IMPLEMENTASI AGENT-BASED MODELING AND SIMULATION DALAM EVAKUASI KEADAAN DARURAT

IMPLEMENTATION OF AGENT-BASED MODELING AND SIMULATION ON EMERGENCY EVACUATION

Varian Vianandha[1], Drs. Jondri, M.Si.[2], Anditya Arifianto, ST., MT.[3]

[1,2,3] Teknik Informatika, Universitas Telkom
Jl. Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia vian_varian@yahoo.co.id,

jondri@telkomuniversity.ac.id, anditya.arifianto@gmail.com

ABSTRAKSI

Teknik penyelamatan diri setiap orang dalam keadaan darurat seperti kebakaran di dalam sebuah gedung seringkali justru menjadi berbahaya ketika hal tersebut merugikan orang lain, seperti saling berdesakan yang dapat membahayakan nyawa diri sendiri maupun orang lain. Simulasi evakuasi sekarang menjadi perhatian khusus pada ilmuwan untuk mengurangi resiko kecelakaan yang dapat merugikan korban. Simulasi menggunakan metode *agent-based* dilakukan untuk mengetahui efek dari setiap atribut yang dimiliki oleh setiap orang. Hasilnya merupakan representasi dari keadaan nyata yang disimulasikan sesuai dengan perilaku manusia. Pada akhirnya, simulasi ini menghasilkan suatu kondisi di mana mirip dengan kondisi aslinya.

Kata kunci : evakuasi, *agent-based modeling and simulation*, darurat, bencana

ABSTRACT

Self rescue techniques for each person in an emergency such as a fire in a building is often a dangerous thing when it harms the others, such as jostle each other or trampled others who may endanger the lives of themselves or others. Evacuation simulation is now a special attention to the scientists to reduce the risk of accidents that can harm the victim. Simulations were performed to determine the effect of each of the attributes possessed by each person. The results are the representation from the real condition that has been simulated according to human behavior. In the end, this simulation produce a condition in which similar to original condition.

Keywords : *evacuation, agent-based modeling and simulation, emergency, disaster*

1. Pendahuluan

1.1. Latar Belakang

Keadaan darurat akibat adanya bencana alam ataupun kesalahan manusia pada suatu tempat seringkali menimbulkan banyak korban jiwa, seperti bencana kebakaran yang melanda gedung bertingkat [10]. Nyawa korban yang berada di dalam gedung bertingkat tersebut dipertaruhkan karena beberapa hal, seperti korban yang tidak bisa menyelamatkan diri karena terluka, jalan pintu darurat yang tertutup, bahkan meninggal di tempat. Kasus yang menimpa banyak korban pun sudah beberapa kali terjadi, seperti penyelamatan diri dari kobaran api di stadium di Buenos Aires, Argentina pada tahun 1968 yang mengakibatkan 75 korban meninggal dan 150 korban luka, pencinta sepak bola yang memaksa masuk ke dalam stadium di Meksiko pada tahun 1985 yang menimbulkan korban meninggal sebanyak 10 orang dan korban luka sebanyak 29 orang, dan di Mekah, Arab Saudi pada tahun 1990 yang diakibatkan oleh terlalu banyaknya orang yang masuk ke dalam terowongan serta didukung oleh sirkulasi udara yang buruk sehingga mengakibatkan 1425 orang meninggal dunia [9].

Beberapa metode evakuasi sudah pernah dipraktikan oleh para peneliti, mulai dari mensimulasikan akses keluar menuju tempat yang lebih aman apabila terdapat keadaan darurat, menyediakan fasilitas yang dapat menyelamatkan orang seperti alat pemadam kebakaran, sampai dengan pemasangan alarm ketika adanya keadaan darurat, namun cara-cara tersebut dirasa kurang efektif untuk mengurangi jumlah dari korban jiwa yang ada serta resiko yang ditimbulkan apabila mensimulasikan menggunakan manusia dikhawatirkan berbahaya. Beberapa metode sekarang sudah dikembangkan karena kebutuhan akan akses evakuasi yang ideal pada suatu tempat yang dipenuhi banyak orang dan sebagai cara untuk mengurangi angka korban jiwa pada suatu bencana.

Solusi yang ditawarkan pada tulisan ini adalah penerapan *agent-based modeling and simulation* (ABMS) dengan menggunakan perangkat lunak *Repast Symphony* karena pemodelan ini sudah digunakan oleh banyak sistem yang ada, ABMS dapat memodelkan sampai ke tingkat perilaku yang dimiliki setiap individu serta proses adaptasi yang memungkinkan adanya perubahan sifat yang dimiliki oleh individu juga membantu dalam memodelkan sistem sehingga pemodelan akan sangat mirip dengan kenyataannya. ABMS juga fleksibel, dapat menggambarkan fenomena evakuasi secara detail, serta memberikan informasi dari sistem yang ada seperti korban jiwa yang mungkin ada selama evakuasi berjalan. Diharapkan dengan menerapkan metode tersebut proses pembuatan simulasi dapat menjadi lebih nyata sehingga

dapat diterapkan pada penempatan posisi pintu keluar pada gedung bertingkat.

1.2. Perumusan Masalah

Pada tulisan ini, rumusan masalah yang akan diteliti adalah bagaimana penerapan *agent-based modeling and simulation* pada simulasi gedung yang terbakar serta waktu yang diperlukan *agent* untuk membebaskan diri dari situasi darurat tersebut.

1.3. Tujuan

Tujuan yang ingin dicapai penulis adalah sebagai berikut :

1. Mengetahui apakah metode ABMS cocok untuk diimplementasikan dalam simulasi gedung yang terbakar.
2. Mencari waktu yang dibutuhkan oleh semua agen untuk membebaskan diri dari dalam ruangan yang dikondisikan sedemikian rupa serta jumlah korban yang ada dengan adanya pengimplementasian berbagai macam kasus.

1.4. Batasan Masalah

Adapun batasan masalah yang terdapat pada tulisan ini adalah sebagai berikut :

1. Kasus yang disimulasikan hanya di dalam gedung tidak bertingkat.
2. Bencana yang ada hanya kebakaran.
3. Proses perpindahan agen tidak memperdulikan api dikarenakan adanya tingkat kepanikan sehingga agen hanya fokus pada jalur keluar.
4. Kecepatan akan bertambah apabila tingkat kepanikan dan kegugupan semakin tinggi, serta berkurang apabila tingkat kelelahan, cedera, serta usia semakin besar.

