

LAPORAN PROYEK AKHIR
MA2151 SIMULASI DAN KOMPUTASI MATEMATIKA
SEMESTER I - 2020/2021
SIMULASI SELULAR AUTOMATA:
PENYEBARAN PENYAKIT MENULAR *STOMACH VIRUS*

Disusun oleh :

- | | |
|------------------------------|----------|
| 1. Sulaiman | 10119049 |
| 2. Charlie RE. | 10119051 |
| 3. Tasha Meiryska Pohan | 10119115 |
| 4. Hieronimus Jevon Valerian | 10819017 |
| 5. Nayla Frisly Mahadewi | 10819027 |



PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
INSTITUT TEKNOLOGI BANDUNG
2020

Daftar Isi

Daftar Isi	2
BAB I Analisis Masalah	3
BAB II Rancangan Model	4
BAB III Implementasi Model	6
BAB IV Diskusi Hasil	12
I. Hasil Visualisasi	12
II. Analisis Faktor-faktor	13
III. Analisis Indikator	16
IV. Kesimpulan	19

BAB I Analisis Masalah

Pada sebuah dunia hipotetikal, suatu virus bernama *Stomach Virus* menyebar di masyarakat sekitar. Kelompok kami akan membantu mensimulasikan penyebaran virus ini serta melakukan beberapa analisis. *Stomach Virus* merupakan salah satu jenis virus yang meresahkan kehidupan masyarakat sekitar. Virus ini menyebabkan sakit pada bagian perut. Pada situasi tersebut, terdapat tiga jenis individu yang hidup di lingkungan tersebut, yaitu *Susceptible* yang merupakan jenis dari individu yang rentan terhadap virus tersebut, *Infectious* yang merupakan jenis individu yang telah terinfeksi dan dapat menularkan, serta *Immune* yang merupakan jenis individu yang telah sembuh dan tahan terhadap virus untuk jangka waktu tertentu.

Terdapat durasi waktu untuk fase *Infectious* dan *Immune*. Individu *Infectious* hanya akan berlangsung selama dua hari terhitung sejak hari mereka terinfeksi, sedangkan individu *Immune* akan kebal dari virus dalam rentang waktu lima hari terhitung sejak mereka sembuh. Setelah lima hari berlalu, individu *Immune* akan kembali menjadi *Susceptible* dan berpeluang terinfeksi kembali. Siklus ini akan berulang secara berkala. Pada analisis kali ini, model yang diterapkan termasuk ke dalam jenis model deterministik, karena pada simulasi kali ini jika seseorang berada di dekat/bersebelahan dengan orang yang *Infectious* maka orang tersebut akan terinfeksi secara pasti. Selain itu, setelah dua hari masa infeksi, individu tersebut dipastikan akan *Immune*. Sehingga, simulasi tidak melibatkan unsur peluang pada proses terinfeksi ataupun kesembuhannya.

BAB II Rancangan Model

a. Pengumpulan data

- Peluang yang akan digunakan pada *probS* dan *probi*
 $probSusceptibleList = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$
 $probInfectiousList = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$

b. Asumsi yang digunakan

- Terdapat lingkungan berupa matriks berukuran 25 x 25.
- Dalam populasi, tiap individu dapat menjadi *susceptible*, *infectious*, atau *immune* terhadap *stomach virus*.
- Menggunakan aturan ketetanggaan Von Neumann (*North, South, West, East*), dimana jika seorang individu rentan dan tetangganya terinfeksi, individu tersebut menjadi terinfeksi.
- Infeksi berlangsung selama dua hari dan imunitas berlangsung hanya lima hari sebelum individu tersebut menjadi rentan kembali.
- Setelah terinfeksi selama 2 hari, individu akan menjadi imun di hari ke-3 sampai hari ke-7 untuk kemudian menjadi rentan kembali.
- Asumsi individu berada di titik grid/kisi/sel. Pada simulasi, nilai pada sebuah titik grid dapat menjadi salah satu dari berikut:
 - 0 : individu susceptible
 - 1, 2 : individu infectious, dimana nilai menunjukkan hari terinfeksi.
 - 3, 4, 5, 6, 7 : individu immune, dimana hari imunitas kesekian sama dengan nilai pada sel dikurangi 2. Misalnya, pada hari ke-1 imunitas, nilai selnya adalah 3.
- Digunakan fungsi kondisi batas yaitu *Reflecting Boundary Condition*.
- Akan dibuat model animasi dengan aturan sebagai berikut :
 1. Individu *susceptible* berwarna hijau pekat.
 2. Individu *infectious* berwarna biru pekat di hari pertama dan berwarna biru cerah di hari kedua.
 3. Individu *immune* berwarna merah pekat pada hari pertama dan warna merah yang lebih cerah di hari-hari berikutnya.

