



e-ISSN Number  
2655 2967

Available online at <https://jurnal.teknologiindustriumi.ac.id/index.php/JCPE/index>

*Journal of Chemical Process Engineering*

Volume 4 Nomor 1 (2019)



SINTA Accreditation  
Number 10/E/KPT/2019

## Penyelesaian Numeris Berbasis Pemrograman

*(Numerical Completion Based on Programming)*

**Andi Aladin, Takdir Syarif**

*Jurusan Teknik Kimia, Fakultas Teknologi Industri, Universitas Muslim Indonesia  
Jl. Urip Sumohardjo km.05 Makassar, Sulawesi Selatan 90231*

**Kata Kunci:** Numerik,  
Program Bahasa QBASIQ,  
MAT-LAB, Kinetika Reaksi,

### Inti Sari

Pada perinsipnya problem-problem matematika dalam teknik sedapat mungkin diselesaikan secara analitis, sebab sifatnya exact memberi hasil hitungan dengan ketelitian mencapai 100%. Namun penyelesaian analitis memiliki kelemahan, yaitu terbatas pada problem-problem sederhana, sementara dalam bidang teknik termasuk teknik kimia, lebih sering dihadapkan pada problem yang lebih kompleks. Pada kondisi seperti ini maka penyelesaian numeris menjadi alternatif. Dalam penyelesaian numeris ada yang sifatnya adalah coba-coba (trial and error) yang menimbulkan persoalan baru sebab membutuhkan sederetan hitungan yang panjang dan berulang-ulang, sehingga cukup melelahkan jika dihitung secara manual. Persoalan ini dapat diatasi dengan bantuan pemrograman komputer. Dalam makalah ini disajikan dua buah contoh kasus, yang pertama evaluasi kinetika reaksi kompleks polimerisasi urea formaldehid dengan program bahasa QBASIQ dan yang kedua adalah penentuan tetapan kesetimbangan untuk meramalkan komposisi gas hasil pada proses gasifikasi arang batubara dengan menggunakan program MATLAB.

### PENDAHULUAN

Pada prinsipnya berbagai problem matematik dalam ilmu-ilmu keteknikan (engineering) diusahakan sedapat mungkin diselesaikan secara analitis. Penyelesaian analitis merupakan penyelesaian yang sifatnya exact (pasti) yang memberikan ketelitian hingga 100%. Keunggulan lain

hitungan secara analitis cukup singkat. Dibalik keunggulan cara analitis, terdapat kelemahan bahwa penyelesaian analitis hanya ampuh pada problem-problem sederhana, sementara dalam ilmu teknik, tidak terkecuali teknik kimia lebih sering diperhadapkan pada problem yang lebih kompleks.

### Published by

Department of Chemical Engineering  
Faculty of Industrial Technology  
Universitas Muslim Indonesia, Makassar

### Address

Jalan Urip Sumohardjo km. 05 (Kampus 2 UMI)  
Makassar- Sulawesi Selatan

### Phone Number

+62 852 5560 3559  
+62 823 4988 0792

### Corresponding Author

[andi.aladin@umi.ac.id](mailto:andi.aladin@umi.ac.id)



### Journal History

Paper received : 06 April 2019  
Received in revised form : 12 Mei 2019  
Accepted : 30 Mei 2019

Alternatif penyelesaian terhadap problem-problem matematik yang lebih kompleks tersebut adalah dengan penyelesaian numeris. Sekalipun memiliki kelemahan, yaitu sifatnya pendekatan yang biasanya dilakukan secara *iteratif (trial and error)*, sehingga hasil hitungan tidak exact, (ketelitian  $< 100\%$ ). Kelemahan numeris ini pada tingkat toleransi tertentu dapat diterima dalam bidang teknik. Permakluman ini ditempuh sebab pada prakteknya juga akan digunakan hasil hitungan yang realistik. Ketika misalnya hasil hitungan diameter pipa diperoleh sebesar  $D=2,5113$  in (ketelitian hingga 4 angka desimal), tetapi dalam prakteknya akhirnya juga dilakukan pembulatan menjadi 2,5 in sesuai ukuran pipa yang tersedia di pasar

### Program Komputer

Kelemahan lain dalam penyelesaian numeris adalah bahwa pada umumnya membutuhkan banyak hitungan (*iterasi*). Dalam kasus tertentu membutuhkan sampai ribuan iterasi yang tentu saja memerlukan waktu sangat lama dan dapat menjemukan bila hitungan dilakukan secara manual.

