DOKUMEN PENGEMBANGAN PRODUK  
MODEL SIMULASI EPIDEMIOLOGI DENGAN PLATFORM GAMA

# Daftar Isi

[Daftar Isi 2](#_Toc51582427)

[1. Pendahuluan 3](#_Toc51582428)

[1.1. Masalah 3](#_Toc51582429)

[1.2. Solusi 3](#_Toc51582430)

[1.2.1. Alternatif Solusi 3](#_Toc51582431)

[1.2.2. Solusi Terpilih 4](#_Toc51582432)

[2. Produk Model Simulasi Epidemiologi 5](#_Toc51582433)

[2.1. Penjelasan Produk 5](#_Toc51582434)

[2.2. Pengembangan Produk 5](#_Toc51582435)

[2.2.1. Berkas dalam Produk 6](#_Toc51582436)

[2.2.2. Pengembangan Parameter Epidemiologis 7](#_Toc51582437)

[2.2.3. Pengembangan Blok Fungsi dan Prosedur 7](#_Toc51582438)

[2.2.4. Penggantian Berkas *Shapefile* 8](#_Toc51582439)

# Pendahuluan

## Latar Belakang Masalah

Pandemi Coronavirus Disease 2019, atau COVID-19, merupakan pandemi yang disebabkan oleh severe acute respiratory syndrome coronavirus 2, atau SARS-CoV-2. Pada 30 Januari 2020, komite darurat dari World Health Organization, atau WHO, setelah mendapatkan laporan 7711 kasus konfirmasi dan 12167 kasus suspek pada Republik Rakyat Cina, serta laporan lainnya, menetapkan pandemi ini sebagai Public Health Emergency of International Concern (PHEIC), [1]. Dalam kesempatan yang sama komite ini menyarankan beberapa hal pada negara-negara, termasuk melakukan usaha mengurangi penularan penyakit, kolaborasi multisektor, dan riset. Dengan terdeteksinya kasus konfirmasi pertama di Indonesia pada 2 Maret 2020 [2], pemerintah menetapkan usaha-usaha pencegahan COVID-19 dengan di antaranya pembentukan Gugus Tugas Percepatan Penanganan COVID-19 [3] dan Pembatasan Sosial Berskala Besar, atau PSBB [4].

Menurut [4], pemberlakuan PSBB dapat diusulkan oleh Pemerintah Daerah dengan mempertimbangkan jumlah kasus, pembatasan kegiatan yang akan dilakukan, dan kebutuhan dasar penduduk yang harus tetap dipenuhi jika PSBB itu diberlakukan. Model epidemiologis dapat membantu pemerintah atau peneliti dalam mengetahui efektivitas dari PSBB, dan model SEIR (*susceptible-exposed-infectious-recovered*) adalah model yang sudah kerap digunakan dalam penyebaran penyakit menular, [5]. Model SEIR adalah suatu sistem persamaan diferensial dengan koefisien-koefisien yang mewakili parameter epidemiologis seperti laju infeksi, laju penyembuhan, dll., sehingga didapatkan fungsi jumlah individu yang tergolong belum terinfeksi, terinfeksi tapi belum bisa menularkan penyakit, terinfeksi dan dapat menularkan penyakit, atau sembuh, [6].

Tetapi, penggunaan model SEIR yang secara keseluruhan merupakan persamaan matematis tersebut kurang bisa menunjukkan dampak dari variasi yang terjadi dalam penentapan PSBB di suatu daerah. Dinamika ini misalnya pilihan durasi PSBB atau pilihan hari ditetapkannya PSBB sejak ditemukannya kasus konfirmasi di daerah itu. Selain itu, terdapat banyak perbedaan dari daerah-daerah di Indonesia, terutama kondisi geografis dan demografisnya, yang pasti akan memberikan hasil berbeda apabila diberikan peraturan PSBB yang sama. Sehingga, diperlukan model selain SEIR yang dapat digunakan untuk bereksperimen mengenai variasi peraturan PSBB yang sesuai dengan karakteristik suatu daerah.

## Solusi

### Alternatif Solusi

Metode selain SEIR (yang sepenuhnya merupakan persamaan matematis) adalah agent-based modelling, atau ABM. Dalam metode ABM, sebuah sistem dimodelkan menjadi objek-objek yang disebut agen, yang dapat melakukan interaksi dengan agen yang sejenis atau berbeda jenis, [7]. Pemodelan epidemiologi dengan ABM dilakukan dengan menyusun agen manusia dan agen bangunan, sehignga interaksi yang terjadi adalah manusia-manusia atau manusia-bangunan. Dengan ini, karakteristik suatu daerah dapat direplikasi dengan agen manusia dan bangunan tersebut. Fenomena penyebaran penyakit juga dapat diwakili dalam simulasi dengan interaksi antar agen manusia.

Solusi berikutnya adalah menggunakan COMOKIT, suatu model epidemiologi yang ditulis dalam program lain, yaitu GAMA Platform. GAMA Platform adalah sarana code sekaligus sarana simulasi, untuk menulis program model dan menjalankan simulasi dengan ABM, [8]. GAMA dapat mengambil masukan file GIS (Geographic Information System), yang mewakili peta bangunan suatu daerah, sehingga bangunan-bangunan pada daerah itu dapat secara otomatis direplikasi ke dalam simulasi, sekaligus berperan sebagai. COMOKIT telah digunakan untuk memperkirakan perkembangan situasi pada lockdown di dua kota di Vietnam, [9].

### Solusi Terpilih

COMOKIT memiliki fitur pemodelan yang sangat detail, di antaranya adalah algoritma-algoritma yang menyebabkan beragamnya aktivitas agent manusia pada simulasi, mulai dari aktivitas bekerja/bersekolah, berbelanja, mengunjungi tetangga, dan hiburan. Pemodelan juga sangat detail pada bagian epidemiologinya, dengan adanya algoritma untuk peraturan lockdown yang membatasi aktivitas agen manusia.

Namun, terdapat kekurangan yang timbul dari banyaknya fitur tersebut. Fitur-fitur semakin detail sehingga dapat menyerupai kejadian epidemiologi sebenarnya pada suatu daerah, tetapi hal ini membuat durasi loading dan durasi simulasi menjadi lebih lambat. Selain itu, kekurangan dari COMOKIT dalam mengakomodasi simulasi epidemiologi di Indonesia adalah karena adanya banyak perbedaan peraturan yang berlaku di Indonesia, misalnya peraturan alur manajemen klinis seseorang yang dikonfirmasi positif terinfeksi SARS-CoV-2.

Sehingga, solusi yang terpilih adalah untuk menyusun program pemodelan dari awal di GAMA Platform untuk simulasi epidemiologi, dengan melakukan simplifikasi di beberapa fiturnya, sehingga program tidak memberi beban komputasi yang terlalu besar. Serta, disusun algoritma untuk peraturan klinis yang sesuai dengan peraturan yang berlaku di Indonesia, sehingga hasil simulasi lebih realistis digunakan di negara ini.

# Produk Model Simulasi Epidemiologi

## Penjelasan Produk

Model Simulasi Epidemiologi yang telah ditulis merupakan model *agent-based modelling* yang memanfaatkan berkas GIS (*Geographic Information System*) yang bertujuan untuk menyimulasikan penyebaran penyakit di suatu daerah tertentu, serta tindakan pemerintah untuk mengatasi penyebaran penyakit ini, dengan pemberlakuan PSBB atau *new normal*. Model ini berupa beberapa jenis berkas yang digabung dalam 1 *folder*, yang dapat di-*import* dalam aplikasi GAMA untuk dijalankan atau dimodifikasi. Daerah tertentu yang dimaksud sebelumnya adalah daerah dalam skala kecamatan, dan daerah yang digunakan pada model adalah representasi GIS untuk Kecamatan Gubeng, Kota Surabaya, dalam format *shapefile*.

Lebih lanjutnya, produk memiliki fitur sebagai berikut:

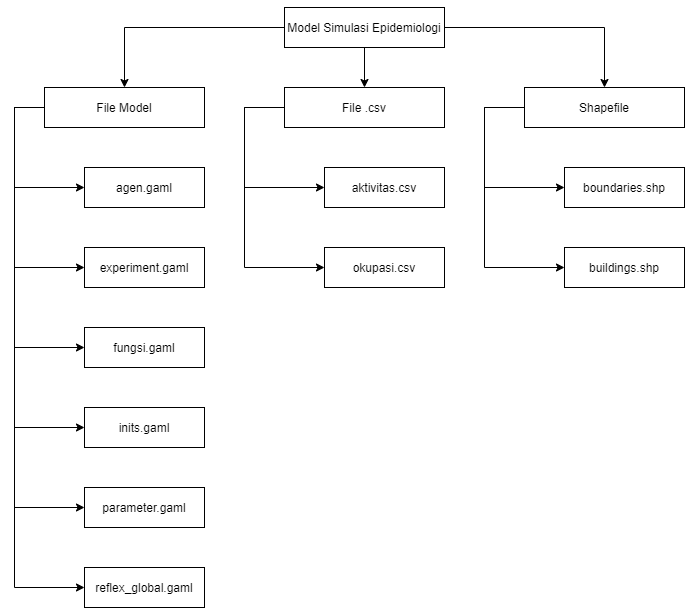
* Penggunaan parameter epidemiologis menurut *paper* terkini untuk memodelkan penyebaran penyakit;
* Perilaku agen manusia yang beragam dengan adanya kegiatan rutin dan kegiatan acak agen manusia, disertai dengan atribut yang dapat mempengaruhi tingkah laku tersebut.
  + Misalnya, atribut usia yang kurang dari usia sekolah akan menyebabkan agen manusia itu memiliki aktivitas pergi ke sekolah secara rutin tiap hari sekolah.
* Status SEIR yang berlaku pada agen manusia;
* Skenario manajemen agen manusia yang terinfeksi untuk dirawat di rumah sakit (manajemen klinis) sesuai dengan peraturan yang berlaku di Indonesia;
* Skenario perubahan peraturan yang mempengaruhi perilaku agen manusia saat simulasi;
* Antarmuka yang memberi kesempatan pengguna untuk mengubah beberapa parameter sebelum simulasi:
  + Jumlah hari yang disimulasikan,
  + Jumlah agen yang terinfeksi di awal simulasi,
  + Batas jumlah agen yang terkonfirmasi positif COVID-19 sehingga memicu PSBB
  + Batas jumlah hari terdapat penurunan kasus konfirmasi per hari sehingga PSBB dilonggarkan menjadi *new normal*.
  + Akurasi tes
  + Efektivitas *contact-tracing*

