PROJECT I

SK5004
PENGANTAR SAINS KOMPUTASI

Mohammad Rizka Fadhli
20921004

23 October 2022

Contents

List of Tables

List of Figures

1 Pendahuluan

Sains komputasi adalah disiplin ilmu yang mempelajari penyelesaian berbagai masalah dalam sains melalui pendekatan komputasi. Salah satunya adalah mencari solusi dari persamaan diferensial yang merupakan fungsi kontinu menggunakan pendekatan yang bersifat diskrit. Sebagai pembahasan pada laporan ini, diberikan satu masalah persamaan diferensial berupa *unconstrained growth and decay* dari suatu permasalahan peluruhan dan pembentukan beberapa zat radioaktif lalu kemudian akan diselesaikan menggunakan metode pendekatan diskrit.

1.1 Masalah

Laju peluruhan suatu zat radioaktif bisa dituliskan dalam persamaan diferensial berikut ini:

$$\frac{dQ}{dt} = -rQ(t)$$

Untuk suatu r bernilai positif (*decay rate*) dan $Q(t)$ adalah fungsi massa zat radioaktif terhadap waktu (t). Suatu zat radioaktif bisa luruh membentuk zat radioaktif lainnya membuat rantai reaksi.

Buatlah model rantai reaksi radioaktif yang berisi 3 elemen: dari zat A luruh menjadi zat B dan luruh menjadi zat C!

1.2 Bahasa Pemrograman

Saya menggunakan bahasa pemrograman **R** versi `4.0.4` untuk membuat program dan melakukan simulasi untuk menyelesaikan permasalahan di atas.

2 Dasar Teori

2.1 *Unconstrained Growth and Decay*

Model *unconstrained growth and decay* pada dasarnya merupakan model pertumbuhan atau peluruhan yang laju perubahannya proporsional dengan populasi (kondisi) saat ini. Populasi akan bertumbuh atau berkurang tanpa ada batasan atau hal yang menghalangi perubahannya. Pada masalah yang dihadapi, suatu zat radioaktif akan meluruh mengikuti persamaan diferensial berikut:

$$\frac{dQ}{dt} = -rQ(t)$$

dan akan bertumbuh juga mengikuti persamaan diferensial berikut:

$$\frac{dQ}{dt} = rQ(t)$$

Perbedaannya adalah pada nilai r yang kelak akan digunakan untuk masing-masing zat radioaktif A, B, dan C.

