

LAPORAN TUGAS UAS

PEMODELAN & SIMULASI IF-41-GAB01

Simulasi Sebaran Penyakit dengan Random Walk

Dosen Pengampu Didit Adytia, Ph.D.



Disusun oleh :

Luqman Haries (1301180072)

PROGRAM STUDI INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2021

DAFTAR ISI

| | |
|--|----------|
| PENDAHULUAN | 3 |
| Deskripsi Permasalahan | 3 |
| Instruksi Tugas | 3 |
| PEMODELAN | 4 |
| Random Walk | 4 |
| Model Individu | 4 |
| Rasio Individu yang terinfeksi dan Waktu Pemulihan | 4 |
| Algoritma | 5 |
| DOKUMENTASI | 6 |
| Screenshot Program | 6 |
| Hasil | 7 |

PENDAHULUAN

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.

Instruksi Tugas

Simulasi penyebaran penyakit/virus dengan menggunakan Random Walk 4
Arah dengan menggunakan variabel-variabel berikut:

- Jumlah individu: 200
- Rasio individu terinfeksi: 5%
- Probabilitas individu bergerak: 80%
- Waktu pemulihan: 10 hari
- Ukuran ruang simulasi: 20 x 20 unit
- Individu yang terkena infeksi dibedakan warnanya pada simulasi

PEMODELAN

Random Walk

Random Walk adalah model simulasi objek yang bergerak secara random dalam suatu lingkungan/matriks. Pada setiap waktu objek tersebut dapat berubah tempat/bergerak dengan syarat memenuhi beberapa kondisi dengan nilai random yang merepresentasikan angka tersebut objek bergerak dengan arah.

Model Individu

```
#individu
pos_x = []
pos_y = []
status = []
recover = []
```

Untuk memvisualisasikan individu yang terdapat pada model akan dibuat beberapa variabel yang akan mempresentasikan posisi, status, dan waktu pulih dari masing masing individu yang disimpan ke dalam list.

Untuk menentukan posisi setiap individu akan di generate secara random dengan batas bawah yaitu 0 dan batas atas adalah 20, dimana batas 0 - 20 ini adalah ukuran grid yang akan digunakan pada visualisasi random walk.

Status adalah variable list yang akan diisi dengan string untuk merepresentasikan bahwa individu tersebut 'infected', dan 'normal'. Untuk individu yang imun menggunakan status = 'normal' dengan recover = -1, untuk individu yang normal atau tidak memiliki imun dengan recover = 0.

