

**ANALISIS PENYEBARAN PENYAKIT MENULAR
MENGUNAKAN RANDOM WALK DARI PARTIKEL
GRANULAR SEBAGAI MODEL BERBASIS AGEN**

Laporan Tugas Akhir

**Diajukan sebagai salah satu syarat kelulusan tahap sarjana di Program Studi
Fisika ITB**

oleh

I Wayan Nadiantara

NIM : 10214071

Program Studi Fisika



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI BANDUNG**

2020

Tugas Akhir

Analisis Penyebaran Penyakit Menular Menggunakan Random Walk dari Partikel Granular Sebagai Model Berbasis Agen.

Oleh:

I Wayan Nadiantara
10214071

Program Studi Fisika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Institut Teknologi Bandung

Tugas Akhir ini dipresentasikan pada:

15 Juni 2020

Telah diperiksa, disetujui, dan disahkan oleh Pembimbing,

Dr. ret. nat. Sparisoma Viridi

197312011999031002

ABSTRAK

ANALISIS PENYEBARAN PENYAKIT MENULAR MENGUNAKAN RANDOM WALK DARI PARTIKEL GRANULAR SEBAGAI MODEL BERBASIS AGEN

oleh

I Wayan Nadiantara

NIM : 10214071

Merebaknya suatu wabah merupakan permasalahan besar yang menghantui peradaban manusia dari masa ke masa. Dalam menghadapi suatu wabah, manusia tidak hanya memerlukan pengobatan dan vaksinasi yang efektif. Meskipun suatu wabah relatif tidak mematikan, jika penyebarannya relatif cepat, wabah tersebut dapat membuat kepanikan masal yang berujung pada terganggunya stabilitas sosial, politik, dan ekonomi dari suatu daerah. Sistem pelayanan kesehatanpun dapat terganggu jika penambahan pasien dalam jumlah besar terjadi pada waktu yang singkat. Pertumbuhan populasi terjangkit wabah yang secara alami bersifat eksponensial akan sangat efektif ditekan jika informasi terkait wabah tersebut diketahui sedini mungkin. Oleh karena itu informasi dengan resolusi yang tinggi terkait bagaimana suatu wabah dapat menyebar akan sangat membantu dalam pencegahan ataupun mitigasi dari mewabahnya suatu patogen.

Pada tugas akhir ini, penulis menggunakan model berbasis agen yang secara umum memiliki resolusi lebih tinggi dari model dengan kerangka lainnya untuk mendeskripsikan bagaimana suatu wabah dapat menyebar. Model ini dibangun berdasarkan gerakan acak dari sistem partikel granular. Random walk merupakan proses stokastik yang jamak digunakan dalam mendeskripsikan gerakan suatu organisme secara matematis, termasuk dalam memodelkan pergerakan atau mobilitas manusia yang mana sangat mempengaruhi bagaimana suatu wabah dapat menyebar.

Kata kunci : *Wabah, Random walk, model berbasis agen, granular.*

ABSTRACT

ANALYSIS OF INFECTIOUS DISEASE TRANSMISSION USING RANDOM WALK ON GRANULAR PARTICLE AS AN AGENT-BASED MODEL

by

I Wayan Nadiantara

NIM : 10214071

The outbreak of infectious disease is a major problem that haunts human civilization from milenia to milenia. In carrying out an outbreak, humans not only need effective treatment and vaccination. Even though an outbreak is relatively non-lethal, if it could spread in relatively rapid time, it could create a mass panic that results in disruption of the social, political, and economic stability of an area. The health care system also could be disrupted if large numbers of patients are coming in a short time. The growth of outbreaks that are naturally exponential will be very effectively suppressed if any pieces of information related to the outbreak are known as early as possible. Therefore information with high resolution related to how an outbreak could spread will be very helpful in preventing or mitigating the outbreak of a pathogen.

In this final project, an agent-based model simulation that generally has a higher resolution than models with other frameworks will be used to describe how an outbreak can spread. This model is built based on random walks of the granular particle system. Random walk is a stochastic process that is commonly used in describing the movements of an organism mathematically, including in modeling human movements or mobility which greatly influences how an outbreak can spread.

Keyword : Epidemic, Random Walk, Agent-based model, granular.

KATA PENGANTAR

Terima kasih saya ucapkan kepada semua pihak yang turut membantu saya dalam menyelesaikan tugas akhir yang berjudul *Analisis Penyebaran Penyakit Menular Menggunakan Random Walk dari Partikel Granular Sebagai Model Berbasis Agen*. Dalam penyelesaian tugas akhir ini, terdapat banyak pihak yang telah senantiasa memberikan kontribusi baik secara langsung maupun tak langsung, moral maupun materil, sehingga pada kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Ni Nengah Nartini dan Ni Made Nadya Prabawanty selaku ibu dan adik dari penulis, yang senantiasa memberi dukungan kepada penulis.
2. Dr. ret. nat. Sparisoma Viridi selaku dosen pembimbing yang selalu memberikan dukungan, saran yang sangat membangun, dan banyak pengetahuan selama pengerjaan tugas akhir ini.
3. Dosen penguji, Dr.Eng. Sidik Permana S.Si.,M.Eng. dan Dr. Fatimah Arofiati Noor S.Si.,M.Si., yang telah memberikan saran dan kritik pada tugas akhir ini.
4. Kepala Prodi dan seluruh dosen pengajar di Prodi Fisika yang telah memberikan ilmu dan dukungan selama masa perkuliahan.
5. Nicholaus Kevin Tanjung, Florentin Anggraeni Purnama, Darius Chandra, Gesti Rahmah, Rizky Adhitama serta teman-teman lainnya di program studi fisika yang telah memberikan banyak bantuan dan dukungan dalam penyelesaian tugas akhir ini.
6. Kepada Pak Sulaeman dan seluruh staff TU Prodi Fisika ITB yang telah membantu penulis dalam proses administratif.
7. Pihak yang secara langsung maupun tidak langsung membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa dalam menyusun tugas akhir ini masih banyak kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun agar penelitian ini menjadi lebih bermanfaat di masa yang akan datang.

Bandung, Maret 2020

Penulis

DAFTAR ISI

ABSTRAK	ii
ABSTRACT	iii
DAFTAR ISI	iv
DAFTAR GAMBAR DAN ILUSTRASI	v
DAFTAR TABEL	vi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1-2
1.3 Tujuan Penelitian	2
1.4 Ruang Lingkup Studi	2
1.5 Metode Penelitian	2-3
1.6 Sistematika Penulisan	3
BAB 2 TEORI DASAR	4
2.1 Random Walk	4
2.3.1 Random Walk dalam Satu Dimensi	5
2.3.1 Random Walk dalam Dua Dimensi	6
2.2 Partikel Granular	7
2.3 Klasifikasi Model Penyebaran Wabah	8
2.3.1 Model Kompartemen	8-9
2.3.2 Model Berbasis Agen	9
BAB 3 METODE PENELITIAN	10
3.1 Alur Kerja Model	10-12
3.2 Asumsi	12
3.3 Hasil Keluaran Simulasi dengan Variasi Nilai Variabel Sembarang	13
3.4 Analisis dan Verifikasi awal Terhadap Hasil Keluaran	19
BAB 4 HASIL	21
4.1 Pendahuluan	22
4.2 Kalibrasi	23
4.3 Validasi Eksternal	40
BAB 5 SIMPULAN DAN SARAN	46
5.1 Simpulan	46
5.2 Saran	46
DAFTAR PUSTAKA	47
LAMPIRAN	50

DAFTAR GAMBAR DAN ILUSTRASI

Gambar 2.1 Sketsa Lintasan Semut	6
Gambar 2.2 Diagram Skematik Model Kompartemen	9
Gambar 2.3 Diagram Skematik Model Berbasis Agen.....	9
Gambar 3.1 Diagram skematik dari alur kerja program.	10
Gambar 3.2a Keadaan awal program.	11
Gambar 3.2b Keadaan program sesaat setelah dijalankan.	11
Gambar 3.3a Tampilan akhir program dengan jumlah total populasi agen sebanyak 400 unit dan jumlah terinfeksi dibuat proporsional 4% dari total populasi..	12
Gambar 3.3b Tampilan akhir program dengan jumlah total populasi agen sebanyak 900 unit dan jumlah terinfeksi dibuat proporsional 4% dari total populasi..	14
Gambar 3.3c Tampilan akhir program dengan jumlah total populasi agen sebanyak 900 unit dan jumlah terinfeksi dibuat proporsional 4% dari total populasi..	14
Gambar 3.4a Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 1% dari total populasi..	15
Gambar 3.4b Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 10% dari total populasi.	15
Gambar 3.4c Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 15% dari total populasi..	16
Gambar 3.5a Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 5% dari total populasi dan waktu untuk sembuh sebesar 50 time steps..	16
Gambar 3.5b Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 5% dari total populasi dan waktu untuk sembuh sebesar 75 time steps..	17
Gambar 3.5c Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 5% dari total populasi dan waktu untuk sembuh sebesar 150 time steps..	17
Gambar 3.6a Tampilan akhir program dengan kecepatan agen maksimum 1px per time steps..	18
Gambar 3.6b Tampilan akhir program dengan kecepatan agen maksimum 2px per time steps..	19
Gambar 3.6c Tampilan akhir program dengan kecepatan agen maksimum 3px per time steps...	20
Gambar 4.1a Hasil pengolahan data terhadap model random walk dengan populasi 900 yang menghasilkan nilai $\beta=0.046$	23

Gambar 4.1b Hasil pengolahan data terhadap model SIR dengan populasi 900 dengan menggunakan nilai $\beta=0.046$ yang diperoleh dari model sebelumnya.	23
Gambar 4.1c Hasil pengolahan data terhadap model random walk dengan populasi 1158 yang menghasilkan nilai $\beta=0.046$	24
Gambar 4.1b Hasil pengolahan data terhadap model SIR dengan populasi 1158 dengan menggunakan nilai $\beta=0.095$ yang diperoleh dari model sebelumnya.	24
Gambar 4.2a Hasil pengolahan data menggunakan model random walk jumlah agen terinfeksi awal 60 yang menghasilkan nilai $\beta=0.079$	25
Gambar 4.2b Hasil pengolahan data menggunakan model SIR dengan jumlah agen terinfeksi awal 60 yang menggunakan nilai $\beta=0.079$	25
Gambar 4.3a Hasil pengolahan data menggunakan model random walk dengan $R_0= 1.78$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.....	27
Gambar 4.3b Hasil pengolahan data menggunakan model SIR dengan $R_0=1.78$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.....	27
Gambar 4.4a Hasil pengolahan data menggunakan model random walk dengan $V_{MAX} = 2$ yang menghasilkan nilai $\beta=0.45$	30
Gambar 4.4b Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai $\beta=0.45$	31
Gambar 4.5a Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai $R_0 = 6$ dan variasi parameter waktu.....	32
Gambar 4.5b Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan $V_{MAX} = 1$	33
Gambar 4.5c Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan $V_{MAX} = 8$	33
Gambar 4.5d Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan $V_{MAX} = 9$	34
Gambar 4.6a Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi $N_{POP} = 500$	35
Gambar 4.6b Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi jumlah populasi $N_{POP} = 500$ dan $V_{MAX} = 1$	35
Gambar 4.6c Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi $N_{POP} = 500$ dan $V_{MAX} = 2$	36

Gambar 4.6d Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi $N_{POP} = 500$ dan $V_{MAX} = 3$	36
Gambar 4.7a Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi $N_{INF} = 5$ dan $R_0 = 6$	37
Gambar 4.7b Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi $N_{INF} = 5$ dan $V_{MAX} = 1$	38
Gambar 4.7c Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi $N_{INF} = 5$ dan $V_{MAX} = 2$	38
Gambar 4.7d Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi $N_{INF} = 5$ dan $V_{MAX} = 3$	39
Gambar 4.7e Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi $N_{INF} = 5$ dan $V_{MAX} = 4$	39
Gambar 4.7f Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi $N_{INF} = 5$ dan $V_{MAX} = 5$	40
Gambar 4.8 Hasil pengolahan salah satu data mentah pada menggunakan model random walk....	41
Gambar 4.9 Hasil pengolahan data penyebaran influenza di sekolah asrama dengan parameter V_{MAX} terkalibrasi.....	41
Gambar 4.10a Hasil pengolahan data SARS di Hong Kong dengan parameter V_{MAX} terkalibrasi...	42
Gambar 4.10b Hasil keluaran salah satu simulasi pada program terkait penyebaran SARS.....	43
Gambar 4.11a Hasil simulasi demam berdarah dengan parameter V_{MAX} dan waktu terkalibrasi.....	43
Gambar 4.11b Salah satu hasil pengolahan data demam berdarah dengan parameter V_{MAX} dan waktu terkalibrasi.....	44
Gambar 4.12a Hasil prediksi jumlah kasus aktif COVID-19 untuk 25 kali simulasi di Kecamatan Coblong dengan dengan parameter V_{MAX} dan waktu terkalibrasi.....	45
Gambar 4.12b Hasil prediksi jumlah kasus aktif COVID-19 untuk 50 kali simulasi di Kecamatan Coblong dengan parameter V_{MAX} dan waktu terkalibrasi.....	45

DAFTAR TABEL

Tabel 4.1a Nilai <i>root mean squared error</i> untuk variasi populasi	25
Tabel 4.1b Nilai <i>mean squared error</i> untuk variasi populasi	25
Tabel 4.2a Nilai <i>root mean squared error</i> untuk variasi jumlah terinfeksi.	27
Tabel 4.2b Nilai <i>mean squared error</i> untuk variasi jumlah terinfeksi.....	27
Tabel 4.3a Nilai <i>root mean squared error</i> untuk variasi VMAX.....	31
Tabel 4.3b Nilai <i>mean squared error</i> untuk variasi VMAX.....	32
Tabel 4.4 Nilai kesalahan untuk pengolahan data observasi penyebaran influenza pada suatu sekolah asrama di Inggris dari tiap model beserta variasinya.....	42
Tabel 4.5 Tabel prediksi beberapa besaran pada penyebaran wabah COVID-19 di Kecamatan Coblong.....	45

BAB I

PENDAHULUAN

1.1 Latar Belakang

Merebaknya suatu wabah penyakit merupakan permasalahan besar yang menghantui peradaban manusia dari masa ke masa. Salah satu wabah yang pertama kali tercatat dalam sejarah misalnya terjadi di Athena pada sekitar tahun 430 SM, menewaskan sekitar 75.000 – 100.000 jiwa atau 25% dari populasi Athena saat itu (Litman, 2009:456-467). Melangkah ke abad 14, dunia dihadapkan kepada pandemi wabah hitam yang merupakan wabah paling mematikan yang pernah tercatat dalam sejarah. Wabah ini menelan sekitar 75-200 juta korban jiwa dengan wilayah terdampak paling parah berada di Benua Eropa – dalam kurun waktu sekitar 7 tahun, setidaknya 60% dari populasi Eropa pada saat itu menjadi korban dari keganasan wabah ini (Duncan, 2005). Pun meski telah memasuki era modern di mana ilmu medis telah mengalami perkembangan signifikan, wabah masih menjadi ancaman bagi manusia. Seperti saat ini contohnya dunia dihadapkan oleh pandemi Covid-19.

Nyatanya, dalam menghadapi wabah penyakit, manusia tidak hanya memerlukan pengobatan dan vaksinasi yang efektif. Meskipun suatu wabah relatif tidak mematikan, jika penyebarannya relatif cepat dapat membuat kepanikan masal yang berujung pada terganggunya stabilitas sosial, politik, dan ekonomi dari suatu daerah. Sistem pelayanan kesehatanpun dapat terganggu jika terjadi penambahan jumlah pasien dalam jumlah besar pada waktu yang relatif singkat. Oleh karena itu, informasi terkait bagaimana suatu wabah dapat menyebar akan sangat membantu dalam pencegahan ataupun mitigasi dari mewabahnya suatu patogen. Pertumbuhan populasi terjangkit wabah yang secara alami bersifat eksponensial akan sangat efektif ditekan jika informasi terkait wabah tersebut diketahui sedini mungkin.

Secara umum terdapat dua jenis model yang dikembangkan untuk memodelkan penyebaran suatu wabah yaitu model kompartemen dan model berbasis agen. Model kompartemen memiliki karakteristik yang secara umum bersifat deterministik sedangkan model berbasis agen bersifat stokastik .

Pada tugas akhir ini, penulis menggunakan model berbasis agen untuk mendeskripsikan bagaimana suatu wabah dapat menyebar. Model pada tugas akhir ini dibangun berdasarkan gerakan acak (*random walk*) dari agen yang direpresentasikan oleh partikel granular. Random walk merupakan proses stokastik yang jamak digunakan dalam mendeskripsikan gerakan suatu organisme secara matematis, termasuk dalam memodelkan pergerakan atau mobilitas manusia yang mana sangat mempengaruhi bagaimana suatu wabah dapat menyebar (Codling,

2008). Interaksi granular digunakan karena dengan model interaksi ini, saat setelah terjadinya kontak atau tumbukan, partikel tidak harus memantul atau berubah arah dengan cepat atau signifikan yang secara intuitif lebih menggambarkan gerakan manusia.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana random walk dari partikel granular dapat digunakan untuk memodelkan penyebaran suatu wabah?
2. Bagaimana hasil simulasi berkorelasi dengan besaran-besaran atau parameter yang diperoleh dari model kompartemen SIR dalam mengolah data riil?
3. Bagaimana hasil prediksi model tersebut jika dibandingkan dengan model lainnya?

1.3 Tujuan Penelitian

Tujuan tugas akhir ini adalah sebagai berikut:

1. Mengidentifikasi parameter dan pengaruh besaran pada model random walk dari partikel granular terhadap penyebaran suatu wabah.
2. Menentukan korelasi model terhadap parameter dalam model kompartemen SIR saat mengolah data riil.
3. Menghasilkan hasil prediksi yang relevan terhadap persebaran suatu wabah berdasarkan model random walk dari partikel granular.

1.4 Metodologi Penelitian

Metodologi yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Studi Literatur
Studi literatur mengenai *random walk*, model berbasis agen, serta partikel granular.
2. Membuat model
Melakukan simulasi berdasarkan formulasi matematis yang diperoleh dari studi literatur.
3. Validasi Model
Menganalisis keluaran dari model serta membandingkannya dengan keluaran model acuan.
4. Evaluasi Model

Evaluasi dilakukan dengan melihat besar nilai simpangan atau *error* dari model saat mengolah data riil.

1.5 Sistematika Penulisan

Penulisan Tugas Akhir terbagi menjadi lima bab, yaitu:

Bab satu merupakan pendahuluan yang terdiri dari latar belakang , rumusan masalah, tujuan, ruang lingkup studi, metodologi penelitian, dan sistematika penulisan.

Bab dua merupakan landasan teori yang terdiri dari *random walk*, partikel granular, dan klasifikasi model penyebaran penyakit.

Bab tiga merupakan metode penelitian yang terdiri alur kerja model, asumsi, dan analisis dan validasi awal dari hasil keluaran

Bab empat merupakan hasil dan pembahasan sementara yang terdiri dari pemaparan hasil, kalibrasi, dan validasi eksternal.

Bab lima merupakan simpulan dan saran yang memaparkan kesimpulan dari penelitian dan saran dari penyusun untuk perbaikan penelitian dimasa selanjutnya.

BAB II

LANDASAN TEORI

2.1 Random Walk

Random walk merupakan proses stokastik yang terbentuk akibat serangkaian pergerakan acak dari suatu variabel independen (Kareiva 1983). Dasar dari teori random walk dapat ditelusuri kembali pada gerakan acak partikel serbuk sari (gerak Brownian) yang diteliti oleh Robert Brown (1828). Sementara istilah "random walk" sendiri pertama kali diperkenalkan oleh Karl Pearson (1905) untuk mendeskripsikan pergerakan nyamuk di hutan dalam suratnya yang dimuat oleh *Nature*. Fisikawan seperti Einstein (1905,1906) dan Smoluchowski (1916) kemudian membawa prinsip random walk ke banyak bidang seperti dalam pemrosesan stokastik, kebisingan acak (random noise), dan analisis spektral.

Model paling sederhana dari random walk adalah model yang tak berkorelasi dan yang tak memiliki bias. Tak berkorelasi di sini berarti bahwa suatu langkah dari partikel yang mengalami random walk tidak memiliki korelasi atau independen dari langkah-langkah sebelumnya. Sedangkan, tak memiliki bias berarti bahwa setiap arah pergerakan partikel yang mengalami random walk memiliki peluang yang sama. Sebagai contoh, random walk satu dimensi dikatakan tidak memiliki bias jika besaran peluang partikel tersebut untuk bergerak ke kanan maupun ke kiri memiliki nilai yang sama. Proses ini dalam dimensi yang lebih tinggi merupakan pendeskripsian dari gerak Brownian dan model serupa dapat digunakan untuk membangun persamaan difusi (Codling, 2008).

Berikutnya terdapat random walk terkorelasi (correlated random walk), dalam model ini suatu langkah memiliki korelasi dengan langkah sebelumnya, pada akhirnya langkah ke- $(n+k)$ akan memiliki korelasi yang sangat kecil dengan langkah ke- n jika nilai k cukup besar. Random walk terkorelasi sering digunakan dalam memetakan pergerakan hewan karena kebanyakan hewan cenderung bergerak ke depan (Renshaw dan Henderson, 1981).

Terdapat juga random walk yang dipengaruhi oleh bias tertentu yang menentukan arah gerak netto pada akhirnya. Ini terjadi jika probabilitas suatu partikel yang mengalami random walk memiliki peluang yang lebih besar untuk bergerak pada arah tertentu ketimbang arah lainnya. Random walk jenis ini dikenal dengan istilah *biased random walk* (BRW) atau jika teramati korelasi antar langkah selama pergerakan partikel dapat juga disebut sebagai *biased and correlated random walk* (BCRW). Bias yang mempengaruhi suatu random walk biasanya disebabkan oleh faktor eksternal hingga spasial dari lingkungan misalnya gradien kimia dalam larutan, medan listrik, medan magnet, perubahan konsentrasi yang tidak homogen, dan lain sebagainya (Pu et.al, 2015).

Pada tugas akhir ini akan digunakan model yang secara visual melibatkan random walk sederhana dalam dua dimensi. Namun, karena di dalam model tiap partikelnya terdistribusi secara homogen dan isotropik, perhitungannya dapat didekati menggunakan model satu dimensi.

2.1.1 Random Walk dalam Satu Dimensi

Misalkan terdapat partikel yang bergerak acak pada satu dimensi. Mula-mula partikel ini berada di titik $x = 0$ dan kemudian bergerak dalam suatu langkah pendek konstan δ dalam waktu singkat τ . Pergerakan partikel dianggap acak tanpa bias dan tak berkorelasi sehingga peluang partikel untuk bergerak ke kanan dan ke kiri memiliki nilai yang sama yaitu $\frac{1}{2}$. Setelah langkah pertama, partikel dapat berada sejauh δ di sebelah kanan atau kiri dari posisi mula-mula. Setelah langkah kedua, partikel dapat berada 2δ di sebelah kanan atau kiri dari posisi mula-mula atau di titik awal ($x = 0$). Perhatikan bahwa, setelah sejumlah genap langkah, partikel berada sejauh genap δ dari titik asal, begitu pula pada jumlah langkah yang ganjil. Sehingga peluang partikel tersebut berada di kanan sejauh $m\delta$ dari titik asal setelah sejumlah n langkah di mana m dan n merupakan bilangan genap diberikan oleh (Codling, 2008) :

$$p(m, n) = \left(\frac{1}{2}\right)^n \binom{n}{\frac{n-m}{2}} = \frac{n!}{2^n ((n+m)/2)! ((n-m)/2)!} \quad (2.1)$$

Ini merupakan bentuk distribusi binomial dengan mean 0 dan varians n . Untuk nilai n yang besar, persamaan ini akan konvergen ke dalam suatu bentuk distribusi normal. Maka untuk nilai waktu $t = n\tau$ yang besar, lokasi $x = m\delta$ terdistribusi secara normal dengan nilai mean 0 dan varians $\frac{\delta^2 t}{\tau}$. Dengan mengambil limit δ dan $\tau \rightarrow 0$ sehingga nilai $\frac{\delta^2 t}{2\tau} = D$, dimana D suatu konstanta yang dikenal sebagai koefisien diffusi, memberikan fungsi kerapatan probabilitas (*probability density function*) dari posisi partikel setelah waktu t :

$$p(x, t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(\frac{-x^2}{4Dt}\right) \quad (2.2a)$$

Untuk kasus yang lebih umum dimana panjang langkah dari partikel tidak konstan tetapi acak namun masih memenuhi distribusi normal (Gaussian), random walk ini dikenal sebagai *Gaussian random walk*. Fungsi probabilitas yang mendeskripsikan probabilitas dari posisi partikel pada kasus ini dapat dituliskan sebagai :

$$p(x, t) = \frac{1}{\sqrt{2\pi t\sigma^2}} \exp\left(\frac{-x^2}{4t\sigma^2}\right) \quad (2.2b)$$

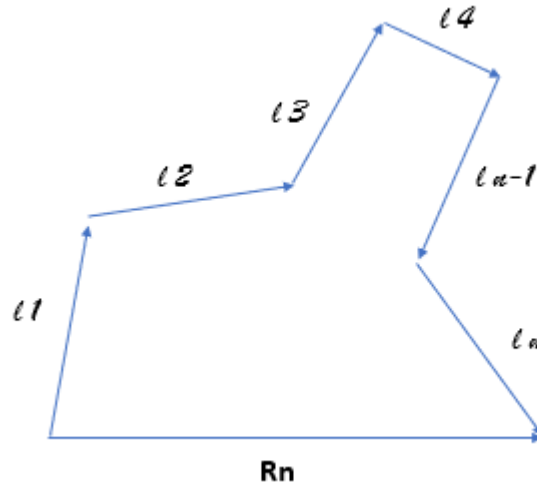
Dengan σ^2 sebagai varians dari nilai langkah acak yang digunakan. Posisi rata-rata dari partikel $E(X_t)$ dan nilai rata-rata kuadrat dari perpindahannya $E(X_t^2)$ kemudian secara statistik didefinisikan oleh :

$$E(X_t) = \int_{-\infty}^{\infty} xp(x, t)dx \quad (2.3a)$$

$$E(X_t^2) = \int_{-\infty}^{\infty} x^2 p(x, t)dx \quad (2.3b)$$

2.1.2 Random Walk dalam Dimensi Dua

Dalam dimensi dua, selain dihadapkan pada probabilitas jarak, suatu partikel yang mengalami random walk juga dihadapkan pada probabilitas sudut pergerakan. Misalkan terdapat suatu organisme katakanlah semut yang gerakannya dapat didekripsikan sebagai partikel yang mengalami random walk di ruang berdimensi dua seperti pada gambar di bawah.



Gambar 2.1 Sketsa lintasan dari seekor semut yang bergerak secara acak pada dimensi dua.

