

# Very Good !!!

## KINETIKA REAKTOR

### KOMPUTASI NUKLIR



Anggota Kelompok 03:

Bagas Yadher Bima N.A.R.H	18/431318/TK/47911
Imam Bayu Prasetya	18/425231/TK/46926
Muhammad Farhan Ramadhany	18/431325/TK/47918
Muhammad Syafiq Fauzan	18/428979/TK/47481
Valentinus Elzha Widatama	18/425242/TK/46937

**PROGRAM STUDI TEKNIK NUKLIR**  
**DEPARTEMEN TEKNIK NUKLIR DAN TEKNIK FISIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS GADJAH MADA**  
**YOGYAKARTA**  
**2021**

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>2</b>
<b>DESKRIPSI MASALAH</b>	<b>1</b>
<b>METODE PENYELESAIAN MASALAH</b>	<b>3</b>
Pemodelan Persamaan	3
Algoritma Program	4
Program utama	4
Subprogram Runge-Kutta orde 6	4
Subprogram sistem persamaan diferensial	5
Diagram Alir Program	7
Program Utama	7
Program Runge-Kutta	8
Program Fungsi	9
<b>IMPLEMENTASI PROGRAM</b>	<b>10</b>
6 Grup	10
Program Utama	10
Program Runge-Kutta	11
Program Fungsi	11
1 Grup	13
Program Utama	13
Program Runge-Kutta	14
Program Fungsi	15
<b>HASIL DAN PEMBAHASAN</b>	<b>16</b>
Hasil	16
Variasi nilai $\rho$ dari $0.2\beta$ , ke $0\beta$ , kemudian ke $-0.2\beta$	16
Variasi nilai $\rho$ dari $0.5\beta$ , ke $0\beta$ , kemudian ke $-0.5\beta$	19
Variasi nilai $\rho$ dari $0.9\beta$ , ke $0\beta$ , kemudian ke $-0.9\beta$	22
Variasi nilai $\rho$ dari $1.1\beta$ , ke $0\beta$ , kemudian ke $-1.1\beta$	25
Pembahasan	28
<b>KESIMPULAN DAN SARAN</b>	<b>30</b>
Kesimpulan	30
Saran	30
<b>DAFTAR ACUAN</b>	<b>31</b>

## A. DESKRIPSI MASALAH

Model reaktor titik merupakan pemodelan yang dapat digunakan untuk mengetahui perubahan daya reaktor ketika terdapat penambahan atau pengurangan reaktivitas ke dalam reaktor. Model reaktor titik merupakan penyederhanaan dari perhitungan persamaan transport atau difusi gayut waktu di mana parameter reaktor dianggap seragam di semua bagian reaktor. Perubahan pada satu titik di reaktor dengan serta merta akan mempengaruhi parameter yang lain. Tentu saja pada model ini terdapat keterbatasan, akan tetapi sebagai tinjauan awal cukup memberikan gambaran tentang dinamika yang terjadi pada reaktor.

Model dinamika yang mencerminkan kondisi reaktor ini disebut dengan dinamika reaktor titik, yang biasa dinyatakan dengan persamaan berikut:

$$\frac{dP}{dt} = \frac{\rho - \beta}{\Lambda} P + \sum_i \lambda_i C_i \quad (1)$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{\Lambda} P - \lambda_i C_i, \quad i = 1, \dots, N_d \quad (2)$$

di mana,

$P$  : daya reaktor

$\rho$  : reaktivitas yang ditambahkan terhadap reaktor

$\beta_i$  : fraksi neutron kasip kelompok ke- $i$

$\Lambda$  : *mean generation time*

$\lambda_i$  : konstanta peluruhan nuklida penghasil neutron kasip kelompok ke- $i$

$C_i$  : daya yang dihasilkan oleh nuklida penghasil neutron kasip kelompok ke- $i$

$N_d$  : banyaknya kelompok neutron kasip

**Tabel 1.** Data neutron kasip.

Kelompok	Energi (MeV)	$T_{1/2}$ (s)	$\lambda_i$ (s <sup>-1</sup> )	$\beta_i$	$\beta_i/\lambda_i$ (s)
1	0,25	55,7	0,0124	0,000215	0,0173
2	0,56	22,7	0,0305	0,001420	0,0466
3	0,43	6,22	0,111	0,001270	0,0114
4	0,62	2,30	0,301	0,002570	0,0085
5	0,42	0,61	1,1	0,000750	0,0007
6	0,23	0,23	3,0	0,000270	0,0001

Untuk perhitungan yang detail, biasanya digunakan enam buah kelompok neutron kasip ( $N_d = 6$ ) dengan data-data tercantum pada Tabel 1. Akan tetapi untuk memudahkan perhitungan, sering kali parameter dinamis ( $\lambda_i$  dan  $\beta_i$ ) diubah menjadi nilai efektifnya.

Seringkali perhitungan dilakukan dengan menggunakan satu kelompok efektif neutron kasip saja. Parameter dinamis dinyatakan sebagai berikut:

Berdasarkan persamaan kinetika reaktor tersebut, diimplementasikan persamaan dinamika titik dan disimulasikan terhadap kondisi reaktor sebagai berikut:

1. Reaktor beroperasi pada kondisi *zero power* dengan daya 1 W. Nilai  $\Lambda = 5 \times 10^{-5}$  detik.
2. Pada saat  $t = 0$ , reaktivitas positif sebesar  $\rho = 0,2\beta$  ditambahkan ke dalam reaktor dan dipertahankan konstan.
3. Pada saat  $t = 10$  detik, batang kendali dimasukkan ke dalam teras reaktor sehingga reaktivitas menjadi  $\rho = 0$  dan dipertahankan konstan.
4. Pada saat  $t = 20$  detik, batang kendali dimasukkan lebih dalam sehingga reaktivitas menjadi  $\rho = -0,2\beta$  dan dipertahankan konstan.

Hasil simulasi program kemudian disusun dan ditentukan nilai dari daya sebagai fungsi waktu selama 60 detik semenjak penambahan reaktivitas positif dengan nilai *time step* dalam orde mikro sekon.

Simulasi dilakukan dengan menggunakan model 6 kelompok dan 1 kelompok neutron kasip (menggunakan  $\lambda_{\text{eff},1}$  dan  $\lambda_{\text{eff},2}$ ). Hasil ketiga perhitungan ditampilkan pada satu plot yang sama kemudian dibandingkan antara kedua hasil tersebut.

Percobaan dilakukan pengulangan sesuai dengan langkah-langkah 2-4 di atas, dengan penambahan reaktivitas positif sebesar 0,  $5\beta$ , 0,  $9\beta$  dan 1,  $1\beta$  serta reaktivitas negatif sebesar  $-0,5\beta$ ,  $-0,9\beta$  dan 1,  $1\beta$ . Hasil yang diperoleh kemudian dibandingkan untuk mencari tahu apakah terdapat perbedaan yang signifikan dari penambahan reaktivitas.

## B. METODE PENYELESAIAN MASALAH

- Pemodelan Persamaan

Persamaan dinamika reaktor titik dinyatakan dengan persamaan:

$$\frac{dP}{dt} = \frac{\rho - \beta}{\Lambda} P + \sum_i \lambda_i C_i \quad (1)$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{\Lambda} P - \lambda_i C_i \quad i = 1, \dots, N_d \quad (2)$$

umur paro  
prekursor

Dari persamaan tersebut menggunakan enam buah kelompok neutron berdasarkan tingkat energinya sehingga nilai  $N_d = 6$ . Nilai dari masing-masing kelompok kemudian dijabarkan menjadi persamaan berikut:

$$\frac{dC_1}{dt} = \frac{\beta_1}{\Lambda} P - \lambda_1 C_1 \quad (3)$$

$$\frac{dC_2}{dt} = \frac{\beta_2}{\Lambda} P - \lambda_2 C_2 \quad (4)$$

$$\frac{dC_3}{dt} = \frac{\beta_3}{\Lambda} P - \lambda_3 C_3 \quad (5)$$

$$\frac{dC_4}{dt} = \frac{\beta_4}{\Lambda} P - \lambda_4 C_4 \quad (6)$$

$$\frac{dC_5}{dt} = \frac{\beta_5}{\Lambda} P - \lambda_5 C_5 \quad (7)$$

$$\frac{dC_6}{dt} = \frac{\beta_6}{\Lambda} P - \lambda_6 C_6 \quad (8)$$

Untuk memudahkan dalam hal simulasi enam kelompok neutron kasip dari persamaan di atas maka dilakukan vektorisasi dari nilai-nilai energi neutron, fraksi neutron kasip, konstanta peluruhan nuklida penghasil neutron kasip, dan daya yang dihasilkan oleh nuklida penghasil neutron kasip. Hasil dari perhitungan daya yang dihasilkan oleh nuklida penghasil neutron kasip ( $C_i$ ) kemudian dilakukan perhitungan daya reaktor ( $P$ ) dengan vektorisasi daya neutron kasip dan konstanta peluruhan tiap kelompok energi.

Setelah dilakukan penyusunan persamaan neutron kasip untuk enam kelompok energi, dilakukan simulasi terhadap kondisi reaktor sebagai berikut:

1. Kondisi *zero power* dengan daya 1 W. Nilai  $\Lambda = 5 \times 10^{-5}$  detik.
2. Saat  $t = 0$ ,  $\rho = 0,2\beta$  dan dipertahankan konstan.
3. Saat  $t = 10$  detik,  $\rho = 0$  dan dipertahankan konstan.

4. Saat  $t = 20$  detik,  $\rho = -0, 2\beta$  dan dipertahankan konstan.

Jumlah prekursor 6 grup pada kondisi awal dengan persamaan  $C(0) = C_0 = \beta_i P_0 / \lambda_i \Lambda$  [1].

Tambahkan  
satuan

Jumlah Awal Prekursor 6 Grup			
Kelompok, i	$\lambda_i$	$\beta_i$	$C_i(0)$
1	0.0124	0.000215	347
2	0.0305	0.00142	931
3	0.111	0.00127	229
4	0.301	0.00257	171
5	1.1	0.00075	14
6	3	0.00027	2

Jumlah prekursor 1 grup pada kondisi awal :

C1 dan C2 lebih baik ditulis  
menjadi  $C_{eff1}$  dan  $C_{eff2}$

Jumlah Awal Prekursor 1 Grup	
$\beta$	0.006495
$\lambda_{eff1}$	0.0767299
$\lambda_{eff2}$	0.3996176
$C_1(0)$	1692.951
$C_2(0)$	325.0608

Percobaan diulang dengan variasi penambahan reaktivitas positif sebesar  $0, 5\beta, 0, 9\beta$  dan  $1, 1\beta$  serta reaktivitas negatif sebesar  $-0, 5\beta, -0, 9\beta$  dan  $-1, 1\beta$ .

