

## Praktikum – SIR

Kode Python ([simulasi\\_model\\_sir.py](#)) yang dibuat oleh Sandy digunakan untuk mensimulasikan model SIR dalam sebuah populasi.

### Model SIR

$$N = S + I + R$$

#### Variabel yang digunakan:

- S = Individu yang tidak terinfeksi tapi rentan terhadapnya
- I = Individu yang dapat menularkan penyakit ke yang rentan
- R = Individu yang sudah sembuh dari penyakit

#### Model SIR sebagai berikut:

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$

$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$

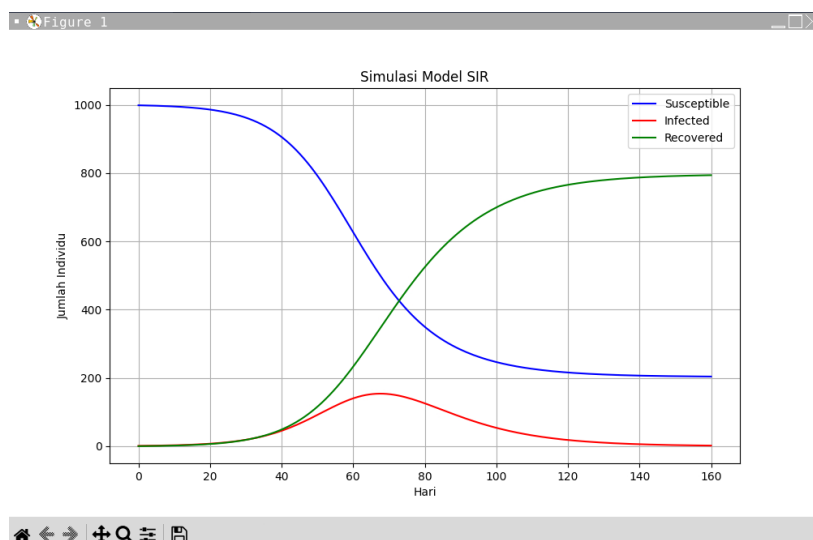
$$\frac{dR}{dt} = \gamma I$$

#### Parameter:

- Beta ( $\beta$ ): Laju Penularan
- Gamma ( $\gamma$ ): Laju Kesembuhan, periode rata” penularan hingga sembuh  $\frac{1}{\gamma}$

*Catatan Presentasi “Compartment Model” FLEARN*

Output yang dihasilkan dalam menjalankan kode pertama file simulasi\_model\_sir.py milik Sandy:



Grafik garis yang dihasilkan menunjukkan perubahan populasi yang rentan (S), terinfeksi (I) dan sembuh (R) terhadap waktu. Kode kedua adalah model tersebut yang dibuat interaktif dengan dapat merubah parameter, kondisi awal (S dan I) dan jangka waktu dengan slider.

Grafik kode pertama menunjukkan populasi yang rentan perlahan menjadi terinfeksi dan populasi sembuh perlahan menaik akibat sembuh dari terinfeksi. Jumlah populasi yang terinfeksi menjadi 0 ketika sudah mencapai kesetimbangan antara yang rentan dan sembuh (tidak dapat sakit lagi). Hal ini dapat digunakan untuk prediksi jangka waktu kapan suatu populasi akan berhenti terinfeksi.

### Penjelasan kode simulasi\_model\_sir.py Arsandy:

#### 1. Mendefinisi Parameter yang digunakan dalam model:

- $\beta = 0.2$  adalah Tingkat infeksi
- $\gamma = 1/10$  adalah Tingkat kesembuhan

#### 2. Menetapkan kondisi awal:

- $S_0 = N-1$  adalah jumlah penduduk yang merupakan sisa dari totalnya
- $I_0 = 1$  adalah jumlah awal yang terinfeksi
- $R_0 = 0$  adalah jumlah awal yang sembuh

#### 3. Mendefinisi jangka waktu:

- $t = \text{np.linspace}(0, 160, 160)$  membuat array/larik dengan jangka 0 hingga 160 sebanyak 160 titik atau per hari.

#### 4. Membuat fungsi SIR:

```
def sir_model(y,t,N,beta,gamma):  
    S,I,R = y  
    dSdt = -beta * S * I / N  
    dIdt = beta * S * I / N - gamma * I  
    dRdt = gamma * I  
    return [dSdt,dIdt,dRdt]
```

Sama seperti Lotka-Volterra, fungsi tersebut dibuat untuk memudahkan penyelesaian sistem persamaan.

#### 5. Menyelesaikan sistem persamaan SIR:

```
y0 = [S0,I0,R0]  
solution = odeint(sir_model, y0, t, args=(N,beta,gamma))  
menggunakan fungsi odeint yang terdapat pada library SciPy untuk  
memudahkan menyelesaikan sistem persamaan diatas.
```

```
S,I,R = solution.T
```

baris diatas memisahkan hasil penyelesaian diatas dengan metode transpose.

#### 6. Membuat grafik:

- Grafik garis perbandingan jumlah dengan waktu dengan tiga variabel (S,I,R).