## Pengembangan Produk

Produk berupa model yang dikembangkan ini ditujukan pada pihak pembuat peraturan untuk digunakan secara seluas-luasnya. Maka dari itu, pengguna model simulasi ini dipersilakan untuk melakukan pengembangan apabila diperlukan. Pengembangan dilakukan dengan mengganti kode program pada model atau mengganti berkas *shapefile* GIS untuk menyimulasikan daerah yang berbeda daripada yang dicantumkan dalam produk. Pengembangan juga dapat dilakukan dengan cara mengubah parameter epidemiologis yang tercantum dalam berkas model. Hal ini dapat bermanfaat apabila di masa depan ditemukan *paper* yang lebih akurat atau lebih sesuai dengan kondisi perkembangan epidemiologis pada masyarakat Indonesia saat itu.

### Berkas dalam Produk

Penggantian kode program pada model atau penggantian berkas *shapefile* GIS dilakukan pada berkas-berkas yang sesuai. Adapun berkas-berkas yang disertakan dalam model ditunjukkan pada gambar 2-1.



Gambar 2-1. Diagram berkas-berkas yang tercantum pada model.

Diperlihatkan bahwa model terdiri dari berkas-berkas model, yakni yang memiliki *extension* .gaml, kemudian data-data dalam format .csv, serta *shapefile* GIS. Berkas .gaml berisi kode-kode yang ditulis dalam GAML, atau *GAMA Modelling Language*. Ia adalah suatu bahasa pemrograman yang sifatnya *object-oriented*. Referensi dan dokumentasi GAML terdapat pada laman situs resmi GAMA (<https://gama-platform.github.io/wiki/GamlReference> ).

Berkas agen.gaml berisi kode untuk memrogram perilaku dan atribut agen. Perilaku agen manusia yang dikodekan adalah perilaku yang terkait dengan penjalanan aktivitas, infeksi, pengetesan, manajemen klinis, dll.. Berkas fungsi.gaml berisi fungsi yang kemudian dipanggil oleh kode di berkas lainnya. Inits.gaml mengkodekan fungsi-fungsi yang harus dilakukan sebelum simulasi berlangsung, yakni *generation* seluruh agen dan atributnya. Kemudian parameter.gaml berisi parameter-parameter epidemiologis maupun demografis yang mempengaruhi perilaku agen. Adapun reflex\_global.gaml berisi fungsi untuk memicu terjadi perubahan peraturan di tengah simulasi. Terakhir, experiment.gaml berisi fungsi untuk memulai simulasi dengan memanggil semua fungsi inisiasi, serta berisi kode untuk memrogram grafik yang muncul saat simulasi.

Berkas .csv berisi data-data tambahan untuk melengkapi atribut agen manusia. Aktivitas.csv berisi nilai atribut pekerjaan manusia yang mungkin, seperti pekerja kantor, pegawai bank, dll.. Sementara, okupasi.csv berisi nilai pembobot untuk atribut pekerjaan. Misalnya, pekerjaan pegawai kantor memiliki bobot lebih tinggi dari pekerjaan petani (seseorang lebih mungkin untuk menjadi pegawai kantor daripada petani di daerah perkotaan).

Berkas *shapefile* menentukan keragaman agen bangunan serta kenampakannya pada peta yang ditampilkan saat simulasi.

### Pengembangan Parameter Epidemiologis

Dalam bagian ini, akan disebutkan parameter-parameter epidemiologis yang digunakan pada model. Berikut disertakan pula referensi *paper* yang memberikan nilai-nilai parameter tersebut.

|  |  |  |
| --- | --- | --- |
| No | Nama parameter | Referensi |
| 1. | *Clinical Fraction* | Davies, N.G., Klepac, P., Liu, Y. et al. Age-dependent effects in the transmission and control of COVID-19 epidemics. Nat Med 26, 1205–1211 (2020). https://doi.org/10.1038/s41591-020-0962-9 |
| 2. | *Susceptibility* |
| 3. | Distribusi Periode Inkubasi | Bi Q, Wu Y, Mei S, Ye C, Zou X, Zhang Z, et al. Epidemiology and Transmission of COVID-19 in Shenzhen China: Analysis of 391 cases and 1,286 of their close contacts. medRxiv. 2020 Mar 27;2020.03.03.20028423. |
| 4. | Distribusi *Serial Interval* |
| 5. | Waktu sejak gejala hingga recovered |
| 6. | Proporsi *hospitalization* | CDCMMWR. Severe Outcomes Among Patients with Coronavirus Disease 2019 (COVID-19) — United States, February 12–March 16, 2020. MMWR Morb Mortal Wkly Rep [Internet]. 2020 [cited 2020 Apr 21];69. Available from: https://www.cdc.gov/mmwr/volumes/69/wr/mm6912e2.htm |
| 7. | Proporsi masuk ICU |
| 8. | Proporsi fatalitas |

Nilai parameter yang diambil disimpan dalam berkas parameter.gaml dalam variabel-variabel dengan tipe data yang beragam. Alasannya, beberapa variabel berbentuk distribusi probabilitas yang membutuhkan dua nilai serta jenis distribusinya untuk didefinisikan. Misalnya, distribusi normal membutuhkan dua nilai yaitu µ dan σ. Sementara, beberapa variabel lain berbentuk distribusi probabilitas yang berbeda-beda untuk kelompok umur yang berbeda. Misalnya pada variabel *susceptibility*, yang menurut distribusi normal, tetapi distribusinya berbeda antara kelompok usia 0 s.d. 9 dengan usia 10 s.d. 19, dan seterusnya.

Lebih detailnya, penulisan nilai variabel-variabel ini dituliskan pada komentar-komentar pada berkas parameter.gaml.

### Pengembangan Blok Fungsi dan Prosedur

Fungsi dan prosedur yang berada pada agen.gaml, experiment.gaml, fungsi.gaml, inits.gaml, dan reflex\_global.gaml dapat pula dimodifikasi jika pengguna merasakan adanya keperluan untuk itu. Penulisan fungsi dan prosedur dalam berkas-berkas mengikuti sintaks GAML.

Lebih detailnya, cara kerja fungsi dan prosedur dituliskan dalam setiap blok fungsi dalam berkas-berkas tersebut.

### Penggantian Berkas *Shapefile*

Berkas *shapefile* yang digunakan dalam produk model terdiri dari dua berkas. Kedua berkas ini wajib disertakan untuk keberjalanan simulasi. Berkas yang pertama yakni suatu *shapefile* berjenis area, yang menyusun semua bangunan berbeda yang terdapat dalam daerah yang dipetakan. Seluruh bangunan ini memiliki atribut jenis bangunannya masing-masing. Jenis objek *shapefile* adalah area, bukan point, karena terdapat interaksi antara agen manusia dengan agen bangunan, yakni agen manusia dapat beraktivitas keluar-masuk bangunan sepanjang simulasi.

Berkas yang kedua yakni *shapefile* berisi hanya satu blok area besar yang menyusun daerah yang dipetakan. Fungsi berkas ini adalah memperlihatkan batas daerah dengan jelas pada peta yang dimunculkan saat simulasi.

Berkas *shapefile* dapat diganti asalkan sesuai dengan keterangan di atas. Sebagai tambahan, pengguna perlu untuk menspesifikkan bangunan sekolah pada berkas *shapefile* pertama. Maksudnya, harus terdapat jenis sekolah yang berbeda menurut jenjangnya, yakni dibedakan menjadi TK, SD, SMP, dan SMA. Hal ini bertujuan agar simulasi lebih mendekati realitanya. Penggantian atribut ini dapat dilakukan dengan menggunakan perangkat lunak pengolah data GIS.

# Daftar Pustaka

[1] World Health Organization, “Statement on the second meeting of the International Health Regulations (2005) Emergency Committee regarding the outbreak of novel coronavirus (2019-nCoV)”, January 30, 2020. [Online]. Available: <https://www.who.int/news-room/detail/30-01-2020-statement-on-the-second-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-outbreak-of-novel-coronavirus-(2019-ncov)>. [Accessed Aug. 23, 2020].

[2] Gorbiano, M.I., “Breaking: Jokowi announces Indonesia’s first two confirmed COVID-19 cases”, *The Jakarta Post*, March 2, 2020. [Online]. Available: <https://www.thejakartapost.com/news/2020/03/02/breaking-jokowi-announces-indonesias-first-two-confirmed-covid-19-cases.html>. [Accessed Aug. 23, 2020].