Dari gambar tersebut maka perpindahan yang dialami semut setelah n langkah adalah :

$$R_n = \sum_{m=1}^n (x_m, y_m) \quad (2.4)$$

Jika panjang lintasan ke- m dari semut diberikan oleh $l_m = \sqrt{x_m^2 + y_m^2}$ maka ekspektansi dari perpindahan dan kuadrat rataanya serta nilai rataan sudut putar yang diukur searah jarum jam diberikan oleh (Kareiva 1983) :

$$E(l) = \int_0^{\infty} lp(l)dl \quad (2.5a)$$

$$E(l^2) = \int_0^{\infty} l^2 p(l)dl \quad (2.5b)$$

$$c = E(\cos\theta) = \int_{-\pi}^{\pi} \cos\theta g(\theta)d\theta \quad (2.5c)$$

$$s = E(\sin\theta) = \int_{-\pi}^{\pi} \sin\theta g(\theta)d\theta \quad (2.5d)$$

Dengan $g(\theta)$ merupakan fungsi probabilitas kepadatan dari sudut sirkular yang salah satunya dapat diberikan oleh (Codling, 2008) :

$$f_{WN}(\theta; \theta_0, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \exp\left[\frac{-(\theta - \theta_0 + 2\pi k)^2}{2\sigma^2}\right] \quad (2.6)$$

Probabilitas perpindahan partikel sejauh \mathbf{r} setelah waktu t diberikan oleh :

$$p(\mathbf{r}, t) = \frac{1}{2\pi t \sigma^2} \exp\left(\frac{-|\mathbf{r}|^2}{4t\sigma^2}\right) \quad (2.7)$$

Ekspektansi perpindahan partikel sejauh \mathbf{R}_n kemudian dirumuskan sebagai (Kareiva 1983) :

$$E(\mathbf{R}_n) = nE(l^2) + E(l)^2 \left\{ \frac{n(c - c^2 - s^2) - c}{(1 - c)^2 + s^2} + \frac{2s^2(c^2 + s^2)^{\frac{(n+1)}{2}}}{[(1 - c)^2 + s^2]^2} \right\} \quad (2.8)$$

Pada gerakan yang acak secara murni, nilai $c = s = 0$, sehingga persamaan 2.8 akan menjadi (Codling, 2008) :

$$E(\mathbf{R}_n) = nE(l^2) \quad (2.9)$$

2.2 Partikel Granular

Secara fisis, suatu partikel granular didefinisikan sebagai bagian dari sistem makroskopis yang interaksi berkelompoknya seringkali membuat sistem ini terlihat memiliki sifat sebagai padatan atau fluida secara bersamaan. Karakteristik dari sistem granular selain sangat ditentukan oleh interaksi antar partikel juga ditentukan oleh interaksi partikel dengan lingkungannya. Jika pengaruh lingkungan dapat diabaikan, maka sistem ini dinamakan sistem granular kering. Brown dan Richards (1970) mengklasifikasi sistem atau material granular sebagai berikut :

- *Material granular umum* terdiri atas padatan diskrit yang hampir setiap waktu tiap partikelnya berada dalam keadaan saling berkontak.
- *Powder* atau bubuk merupakan sistem granular yang ukuran diameter tiap individu partikelnya kurang dari 100 μm . Bubuk kemudian diklasifikasikan Kembali menjadi tiga jenis yaitu bubuk granular (10-100 μm), bubuk terhaluskan(1-10 μm), dan bubuk sangat terhaluskan(0,1-1 μm).
- *Padatan granular* merupakan sistem granular yang ukuran diameter tiap individu partikelnya berada pada rentang (100-3000 μm).

Partikel granular dalam interaksinya juga dapat mengalami deformasi atau perubahan bentuk. Pada model dalam tugas akhir ini, partikel dianggap memiliki

sifat yang sangat lembut sehingga saat berinteraksi, partikel dapat beririsan satu sama lain tanpa gaya hambat ataupun gaya tolak.

2.3 Klasifikasi Model Penyebaran Wabah

Dalam memodelkan penyebaran penyakit menular terdapat dua kerangka yang umumnya digunakan yaitu model berbasis kompartemen dan model berbasis agen. Model kompartemen memiliki sejarah yang lebih panjang dalam pemodelan penyebaran penyakit jika dibandingkan model berbasis agen dan melibatkan perhitungan yang lebih cepat secara analitik ataupun numerik ketimbang model berbasis agen. Sementara model berbasis agen melibatkan perhitungan jauh lebih banyak data dan relatif baru populer setelah mapannya teknologi informasi dewasa ini yang memungkinkan perhitungan pada model ini dikerjakan oleh kalangan yang lebih luas. Fokus dari model ini pun berbeda, model kompartemen berfokus pada laju perubahan keadaan dari populasi di tiap-tiap kompartemen yang telah didefinisikan, sedangkan model berbasis agen berfokus mengamati laju interaksi antar individu agen dan faktor-faktor yang mempengaruhinya (Brauer, 2008).

Meskipun memiliki basis pendekatan yang berbeda, penelitian menunjukkan bahwa masing-masing kerangka dapat memberikan hasil yang serupa. Model berbasis agen biasanya memberikan hasil penelitian dengan resolusi yang lebih tinggi namun dengan lebih banyak perhitungan serta data verifikasi yang lebih sulit diperoleh. (Riley et.al, 2015) Sedangkan model kompartemen melibatkan perhitungan yang relatif sedikit, cepat serta data verifikasi yang mudah diperoleh, tetapi memiliki resolusi yang lebih rendah. (Tracy et.al, 2018) Hal ini membuat para peneliti juga mulai membangun model hibrida yang melibatkan kedua kerangka dalam menyelesaikan permasalahan terkait model penyebaran wabah.

2.3.1 Model Kompartemen

Dalam epidemiologi, model kompartemen dikenali melalui sifatnya yang umumnya deterministik. Bentuk awal model ini diperkenalkan pertama kali secara formal oleh Kermack dan McKendrick (1927). Dalam modelnya, Kermack dan McKendrick membagi total populasi N ke dalam tiga kompartemen yaitu kompartemen *susceptible*, *infectious*, dan *recovered* (SIR) yang mencerminkan keadaan tiap populasi di dalamnya. Perubahan populasi pada tiap kompartemen ini kemudian diberikan oleh sistem persamaan diferensial sebagai berikut (Brauer, 2008).

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta IS}{N} \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}\tag{2.10}$$

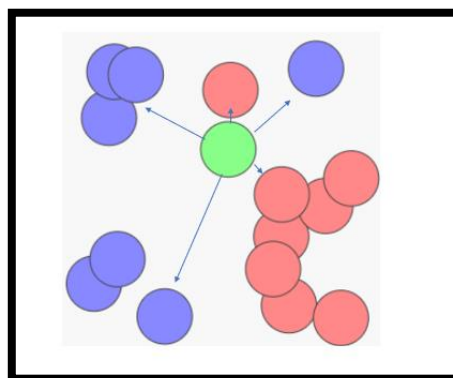


Gambar 2.2 Diagram skematik dari model kompartemen SIR.

Dari model ini didefinisikan dua konstanta yaitu β yang menyatakan frekuensi kontak rata-rata tiap waktu dari individu pada populasi tersebut dan γ menyatakan frekuensi populasi terinfeksi berubah keadaan menjadi sembuh tiap waktu. Dari dua konstanta ini, didefinisikan variabel baru $R_0 = \frac{\beta}{\gamma}$ sebagai bilangan reproduksi dasar yang menyatakan seberapa banyak individu dapat menginfeksi individu lainnya dalam suatu wabah. Jika $R_0 < 1$ maka wabah akan menghilang dalam jangka panjang, dan jika $R_0 > 1$ wabah akan meluas. Model ini dapat dikembangkan dengan menambah kompartemen serta memodifikasi persamaan diferensialnya menjadi beberapa jenis, sesuai dengan karakteristik yang teramati pada tiap populasi pada observasi. Seperti contohnya model SEIR yang menambahkan kompartemen E (*exposed*) untuk mendeskripsikan fraksi populasi yang telah melakukan kontak dengan individu terinfeksi namun belum menunjukkan gejala terinfeksi (Brauer, 2008).

2.3.2 Model Berbasis Agen

Model berbasis agen (*agent-based model*) dalam epidemiologi secara alami merupakan model yang bersifat stokastik. Model ini dibangun dengan mendeskripsikan bagaimana suatu agen bergerak hingga kemudian berinteraksi dengan agen lainnya. Gerak dari agen merupakan gerak acak dengan tingkat keacakan berbeda-beda bergantung lingkungan yang ingin dimodelkan. Dengan melihat karakteristik ini berbagai peristiwa yang melibatkan gerak acak yang terjadi di alam atau secara matematis seperti gerak Brownian, *cellular automata*, difusi partikel, gerak acak dari partikel gas pada suhu tertentu dapat digunakan sebagai model berbasis agen (Tracy et.al, 2018).



Gambar 2.3 Gambaran interaksi antar agen pada model penyebaran penyakit berbasis agen.

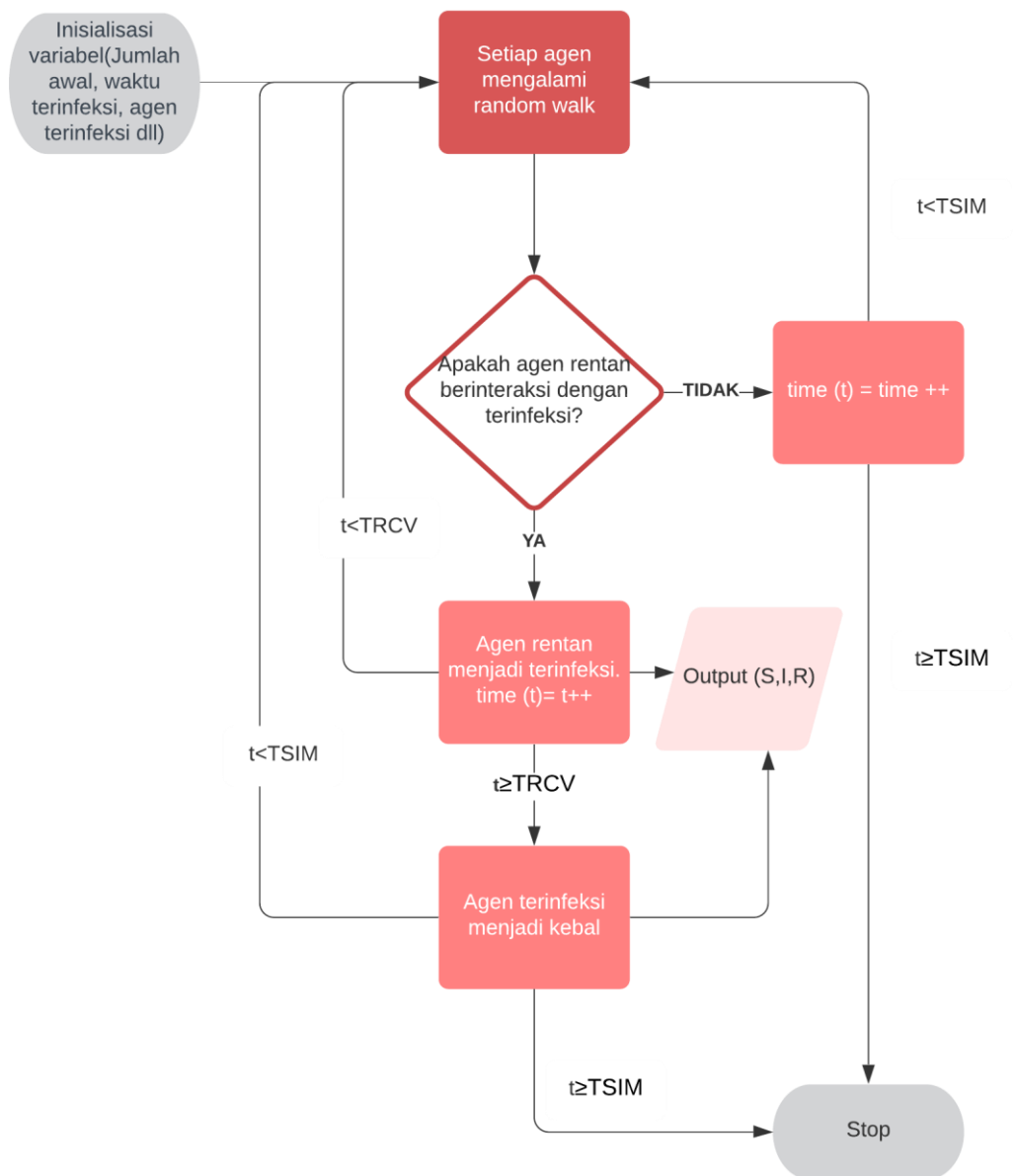
BAB III

METODE PENELITIAN

3.1 Alur Kerja Model

Model ini dibuat menggunakan bahasa pemrograman JavaScript. Model bekerja diawali dengan menginisialisasi jumlah agen yang sebelumnya telah ditentukan serta posisinya secara acak. Jarak terdekat rata-rata agen adalah r , dengan r merupakan jari-jari agen yang secara proporsional berubah sesuai jumlah populasi awal yang dimasukan. Agen dibagi ke dalam tiga kelompok, agen berwarna biru menggambarkan individu yang rentan (*susceptible*), agen berwarna merah menggambarkan individu yang terinfeksi (*infectious*), serta agen berwarna hijau menggambarkan individu yang telah sembuh (*recovered*). Pada mulanya hanya terdapat dua jenis agen yaitu agen yang rentan dan terinfeksi.

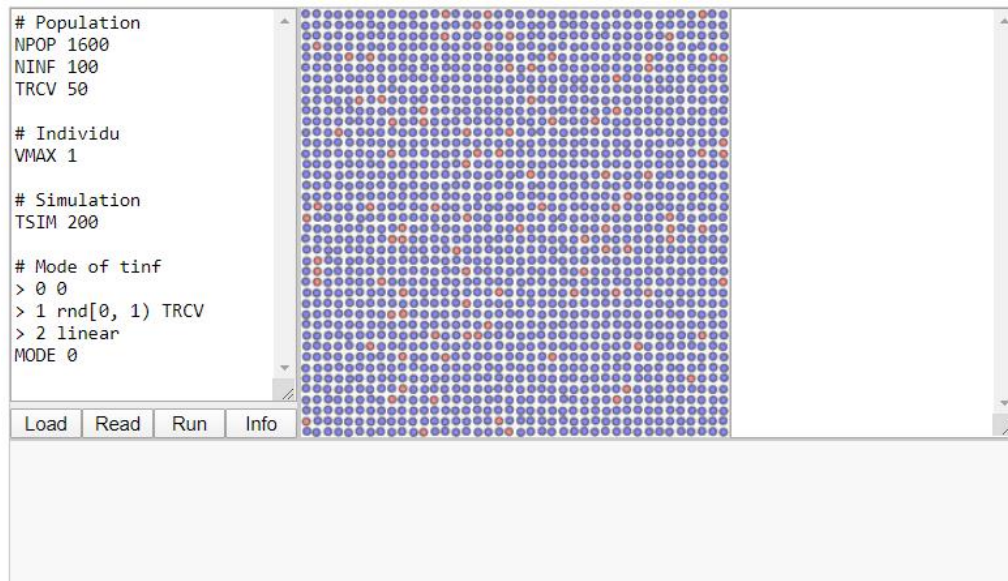
Interaksi diatur sedemikian rupa sehingga tidak ada tumbukan antar agen. Jika berada pada posisi yang sama atau saling berdekatan, agen akan saling beririsan atau bersentuhan tanpa adanya tolakan satu sama lain. Jika agen yang terinfeksi bersentuhan dengan agen yang rentan maka agen yang rentan tersebut akan berubah keadaannya menjadi agen terinfeksi. Setelah waktu yang ditentukan agen terinfeksi kemudian dapat berubah keadaannya menjadi sembuh. Agen yang sembuh dianggap telah memiliki imunitas dan tidak dapat terinfeksi kembali seperti pada model SIR. Ini kemudian terus diulangi hingga waktu simulasi yang telah ditentukan berakhir. Alur kerja dari model ini secara skematis digambarkan oleh gambar 3.1.



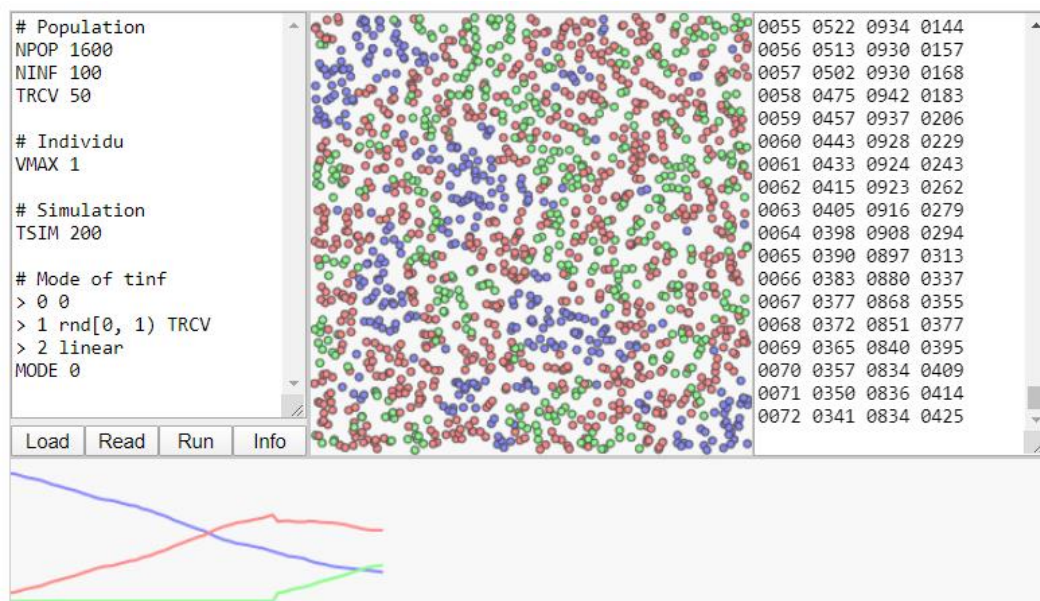
Gambar 3.1 Diagram alir dari alur kerja program .

Gambar 3.2a merupakan keadaan standar dari program ini. Kotak bagian kiri berisi variabel-variabel yang besarannya dapat diubah-ubah. Di bawah menu “#Population” terdapat variabel NPOP menyatakan jumlah populasi, NFIF menyatakan jumlah terinfeksi di awal program sebelum dijalankan serta TRCV sebagai waktu yang dibutuhkan oleh agen terinfeksi untuk sembuh atau berubah warna dari merah menjadi hijau.

3.1.1 Penjelasan mengenai Variabel pada Program dan Panduan Penggunaan



Gambar 3.2a Keadaan awal program.



Gambar 3.2b Keadaan program sesaat setelah dijalankan

Di bawahnya terdapat menu “# Individu” yang berisi variabel VMAX. Variabel ini menyatakan kecepatan maksimal dalam dua dimensi yang dapat ditempuh agen tiap *time step*-nya. Menu “# Simulasi” berisi variabel TSIM yang menyatakan jumlah *time step* tiap simulasi.

Menu “# Mode of tinf” berisi variabel “MODE” yang menyatakan bagaimana agen yang terinfeksi di awal program akan berubah keadaannya menjadi agen yang sembuh. Jika dipilih “MODE 0”, maka agen yang terinfeksi di awal akan sembuh secara bersamaan setelah $TSIM = TRCV$. Jika “MODE 1” yang dipilih maka agen yang terinfeksi di awal akan sembuh dalam waktu yang acak. Jika “MODE 2” yang dipilih, maka agen akan sembuh secara linear, misal agen dengan indeks 0 akan sembuh setelah $TSIM = TRCV$, agen dengan indeks 1 akan sembuh setelah $TSIM=1+TRCV$, agen dengan indeks 2 akan sembuh setelah $TSIM=2+TRCV$ dan seterusnya.

Di bawah menu ini terdapat tombol “Load”, “Read”, “Run”, dan “Info”. Tombol “Load” berfungsi untuk memuat keadaan standar dari program, tombol “Read” untuk memuat perubahan jika terdapat variabel pada menu di atasnya yang diubah oleh pengguna, tombol “Run” berfungsi untuk menjalankan program, dan tombol “Info” yang saat diklik akan memunculkan informasi tentang program di kotak sebelah kanan.

Pada bagian terbawah, terdapat kotak memanjang yang saat program dijalankan akan memunculkan grafik dari jumlah populasi dari tiap agen seperti yang tampak pada gambar 3.2b. Garis biru menyatakan jumlah agen yang rentan, garis merah menyatakan jumlah agen terinfeksi dan garis hijau menyatakan jumlah agen yang telah sembuh.

Sementara, dibagian kanan program terdapat kotak yang memuat angka-angka saat program dijalankan seperti yang nampak pada gambar 3.2b. Secara berturut-turut dari kiri ke kanan angka-angka ini menyatakan waktu simulasi, jumlah agen yang rentan, serta jumlah agen yang telah sembuh.

3.1.2 Penjelasan mengenai Variabel pada Program dan Panduan Penggunaan

Dalam melakukan simulasi, program ini kurang efektif jika melakukan simulasi di atas 5000 agen. Data dengan fraksi terinfeksi di atas $1/5000$ akan kurang relevan jika disimulasikan menggunakan program ini. Untuk mengatasi hal ini dapat digunakan pengolahan data yang berfokus mengetahui distribusi penyebaran wabah pada populasi yang akan terinfeksi dan bukan keseluruhan populasi.

3.2 Asumsi

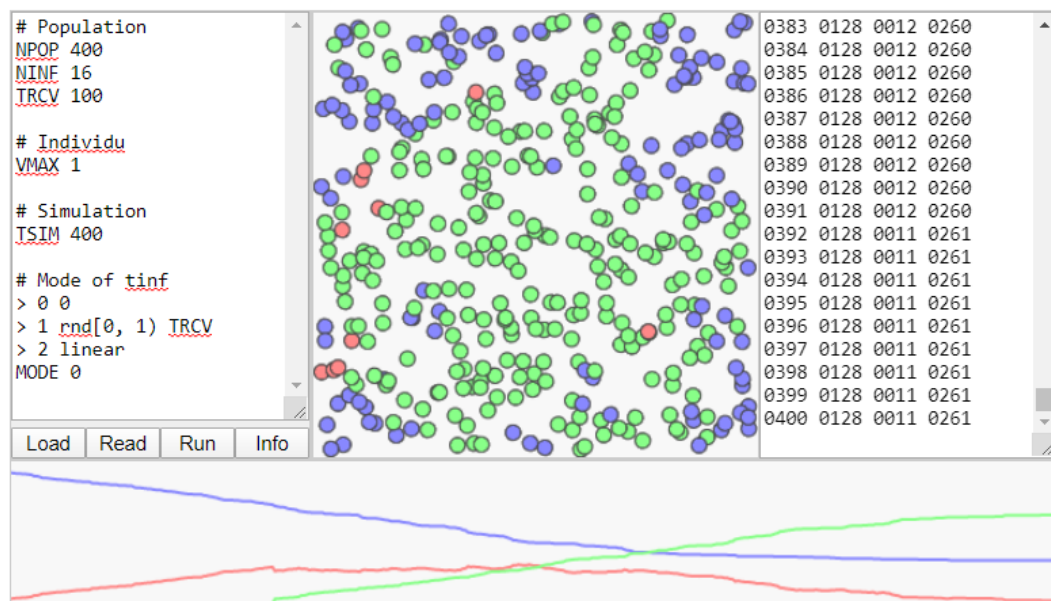
Model ini bekerja berdasarkan asumsi-asumsi berikut yaitu :

1. Agen terdistribusi secara homogen dan dapat dilihat sebagai sistem yang isotropis.
2. Agen yang telah sembuh dianggap telah memiliki kekebalan dan tidak dapat terinfeksi kembali.
3. Jumlah agen tiap waktunya tetap.

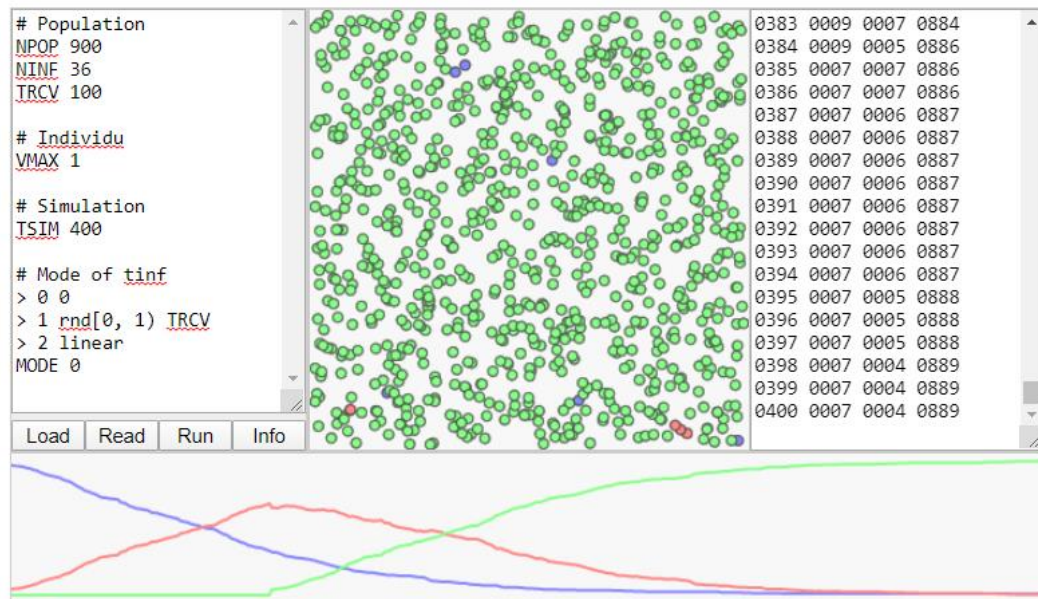
3.3 Hasil Keluaran Simulasi dengan Variasi Nilai Variabel Sembarang

Pada subbab berikut akan diperlihatkan hasil keluaran simulasi dengan variasi nilai variabel sembarang, sembarang berarti berubahnya nilai variabel tidak disesuaikan dengan data riil.

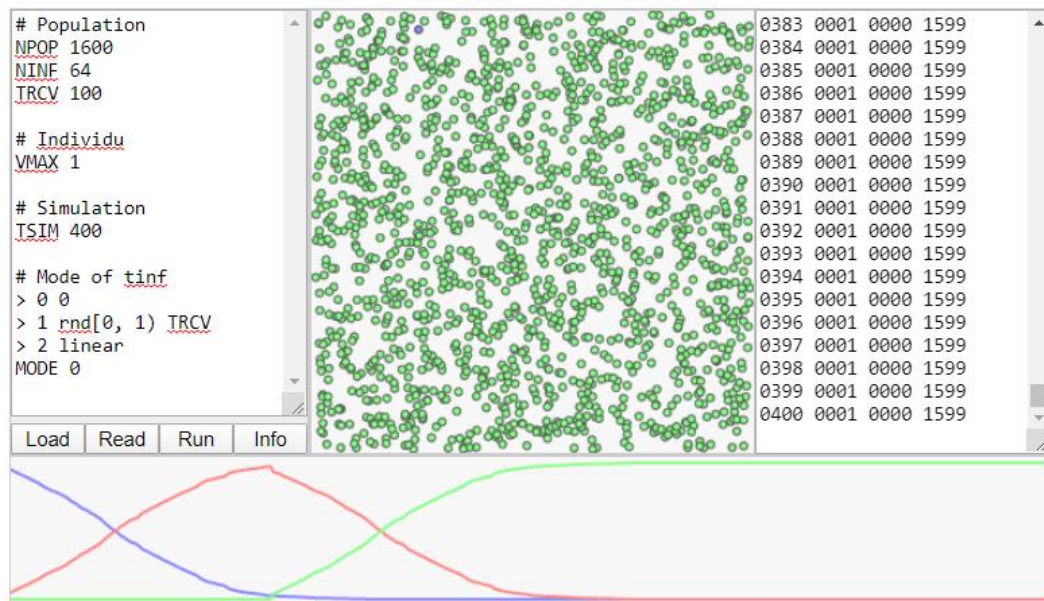
3.2.1 Hasil keluaran program dengan variasi jumlah populasi



Gambar 3.3a Tampilan akhir program dengan jumlah total populasi agen sebanyak 400 unit dan jumlah terinfeksi dibuat proporsional 4% dari total populasi.