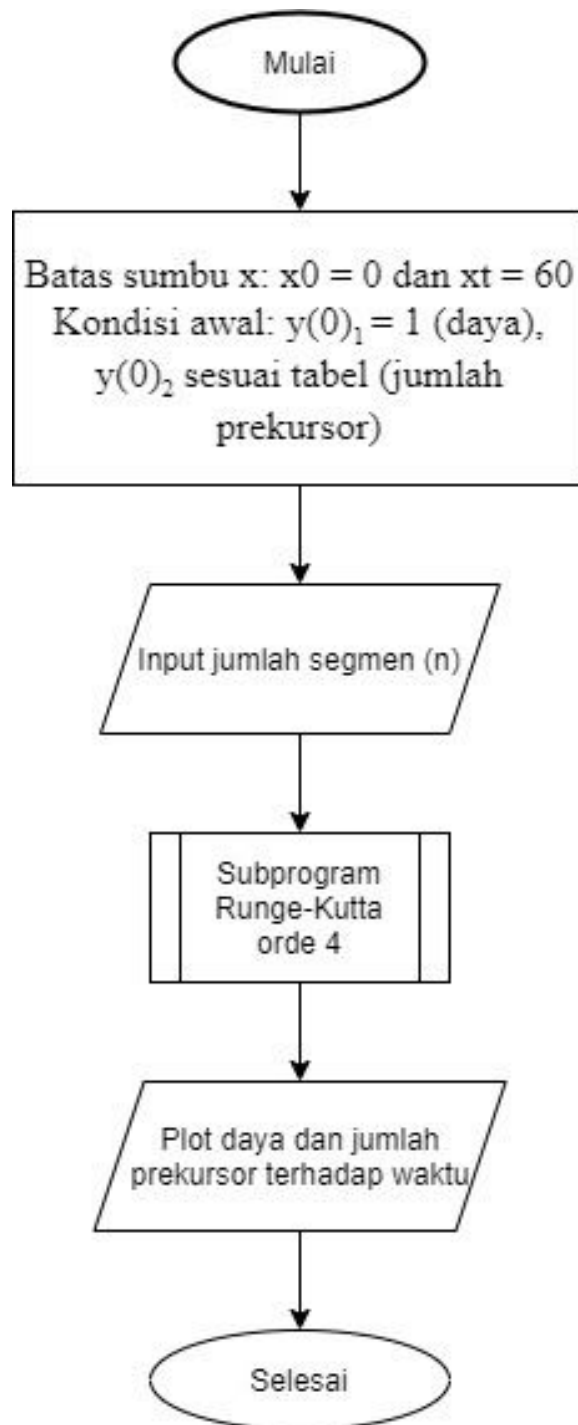
- Algoritma Program
  - Program utama
    1. Tentukan kondisi awal tiap sumbu dari sistem persamaan diferensial :
      - a. Batas awal sumbu x,  $x_0 = 0$  dan  $x_t = 60$
      - b. Batas awal sumbu y,  $y(0)_1 = 1$  (daya),  $y(0)_2$  sesuai tabel (jumlah prekursor)
    2. Inputkan jumlah segmen (n),
    3. Masuk ke subprogram metode Runge-Kutta orde 6,
    4. Plot daya dan jumlah prekursor terhadap waktu.
  - Subprogram Runge-Kutta orde 6
    1. Hitung lebar interval tiap segmen,  $h = (x_t - x_0)/n$ ,
    2. Buat matriks sebagai tebakan pertama :
      - a.  $Y = \text{zeros}(n + 1, \text{length}(y_0))$
      - b.  $T = \text{zeros}(n + 1, 1)$

3. Definisikan syarat awal :
    - a.  $T(1) = x_0$ ;
    - b.  $Y(1,:) = y_0$ ;
  4. Untuk  $i = 1$  sampai dengan  $i \leq n$ , lakukan :
    - a. Hitung nilai konstanta lalu masuk ke subprogram fungsi :
      - i.  $k_1 = h * f(T(i), Y(i,:))$
      - ii.  $k_2 = h * f(T(i) + 1/4 * h, Y(i,:) + 1/4 * k_1)$
      - iii.  $k_3 = h * f(T(i) + 1/4 * h, Y(i,:) + 1/8 * k_1 + 1/8 * k_2)$
      - iv.  $k_4 = h * f(T(i) + 1/2 * h, Y(i,:) - 1/2 * k_2 + k_3)$
      - v.  $k_5 = h * f(T(i) + 3/4 * h, Y(i,:) + 3/16 * k_1 + 9/16 * k_4)$
      - vi.  $k_6 = h * f(T(i) + h, Y(i,:) - 3/7 * k_1 + 2/7 * k_2 + 12/7 * k_3 - 12/7 * k_4 + 8/7 * k_5)$
    - b. Hitung nilai Y dan T pada segmen berikutnya :
      - i.  $T(i + 1) = T(i) + h$
      - ii.  $Y(i+1,:) = Y(i,:) + 1/90 * (7 * k_1 + 32 * k_3 + 12 * k_4 + 32 * k_5 + 7 * k_6)$
  5. Kembali ke program utama.
- Subprogram sistem persamaan diferensial
1. Nilai x dan y didapatkan dari subprogram Runge-Kutta,
  2. Definisikan konstanta tiap grup seperti  $\Lambda$ ,  $\lambda_i$ ,  $\beta_i$ , dan  $\beta_i/\lambda_i$  :
    - a.  $li(\lambda_i) = [0.0124 \ 0.0305 \ 0.111 \ 0.301 \ 1.1 \ 3.0]$
    - b.  $Bi(\beta_i) = [0.000215 \ 0.001420 \ 0.001270 \ 0.002570 \ 0.000750 \ 0.000270]$
    - c.  $Bli(\beta_i/\lambda_i) = Bi ./ li$
    - d.  $\text{lamb}(\Lambda) = 5 * 10^{-5}$
  3. Hitung parameter dinamis  $\beta$ ,  $\lambda_{eff1}$ , dan  $\lambda_{eff2}$  :
    - a.  $B(\beta) = \text{sum}(Bi)$
    - b.  $\lambda_{eff1}(\lambda_{eff1}) = (1/B * \text{sum}(Bi ./ li))^{-1}$ ;
    - c.  $\lambda_{eff2}(\lambda_{eff2}) = 1/B * \text{sum}(Bi .* li)$ ;
  4. Jika nilai  $x < 10$ , maka  $\rho(\rho) = 1.1 * B$ ,
  5. Jika nilai  $x \geq 20$ , maka  $\rho = -1.1 * B$ ,
  6. Alih-alih  $\rho = 0$ ,
  7. Hitung nilai dari sistem persamaan diferensial berikut :
    - a.  $f_{xy}(1) = (\rho - B)/\text{lamb} * y(1) + li(1) * y(2) + li(2) * y(3) + li(3) * y(4) + li(4) * y(5) + li(5) * y(6) + li(6) * y(7)$ ;
    - b.  $f_{xy}(2) = (Bi(1)/\text{lamb}) * y(1) - li(1) * y(2)$ ;
    - c.  $f_{xy}(3) = (Bi(2)/\text{lamb}) * y(1) - li(2) * y(3)$ ;
    - d.  $f_{xy}(4) = (Bi(3)/\text{lamb}) * y(1) - li(3) * y(4)$ ;
    - e.  $f_{xy}(5) = (Bi(4)/\text{lamb}) * y(1) - li(4) * y(5)$ ;

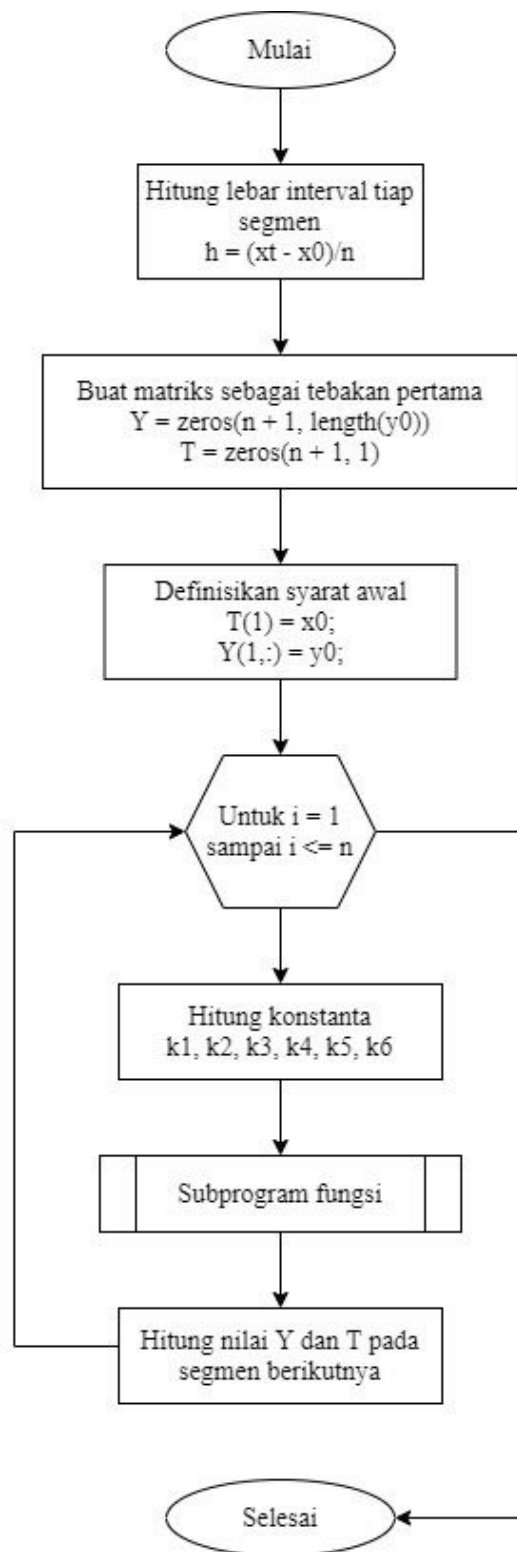
- f.  $fx_y(6) = (Bi(5)/lamb)*y(1) - li(5) * y(6);$
- g.  $fx_y(7) = (Bi(6)/lamb)*y(1) - li(6) * y(7);$
- 8. Kembali ke subprogram Runge-Kutta.



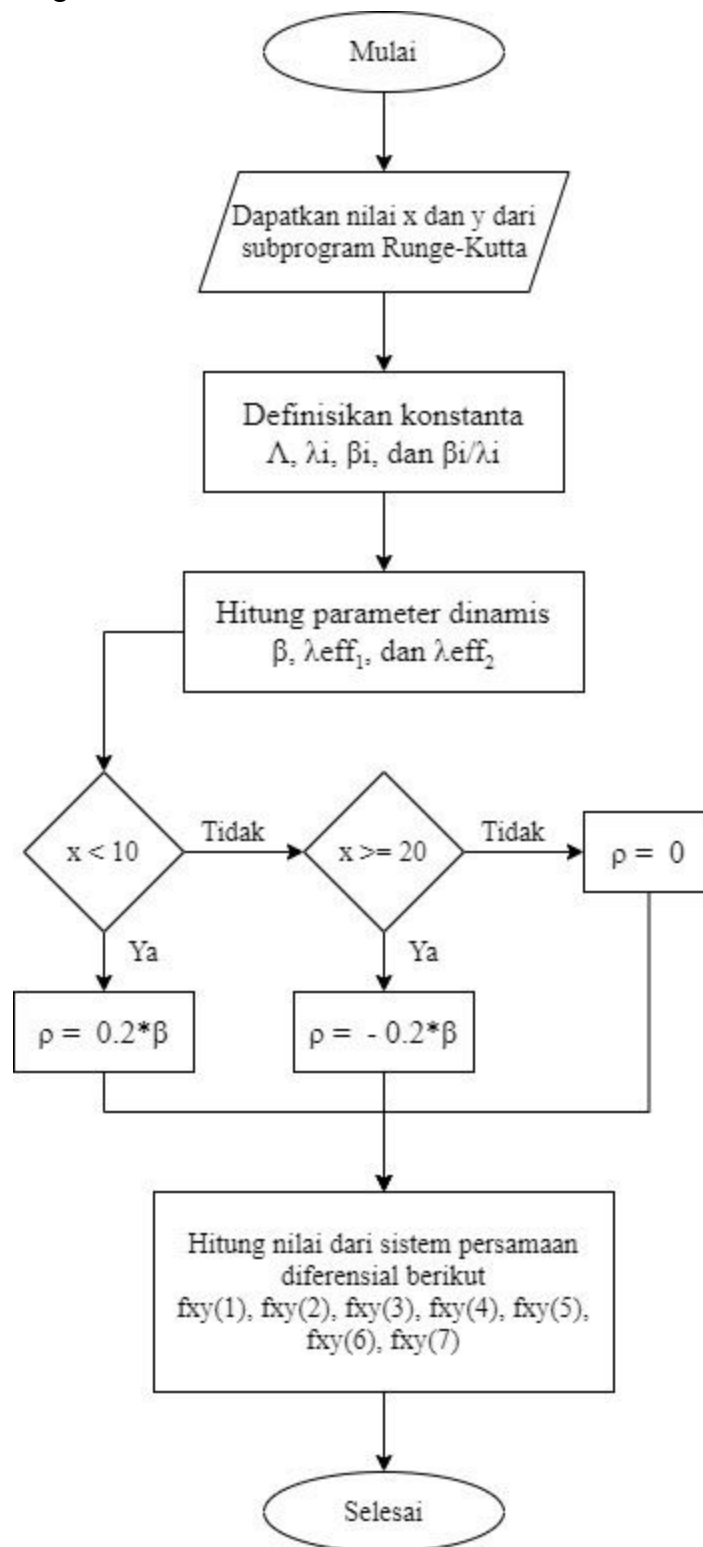
- Diagram Alir Program
  - Program Utama



