

# **LAPORAN TUGAS UAS**

## **PEMODELAN & SIMULASI IF-41-GAB02**

*Penyebaran penyakit/virus dengan Random Walk*



Oleh:

Haura Athaya Salka - 1301183454

Tri Ayu Syifa'ur Rohmah - 1301180254

**IF 41-GAB02**  
**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**UNIVERSITAS TELKOM**  
**BANDUNG**  
**2021**

## **DAFTAR ISI**

1.	Deskripsi Permasalahan	3
2.	Pemodelan Matematika	3
3.	Dokumentasi	5
3.1	Algoritma	5
3.2	Screenshot Program	5
3.3	Hasil Simulasi	6
4.	Kesimpulan	7

## **1. DESKRIPSI PERMASALAHAN**

Proses penyebaran suatu penyakit/virus dapat disimulasikan secara sederhana dengan menggunakan Random Walk. Pada metode ini, setiap individu direpresentasikan sebagai partikel yang bergerak bebas secara acak. Proses simulasi diawali dengan mendefinisikan sejumlah individu dari suatu komunitas yang sudah terinfeksi. Setelah itu, simulasi dilakukan dengan mendefinisikan perubahan posisi dari masing-masing individu secara acak. Secara sederhana, proses infeksi terjadi pada saat individu sehat berada pada posisi yang sama dengan individu yang terinfeksi. Selain itu, individu yang sudah sembuh diasumsikan memiliki imun terhadap penyakit/virus sehingga tidak akan terinfeksi untuk kedua kalinya. Proses simulasi berakhir setelah tidak ada lagi individu yang terinfeksi.

Secara lebih detail, ruang simulasi perlu didefinisikan untuk menghindari pergerakan individu yang terlalu menyebar. Terkait hal ini, maka individu yang bergerak melebihi batas area perlu dikontrol dengan menggunakan metode periodic boundary condition (PBC). Selain itu, penerapan karantina wilayah pada level tertentu dapat direpresentasikan dengan mendefinisikan suatu variabel yang menentukan probabilitas suatu individu untuk bergerak. Hasil simulasi tersebut dapat menunjukkan fluktuasi jumlah individu yang terinfeksi tiap harinya dan waktu yang diperlukan oleh komunitas untuk pulih dari wabah penyakit/virus atau tidak ada lagi individu yang terinfeksi. Pada kasus ini, satu iterasi diasumsikan sebagai satu hari.

## **2. PEMODELAN MATEMATIKA**

Random Walk (RW) merupakan metode mengestimasi nilai future tanpa memperhatikan past observations, nilai yang muncul dari Random Walk merupakan suatu keacakan. Random Walk memiliki karakteristik panjang langkah atau jaraknya konstan. Random Walk banyak diimplementasikan untuk memodelkan data-data yang seringkali sulit ditebak. Dalam banyak hal beberapa model advanced melibatkan proses RW didalamnya.

Pada simulasi Random Walk ini, digunakan juga Periodic Boundary Condition. PBC pada simulasi penyebaran virus ini berguna untuk mempertahankan pengamatan terhadap individu di suatu lokasi sehingga hasil update posisi diharapkan tidak keluar dari batas ruang simulasi (20x20) yang sudah ditentukan.

Selain itu, digunakan juga Brownian Motion untuk menentukan prediksi arah gerak yang dilakukan oleh individu.

