



TESIS - TE142599

**SIMULASI EVAKUASI KEBAKARAN DI DALAM
GEDUNG PADA KERUMUNAN HETEROGEN
MENGUNAKAN RECIPROCAL VELOCITY
OBSTACLE AGENT-BASED**

FADIL MUHAMMAD
07111550052009

DOSEN PEMBIMBING
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
Mochamad Hariadi, S.T., M.Sc., Ph.D.

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018



TESIS - TE142599

**SIMULASI EVAKUASI KEBAKARAN DI DALAM
GEDUNG PADA KERUMUNAN HETEROGEN
MENGUNAKAN RECIPROCAL VELOCITY
OBSTACLE AGENT-BASED**

FADIL MUHAMMAD
07111550052009

DOSEN PEMBIMBING
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
Mochamad Hariadi, S.T., M.Sc., Ph.D.

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Fadil Muhammad
NRP. 07111550052009

Tanggal Ujian : 26 Juni 2018
Periode Wisuda : September 2018

Disetujui oleh:

1. Dr. Supeno Mardi Susiki N, S.T., M.T. (Pembimbing I)
NIP: 197003131995121001

2. Mochamad Hariadi, S.T., M.Sc., Ph.D. (Pembimbing II)
NIP: 196912091997031002

3. Prof. Dr. Ir. Yoyon K Suprpto, M.Sc. (Penguji)
NIP: 195409251978031001

4. Dr. Eko Mulyanto Yuniarno, S.T., M.T. (Penguji)
NIP: 196806011995121009



Dekan Fakultas Teknologi Elektro

Dr. Tri Arief Sardjono, S.T., M.T.
NIP: 197002121995121001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul “**SIMULASI EVAKUASI KEBAKARAN DI DALAM GEDUNG PADA KERUMUNAN HETEROGEN MENGGUNAKAN RECIPROCAL VELOCITY OBSTACLE AGENT-BASED**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018

Fadil Muhammad

07111550052009

Halaman ini sengaja dikosongkan

SIMULASI EVAKUASI KEBAKARAN DI DALAM GEDUNG PADA KERUMUNAN HETEROGEN MENGGUNAKAN RECIPROCAL VELOCITY OBSTACLE AGENT-BASED

Nama Mahasiswa : Fadil Muhammad
NRP : 07111550052009
Pembimbing : 1. Dr. Supeno Mardi Susiki N, S.T., M.T.
2. Mochamad Hariadi, S.T., M.Sc., Ph.D.

ABSTRAK

Terjadinya bencana merupakan salah satu hal yang tidak dapat diduga oleh manusia termasuk bencana di dalam gedung atau ruangan. Umumnya setiap orang memiliki respon yang berbeda dalam menghadapi bencana tergantung situasi orang tersebut. Pada penelitian ini akan dibuat suatu simulasi evakuasi bencana dengan keadaan *agent* yang memiliki respon bencana yang berbeda atau heterogen.

Pada penelitian ini nantinya akan terdapat *multi-agent* yang memiliki status serta respon yang berbeda-beda. Status tiap *agent* dapat berupa kecepatan yang berbeda pada tiap *agent*. Pada penelitian ini akan dibuat relasi antar *agent* sehingga menciptakan respon yang berbeda pula dalam menghadapi suatu bencana. Agar *agent* dapat menghindari bencana serta dapat mencapai tujuan evakuasi maka diperlukan *path following* sebagai petunjuk serta dibutuhkan *collision avoidance* agar nantinya setiap *agent* tidak bertabrakan satu sama lain. Simulasi kerumunan ini menggunakan metode *reciprocal velocity obstacle* (RVO) berbasis *agent-based model* sebagai *collision avoidance* yang merangkap sebagai navigasi lokal.

Hasil dari penelitian pada simulasi ditemukan tabrakan paling banyak sebesar 7781 tabrakan dengan waktu simulasi paling lama sebesar 151 detik dan FPS paling kecil dengan jumlah FPS sebesar 33 FPS pada simulasi dengan jumlah *agent* sebanyak 300 *agent*. Pada desain yang telah ditentukan terdapat *agent follower* yang dalam kondisi tersesat sebanyak 30 *agent* pada simulasi heterogen tipe C.

Kata Kunci : *Crowd simulation, Multi-agent, Collision avoidance, Reciprocal collision avoidance, Agent-based model*, Evakuasi

CROWDS FIRE EVACUATION SIMULATION IN THE BUILDING ON HETEROGENEOUS AGENT USING RECIPROCAL VELOCITY OBSTACLE AGENT-BASED

By : Fadil Muhammad
Student Identity Number : 07111550052009
Supervisors : 1. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
2. Mochamad Hariadi, S.T., M.Sc., Ph.D.

ABSTRACT

Crowd simulation becomes one of the active research over the last three decade. Crowd simulation research is part of research about human behavior. One of research focus on crowd simulation was simulation at disaster evacuation. People who trapped in disaster usually has different behavior and abilities when evacuating. So the simulation looks more real, there are several simulations scenario used when disaster simulation like invariant speed and relation between virtual agents scenario.

In this research will be used agent-based reciprocal velocity obstacle method. The agent-based method used because applying a microscopic approach. With agent-based method then each virtual agents can have different behavior and abilities. Reciprocal velocity obstacle(RVO) used as virtual agents collision avoidance in crowds simulation.

Experiment result demonstrates the simulation of hundreds of agents in a three evacuation scenario. Leader-follower relations on heterogeneous agents result in the possibility of collisions between agents getting bigger with the number of collisions is 7781 collisions. The longest simulation time is 151 seconds with the smallest FPS number is 33 FPS. In the design of the simulation set there is agent follower who is in a lost state in heterogeneous simulation type C with the number agent who in a lost state is 30 agent.

Keywords : Crowd Simulation, Multi-Agent, Collision Avoidance, Reciprocal Velocity Obstacle, Agent-Based Model, Evacuation

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas terselesaikannya Thesis dengan judul *“Simulasi Evakuasi Kebakaran di Dalam Gedung pada Kerumunan Heterogen Menggunakan Reciprocal Velocity Obstacle Agent-Based”*. Thesis ini merupakan salah satu syarat kelulusan mahasiswa di jurusan Teknik Elektro FTE ITS.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Jurusan Teknik Elektro ITS, Bidang keahlian Jaringan Cerdas Multimedia, serta digunakan sebagai persyaratan menyelesaikan pendidikan S2. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. selaku Dekan Fakultas Teknologi Elektro ITS , Bapak Dr. Ir. Wirawan, DEA selaku Ketua Program Pasca Sarjana FTE Teknik Elektro ITS, Serta Bapak Dr. Eko Mulyanto, S.T., M.T. Selaku Ketua program bidang keahlian Jaringan Cerdas Multimedia FTE Teknik Elektro ITS.
3. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. dan Bapak Mochamad Hariadi, S.T., M.Sc, Ph.D. atas bimbingan selama mengerjakan thesis.
4. Ibu Susi Juniastuti, S.T., M.Eng. yang telah ikut serta membimbing dalam pengerjaan thesis serta bapak-ibu dosen pengajar Bidang keahlian Jaringan Cerdas Multimedia, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juli 2018

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
NOMENKLATUR	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Manfaat	3
2 KAJIAN PUSTAKA	5
2.1 Simulasi Kerumunan	5
2.2 <i>Multi-Agent</i>	6
2.3 Model Evakuasi Kebakaran	10
2.3.1 Faktor Afeksi dari pergerakan orang-orang	10
2.3.2 Model Pemilihan Rute	13
2.3.3 Ketidakpastian Perilaku	14
2.3.4 Integrasi dengan model lainnya	15
2.3.5 Metode Validasi	16
2.4 <i>Reciprocal Velocity Obstacles</i>	16
2.4.1 <i>Hybrid Reciprocal Velocity Obstacle</i>	19
2.4.2 <i>Optimal Reciprocal Collision Avoidance</i>	20
2.5 <i>Agent-Based Model</i>	22
2.5.1 <i>Collision Avoidance</i>	23
2.5.2 <i>Velocity Matching</i>	23
2.5.3 <i>Flock Centering</i>	24

2.6	<i>Crowd Simulation Hybrid Model</i>	25
2.6.1	<i>Zone-Based Model</i>	25
2.6.2	<i>Layers-Based Model</i>	26
2.6.3	<i>Sequential Models</i>	27
2.7	Algoritma A*	27
3	METODOLOGI PENELITIAN	33
3.1	Membangkitkan Lingkungan dan Karakteristik <i>Agent</i>	33
3.1.1	Lingkungan Simulasi	34
3.1.2	Karakteristik <i>Agent</i> Berdasarkan Koleksi dan Analisa Data	37
3.2	Navigasi Global dan Pergerakan Lokal	41
3.2.1	<i>Seeking</i>	42
3.2.2	<i>Steering</i>	44
3.2.3	<i>Collision Avoidance</i>	45
3.3	Penerapan <i>Agent</i> Heterogen dengan <i>Agent-Based Model</i>	46
4	PENGUJIAN	53
4.1	Pengujian Sistem Simulasi	53
4.1.1	Pengujian <i>Seeking</i>	53
4.1.2	Pengujian <i>Steering</i>	55
4.1.3	Pengujian <i>Collision Avoidance</i>	56
4.1.4	Pengujian <i>Collision Detection</i>	58
4.1.5	Pengujian Ekstraksi Data	58
4.2	Hasil Simulasi	59
5	PENUTUP	71
5.1	Kesimpulan	71
5.2	Saran	71
	DAFTAR PUSTAKA	73
	LAMPIRAN	82

DAFTAR GAMBAR

2.1	Ilustrasi pembagian pendekatan simulasi kerumunan pada tiga pendekatan(Ijaz et al., 2015)	6
2.2	Ilustrasi <i>agent</i> terhadap lingkungan	6
2.3	Topologi kooperasi <i>multi-agent</i> (Doran et al., 1997)	7
2.4	<i>multi-agent</i> homogen non-komunikasi(Glavic, 2006)	8
2.5	<i>multi-agent</i> homogen berkomunikasi(Glavic, 2006)	9
2.6	<i>multi-agent</i> heterogen non-komunikasi(Glavic, 2006)	9
2.7	<i>multi-agent</i> heterogen berkomunikasi(Glavic, 2006)	10
2.8	Osilasi yang terjadi pada <i>collision avoidance</i> menggunakan <i>velocity obstacles</i> (Van Berg et al., 2008)	17
2.9	Penghindaran tabrakan oleh <i>velocity obstacle</i> (kiri) dan penghindaran tabrakan oleh RVO(kanan)(Van Berg et al., 2008)	18
2.10	Gerak saat <i>agent</i> menghindari tabrakan dengan <i>agent</i> lain menggunakan RVO(Van Berg et al., 2008)	18
2.11	<i>hybrid reciprocal velocity obstacle HRVO_{A B}</i> pada <i>agent A</i> yang ingin menghindari <i>agent B</i> (Snape et al., 2011)	19
2.12	Dua <i>agent A</i> dan <i>agent B</i> (kiri). Potongan kecepatan rintangan <i>VO</i> untuk <i>agent A</i> yang terinduksi oleh <i>agent B</i> (tengah). <i>Half-plane</i> pada <i>optimal reciprocal collision avoidance</i> dengan kecepatan <i>ORCA</i> yang diizinkan untuk <i>agent A</i> dan <i>agent B</i> (kanan)(Berg et al., 2011)	21
2.13	<i>Separation</i>	23
2.14	<i>Aligment</i>	24
2.15	<i>Cohession</i>	25
2.16	Cara pembagian pendekatan simulasi kerumunan berdasarkan model berbasis zona(Xiong et al., 2012)	26
2.17	Cara pembagian pendekatan simulasi berdasarkan model berbasis lapisan(<i>layer</i>)(Cristian et al., 2012)	27
2.18	Contoh <i>graph</i>	28
2.19	Perbedaan penentuan jalur <i>breadth-first search</i> dan <i>best-first search</i>	30
3.1	Metodologi Penelitian	33
3.2	Gedung robotika ITS(rob, 2018)	34

3.3	Lingkungan gedung robot tampak luar	35
3.4	Lingkungan gedung robot tampak dalam	35
3.5	Lingkungan gedung acara robot tampak dalam dilihat dari atas	36
3.6	Peta saat simulasi dilakukan	36
3.7	Lingkungan gedung robot bengkel tampak dalam dilihat dari atas	37
3.8	Lingkungan gedung robot laboratorium tampak dalam dilihat dari atas	37
3.9	<i>Seeking</i> pada <i>agent leader</i> (kiri), <i>seeking</i> pada <i>agent</i> independen(kanan)	43
3.10	<i>Seeking</i> pada <i>agent follower</i> terhadap <i>agent</i> independen(kiri) , <i>seeking</i> pada <i>agent follower</i> terhadap <i>agent leader</i> (kanan), lingkaran merah adalah wilayah <i>line of sight</i>	43
3.11	Pembagian kondisi berdasarkan jenis <i>agent</i>	47
3.12	FSM pada <i>agent follower</i> pada skenario heterogen tipe A	48
3.13	FSM pada <i>agent follower</i> pada skenario heterogen tipe B	50
3.14	FSM pada <i>agent follower</i> pada skenario heterogen tipe C	51
4.1	Seeking pada <i>agent leader</i> (kiri), seeking pada <i>agent</i> independen(kanan)	54
4.2	Seeking pada <i>agent follower</i> terhadap <i>agent</i> independen(kiri) , seeking pada <i>agent follower</i> terhadap <i>agent leader</i> (kanan) . . .	54
4.3	Proses seeking pada kerumunan	55
4.4	Pada awal <i>agent follower</i> tertinggal <i>agent leader</i> (kiri), <i>agent follower</i> dan <i>leader</i> menyesuaikan agar dapat berdekatan(kanan)	56
4.5	Pada awal <i>agent follower</i> tertinggal <i>agent</i> independen(kiri), <i>agent follower</i> menyesuaikan kecepatan dengan <i>agent</i> independen agar dapat berdekatan(kanan)	56
4.6	<i>Agent</i> menghindari rintangan statis	57
4.7	<i>Agent</i> menghindari rintangan dinamis(<i>agent</i> lainnya)	57
4.8	<i>Agent</i> saat belum tabrakan(atas) dan saat mengalami tabrakan dengan (<i>agent</i> lainnya)(bawah).	58
4.9	Peta saat simulasi dilakukan	60
4.10	Simulasi dengan jumlah <i>agent</i> yang relatif sedikit	61
4.11	Simulasi dengan jumlah <i>agent</i> yang banyak	62
4.12	Grafik jumlah tabrakan pada setiap simulasi	63
4.13	Grafik waktu simulasi pada setiap simulasi	65
4.14	Grafik <i>frame per second</i> (FPS) pada setiap simulasi	67

4.15 Grafik <i>agent follower</i> yang tersesat pada simulasi heterogen tipe A, B dan C	68
------------------------------------------------------------------------------------------------------	----

Halaman ini sengaja dikosongkan

DAFTAR TABEL

3.1	Tabel Proses seeking pada setiap <i>agent</i>	43
4.1	Tabel ekstraksi data	59
1	Hasil Simulasi <i>agent</i> dengan kecepatan sama dan tanpa relasi .	77
2	Hasil simulasi <i>agent</i> dengan kecepatan berbeda dan tanpa relasi	78
3	Hasil Simulasi Heterogen Tipe A	79
4	Hasil Simulasi Heterogen Tipe B	80
5	Hasil Simulasi Tipe C	81

Halaman ini sengaja dikosongkan

NOMENKLATUR

\mathbf{v}_n	Kecepatan pada <i>agent</i> n
VO_n^m	<i>Velocity obstacle agent</i> n terhadap <i>agent</i> m
RVO_n^m	RVO <i>agent</i> n terhadap <i>agent</i> m
$HRVO_{n m}$	HRVO <i>agent</i> n terhadap <i>agent</i> m
$ORCA_{n m}$	ORCA <i>agent</i> n terhadap <i>agent</i> m
u	Vektor dari kecepatan relatif ORCA
$f(b)$	Fungsi pada algoritma A* untuk mencari bobot minimal
$g(n)$	Fungsi biaya perpindahan dari <i>node</i> pertama hingga ke-n
$h(n)$	Fungsi <i>heuristic</i> dari n menuju tujuan
V_{leader}	Kecepatan <i>leader</i>
$V_{initialL}$	Kecepatan awal <i>leader</i>
S_{agent}	Posisi antara <i>leader</i> dan <i>follower</i>
S_x	Jarak yang diinginkan antara <i>leader</i> dan <i>follower</i>
S_{LoSL}	Radius <i>Line of Sight</i> dari <i>agent leader</i>
$V_{follower}$	Kecepatan <i>follower</i>
$V_{initialF}$	Kecepatan awal <i>follower</i>
S_{agent}	Posisi antara <i>leader</i> dan <i>follower</i>
S_{LoSF}	Radius <i>Line of Sight</i> dari <i>agent follower</i>

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Simulasi kerumunan serta pergerakan dari manusia telah menjadi penelitian yang aktif selama tiga dekade terakhir ini. Penelitian ini merupakan bagian dari penelitian tentang sifat serta tingkah laku dari manusia. Terdapat beberapa tempat yang menjadi fokus penelitian ini baik itu tempat terbuka seperti di stadion, taman maupun di zona pejalan kaki serta tempat tertutup seperti gedung, stasiun, bandara atau rumah sakit. Tujuan dari penelitian ini pun bermacam-macam seperti proses antrian, evakuasi maupun sebuah proses kerumunan dari pejalan kaki.

Terdapat dua jenis pendekatan saat melakukan simulasi kerumunan ini yaitu pendekatan makroskopis dan pendekatan mikroskopis. Pada pendekatan makroskopis kerumunan dianggap suatu kumpulan individu yang bersifat homogen. Karena pada pendekatan ini tidak terlalu memperhatikan tiap individu dalam kerumunan maka proses komputasi pada kerumunan ini dapat berlangsung cepat namun kurang terlihat realistis. Metode-metode yang termasuk pada pendekatan makroskopis ini adalah *regression* dan *fluid-dynamics model*. Pada simulasi kerumunan yang menggunakan pendekatan mikroskopis akan menganggap suatu kerumunan adalah kumpulan dari proses tiap-tiap individu. Dengan begitu tiap individu dalam suatu kerumunan akan dianggap sebagai *agent* otonom yang memiliki proses pengambilan keputusan, tindakan, suatu perilaku khusus dan interaksi dengan *agent* lainnya di dalam kerumunan. Berkebalikan dengan pendekatan makroskopis, pendekatan ini memerlukan proses komputasi yang lebih lambat dibanding proses mikroskopis namun lebih terlihat realistis. Pada pendekatan mikroskopis terdapat 3 model atau metode yang digunakan untuk melakukan simulasi kerumunan yaitu *cellular automata(CA)*, *social force model(SFM)* dan *agent-based model*.

Selain harus memperhatikan navigasi secara keseluruhan atau navigasi global, dalam suatu simulasi diperlukan navigasi lokal yang dapat menyelesaikan permasalahan pada rintangan dinamis. Navigasi global yang dibuat hanya memperhatikan adanya jalur-jalur yang akan dilewati oleh *agent* dengan rintangan yang statis. Padahal dalam suatu kerumunan evakuasi diperlukan navigasi lokal yang memperhatikan adanya rintangan-rintangan yang berge-

rak secara dinamis agar *agent* dapat bergerak melewati rintangan tersebut dan simulasi dapat berjalan dengan lancar. Terdapat beberapa metode yang digunakan dalam hal mengatasi rintangan secara dinamis ini yaitu *velocity obstacle*(VO) serta *reciprocal velocity obstacle* (RVO). Selain itu terdapat pula metode pengembangan dari RVO untuk mengatasi permasalahan *reciprocal dances* dari RVO seperti metode *hybrid reciprocal velocity obstacle*(HRVO) dan *optimal reciprocal collision avoidance* (ORCA).

Simulasi evakuasi merupakan salah satu tema penelitian pada simulasi kerumunan. Simulasi evakuasi dapat terjadi di beberapa tempat baik tempat terbuka maupun tempat tertutup. Simulasi evakuasi terjadi saat terjadi bencana yang dapat disebabkan oleh alam seperti adanya gempa bumi, longsor, tsunami maupun kesalahan manusia terjadinya kebakaran, kebocoran gas dan bencana lainnya. Pada tempat-tempat yang umum dan tertutup terjadinya bencana dapat menimpa pada beragam jenis manusia atau *agent*. Adanya keragaman *agent* pada wilayah umum ini mengakibatkan kerumunan *agent* pada suatu tempat bersifat heterogen. sifat-sifat heterogen ini dapat berupa perbedaan pergerakan, respon serta hubungan antara *agent*. Pergerakan serta respon tiap *agent* dapat ditentukan dengan perbedaan usia atau perbedaan jenis kelamin antar *agent*. Sedangkan hubungan antar *agent* dapat dibuat hubungan dapat mempengaruhi respon pada *agent*. Perbedaan dengan adanya kerumunan yang tiap individu yang bersifat heterogen maka simulasi kerumunan ini akan memperhatikan tiap-tiap individunya. Dengan begitu simulasi ini nantinya akan terlihat lebih realistis.

1.2 Rumusan Masalah

1. Banyaknya *agent* menyebabkan besarnya proses komputasi.
2. Sifat *multi-agent* yang heterogen dalam kerumunan membuat simulasi memiliki kompleksitas yang tinggi.
3. Tingginya kompleksitas yang terjadi pada simulasi dapat memberikan resiko galat yang pada *agent*
4. *Agent* dengan jumlah yang banyak memiliki resiko besar kesalahan navigasi serta tabrakan antar *agent*.

1.3 Tujuan

1. Memberikan respon serta status pada *multi-agent* secara heterogen pada suatu kerumunan.
2. Memberikan kendali kerumunan pada *multi-agent* menggunakan metode *reciprocal collision avoidance* dan *agent-based*.

1.4 Batasan Masalah

1. Simulasi berupa simulasi evakuasi bencana.
2. Lingkungan simulasi berupa gedung satu lantai.
3. Skenario simulasi berdasarkan data koleksi dan analisa yang sudah ada.
4. Kendali kerumunan menggunakan *reciprocal collision avoidance* dan *agent-based*.

1.5 Manfaat

Dengan adanya kerumunan *multi-agent* yang heterogen diharapkan simulasi evakuasi bencana terlihat lebih nyata dan sesuai dengan kondisi yang sebenarnya.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

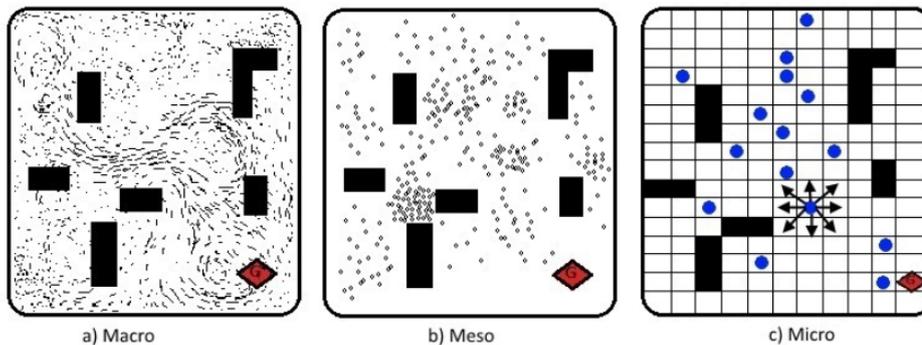
2.1 Simulasi Kerumunan

Simulasi kerumunan merupakan salah satu dari simulasi perilaku manusia. Secara umum terdapat dua pendekatan saat melakukan simulasi kerumunan ini yaitu menggunakan pendekatan makroskopis dan pendekatan mikroskopis. Pada pendekatan makroskopis suatu kerumunan dianggap sekumpulan individu yang homogen. Pada pendekatan ini kemampuan-kemampuan dari tiap-tiap individu seperti pergerakan dan tindakan tidak terlalu diperhatikan. Dengan begitu simulasi kerumunan menggunakan pendekatan makroskopis ini kurang realistis. Metode-metode yang termasuk pada pendekatan makroskopis ini adalah *regression* dan *fluid-dynamics model*.

Pada simulasi kerumunan yang menggunakan pendekatan mikroskopis akan menganggap suatu kerumunan adalah kumpulan dari proses tiap-tiap individu. Dengan begitu tiap individu dalam suatu kerumunan akan dianggap sebagai *agent* otonom yang memiliki proses pengambilan keputusan, tindakan, suatu perilaku khusus dan interaksi dengan *agent* lainnya di dalam kerumunan. Pada pendekatan mikroskopis terdapat tiga model atau metode yang digunakan untuk melakukan simulasi kerumunan yaitu *cellular automata*(CA), *social force model*(SFM) dan *agent-based model*. Pendekatan CA memodelkan pergerakan tiap individu dalam waktu diskrit dan ruang yang diwakili oleh sel-sel yang berinteraksi(Blue et al., 1997)(Blue and Adler, 1998)(Blue and Adler, 2001). SFM menganggap tiap-tiap individu sebagai partikel pasif yang dikendalikan oleh suatu kerumunan(Helbing and Molnar, 1995).*agent-based model* mengansumsikan setiap *agent* dalam kerumunan adalah individu yang memiliki perilaku yang otonom(Reynolds, 1987).

Selain melalui dua pendekatan, ada beberapa penelitian yang menambah pendekatan *mesoscopic* sebagai pendekatan lainnya(Ijaz et al., 2015). Pendekatan *mesoscopic* ini merupakan irisan antara pendekatan mikroskopis dan pendekatan makroskopis. Pada pendekatan *mesoscopic* pergerakan pada kerumunan diserahkan pada setiap individu namun simulasi relatif dalam skala yang lebih besar. Pada penelitian ini metode *cellular automata* dimasukkan pada pendekatan *mesoscopic* penelitian ini. *Cellular automata* resolusi spasial yang memiliki komputasi yang sedikit lebih besar dibandingkan dengan

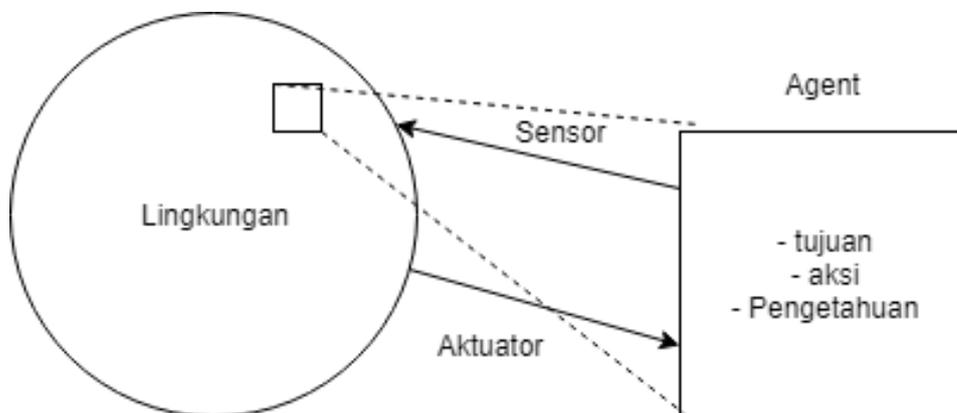
pendekatan makroskopis. Namun berbeda dengan pendekatan makroskopis, pendekatan ini memakai tiap individu sebagai pergerakan pada kerumunan. Pada Gambar 2.1 diilustrasikan pembagian jenis pendekatan dengan tiga pendekatan simulasi kerumunan.