c. Variabel input dan output

Input

- Peluang seseorang menjadi rentan (*susceptible*) : *probS*
- Peluang seseorang terinfeksi (*infectious*) : *probl*
- Banyaknya iterasi

Output

- Array matriks seluler automata (setiap *day*) untuk masing-masing *probS* dan *probl*
- Visualisasi proses penyebaran stomach virus untuk setiap *probS* dan *probl*

d. Desain simulasi

Kita memiliki lingkungan berupa matriks 25 x 25 dengan populasi berupa 3 jenis individu yaitu individu *susceptible*, individu *infectious*, dan individu *immune*. Matriks ini akan dikenakan iterasi sebanyak 50 hari dengan fungsi *init_grid*. Individu yang terinfeksi akan berdistribusi seragam antara hari ke-1 dan hari ke-2 terinfeksi. Dalam inisialisasi, individu yang imun akan berdistribusi seragam dengan nilai antara 3 sampai 7, atau antara hari ke-1 sampai hari ke-5 imunitas. Kemudian, dengan menggunakan kondisi batas Reflecting Boundary Condition, akan dibangun *matrix_ext* yang merupakan grid yang diperpanjang satu sel disetiap sisinya dengan fungsi *reflect_bound*. Selanjutnya, bangun fungsi *spread(matrix_ext)* untuk meng-update *matrix_ext*. Akan dilakukan looping untuk memperbaharui kondisi/status setiap individu pada grid sampai $t = 50$ **atau** sampai waktu di mana sudah semua sel bernilai *susceptible*.

Fungsi *spread* menggunakan ketetanggaan aturan ketetanggaan Von Neumann (*North*, *South*, *West*, *East*), dimana jika seorang individu rentan (*susceptible*) dan tetangganya terinfeksi, individu tersebut menjadi terinfeksi. Fungsi ini mengembalikan grid dalam simulasi penyebaran penyakit yang kondisi sel-selnya telah diubah sesuai aturan. Bila suatu sel sebelumnya bernilai 0 (*susceptible*) dan memiliki tetangga yang *infectious* (1 atau 2) maka nilainya diubah menjadi 1 (*infectious*). Sedangkan apabila sebelumnya nilainya dari 1 sampai 6 (*infectious* atau *immune*) maka naikan satu, dan bila sebelumnya nilainya 7 (hari terakhir *immune*) maka ubah nilainya menjadi 0.

BAB III Implementasi Model

Menggunakan algoritma yang telah dikembangkan, implementasi pada program bahasa *python* dapat dilihat di:

[JUPYTER NOTEBOOK SIMULASI](#)

1. Simpan konstanta berikut sebagai konstanta global:
 - ROWS = 25
 - COLS = 25
2. Pertama, kita deklarasi fungsi *init_grid* untuk menginisialisasi sistem dari lingkungan yang akan kita gunakan pada simulasi

init_grid(r, c, probS, probI)

Fungsi untuk mengembalikan matriks berukuran $r \times c$ dengan setiap elemennya berisi antara *SUSCEPTIBLE*, *INFECTIOUS*, *IMMUNE* dengan probS peluang seseorang menjadi *Susceptible* dan probI peluang seseorang menjadi *Infectious*.

Pre:

- r is the row size of the grid and c is the column size of the grid where r, c are positive
- *probS* peluang seseorang menjadi *Susceptible*
- *probI* peluang seseorang menjadi *Infectious*
- *matrix* berukuran $r \times c$ dengan setiap elemen diberi nilai awal 0

Post:

- Matriks dengan setiap elemen berisi satu dari tiga tipe manusia (*SUSCEPTIBLE*, *INFECTIOUS*, *IMMUNE*) akan dikembalikan.

Algorithm:

```
for i from 1 to r do
  for j from 1 to c do
    rand_1 := generate random number in range [0,1)
    rand_2 := generate random number in range [0,1)
```

```

if rand_1 < probaS then
    matrix[i, j] = 0           //susceptible individual

else if rand_2 < probaI then
    rand_3 := generate random number between 1 and 2
    matrix[i, j] := rand_3    //infectious individual

else
    rand_3 := generate random number among [3,4,5,6,7]
    matrix[i, j] := rand_3    //immune individual

return matrix

```

3. Kedua, kita berikan kondisi batas untuk lingkungannya dengan mendeklarasi fungsi *reflect_bound*.

reflect_bound(matrix)

Fungsi untuk mengembalikan grid yang telah diperluas di sisi-sisinya dengan menggunakan prinsip *reflecting boundary condition*.

Pre:

- *matrix* berukuran *r* x *c* dengan setiap elemen berisi nilai antara *SUSCEPTIBLE*, *INFECTIOUS*, *IMMUNE*.

Post:

- *matrix_ext* dengan setiap grid diperluas pada masing-masing sisinya dengan *reflecting boundary condition*.

