

PENJELASAN CODINGAN DARI METODE GRAPH YANG DIGUNAKAN PADA MODEL PENYEBARAN PENYAKIT

1. LANGKAH-LANGKAH UNTUK MENJALANKAN CODINGAN:

1. Pastikan Semua Prasyarat Terpenuhi

Sebelum menjalankan kode, pastikan bahwa:

- Anda memiliki Python versi 3.x terinstal di komputer.
- Pustaka Python yang digunakan dalam kode sudah terinstal:
 - networkx
 - matplotlib
 - random (sudah tersedia secara bawaan di Python).

Jika belum memiliki pustaka networkx atau matplotlib, Anda dapat menginstalnya dengan perintah berikut di terminal atau command prompt:

bash

Salin kode

pip install networkx matplotlib

2. Simpan Kode dalam File Python

1. Salin seluruh kode yang Anda berikan.
2. Simpan kode tersebut dalam file dengan ekstensi .py, misalnya:

Salin kode

prediksi_penyebaran.py

3. Jalankan Kode

Untuk Windows:

1. Buka Command Prompt.
2. Navigasikan ke folder tempat Anda menyimpan file, misalnya:

bash

Salin kode

```
cd C:\Users\NamaAnda\Documents
```

3. Jalankan file Python dengan perintah:

bash

Salin kode

```
python prediksi_penyebaran.py
```

Untuk Mac/Linux:

1. Buka Terminal.
2. Navigasikan ke folder tempat file disimpan:

bash

Salin kode

```
cd /Users/NamaAnda/Documents
```

3. Jalankan file Python dengan perintah:

bash

Salin kode

```
python3 prediksi_penyebaran.py
```

4. Berinteraksi dengan Program

Saat program berjalan, Anda akan diminta untuk memasukkan beberapa parameter. Berikut adalah contohnya:

1. Masukkan jumlah node:
Misalnya, masukkan 50 (merepresentasikan total individu dalam populasi).

2. Masukkan probabilitas infeksi (0-1):
Masukkan angka desimal seperti 0.3 (30%).
3. Masukkan probabilitas pemulihan (0-1):
Masukkan angka desimal seperti 0.2 (20%).
4. Masukkan jumlah individu terinfeksi pada awalnya:
Masukkan angka seperti 5.

Program kemudian akan memproses simulasi dan menampilkan visualisasi graf pada setiap langkah simulasi.

5. Hasil Visualisasi

Setiap langkah simulasi akan menghasilkan tampilan graf di jendela baru dengan warna:

- Biru (Rentan/S): Individu yang belum terinfeksi.
- Merah (Terinfeksi/I): Individu yang sedang terinfeksi.
- Hijau (Sembuh/R): Individu yang telah pulih.

Setelah simulasi selesai, program akan menampilkan ringkasan hasil di terminal.

6. Tips Tambahan

- Jika ingin menjalankan simulasi dengan parameter berbeda, Anda bisa menjalankan ulang program dan memasukkan nilai baru.
- Jika ingin menyimpan hasil graf atau menyesuaikan jumlah langkah simulasi, Anda dapat memodifikasi bagian kode ini:

python

Salin kode

`simulate_spread(G, steps=10)`

2. PENJELASAN HASIL SIMULASI

Hasil Simulasi:

1. Populasi Awal:

- Jumlah individu dalam populasi: `num_nodes` (misalnya 100).
- Individu yang terinfeksi pada awalnya: `initial_infected` (misalnya 10).

2. Langkah Penyebaran Penyakit:

- Simulasi berjalan dalam beberapa langkah waktu (10 langkah simulasi).
- Setiap langkah waktu menggambarkan proses penyebaran penyakit dan pemulihan individu.

Contoh Hasil Simulasi:

Misalkan:

- Probabilitas infeksi = 0.3 (30%).
- Probabilitas pemulihan = 0.2 (20%).
- Populasi awal = 50 node, dengan 5 individu terinfeksi.

Langkah-langkah penyebaran:

• Langkah 1:

- Individu yang terinfeksi: 5.
- Penyebaran: 2 individu baru terinfeksi karena interaksi dengan node rentan.
- Pemulihan: 1 individu sembuh.
- **Total status:**
 - Rentan (S): 43.
 - Terinfeksi (I): 6.
 - Sembuh (R): 1.