- Program Runge-Kutta



○ Program Fungsi



### C. IMPLEMENTASI PROGRAM

- 6 Grup
  - Program Utama

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Utama;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;clc;clf;

%Y1(0)=daya
%Y2(0)=prekursor

x0 = 0.0;                %Waktu awal / Batas awal
t  = 60;                 %Waktu akhir
y0 = [1 347 931 229 171 14 2];      %Syarat batas awal
n  = input('Masukan jumlah segment (n) : ') %Lebar interval

tic;
[x,yr] = rk(@f, x0, y0, t, n); %Masuk ke subprogram RK.m

%plot daya
subplot(2,1,1);
plot(x,yr(:,1),'m')
legend('Daya');
xlabel('t(s)');
ylabel('P(t)');
xlim([x0 t]); %ylim([0 2.5]);
title('Daya');

%plot prekursor
subplot(2,1,2);
plot(x,yr(:,2),'k', x,yr(:,3),'r', x,yr(:,4),'g', x,yr(:,5),'b',
x,yr(:,6),'c', x,yr(:,7),'m')
legend('C1','C2','C3','C4','C5','C6');
xlabel('t(s)');
ylabel('C(t)');
xlim([x0 t]); %ylim([0 2.5]);
title('Jumlah Prekursor');
toc;
```

○ Program Runge-Kutta

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Runge Kutta Orde 6
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T,Y] = rk(f, x0, y0, t, n)

%fungsi Runge Kutta orde 6
%penyelesaian numerik persamaan diferensial dengan metode Runge
Kutta orde 6

h = (t-x0)/n;           %Banyak segmen
Y = zeros(n+1, length(y0)); % variabel terikat awal
T = zeros(n+1, 1); % variabel waktu awal

%syarat awal
T(1) = x0;
Y(1,:) = y0;

%jalankan sebanyak n langkah
for i = 1:n
    k1 = h*f(T(i), Y(i,:));
    k2 = h*f(T(i) + 1/4*h, Y(i,:) + 1/4*k1);
    k3 = h*f(T(i) + 1/4*h, Y(i,:) + 1/8*k1 + 1/8*k2);
    k4 = h*f(T(i) + 1/2*h, Y(i,:) - 1/2*k2 + k3);
    k5 = h*f(T(i) + 3/4*h, Y(i,:) + 3/16*k1 + 9/16*k4);
    k6 = h*f(T(i) + h, Y(i,:) - 3/7*k1 + 2/7*k2 + 12/7*k3 - 12/7*k4
+ 8/7*k5);
    %Menentukan Y pada segmen berikutnya
    Y(i+1,:) = Y(i,:) + 1/90 * (7*k1 + 32*k3 + 12*k4 + 32*k5 +
7*k6);

    %Menentukan T pada segmen berikutnya
    T(i+1) = T(i) + h;
endfor
end

```

○ Program Fungsi

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Fungsi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fxy = f(x,y)

```

```

%Lambda kecil
li = [0.0124 0.0305 0.111 0.301 1.1 3.0];

%Fraksi Neutron Kasip
Bi = [0.000215 0.001420 0.001270 0.002570 0.000750 0.000270];
Bli = Bi ./ li;

%Parameter Dinamis
B = sum(Bi);

%kondisi 2
if (x < 10)
    rho = 1.1 * B;
elseif (x >= 20)
    rho = -1.1 * B;
else
    rho = 0;
endif

%Lamda Besar
lamb = 5 * 10^-5;

%Perhitungan daya
fxy(1) = (rho-B)/lamb*y(1) + li(1) * y(2) + li (2) * y(3) + li(3) *
y(4) + li(4) * y(5) + li(5) * y(6) + li(6) * y(7);

%Perhitungan jumlah prekursor 6 grup
fxy(2) = (Bi(1)/lamb)*y(1) - li(1) * y(2);
fxy(3) = (Bi(2)/lamb)*y(1) - li(2) * y(3);
fxy(4) = (Bi(3)/lamb)*y(1) - li(3) * y(4);
fxy(5) = (Bi(4)/lamb)*y(1) - li(4) * y(5);
fxy(6) = (Bi(5)/lamb)*y(1) - li(5) * y(6);
fxy(7) = (Bi(6)/lamb)*y(1) - li(6) * y(7);

end

```

- 1 Grup
  - Program Utama

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Utama;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;clc;clf;

%Y1(0)=daya
%Y2(0)=prekursor
x0 = 0.0;                                %Waktu awal / Batas
awal
t = 60;                                %Waktu akhir
y0 = [1 325.0607972];                    %Syarat batas awal
n = input('Masukan jumlah segment (n) : ') %Lebar interval

tic;
[x,yr] = rk(@f2, x0, y0, t, n); %Masuk ke subprogram RK.m

%plot Daya 1
subplot(2,1,1);
plot(x,yr(:,1),'b')
legend('Daya 2');
xlabel('t(s)');
ylabel('P(t)');
xlim([x0 t]); %ylim([0 2.5]);
title('Daya Pendekatan Lamda 2');

%subplot(2,2,2);
%plot(x,yr(:,2),'r')
%legend('Daya 2');
%xlabel('t(s)');
%ylabel('P(t)');
%xlim([x0 t]); %ylim([0 2.5]);
%title('Daya Pendekatan Lamda 2');

subplot(2,1,2);
plot(x,yr(:,2),'b')
legend('C lambda 2');
xlabel('t(s)');
ylabel('C(t)');
xlim([x0 t]); %ylim([0 2.5]);
title('Jumlah Prekursor Pendekatan Lamda 2');

```

```

%subplot(2,2,4);
%plot(x,yr(:,4),'b')
%legend('C lambda 2');
%xlabel('t(s)');
%ylabel('C(t)');
%xlim([x0 t]); %ylim([0 2.5]);
%title('Jumlah Prekursor Pendekatan Lamda 2');
toc;

```

# o Program Runge-Kutta

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Runge Kutta Orde 6
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T,Y] = rk(f, x0, y0, t, n)

%fungsi Runge Kutta orde 6
%penyelesaian numerik persamaan diferensial dengan metode Runge
Kutta orde 6

h = (t-x0)/n; %Banyak segmen
Y = zeros(n+1, length(y0)); % variabel terikat awal
T = zeros(n+1, 1); % variabel waktu awal

%syarat awal
T(1) = x0;
Y(1,:) = y0;

%jalankan sebanyak n langkah
for i = 1:n
    k1 = h*f(T(i), Y(i,:));
    k2 = h*f(T(i) + 1/4*h, Y(i,:) + 1/4*k1);
    k3 = h*f(T(i) + 1/4*h, Y(i,:) + 1/8*k1 + 1/8*k2);
    k4 = h*f(T(i) + 1/2*h, Y(i,:) - 1/2*k2 + k3);
    k5 = h*f(T(i) + 3/4*h, Y(i,:) + 3/16*k1 + 9/16*k4);
    k6 = h*f(T(i) + h, Y(i,:) - 3/7*k1 + 2/7*k2 + 12/7*k3 - 12/7*k4
+ 8/7*k5);
    %Menentukan Y pada segmen berikutnya
    Y(i+1,:) = Y(i,:) + 1/90 * (7*k1 + 32*k3 + 12*k4 + 32*k5 +
7*k6);

    %Menentukan T pada segmen berikutnya

```



```

        T(i+1) = T(i) + h;
    endfor
end

```

# ○ Program Fungsi

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Fungsi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fxy = f2(x,y)

%Lambda kecil
li = [0.0124 0.0305 0.111 0.301 1.1 3.0];
%Fraksi Neutron Kasip
Bi = [0.000215 0.001420 0.001270 0.002570 0.000750 0.000270];
Bli = Bi ./ li;

%Parameter Dinamis
B = sum(Bi);
leff1 = (1/B * sum(Bi ./ li)) ^ -1;
leff2 = 1/B * sum(Bi .* li);

%kondisi 2
if (x < 10)
    rho = 1.1 * B;
elseif (x >= 20)
    rho = -1.1 * B;
else
    rho = 0;
endif

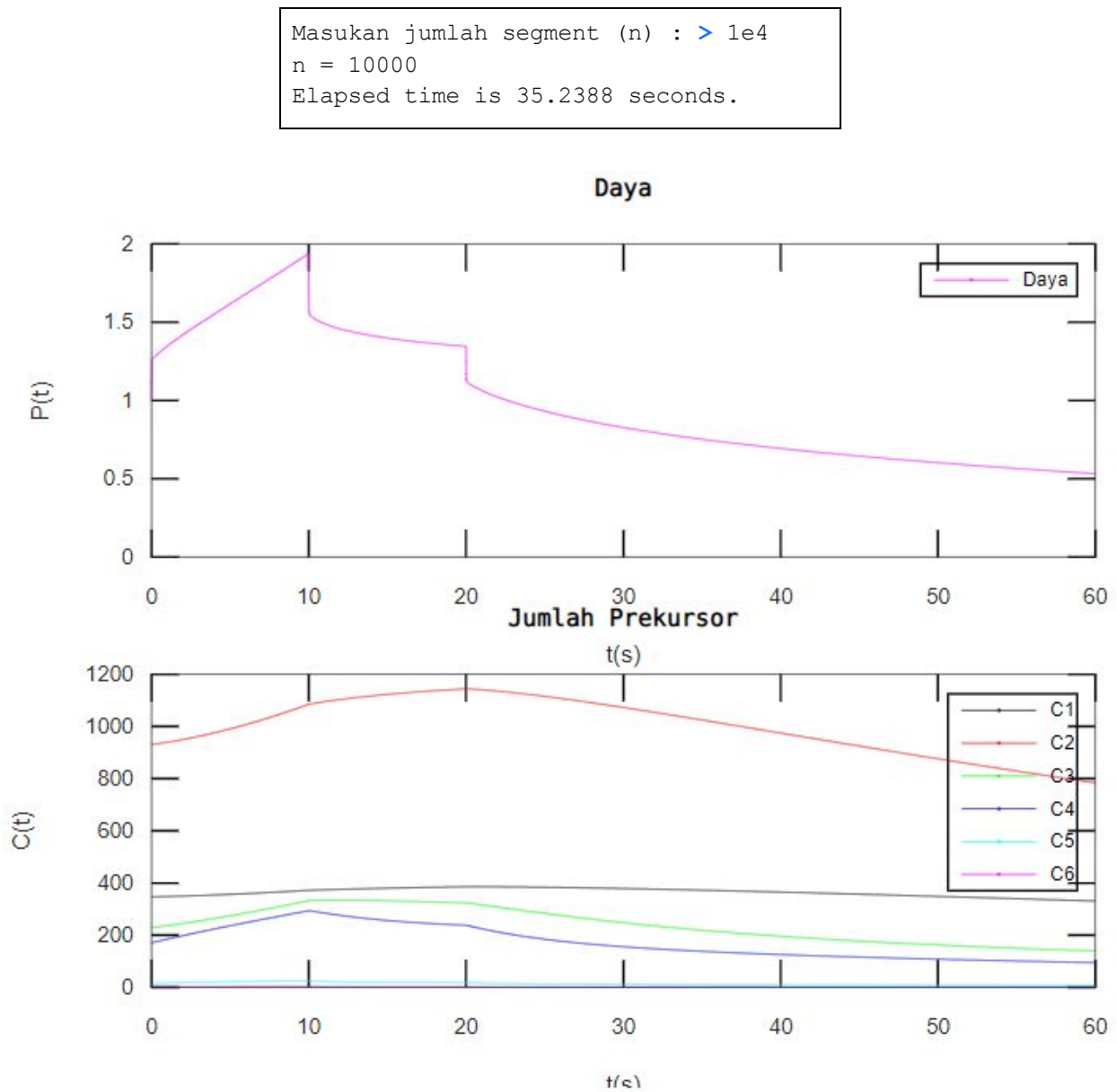
%Lamda Besar
lamb = 5 * 10^-5;

%Perhitungan daya
fxy(1) = (rho-B)/lamb*y(1) + leff1 * y(2); %dengan pendekatan
lambda 1
fxy(1) = (rho-B)/lamb*y(1) + leff2 * y(2); %dengan pendekatan
lambda 2
%Perhitungan jumlah prekursor 1 grup
fxy(2) = (B/lamb)*y(1) - leff1 * y(2); %dengan pendekatan lambda 1
fxy(2) = (B/lamb)*y(1) - leff2 * y(2); %dengan pendekatan lambda 2
end