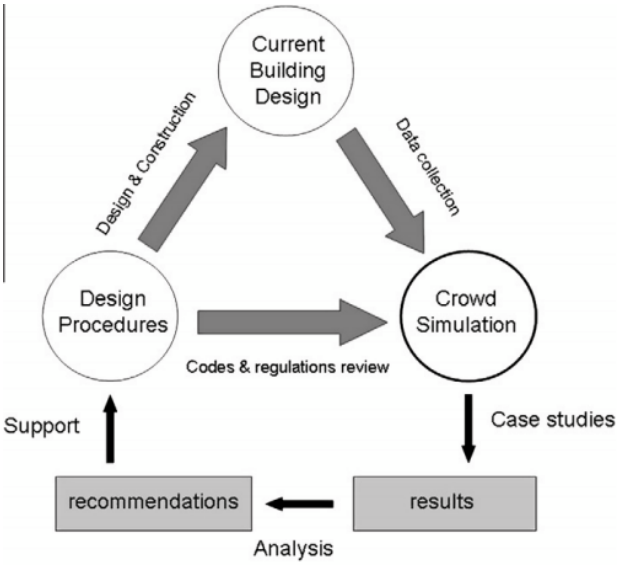
2. Landasan Teori

2.1. *Crowd Simulation*

Crowd simulation adalah metode yang digunakan untuk mensimulasikan keramaian dengan proses asumsi sifat yang dimiliki oleh tiap *agent* sehingga didapatkan lingkungan yang terlihat seperti aslinya [8].

Crowd simulation merupakan proses simulasi pergerakan banyak entitas ataupun karakter, teknik ini sekarang sudah menggunakan grafik dari komputer dalam bentuk tiga dimensi. Ketika mensimulasikan keramaian, objek manusia yang diteliti beserta interaksinya direplikasi untuk mengumpulkan sifat-sifat yang tampak seperti nyata [7].

Keramaian yang maya biasanya dimodelkan sebagai kumpulan agen yang berinteraksi. Pada *behavioral models*, pergerakan sekelompok agen akan dipengaruhi oleh masing-masing agen dan pergerakan agen tersebut akan saling mempengaruhi. Tingkah laku agen ini didefinisikan menggunakan peraturan yang sederhana, seperti "bergeraklah dengan kecepatan sedang" atau "jaga agar Anda tidak bertabrakan dengan orang lain" [7].



Gambar 2-1 Desain dari Simulasi Keramaian secara Umum

Simulasi keramaian dapat dibagi menjadi dua kondisi, pertama adalah situasi normal di mana agen bergerak secara perlahan tanpa adanya rasa takut akan bahaya, kedua adalah situasi panik di mana agen dapat bertingkah di luar kebiasaannya. Pada situasi yang panik biasanya berlaku hal sebagai berikut [8]:

- Dalam situasi panik, individu akan menjadi stress karena mereka merasa takut dan mementingkan untuk menyelamatkan diri sendiri, missal mereka mulai melakukan aksi "buta", bergerak tanpa tujuan.
- Orang akan bergerak lebih cepat dari biasanya.
- Individu mulai mendorong satu sama lain, pada intinya mereka melakukan kontak fisik secara alami lebih banyak dari biasanya.
- Pada saat bergerak biasanya akan terjadi *bottleneck*.
- Pada pintu keluar, akan terjadi kemacetan oleh banyaknya orang.
- Interaksi pada keramaian yang macet dapat membahayakan orang yang terkontak fisik.
- Proses evakuasi menjadi sangat lambat karena terdapat korban yang meninggal dan menjadi "penghalang" korban yang masih hidup.
- Jalan keluar alternatif biasanya ada dan tidak digunakan secara efektif oleh korban.

Simulasi keramaian sekarang menjadi tantangan tersendiri bagi peneliti. Banyak model keramaian sudah dibuat seperti *cellular automata*, *agent-based*, *regression*, dan lain-lain. Model keramaian yang umum dibagi menjadi dua, yaitu *macroscopic models* dan *microscopic models* [7][9][10]. *Macroscopic* mencakup pemodelan *regression* dan *fluid dynamic models*. Sedangkan *microscopic* mencakup pemodelan *rule-based models* dan *agent-based models*.

2.2. *Agent-Based Modeling and Simulation*

Agent-Based Modeling and Simulation (ABMS) merupakan metode pemodelan yang bertujuan untuk memodelkan masalah yang kompleks berdasarkan kasus nyata yang ada. Dalam kasus ini model yang ada dapat digunakan untuk merencanakan serta menyiapkan keadaan darurat jika suatu saati terjadi hal yang tidak diinginkan [7].

- Kemampuan untuk membuat lingkungan seperti keadaan aslinya, seperti kursi, tembok, dan batasan-batasan lainnya layaknya seperti dunia nyata.
- Kemampuan untuk mensimulasikan posisi api ketika gedung terbakar sehingga dapat membuat pencegahannya.
- Kemampuan untuk mensimulasikan beberapa kenario keselamatan, seperti jalur keselamatan, dengan tanpa biaya.

Tujuan dari penggunaan metode ini adalah sebagai berikut [7]:

- Berguna : Mengapa ABMS merupakan metode pemodelan yang cocok untuk memodelkan sistem yang besar dan memiliki cara-cara memodelkan yang lebih canggih untuk kasus lainnya.
- Layak digunakan : Bagaimana ABMS dapat diimplementasikan kepada perangkat lunak dengan tingkat pengembangan yang lebih baik.
- Dapat digunakan : Bagaimana ABMS digunakan untuk menyelesaikan masalah dalam keadaan seperti dunia nyata.

Sistem ABS didesain untuk memodelkan suatu tempat yang memiliki kursi, jalan, pintu keluar, manusia, dan lain-lain. Hal ini memungkinkan untuk memodelkan banyak manusia dan api pada satu tempat. Tujuan dari system ini adalah untuk mensimulasikan evakuasi keramaian pada bangunan besar seperti stadium, auditorium, atau gedung konser. Sistem harus mampu memperlihatkan bagaimana agen berinteraksi satu sama lain ketika dalam keadaan darurat, bagaimana keadaan api serta tata ruang yang juga dapat mengganggu jalannya proses evakuasi, serta agen yang harus menghindari api sebisa mungkin. Api juga direpresentasikan sebagai agen yang dibuat dan menghasilkan asap. Sistem ini digunakan untuk merencanakan dan mempersiapkan kasus darurat dan menawarkan keuntungan dari proses evakuasi di lapangan sehingga korban yang berjatuhan tidak akan banyak [8].

- Kemampuan untuk menggambarkan lingkungan dengan susunan kursi, jalan, pintu keluar, dan posisi agen. Hal ini memungkinkan pihak manajemen gedung untuk mereplikasikan kondisi gedung ketika terdapat bencana.
- Kemampuan untuk membuat beberapa api dengan proses penyebaran api dan tingkat produksi asap yang dapat ditentukan.
- Kemampuan untuk mensimulasikan beberapa cenario dan perhitungan keamanan, seperti jalan yang lebih lebar atau pintu keluar yang lebih banyak secara maya dan tanpa biaya.