Rasio Individu yang terinfeksi dan Waktu Pemulihan

```
[5] def Initial():
    for i in range(n_individu):
        #random position
        pos_x.append(random.randint(0, 20))
        pos_y.append(random.randint(0, 20))

        #random prob infected
        if (np.random.uniform(0, 1) <= infected_proba):
            status.append('infected')
            recover.append(recover_time)
            sum_infected.append(1)

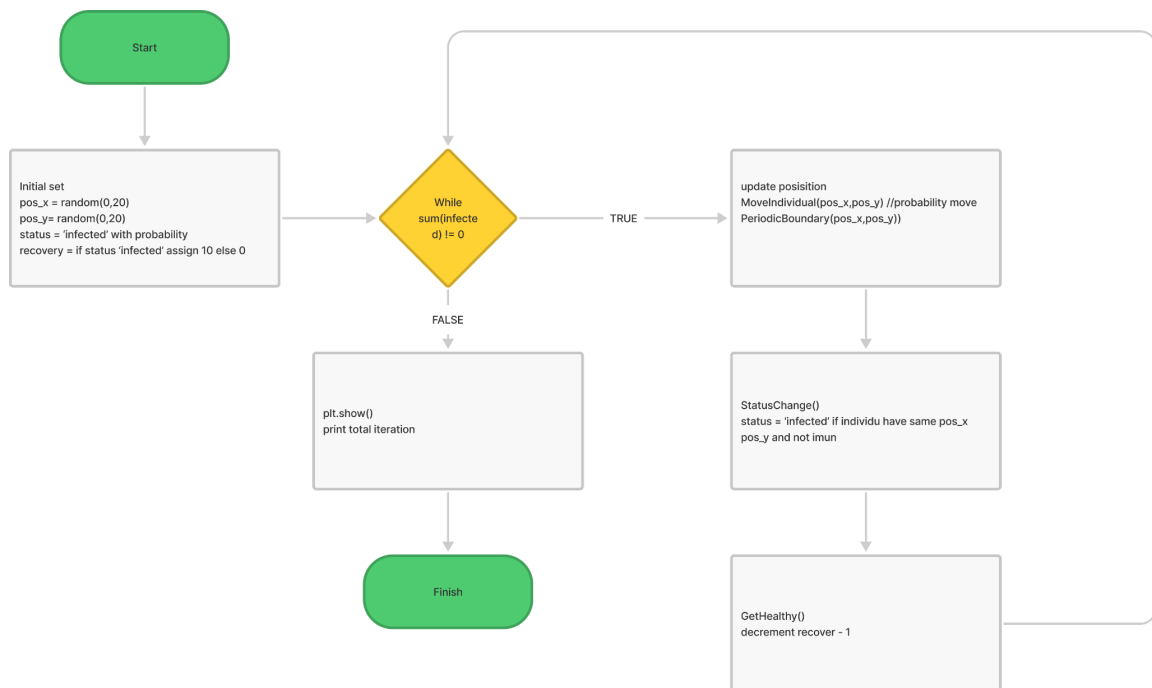
        #status normal recover 0
        else:
            status.append('normal')
            recover.append(0)
            sum_infected.append(0)
```

```
def GetHealthy():
    for i in range(n_individu):
        #defected
        if (recover[i] > 0):
            recover[i] -= 1

        #imun
        elif (recover[i] == 0 and status[i] == 'infected'):
            recover[i] = -1
            status[i] = 'normal'
```

Rasio individu inisialisasi pada saat initial state akan berstatus 'infected' secara random dengan melakukan generate uniform dari 0 - 1, dan waktu pemulihan akan inisialisasikan recover = 10 dengan status = 'infected'. Individu yang akan mendapatkan status = 'imun' adalah individu yang pernah ber status = 'infected' dan kemudian pulih.

Algoritma



DOKUMENTASI

Screenshot Program

```
#INIT
fig = plt.figure()
camera = Camera(fig)

#individu
pos_x = []
pos_y = []
status = []
recover = []

#statement
iteration = 0
sum_iteration = []
infected = []
sum_infected = []

#global
n_individu = 200
recover_time = 10

#probabilitas
infected_proba = 0.05
move_proba = 0.8

#plotting variable iteration
infected_indv_x_pos = []
infected_indv_y_pos = []
healthy_indv_x_pos = []
healthy_indv_y_pos = []
```

```
def Initial():
    for i in range(n_individu):
        #random position
        pos_x.append(random.randint(0, 20))
        pos_y.append(random.randint(0, 20))

        #random prob infected
        if (np.random.uniform(0, 1) <= infected_proba):
            status.append('infected')
            recover.append(recover_time)
            sum_infected.append(1)

        #status normal recover 0
        else:
            status.append('normal')
            recover.append(0)
            sum_infected.append(0)
```

```
def MoveIndividual(pos_x, pos_y):
    rand = random.random()
    if (np.random.uniform(0, 1) >= move_proba):
        #right
        if(rand > 0.125) and (rand <= 0.25):
            pos_x += 1

        #down
        elif(rand > 0.375) and (rand <= 0.5):
            pos_y -= 1

        #left
        elif(rand > 0.625) and (rand <= 0.75):
            pos_x -= 1

        #up
        else:
            pos_y += 1

    return pos_x, pos_y
```

```
def PeriodicBoundary(pos_x, pos_y, min_pos, max_pos):
    range_pos = [(max_pos - min_pos), (max_pos - min_pos)]

    if(pos_x > max_pos):
        pos_x -= range_pos[0]
    if(pos_x < min_pos):
        pos_x += range_pos[0]

    if(pos_y > max_pos):
        pos_y -= range_pos[1]
    if(pos_y < min_pos):
        pos_y += range_pos[1]

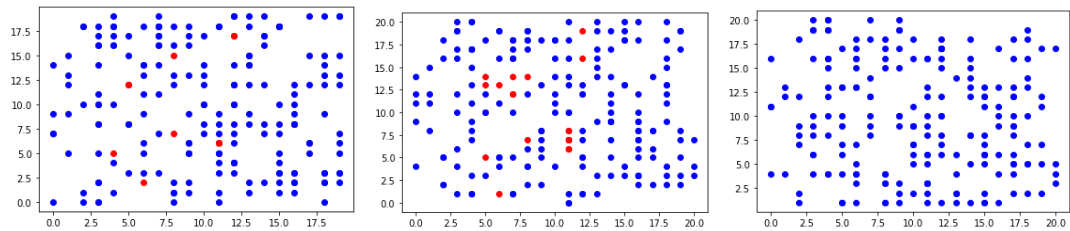
    return pos_x, pos_y
```

```
def StatusChange():
    #loop x
    for i in range(n_individu):
        #loop y
        for j in range(n_individu):
            #individu tidak sama
            if (i != j):
                #cek posisi sama dan imun
                if ((pos_x[i] == pos_x[j]) and (pos_y[i] == pos_y[j])) and ((recover[i] != -1) and (recover[j] != -1)):
                    #infected
                    if (status[i] == 'infected') and (status[j] == 'normal'):
                        status[j] = 'infected'
                        recover[j] = recover_time
                    #infected
                    elif (status[i] == 'normal') and (status[j] == 'infected'):
                        status[i] = 'infected'
                        recover[i] = recover_time
```