Seiring dengan kecanggihan perkembangan teknologi pemrograman komputer, maka hitungan yang memerlukan waktu lama tersebut, dapat diselesaikan dalam waktu relatif singkat. Waktu hitungan satu hari secara manual dapat dipersingkat menjadi satu menit dengan komputer. Bahkan dengan *super-komputer*, waktu hitungan 1 bulan dapat dipersingkat menjadi waktu hitungan 1 detik.

Dalam penyelesaian numeris pada berbagai problem hitungan keteknikan berbagai bahasa pemrograman biasa digunakan seperti software BASIC, C++, FORTRAN dan

PASCAL. Juga tersedia software semi aplikasi seperti MATLAB, dan bahkan MS Excell.

### Dua Langkah Utama

Dalam penanganan kasus-kasus keteknikan pada umumnya dan keteknik-kimiaan pada khususnya dengan penyelesaian numeris menggunakan program komputer diperlukan dua langkah utama sebagai berikut (Sediawan, dan Prasetya 1997):

#### ☞ *Modeling*

Setiap kasus perlu terlebih dahulu diformulasikan ke dalam bentuk persamaan matematik (model matematik). Dalam penyusunan model matematik (*modeling*) untuk mendekati peristiwa-peristiwa keteknik-kimiaan didasarkan pada *chemical engineering tools (material balance, energy balance, equilibrium, dan rate processes)*. Perlu dimaklumi bahwa dalam modeling untuk setiap peristiwa memungkinkan terbentuk beberapa model matematik yang berbeda, tergantung pada asumsi-asumsi yang dilakukan, sehingga pemodelan bersifat *open ended*.

#### ☞ *Programing*

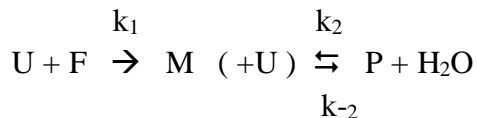
Dari model yang telah disusun pada langkah pertama di atas (terbuka lebih dari satu model), selanjutnya penyelesaiannya dilakukan secara numeris berbasis pemrograman dengan bantuan komputer. Disusun programnya dalam urutan yang logis dengan menggunakan bahasa komputer yang dikuasai (seperti bahasa QBASIC). Bentuk dan susunan program pun dapat berbeda antara programmer yang satu dengan yang lainnya, tergantung

kreativitas dan daya imajinasi seseorang. Dari hasil hitungan program ini sekaligus dapat digunakan untuk mengevaluasi dan memilih model terbaik dari beberapa alternatif model yang dihasilkan dalam langkah pertama.

### Contoh Kasus 1

#### Perhitungan Konstanta Laju Reaksi Kompleks

Aladin (1998) telah melakukan penelitian kinetika polimerisasi resin urea formaldehid berdasarkan mekanisme reaksi seri-paralel bolak-balik, sebagai berikut:



dengan F=formaldehid, U=urea, M=monomer, P = polimer,  $k_1$  = konstanta laju adisi,  $k_2$  = konstanta laju kondensasi arah maju dan  $k_{-2}$  = konstanta laju kondensasi arah balik. Diperoleh data pengamatan laboratorium; dari waktu 0 sampai 120 menit, untuk selang waktu pengamatan tiap 15 menit, konsentrasi formaldehid berturut-turut: 9,123; 5,3081; 4,4223; 3,9831, 3,5154; 3,3373; 3,2950; 3,2695; 3,2526 (mol/L). Berdasarkan data tersebut ingin dihitung nilai konstanta laju reaksi  $k_1$ ,  $k_2$  dan  $k_{-2}$ .

