

**IMPLEMENTASI ALGORITMA *RECIPROCAL VELOCITY*
OBSTACLE UNTUK *CROWD SIMULATION* WISATA
ALUN-ALUN KOTA BATU**

SKRIPSI

**OLEH :
MUHAMMAD FURQON
NIM 14650054**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

**IMPLEMENTASI ALGORITMA *RECIPROCAL VELOCITY OBSTACLE*
UNTUK *CROWD SIMULATION* WISATA ALUN-ALUN KOTA BATU**

SKRIPSI

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :
MUHAMMAD FURQON
NIM. 14650054**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

LEMBAR PERSETUJUAN

IMPLEMENTASI ALGORITMA *RECIPROCAL VELOCITY OBSTACLE* UNTUK *CROWD SIMULATION* WISATA ALUN-ALUN KOTA BATU

SKRIPSI

Oleh :

MUHAMMAD FURQON

NIM. 14650054

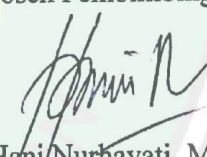
Telah Diperiksa dan Disetujui

Tanggal : Desember 2019

Dosen Pembimbing I


Yunifa Miftachul Arif, M.T
NIP. 198306162011011004

Dosen Pembimbing II


Hani Nurhayati, M.T
NIP. 197806252008012006

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dedyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

IMPLEMENTASI ALGORITMA *RECIPROCAL VELOCITY OBSTACLE* UNTUK *CROWD SIMULATION* WISATA ALUN-ALUN KOTA BATU

SKRIPSI

Oleh:
MUHAMMAD FURQON
NIM. 14650054

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: Desember 2019

Susunan Dewan Penguji

Penguji Utama : Dr. M. Faisal, M.T
NIP. 19740510 200501 1 007

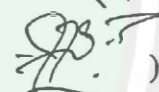
Ketua Penguji : Khadijah F.H. Holle, M.Kom
NIDT. 19900626 20160801 2 077

Sekretaris Penguji : Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004

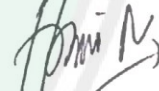
Anggota Penguji : Hani Nurhayati, M.T
NIP. 19780625 200801 2 006

Tanda Tangan

()

()

()

()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini,

Nama : Muhammad Furqon

NIM : 14650054

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : IMPLEMENTASI ALGORITMA RECIPROCAL VELOCITY
OBSTACLE UNTUK CROWD SIMULATION WISATA
ALUN-ALUN KOTA BATU

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 28 Desember 2019

Yang membuat pernyataan,



Muhammad Furqon
NIM. 14650054

MOTTO

“Harapan Itu Selalu Ada, Harapan Yang Membuat Kita Selalu Termotivasi Untuk Bisa Lebih Baik. Maka Pupuklah Mimpi Dan Harapan Beserta Do’a dan Ikhtiar. Insya Allah Apa Yang Kita Impikan Akan Terwujud Meskipun Ujian Datang Berkali-Kali.”

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah rabbil'alamiin, sebuah karya sederhana ini bisa terselesaikan tidak lain dan tidak bukan karena adanya nikmat dan rahmat serta karuniaNya. Hadirnya berbagai bantuan, bimbingan, masukan, serta do'a yang dipanjatkan dari berbagai pihak juga sangat memberi kekuatan bagi saya untuk menyelesaikan skripsi ini. Saya persembahkan sebuah karya tulis sederhana yang masih jauh dari kata sempurna ini kepada:

Kedua orang tua yang amat sangat saya cintai yang telah memberikan dukungan baik berupa doa maupun materil yang terus mengalir, dan tak lupa kedua kakak saya yang telah menyemangati serta memberikan masukan agar tetap semangat dan tidak menyerah serta seluruh keluarga besar Bani Sa'id yang tidak bisa saya sebutkan satu persatu.

Bapak Yunifa Miftachul Arif, M.T dan Hani Nurhayati, M.T selaku dosen pembimbing yang selalu memberikan masukan dan masukan dalam menyelesaikan skripsi ini, Bapak Cahyo Crysdian selaku kajar Teknik Informatika, seluruh dosen Teknik Informatika, Mbak Citra selaku Administrasi Jurusan Teknik Informatika serta seluruh guru-guruku yang telah membimbing dan memberikan ilmu kepada saya.

Seluruh rekan rekan saya di Teknik Informatika 2014 khususnya TI B Arief, Khojen, Gimen, Adam, Ulung, Hilal, Ifa, Rindah, Insan, Yolan, Cicin, yang tak pernah telah menyemangati serta mengingatkan untuk segera menyelesaikan skripsi ini, juga kepada rekan kontrakan keat sentosa, private class.

Rekan Pondok Pesantren Sabilurrosyad khususnya kamar power ranger, rekan rekan sahabat ngopi, teman teman se-Pasuruan yang ada di Malang, teman teman Gasek Keren Sholeh, Gasek wibu sejahtera dan seluruh rekan yang tidak bisa saya sebutkan satu persatu, terima kasih semuanya. *Jazakumullahu ahsanul Jaza.*

Juga kupersembahkan kepada siapapun yang hendak memanfaatkan atau melanjutkan penelitian ini, meskipun memang masih jauh dari kata sempurna dan masih banyak kekurangan, semoga bisa dikembangkan dan bisa bermanfaat bagi masyarakat.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT, karena atas rahmat, hidayah serta karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Algoritma Reciprocal Velocity Obstacle Untuk Crowd Simulation Wisata Alun-Alun Kota Batu” sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Salawat serta Salam senantiasa terlimpahkan kepada Nabi Muhammad SAW, keluarga dan para sahabat yang telah membimbing umat dari gelapnya alam jahiliyah menuju cahaya Islam yang diridoi Allah SWT.

Penulis menyadari adanya banyak keterbatasan yang penulis miliki, sehingga ada banyak pihak yang telah memberikan bantuan baik moril maupun materil dalam menyelesaikan penelitian ini. Maka dari itu dengan segenap kerendahan hati penulis mengucapkan terimakasih kepada:

1. Prof Dr H Abd. Haris, M.Ag selaku rektor UIN Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si. selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
4. Yunifa Miftachul Arif, M.T selaku dosen pembimbing I dan Hani Nurhayati, M.T selaku dosen pembimbing II yang senantiasa meluangkan waktu untuk membimbing, mengarahkan penulis, dan memberi masukan.
5. Seluruh Dosen Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang yang telah memberikan ilmu dan pengetahuan serta pengalaman.

6. Segenap civitas akademik Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
7. Kedua orang tua serta seluruh keluarga besar penulis yang senantiasa mendukung.
8. Sahabat-sahabat seperjuangan Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

Penulis menyadari dalam karya ini masih banyak kekurangan. Oleh karena itu penulis selalu menerima segala kritik dan saran dari pembaca. Semoga karya ini bermanfaat bagi seluruh pihak.