Algorithm:

```

r := ROWS
c := COLS

```

```

matrix_ext := declare matrix with extended one cell in
each direction from "matrix" with each element contains
zero value.

```

```

for i from 2 to r+1 do
  for j from 2 to c+1 do
    matrix_ext[i, j] := matrix[i-1, j-1]

for j from 2 to c+1 do
  matrix_ext[1, j] := matrix_ext[2, j]
  matrix_ext[r+2, j] := matrix_ext[r+1, j]

for k from 1 to r+2 do
  matrix_ext[k, 1] := matrix_ext[k, 2]
  matrix_ext[k, c+2] := matrix_ext[k, c+1]

return matrix_ext

```

4. Ketiga, kita deklarasi fungsi *spread* yang menerapkan aturan transisi pada setiap iterasi. Sehingga ada kemungkinan kondisi setiap elemen akan berubah setiap iterasi berlangsung.

spread(matrix_ext)

Fungsi untuk mengembalikan matriks hasil simulasi penyebaran penyakit dengan merubah kondisi pada setiap elemen matriks jika diperlukan berdasarkan aturan transisi.

Pre:

- *matrix_ext* dengan setiap grid diperluas pada masing-masing sisinya dengan *reflecting boundary condition*.

Post:

- *matrix_ext* dengan pembaharuan kondisi pada setiap elemen berdasarkan aturan transisi yang berlaku.

Algorithm:

```

r := ROWS + 2
c := COLS + 2

```



```

for i from 2 to r-1 do
  for j from 2 to c-1 do
    if matrix_ext[i, j] == 0 then
      neighbor := [matrix_ext[i-1, j], matrix_ext[i, j+1],
                  matrix_ext[i+1, j], matrix_ext[i, j-1]]

      if (1 in neighbor) OR (2 in neighbor) then
        matrix[i, j] := 1 //first day become infectious

    else if 1 <= matrix_ext[i, j] <= 6 do
      matrix_ext[i, j] := matrix_ext[i, j] + 1

    else if matrix_ext[i, j] == 7 then //last day as an
immune individual
      matrix_ext[i, j] := 0 //back into susceptible
individual

```

5. Keempat, kita akan membuat algoritma *simulasi* terkait “Persebaran *Stomach Virus*”

Simulation Algorithm of The Spread of Stomach Virus

Initialize:

- *probSusceptibleList* array berisi seluruh *probS* yang akan digunakan pada simulasi.
- *probInfectiousList* array berisi seluruh *probI* yang akan digunakan pada simulasi.
- *simulation_length*
- *list_spread* adalah matriks berukuran 9 x 9 dengan setiap elemen berisi hasil pada setiap iterasi untuk masing-masing *probS* dan *probI*.
- *rows, cols* ukuran baris dan kolom pada *matrix*

Algorithm:

```

for i from 1 to length(probSusceptibleList) do
  for j from 1 to length(probInfectiousList) do

```

```

    probSusceptible := probSusceptibleList[i]
    probInfectious := probInfectiousList[j]

    //Inisialisasi Sistem
        spread_mat := init_grid(rows, cols,
probSusceptible, probInfectious)
    add the spread_mat onto list_spread[i, j]

    stop := FALSE
    //Proses Penyebaran Virus
    k := 0

    while (k < simulation_length) AND NOT stop do
        spread_mat_ext := list of
reflect_bound(spread_mat)
        spread(spread_mat_ext)
        add spread_mat_ext onto list_spread[i, j]

        //Update Kondisi untuk Setiap Elemen di
spread_mat
        stop := TRUE
        for l from 1 to rows do
            for m from 1 to cols do
                spread_mat[l, m] := spread_mat[l+1, m+1]

            if stop then
                if spread_mat[l, m] then //Jika belum
//semua SUSCEPTIBLE, loop
berlanjut
                    stop := FALSE

        k := k + 1

```

6. Kelima, kita buat fungsi ***showanim*** untuk menganimasikan proses pada masing-masing *probS* dan *probl*.

showanim(probS, probl)

Fungsi yang menganimasikan proses penyebaran penyakit untuk masing-masing *probS* dan *probI*

Pre:

- *probS* peluang seseorang menjadi *SUSCEPTIBLE*
- *probI* peluang seseorang menjadi *INFECTIOUS*

Post:

- Animasi terkait proses penyebaran virus untuk masing-masing *probS* dan *probI*

Algorithm:

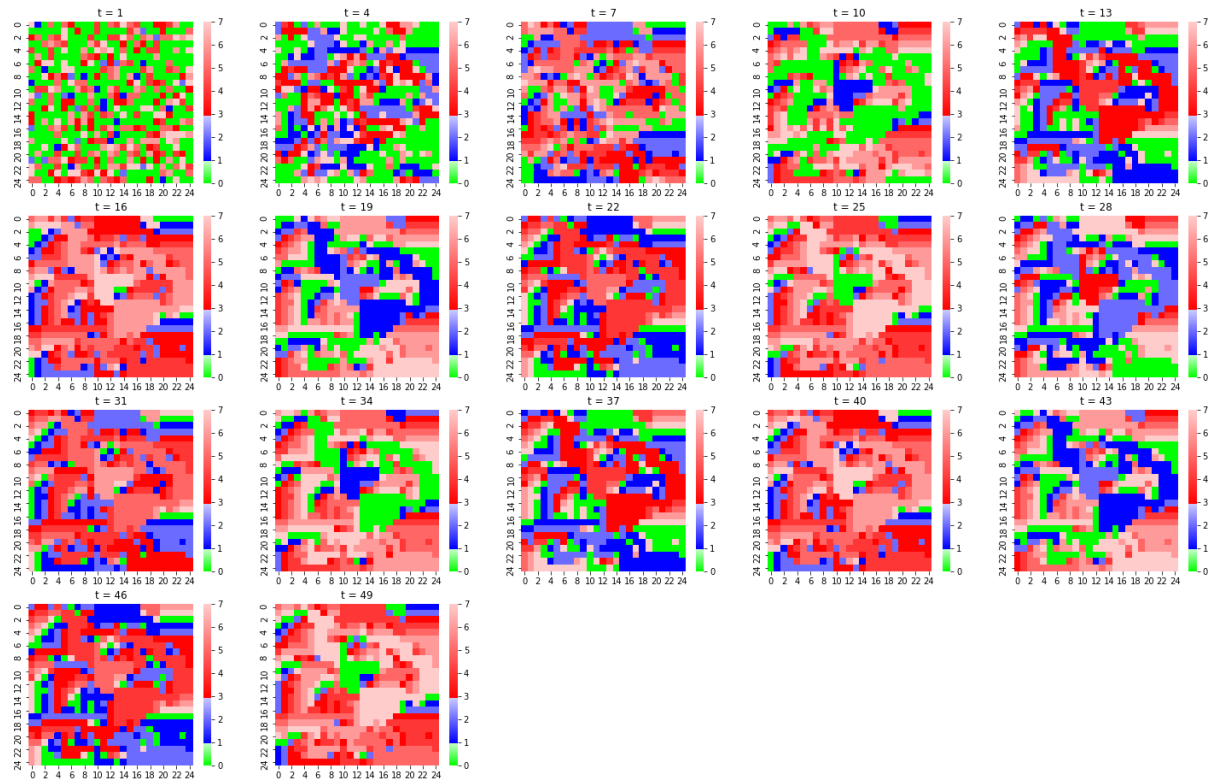
```
x := INT(probS / 0.1) - 1
y := INT(probI / 0.1) - 1
t := length(list_spread[x, y])
```

```
animate all the conditions in each element for each
iteration process that lasts t times
```

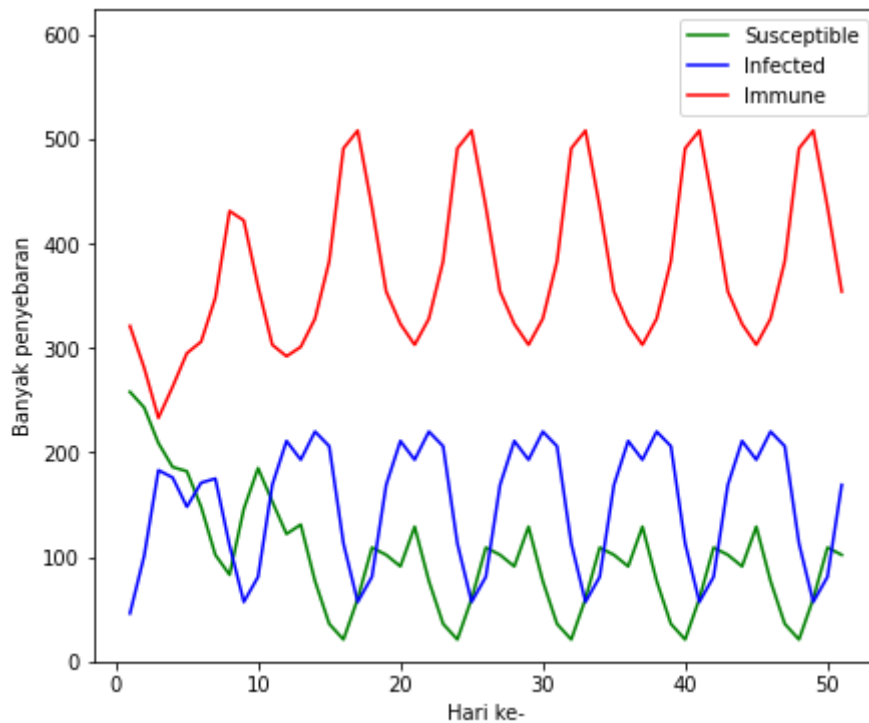
BAB IV Diskusi Hasil

I. Hasil Visualisasi

Hasil selular automata yang telah dianimasikan dapat dilihat pada **LAMPIRAN_ANIMASI**. Untuk mendapat sedikit gambaran, berikut ini adalah animasi untuk $probSusceptible = 0.5$ dan $probInfectious = 0.1$ yang telah dipisahkan dalam masing-masing frame.