Solusi analitik dari peluruhan adalah: $P = P_0 e^{-rt}$

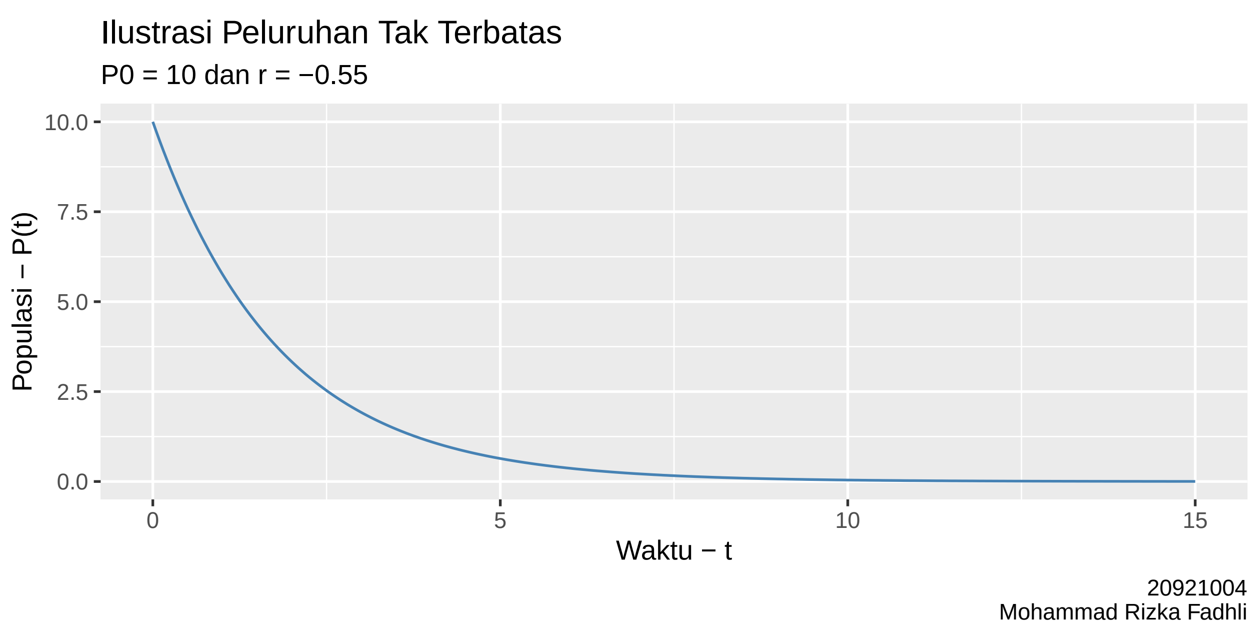

Figure 1: Ilustrasi Peluruhan

sedangkan solusi analitik dari pertumbuhan adalah: $P = P_0 e^{rt}$

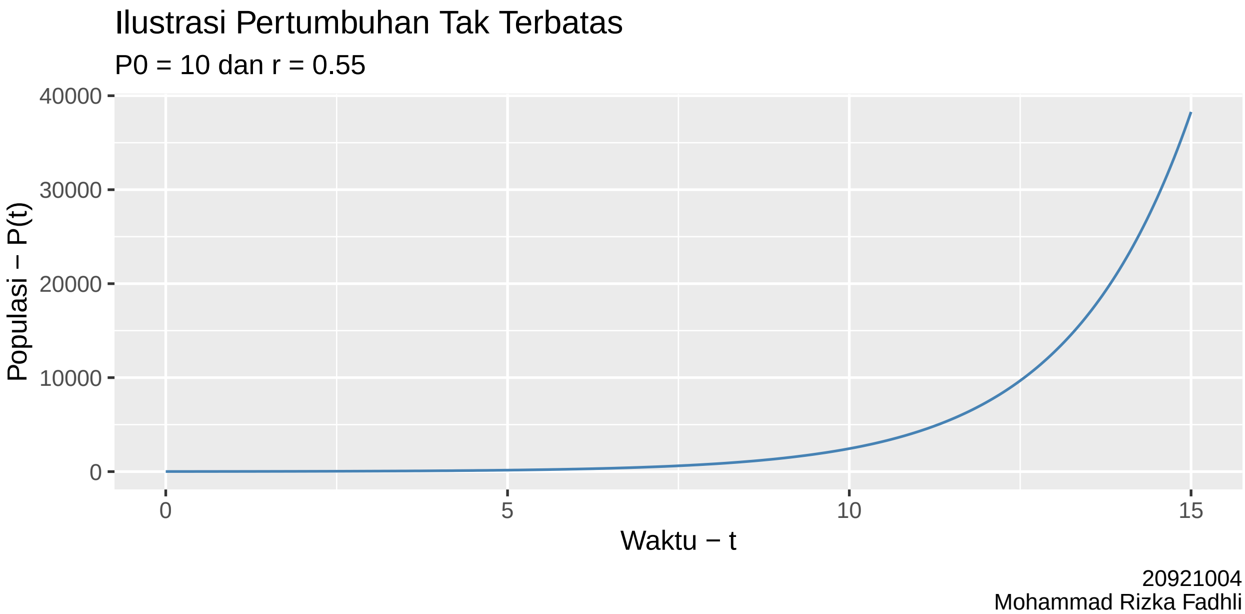

Figure 2: Ilustrasi Pertumbuhan

2.2 *Finite Difference Equation*

Pendekatan diskrit untuk menyelesaikan persamaan diferensial menggunakan komputer adalah dengan membuat persamaan kontinu di atas menjadi bentuk persamaan beda. Komputer tidak bisa menyelesaikan masalah pada t kontinu, oleh karena itu dibutuhkan pendekatan diskrit berupa laju perubahan pada Δt yang relatif kecil. Bentuk umum persamaan beda adalah sebagai berikut:

$$\text{new value} = \text{old values} + \text{change in value}$$

Berdasarkan bentuk di atas, saya akan membuat dua algoritma untuk menyelesaikannya, yakni:

2.3 Algoritma Euler

Algoritma Euler untuk menyelesaikan persamaan diferensial adalah sebagai berikut:

```
initialize
  sim_length
  population
```

```
  rate
  dt
compute:
  rate_per_step = rate * dt
  num_iter = sim_length / dt
for i: 1 to num_iter do
  population = population + rate_per_step * population
  t = i * dt
  print(t, population)
```

2.4 Algoritma Runge Kutta 4^{th}

Bentuk umum dari metode Runge Kutta orde 4 adalah sebagai berikut:

$$y_{n+1} = y_n + h \sum_{i=1}^{n} b_i k_i$$

dimana:

1. $y(t = 0)$ diketahui.
2. k_i adalah konstanta yang harus dicari, yakni:

$$\begin{gathered} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \tfrac{h}{2}, y_n + h\tfrac{k_1}{2}) \\ k_3 = f(t_n + \tfrac{h}{2}, y_n + h\tfrac{k_2}{2}) \\ k_4 = f(t_n + h, y_n + hk_3) \end{gathered}$$

Berikut adalah algoritmanya:

```
initialize
  f # fungsi persamaan diferensial
  x0
  y0
  h
  sim_length

compute:
  num_iter = sim_length / h

for i: 1 to num_iter do
  compute:
    k1 = f(x0,y0)
    k2 = f(x0 + 0.5*h,y0 + 0.5*k1*h)
    k3 = f(x0 + 0.5*h,y0 + 0.5*k2*h)
    k4 = f(x0 + h,y0 + k3*h)
    y0 = y0 + (1/6)*(k1 + 2*k2 + 2*k3 + k4) * h
    x0 = x0 + h
    print(x0,y0)
```

3 Model Matematika

Untuk membuat model matematika dari permasalahan ini, saya akan membuat diagram dari model sehingga hubungan antara ketiga zat radioaktif tersebut bisa terlihat dengan jelas.

3.1 Diagram Model

Berikut ini adalah diagram model:

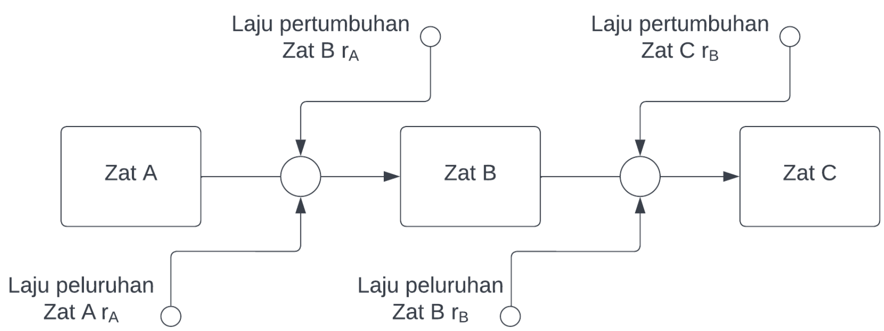

Figure 3: Diagram Model

Zat A akan luruh dengan laju sebesar r_A membentuk zat B dengan laju pertumbuhan sebesar r_A juga. Kemudian zat B akan luruh dengan laju sebesar r_B membentuk zat C dengan laju pertumbuhan sebesar r_B juga.

3.2 Persamaan Diferensial

Berdasarkan diagram dan keterangan di atas, kita dapatkan bahwa:

1. Perubahan massa zat A hanya bergantung pada peluruhan saja.
2. Perubahan massa zat B bergantung pada pembentukan dan peluruhan.
3. Perubahan massa zat C bergantung pada pembentukan saja.