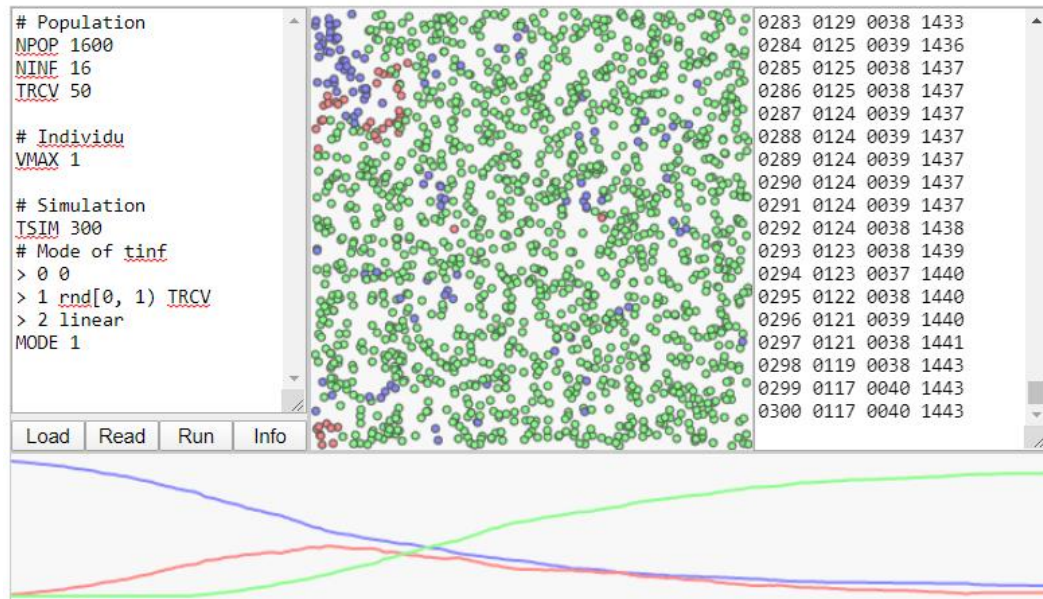


Gambar 3.3b Tampilan akhir program dengan jumlah total populasi agen sebanyak 900 unit dan jumlah terinfeksi dibuat proporsional 4% dari total populasi.

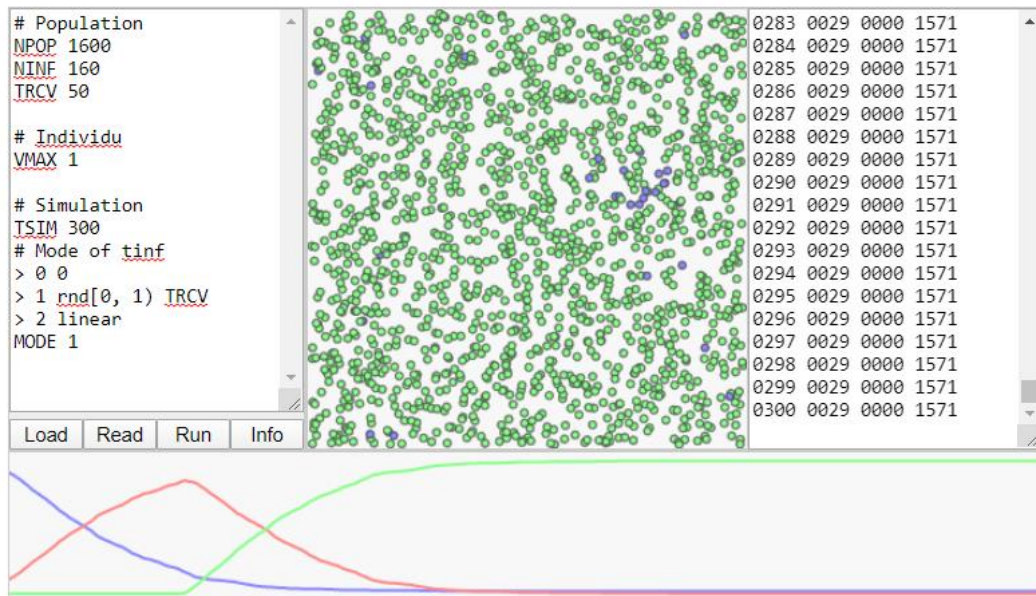


Gambar 3.3c Tampilan akhir program dengan jumlah total populasi agen sebanyak 900 unit dan jumlah terinfeksi dibuat proporsional 4% dari total populasi.

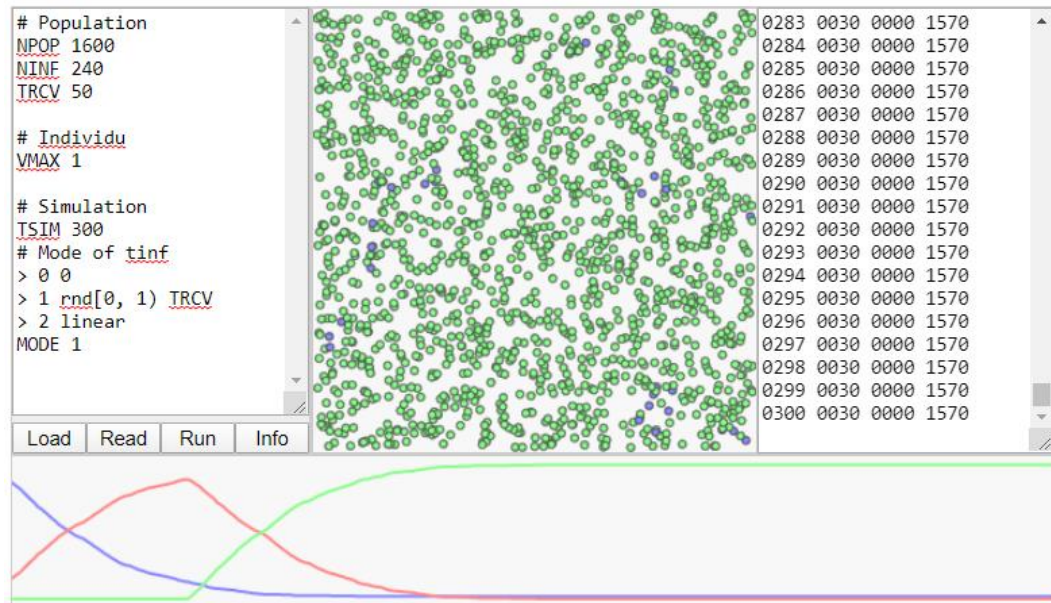
3.2.2 Hasil keluaran dengan variasi jumlah terinfeksi awal



Gambar 3.4a Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 1% dari total populasi.

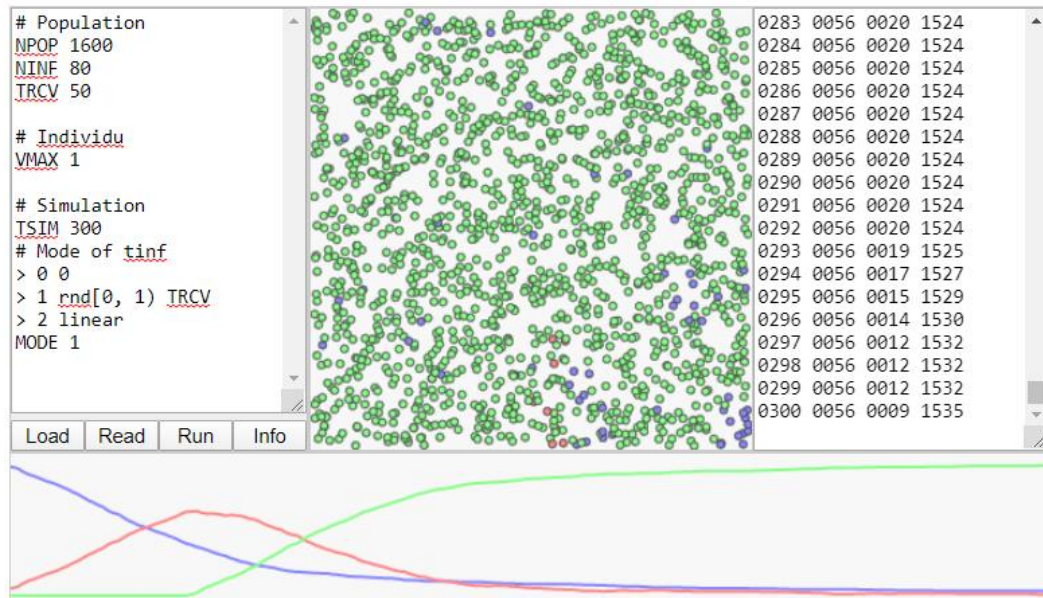


Gambar 3.4b Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 10% dari total populasi.

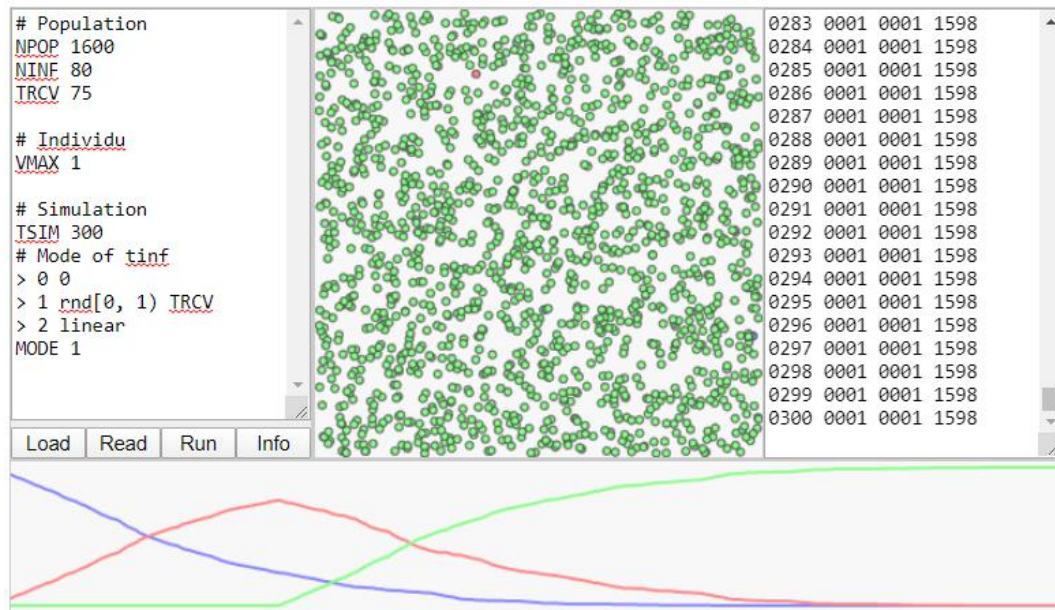


Gambar 3.4c Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 15% dari total populasi.

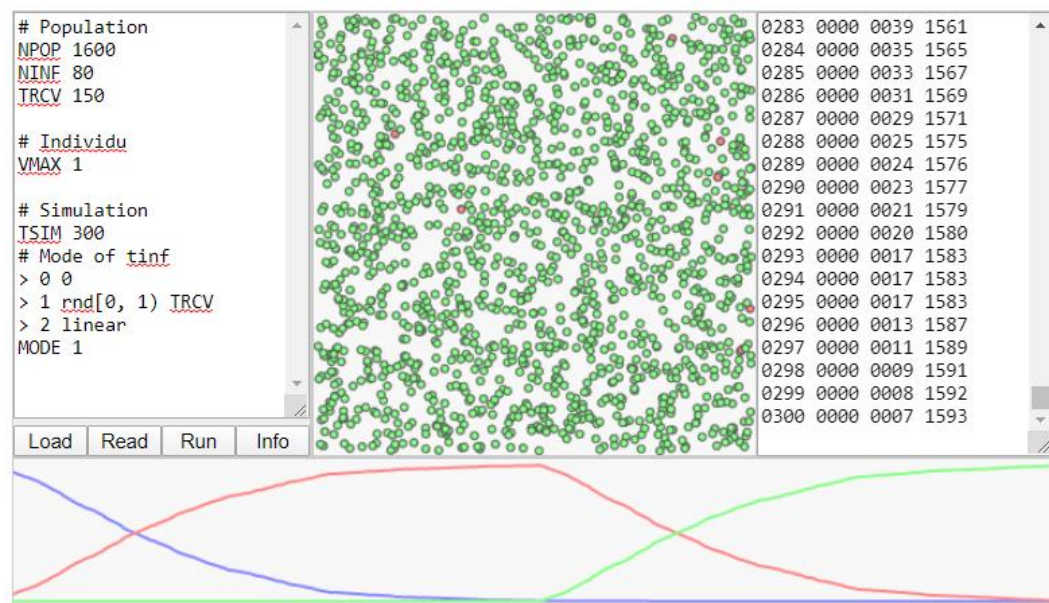
3.2.3 Hasil keluaran dengan variasi waktu terinfeksi



Gambar 3.5a Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 5% dari total populasi dan waktu untuk sembuh sebesar 50 *time steps*.

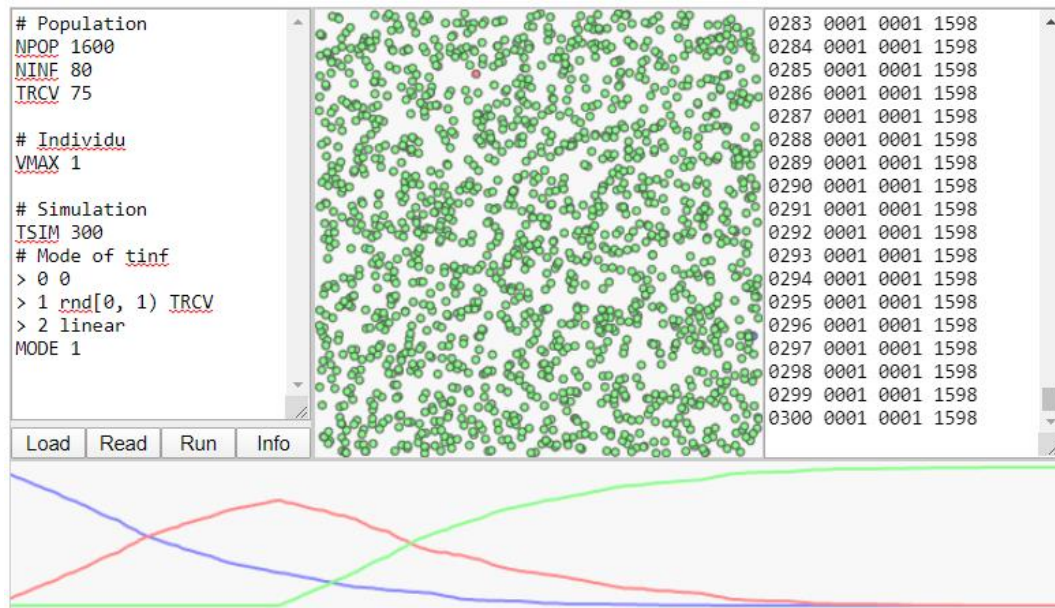


Gambar 3.5b Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 5% dari total populasi dan waktu untuk sembuh sebesar 75 *time steps*.

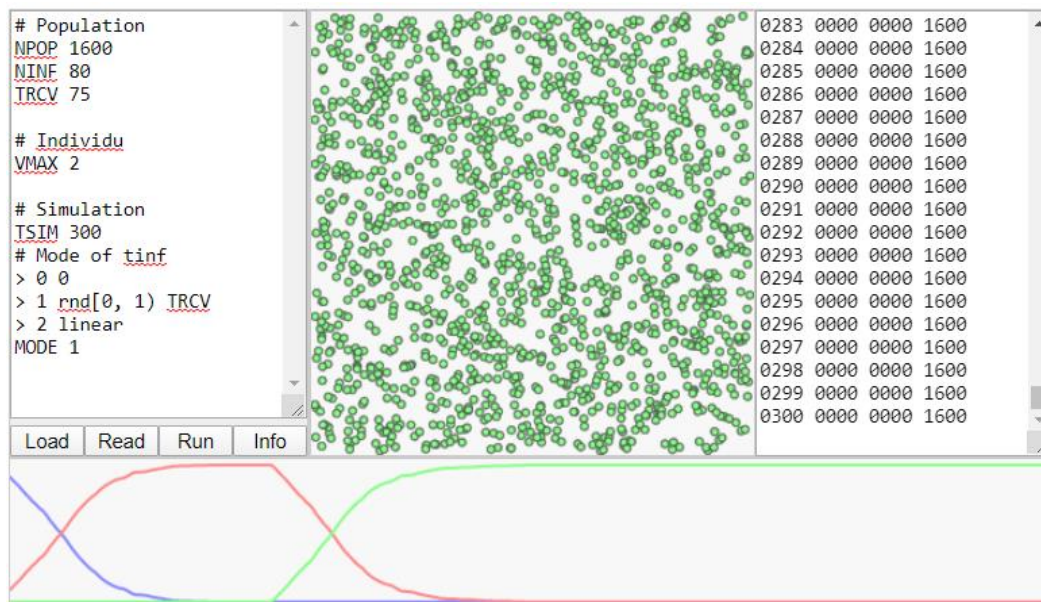


Gambar 3.5c Tampilan akhir program dengan jumlah populasi terinfeksi di awal sebesar 5% dari total populasi dan waktu untuk sembuh sebesar 150 *time steps*.

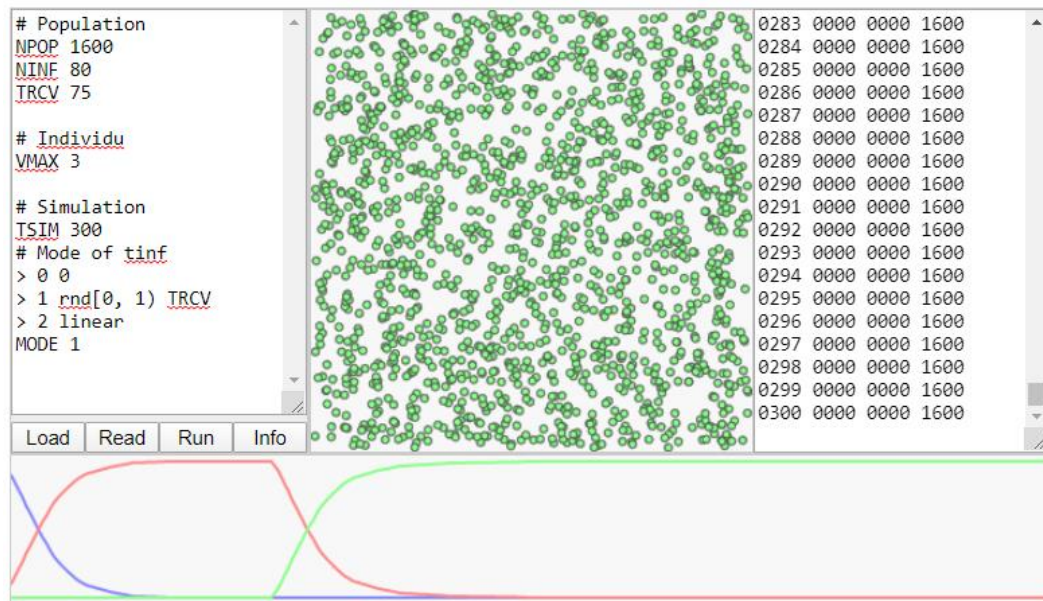
3.2.4 Hasil keluaran dengan variasi kecepatan partikel



Gambar 3.6a Tampilan akhir program dengan kecepatan agen maksimum 1px per *time steps*.



Gambar 3.6b Tampilan akhir program dengan kecepatan agen maksimum 2px per *time steps*.



Gambar 3.6c Tampilan akhir program dengan kecepatan agen maksimum 3px per *time steps*.

3.4 Analisis dan Verifikasi awal Terhadap Hasil Keluaran

Pada keadaan awal terlihat bahwa sistem partikel menyerupai suatu kisi homogen atau isotropik dengan jarak rata-rata r antar agennya. Nilai r berubah secara proporsional bergantung pada jumlah agen yang digunakan saat simulasi. Saat jumlah agen sedikit, nilai r akan membesar begitu pula sebaliknya. Jika kecepatan agen tetap dan jarak antar agen yang membesar, maka peluang agen untuk berinteraksi akan mengecil. Ini membuat peluang tiap agen untuk terinfeksi membesar jika jumlah total populasi agen bertambah seperti yang terlihat secara berurutan pada Gambar 3.3a, 3.3b, dan 3.3c. Secara analitik ini telah sesuai dengan persamaan 2.2b. Agar tetap relevan dengan model SIR, menaikkan VMAX dapat mengompensasi pengurangan jumlah populasi yang digunakan. Kalibrasi terhadap pengaruh VMAX dan jumlah populasi yang digunakan akan dijelaskan pada bab selanjutnya.

Sementara dari Gambar 3.4a, 3.4b dan 3.4c, terlihat jika populasi agen terinfeksi di awal ditambah, grafik pertumbuhan populasi terinfeksi akan menanjak lebih cepat. Ini terjadi karena dengan bertambahnya jumlah populasi agen terinfeksi, maka peluang agen yang rentan untuk berinteraksi dengan agen terinfeksi juga akan bertambah.

Berikutnya hasil simulasi memperlihatkan pada gambar 3.5a, 3.5b, dan 3.5c perubahan grafik pertumbuhan populasi agen bertambah lebih banyak jika waktu

agen untuk sembuh dari keadaan terinfeksi yaitu TRCV ditambah. Ini menunjukkan bahwa waktu untuk sembuh mempengaruhi peluang agen terinfeksi untuk menginfeksi agen lainnya.

Variasi terakhir dari model ini adalah variasi kecepatan agen, di sini terlihat grafik pertumbuhan populasi agen terinfeksi meningkat sangat cepat jika kecepatan maksimal agen ditambah. Ini berarti bahwa peluang terinfeksi suatu agen akan meningkat sangat cepat jika agen tersebut memiliki mobilitas yang tinggi. Secara analitik hasil ini akan diberikan oleh persamaan (2.2b) dan (2.3b). Secara fisis, kecepatan merupakan suatu besaran yang berbanding terbalik dengan waktu. Demikian juga dalam model ini, jika waktu simulasi diperpendek secara ekstrim misalkan menjadi hanya 30 *time step*, agar model ini tetap selaras dengan SIR, maka VMAX juga harus dinaikan, kalibrasi terkait hal ini akan dilakukan pada subab selanjutnya.

BAB IV

HASIL

4.1 Pendahuluan

Menurut Auchincloss dan Garcia (2015), suatu model berbasis agen yang digunakan dalam memodelkan persebaran suatu wabah agar bekerja dengan efektif harus melalui tahapan-tahapan berikut yaitu verifikasi, kalibrasi, dan validasi eksternal. Verifikasi merupakan proses pemeriksaan apakah kode pemrograman yang digunakan dalam pembuatan telah menghasilkan model yang bekerja sesuai formulasi matematis. Proses ini telah dilakukan di bagian akhir bab sebelumnya. Berikutnya adalah proses kalibrasi. Proses kalibrasi model dapat dilakukan secara kualitatif menggunakan data abstrak ataupun riil dengan membandingkan pengolahan model ini dengan model yang telah mapan. Bagian terakhir merupakan validasi eksternal yang melibatkan data riil untuk melihat seberapa terpercayanya model yang telah dikembangkan dengan membandingkan besaran-besaran yang dihasilkan secara kuantitatif dengan model yang ada (data yang digunakan harus berbeda dengan yang digunakan pada kalibrasi) atau mencoba melakukan prediksi terhadap data nyata.

4.2 Kalibrasi

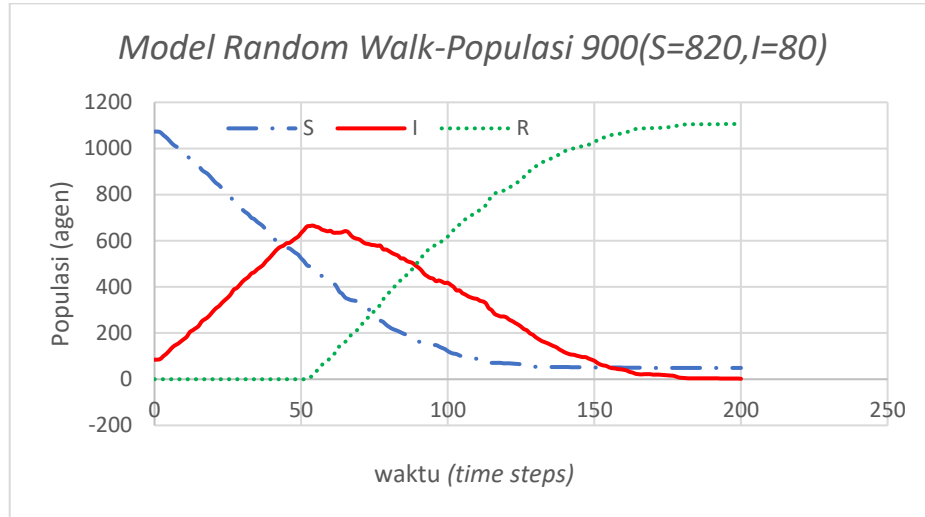
Dalam subbab ini penulis akan membandingkan simulasi model sebelumnya dengan model SIR yang dibuat menggunakan bahasa pemrograman Python untuk kemudian menentukan beberapa parameter dari model SIR seperti koefisien transmisi dan bilangan reproduksi dasar. Korelasi model berbasis agen ini terhadap model SIR sebenarnya dapat ditentukan secara analitik maupun numerik, namun karena menggunakan relatif banyak agen, korelasi ini lebih mudah dicari secara numerik.

Pada subbab ini akan ditentukan korelasi model terhadap parameter dari model SIR secara numerik dengan mengatur variasi populasi, jumlah terinfeksi, dan kecepatan maksimum agen. Sedangkan nilai variabel lainnya dibuat tetap. Variabel S, I, R dalam tiap grafik menyatakan secara berurutan jumlah atau fraksi populasi yang (i) rentan atau *susceptible*, (ii) terinfeksi atau *infected*, dan (iii) sembuh atau *recovered*.

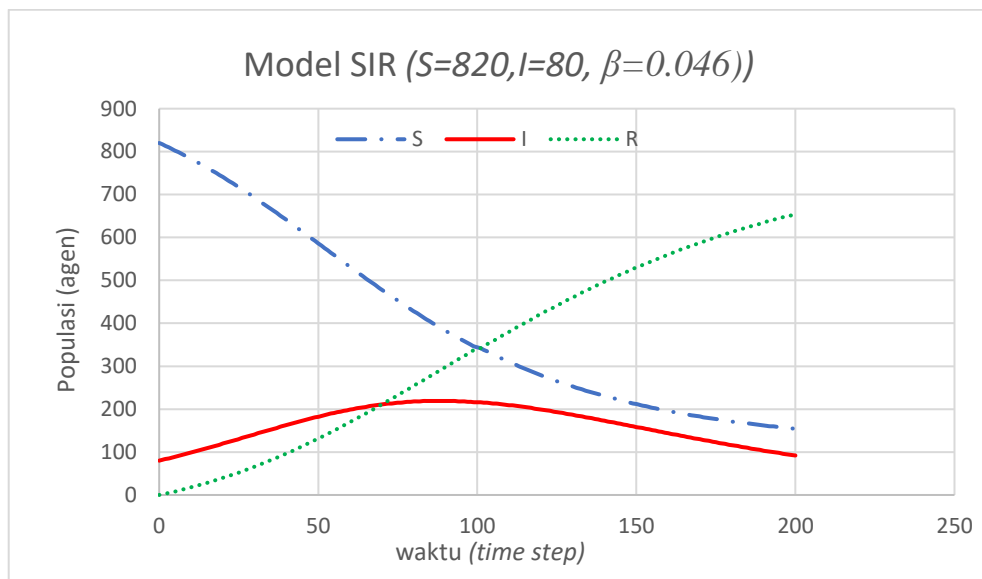
4.2.1 Menentukan Korelasi terhadap Model SIR dengan Variasi Populasi

Variasi pertama menggunakan populasi 900 agen, dengan jumlah agen awal terinfeksi sebanyak 80 unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel akan diperoleh nilai koefisien transmisi penyakit

$\beta=0.046$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 2.3$. Dari perbandingan grafik pada gambar 4.1a dan 4.1b terlihat pada keadaan ini, model ini belum begitu mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.