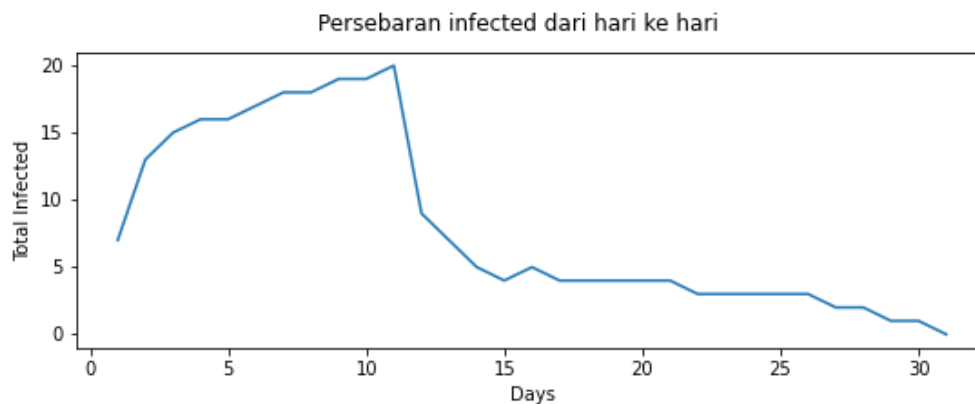
```
] def GetHealthy():
    for i in range(n_individu):
        #defected
        if (recover[i] > 0):
            recover[i] -= 1

        #imun
        elif (recover[i] == 0 and status[i] == 'infected'):
            recover[i] = -1
            status[i] = 'normal'
```

Hasil



Individu direpresentasikan dengan titik-titik berwarna, titik berwarna biru adalah individu yang memiliki status = 'normal', dan titik yang berwarna merah adalah individu yang berstatus = 'infected'. Persebaran virus terjadi karena individu saling berdekatan dan tidak memiliki imun, dan individu yang sembuh akan menjadi imun dan tidak akan terjangkit virus di hari-hari berikutnya.



Dari simulasi yang sudah dilakukan didapat bahwa terjadi lonjakan pada hari ke-12 dimana individu yang berstatus infected mencapai < 20 individu yang terjangkit virus, tetapi setelah lonjakan tersebut terdapat penurunan angka total infected yang terus menurun hingga 0 yang terjadi pada hari ke-31 semua individu dinyatakan sembuh dari virus.

Video Simulasi: <https://youtu.be/XTi5NBcrA0I>

Source Code:

<https://colab.research.google.com/drive/1alidWW89ztOFKtXZtkLT3G9k8N4X0CSM?usp=sharing>