Dari sini, maka didapatkan sistem persamaan diferensial sebagai berikut:

Model perubahan massa zat A:

$$\frac{dA}{dt} = -r_A A$$

Model perubahan massa zat B:

$$\frac{dB}{dt} = r_A A - r_B B$$

Model perubahan massa zat C:

$$\frac{dC}{dt} = r_B B$$

3.3 Algoritma Penyelesaian

Berikut adalah algoritma penyelesaian dengan cara mengubah persamaan diferensial menjadi persamaan beda hingga:

```
# definisi dan initial condition
r_a   # rate peluruhan A dan pertumbuhan B
r_b   # rate peluruhan B dan pertumbuhan C

q_a   # massa awal zat radioaktif A
q_b   # massa awal zat radioaktif B
q_c   # massa awal zat radioaktif C
t     # waktu awal t = 0

dt           # delta t
iter_length  # panjang iterasi
num_iter = iter_length / dt  # berapa banyak iterasi

# proses iterasi
for i in 1 to num_iter do
    # peluruhan A dan pertumbuhan B
    rate_1 = r_a * q_a[i-1] * dt

    # peluruhan B dan pertumbuhan C
    rate_2 = r_b * q_b[i-1] * dt

    # perhitungan massa zat A
    q_a[i] = q_a[i-1] - rate_1
```

```
    # perhitungan massa zat B
    q_b[i] = q_b[i-1] + rate_1 - rate_2
    # perhitungan massa zat C
    q_c[i] = q_c[i-1] + rate_2
    # perhitungan massa waktu
    t[i] = t[i-1] + dt

print(t,q_a,q_b,q_c)
```

Algoritma di atas merupakan penerapan metode Euler untuk menyelesaikan sistem persamaan diferensial.

3.4 Program Model Matematika

Pada bagian ini, saya akan membuat dua program untuk menyelesaikan permasalahan ini, yakni:

3.4.1 Program dengan Metode Euler

```
# ==============================================================================
# INPUT dari user:
  # rate peluruhan A dan pertumbuhan B
  r_a = readline(prompt = "Rate peluruhan A: ") %>% as.numeric()
  # rate peluruhan B dan pertumbuhan C
  r_b = readline(prompt = "Rate peluruhan B: ") %>% as.numeric()

  # massa awal zat radioaktif A
  qa0 = readline(prompt = "massa awal zat radioaktif A: ") %>% as.numeric()
  # massa awal zat radioaktif B
  qb0 = readline(prompt = "massa awal zat radioaktif B: ") %>% as.numeric()
  # massa awal zat radioaktif C
  qc0 = readline(prompt = "massa awal zat radioaktif C: ") %>% as.numeric()
  # delta t
  dt0 = readline(prompt = "nilai delta t: ") %>% as.numeric()
  # panjang iterasi
  iter_length = readline(prompt = "seberapa panjang iterasi dilakukan: ") %>%
                as.numeric()
```

```
# proses perhitungan dengan metode Euler
q_a = c(qa0)  # array massa zat radioaktif A
q_b = c(qb0)  # array massa zat radioaktif B
q_c = c(qc0)  # array massa zat radioaktif C
t = c(0)      # waktu awal t = 0
dt = dt0
num_iter = (iter_length / dt) + 1

# proses iterasi
for(i in 2:num_iter){
    # peluruhan A dan pertumbuhan B
    rate_1 = r_a * q_a[i-1] * dt
    # peluruhan B dan pertumbuhan C
    rate_2 = r_b * q_b[i-1] * dt

    # perhitungan massa zat A
    q_a[i] = q_a[i-1] - rate_1
    # perhitungan massa zat B
    q_b[i] = q_b[i-1] + rate_1 - rate_2
    # perhitungan massa zat C
    q_c[i] = q_c[i-1] + rate_2
    # perhitungan massa waktu
    t[i] = t[i-1] + dt
}

# membuat data output
df = data.frame(t,q_a,q_b,q_c)

# print output pada layar
print(df)
# ==============================================================================
```

Berikut adalah *screenshoot* saat program dijalankan dengan parameter sebagai berikut:

Table 1: Parameter Simulasi Program Euler

Parameter	Value
r__a	0.50
r__b	0.40
qa0	10.00
qb0	0.00
qc0	0.00
dt0	0.25
iter__length	5.00

```
> source("Program Euler.R")
Rate peluruhan A: .5
Rate peluruhan B: .4
massa awal zat radioaktif A: 10
massa awal zat radioaktif B: 0
massa awal zat radioaktif C: 0
nilai delta t: .25
seberapa panjang iterasi dilakukan: 5
      t        q_a      q_b       q_c
1  0.00 10.0000000 0.000000 0.0000000
2  0.25  8.7500000 1.250000 0.0000000
3  0.50  7.6562500 2.218750 0.1250000
4  0.75  6.6992188 2.953906 0.3468750
5  1.00  5.8618164 3.495918 0.6422656
6  1.25  5.1290894 3.879053 0.9918574
7  1.50  4.4879532 4.132284 1.3797627
8  1.75  3.9269590 4.280050 1.7929912
9  2.00  3.4360892 4.342915 2.2209961
10 2.25  3.0065780 4.338134 2.6552876
11 2.50  2.6307558 4.280143 3.0891010
12 2.75  2.3019113 4.180973 3.5171154
13 3.00  2.0141724 4.050615 3.9352127
14 3.25  1.7624008 3.897325 4.3402742
15 3.50  1.5421007 3.727893 4.7300067
16 3.75  1.3493381 3.547866 5.1027959
17 4.00  1.1806709 3.361747 5.4575825
18 4.25  1.0330870 3.173156 5.7937572
19 4.50  0.9039511 2.984976 6.1110728
20 4.75  0.7909572 2.799472 6.4095704
21 5.00  0.6920876 2.618395 6.6895176
```

Figure 4: Hasil Simulasi Program Euler

Program tersebut saya simpan dengan nama **Program Euler.R** dan saya lampirkan bersamaan dengan laporan ini. Untuk menggunakannya, silakan buka **R** dan ketikkan perintah `source("Program Euler.R")`.