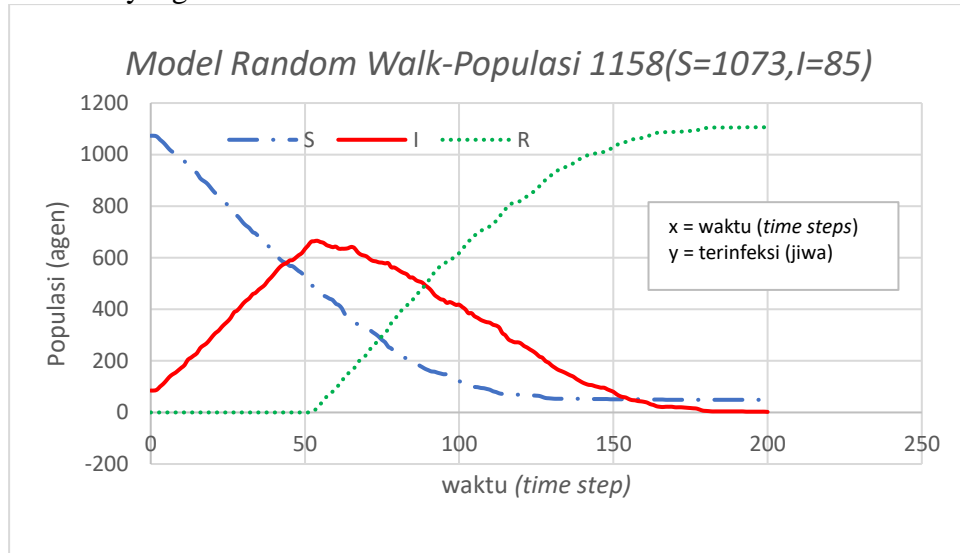
Gambar 4.1a Hasil pengolahan data terhadap model random walk dengan populasi 900 yang menghasilkan nilai $\beta=0.046$.



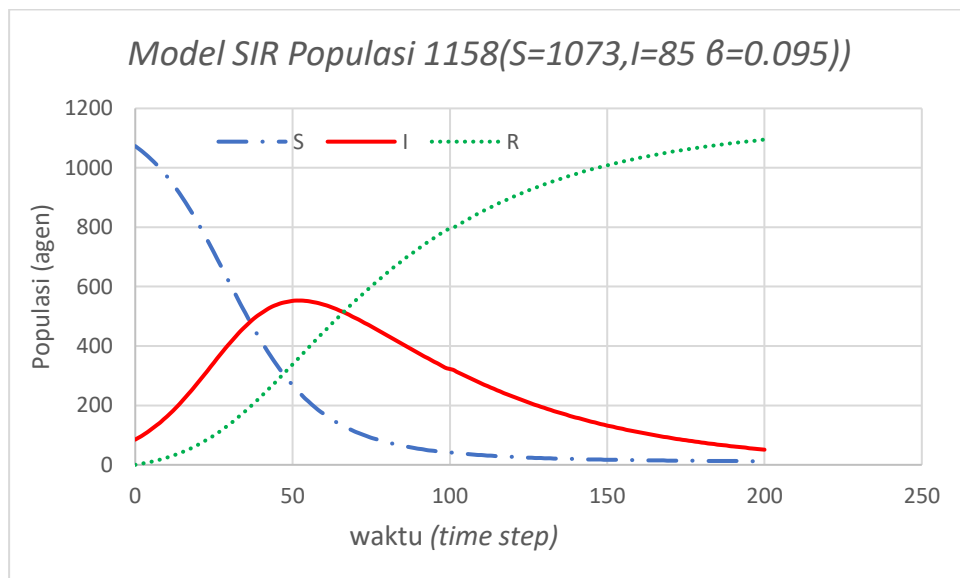
Gambar 4.1b Hasil pengolahan data terhadap model SIR dengan populasi 900 dengan menggunakan nilai $\beta=0.046$ yang diperoleh dari model sebelumnya.

Variasi kedua menggunakan populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 805unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel akan diperoleh nilai koefisien

transmisi penyakit $\beta=0.095$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 4.6$. Dari perbandingan grafik pada gambar 4.1c dan 4.1d terlihat pada keadaan ini, model ini telah terlihat mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.



Gambar 4.1c Hasil pengolahan data terhadap model random walk dengan populasi 1158 yang menghasilkan nilai $\beta=0.095$.



Gambar 4.1d Hasil pengolahan data terhadap model SIR dengan populasi 1158 dengan menggunakan nilai $\beta=0.095$ yang diperoleh dari model sebelumnya.

Variasi ketiga menggunakan populasi 1600 agen, dengan jumlah agen awal terinfeksi sebanyak 86 unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit

$\beta=0.19$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 8.5$ Dari perbandingan grafik pada lampiran yaitu gambar A.1.1 dan A.1.2 terlihat pada keadaan ini, model ini belum begitu mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.

Jika model SIR dianggap sebagai acuan, maka *relative squared error*, *root relative squared error*, dan *mean squared error* dari model ini terhadap model SIR untuk variasi jumlah agen ditampilkan pada tabel di bawah ini.

Tabel 4.1a Nilai *root mean squared error* untuk variasi populasi

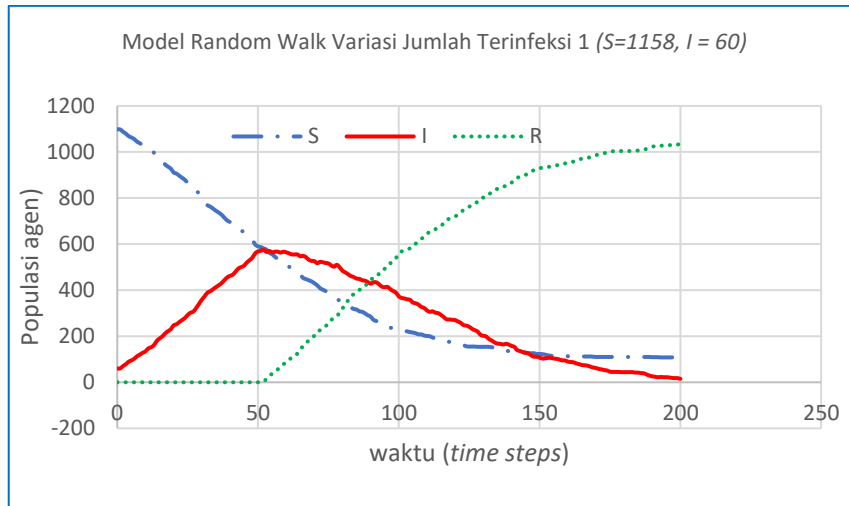
Jumlah Agen = 900	Jumlah Agen = 1158	Jumlah Agen = 1600
Root Mean Squared Error		
174.505	70.14093	310.2954

Tabel 4.1b Nilai *mean squared error* untuk variasi populasi

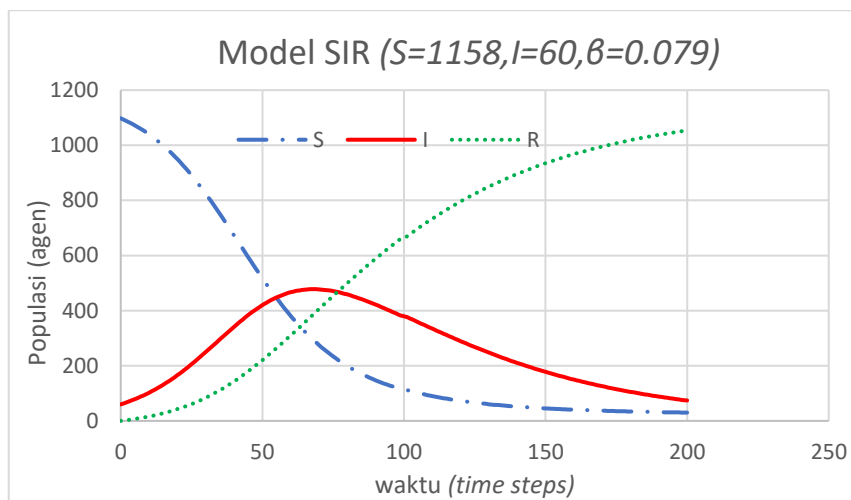
Jumlah Agen = 900	Jumlah Agen = 1158	Jumlah Agen = 1600
Mean Squared Error		
30451.65	4919.75	96283.25

4.2.2 Menentukan Korelasi terhadap Model SIR dengan Variasi Jumlah Terinfeksi

Variasi pertama menggunakan populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 60 unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.079$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 3.395$. Dari perbandingan grafik pada gambar 4.2a dan 4.2b terlihat pada keadaan ini, model ini mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.



Gambar 4.2a Hasil pengolahan data menggunakan model random walk jumlah agen terinfeksi awal 60 yang menghasilkan nilai $\beta=0.079$.



Gambar 4.2b Hasil pengolahan data menggunakan model SIR dengan jumlah agen terinfeksi awal 60 yang menggunakan nilai $\beta=0.079$.

Variasi kedua menggunakan populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 98 unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.089$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 4.45$. Dari perbandingan grafik pada lampiran yaitu gambar A.2.1 dan A.2.2 terlihat pada keadaan ini, model ini belum begitu mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.

Variasi ketiga menggunakan populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 168 unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.11$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 5.5$. Dari perbandingan grafik pada lampiran yaitu gambar A.2.3 dan A.2.4 terlihat pada keadaan ini, model ini belum begitu mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.

Jika model SIR dianggap sebagai acuan, maka *relative squared error*, *root relative squared error*, dan *mean squared error* dari model ini terhadap model SIR untuk variasi jumlah terinfeksi di awal ditampilkan pada tabel di bawah ini.

Tabel 4.2a Nilai *root mean squared error* untuk variasi jumlah terinfeksi.

Jumlah Agen Terinfeksi = 60	Jumlah Agen = 98	Jumlah Agen Terinfeksi = 168
Root Mean Squared Error		
92.65989	61.23194	65.55528

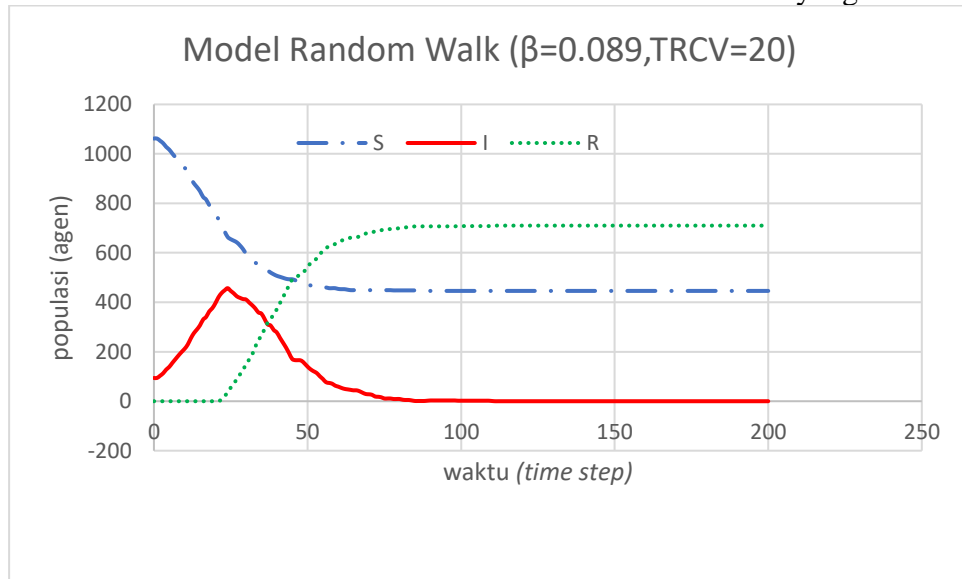
Tabel 4.2b Nilai *mean squared error* untuk variasi jumlah terinfeksi.

Jumlah Agen Terinfeksi = 60	Jumlah Agen = 98	Jumlah Agen Terinfeksi = 168
Mean Squared Error		
8585.855	3749.35	4297.495

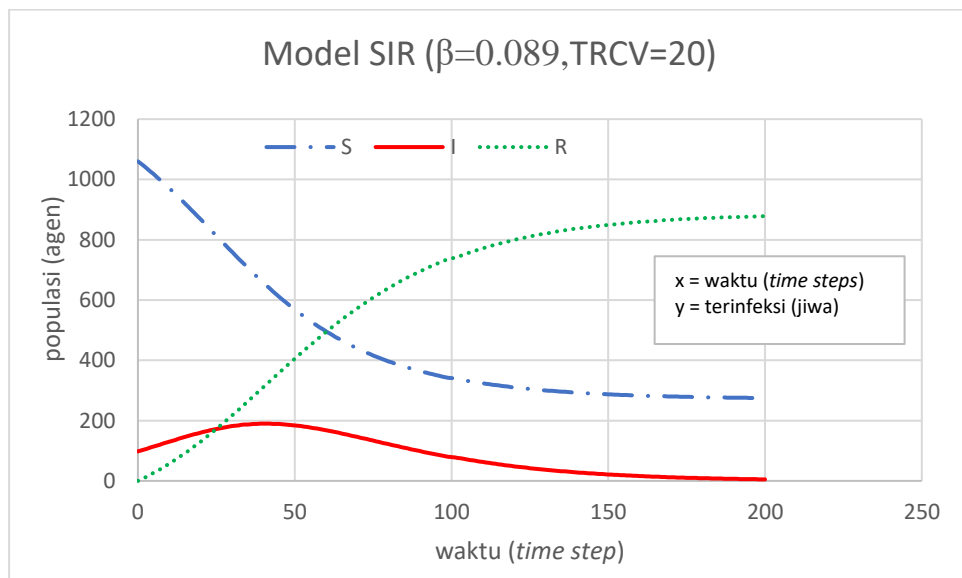
4.2.3 Menentukan Korelasi terhadap Model SIR dengan Variasi Waktu Terinfeksi

Variasi pertama menggunakan waktu agen terinfeksi untuk sembuh selama 20 *time steps* populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 98 unit. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.089$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 1.78$. Dari

perbandingan grafik pada gambar 4.3a dan 4.3b terlihat pada keadaan ini, model ini belum mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.



Gambar 4.3a Hasil pengolahan data menggunakan model random walk dengan $R_0 = 1.78$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



Gambar 4.3b Hasil pengolahan data menggunakan model SIR dengan $R_0 = 1.78$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.

Variasi kedua menggunakan waktu agen terinfeksi untuk sembuh selama 35 *time steps* populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 98 unit. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.089$.

Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 3.1$. Dari perbandingan grafik pada lampiran yaitu gambar A.3.1 dan A.3.2 terlihat pada keadaan ini, model ini mulai mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.

Variasi ketiga menggunakan waktu agen terinfeksi untuk sembuh selama 60 *time steps* populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 98 unit. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.089$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 5.34$. Dari perbandingan grafik pada lampiran yaitu gambar A.3.3 dan A.3.4 terlihat pada keadaan ini, model ini masih mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.

Variasi keempat menggunakan waktu agen terinfeksi untuk sembuh selama 70 *time steps* populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 98 unit. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.089$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 6.23$. Dari perbandingan grafik pada gambar pada lampiran yaitu gambar A.3.5 dan A.3.6 terlihat pada keadaan ini, model ini masih mencerminkan keluaran dari model SIR untuk nilai variabel yang sama meskipun mulai menunjukkan perbedaan yang cukup mencolok.

Variasi kelima menggunakan waktu agen terinfeksi untuk sembuh selama 80 *time steps* populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 98 unit. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.089$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 7.12$. Dari perbandingan grafik pada gambar lampiran yaitu gambar A.3.7 dan A.3.8 terlihat pada keadaan ini, model ini masih mencerminkan meskipun mulai menunjukkan perbedaan yang lebih mencolok dari sebelumnya.

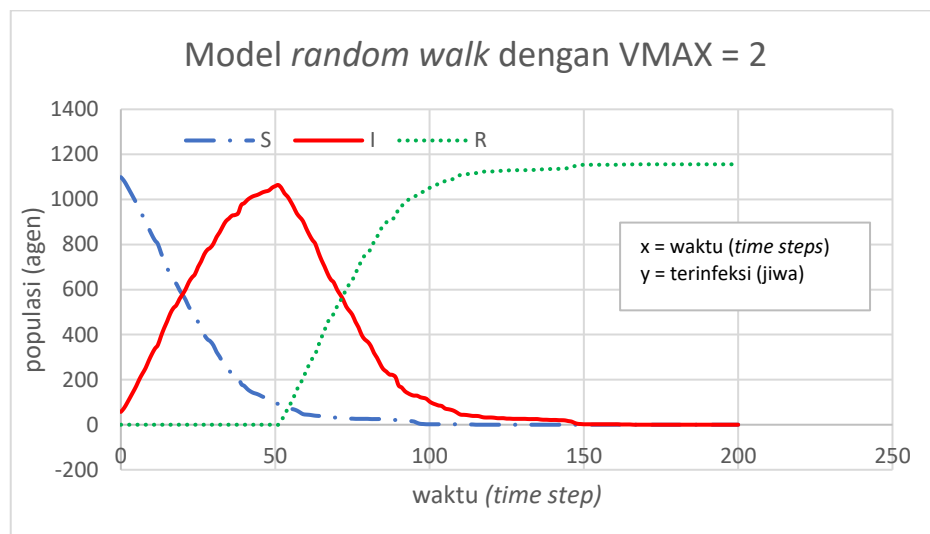
Variasi keenam menggunakan waktu agen terinfeksi untuk sembuh selama 100 *time steps* populasi 1158 agen, dengan jumlah agen awal terinfeksi sebanyak 98 unit. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.089$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 8.9$. Dari perbandingan grafik pada gambar A.3.9 dan A.3.10 terlihat pada keadaan ini, model ini masih mencerminkan meskipun mulai menunjukkan perbedaan yang lebih mencolok dari sebelumnya

Dari variasi waktu agen terinfeksi untuk sembuh ini terlihat bahwa model akan memiliki korelasi relative dekat dengan model SIR pada rentang waktu terinfeksi

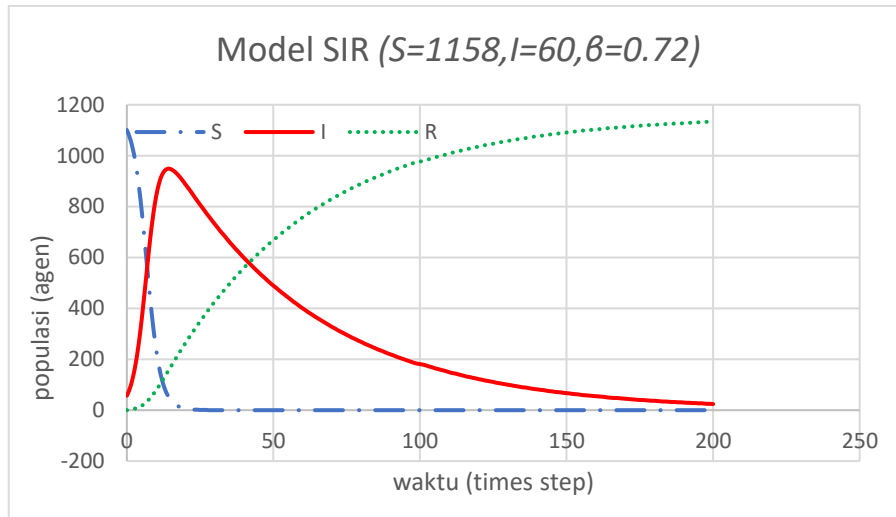
40 – 65 *time steps*, atau dapat dikatakan jika model ini dikalibrasi dengan 1158 populasi dengan jumlah agen awal terinfeksi 8.5% dari populasi, maka rentang nilai R_0 di mana model akan efektif bekerja adalah pada 4.5 – 6. Tentunya rentang ini dapat diperluas dengan mengatur variasi jumlah agen awal terinfeksi ataupun VMAX.

4.2.4 Menentukan Korelasi terhadap Model SIR dengan Variasi Kecepatan

Kalibrasi berikutnya akan menggunakan variasi kecepatan maksimum agen pada simulasi. Akan dilakukan dua variasi yaitu dengan kecepatan maksimum $V_{MAX} = 2$ dan $V_{MAX} = 3$. Pada $V_{MAX}=2$ Variasi ini dilakukan dengan menggunakan populasi 1158 agen, karena simulasi menunjukkan dengan jumlah populasi ini nilai error relatif kecil serta dengan jumlah agen awal terinfeksi sebanyak 60 unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.45$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 22.3$. Dari perbandingan grafik pada gambar 4.1a dan 4.1b terlihat pada keadaan ini, model ini belum begitu mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.



Gambar 4.4a Hasil pengolahan data menggunakan model random walk dengan $V_{MAX} = 2$ yang menghasilkan nilai $\beta=0.45$.



Gambar 4.4b Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai $\beta=0.45$.

Pada VMAX=3 Variasi ini dilakukan dengan menggunakan populasi 1158 agen, karena simulasi menunjukkan dengan jumlah populasi ini nilai error relatif kecil serta dengan jumlah agen awal terinfeksi sebanyak 60 unit. Waktu populasi terinfeksi untuk sembuh diatur menjadi 50 *time steps*. Dari nilai variabel ini diperoleh data pada tabel dalam lampiran. Dengan mengolah data pada tabel diperoleh nilai koefisien transmisi penyakit $\beta=0.72$. Dengan mengambil definisi $R_0 = \frac{\beta}{\gamma}$ maka diperoleh nilai $R_0 = 36$. Dari perbandingan grafik pada gambar A.4.1 dan A.4.2 pada lampiran terlihat pada keadaan ini, model ini belum begitu mencerminkan keluaran dari model SIR untuk nilai variabel yang sama.

Jika model SIR dianggap sebagai acuan, maka *relative squared error*, *root relative squared error*, dan *mean squared error* dari model ini terhadap model SIR untuk variasi kecepatan maksimal ditampilkan pada tabel di bawah ini.

Tabel 4.3a Nilai *root mean squared error* untuk variasi VMAX

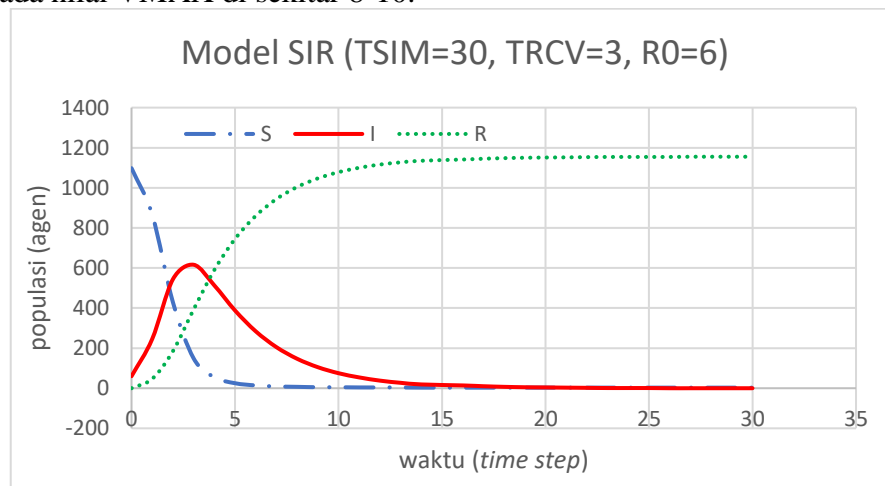
VMAX = 2	VMAX = 3
Root Mean Squared Error	
201.2968	129.3143

Tabel 4.3b Nilai *root mean squared error* untuk variasi VMAX

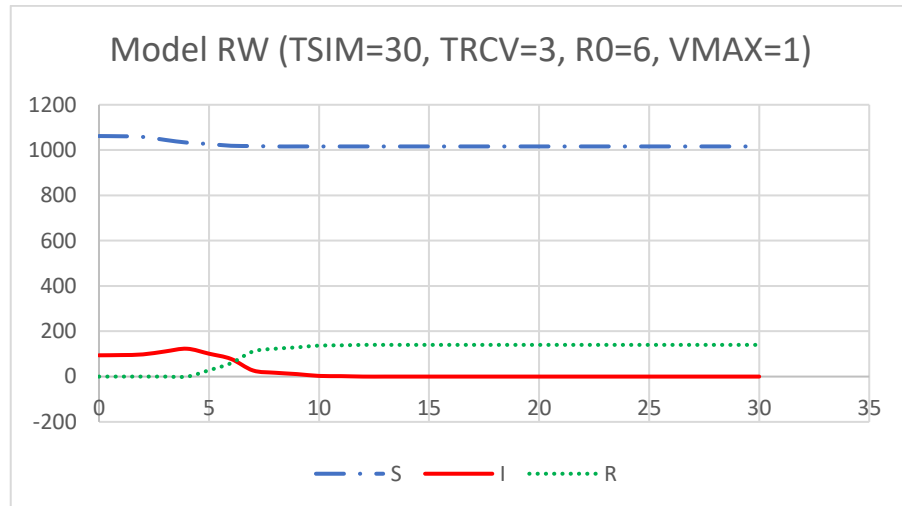
VMAX = 2	VMAX =3
Mean Squared Error	
40520.42	16722.19

4.2.5 Menentukan Rentang Korelasi terhadap Model SIR pada Variasi Kecepatan dan Waktu Simulasi.

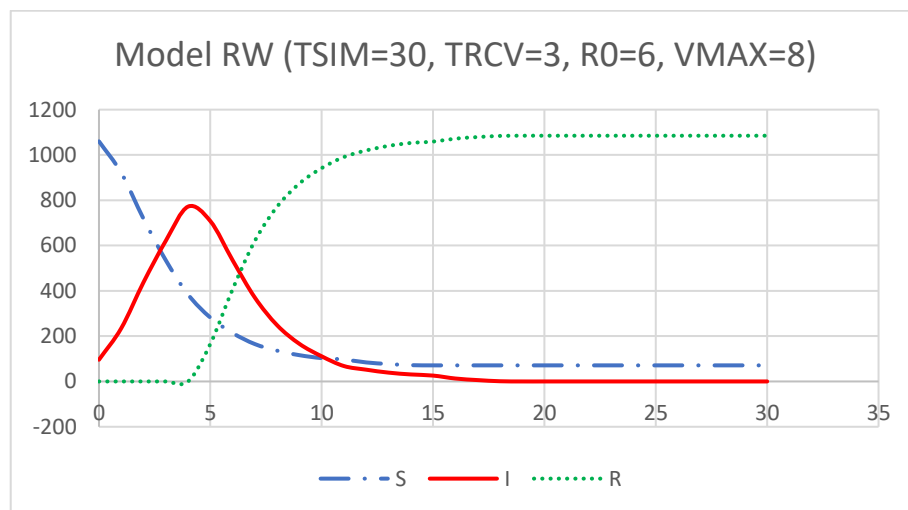
Telah dijelaskan pada bab sebelumnya bahwa waktu simulasi yang singkat dapat dikompensasi dengan menambah VMAX atau sebaliknya. Ini terjadi jika kita misalnya ingin melakukan simulasi dengan parameter sesuai keadaan riil. Misalkan kita ingin melakukan simulasi dengan “hari” sebagai parameter waktu. Biasanya banyak data penyakit menular seperti flu disimulasikan dalam waktu 14-64 hari, dengan waktu agen terinfeksi untuk sembuh 2-5 hari. Waktu ini sangat singkat jika dibandingkan pengaturan standar pada model simulasi ini yaitu dengan 200 *time steps*, sehingga hasil simulasi akan jauh berbeda dengan data ataupun yang diolah dengan model kompartemen SIR. Misalkan kita ingin mengolah suatu data penyakit menular dengan $R_0 = 6$, waktu agen terinfeksi untuk sembuh TRCV = 3 hari, dan waktu simulasi TSIM = 30 hari pada 1158 populasi dengan jumlah agen terinfeksi di awal sebanyak 98 orang, maka hasil simulasi akan relative dekat dengan model SIR pada nilai VMAX di sekitar 8-10.