Malang, 28 Desember 2019

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN KEASLIAN TULISAN	iv
MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
ABSTRAK	xiii
ABSTRACT.....	xiv
المخلص	xv
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Tujuan Penelitian.....	4
1.4. BatasandMasalah.....	4
1.5. Manfaat Penelitian.....	5
BAB II.....	6
KAJIAN PUSTAKA.....	6
2.1. Landasan Teori	6
2.1.1. <i>Agent-Based Modelling</i>	6
2.1.2. <i>Crowd Simulation</i>	7
2.1.3. <i>Crowd Behavior</i>	9
2.1.4. <i>Algoritma Reciprocal Velocity Obstacle</i>	14
2.1.5. <i>Tools Unity3D</i>	22
2.1.6. <i>Pariwisata</i>	22
2.1.7. <i>Alun-Alun Kota Batu</i>	24

2.2. Penelitian Terkait	24
BAB III	28
DESAIN DAN RANCANGAN SIMULASI	28
3.1 Tahapan Penelitian	28
3.1.1 Studi literatur.....	28
3.1.2 Desain skenario	29
3.1.3 Desain Agen dan <i>Enviroment</i>	29
3.1.4 Perancangan Simulasi	29
3.1.5 Uji Coba Simulasi	29
3.1.6 Penyusunan Laporan	29
3.2 Desain dan Perancangan Simulasi.....	31
3.3 Implementasi Algoritma <i>Reciprocal Velocity Obstacle</i>	34
3.3.1 Parameter.....	34
3.3.2 Perhitungan RVO.....	35
3.4 Pembuatan Lingkungan Simulasi.....	40
3.4.1 Deskripsi Agen.....	45
3.4.2 Deskripsi Alun-Alun	46
BAB IV	49
HASIL DAN PEMBAHASAN.....	49
4.1 Pengujian Sistem Simulasi	49
4.1.1 Perangkat Keras yang Dibutuhkan.....	49
4.1.2 Perangkat Lunak yang Dibutuhkan.....	49
4.2 Pengujian RVO.....	50
4.2.1 <i>Seeking</i>	50
4.2.2 <i>Steering</i>	50
4.2.3 <i>Collision Avoidance</i>	51
4.2.4 Ekstraksi data	52
BAB V.....	62
PENUTUP.....	62
5.1 Kesimpulan.....	62
5.2 Saran.....	63
Daftar Pustaka	64

DAFTAR GAMBAR

Gambar 2. 1 penghindaran <i>obstacle</i> menggunakan <i>Velocity Obstacle</i>	15
Gambar 2. 2 area <i>velocity obstacle</i>	16
Gambar 2. 3 Osilasi ketika <i>collision avoidance</i> menggunakan <i>Velocity Obstacle</i>	18
Gambar 2. 4 Pergerakan agen yang bebas osilasi	20
Gambar 2. 5 <i>RVOBA</i> (v_b, v_a) dari agen B ke agen A.....	21
Gambar 2. 6 Multi agen navigasi menggunakan RVO	22
Gambar 3. 1 Diagram alur penelitian	28
Gambar 3. 2 <i>Flowchart</i> Alur pergerakan agen.....	31
Gambar 3. 3 <i>Flowchart Collision detection</i>	33
Gambar 3. 4 Dua agen yang bergerak berlawanan	37
Gambar 3. 5 Agen merah menghitung wilayah <i>collider</i>	38
Gambar 3. 6 VO pada agen merah terhadap agen biru	38
Gambar 3. 7 RVO pada agen merah kepada agen biru	39
Gambar 3. 8 RVO dari agen merah dengan beberapa agen biru.....	39
Gambar 3. 9 Area <i>Cross Scenario</i>	40
Gambar 3. 10 Area <i>Narrow Passage Scenario</i>	41
Gambar 3. 11 Area <i>walkable</i> untuk agen	42
Gambar 3. 12 Area skenario alun-alun Kota Batu	43
Gambar 3. 13 bentuk agen dalam simulasi	45
Gambar 3. 14 Gambar <i>collider</i> yang berada pada tubuh agen	46
Gambar 3. 15 area <i>walkable</i>	46
Gambar 4. 1 <i>seeking</i>	50
Gambar 4. 2 <i>pengujian agen menuju titik tujuannya</i>	51
Gambar 4. 3 <i>pengujian collision avoidance</i>	51
Gambar 4. 4 Grafik total Collision pada cross scenario.....	53
Gambar 4. 5 Grafik waktu simulasi	53
Gambar 4. 6 Grafik rata-rata <i>framerate</i>	54
Gambar 4. 7 Grafik total collision pada <i>Narrow Passage Scenario</i>	55
Gambar 4. 8 Grafik waktu simulasi	56
Gambar 4. 9 Grafik Rata-rata <i>frame</i>	56
Gambar 4. 10 Grafik total collision pada alun-alun scenario.....	57
Gambar 4. 11 Waktu simulasi dari alun-alun skenari	58
Gambar 4. 12 Rata-rata Frame rate dari alun-alun simulasi	59

DAFTAR TABEL

Tabel 3. 1 Tabel 3.1 Titik awal dan tujuan dalam cross scenario	41
Tabel 3. 2 Tabel titik awal dan tujuan agen	43
Tabel 3. 3 Titik awal dan destinasi dari alun-alun skenario.....	44
Tabel 4. 1 Ekstraksi Data	52



ABSTRAK

Furqon, Muhammad. 2019. **Implementasi Algoritma *Reciprocal Velocity Obstacle* Untuk *Crowd Simulation* Wisata Alun-Alun Kota Batu**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Yunifa Miftachul Arif, M.T (II) Hani Nurhayati, M.T

Kata Kunci : Alun-alun Kota Batu, Simulasi kerumunan, RVO(*Reciprocal Velocity Obstacle*).

Batu dikenal sebagai Kota yang kaya akan tempat pariwisata. Salah satu tempat wisata yang sering dikunjungi di Kota Batu adalah Alun-alun Batu karena selain gratis, alun-alun Kota Batu juga berada pada tempat yang strategis. Di alun-alun Kota Batu sering terjadi kepadatan pengunjung hal ini dikarenakan banyaknya wisatawan yang berkunjung ke alun-alun Kota batu pada waktu yang sama. Pada era yang serba modern ini segala sesuatu bisa ditampilkan dengan komputerisasi dalam bentuk visual. Untuk mengetahui kepadatan yang terjadi pada alun-alun Kota Batu maka dilakukan visualisasi dalam bentuk simulasi kerumunan.