ABMS adalah teknik pemodelan simulasi yang sangat berguna, dan sudah terdapat beberapa aplikasi yang dibuat dalam beberapa tahun terakhir, termasuk aplikasi yang menggambarkan bisnis yang nyata. Di ABMS, sistem dimodelkan sebagai kumpulan objek yang dapat membuat kesimpulan masing-masing yang disebut agen [9]. Setiap agen merepresentasikan seseorang beserta lingkungannya dan interaksi dengan agen lain untuk membuat kesimpulan dari aturan yang sudah dibuat sebelumnya. Terdapat pula agen yang berpengalaman, seperti mereka yang sudah mengetahui jalur evakuasi yang secepat mungkin dapat mengeluarkan mereka dari suasana berbahaya.

Keuntungan dari menggunakan ABMS untuk memodelkan adalah fleksibel, menggambarkan kondisi nyata, dan memberikan cara agar proses evakuasi dapat dilaksanakan secepat mungkin tanpa ada korban yang berjatuhan. Jadi, ABMS idealnya menawarkan solusi secara mekanisme sebelum dan setelah bencana itu terjadi sehingga dapat dijadikan tolak ukur bagi pihak manajemen gedung untuk dapat membuat kondisi nyata seefektif mungkin untuk proses evakuasi.

2.3. Pengukuran Kelayakan *Agent-Based Simulation*

Pada jurnal [3] membahas mengenai pengukuran kompleksitas yang sudah dibuktikan melalui kasus tertentu. Pengukuran ini berguna untuk menguji apakah metode *agent-based* layak untuk diimplementasikan atau tidak.

Berikut adalah perhitungan kelayakan beserta penjabarannya.

- NAT (*Number of Agent Types*) adalah pengukuran dari tingkat perbedaan agen di suatu model. Pada dasarnya pengukuran ini menghasilkan variasi model dari semua kelas seperti pada *object-oriented* dan dapat dengan mudah dihitung pada model simulasi. Tingkat tertinggi dari pengukuran ini adalah ketika semua agen memiliki atribut yang berbeda satu sama lain serta kelas yang berbeda. Contoh, jika hanya parameter dan bobot yang didefinisikan berbeda namun masih menggunakan kelas yang sama, maka NAT = 1, walaupun isi dari atribut kelas tersebut benar-benar berbeda.
- NRT (*Number of Resource Types*) sama seperti pengukuran NAT, namun untuk entitas pasif dari lingkungan yang ada.
- MNA (*Maximum Number of Agents*) adalah pengukuran yang paling mudah karena hanya melihat jumlah agen selama proses simulasi berjalan.
- MNR (*Minimum Number of Resources*) adalah pengukuran yang mendukung MNA, *resource*

yang dibutuhkan untuk sekian MNA didefinisikan di sini.

- MDA (*Maximum Delta of Agent Population*) adalah pengukuran untuk mendefinisikan perbedaan jumlah populasi dari rentang waktu yang diperlukan, pada umumnya satu langkah simulasi. MDA juga mengukur pergantian populasi di setiap langkah simulasi.
- MDR (*Maximum Delta of Resource Population*) adalah pendukung dari perhitungan MDA.
- ARR (*Agent-Resource Relation*) adalah jumlah agen dibagi dengan banyaknya *resource* yang dibutuhkan (MNA dibagi MNR).
- NASh (*Number of Agent Shapes*) adalah pengukuran untuk kompleksitas yang spasial. Contoh pada perhitungan jumlah geometri pada satu agen yang memungkinkan. Pengukuran ini hanya dapat dilakukan melalui simulasi yang berbasis spasial map.
- NRSh (*Number of Resource Shapes*) adalah pendukung dari perhitungan NASh.
- MRS (*Maximum Resource Status Size*) adalah perhitungan yang melibatkan maksimum kemungkinan perubahan atribut apabila terdapat penghalang.
- MRP (*Maximum Resource Parameter*) adalah perhitungan parameter yang mempengaruhi suatu nilai dari variabel lainnya.

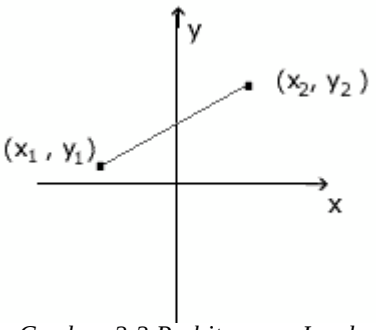
Pada kasus ABMS, simulasi *sugarscape* sangat identik dengan penelitian di tulisan ini. Perlu diketahui bahwa simulasi *sugarscape* merupakan simulasi yang agennya berupa semut, dia harus mencari tumpukan gula untuk dapat selamat dan gulanya dapat bertambah sesuai dengan peluang pertambahannya. Namun semut tersebut dapat mati apabila gula sudah habis dimakan oleh semut lain atau perkembangan gulanya sedikit. Berikut adalah contoh perhitungan dari berbagai macam variabel di atas menggunakan kasus *sugarscape*.

Tabel 2-1 Perhitungan Kasus Sugarscape

Perhitungan	Sugarscape
NAT	1
NRT	1
MNA	Sekitar 15
MNR	2500
MDA	-13
MDR	0
ARR	0.12
NASh	1
NRSh	1
MRS	1
MRP	2

2.4. Perhitungan Jarak dan Sudut

Pada tahapan ini jarak dan sudut pergerakan agen perlu dihitung untuk mendapatkan hasil yang diinginkan. Rumus untuk menghitung jarak 2 titik pada bidang 2 dimensi adalah sebagai berikut [12].

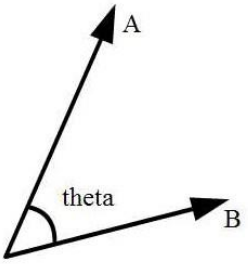


Gambar 2-2 Perhitungan Jarak

Dengan jarak yang didapatkan melalui perhitungan berikut. Nilai x dan y didapatkan dari koordinat satu agen dengan agen lain atau pintu keluar.

$$jarak = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Sedangkan untuk perhitungan sudut antara 2 vektor adalah sebagai berikut [13].



Gambar 2-3 Perhitungan Sudut

Dengan sudut yang didapatkan melalui perhitungan berikut. Nilai vektor A dan B didapatkan dari letak agen dengan tujuan pergerakan agen tersebut.

$$sudut = cos^{-1}\left(\frac{A \cdot B}{|A||B|}\right)$$

Perhitungan jarak dan sudut dilakukan untuk menentukan pergerakan arah agen terutama *human* yang menentukan arah pergerakannya berdasarkan jarak terdekat dengan *leader* atau pintu keluar.