[3] Presiden Republik Indonesia. (Mar. 13, 2020). *Keppres Nomor 7 Tahun 2020 tentang Gugus Tugas Percepatan Penanganan Coronavirus Disease 2019 (COVID-19).*

[4] Presiden Republik Indonesia. (Mar. 31, 2020). *PP Nomor 21 Tahun 2020 tentang Pembatasan Sosial Berskala Besar Dalam Rangka Percepatan Penanganan Coronavirus Disease 2019 (COVID-19).*

[5] Biswas, Md., De Pinho, M., and Paiva L.. “A SEIR Model for Control of Infectious Diseases with Constraints” in *Mathematical Biosciences and Engineering: MBE 11(4)*. Aug. 2014. [doi: 10.3934/mbe.2014.11.761]

[6] Godio, A., Pace, F. and Vergnano A.. “SEIR Modeling of the Italian Epidemic of SARS-CoV-2 Using Computational Swarm Intelligence” in *Int J Environ Res Public Health 17 (10)*. May 18 2020. [doi:10.3390/ijerph17103535]

[7] Bonabeau, E.. “Agent-based modeling: Methods and techniques for simulating human systems” in *Proceedings of the National Academy of Sciences 99 (suppl 3)*. May 14, 2020. [doi: 10.1073/pnas.082080899]

[8] Taillandier, P., Gaudou, B., Grignard, A.,Huynh, Q.-N., Marilleau, N., P. Caillou, P., Philippon, D., & Drogoul, A.. “Building, composing and experimenting complex spatial models with the GAMA platform” in *Geoinformatica*. 2019. 23 (2), pp. 299-322, [doi:10.1007/s10707-018-00339-6]

[9] Drogoul, A., Taillandier, P., Gaudou, B., Choisy, M., Chapuis, K., Quang Nghi, H., Doanh, N., Philippon, D., Brugière, A., and Larmande, P.. ”Designing social simulation to (seriously) support decision-making: COMOKIT, an agent-based modeling toolkit to analyze and compare the impacts of public health interventions against COVID-19” in *Review of Artificial Societies and Social Simulation*. Apr. 27, 2020. [Online] Available: <https://rofasss.org/2020/04/27/comokit> .

# Lampiran Desain Sistem

## Sistem Level 1

Berikut adalah sistem yang direncanakan untuk produk yang dikembangkan, dan akan dijelaskan arsitektur utama, interaksi pengguna, algoritma utama, dan penjelasan blok-subblok.

### Arsitektur Utama

**A screenshot of a cell phone

Description automatically generated**

Gambar 3-1. Arsitektur Utama

Gambar 3-1 menunjukkan arsitektur utama, dengan bagian kotak biru adalah blok-blok program yang didesain oleh mahasiswa. Gabungan blok program dan berkas GIS menyusun produk.

Dengan asumsi bahwa program model telah sepenuhnya ditulis dan berkas GIS tersedia, maka simulasi dimulai dengan menyiapkan simulasi terlebih dahulu, yang direpresentasikan oleh blok warna hijau di sebelah kiri. Persiapan simulasi dipicu oleh perintah pengguna melalui tombol pada aplikasi GAMA. Selanjutnya, persiapan simulasi itu dilakukan secara otomatis oleh GAMA, yang dilakukan dengan membaca berkas model simulasi. Model terdiri dari beberapa blok program. Setelah blok-blok ini dijalankan, maka tepat sebelum simulasi dijalankan, akan dijalankan fungsi untuk menerima masukan dari pengguna. Setelah itu, simulasi berjalan bersama dengan *display* grafik simulasi.

Blok-blok program yang didesain meliputi parameter model, *behavior* agen, manajemen klinis, *import* berkas GIS, generasi agen, penetapan aktivitas agen, baca input simulasi, dan display grafik-grafik. Dalam blok parameter model, akan didaftarkan parameter-parameter epidemiologis, misalnya proporsi kematian manusia, periode inkubasi, dll., serta parameter demografis, misalnya proporsi manusia berusia kerja, proporsi manusia berusia tua, dll.. Seluruh parameter ini dibentuk dalam variabel-variabel yang akan digunakan di blok program selanjutnya. Kemudian, blok *behavior* agen berisi kode program untuk menentukan tingkah laku agen manusia. Untuk membuat simulasi realistis, diperlukan adanya interaksi agen manusia dengan agen bangunan atau interaksi sesama agen manusia. Interaksi manusia-bangunan ditunjukkan dengan perginya manusia dari satu bangunan ke bangunan lain. Sedangkan interaksi manusia-manusia berupa penularan dari individu *infectious* ke individu *susceptible*. Interaksi-interaksi ini digambarkan melalui tingkah laku manusia: menjalankan aktivitas sehari-hari dan menjalankan aktivitas rekreasi. Selain itu, blok *behavior* agen juga meliputi kode-kode mengenai perubahan-perubahan atribut epidemiologis dari agen manusia, seperti perubahan status agen dari *susceptible* ke *exposed* ke *infectious* ke *recovered*.

Blok manajemen klinis berisi kode-kode yang memungkinkan agen manusia melakukan tes apabila status penyakitnya sudah menunjukkan gejala. Kemudian, agen manusia tersebut akan diberikan hasil tesnya: *true positive*, *false positive*, *true negative*, dan *false negative*. Agen manusia akan diberikan penanganan sesuai dengan hasilnya. Salah satu fungsi yang diperlukan pada blok ini adalah fungsi surveilans atau *contact tracing*, yaitu fungsi yang dipicu oleh hasil positif dari seorang agen manusia, yang menyebabkan agen manusia lain yang telah melakukan kontak dengan agen manusia yang pertama tadi untuk melakukan isolasi mandiri selama 14 hari, sesuai dengan peraturan yang berlaku di Indonesia.

Blok *import* berkas GIS berfungsi untuk membaca berkas GIS suatu daerah. Berkas GIS itu berisi bangunan-bangunan yang dimasukkan ke simulasi dalam bentuk agen bangunan. Kemudian, generasi agen berfungsi untuk membuat agen-agen manusia serta atribut-atribut dasarnya seperti usia, dan jenis kelamin. Dilanjutkan dengan blok penetapan aktivitas agen, yaitu penetapan atribut status pekerjaan serta jadwal melakukan pekerjaan tersebut.

Kemudian, dilanjutkan dengan blok untuk meminta *input* dari pengguna, yang berkenaan dengan variasi peraturan PSBB. Variasi peraturan PSBB ini akan mempengaruhi perkembangan simulasi. Misalnya, *input* penetapan mulainya PSBB yang dimulai setelah adanya 20 pasien positif di daerah itu akan memiliki hasil penyebaran penyakit yang lebih parah daripada *input* penetapan mulainya PSBB setelah adanya 2 pasien positif. Kemudian, blok selanjutnya yakni penampilan grafik, berfungsi untuk menunjukkan perkembangan epidemi. Grafik yang dimaksud adalah grafik jumlah agen manusia yang terkonfirmasi positif dalam suatu waktu pada daerah itu. Pada saat penampilan grafik ini, simulasi agen-agen manusia dan bangunan sudah dilakukan. Blok terakhir adalah perubahan peraturan, yakni perubahan dari situasi normal ke PSBB, atau dari PSBB kemudian dilonggarkan. Efeknya diberlakukannya PSBB dalam simulasi adalah pembatasan gerak agen manusia dan diturunkannya probabilitas seseorang untuk terinfeksi penyakit. Penurunan probabilitas ini dilakukan untuk merepresentasikan kewajiban menggunakan masker, budaya cuci tangan, dan meningkatnya kewaspadaan masyarakat ketika diberlakukan PSBB. Perubahan peraturan ini terjadi pada suatu titik di tengah simulasi.

### Interaksi Pengguna

Pengguna berinteraksi dengan program dalam beberapa jalan. Pertama, pengguna dapat mengganti berkas GIS untuk mengganti daerah yang akan disimulasikan. Kedua, yaitu pengguna dapat memencet tombol simulasi pada program GAMA untuk memulai simulasi. Setelah itu, proses akan diotomatisasi sampai dengan selangkah sebelum berjalannya simulasi. Saat itu adalah interaksi ketiga, yaitu memberikan *input* variasi peraturan PSBB. Interaksi keempat adalah perintah pengguna untuk menghentikan sementara simulasi, yang kemudian pengguna dapat menginspeksi agen simulasi atau sekadar meneliti grafik, atau menutup simulasi. Interaksi keempat ini dilakukan dengan tombol yang disediakan GAMA.

### Algoritma Utama

**A close up of a map

Description automatically generated**

Gambar 3-2. Algoritma Utama

Seperti pada gambar 3-2, algoritma utama dimulai dengan ditekannya tombol simulasi program yang terdapat pada GAMA. Program akan memulai dengan memuat parameter-parameter, fungsi-fungsi, yang akan digunakan nanti. Setelah itu, program menjalankan blok fungsi untuk meng-*import* berkas GIS, yang kemudian dari berkas tersebut akan dijalankan blok fungsi untuk membuat agen-agen bangunan. Setelah membuat agen bangunan, dijalankan fungsi untuk membuat agen manusia serta atribut-atribut sederhananya. Kemudian, dilanjutkan dengan fungsi pendefinisian aktivitas manusia, seperti bekerja, bersekolah, atau menganggur. Untuk agen manusia yang bekerja atau bersekolah, blok fungsi selanjutnya akan menentukan jadwal dari agen manusia tersebut. Setelah agen-agen lengkap, kemudian dijalankan fungsi untuk menerima *input* nilai variabel peraturan dari pengguna. Kemudian simulasi dijalankan, bersama dengan penampilan grafik data simulasi. Simulasi berjalan sampai dengan pengguna menghentikan simulasi melalui tombol di aplikasi GAMA.

### Subblok

Dari blok-blok program yang disebutkan pada bagian arsitektur utama di atas, dapat diturunkan subblok yang bersesuaian.