**Langkah pertama:** *Pemodelan* kinetika polimerisasi urea-formaldehid berdasarkan mekanisme seri paralel bolak balik di atas dapat ditulis sebagai berikut:

$$-dC_F/dt = k_1 C_U C_F \quad (1)$$

$$+dC_M/dt = k_1 C_U C_F - k_2 C_U C_M + k_{-2} C_P \quad (2)$$

Jika ratio  $C_{F0}:C_{U0} = 3:2$ , berdasarkan kaedah neraca massa maka kedua persamaan terakhir di atas dapat disederhanakan menjadi :

$$dC_F/dt = k_1 C_F [ 4/3 C_{F0} - 2 C_F - C_M ] \quad (3)$$

$$dC_M/dt = [ k_1 C_F - k_2 C_M ] [-4/3 C_{F0} + 2 C_F + C_M + k_{-2} [ C_{F0} - C_F - C_M ] \quad (4)$$

Kedua persamaan diferensial (PD) terakhir di atas terlihat cukup kompleks, yang sulit diselesaikan dengan cara analitis, sehingga alternatifnya diselesaikan secara numeris.

**Langkah kedua** adalah *Programing*. Ditebak nilai  $k_1$ ,  $k_2$  dan  $k_{-2}$  dalam persamaan di atas kemudian kedua PD diselesaikan secara simultan dengan metode *Runge Kutta* (Stroud, 1987). Tebakan yang benar adalah yang memberi nilai *the sum of squared errors* (SSE) minimum. Minimasi SSE dengan tiga variabel  $k_1$ ,  $k_2$  dan  $k_{-2}$  dilakukan dengan metode *Hooke-Jeeves* multi variabel. Selanjutnya disusun program dengan memilih salah satu bahasa pemrograman, seperti bahasa QBASIC.

Algoritma hitungan di atas adalah sebagai berikut:

1. Input semua data yang menjadi ketetapan,
2. Input/baca data konsentrasi formaldehid hasil pengamatan laboratorium  $C_{Flab}$ .
3. Tebak nilai  $k_1$ ,  $k_2$  dan  $k_{-2}$
4. Selesaikan kedua PD (persamaan 3 dan 4) secara simultan dengan metode Runge Kutta
5. catat setiap konsentrasi formaldehid pada selang waktu tertentu (sesuai dengan selang waktu penelitian) yang terhitung dari langkah 4 sebagai  $C_{Fhitung}$
6. Hitung *the Sum of Squared Errors*,  
 $SSE = \sum [C_{Fhitung} - C_{Flab}]^2$
7. Tentukan dengan metode *Hooke Jeeves*, apakah SSE telah minimum ?
8. Jika SSE belum minimum maka ulangi langkah 3 s/d 7
9. Cetak hasil tebakan  $k_1$ ,  $k_2$  dan  $k_{-2}$  yang memberikan SSE minimum
10. Selesai

Setelah komputer menghitung sebanyak 804 kali iterasi, akhirnya diperoleh nilai konstanta laju rekasi  $k_1$ ,  $k_2$  dan  $k_3$  berturut-turut adalah  $9,13 \cdot 10^{-3}$ ;  $4,44 \cdot 10^{-4}$  dan  $3,36 \cdot 10^{-3}$  (L/grek/menit). List program dan print out hasil hitungan terlampir 1 (Aladin, 1998).

### Contoh Kasus 2

*Penentuan tetapan kesetimbangan untuk meramalkan komposisi gas hasil pada proses gasifikasi arang batubara.*

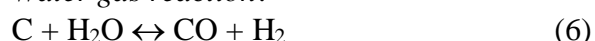
Perhitungan komposisi gas hasil dari proses gasifikasi arang batubara dapat dilakukan dengan pendekatan *stoikiometri* dan *non-stoikiometri*. Pendekatan *stoikiometri* dianggap terbaik selama persamaan reaksi diketahui (Acharya, dkk., 2010; Jarunghammachote dan Dutta, 2007; Yan, dkk., 2006).