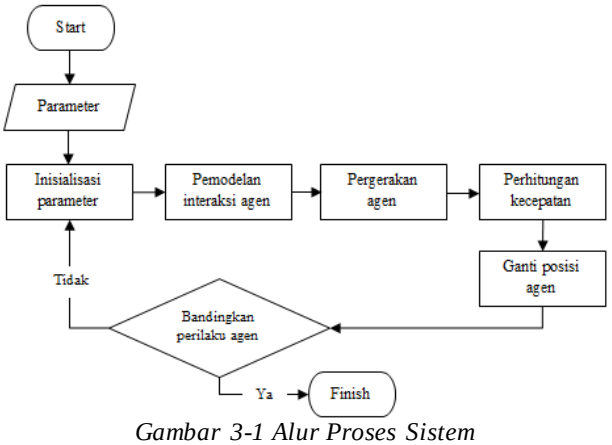
3. Analisis dan Perancangan Sistem

3.1. Deskripsi Sistem

Sistem yang akan dibangun dalam Tugas Akhir ini merupakan sistem yang mengimplementasikan ABMS untuk menentukan kecepatan serta arah gerak agens elama proses perpindahan posisi sehingga mendapatkan waktu seefektif mungkin untuk dapat mencapai pintu keluar dengan kemampuan yang sudah didefinisikan.

3.2. Perancangan Sistem

Pada penelitian tulisan ini, proses yang terdapat pada sistem digambarkan dengan diagram *flow chart* sebagai berikut.



Gambar 3-1 Alur Proses Sistem

Inputan awal yang harus ada adalah jumlah agen beserta tata ruang di dalam gedung (api dan dinding). Jumlah agen yang diinputkan diantara 1 sampai 10.000. Pemrosesan waktu lamanya proses evakuasi dihitung berdasarkan satuan menit dan nilainya didapatkan setelah proses simulasi selesai. Sehingga hasil akhirnya adalah waktu yang diperlukan sekian agen untuk meloloskan diri dari gedung, serta jumlah korban meninggal apabila memungkinkan.

Proses yang terjadi di dalam sistem adalah sistem menerima inputan berupa tata ruang yang ada di dalam gedung beserta jumlah agen, setelah itu memodelkan dengan tampilan yang menggambarkan orang yang terdapat di dalam ruangan. Setelah itu terdapat pemrosesan menggunakan *agent-based modeling and simulation* ketika orang tersebut bergerak untuk menentukan kecepatan dan arah gerak. Pada akhirnya waktu total didapatkan setelah orang terakhir yang masih hidup (asumsi terdapat orang yang meninggal) keluar dari gedung. Waktu dari pemrosesan tersebut dicatat dan pemrosesan sistem diulang dari awal dengan inputan waktu yang berbeda sesuai kesepakatan awal.

Pada pemodelan sistem berbasis *agent-based* ini, beberapa tipe dari individu diketahui, seperti banyaknya laki-laki, perempuan, anak-anak, pihak keamanan, dan pemimpin.

Lingkungan yang akan dibuat pun ditentukan di awal yang terdiri dari :

- Jumlah orang yang ada di area tersebut,
- Jumlah dan posisi pintu keluar,
- Letak dinding,
- Intensitas api, dan
- Tingkat bahaya pengaruh api terhadap orang.

Orang yang dimaksud terdiri dari *human* (orang dengan pengetahuan area rendah) serta *leader* (orang dengan pengetahuan area tinggi).

Adapun atribut yang sekiranya perlu ada di awal pembangungan sistem adalah sebagai berikut :

Tabel 3-1 Atribut yang Digunakan pada Sistem

Atribut	Tipe data	Penjelasan
usia	int	[18,55]
		mengurangi kecepatan sampai dengan 30% pada usia 55 tahun
pengetahuan area	double	[0,1]
individualistis	int	1 : mengikuti pemimpin
		0 : tidak mengikuti pemimpin
posisi	(x,y)	koordinat
kecepatan	double	semakin besar angka, pergerakan semakin cepat
tingkat kelelahan	double	[0,1]
		semakin besar angka, pergerakan semakin lambat
tingkat kepanikan	double	[0,1]
		semakin besar angka, pergerakan semakin cepat
tingkat cedera	double	[0,1]
		semakin besar angka, pergerakan semakin lambat
kecepatan	double	tergantung umur

Karakteristik lingkungan di atas mempengaruhi waktu yang berdampak pada hasil dari evakuasi. Seperti pemimpin yang dapat menggunakan jalur berbeda untuk menemukan jalan keluar yang tersedia karena mereka memiliki tingkat pengetahuan yang lebih tinggi. Proses evakuasi orang yang tidak tahu jalan keluar seharusnya mengikuti pemimpin yang sudah lebih tahu jalan keluar. Namun adanya tingkat cedera yang tinggi dapat memperlambat proses evakuasi karena pergerakan agen menjadi lebih lambat, bahkan tidak mustahil orang yang memiliki tingkat cedera tinggi akan lebih cepat meninggal karena terpapar oleh api. Umur pun dapat menentukan kecepatan yang seharusnya dimiliki oleh individu tersebut, biasanya hal itu menjadi masalah juga untuk proses evakuasi karena dikhawatirkan korban akan bertambah banyak.

3.2.1. Perhitungan Jarak 2 Koordinat

Proses perpindahan agen ini menggunakan jarak dari 2 koordinat, yaitu koordinat agen dan pintu keluar menggunakan rumus pada Gambar 2-3.

Setelah mendapatkan jarak maka penentuan kemana agen *human* harus berpindah diputuskan melalui perhitungan jarak terdekat di antara agen *leader* atau pintu keluar.

3.2.2. Perhitungan Sudut antara 2 Vektor

Proses ini bertujuan untuk mendapatkan sudut pergerakan agen terhadap sesuatu yang ditujunya dengan menggunakan rumus pada Gambar 2-4.

Sudut yang dapat diproses oleh sistem hanya dalam rentang 0 sampai dengan 2π dengan rincian sebagai berikut.

Tabel 3-2 Nilai Sudut Berdasarkan Arah Gerak

Arah Gerak	Nilai Sudut
Utara	$\pi/2$ (1.5708)
Barat	π (3.1416)
Selatan	$3\pi/2$ (4.7124)
Timur	0

Sehingga apabila ditemukan nilai sudut di luar rentang yang sudah ditentukan maka akan dikonversi sehingga sesuai rentang agar mudah dalam pemrosesan apabila terdapat proses yang memerlukan nilai sudut.

3.2.3. Perhitungan Kecepatan Agen

Pada proses ini kecepatan agen dipertimbangkan dengan menggunakan atribut yang sudah didefinisikan sebelumnya. Perubahan kecepatan ini merupakan hal yang penting karena akan mengubah hasil dari simulasi yang dilakukan ke depannya.

Proses inisialisasi kecepatan awal agen bergerak adalah $0{,}75\ ^{m}/_{s}$ (kecepatan rata-rata orang bergerak saat keadaan normal). Namun karena adanya pengaruh lain seperti yang telah didefinisikan di atribut maka kecepatan akan berubah sesuai dengan input tingkatan masing-masing atribut. Berikut terlampir contoh perubahan kecepatan sesuai dengan atributnya (diasumsikan perubahan kecepatan setiap tingkat melebihi 50%).