### 3. ALGORITMA

#### 3.1 Algoritma Program

##### 3.1.1. Import Library

```
1  import random
2  from matplotlib import animation, rc
3  import matplotlib.pyplot as plt
4  from IPython.display import HTML
5  from celluloid import Camera as Camera
```

##### 3.1.2. Inisialisasi Variable Scalar

```
10  ## INISIALISASI VARIABLE SCALAR ##
11  individu = 200
12  rasio = 5 # 5%
13  prob = 0.8 # 80%
14  waktuPulih = 10 # hari pemulihan
15
16  jml_terinfeksi = 0 # inisialisasi jumlah terinfeksi
17
18  # UKURAN RUANG SIMULASI #
19  x_min = int(0)
20  x_max = int(20)
21  y_min = int(0)
22  y_max = int(20)
23  x_range = x_max - x_min
24  y_range = y_max - y_min
```

### 3.1.3. Inisialisasi Variable List

```
26  ## INISIALISASI VARIABEL LIST ##
27  statusHealth = [] # status terinfeksi / tidak
28  waktuInfeksi = [] # waktu terinfeksi / recovery
29  totalInfeksi = [] # total individu yang terinfeksi
30  hari = []
31
32  x_pos = [] # posisi X
33  y_pos = [] # posisi Y
```

### 3.1.4. Status Individu Awal

```
35  # STATUS INDIVIDU AWAL #
36  for i in range(individu):
37      x_infeksi = []
38      y_infeksi = []
39      x_sehat = []
40      y_sehat = []
41
42      x_pos.append(random.randint(x_min, x_max))
43      y_pos.append(random.randint(y_min, y_max))
44
45      randRasio = (random.randint(1, 100))
46      if randRasio <= rasio:
47          status = "terinfeksi"
48          statusHealth.append(status)
49          x_infeksi.append(random.randint(x_min, x_max))
50          y_infeksi.append(random.randint(y_min, y_max))
51          jml_terinfeksi += 1
52          waktuInfeksi.append(1)
53      else:
54          status = "sehat"
55          statusHealth.append(status)
56          waktuInfeksi.append(0)
57          x_sehat.append(random.randint(x_min, x_max))
58          y_sehat.append(random.randint(y_min, y_max))
```

### 3.1.5. Iterasi

```
60 # ITERASI #
61
62 hari_ke = 1
63 hari.append(hari_ke)
64 totalInfeksi.append(jml_terinfeksi)
65
66 while (jml_terinfeksi > 0):
67     for i in range(individu):
68         x_infeksi = []
69         y_infeksi = []
70         x_sehat = []
71         y_sehat = []
```

#### 3.1.5.1. Update Posisi Individu

```
73 # UPDATE POSISI
74 probRand = random.uniform(0, 1)
75 if (probRand >= prob):
76     arah = random.uniform(0, 1)
77     if arah <= 0.25: # kanan
78         x_pos[i] = x_pos[i-1] + 1
79         y_pos[i] = y_pos[i-1]
80     elif arah <= 0.5: # kiri
81         x_pos[i] = x_pos[i-1] - 1
82         y_pos[i] = y_pos[i-1]
83     elif arah <= 0.75: # atas
84         x_pos[i] = x_pos[i-1]
85         y_pos[i] = y_pos[i-1] + 1
86     elif arah <= 1: # bawah
87         x_pos[i] = x_pos[i-1]
88         y_pos[i] = y_pos[i-1] - 1
89
```

### 3.1.5.2. Koreksi Dengan PBC

```
90 # KOREKSI DGN PBC
91 if x_pos[i] > x_max:
92     x_pos[i] = x_pos[i] - 1
93 elif x_pos[i] < x_min:
94     x_pos[i] = x_pos[i] + 1
95 elif y_pos[i] > y_max:
96     y_pos[i] = y_pos[i] - 1
97 elif y_pos[i] < y_min:
98     y_pos[i] = y_pos[i] + 1
```

### 3.1.5.3. Pengecekan Penyebaran Virus (Posisi Sama)

```
# PENGECEKAN PENYEBARAN VIRUS (INDIVIDU PADA POSISI YG SAMA)
for j in range(individu):
    if j != i:
        if (x_pos[j] == x_pos[i] and y_pos[j] == y_pos[i]):
            if statusHealth[j] != "terinfeksi" and statusHealth[j] != "imun":
                if statusHealth[i] == "terinfeksi":
                    statusHealth[j] = "terinfeksi"
                    waktuInfeksi[j] = 1
                    jml_terinfeksi += 1
            elif statusHealth[j] == "terinfeksi":
                if statusHealth[i] == "sehat":
                    statusHealth[i] = "terinfeksi"
                    waktuInfeksi[i] = 1
                    jml_terinfeksi += 1
```

