi komponen
uap:', +string(yi),'', 'Fraksi
komponen cairan:', +string(xi)]
end

```

Pada program utama tersebut dapat dibagi menjadi 5 bagian besar, yaitu:

1. Membuka dan membaca excel, dengan perintah **readxls**.

File dengan nama: "DataFisbs.xls" tidak muncul secara eksplisit. Gambar 3 menunjukkan posisi letak dari data tekanan uap murni dari komponen pada *sheet* (lembar) ke 3 dan posisi data pada kolom tertentu dengan menggunakan piranti lunak OpenOffice.calc. Yang perlu mendapat perhatian, pada saat menyimpan file harus dengan ekstensi xls. Pada program terlihat, sebagai alat interaksi adalah nama komponen yang terletak pada posisi di kolom → `dtlbl(:,2)`.

Data Files - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

Arial 12 B I U

f(x) Σ =

	A	B	C	D	E
1	No.		A	B	C
2	1	ACETALDEHYDE	16.2481	2465.15	-37.15
3	2	ACETIC ACID	16.8080	3405.57	-56.34
4	3	ACETIC ANHYDRIDE	16.3982	3287.56	-75.11
5	4	ACETONE	16.6513	2940.46	-35.93
6	5	ACETONITRILE	16.2874	2945.47	-49.15
7	6	ACETYL CHLORIDE	15.7514	2447.33	-55.33
8	7	ACETYLENE	16.3481	1637.18	-19.77
9	8	ACROLEIN	15.9057	2606.53	-45.15
10	9	ACRYLIC ACID	16.5617	3319.18	-80.15
11	10	ACRYLONITRILE	15.9253	2782.21	-51.15
12	11	ALLYL ALCOHOL	16.9066	2928.20	-85.15
13	12	ALLYL CHLORIDE	15.9772	2531.92	-47.15
14	13	ALLYL CYANIDE	16.0019	3128.75	-58.15
15	14	ALPHA-METHYL STYRENE	16.3308	3644.30	-67.15
16	15	AMMONIA	16.9481	2132.50	-32.98
17	16	ANILINE	16.6748	3857.52	-73.15
18	17	ANTHRACENE	405.3100	53.70	0.00
19	18	ARGON	15.2337	700.51	-5.84
20	19	BENZALDEHYDE	16.3501	3748.62	-66.12

Data BM MP BP T Pc Antn mmHg Cp uap hf Gf Hv

Sheet 3 / 5 PageStyle\_Antn\_mmHg % STD Sum=0

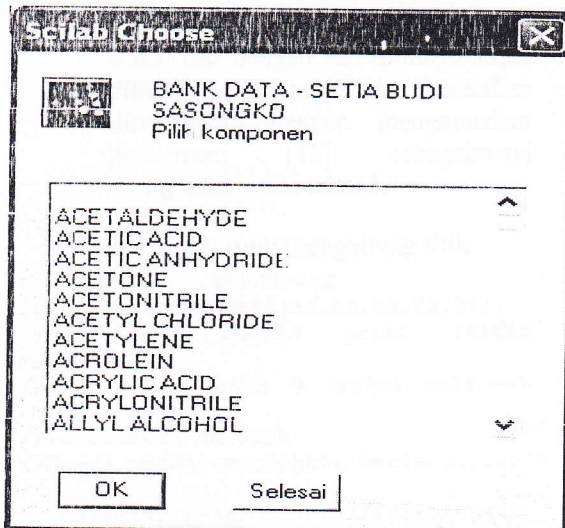
Sheet ke 3

Data diambil

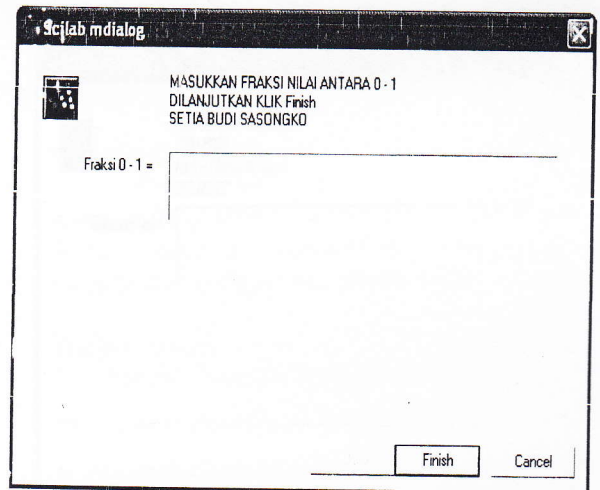
Gambar 3: Data-base pada OpenOffice.Calc

2. Pengguna memilih komponen dengan menggunakan fasilitas GUI dari program, dengan perintah **x\_choose** (gambar 4a) dan dilanjutkan dengan memasukkan komposisi komponen dalam bentuk fraksi dengan nilai antara 0 dan 1 dengan fasilitas GUI **x\_mdialog** (gambar 4b).

Berdasarkan urutan komponen yang dipilih oleh pengguna, program akan mengambil data yang diperlukan (pada kasus ini konstanta tekanan uap murni komponen,  $A_i$ ,  $B_i$  dan  $C_i$ ) untuk disimpan dalam bentuk matriks



4a. Pemilihan komponen



4b. Memasukkan komposisi komponen (fraksi)

Gambar 4. Pemilihan komponen dan komposisi

3. Program akan menghitung titik embun dan titik gelembung yang dieksekusi dari sub rutin dewpt1 dan bubpt1 sebagaimana ditunjukkan pada tabel 2 dan tabel 3. Untuk menghitung titik embun (*dew point*), maka semua fraksi komponen masuk dianggap sebagai uap  $y_k$ . Berdasarkan hubungan kesetimbangan, persamaan [4] dan  $\sum x_k = 1$  akan didapat titik embun berdasarkan persamaan [13] dengan menggunakan metoda newton raphson (Constantinides and Mostoufi, 2000), sub rutin tersebut dapat dilihat pada tabel 4.

Tabel 2: sub rutin menghitung titik embun