```

#### D. HASIL DAN PEMBAHASAN

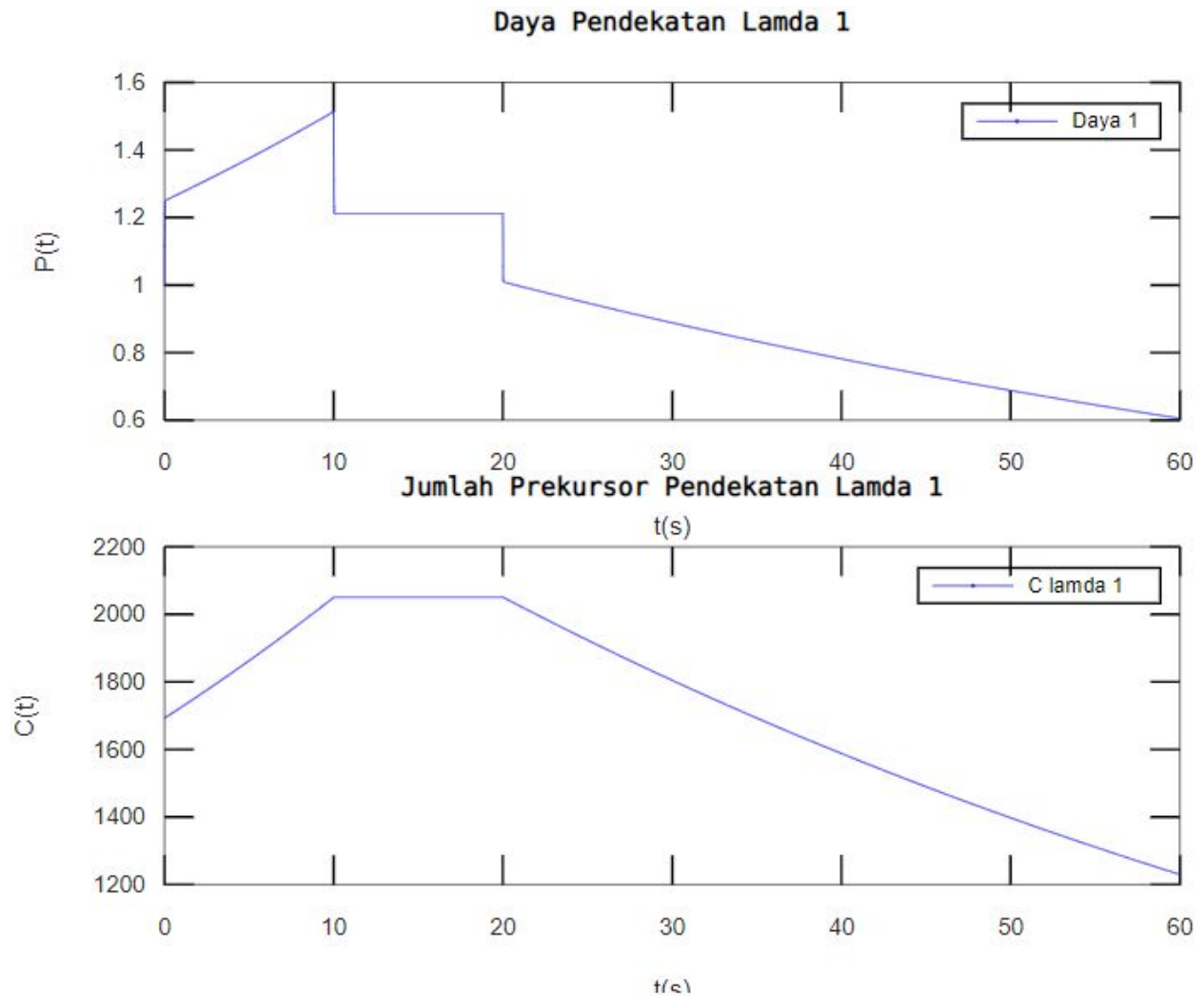
- Hasil
  - Variasi nilai  $\rho$  dari  $0.2\beta$ , ke  $0\beta$ , kemudian ke  $-0.2\beta$ 
    - 6 Grup



Gambar 4.1. Hasil Komputasi dengan 6 Grup Prekursor pada  $\rho_1$

■ 1 Grup  $\lambda_{\text{eff},1}$

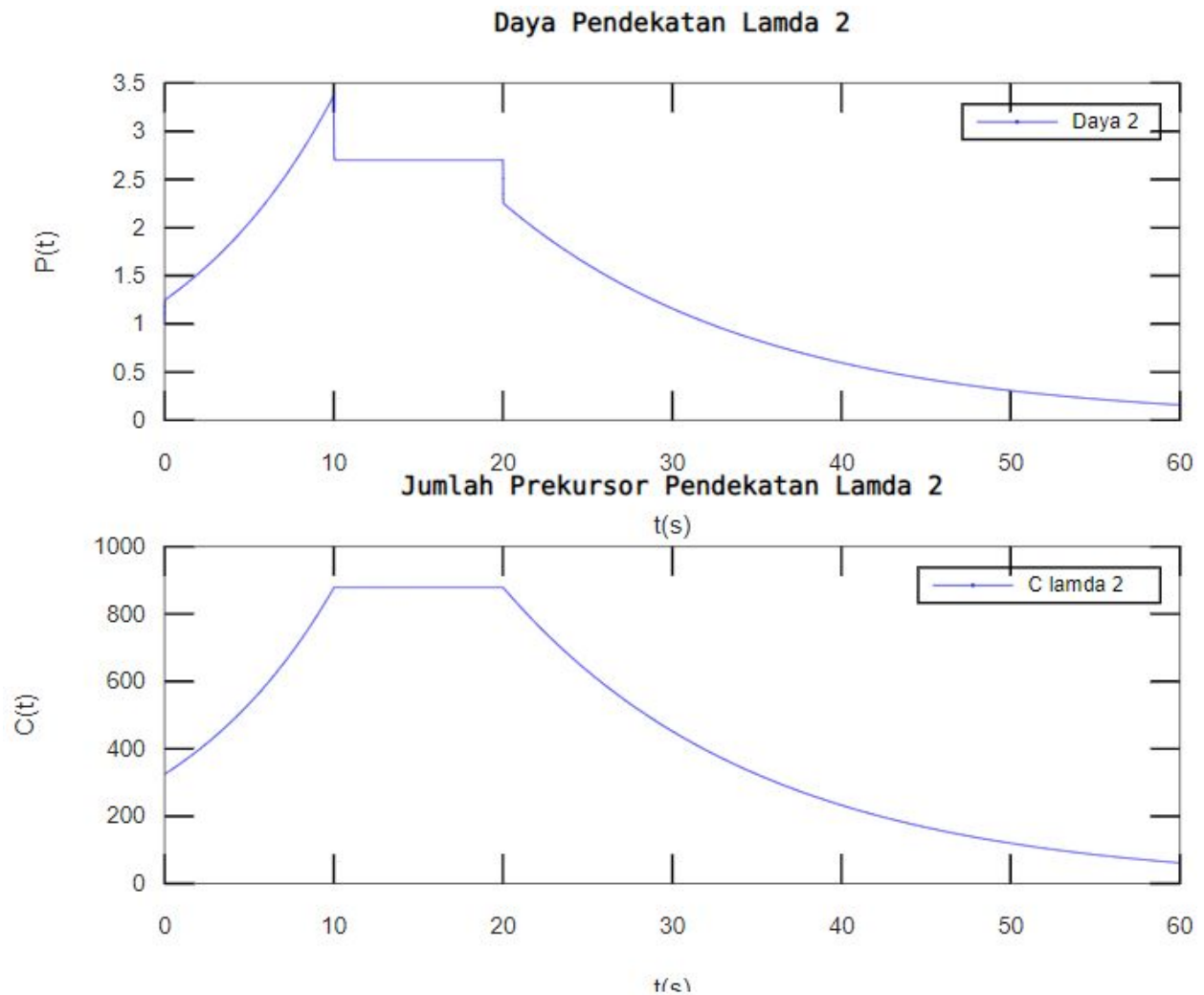
Masukan jumlah segment (n) : > 1e4  
n = 10000  
Elapsed time is 7.58254 seconds.