Gambar 2.1: Ilustrasi pembagian pendekatan simulasi kerumunan pada tiga pendekatan (Ijaz et al., 2015)

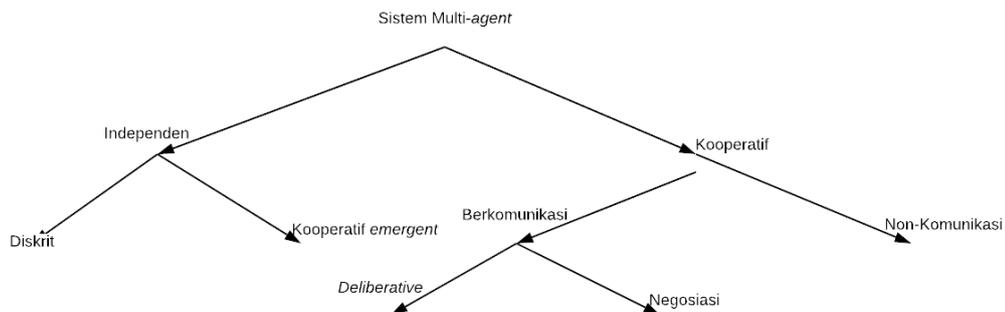
2.2 Multi-Agent

Agent adalah sesuatu yang dapat memahami lingkungan sekitar melalui sensor yang dimilikinya dan bertindak atas lingkungan tersebut melalui aktuator (Norvig). Agar *agent* dapat bertindak terhadap suatu lingkungan maka diperlukan beberapa hal yang terdapat pada suatu *agent* seperti tujuan, aksi dan pengetahuan. Ilustrasi perilaku *agent* terhadap lingkungan dapat dilihat pada Gambar 2.2. Apabila *Agent* manusia memiliki mata, telinga, kulit dan organ lainnya sebagai sensor serta memiliki tangan, mulut serta beberapa organ lainnya sebagai aktuator. *Agent* robot memiliki kamera sebagai sensor dan motor sebagai aktuatornya.



Gambar 2.2: Ilustrasi *agent* terhadap lingkungan

Pada suatu lingkungan yang memiliki beberapa agent maka disebut pula *multi-agent*. Pada lingkungan yang memiliki banyak *agent* atau *multi-agent* terdapat dua keadaan pada tiap *agent*. Suatu *agent* dapat independen ataupun kooperatif terhadap *agent* lainnya (Doran et al., 1997). Ilustrasi topologi kooperasi dari *multi-agent* dapat dilihat pada Gambar 2.3. Suatu *agent* dapat tidak terpengaruh terhadap *agent* lainnya pada suatu lingkungan. Namun *agent* yang independen ini juga dapat terpengaruh apabila terdapat perilaku *agent* lainnya yang membuat suatu *agent* merubah responnya terhadap lingkungannya. Dengan begitu kehadiran *agent* dapat merubah keputusan *agent* lainnya meskipun *agent* tersebut tidaklah berkooperasi. Suatu *agent* juga dapat bersifat kooperatif terhadap *agent* lainnya. sifat kooperatif ini dapat berupa komunikasi antar *agent* maupun tidak. *Agent* yang tidak berkomunikasi dengan *agent* lainnya hanya memperhatikan serta merespon tanpa adanya komunikasi dari *agent* lainnya. Pada *multi-agent* yang berkomunikasi antar *agent* terdapat sifat *deliberative* dan negosiasi. Pada kasus *multi-agent deliberative* setiap *agent* akan menjalankan aksinya dengan cara berkooperasi dengan *agent* lainnya. perbedaan antara *deliberative* dan negosiasi terletak pada kompetisi antar *agent*. Saat *agent* memutuskan untuk negosiasi dengan *agent* lainnya berarti mereka tidak berkompetisi dengan *agent* lainnya.

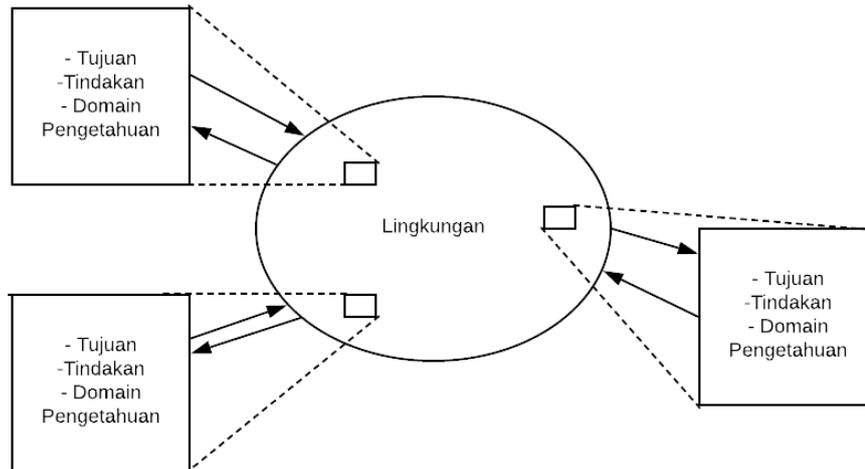


Gambar 2.3: Topologi kooperasi *multi-agent*(Doran et al., 1997)

Dengan adanya sistem *multi-agent* ini maka terdapat pembagian mengenai karakteristik yang dimiliki oleh *multi-agent* (Glavic, 2006). Berikut adalah pembagian jenis *agent* pada sistem *multi-agent*:

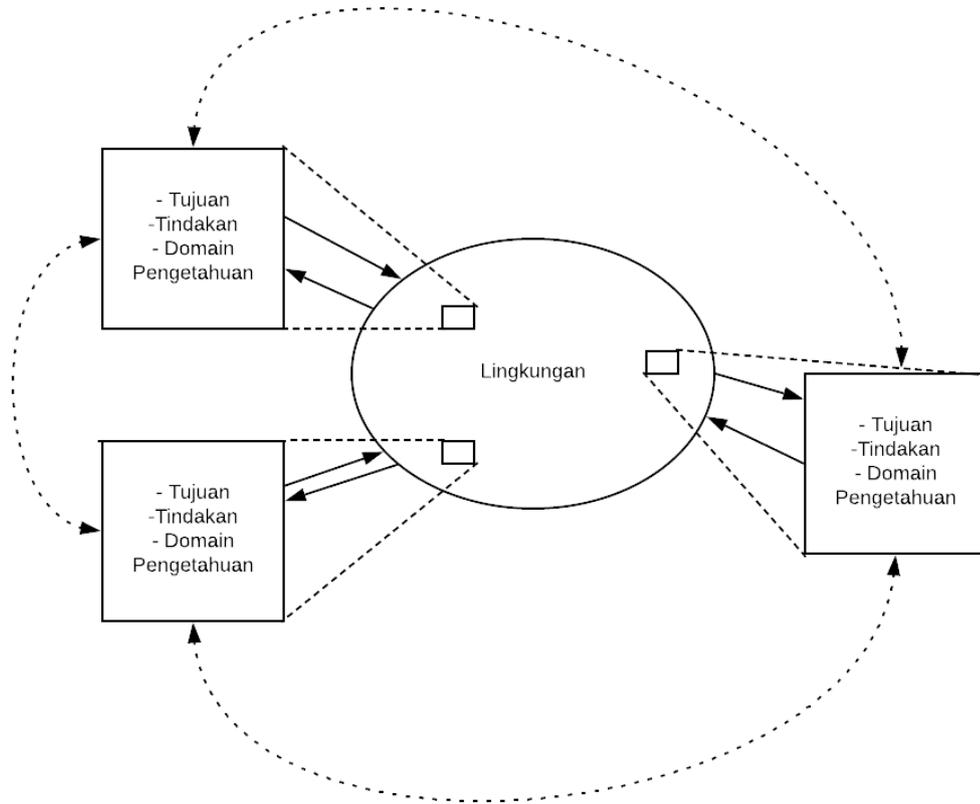
1. **Sistem *multi-agent* homogen non-komunikasi.** Pada sistem *multi-agent* ini setiap *agent* yang berada pada suatu lingkungan memiliki pengetahuan serta kemampuan yang sama dengan *agent* lainnya dan akhir-

nya memiliki aksi yang serupa antar *agent*. Selain itu sistem *multi-agent* ini tidak memiliki komunikasi antar *agent*. Pada Gambar 2.4 diilustrasikan sistem *multi-agent* homogen tanpa adanya komunikasi.

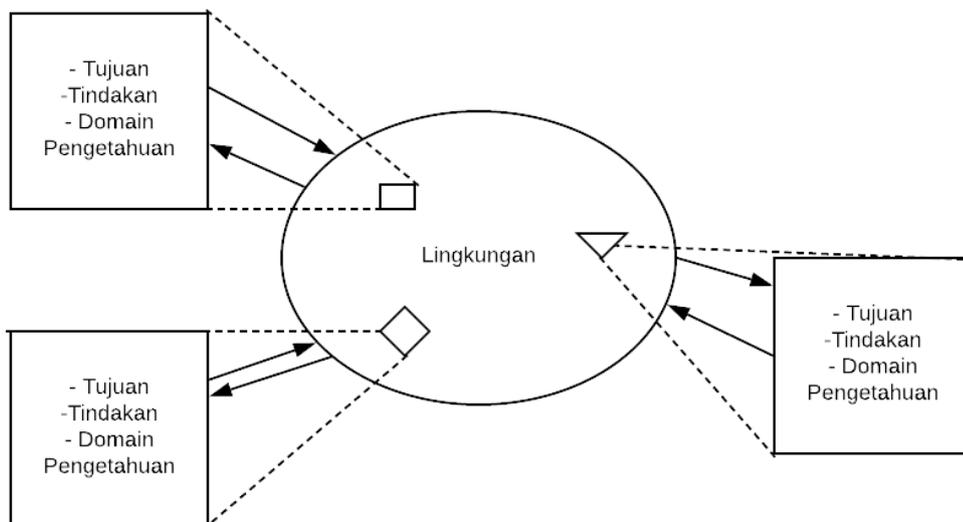


Gambar 2.4: *multi-agent* homogen non-komunikasi (Glavic, 2006)

2. **Sistem *multi-agent* homogen berkomunikasi.** Sama seperti kategori sebelumnya sistem *multi-agent* ini memiliki kemampuan yang sama antar *agent* namun yang membedakan adalah terjadinya komunikasi yang terjalin antar *agent*. Komunikasi ini dapat berupa negoisasi maupun *deliberative*. Pada Gambar 2.5 diilustrasikan sistem *multi-agent* homogen dengan adanya komunikasi.
3. **Sistem *multi-agent* heterogen non-komunikasi.** Berkebalikan dengan sistem *multi-agent* homogen, pada sistem *multi-agent* heterogen memiliki kemampuan serta pengetahuan yang berbeda antar *agent*. Dengan adanya perbedaan kemampuan ini, aksi yang dilakukan oleh *agent* bisa jadi berbeda meskipun memiliki tujuan yang sama. Pada sistem *multi-agent* ini, *agent* tidak dapat berkomunikasi dengan *agent* lainnya. Pada Gambar 2.6 diilustrasikan sistem *multi-agent* heterogen tanpa dapat berkomunikasi.
4. **Sistem *multi-agent* heterogen berkomunikasi.** Pada sistem ini sama seperti sistem *multi-agent* heterogen non-komunikasi hanya saja pada sistem ini terdapat komunikasi yang terjadi antar satu *agent* dengan

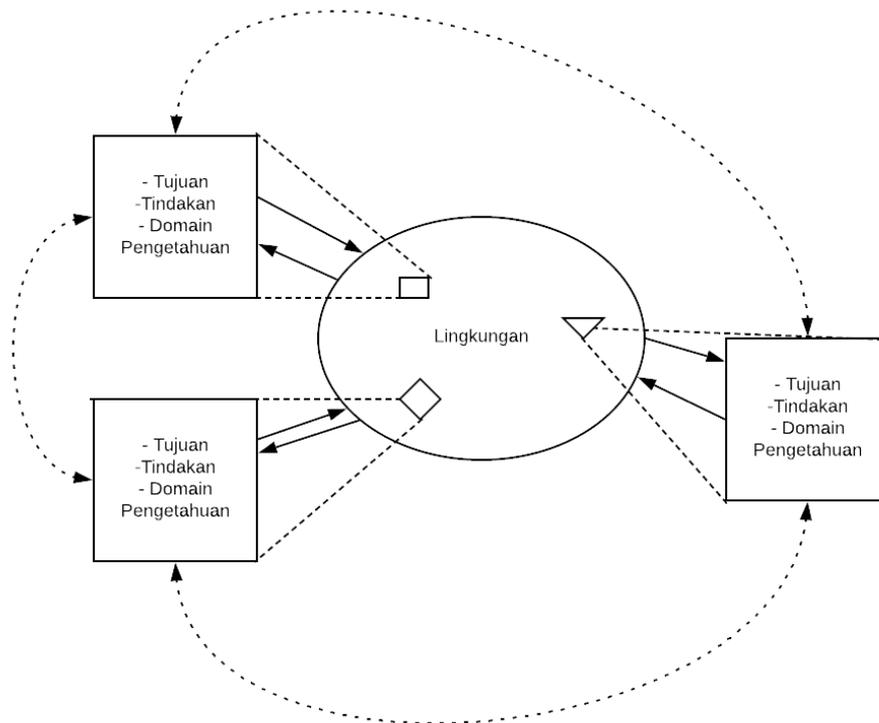


Gambar 2.5: *multi-agent* homogen berkomunikasi (Glavic, 2006)



Gambar 2.6: *multi-agent* heterogen non-komunikasi (Glavic, 2006)

agent lainnya. Dengan begitu terjadi interaksi antar *agent* dan dapat mempengaruhi keputusan dari *agent*. Pada Gambar 2.7 diilustrasikan sistem *multi-agent* heterogen yang dapat berkomunikasi.



Gambar 2.7: *multi-agent* heterogen berkomunikasi (Glavic, 2006)

2.3 Model Evakuasi Kebakaran

Pemodelan evakuasi menggunakan komputer merupakan riset yang telah lama dilakukan dan terus berkembang hingga kini. Terdapat beberapa jenis evakuasi tergantung kondisi, tempat serta jenis evakuasi. Pada penelitian Gwynne dkk (Gwynne et al., 1999) terdapat dua puluh dua jenis model evakuasi. Salah satu dari bentuk evakuasi bencana yaitu adalah evakuasi dari kebakaran. Ronchi E (Ronchi, 2016) menjelaskan peta penelitian pada model evakuasi kebarakaran yaitu terdiri dari faktor afeksi dari pergerakan orang-orang, pemodelan dalam pencarian rute, ketidakpastian dari perilaku orang-orang, gabungan(integrasi) dengan model lainnya, validitas suatu metode.

2.3.1 Faktor Afeksi dari pergerakan orang-orang

Representasi gerakan orang dalam model evakuasi umumnya didasarkan pada simulasi kecepatan berjalan tanpa hambatan yang diadopsi oleh pada

korban evakuasi serta beberapa faktor yang mempengaruhi pergerakan tersebut. Beberapa model yang digunakan dalam evakuasi ini adalah *steering behaviour*, *social force model*, *floor field cellular automaton* dan model lainnya. Model evakuasi ini biasanya membutuhkan input distribusi kecepatan agar dapat menjalankan simulasi kemudian menggunakan metode lainnya untuk menghitung kecepatan yang terhambat oleh halang rintang. Terhambatnya suatu pergerakan para korban evakuasi dapat disebabkan oleh dua faktor yaitu karena faktor interaksi dengan korban evakuasi lainnya dan faktor lingkungan sekitar. Metode ini disebut pula dengan fundamental diagram untuk simulasi gerakan manusia untuk menguji perilaku yang muncul yang berasal dari metode pemodelan yang digunakan. Ketika metode ini menggunakan parameter kelajuan berjalan, kepadatan serta aliran kerumunan, maka terdapat beberapa faktor yang akan mempengaruhi hambatan dari kecepatan berjalan. Pada model ini setidaknya ada beberapa faktor eksplisit yang dapat mempengaruhi simulasi evakuasi yaitu 1) Penggunaan tenaga fisik pada *agent* dalam simulasi, 2) dampak motivasi serta persepsi resiko, 3) dampak adanya asap, 4) perbedaan kebiasaan serta pengabungan aliran kerumunan.

Penggunaan tenaga fisik merupakan salah satu kunci dalam pengerahan kecepatan dalam pergerakan para korban evakuasi, khususnya dalam evakuasi yang memiliki jarak yang relatif panjang serta evakuasi yang tidak hanya satu lantai. Pada saat ini infrastruktur gedung-gedung sendiri memiliki tinggi beberapa ratus meter dan tergolong *high rise building*. Dengan semakin besar serta kompleksnya tipe gedung-gedung maka tenaga fisik merupakan salah satu faktor yang menentukan dalam proses evakuasi. Sebagai contohnya saat seseorang memanjat suatu tempat dikarenakan suatu eskalator yang terdapat pada gedung tersebut berhenti berjalan maka akan memakan banyak tenaga. Selain itu dengan banyaknya tenaga yang dikeluarkan maka akan membuat seseorang akan memperlambat langkahnya serta berhenti saat proses evakuasi. Melambatnya kelajuan pergerakan saat evakuasi ini dapat mengakibatkan terhambatnya aliran kerumunan pada evakuasi. Pengembangan kedepannya suatu evakuasi model seharusnya memasukan parameter penggunaan tenaga fisik yang akan mempengaruhi kecepatan para korban evakuasi serta hambatan yang terjadi pada aliran kerumunan. Hal ini termasuk waktu maksimal yang dapat ditempuh oleh pejalan kaki sebelum akhirnya para pejalan kaki merasa lelah dan memperlambat kelajuannya bahkan hingga akhirnya berhenti beristirahat. Hal ini merefleksikan bagaimana penggunaan tenaga fisik

sangat mempengaruhi konsep dari fundamental diagram. Selain itu perlu pula dilakukan penelitian bagaimana besar badan seseorang, usia, gender dapat mempengaruhi pergerakan suatu manusia yang akhirnya akan mempengaruhi aliran suatu kerumunan.

Salah satu yang menjadi faktor afeksi dari pergerakan suatu manusia adalah adanya motivasi pada suatu manusia. Dalam suatu evakuasi adanya motivasi ini dapat menjadi faktor tambahan selain tenaga fisik. Semakin besar motivasi maka dapat menambah kelajuan serta tenaga fisik pada manusia. Motivasi ini dapat dipengaruhi oleh beberapa hal misalnya saja seseorang dapat meningkatkan motivasinya apabila menemukan tujuan yang dicapai (Khisty, 1985). Representasi dari kebiasaan manusia serta motivasi dari pergerakan itu sendiri dapat dihubungkan kepada simulasi terhadap persepsi resiko dari seseorang. Pada proses evakuasi pun terdapat satu faktor yang dapat mempengaruhi suatu pergerakan pada manusia yaitu kondisi panik yang terjadi pada manusia. Pada penelitiannya Fahy dkk (Fahy et al., 2009) terdapat beberapa kasus yang disebabkan oleh kepanikan yang terjadi saat bencana kebakaran. Terdapat pula beberapa kasus orang-orang yang dapat mengendalikan kepanikannya akhirnya dapat mengatasi bencana kebakaran dan dapat melakukan evakuasi.

Selain faktor internal seperti tenaga fisik yang terdapat pada manusia serta kebiasaan serta motivasi pada manusia, terdapat pula faktor eksternal yang mempengaruhi pergerakan dari korban evakuasi. Pada bencana kebakaran faktor yang tidak dapat dikesampingkan adalah dampak adanya asap terhadap pergerakan para korban evakuasi. Dengan adanya asap yang disebabkan oleh api saat kebakaran dapat menghalangi serta menghambat pergerakan para korban evakuasi. Penurunan pergerakan ini salah satu disebabkan karena berkurangnya penglihatan pada korban evakuasi. Dengan begitu terdapat penurunan pergerakan serta perpindahan yang terjadi saat evakuasi.

Beberapa model evakuasi seperti *steering model* dan *social force model* memiliki karakteristik *agent* yang memiliki kondisi serta kemampuan yang sama atau bisa disebut juga *agent* yang homogen. Pada kenyataannya didalam suatu evakuasi beberapa orang akan memiliki serta kemampuan yang berbeda-beda atau yang bisa disebut dengan heterogen. Selain itu aliran kerumunan juga perlu diperhatikan dalam suatu evakuasi. Dengan adanya pemisahan aliran membuat suatu evakuasi dapat berjalan lancar atau mungkin dapat menghambat suatu evakuasi. Dengan adanya pemisahan aliran dapat mem-

buat kepadatan suatu evakuasi dapat dikurangi. Dengan begitu keadaan *agent* yang heterogen serta memiliki aliran yang terbagi-bagi sehingga menciptakan kepadatan yang rendah dapat mempengaruhi faktor afeksi dari pergerakan orang-orang dalam evakuasi

2.3.2 Model Pemilihan Rute

Beberapa perilaku yang dapat diadopsi dalam pemodelan rute ini antara lain:

1. ***path-finding***: Proses pencarian titik tujuan *agent* pada suatu simulasi.
2. ***local movement***: Pergerakan *agent* dalam menghadapi lingkungan sekitarnya seperti menghindari rintangan baik rintangan statis maupun rintangan dinamis.
3. ***locomotion***: Representasi pergerakan individu dari manusia termasuk dari *biomechanic* dari pergerakan tubuh. Pergerakan tubuh ini dapat direpresentasikan dari struktur tubuh manusia, cara berjalan maupun cara berbalik dari manusia.

Pada saat ini penelitian tentang eksplorasi pergerakan tubuh manusia masih pada tahapan awal dalam suatu penelitian evakuasi kerumunan. Oleh karena itu model pemilihan rute dalam evakuasi kerumunan masih menggunakan *path-finding* atau pencarian jalur serta *local movement* atau pergerakan lokal. Kedepannya diperlukan penelitian serta implementasi dari *biomechanic* manusia agar simulasi evakuasi dapat terlihat sesuai dengan keadaan sebenarnya sehingga hasil simulasi seakurat mungkin. Terdapat beberapa metode yang merepresentasikan dari pemilihan rute dan *agent* dalam simulasi akan melakukan pencarian rute terpendek serta tercepat dalam berbagai kondisi. Beberapa metode juga memperhatikan adanya variabel yang terjadi saat adanya bencana seperti adanya asap, adanya antrian dalam kerumunan, adanya hubungan kekeluargaan pada *agent*, jumlah jalan keluar yang tersedia dan variabel-variabel lainnya yang dapat mempengaruhi pemilihan rute dan pergerakan lokal dari *agent*. Pencarian rute dapat dilakukan dengan cara menggunakan *deterministic* atau algoritma probabilitas.

Pertanyaan mendasar dari penggunaan rute dalam simulasi evakuasi adalah: apakah penggunaan rute tersebut dapat dilakukan pada proses evakuasi? Apa saja faktor-faktor yang membuat pemilihan rute sesuai dengan proses

evakuasi pada keadaan yang sebenarnya? Untuk saat ini algoritma yang digunakan dalam pemilihan rute pada simulasi evakuasi berdasarkan asumsi berdasarkan pemodelan seperti rute terpendek atau rute tercepat dibandingkan menggunakan *data-driven* yaitu data-data berdasarkan kejadian yang sudah terjadi. Penggunaan *data-driven* sebenarnya lebih baik digunakan karena sesuai dengan statistik yang terjadi pada kejadian-kejadian sebelumnya. Pendekatan ini lebih masuk akal karena dalam skenario evakuasi terdapat beberapa keterbatasan dalam hal pencarian jalur. Namun masalah terbesar dalam penggunaan *data-driven* ini adalah masih terbatasnya penelitian yang memahami faktor-faktor mendasar yang mempengaruhi pilihan rute para korban evakuasi saat evakuasi berlangsung. Apalagi ditambah dengan adanya masalah kompleksitas pada suatu bangunan yang akhirnya menambah jumlah faktor-faktor apa saja yang membuat para korban evakuasi memilih rute yang mereka pilih.

Saat ini, bangunan modern dibuat dengan konsep bangunan pintar dengan sistem dinamis serta informasi apabila terjadi proses evakuasi pada orang-orang yang di dalam gedung. Misalnya saja dengan adanya informasi dalam pertolongan pertama, adanya sistem akustik yang baik serta tanda-tanda evakuasi dalam gedung. Pada penelitian lebih lanjut seharusnya mengikut sertakan karakteristik dari bangunan pintar modern sebagai suatu faktor pada proses simulasi evakuasi. Hal inilah yang akan menjadi salah satu variabel yang digunakan dalam pemilihan suatu rute *agent* dalam proses evakuasi. Selain itu perlu pula menambahkan proses interaksi antar *agent* dalam suatu proses pencarian jalur. Dengan begitu proses simulasi evakuasi ini akan lebih akurat.

2.3.3 Ketidakpastian Perilaku

Pada model simulasi evakuasi kebakaran yang menggunakan pendekatan probabilitas akan melakukan distribusi dari variabel-variabel yang mempengaruhi proses evakuasi menggunakan model stokastik. Beberapa perilaku yang tidak pasti pada perilaku manusia yang terjebak evakuasi berdasarkan model stokastik maka akan disebut dengan ketidakpastian perilaku. Model simulasi evakuasi ini biasanya menggunakan sampling *pseudo-random* dari distribusi untuk merepresentasikan bagian mana saja yang menjadi kunci dalam proses evakuasi seperti kelajuan dalam berjalan, waktu sebelum evakuasi serta variabel lainnya. Sampling *pseudo-random* ini dapat bekerja menggunakan

metode numerik untuk generasi *pseudo-random* yang akan diberikan pada distribusi. Penggunaan *pseudo-random* ini dilakukan karena nilai yang dihasilkan memenuhi pegujian statistik pada acak namun juga tetap memakai prosedur matematika. Metode probabilitas yang digunakan akan menggunakan implementasi model masukan dengan menggunakan hukum distribusi. Pada kasus ini, output akhir yang akan dihasilkan secara otomatis oleh model evakuasi. Beberapa skenario yang sama akan dihasilkan untuk menentukan konvergensi hasil. Terdapat beberapa metode yang digunakan dalam menggunakan hukum distribusi seperti analisa fungsi dan statistik *inferential*, model *markov* yang telah dimodifikasi serta model lainnya.