Tabel 3-3 Contoh Hasil Perhitungan Kecepatan

Kecepatan Awal	Tingkat Kepanikan	Tingkat Kegugupan	Tingkat Kelelahan	Tingkat Cedera	Usia (18-55)	Kecepatan Baru
	30%	20%	-20%	-20%	-30%	
0.75	0	0	1	1	55	0.336
0.75	1	1	0	0	18	1.17
0.75	0.364	0.661	0.7	0.596	29	0.64324525
0.75	0.918	0.126	0.871	0.553	31	0.69952787
0.75	0.213	0.494	0.264	0.466	44	0.59189189
0.75	0.65	0.81	0.105	0.918	21	0.74662111
0.75	0.81	0.502	0.611	0.166	47	0.65065999
0.75	0.1	0.713	0.184	0.648	54	0.5422105

Tabel tersebut memperlihatkan bahwa perubahan akan terjadi seiring dengan tingkatan yang nilainya pada sistem akan diacak pada setiap agen (pada *human* dan *leader*). Seperti tingkat kepanikan akan menambah kecepatan sebesar 30%. Dapat disimpulkan bahwa kecepatan minimum adalah $0{,}336\ ^{m}/_{s}$ serta kecepatan maksimum adalah $1{,}17\ ^{m}/_{s}$.

3.2.4. Proses Pergerakan Agen

Tahapan ini merupakan langkah terakhir dalam penentuan hasil dari gerakan semua agen. Pada tahap ini semua data jarak, sudut, serta kecepatan diambil untuk menentukan gerakan mana yang harus diambil oleh agen. Khusus untuk agen *human*, masih ada satu tahapan lagi yang terlewat yaitu penentuan kemana agen tersebut akan bergerak. Tahapan tersebut akan dijelaskan dengan menggunakan ilustrasi berikut.

Tabel 3-4 Contoh Penentuan Arah Gerak

Jarak *human* dengan *exit* terdekat (meter)	Jarak *human* dengan *leader* terdekat (meter)	Arah Gerak
15.5	12	Leader
8.73	17.22	Exit
4.22	4.01	Leader
20.63	2.1	Leader
1.28	11.2	Exit

3.2.5. Pertemuan Agen dengan Rintangan

Pada tahap ini agen akan bertemu dengan rintangan, yaitu api atau dinding. Tahapan ini bersifat opsional dan hanya akan dieksekusi apabila agen bertemu rintangan tersebut.

Khusus untuk api diasumsikan tidak ada pergerakan untuk menghindari api, namun ditambahkan efek tambahan apabila mendekati api yaitu *health* agen berkurang 1 apabila memasuki area api. Area api ditentukan pada sistem, yaitu sejauh 1 ke segala arah.

Apabila agen bertemu dengan dinding maka agen akan menghindar dengan mengubah sudut pergerakan dengan mengkonversi sudut tersebut ke arah sudut terdekat (utara, barat, selatan, atau timur) sehingga agen akan memproses sudut baru yang didapatkan dari perhitungan yang dijabarkan melalui tabel berikut.

Tabel 3-5 Arah Gerak yang Baru Berdasarkan Arah Sebelumnya

Arah Gerak	Pilihan Pergerakan
Utara	Barat atau Timur
Barat	Utara atau Selatan
Selatan	Barat atau Timur
Timur	Utara atau Selatan

Untuk penentuan salah satu arah gerak yang baru masih memakai variabel acak sehingga gerakannya pun masih acak hingga menemukan jalan untuk membebaskan diri dari dinding yang menghalangi langkah agen.

3.3. Implementasi Sistem

Sistem implementasi *agent-based modeling and simulation* ini dilakukan menggunakan aplikasi berbasis *desktop*. Sistem ini dibangun dengan *tools* pemodelan simulasi berbasis *Java*.

Adapun spesifikasi perangkat keras dan lunak yang digunakan pada implementasi sistem ini adalah sebagai berikut.

1. Perangkat keras
   - Prosesor : Intel® Core™ i3-2330M
   - Kecepatan prosesor : @ 2.20 GHz (4 CPUs)
   - Memori : 2048 MB
2. Perangkat lunak
   - Sistem operasi : Windows 8 Pro 64-bit (6.2, Build 9200)
   - IDE : Repast Simphony 2.1 (Eclipse Kepler)
   - Java version : 1.8.0_05

4. Pengujian dan Analisis Sistem

4.1. Pengujian Sistem

Pada bagian ini dijelaskan proses pengujian dari sistem yang telah dibuat, kemudian dilanjutkan dengan analisis dari hasil pengujian.

4.1.1. Tujuan Pengujian

Berikut merupakan tujuan dilakukan pengujian,

1. Mencari waktu yang dibutuhkan agen untuk menuju pintu keluar dan selamat dengan adanya api di dalam ruangan,
2. Menganalisis perubahan waktu keluar serta jumlah kematian disebabkan oleh api karena tidak adanya agen *leader* di dalam ruangan,

3. Menganalisis perubahan waktu keluar dengan berbagai macam jumlah pintu, dan
4. Menganalisis perubahan waktu keluar berdasarkan ada atau tidaknya dinding di ruangan tersebut.

4.1.2. Strategi Pengujian

Berdasarkan alur sistem yang ada pada bab 3, pengujian akan dilakukan beberapa kali dengan menggunakan tata letak yang berbeda untuk mendapatkan waktu yang dibutuhkan oleh agen untuk melarikan diri dari ruangan.

Langkah pengujian dimulai dengan menentukan parameter panjang dan lebar ruangan, jumlah agen (*human* dan *leader*), dan bentuk tata letak ruangan (dinding dan jumlah api). Setiap pengujian akan dilakukan 10 kali untuk mencari rata-rata waktu dan jumlah agen yang berhasil keluar atau mati.

Pada pengujian kali ini parameter *health* bernilai tetap, yaitu minimum 5 dan maksimum 10, serta *layout* ruangan pun tetap yaitu panjang dan lebar 30.

Tabel 4-1 Data Pengujian

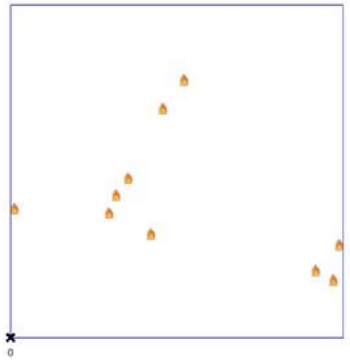
No	*Layout* Ruangan		Jumlah Agen		*Health*		Jumlah Api	Jumlah *Exit*	Dinding
	Panjang	Lebar	*Leader*	*Human*	Min	Maks			
1a	30	30	5	100	5	10	10	1	Tidak ada
1b	30	30	10	200	5	10	10	1	Tidak ada
1c	30	30	15	300	5	10	10	1	Tidak ada
1d	30	30	20	400	5	10	10	1	Tidak ada
2a	30	30	0	100	5	10	10	1	Tidak ada
2b	30	30	0	400	5	10	10	1	Tidak ada
3a	30	30	5	100	5	10	10	2	Tidak ada
3b	30	30	5	100	5	10	10	4	Tidak ada
4	30	30	5	100	5	10	10	1	Ada

4.2. Analisis Hasil Pengujian

Dari pengujian yang sudah dilakukan, selanjutnya hasil yang didapat dianalisis sesuai dengan skenario pengujian yang ada.