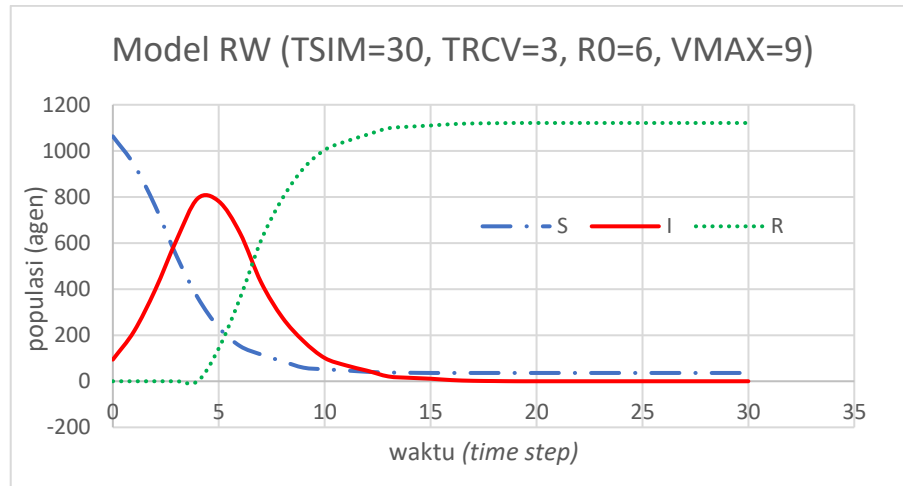
Gambar 4.5a Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai $R_0 = 6$ dan variasi parameter waktu.



Gambar 4.5b Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan VMAX =1.



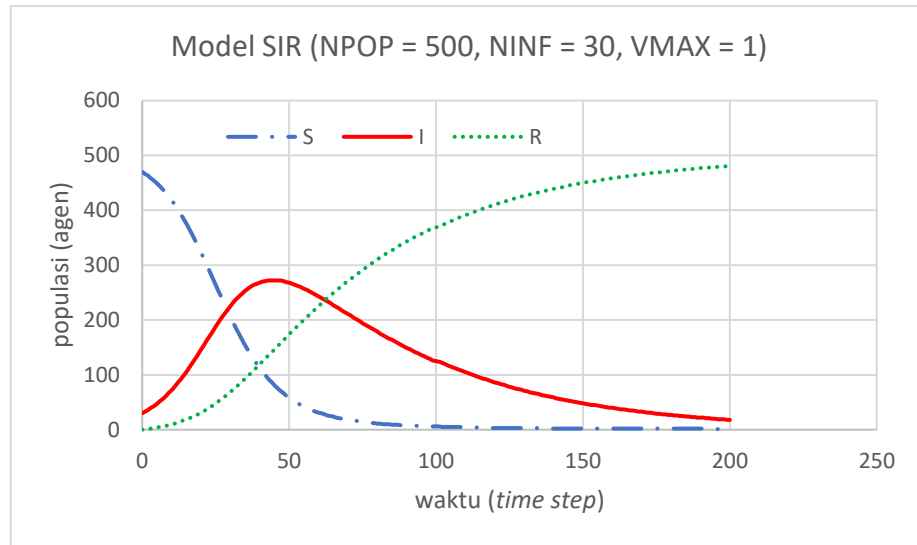
Gambar 4.5c Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan VMAX = 8.



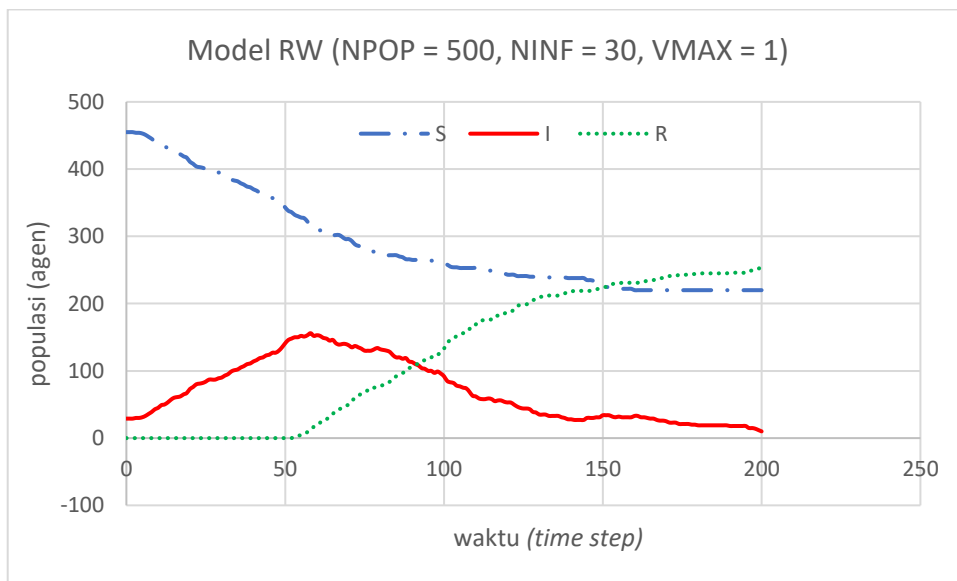
Gambar 4.5d Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan VMAX = 9.

4.2.6 Menentukan Rentang Korelasi terhadap Model SIR pada Variasi Kecepatan dan Jumlah Populasi.

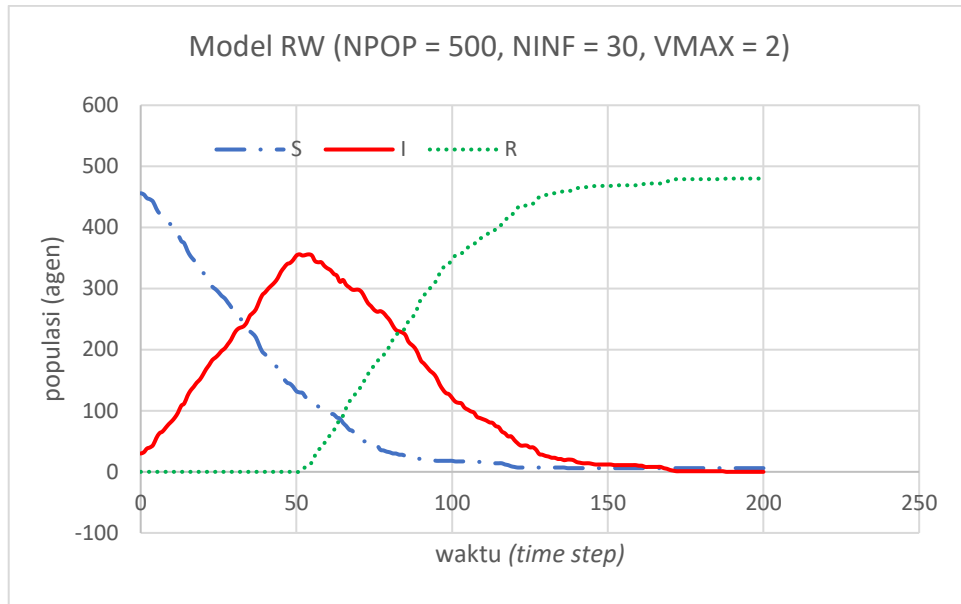
Pada bab sebelumnya telah dijelaskan bahwa mengurangi jumlah populasi pada model ini akan menaikkan jarak rata-rata antar agen akan mengurangi peluang interaksi antar partikel. Hal ini secara matematis juga terlihat dari persamaan 2.2a, 2.2b, dan 2.7. Dari persamaan ini terlihat bahwa nilai absolut suatu jarak berbanding terbalik secara eksponensial dengan peluang suatu agen berada di jarak tersebut pada waktu t . Ini menunjukkan bahwa penambahan jarak akan mengurangi peluang interaksi partikel secara signifikan. Oleh karena itu, agar tetap berkorelasi dengan model kompartemen SIR, maka jika populasi agen dikurangi secara signifikan dari populasi standar yang digunakan (1158), maka VMAX harus dinaikan sebagai kompensasi. Pada subbab berikut akan ditentukan titik korelasi model dengan mencari nilai VMAX yang cocok dengan model SIR untuk jumlah populasi (NPOP) 500 agen, waktu simulasi (TSIM) 200 *time steps*, jumlah agen terinfeksi awal (NINF) 30, waktu agen terinfeksi untuk sembuh (TRCV) 60 *time steps*, serta nilai $R_0 = 6$.



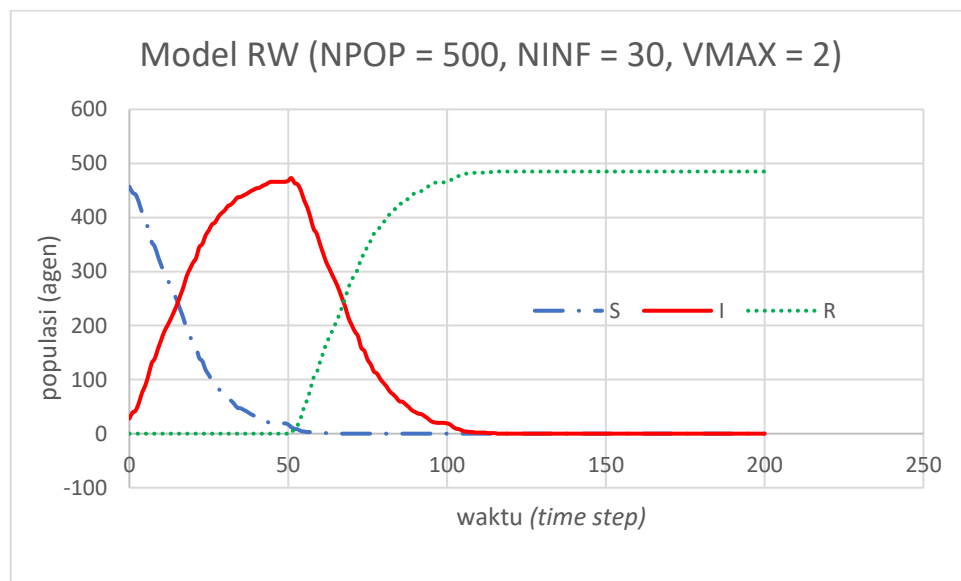
Gambar 4.6a Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi NPOP = 500.



Gambar 4.6b Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi jumlah populasi NPOP = 500 dan VMAX = 1.



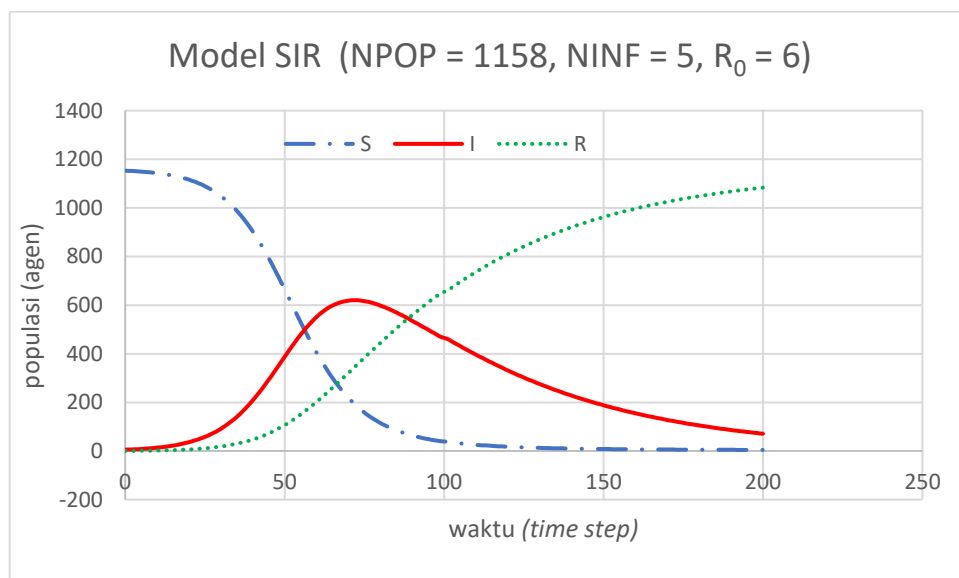
Gambar 4.6c Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi NPOP = 500 dan VMAX = 2.



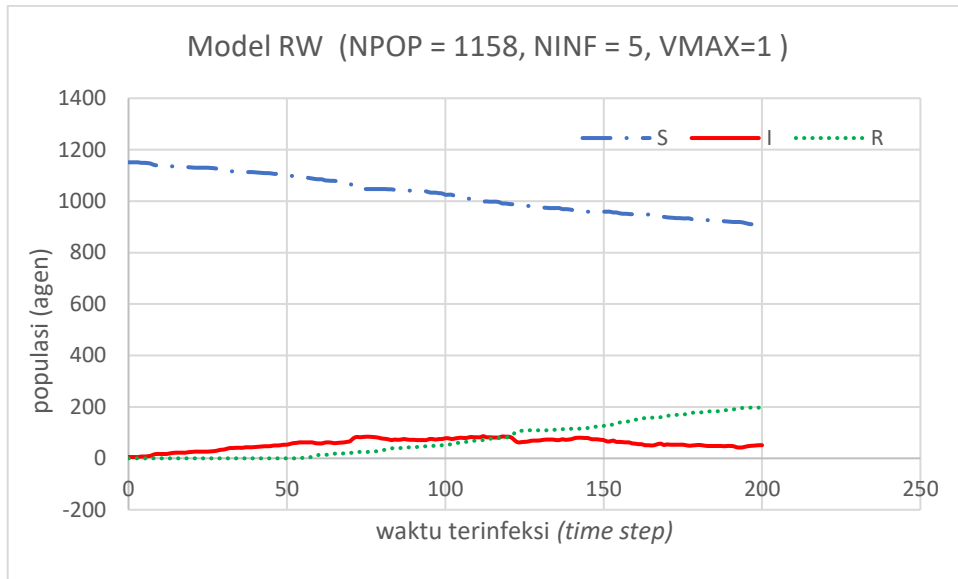
Gambar 4.6d Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi NPOP = 500 dan VMAX = 3.

4.2.7 Menentukan Rentang Korelasi terhadap Model SIR dengan Variasi Kecepatan dan Jumlah Populasi Terinfeksi di Awal.

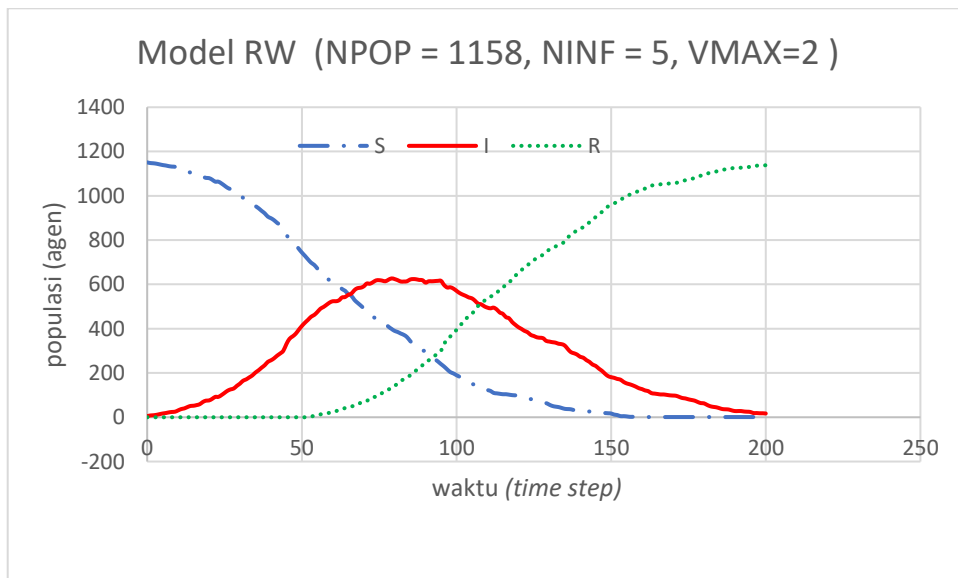
Salah satu perbedaan model berbasis agen dengan model kompartemen adalah bagaimana model berbasis agen sangat dipengaruhi elemen spasial seperti posisi, jarak, dan kecepatan agen. Dalam model SIR, pengaruh-pengaruh ini disederhanakan dalam suatu konstanta β yang menyatakan frekuensi kontak unit terinfeksi dengan unit lainnya. Karena dinyatakan oleh satu konstanta, setiap unit populasi memiliki peluang yang sama untuk melakukan kontak dengan satu unit terinfeksi tanpa dipengaruhi oleh parameter jarak. Telah dijelaskan di bab sebelumnya bahwa pada model ini peluang kontak antar partikel dipengaruhi oleh jarak. Untuk jumlah populasi $NPOP = 1200$, $VMAX = 1$ dan waktu simulasi $TSIM = 200$ *time steps*, nilai peluang partikel di ujung kiri bawah diagonal kotak untuk melakukan kontak dengan partikel di ujung kanan atas diagonal dapat dikatakan sangat kecil. Karena hal ini, untuk jumlah populasi awal terinfeksi yang cukup kecil ($NINF \leq 1\% NPOP$), hasil simulasi model ini akan memperlihatkan perbedaan hasil yang cukup mencolok jika dibandingkan dengan model SIR. Pengurangan jumlah populasi awal terinfeksi juga dapat dikompensasi dengan penambahan nilai $VMAX$, secara intuitif $VMAX$ yang lebih besar akan memperbesar peluang kontak agen dengan agen lainnya dan mengkompensasi ketidakhadiran partikel terinfeksi. Pada subab ini akan ditentukan rentang korelasi model ini terhadap model SIR jika Jumlah populasi awal terinfeksi $NINF = 5$, $NPOP = 1158$, $R_0 = 6$, $TSIM = 200$, dan $TRCV = 50$. Terlihat dari gambar di bawah, model ini menunjukkan korelasi pada model SIR saat nilai $VMAX$ berada dalam rentang 2-4.



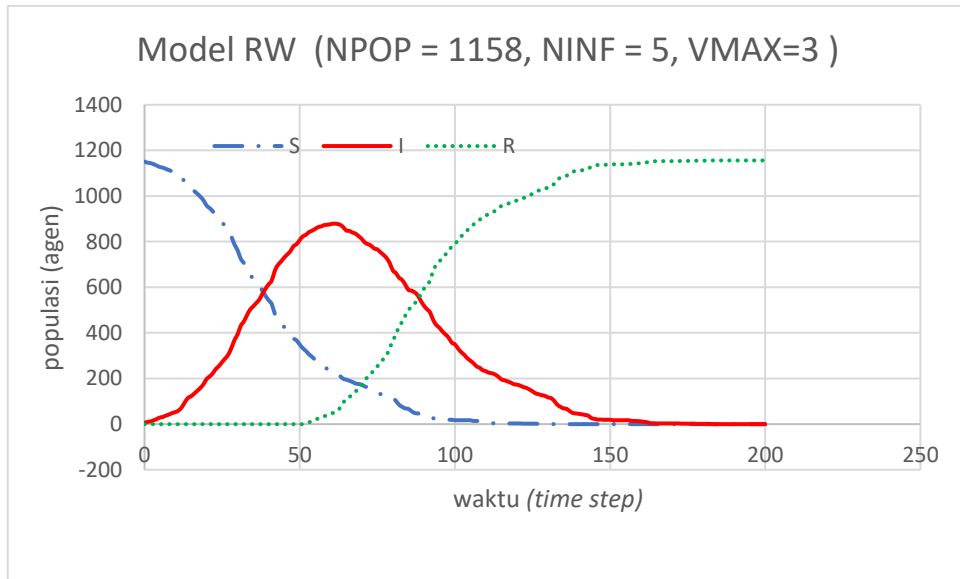
Gambar 4.7a Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai variasi jumlah populasi $NINF = 5$ dan $R_0 = 6$.



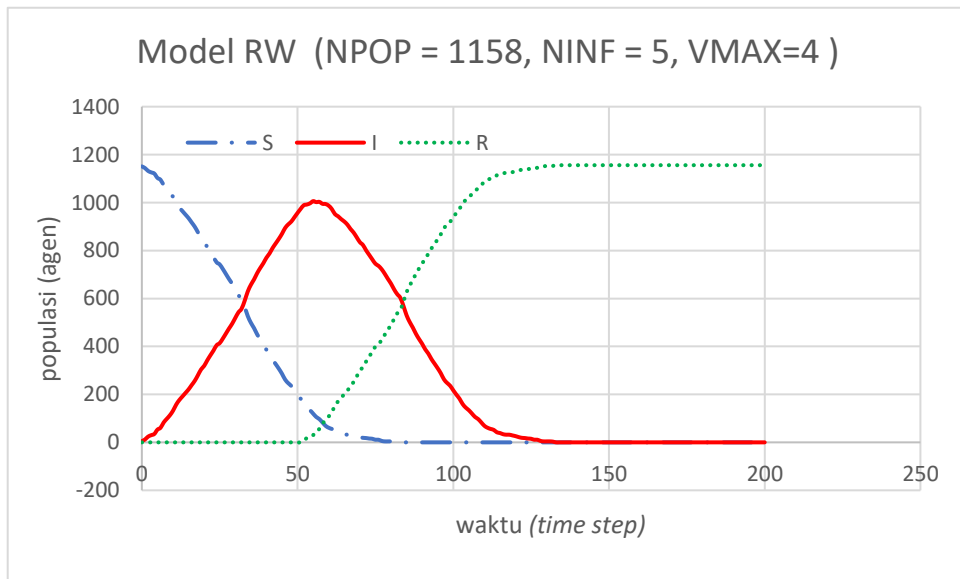
Gambar 4.7b Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi NINF= 5 dan VMAX =1.



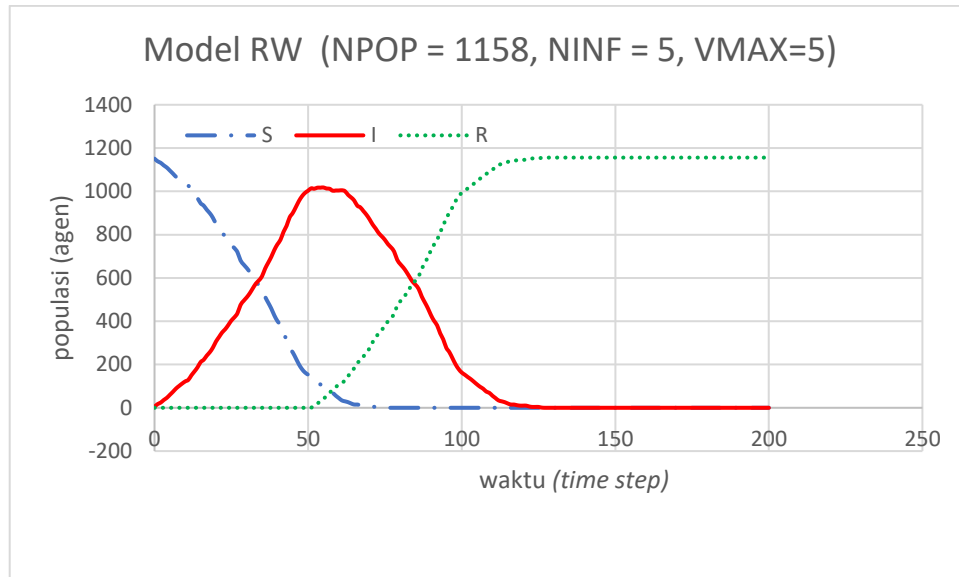
Gambar 4.7c Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi NINF= 5 dan VMAX =2.



Gambar 4.7d Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi NINF= 5 dan VMAX =3.



Gambar 4.7e Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi NINF= 5 dan VMAX =4.



Gambar 4.7f Hasil pengolahan data menggunakan model RW dengan menggunakan nilai variasi jumlah populasi NINF= 5 dan VMAX =5.

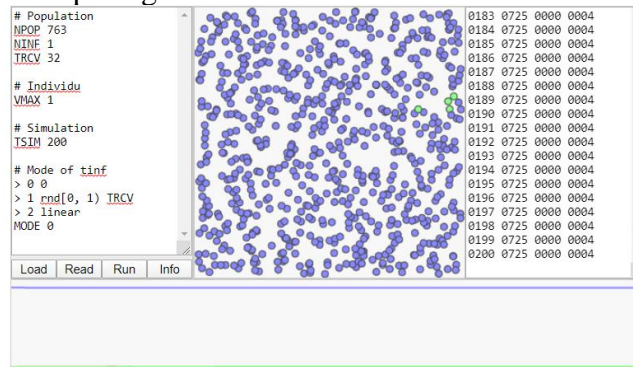
4.3 Validasi Eksternal

Dalam subab ini akan dilakukan pengujian model dengan membandingkan hasil pengolahannya terhadap data nyata. Pengujian akan dilakukan melalui dua metode yaitu menggunakan parameter mentah dan parameter terstandarisasi. Parameter mentah yang dimaksud di sini adalah dengan memasukan jumlah agen sesuai dengan data yang ada pada literatur tanpa diolah sebelumnya. Sedangkan jika parameter telah terstandarisasi berarti bahwa parameter tersebut dimasukan setelah sebelumnya disesuaikan dengan cara kerja model dan kalibrasi pada subab sebelumnya. Karena sulitnya menemukan data dengan parameter yang sesuai, beberapa data diolah dengan penyesuaian terhadap model. Misalnya jumlah agen yang dapat diolah secara efektif oleh program ini sekitar 5000 unit, untuk kasus melibatkan populasi di atas 5000 unit maka perlu dilakukan normalisasi terhadap data. Hal ini juga membuat suatu wabah dengan fraksi jumlah terinfeksi awal di bawah $\frac{1}{5000}$ total populasi akan kurang kurang relevan untuk diolah menggunakan model ini.

4.3.1 Validasi Eksternal dengan menggunakan Parameter Mentah

Validasi eksternal pertama dengan menggunakan parameter mentah menggunakan data dari penyebaran penyakit flu di sebuah sekolah asarman Inggris pada 1978 yang diolah menggunakan model SIR pada Marinov (2014). Di sini, setelah libur musim dingin, sebagian besar dari 763 siswa mengalami flu. Pada hari ke-0 dianggap hanya satu siswa yang terkena flu. Waktu sembuh setelah terpapar flu diperoleh sekitar 2.23 hari. Karena pada model random walk ini perubahan keadaan populasi baru terlihat pada *time step* yang cukup panjang jika dibandingkan dengan waktu pengamatan pada data, maka dilakukan normalisasi terhadap waktu pengamatan agar sesuai dengan model. Dari normalisasi diperoleh untuk 200 *time*

steps maka waktu untuk sembuh pada data eqivalen dengan 32 *time steps*. Hasil pengolahannya terlihat pada gambar di bawah ini.

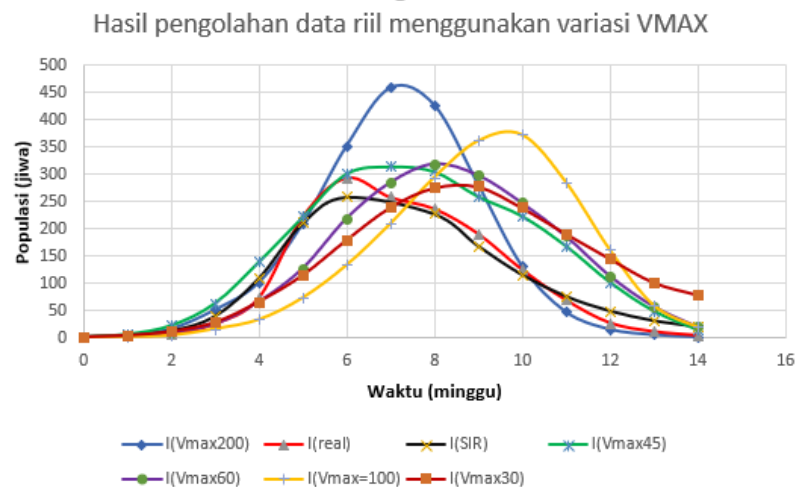


Gambar 4.8 Hasil pengolahan salah satu data mentah pada menggunakan model *random walk*.

Dari gambar tersebut terlihat bahwa flu hanya menyebar pada sedikit populasi sebelum akhirnya “punah” yang mana bertentangan dengan pengamatan dan model yang diberikan. Hasil ini menunjukkan bahwa model tidak bisa digunakan untuk mengolah data mentah. Model yang digunakan untuk mengolah data haruslah yang telah dikalibrasi terlebih dahulu dan data yang tersedia harus distandarisasi secara statistik agar sesuai dengan model terkalibrasi. Hasil pengolahan menggunakan parameter ternormalisasi pada subab berikutnya.