3.4.2 Program dengan Metode Runge Kutta 4^{th}

Berikut adalah program yang saya buat menggunakan metode Runge Kutta 4^{th}:

```
# ==============================================================================
# INPUT dari user:
  # rate peluruhan A dan pertumbuhan B
  r_a = readline(prompt = "Rate peluruhan A: ") %>% as.numeric()
  # rate peluruhan B dan pertumbuhan C
  r_b = readline(prompt = "Rate peluruhan B: ") %>% as.numeric()

  # massa awal zat radioaktif A
  q_a = readline(prompt = "massa awal zat radioaktif A: ") %>% as.numeric()
  # massa awal zat radioaktif B
  q_b = readline(prompt = "massa awal zat radioaktif B: ") %>% as.numeric()
  # massa awal zat radioaktif C
  q_c = readline(prompt = "massa awal zat radioaktif C: ") %>% as.numeric()

  # panjang iterasi
  iter_length = readline(prompt = "seberapa panjang iterasi dilakukan: ") %>%
                as.numeric()
  # h
  h = readline(prompt = "selang h: ") %>% as.numeric()

# initial condition
t   = 0   # waktu awal t = 0

# kalkulasi banyak iterasi yang dilakukan
num_iter = (iter_length / h) + 1

# definisi fungsi persamaan diferensial
  # zat radio aktif A
  d_a = function(t,q_a){(-r_a * q_a)}
```

```
  # zat radio aktif B
  d_b = function(t,q_a,q_b){((r_a * q_a) - (r_b * q_b))}

  # zat radio aktif C
  d_c = function(t,q_b,q_c){(r_b * q_b)}

# persiapan array utk iterasi
A = c(q_a)
B = c(q_b)
C = c(q_c)
t_ = c(t)


# proses iterasi
for(i in 2:num_iter){
  # kita akan hitung dulu zat a
  k1 = d_a(t,A[i-1])
  k2 = d_a(t + 0.5*h,A[i-1] + 0.5*k1*h)
  k3 = d_a(t + 0.5*h,A[i-1] + 0.5*k2*h)
  k4 = d_a(t + h,A[i-1] + k3*h)

  A[i] = A[i-1] + (1/6)*(k1 + 2*k2 + 2*k3 + k4) * h

  # kita hitung zat b
  k1 = d_b(t,A[i-1],B[i-1])
  k2 = d_b(t + 0.5*h,A[i-1] + 0.5*k1*h,B[i-1] + 0.5*k1*h)
  k3 = d_b(t + 0.5*h,A[i-1] + 0.5*k2*h,B[i-1] + 0.5*k2*h)
  k4 = d_b(t + h,A[i-1] + k3*h,B[i-1] + k3*h)

  B[i] = B[i-1] + (1/6)*(k1 + 2*k2 + 2*k3 + k4) * h

  # kita hitung zat c
  k1 = d_c(t,B[i-1],C[i-1])
  k2 = d_c(t + 0.5*h,B[i-1] + 0.5*k1*h,C[i-1] + 0.5*k1*h)
  k3 = d_c(t + 0.5*h,B[i-1] + 0.5*k2*h,C[i-1] + 0.5*k2*h)
```

```
  k4 = d_c(t + h,B[i-1] + k3*h,C[i-1] + k3*h)

  C[i] = C[i-1] + (1/6)*(k1 + 2*k2 + 2*k3 + k4) * h

  t_[i] = t_[i-1] + h

}

# output
output = data.frame(t_,A,B,C)

# mengeluarkan output ke layar
print(output)
# ==============================================================================
```

Program tersebut saya simpan dengan nama **Program RK 4.R** dan saya lampirkan bersamaan dengan laporan ini. Untuk menggunakannya, silakan buka **R** dan ketikkan perintah `source("Program RK 4.R")`.

4 Diskusi

Sekarang saya akan *run* program Euler dan Runge Kutta 4^{th} menggunakan parameter yang ada di soal, yakni:

Table 2: Parameter di Soal

Parameter	Value
r_a	0.0137
r_b	0.0510
qa0	0.0000
qb0	0.0000
qc0	0.0000
dt0	0.5000
iter_length	10.0000

Massa zat A adalah $S \times 10^{-8}$ dengan S adalah 2 digit terakhir NIM saya (20921004). Sehingga $S = 4$.

4.1 Menggunakan Program Euler

4.1.1 Tabel Simulasi

Berikut ini adalah hasil simulasi menggunakan program Euler:

```
##       t          q_a          q_b          q_c
## 1   0.0 4.000000e-08 0.000000e+00 0.000000e+00
## 2   0.5 3.972600e-08 2.740000e-10 0.000000e+00
## 3   1.0 3.945388e-08 5.391361e-10 6.987000e-12
## 4   1.5 3.918362e-08 7.956472e-10 2.073497e-11
## 5   2.0 3.891521e-08 1.043766e-09 4.102397e-11
## 6   2.5 3.864864e-08 1.283719e-09 6.764001e-11
## 7   3.0 3.838390e-08 1.515727e-09 1.003748e-10
## 8   3.5 3.812097e-08 1.740006e-09 1.390259e-10
## 9   4.0 3.785984e-08 1.956765e-09 1.833961e-10
## 10  4.5 3.760050e-08 2.166207e-09 2.332935e-10
## 11  5.0 3.734294e-08 2.368532e-09 2.885318e-10
## 12  5.5 3.708714e-08 2.563934e-09 3.489294e-10
```

```
## 13  6.0 3.683309e-08 2.752600e-09 4.143097e-10
## 14  6.5 3.658078e-08 2.934716e-09 4.845010e-10
## 15  7.0 3.633020e-08 3.110459e-09 5.593363e-10
## 16  7.5 3.608134e-08 3.280004e-09 6.386530e-10
## 17  8.0 3.583419e-08 3.443521e-09 7.222931e-10
## 18  8.5 3.558872e-08 3.601175e-09 8.101028e-10
## 19  9.0 3.534494e-08 3.753128e-09 9.019328e-10
## 20  9.5 3.510283e-08 3.899536e-09 9.976376e-10
## 21 10.0 3.486237e-08 4.040552e-09 1.097076e-09
```

