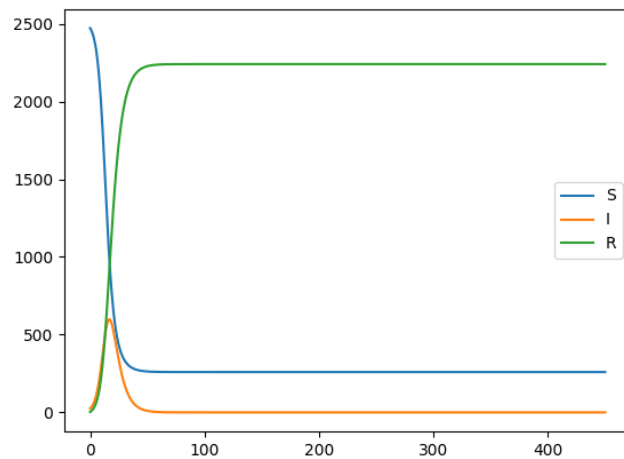


**PRAKTIKUM FISIKA KOMPUTASI**  
**PENYELESAIAN MODEL SIR MENGGUNAKAN METODE EULER**

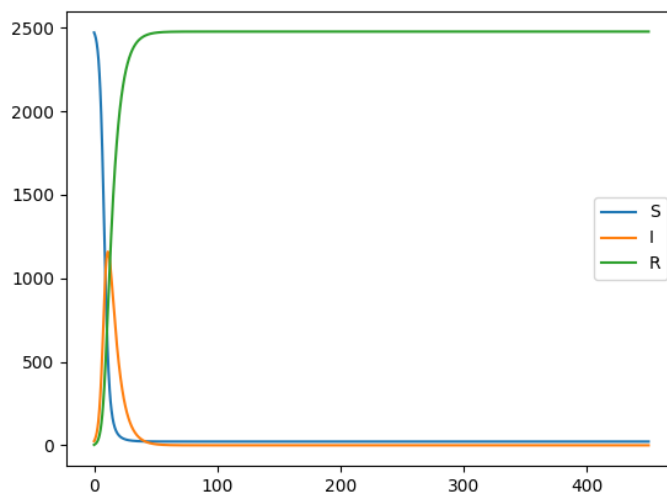
Darniel Trio Apriliansyah

NIM. 1227030009

1. **Buat grafik penyebaran virus Covid-19 dalam kurun waktu 450 hari dari 2500 penduduk di suatu lingkungan dengan jumlah awal terinfeksi 25 dan jumlah awal sembuh 3. Laju penularan Covid-19 sebesar 0.5 sedangkan laju pemulihan sebesar 0.2!**



2. **Buat grafik dengan laju pemulihan sebesar 0.15 dan laju penularan sebesar 0.7!**



### 3. Analisis grafik penyebaran virus Covid-19 pada soal 1&2!

Pada simulasi pertama ( $\beta = 0.5$ ,  $\gamma = 0.2$ ). Jumlah individu yang rentan (S) menurun dengan cepat di awal, menunjukkan banyak orang yang terpapar virus dalam waktu singkat. Namun, laju pemulihan yang cukup cepat ( $\gamma = 0.2$ ) menyebabkan jumlah orang yang sembuh (R) bertambah dengan stabil, sehingga jumlah orang terinfeksi (I) mencapai puncaknya lebih rendah dan mulai menurun lebih cepat. Ini menggambarkan situasi di mana wabah dapat dikendalikan dengan baik karena penularan tidak terlalu cepat dan pemulihan yang cepat.

Pada simulasi kedua ( $\beta = 0.7$ ,  $\gamma = 0.15$ ). Jumlah individu rentan (S) berkurang lebih cepat dibandingkan simulasi pertama, karena laju penularan yang lebih tinggi ( $\beta = 0.7$ ). Hal ini mengakibatkan lonjakan lebih besar pada jumlah orang terinfeksi (I), dengan puncak infeksi yang lebih tinggi dan terjadi lebih cepat. Laju pemulihan yang lebih lambat ( $\gamma = 0.15$ ) memperpanjang waktu yang dibutuhkan bagi orang yang terinfeksi untuk sembuh (R), sehingga jumlah kasus infeksi bertahan lebih lama sebelum mulai menurun. Ini menunjukkan skenario di mana wabah menjadi lebih sulit dikendalikan, dengan lebih banyak orang terinfeksi secara bersamaan dan pemulihan yang lambat.

Secara umum, kedua grafik berhasil menggambarkan model SIR dengan benar sesuai dengan rumus atau formula dari Euler. Dimana nilai S (rentan) selalu menurun seiring berjalannya waktu, nilai I (terinfeksi) selalu naik dan kemudian akan turun seiring berjalannya waktu, dan nilai R (sembuh) selalu naik seiring dengan berjalannya waktu.