Simulasi kerumunan merupakan salah satu tema penelitian yang digunakan dalam bidang navigasi multi agen. Kepadatan agen serta banyaknya *obstacle* dapat menyebabkan masalah dalam pergerakan kerumunan agen. Oleh sebab itu dibutuhkan adanya suatu metode navigasi yang dapat digunakan untuk menavigasi agen agar dapat mencapai titik tujuan dengan aman.

Salah satu metode dalam navigasi multi agen adalah *Reciprocal Velocity Obstacle* (RVO), RVO memungkinkan agen bergerak tanpa adanya komunikasi eksplisit dengan agen lain dalam melakukan penghindaran tabrakan serta bebas dari gerakan osilasi yang terjadi pada *Velocity Obstacle*. Dengan adanya RVO diharapkan dapat memberikan kontribusi untuk pengembangan deteksi tabrakan antar objek serta dapat digunakan sebagai salah satu metode simulasi kerumunan (*Crowd Simulation*).

Hasil dari penelitian ini menunjukkan bahwa tabrakan akan terjadi pada tingkat jumlah kerumunan tertentu. Hal ini dipengaruhi oleh kepadatan kerumunan yang tinggi sehingga menghasilkan perhitungan yang lebih kompleks untuk navigasi agen. Sedangkan waktu yang diperlukan untuk melakukan satu periode simulasi akan terus meningkat seiring dengan banyaknya jumlah agen yang disimulasikan, selain itu luas area simulasi dan banyaknya jumlah *obstacle* juga mempengaruhi waktu yang diperlukan untuk melakukan simulasi.

ABSTRACT

Furqon, Muhammad. 2019. **Implementation of Reciprocal Velocity Obstacle Algorithm for Crowd Simulation in Batu Town Square**. Undergraduate Thesis. Departement of Informatics Engineering. Faculty of Science and Technology Islamic State Maulana Malik Ibrahim Malang. Supervisor : (I) Yunifa Miftachul Arif, M.T (II) Hani Nurhayati, M.T

Keyword : Batu Town Square, Crowd Simulation, RVO(*Reciprocal Velocity Obstacle*).

Batu is known as a city that is rich in tourism. One of the tourist attractions that are often visited in Batu City is Batu Town Square because it is free, Batu Town Square is also in a strategic place. In Batu Town Square there is often a density of visitors this is due to the large number of tourists visiting Batu Town Square at the same time. In this modern era everything can be displayed by computerization in visual form. To find out the density that occurred in Batu Town Square, visualization was done in the form of crowd simulation.

Crowd simulation is one of the research themes used in the multi agent navigation field. The density of the agent and the number of obstacles can cause problems in the movement of the agent crowd. Therefore we need a navigation method that can be used to navigate agents to reach their destination safely.

One method in multi-agent navigation is Reciprocal Velocity Obstacle (RVO), RVO allows agents to move without explicit communication with other agents in collision avoidance and free from oscillation movements that occur in Velocity Obstacle. With the RVO is expected to contribute to the development of collision detection between objects and can be used as a crowd simulation method (Crowd Simulation).

The results of this study indicate that collisions will occur at a certain level of crowd size. This is influenced by a high crowd density resulting in more complex calculations for agent navigation. While the time needed to carry out a simulation period will continue to increase along with the number of agents being simulated, besides the area of the simulation and the large number of obstacles also affect the time needed to do the simulation.

الملخص

فرقان ، محمد. ٢٠١٩. تنفيذ خوارزمية عقبة السرعة المتبادلة لمحاكاة الحشود في ساحة مدينة باتو. البحث العلمي قسم هندسة المعلوماتية. كلية العلوم والتكنولوجيا . جامعة مولانا الك إبراهيم الاسلامية الحكومية مالانج. المشرف : (١) يوننفا مفتخ الاريف الما جستي, (٢) هاني نور حيا تي الما جيستير.

الكلمات المفتاحية : ساحة مدينة باتو, محاكاة الحشد, RVO (عقبة السرعة المتبادلة) واحدة من مناطق الجذب السياحي التي يتم زيارتها Alun-alun باتو معروفة كمدينة غنية بالسياحة. تعد في مكان باتو مدينة Alun-alun لأنها مجانية ، كما يقع ميدان باتو في كثير من الأحيان في مدينة استراتيجي. في ساحة كوتا باتو ، غالبًا ما يكون هناك كثافة من الزوار وذلك بسبب العدد الكبير من السياح الذين يزورون ساحة كوتا باتو في نفس الوقت. في هذا العصر الحديث ، يمكن عرض كل شيء عن مدينة باتو ، تم التصور على طريق الحوسبة في شكل بصري. لمعرفة الكثافة التي حدثت في ساحة شكل محاكاة للحشود

الحشود هي واحدة من الموضوعات البحثية المستخدمة في مجال الملاحة متعدد الوكلاء. يمكن أن تسبب كثافة العامل وعدد العقبات مشاكل في حركة حشد الوكيل. لذلك نحتاج إلى طريقة تنقل يمكن استخدامها للتنقل بين الوكلاء للوصول إلى وجهتهم بأمان

RVO طريقة واحدة في الملاحة متعددة الوكلاء هي Reciprocal Velocity Obstacle (RVO) تسمح للعاملين بالتحرك دون اتصال واضح مع الوكلاء الآخرين في تجنب الاصطدام وخالية من في تطوير . RVO من المتوقع أن يساهم Velocity Obstacle. حركات التذبذب التي تحدث في اكتشاف التصادم بين الكائنات ويمكن استخدامه كأسلوب محاكاة الحشد

تشير نتائج هذه الدراسة إلى أن الاصطدامات ستحدث عند مستوى معين من حجم الحشود. يتأثر هذا بكثافة حشد عالية مما يؤدي إلى عمليات حسابية أكثر تعقيدًا للتنقل عبر الوكيل. في حين أن الوقت اللازم لتنفيذ فترة المحاكاة سيستمر في الزيادة إلى جانب عدد العوامل المحاكاة ، إلى جانب مساحة المحاكاة وعدد كبير من العقبات تؤثر أيضًا على الوقت اللازم لإجراء المحاكاة

BAB I

PENDAHULUAN

1.1. Latar Belakang

Batu dikenal sebagai Kota yang kaya akan potensi pariwisata. Pariwisata merupakan suatu perjalanan yang dilakukan untuk sementara waktu dari suatu tempat ke tempat lain dengan maksud tidak untuk mencari nafkah di tempat yang dikunjungi tapi hanya semata untuk menikmati perjalanan tersebut untuk mencapai kepuasan (UU Kepariwisata, 1990).