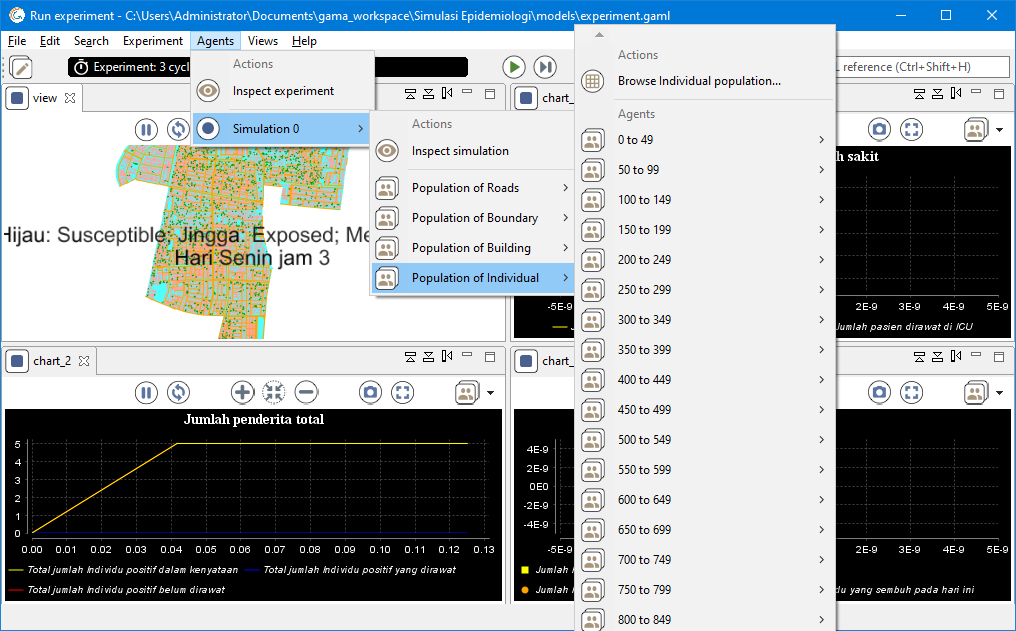
1. Blok parameter model bertujuan mendaftarkan seluruh parameter epidemiologis, parameter demografis, dan parameter lainnya yang dibutuhkan dalam simulasi. Blok ini tidak memerlukan subblok apapun.
2. Blok *behavior* agen bertujuan mendefinisikan tingkah laku agen manusia terhadap agen manusia lain atau agen bangunan. Blok ini akan memerlukan subblok atribut agen manusia, subblok *update* atribut epidemiologis, subblok penjalanan aktivitas, dan subblok penularan.
   1. Subblok atribut agen manusia bertujuan untuk mendaftarkan apa saja atribut yang mungkin dimiliki agen manusia. Selain atribut dasar seperti usia dan jenis kelamin, juga didaftarkan atribut epidemiologis seperti periode inkubasi dan status SEIR. Selain itu, juga didaftarkan atribut agenda mingguan yang akan digunakan sebagai acuan menjalankan aktivitas. Dalam subblok ini, atribut-atribut tersebut belum diberi nilai, hanya didefinisikan variabel kosongnya saja.
   2. Subblok *update* atribut epidemiologis, bertujuan untuk mengecek berapa lama sebuah agen manusia sudah terinfeksi, dan memberikan perubahan atribut apabila waktunya sudah mencapai parameter epidemiologis tertentu. Setelah mencapai *serial interval,* atribut status SEIR akan berubah dari *exposed* menjadi *infectious*. Sedangkan setelah mencapai periode inkubasi, seorang agen manusia akan dianggap menunjukkan gejala.
   3. Subblok penjalanan aktivitas bertujuan untuk memungkinkan agen manusia untuk berpindah lokasi dari satu bangunan ke bangunan lain.
   4. Subblok penularan bertujuan untuk memungkinkan penularan penyakit dari agen manusia beratribut status *infectious* ke agen berstatus *susceptible*.
3. Blok manajemen klinis bertujuan untuk memungkinkan alur yang dijalani agen manusia sejak terinfeksi, tes, hingga sembuh atau mati. Blok ini memerlukan subblok pengetesan, subblok isolasi, dan subblok *contact tracing*.
   1. Subblok pengetesan berfungsi untuk melakukan tes pada agen manusia yang menunjukkan gejala pada atributnya. Kemudian, hasil tes akan dikeluarkan menurut probabilitas tes dan dipengaruhi oleh usia agen manusia.
   2. Subblok isolasi bertujuan untuk menetapkan nilai dari atribut berupa status karantina. Status karantina akan dikembalikan nilainya ke semula ketika isolasi sudah selesai.
   3. Subblok *contact tracing* bertujuan untuk memforsir agen manusia ke isolasi mandiri di tempat tinggal masing-masing. Fungsi forsir ini dipicu oleh adanya agen manusia lain yang merupakan kontak erat dan dikonfirmasi positif melalui subblok pengetesan.
4. Blok *import* GIS bertujuan untuk menerima berkas GIS dan membuat agen bangunan yang sesuai dengan berkas GIS tersebut, sehingga saat simulasi dapat ditampilkan peta yang sesuai. Blok ini tidak memerlukan subblok apapun.
5. Blok generasi agen bertujuan untuk membuat semua agen manusia. Dalam blok ini juga ditetapkan nilai beberapa atribut dasar seperti usia dan jenis kelamin, berdasarkan probabilitas yang dituliskan sebagai parameter model. Blok ini tidak memerlukan subblok apapun.
6. Blok penetapan aktivitas bertujuan untuk menetapkan atribut aktivitas agen manusia, serta agendanya, sehingga terdapat jadwal melakukan aktivitas untuk setiap agen manusia. Blok ini memerlukan subblok berupa subblok pemilihan aktivitas dan subblok penentuan agenda.
   1. Subblok pemilihan aktivitas bertujuan untuk memilih menurut probabilitas, aktivitas agen manusia. Opsi yang mungkin adalah bekerja, bersekolah, atau menganggur.
   2. Subblok penentuan agenda bertujuan untuk menentukan jadwal aktivitas untuk agen manusia yang bekerja atau bersekolah. Untuk agen manusia yang bekerja, hari kerja ditentukan secara acak kecuali pekerjaan di kantor. Sedangkan untuk agen manusia yang bersekolah, hari sekolah ditentukan Senin sampai Jumat.
7. Blok pembacaan *input* simulasi bertujuan untuk menerima *input* dari pengguna berupa nilai variabel yang menentukan peraturan PSBB, misalnya durasi PSBB dan waktu pemicu pemberlakuan PSBB. Blok ini tidak memerlukan subblok lain.
8. Blok grafik simulasi bertujuan untuk memperlihatkan grafik data simulasi yang berisi jumlah agen manusia yang terinfeksi seiring waktu. Blok ini tidak memerlukan subblok lain.
9. Blok perubahan peraturan bertujuan merealisasikan transisi dari situasi normal ke situasi PSBB atau dari situasi PSBB ke situasi PSBB yang lebih longgar (*new normal*). Blok ini akan terbagi menjadi subblok *lockdown* dan subblok *new normal*.
   1. Subblok *lockdown* mengubah situasi dari normal ke PSBB dengan cara melakukan pengurangan hari kerja, peliburan sekolah, dan pengurangan aktivitas lain. Tujuannya untuk mewakili pembatasan gerak manusia ketika PSBB. Selain itu, akan dilakukan penurunan peluang transmisi penyakit, untuk merepresentasikan meningkatnya kewaspadaan masyarakat terhadap COVID-19 dan kewajiban meningkatkan kualitas hidup sehat.
   2. Subblok *new normal* akan melonggarkan *lockdown*, dengan tetap mempertahankan variabel-variabel penurunan peluang transmisi penyakit, tapi memperbolehkan aktivas bekerja dan aktivitas lain.

### Verifikasi Sistem

Untuk verifikasi sistem, terdapat kebutuhan berupa menjalankan aplikasi GAMA terlebih dahulu, lalu menjalankan simulasi, lalu melakukan inspeksi agen eksperimen: manusia dan bangunan. Karena itu, sebelum verifikasi setiap spesifikasi, akan dijelaskan prosedur menginspeksi agen-agen eksperimen simulasi, sebagai prosedur lanjutan dari yang telah dijelaskan di Bab 3.

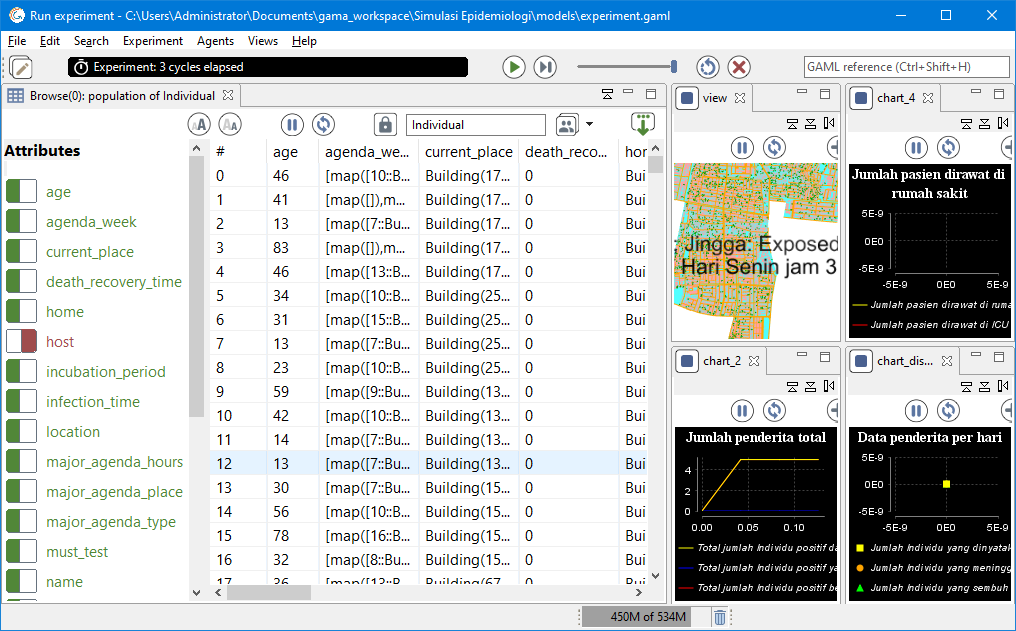
Berikut adalah langkah-langkah tersebut:

1. Pada saat simulasi, masuk ke *dropdown menu* “Agents” seperti pada gambar 6 sehingga inspeksi populasi manusia dapat dilakukan. Kemudian pilih “Browse Individual population”.