Visualisasi berupa grafik juga kami sertakan dalam **LAMPIRAN_GRAFIK**. Untuk $probSusceptible$ dan $probInfectious$ bernilai (0.4, 0.1), berikut grafik penyebaran kasus (kumulatif) untuk tiap harinya.



II. Analisis Faktor-faktor

Dari semua hasil simulasi yang kami dapatkan, dapat diamati bahwa penyebaran *Stomach Virus* memiliki pola penyelesaian yang unik. Terdapat dua jenis pola selesainya penyebaran *Stomach Virus*:

- Semua sel telah menjadi Susceptible pada hari ke-t
- Pola penyebaran berulang-ulang (periodik) dan tidak pernah selesai (selalu ada yang masih *Infectious*)

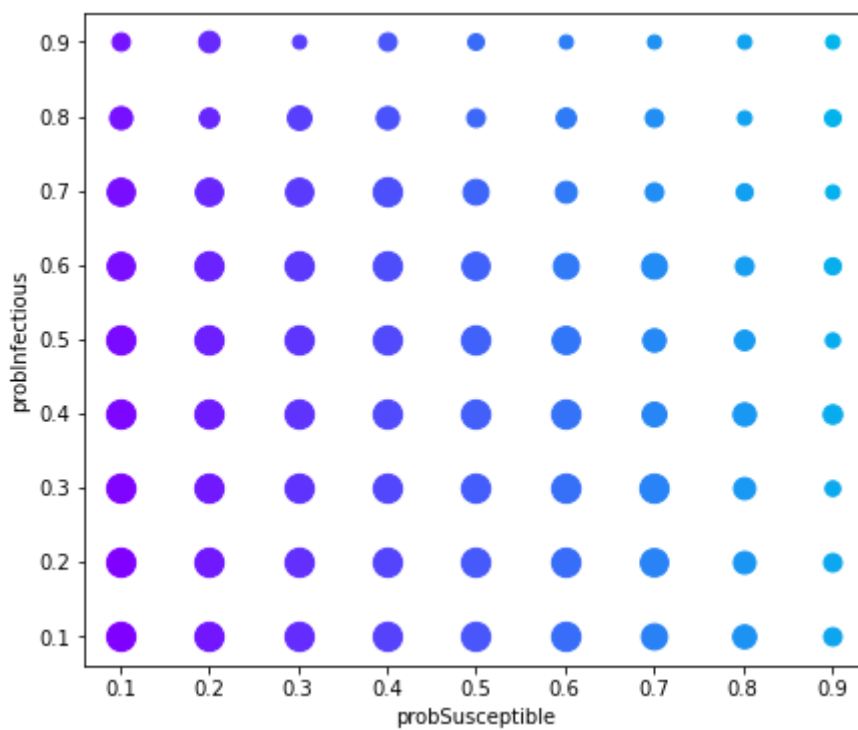
Untuk itu, kami mencoba menguji apakah terdapat korelasi antara *probSusceptible* dan *probInfectious* terhadap pola penyelesaian ini.

Pertama-tama, diuji terlebih dahulu korelasi antara probabilitas-probabilitas tersebut terhadap total hari penyebaran (sampai penyebaran selesai). Untuk keakuratan yang lebih tinggi, setiap simulasi (dengan *probSusceptible* dan *probInfectious* tertentu) kami lakukan **sepuluh kali** lalu diambil rata-ratanya.

Tabel yang lengkap dapat dilihat pada **LAMPIRAN_TABEL1**. Kolom "*size*" memiliki arti "banyaknya hari penyebaran". Data-data yang pertama ditunjukkan pada tabel di bawah ini.

	probS	probI	size
0	0.1	0.1	51.0
1	0.1	0.2	51.0
2	0.1	0.3	51.0
3	0.1	0.4	51.0
4	0.1	0.5	51.0
5	0.1	0.6	47.2
6	0.1	0.7	47.2
7	0.1	0.8	31.5
8	0.1	0.9	19.3
9	0.2	0.1	51.0
10	0.2	0.2	51.0
11	0.2	0.3	51.0
12	0.2	0.4	51.0
13	0.2	0.5	51.0
14	0.2	0.6	51.0

Plot dari data tersebut ditunjukkan pada grafik berikut.