Kota Batu merupakan Kota yang sangat diminati wisatawan baik lokal maupun mancanegara. Hal ini dapat diketahui dari data jumlah wisatawan yang mengunjungi Kota Batu pada 2017 mencapai 4,7 juta (Dinas Pariwisata Kota Batu, 2018) data tersebut melampaui target yang diperkirakan hanya sampai 4,2 juta. Salah satu destinasi wisata Batu yang banyak digemari adalah alun-alun Kota Batu. Karena selain lokasinya yang strategis, untuk menikmati alun-alun Batu tidak dipungut biaya. Selain itu, alun-alun Kota Batu juga dapat digunakan sebagai sarana rekreasi, edukasi dan olahraga.

Dalam berpariwisata, hal yang sering dialami oleh wisatawan adalah kurangnya pengetahuan mengenai tempat wisata yang akan dikunjungi. Oleh karena itu dibutuhkan sebuah sistem visualisasi untuk mempermudah wisatawan dalam memahami tempat wisata yang akan dikunjungi. Karena sebagian besar seseorang lebih mudah mengingat sesuatu hal ketika ia melihat dan terlibat secara langsung dibandingkan dengan jika ia membaca atau mendengarnya dari orang lain (Muhammad Badrul, 2017).

Salah satu cara untuk memvisualisasikan pengunjung wisata adalah dengan melakukan simulasi kerumunan. Simulasi kerumunan merupakan suatu pemodelan gerakan dari sekumpulan entitas bergerak yang di sebut agen dalam jumlah besar dengan kepadatan tinggi (Moch Fahri, 2016). Simulasi kerumunan digunakan untuk melakukan peniruan terhadap pengunjung dalam suatu tempat wisata sebagai langkah optimasi mencari solusi kemungkinan terbaik. (F. Feurtey, 2000) dan (Y Matsuo dkk, 2001) merupakan peneliti yang mempopulerkan simulasi kerumunan. Namun, dalam penelitian yang mereka lakukan terdapat beberapa kelemahan seperti masing-masing agen tidak dapat memperhitungkan bahwa agen lain juga memiliki kemampuan pengambilan keputusan yang sama untuk menghindari tabrakan, sehingga gerakan yang dihasilkan cenderung mengandung osilasi yang tidak diinginkan dan tidak realistis.

Berdasarkan permasalahan yang sudah diuraikan oleh penulis tersebut dibutuhkan sebuah metode simulasi kerumunan sebagai langkah mencari solusi kemungkinan terbaik. Terdapat beberapa metode yang digunakan dalam simulasi kerumunan seperti *Algoritma Menge*, *A star*, *Velocity Obstacle(VO)*, *Optimal Reciprocal Collision Avoidance(ORCA)*, dan *Reciprocal Velocity Obstacle (RVO)*. Namun dari beberapa metode tersebut yang memiliki kelebihan dalam melakukan gerakan yang bebas osilasi adalah RVO. Algoritma RVO memperhitungkan perilaku reaktif dari agen dengan mengasumsikan bahwa agen lain melakukan perilaku penghindaran tabrakan yang sama agar menghasilkan gerakan agen yang bebas dari tabrakan dan osilasi. Dengan adanya metode *Reciprocal Velocity Obstacle* diharapkan dapat menangani masalah menavigasi yang aman dan bebas osilasi di antara banyak agen dalam tempat wisata.

Di dalam Al-Quran diperoleh banyak isyarat untuk melakukan aktivitas pariwisata. Pariwisata sebagai salah satu sektor yang bisa mendatangkan pendapatan individu, masyarakat dan pendapatan bagi negara. Dalam surah Yusuf ayat 109 Allah SWT berfirman :

وَمَا أَرْسَلْنَا مِنْ قَبْلِكَ إِلَّا رِجَالًا نُوحِي إِلَيْهِمْ مِنْ أَهْلِ الْقُرَىٰ ۖ أَفَلَمْ يَسِيرُوا فِي الْأَرْضِ فَيَنْظُرُوا كَيْفَ

كَانَ عَاقِبَةُ الَّذِينَ مِنْ قَبْلِهِمْ ۖ وَلَنَارُ الْآخِرَةِ خَيْرٌ لِلَّذِينَ اتَّقَوْا ۖ أَفَلَا تَعْقِلُونَ

Artinya :*" Kami tidak mengutus sebelum kamu, melainkan orang laki-laki yang Kami berikan wahyu kepadanya diantara penduduk negeri. Maka tidakkah mereka bepergian di muka bumi lalu melihat bagaimana kesudahan orang-orang sebelum mereka (yang mendustakan rasul) dan sesungguhnya kampung akhirat adalah lebih baik bagi orang-orang yang bertakwa. Maka tidakkah kamu memikirkannya?"*.

Maksud dari ayat tersebut adalah menganjurkan agar menelusuri berbagai negeri. Ini merupakan isyarat bahwa manusia perlu mengenal budaya masyarakat dari berbagai belahan dunia. Kebudayaan suatu masyarakat tidak dapat dipahami secara holistik tanpa terlibat langsung di dalamnya. Ayat ini juga menegaskan betapa banyak umat terdahulu telah dibinasakan Allah karena kedurhakaan mereka kepada ayat-ayatnya. Karena itu manusia harus menggunakan daya pikirnya untuk menemukan kebenaran, dan sekaligus meyakini kebenaran berita yang disampaikan Al-Quran.

Penelitian ini bertujuan untuk mengimplementasikan simulasi kerumunan kedalam model 3D. Karena penulis beranggapan bahwa simulasi akan menarik dan realistis ketika diimplementasikan kedalam suatu *crowd simulation*.

1.2. Rumusan Masalah

Dengan memperhatikan permasalahan yang sudah diuraikan diatas maka ditentukan rumusan masalah dalam penelitian ini: Bagaimana membuat pergerakan agen agar dapat menghindari *obstacle* berupa objek diam maupun bergerak sehingga dapat saling menghindari tabrakan antara satu agen dengan yang lainnya menggunakan Algoritma RVO (*Reciprocal Velocity Obstacle*).

1.3. Tujuan Penelitian

Dengan memperhatikan rumusan masalah di atas maka tujuan dari penelitian ini adalah: menghasilkan pergerakan agen agar dapat menghindari *obstacle* berupa objek diam maupun bergerak sehingga dapat saling menghindari tabrakan antara satu agen dengan yang lainnya menggunakan Algoritma RVO (*Reciprocal Velocity Obstacle*).

1.4. Batasan Masalah

Dalam penelitian ini batasan masalah digunakan untuk menghindari pembahasan yang tidak sesuai dengan topik penelitian. Adapun batasan-batasan tersebut sebagai berikut:

1. Simulasi hanya dibatasi pada kasus pergerakan agen di alun-alun Batu.
2. Jumlah maksimum agen dalam simulasi ini dibatasi 200 agen.
3. Target yang dituju pada simulasi ini adalah *waypoint-waypoint* yang sudah ditentukan.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dari hasil penelitian ini adalah memberikan kontribusi untuk pengembangan deteksi tabrakan antar objek atau perilaku sekelompok agen dalam bidang animasi komputer, serta dapat digunakan sebagai salah satu metode simulasi kerumunan (*crowd simulation*).