Gambar 3-3. Menu untuk inspeksi agen manusia

1. Tabel populasi manusia akan ditunjukkan. Kolom menunjukkan atribut agen manusia dan baris menunjukkan populasi manusia. Terlihat terdapat atribut-atribut yang di antaranya adalah usia, agenda mingguan, tempat tinggal, jenis agenda mingguan, periode inkubasi, dll..



Gambar 3-4. Tabel untuk inspeksi agen manusia

Untuk verifikasi tiap spesifikasi, akan diperlukan untuk mengobservasi atribut dari seorang atau lebih agen manusia.

Spesifikasi sistem serta prosedur verifikasinya ditulis pada tabel-tabel di bawah.

Tabel 3-1. Spesifikasi Sistem Level 1-1

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 1 | |
| Hal | Seluruh agen manusia dan bangunan yang didefinisikan di model dapat dibuat dalam simulasi |
| Asumsi | - |
| Kebutuhan Verifikasi | *Personal computer* dengan *operating system* Windows, MacOS, atau Linux, dan perangkat lunak GAMA. |
| Prosedur Verifikasi | Jalankan simulasi. Inspeksi agen simulasi, pastikan terdapat jumlah bangunan dan jumlah manusia yang sesuai dengan parameter model. |

Tabel 3-2. Spesifikasi Sistem Level 1-2

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 2 | |
| Hal | Blok fungsi *behavior* agen manusia bekerja tanpa kesalahan. |
| Asumsi | - |
| Kebutuhan Verifikasi | *Personal computer* dengan *operating system* Windows, MacOS, atau Linux, dan perangkat lunak GAMA. |
| Prosedur Verifikasi | Variasikan simulasi dengan memastikan jumlah agen manusia yang dibuat hanya sedikit. Jalankan simulasi. Inspeksi agen manusia, inspeksi atribut yang berkaitan dengan agenda dan aktivitas. Pastikan agenda/aktivitas sesuai dengan lokasi agen. Pastikan juga agenda/aktivitas sesuai dengan waktu simulasi. |

Tabel 3-3. Spesifikasi Sistem Level 1-3

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 3 | |
| Hal | Blok fungsi manajemen klinis untuk agen manusia terinfeksi berfungsi tanpa kesalahan. |
| Asumsi | - |
| Kebutuhan Verifikasi | *Personal computer* dengan *operating system* Windows, MacOS, atau Linux, dan perangkat lunak GAMA. |
| Prosedur Verifikasi | Variasikan simulasi dengan memastikan jumlah agen manusia yang dibuat hanya sedikit. Jalankan simulasi. Inspeksi agen manusia yang terinfeksi di awal. Observasi apakah atribut status SEIR dan status karantina seiring waktu sesuai dengan manajemen klinis yang semestinya. |

Tabel 3-4. Spesifikasi Sistem Level 1-4

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 4 | |
| Hal | Blok fungsi manajemen klinis di bagian *contact tracing* berfungsi tanpa kesalahan. |
| Asumsi | - |
| Kebutuhan Verifikasi | *Personal computer* dengan *operating system* Windows, MacOS, atau Linux, dan perangkat lunak GAMA. |
| Prosedur Verifikasi | Variasikan simulasi dengan memastikan jumlah agen manusia yang dibuat hanya sedikit. Jalankan simulasi. Inspeksi agen manusia yang terinfeksi di awal. Tunggu sampai agen manusia yang pertama dinyatakan positif. Observasi anggota keluarga dari agen tersebut. |

Tabel 3-5. Spesifikasi Sistem Level 1-5

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 5 | |
| Hal | Grafik yang ditampilkan dalam simulasi sesuai dengan kenyataan pada agen manusia. |
| Asumsi | - |
| Kebutuhan Verifikasi | *Personal computer* dengan *operating system* Windows, MacOS, atau Linux, dan perangkat lunak GAMA. |
| Prosedur Verifikasi | Variasikan simulasi dengan memastikan jumlah agen manusia yang dibuat hanya sedikit. Jalankan simulasi. Setelah grafik menunjukkan individu terinfeksi sebanyak 20, hentikan sementara simulasinya . Inspeksi agen manusia. Pastikan dari atribut SEIR bahwa jumlah individu terinfeksi memang 20. |

Tabel 3-6. Spesifikasi Sistem Level 1-6

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 6 | |
| Hal | Pemberlakuan PSBB pada simulasi bekerja tanpa kesalahan. |
| Asumsi | - |
| Kebutuhan Verifikasi | *Personal computer* dengan *operating system* Windows, MacOS, atau Linux, dan perangkat lunak GAMA. |
| Prosedur Verifikasi | Variasikan simulasi dengan memastikan jumlah agen manusia yang dibuat hanya sedikit. Jalankan simulasi hingga pemberlakuan PSBB. Pastikan melalui peta yang ditunjukkan pada simulasi bahwa aktivitas manusia berkurang. |

# Lampiran Implementasi Sistem

## Sistem Level 1-1

### Implementasi

Sistem level 1-1 adalah sistem untuk membuat seluruh agen manusia dan bangunan yang ada dalam simulasi. Keluarga dibuat per bangunan rumah, jadi jumlah agen manusia menurut pada jumlah agen bangunan rumah. Implementasinya dalam *source code* ditunjukkan berikut.

action population\_generation {

/\*

\* Action untuk membuat populasi entitas Individual.

\* Melibatkan penentuan bangunan rekreasi. Tiap anggota keluarga memiliki jenis rekreasi persis.

\* Kemudian keluarga dibentuk dengan probabilitas-probabilitas di parameters.gaml.

\* Dalam action ini ditentukan atribut-atribut seperti usia, jenis kelamin, komorbiditas, rumah,

\* dan bangunan rekreasi.

\*/

ask **num\_homes** **among** **homes** {

/\*

\* Dilakukan dengan pertama-tama memilih

\* entitas Bangunan rumah secara random, kemudian memilih 3 bangunan untuk rekreasi,

\* 1 bangunan untuk belanja, dan memilih rumah-rumah lain yang berada dalam radius tertentu,

\* untuk dijadikan tetangga. Semua bangunan digabung dalam list minor\_buildings\_temp.

\*/

list<**Building**> minor\_buildings\_temp **<-** [];

loop **times**: 3 {

string recreation **<-** **one\_of**(**possible\_recreations**);

**minor\_buildings\_temp** **<<** **one\_of**(**buildings\_per\_activity**[**recreation**]);

}

string shopping **<-** **one\_of** (**possible\_shopping**);

**minor\_buildings\_temp** **<<** **one\_of** (**buildings\_per\_activity**[**shopping**]);

list<**Building**> neighbors;

ask **Boundary** {

if (*myself* **overlaps** *self*) {

**neighbors** **<-** **homes\_at\_rw** **at\_distance** 25;

}

}

loop b **over**:**neighbors** {**minor\_buildings\_temp** **<<** **b**;}

if (**flip**(**proba\_active\_family**)) {

// Keluarga aktif didefinisikan sebagai keluarga yang setidaknya ada ayah dan ibu.

create **Individual** {

**age** **<-** **rnd**(**min\_working\_age**,**max\_working\_age**);

**sex** **<-** 1;

**home** **<-** *myself*;

*myself*.**residents** **<<** *self*;

**recreation\_buildings** **<-** **minor\_buildings\_temp**;

}

create **Individual** {

**age** **<-** **rnd**(**min\_working\_age**,**max\_working\_age**);

**sex** **<-** 0;

**home** **<-** *myself*;

*myself*.**residents** **<<** *self*;

**recreation\_buildings** **<-** **minor\_buildings\_temp**;

}

int num\_children **<-** **rnd**(0,**max\_num\_children**);

loop **times**: **num\_children** {

create **Individual** {

**age** **<-** **rnd**(**min\_age**,**max\_student\_age**);

**sex** **<-** **rnd**(0,1);

**home** **<-** *myself*;

*myself*.**residents** **<<** *self*;

**recreation\_buildings** **<-** **minor\_buildings\_temp**;

}

}

if (**flip**(**proba\_grandfather**)) {

create **Individual** {

**age** **<-** **rnd**(**max\_working\_age+**1,**max\_age**);

**sex** **<-** 1;

**home** **<-** *myself*;

*myself*.**residents** **<<** *self*;

**recreation\_buildings** **<-** **minor\_buildings\_temp**;

}

}

if(**flip**(**proba\_grandmother**)) {

create **Individual** {

**age** **<-** **rnd**(**max\_working\_age+**1,**max\_age**);

**sex** **<-** 0;

**home** **<-** *myself*;

*myself*.**residents** **<<** *self*;

**recreation\_buildings** **<-** **minor\_buildings\_temp**;

}

}

if(**flip**(**proba\_rewang**)) {

create **Individual** {

**age** **<-** **rnd**(**min\_working\_age**,**max\_working\_age**);

**sex** **<-** **rnd**(0,1);

**home** **<-** *myself*;

*myself*.**residents** **<<** *self*;

**recreation\_buildings** **<-** **minor\_buildings\_temp**;

}

}

} else {

// Individual yang tinggal sendirian

create **Individual** {

**age** **<-** **rnd**(**min\_working\_age**,**max\_age**);

**sex** **<-** **rnd**(0,1);

**home** **<-** *myself*;

*myself*.**residents** **<<** *self*;

**recreation\_buildings** **<-** **minor\_buildings\_temp**;

}

}

}

**num\_individuals\_init** **<-** **length** (**Individual**);

}

*Source code* di atas menunjukkan implementasi melalui *action* population\_generation. Kata kunci *action* dan *reflex* adalah representasi *function* dalam GAML (GAMA *Modelling Language*), dipakai karena *source code* memang ditulis dalam bahasa itu. Perbedaannya, *action* dijalankan ketika pemanggilan fungsi, sedangkan *reflex* dikerjakan pada tiap langkah simulasi.