```
function tdp=dewpt1(yk,Ak,Bk,Ck,Pt);
//Menghitung dew point (titik embun)
//fraksi mol masuk fraksi uap --> yk = zk
//Pendekatan Antoine
//Semua parameter -> vektor baris.....

    param(1,:)=yk; //fraksi uap
    param(2,:)=Ak;
    param(3,:)=Bk;
    param(4,:)=Ck;

uk=size(yk);
    param(5,:)=ones(1,uk(1,2))*Pt;

Taw=300 //Suhu tebakan awal satuan K
    tdp=newraph(sigmintux1,Taw,param)
endfunction
```

Tabel 3: sub rutin dew point berdasarkan rumus [13]

```
function jmlx=sigmintux1(T,param);
//fungsi sigmax - 1
//untuk menghitung dewpcint (titik embun)
//Pengambilan nilai parameter:
    yk=param(1,:);
    Ak=param(2,:);
    Bk=param(3,:);
    Ck=param(4,:);
    Pt=param(5,1);
    pk=exp(Ak-Bk./(T+Ck));
    Kk=pk./Pt;
    jmlx=sum(yk./Kk)-1;
endfunction
```

Tabel 4: sub rutin newton raphson

```
function
hsl=newraph(penolin,Xa,varargin)
//sub rutin newton raphson
//setia budi sasongko
    X=Xa;
    tol=0.0001;
    eps=0.001;
    tolhit=1;
    while tolhit>tol
        fungawal=penolin(X)
        Xmin=X-eps;
        fmin=penolin(Xmin);
        Xplus=X+eps;
        fpls=penolin(Xplus);
        dfx=(fpls-fmin)/(2*eps);
        Xb=X-(fungawal/dfx);
        tolhit=abs(Xb-X);
        X=Xb;
    end
    hsl=X;
endfunction
```

Untuk menghitung titik gelembung dilakukan dengan metoda hampir yang sama, hanya saja semua fraksi



komponen masuk dianggap sebagai fraksi cair dengan sub rutinnya dapat dilihat pada tabel 5, kemudian diselesaikan dengan menggunakan persamaan [12] sebagaimana ditunjukkan pada tabel 6.

Tabel 5: sub rutin menghitung titik gelembung