Hingga saat ini hanya ada beberapa model evakuasi yang memungkinkan menjalankan beberapa skenario yang sama secara bersamaan. Selain itu umumnya model ini tidak mempelajari bagaimana ketidakpastian perilaku pada penggunaan distribusi. Dengan begitu ketidakpastian perilaku ini masih jarang digunakan pada model evakuasi. Hal ini dikarenakan akan menambah sumber daya komputasi yang dilakukan saat simulasi. Dengan adanya ketidakpastian perilaku dalam simulasi akan menambah kompleksitas dalam suatu simulasi. Dengan adanya kompleksitas ini membuat tingginya komputasi yang terjadi. Oleh karena itu metode ini perlu dipertimbangkan sesuai dengan sumber daya yang akan digunakan dalam proses simulasi evakuasi. Semakin tingginya tingkat kompleksitas dalam suatu simulasi evakuasi dengan menambah beberapa parameter membuat simulasi terlihat lebih nyata namun akan membutuhkan sumber daya komputasi yang tinggi pula.

2.3.4 Integrasi dengan model lainnya

Mengingat perlunya simulasi evakuasi pada skala yang lebih besar seperti pada kebakaran hutan dan pengembangan pada BIM(*building information modelling*) dalam semua proses desain bangunan, maka diperlukan model simulasi evakuasi yang sesuai untuk pengembangan lebih lanjut. Kedepannya model evakuasi perlu didesain dengan memperhatikan aspek keselamatan dengan skala simulasi evakuasi yang lebih besar pula. Untuk itu, diperlukan penambahan kemampuan simulasi evakuasi dengan cara mengintegrasikan antar model evakuasi. Saat ini model simulasi evakuasi menggunakan teknik keselamatan kebakaran termasuk integrasi pada alat-alat teknik keselamatan evakuasi atau dengan adanya model lalu-lintas kerumunan. Disarankan pada masa yang akan datang akan ada perluasan integrasi pada model dengan ada-

nya alat-alat lainnya seperti dengan pemodelan struktur, pemodelan cahaya dan lainnya. Dengan begitu integrasi ini akan menambah keakuratan suatu proses evakuasi serta menambah model keselamatan dalam pembuatan suatu bangunan dalam hal evakuasi.

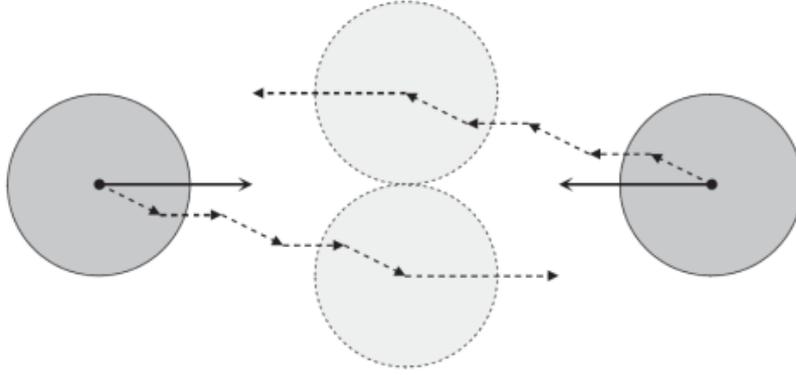
2.3.5 Metode Validasi

Saat ini terdapat beberapa metode yang tersedia dalam menganalisa model evakuasi mana yang memiliki kecocokan dengan data lapangan. Para peneliti menyadari masih kurangnya data-data mengenai perilaku manusia sehingga perlu diusahakan dalam menyelesaikan permasalahan tersebut. Sebaliknya kurangnya homogenitas dalam metode yang diadopsi untuk mengevaluasi hasil model simulasi evakuasi dalam penelitian validasi juga belum diteliti lebih mendalam. Berbagai metode serta pengujian model simulasi evakuasi masih memiliki perbedaan rutinitas baik analisa waktu simulasi evakuasi secara keseluruhan, analisa waktu simulasi waktu kedatangan, analisa rute perjalanan evakuasi atau pemilihan rute, analisa antrian kerumunan, jarak perjalanan, kepadatan serta hubungannya dengan aliran kerumunan, kecepatan dan penelitian tentang pola yang muncul pada kerumunan serta penelitian lainnya. Metode kini yang tersedia sering berbeda sesuai dengan besar variasi dalam prosedur yang diadopsi untuk pengujian. Selain itu, setiap metode dapat memberikan kelebihan dan kekurangan.

2.4 *Reciprocal Velocity Obstacles*

Reciprocal Velocity Obstacles(RVO) adalah suatu konsep pada *local reactive collision avoidance* yang digunakan yang digunakan pada navigasi *multi-agent* dimana setiap *agent* tidak memiliki komunikasi secara eksplisit (Van Berg et al., 2008). RVO merupakan perbaikan konsep dari *Velocity Obstacle* yang masih terdapat osilasi pada setiap gerakan *agent* saat terjadi halang rintang di sekitarnya. Pada Gambar 2.8 diilustrasikan adanya osilasi pada saat kedua *agent* yang akan bertabrakan saling menghindar menggunakan *velocity obstacle*. Pada Gambar 2.8 dua *agent* A dan B yang memiliki kecepatan \mathbf{v}_A dan \mathbf{v}_B . Apabila kedua *agent* tetap melanjutkan kecepatan \mathbf{v}_A dan \mathbf{v}_B maka akan terjadi tabrakan antara keduanya. Untuk menghindari *agent* yang berada di depannya maka setiap *agent* A dan B akan mengubah kecepatan \mathbf{v}_A dan \mathbf{v}_B diluar *velocity obstacle*-nya. Masalah muncul saat kedua *agent* telah berada diluar *velocity obstacle*, maka *agent* akan kembali ke \mathbf{v}_A dan \mathbf{v}_B awal karena mencari titik tujuan terpendek. dan akhirnya *agent* akan kembali kelu-

ar dari *velocity obstacle* dan akan begitu seterusnya. Hal inilah yang membuat terjadinya osilasi pada gerakan *agent* tersebut.



Gambar 2.8: Osilasi yang terjadi pada *collision avoidance* menggunakan *velocity obstacles* (Van Berg et al., 2008)

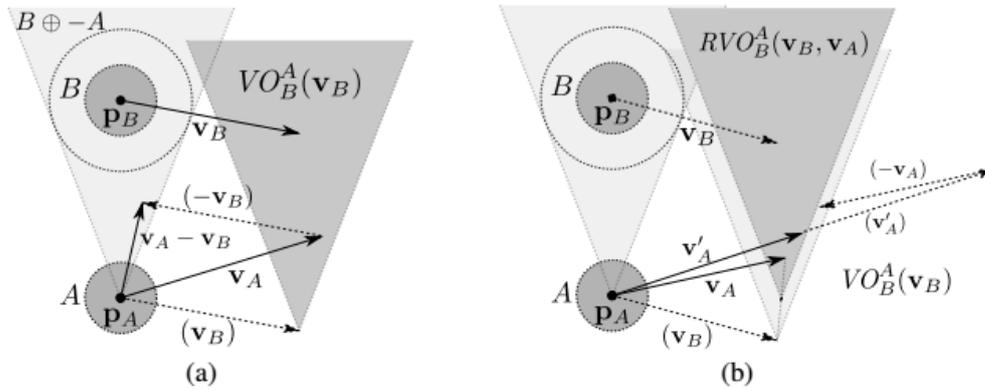
Untuk menghindari gerak osilasi inilah, Jur van der Berg (Van Berg et al., 2008), dkk merumuskan metode baru yang dikembangkan dari *Velocity Obstacle* yaitu *Reciprocal Velocity Obstacle* (RVO). Kecepatan (*velocity*) baru yang dipilih untuk menghindari tabrakan merupakan rerata dari kecepatan *agent* sekarang dan kecepatan yang berada diluar *Velocity Obstacle agent* lain. RVO didefinisikan sebagai:

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v}'_A | \mathbf{v}'_A - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\} \quad (2.1)$$

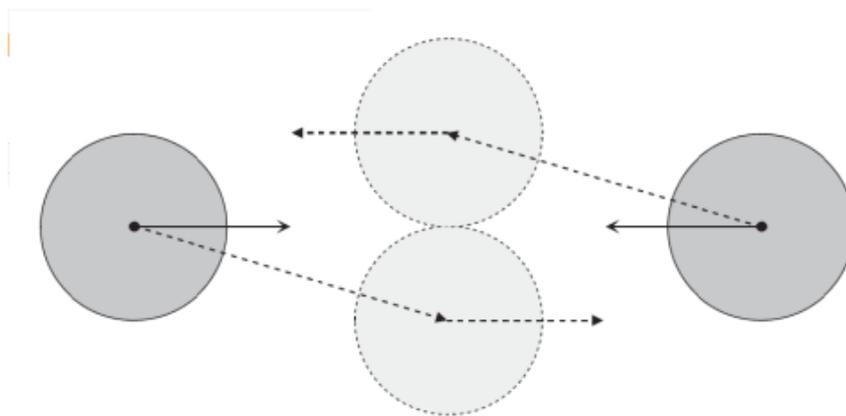
Berbeda dengan *velocity obstacle* yang berfikir bahwa *agent* lainnya tidak akan merubah kecepatannya, pada metode RVO setiap *agent* akan berfikir bahwa *agent* yang berada di dekatnya juga akan berganti kecepatan. Dengan begitu RVO akan membuat *agent* hanya mengambil setengah dari kecepatan penghindaran tabrakan dan berasumsi *agent* lainnya juga akan mengambil bagian setengah dengan mengganti kecepatan dalam menghindari tabrakan. Dengan begitu setiap *agent* nantinya akan terhindar dari tabrakan dan osilasi antar *agent* lainnya.

Reciprocal Velocity Obstacle $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$ dari *agent* B ke *agent* A berisi semua kecepatan *agent* A yang merupakan rerata kecepatan \mathbf{v}_A dan kecepatan yang berada didalam *Velocity Obstacle* $VO_B^A(\mathbf{v}_B)$ *agent* B. Kemudian RVO_B^A secara geometris dapat diinterpretasikan sebagai *Velocity Obstacle* VO_B^A

yang ditranslasikan supaya nilai puncaknya berada di $\frac{\mathbf{v}_A + \mathbf{v}_B}{2}$. Proses ini dapat dilihat pada Gambar 2.9. Gerakan penghindaran yang dihasilkan dapat dilihat pada Gambar 2.10.



Gambar 2.9: Penghindaran tabrakan oleh *velocity obstacle* (kiri) dan penghindaran tabrakan oleh RVO (kanan) (Van Berg et al., 2008)



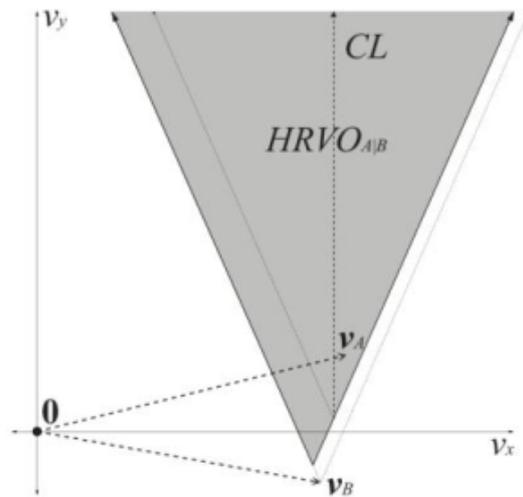
Gambar 2.10: Gerak saat *agent* menghindari tabrakan dengan *agent* lain menggunakan RVO (Van Berg et al., 2008)

Jika setiap *agent* memilih kecepatan baru yang terdekat dengan kecepatannya saat itu maka setiap *agent* akan secara otomatis melewati *agent* lainnya di sisi yang sama. Alih-alih memilih kecepatan terdekat dengan kecepatannya saat itu, agar dapat cepat sampai tujuan *agent* memilih kecepatan baru berdasarkan preferensi kecepatan oleh *agent* yang biasanya dengan arah terdekat dengan posisi tujuan dari *agent* tersebut. Dengan keadaan *agent* yang memilih kecepatan preferensi tersebut dibandingkan dengan kecepatan yang mendekati

dengan kecepatan sebelumnya dapat membuat *agent* terjebak dalam kondisi osilasi yang disebut dengan *reciprocal dances*. Kondisi *reciprocal dances* ini juga dapat ditambah dengan bertambahnya *agent* saat terjadinya penghindaran tabrakan. *Reciprocal dances* dapat mengakibatkan *agent* mengalami jalan buntu sehingga tidak dapat bergerak secara sempurna.

2.4.1 Hybrid Reciprocal Velocity Obstacle

Setelah adanya permasalahan *reciprocal dances* yang mengakibatkan *agent* dalam kondisi *deadlock* maka Snape dkk (Snape et al., 2011) membuat metode baru dengan menggabungkan antara *velocity obstacle*(VO) dan *reciprocal velocity obstacle*(RVO). Metode ini bertujuan agar setiap *agent* dapat menghindari *agent* lainnya dengan sisi yang berbeda satu sama lain. Cara kerja dari *hybrid reciprocal velocity obstacle*(HRVO) adalah jika kecepatan suatu *agent* berada disebelah kanan dari pusat RVO dengan *agent* lainnya maka *agent* tersebut akan bergerak ke kanan begitu pula sebaliknya. Dengan begitu kecepatan *agent* tidak lagi ditentukan dari seberapa dekat posisi tujuan akhir demi menghindari adanya *reciprocal dances*. Untuk dapat mendorong hal tersebut maka RVO akan menambah hambatan kecepatan bagi sisi yang tidak seharusnya dilewati oleh *agent*. Interpretasi dari geometri *hybrid reciprocal velocity obstacle* $HRVO_{A|B}$ pada *agent* A yang ingin menghindari *agent* B termasuk lokasi dari pusat HRVO dapat dilihat pada Gambar 2.11.



Gambar 2.11: *hybrid reciprocal velocity obstacle* $HRVO_{A|B}$ pada *agent* A yang ingin menghindari *agent* B (Snape et al., 2011)

Jika *agent* tersebut ingin berada di sisi kanan maka sisi sebelah kiri diberikan tambahan hambatan agar kecepatan *agent* tidak berpindah ke sisi kiri. Penambahan hambatan kecepatan ini menggunakan proses *mirroring* dengan kecepatan yang akan dituju. Dengan begitu apabila suatu *agent* akan mengambil arah kecepatan ke arah kanan maka arah kecepatan di kiri ditambah hambatan kecepatannya.

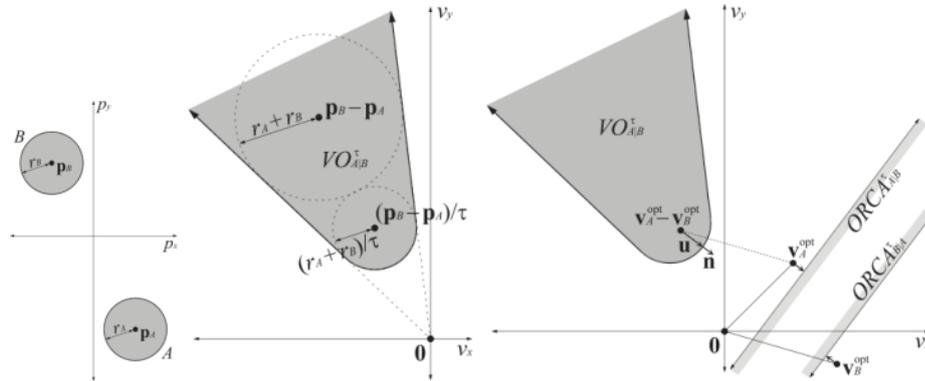
2.4.2 *Optimal Reciprocal Collision Avoidance*

Optimal reciprocal collision avoidance (ORCA) menyelesaikan permasalahan *reciprocal dances* dengan cara yang berbeda dengan HRVO. Metode ini dikembangkan oleh Van den Berg Dkk (Berg et al., 2011) dengan menambahkan suatu bidang *half-plane*. Bidang *half-plane* dapat memisahkan kecepatan mana saja yang dapat dilalui oleh setiap *agent* maupun yang tidak dapat dilewati. Tujuan dari metode ini adalah dapat membuat setiap *agent* yang akan berada pada kerumunan akan bebas dari tabrakan antar sesama *agent* dengan simulasi yang tidak terdapat osilasi walaupun dengan kerumunan yang padat.

Metode *half-plane* dari *optimal reciprocal collision avoidance* $ORCA_{A|B}$ pada *agent* A terhadap *agent* B didefinisikan sebagai berikut. Sesuai dengan Gambar 2.12, u menjadi vektor dari kecepatan relatif $\mathbf{v}_A - \mathbf{v}_b$ dari *agent* A dan *agent* B menuju titik terdekat pada batas kecepatan rintangan yang terpotong bagi *agent* A yang terinduksi oleh *agent* B . Sedangkan n menjadi batas luar dari kecepatan rintangan $\mathbf{v}_A - \mathbf{v}_b + \mathbf{u}$. Dengan begitu u menjadi faktor perubah terkecil kecepatan relatif *agent* untuk menghindari tabrakan. Untuk itu, setiap *agent* setidaknya menambahkan kecepatan paling sedikit $\frac{1}{2}\mathbf{u}$ untuk menghindari adanya tabrakan. Karena itu kecepatan yang diijinkan oleh ORCA pada *half-plane* ke arah n dimulai pada kecepatan $\mathbf{v}_A + \frac{1}{2}\mathbf{u}$ seperti yang terlihat pada Gambar 2.12.

Pada HRVO dan ORCA memilih kecepatan baru dengan cara menghitung kecepatan terdekat dengan kecepatan preferensi dan bebas dari rintangan. Jika titik tujuan yang akan dicapai terlihat oleh *agent* maka kecepatan preferensi adalah arah titik tujuan tersebut. Namun jika titik tujuan belum terlihat oleh *agent* maka kecepatan preferensi akan sesuai dengan *node* terdekat dengan jalur yang telah dipilih sesuai navigasi yang dilakukan.

ORCA menggunakan algoritma *linear programming* dengan efisiensi acak yang menambahkan pembatas satu persatu dalam urutan acak sambil terus



Gambar 2.12: Dua *agent* *A* dan *agent* *B*(kiri). Potongan kecepatan rintangan *VO* untuk *agent* *A* yang terinduksi oleh *agent* *B*(tengah). *Half-plane* pada *optimal reciprocal collision avoidance* dengan kecepatan *ORCA* yang diizinkan untuk *agent* *A* dan *agent* *B*(kanan)(Berg et al., 2011)

melacak kecepatan baru yang optimal untuk *agent* dalam batasan grup. *Linear programming* adalah suatu teknik optimasi yang biasanya digunakan untuk riset operasi untuk menemukan satu solusi untuk satu set persamaan linear dan batasan ketidaksetaraan yang mengoptimalkan fungsi linear variabel yang diberikan. Secara geometri, algoritma *linear programming* akan menghitung titik dalam poligon dimana fungsi dengan nilai maksimum dan nilai minimum jika titik tersebut ada. Algoritma *linear programming* yang bersifat acak akan menambahkan batas linear dalam urutan acak untuk menghitung solusi optimal. Untuk setiap *agent* yang berada dalam kerumunan, algoritma *linear programming* memiliki perkiraan waktu berjalan linear sehubungan dengan jumlah *agent* yang dimasukkan ke dalam algoritma. Algoritma ini menghitung kecepatan yang terdekat dengan kecepatan preferensi *agent* tersebut dan akan melaporkan kegagalan apabila terdapat *linear program* yang tidak layak.

Efisiensi dan skalabilitas dalam perhitungan kecepatan baru untuk *agent* dapat ditingkatkan dengan tidak memasukan pembatas semua *agent* tetapi hanya *agent* terdekat saja yang dimasukkan. *Agent* yang jauh memiliki kemungkinan kecil untuk berinteraksi dengan *agent* tersebut dalam interval waktu terdekat sehingga secara substansial tabrakan tidak mungkin terjadi. *library* *ORCA* menggunakan *kd-tree* untuk menghitung tetangga yang berupa *agent* terdekat. *Kd-tree* ini terus diperbaharui di setiap perjalanan *agent* sehingga mengetahui mana saja *agent-agent* lain yang dekat dengan *agent* tersebut.

Pada keadaan kerumunan padat, dimana setiap kerapatan antar *agent*

sangat tinggi memungkinkan tidak dapat ditemukannya kecepatan baru yang sesuai dengan batas pada *linear program* dan algoritma *linear programming* akan mengembalikan kecepatan *linear program* menjadi tidak layak. Apabila hal itu terjadi, maka ORCA akan menghitung kemungkinan terbaik serta teraman dari gangguan tabrakan pada setiap *agent*, dengan kecepatan dengan penghindaran minimal terhadap *agent* lainnya. Secara geometris seperti mengurakan tepi pada bidang *half-plane* tegak lurus diluar dengan kecepatan yang sama. Kecepatan ini dapat dihitung dengan *linear program* dengan dimensi tiga. Hampir sama dengan dimensi dua, penggunaan dimensi tiga hanya menambah satu dimensi untuk mencari solusi kecepatan baru yang akan digunakan oleh *agent*. Waktu yang digunakan pada algoritma ini masih sesuai dengan jumlah *agent*.

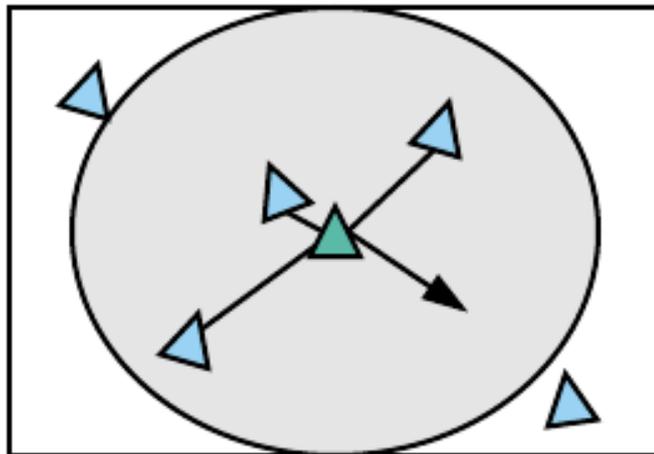
2.5 Agent-Based Model

Pada suatu simulasi kerumunan *Agent-based model*, setiap *agent* memiliki kecerdasan individu yang nantinya akan menentukan pergerakan di masing-masing *agent*. Salah satu dari contoh *agent-based model* ini adalah flocking behavior yang diinisiasi oleh Reynolds(Reynolds, 1987). *Flocking behavior* merupakan suatu sebuah tingkah laku dalam menirukan suatu kerumunan yang ditiru dari kerumunan hewan seperti kerumunan burung terbang, kerumunan ikan saat berenang, kerumunan serangga ataupun kerumunan hewan lainnya. Menurut Reynolds(Reynolds, 1987)], untuk mensimulasikan *flocking behavior* ini diperlukan beberapa hal yaitu *collision avoidance*, *flock centering* dan *velocity matching*.

Selain dari tiga dasar *flocking* terdapat pula beberapa perilaku yang dapat menjadi dasar dari *agent-based model*. Beberapa perilaku tersebut seperti *flee*(melarikan diri). Pada perilaku *flee* nantinya setiap *agent* akan diberikan tambahan gaya yang menjadi kecepatan untuk menghindari target. Berkebalikan dari *flee*, suatu *agent* dapat mengalami perilaku *arrival*(menghampiri). Pada perilaku *arrival* suatu *agent* nantinya akan dikurangi kecepatannya saat mendekati target. Selain itu ada pula beberapa contoh perilaku pada *agent-based model* ini seperti *wander*(berkelana), *pursuit*(mengejar), *evade*(menjauh), *queue*(mengantri), *path following*(mengikuti jalur) dan *leader following* (mengikuti *agent* lain).

2.5.1 *Collision Avoidance*

Menjaga jarak dengan anggota *flocking* yang berada di dekatnya. Perilaku ini dapat disebut pula dengan *separation*. Pada *separation* setiap anggota *flocking* dijaga agar tidak dapat menabrak anggota *flocking* lainnya. Dengan begitu setiap anggota *flocking* dalam suatu kerumunan dibuat agar memiliki jarak dengan anggota *flocking* lainnya. Pada Gambar 2.13 diberikan ilustrasi dari perilaku *separation*. Perilaku *separation* ini apabila diterapkan pada kerumunan *agent* NPC yang berbentuk kumpulan manusia berfungsi agar setiap NPC dapat menjaga jarak dengan NPC lainnya sehingga dengan begitu antar NPC tidak bertabrakan.

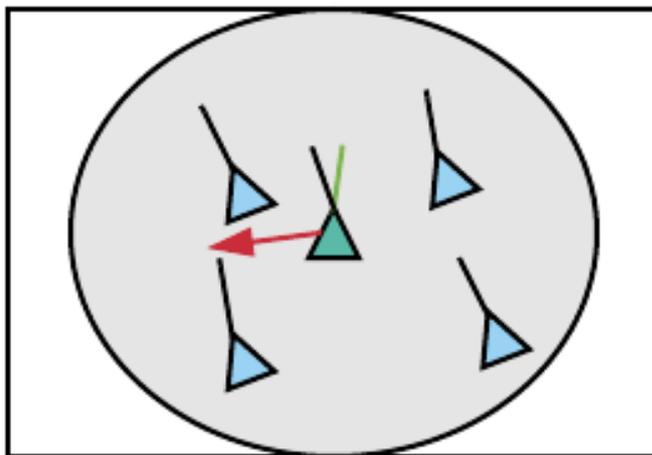


Gambar 2.13: *Separation*

2.5.2 *Velocity Matching*

Memiliki kesamaan kecepatan dengan anggota *flocking* lainnya. Pada *velocity matching* memperhatikan dua hal yaitu kecepatan dari setiap anggota *flocking* serta arah yang akan dituju oleh kumpulan anggota *flocking*. Pada arah setiap anggota *flocking* akan dibuat memiliki arah yang sama dan untuk kecepatan setiap anggota *flocking* dalam suatu kerumunan diperlukan menjaga kecepataannya sesuai dengan kecepatan dari kerumunan. Apabila suatu anggota *flocking* dirasa memiliki kecepatan melebihi kecepatan lainnya maka anggota *flocking* itu akan memperlambat kecepataannya dan apabila suatu

anggota *flocking* memiliki kecepatan yang lebih lambat dari anggota *flocking* lainnya maka diharapkan anggota *flocking* itu mempercepat kecepataannya agar menyesuaikan kondisi kecepatan dari kerumunan tersebut. *Velocity matching* ini saat ini dapat disebut pula dengan *alignment*. Penggunaan *alignment* pada NPC biasanya digunakan agar setiap NPC memiliki arah dan kecepatan yang sama sehingga kumpulan dari NPC dapat menuju ke suatu target tanpa adanya NPC yang berpisah karena tidak memiliki arah serta menghindari adanya NPC yang berjalan terlalu cepat dan NPC yang tertinggal karena memiliki kecepatan yang berbeda dengan NPC lainnya. Pada Gambar 2.14 akan diberikan ilustrasi dari pergerakan *alignment* atau *velocity matching* pada suatu kerumunan.