Takdir Syarif, dkk (2018) telah melakukan penelitian mengenai penggunaan model kesetimbangan kimia dalam meramalkan komposisi gas hasil gasifikasi. Dalam penelitian tersebut digunakan persamaan reaksi kimia sebagai berikut:

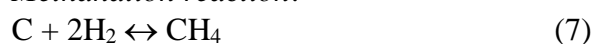
*Boudouard reaction:*



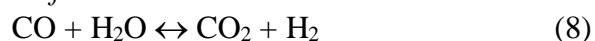
*Water gas reaction:*



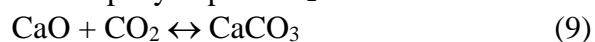
*Methanation reaction:*



*Shift Conversion:*



Reaksi penyerapan  $CO_2$ :



**Langkah pertama Pemodelan.** Dari persamaan 5 dan 6 selanjutnya disusun persamaan kesetimbangan sebagai berikut:

$$K_1 = \frac{y_{CO}^2}{y_{CO_2}} P \quad (10)$$

$$K_2 = \frac{y_{CO} \cdot y_{H_2}}{y_{H_2O}} P \quad (11)$$

$$K_3 = \frac{y_{CH_4}}{y_{H_2}^2} \frac{1}{P} \quad (12)$$

$$K_4 = \frac{1}{y_{CO_2}} \quad (13)$$

Agar komposisi dari masing-masing gas dapat dihitung, diperlukan nilai tetapan kesetimbangan  $K_1$  sampai  $K_4$ . Nilai dari masing-masing tetapan kesetimbangan dapat didekati dengan persamaan empiris sebagai berikut (Smith et al., 2005):

$$K_i = \exp\left(K_{0i} + \frac{K_{1i}}{T}\right) \quad (14)$$

Dengan menyusun kembali persamaan (10) – (13) akan diperoleh:

$$f_1 = K_1 y_{CO_2} - P y_{CO}^2 = 0 \quad (15)$$

$$f_2 = K_2 y_{H_2O} - P y_{CO} y_{H_2} = 0 \quad (16)$$

$$f_3 = K_3 y_{H_2}^2 P - y_{CH_4} = 0 \quad (17)$$

$$f_4 = K_4 y_{CO_2} - 1 = 0 \quad (18)$$

Nilai dari tetapan kesetimbangan  $K_i$  diperoleh dari persamaan (14) sedangkan nilai tetapan  $K_{0i}$  dan  $K_{1i}$  diperoleh dengan menggunakan metode kwadrat terkecil. Agar persamaan (15) – (16) dapat diselesaikan, dibutuhkan konstrain yang diperoleh dari penyusunan neraca massa komponen sebagai berikut:

$$f_5 = N_C - n_g (y_{CO} + y_{CO_2} + y_{CH_4}) = 0 \quad (19)$$

$$f_6 = N_H - n_g (2y_{H_2} + 4y_{CH_4} + 2y_{H_2O}) = 0 \quad (20)$$

$$f_7 = N_O - n_g (y_{CO} + 2y_{CO_2} + y_{H_2O}) = 0 \quad (21)$$

$$f_8 = y_{H_2} + y_{CO} + y_{CO_2} + y_{CH_4} + y_{H_2O} - 1 = 0 \quad (22)$$

**Langkah kedua** adalah *Programing*. Persamaan (15) – (18) diselesaikan dengan menggunakan metode *trial and error Newton-Raphson* yang dikombinasikan dengan optimasi multi variabel *Hooke Jeeves*. Algoritma perhitungan adalah sebagai berikut:

1. Masukkan data berupa suhu operasi (600, 700, dan 800 °C), fraksi mol gas ( $y_{i,data}$ ).
2. Masukkan nilai tebakan  $K_{0i}$  dan  $K_{1i}$ .
3. Hitung  $K_i$  masing-masing reaksi dengan menggunakan persamaan (14)
4. Masukkan nilai fraksi mol komponen tebakan  $y_{i,old}$ .
5. Hitung nilai fungsi ( $f_i$ ) Persamaan (15) – (22) pada  $y_{i,old}$ .
6. Hitung nilai turunan fungsi pada  $y_{i,old}$ .
7. Hitung  $y_{i,new}$  dengan menggunakan persamaan matriks sebagai berikut:

$$Y_{i,new} = Y_{i,old} - \frac{\partial f_i}{\partial y_i} * f_i \quad (23)$$

8. Bandingkan, apakah  $y_{i,new} \approx y_{i,old}$  ?
9. Jika ya, lanjut ke langkah 10. Jika tidak, kembali ke langkah 4, dimana  $y_{i,old} = y_{i,new}$ .
10. Hitung SSE
11. Cek, apakah SSE minimum ? . Jika Ya  $\rightarrow$  selesai. Jika tidak, update nilai  $K_{0i}$  dan  $K_{1i}$  kemudian kembali ke langkah 3