4.3.2 Validasi Eksternal dengan menggunakan model terkalibrasi (wabah influenza di suatu sekolah asrama Inggris)

Pada subab ini akan dilakukan validasi eksternal dengan melakukan variasi kecepatan maksimal agen untuk memperoleh hasil sedekat mungkin dengan data observasi. Variasi dilakukan menggunakan VMAX dari 30 – 100 yang hasilnya ditampilkan pada gambar 4.8 di bawah untuk data penyebaran influenza di sekolah asrama Inggris.



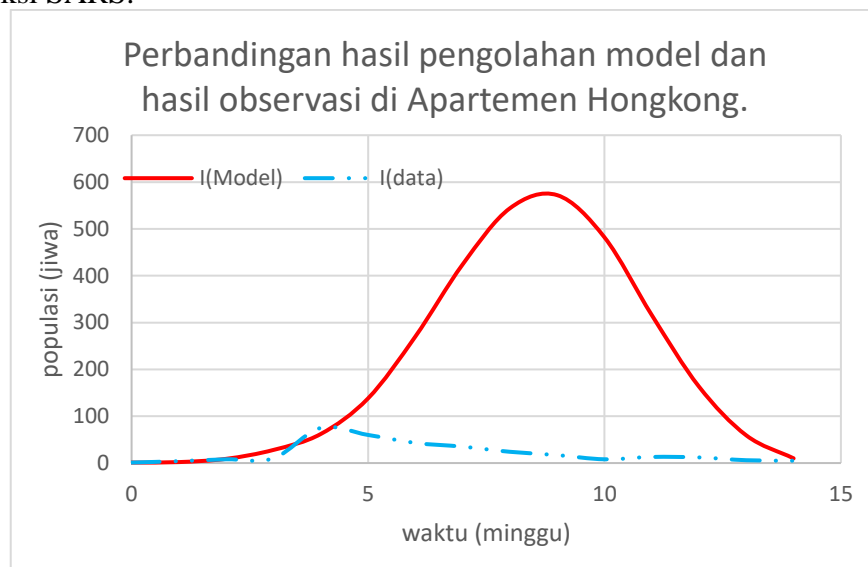
Gambar 4.9 Hasil pengolahan data penyebaran influenza di sekolah asrama dengan parameter VMAX terkalibrasi.

Tabel 4.4 Nilai kesalahan untuk pengolahan data observasi penyebaran influenza pada suatu sekolah asrama di Inggris dari tiap model beserta variasinya.

Model dan Variasi	MSE	RMSE
SIR	1208	34.766
RW(VMAX30)	5954	77.167
RW(VMAX45)	3058	55.300
RW(VMAX60)	4795.1	69.246
RW(VMAX100)	14167	119.02
RW(VMAX200)	6026.0	77.630

4.3.3 Validasi Eksternal dengan menggunakan model terkalibrasi (SARS di apartement Amoy Garden Hong Kong)

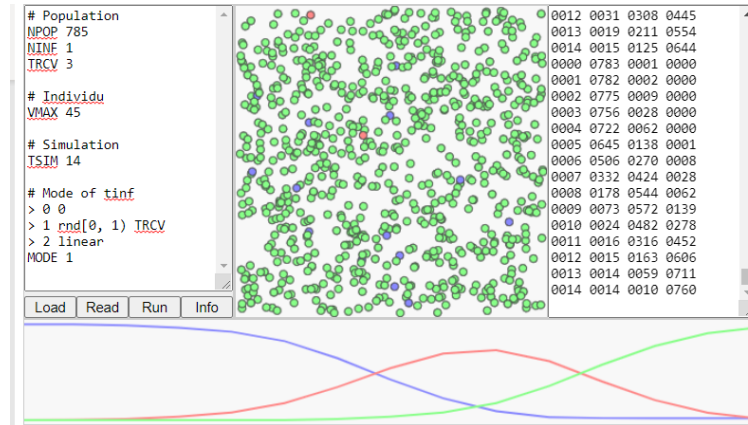
Pada tahun 2003, terjadi wabah SARS di Hong Kong. Pada subab ini akan dilakukan pengolahan data secara spesifik pada salah satu apartemen di Hongkong. Selama 20 hari di bulan Maret-April, sejumlah 321 dari 788 penghuni dinyatakan terinfeksi SARS.



Gambar 4.10a Hasil pengolahan data SARS di Hong Kong dengan parameter VMAX terkalibrasi

Dari model ini terlihat bahwa hasil keluaran model sangat menyimpang dari data observasi. Ini terjadi karena di hari ke-empat, pemerintah kota Hong Kong melakukan karantina terhadap penghuni apartemen sehingga peluang interaksi antar individu di apartemen tersebut berkurang secara signifikan. Sementara setelah hari ke empat pada model simulasi ini, jumlah agen yang berinteraksi tetap sehingga hasil yang digambarkan dari keluaran ini adalah hasil simulasi jika pemerintah kota seandainya tidak melakukan karantina terhadap penghuni apartemen. Jumlah pasien yang tercatat setelah karantina merupakan pasien yang telah terekspos namun belum menunjukkan gejala terinfeksi. Total jumlah akhir terinfeksi yang diperoleh dari model *random walk* ini secara rata-rata sebanyak 760 jiwa, model referensi (Mkhatshwa dan Mummert, 2010) yang digunakan menunjukkan total terinfeksi di akhir sebanyak 766 jiwa atau hampir seluruh penghuni akan terinfeksi. Model pada

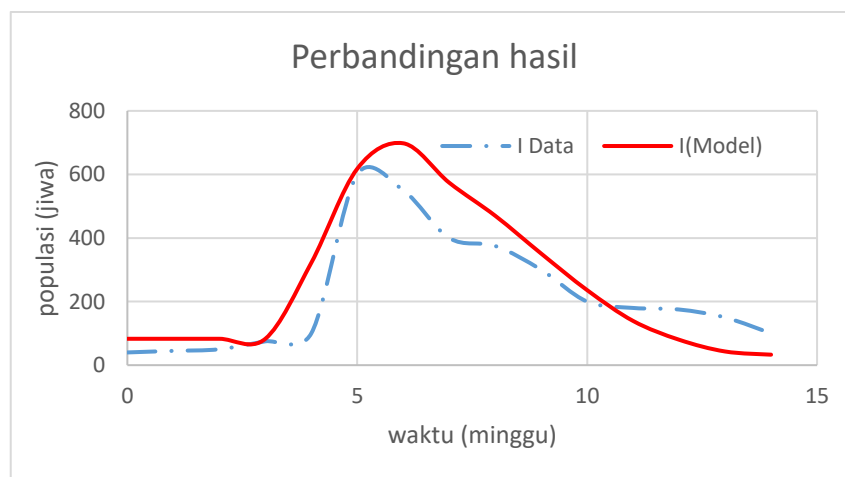
referensi menggunakan model SIR yang dimodifikasi dengan menambah kompartemen P sebagai *superspreader*. *Superspreader* merupakan individu yang dianggap mampu mentransmisikan suatu penyakit ke lebih banyak individu jika dibandingkan individu terinfeksi biasa.



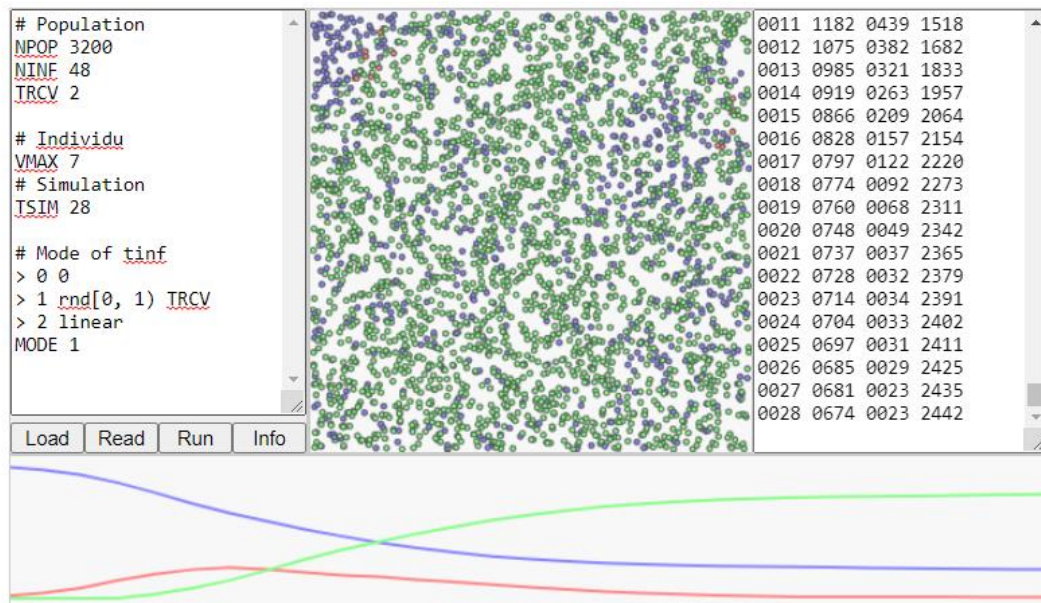
Gambar 4.10b Hasil keluaran salah satu simulasi pada program terkait penyebaran SARS.

4.3.4 Validasi Eksternal dengan menggunakan model terkalibrasi (Demam berdarah Sulawesi Selatan)

Pada subab ini akan diolah data penyebaran penyakit demam berdarah di Sulawesi Selatan pada 2008 (Side dan Noorani, 2013), fokus pengolahan dilakukan pada periode dua minggu di mana dalam rentang tersebut angka kenaikan pasien terinfeksi bertambah signifikan. Dalam rentang ini tercatat lebih dari 3100 jiwa menderita demam berdarah. Karena keterbatasan model untuk mengolah keseluruhan populasi, pada model ini digunakan keseluruhan populasi terinfeksi sebagai batas pengolahan. Hasil yang diperoleh merupakan distribusi dari tercatatnya penderita demam berdarah dalam tiap rentang waktu.



Gambar 4.11a Hasil simulasi demam berdarah dengan parameter VMAX dan waktu terkalibrasi



Gambar 4.11b Salah satu hasil pengolahan data demam berdarah dengan parameter VMAX dan waktu terkalibrasi

Pada umumnya, model kompartemen penyebaran wabah demam berdarah dilakukan dengan menggunakan model SIR dengan memperhitungkan populasi nyamuk sebagai vektor transmisi penyakit. Begitu pula pada referensi yang digunakan sebagai sumber data (Side dan Noorani, 2013). Pada hasil pengolahan model *random walk* di atas, keterbatasan jumlah agen pada model yang dapat diolah mengakibatkan perhitungan ini tidak melibatkan populasi nyamuk dan hanya menghitung bagaimana populasi terinfeksi demam berdarah di Sulawesi Selatan bertambah pada suatu rentang waktu.

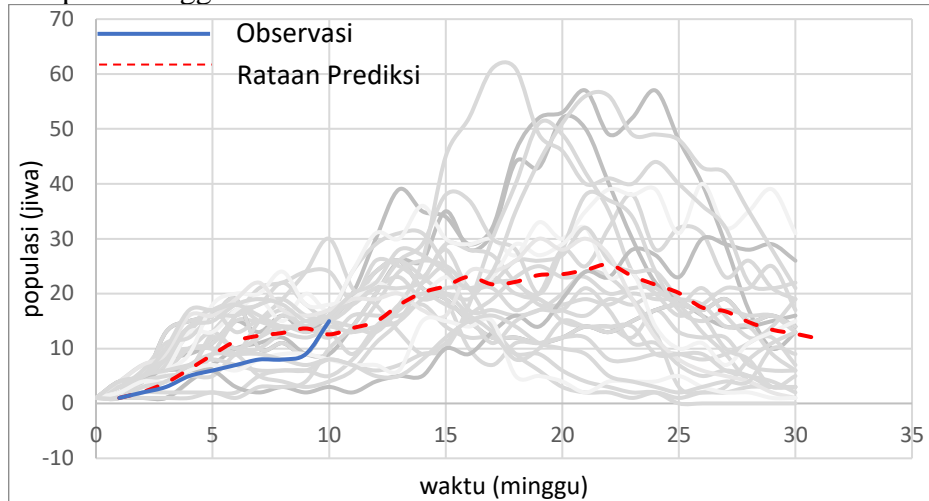
Tabel 4.5 Nilai kesalahan untuk pengolahan data observasi pertambahan populasi terinfeksi demam berdarah di Sulawesi Selatan.

MSE	RMSE
9720	98.590

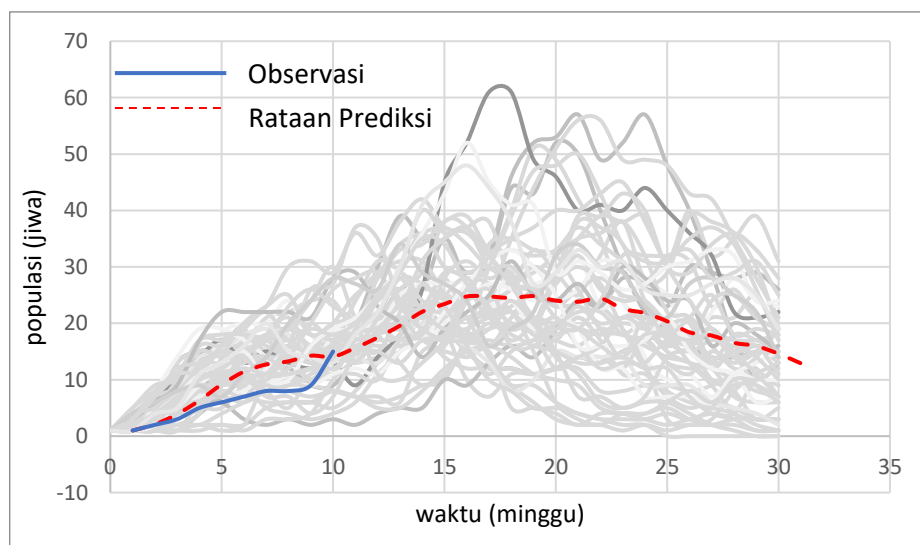
4.3.5 Prediksi jumlah populasi terinfeksi Covid-19 di Kecamatan Coblong Kota Bandung.

Pada subab berikut akan dilakukan prediksi terhadap pertambahan jumlah kasus aktif (populasi terinfeksi) COVID-19 di Kecamatan Coblong Kota Bandung selama 30 minggu yaitu dari awal April 2019 hingga awal tahun 2021. Pada prediksi ini dianggap jumlah terinfeksi maksimal dari suatu populasi adalah sekitar 0.5%. Besaran ini diambil berdasarkan perbandingan terinfeksi per populasi di negara-negara Eropa yang notabene memiliki pencatatan kasus yang baik dan dapat dianggap telah berada di fasa akhir gelombang pertama pandemi COVID-19.

Karena model bersifat stokastik, diperlukan pengambilan data berulang dari hasil keluaran simulasi. Gambar 4.10a menunjukkan hasil prediksi berdasarkan 25 kali pengambilan data dan gambar 4.10b menunjukkan hasil pengolahan untuk 50 kali pengambilan data. Dari hasil pengolahan ini terlihat bahwa tanggal 15 Juni 2020 yang masuk dalam minggu ke sepuluh masih masuk dalam fase awal pertambahan jumlah terinfeksi. Prediksi baru menampilkan penurunan jumlah populasi pada minggu ke 22-25.



Gambar 4.12a Hasil prediksi jumlah kasus aktif COVID-19 untuk 25 kali simulasi di Kecamatan Coblong dengan dengan parameter VMAX dan waktu terkalibrasi



Gambar 4.12b Hasil prediksi jumlah kasus aktif COVID-19 untuk 50 kali simulasi di Kecamatan Coblong dengan parameter VMAX dan waktu terkalibrasi

Tabel 4.6 Tabel prediksi beberapa besaran pada penyebaran wabah COVID-19 di Kecamatan Coblong

Total terinfeksi (jiwa)	Terinfeksi maksimal (jiwa)	Periode penambahan (minggu)	Periode Penurunan (minggu)	Puncak (minggu)
127	25	0-18	22 ke atas	18-22

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

1. Dalam model *random walk* ini, jarak sangat mempengaruhi peluang interaksi partikel; penambahan jarak akan mengecilkan peluang interaksi secara kuadrat eksponensial, perubahan variabel yang mengakibatkan berubahnya jarak antar agen seperti pengurangan dan penambahan populasi ataupun jumlah awal agen terinfeksi harus dilakukan secara hati-hati jika ingin dijaga agar tetap relevan dengan model epidemi lainnya.
2. Waktu simulasi yang singkat, jumlah agen total ataupun terinfeksi yang sedikit pada observasi dapat dikompensasi dengan penambahan laju kecepatan agen (VMAX) yang secara analitis memperbesar peluang interaksi partikel secara kuadrat eksponensial dalam model.
3. Secara keseluruhan dari hasil pengolahan data terlihat bahwa jumlah populasi terinfeksi dari hasil simulasi berada di atas data observasi, ini terjadi karena setelah terinfeksi agen tetap bergerak dengan kecepatan yang sama. Sementara di dunia nyata terinfeksi individu dapat memperlambat gerak individu tersebut bahkan membuat geraknya terbatas sama sekali akibat dirawat di rumah sakit atau dikarantina.

5.2 Saran

Seperti umumnya model berbasis agen, model ini kesulitan untuk memperoleh data dengan parameter yang sesuai sebagai validasi eksternal. Untuk memperluas cakupan validasi tersebut, kedepannya program perlu diatur dan juga diperlukan perangkat komputasi yang lebih mumpuni agar tetap efektif mengolah perhitungan dengan jumlah agen di atas 5000 unit.

Sensitivitas program juga bisa ditambah dengan menjadikan perubahan karakteristik dari agen saat berinteraksi menjadi suatu variabel stokastik. Pada model ini agen yang rentan akan langsung menjadi terinfeksi saat berinteraksi dengan agen terinfeksi. Variabel stokastik ini juga berguna untuk memperluas penggunaan model, seperti misalnya pada kasus di mana terdapat suatu *superspreader* dan individu terekspos yang belum menunjukkan gejala terinfeksi seperti yang telah dibahas pada Subab 4.3.

Sebagai model berbasis agen, model ini dipengaruhi berbagai ketidakpastian, oleh karena itu untuk melakukan prediksi yang relevan secara statistik, data dari model ini harus diambil berulang kali. Pada tugas akhir ini pengulangan dilakukan dalam rentang 25-50 kali akibat keterbatasan perangkat komputasi.

DAFTAR PUSTAKA


- Auchincloss, A. H., & Garcia, L. M. T. (2015). Brief introductory guide to agent-based modeling and an illustration from urban health research. *Cadernos de saude publica*, 31, 65-78.
- Badan Pusat Statistik Kota Bandung (2016); *Statistik Daerah Kecamatan Coblong*. Ainan Dhinan M, Badan Pusat Statistik Kota Bandung.
- Brauer, F. (2008). Compartmental models in epidemiology. In *Mathematical epidemiology* (pp. 19-79). Springer, Berlin, Heidelberg.
- Codling, E. A., Plank, M. J., & Benhamou, S. (2008). Random walk models in biology. *Journal of the Royal Society Interface*, 5(25), 813-834.
- Duncan, C. J., & Scott, S. (2005). What caused the black death?. *Postgraduate medical journal*, 81(955), 315-320..
- Duran, Jacques (2000). *Sands, Powders, and Grains: An Introduction to the Physics of Granular Materials*. Partially Ordered Systems (dalam bahasa Inggris). New York: Springer-Verlag. ISBN 978-0-387-98656-2.
- Guo, D., Li, K. C., Peters, T. R., Snively, B. M., Poehling, K. A., & Zhou, X. (2015). Multi-scale modeling for the transmission of influenza and the evaluation of interventions toward it. *Scientific reports*, 5, 8980.
- Gandon, S., Day, T., Metcalf, C. J. E., & Grenfell, B. T. (2016). Forecasting epidemiological and evolutionary dynamics of infectious diseases. *Trends in ecology & evolution*, 31(10), 776-788.
- Hackl, J., & Dubernet, T. (2019). Epidemic spreading in urban areas using agent-based transportation models. *Future Internet*, 11(4), 92.
- Kareiva, P. M. Shigesada (1983) "Analyzing insect movement as a correlated random walk." *Oecologia*56: 234-238.
- Khalil, Khaled M.; Abdel-Aziz, M.; Nazmy, Taymour T.; Salem, Abdel-Badeeh M. (2012). Lu, Jie; Jain, Lakhmi C.; Zhang, Guangquan, ed. *Handbook on Decision Making: Vol 2: Risk Management in Decision Making*. Intelligent Systems Reference Library (dalam bahasa Inggris). Berlin, Heidelberg: Springer. hlm. 205–218. doi:10.1007/978-3-642-25755-1_11. ISBN 978-3-642-25755-1.
- Littman, Robert J. "The plague of Athens: epidemiology and paleopathology." *Mount Sinai Journal of Medicine: A Journal of Translational and*

Personalized Medicine: *A Journal of Translational and Personalized Medicine* 76.5 (2009): 456-467.

Marinov, Tchavdar T.; Marinova, Rossitza S.; Omojola, Joe; Jackson, Michael (2014-07-01). Inverse problem for coefficient identification in SIR epidemic models (dalam bahasa Inggris). *67* (12): *Computers & Mathematics with Applications*. 2218–2227. doi:10.1016/j.camwa.2014.02.002. ISSN 0898-1221.

Miksch, F., Urach, C., Einzinger, P., & Zauner, G. (2014, April). A Flexible Agent-Based Framework for Infectious Disease Modeling. In *Information and Communication Technology-EurAsia Conference* (pp. 36-45). Springer, Berlin, Heidelberg..

Mkhatshwa, Theminkosi, and Anna Mummert. "Modeling super-spreading events for infectious diseases: case study SARS." *arXiv preprint arXiv:1007.0908* (2010).

Perez, Liliana; Dragicevic, Suzana (2009). "An agent-based approach for modeling dynamics of contagious disease spread". *International Journal of Health Geographics* (dalam bahasa Inggris). *8* (1): 50. doi:10.1186/1476-072X-8-50. ISSN 1476-072X. PMC 2729742 . PMID 19656403.

Pettarin, Alberto; Pietracaprina, Andrea; Pucci, Geppino; Upfal, Eli (2011-01-25). "Infectious Random Walks". Library (dalam bahasa Inggris). *arXiv:1007.1604 [cs]*.

Pu, Cunlai; Li, Siyuan; Yang, Jian (2015-08-15). "Epidemic spreading driven by biased random walks". *Physica A: Statistical Mechanics and its Applications* (dalam bahasa Inggris). *432*: 230–239. doi:10.1016/j.physa.2015.03.035. ISSN 0378-4371.

Renshaw, E., & Henderson, R. (1981). The correlated random walk. *Journal of Applied Probability*, 18(2), 403-414.

Riley, Steven; Eames, Ken; Isham, Valerie; Mollison, Denis; Trapman, Pieter (2015-03-01). "Five challenges for spatial epidemic models". *Epidemics. Challenges in Modelling Infectious Disease Dynamics* (dalam bahasa Inggris). *10*: 68–71. doi:10.1016/j.epidem.2014.07.001. ISSN 1755-4365. PMC 4383807 . PMID 25843387.

Shiflet, Angela B.; Shiflet, George W. (2014-01-01). "An Introduction to Agent-based Modeling for Undergraduates". *Procedia Computer Science*. 2014 International Conference on Computational Science (dalam bahasa

Inggris). **29**: 1392–1402. doi:10.1016/j.procs.2014.05.126. ISSN 1877-050

Side, Syafruddin & Noorani, Moh. (2015). A SIR model for spread of dengue fever disease (simulation for south sulawesi Indonesia and selangor Malaysia). 10.13140/RG.2.1.5042.6721.

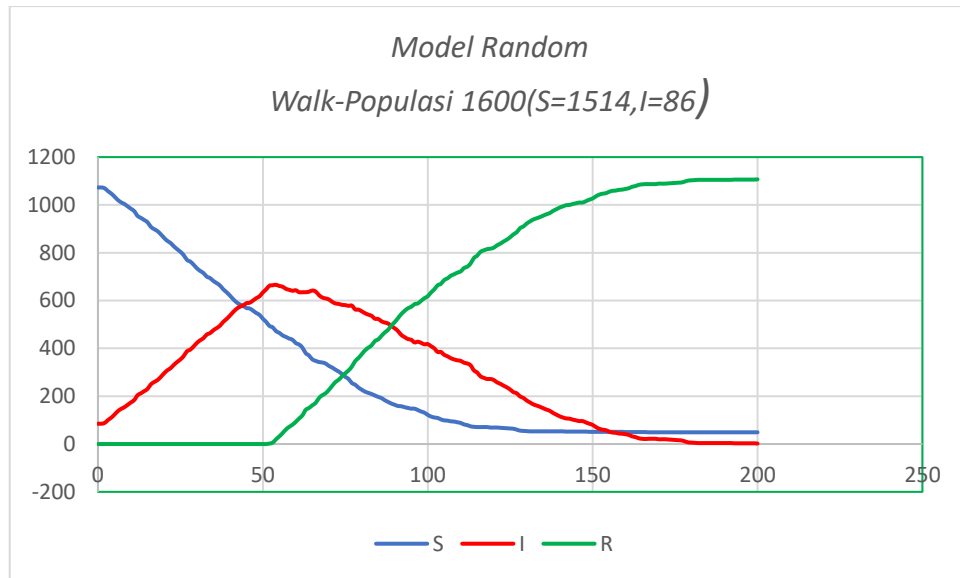
Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health: current applications and future directions. *Annual review of public health*, 39, 77-94.

Venkatramanan, S., Lewis, B., Chen, J., Higdon, D., Vullikanti, A., & Marathe, M. (2018). Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics*, 22, 43-49

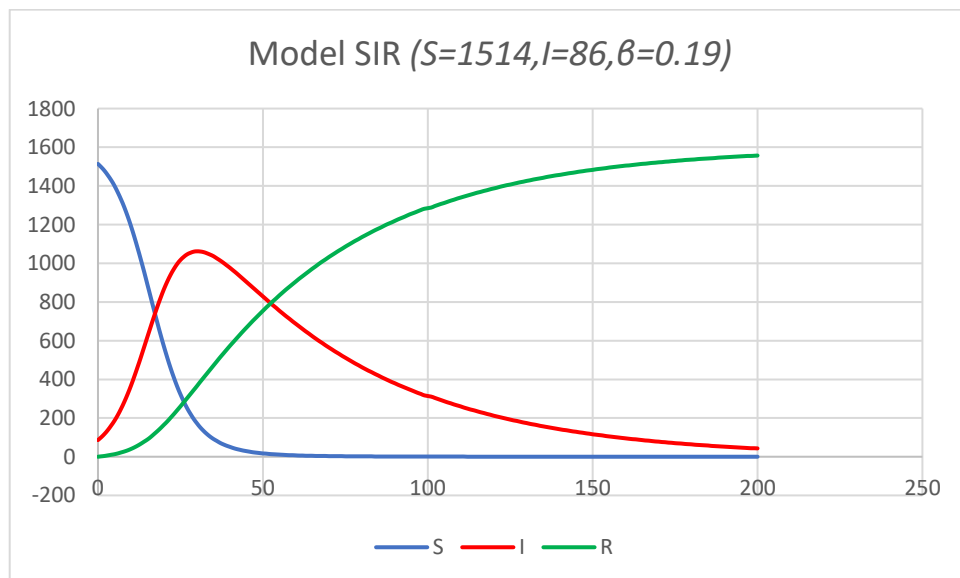
LAMPIRAN

Lampiran A Grafik Hasil Pengolahan Data Simulasi

A.1 Korelasi terhadap Model SIR dengan Variasi Populasi

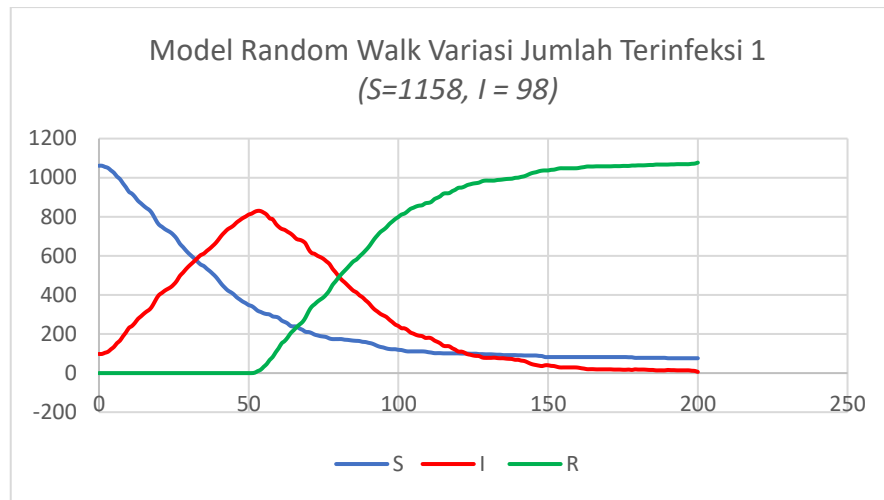


Gambar A.1.1 Hasil pengolahan data terhadap model random walk dengan populasi 1600 yang menghasilkan nilai $\beta=0.19$.