4.1.2 Grafik Hasil Simulasi

Berikut ini adalah grafiknya jika saya perpanjang nilai t-nya menjadi 300 dan $dt = 0.05$:

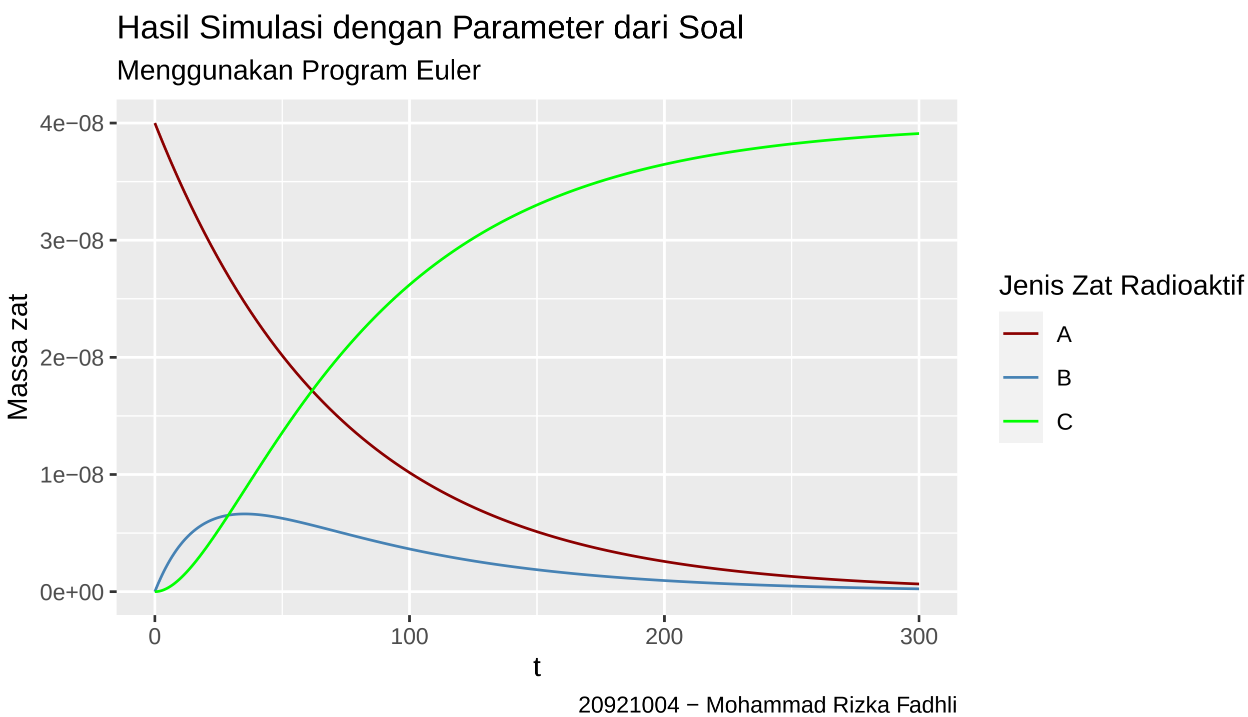

Figure 5: Ilustrasi Hasil Simulasi Euler Menggunakan Parameter Soal

Terlihat bahwa zat A luruh sepenuhnya membentuk zat B yang kemudian luruh sepenuhnya sehingga hanya menyisakan zat C yang terus bertumbuh hingga massanya sebesar $Q_A(t = 0) = 4 \times 10^{-8}$ gram di akhir waktu.

4.2 Menggunakan Program Runge Kutta 4^{th}

4.2.1 Tabel Hasil Simulasi

Berikut ini adalah hasil simulasi menggunakan program Runge Kutta 4^{th}:

```
##      t_            A            B            C
## 1   0.0 4.000000e-08 0.000000e+00 0.000000e+00
## 2   0.5 3.972694e-08 2.714608e-10 0.000000e+00
## 3   1.0 3.945574e-08 5.342103e-10 7.011263e-12
## 4   1.5 3.918639e-08 7.884813e-10 2.080880e-11
## 5   2.0 3.891888e-08 1.034500e-09 4.117362e-11
## 6   2.5 3.865320e-08 1.272489e-09 6.789260e-11
## 7   3.0 3.838933e-08 1.502662e-09 1.007583e-10
## 8   3.5 3.812726e-08 1.725229e-09 1.395689e-10
## 9   4.0 3.786698e-08 1.940395e-09 1.841280e-10
## 10  4.5 3.760848e-08 2.148358e-09 2.342443e-10
## 11  5.0 3.735174e-08 2.349313e-09 2.897319e-10
## 12  5.5 3.709675e-08 2.543449e-09 3.504097e-10
## 13  6.0 3.684351e-08 2.730950e-09 4.161017e-10
## 14  6.5 3.659199e-08 2.911995e-09 4.866364e-10
## 15  7.0 3.634219e-08 3.086760e-09 5.618472e-10
## 16  7.5 3.609410e-08 3.255414e-09 6.415717e-10
## 17  8.0 3.584770e-08 3.418124e-09 7.256523e-10
## 18  8.5 3.560298e-08 3.575050e-09 8.139352e-10
## 19  9.0 3.535994e-08 3.726352e-09 9.062713e-10
## 20  9.5 3.511855e-08 3.872181e-09 1.002515e-09
## 21 10.0 3.487881e-08 4.012688e-09 1.102526e-09
```

4.2.2 Grafik Hasil Simulasi

Berikut ini adalah grafiknya jika saya perpanjang nilai t-nya menjadi 300 dan $h = 0.05$:

Terlihat bahwa grafik yang dihasilkan program Runge Kutta 4^{th} sama. Hasilnya adalah zat A luruh sepenuhnya membentuk zat B yang kemudian luruh sepenuhnya sehingga hanya menyisakan zat C yang terus bertumbuh hingga massanya sebesar $Q_A(t = 0) = 4 \times 10^{-8}$ gram di akhir waktu.