4.2.1. Waktu yang Dibutuhkan Agen untuk Menuju Pintu Keluar

Untuk kasus pengujian nomor 1 ruangan berbentuk persegi dengan sisi 30 meter, perbandingan jumlah *human* dengan *leader* adalah 20:1 serta diambil kelipatan 100 untuk jumlah *human*, jumlah api adalah 10 buah dan diletakan secara acak di dalam ruangan, serta pintu keluar diletakan di koordinat (0,0) seperti pada Gambar 4-1 tanpa diberi halangan (dinding).

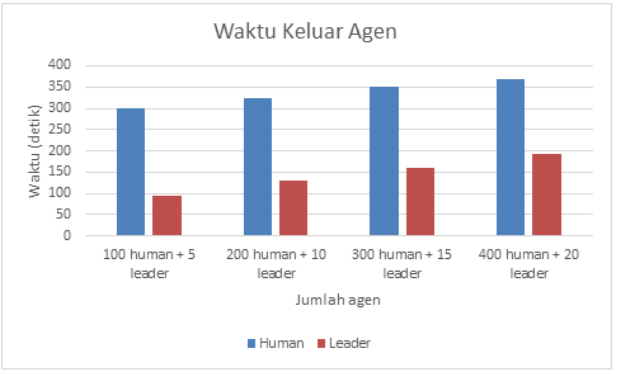


Gambar 4-1 Letak Pintu Keluar (1)

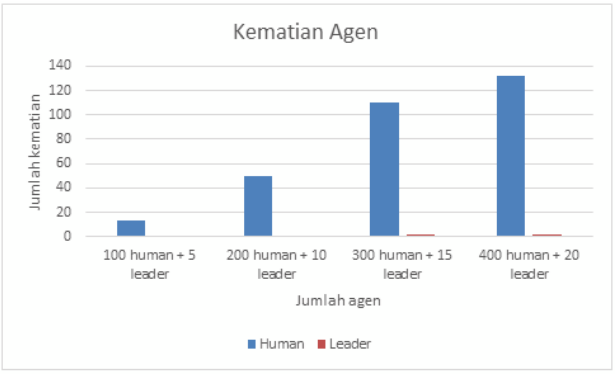
Untuk pengujian pertama didapatkan rata-rata setiap pasang jumlah *human* dan *leader* dengan hasil seperti berikut.

Tabel 4-2 Hasil Pengujian Nomor 1

Jumlah Agen	Waktu keluar agen (detik)			Jumlah agen yang mati		
	Semua *human*	90% *human*	*Leader*	*Human*	*Leader*	Total
100	301	127	93	14	0	14
200	324	136	131	49	1	50
300	349	143	158	110	1	111
400	369	160	192	132	1	133



Gambar 4-2 Grafik Waktu Keluar Agen



Gambar 4-3 Grafik Jumlah Kematian Agen

Dari grafik yang tertera di atas menunjukkan bahwa semakin banyak jumlah agen di dalam ruangan maka jumlah kematian yang diakibatkan pun akan semakin tinggi, terlihat pada 400 *human* serta 20 *leader* angka rata-rata waktu keluar serta kematian paling tinggi.

Pada Tabel 4-2 terdapat kolom "90% *human*" yang maksudnya adalah 90% agen *human* yang selamat melarikan diri pada waktu tersebut. Dapat terlihat bahwa mayoritas *human* sudah terlebih dahulu melarikan diri dengan cepat, namun 10% sisanya biasanya memiliki kecepatan yang rendah sehingga pergerakannya relatif lambat.

Agen *leader* pun keluar lebih cepat dibandingkan *human* karena mereka memiliki tingkat pengetahuan area yang lebih tinggi dibandingkan *human*, serta tingkat kematian *human* yang cukup tinggi pun dikarenakan rendahnya pengetahuan mereka terhadap ruangan. Semakin cepat agen membebaskan diri maka semakin kecil kemungkinan agen tersebut menjadi korban di ruangan tersebut.

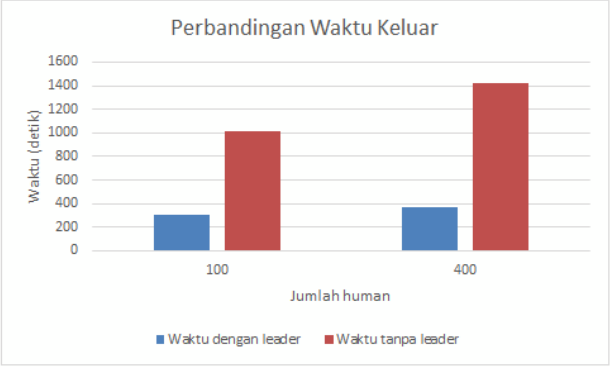
4.2.2. Waktu yang Dibutuhkan Agen Tanpa Adanya *Leader*

Untuk pengujian Tabel 4-1 nomor 2 posisi pintu keluar seperti pada Gambar 4-1 dengan penghapusan agen *leader*. Hasilnya seperti berikut.

Tabel 4-3 Hasil Pengujian Nomor 2

Jumlah Agen	Waktu keluar (detik)	Kematian
100	1015	53
400	1424	195

Apabila data pada Tabel 4-3 dibandingkan dengan data pada Tabel 4-2 maka hasilnya akan menjadi seperti berikut.



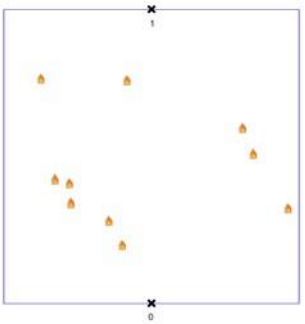
Gambar 4-4 Grafik Perbandingan Waktu Keluar

Gambar 4-5 Grafik Perbandingan Jumlah Kematian

Terlihat dari hasil yang dibandingkan dengan kasus pada nomor 1 yaitu ruangan tanpa adanya agen *leader* membutuhkan waktu yang lebih lama untuk keluar serta menghasilkan jumlah kematian yang lebih tinggi. Hal tersebut dikarenakan pengetahuan dari *human* yang sama sehingga tidak adanya sosok pemimpin di dalam ruangan tersebut sehingga tidak adanya pemandu jalan yang tahu secara persis posisi pintu keluar, yang memungkinkan memperlambat evakuasi serta meningkatkan jumlah korban kematian.

4.3.3. Waktu yang Dibutuhkan Untuk Keluar dengan Banyak Pintu

Untuk kasus pengujian nomor 3 ruangan berbentuk persegi dengan sisi 30 meter, jumlah *human* adalah 100 orang dan *leader* adalah 5 orang, jumlah api adalah 10 buah dan diletakan secara acak di dalam ruangan, serta pintu keluar seperti pada Gambar 4-6 (pintu diberi tanda X) tanpa diberi halangan (dinding). Pada 4 pintu letaknya ditaruh di setiap pojok ruangan.