Gambar 4.2. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},1}$  pada  $\rho_1$

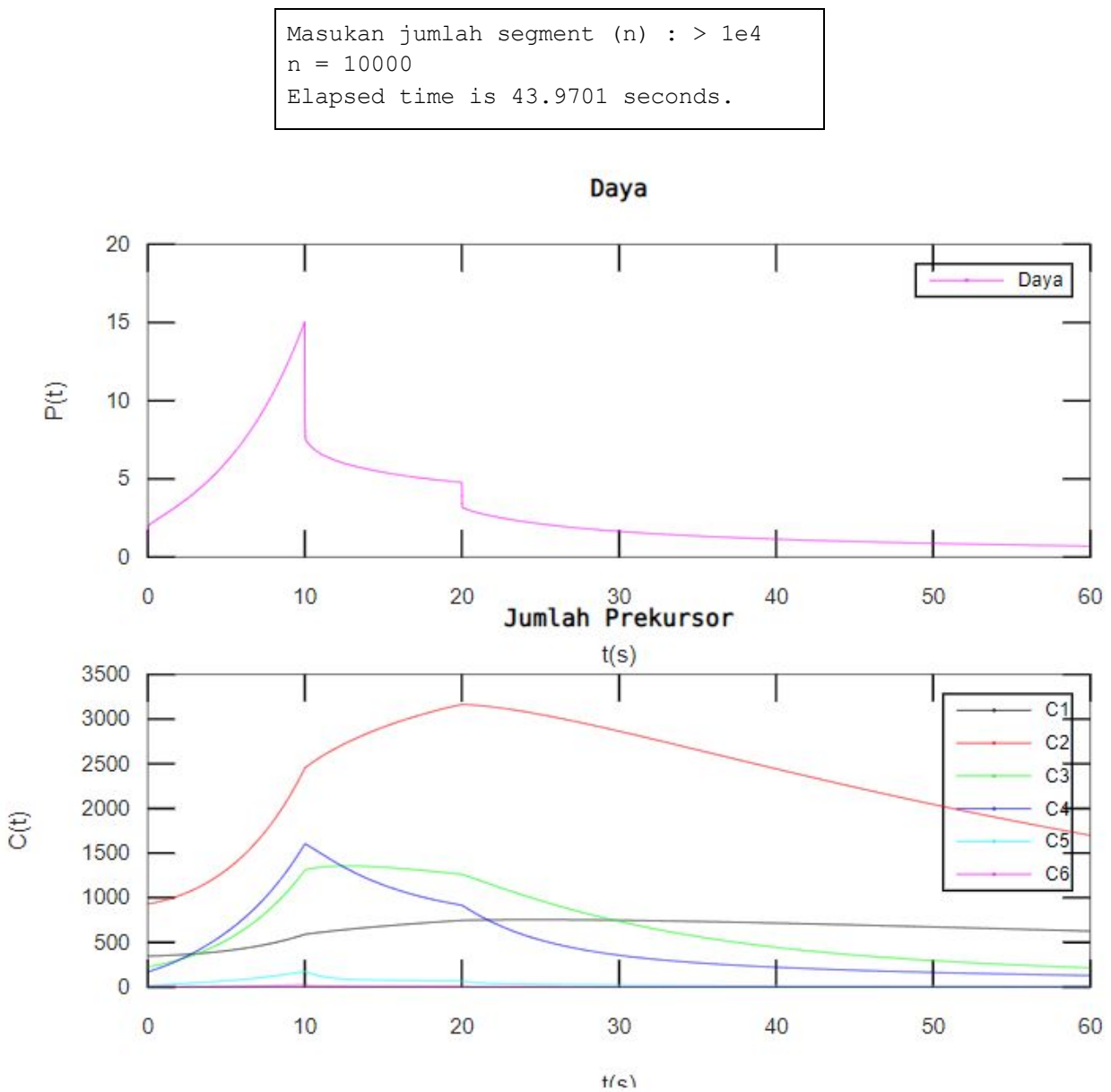
■ 1 Grup  $\lambda_{\text{eff},2}$

Masukan jumlah segment (n) : > 1e4  
n = 10000  
Elapsed time is 7.22884 seconds.



Gambar 4.3. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},2}$  pada  $\rho_1$

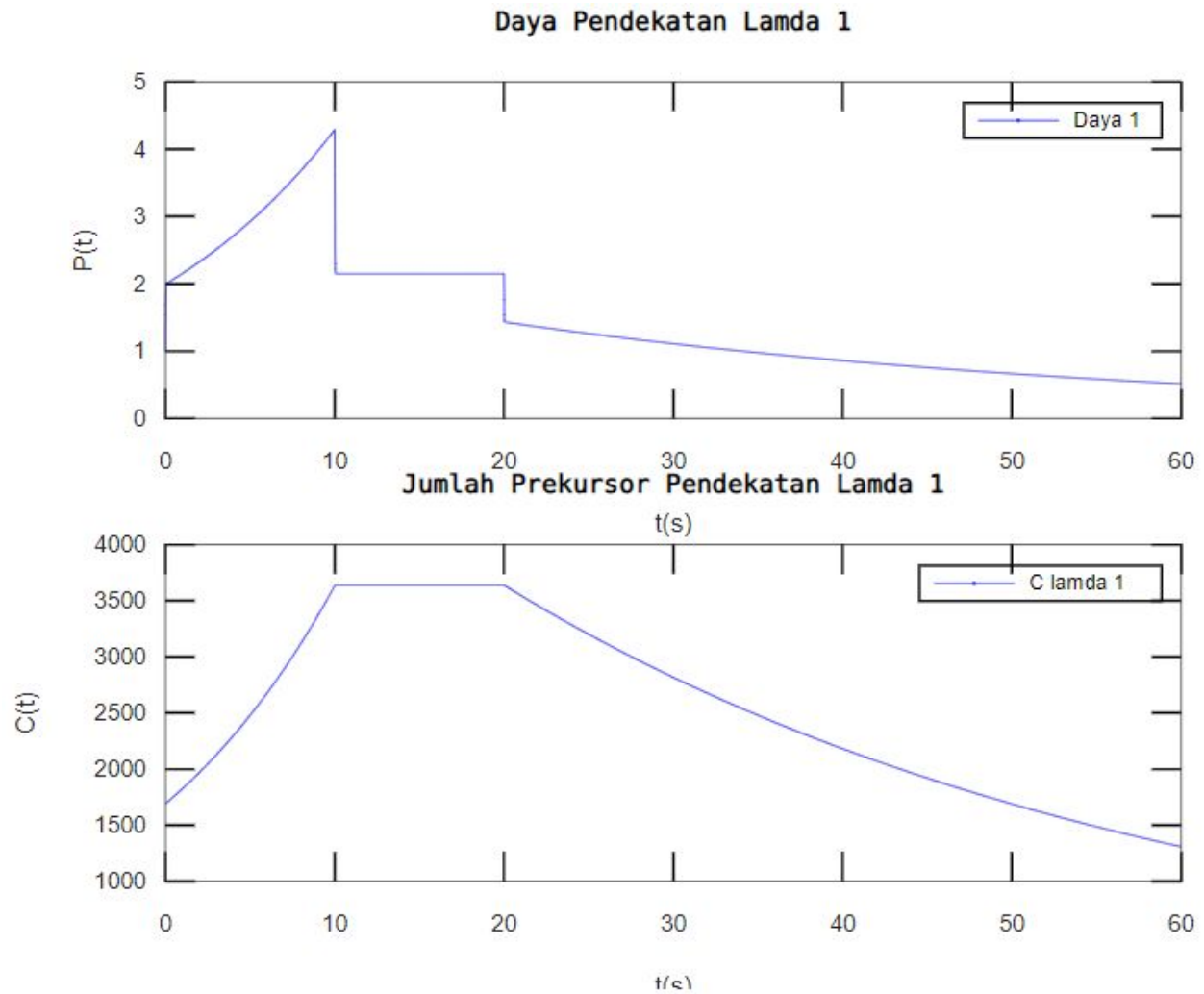
- Variasi nilai  $\rho$  dari  $0.5\beta$ , ke  $0\beta$ , kemudian ke  $-0.5\beta$
- 6 Grup



Gambar 4.4. Hasil Komputasi dengan 6 Grup Prekursor pada  $\rho_2$

■ 1 Grup dengan  $\lambda_{\text{eff},1}$

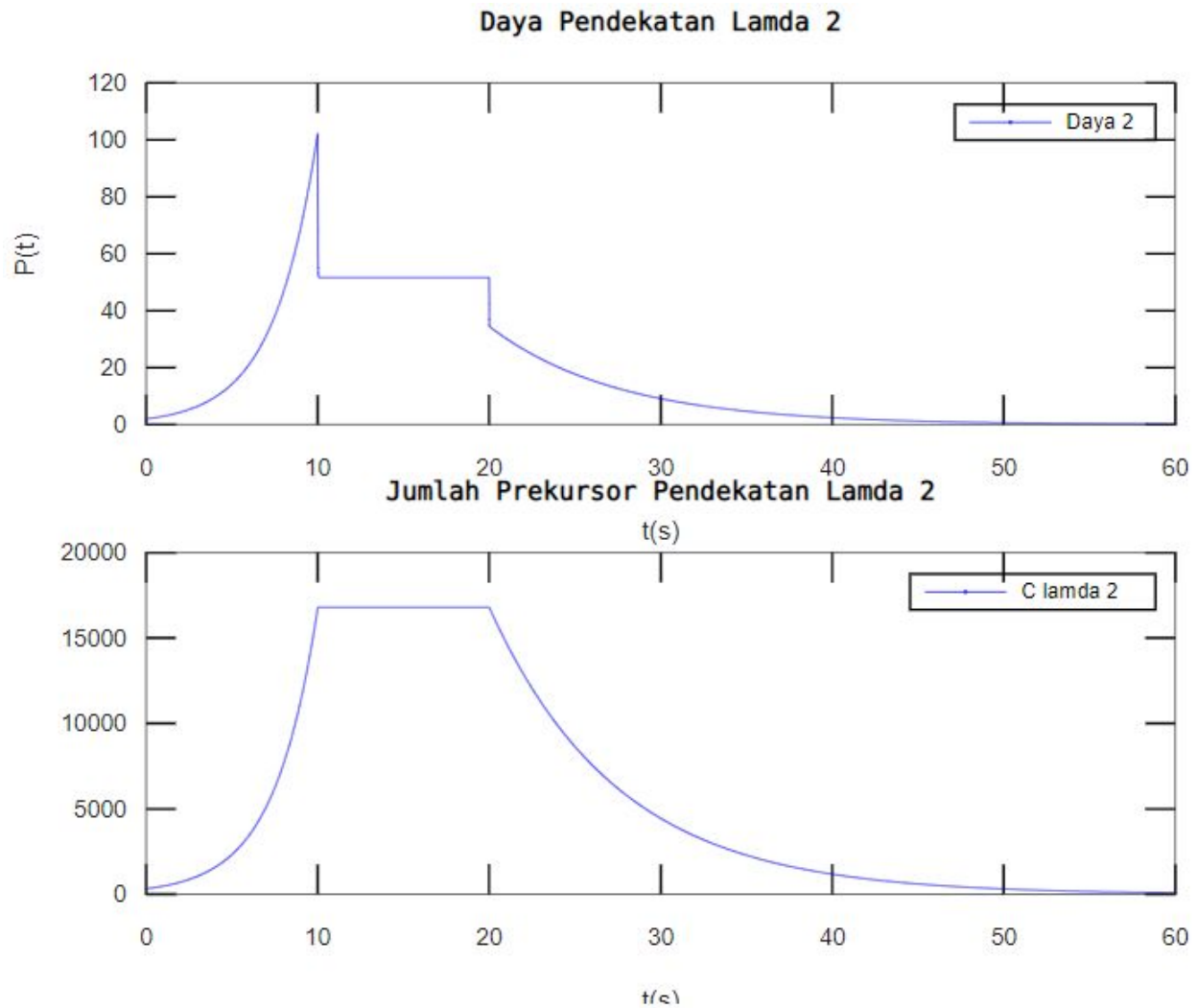
Masukan jumlah segment (n) : > 1e4  
n = 10000  
Elapsed time is 7.01045 seconds.