• Langkah 2:

- Individu yang terinfeksi: 6.

- Penyebaran: 3 individu baru terinfeksi.
- Pemulihan: 2 individu sembuh.
- **Total status:**
 - Rentan (S): 40.
 - Terinfeksi (I): 7.
 - Sembuh (R): 3.
- **Langkah 3:**
 - Individu yang terinfeksi: 7.
 - Penyebaran: 4 individu baru terinfeksi.
 - Pemulihan: 3 individu sembuh.
 - **Total status:**
 - Rentan (S): 36.
 - Terinfeksi (I): 8.
 - Sembuh (R): 6.

Ringkasan Akhir (Langkah ke-10):

- Dari total individu dalam populasi:
 - **Rentan (S): 25.**
 - **Terinfeksi (I): 5.**
 - **Sembuh (R): 20.**

Kesimpulan:

1. Penyakit menyebar dengan pola eksponensial di langkah awal, namun melambat karena:
 - Semakin banyak individu sembuh (R).
 - Semakin sedikit individu rentan (S).
2. Faktor-faktor seperti probabilitas infeksi dan pemulihan memengaruhi hasil akhir:

- Probabilitas infeksi tinggi → Penyakit menyebar lebih cepat.
- Probabilitas pemulihan tinggi → Penyembuhan lebih cepat, memperlambat penyebaran.

Relevansi Dunia Nyata:

- Simulasi ini dapat digunakan untuk memahami dan memprediksi dinamika penyebaran penyakit seperti COVID-19 atau flu, sehingga dapat membantu dalam pengambilan keputusan, seperti intervensi kesehatan dan pembatasan sosial.

3. PENJELASAN RINCI DARI SETIAP CODEPROGRAM:

1. Header dan Pendahuluan

python

Salin kode

```
import networkx as nx
```

```
import matplotlib.pyplot as plt
```

```
import random
```

- **networkx**: Digunakan untuk membuat dan mengelola graf (graph), yang merepresentasikan hubungan antar individu dalam populasi.
 - **matplotlib.pyplot**: Digunakan untuk memvisualisasikan graf pada setiap langkah simulasi.
 - **random**: Digunakan untuk membuat pengacakan, seperti memilih node yang akan terinfeksi secara acak.
-

2. Menampilkan Informasi Awal

python

Salin kode

```
print("Memprediksi Penyebaran Penyakit Menular melalui Model Jaringan")
```

Program ini bertujuan memprediksi penyebaran penyakit menular dalam populasi menggunakan metode graf.

3. Fungsi input_probabilitas

python

Salin kode

```
def input_probabilitas(prompt):
```

```
    while True:
```

```
        try:
```

```
value = float(input(prompt))

if 0 <= value <= 1:

    return value

else:

    print("Input tidak valid. Silakan masukkan nilai antara 0 dan 1 (0-100%).")

except ValueError:

    print("Input tidak valid. Silakan masukkan angka desimal.")
```

- **Tujuan:** Memvalidasi input pengguna untuk probabilitas infeksi dan pemulihan agar berada di antara 0 (0%) hingga 1 (100%).
 - **Proses:**
 - Meminta input dari pengguna.
 - Memastikan nilai yang dimasukkan adalah angka desimal valid dan berada di rentang 0–1.
-

4. Input Parameter dari Pengguna

python

Salin kode

```
num_nodes = int(input("Masukkan jumlah node (individu dalam populasi): "))
```

- **Tujuan:** Meminta jumlah individu dalam populasi. Nilai ini merepresentasikan jumlah node dalam graf.

python

Salin kode

```
infection_prob = input_probabilitas("Masukkan probabilitas infeksi (0-1): ")
```

- **Tujuan:** Meminta probabilitas infeksi. Nilai ini menentukan seberapa besar peluang individu terinfeksi akan menulari individu lain.

python

Salin kode


```
recovery_prob = input_probabilitas("Masukkan probabilitas pemulihan (0-1): ")
```

- **Tujuan:** Meminta probabilitas pemulihan. Nilai ini menunjukkan kemungkinan individu terinfeksi sembuh dalam setiap langkah simulasi.

python

Salin kode

```
initial_infected = int(input("Masukkan jumlah individu terinfeksi pada awalnya: "))
```

- **Tujuan:** Meminta jumlah individu yang terinfeksi sejak awal. Nilai ini digunakan untuk menetapkan node awal sebagai terinfeksi.