```
function tbp=bubpt1(xk,Ak,Bk,Ck,Pt);
//Menghitung bubble point (titik gelembung)
//fraksi mol masuk → fraksi cair -->
xk = zk
//Pendekatan Antoine
//Semua parameter vektor baris.....

    param(1,:)=xk;           //fraksi cair
    param(2,:)=Ak;
    param(3,:)=Bk;
    param(4,:)=Ck;

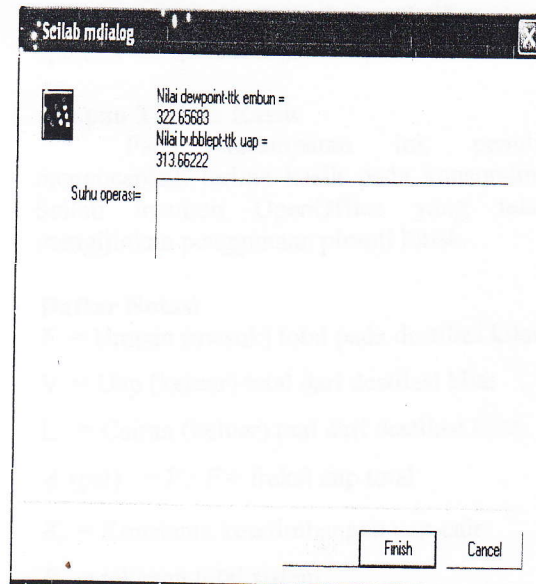
uk=size(xk);
    param(5,:)=cnes(1,uk(1,2))*Pt;

Taw=300 //Suhu tebakan awal satuan K
    tbp=newraph(sigmintuy1,Taw,param)
endfunction
```

Tabel 6: sub rutin dew point berdasarkan rumus [12]

```
function jmlly=sigmintuy1(T,param);
//fungsi sigmay - 1
//untuk menghitung bubblepoint (titik uap)
//Pengambilan nilai parameter:
xk=param(1,:);
Ak=param(2,:);
Bk=param(3,:);
Ck=param(4,:);
Pt=param(5,1);
    pk=exp(Ak-Bk./(T+Ck));
    Kk=pk./Pt;
    jmlly=sum(xk.*Kk)-1;
endfunction
```

- Hasil perhitungan dari titik embun dan titik gelembung ditaripilkan pada jendela dialog, yang selanjutnya pengguna diminta untuk memasukkan suhu operasi yang nilainya diantara titik embun dan titik gelembung untuk mendapatkan hasil proses pemisahan dengan destilasi kilat berupa perbandingan uap/umpan atau V/F beserta komposisi uap dan cairannya. Jendela dialog sebagai interaksi antara pengguna dan program dapat dilihat pada gambar 5, untuk memasukkan suhu operasi dari destilasi kilat.



Gambar 5: Jendela dialog untuk memasukkan suhu operasi

- Pada tahap ini, apabila pengguna mengisikan suhu operasi diantara titik gelembung dan titik embun maka program akan menghitung fraksi uap total (V/F) dengan sub rutin dapat dilihat pada tabel 7 dan tabel 8. Perhitungan fraksi uap total diselesaikan dengan menggunakan rumus [11] dengan menggunakan metoda newton raphson.

Tabel 7: sub rutin destilasi kilat