Gambar 4.5. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},1}$  pada  $\rho_2$

■ 1 Grup dengan  $\lambda_{\text{eff},2}$

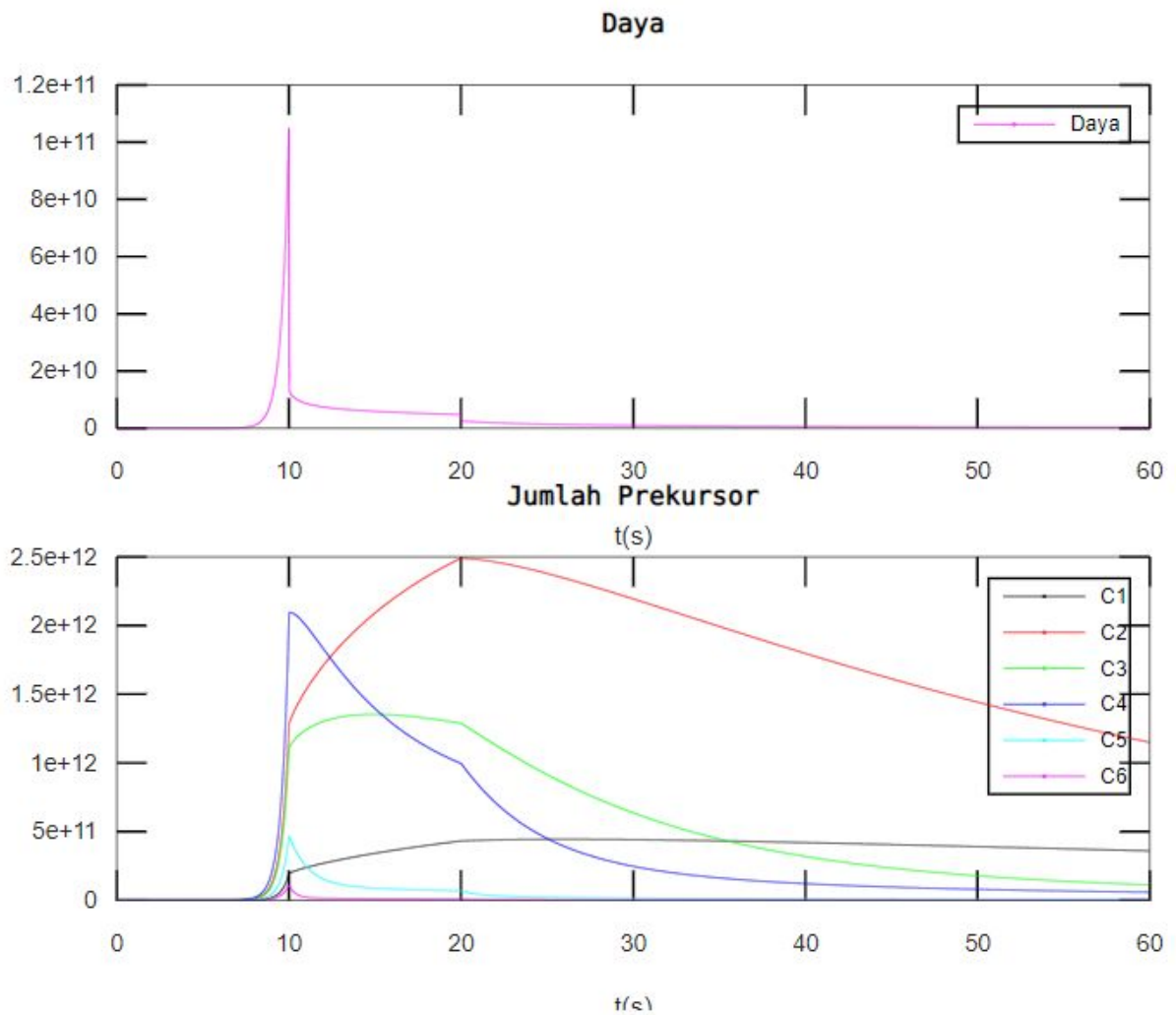
Masukan jumlah segment (n) : > 1e4  
n = 10000  
Elapsed time is 7.18754 seconds.



Gambar 4.6. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},2}$  pada  $\rho_2$

- Variasi nilai  $\rho$  dari  $0.9\beta$ , ke  $0\beta$ , kemudian ke  $-0.9\beta$
- 6 Grup

Masukan jumlah segment (n) : > 1e5  
 n = 100000  
 Elapsed time is 234.804 seconds.

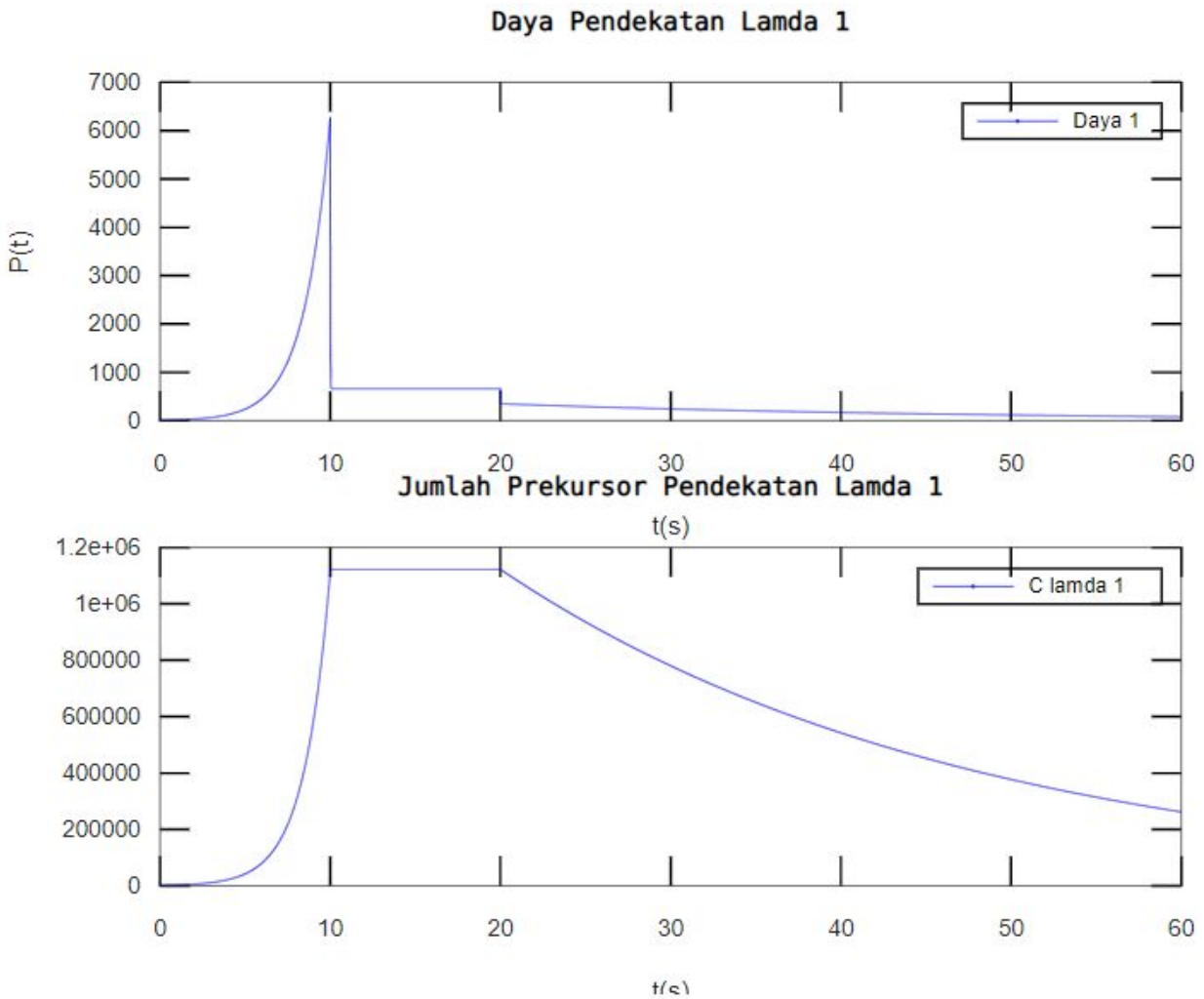


Gambar 4.7. Hasil Komputasi dengan 6 Grup Prekursor pada  $\rho_3$



■ 1 Grup dengan  $\lambda_{\text{eff},1}$

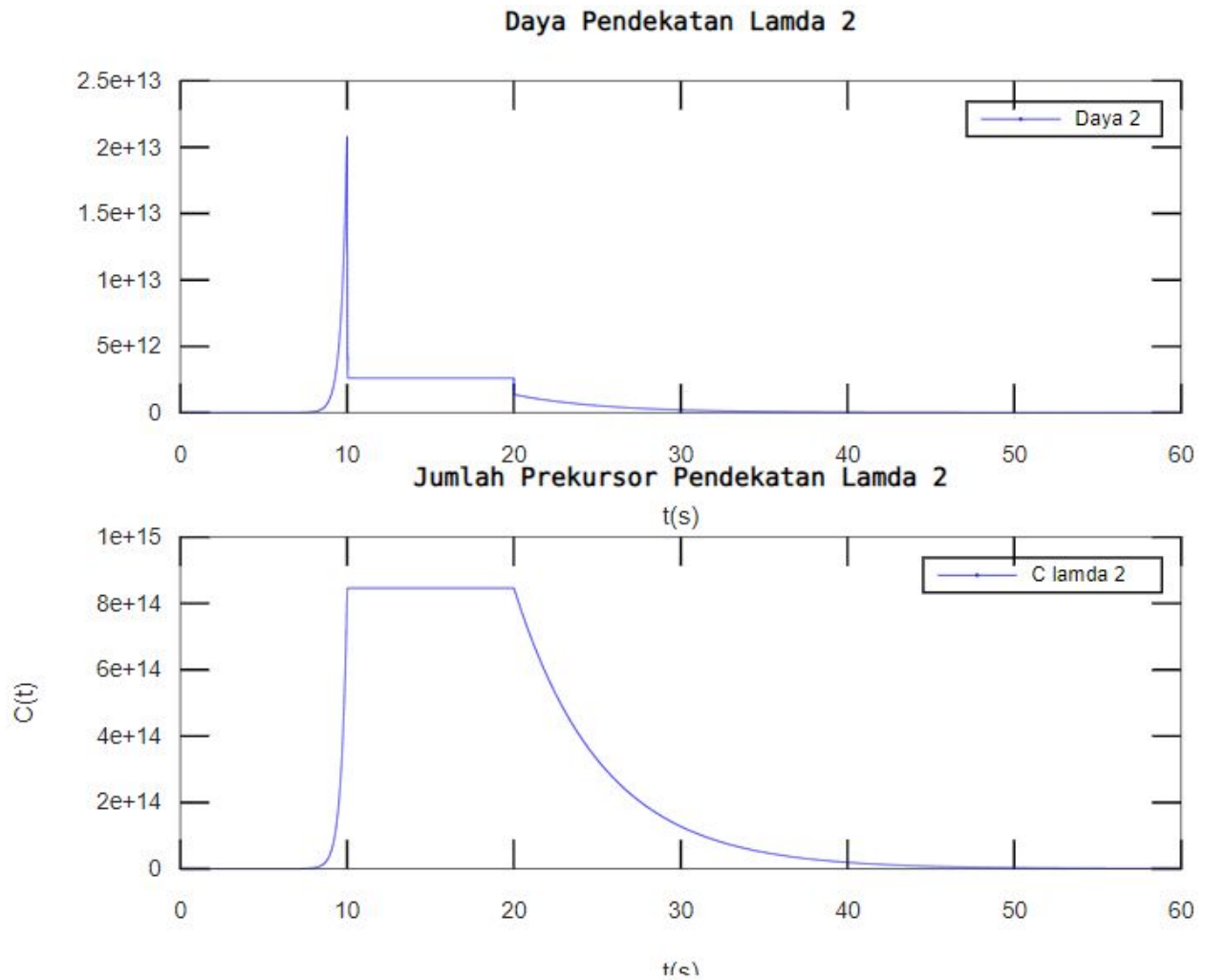
Masukan jumlah segment (n) : > 1e5  
n = 100000  
Elapsed time is 65.0683 seconds.