Keterangan dari persamaan (23) :

$$Y_{i,new} = \begin{bmatrix} y_{1,new} \\ y_{2,new} \\ \vdots \\ y_{n,new} \end{bmatrix}, \quad Y_{i,old} = \begin{bmatrix} y_{1,old} \\ y_{2,old} \\ \vdots \\ y_{n,old} \end{bmatrix}$$

$$\frac{\partial f_i}{\partial y_i} = \begin{bmatrix} \partial f_1 / \partial y_1 & \dots & \partial f_1 / \partial y_n \\ \partial f_2 / \partial y_1 & \dots & \partial f_2 / \partial y_n \\ \vdots & \ddots & \vdots \\ \partial f_n / \partial y_1 & \dots & \partial f_n / \partial y_n \end{bmatrix}^{-1} \quad \text{dan}$$

$$f_i = \begin{bmatrix} f_1(y_{1,old}, \dots, y_{n,old}) \\ f_2(y_{1,old}, \dots, y_{n,old}) \\ \vdots \\ f_n(y_{1,old}, \dots, y_{n,old}) \end{bmatrix}$$

Dengan menggunakan program komputer (MATLAB), diperoleh hasil perhitungan (tabel 1) dengan list program terlampir 2 (Syarif, dkk., 2018):

**Tabel 1. Nilai parameter**

Persamaan	$K_{0i}$	$K_{1i}$
5	-6.5693	$7.5001 \times 10^3$
6	11.309	$-9.0457 \times 10^3$
7	-23.734	$2.0694 \times 10^4$
9	1.6360	$1.4197 \times 10^3$

## KESIMPULAN

Persoalan matematik dalam bidang keteknik-kimia yang kompleks, penyelesaian dengan metode analitis sering sekali tidak ampuh lagi digunakan, maka alternatifnya adalah menggunakan metode numeris. Hitungan dengan metode numris yang sifatnya iteratif, yang membutuhkan *trial and error* hingga ratusan kali, membutuhkan bantuan komputer dengan bahasa pemrograman dasar seperti BASIQ atau bahasa pemrograman semi aplikasi seperti MATLAB.

## DAFTAR PUSTAKA

- Aladin, A., 1998, "Kinetika Polimerisasi Urea Formaldehid Berdasarkan Mekanisme Reaksi Seri Paralel Bolak-Balik", Tesis S2, Teknik Kimia, UGM Yogyakarta.
- Acharya, B., Dutta, A., Basu, P., 2010, "An investigation into steam gasification of biomass for hydrogen enriched gas

- production in presence of CaO*". Int. J. Hydrogen Energy 35, 1582–1589.
- Beers, K., J., 2007, "Numerical Methods for Chemical Engineering, Applications in MATLAB", CAMBRIDGE University Press.
- Davis, M.E., "Numerical Methods and Modeling for Chemical Engineers" John Wiley & Sons, New York.
- Insap Santosa, "Quick Basic", penerbit ANDI Yogyakarta.
- Jarungthammachote, S., Dutta, A.Ã., 2007, "Thermodynamic equilibrium model and second law analysis of a downdraft waste gasifier" 32, 1660–1669.
- Jeffreys, G.V. and Jenson, V.G., "Mathematical Methods in Chemical Engineering", Academic Press London
- Sediawan, W.B. dan Prasetya, A., 1997, "Pemodelan Matematis dan Penyelesaian Numeris Dalam Teknik Kimia", penerbit ANDI, Yogyakarta.
- Stroud, K.A., 1987, "Engineering Mathematics", 3rd edition, the Macmillan Press Ltd.
- Syarif, T., Sulisty, H., Sediawan, W. B., Budhijanto., 2018, "Thermochemical Equilibrium Modelling of Steam Gasification of Char From Pattukku Coal Using Cao as Co Absorbent", IOP Conf. Ser.: Earth Environ. Sci. 175 012027.
- Yan, Q., Guo, L., Lu, Y., 2006. Thermodynamic analysis of hydrogen production from biomass gasification in supercritical water. Energy Convers. Manag. 47, 1515–1528.