Perlu dicatat bahwa *bubble* dengan ukuran terbesar ($size = 51$) menandakan bahwa penyebaran tersebut *tidak berhenti* sampai panjang simulasi 50, yang belum dapat kita simpulkan darinya penyebaran itu akan berhenti atau periodik. Oleh karena itu, untuk data ini kita perhatikan daerah di plot dengan $probSusceptible > 0.5$ dan $probInfectious > 0.5$

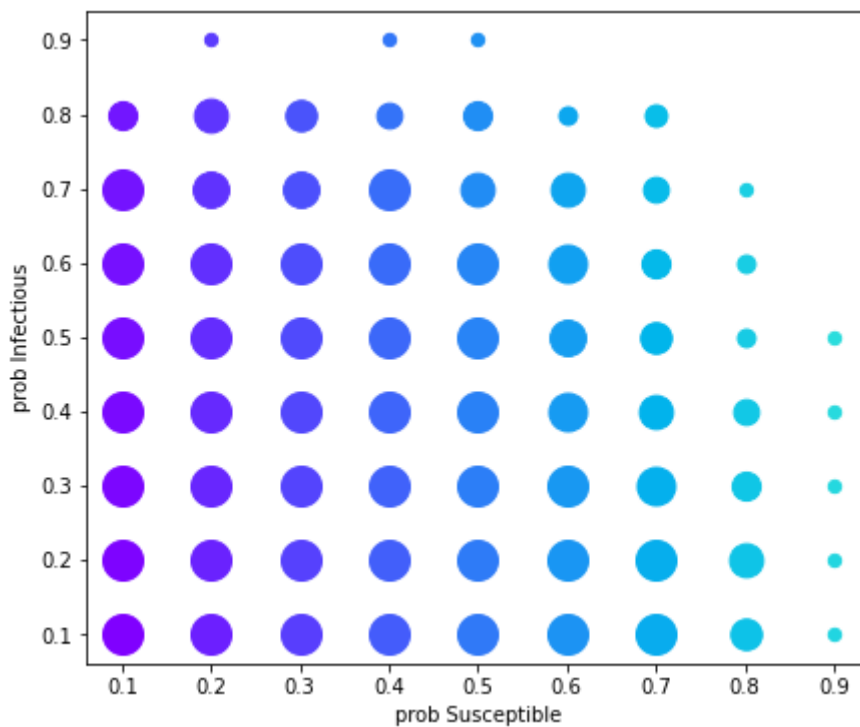
Pola penyebaran ini memiliki tren, **semakin besar *probSusceptible* atau *probInfectious*, maka total hari penyebaran semakin kecil.**

Pengamatan kedua yang kami lakukan ialah menguji keperiodikan penyebaran. Untuk itu, panjang simulasi perlu ditingkatkan dari 50 (seperti yang dinyatakan pada *desain simulasi*) menjadi 200. Diasumsikan bahwa pola pasti periodik apabila penyebaran masih belum selesai saat simulasi berakhir. Seperti sebelumnya, di sini digunakan simulasi berulang sebanyak masing-masing sepuluh kali.

Data yang lengkap terdapat pada **LAMPIRAN_TABEL2**. Data-data pertama darinya ditunjukkan di bawah ini.

	probS	probI	periodic_count (persen)
0	0.1	0.1	100
1	0.1	0.2	100
2	0.1	0.3	100
3	0.1	0.4	100
4	0.1	0.5	100
5	0.1	0.6	100
6	0.1	0.7	100
7	0.1	0.8	50
8	0.1	0.9	0
9	0.2	0.1	100
10	0.2	0.2	100
11	0.2	0.3	100
12	0.2	0.4	100
13	0.2	0.5	100
14	0.2	0.6	100

Representasi grafiknya ditunjukkan pada berikut.



Pengamatan ini penting sebab kita dapat melihat bahwa **probabilitas yang besar tidak menjamin terselesaikannya penyebaran**. Selain itu, untuk daerah yang dibatasi $probSusceptible \leq 0.5$ dan $probInfectious \leq 0.5$ **pola penyebarannya selalu periodik** (100% untuk setiap percobaan). Seperti yang kita lihat sebelumnya, data ini tidak bertentangan sebab pengamatan ini juga menunjukkan tren, semakin besar $probSusceptible$ atau $probInfectious$ maka pola periodik semakin jarang terjadi, yang artinya semakin jarang pula terdapat simulasi yang lama penyebarannya besar.