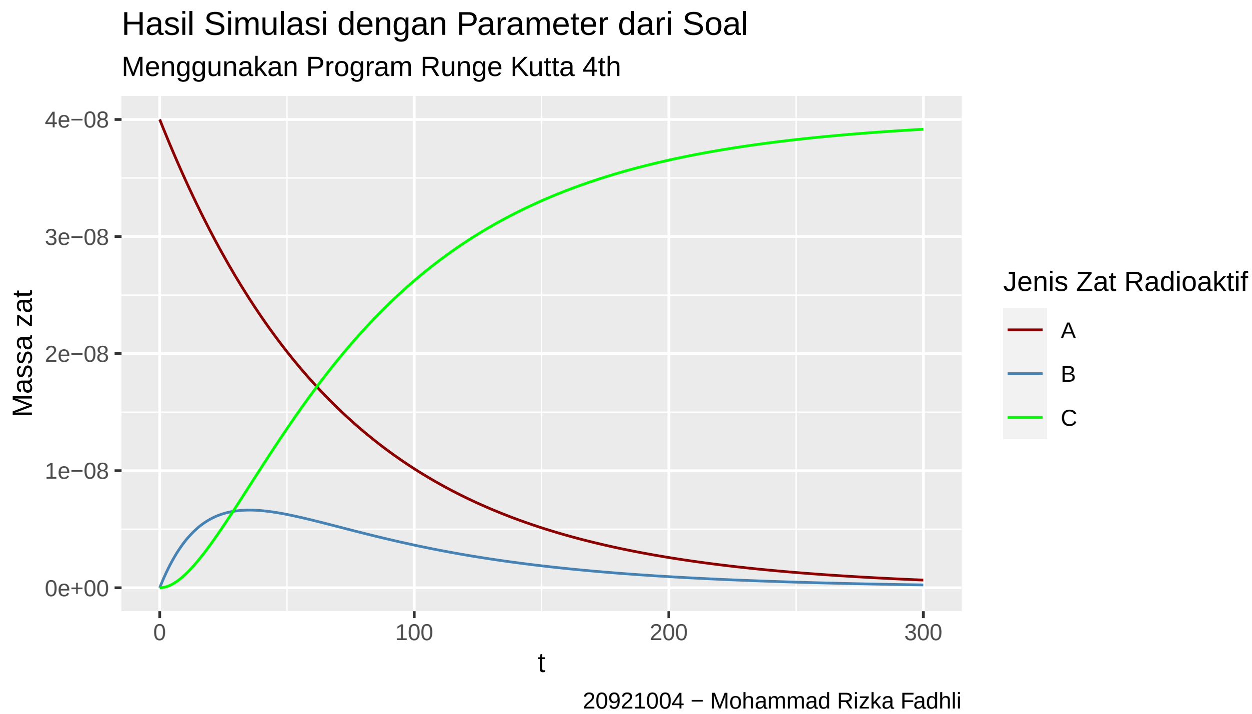

Figure 6: Ilustrasi Hasil Simulasi RK4 Menggunakan Parameter Soal

4.3 Nilai Maksimum Massa Zat B

Jika kita lihat pada grafik sebelumnya, nilai massa zat B akan maksimal di suatu waktu t tertentu. Kita akan coba aproksimasi nilainya.

Secara analitik, nilai maksimum terjadi saat $\frac{dB}{dt} = 0$.

$$\frac{dB}{dt} = r_A A - r_B B = 0$$

$$r_A A = r_B B$$

$$0.0137A = 0.051B$$

$$\frac{A}{B} = \frac{0.051}{0.0137} = 3.722628$$

B akan mencapai maksimum saat rasio $\frac{A}{B}$ di suatu t tertentu bernilai 3.722628. Oleh karena itu, saya akan *run* ulang **Program RK4.R** dengan parameter h dibuat kecil ($h = 0.0001$).

Kemudian saya akan cari t tertentu saat rasio $\frac{A}{B}$ sesuai dengan perhitungan analitik di atas.

Saya dapatkan:

```
##        t_            A            B            C
## 1 35.2394 2.468263e-08 6.630437e-09 8.686955e-09
```

Saya dapatkan massa B maksimum adalah sekitar $B = 6.630437 \times 10^{-9}$ gram pada $t = 35.2394$.

Rasio $\frac{A}{B} = 3.7226249$ hampir sama dengan rasio hasil analitik.

4.4 Perubahan Laju Peluruhan r_A dan r_B

Apa yang akan terjadi jika:

4.4.1 $r_A < r_B$

Kasus ini sama persis dengan apa yang sudah disimulasikan pada bagian sebelumnya. Hal yang akan terjadi adalah:

- Peluruhan zat A akan relatif lebih lama sehingga pembentukan zat B juga akan sama lamanya.
- Namun zat B akan lebih cepat luruh dibandingkan terbentuk.
- Akibatnya:
  - C akan lebih cepat terbentuk.
  - B akan lebih cepat mencapai maksimum sebelum A luruh sepenuhnya.
  - B akan lebih cepat luruh sebelum A luruh.

Berikut ini adalah beberapa simulasi lain menggunakan parameter r_A dan r_B yang berbeda-beda.