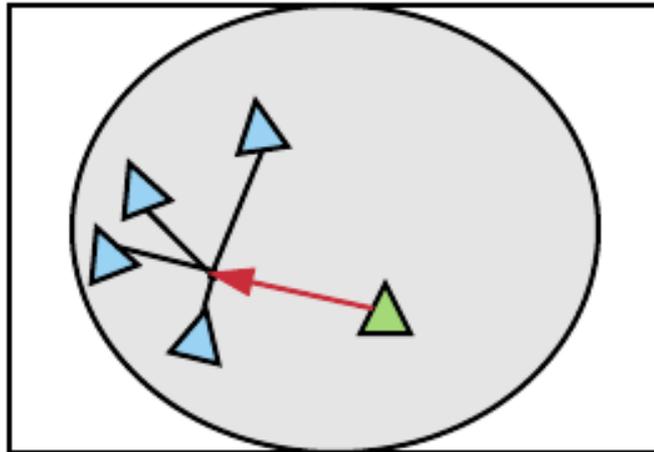


Gambar 2.14: *Alignment*

2.5.3 *Flock Centering*

Suatu perilaku untuk tetap berada dekat dengan anggota *flocking* lainnya. Perilaku *flock centering* ini digunakan agar setiap anggota *flocking* kerumunan tidak berjauhan dengan anggota *flocking* lainnya. Dengan adanya perilaku *flocking* ini maka setiap anggota *flocking* dapat tetap berada dikerumunannya. *flock centering* ini juga biasa disebut dengan *cohesion*. Untuk dapat membuat suatu anggota *flocking* berada mendekati suatu kerumunan maka dibuat agar setiap anggota *flocking* dapat bergerak ke suatu titik kepadatan tertentu. Pada

Gambar 2.15 diberikan ilustrasi dari perilaku *cohesion*. Penggunaan *cohesion* pada NPC membuat suatu kumpulan NPC dapat terus berjalan berkelompok dan membuat NPC yang terpisah dapat kembali pada kelompoknya.



Gambar 2.15: Cohesion

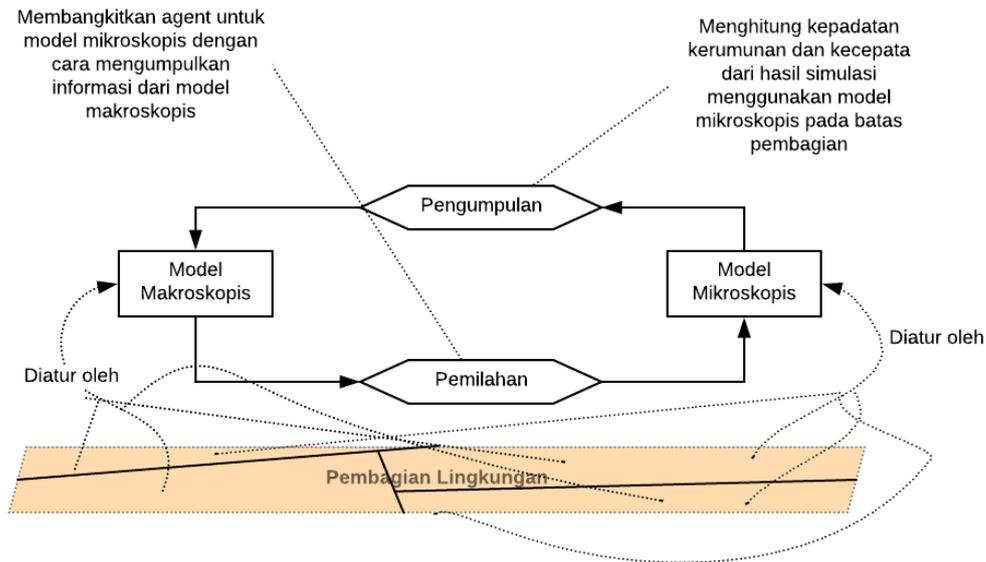
2.6 Crowd Simulation Hybrid Model

Model *hybrid* pada simulasi kerumunan merupakan suatu model yang menggabungkan dari berbagai model yang telah. Menurut survey yang dilakukan oleh Ijaz, terdapat tiga jenis model *hybrid* pada simulasi kerumunan (Ijaz et al., 2015). Ketiga jenis model dari simulasi kerumunan ini adalah:

2.6.1 Zone-Based Model

Pada model simulasi kerumunan ini, Setiap ruang simulasi akan dibagi menjadi beberapa zona. Penerapan *hybrid* model berdasarkan zona yang ada. Biasanya pada model ini zona simulasi yang mencakup *macrosimulation* berada pada zona lingkungan yang umum dan besar serta tidak terlalu membutuhkan pengambilan keputusan secara individu. Sedangkan pada zona simulasi yang mencakup *microsimulation* berada pada zona-zona yang membutuhkan keputusan individu seperti di persimpangan, pintu dan tempat-tempat lainnya. Pada penelitian yang dilakukan oleh Xiong dkk, terdapat pembagian zona antara model mikroskopis dan model makroskopis (Xiong et al. (2012). Ilustrasi

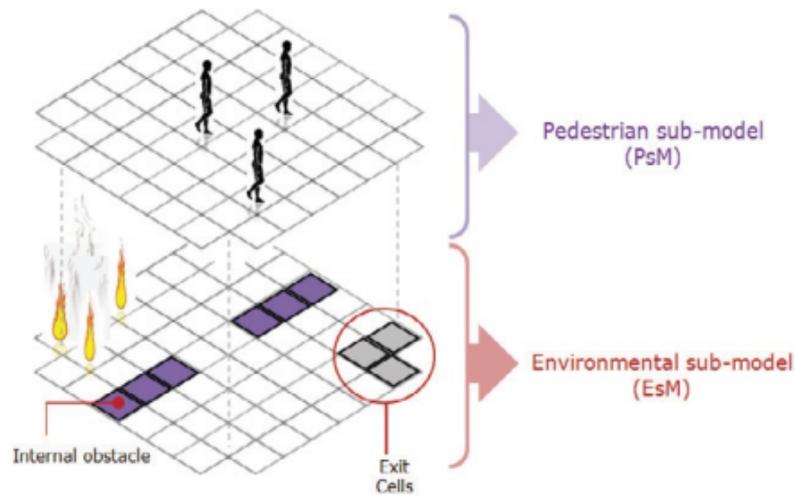
mengenai cara pembagian menurut zona dapat dilihat pada Gambar 2.16.



Gambar 2.16: Cara pembagian pendekatan simulasi kerumunan berdasarkan model berbasis zona (Xiong et al., 2012)

2.6.2 Layers-Based Model

Pada model *hybrid* ini pembagian simulasi kerumunan dibagi berdasarkan lapisan. Pada dasarnya pembagian lapisan ini terbagi menjadi dua. Pada lapisan pertama diberikan pada pergerakan secara keseluruhan diberikan model makroskopis atau *mesoscopic*. Hal ini agar komputasi yang terjadi tidak terlalu besar. Lapisan yang kedua digunakan pendekatan mikroskopis sebagai pergerakannya. Lapisan kedua merupakan pergerakan tiap individu yang dapat mengenali lingkungan sekitarnya seperti pedestrian lainnya. Pada penelitian Cristian model simulasi kerumunan dibagi menjadi dua lapisan yaitu lapisan pada lingkungan dan lapisan pada pedestrian (Cristian et al., 2012). Pada lapisan lingkungan setiap *agent* pedestrian menggunakan CA untuk dapat mengetahui kondisi lingkungan serta dapat melakukan menghindari halangan yang ada di lingkungan. Pada lapisan pedestrian, setiap *agent* dapat melakukan interaksi dengan lingkungan lainnya seperti menghindari *agent* lainnya. Pada Gambar 2.17 diberikan ilustrasi bagaimana pendekatan simulasi dibagi berdasarkan lapisan.



Gambar 2.17: Cara pembagian pendekatan simulasi berdasarkan model berbasis lapisan (*layer*) (Cristian et al., 2012)

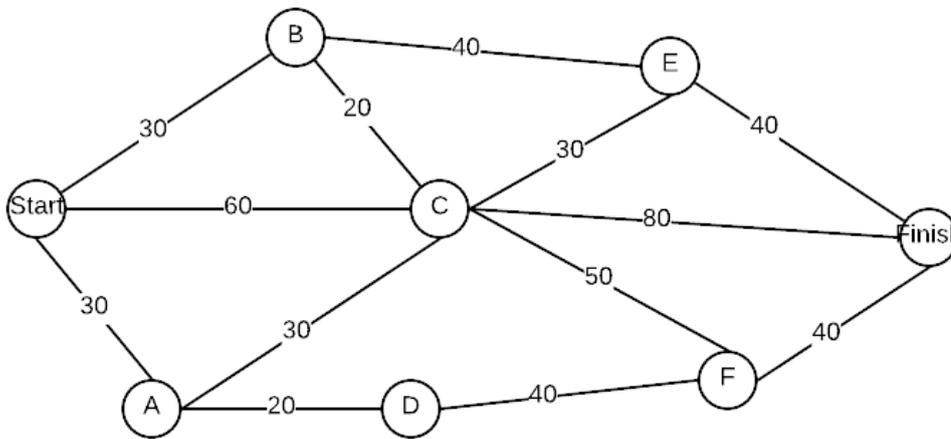
2.6.3 Sequential Models

Model sekuensial ini hampir mirip dengan model yang berbasis lapisan. Pada model ini juga menggunakan pendekatan makroskopis pada pergerakan keseluruhan kerumunan dan menggunakan pendekatan mikroskopis di kerumunan yang sama dalam pendekatan individu. Kedua pendekatan ini dijalankan secara berurutan. Pada model sekuensial ini terdapat modul sinkronisasi yang berperan untuk mengubah dari pendekatan makroskopis menjadi pendekatan mikroskopis dan sebaliknya. Pada awalnya pendekatan makroskopis berjalan berdasarkan hubungan kecepatan dan kepadatan untuk mensimulasikan pergerakan kerumunan. Setelah berganti ke pendekatan mikroskopis melalui modul sinkronisasi, pendekatan mikroskopis akan menjalankan eksekusi dari pola pergerakan berdasarkan pendekatan makroskopis sebelumnya. Pada Penelitian Xiong (Xiong et al., 2009) Pendekatan makroskopis digunakan saat kerumunan bersifat stabil dan Pendekatan mikroskopis digunakan apabila terdapat sesuatu yang mengganggu dalam kerumunan saat pendekatan makroskopis.

2.7 Algoritma A*

Dalam ilmu komputer algoritma A* (baca A-star) digunakan sebagai algoritma pencarian *path finding*. *Path finding* digunakan untuk menyelesaikan masalah pada sebuah *graph*. *Graph* merupakan himpunan titik atau yang biasa disebut dengan *node* yang terhubung oleh *edge*. *Edge* yang menghubungkan

setiap node merupakan suatu vektor yang memiliki arah dan besaran tertentu. Untuk dapat menemukan jalur dari *node* awal hingga *node* tujuan yang akan dicapai maka diperlukan penelusuran terhadap *graph* tersebut. Penelusuran dilakukan dengan mengikuti arah *edge* yang menghubungkan antar *node*. *Edge* yang menghubungkan antar node biasanya memiliki bobot-bobot yang nantinya akan mempengaruhi biaya perjalanan dalam suatu *graph*. Biaya perjalanan inilah yang nantinya akan dijadikan pertimbangan dalam pemilihan jalur. Pada Gambar 2.18 diberikan contoh *graph* dari titik awal start menuju titik tujuan finish dengan beberapa *node* yang dihubungkan oleh *edge* yang telah diberikan bobot.



Gambar 2.18: Contoh *graph*

terdapat beberapa jenis algoritma pencarian yang menggunakan *graph* (Patel, 2014), beberapa dari algoritma tersebut yaitu:

1. ***Breadth-First Search*** yaitu algoritma yang mengeksplorasi semua daerah secara sama. Algoritma ini termasuk algoritma yang sering digunakan tidak hanya pada pencarian jalur pada umumnya, namun juga dipakai pada prosedur pembuatan peta, pemetaan aliran bidang, peta jarak serta beberapa analisa peta lainnya.
2. **Algoritma Dijkstra's** atau biasa juga disebut *uniform cost search* yang akan memprioritaskan jalur-jalur tertentu untuk dieksplor. Berbeda dengan algoritma *breadth-first search* yang mengeksplorasi semua daerah algoritma dijkstra's hanya mengeksplorasi tempat tertentu untuk menghemat biaya operasional pencarian. Pada algoritma ini akan menghindari

jalur-jalur yang memiliki biaya operasional pencarian yang tinggi. Biasanya untuk jalur-jalur yang memiliki biaya operasional yang beragam algoritma ini dipilih dibandingkan algoritma *breadth-first search*.

3. **A*** merupakan modifikasi dari algoritma dijkstra's yang akan mengoptimasi pada satu tujuan saja. Jika algoritma dijkstra's dapat mencari semua jalur, A* hanya mencari satu jalur saja. Hal ini memprioritaskan ke tujuan yang akan dicapai.

Algoritma A* diperkenalkan oleh Peter Hart dkk (Hart and Nils, 1968). Pada algoritma A* suatu tempat yang akan dijadikan jalur akan dibagi menjadi kisi-kisi yang akan dijadikan titik-titik yang akan dihubungkan oleh menjadi sebuah jalur yang disebut *node*. Algoritma ini akan mencari jalur tercepat yang akan ditempuh oleh suatu *agent*. Algoritma A* adalah algoritma *best-first search* yang berarti mengeksplorasi suatu tempat dengan cara memperluas serta memilih *node* yang paling menjanjikan dengan aturan-aturan tertentu dalam hal ini mencari jalur dengan biaya terkecil atau jarak terpendek. Setiap jalur yang akan melewati kisi-kisi akan dihitung dan diberikan bobot sehingga bobot-bobot tersebut nanti dapat dijumlahkan dan akan dapat dicari jalur dengan bobot terendah. Untuk itu dibuat bobot *graph* untuk mencari bobot terendah agar dapat dicari jalur terdekat. Dimulai dari *graph* tertentu setelahnya dibuat pohon jalur dari jalur awal hingga jalur terakhir yang dilalui. Setelahnya dilakukan penelusuran setiap jalur dengan waktu diskrit hingga ditemukan jalur tujuan. Jalur dengan bobot *graph* terendah menjadi jalur acuan. Hal ini berbeda dengan *breadth-first search* yang menelusuri semua *grid* sehingga algoritma *best-first search* akan memiliki biaya yang lebih rendah serta waktu tempuh yang lebih cepat.

Penggunaan *best-first search* serta *breadth-first search* biasanya tergantung informasi yang terdapat pada sebuah peta yang biasanya dalam bentuk *grid* atau kisi-kisi yang dapat berbentuk peta. Apabila peta tersebut tidak memiliki informasi(*uninformed*) maka diperlukan adanya *breadth-first search* karena *agent* yang tidak memiliki informasi apapun dalam mencapai tujuan. Namun apabila terdapat informasi(*heuristic*) maka *best-first search* dapat digunakan karena terdapat aturan-aturan serta informasi dalam melakukan perjalanan hingga tujuan. Dengan begitu informasi inilah yang membuat waktu perjalanan serta biaya operasional perjalanan lebih efektif dibandingkan dengan algoritma *breadth-first search*.

Dimana n adalah *node* terakhir pada jalur, $g(n)$ adalah biaya perpindahan dari jalur dengan *node* pertama hingga *node* ke- n , dan $h(n)$ adalah *heuristic* yang mengestimasi biaya perpindahan dari jalur termurah dari n menuju tujuan. Estimasi *heuristic* dari jarak menuju goal haruslah dapat diterima yang artinya tidak melebih-lebihkan biaya sebenarnya untuk sampai pada *node* tujuan terdekat. Implementasi dari algoritma A* menggunakan antrian prioritas untuk melakukan pemilihan berulang dari estimasi minimal biaya perpindahan *node* yang akan dijelajah. Antrian prioritas ini dikenal juga dengan sebutan *open set* atau *fringe*. Setiap langkah dalam algoritma, *node* dengan nilai $f(x)$ terkecil akan dihapus dari antrian. Nilai f dan g dari tetangganya akan diperbaharui terus dan akan dimasukkan kedalam antrian. Algoritma ini akan terus berlanjut hingga *node* tujuan memiliki nilai f lebih rendah dibanding nilai *node* lainnya di dalam antrian atau saat antrian kosong. Nilai f dari tujuan adalah yang memiliki panjang jarak terpendek karena h pada tujuan adalah nol didalam *heuristic* yang diterima.

Sejauh ini yang dijelaskan dari algoritma A* adalah memberikan panjang jalur terpendek. Untuk menemukan urutan langkah yang sebenarnya, maka setiap algoritma dapat mudah direvisi sehingga setiap *node* di suatu jalur akan melacak pendahulunya. Setelah algoritma dijalankan, *node* terakhir akan menunjuk ke pendahulunya dan seterusnya hingga pendahulunya adalah *node* awal. Sebagai contoh, ketika melakukan pencarian rute terdekat pada suatu peta, $h(x)$ akan merepresentasikan jarak dalam bentuk garis lurus untuk mencapai tujuan. karena jarak tersebutlah yang memungkinkan untuk mencapai rute terpendek diantara dua titik. Jika *heuristic* h memenuhi kondisi tambahan $h(x) \leq d(x, y) + h(y)$ untuk setiap $edge(x, y)$ dari *graph* (dimana d menunjukkan panjang dari *edge* tersebut), maka h disebut monoton atau konsisten. Dalam kasus seperti itu, A* dapat diimplementasikan secara lebih efisien. Secara kasarnya tidak ada *node* yang perlu diproses lebih dari sekali dan A* setara dengan operasi algoritma Dijkstra dengan biaya perpindahan yang dikurangi $d'(x, y) = d(x, y) + h(y) - h(x)$.

Halaman ini sengaja dikosongkan

BAB 3

METODOLOGI PENELITIAN

Pada penelitian ini nantinya akan terdiri dari tiga langkah-langkah yaitu menyiapkan lingkungan simulasi kerumunan serta karakteristik dari *agent* berdasarkan analisa data dari buku *data collection and anlysis of evacuation situations*(Rinne et al., 2010). Setelahnya akan disiapkan navigasi global dengan menggunakan algoritma A* yang telah terdapat pada *engine* Unity 3d, Menyiapkan navigasi lokal yang berpusat pada penghindaran tabrakan dengan *optimal reciprocal collision avoidance* yang merupakan metode lanjutan dari *reciprocal velocity obstacle*, desain *agent* heterogen menggunakan *agent-based model* dengan skenario yang didapatkan dari karakteristik *agent* pada langkah pertama seperti yang terlihat pada Gambar 3.1 . Pada penelitian ini memiliki tujuan umum yaitu membuat simulasi kerumunan untuk evakuasi bencana pada *agent* yang heterogen. Berikut adalah penjelasan dari setiap langkah-langkah yang nantinya akan digunakan.



Gambar 3.1: Metodologi Penelitian

3.1 Membangkitkan Lingkungan dan Karakteristik *Agent*

Pada tahapan ini akan dibuat *environment* atau lingkungan tempat simulasi kerumunan dilakukan. Simulasi kerumunan pada penelitian ini berupa simulasi evakuasi pada gedung tertutup dengan begitu lingkungan yang akan di-*generate* berupa gedung atau tempat tertutup. Pada penelitian ini lingkungan yang digunakan untuk simulasi adalah gedung pusat robotika ITS. Gedung pusat robotika ITS nantinya akan dibentuk model tiga dimensi dengan skala disesuaikan dengan skala pada simulasi. Dengan begitu skala yang dibuat pada gedung robotika dan orang yang sesungguhnya akan sesuai. Selain itu akan dipersiapkan karakteristik dari *agent* sesuai dengan koleksi data dan analisa pada kejadian yang sesungguhnya sehingga dapat dibuat skenario

pada *agent*. Koleksi data ini berupa dimensi fisik *agent* yang akan disesuaikan dengan bentuk tubuh anatomi orang Indonesia serta beberapa perilaku yang dilakukan oleh orang-orang saat menghadapi kepanikan. Perilaku dari orang-orang inilah nantinya akan dijadikan patokan skenario pergerakan kerumunan pada simulasi.

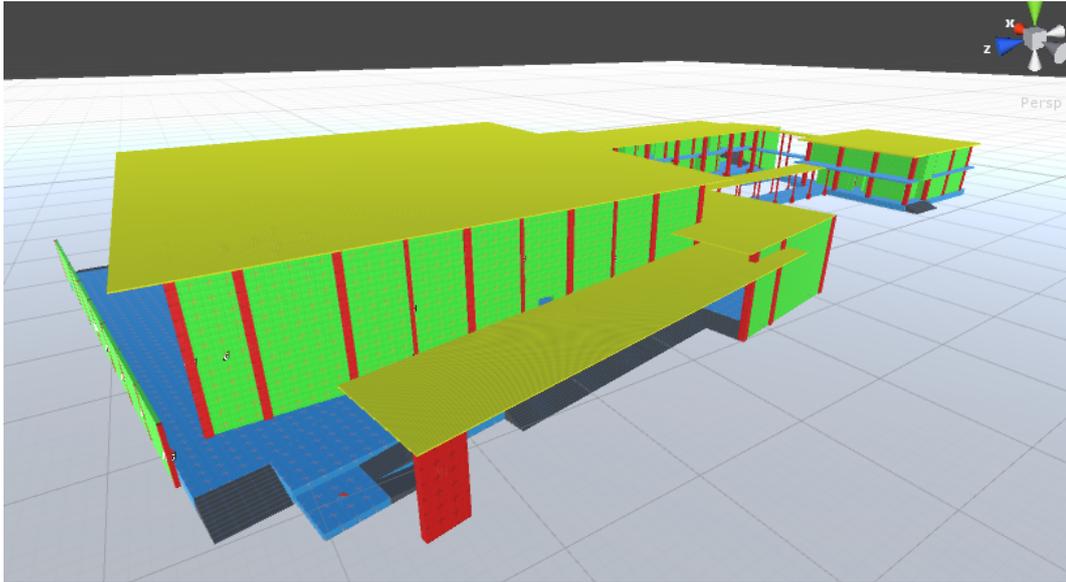
3.1.1 Lingkungan Simulasi

Proses simulasi evakuasi akan dilakukan pada model tiga dimensi gedung robot ITS dengan skala 1 meter per 1 satuan unit di Unity 3d. Foto gedung robotika ITS dapat dilihat pada Gambar 3.2. Ilustrasi lingkungan dapat dilihat pada Gambar 3.3 dan Gambar 3.4. Gedung robot ITS memiliki tiga bagian yaitu gedung acara, gedung laboratorium dan gedung bengkel(rob, 2018). Gedung acara digunakan untuk diselenggarakannya acara-acara seperti kontes robot dan acara lainnya seperti acara seminar, tes masuk, pameran serta acara-acara lainnya yang membutuhkan ruangan yang besar. Selain digunakan pihak ITS, gedung acara juga dapat disewakan kepada pihak luar apabila terdapat acara-acara yang dirasa membutuhkan tempat yang besar. Dengan adanya tribun yang terdapat pada lantai atas pada gedung ini membuat gedung menjadi gedung yang serbaguna. Pada acara kontes robot ataupun acara-acara perlombaan lainnya, tribun dapat digunakan sebagai tempat duduk penonton untuk menonton perlombaan yang diselenggarakan.