Dalam sistem ini, agen bangunan dibuat langsung dari berkas *shapefile* yang disiapkan, dan proses *import* berkas *shapefile* ini telah dipermudah dalam aplikasi GAMA sendiri, sehingga hal selanjutnya yang diperlukan adalah menyalin atribut jenis bangunan dari *shapefile* ke setiap agen bangunan yang dibuat. Kemudian, untuk setiap agen bangunan yang berjenis rumah, dibuatlah beberapa agen manusia. Terdapat probabilitas bahwa agen manusia dibuat lebih dari satu, merepresentasikan keluarga, dan probabilitas bahwa agen manusia dibuat hanya satu saja, merepresentasikan orang yang tinggal sendiri. Agen manusia dibuat mulai dari ayah dan ibu, kemudian sejumlah anak menurut probabilitas jumlah anak, kemudian kakek dan nenek sesuai probabilitasnya. Selain itu, terdapat pula probabilitas untuk menambahkan satu agen manusia lagi berupa asisten rumah tangga.

Dalam *action* yang sama, juga ditentukan bangunan rekreasi dan penentuan bangunan tetangga. Pemilihan bangunan rekreasi secara acak, sedangkan tetangga ditentukan dengan radius kedekatan bangunan. Jika terdapat rumah dalam radius 25 satuan, maka rumah itu dianggap sebagai tetangga. Penentuan rekreasi dan tetangga dilakukan secara kolektif per keluarga, salah satu tujuannya untuk simplifikasi pemodelan.

### Pengujian

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 - 1 | |
| Hal | Seluruh agen manusia dan bangunan yang didefinisikan di model dapat dibuat dalam simulasi |
| Hasil | Setelah inspeksi, jumlah agen bangunan sama dengan yang ditentukan sebelumnya dalam parameter model. |

## Sistem Level 1-2

### Implementasi

Sistem level 1-2 adalah sistem untuk memungkinkan agen manusia memiliki tingkah laku yang sesuai yang diinginkan dalam simulasi. Tingkah laku manusia di model dibagi 2: aktivitas besar dan aktivitas kecil. Aktivitas besar mewakili kegiatan yang rutin dan wajib dilakukan setiap minggu, yakni bekerja atau bersekolah. Aktivitas kecil menanadakan kegiatan rekreasi, seperti pergi ke restoran, yang dilakukan menurut probabilitas aktivitas, alih-alih wajib dilakukan. Implementasi dalam *source code* dicantumkan di bawah.

action assign\_school\_work {

/\*

\* action untuk meng-assign jenis dan lokasi sekolah/kerja (tapi bukan agendanya)

\* Sekolah yang sesuai dengan usia dipilih berdasarkan kedekatan (asumsikan zonasi).

\* major\_agenda\_type akan berisi jenis pekerjaan, atau "sekolah", atau "none".

\*/

ask **Individual** {

do **enter\_building**(**home**);

if (**age** **>=** **min\_student\_age** **and** **age** **<=** **max\_student\_age**) {

list<**int**> L **<-** **match\_age**(**possible\_schools**.**keys**);

string temp **<-** **possible\_schools**[**L**];

list<**Building**> schools **<-** **buildings\_per\_activity**[**temp**];

**major\_agenda\_place** **<-** **schools** **closest\_to** *self*;

**major\_agenda\_place**.**residents** **<<** *self*;

**major\_agenda\_type** **<-** "school";

} else if (**age** **>=** **min\_working\_age** **and** **age** **<=** **max\_working\_age**) {

bool is\_employed;

if (**sex=**1) {

**is\_employed** **<-** **flip**(**proba\_employed\_male**);

} else {

**is\_employed** **<-** **flip**(**proba\_employed\_female**);

}

if (**is\_employed**) {

if (**flip**(**proba\_works\_at\_home**)) {

// Memperhitungkan kemungkinan orang yang memang kerjanya

// dari rumah.

**major\_agenda\_place** **<-** **home**;

**major\_agenda\_type** **<-** "swasta";

} else {

**major\_agenda\_type** **<-** **rnd\_choice**(**possible\_workplaces**);

list<**Building**> working\_places;

if (**major\_agenda\_type** **=** "guru") {

string school\_type **<-** **one\_of**(**possible\_schools**.**values**);

**working\_places** **<-** **buildings\_per\_activity**[**school\_type**];

} else {

**working\_places** **<-** **buildings\_per\_activity**[**major\_agenda\_type**];

}

**major\_agenda\_place** **<-** **one\_of**(**working\_places**);

**major\_agenda\_place**.**residents** **<<** *self*;

}

} else {

**major\_agenda\_type** **<-** "none";

}

} else {

**major\_agenda\_type** **<-** "none";

}

}

**individuals\_per\_profession** **<-** (**Individual** **group\_by** (*each*.**major\_agenda\_type**));

}

action assign\_major\_agenda {

/\*

\* action untuk menambahkan agenda sekolah/kerja sesuai dengan yang sudah

\* di-assign di action sebelumnya. Atribut Individual yang diisi adalah

\* agenda\_week.

\*

\*/

ask **Individual** {

loop i **from**: 0 **to**: 6 {

**agenda\_week** **<<** [];

}

if (**major\_agenda\_type** **=** "school") {

loop L **over**: **possible\_schools**.**keys** {

if (**age** **>=** **min**(**L**) **and** **age** **<=** **max**(**L**)) {

string temp **<-** **possible\_schools**[**L**];

loop d **over**: [0,1,2,3,4] {

int start\_hour;

int end\_hour;

// Penentuan jam masuk dan pulang untuk setiap jenis sekolah.

switch **temp** {

match "tk" {

**start\_hour** **<-** 7;

**end\_hour** **<-** 10;

}

match "sd" {

**start\_hour** **<-** 7;

**end\_hour** **<-** 13;

}

match "smp" {

**start\_hour** **<-** 7;

**end\_hour** **<-** 15;

}

match "sma" {

**start\_hour** **<-** 7;

**end\_hour** **<-** 15;

}

match "college" {

**start\_hour** **<-** **rnd**(7,10);

**end\_hour** **<-** **rnd**(14,18);

}

}

**major\_agenda\_hours**[**d**] **<-** [**start\_hour**, **end\_hour**];

(**agenda\_week**[**d**])[**start\_hour**] **<-** *self*.**major\_agenda\_place**;

(**agenda\_week**[**d**])[**end\_hour**] **<-** *self*.**home**;

}

}

}

} else if (!(**major\_agenda\_type** **in** ["none", "school"])) {

// Penetapan jam kerja dan pulang.

if (**major\_agenda\_type** **in** **possible\_offices**) {

// Untuk yang bekerja di kantor, maka hari kerjanya 0-4

// yakni hari Senin-Jumat.

loop d **over**: [0,1,2,3,4] {

int start\_hour **<-** **rnd**(7,10);

int end\_hour **<-** **rnd**(15,19);

**major\_agenda\_hours**[**d**] **<-** [**start\_hour**, **end\_hour**];

**agenda\_week**[**d**][**start\_hour**] **<-** *self*.**major\_agenda\_place**;

**agenda\_week**[**d**][**end\_hour**] **<-** *self*.**home**;

}

} else if (**major\_agenda\_type** **!=** **none**) {

// Untuk yang bekerja di selain kantor, hari kerja acak.

list<**int**> working\_days **<-** [];

loop i **over**: [0,1,2,3,4,5,6] {

**working\_days** **<<** **i**;

}

// Sebanyak 3 hari kerja dibuang secara random dari semua

// hari yang mungkin (0-6)

int i **<-** **one\_of**(**working\_days**);

**working\_days >> i;**

**i** **<-** **one\_of**(**working\_days**);

**working\_days >> i;**

**i** **<-** **one\_of**(**working\_days**);

**working\_days >> i;**

// Penetapan jam kerja dan pulang

loop d **over**: **working\_days** {

int start\_hour **<-** **rnd** (7,10);

int end\_hour **<-** **rnd**(16,21);

**major\_agenda\_hours**[**d**] **<-** [**start\_hour**, **end\_hour**];

(**agenda\_week**[**d**])[**start\_hour**] **<-** *self*.**major\_agenda\_place**;

(**agenda\_week**[**d**])[**end\_hour**] **<-** *self*.**home**;

}

}

}

}

}

*Source code* menunjukkan *action* assign\_school\_work dan assign\_major\_agenda. Pada *action* yang disebutkan lebih dulu, pembagian jenis aktivitas besar dilakukan untuk semua agen manusia yang berusia sekolah sampai usia pensiun. Tetapi tetap ada probabilitas menganggur untuk manusia berusia kerja. Dengan ini, manusia yang menganggur terdiri dari populasi manusia yang belum usia sekolah, yang berusia lanjut, dan beberapa dari yang berusia kerja. Semua manusia usia sekolah dianggap bersekolah.

Dalam *action* yang disebutkan lebih akhir, ditentukan hari kerja atau hari sekolah. Hari kerja untuk orang-orang yang bekerja di kantor serta hari sekolah ditentukan tetap yakni hari Senin sampai Jumat. Sedangkan untuk orang-orang yang bekerja di tempat lain, hari kerja diacak tetapi tetap diberi batas jumlah hari kerja. Setelah hari kerja ditentukan, maka jam kerja atau jam sekolah serta jam pulang ditentukan. Jam-jam ini diacak untuk manusia yang bekerja, sementara untuk jam sekolah ditetapkan sesuai jenjang pendidikannya (misalnya, murid SMA masuk pukul 7.00 dan pulang pukul 15.00).