Gambar 4.8. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},1}$  pada  $\rho_3$

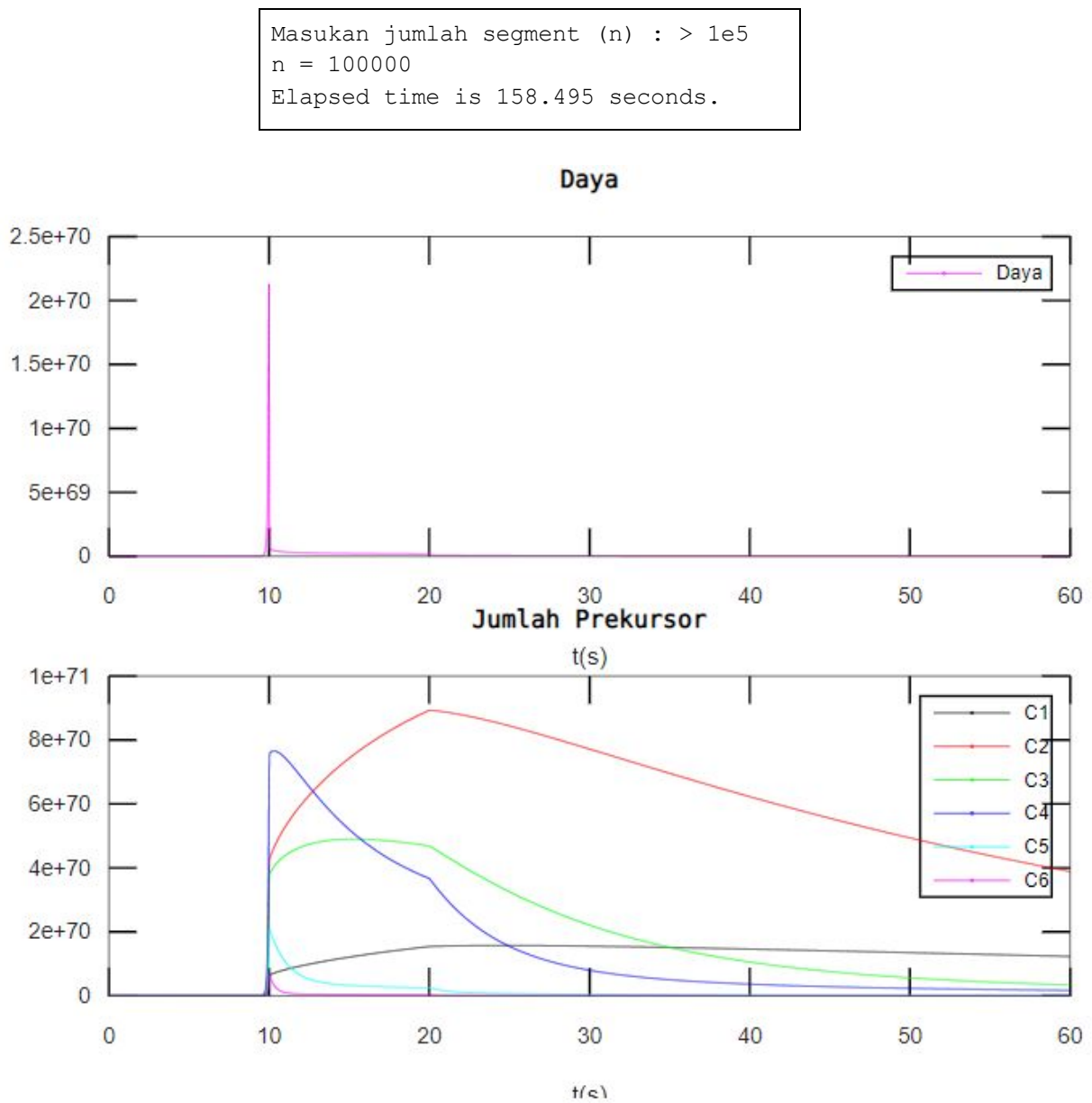
■ 1 Grup dengan  $\lambda_{\text{eff},2}$

Masukan jumlah segment (n) : > 1e5  
n = 100000  
Elapsed time is 67.8637 seconds.



Gambar 4.9. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},2}$  pada  $\rho_3$

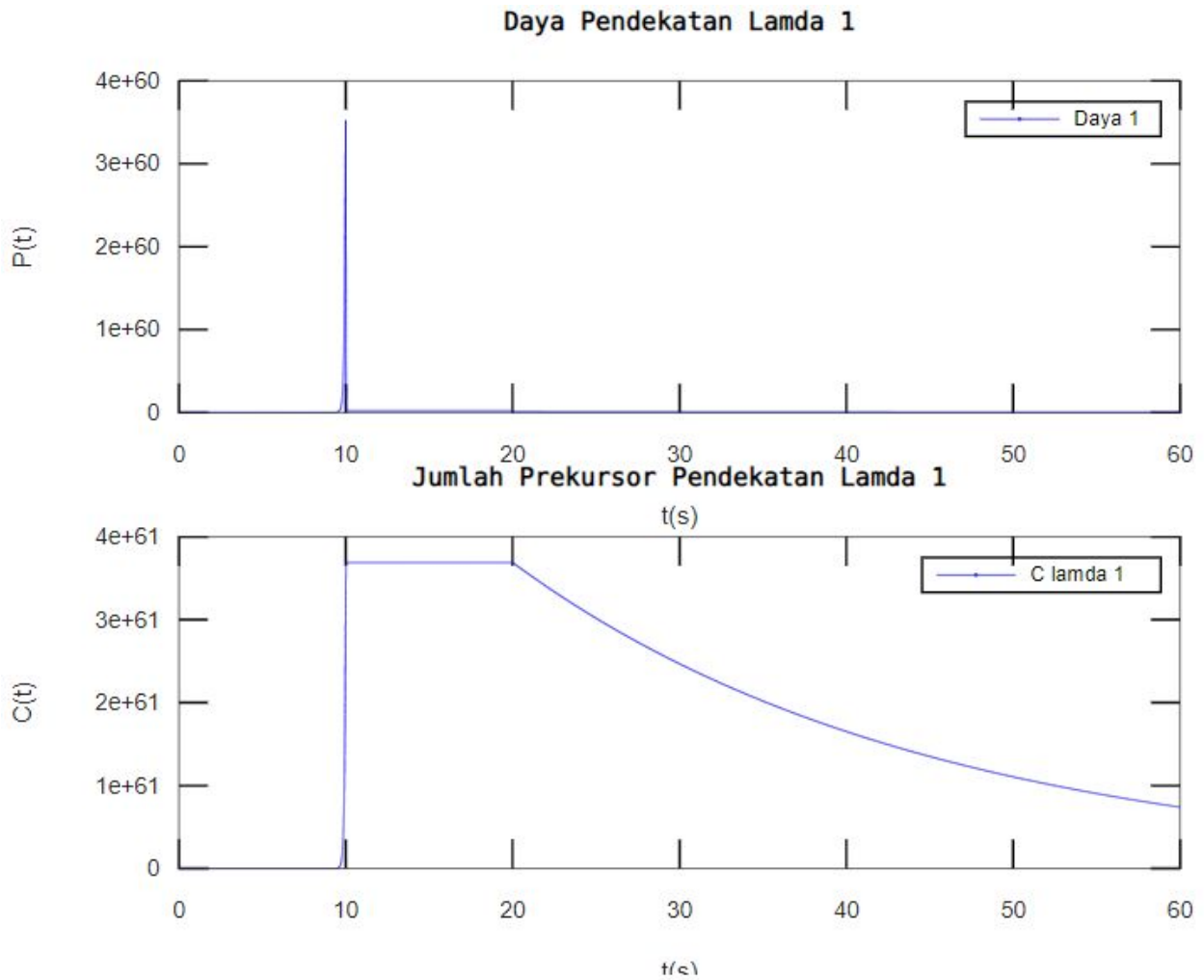
- Variasi nilai  $\rho$  dari  $1.1\beta$ , ke  $0\beta$ , kemudian ke  $-1.1\beta$
- 6 Grup



Gambar 4.10. Hasil Komputasi dengan 6 Grup Prekursor pada  $\rho_4$

■ 1 Grup dengan  $\lambda_{\text{eff},1}$

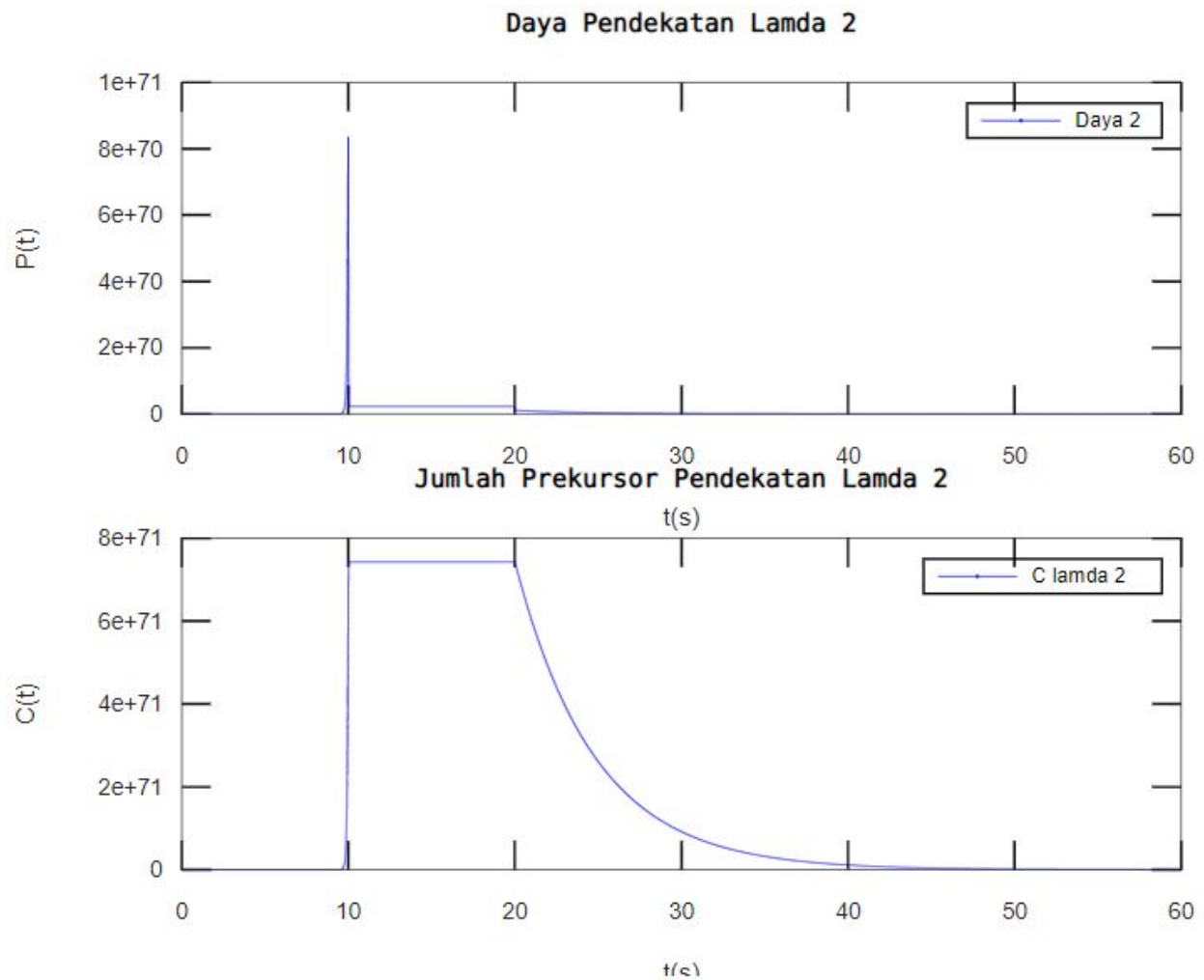
Masukan jumlah segment (n) : > 1e5  
n = 100000  
Elapsed time is 65.6739 seconds.