BAB II

KAJIAN PUSTAKA

2.1. Landasan Teori

2.1.1. *Agent-Based Modelling*

Simulasi komputer merupakan suatu program komputer yang digunakan untuk meniru suatu kejadian atau sistem tertentu. Melalui simulasi konsep - konsep yang bersifat abstrak dapat di visualisasikan dalam bentuk gambar dan animasi. Melalui simulasi juga user dapat langsung berinteraksi, melakukan percobaan dan langsung melihat pengaruh yang ditimbulkan pada sebuah sistem. Simulasi berbasis agen (*Agent-based Modeling*) merupakan salah satu metode yang relatif baru dalam mengembangkan simulasi. Simulasi berbasis agen telah digunakan secara luas oleh para peneliti dalam mempelajari dan memvisualisasi fenomena seperti interaksi individu, reaksi kimia dan perilaku serangga.

Kelebihan dari *agent-based modeling* terletak pada kemampuannya memodelkan sistem dunia nyata yang semakin kompleks. *Agent-based modeling* juga dapat menghasilkan perilaku sistem yang kompleks dimana perilaku ini dihasilkan dari interaksi agen - agen sederhana didalamnya. *Agent-Based Modeling* suatu sistem dimodelkan dengan kumpulan entitas - entitas yang dapat mengambil keputusan secara otonom, entitas ini disebut 'agen'. Setiap agen bertindak dan berperilaku berdasarkan aturan tertentu di dalam lingkungannya (Muhammad Badrul, 2017).

2.1.2. Crowd Simulation

Menurut Siamak Sarmady (2008), “*Crowd* (kerumunan) dapat didefinisikan sebagai kelompok pejalan kaki yang berkumpul di tempat yang sama atau kadang-kadang mempunyai tujuan yang berbeda”. Menurut Ming-Liang Xu (2014), kerumunan adalah sistem yang kompleks yang mengandung koleksi individu, seperti kelompok manusia, ternak hewan, kawanan serangga, dan arus kendaraan di lingkungan fisik yang sama. Mereka memiliki perilaku yang berbeda ketika dalam kerumunan jika dibandingkan saat sendiri.

Simulasi kerumunan dapat dibagi menjadi dua bidang yaitu berfokus pada aspek perilaku realisme dan visualisasi berkualitas tinggi. Pada fokus yang pertama, tuntutan penelitian lebih sederhana. Hasil penelitian biasanya berbentuk visualisasi 2 dimensi seperti simulator evakuasi, model kerumunan sosiologis, atau kerumunan dinamika model. Untuk cakupan penelitian tersebut perilaku simulasi biasanya sempit, rentang area penelitiannya dikendalikan (misalnya, orang pergi keluar atau orang yang membentuk struktur lingkaran kerumunan) dengan adanya upaya validasi korespondensi dari hasil pengamatan dunia nyata untuk situasi tertentu. Visualisasi digunakan untuk membantu memahami hasil simulasi. Sementara dalam fokus kedua yaitu memusatkan penekanan pada kualitas rendering yang tinggi dan teknik animasi untuk mencapai efek visual yang nyata. Penerapan kasus ini mengarah pada produksi film dan video *game* Sehingga perilaku model yang nyata tidaklah penting (Daniel Thalmann 2007).

Menurut (Wolinski, 2016) lingkup simulasi kerumunan dikelompokkan menjadi dua basis, yakni makrokopis (*continuum based*) dan mikrokopis (*agent based*). Lingkup makrokopis adalah simulasi kerumunan yang berada dalam level

global atau dalam jumlah agen yang sangat besar. Selain itu digunakan untuk keperluan statistik seperti arus dan kepadatan agen dalam simulasi. Simulasi yang berbasis makroskopis tidak terfokus pada interaksi antar agen. Sedangkan mikroskopis merupakan kerumunan yang terbentuk berbasis agen yang saling berinteraksi. Salah satu contoh algoritma yang berada dalam lingkup mikroskopis adalah Algoritma *boid* milik *Reynolds*. Dimana agen *boid* berinteraksi dengan agen *boid* lainnya menggunakan tiga aturan *Reynolds*. Aturan tersebut ialah *cohesion*, *alignment* dan *separation*. Mensimulasikan kerumunan manusia seperti di dunia nyata menjadi sesuatu kebutuhan pemodelan yang interaktif dan realistis. Simulasi kerumunan yang *real-time* ini sulit dilakukan karena perilaku yang dipamerkan pada kelompok besar ini sangat kompleks dan rumit.

Sebuah model kerumunan tidak hanya meliputi pergerakan individu manusia dalam lingkungannya dengan rintangan / batasan yang ada, akan tetapi juga interaksi dinamis antara orang-orangnya. Lebih jauh lagi, model harus dapat mencerminkan *intelligent path planning* melalui perubahan lingkungannya. Dan selanjutnya, orang-orang yang berada di dalamnya harus dapat menyesuaikan jalan mereka secara dinamis. Pada prinsipnya pergerakan kerumunan adalah bagaimana penanganan terhadap permasalahan *path planning* (Perencanaan Jalur), *collision avoidance* (penghindaran tabrakan), serta *separation* (pemisahan jarak). Jika hanya menggunakan penghindaran tabrakan, maka model kerumunan yang nyata tidak akan dapat dihasilkan saat sekelompok orang tersebut mencapai tujuan. Konsekuensinya, banyak model kerumunan yang menggabungkan antara menghindari tabrakan (*collision*

avoidance) dan perencanaan global (*global navigation*) (Arik kurniawati dkk, 2010).

2.1.3. Crowd Behavior

(Krisna, 2009) Dalam konsep ilmu sosial istilah kerumunan menjadi penting setelah Le Bon menerbitkan buku *The Crowd: A study of the Popular Mind*. Le Bon berpendapat bahwa dalam pengertian sehari-hari istilah kerumunan berarti sejumlah individu yang berkumpul bersama, namun dari segi psikologis istilah kerumunan mempunyai makna sekumpulan orang yang mempunyai ciri baru yang berbeda yaitu berhaluan sama dan kesadaran perseorangan lenyap dan terbentuknya satu makhluk tunggal kerumunan terorganisasi (*organized crowd*) atau kerumunan psikologis (*psychological crowd*). Sederhananya kita bersandar pada definisi yang diberikan Yusron Razak bahwa kerumunan adalah kumpulan orang yang bersifat sementara dan yang memberikan reaksi secara bersama terhadap suatu rangsangan. Terdapat beberapa faktor yang menyebabkan terjadi perilaku kerumunan yaitu:

- a. Anonimitas. Karena faktor kebersamaan dengan berkumpulnya individu-individu yang semula dapat mengendalikan diri, merasa dapat kekuatan luar biasa yang mendorongnya untuk tunduk pada dorongan naluri dan terlebur dalam kerumunan sehingga perasaan menyatu dan tidak dikenal mampu melakukan hal hal yang tidak bertanggung jawab. Semakin tinggi kadar anonimitas suatu kerumunan, semakin besar pula kemungkinannya untuk menimbulkan tindakan ekstrim karena anonimitas mengikis rasa individualitas para anggota kerumunan itu.