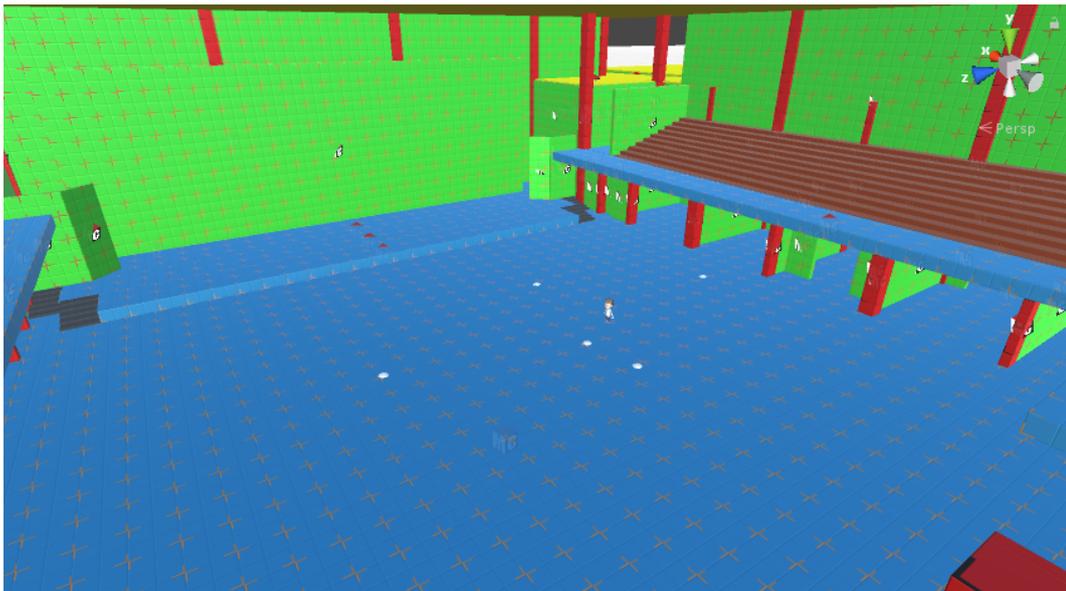


Gambar 3.2: Gedung robotika ITS(rob, 2018)

Selain itu terdapat pula gedung bengkel yang digunakan oleh tim robotika ITS sebagai bengkel robotika dan gedung laboratotium yang terdiri dari laboratorium dasar mekatronika dan otomasi industri, laboratorium sensor dan



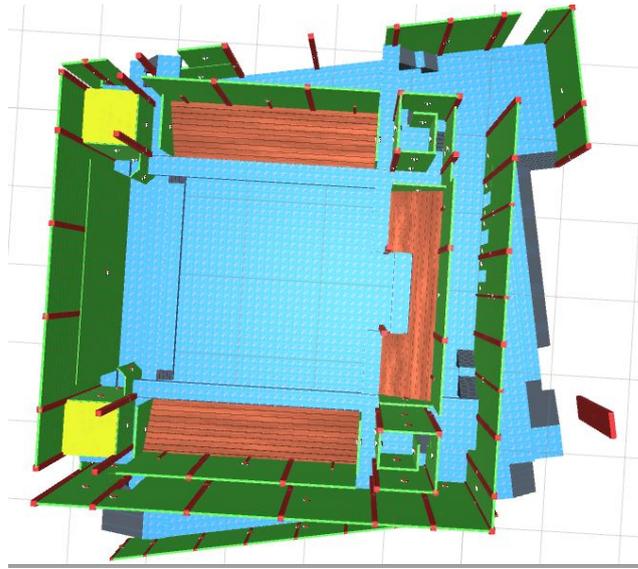
Gambar 3.3: Lingkungan gedung robot tampak luar



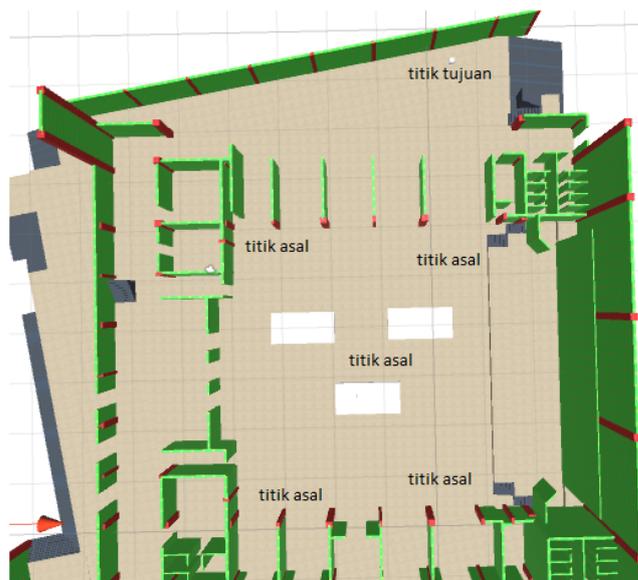
Gambar 3.4: Lingkungan gedung robot tampak dalam

laboratorium aplikasi manufaktur. Laboratorium digunakan oleh pihak ITS sebagai tempat riset beberapa penelitian yang menjadi penelitian unggulan dari ITS. Pada penelitian ini hanya akan menggunakan gedung acara robotika sebagai lingkungan simulasi evakuasi. Hal ini dikarenakan gedung acara dapat memuat kerumunan dalam jumlah besar dan dapat diaplikasikan pada penelitian ini. Pada penelitian ini hanya akan menggunakan lantai satu dengan tambahan rintangan yang berada di tengah-tengah ruangan lantai satu. Keadaan dalam gedung acara robotika pada bagian atas dapat dilihat pada

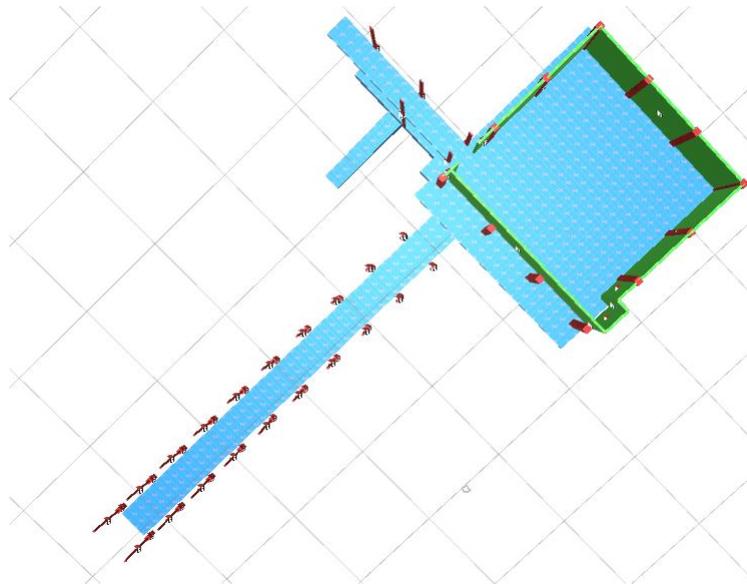
Pada Gambar 3.5. Pada gedung robotika terdapat beberapa pintu namun pada penelitian ini akan menggunakan satu pintu belakang sebagai tempat tujuan evakuasi seperti yang terlihat pada Gambar 3.6. Keadaan gedung bengkel dan laboratorium dari gedung pusat robotika dapat dilihat pada Gambar 3.7 dan Gambar 3.8.



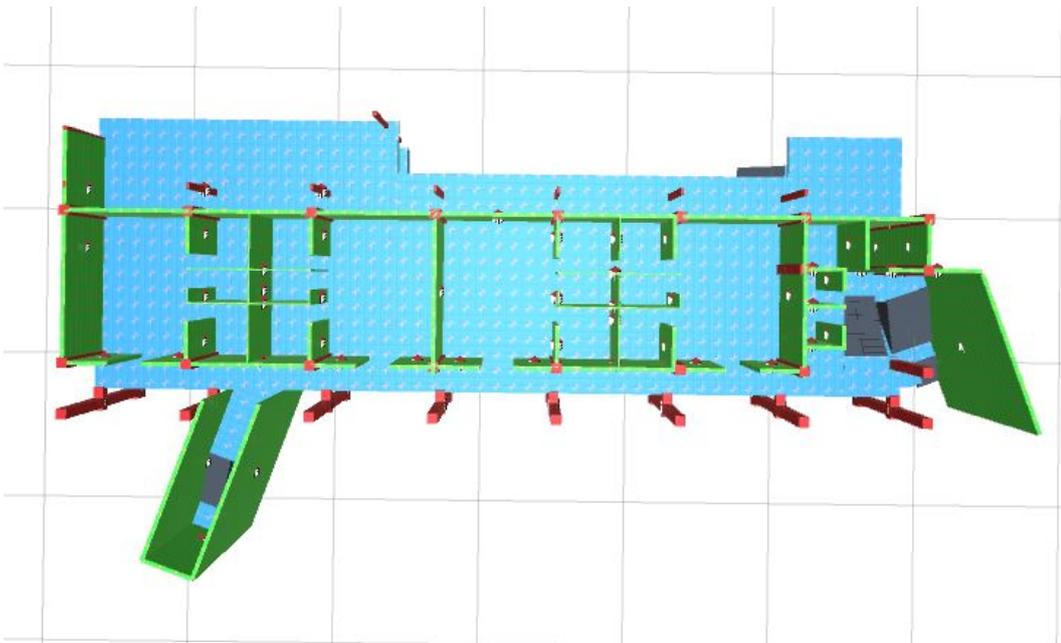
Gambar 3.5: Lingkungan gedung acara robot tampak dalam dilihat dari atas



Gambar 3.6: Peta saat simulasi dilakukan



Gambar 3.7: Lingkungan gedung robot bengkel tampak dalam dilihat dari atas



Gambar 3.8: Lingkungan gedung robot laboratorium tampak dalam dilihat dari atas

3.1.2 Karakteristik *Agent* Berdasarkan Koleksi dan Analisa Data

Untuk dapat membuat simulasi evakuasi semirip mungkin dengan kejadian sebenarnya maka diperlukan membuat kondisi *agent* semirip mungkin dengan kondisi lapangan pada saat simulasi. Pada penelitian akan mengambil koleksi data dari buku *data collection and analysis of evacuation situation* (Rinne

et al., 2010) pada skenario *agent* serta jurnal yang berjudul *evacuation time and movement in apartment building** (Proulx, 1995) dan *technical note* yang berjudul *overall and local movement speeds during fire drill evacuations in buildings up to 31 stories* (Peacock et al., 2010). Informasi data koleksi data akan dijadikan dasar skenario *agent* serta dasar beberapa informasi seperti kecepatan bergerak *agent*. Bentuk *agent* pada simulasi ini berbentuk kapsul pada *asset* dasar *engine* Unity 3d dengan tinggi serta lebar badan yang akan disesuaikan skala dengan rata-rata tinggi badan orang indonesia yaitu 169 cm untuk laki-laki dan 155 cm untuk perempuan dan lebar bahu 43 cm untuk laki-laki dan 38 cm untuk perempuan dan kedalaman dada (*chest depth*) sebesar 21 cm pada laki-laki dan 22 cm pada perempuan (Chuan et al., 2010). Tinggi badan dan lebar bahu ini akan disesuaikan dengan tinggi serta lebar kapsul sesuai dengan skala pada *engine* unity 3d yaitu skala satu unit satu meter.

Pada data yang didapatkan pada buku *data collection and analysis of evacuation situation* terdapat beberapa perbedaan kecepatan saat evakuasi diantara beberapa orang. Nilai tengah pada saat situasi normal orang-orang yang sedang melakukan evakuasi pada bidang yang horizontal adalah 1.3 m/s untuk orang dewasa, 1.5 m/s untuk anak-anak dan 2.1 m/s untuk beberapa orang yang telah menemukan titik tujuannya (*goal-oriented*). Kecepatan ini akan berpengaruh terhadap waktu simulasi secara keseluruhan selain itu proses evakuasi akan dipengaruhi pula dengan jumlah orang yang berada di ruangan. Contohnya pada ruangan kelas yang diisi guru dan murid dengan jumlah 10 hingga 25 orang akan membutuhkan waktu evakuasi selama 13 detik hingga 98 detik. Kecepatan evakuasi juga nantinya akan dipengaruhi oleh bentuk gedung pada saat evakuasi termasuk jumlah lantai gedung tersebut. Jumlah lantai dapat pula mempengaruhi kecepatan orang-orang saat evakuasi karena pergerakan orang-orang saat menelusuri bidang horizontal dan vertikal berbeda.

Sebelum terjadinya aksi evakuasi terdapat waktu yang disebut dengan pra-evakuasi yang nantinya akan menentukan arahan saat terjadinya evakuasi. Aksi-aksi pada saat pra evakuasi ini yaitu reaksi terhadap alarm, aksi pertama saat mendengar alarm, tindakan dalam suatu kelompok serta aksi-aksi dari pegawai yang telah mengenal gedung dengan baik. Reaksi terhadap alarm kebakaran tergantung dari jenis bangunan. Di sekolah terdapat alarm-alarm yang biasanya dibunyikan saat waktu-waktu tertentu, begitu pula pada pusat perbelanjaan yang menjadikan alarm sebagai pengumuman yang bersi-

fat komersial. Dengan begitu alarm pada proses evakuasi tidak dapat berjalan sesuai yang diharapkan. Adanya pengumuman yang terjadi dua kali juga dapat membuat orang-orang menjadi kebingungan sehingga hal ini tidak terlalu efektif. Namun pada tempat yang memiliki geometri yang kompleks seperti kantor ataupun toko-toko kehadiran setidaknya dapat membuat orang-orang mengambil langkah dalam evakuasi dibandingkan dengan pemberitahuan para pegawai yang harus melewati semua tempat yang kompleks tersebut. Namun secara umum pemberitahuan manual masih lebih dibandingkan pemberitahuan otomatis, namun terdapat beberapa hal yang harus diperhatikan seperti pesan yang singkat serta penting dan pengulangan pesan yang lambat.

Setelah mendengarkan alarm terdapat beberapa reaksi dari orang-orang saat terjadinya pra-evakuasi. Pada beberapa kasus orang-orang tetap melakukan aktifitasnya bahkan setelah diperintahkan untuk mengungsi. Pada kasus yang terjadi di supermarket saat orang-orang belum mengira terjadinya bencana orang-orang masih melakukan kegiatan berbelanja dan akhirnya berdampak terjadinya kemacetan pada meja-meja kasir. Pada kasus yang terjadi di sekolah pergerakan untuk proses evakuasi baru terjadi saat guru serta pegawai sekolah memerintahkan orang-orang untuk melakukan proses evakuasi dengan menyuruh meninggalkan gedung sekolah. Dengan begitu adanya komunikasi antar orang dapat mempengaruhi langkah-langkah pertama saat terjadinya evakuasi.

Perilaku berkelompok terjadi bila terdapat pegawai, keamanan, atau orang-orang yang dapat memberikan saran serta memberikan perintah terhadap suatu kelompok kecil untuk melakukan evakuasi. Dengan adanya orang-orang yang dapat memberi perintah ini dapat membuat suatu perilaku kelompok yang menjadi suatu kerumunan yang dapat membantu proses evakuasi dalam hal ini dapat berupa penunjuk jalan serta pemberi arahan. Adanya perilaku kelompok ini dapat efektif pada kelompok-kelompok dengan skala yang kecil. Pada skala yang lebih besar dibutuhkan lebih dari satu orang untuk mengarahkan suatu kelompok. Dengan adanya perilaku kelompok ini dapat menarik individu-individu sekitar untuk masuk kedalam kelompok ini. Misalnya pada evakuasi terdapat sekelompok wanita yang berjumlah tiga orang yang memiliki rute yang mereka yakini lebih baik maka kelompok ini dapat bertambah seiring dengan adanya individu-individu lain yang mengikuti kelompok tersebut. Pada suatu kerumunan yang lebih besar perilaku ini dapat dikenali saat terdapat reaksi berantai pada orang-orang misalnya saja dalam memilih pintu

yang sama saat proses evakuasi.

Peran serta status seseorang dapat berpengaruh saat terjadinya evakuasi. Seorang pegawai yang telah mendapatkan pelatihan evakuasi sebelumnya dapat membantu proses evakuasi yang terjadi. Dengan begitu pegawai-pegawai ini memiliki pekerjaan lain saat situasi yang tidak terduga seperti saat terjadinya evakuasi dalam gedung di tempat mereka bekerja. Namun dengan adanya sistem *shift* serta keadaan yang tidak terduga membuat para pegawai ini terkadang bingung saat menelusuri proses evakuasi dengan efektif. Pada kasus di bioskop dengan jumlah pegawai yang sedikit setiap pegawai diberikan tanggung jawab tempat-tempat mana saja yang menjadi tanggung jawab mereka saat terjadinya evakuasi saat pelatihan evakuasi. Pengalaman suatu pegawai juga dapat mempengaruhi proses perilaku pegawai nantinya saat terjadi evakuasi. Pada kasus-kasus yang terdapat di tempat pekerja-pekerja organisasi keselamatan di perusahaan tersebut dapat mempengaruhi respon-respon orang-orang saat terjadinya evakuasi. Keuntungan yang didapat dengan adanya pegawai yang memandu sekelompok orang ini ada dua yaitu pertama petugas pemadam dapat memperoleh informasi apakah masih terdapat orang-orang yang terjebak dalam gedung atau tidak. Yang kedua para pegawai ini tentunya lebih paham jalur evakuasi di dalam gedung sehingga memudahkan proses evakuasi.

Fase Evakuasi dimulahi saat orang-orang menerima alarm dan memutuskan untuk melakukan evakuasi keluar gedung. Waktu evakuasi, atau fase-fase perpindahan tergantung pada kecepatan berjalan seseorang serta kepadatan kerumunan saat proses evakuasi dan juga jarak yang ditempuh saat evakuasi terjadi. Parameter lainnya yaitu pemilihan rute serta pemilihan pintu yang akan ditempuh saat terjadinya evakuasi. Pemilihan rute tergantung pada seberapa mengenalnya orang-orang terhadap gedung saat terjadinya evakuasi, jarak penglihatan tanda-tanda evakuasi, aspek mayoritas-minoritas (seperti adanya perilaku kelompok yang mengikuti pemandu) ancaman kebakaran dan lain-lain. Pemilihan pintu merupakan faktor terakhir yang akan mempengaruhi proses evakuasi yang terjadi.

Pada umumnya orang-orang akan memilih pintu utama sebagai jalan keluar evakuasi. Selain itu pada kasus kebanyakan orang-orang cenderung memilih tempat mereka masuk saat memutuskan untuk keluar gedung. Selain itu orang-orang dapat mencari tempat keluar lainnya apabila terdapat orang-orang yang membimbing kerumunan tersebut mencari tempat keluar yang lain-

nya. Hal ini terjadi jika terdapat pegawai-pegawai yang terdapat pada gedung yang akhirnya menngarahkan kerumunan untuk melalui rute evakuasi yang telah ditunjukkan. Sebagai contoh saat pegawai yang berprofesi sebagai petugas keselamatan dapat mengarahkan kerumunan untuk melewati jalur evakuasi yang telah ditentukan. Selain itu orang-orang dapat mencari rute lain apabila pintu utama tersebut tidak dapat dilewati saat terjadinya proses evakuasi. Dalam beberapa kasus rute evakuasi ditentukan saat adanya pengarah singkat sebelum proses evakuasi terjadi sehingga terbentuk kerumunan melewati jalur evakuasi yang telah ditentukan.

Pemilihan dan penggunaan pintu merupakan salah satu faktor penting dalam suatu proses evakuasi. Orang-orang cenderung memilih pintu-pintu yang telah dilewati orang-orang sebelumnya dengan alasan merupakan jalan keluar yang sudah dapat dilewati. Dengan pemilihan pintu yang sama akhirnya menambah kepadatan yang terjadi disekitar pintu tersebut. Bahkan pada banyak kasus pintu tambahan yang dibuka hanya diikuti oleh sebagian kecil orang sehingga tidak mengurangi kepadatan kerumunan pintu sebelumnya. Bahkan pada pintu ganda pun biasanya hanya dibuka sebagian oleh orang-orang saat terjadinya evakuasi. Orang-orang yang terjebak dalam evakuasi juga lebih menghindari pemilihan pintu-pintu yang terkunci baik itu pada simulasi evakuasi maupun pada kehidupan nyata. Orang-orang lebih cenderung memilih rute-rute dengan pintu yang biasa mereka lewati. Bahkan pada saat latihan beberapa orang tidak mengetahui apakah boleh untuk melakukan pengrusakan pada pintu yang dikunci.

Faktor-faktor seperti pengaturan fisik pada jalan keluar alternatif, hambatan, kapasitas jalan, jarak penglihatan serta komposisi kerumunan menjadi faktor-faktor yang akan mempengaruhi seseorang dalam mencari jalur alternatif. Dengan begitu pencarian jalur-jalur alternatif ini memerlukan banyak faktor sehingga terlihat wajar bagaimana saat pintu-pintu alternatif yang dapat digunakan tetapi tidak digunakan oleh para pengungsi.

3.2 Navigasi Global dan Pergerakan Lokal

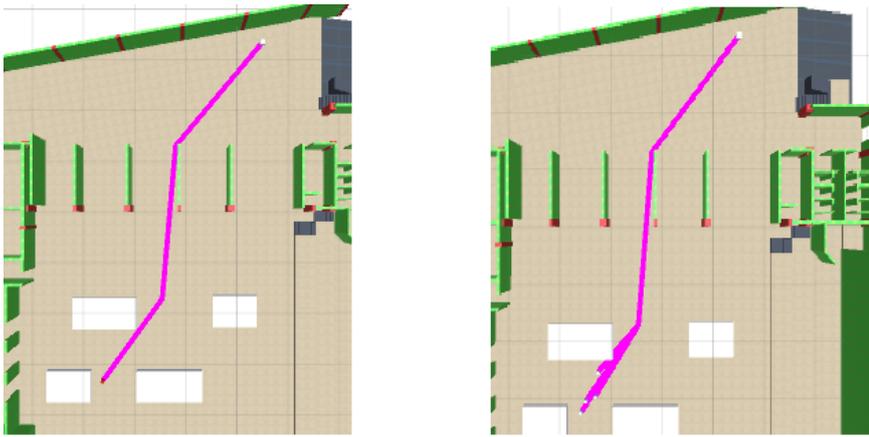
Navigasi global dilakukan untuk mencapai titik tujuan simulasi. Dengan adanya navigasi global ini *agent* dapat melakukan perpindahan dari titik awal *agent* menuju titik tujuan. Bentuk navigasi global pada *agent* ini adalah proses *seeking* pada *agent*. Pada tahapan penerapan navigasi global setiap *agent* hanya dapat menghindari adanya rintangan yang berbentuk statik dan tidak

dipersiapkan untuk menghindari adanya rintangan yang berbentuk dinamis. Untuk itu digunakanlah metode RVO sebagai metode yang membuat setiap *agent* dapat menghindari rintangan yang bersifat dinamis. RVO ini nantinya juga akan bertindak sebagai navigasi lokal sehingga kecepatan *agent* yang sebelumnya telah ditetapkan saat pencarian posisi akhir dengan navigasi global akan berubah dengan adanya navigasi lokal ini. Pada penerapan RVO sebagai navigasi lokal terdapat dua proses yang terbentuk yaitu proses *steering* dan proses *collision avoidance*. Proses *steering* ini adalah proses menentukan kecepatan preferensi yang akan dipilih oleh *agent* sedangkan proses *collision avoidance* akan digunakan agar *agent* dapat menghindari tabrakan pada rintangan baik itu rintangan statis dan rintangan dinamis.

3.2.1 *Seeking*

Pada bagian ini setiap *agent* akan melakukan proses *seeking* untuk mencari titik tujuan *agent*. Karena pada penelitian ini nantinya akan terdapat proses relasi antar *agent* maka setiap *agent* akan memiliki titik tujuan yang berbeda-beda. Pada *agent* yang bertindak sebagai *follower* maka titik acuan dari *agent* tersebut adalah *leader*, sedangkan pada *leader* akan diberikan titik acuan suatu tempat yang membuat *agent* dapat melakukan evakuasi dalam gedung. Posisi *leader* pada simulasi ini adalah sebagai penunjuk arah pada *agent* yang tidak memiliki pengetahuan jalur evakuasi yang harus dilewati. Dengan begitu dengan adanya *agent leader* dapat menuntun *agent* yang bersifat *follower* melewati jalur evakuasi yang telah ditetapkan. Selain itu juga terdapat *agent* yang bersifat independen yang melakukan proses *seeking* seperti pada *agent* yang bersifat *leader* yaitu langsung ke tempat tujuan. Proses *seeking* pada *agent* yang bersifat *leader* dan independen dapat dilihat pada Gambar 3.9. Proses pencarian titik tujuan dapat dilihat pada Tabel 3.1.

Pada *agent* yang bertindak sebagai *follower* proses *seeking* terhadap *agent* yang bersifat *leader* terdapat batas yang disebut dengan LoS(*Line of Sight*). Batas ini merupakan batas jarak pencarian *agent* yang bersifat *follower* terhadap *agent* yang bersifat *leader* seperti yang terlihat pada Gambar 3.10. Dengan begitu jika *agent* yang bersifat *follower* tidak dapat melihat *agent* yang bersifat *leader* dalam *line of sight* maka posisi *agent* tersebut dalam keadaan tersesat sehingga tidak dapat melakukan proses evakuasi. Proses pencarian titik tujuan pada *leader* dan *follower* akan dibahas lebih lanjut pada bab 3.3. Pada penelitian ini penggunaan navigasi global digunakan agar setiap *agent*



Gambar 3.9: *Seeking* pada *agent leader*(kiri), *seeking* pada *agent independen*(kanan)

Tabel 3.1: Tabel Proses *seeking* pada setiap *agent*

no	Agent	Titik Tujuan <i>Seeking</i>
1	<i>Leader</i>	Titik akhir evakuasi
2	<i>Independen</i>	Titik akhir evakuasi
3	<i>Follower</i>	Agent lain (<i>leader</i> / <i>independen</i>)

dapat melakukan proses *seeking* posisi yang akan diinginkan. Proses *seeking* ini akan menggunakan algoritma A* (baca A-star) sebagai *path finding*. Pada *engine* Unity terdapat fitur *navmesh* yang merupakan fitur *path finding* yang menggunakan algoritma A*.



Gambar 3.10: *Seeking* pada *agent follower* terhadap *agent independen*(kiri) , *seeking* pada *agent follower* terhadap *agent leader*(kanan), lingkaran merah adalah wilayah *line of sight*

3.2.2 Steering

Proses *seeking* yang telah dilakukan sebelumnya akan berpengaruh dalam proses *steering* yang akan dilakukan selanjutnya. Pada proses ini, setiap agent akan memiliki kecepatan maksimal v_{max} dan kecepatan preferensi. Kecepatan preferensi ditentukan oleh navigasi lokal yang dilakukan oleh RVO namun juga dipengaruhi dari navigasi global yang telah ditentukan. Navigasi global membuat proses kecepatan preferensi yang dihasilkan sebisa mungkin mendekati navigasi global. Pada penelitian ini akan terdapat tiga macam *agent* yaitu *agent* yang bersifat independen, *agent* yang bersifat sebagai pemimpin atau *leader* serta *agent* yang bersifat *follower* yang akan mengikuti pemimpin. Dengan adanya tiga jenis *agent* dalam simulasi maka setiap *agent* akan memiliki kecepatan preferensi yang berbeda-beda pula. Proses *steering* pada *agent* yang bersifat independen adalah proses perpindahan dari titik mula hingga ke titik tujuan. Proses *steering* pada *agent* independen tidak terpengaruh oleh *agent* lainnya karena *agent* independen tidak melakukan kooperasi dengan *agent* lainnya. Pengaruh *agent* lain pada *agent* independen hanya pada proses *collision avoidance*.