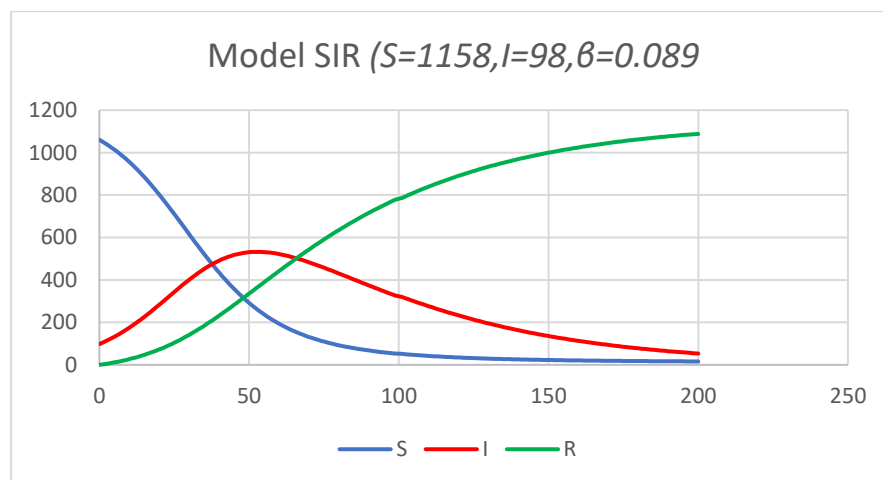


Gambar A.1.2 Hasil pengolahan data terhadap model SIR dengan populasi 1600 yang menghasilkan nilai $\beta=0.19$.

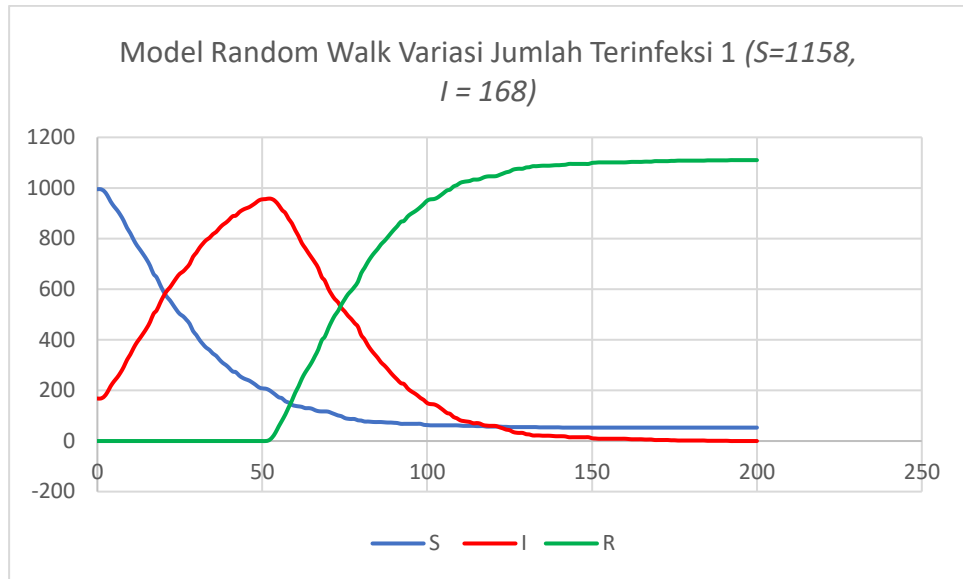
A.2 Korelasi terhadap Model SIR dengan Variasi Jumlah Terinfeksi



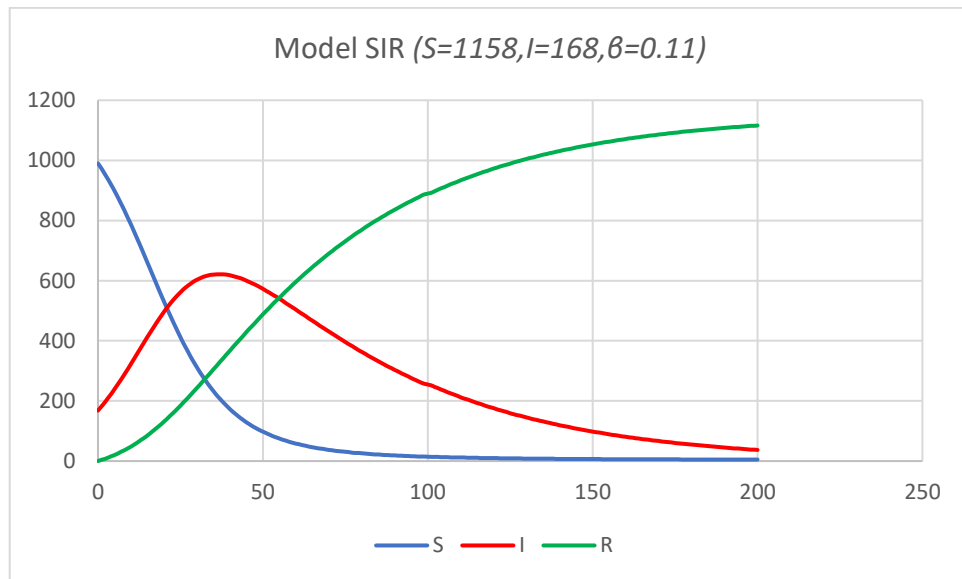
Gambar A.2.1 Hasil pengolahan data menggunakan model random walk jumlah agen terinfeksi awal 98 yang menghasilkan nilai $\beta=0.089$.



Gambar A.2.2 Hasil pengolahan data menggunakan model SIR dengan jumlah agen terinfeksi awal 98 yang menggunakan nilai $\beta=0.089$.

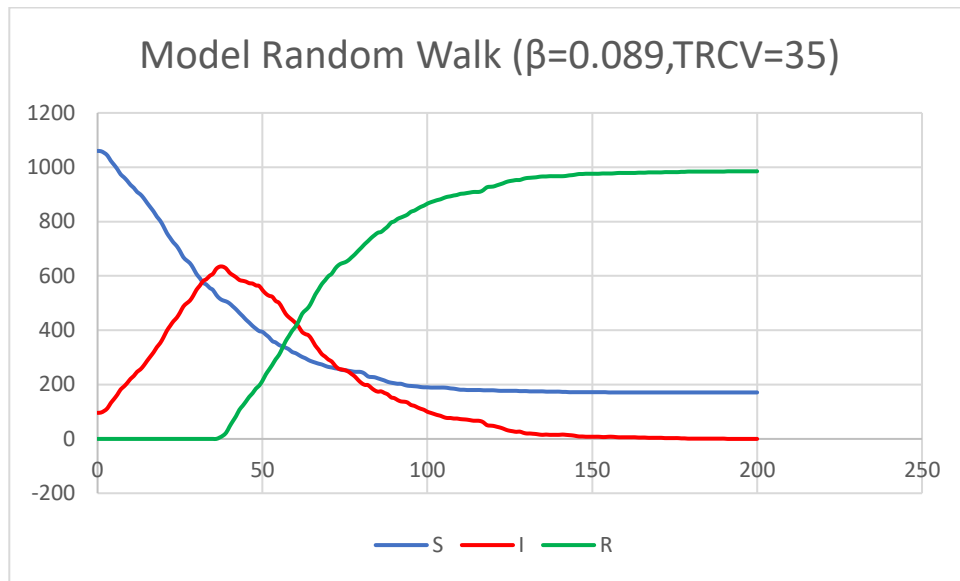


Gambar A.2.3 Hasil pengolahan data menggunakan model random walk jumlah agen terinfeksi awal 168 yang menghasilkan nilai $\beta=0.11$.

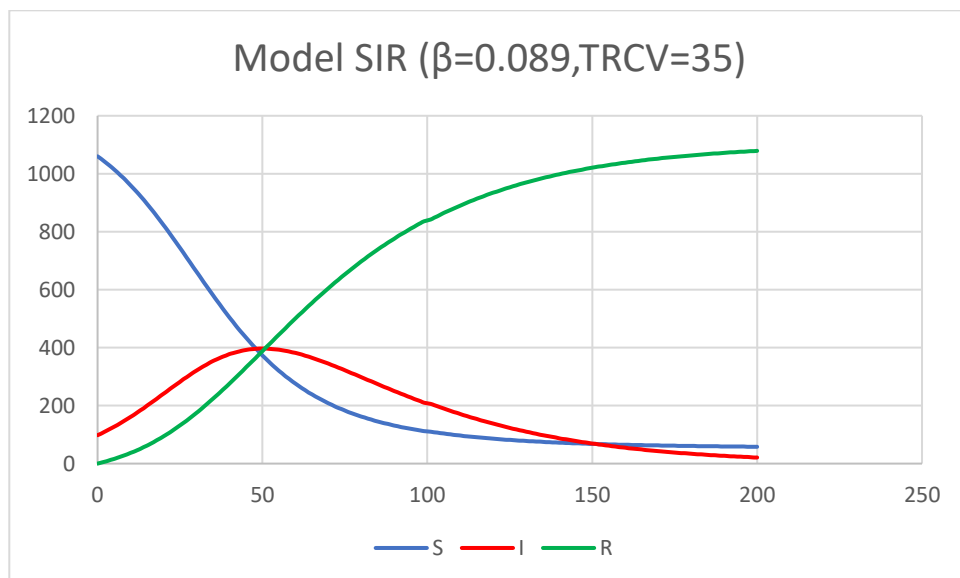


Gambar A.2.4 Hasil pengolahan data menggunakan model SIR dengan jumlah agen terinfeksi awal 168 yang menggunakan nilai $\beta=0.11$.

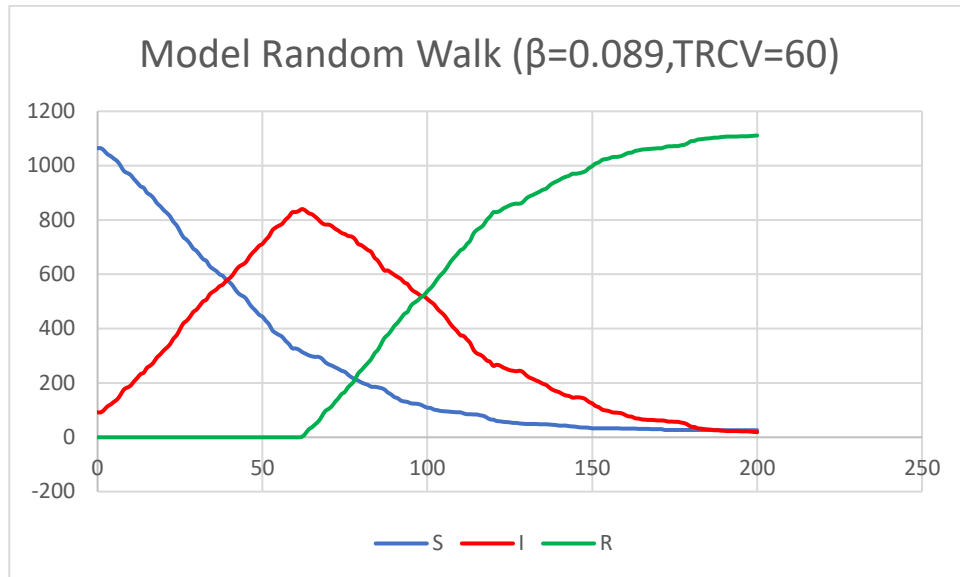
A.3 Korelasi terhadap Model SIR dengan Variasi Waktu Terinfeksi



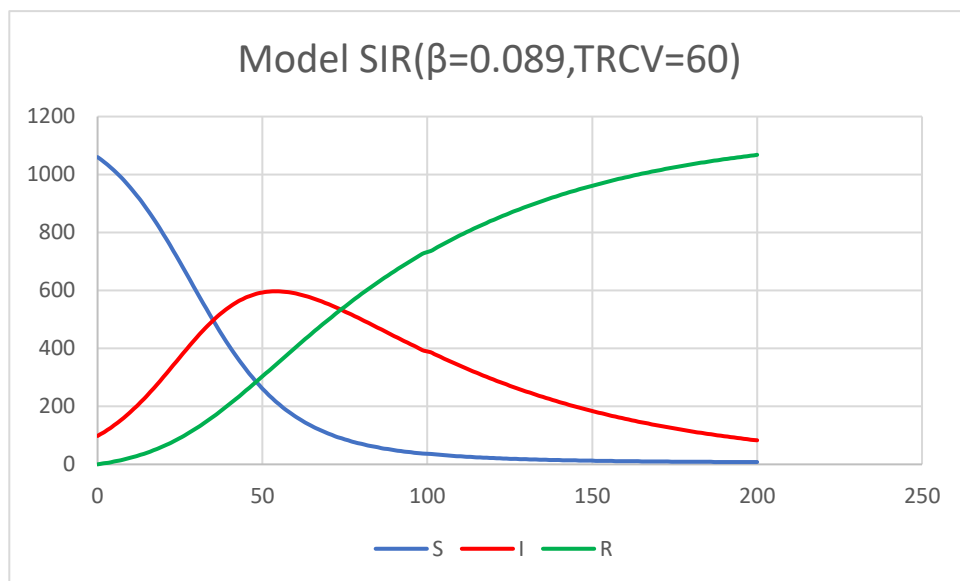
Gambar A.3.1 Hasil pengolahan data menggunakan model random walk dengan $R_0 = 3.10$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



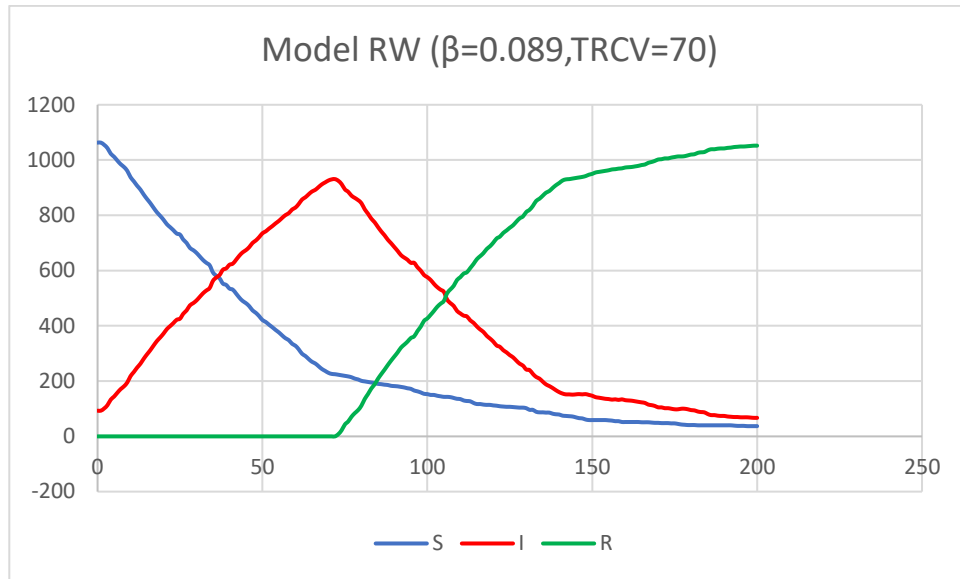
Gambar A.3.2 Hasil pengolahan data menggunakan model SIR dengan $R_0 = 3.10$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



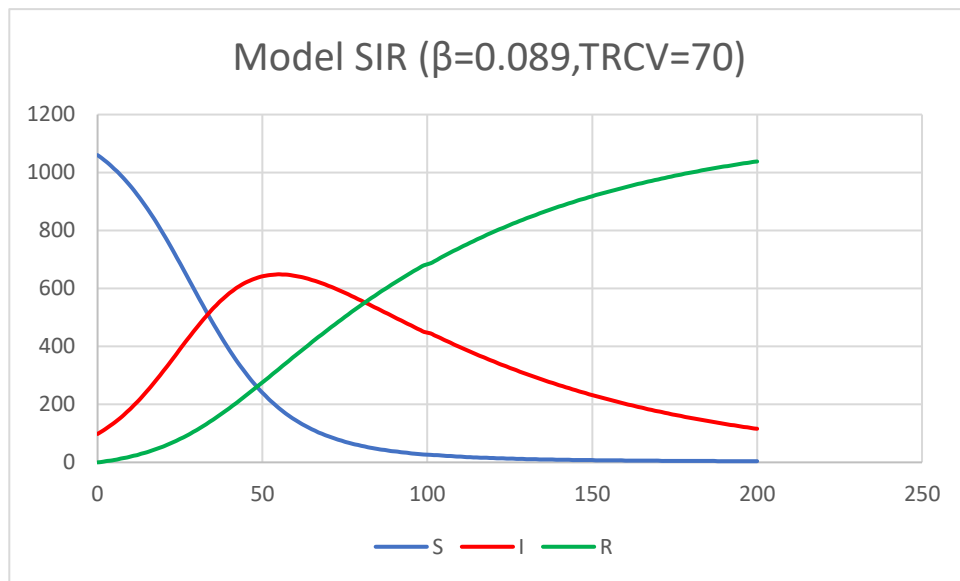
Gambar A.3.3 Hasil pengolahan data menggunakan model random walk dengan $R_0 = 5.34$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



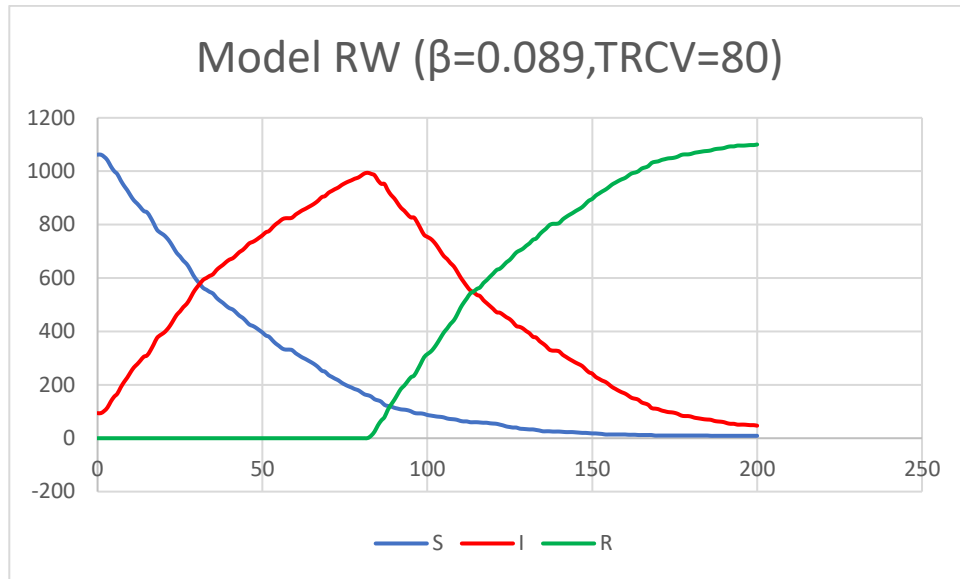
Gambar A.3.4 Hasil pengolahan data menggunakan model SIR dengan $R_0 = 5.34$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



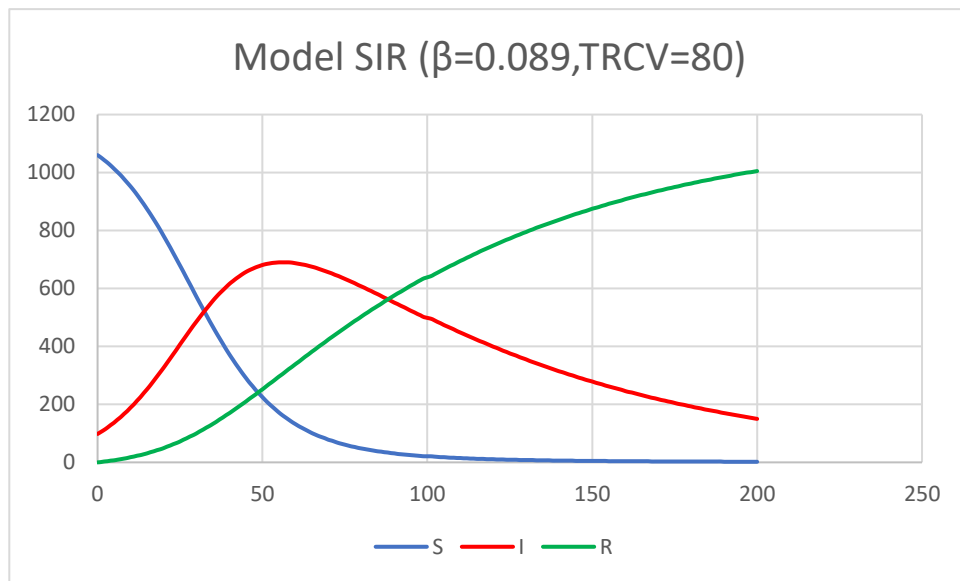
Gambar A.3.5 Hasil pengolahan data menggunakan model random walk dengan $R_0 = 6.23$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



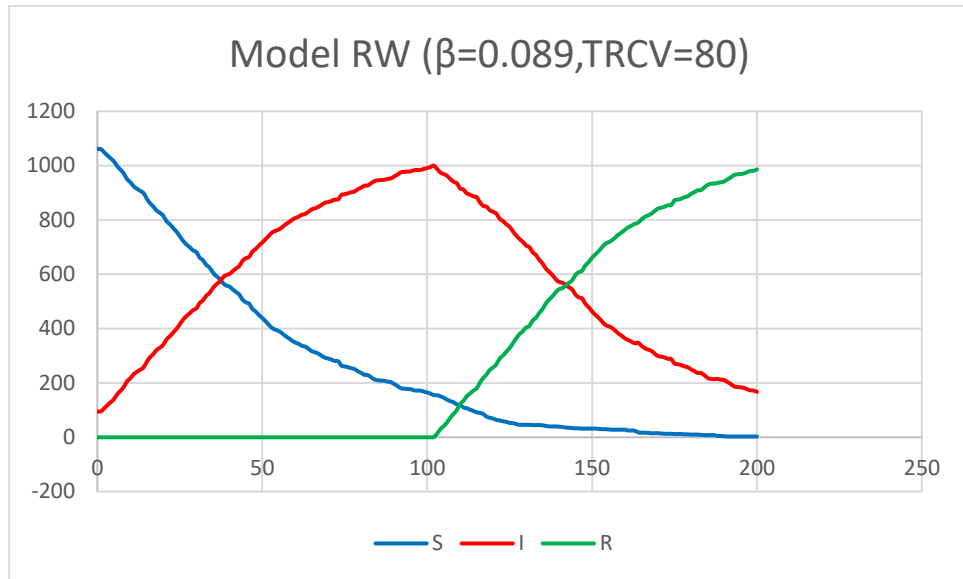
Gambar A.3.6 Hasil pengolahan data menggunakan model SIR dengan $R_0 = 6.23$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



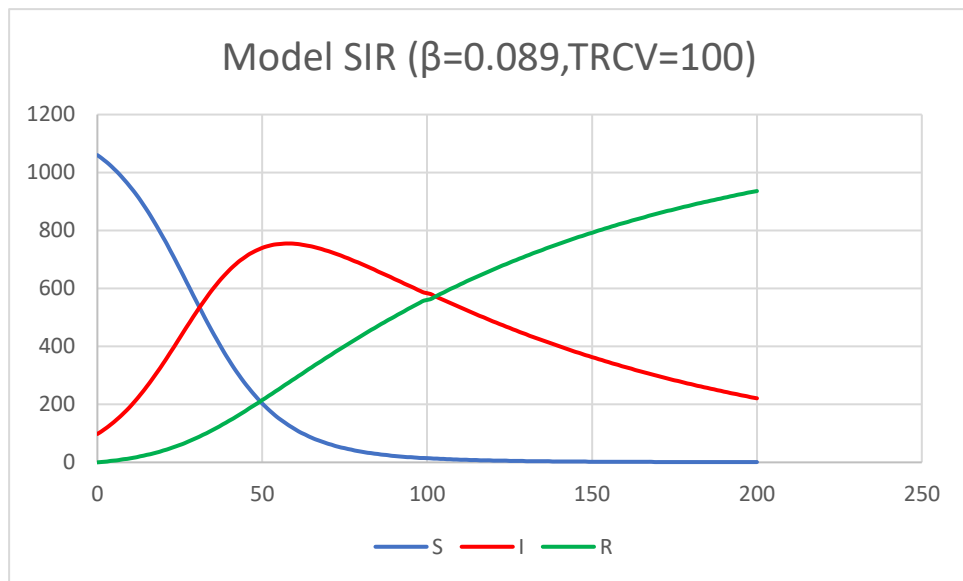
Gambar A.3.7 Hasil pengolahan data menggunakan model random walk dengan $R_0 = 7.12$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.



Gambar A.3.8 Hasil pengolahan data menggunakan model SIR dengan $R_0 = 7.12$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen

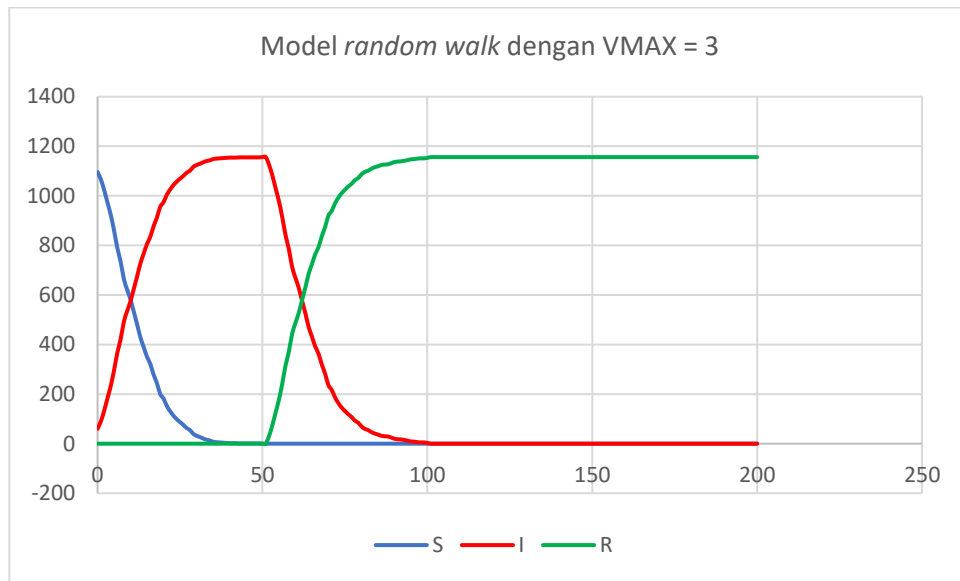


Gambar A.3.9 Hasil pengolahan data menggunakan model random walk dengan $R_0 = 8.90$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.