**Lampiran 1:**

```

CLS
tt = 160
DIM k(30), R(30), T(tt + 5), CFlab(30), CM(tt + 5)
DIM CF(tt + 5), CP(tt + 5), CU(tt + 5)
BISMILLAH:
  GOSUB Tebakan
  FOR lab = 1 TO 8
    READ CFlab(lab)  '...membaca data penelitian
  NEXT lab
  GOSUB RungeKutta  '...menghitung 2 PD simultan
  SSEmin = y
  GOSUB cetak
  GOSUB HookJeeves  '...mencari k1,k2 dan k-2 optimum
  GOSUB ralat
  GOSUB Kesimpulan
END '-----

HookJeeves:
cetak = 1
75 '....ekspolarasi:
  kodel = 0: kode2 = 0: kode3 = 0
  k1 = klopt + delk1: k2 = k2opt: k3 = k3opt
  GOSUB RungeKutta
  IF y >= SSEmin THEN 100
  klopt = k1: SSEmin = y: kode1 = 1
  GOTO 125
100  k1 = klopt - delk1
  GOSUB RungeKutta
  IF y >= SSEmin THEN 125
  klopt = k1: SSEmin = y: kode1 = -1
125  k1 = klopt: k2 = k2opt + delk2: k3 = k3opt
  GOSUB RungeKutta
  IF y >= SSEmin THEN 150
  k2opt = k2: SSEmin = y: kode2 = 1
  GOTO 175
150  k2 = k2opt - delk2
  GOSUB RungeKutta
  IF y >= SSEmin THEN 175
  k2opt = k2: SSEmin = y: kode2 = -1
175  k1 = klopt: k2 = k2opt: k3 = k3opt + delk3
  GOSUB RungeKutta
  IF y >= SSEmin THEN 200
  k3opt = k3: SSEmin = y: kode3 = 1
  GOTO 225
200  k3 = k3opt - delk3
  GOSUB RungeKutta
  IF y >= SSEmin THEN 225
  k3opt = k3: SSEmin = y: kode3 = -1
225  IF ABS(kode1) > .2 THEN 250
  IF ABS(kode2) > .2 THEN 250
  IF ABS(kode3) > .2 THEN 250
  IF delk1 < tolk1 AND delk2 < tolk2 AND delk3 < tolk3 THEN 300
  '.....kecilkan delta
  delk1 = delk1 * rasio: delk2 = rasio * delk2: delk3 = rasio * delk3
  GOTO 75 'untuk ekspolarasi lebih lanjut....!
250  'mengulang langkah sukses
  k1 = klopt + kode1 * delk1
  k2 = k2opt + kode2 * delk2
  k3 = k3opt + kode3 * delk3
  GOSUB RungeKutta

```

```

    IF y >= SSEmin THEN 75
    klopt = k1: k2opt = k2: k3opt = k3: SSEmin = y
    GOTO 250
300  RETURN

```

**RungeKutta:**

```

T(0) = t0: CN(0) = CM0: CF(0) = CF0: CP(0) = CP0: CU(0) = CU0
FOR menit = 0 TO tt
  T = T(menit): CF = CF(menit): CU = CU(menit): CM = CM(menit):CP = CP(menit)
  GOSUB PD
  RKa1 = PD1b * delT: RKb1 = PD2b * delT
  T = T(menit) + delT / 2:CF = CF(menit) + RKa1 / 2:CM = CM(menit) + RKb1 / 2
  GOSUB PD
  RKa2 = PD1b * delT: RKb2 = PD2b * delT
  T = T(menit) + delT / 2:CF = CF(menit) + RKa2 / 2:CM = CM(menit) + RKb2 / 2
  GOSUB PD
  RKa3 = PD1b * delT: RKb3 = PD2b * delT
  T = T(menit) + delT / 2: CF = CF(menit) + RKa3: CM = CM(menit) + RKb3
  GOSUB PD
  RKa4 = PD1b * delT: RKb4 = PD2b * delT
  T(menit + 1) = T(menit) + delT
  CF(menit + 1) = CF(menit) + (RKa1 + 2 * RKa2 + 2 * RKa3 + RKa4) / 6
  CM(menit + 1) = CM(menit) + (RKb1 + 2 * RKb2 + 2 * RKb3 + RKb4) / 6
  T = T(menit + 1): CF = CF(menit + 1): CM = CM(menit + 1)
  CP(menit + 1) = CF0 - CF - CM
  CU(menit + 1) = CU0 - (CM + 2 * CP(menit + 1))
NEXT menit
sse = 0
FOR i = 1 TO 8
  sse = sse + (CF(15 * i) - CFlab(i)) ^ 2
NEXT i
y = sse: No = No + 1
RETURN