### 4. Jelaskan algoritma pemrograman yang digunakan untuk menyelesaikan persoalan di atas!

Algoritma pemrograman ini mensimulasikan penyebaran penyakit dalam sebuah populasi menggunakan model SIR (Susceptible, Infected, Recovered). Simulasi dimulai dengan inisialisasi parameter waktu, populasi, serta jumlah individu yang rentan, terinfeksi, dan sembuh. Populasi dibagi ke dalam tiga kategori: orang yang rentan tertular (S), terinfeksi (I), dan sembuh (R), dengan jumlah awal yang ditentukan. Program ini menggunakan metode Euler untuk memecahkan sistem yang menggambarkan perubahan dalam setiap kategori seiring waktu. Dihitung berapa banyak orang yang pindah dari kategori rentan ke terinfeksi, dan dari terinfeksi ke sembuh, berdasarkan laju penularan ( $\beta$ ) dan laju pemulihan ( $\gamma$ ). Perubahan ini dihitung untuk setiap langkah waktu dengan langkah kecil yang ditentukan ( $h$ ). Setelah semua data didapat, jumlah orang dalam kategori S, I, dan R diplot dan ditampilkan dalam bentuk grafik, dimana nilai S selalu menurun, nilai I selalu naik dan turun, yang terakhir nilai R selalu naik.

```
import numpy as np
import matplotlib.pyplot as plt
```

Library Numpy digunakan untuk operasi matematika dan array

Library matplotlib.pyplot digunakan untuk memplot grafik dari hasil simulasi

```
t0 = 0
tn = 450
ndata = 1000
```

t0 adalah waktu awal

tn adalah durasi simulasi

ndata adalah jumlah data dalam simulasi

```
t = np.linspace(t0, tn, ndata)
h = t[2] - t[1]
```

t adalah array yang berisi deret waktu

h adalah selisih waktu antara dua titik waktu dalam array

```
N = 2500
I0 = 25
R0 = 3
S0 = N - I0 - R0
```

N adalah total jumlah populasi

I0 adalah jumlah awal terinfeksi

R0 adalah jumlah awal sembuh

S0 adalah jumlah awal individu yang rentan

```
I = np.zeros(ndata)
S = np.zeros(ndata)
R = np.zeros(ndata)
```

I, S, R adalah array yang masing masing akan menyimpan jumlah individu terinfeksi, sembuh, dan rentan

```
I[0] = I0
S[0] = S0
R[0] = R0
```

Set nilai awal hari ke 0

```
beta = 0.5
gamma = 0.2
```

beta adalah laju penularan

gamma adalah laju pemulihan

```
for n in range(0, ndata-1):
    S[n+1] = S[n] - h*beta/N*S[n]*I[n]
    I[n+1] = I[n] + h*beta/N*S[n]*I[n] - h*gamma*I[n]
    R[n+1] = R[n] + h*gamma*I[n]
```

Metode euler yang digunakan untuk menghitung model SIR

```
plt.plot(t, S, label='S')
plt.plot(t, I, label='I')
plt.plot(t, R, label='R')
plt.legend()
```

```
plt.show()
```

plt.plot digunakan untuk membuat plot dari data S, I dan R

plt.legend digunakan untuk membedakan antara S, I, dan R

plt.show digunakan untuk menampilkan grafik hasil simulasi

### Kode Program soal 1

```
import numpy as np
import matplotlib.pyplot as plt

t0 = 0                # waktu awal
tn = 450              # dalam waktu 300 hari
ndata = 2500          # jumlah data

t = np.linspace(t0, tn, ndata)
h = t[2] - t[1]

N = 2500              # jumlah populasi
I0 = 25               # jumlah awal individu terinfeksi
R0 = 3                # jumlah awal individu sembuh
S0 = N - I0 - R0      # jumlah awal individu rentan

I = np.zeros(ndata)
S = np.zeros(ndata)
R = np.zeros(ndata)

I[0] = I0
S[0] = S0
R[0] = R0

beta = 0.5            # laju penularan
gamma = 0.2           # laju pemulihan

for n in range(0, ndata-1):
    S[n+1] = S[n] - h*beta/N*S[n]*I[n]
    I[n+1] = I[n] + h*beta/N*S[n]*I[n] - h*gamma*I[n]
    R[n+1] = R[n] + h*gamma*I[n]

plt.plot(t, S, label='S')
plt.plot(t, I, label='I')
plt.plot(t, R, label='R')
plt.legend()
plt.show()
```

## Kode Program Soal 2

```
import numpy as np
import matplotlib.pyplot as plt

t0 = 0          # waktu awal
tn = 450        # dalam waktu 300 hari
ndata = 2500    # jumlah data

t = np.linspace(t0, tn, ndata)
h = t[2] - t[1]

N = 2500        # jumlah populasi
I0 = 25         # jumlah awal individu terinfeksi
R0 = 3          # jumlah awal individu sembuh
S0 = N - I0 - R0 # jumlah awal individu rentan

I = np.zeros(ndata)
S = np.zeros(ndata)
R = np.zeros(ndata)

I[0] = I0
S[0] = S0
R[0] = R0

beta = 0.7      # laju penularan
gamma = 0.15    # laju pemulihan

for n in range(0, ndata-1):
    S[n+1] = S[n] - h*beta/N*S[n]*I[n]
    I[n+1] = I[n] + h*beta/N*S[n]*I[n] - h*gamma*I[n]
    R[n+1] = R[n] + h*gamma*I[n]

plt.plot(t, S, label='S')
plt.plot(t, I, label='I')
plt.plot(t, R, label='R')
plt.legend()
plt.show()
```