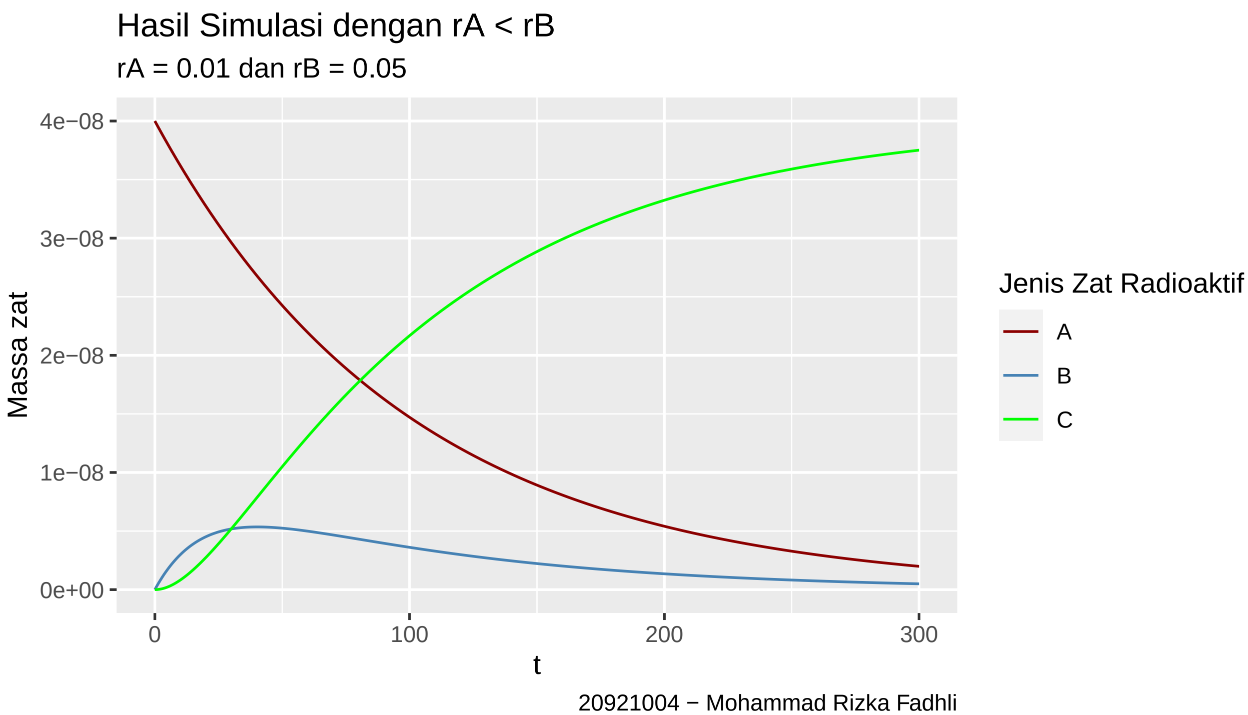

Figure 7: Ilustrasi Hasil Simulasi Saat rA $<$ rB (1)

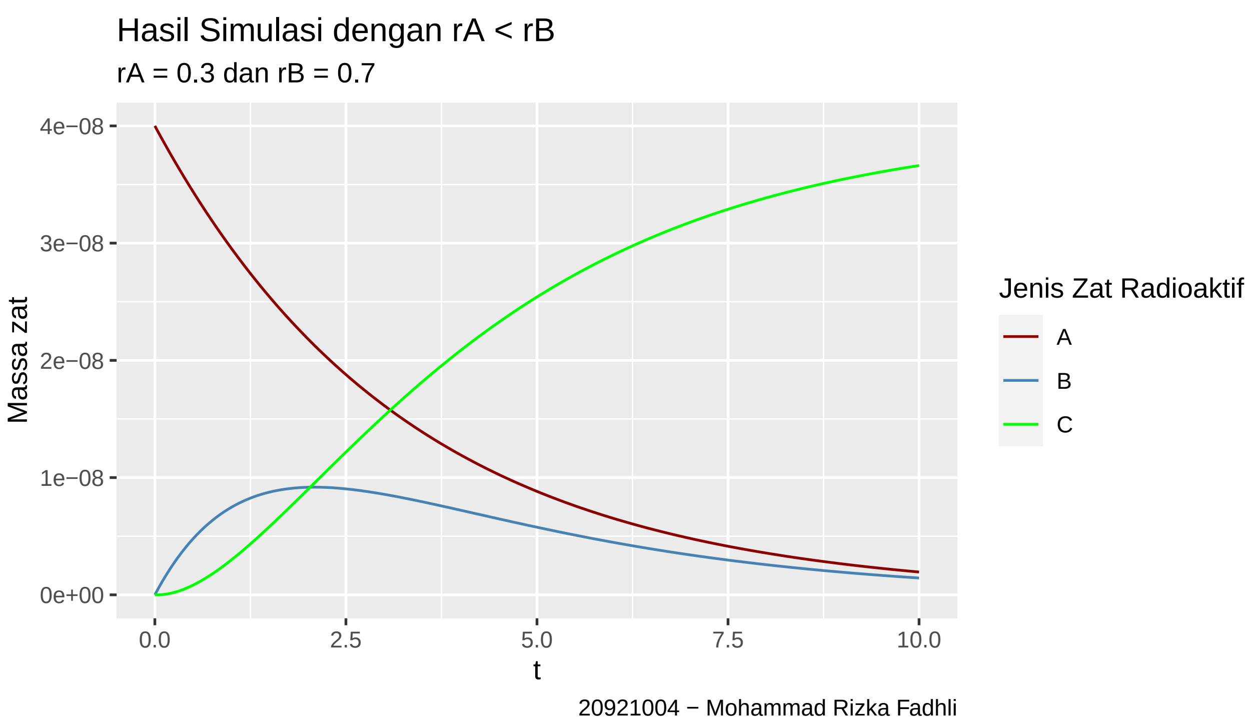

Figure 8: Ilustrasi Hasil Simulasi Saat rA $<$ rB (2)

4.4.2 $r_A > r_B$

Jika kondisi yang terjadi adalah $r_A > r_B$, maka:

- Peluruhan zat A akan relatif lebih cepat sehingga pembentukan zat B juga akan sama cepatnya.
- Zat B akan lebih cepat terbentuk dibandingkan meluruh.
- Akibatnya:
  - C akan lebih lambat terbentuk.
  - B akan lebih cepat mencapai maksimum bersamaan dengan A luruh sepenuhnya.
  - A akan lebih cepat luruh sebelum B luruh.

Berikut ini adalah beberapa simulasi lain menggunakan parameter r_A dan r_B yang berbeda-beda.