Gambar 4.11. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},1}$  pada  $\rho_4$

■ 1 Grup dengan  $\lambda_{\text{eff},2}$

Masukan jumlah segment (n) : > 1e5  
 n = 100000  
 Elapsed time is 67.4615 seconds.



Gambar 4.12. Hasil Komputasi dengan 1 Grup Prekursor Menggunakan  $\lambda_{\text{eff},2}$  pada  $\rho_4$

- Pembahasan

Simulasi dilakukan dengan menggunakan pendekatan metode Runge-Kutta orde 6. Hal ini dilakukan untuk memperoleh hasil simulasi yang lebih akurat karena dalam perhitungan dibandingkan menggunakan metode Runge-Kutta orde 4. Persamaan kinetika titik disusun dengan menggunakan vektorisasi variabel-variabel tiap 6 grup dan 1 grup neutron kasip. Kemudian untuk menjalankan program, digunakan nilai *time step* dalam orde yang cukup kecil agar dapat diperoleh hasil yang stabil, serta untuk mengetahui fenomena fisis yang terjadi.

Dengan adanya penambahan reaktivitas positif akan menyebabkan naiknya daya secara eksponensial karena reaksi fisi yang terjadi semakin banyak. Semakin besar penambahan reaktivitas akan menyebabkan nilai daya yang dicapai dalam waktu yang sama, semakin besar. Sedangkan karena adanya penambahan reaktivitas negatif pada reaktor menyebabkan reaksi fisi semakin berkurang, dan daya yang dihasilkan ikut berkurang. Semakin besar penambahan reaktivitas negatif akan menyebabkan penurunan daya yang lebih besar dalam kurun waktu yang sama. Fenomena ini secara signifikan dapat dilihat pada variasi pemberian reaktivitas sebesar  $0.9\beta$  dan  $1.1\beta$  di mana terjadi lonjakan daya yang begitu besar karena penurunan periode reaktor yang disebabkan pemberian reaktivitas undak yang besar.

Perbedaan perhitungan daya dengan menggunakan 6 grup neutron kasip dan 1 grup neutron kasip adalah pada 1 grup diasumsikan bahwa 6 grup neutron kasip dapat digantikan dengan satu kelompok neutron kasip dengan fraksi yield efektif ( $\beta_{\text{eff}}$ ) dan konstanta peluruhan efektif ( $\lambda_{\text{eff}}$ ). Penyusunan untuk persamaan kinetika reaktor satu grup dengan menggunakan nilai  $\lambda_{\text{eff},1}$  dan  $\lambda_{\text{eff},2}$ . Lalu program pendekatan 1 grup digunakan kedua pendekatan lambda efektif agar hasil yang didapatkan dapat dibandingkan dengan perhitungan daya pada 6 grup .

Dapat dilihat pada hasil komputasi, daya yang dihasilkan oleh komputasi dengan 6 grup akan lebih lebih besar dibandingkan dengan 1 grup yang menggunakan  $\lambda_{\text{eff},1}$ , dan akan lebih kecil dibandingkan dengan 1 grup yang menggunakan  $\lambda_{\text{eff},2}$ . Hal ini disebabkan karena penggunaan nilai lambda efektif dapat digunakan pada dua kondisi. Pertama, untuk kondisi transien yang lambat ( $1/T \ll \lambda_i$ ) maka akan digunakan nilai  $\lambda_{\text{eff},1}$  pada pendekatan 1 grup. Kedua, untuk kondisi transien yang cepat ( $1/T \gg \lambda_i$ ) maka akan digunakan  $\lambda_{\text{eff},2}$ .

Berdasarkan tinjauan grafik prekursor pada 6 grup, jumlah prekursor dipengaruhi oleh nilai reaktivitas yang diberikan. Pada rentang waktu dari 0 hingga 10 detik, seluruh prekursor jumlahnya naik secara signifikan karena adanya penambahan reaktivitas positif yang diberikan. Dengan adanya penambahan reaktivitas positif menyebabkan jumlah reaksi fisi meningkat sehingga jumlah prekursor ikut meningkat. Pada rentang waktu dari 10 hingga 20 detik dimana dilakukan pemberian reaktivitas bernilai nol, prekursor C4,

Solusi dengan model 6 grup adalah eksak. Model 1 grup yang satu akan overpredict, yang satunya lagi akan underpredict

Tambah penjelasan, apakah penggunaan model 1 grup dianjurkan atau tidak jika dibandingkan dengan model 6 grup? Kaitkan juga dengan waktu komputasi.

C5, dan C6 jumlahnya menurun, hal ini disebabkan karena prekursor-prekursor tersebut memiliki waktu paruh yang jauh lebih pendek dibandingkan dengan waktu pembentukannya. Sedangkan untuk prekursor C1, C2, dan C3, jumlahnya cenderung meningkat karena waktu paruhnya lebih besar dibandingkan dengan waktu pembentukannya. Pada rentang waktu dari 20 detik hingga 60 detik, seluruh jumlah prekursor mengalami penurunan karena penambahan reaktivitas negatif. Dengan adanya penambahan reaktivitas negatif di reaktor maka jumlah reaksi fisi berkurang, sehingga jumlah prekursor yang dihasilkan ikut berkurang. Semakin besar penambahan reaktivitas positif/negatif, semakin besar kenaikan/penurunan jumlah prekursor.

Hasil dari grafik perhitungan 1 grup prekursor juga mengalami perubahan sesuai dengan penambahan reaktivitas positif/negatif yang berbanding lurus dengan kenaikan/penurunan jumlah prekursor. Akan tetapi ketika reaktivitas bernilai nol pada detik ke 10 sampai 20, maka daya akan turun hingga nilai tertentu dan akan bernilai konstan seiring dengan penambahan waktu. Hal ini dikarenakan tidak adanya penambahan prekursor pada saat reaktivitas bernilai nol sehingga pada grafik tidak mengalami perubahan atau konstan.

Kemudian jika ditinjau dari lama komputasi, waktu yang dibutuhkan untuk mencari solusi tiap segmen dan membuat plot pada kinetika reaktor 6 grup prekursor akan lebih lama dibandingkan dengan pendekatan 1 grup prekursor. Hal ini dapat dilihat pada salah satu hasil simulasi, yaitu variasi nilai  $\rho$  dari  $0.2\beta$ , ke  $0\beta$ , kemudian ke  $-0.2\beta$ . Untuk menyelesaikan komputasi 6 grup membutuhkan waktu sekitar 35 detik sedangkan 1 grup hanya membutuhkan waktu 7,5 detik.

*Octave* akan melakukan komputasi secara simultan terhadap sistem persamaan diferensial dimana tiap persamaan diferensial tersebut nilai turunannya akan tergantung pada variabel-variabel persamaan diferensial lain yang masih belum diketahui. Untuk kinetika reaktor 6 grup prekursor, terdapat 7 buah persamaan diferensial sehingga *Octave* akan melakukan perhitungan ke 7 persamaan diferensial tersebut terlebih dahulu sebelum melanjutkan ke segmen berikutnya. Sama halnya dengan pendekatan 1 grup, dimana hanya terdapat dua buah persamaan diferensial. Dari hal tersebut dapat diamati bahwa untuk menjalankan komputasi pada 6 grup dapat memakan waktu yang lebih lama dibandingkan dengan komputasi pada pendekatan 1 grup.

## E. KESIMPULAN DAN SARAN

- Kesimpulan

Berdasarkan hasil yang telah didapatkan, dapat disimpulkan bahwa:

1. Jumlah prekursor berbanding lurus dengan nilai reaktivitas reaktor yang diberikan.
2. Daya yang dihasilkan oleh komputasi dengan 6 grup prekursor akan lebih lebih besar dibandingkan dengan 1 grup prekursor yang menggunakan  $\lambda_{\text{eff},1}$ , dan akan lebih kecil dibandingkan dengan 1 grup prekursor yang menggunakan  $\lambda_{\text{eff},2}$
3. Untuk pendekatan 6 grup, terjadi fenomena sebagai berikut:
  - Pada rentang waktu dari 0 hingga 10 detik seluruh prekursor jumlahnya naik secara signifikan karena adanya penambahan reaktivitas positif yang diberikan.
  - Pada rentang waktu dari 10 hingga 20 detik dimana dilakukan pemberian reaktivitas bernilai nol, prekursor C4, C5, dan C6 jumlahnya menurun sedangkan prekursor C1, C2, dan C3 jumlahnya cenderung meningkat. Daya reaktor akan turun pada nilai tertentu dan seiring berjalannya waktu akan mengalami penurunan yang tidak signifikan.
  - Pada rentang waktu dari 20 detik hingga 60 detik, seluruh jumlah prekursor mengalami penurunan karena penambahan reaktivitas negatif.
4. Untuk pendekatan 1 grup, ketika reaktivitas bernilai nol pada detik ke 10 sampai 20, maka daya akan turun hingga nilai tertentu dan akan bernilai konstan seiring dengan penambahan waktu karena tidak adanya penambahan prekursor pada saat reaktivitas bernilai nol sehingga pada grafik tidak mengalami perubahan atau konstan.
5. Waktu yang dibutuhkan untuk mencari solusi tiap segmen dan membuat plot pada kinetika reaktor 6 grup prekursor yaitu sekitar 35 detik, lebih lama dibandingkan dengan pendekatan 1 grup prekursor yang hanya memerlukan 7,5 detik.

- Saran

Sebaiknya dalam membuat plot jumlah prekursor dengan 6 grup digabungkan dengan jumlah prekursor dengan 1 grup pada tiap pendekatan nilai lambda efektif agar dapat dibandingkan secara langsung hasil yang didapatkan. Selain itu sebaiknya menggunakan metode Runge-Kutta adaptif agar plot grafik yang dihasilkan lebih baik karena akan dijumpai kondisi yang memiliki perubahan nilai yang drastis.

Dapat dicoba  
sebagai tugas  
ekstra



## F. DAFTAR ACUAN

- [1] M. Ismail, D. Abulaban, F. Genco, dan M. Alkhedher, "Solution of the Reactor Point Neutron Kinetic Equations with Temperature Feedback Control Using MATLAB-Simulink Toolbox," pada ICTEA 10th International Conference on Thermal Engineering: Theory and Applications, Muscat, 2017.

### Saran terkait coding:

- Data-data fisis seperti nilai  $\lambda$ ,  $\Lambda$ ,  $\beta$  didefinisikan di program utama sebagai variabel global.
- Nilai awal berupa vektor  $y_0$  dinyatakan dengan nama variabel, tidak langsung menggunakan angka, sehingga lebih mudah dilakukan perubahan. Misal nilai  $P_0$  berubah, maka nilai awal prekursor akan secara otomatis dihitung oleh Octave (tidak dihitung manual dulu kemudian dimasukkan secara manual ke dalam nilai  $y_0$ ).
- Nilai  $p(t)$  dan waktu perubahannya lebih baik jika didefinisikan di program utama, tidak di bagian function persamaan diferensial. Ini akan membuat program menjadi lebih fleksibel.