### Pengujian

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 - 2 | |
| Hal | Blok fungsi *behavior* agen manusia bekerja tanpa kesalahan. |
| Hasil | Setelah inspeksi agen manusia dan atribut “major\_agenda\_type”, ditemukan bahwa semua agen memiliki nilai untuk atribut ini, yakni jenis pekerjaan, status bersekolah, atau “none” jika agen menganggur. Untuk agen beratribut “major\_agenda\_type” yang tidak bernilai “none”, maka “agenda\_week” berisi hari dan jam yang dipetakan ke jenis bangunan yang sesuai dengan pekerjaan/sekolah. Pada hari dan jam yang sesuai, agen manusia terdapat di lokasi yang sesuai. |

## Sistem Level 1-3

### Implementasi

Sistem level 1-3 adalah sistem manajemen klinis. Sistem diterapkan dengan menggunakan dua fungsi: fungsi pengetesan manusia dan fungsi durasi isolasi. Implementasi dalam *source code* dicantumkan di bawah.

/\*

\* Individu akan melakukan tes dengan syarat must\_test = true atau karena keinginan atau random.

\* must\_test true jika memiliki gejala.

\* Tes karena keinginan/random menurut proba\_voluntary\_random\_test (parameters.gaml)

\*

\* Mekanisme tes dibagi menjadi dua.

\* Yakni ketika penerima tes memang sudah terinfeksi penyakit, dan ketika penerima tidak terinfeksi.

\* Untuk yang sudah terinfeksi penyakit, terdapat dua kemungkinan: True Positive dan False Negative.

\* Untuk yang tidak terinfeksi, terdapat dua kemungkinan: True Negative dan False Positive.

\* Individu yang tes karena gejala (suspek, probabel) sudah pasti terinfeksi.

\* Individu yang tes karena keinginan/random belum pasti terinfeksi.

\*

\*/

reflex take\_test when: ((**current\_hour** **=** 6) and ((**must\_test** **and** **flip**(**proba\_test**)) **or** (**flip**(**proba\_voluntary\_random\_test**) **and** **status** **!=** **recovered** **and** **quarantine\_status** **=** **none**)) ) {

Building the\_hospital **<-** **one\_of** (**buildings\_per\_activity**["hospital"]);

do **enter\_building**(**the\_hospital**);

//Jika statusnya memang terinfeksi (exposed atau infectious)

if (**status** **in** [**exposed**, **infectious**]) {

float proba **<-** **sensitivity\_pcr** **\*** **test\_accuracy**;

bool positive\_result **<-** **flip**(**proba**);

// Jika True Positive

if (**positive\_result**) {

// Penentuan tempat perawatan

switch **severity** {

match **mild** {

**quarantine\_status** **<-** **hospital**;

}

match **moderate** {

**quarantine\_status** **<-** **hospital**;

}

match **severe** {

**quarantine\_status** **<-** **ICU**;

}

match **deadly** {

**quarantine\_status** **<-** **ICU**;

}

match **none** {

**quarantine\_status** **<-** **hospital**;

}

}

// Reset waktu karantina. Berguna untuk kasus orang yang sebelum ini lakukan karantina mandiri di rumah.

**quarantine\_time** **<-** 0;

do **contact\_trace**;

**must\_test** **<-** false;

} else {

// Jika False Negative, dipulangkan tapi must\_test tidak dijadikan false.

// Dengan tujuan Individual ini akan mengambil tes lagi keesokan hari.

**quarantine\_status** **<-** **none**;

do **enter\_building**(**home**);

}

// Jika sebenarnya tidak infeksi (susceptible, recovered)

} else {

float proba **<-** **specificity\_pcr** **\*** **test\_accuracy**;

if (**flip**(**proba**)) {

// Jika True Negative, dipulangkan.

**quarantine\_status** **<-** **none**;

do **enter\_building**(**home**);

**must\_test** **<-** false;

} else {

// Jika False Positive

**quarantine\_status** **<-** **hospital**;

do **contact\_trace**;

**death\_recovery\_time** **<-** 24 **\*** 14;

**quarantine\_time** **<-** 0;

**must\_test** **<-** false;

}

}

}

reflex update\_quarantine when: (**quarantine\_status** **in** ["home", **hospital**, **ICU**]) {

/\*

\* Reflex untuk tracking waktu karantina. Jika sudah recovered, Individual akan

\* dilepas dari karantina atau mati. Atau jika sebenarnya tidak terinfeksi tapi

\* masuk rumah sakit (false positive), dikeluarkan setelah 14 hari.

\*/

**quarantine\_time** **<-** **quarantine\_time** **+** 1;

if (**status** **=** **recovered**) {

**quarantine\_status** **<-** **none**;

**quarantine\_time** **<-** 0;

do **enter\_building**(**home**);

} else if (**status** **in** [**susceptible**, **recovered**]) {

if (**quarantine\_time** **>** (24 **\*** 14)) {

**quarantine\_status** **<-** **none**;

**quarantine\_time** **<-** 0;

}

}

Untuk implementasi, sesuai *source code*, digunakan *reflex* take\_test serta during treatment. take\_test memungkinkan agen manusia untuk menjalani tes dengan beberapa syarat. Untuk simplifikasi, tes hanya dilakukan pada pukul 6.00; tujuannya untuk mengurangi beban komputasi. Selain itu, syaratnya adalah agen tersebut memiliki gejala COVID-19. Tetapi, terdapat pula peluang bagi agen manusia yang tidak memiliki gejala untuk mengambil tes juga, justifikasi adanya peluang ini adalah mewakili agen manusia yang merasa perlu untuk mengambil tes karena cemas dengan gejala yang ia miliki, padahal ia sebenarnya tidak terinfeksi. *Reflex* ini memungkinkan 4 hasil tes: *true positive, true negative, false positive,* dan *false negative*. Empat hasil ini dimungkinkan karena pengguna akan diminta memberikan input akurasi tes di awal simulasi, yang akan memungkinkan hasil tes yang salah (*false positive* dan *false negative*).

Dalam *reflex* update\_quarantine, diatur atribut agen manusia yang berkaitan dengan status karantina. Untuk agen manusia yang sedang dikarantina, atribut waktu karantina di-*increment*. Untuk kasus karantina mandiri. ketika waktu karantina ini melebihi 14 hari, maka karantina akan diakhiri, kemudian nilai atribut status karantina diganti menjadi “none”. Sedangkan untuk kasus karantina di rumah sakit atau ICU, agen akan dilepas dari karantina apabila status SEIR telah berganti dari *infectious* menjadi *recovered* atau meninggal. Atribut agen manusia berupa status karantina ini kemudian diusahakan menjadi syarat agen manusia tersebut dapat memanggil *action* dan *reflex* yang berkaitan dengan aktivitas. Sehingga, individu yang sedang karantina tidak dapat melakukan aktivitas dengan normal, tetap diam di rumah/rumah sakit/ICU.

### Pengujian

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 - 3 | |
| Hal | Blok fungsi manajemen klinis untuk agen manusia terinfeksi berfungsi tanpa kesalahan. |
| Hasil | Ketika diobservasi agen manusia yang terinfeksi, status SEIR berubah dari *exposed* ke *infectious* sesuai dengan durasi *serial interval*. Ketika agen mengalami pengetesan, status karantina sesuai, berubah dari tidak ada status karantina menjadi berstatus karantina “rumah sakit”. |

## Sistem Level 1-4

Implementasi

Sistem level 1-4 adalah sistem manajemen klinis bagian *contact tracing*. Sistem melibatkan fungsi yang memaksa agen manusia dengan kontak erat untuk menjalani isolasi mandiri. Dalam isolasi tersebut, jika kemudian muncul gejala, maka harus melakukan pengetesan. Dalam implementasinya, kontak erat yang ditentukan oleh fungsi berisi anggota keluarga dan teman kerja di tempat kerja. Implementasi dalam *source code* dicantumkan di bawah.

action contact\_trace {

/\* Action untuk melakukan contact tracing pada keluarga dan teman kerja

\* dari seorang Individual yang dinyatakan positif terinfeksi oleh

\* rumah sakit.

\* Kontak-kontak yang dimaksud akan diubah di variabel must\_test

\* menjadi bernilai true.

\*/

// Mengisi contacts dengan anggota keluarga dan teman kerja.

list<**Individual**> contacts **<-** *self*.**home**.**residents** **-** *self*;

if (not (**major\_agenda\_type** **in** ["none", "swasta"])) {

**contacts** **<-** **contacts** **+** *self*.**major\_agenda\_place**.**residents** **-** *self*;

}

// Menentukan jumlah kontak yang gagal di-trace, berdasarkan tracing effectivity.

int num\_contacts\_untraced **<-** **round**((1**-tracing\_effectivity**)**\*length**(**contacts**));

loop **times**: **num\_contacts\_untraced** {

**contacts >> one\_of(contacts);**

}

// Mengubah status karantina Individu menjadi karantina mandiri.

// Status bisa berubah jika Individu terinfeksi dan simptomatik,

// dan sudah melewati periode inkubasi (reflex during\_infection)

if (**contacts** **!=** []) {

ask **contacts** {

if (**quarantine\_status** **=** **none**) {

**quarantine\_status** **<-** "home";

}

}

}

}

Seperti pada *source code*, *action* contact\_trace akan memungkinkan mekanisme surveilans kontak erat dari agen yang terkonfirmasi positif oleh tes pada sistem level 1-3. Ketika hasilnya positif (*true positive* maupun *false positive*), *action* ini akan dipanggil. Hal ini memicu untuk mengubah atribut dari agen manusia yang termasuk anggota keluarga dan teman kerja. Atribut yang diganti adalah atribut status karantina, dan diubah sehingga agen ditahan di bangunan rumah sebagai representasi isolasi mandiri. Tetapi, agen dapat keluar apabila dalam isolasi mandiri 14 hari tersebut, ia menunjukkan gejala. Hal itu akan memicu *reflex* take\_test pada sistem level 1-3.