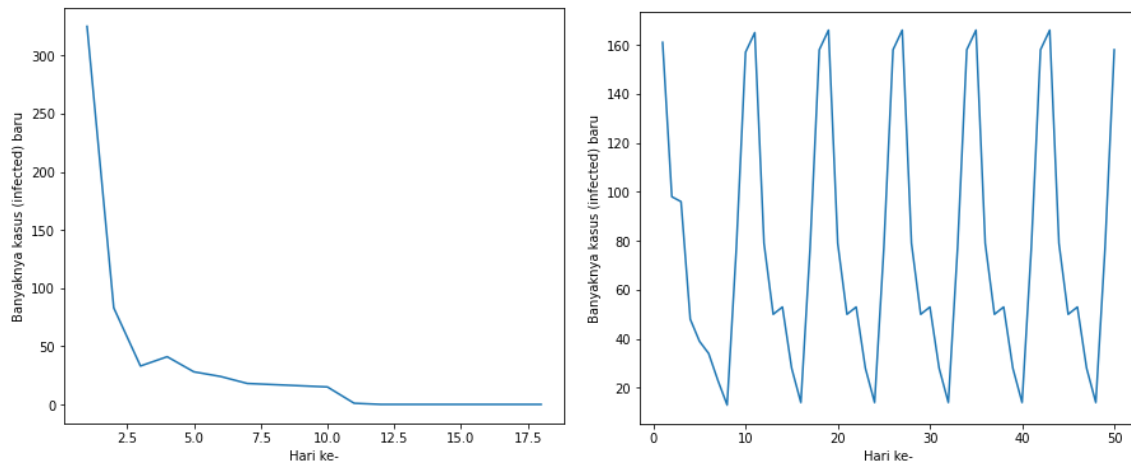
III. Analisis Indikator

Dari hasil analisis bagian sebelumnya, muncul pertanyaan yaitu seandainya kita berada dalam simulasi tersebut, maka *apa yang dapat kita lakukan untuk memprediksi apakah penyebaran virus akan berakhir?*

Periodik atau tidaknya suatu penyebaran dapat ditunjukkan oleh suatu indikator. Setelah menggali data lebih dalam, kami mendapatkan bahwa **lonjakan terbesar dari *get_infected*** dapat menjadi indikator yang baik.

Sebagai tambahan dari *susceptible*, *infectious*, dan *immune* sekarang kita miliki *get_infected*. Didefinisikan *get_infected* untuk sebuah frame (*day* tertentu) selular automata bernilai sama dengan banyaknya sel yang bernilai 1 (*infectious*) **dan** pada *day-1* bernilai 0 (*susceptible*). Dengan kata lain, array *get_infected* berisi penambahan **kasus baru** setiap harinya.

Berikut contoh representasi grafik dari *get_infected* untuk probS , $\text{probI} = (0.9, 0.1)$ dan $(0.5, 0.3)$.



Dari nilai-nilai *get_infected* ini, kita ambil **lonjakan terbesarnya (*biggest jump*)** sebagai indikator. Secara algoritmis, lonjakan terbesar dari sebuah selular automata (hingga seluruh *day*) ialah $\max(\text{get_infected}[i+1] - \text{get_infected}[i]) / \text{populasi}$. Untuk simulasi ini digunakan $\text{populasi} = 25 \times 25 = 625$.

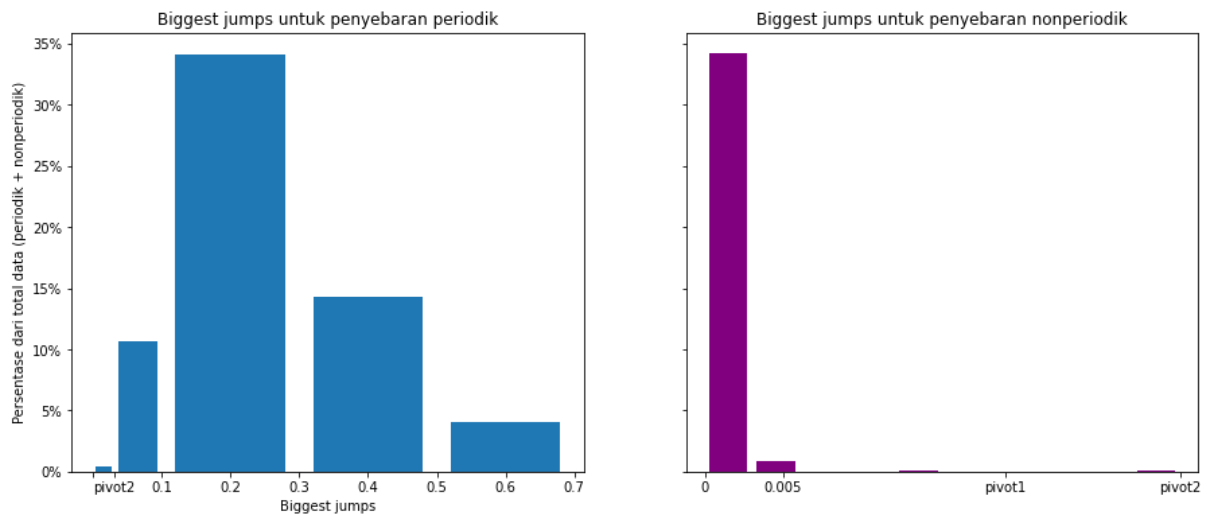
Berikut hasil pengamatan biggest jumps (vs keperiodikan) untuk *probSusceptible* dan *probInfectious* $[0.1, 0.2, \dots, 0.9] \times [0.1, 0.2, \dots, 0.9]$ dengan masing-masing diulangi sebanyak 10 kali (tidak dirata-ratakan). Hasil lengkap dapat dilihat di **LAMPIRAN_TABEL3**.