```

```

PD: PD1b = k1 * CF * (4 / 3 * CF0 - 2 * CF - CM)
    PD2b = (k1 * CF - k2 * CM) * (-4 / 3 * CF0 + 2 * CF + CM)
    PD2b = PD2b + k3 * (CF0 - CF - CM)

```

```
RETURN
```

```
Tebakan:
```

```
klopt = .001: k2opt = .001: k3opt = .001
```

```
KondisiReaksi:
```

```
suhu = 90: pH = 8: ratio = 1.5
```

```
Kondisiawal:
```

```
t0 = 0: CM0 = 0: CP0 = 0
```

```
CF0 = 12.4074 * 50 / 68: CU0 = CF0 / ratio
```

```
Ketentuan:
```

```
delk1 = .0001: delk2 = .0001: delk3 = .00001
```

```
tolk1 = 1E-15: tolk2 = .00001: tolk3 = 1E-15
```

```
k1 = klopt: k2 = k2opt: k3 = k3opt
```

```
delT = 1: rasio = .6: No = 0: jumlahdata = 8
```

```
RETURN
```

```
ralat:
```

```
k1 = klopt: k2 = k2opt: k3 = k3opt: GOSUB RungeKutta
```

```
SSEmin = y: GOSUB cetak
```

```
'PRINT " No Waktu CFlab CFhit CUhit CMhit CPhit Ralat"
```

```
'PRINT " (menit) (grek/L) (grek/L) (grek/L) (grek/L) (grek/L) (%)"
```

```
'PRINT "-----"
```

```
mus$ = "## ## ##.### ##.### ##.### ##.### ##.### ##.### ##.###" "
```

```
ralat = 0
```

```
FOR k = 0 TO jumlahdata
```

```
T = 15 * k: CFlab(0) = CF(0)
```

```
R(k) = ABS((CFlab(k) - CF(T)) / CF(T)) * 100
```



```

    'PRINT USING mus$; k; T; CFlab(k); CF(T); CU(T); CM(T); CP(T); R(k)
    IF k > 8 OR k < 1 THEN GOTO 420
    ralat = ralat + R(k)
420 NEXT k
ralat = ralat / jumlahdata
'PRINT "-----"
'PRINT "Ralat rerata                               ="; ralat
RETURN
Kesimpulan:
PRINT: PRINT "Kesimpulan:"
PRINT "Kondisi polimerisasi urea-formaldehid:"
PRINT "pH ="; pH; ", suhu T :"; suhu;
PRINT "oC, ratio F:U = "; ratio
PRINT "Maka diperoleh konstanta laju  :"
PRINT "konstanta tahap adisi, k1                :"; k1; " (L/grek/menit)"
PRINT "konstanta tahap kondensasi arah maju, k2  :"; k2; " (L/grek/menit)"
PRINT "konstanta tahap kondensasi arah balik, k-2: "; k3; " (L/grek/menit)"
PRINT "Persen Ralat                             :"; ralat; "%"
PRINT "The Sum of Squared Errors (SSE)          :"; SSEmin
RETURN
DATA LABORATORIUM :
DATA 5.3081,4.4223,3.9831,3.5154,3.3373,3.2950,3.2695,3.2526

```

### Hasil Running:

```

Kesimpulan:
Kondisi polimerisasi urea-formaldehid:
pH = 8 , suhu T : 90 oC, ratio F:U = 1.5
Maka diperoleh konstanta laju  :
konstanta tahap adisi, k1                : 9.13357E-03 (L/grek/menit)
konstanta tahap kondensasi arah maju, k2  : 4.443807E-04 (L/grek/menit)
konstanta tahap kondensasi arah balik, k-2: 3.362608E-03 (L/grek/menit)
Persen Ralat                             : 1.754841 %
The Sum of Squared Errors (SSE)          : 6.292847E-02