### Pengujian

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 4 | |
| Hal | Blok fungsi manajemen klinis di bagian *contact tracing* berfungsi tanpa kesalahan. |
| Hasil | Ketika diobservasi agen manusia yang terinfeksi, kemudian dipindahkan status karantinanya menjadi “rumah sakit”, maka ini merupakan kasus konfirmasi positif. Ketika diinspeksi anggota keluarga dan teman kerja, atribut status karantina agen-agen tersebut bernilai “rumah”, artinya melakukan isolasi mandiri sesuai manajemen klinis. |

## Sistem Level 1-5

### Implementasi

Sistem level 1-5 adalah sistem pembuatan grafik simulasi. Dibuat 3 grafik simulasi, yaitu grafik jumlah agen manusia terkonfirmasi positif per hari (termasuk *false positive*), grafik jumlah agen manusia terinfeksi sebenarnya, serta grafik jumlah pasien di rumah sakit / ICU. Implementasi dalam *source code* dicantumkan di bawah.

output {

layout #*split* **consoles**:false **editors**:false **navigator**:false;

display chart\_1 **refresh**:(**current\_hour** **=** 0) {

chart "Data harian" **type:** **xy** **background**: #*white* **axes**:#*black* **color**: #*black* **tick\_line\_color**: #*grey* {

data "Jumlah Individu yang dinyatakan positif pada hari ini"

**value:** {**current\_day**, **positives\_today**} **color**: #*red* **line\_visible**:false;

data "Jumlah Individu yang meninggal pada hari ini"

**value:** {**current\_day**, **deaths\_today**} **color**: #*green* **line\_visible**:false;

data "Jumlah Individu yang sembuh pada hari ini"

**value:** {**current\_day**, **recoveries\_today**} **color**: #*blue* **line\_visible**:false;

}

}

display view **draw\_env**:false **type:opengl** {

species **Boundary** **aspect:geom**;

species **Roads** **aspect:geom**;

species **Building** **aspect:geom**;

species **Individual** **aspect:circle**;

graphics **title** {

draw **simulation\_name** **color**: #*black* **anchor**: #*top\_center*;

draw **legend** **color**: #*black* **anchor**: #*bottom\_center*;

}

}

display chart\_2 **refresh**:(**current\_hour** **=** 0) {

chart "Data total" **type:** **xy** **background**: #*white* **axes**:#*white* **color**: #*black* **tick\_line\_color**: #*grey*{

data "Jumlah Individu terinfeksi dalam kenyataannya"

**value:** {**cycle/**24, **infections**} **color**: #*green* **marker**:false;

data "Jumlah Individu dirawat di rumah sakit, termasuk ICU"

**value:** {**cycle/**24, **hospitalizations**} **color**: #*blue* **marker**:false;

data "Jumlah Individu dirawat di ICU"

**value:** {**cycle/**24, **treated\_ICU**} **color**: #*red* **marker**:false;

}

}

display chart\_3 **refresh**:(**current\_hour** **=** 0) {

chart "Data sembuh dan meninggal" **type:** **xy** **background**: #*white* **axes**:#*white* **color**: #*black* **tick\_line\_color**: #*grey*{

data "Jumlah Individu dirawat di rumah sakit, termasuk ICU"

**value:** {**cycle/**24, **hospitalizations**} **color**: #*green* **marker**:false;

data "Jumlah Individu yang sembuh"

**value:** {**cycle/**24, **recoveries**} **color**: #*blue* **marker**:false;

data "Jumlah Individu yang meninggal"

**value:** {**cycle/**24, **deaths**} **color**: #*red* **marker**:false;

}

}

}

Data grafik ini didapatkan dari perintah untuk menghitung jumlah agen manusia dengan atribut tertentu, yang dilakukan dengan mengakses *built-in function*.

### Pengujian

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 5 | |
| Hal | Grafik yang ditampilkan dalam simulasi sesuai dengan kenyataan pada agen manusia. |
| Hasil | Ketika dilakukan inspeksi di tengah simulasi, jumlah agen manusia terkonfirmasi positif di grafik sesuai dengan jumlah sebenarnya pada menu inspeksi agen. |

## Sistem Level 1-6

### Implementasi

Sistem level 1-6 adalah sistem yang berisi fungsi pemberlakuan PSBB. Dalam implementasinya, ditambahkan pula keadaan pelonggaran PSBB, atau biasa disebut keadaan *new normal*. Perbedaan PSBB dan *new normal* adalah bahwa aktivitas bekerja pada saat *new normal* tidak diliburkan, tetapi parameter yang menurunkan probabilitas infeksi dipertahankan. Dengan asumsi, bahwa kewaspadaan masyarakat tetap tinggi terhadap COVID-19, dan masyarakat dihimbau untuk mempertahankan gaya hidup sehat. Implementasi sistem ini dicantumkan di bawah.

reflex enforce\_lockdown when: **hospitalizations** **>=** **lockdown\_threshold**

**and** not **lockdown** **and** not **new\_normal** {

/\*

\* reflex global untuk melakukan lockdown, dengan syarat total

\* Individual yang dirumahsakitkan melebihi threshold.

\*/

loop occupation **over**: **individuals\_per\_profession**.**keys** {

if (**occupation** **in** (**possible\_shopping** **+** **possible\_offices**)) {

// Menyuruh Individual yang bekerja di tempat belanja atau kantor

// untuk wfh pada 3 hari dari keseluruhan hari kerjanya.

ask **individuals\_per\_profession**[**occupation**] {

int wfh\_day\_1 **<-** **one\_of** (**major\_agenda\_hours**.**keys**);

int wfh\_day\_2 **<-** **one\_of** (**major\_agenda\_hours**.**keys** **-** **wfh\_day\_1**);

int wfh\_day\_3 **<-** **one\_of** (**major\_agenda\_hours**.**keys** **-** **wfh\_day\_1** **-** **wfh\_day\_2**);

int start\_hour **<-** **min**(**major\_agenda\_hours**[**wfh\_day\_1**]);

**agenda\_week**[**wfh\_day\_1**][**start\_hour**] **<-** **home**;

**start\_hour** **<-** **min**(**major\_agenda\_hours**[**wfh\_day\_2**]);

**agenda\_week**[**wfh\_day\_2**][**start\_hour**] **<-** **home**;

**start\_hour** **<-** **min**(**major\_agenda\_hours**[**wfh\_day\_3**]);

**agenda\_week**[**wfh\_day\_3**][**start\_hour**] **<-** **home**;

}

} else if (**occupation** **!=** **hospital** **and** **occupation** **!=** **none**) {

// Menyuruh Individual yang bekerja di tempat selain kantor dan tempat

// belanja, untuk wfh secara menyeluruh

ask **individuals\_per\_profession**[**occupation**] {

loop agenda\_day **over**:**agenda\_week** {

int start\_hour **<-** **min**(**agenda\_day**.**keys**);

**agenda\_day**[**start\_hour**] **<-** **home**;

}

}

}

}

**lockdown** **<-** true;

**activity\_reduction\_factor** **<-** 0.75;

**mask\_usage\_proportion** **<-** 0.9;

**infection\_reduction\_factor** **<-** 0.75;

}

reflex lift\_lockdown when: **pos\_decrease\_counter** **=** **new\_normal\_threshold**

**and** **lockdown** **and** not **new\_normal** {

/\*

\* Melakukan pengangkatan lockdown ketika counter penurunan

\* jumlah positif mencapai threshold.

\* Tiap Individual yang bekerja tidak di rumah sakit dapat

\* bekerja kembali seperti biasa.

\*/

loop occupation **over**: **individuals\_per\_profession**.**keys** {

if (**occupation** **!=** **none**) {

ask **individuals\_per\_profession**[**occupation**] {

loop agenda\_day **over**: **agenda\_week** {

int start\_hour **<-** **min**(**agenda\_day**.**keys**);

**agenda\_day**[**start\_hour**] **<-** **major\_agenda\_place**;

}

}

}

}

**lockdown** **<-** false;

**new\_normal** **<-** true;

**activity\_reduction\_factor** **<-** 0.25;

**mask\_usage\_proportion** **<-** 0.9;

**infection\_reduction\_factor** **<-** 0.75;

}

Seperti pada *source code*, *reflex* dilakukan ketika situasi simulasi sudah mencapai syarat. Yakni, ketika jumlah agen manusia yang terkonfirmasi positif sudah mencapai batas nilai yang ditentukan sendiri oleh pengguna pada *prompt* di awal simulasi. Peran *reflex* enforce\_lockdown dalam pembatasan aktivitas adalah mempengaruhi atribut agenda mingguan dari semua agen manusia. Agenda mingguan berisi nilai hari dan jam yang dipetakan dengan bangunan; karena diterapkan kerja-dari-rumah ketika PSBB, maka *reflex* ini mengubah nilai bangunan dalam agenda mingguan menjadi rumah. Sehingga agen manusia tidak berpindah lokasi ke tempat kerja atau sekolah.

### Pengujian

|  |  |
| --- | --- |
| Spesifikasi Sistem Level 1 – 6 | |
| Hal | Pemberlakuan PSBB pada simulasi bekerja tanpa kesalahan. |
| Hasil | Dalam simulasi, terdapat penurunan grafik ketika PSBB sudah dilaksanakan. Perubahan peraturan juga memicu perubahan pesan yang ditampilkan pada simulasi, memberikan keterangan bahwa simulasi berada pada skenario PSBB. Selain penurunan grafik, aktivitas manusia pada peta simulasi juga terlihat berkurang setelah adanya PSBB. |