### 3.1.5.4. Update Status Kesehatan Individu

```
# PENGECEKAN PERUBAHAN STATUS TERINFEKSI -> IMUN
if statusHealth[i] == "terinfeksi":
    if waktuInfeksi[i] <= waktuPulih:
        waktuInfeksi[i] += 1
        x_infeksi.append(x_pos[i])
        y_infeksi.append(y_pos[i])
    else:
        statusHealth[i] = "imun"
        jml_terinfeksi -= 1
        x_sehat.append(x_pos[i])
        y_sehat.append(y_pos[i])
else:
    x_sehat.append(x_pos[i])
    y_sehat.append(y_pos[i])
```

### 3.1.6. Plotting

```
plt.figure(1)
plt.subplot(1, 2, 1)
plt.title('Simulasi Random Walk Penyebaran Virus')
plt.plot(x_infeksi, y_infeksi, 'ro')
plt.plot(x_sehat, y_sehat, 'go')
```

```
# PERUBAHAN HARI
hari_ke += 1
hari.append(hari_ke)

# PERUBAHAN INFEKSI
totalInfeksi.append(jml_terinfeksi)
plt.subplot(1, 2, 2)
plt.plot(totalInfeksi, color='blue')
plt.title("Grafik Penyebaran Virus")
plt.xlabel('Jumlah Hari')
plt.ylabel('Jumlah Terinfeksi')
Camera.snap()
```

```
# Animasi Plot
anim = Camera.animate(interval=1000)
plt.grid(True, which="both")
anim.save('plot dan grafik.mp4')
rc('animation', html='jshtml')
plt.show()
```

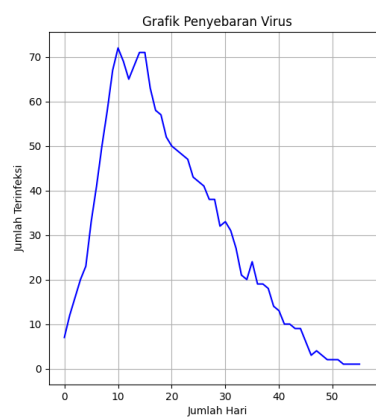
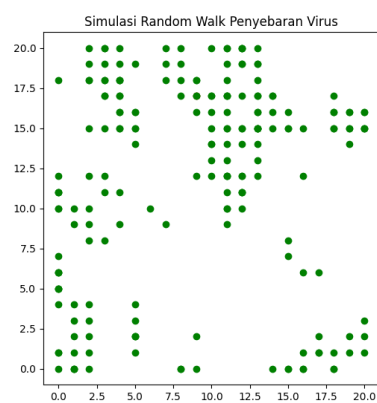
## 3.2 Hasil Simulasi