Pada *agent* yang bersifat dependen seperti pada *agent* yang berperan sebagai *leader* dan *agent* yang berperan sebagai *follower* maka keberadaan *agent* lain akan mempengaruhi proses *steering* dari *agent* tersebut. Pada *agent* yang bertindak sebagai *leader*, proses *steering* dilakukan dengan cara berpindah pada titik semula menuju titik-titik tujuan evakuasi. Namun dalam perjalanannya kecepatan maksimal pada *agent* yang bersifat *leader* akan berpengaruh terhadap adanya *agent* yang bersifat *follower* di sekitar *line of sight* *agent* yang bersifat *leader* tersebut. Bila *agent* yang bersifat *follower* tersebut semakin jauh dari *leader* maka *agent* yang bersifat *leader* akan memperlambat kecepatannya. Perlambatan kecepatan pada *leader* sesuai dengan persamaan 3.1.

$$V_{leader} = \frac{V_{initialL}}{\exp\left(\frac{S_{agent}-S_x}{s_{LoSL}}\right)} \quad (3.1)$$

V_{leader} = Kecepatan *leader*

$V_{initialL}$ = Kecepatan awal *leader*

S_{agent} = Posisi antara *leader* dan *follower*

S_x = Jarak yang diinginkan antara *leader* dan *follower*

S_{LoSL} = Radius *Line of Sight* dari *agent leader*

Proses *steering* pada *agent* yang bersifat *follower* berbeda dengan *agent* lainnya. Dikarenakan *agent follower* tidak dapat mencari titik tujuan maka *agent follower* akan mengikuti *agent* lainnya. *agent* ini dapat berupa *agent leader* atau *agent* independen. FSM dari pergerakan dapat dilihat pada sub bab Penerapan *Agent Heterogen* dengan *Agent-Based Model*. Dengan begitu preferensi kecepatan dari *agent follower* juga tergantung dengan *agent* lainnya. Kecepatan dari *agent follower* akan tergantung dari jarak dengan *agent* yang akan dituju baik itu *agent leader* maupun *agent* independen. Semakin jauh maka *agent follower* akan menambah kecepatannya agar dapat menyusul *agent* yang menjadi target. Penambahan kecepatan ini sesuai dengan perilaku manusia yang cenderung menambahkan kecepatannya saat melihat target yang berada di depannya. Dengan begitu *agent* yang bersifat *follower* akan cenderung memiliki kelajuan yang lebih besar dibandingkan dengan *agent* yang bersifat *follower*.

$$V_{follower} = V_{initialF} \times \exp\left(\frac{S_{agent} - S_x}{S_{LoSF}}\right) \quad (3.2)$$

$V_{follower}$ = Kecepatan *follower*

$V_{initialF}$ = Kecepatan awal *follower*

S_{agent} = Posisi antara *leader* dan *follower*

S_x = Jarak yang diinginkan antara *leader* dan *follower*

S_{LoSF} = Radius *Line of Sight* dari *agent follower*

3.2.3 *Collision Avoidance*

Proses dari *collision avoidance* memungkinkan *agent* tidak bertabrakan dengan objek lain seperti *agent* lain maupun halang rintang yang berupa tembok dan halang rintang lainnya. Terdapat dua jenis tabrakan yang akan dihindari oleh *agent* dalam simulasi ini yaitu halang rintang yang bersifat statis dan halang rintang yang bersifat dinamis. Pada halang rintang yang bersifat statis memiliki kecepatan $v = 0$ sedangkan halang rintang dinamis memiliki pergerakan. Pada metode *agent-based* memiliki kelemahan dalam menghindari rintangan dinamis. Saat *agent* bertemu dengan *agent* lainnya maka dalam metode *agent-based* akan menghindari *agent* tersebut. Untuk menghindari

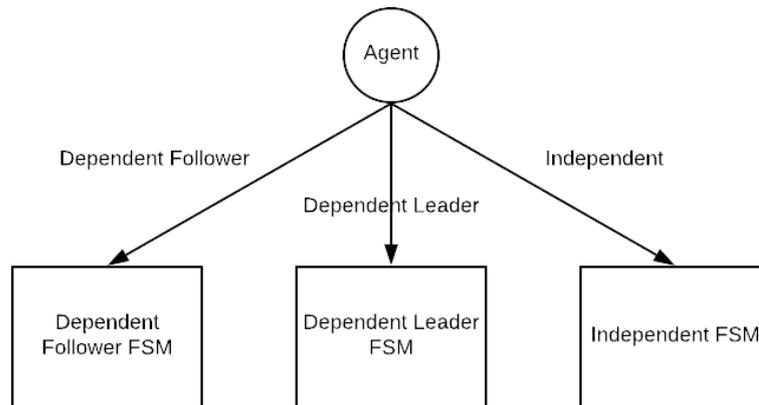
agent yang berada di dekatnya maka *agent* tersebut akan mencari kecepatan preferensi yang baru. Namun hal ini tidak hanya berlaku pada *agent* tersebut tetapi juga *agent* yang berada di dekatnya. Dengan begitu *agent* di dekatnya akan mengganti pula preferensi kecepatan. Hal ini membuat *agent* mengubah kembali preferensinya kembali kekecepatan semula. Hal inilah yang menyebabkan adanya osilasi pada pergerakan *agent* dalam simulasi. Untuk menghindari hal tersebut maka digunakanlah metode RVO untuk melakukan penghindaran halang rintang tersebut.

Pada penelitian ini metode RVO yang digunakan adalah metode terbaru yaitu *optimal n-body reciprocal collision avoidance*. Metode ini akan menyempurnakan metode RVO terutama mengatasi masalah *reciprocal dances* yang terjadi pada RVO. Dengan begitu *agent* tidak lagi bingung memilih sisi mana yang akan dilewati. Selain itu pada metode ini juga akan diberikan batasan jumlah *neighbour* yang berada disekitar *agent*. Pembatasan *neighbour* ini akan berpengaruh pada jumlah maksimal rintangan yang dapat dihindari. Pembatasan ini bertujuan agar komputasi yang dilakukan *agent* tidak begitu memberatkan proses simulasi evakuasi.

3.3 Penerapan *Agent* Heterogen dengan *Agent-Based Model*

Untuk dapat menerapkan kondisi simulasi kerumunan dengan *multi-agent* yang bersifat heterogen diperlukan pendekatan metode lainnya selain metode RVO. *Agent-based model* dinilai tepat karena setiap *agent* dalam sistem *multi-agent* diperhatikan secara individu bukan secara kerumunan. Dengan adanya perbedaan kecepatan serta keadaan *agent* yang heterogen maka setiap *agent* akan diberikan kondisi yang berbeda-beda. Dengan begitu setiap *agent* akan memiliki kondisi masing-masing sesuai dengan Gambar 3.11.

Agar simulasi evakuasi kerumunan ini dapat lebih realistis maka *agent* yang heterogen pada penelitian ini akan menerima perbedaan kemampuan yang dalam hal ini seperti waktu respon dan kecepatan bergerak dari *agent*. Perbedaan kemampuan ini dapat dibedakan melalui perbedaan usia, gender dan juga status setiap *agent*. Didalam kerumunan nantinya akan terdapat dua jenis *agent* yaitu *agent* yang independen dan *agent* yang *dependent*. *Agent* yang independen tidak memiliki relasi apapun dengan *agent* lainnya, sedangkan *agent* yang *dependent* memiliki relasi dengan *agent* lainnya. Relasi ini bisa berupa relasi antar saudara dan akhirnya dapat menjadi relasi *leader-follower*



Gambar 3.11: Pembagian kondisi berdasarkan jenis *agent*

yang terjadi antar *agent*. Relasi *leader-follower* ini dalam dunia nyata dapat terjadi pada orang tua dan anak yang melakukan evakuasi. Dengan begitu anak akan mengikuti pergerakan orang tuanya dan orang tua akan mencari anaknya saat terjadi evakuasi. Selain itu seperti pada artikel *data collection and analysis of evacuation situation* terdapat beberapa hal yang membuat *agent* mengikuti *agent* lainnya seperti karena adanya pegawai gedung yang mengarahkan kerumunan untuk mencari jalan keluar alternatif, adanya guru sekolah yang menuntun murid-muridnya untuk melakukan proses evakuasi, adanya petugas keselamatan (*safety*) yang menuntun proses evakuasi dalam suatu lingkungan kerja ataupun adanya gerombolan yang akhirnya diikuti individu lainnya karena memiliki alternatif jalan dalam proses evakuasi.

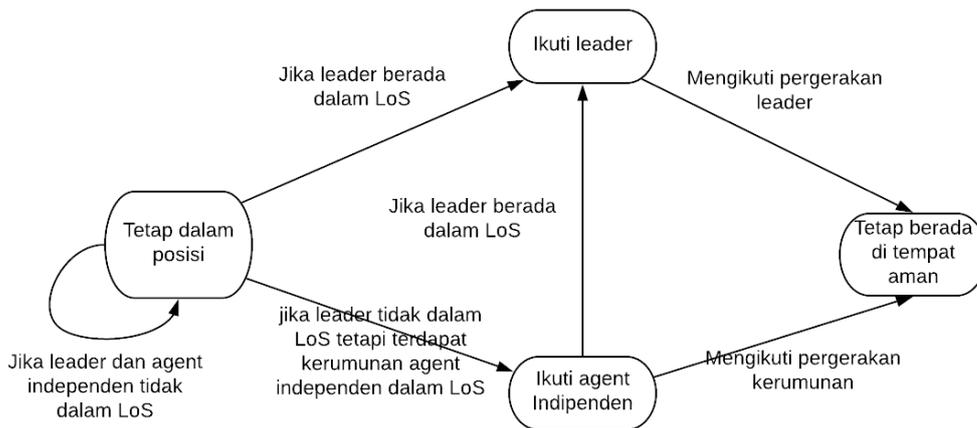
Pada penelitian ini *agent* yang *dependent* akan terdapat dua jenis *agent* yaitu *agent* yang menjadi *leader* dan *agent* yang menjadi *follower*. Karena bersifat *dependent* maka perilaku *agent* tersebut dapat berbeda dengan perilaku *agent* yang *independent*. Pada *agent* yang menjadi follower, maka *agent* mengikuti *leader* apabila berada di dekat *leader* berada dalam *line of sight* *agent* tersebut. Jika tidak terdapat *leader* di *line of sight* *agent* follower maka akan dilakukan pengecekan apakah ada kerumunan *agent* independen yang tertangkap dalam radius *line of sight* *agent* tersebut. Jika ada maka *agent* follower dapat mengikuti *agent* independen tersebut. Jika tidak ada *agent* leader maupun *agent* independen maka *agent* tersebut dinyatakan dalam keadaan tersesat.

Perbedaan antara perilaku *agent* leader dengan *agent* independen saat

diikuti oleh *agent follower* adalah *agent leader* akan menyesuaikan kecepatan sehingga *agent follower* tidak tertinggal sedangkan *agent independen* tidak melakukan hal tersebut. Pada tahapan terjadinya relasi *leader follower* ini pula akan terjadi *velocity matching* antara *agent* yang menjadi *leader* dan *follower*.

Pada penelitian ini akan diberikan tiga skenario *agent* heterogen sesuai dengan karakteristik *agent* berdasarkan koleksi dan analisa yang telah dilakukan sebelumnya. Berikut adalah skenario yang terjadi akan diterapkan:

1. **Skenario heterogen tipe A.** Pada skenario ini dalam suatu kerumunan terdapat tiga jenis *agent* yaitu *agent* yang bersifat independen, *agent* yang bersifat *leader* dan *agent* yang bersifat *follower*. *Agent* yang bersifat *leader* dan independen akan langsung menuju tempat titik akhir evakuasi sedangkan *agent* yang bersifat *follower* akan mengikuti *leader* atau independen. Namun untuk dapat mengikuti *agent leader* dan independen maka *agent* yang akan diikuti harus terlihat dalam *line of sight* *agent* yang bersifat *follower* tersebut. Pada skenario ini *agent* yang bersifat *follower* akan memprioritaskan mengikuti *agent leader* dibandingkan *agent* yang bersifat independen. FSM *agent* yang bersifat *follower* dapat dilihat pada Gambar 3.12.



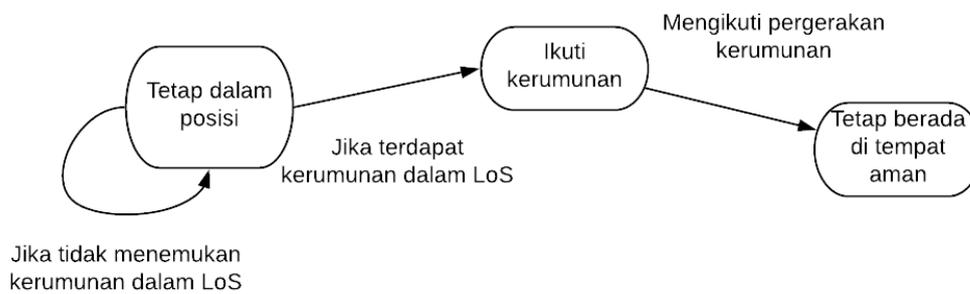
Gambar 3.12: FSM pada *agent follower* pada skenario heterogen tipe A

Pada *agent* yang bersifat *leader* disini juga akan memperhatikan kehadiran *agent* yang bersifat *follower*. Untuk itu *agent* yang bersifat *leader* akan memiliki *line of sight* yang dapat mendeteksi kehadiran *agent*

yang bersifat *follower*. Bila terdapat *agent* yang bersifat *follower* di daerah *line of sight agent leader*, maka *agent* tersebut akan menyesuaikan dengan mengurangi kecepatannya. Dengan begitu *agent* yang bersifat *follower* tidak akan tertinggal jauh dari *agent* yang bersifat *leader*. Pembuatan skenario ini dikarenakan dalam suatu evakuasi biasanya terdapat beberapa orang yang dapat menjadi penunjuk arah keluar saat proses evakuasi. Dalam proses evakuasi *agent* yang bersifat *leader* biasanya adalah orang-orang yang sudah paham tempat terjadinya proses evakuasi dan memiliki tanggung jawab dalam proses evakuasi seperti para pegawai yang menuntun para pengunjung gedung, atau petugas keselamatan yang bertanggung jawab atas keselamatan para pekerja dalam suatu lingkungan pekerja. *Agent* yang bersifat *leader* ini juga dapat berupa guru atau orang tua yang menuntun anak-anak didiknya dalam proses evakuasi. Sedangkan peran pada *agent* yang bersifat independen terjadi pada orang-orang yang mengetahui letak gedung saat proses evakuasi namun tidak bertanggung jawab pada proses evakuasi. Dengan begitu *agent* yang bersifat independen ini tidak memiliki keterikatan dengan *agent* lainnya. *agent* yang bersifat *follower* diibaratkan orang-orang yang baru yang tidak mengetahui geometri gedung ataupun anak-anak yang butuh bantuan orang lain dalam proses evakuasi. Dengan begitu *agent* yang bersifat *follower* ini membutuhkan *agent* lain dalam proses evakuasinya.

- 2. Skenario heterogen tipe B.** Pada skenario heterogen tipe ini dalam proses evakuasi tidak terdapat *agent* yang bersifat *leader* seperti pada skenario heterogen tipe A. Dengan begitu hanya ada *agent* yang bersifat independen dan *agent* yang bersifat *follower* saja. *Agent* yang bersifat independen pada skenario memiliki navigasi global menuju titik akhir evakuasi. Dengan begitu *agent* yang bersifat independen pada skenario ini tau geometri gedung dan dapat melakukan evakuasi tanpa bantuan *agent* manapun. *Agent* yang bersifat *follower* akan mengikuti pergerakan kerumunan *agent* yang bersifat independen yang telah mengetahui arah titik akhir evakuasi. Dengan begitu *agent* yang bersifat *follower* ini membutuhkan *agent* lainnya dalam proses evakuasi karena tidak memiliki pengetahuan untuk melakukan proses pencarian titik akhir evakuasi. Sama seperti skenario sebelumnya, untuk dapat mengetahui letak *agent* yang bersifat independen serta mengikuti pergerakannya maka terdapat *line of sight* pada *agent* yang bersifat *follower*. Apabila terdapat *agent*

yang bersifat independen pada *line of sight*, maka *agent* yang bersifat *follower* tersebut akan mengikuti *agent* yang berada pada *line of sight* tersebut. Namun apabila tidak terdapat *agent* dalam *line of sight* maka *agent* yang bersifat *follower* akan dalam kondisi tersesat dan tidak dapat bergerak. *Agent* yang bersifat independen tidak akan memperhatikan kondisi *agent* lainnya disekitar tempat evakuasi. Dengan begitu *agent* yang bersifat independen tidak akan terpengaruh oleh kehadiran *agent* lainnya termasuk *agent* yang bersifat *follower* kecuali dalam hal penghindaran tabrakan. FSM dari *agent* yang bersifat *follower* dapat dilihat pada Gambar 3.13.

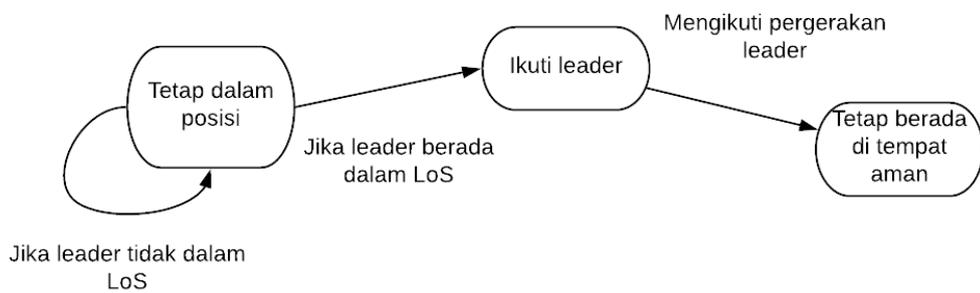


Gambar 3.13: FSM pada *agent follower* pada skenario heterogen tipe B

Pembuatan skenario ini berdasarkan terdapat kecenderungan orang yang mengikuti orang lainnya dalam proses evakuasi terlebih bila orang tersebut tidak mengetahui atau tidak terlalu mengenal tempat dia berada. Hal ini diibaratkan pada *agent* yang bersifat *follower* yang akan mengikuti pergerakan kerumunan *agent* independen yang sudah mengetahui titik akhir evakuasi. Selain itu pada proses evakuasi seseorang yang tidak memiliki tanggung jawab dalam suatu proses evakuasi cenderung tidak menghiraukan sekitarnya dan akan bertindak mencari tempat aman sesuai dengan pengetahuan yang telah dia miliki. Hal ini diibaratkan pada *agent* yang bersifat independen yang tidak mepedulikan lingkungan sekitar evakuasi termasuk *agent* lainnya.

3. **Skenario heterogen tipe C.** Pada skenario ini hampir sama dengan skenario heterogen tipe A yang terdiri dari *agent* yang bersifat *leader*, *agent* yang bersifat *follower* dan *agent* yang bersifat independen. Perbedaan mendasar skenario tipe C dengan skenario tipe A adalah *agent*

yang bersifat *follower* hanya akan mengikuti *agent* yang bersifat *leader* saja tanpa dapat mengikuti *agent* lainnya yaitu *agent* yang bersifat independen. Dengan begitu apabila hanya terdapat *agent* yang bersifat independen pada *line of sight agent* yang bersifat *follower* maka *agent* tersebut akan mengalami *state* tersesat. Namun jika terdapat *agent* yang bersifat *leader* pada *line of sight agent* yang bersifat *follower* maka *agent* tersebut akan mengikuti *agent* yang bersifat *leader* tersebut. FSM pada *agent* yang bersifat *follower* dapat dilihat pada Gambar 3.14.



Gambar 3.14: FSM pada *agent follower* pada skenario heterogen tipe C

Pembuatan skenario ini berdasarkan kepanikan anak-anak yang hanya akan mengikuti orang yang dikenalnya saja dan akan cenderung diam apabila tidak dapat menemukan orang yang dikenalnya. Dengan begitu *agent* yang bersifat *follower* hanya dapat mengikuti *agent* yang bersifat *leader* saja tanpa dapat mengikuti *agent* lainnya.

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN

Pada bab ini akan dilakukan pengujian dan analisa terhadap desain dan implementasi sistem yang telah diterapkan pada Bab 3. Pengujian dilakukan untuk mengetahui apakah simulasi kerumunan evakuasi bencana pada *agent* heterogen telah berjalan dengan baik menggunakan metode *agent-based* dan *reciprocal velocity obstacle*(RVO). Pada bab ini akan dilakukan dua jenis pengujian yaitu pengujian sistem simulasi untuk dapat melihat apakah setiap variable dalam simulasi dapat berjalan dengan baik atau tidak. Setelah itu dapat dilakukan pengujian hasil simulasi dengan menggunakan berbagai macam kondisi dari setiap *agent*. Pengujian dilakukan pada perangkat lunak Unity Versi 2017.3.1f1 *personal edition* pada sistem operasi Windows 10 64 bit.

4.1 Pengujian Sistem Simulasi

Pengujian ini dilakukan untuk memastikan simulasi kerumunan dapat berjalan dengan baik atau tidak. Pengujian akan dilakukan pada parameter *agent* serta ekstraksi data yang nantinya akan digunakan sebagai parameter dalam suatu pengujian. Pada parameter terdapat beberapa hal yang menentukan *agent* dapat berjalan dengan baik yaitu parameter *seeking*, *steering*, *collision avoidance*, *collision detection*. Selain parameter dari *agent* akan dilakukan juga pengujian terhadap ekstraksi data. Parameter-parameter tersebut yang akan menjadi bahan analisa saat akan melakukan simulasi evakuasi. Apabila terdapat salah satu parameter yang tidak dapat berjalan dengan baik membuat simulasi tidak dapat dianalisa dengan baik.

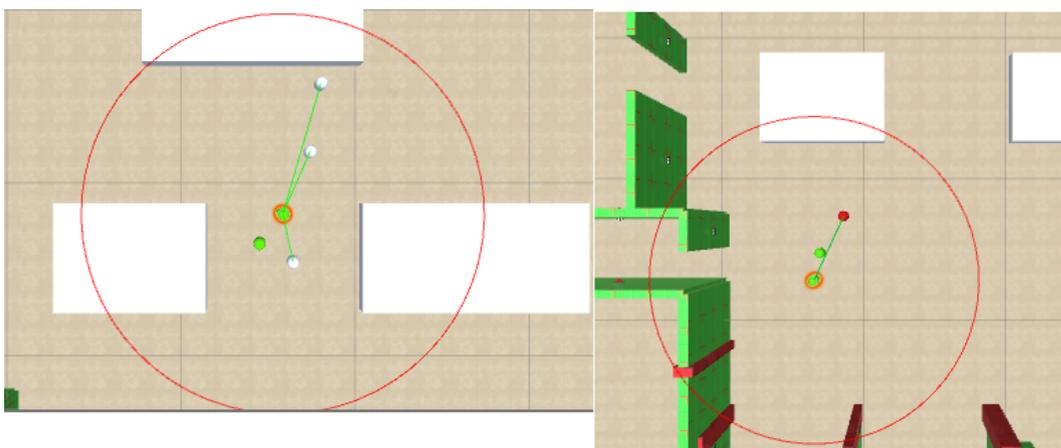
4.1.1 Pengujian *Seeking*

Proses *seeking* adalah suatu proses dimana *agent* mencari titik destinasi yang akan dituju. Pada penelitian ini proses *seeking* dilakukan menggunakan algoritma A* yang terdapat pada perangkat lunak Unity *Engine* dalam bentuk *navmesh*. Proses ini akan mencari navigasi global yang nantinya akan turut serta mempengaruhi proses *steering* dalam hal mencari kecepatan preferensi yang baru. Proses ini harus dapat berjalan agar setiap *agent* dapat bergerak menuju tempat yang diharapkan. Pada penelitian ini terdapat relasi antar *agent* yang mengakibatkan perbedaan destinasi pada *agent*. Pada *agent* yang bersifat independen serta *agent* yang bersifat *leader* memiliki des-

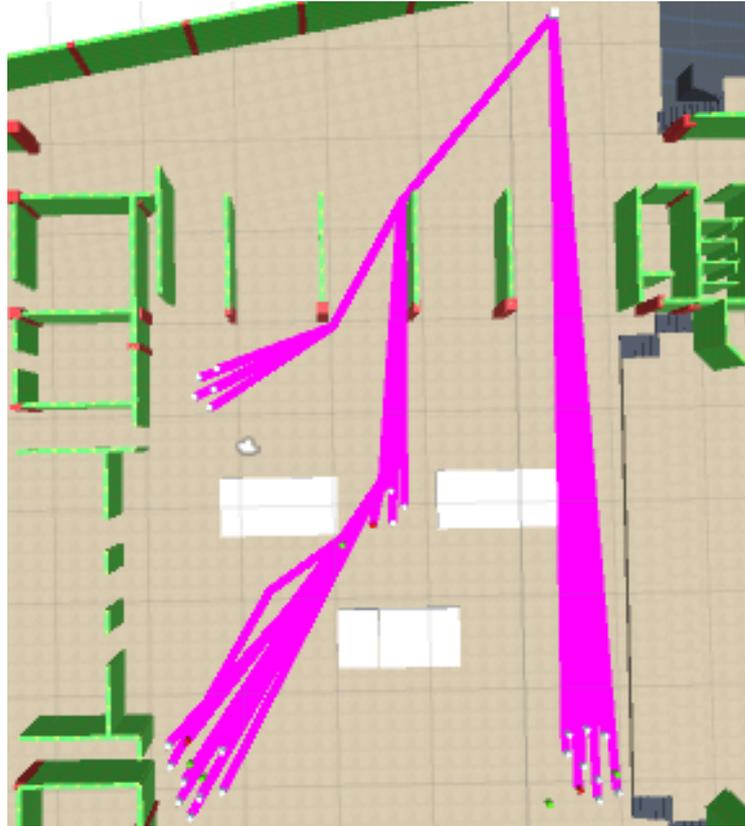
tinasi titik evakuasi seperti yang terlihat pada Gambar 4.1. Sedangkan pada *agent* yang bersifat *follower* maka titik destinasi *agent* tersebut adalah *agent leader* jika terdapat *agent leader* di sekitar *line of sight agent* tersebut. Jika disekitar *agent* tersebut tidak ada *agent leader* maka akan dicari *agent* independen lain(apabila skenario membolehkan *agent* mengikuti *agent* yang bersifat independen) untuk diikuti seperti pada Gambar 4.2. Pada Gambar 4.3 diilustrasikan proses *seeking* pada kerumunan secara keseluruhan menuju tempat evakuasi.



Gambar 4.1: Seeking pada *agent leader*(kiri), seeking pada *agent* independen(kanan)



Gambar 4.2: Seeking pada *agent follower* terhadap *agent* independen(kiri) , seeking pada *agent follower* terhadap *agent leader*(kanan)



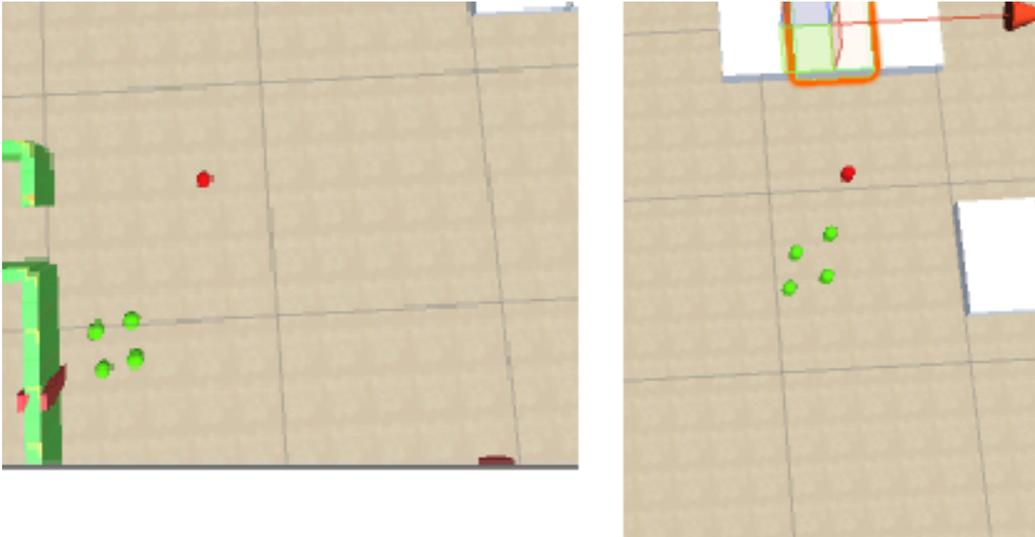
Gambar 4.3: Proses seeking pada kerumunan

4.1.2 Pengujian *Steering*

Setelah proses *seeking* maka setiap *agent* akan mendapatkan preferensi *velocity* sesuai dengan *seeking* yang telah dilakukan sebelumnya. Pada pengujian ini akan dilakukan pengecekan kecepatan *agent follower* yang akan mengikuti *agent leader* maupun *agent independen*. Apabila *agent follower* tersebut jauh maka akan menambah kecepatan, sebaliknya semakin dekat akan mengurangi kecepatannya. Selain itu *agent leader* juga akan mengurangi kecepatannya jika terdeteksi adanya follower di dekat *agent* tersebut. Pengurangan kecepatan ini juga tergantung dengan jarak dengan *agent follower* tersebut. Semakin jauh maka kecepatan *agent leader* akan semakin rendah.