```
function
[psi,xk,yk]=flash1(zk,Ak,Bk,Ck,Pt,T)
    param(1,:)=zk;
    param(2,:)=Ak;
    param(3,:)=Bk;
    param(4,:)=Ck;
    uk=size(zk);
    param(5,:)=ones(1,uk(1,2))*Pt;
    param(6,:)=ones(1,uk(1,2))*T;
    tebak=0.0001;

psi=newraph(sigpsi,tebak,param);

    pk=exp(Ak-Bk./(T+Ck));
    Kk=pk./Pt;

    xk=zk./(1+psi*(Kk-1));
    yk=Kk.*xk;
endfunction
```

Tabel 8: sub rutin destilasi kilat persamaan [11]

```
function jmlpsi=sigpsi(vf,param)
```



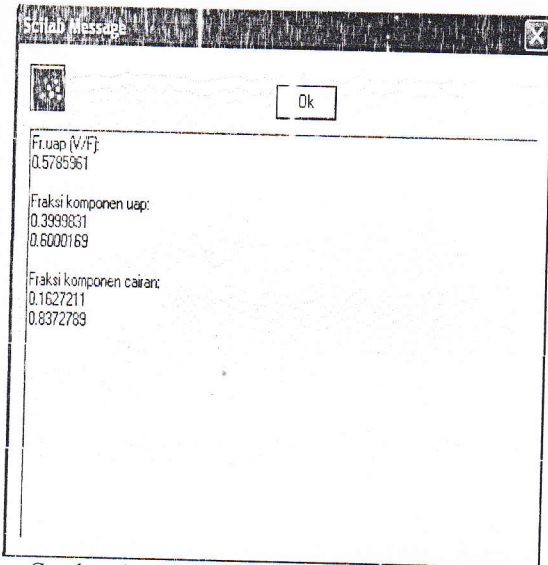
```

zk=param(1,:);
Ak=param(2,:);
Bk=param(3,:);
Ck=param(4,:);
Pt=param(5,1);
T=param(6,1);
uk=size(zk);

pk=exp(Ak-Bk./(T+Ck));
Kk=pk./Pt;
jmlpsi=sum(zk.*(1-
Kk)./(1+vf.*(Kk-1)));
endfunction

```

Hasil perhitungan akan ditambahkan dalam bentuk jendela pesan dengan menggunakan fasilitas GUI dari `x_message_modeless`, sebagaimana yang ditampilkan pada gambar 6.



Gambar 6: Jendela pesan hasil akhir simulasi.

### Kesimpulan

Penyusunan program komputasi dengan menggunakan piranti lunak bebas biaya (*open source software*) dan bersifat akrab pengguna (*user friendly*) pada destilasi kilat sebagai studi kasus telah dapat

dieksekusi dengan baik. Program dengan sifat akrab pengguna ini dapat dikembangkan lebih lanjut sebagai bagian dari pelatihan bagi operator ataupun sebagai alat pendidikan.

### Ucapan Terima Kasih

Pada kesempatan ini, penulis mengucapkan terima kasih pada konsorsium Scilab maupun OpenOffice yang telah mengijinkan penggunaan piranti lunak.

### Daftar Notasi

$F$  = Umpan (masuk) total pada destilasi kilat

$V$  = Uap (keluar) total dari destilasi kilat

$L$  = Cairan (keluar) total dari destilasi kilat.

$\phi$  (psi) =  $V/F$  = fraksi uap total

$K_i$  = Konstanta kesetimbangan uap-cair

$P$  = tekanan total sistem

$y_k$  = fraksi uap komponen  $k$

$x_k$  = fraksi cair komponen  $k$

### Daftar Pustaka

Arief, S. (2008) "Sekilas tentang Scilab" (<http://ilmukomputer.com/2008/04/09/sekilas-tentang-scilab>)

Bequette, B.W., "Process Dynamics, modeling, analysis and simulation", Prentice Hall PTR, NJ, 1998.

Chandler, G. And Roberts, S., (2002) "Introduction to Scilab", (<http://comptisci.anu.edu.au/scilab/primer.pdf>)

Constantinides, A., and Mostoufi, N. (2000), "Numerical Methods for Chemical Engineers with MATLAB Applications", Prentice Hall PTR, NJ.

ISSN 1858-2907



9 771858 290721