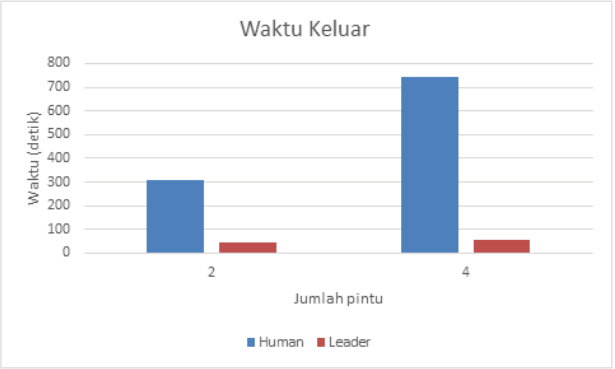


Gambar 4-6 Letak Pintu Keluar (2)

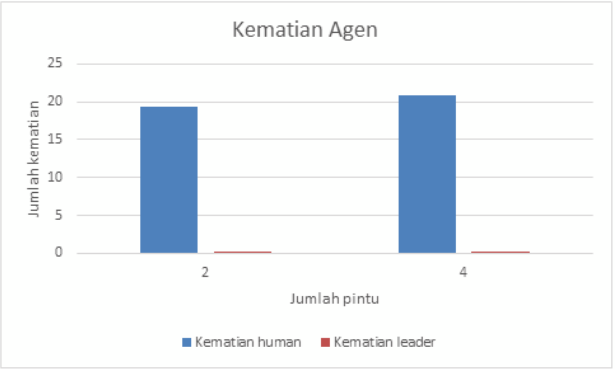
Hasil dari pengujian tersebut adalah sebagai berikut.

Tabel 4-4 Hasil Pengujian Nomor 3

Jumlah pintu	Waktu *human* keluar (detik)	*Leader* keluar (detik)	Kematian *human*	Kematian *leader*
2	308	47	19	0
4	745	54	21	0



Gambar 4-7 Grafik Waktu Keluar



Gambar 4-8 Grafik Kematian Agen

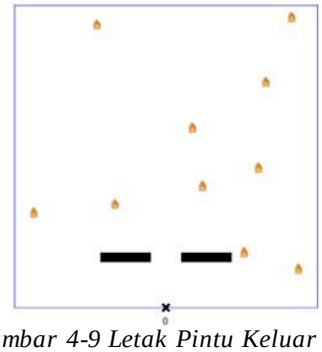
Pada tabel dan grafik terlihat bahwa waktu evakuasi pada ruangan yang memiliki 4 pintu lebih lama dibandingkan dengan 2 pintu. Hal ini disebabkan karena agen *human* yang cenderung berputar-putar di dalam ruangan karena mereka kebingungan untuk memilih pintu yang akan dilaluinya. Agen *leader* pun mengalami hal yang sama, mereka cenderung bingung akan memilih pintu yang mana yang akan digunakannya untuk membebaskan diri, namun mereka masih sangat cepat untuk menyelamatkan dirinya, hal itu dapat terjadi karena tingkat pengetahuan ruangan yang tinggi.

Untuk tingkat kematian pada ruangan yang memiliki 4 pintu sedikit lebih tinggi dikarenakan oleh lamanya waktu evakuasi pada agen *human*. Banyaknya pintu sebenarnya dapat meminimkan korban di dalam ruangan

seperti yang terlihat pada jumlah kematian agen *leader*, asalkan pengetahuan agen terhadap area semakin tinggi sehingga waktu evakuasi pun semakin cepat.

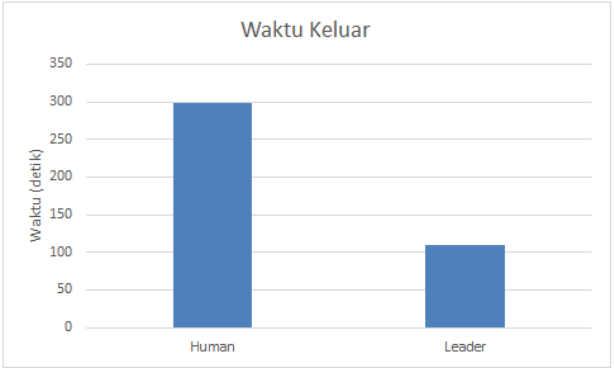
4.3.4. Waktu yang Dibutuhkan Untuk Keluar dengan Adanya Dinding

Untuk kasus pengujian nomor 4 ruangan berbentuk persegi dengan sisi 30 meter, jumlah *human* adalah 100 orang dan leader adalah 5 orang, jumlah api adalah 10 buah dan diletakan secara acak di dalam ruangan, serta pintu keluar (pintu diberi tanda X) dengan ditambahkan dinding pada koordinat (9,5) sampai dengan (21,5) dengan sedikit celah pada koordinat (14,5) sampai dengan (16,5) seperti pada Gambar 4-9 dengan tujuan sedikit menghalangi agen untuk mencapai pintu keluar.

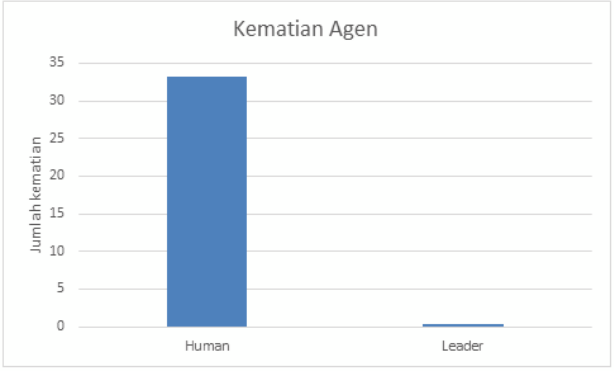


Gambar 4-9 Letak Pintu Keluar (3)

Hasil dari pengujian tersebut adalah sebagai berikut.

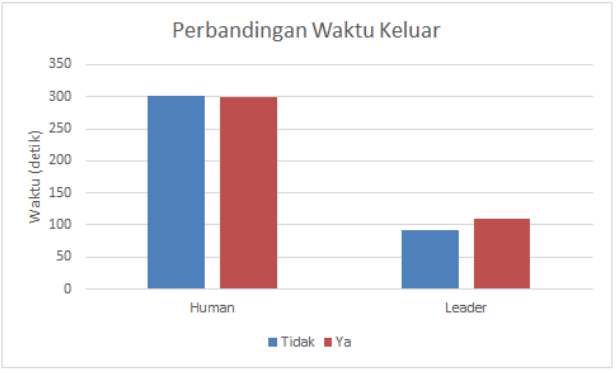


Gambar 4-10 Grafik Waktu Keluar

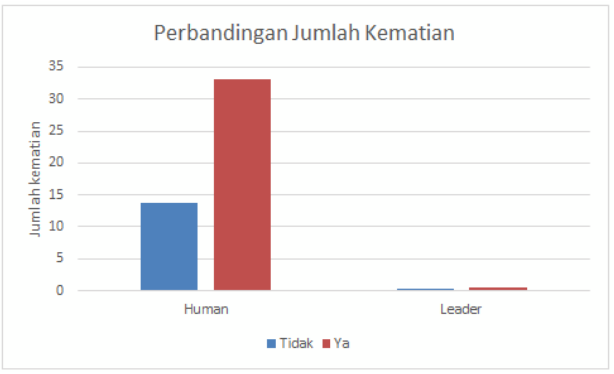


Gambar 4-11 Grafik Kematian Agen

Perbandingan hasil pada adanya dinding dengan tidak adanya dinding adalah sebagai berikut.