Selain itu jika tidak ditemukan adanya *agent leader* di sekitar *line of sight agent follower*, maka *agent follower* akan mengecek adanya kerumunan *agent independen* lainnya dan akan mengikuti *agent* tersebut. Bedanya pada *agent independen* tidak menyesuaikan kecepatannya terhadap *agent follower*. Jadi pada kasus ini hanya *agent follower* yang menambah kecepatan mengikuti kecepatan *agent independen* tanpa adanya penyesuaian kecepatan *agent inde-*

penden terhadap *agent follower*. Pada Gambar 4.4 diilustrasikan bagaimana *agent follower* dapat mengikuti serta mendekati *agent leader* dengan cara *agent leader* dan *agent follower* menyesuaikan kecepatannya. dan Pada Gambar 4.5 diilustrasikan bagaimana *agent follower* apabila mengikuti kerumunan *agent independen*.



Gambar 4.4: Pada awal *agent follower* tertinggal *agent leader*(kiri), *agent follower* dan *leader* menyesuaikan agar dapat berdekatan(kanan)

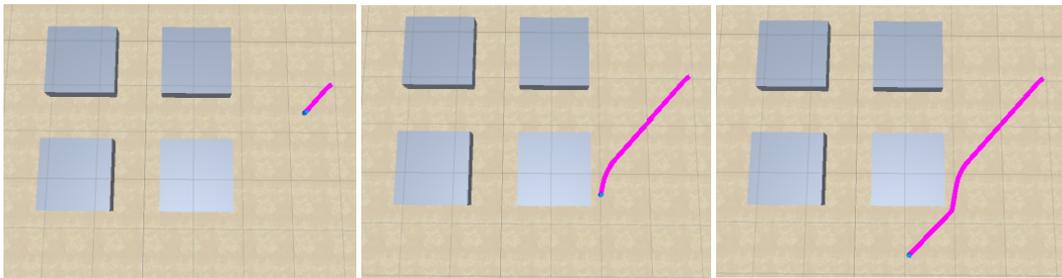


Gambar 4.5: Pada awal *agent follower* tertinggal *agent independen*(kiri), *agent follower* menyesuaikan kecepatan dengan *agent independen* agar dapat berdekatan(kanan)

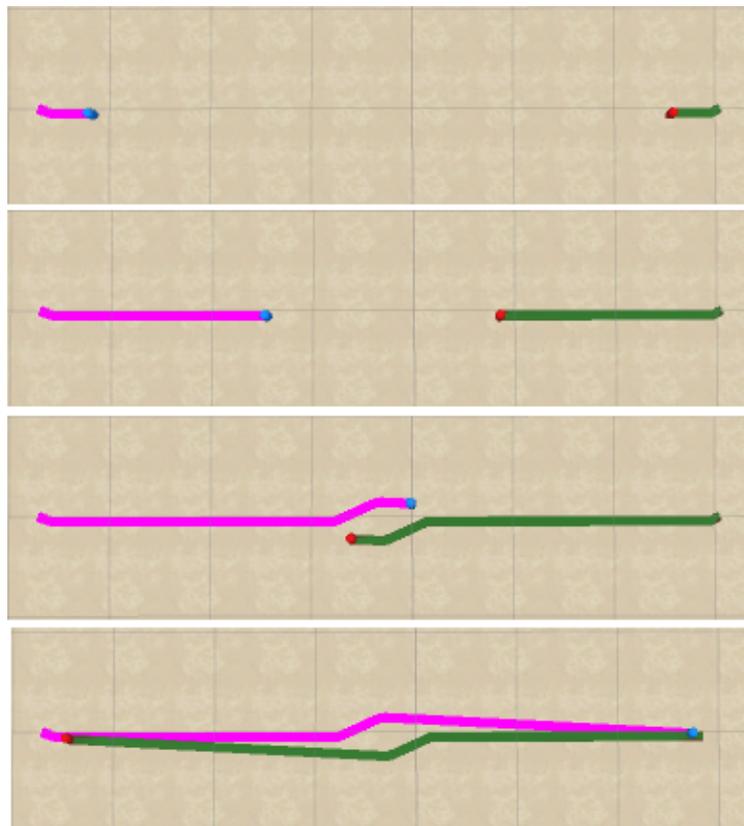
4.1.3 Pengujian *Collision Avoidance*

Pada pengujian ini akan dilakukan pengecekan apakah *agent* dapat melakukan *collision avoidance* terhadap rintangan-rintangan yang ada di dalam si-

mulasi atau tidak. Rintangan dalam suatu simulasi dapat berbentuk rintangan statis maupun dinamis. Pada rintangan statis maka rintangan tersebut tidak memiliki kecepatan atau $\mathbf{v}_{obs} = 0$. Pada Gambar 4.6 diilustrasikan bagaimana *agent* menghindari rintangan yang bersifat statis. Pada rintangan yang dinamis maka setiap rintangan akan memiliki kecepatan. Rintangan dinamis ini biasanya berupa *agent* lain yang berada dalam simulasi. Pada Gambar 4.7 diperlihatkan bagaimana proses *agent* menghindari rintangan dinamis yang berbentuk *agent* lainnya.



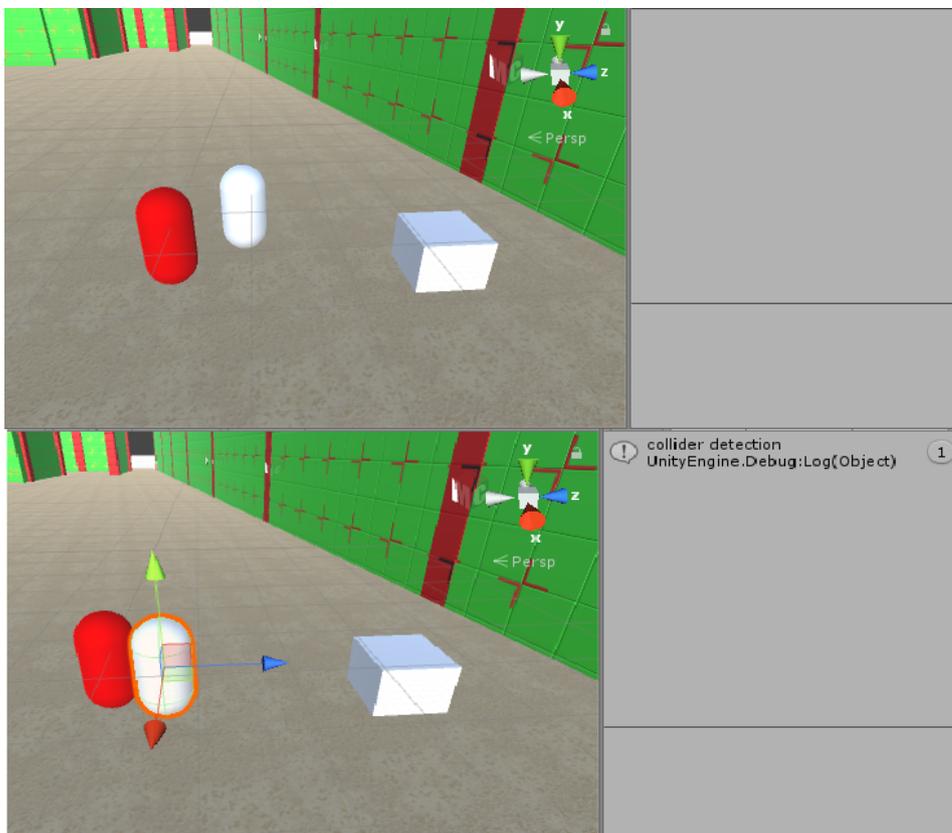
Gambar 4.6: *Agent* menghindari rintangan statis



Gambar 4.7: *Agent* menghindari rintangan dinamis(*agent* lainnya)

4.1.4 Pengujian *Collision Detection*

Pengujian ini dilakukan untuk melihat apakah salah satu parameter dalam pengujian simulasi nanti dapat berjalan dengan baik atau tidak. *Collision detection* akan menentukan apakah suatu *agent* dapat menghindari tabrakan atau tidak. Jika terdapat banyak *collision detection* dalam suatu simulasi maka akan terdapat banyak tabrakan yang terjadi. Pada simulasi ini *collision detection* dapat berupa tabrakan dengan rintangan statis maupun dinamis. Pada Gambar 4.8 diilustrasikan adanya peringatan saat *agent* mengalami tabrakan dengan *agent* lainnya sehingga terjadi peringatan pada *collision detection*.



Gambar 4.8: *Agent* saat belum tabrakan(atas) dan saat mengalami tabrakan dengan (*agent* lainnya)(bawah).

4.1.5 Pengujian Ekstraksi Data

Pada pengujian ini akan memastikan jalannya parameter-parameter yang akan dilakukan pada saat simulasi. Parameter-parameter ini digunakan untuk pengambilan data yang akan dibandingkan saat terjadinya simulasi. Pengujian ini dilakukan dengan mencoba semua fungsi ekstraksi data dan melihat apakah semua parameter dalam ekstraksi data dapat berjalan dengan baik atau tidak.

Tabel 4.1: Tabel ekstraksi data

No	Nama Data	Fungsi Dapat Berjalan/Tidak
1	Jumlah tabrakan	Ya
2	Waktu simulasi	Ya
3	Rata-rata <i>frame rate</i>	Ya
4	Jumlah Agent tersesat	Ya

Parameter yang diambil akan sesuai dengan perbandingan sesuai dengan desain sistem yang telah dibuat. Hasil yang didapat pada simulasi dapat dilihat pada Tabel 4.1.

4.2 Hasil Simulasi

Pada bagian ini akan dilakukan simulasi dengan beberapa skenario. Sebagai skenario pembanding maka akan dibuat simulasi dengan *agent* yang bersifat homogen dengan kondisi kecepatan *agent* sama dan kondisi tidak terdapat relasi pada *agent* manapun di dalam simulasi. Skenario selanjutnya adalah adanya perbedaan maksimal kecepatan dari setiap *agent* di dalam simulasi namun tidak terdapat relasi yang terjadi pada setiap *agent*. Selanjutnya akan dibuat skenario akan terdapat beberapa *agent* yang memiliki relasi dan memiliki kecepatan maksimal yang berbeda-beda sesuai dengan desain skenario yang telah dibuat pada desain sistem.

Simulasi akan dilakukan pada *notebook* dengan spesifikasi prosesor Intel i3-3110M 2.40GHz dengan RAM sebesar 6GB dan GPU Nvidia Geforce 610M. Simulasi dilakukan pada perangkat lunak *engine* Unity 3d yang berjalan pada sistem operasi Windows 10 64 bit. Simulasi akan dilakukan dengan jumlah *agent* yang bervariasi mulai dari 20 hingga 300 *agent* dalam satu simulasi. Simulasi akan dilakukan di dalam peta gedung acara robotika ITS dengan titik asal sebanyak lima titik dengan satu titik tujuan. Simulasi dimulai saat setiap *agent* mulai bergerak hingga semua *agent* sampai pada titik akhir proses evakuasi. Di setiap titik asal akan ditempatkan berbagai jenis *agent* dari mulai *agent* yang bersifat independen, *agent* yang bersifat *leader* dan *agent* yang bersifat *follower*. Penempatan di titik-titik mulai tersebut akan dibuat acak. Setiap kali simulasi akan dihitung waktu simulasi, jumlah rata-rata *frame per second*, jumlah *collision* yang terjadi saat simulasi baik *collision* pada rintangan statis maupun pada rintangan dinamis serta jumlah *agent* yang bersifat

follower yang tertinggal jika terdapat *agent* yang bersifat *follower* yang terse-
sat. Peta simulasi yang menunjukkan titik asal dan titik akhir evakuasi dapat
dilihat pada Gambar 4.9.



Gambar 4.9: Peta saat simulasi dilakukan

Terdapat lima jenis simulasi yang akan diujikan. Simulasi pertama setiap *agent* akan bersifat homogen dengan memiliki kecepatan yang sama dan tidak memiliki relasi antar *agent*. Pada simulasi kedua terdapat perbedaan kelajuan pada *agent* namun tidak terdapat relasi pada semua *agent* di dalam simulasi. Selbihnya akan ada tiga jenis simulasi dengan kondisi *agent* yang heterogen. berikut ketiga jenis simulasi tersebut:

1. **Simulasi heterogen tipe A** : Pada simulasi ini terdapat tiga jenis *agent* yaitu *agent* independen, *leader* dan *follower*. *Follower* mengikuti *leader* dan independen jika tidak terdapat *leader*. Pada skenario heterogen tipe A ini *follower* akan lebih memilih mengikuti *leader* dibandingkan independen apabila kedua *agent* sama-sama berada pada *line of sight agent* yang bersifat *follower*.
2. **Simulasi heterogen tipe B** : Pada simulasi ini tidak terdapat *agent leader* sehingga *agent follower* hanya dapat mengikuti *agent independen*

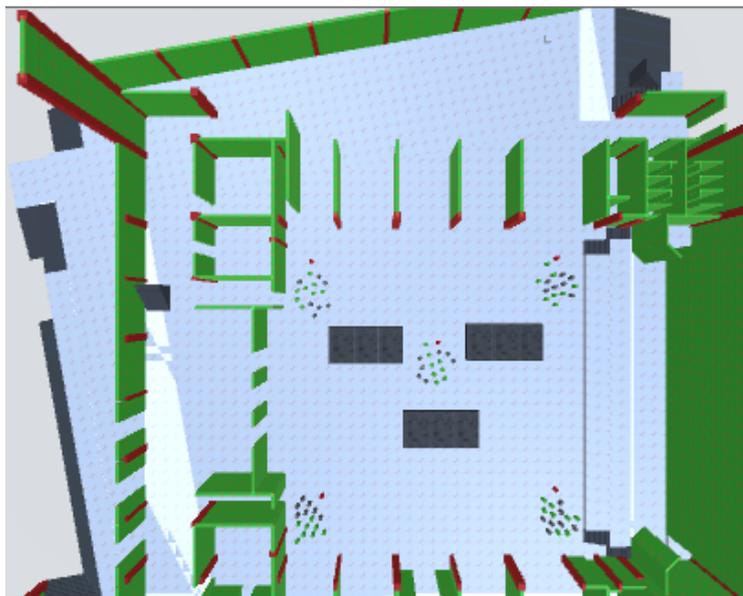
yang berada pada *line of sight agent* tersebut.

3. **Simulasi heterogen tipe C** : Pada simulasi ini semua *agent* lengkap namun akan dibuat kondisi *agent follower* hanya dapat mengikuti *leader* dan tidak dapat mengikuti *agent* independen. Dengan begitu biarpun *agent* yang bersifat independen terdapat pada *line of sight agent* yang bersifat *follower* tetap *agent* yang bersifat *follower* tidak akan mengikuti *agent* tersebut.

Dalam simulasi akan diambil beberapa parameter yang nantinya akan dibandingkan di setiap simulasi. Parameter-parameter tersebut adalah:

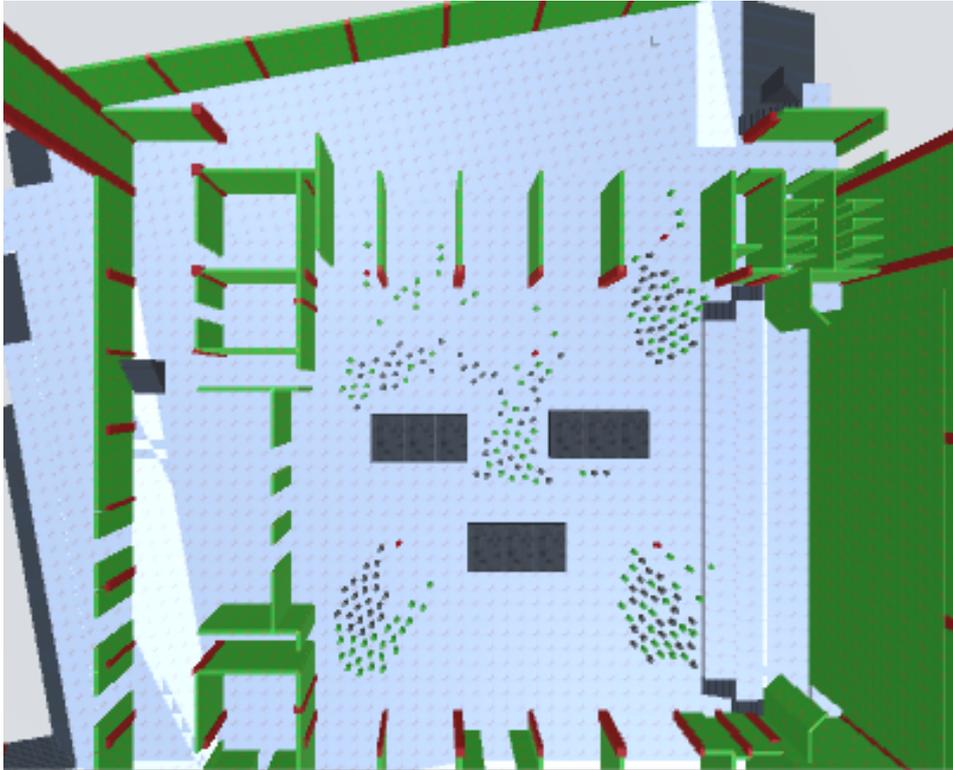
1. Parameter waktu simulasi.
2. Parameter *frame* dalam simulasi.
3. Parameter *collision* yang terjadi saat simulasi.
4. Parameter *agent* yang tersesat pada simulasi khususnya pada simulasi *agent* heterogen.

Secara garis besar parameter-parameter pada simulasi tidak terlalu naik signifikan pada simulasi dengan jumlah *agent* yang masih sedikit seperti pada Gambar 4.10. Kenaikan simulasi baru terasa saat jumlah *agent* meningkat lebih dari 100 *agent*.



Gambar 4.10: Simulasi dengan jumlah *agent* yang relatif sedikit

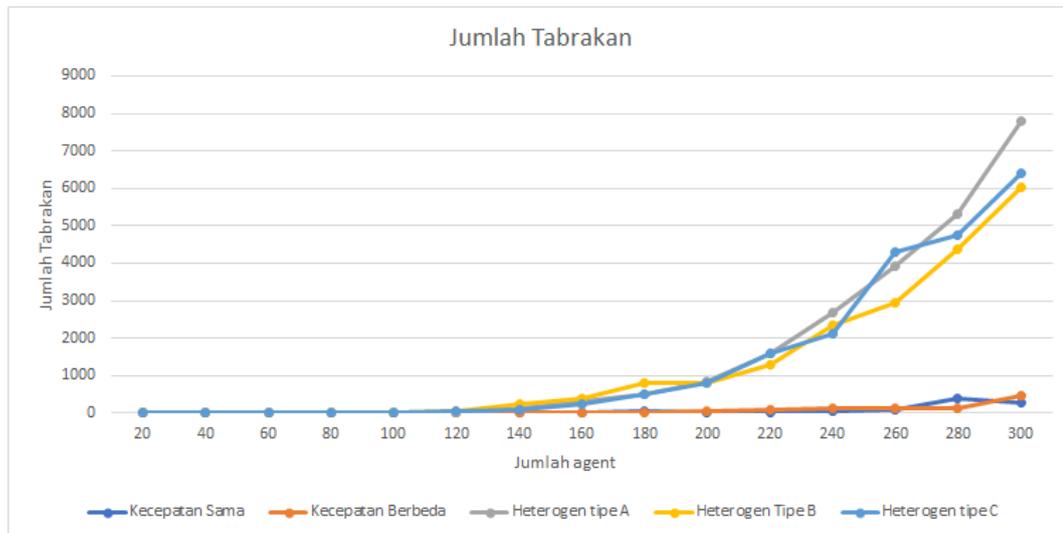
Kenaikan parameter simulasi dengan terjadi pada *agent* yang relatif banyak seperti pada Gambar 4.11. Secara keseluruhan terdapat peningkatan yang signifikan pada saat *agent* yang berjumlah banyak seperti pada parameter *collision*, waktu simulasi maupun penurunan FPS.



Gambar 4.11: Simulasi dengan jumlah *agent* yang banyak

Pada jumlah tabrakan yang terjadi setiap simulasi baru mulai terjadi tabrakan pada saat *agent* berjumlah 140 seperti pada Gambar 4.12. Namun pada skenario *agent* yang tidak memiliki relasi, tabrakan tidak terlalu terlihat dan hanya baru terlihat tabrakan pada jumlah *agent* diatas 260 *agent* dan jumlah tabrakan yang terjadi pun tidak terlalu banyak seperti jumlah tabrakan pada skenario *agent* yang memiliki relasi. Tabrakan terjadi karena banyaknya jumlah *agent* yang berada disekitar *agent* dan menyebabkan banyaknya perhitungan untuk mencari kecepatan preferensi pada *agent*. Pada metode RVO sendiri terdapat parameter yang digunakan untuk membatasi jumlah *agent* yang berada disekitar untuk menghindari beratnya proses komputasi.

Pada *agent* yang bersifat homogen dengan kelajuan yang sama memiliki jumlah tabrakan terkecil dengan jumlah tabrakan paling banyak sebanyak 377 tabrakan disusul dengan *agent* yang tidak memiliki relasi namun berbeda kela-



Gambar 4.12: Grafik jumlah tabrakan pada setiap simulasi

jumlah dengan jumlah tabrakan 464 tabrakan. Hal ini terjadi dikarenakan setiap *agent* fokus terhadap titik tujuan dan tidak terdapat *agent* yang bergantung terhadap *agent* lainnya. Dengan begitu tidak ada perubahan kecepatan preferensi yang signifikan diantara para *agent*. Setelah itu terdapat kenaikan yang signifikan pada saat simulasi heterogen terjadi. Hal ini dikarenakan adanya relasi pada *agent* yang menyebabkan *agent* memiliki kecepatan preferensi yang mudah berubah drastis karena perubahan titik tujuan *agent follower*.

Diantara ketiga jenis simulasi *agent* heterogen, Simulasi heterogen tipe B memiliki tabrakan paling sedikit karena *agent follower* lebih bebas dalam memilih *leader* yang akan diikutinya dengan jumlah 6032 tabrakan. Dengan adanya kebebasan memilih *agent* yang diikuti dan tidak terpakunya pada *agent* tertentu membuat pergerakan *follower* lebih leluasa dan tidak menghalangi *agent* lainnya. Selain itu pada simulasi ini juga tidak terdapat *agent* yang bersifat *leader* yang kelajuan preferensinya dapat berubah tergantung jarak *agent* tersebut dengan *agent* yang bersifat *follower*. Karena hal tersebut akhirnya malah dapat menghambat pergerakan *agent* lainnya yang berada disekitar *agent* tersebut.

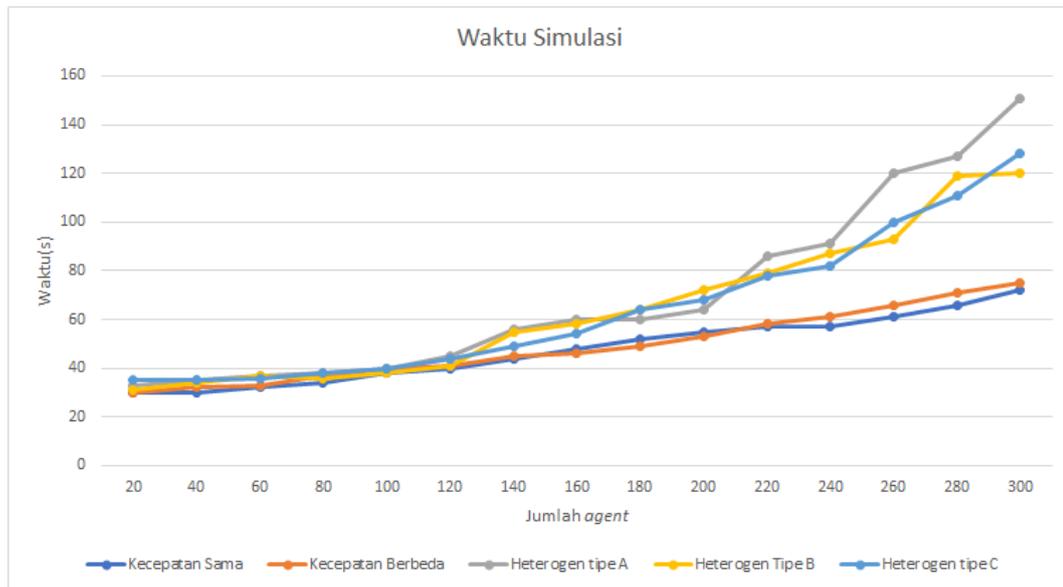
Hal ini berbeda dengan simulasi heterogen tipe A dan C yang memiliki lebih banyak tabrakan dengan simulasi heterogen tipe C sebanyak 6397 tabrakan. Pada skenario ini lebih sedikit dibanding skenario tipe A dikarenakan adanya *agent* yang tersesat saat terjadinya simulasi sedangkan tabrakan banyak terjadi saat melewati lorong pintu keluar proses evakuasi. Dengan adanya *agent* yang tersesat dan akhirnya tertinggal ini membuat jumlah *agent* yang

nantinya melewati lorong keluar menjadi lebih sedikit dibandingkan skenario lainnya.

Skenario pada *agent* heterogen tipe A mendapatkan jumlah tabrakan paling banyak dibanding simulasi lainnya dengan jumlah tabrakan paling banyak sebanyak 7781 tabrakan. Hal ini disebabkan karena skenario ini lebih kompleks dibandingkan dengan skenario lainnya. FSM *follower* pada skenario ini membuat *follower* dapat mengikuti *leader* dan *agent* independen jika tidak terdapat *leader* di sekitar *agent* tersebut. Selain itu *agent* yang bersifat *leader* ini juga akan melakukan penyesuaian kecepatan sesuai dengan jarak *follower*-nya. Dengan begitu kompleksitas pada setiap *agent* inilah yang membuat kecepatan preferensi pada setiap *agent* mudah berubah dan menyebabkan tabrakan. Selain itu juga pergerakan penyesuaian *agent* terhadap *agent* lainnya juga menghambat proses evakuasi.