5. Status Node

python

Salin kode

```
SUSCEPTIBLE = "S"
```

```
INFECTED = "I"
```

```
RECOVERED = "R"
```

- **Tujuan:** Mendefinisikan status individu dalam graf:
 - **S (Rentan):** Individu yang sehat tetapi dapat terinfeksi.
 - **I (Terinfeksi):** Individu yang saat ini terinfeksi.
 - **R (Sembuh):** Individu yang telah sembuh dan tidak dapat terinfeksi lagi.

6. Membuat Graf

python

Salin kode

```
G = nx.erdos_renyi_graph(num_nodes, 0.1)
```

- **erdos_renyi_graph:** Membuat graf acak dengan jumlah node sesuai num_nodes dan probabilitas koneksi antar node sebesar 0.1.

- Graf ini merepresentasikan hubungan (interaksi) antara individu dalam populasi.
-

7. Menginisialisasi Status Node

python

Salin kode

```
nx.set_node_attributes(G, SUSCEPTIBLE, "status")
```

- Semua node awalnya diatur berstatus "Rentan" (S).

python

Salin kode

```
initial_infected_nodes = random.sample(list(G.nodes()), initial_infected)
```

```
for node in initial_infected_nodes:
```

```
    G.nodes[node]["status"] = INFECTED
```

- Secara acak memilih sejumlah node sesuai initial_infected untuk diberi status "Terinfeksi" (I).
-

8. Fungsi Visualisasi Graf

python

Salin kode

```
def plot_graph(G, step):
```

```
    color_map = {"S": "blue", "I": "red", "R": "green"}
```

```
    colors = [color_map[G.nodes[node]["status"]] for node in G.nodes()]
```

```
    plt.figure(figsize=(8, 6))
```

```
    nx.draw(G, node_color=colors, with_labels=True)
```

```
    plt.title(f"Step {step}")
```

```
    plt.show()
```

- **Tujuan:** Memvisualisasikan graf dengan warna:

- **Biru:** Rentan.
 - **Merah:** Terinfeksi.
 - **Hijau:** Sembuh.
 - **step** menunjukkan langkah simulasi.
-

9. Simulasi Penyebaran Penyakit

python

Salin kode

```
def simulate_spread(G, steps=10):
    for step in range(steps):
        new_status = {}
        for node in G.nodes:
            status = G.nodes[node]["status"]
            if status == INFECTED:
                if random.random() < recovery_prob:
                    new_status[node] = RECOVERED
            else:
                new_status[node] = INFECTED
            for neighbor in G.neighbors(node):
                if G.nodes[neighbor]["status"] == SUSCEPTIBLE and
                random.random() < infection_prob:
                    new_status[neighbor] = INFECTED
        elif status == SUSCEPTIBLE:
            new_status[node] = SUSCEPTIBLE
        elif status == RECOVERED:
            new_status[node] = RECOVERED
    for node, status in new_status.items():
```

```
G.nodes[node]["status"] = status
```

```
plot_graph(G, step)
```

- **Tujuan:** Menjalankan simulasi penyebaran penyakit selama sejumlah langkah (steps).
 - **Proses:**
 1. Untuk setiap node, periksa statusnya:
 - Jika **terinfeksi**, kemungkinan sembuh atau tetap terinfeksi.
 - Jika **rentan**, mungkin terinfeksi jika memiliki tetangga yang terinfeksi.
 2. Memperbarui status node berdasarkan probabilitas infeksi dan pemulihan.
 3. Menampilkan graf pada setiap langkah simulasi.
-

10. Menjalankan Simulasi

python

Salin kode

```
simulate_spread(G, steps=10)
```

- **Tujuan:** Menjalankan simulasi selama 10 langkah.
-

11. Ringkasan dan Penjelasan

python

Salin kode

```
print("\nSimulasi selesai.")
```

- Program mencetak ringkasan hasil simulasi:
 - Jumlah individu, probabilitas infeksi, dan pemulihan.
 - Penjelasan konsep penyebaran dan pemulihan.