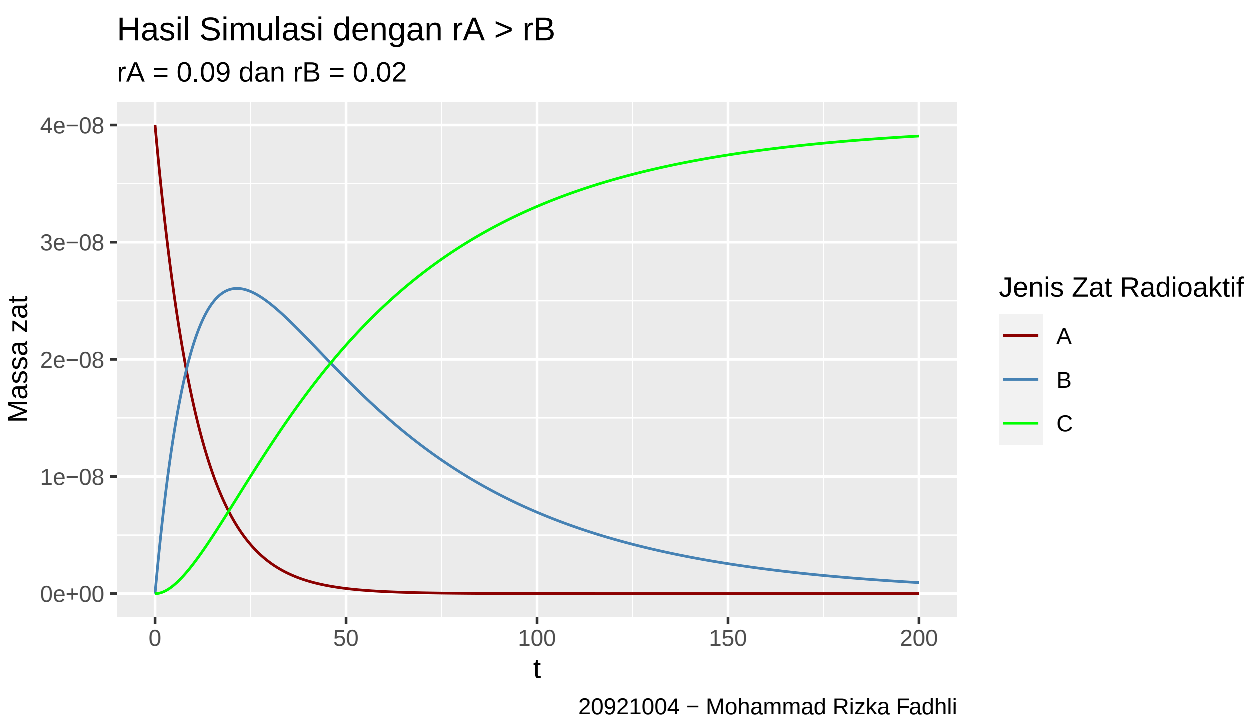

Figure 9: Ilustrasi Hasil Simulasi Saat rA > rB (1)

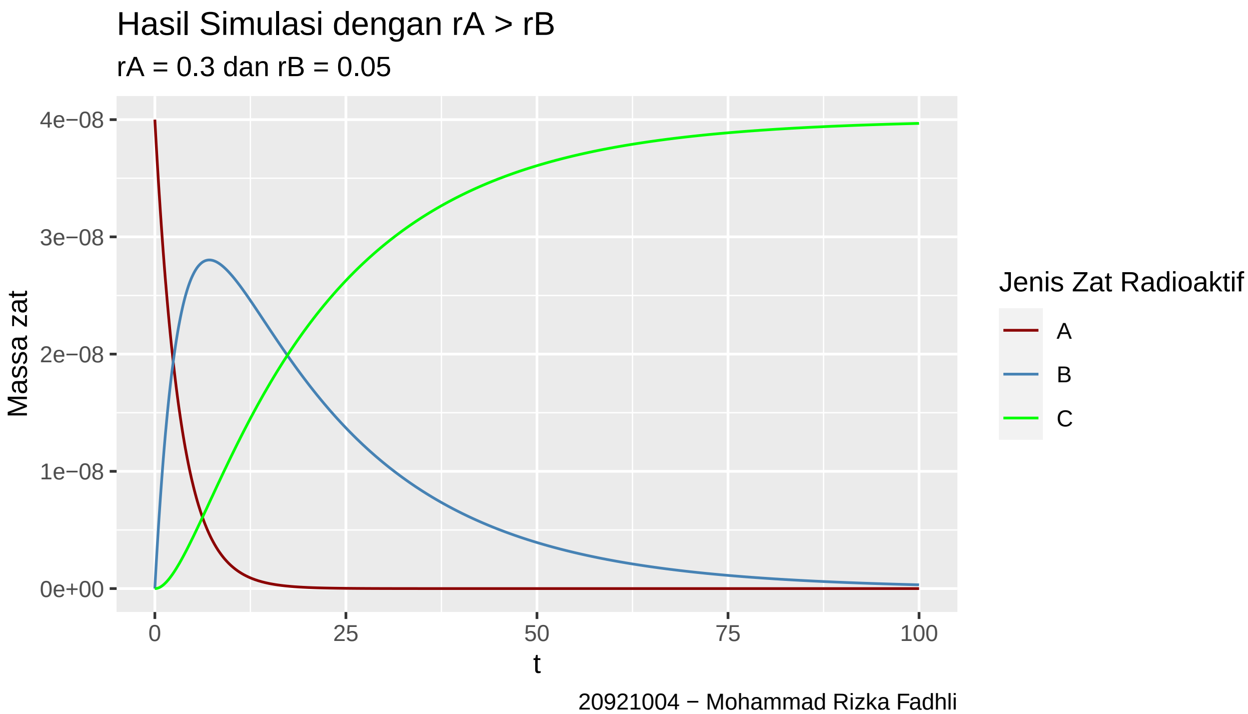

Figure 10: Ilustrasi Hasil Simulasi Saat rA > rB (2)

5 Kesimpulan

- Masalah persamaan diferensial bisa diselesaikan dengan menggunakan metode aproksimasi diskrit seperti algoritma Euler dan Runge Kutta 4^{th} *order*. Hasil kedua algoritma tersebut bisa mendekati hasil analitik jika kita perkecil Δt yang digunakan pada saat kita *run* program tersebut.
- Dinamika model akan bisa terlihat saat kita mengubah-ubah nilai laju peluruhan zat (r_A dan r_B).