```

### Lampiran 2:

#### Listing Program Evaluasi Tetapan Kesetimbangan:

```

function Model_K_CaO
clc
clear all
global channel lokasi1 lokasi2 lokasi3
disp('-----');
disp('1. Suhu 600 oC                               ');
disp('2. Suhu 700 oC                               ');
disp('3. Suhu 800 oC                               ');
disp('4. Semua Suhu (600, 700, 800 oC)             ');
disp('5. Exit                                       ');
disp('-----');
pilih=input('Tekan Nomor : ');
if pilih>=5

```

```
disp('<<< Selesai >>>');
else
channel = ddeinit('excel','Publikasi_1_International.xlsx');
if pilih==1
disp('<< Running Suhu 600 oC >>');
y_data = ddereq(channel,'r3c2:r7c2');
var_optim = ddereq(channel,'r20c3:r23c3');
x = ddereq(channel,'r20c2:r24c2');
data_konst = ddereq(channel,'r12c2:r15c2');
lokasi1='r3c5:r7c5';
lokasi2='r20c6:r24c6';
lokasi3='r9c5:r9c5';
elseif pilih==2
disp('<< Running Suhu 700 oC >>');
y_data = ddereq(channel,'r3c3:r7c3');
var_optim = ddereq(channel,'r20c4:r23c4');
x = ddereq(channel,'r20c2:r24c2'); %Fraksi mol tebakan
data_konst = ddereq(channel,'r12c3:r15c3');
lokasi1='r3c6:r7c6';
lokasi2='r20c7:r24c7';
lokasi3='r9c6:r9c6';
elseif pilih==3
disp('<< Running Suhu 800 oC >>');
y_data = ddereq(channel,'r3c4:r7c4');
var_optim = ddereq(channel,'r20c5:r23c5');
x = ddereq(channel,'r20c2:r24c2'); %Fraksi mol tebakan
data_konst = ddereq(channel,'r12c4:r15c4');
lokasi1='r3c7:r7c7';
lokasi2='r20c8:r24c8';
lokasi3='r9c7:r9c7';
%else
%disp('<< Running Semua Suhu >>');
end
lb=[0;0;0;0];
ub=[];
option=optimset('Algorithm','interior-
point','DerivativeCheck','on','GradConstr','on','TolX',1e-15);

[K_hit,fmin2,exitflag,output]=fmincon(@(var_optim)
hit_SSE(var_optim,x,y_data,data_konst),var_optim,[],[],[],[],lb,ub,[],option);
end
```

```
function SSE=hit_SSE(var_optim,x,y_data,data_konst)
global channel lokasi1 lokasi2 lokasi3
options=optimset('MaxFunEvals',1000,'Algorithm','levenberg-marquardt');
[x,fval] = fsolve(@(x) hit_fraksi(x,var_optim,data_konst),x,options);
SSE=sum(((y_data-x)./y_data).^2);
rc = ddepoke(channel,lokasi1,x);
rc = ddepoke(channel,lokasi2,var_optim);
rc = ddepoke(channel,lokasi3,SSE);
function Fraksimol=hit_fraksi(x,var_optim,data_konst)
K1=var_optim(1);
K2=var_optim(2);
K3=var_optim(3);
K5=var_optim(4);
yCH4=x(1);
yCO=x(2);
yH2=x(3);
yCO2=x(4);
yH2O=x(5);
C=data_konst(1);
H=data_konst(2);
O=data_konst(3);
NG=data_konst(4);
F1=K1*yCO2-yCO^2;
F2=K2*yH2O-yH2*yCO;
F3=K3*yH2^2-yCH4;
F5=1-K5*yCO2;
F6=yCO+yCO2+yCH4-C/NG;
F7=2*yH2+4*yCH4+2*yH2O-H/NG;
F9=yCO2+yCO+yH2+yH2O+yCH4-1;
Fraksimol=[F1;F2;F3;F5;F6;F7;F9];
```