Total waktu pemulihan yang diperlukan oleh komunitas: 58



Hari ke 1 Jumlah yang Terinfeksi: 7	Hari ke 13 Jumlah yang Terinfeksi: 65	Hari ke 26 Jumlah yang Terinfeksi: 42	Hari ke 39 Jumlah yang Terinfeksi: 18
Hari ke 2 Jumlah yang Terinfeksi: 12	Hari ke 14 Jumlah yang Terinfeksi: 68	Hari ke 27 Jumlah yang Terinfeksi: 41	Hari ke 40 Jumlah yang Terinfeksi: 14
Hari ke 3 Jumlah yang Terinfeksi: 16	Hari ke 15 Jumlah yang Terinfeksi: 71	Hari ke 28 Jumlah yang Terinfeksi: 38	Hari ke 41 Jumlah yang Terinfeksi: 13
Hari ke 4 Jumlah yang Terinfeksi: 20	Hari ke 16 Jumlah yang Terinfeksi: 71	Hari ke 29 Jumlah yang Terinfeksi: 38	Hari ke 42 Jumlah yang Terinfeksi: 10
Hari ke 5 Jumlah yang Terinfeksi: 23	Hari ke 17 Jumlah yang Terinfeksi: 63	Hari ke 30 Jumlah yang Terinfeksi: 32	Hari ke 43 Jumlah yang Terinfeksi: 10
Hari ke 6 Jumlah yang Terinfeksi: 33	Hari ke 18 Jumlah yang Terinfeksi: 58	Hari ke 31 Jumlah yang Terinfeksi: 33	Hari ke 44 Jumlah yang Terinfeksi: 9
Hari ke 7 Jumlah yang Terinfeksi: 41	Hari ke 19 Jumlah yang Terinfeksi: 57	Hari ke 32 Jumlah yang Terinfeksi: 31	Hari ke 45 Jumlah yang Terinfeksi: 9
Hari ke 8 Jumlah yang Terinfeksi: 50	Hari ke 20 Jumlah yang Terinfeksi: 52	Hari ke 33 Jumlah yang Terinfeksi: 27	Hari ke 46 Jumlah yang Terinfeksi: 6
Hari ke 9 Jumlah yang Terinfeksi: 58	Hari ke 21 Jumlah yang Terinfeksi: 50	Hari ke 34 Jumlah yang Terinfeksi: 21	Hari ke 47 Jumlah yang Terinfeksi: 3
Hari ke 10 Jumlah yang Terinfeksi: 67	Hari ke 22 Jumlah yang Terinfeksi: 49	Hari ke 35 Jumlah yang Terinfeksi: 20	Hari ke 48 Jumlah yang Terinfeksi: 4
Hari ke 11 Jumlah yang Terinfeksi: 72	Hari ke 23 Jumlah yang Terinfeksi: 48	Hari ke 36 Jumlah yang Terinfeksi: 24	Hari ke 49 Jumlah yang Terinfeksi: 3
Hari ke 12 Jumlah yang Terinfeksi: 69	Hari ke 24 Jumlah yang Terinfeksi: 47	Hari ke 37 Jumlah yang Terinfeksi: 19	Hari ke 50 Jumlah yang Terinfeksi: 2
	Hari ke 25 Jumlah yang Terinfeksi: 43	Hari ke 38 Jumlah yang Terinfeksi: 19	Hari ke 51 Jumlah yang Terinfeksi: 2

Hari ke 52 Jumlah yang Terinfeksi: 2
Hari ke 53 Jumlah yang Terinfeksi: 1
Hari ke 54 Jumlah yang Terinfeksi: 1
Hari ke 55 Jumlah yang Terinfeksi: 1
Hari ke 56 Jumlah yang Terinfeksi: 1
Hari ke 57 Jumlah yang Terinfeksi: 1
Hari ke 58 Jumlah yang Terinfeksi: 0



#### **4. KESIMPULAN**

PBC dan Random Walk cocok digunakan untuk simulasi sederhana pada penyebaran virus. Pada kasus ini penyebaran virus di implementasikan dengan Random Walk yang menyebabkan setiap running menghasilkan angka yang berbeda, dan individu yang diimplementasikan dikoreksi perubahan posisinya agar tidak melewati batasan ruang menggunakan PBC. Dari simulasi yang telah dilakukan diketahui bahwa penyebaran virus pada suatu ruang dengan jumlah individu 200 terjadi selama 58 hari, dengan puncak individu terinfeksi pada periode hari ke 10 hingga ke 16. Jumlah tertinggi individu yang terinfeksi yaitu 72 orang dan terjadi pada hari ke 11. Setelah penyebaran virus mengalami masa puncak, angka tersebut berangsur-angsur turun hingga sudah tidak ada lagi individu yang terinfeksi pada hari ke 58.