- b. *Contagion* (penularan). Penularan Sosial (*social contagion*), adalah penyebaran suasana hati, perasaan atau suatu sikap, yang tidak rasional, tanpa disadari dan secara relatif berlangsung cepat. Penularan ini oleh Le Bon dapat dianggap suatu gejala hipnotis. Individu yang telah tertular oleh perasaan dan tindakan orang lain sudah tidak memikirkan kepentingan individu melainkan kepentingan bersama.
- c. Konvergensi (keterpaduan). Orang-orang yang akan menonton festival musik Pop, dengan orang-orang yang menonton festival musik Rock akan memiliki ciri-ciri yang berbeda. Orang-orang yang menonton festival musik rock cenderung akan lebih mudah menimbulkan keributan dibanding dengan orang-orang yang menonton festival musik Pop.
- d. *Suggestibility* (mudahnya dipengaruhi). Kerumunan biasanya tidak berstruktur, tidak dikenal adanya pemimpin yang mapan atau pola perilaku yang dapat dipanuti oleh para anggota kerumunan itu sehingga dalam suasana seperti itu, orang berperilaku tidak kritis dan menerima saran begitu saja, terutama jika saran itu meyakinkan dan bersifat otoritatif. Akan tetapi siapa induk atau yang memulai sulit ditentukan.

2.1.3.1 *Path Finding*

Pergerakan dalam suatu simulasi sangat erat kaitannya dengan *path finding* (pencarian jalan). *Path finding* adalah salah satu dasar algoritma dalam konsep menggerakkan karakter. Pergerakan obyek menuju sasaran dengan mengambil nilai *potential field* (Arik kurniawati dkk, 2010). Metode *path finding* mudah diketahui pada game-game bertipe strategi dimana kita menunjuk suatu tokoh untuk digerakkan ke lokasi tertentu dengan mengklik lokasi yang di inginkan.

Tokoh tersebut akan bergerak ke arah yang di telah ditentukan, dan secara cerdas dapat menemukan jalur terpendek dan menghindari dari rintangan - rintangan yang ada.

2.1.3.2 Collision Detection

Collision detection atau pendeteksian tabrakan adalah proses pengecekan apakah beberapa buah objek spasial saling bertabrakan atau tidak. Jika ternyata ada paling sedikit dua buah objek yang bertabrakan, maka kedua objek tersebut dikatakan saling bertabrakan. Pada ruang spasial dua dimensi, Objek yang bertumpuk berarti objek spasialnya beririsan. Teknik pendeteksian tabrakan bisa dikelompokkan menjadi dua macam yaitu *priori detection* dan *post detection*. *Priori detection* adalah pengecekan tabrakan sebelum tabrakan tersebut terjadi, sedangkan *post detection* adalah pengecekan tabrakan setelah tabrakan tersebut terjadi (Riski Maulana, 2010).

Objek-objek ini bisa saja memiliki bentuk yang sangat bervariasi, ada yang berbentuk kotak, segi-n, sampai bentuk pesawat pemain yang sangat mendetail. Untuk mempercepat proses pada collision detection, umumnya objek-objek ini direpresentasikan secara logik dengan bentuk primitif seperti segiempat dan lingkaran (jika pada koordinat dua dimensi), atau kubus dan bola (jika pada koordinat tiga dimensi). Bentuk primitif yang merepresentasikan objek ini biasa disebut sebagai *Bounding Box* atau *Bounding Circle*.

2.1.3.3 Collision Avoidance

Collision avoidance atau penghindaran benturan adalah masalah mendasar dalam robotika. secara umum dapat didefinisikan sebagai robot bergerak secara otonom yang dapat menavigasi di lingkungan dengan adanya rintangan atau entitas bergerak lainnya, di mana robot menggunakan siklus penginderaan dan perilaku terus menerus. Dalam setiap siklus, tindakan untuk robot harus dihitung berdasarkan pengamatan lokal terhadap lingkungan, sehingga robot tetap bebas dari tabrakan dengan rintangan dan entitas bergerak lainnya sambil bergerak menuju tujuannya.

Penghindaran tabrakan antar orang ternyata dapat menimbulkan banyak permasalahan ketika berurusan dalam kerumunan. Sebuah metode untuk menghindari tabrakan antar individu menjadi tidak efisien ketika digunakan dalam kerumunan banyak orang. Ada lebih banyak kendala yang terjadi dalam sebuah variabel lingkungan yang kompleks (dengan *obstacle* yang tetap, rintangan *mobile*/bergerak dan daerah yang kecil untuk berjalan) dan mencakup banyak orang. Namun, jika struktur grup dalam kelompok harus dipertahankan, maka harus ada penambahan parameter lain dalam kerumunan kompleksitas untuk menghindari tabrakan. Walaupun *collision avoidance* (penghindaran tabrakan) terlihat begitu kompleks dalam realitanya, namun hal ini dapat didefinisikan menjadi beberapa aturan yang sederhana (Arikurniawati, 2010).

Berikut ini adalah cara-cara bagaimana mengantisipasi berbagai jenis tabrakan:

1. *Toward collisions*

Tahap pertama adalah menentukan apakah akan bertabrakan di sebelah kiri atau kanan dari seseorang. Pengamatan menunjukkan bahwa seseorang akan memiliki tiga cara yang berbeda untuk menghindari tabrakan :

- a. Mengubah arah saja
- b. Mengubah kecepatan saja
- c. Mengubah arah dan kecepatan

Jika tidak ada perilaku yang ditemukan untuk menghindari tabrakan, maka seseorang tersebut hanya akan berhenti berjalan. Setelah orang lain keluar dari jalur tabrakan maka seseorang tersebut akan melanjutkan jalannya.

2. *Away Collisions*

Jika seseorang berjalan di belakang orang lain atau dengan kata lain seseorang tersebut tepat di belakang orang lain, sehingga akan terjadi tabrakan dengan orang lain yang ada di depannya. Untuk mengatasi situasi ini, maka seseorang tersebut ada dua pilihan:

- a. Menyesuaikan kecepatan dengan orang yang ada di depannya dengan memperlambat kecepatan berjalan.
- b. Berjalan dengan kecepatan tinggi dengan mengambil jalur di samping depan orang lain.