Pengujian waktu simulasi dilakukan untuk melihat pengaruh dari skenario *agent* terhadap waktu proses evakuasi yang akan dilakukan. Selain itu dalam suatu proses evakuasi kebakaran waktu proses evakuasi sangat penting untuk diketahui. Setiap detik dalam proses evakuasi dapat berarti satu nyawa dari para korban proses evakuasi. Secara keseluruhan waktu evakuasi dengan *agent* yang tidak memiliki relasi memiliki waktu simulasi yang lebih cenderung sebentar dibandingkan dengan waktu simulasi dengan skenario *agent* yang memiliki relasi. Kenaikan waktu simulasi di tiap simulasi mulai berbeda jauh saat simulasi dengan jumlah *agent* lebih dari 140 *agent* seperti pada Gambar 4.13. Diantara semua simulasi simulasi dengan *agent* yang homogen memiliki jumlah simulasi yang paling sebentar dengan lama simulasi terlama sebesar 72 detik. Waktu simulasi ini disusul oleh *agent* yang memiliki kecepatan yang berbeda namun tidak memiliki relasi dengan lama simulasi sebesar 75 detik. Kedua skenario yang tidak memiliki relasi ini memiliki perbedaan waktu yang tidak terlalu jauh dan cenderung setiap simulasi bergantian jumlah terbesar diantara keduanya.

Waktu simulasi mulai berbeda jauh pada skenario simulasi heterogen yang memiliki relasi di tiap *agent* yang terdapat pada simulasi tersebut. Pada *agent* yang tidak memiliki relasi memiliki waktu simulasi yang relatif lebih sebentar dibanding yang lain dikarenakan setiap *agent* akan langsung pergi ke titik tujuan evakuasi sehingga tidak perlu mengikuti *agent* lainnya. Dengan begitu kecepatan preferensi setiap *agent* tidak serumit dengan kecepatan simulasi *agent* yang memiliki relasi.



Gambar 4.13: Grafik waktu simulasi pada setiap simulasi

Pada simulasi heterogen tipe A, B dan C, simulasi heterogen tipe B memiliki waktu simulasi lebih sedikit dibanding simulasi heterogen lainnya dengan waktu maksimal simulasi selama 120 detik. Pada simulasi ini *agent follower* memiliki lebih banyak pilihan dalam mencari *agent* yang akan diikutinya sehingga hal ini membuat *agent follower* tidak terlalu membatasi pergerakan *agent* lainnya dibanding simulasi heterogen tipe lain. Selain itu dengan tidak adanya *agent* yang bersifat *leader* membuat simulasi skenario heterogen tipe B ini menjadi lebih cepat dibandingkan simulasi heterogen lainnya karena sifat *agent* yang bersifat *leader* yang harus menyesuaikan kecepatan *agent* yang bersifat follower. Dengan begitu hanya *agent* yang bersifat *follower* saja yang akan menambah kecepatannya mengikuti *agent* yang bersifat independen yang diikutinya.

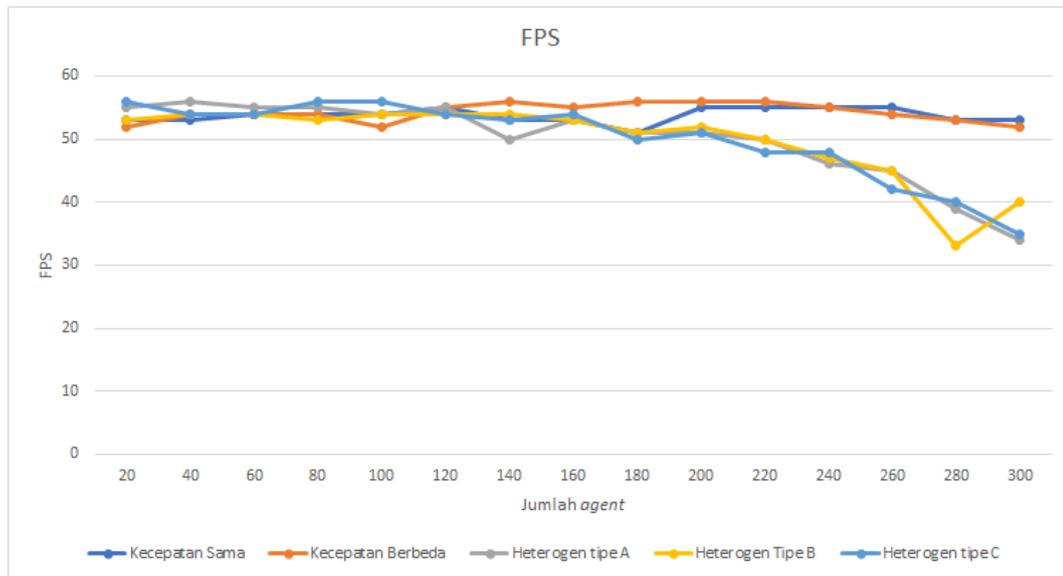
Simulasi heterogen tipe C lebih lama sedikit dibandingkan simulasi heterogen tipe B namun masih lebih cepat dibandingkan simulasi heterogen tipe A. Simulasi heterogen tipe B dan C terkadang bergantian dalam tiap variasi jumlah *agent* dalam hal lama waktu simulasi dengan waktu maksimal simulasi 128 detik. Simulasi ini lebih lama dibanding simulasi heterogen tipe B karena banyaknya *agent follower* yang berada dalam keadaan tersesat karena hanya dapat mengikuti *leader* tanpa dapat mengikuti kerumunan *agent* independen. Dengan begitu *agent follower* yang tersesat ini akan berada dalam kondisi $V_{pref} = 0$ sehingga dapat menghambat gerakan *agent* lainnya.

Simulasi heterogen tipe A memiliki waktu simulasi paling lama diban-

dingkan simulasi lainnya dengan waktu maksimal 151 detik. Waktu lamanya simulasi ini disebabkan kompleksitas skenario *agent* yang bersifat *follower* lebih kompleks dibandingkan simulasi lainnya. Simulasi heterogen tipe A terdapat *agent leader* didalam simulasi memiliki penyesuaian kecepatan terhadap *agent* lainnya. Dengan begitu *agent* yang *leader* akan bereaksi terhadap *agent follower* dengan cara menurunkan kelajuannya dimana hal ini tidak terjadi pada *agent* independen. Selain memperlambat *agent* yang bersifat *leader* hal ini juga memperlambat *agent* yang bersifat *follower* karena pergerakannya menjadi lebih lambat. Hal ini juga berdampak pada kerumunan yang akhirnya menghambat pergerakan kerumunan secara keseluruhan.

Pengujian FPS(*frame per second*) dilakukan untuk mengetahui pengaruh jumlah *agent* serta tipe skenario simulasi terhadap performa *frame rate* yang terjadi saat simulasi terjadi. Pengujian ini juga akan melihat pengaruh kompleksitas serta jumlah agent pada skenario yang mempengaruhi performa FPS. Pada keseluruhan simulasi FPS awalnya stabil diantara 50 hingga 60 FPS pada jumlah *agent* kurang dari 200 *agent* seperti yang terlihat pada Gambar 4.14. Pada simulasi yang tidak memiliki relasi antar *agent* baik dengan kecepatan yang sama maupun dengan kecepatan yang berbeda, FPS cenderung stabil hingga *agent* berjumlah 300 *agent*. Pada skenario dengan *agent* yang memiliki kecepatan yang sama, FPS paling kecil terjadi saat *agent* berjumlah 180 *agent* dengan nilai FPS sebesar 51 FPS. Pada *agent* dengan kecepatan yang berbeda FPS paling kecil terjadi saat *agent* berjumlah 300 *agent* dan 100 *agent* dengan nilai FPS sebesar 52 FPS. Bahkan pada skenario ini pada jumlah *agent* sebanyak 140 nilai FPS dapat memiliki nilai yang besar sebesar 56 FPS.

Pada skenario *agent* dengan menggunakan relasi antar *agent* baik itu pada simulasi heterogen tipe A, tipe B dan tipe C terdapat penurunan FPS secara signifikan pada saat jumlah *agent* sebesar 200 FPS. Namun diantara ketiga simulasi heterogen ini tidak memiliki perbedaan FPS yang signifikan sesama skenario *agent* heterogen lainnya. Diantara tiga tipe simulasi heterogen, tipe B memiliki FPS yang paling tidak stabil dan memiliki FPS paling kecil sebesar 33 FPS pada jumlah *agent* sebesar 280 *agent*. Namun pada jumlah *agent* sebesar 300 *agent* FPS pada skenario ini kembali naik dengan nilai 40 FPS. FPS yang tiba-tiba turun pada skenario simulasi ini dapat dikarenakan terhalangnya proses simulasi kerumunan yang disebabkan oleh rintangan baik rintangan yang statis maupun rintangan dinamis.



Gambar 4.14: Grafik *frame per second*(FPS) pada setiap simulasi

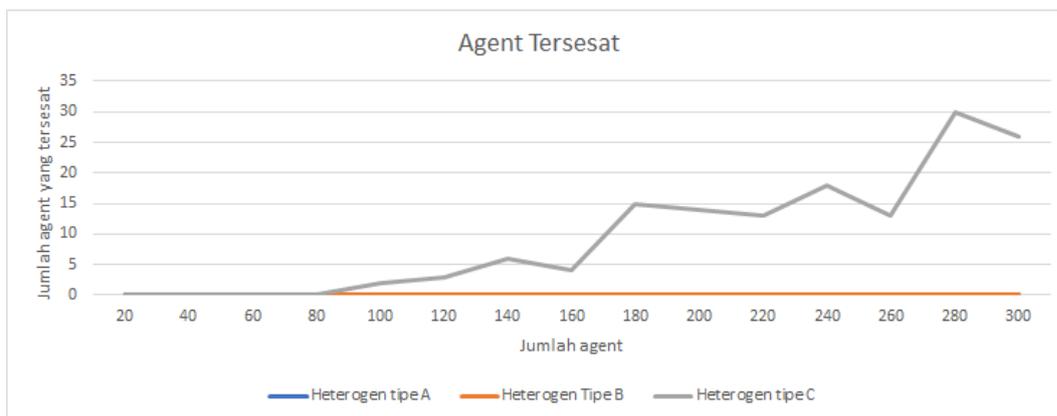
Hal ini juga terjadi pada simulasi tipe C yang memiliki FPS lebih stabil dibanding FPS simulasi heterogen lainnya dengan jumlah terkecil FPS sebesar 35 FPS. Perbedaan FPS pada tipe C ini juga tidak terlalu besar dibandingkan dengan FPS skenario tipe A. Hal ini kemungkinan disebabkan adanya *agent* yang tersesat dan akhirnya tidak melanjutkan perjalanan. Dengan tidak melanjutkan perjalanan ini maka komputasi dengan skenario tipe C ini cenderung lebih sedikit dibandingkan skenario tipe lainnya. Dengan begitu akhirnya berpengaruh terhadap *frame rate* yang terdapat pada simulasi.

Penurunan FPS pada skenario tipe A hampir sama dibanding skenario heterogen lainnya namun sedikit lebih menurun dibandingkan dengan skenario tipe C dengan jumlah FPS paling kecil sebesar 34 FPS. Simulasi tipe A ini memiliki kompleksitas lebih besar dibanding dengan skenario lainnya. Dengan begitu proses komputasi yang terjadi pada simulasi heterogen tipe menjadi lebih banyak dibanding simulasi lainnya. Dengan begitu *frame rate* pada simulasi ini juga cenderung lebih kecil namun pada beberapa percobaan seperti pada saat jumlah *agent* sebesar 280 *agent* skenario tipe A ini dapat memiliki FPS lebih besar dibandingkan FPS tipe B. Hal ini kemungkinan pada saat itu simulasi tipe B mengalami sedikit hambatan saat simulasi yang akhirnya menyebabkan komputasi sedikit terganggu dan *frame rate* menurun.

Parameter berikutnya terdapat pada simulasi ini ini adalah parameter *agent* yang tersesat. Hal ini untuk mengetahui serta membandingkan *agent* yang tersesat di dalam suatu simulasi. Parameter *agent* yang tersesat diperun-

tukan kepada simulasi yang memiliki relasi antar *agent* seperti pada simulasi heterogen tipe A, B dan C. Pada simulasi heterogen terdapat relasi *leader-follower* yang menyebabkan *agent* mengikuti *agent* lainnya yang berada dalam *line of sight* *agent* tersebut. Apabila ternyata tidak terdapat *agent* dalam *line of sight* pada *agent* tersebut hingga akhir proses evakuasi maka *agent* tersebut akan mengalami kondisi tersesat dan tidak dapat melakukan proses evakuasi.

Setelah dilakukan simulasi didapatkan hasil hanya simulasi heterogen tipe C saja yang terdapat *agent* yang tersesat seperti yang terlihat pada Gambar 4.15. Pada simulasi heterogen tipe A dan tipe B tidak terdapat *agent* yang tersesat sehingga setiap *agent* dapat sampai pada titik evakuasi. Hal ini dikarenakan pada simulasi tipe A dan tipe B *agent follower* memiliki banyak pilihan *agent* yang akan diikuti. Selain *agent leader*, *agent follower* dapat mengikuti kerumunan *agent* independen.



Gambar 4.15: Grafik *agent follower* yang tersesat pada simulasi heterogen tipe A, B dan C

Hal ini yang tidak terjadi pada simulasi heterogen tipe C dimana *agent follower* hanya dapat mengikuti mengikuti *agent leader* dan tidak dapat mengikuti *agent* independen. Pada simulasi heterogen tipe C mulai terdapat *agent* yang tersesat pada saat simulasi dengan jumlah 100 *agent* dengan jumlah *agent* yang tersesat sebanyak dua *agent*. Seiring dengan bertambahnya jumlah *agent* yang terdapat pada simulasi, *agent* yang tersesat dalam simulasi juga jumlahnya bertambah.

Jumlah terbanyak *agent* yang tersesat saat simulasi dengan *agent* sebanyak 280 *agent* dengan jumlah *agent* yang tersesat sebanyak 30 *agent*. Kemungkinan tersesatnya *agent* dikarenakan karena adanya antrian saat *agent* harus melewati rintangan statis dan saat itu *agent follower* tertinggal *agent*

leader dan setelahnya *agent leader* tidak terdapat di dalam radius *line of sight* *agent follower*. Dengan begitu akhirnya *agent follower* akan tersesat karena ditinggal oleh *agent leader*. Dengan adanya *agent follower* yang tersesat ini juga akhirnya memperlambat laju *agent* lainnya karena *agent follower* dalam keadaan posisi diam dan $V_{pref} = 0$.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian yang dilakukan dengan lima jenis skenario simulasi maka dapat diberikan kesimpulan sebagai berikut:

1. Dari hasil simulasi evakuasi kerumunan terdapat adanya tabrakan saat jumlah *agent* lebih besar dari 120 *agent* dengan jumlah tabrakan paling sedikit pada skenario percobaan simulasi *agent* homogen dengan jumlah tabrakan sebanyak 377 tabrakan dan paling banyak pada skenario *agent follower* dapat mengikuti *agent leader* maupun *agent* independen yaitu pada skenario heterogen tipe A terdapat tabrakan dengan jumlah 7781 tabrakan.
2. Waktu simulasi yang ditempuh dalam lima skenario percobaan simulasi didapatkan waktu terkecil pada skenario *agent* yang homogen dengan paling lama sebesar 72 detik. Sedangkan waktu terlama terjadi pada skenario heterogen tipe A dengan *agent follower* dapat mengikuti *agent leader* maupun *agent* independen dengan lama simulasi paling lama sebesar 151 detik.
3. Dari kelima skenario simulasi percobaan FPS cenderung stabil diantara 50 FPS hingga 56 FPS pada saat *agent* berjumlah kurang dari 220 *agent*. Pada skenario *agent* tidak memiliki relasi FPS cenderung stabil hingga *agent* berjumlah 300 *agent*. Penurunan FPS paling tinggi terjadi pada skenario heterogen tipe B pada saat tidak terdapat *agent leader* sehingga *agent follower* hanya dapat mengikuti independen dengan jumlah FPS sebesar paling kecil 33 FPS.
4. Jumlah *agent* yang tersesat terdapat pada skenario tipe C yaitu pada skenario *agent follower* hanya dapat mengikuti *leader* tetapi tidak dapat mengikuti kerumunan *agent* independen terdapat *agent follower* yang hilang dengan saat simulasi lebih dari 80 *agent* dengan jumlah *agent* yang hilang paling banyak sebesar 30 *agent follower*.

5.2 Saran

Untuk pengembangan penelitian selanjutnya maka ada beberapa hal yang dapat diperhatikan:

1. Skenario simulasi dapat dibuat lebih kompleks lagi dengan beberapa tambahan parameter lain selain *agent*.
2. Lingkungan, jumlah *agent* serta perangkat keras saat melakukan simulasi dapat ditingkatkan lebih baik lagi.

DAFTAR PUSTAKA

- V J Blue, M J Embrechts, and J L Adler. CELLULAR AUTOMATA MODELING OF PEDESTRIAN MOVEMENTS. IEEE Conference on Systems, Man, and Cybernetics, pages 2320–2323, 1997.
- V J Blue and Jeffrey L Adler. Emergent Fundamental Pedestrian Flows from Cellular Automata Microsimulation. Transportation Research Record, (98): 29–36, 1998.
- V J Blue and Jeffrey L Adler. Cellular Automata Microsimulation of bi-directional pedestrian flow. transportation research record, (99):135–141, 2001.
- Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. Physical Review, pages 4282–4286, 1995. doi: 10.1103/PhysRevE.51.4282.
- Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH Computer Graphics, 21(4):25–34, 1987. ISSN 00978930. doi: 10.1145/37402.37406. URL <http://portal.acm.org/citation.cfm?doid=37402.37406>.
- Kiran Ijaz, Shaleeza Sohail, and Sonia Hashish. A Survey of Latest Approaches for Crowd Simulation and Modeling using Hybrid Techniques. 2015. doi: 10.1109/UKSim.2015.46.
- Peter Norvig. Artificial Intelligence : A Modern Approach. ISBN 0137903952. URL http://thuvienso.thanglong.edu.vn/handle/DHTL_123456789/4010.
- J E Doran, S Franklin, and N R Jennings. On cooperation in multi-agent systems *. The knowledge Engineering Review, 12(1994):309–314, 1997.
- Mevludin Glavic. Agents and Multi-Agent Systems : A Short Introduction for Power Engineers. pages 1–21, 2006.
- S Gwynne, E R Galea, M Owen, P J Lawrence, and L Filippidis. A Review of the Methodologies Used in Evacuation Modelling. Fire and Materials, 388 (September 1998):383–388, 1999.

- Enrico Ronchi. A RESEARCH ROADMAP FOR EVACUATION MODELS USED IN FIRE SAFETY ENGINEERING. Fire and Evacuation Modelling Technical Conference 2016, 2016.
- CJ Khisty. Pedestrian Flow Characteristics on Stairways during Disaster Evacuation. 034, 1985.
- Rita Fahy, G Proulx, and Lata Aiman. ' Panic ' and human behaviour in fire. Proceedings of the 4th International Symposium on Human Behaviour in Fire, pages 387–398, 2009.
- Jur Den Van Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. Proceedings - IEEE International Conference on Robotics and Automation, pages 1928–1935, 2008. ISSN 10504729. doi: 10.1109/ROBOT.2008.4543489.
- Jamie Snape, Student Member, Stephen J Guy, and Dinesh Manocha. The Hybrid Reciprocal Velocity Obstacle. IEEE Transactions on Robotics (T-RO), 27(October):1–11, 2011.
- Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n -body Collision Avoidance. Robotics Research: The 14th International Symposium ISRR, Springer Tracts in Advanced Robotics, 70:3–19, 2011.
- Muzhou Xiong, Michael Lees, Wentong Cai, Suiping Zhou, Malcolm Yoke, and Hean Low. Hybrid modelling of crowd simulation. Procedia Computer Science, 1(1):57–65, 2012. ISSN 1877-0509. doi: 10.1016/j.procs.2010.04.008. URL <http://dx.doi.org/10.1016/j.procs.2010.04.008>.
- Pablo Cristian, A Marcela Printista, and Emilio Luque. A Hybrid Simulation Model to Test Behaviour Designs in an Emergency Evacuation. Procedia - Procedia Computer Science, 9:266–275, 2012. ISSN 1877-0509. doi: 10.1016/j.procs.2012.04.028. URL <http://dx.doi.org/10.1016/j.procs.2012.04.028>.
- Muzhou Xiong, Wentong Cai, Suiping Zhou, Malcolm Yoke Hean Low, Feng Tian, Dan Chen, Darren Wee Sze Ong, and Benjamin D. Hamilton. A case study of multi-resolution modeling for crowd simulation. pages 17:1–17:8, 2009. URL <http://dl.acm.org/citation.cfm?id=1639809.1639827>.

Amit Patel. Amit's A* Pages, 2014. URL <https://www.redblobgames.com/pathfinding/a-star/introduction.html>.

Peter E Hart and J Nils. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE transactions on Systems Science and Cybernetics, 4(2):100–107, 1968.

Tuomo Rinne, Kati Tillander, and Peter Grönberg. Data collection and analysis of evacuation situations. JULKAISIJA – UTGIVARE – PUBLISHER, 2010. ISBN 9789513876739. URL <http://www.vtt.fi/publications/index.jsp>.

Laboratorium Robotika, 2018. URL <https://www.its.ac.id/riset/laboratorium-unggulan/laboraturium-robotika/>.

G Proulx. Evacuation Time and Movement in Apartment Buildings*. Fire Safety Journal, 24:229–246, 1995.

Richard D Peacock, Bryan L Hoskins, and Erica D Kuligowski. Overall and Local Movement Speeds During Fire Drill Evacuations in Buildings up to 31 Stories. National Institute of Standards and Technology Technical Note 1675, 2010.

Tan Kay Chuan, Markus Hartono, and Naresh Kumar. International Journal of Industrial Ergonomics Anthropometry of the Singaporean and Indonesian populations. International Journal of Industrial Ergonomics, 40(6):757–766, 2010. ISSN 0169-8141. doi: 10.1016/j.ergon.2010.05.001. URL <http://dx.doi.org/10.1016/j.ergon.2010.05.001>.

Halaman ini sengaja dikosongkan

LAMPIRAN

Tabel 1: Hasil Simulasi *agent* dengan kecepatan sama dan tanpa relasi

Jumlah Agent	Waktu Simulasi	Jumlah Frame	Jumlah Tabrakan	FPS
20	30	1582	0	53
40	30	1614	0	53
60	32	1734	0	54
80	34	1839	0	54
100	38	2069	2	54
120	40	2220	4	55
140	44	2333	4	53
160	48	2552	18	53
180	52	2660	40	51
200	55	3049	28	55
220	57	3168	25	55
240	57	3142	36	55
260	61	3354	77	55
280	66	3531	377	53
300	72	3833	258	53

Tabel 2: Hasil simulasi *agent* dengan kecepatan berbeda dan tanpa relasi

Jumlah Agent	Waktu Simulasi	Jumlah Frame	Jumlah Tabrakan	FPS
20	30	1552	0	52
40	32	1731	0	54
60	33	1793	0	54
80	37	2027	0	54
100	40	2091	0	52
120	41	2268	0	55
140	45	2534	0	56
160	46	2528	20	55
180	49	2739	2	56
200	53	2963	36	56
220	58	3228	98	56
240	61	3364	131	55
260	66	3564	108	54
280	71	3786	134	53
300	75	3889	464	52

Tabel 3: Hasil Simulasi Heterogen Tipe A

Jumlah Agent	Waktu Simulasi	Jumlah Frame	Jumlah Tabrakan	FPS	Jumlah Agent yang Tersesat
20	33	1829	0	55	0
40	35	1948	0	56	0
60	37	2040	2	55	0
80	38	2095	14	55	0
100	40	2179	8	54	0
120	45	2462	16	55	0
140	56	2803	136	50	0
160	60	3208	306	53	0
180	60	3087	483	51	0
200	64	3310	856	51	0
220	86	4340	1589	50	0
240	91	4259	2691	46	0
260	120	5380	3937	45	0
280	127	4958	5307	39	0
300	151	5143	7781	34	0

Tabel 4: Hasil Simulasi Heterogen Tipe B

Jumlah Agent	Waktu Simulasi	Jumlah Frame	Jumlah Tabrakan	FPS	Jumlah Agent yang Tersesat
20	31	1642	0	53	0
40	34	1837	0	54	0
60	37	1997	6	54	0
80	36	1925	8	53	0
100	38	2061	6	54	0
120	41	2221	48	54	0
140	55	2980	252	54	0
160	58	3055	378	53	0
180	64	3253	808	51	0
200	72	3764	811	52	0
220	79	3973	1301	50	0
240	87	4106	2356	47	0
260	93	4221	2938	45	0
280	129	4263	4370	33	0
300	120	4867	6032	40	0

Tabel 5: Hasil Simulasi Tipe C

Jumlah Agent	Waktu Simulasi	Jumlah Frame	Jumlah Tabrakan	FPS	Jumlah Agent yang Tersesat
20	35	1960	0	56	0
40	35	1902	0	54	0
60	36	1957	0	54	0
80	38	2119	0	56	0
100	40	2243	2	56	2
120	44	2394	54	54	3
140	49	2630	86	53	6
160	54	2915	252	54	4
180	64	3245	504	50	15
200	68	3508	805	51	14
220	78	3748	1586	48	13
240	82	3918	2108	48	18
260	100	4255	4310	42	13
280	111	4513	4759	40	30
300	128	4571	6397	35	26

Halaman ini sengaja dikosongkan



Penulis dilahirkan pada tanggal 17 April 1991 di sebuah rumah sakit umum di Kota Serang. Penulis menyelesaikan pendidikan S1 di jurusan Teknik Elektro FTI ITS dengan bidang studi Teknik Komputer dan Telematika pada tahun 2015. Penulis mengambil tugas akhir berjudul Perilaku Pasukan *Non-Playable Character* Berformasi pada Permainan *Real-Time Strategy*. Penulis melanjutkan Study di kampus yang sama pada Jurusan Teknik elektro FTE ITS dengan bidang minat Jaringan Cerdas Multimedia. Penulis mengambil tesis dengan judul Simulasi Evakuasi Kebakaran Di Dalam Gedung Pada Kerumunan Heterogen Menggunakan *Reciprocal Velocity Obstacle Agent-Based* . Penulis dapat dihubungi

pada email : fadilmuhammad703@gmail.com atau telepon di +6281316627621

Halaman ini sengaja dikosongkan