	probS	probI	biggest jumps	is periodic?
0	0.1	0.1	0.1824	True
1	0.1	0.1	0.2112	True
2	0.1	0.1	0.1328	True
3	0.1	0.1	0.2672	True
4	0.1	0.1	0.1968	True
5	0.1	0.1	0.2672	True
6	0.1	0.1	0.1120	True
7	0.1	0.1	0.3200	True
8	0.1	0.1	0.2896	True
9	0.1	0.1	0.1600	True
10	0.1	0.2	0.2016	True
11	0.1	0.2	0.1216	True
12	0.1	0.2	0.1104	True
13	0.1	0.2	0.0720	True
14	0.1	0.2	0.1408	True
15	0.1	0.2	0.0672	True
16	0.1	0.2	0.0656	True
17	0.1	0.2	0.1344	True
18	0.1	0.2	0.0864	True
19	0.1	0.2	0.0512	True

Nilai-nilai *biggest jumps* ini kami duga memiliki hubungan kuat dengan keperiodikan. Untuk itu, kita amati beberapa sifat-sifatnya:

- Banyaknya semua data adalah 810
- Sebanyak 35.30% darinya nonperiodik
- Nilai terkecil dari *biggest jumps* untuk data yang periodik ialah 0.0192 (pivot1)
- Nilai terbesar dari *biggest jumps* untuk data nonperiodik ialah 0.0034 (pivot2)
- Sebanyak 99% dari seluruh data yang periodik memiliki biggest jump > pivot2
- Sebanyak 99% dari seluruh data nonperiodik memiliki biggest jump < pivot1

Histogram penyebaran nilai *biggest jumps* ini ditunjukkan pada gambar berikut.



Secara signifikan kita dapat melihat bahwa interval nilai *biggest jumps* untuk data yang periodik dan data yang nonperiodik hampir tidak beririsan. Terdapat sejenis polarisasi nilai *biggest jumps* antara kedua jenis data. Polarisasi ini kuat sekali, sebab kita menemukan 99% data periodik ditemukan pada daerah *biggest jumps* $>$ pivot2 dan 99% data nonperiodik ditemukan pada daerah *biggest jumps* $<$ pivot1, dengan fakta bahwa pivot1 dan pivot2 hanya berselisih 0.01.

Berdasarkan informasi tersebut, dapat kita tarik pemahaman bahwa **seandainya kita berada di simulasi itu, dan pada suatu hari banyaknya kasus baru melonjak sebanyak 5% maka hampir dapat dipastikan penyebaran itu tidak akan pernah berakhir.**

IV. Kesimpulan

Dari diskusi dan analisis data-data yang telah disajikan pada dua bagian sebelumnya, dapat kita tarik tiga hal:

- 1) Meskipun *setelah* inisialisasi model bersifat deterministik, tetap ditemukan variasi-variasi dari hasilnya, sehingga model masih bersifat stokastik *menurut* inisialisasi model
- 2) Hasil simulasi dapat dikategorikan menjadi penyebaran periodik dan penyebaran nonperiodik. Bila penyebaran periodik, maka akan selalu ada individu yang *infectious* sehingga penyebaran virus tidak pernah berakhir. Untuk penyebaran nonperiodik, terdapat suatu waktu di mana semua individu telah kembali *susceptible*. Probabilitas pada inisialisasi (*probSusceptible* dan *probInfectious*) mempengaruhi keperiodikan ini dalam cara:
 - Semakin kecil *probSusceptible* atau *probInfectious*, maka peluang terjadi dinya pola periodik semakin kecil. Bila *probSusceptible* \leq 0.5 dan *probInfectious* \leq 0.5, hampir dapat dipastikan akan terjadi pola periodik ($>99\%$).

- Semakin besar *probSusceptible* atau *probInfectious*, maka lama penyebaran cenderung semakin kecil. Oleh karena itu, semakin jarang pula ditemukan pola periodik.

3) Seandainya kita berada dalam simulasi tersebut, kita dapat memanfaatkan indikator *biggest jump* untuk memprediksi apakah penyebaran akan berakhir (nonperiodik). Bila suatu hari ditemukan lonjakan *get_infected* sebesar 5% dari populasi, maka hampir dapat dipastikan (>99%) penyebaran tersebut tidak akan pernah berakhir (periodik).