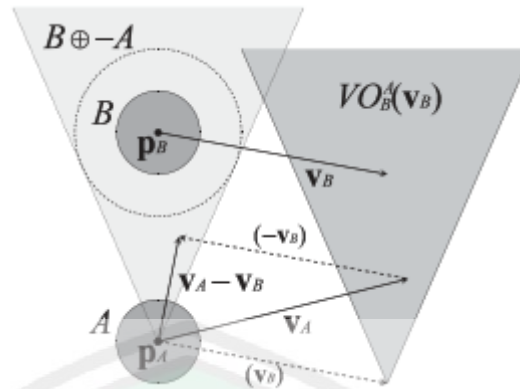
2.1.4. Algoritma *Reciprocal Velocity Obstacle*

2.1.4.1 *Velocity Obstacle*

Dalam robotika dan perencanaan gerak, *velocity obstacle* atau biasa disingkat VO adalah himpunan semua kecepatan agen yang akan menghasilkan tabrakan dengan agen lain, dengan asumsi bahwa agen lain mempertahankan kecepatannya saat ini. Jika agen memilih kecepatan di dalam *velocity obstacle* maka kedua agen itu akan bertabrakan, untuk itu agen harus memilih kecepatan baru di luar *velocity obstacle*, agar tabrakan itu tidak terjadi.

(Fiorini dan Shiller 1998) merupakan yang pertama yang memperkenalkan pendekatan ini untuk digunakan dalam perencanaan jalan dan navigasi *robot mobile* di lingkungan yang dinamis yang terdiri dari rintangan statis dan rintangan bergerak. Mereka menyebutkan Fitur utama dari VO (*Velocity Obstacle*) sebagai berikut :

- a. VO memberikan representasi geometrik sederhana dari manuver penghindaran potensial.
- b. Sejumlah rintangan yang bergerak dapat dihindari dengan mempertimbangkan VO mereka.
- c. menyatukan penghindaran bergerak serta *obstacle* stasioner.
- d. memungkinkan untuk pertimbangan sederhana dari dinamika robot.



Gambar 2. 1 penghindaran *obstacle* menggunakan *Velocity Obstacle*

Gambar di atas menunjukkan konsep penghindaran *obstacle* dengan menggunakan *Velocity Obstacle* yang di perkenalkan oleh Jur van der Berg, dkk (2008) sebagai berikut. Misalkan agen A adalah agen yang sedang bertraslasi di bidang dengan titik referensinya pada $\mathbf{p}_A$, dan B adalah *obstacle* planar (bergerak) dengan titik referensinya pada $\mathbf{p}_B$. *Velocity Obstacle* $VO_B^A(v_B)$ dari rintangan B ke agen A adalah himpunan yang terdiri dari semua kecepatan v_A untuk A yang akan menghasilkan tabrakan pada suatu waktu bersamaan dengan *obstacle* B yang bergerak dengan kecepatan v_B .

Velocity Obstacle dari agen A terhadap rintangan B tersebut dapat didefinisikan secara geometri sebagai berikut:

$$A \oplus B = \{a+b \mid a \in A, b \in B\}, -A = \{-a \mid a \in A\}$$

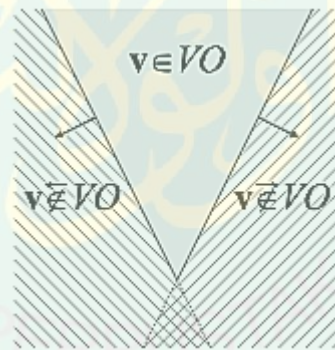
$A \oplus B$ merupakan penjumlahan Minkowski dari dua objek A dan B dimana $-A$ adalah refleksi dari objek A terhadap titik referensinya. Misal $\lambda(\mathbf{p}, \mathbf{v})$ menyatakan ray yang bermula di $\mathbf{p}$ dan bergerak ke arah $\mathbf{v}$:

$$\lambda(p, v) = \{p + tv | t \geq 0\}$$

Jika *ray* yang bermula pada P_A dan mengarah ke kecepatan relatif dari A dan B ($v_A - v_B$) memotong jumlah Minkowski dari B dan A yang berpusat di P_B , *Velocity obstacle* v_A berada di dalam *Velocity obstacle* dari B, sehingga *Velocity obstacle* B ke A didefinisikan sebagai :

$$VO_B^A(v_B) = \{v_A | \lambda(p_A, v_A - v_B) \cap B \oplus -A \neq \emptyset\}.$$

Yang berarti bahwa jika $v_A \in VO_B^A(v_B)$, A dan B akan bertabrakan pada suatu titik dan waktu tertentu. Jika v_A berada diluar *velocity obstacle* dari B, maka kedua agen tidak akan bertabrakan namun jika v_A ada pada perbatasan *velocity obstacle*, maka A akan bersentuhan dengan B pada suatu momen waktu. Seperti pada gambar berikut.



Gambar 2. 2 area *velocity obstacle*

Konsep *Velocity Obstacle* dapat digunakan untuk navigasi multiagen dengan tiap agen menganggap agen lain sebagai rintangan bergerak dan memilih kecepatan untuk dirinya sendiri yang berada diluar *Velocity Obstacle* yang diinduksikan dari agen lain. Namun, pendekatan ini menghasilkan pergerakan osilasi yang tidak diinginkan.

2.1.4.2 Sifat- sifat *Velocity Obstacle*

Menurut (Jur van der Berg, dkk, 2008) *Velocity obstacle* memiliki beberapa sifat dasar, diantaranya:

- a. Teorema 1 (*Symetry*)

$$\mathbf{v}_A \in VO_B^A(\mathbf{v}_B) \Leftrightarrow \mathbf{v}_B \in VO_A^B(\mathbf{v}_A).$$

- b. Teorema 2 (*Traslasi*)

$$\mathbf{v}_A \in VO_B^A(\mathbf{v}_B) \Leftrightarrow \mathbf{v}_B \in VO_A^B(\mathbf{v}_A).$$

Dengan mempertimbangkan wilayah di luar dari *velocity obstacle*, maka dapat dibedakan wilayah ke kiri dan wilayah ke kanan dari *velocity obstacle*.