Gambar 4-12 Perbandingan Waktu Keluar



Gambar 4-13 Perbandingan Jumlah Kematian

Terlihat jelas untuk perbandingan waktu keluar antara adanya dinding dengan tidak adanya dinding untuk agen *human* perbedaan tidak terlalu signifikan, hanya saja untuk agen *leader* waktu evakuasi menjadi lebih lama dibandingkan dengan tidak adanya dinding karena mereka cenderung untuk menghindar dinding. Namun hasil waktu evakuasi yang cenderung sama pada agen *human* tidak didukung dengan jumlah kematian yang lebih tinggi apabila adanya dinding di dalam ruangan. Hal tersebut diakibatkan oleh posisi api yang memungkinkan menghalangi posisi celah dinding sehingga *human* akan menjadi korban lebih cepat, beda halnya dengan *leader* yang keduanya sama-sama memberikan hasil korban yang nihil.

4.3.5. Perhitungan Nilai Kelayakan Sistem

Pada kasus ini penulis akan menggunakan perhitungan NAT, NRT, MNA, dan MRP karena perhitungan ters ebut memiliki pengaruh yang besar terhadap kelayakan sistem.

Pada Tabel 2-1 dijelaskan bahwa kasus *sugarscape* dapat dinyatakan layak apabila memenuhi angka dari perhitungan yang sudah didefinisikan sebelumnya. Penulis akan mencoba menghitung nilai kelayakan berdasarkan variabel yang ada dan dibandingkan dengan kasus *sugarscape*.

- NAT

  Untuk perhitungan ini penulis menggunakan 2 kelas yang berbeda, yaitu *Human.java* serta *Leader.java*, meskipun terdapat 2 kelas tambahan yaitu *Dead.java* serta *DeadLeader.java* untuk mendefinisikan agen yang telah mati namun kelas tersebut tidak

berpengaruh bagi lingkungan. Sehingga NAT = 2.

- NRT

  Untuk perhitungan ini penulis membangun beberapa kelas tambahan, seperti *Dead.java*, *DeadLeader.java*, *Wall.java*, *Fire.java*, serta *Exit.java*. Maka NRT = 5.
- MNA

  Pada perhitungan ini maksimum agen yang dapat diinputkan adalah 10 ribu dari kombinasi *human* dan *leader* untuk mencegah sistem yang melambat karena terlalu banyak agen yang diinputkan. Sehingga MNA = 10.000.
- MRP

  Untuk perhitungan ini penulis menggunakan beberapa atribut yang mendukung, seperti *speed*, *angle*, *health*, *knowledge*, *injury*, *fatigue*, *panic*, serta *nervousness*. Atribut tersebut sangat mempengaruhi pola perpindahan agen. Maka MRP = 8.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan pada Tugas Akhir ini dapat disimpulkan beberapa hal, yaitu sebagai berikut.

1. Metode ABMS dapat diimplementasikan dengan sempurna untuk simulasi karena memiliki tingkat perhitungan yang tinggi, seperti NAT, NRT, MNA, dan MRP yang melebihi perhitungan pada kasus *sugarscape*, serta performansi sistem yang cepat menjadikan ABMS sebagai simulasi yang dapat menampung banyak variabel dengan komputasi yang ringan.
2. Waktu yang dibutuhkan oleh agen untuk membebaskan diri serta korban yang ditimbulkan dari berbagai macam kasus pun dilihat cukup mewakili dari kasus yang diujikan sebelumnya.

5.2. Saran

Karena cakupan ABMS sangat luas maka sebaiknya tambahkan parameter ABMS yang baru karena semakin banyak parameter yang ditambahkan maka simulasinya akan semakin sesuai dengan dunia nyata, seperti tambahan untuk menghindari api, dinding, dan sebagainya karena parameter ABMS dapat menentukan jalur keluar apabila didefinisikan sedetail mungkin, seperti interaksi antar agen, pencegahan apabila agen bertemu dengan penghalang, kondisi api yang dapat membesar dalam kurun waktu tertentu, dan lain-lain.

Daftar Pustaka

[1] Beltaief, O., Hadouaj, S. E., & Ghedira, K. (2011). Multi-Agent Simulation Model of Pedestrians Crowd Based on Psychological Theories. 150-156.

[2] Helbing, D., Farkas, I. J., Molnar, P., & Vicsek, T. (n.d.). Simulation of Pedestrian Crowds in Normal and Evacuation Situations.

[3] Klugl, F. (n.d.). Measuring Complexity of Multi-Agent Simulations - An Attempt Using Metrics.

[4] Macal, C. M., & North, M. J. (2009). Agent-Based Modeling and Simulation. 86-98.

[5] Qingsong, Z., Guomin, Z., & Jinlan, L. (2009). Performance-Based Design for Large Crowd Venue Control Using a Multi-Agent Model. 352-359.

[6] Ren, C., Yang, C., & Jin, S. (2009). Agent-Based Modeling and Simulation on Emergency Evacuation. 1451-1461.

[7] Shendarkar, A., & Vasudevan, K. (2006). Crowd Simulation for Emergency Response using BDI Agent on Virtual Reality. 545-553.

[8] Thalmann, D., & Musse, S. R. (2007). *Crowd Simulation*. Springer.

[9] Wagner, N., & Agrawal, V. (2014). An Agent-Based Simulation System for Concert Venue Crowd Evacuation Modeling in the Presence of a Fire Disaster. 2807-2815.

[10] Xiaoping, Z., Tingkuan, Z., & Mengting, L. (2009). Modeling Crowd Evacuation of a Building Based On Seven Methodological Approaches. 437-445.

[11] Xiong, M., Lees, M., Cai, W., Zhou, S., & Low, M. Y. (2009). A Rule-Based Motion Planning for Crowd Simulation. 88-95.

[12] *About.com Mathematics*. (2015). Retrieved January 2, 2015, from About.com: http://math.about.com/library/bldistance.htm

[13] *tutsplus*. (2014). Retrieved January 3, 2015, from Envato Pty Ltd.: http://gamedevelopment.tuts plus.com/tutorials/how-to-detect-when-an-object-has -been-circled-by-a-gesture--gamedev-336