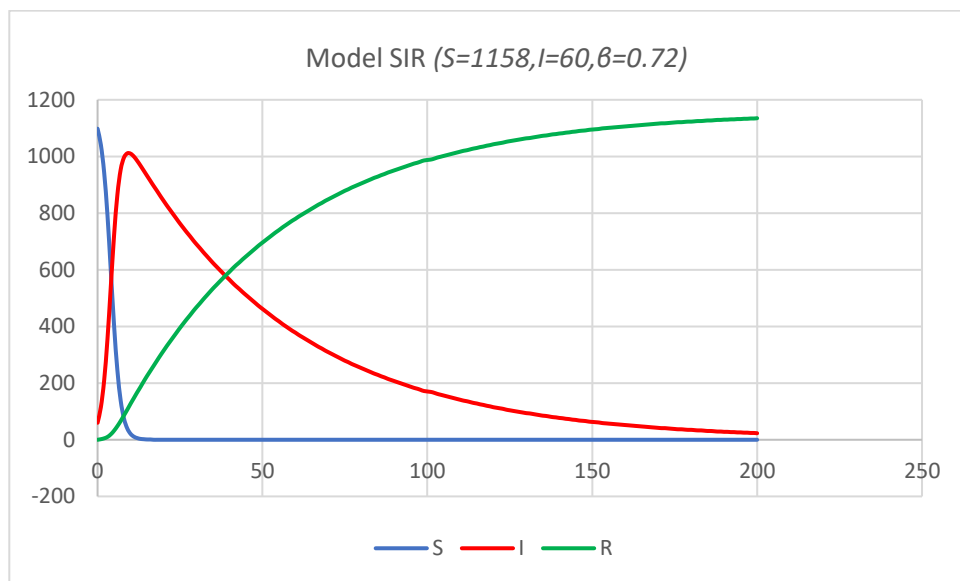


Gambar A.3.10 Hasil pengolahan data menggunakan model random walk dengan $R_0 = 8.90$ untuk populasi 1158 dan jumlah terinfeksi awal 98 agen.

A.4 Pengolahan data dengan variasi VMAX

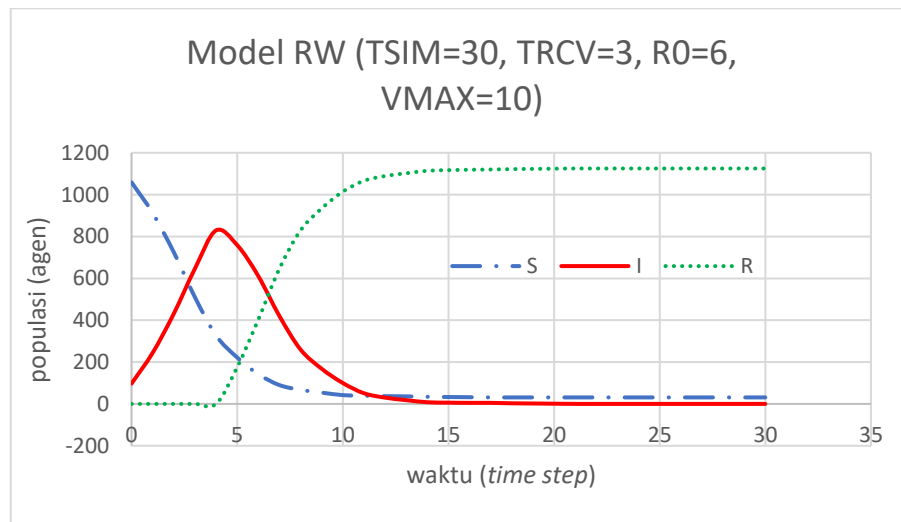


Gambar A.4.1 Hasil pengolahan data menggunakan model random walk dengan VMAX = 2 yang menghasilkan nilai $\beta=0.72$.

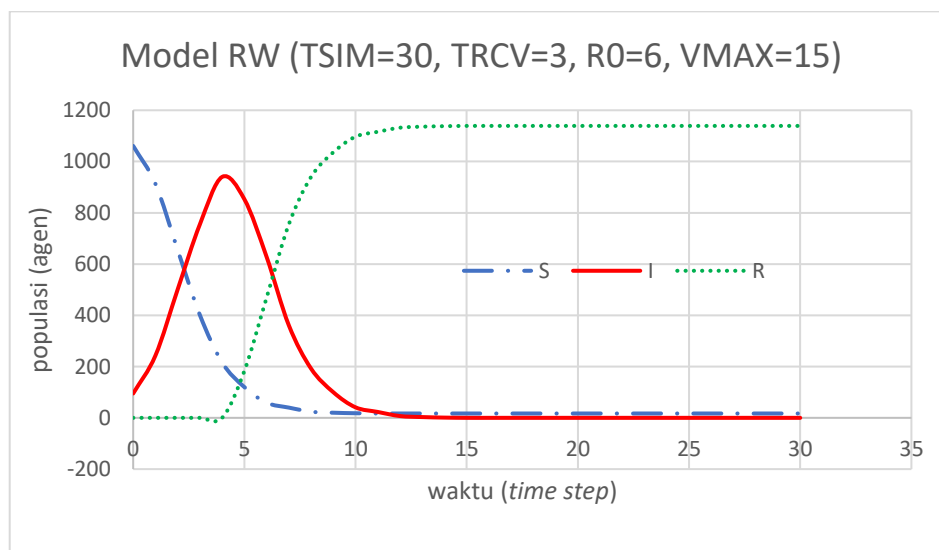


Gambar A.4.2 Hasil pengolahan data menggunakan model SIR dengan menggunakan nilai $\beta=0.72$.

A.5 Pengolahan data dengan variasi kecepatan (VMAX) dan waktu simulasi (TSIM)



Gambar A.5.1 Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan VMAX = 10.



Gambar A.5.2 Hasil pengolahan data menggunakan model random walk dengan menggunakan nilai variasi parameter waktu simulasi dengan VMAX = 16.

A.4 Pengolahan data dengan variasi kecepatan (VMAX) dan waktu simulasi (TSIM)

Lampiran B Kode Program Utama Dengan JavaScript

```
1  /*
2   wtrwgpmmids.js
3   Walk-through and random-walk of granular particles
4   as an alternative for mathematical modelling
5   of infectious disease spread
6
7   Sparisoma Viridi | https://github.com/dudung
8   Nuning Nuraini | nuning@math.itb.ac.id
9   Armi Susandi | armisusandi@yahoo.com
10  Intan Taufik | i.taufik@sith.itb.ac.id
11  Pingkan Aditiawati | pingkan@sith.itb.ac.id
12
13  20200308
14  1059 Beg bei Sukahaji.
15  1238 Fin temporaer tata letak.
16  1247 Update to Github.
17  1320 Finish button events and interactions.
18  1359 Can draw initial position for all particles.
19  1430 Can view all particle but only with infection
20  | scenario.
21  1509 Two states are created -1, 0, 1.
22  1645 Con at home.
23  1654 Change name from wtktgmmid to wtrwgpmmids.
24  1659 Info button ok.
25  1728 Fix color and output in 4 digits.
26  1802 Parameters can be given from UI.
27  1939 Finish for today. Update xphysics at Github.
28  20200309
29  0744 Correction: Amri --> Armi.
30  20200311
31  0533 Correct initial incubation time. Random.
32  0619 Add mode of initial tinf: 0, 1, 2.
33  20200320
```

```

39
40 // Define some global parameters
41 var pname = "WTRWGPMIDS";
42 var btLoad, btRead, btRun, btInfo;
43 var taIn, taOut, caOut, caOut2;
44 var bgColor = "#f8f8f8";
45 var N, xx, yy, dd, cc, hh, ii;
46 var xmin, ymin, xmax, ymax;
47 var XMIN, YMIN, XMAX, YMAX;
48 var xmin2, ymin2, xmax2, ymax2;
49 var XMIN2, YMIN2, XMAX2, YMAX2;
50 var proc, iter, Niter;
51 var colors = ["#888", "#f88", "#88f", "#8f8"];
52 var D, vmax;
53 var TRCV, NINF, MODE;
54 var S, I, R;
55
56 // Start the program
57 main()
58
59
60 // Main program
61 function main() {
62     cout("Program start");
63
64     layout();
65     initParams();
66 }
67
68
69 // Start the program
70 main()
71
72
73 // Main program
74 function main() {
75     cout("Program start");
76
77     layout();
78     initParams();
79 }
80
81
82 // Simulate
83 function simulate() {
84     if(iter >= Niter) {
85         clearInterval(proc);
86         stateButton(1, 1, 1, 1);
87         btRun.innerHTML = "Run";
88     }
89
90     var sir = getSIR();
91     tout(
92         ("0000" + iter).slice(-4) + " "
93         + ("0000" + sir[0]).slice(-4) + " "
94         + ("0000" + sir[1]).slice(-4) + " "
95         + ("0000" + sir[2]).slice(-4) + "\n"
96     );
97 }

```



```

85     S.push(sir[0]);
86     I.push(sir[1]);
87     R.push(sir[2]);
88
89     changePositions();
90     developImmunity();
91     propagateInfection();
92     clearX();
93     drawPositions();
94
95     clearX2();
96     drawSIRChart();
97
98     iter++;
99 }
100
101
102 // Develop immunity
103 function developImmunity() {
104     for(var i = 0; i < N; i++) {
105         var tinf = (ii[i] > -1) ? (iter - ii[i]) : 0;
106         if(tinf > TRCV) {
107             hh[i] = 1;
108             cc[i] = colors[hh[i] + 2];
109         }
110     }
111 }

```

```

113 // Propagate infection
114 function propagateInfection() {
115     for(var i = 0; i < N; i++) {
116         for(var j = i + 1; j < N; j++) {
117             var xi = xx[i];
118             var yi = yy[i];
119             var ri = 0.5 * dd[i];
120
121             var xj = xx[j];
122             var yj = yy[j];
123             var rj = 0.5 * dd[j];
124
125             var dx = xi - xj;
126             var dy = yi - yj;
127             var dr = Math.sqrt(dx*dx + dy*dy);
128             var ksi = (ri + rj) - dr;
129             ksi = (ksi < 0) ? 0 : ksi;
130
131             if(ksi > 0) {
132                 if(hh[i] == -1 && hh[j] == 0) {
133                     hh[j] = hh[i];
134                     cc[j] = cc[i];
135                     ii[j] = iter;
136                 }
137                 if(hh[j] == -1 && hh[i] == 0) {
138                     hh[i] = hh[j];
139                     cc[i] = cc[j];
140                     ii[i] = iter;
141                 }
142             }
143         }
144     }

```

```

144     }
145 }
146
147
148 // Change particles position
149 function changePositions() {
150     for(var i = 0; i < N; i++) {
151         var xold = xx[i];
152         var yold = yy[i];
153         var r = 0.5 * dd[i];
154
155         var theta = 2 * Math.PI * Math.random();
156         var vx = vmax * Math.cos(theta);
157         var vy = vmax * Math.sin(theta);
158         var dt = 1;
159
160         var xnew = xold + vx * dt;
161         var ynew = yold + vy * dt;
162
163         if(xnew < xmin + r || xmax - r < xnew) {
164             xnew = xold;
165         }
166         if(ynew < ymin + r || ymax - r < ynew) {
167             ynew = yold;
168         }
169
170         xx[i] = xnew;
171         yy[i] = ynew;
172     }
173 }
174
175 // Calculate infected
176 function getSIR() {
177     var I = 0;
178     var S = 0;
179     var R = 0;
180     for(var i = 0; i < N; i++) {
181         if(hh[i] == -1) {
182             I++;
183         }
184         if(hh[i] == 0) {
185             S++;
186         }
187         if(hh[i] == 1) {
188             R++;
189         }
190     }
191     return [S, I, R];
192 }
193
194
195 // Initialize parameters
196 function initParams() {
197     xmin = 0;
198     ymin = 0;
199     xmax = 300;
200     ymax = 300;
201
202     XMIN = 0;
203     YMIN = parseInt(caOut.height);
204     XMAX = parseInt(caOut.width);
205     YMAX = 0;

```

```

220 // Load parameters
221 function loadParams() {
222     var p = "";
223     p += "# Population\n";
224     p += "NPOP 1600\n";           // N
225     p += "NINF 100\n";           // NINF
226     p += "TRCV 50\n";            // TRCV
227     p += "\n";
228     p += "# Individu\n";
229     p += "VMAX 1\n";             // vmax
230     p += "\n";
231     p += "# Simulation\n";       // N
232     p += "TSIM 200\n";           // Niter
233     p += "\n";
234     p += "# Mode of tinf\n";     // N
235     p += "> 0 0\n";              // N
236     p += "> 1 rnd[0, 1) TRCV\n"; // N
237     p += "> 2 linear\n";        // N
238     p += "MODE 0\n";            // Niter
239     tin(p);
240 }

```

```

264 // Initialize all positions
265 function initPositions() {
266     var Nx = Math.floor(Math.sqrt(N));
267     var Ny = Nx;
268     var lx = (xmax - xmin) / Nx;
269     var ly = (ymax - ymin) / Ny;
270
271     D = 0.34 * (lx + ly);
272
273     xx = [];
274     yy = [];
275     dd = [];
276     cc = [];
277     hh = [];
278     ii = [];
279
280     for(var iy = 0; iy < Ny; iy++) {
281         for(var ix = 0; ix < Nx; ix++) {
282             var x = (ix + 0.5) * lx;
283             var y = (iy + 0.5) * ly;
284
285             x += 0.2 * D * (Math.random() - 0.5);
286             y += 0.2 * D * (Math.random() - 0.5);
287
288             xx.push(x);
289             yy.push(y);
290             dd.push(D);
291             hh.push(0);
292             ii.push(-1);
293             cc.push(colors[2]);

```

```

295     S = [];
296     I = [];
297     R = [];
298 }
299
300
301 for(var j = 0; j < NINF; j++) {
302     var i = Math.floor(Math.random() * N);
303     hh[i] = -1;
304     cc[i] = colors[hh[i] + 2];
305     if(MODE == 0) {
306         ii[i] = 0;
307     } else if(MODE == 1) {
308         ii[i] = Math.random() * TRCV;
309     } else {
310         ii[i] = Math.floor((TRCV / NINF) * j);
311     }
312 }
313 }
314
315 // Draw all chart
316 function drawSIRChart() {
317     var cx = caOut2.getContext("2d");
318     ---
319     var NS = S.length;
320     cx.lineWidth = "2";
321     cx.strokeStyle = colors[2];
322     cx.beginPath();
323     for(var i = 0; i < NS; i++) {
324         var X = Xx2(i);
325         var Y = Yy2(S[i] / N);
326         if(i == 0) {
327             cx.moveTo(X, Y);
328         } else {
329             cx.lineTo(X, Y);
330         }
331     }
332     cx.stroke();
333
334     var NI = I.length;
335     cx.strokeStyle = colors[1];
336     cx.lineWidth = "2";
337     cx.beginPath();
338     for(var i = 0; i < NI; i++) {
339         var X = Xx2(i);
340         var Y = Yy2(I[i] / N);
341         if(i == 0) {
342             cx.moveTo(X, Y);
343         } else {
344             cx.lineTo(X, Y);
345         }
346     }
347     cx.stroke();
348 }
349
350

```

```

351     var NR = R.length;
352     cx.strokeStyle = colors[3];
353     cx.lineWidth = "2";
354     cx.beginPath();
355     for(var i = 0; i < NR; i++) {
356         var X = Xx2(i);
357         var Y = Yy2(R[i] / N);
358         if(i == 0) {
359             cx.moveTo(X, Y);
360         } else {
361             cx.lineTo(X, Y);
362         }
363     }
364     cx.stroke();
365
366     function Xx2(x) {
367         var X = (x - xmin2) / (xmax2 - xmin2);
368         X *= (XMAX2 - XMIN2);
369         X += XMIN2;
370         return X;
371     }
372     function Yy2(y) {
373         var Y = (y - ymin2) / (ymax2 - ymin2);
374         Y *= (YMAX2 - YMIN2);
375         Y += YMIN2;
376         return Y;
377     }
378 }
379
380 // Draw all positions
381 function drawPositions() {
382     var cx = caOut.getContext("2d");
383     for(var i = 0; i < N; i++) {
384         var X = Xx(xx[i]);
385         var Y = Yy(yy[i]);
386         var D = 0.5 * (Xx(xx[i] + dd[i]) - X);
387         var C = cc[i];
388         cx.beginPath();
389         cx.fillStyle = C;
390         cx.arc(X, Y, D, 0, 2 * Math.PI);
391         cx.fill();
392         cx.beginPath();
393         cx.strokeStyle = "#444";
394         cx.arc(X, Y, D, 0, 2 * Math.PI);
395         cx.stroke();
396     }
397 }
398
399 function Xx(x) {
400     var X = (x - xmin) / (xmax - xmin);
401     X *= (XMAX - XMIN);
402     X += XMIN;
403     return X;
404 }
405 function Yy(y) {
406     var Y = (y - ymin) / (ymax - ymin);
407     Y *= (YMAX - YMIN);
408     Y += YMIN;
409     return Y;
410 }
411 }

```

```

414 // Set layout
415 function layout() {
416     btLoad = document.createElement("button");
417     btLoad.innerHTML = "Load";
418     btLoad.style.width = "50px";
419     btLoad.style.height = "21px";
420
421     btRead = document.createElement("button");
422     btRead.innerHTML = "Read";
423     btRead.style.width = "50px";
424     btRead.style.height = "21px";
425
426     btRun = document.createElement("button");
427     btRun.innerHTML = "Run";
428     btRun.style.width = "50px";
429     btRun.style.height = "21px";
430
431     btInfo = document.createElement("button");
432     btInfo.innerHTML = "Info";
433     btInfo.style.width = "50px";
434     btInfo.style.height = "21px";
435
436     taIn = document.createElement("textarea");
437     taIn.style.width = "194px";
438     taIn.style.height = "269px";
439     taIn.style.overflowY = "scroll";
440
441     taOut = document.createElement("textarea");
442     taOut.style.width = "194px";
443     taOut.style.height = "294px";

```

```

454     caOut2 = document.createElement("canvas");
455     caOut2.width = "702";
456     caOut2.height = "100";
457     caOut2.style.width = caOut2.width + "px";
458     caOut2.style.height = caOut2.height + "px";
459     caOut2.style.background = bgColor;
460     caOut2.style.border = "1px solid #ccc";
461
462     var divLeft = document.createElement("div");
463     divLeft.style.border = "1px solid #ccc";
464     divLeft.style.background = "#eee";
465     divLeft.style.width = "200px";
466     divLeft.style.height = "300px";
467     divLeft.style.float = "left";
468
469     var divRight = document.createElement("div");
470     divRight.style.border = "1px solid #ccc";
471     divRight.style.background = "#eee";
472     divRight.style.width = "500px";
473     divRight.style.height = "300px";
474     divRight.style.float = "left";
475
476     document.body.append(divLeft);
477     divLeft.append(taIn);
478     divLeft.append(btLoad);
479     divLeft.append(btRead);
480     divLeft.append(btRun);
481     divLeft.append(btInfo);
482     document.body.append(divRight);
483     divRight.append(caOut);
484     divRight.append(taOut);
485     document.body.append(caOut2);
486
487     btLoad.addEventListener("click", clickButton);
488     btRead.addEventListener("click", clickButton);
489     btRun.addEventListener("click", clickButton);
490     btInfo.addEventListener("click", clickButton);
491
492     stateButton(1, 0, 0, 1);
493
494
495
496     // Click button
497     function clickButton() {
498         var t = event.target;
499         var n = t.innerHTML;
500         cout("Button click -> " + n);
501
502         if(n == "Load") {
503             stateButton(1, 1, 0, 1);
504             clearTin();
505             loadParams();
506         } else if(n == "Read") {

```



```

507         stateButton(1, 1, 1, 1);
508         readParams();
509         iter = 0;
510         initPositions();
511         clearX();
512         drawPositions();
513     } else if(n == "Run") {
514         stateButton(0, 0, 1, 0);
515         btRun.innerHTML = "Stop";
516         proc = setInterval(simulate, 10);
517     } else if(n == "Stop") {
518         stateButton(1, 1, 1, 1);
519         btRun.innerHTML = "Run";
520         clearInterval(proc);
521     } else if(n == "Info") {
522         showInfo();
523     }
524 }
525
526
527 // Show information
528 function showInfo() {
529     var msg = "wtrwgpmmids.js\n" +
530     "Walk-through and random-walk of granular particles\n" +
531     "as an alternative for mathematical modelling\n" +
532     "of infectious disease spread\n\n" +
533     "Sparisoma Viridi | https://github.com/dudung\n" +
534     "Nuning Nuraini | nuning@math.itb.ac.id\n" +
535     "Armi Susandi | armisusandi@yahoo.com\n" +
536     "Intan Taufik | i.taufik@sith.itb.ac.id\n" +
537     "Pingkan Aditiawati | pingkan@sith.itb.ac.id\n" +
538     "Version 20200308";
539
540 // Set buttons state
541 function stateButton() {
542     var sL = arguments[0];
543     var sR = arguments[1];
544     var sS = arguments[2];
545     var sI = arguments[3];
546
547     btLoad.disabled = !sL;
548     btRead.disabled = !sR;
549     btRun.disabled = !sS;
550     btInfo.disabled = !sI;
551 }
552
553 // Clear canvas
554 function clearX() {
555     var cx = caOut.getContext("2d");
556     cx.clearRect(XMIN, YMAX, XMAX, YMIN);
557 }
558
559 // Clear 2nd canvas
560 function clearX2() {
561     var cx = caOut2.getContext("2d");
562     cx.clearRect(XMIN2 - 4, YMAX2 - 4, XMAX2 + 4, YMIN2 + 4);
563 }

```

Lampiran C Kode Program Tambahan Untuk Kalibrasi Dengan Python

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed May 20 02:17:45 2020
4
5 @author: ASUS
6 """
7
8 import numpy as np
9 from scipy.integrate import odeint
10 import matplotlib.pyplot as plt
11 import csv
12 import pandas as pd
13
14 dataset = pd.read_csv('save_progress.csv')
15 # Total population, N.
16 N = 9000
17 # Initial number of infected and recovered individuals, I0 and R0.
18 I0, R0 = 80, 0
19 # Everyone else, S0, is susceptible to infection initially.
20 S0 = N - I0 - R0
21 # Contact rate, beta, and mean recovery rate, gamma, (in 1/days).
22 beta, gamma = 0.046, 1./50
23 # A grid of time points (in days)
24 t = np.linspace(0, 200, 200)
25
26 # The SIR model differential equations.
27 def deriv(y, t, N, beta, gamma):
28     S, I, R = y
29     dSdt = -beta * S * I / N
30     dIdt = beta * S * I / N - gamma * I
31     dRdt = gamma * I
32     return dSdt, dIdt, dRdt
33
34 # Initial conditions vector
35 y0 = S0, I0, R0
36 # Integrate the SIR equations over the time grid, t.
37 ret = odeint(deriv, y0, t, args=(N, beta, gamma))
38 S, I, R = ret.T
39
40 #save data
41 save_data = open('save_progress1.csv', 'w')
42 save_data.write('t,S,I,R\n')
43 for i in range(len(t)):
44     str_save = '%s,%s,%s,%s\n'%(round(t[i]), round(S[i]), round(I[i]), round(R[i]))
45     save_data.write(str_save)
46 save_data.close()
47
48 # Plot the data on three separate curves for S(t), I(t) and R(t)
49 fig = plt.figure(facecolor='w')
50 ax = fig.add_subplot(111, facecolor='#dddddd', axisbelow=True)
51 ax.plot(t, S/1000, 'b', alpha=0.5, lw=2, label='Susceptible')
52 ax.plot(t, I/1000, 'r', alpha=0.5, lw=2, label='Infected')
53 ax.plot(t, R/1000, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
54 ax.set_xlabel('Time steps')
55 ax.set_ylabel('Number (1000s)')
56 ax.set_ylim(0,2)
57 ax.yaxis.set_tick_params(length=0)
58 ax.xaxis.set_tick_params(length=0)
59 ax.grid(b=True, which='major', c='w', lw=2, ls='--')
60 legend = ax.legend()
61 legend.get_frame().set_alpha(0.5)
62 for spine in ('top', 'right', 'bottom', 'left'):
63     ax.spines[spine].set_visible(False)
64 plt.show()
65

```

Lampiran D Tabel**D.1 Tabel hasil pengolahan dengan variasi populasi**

Time	S	I	R
0	820	80	0
1	820	80	0
2	816	84	0
3	813	87	0
4	803	97	0
5	793	107	0
6	781	119	0
7	777	123	0
8	772	128	0
9	762	138	0
10	758	142	0
11	746	154	0
12	737	163	0
13	725	175	0
14	715	185	0
15	706	194	0
16	700	200	0
17	688	212	0
18	673	227	0
19	661	239	0
20	656	244	0
21	648	252	0
22	638	262	0
23	631	269	0
24	619	281	0
25	607	293	0
26	595	305	0
27	580	320	0
28	567	333	0
29	553	347	0
30	542	358	0
31	534	366	0
32	523	377	0
33	517	383	0
34	505	395	0
35	498	402	0
36	483	417	0
37	471	429	0
38	464	436	0
39	457	443	0
40	451	449	0
41	444	456	0
42	438	462	0
43	430	470	0

44	421	479	0
45	412	488	0
46	406	494	0
47	397	503	0
48	384	516	0
49	381	519	0
50	367	533	0
51	357	543	0
52	352	546	2
53	348	544	8
54	342	545	13
55	336	540	24
56	331	535	34
57	322	532	46
58	319	530	51
59	314	529	57
60	308	522	70
61	306	516	78
62	301	507	92
63	293	504	103
64	290	494	116
65	288	485	127
66	280	481	139
67	276	475	149
68	272	466	162
69	267	456	177
70	259	451	190
71	259	445	196
72	255	438	207
73	253	429	218
74	252	421	227
75	249	411	240
76	246	399	255
77	241	391	268
78	237	377	286
79	231	370	299
80	229	358	313
81	227	347	326
82	225	341	334
83	224	330	346
84	222	322	356
85	221	310	369
86	221	302	377
87	216	290	394
88	216	275	409
89	216	267	417

90	216	260	424
91	210	259	431
92	207	253	440
93	204	249	447
94	197	246	457
95	194	239	467
96	193	229	478
97	190	224	486
98	188	217	495
99	186	201	513
100	186	195	519
101	184	183	533
102	184	173	543
103	183	169	548
104	182	166	552
105	180	162	558
106	179	157	564
107	179	152	569
108	178	144	578
109	173	146	581
110	173	141	586
111	172	136	592
112	171	135	594
113	168	133	599
114	167	126	607
115	165	125	610
116	162	126	612
117	160	120	620
118	158	118	624
119	157	115	628
120	152	115	633
121	151	108	641
122	146	113	641
123	145	110	645
124	145	108	647
125	145	107	648
126	144	105	651
127	143	103	654
128	143	98	659
129	143	94	663
130	142	89	669
131	140	89	671
132	139	88	673
133	139	86	675
134	139	85	676
135	139	83	678

136	138	83	679
137	134	87	679
138	134	82	684
139	134	82	684
140	134	82	684
141	132	84	684
142	131	79	690
143	129	78	693
144	127	77	696
145	126	71	703
146	122	72	706
147	122	71	707
148	122	68	710
149	122	66	712
150	121	65	714
151	120	66	714
152	120	64	716
153	120	64	716
154	119	64	717
155	118	64	718
156	117	63	720
157	115	64	721
158	115	64	721
159	115	63	722
160	115	58	727
161	114	59	727
162	114	58	728
163	114	57	729
164	114	54	732
165	114	53	733
166	113	52	735
167	113	49	738
168	113	47	740
169	112	46	742
170	110	47	743
171	110	42	748
172	110	41	749
173	110	36	754
174	110	35	755
175	110	35	755
176	110	35	755
177	110	34	756
178	110	33	757
179	110	33	757
180	110	33	757
181	109	33	758

182	109	31	760
183	109	30	761
184	109	30	761
185	109	30	761
186	108	31	761
187	108	30	762
188	108	26	766
189	108	26	766
190	108	26	766
191	108	26	766
192	108	24	768
193	106	25	769
194	106	23	771
195	105	22	773
196	104	22	774
197	101	21	778
198	101	21	778