$$\mathbf{v}_A \notin \overleftarrow{VO}_B^A(\mathbf{v}_B)$$

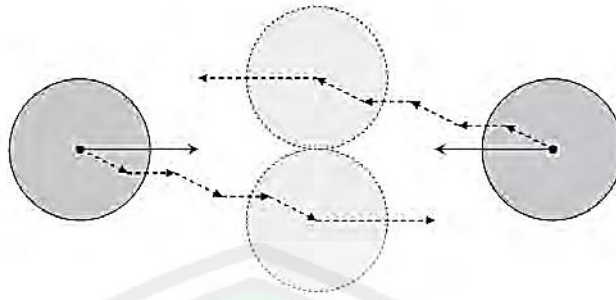
Jika $\mathbf{V}_A$ berada pada setengah bidang di sebelah kiri $\mathbf{VO}$ maka kecepatan seperti itu membiarkan A melewati B di sisi kiri. Begitupun sebaliknya jika $\mathbf{V}_A$ berada pada setengah bidang di sebelah kanan $\mathbf{VO}$ maka kecepatan seperti itu membiarkan A melewati B di sisi kiri

$$\mathbf{v}_A \notin \overrightarrow{VO}_B^A(\mathbf{v}_B)$$

- c. Terema 3 (*konveksitas*)

$$\mathbf{v}_A \notin \overrightarrow{VO}_B^A(\mathbf{v}_B) \wedge \mathbf{v}'_A \notin \overrightarrow{VO}_B^A(\mathbf{v}_B) \Rightarrow (1 - \alpha)\mathbf{v}_A + \alpha\mathbf{v}'_A \notin \overrightarrow{VO}_B^A(\mathbf{v}_B), \text{ for } 0 \leq \alpha \leq 1.$$

2.1.4.3 Osilasi



Gambar 2. 3 Osilasi ketika *collison avoidance* menggunakan *velocity obstacle*

Misalkan terdapat dua agen A dan B bergerak bersamaan dengan kecepatan masing-masing v_A dan v_B , sehingga $v_A \in VO_B^A(v_B)$ dan $v_B \in VO_A^B(v_A)$. Karenanya, Jika kedua agen tetap mempertahankan kecepatan saat ini maka akan menghasilkan tabrakan antara keduanya. Untuk menghindarinya, agen A memutuskan untuk mengubah kecepatannya menjadi v'_A yang berada pada luar *velocity obstacle* B ($v_A \notin VO_B^A(v_B)$). Pada waktu yang sama, agen B mengubah kecepatannya menjadi v'_B sehingga berada diluar *velocity obstacle* A ($v_B \notin VO_A^B(v_A)$).

Masalah muncul ketika kedua agen berada pada kecepatan lama v_A dan v_B , kecepatan lama (v_A dan v_B) masing-masing berada di luar *velocity obstacle* B dan A (misal $v_A \notin VO_B^A(v'_B)$ dan $v_B \notin VO_A^B(v'_A)$). Jika kedua agen lebih memilih kecepatan yang lama karena mengarahkan mereka langsung ke tujuan mereka, maka mereka akan memilih kecepatan ini lagi. Sehingga pada siklus berikutnya, kedua agen dengan kecepatan lama akan menghasilkan tabrakan, dan mereka mungkin akan memilih v'_A dan v'_B lagi, dan seterusnya. Dengan demikian agen

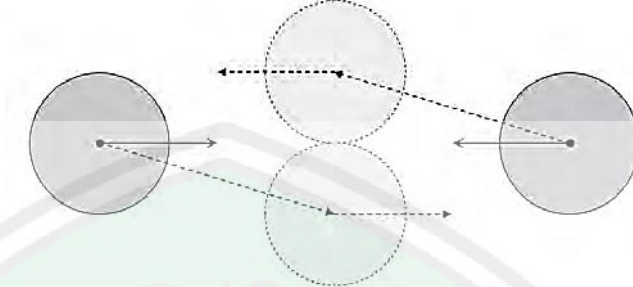
akan mengalami osilasi di antara dua kecepatan ini ketika *velocity obstacle* digunakan untuk menghindari satu sama lain (Fiorini dan Shiller 1998).

2.1.4.4 Reciprocal Velocity Obstacle

Reciprocal Velocity Obstacle, atau RVO, adalah sebuah konsep untuk menghindari tabrakan reaktif lokal yang digunakan dalam navigasi multi-agen, di mana setiap agen menavigasi dan memilih rute secara mandiri tanpa komunikasi eksplisit satu sama lain. Jadi, Algoritma RVO memungkinkan penghindaran tabrakan antar agen tanpa koordinasi satu sama lain. RVO merupakan pengembangan dari konsep *velocity obstacle* atau VO yang di gunakan untuk navigasi pada obstacle yang bergerak. Pendekatan RVO memperhitungkan perilaku reaktif dari agen lain yang mengasumsikan bahwa agen lain melakukan perilaku penghindaran tabrakan yang sama dengan tujuan menghasilkan gerakan agen yang bebas dari tabrakan, baik itu dengan agen lain maupun *obstacle*, dan serta bebas dari osilasi yang dapat terjadi dalam VO.

Konsep *reciprocal velocity obstacle* (RVO) diperkenalkan oleh J. Van den Berg dkk (2008) masalah yang mereka coba pecahkan adalah masalah osilasi. Osilasi pada *velocity obstacle* terjadi ketika agen kedua dimasukkan kedalam lingkungan. Mereka membuat kesimpulan bahwa alasan terjadinya osilasi adalah ketika kedua agen menganggap bahwa agen yang lain tidak akan mengubah kecepatannya dan dengan demikian mereka harus melakukan manuver penghindaran penuh. Dengan kata lain para agen saling mempertimbangkan untuk menjadi *obstacle*. Konsep RVO adalah mengubah asumsi ini. Daripada menganggap satu sama lain sebagai obstacle dan mengambil tanggung jawab

penyuh, para agen menganggap bahwa mereka akan berbagi tanggung jawab untuk menghindari.



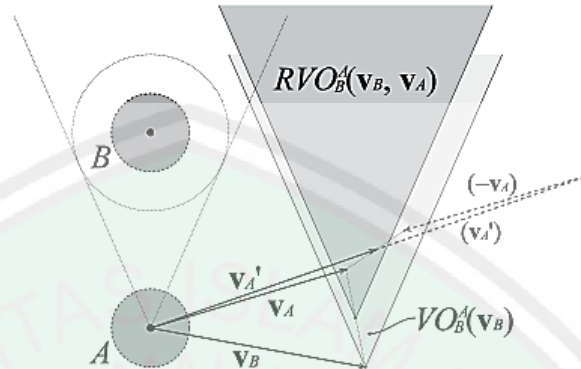
Gambar 2. 4 Pergerakan agen yang bebas osilasi

Reciprocal Velocity Obstacle merupakan konsep baru dalam pencegahan tabrakan reaktif lokal yang secara terstruktur menganggap bahwa agen lain dapat membuat penalaran dalam menghindari tabrakan yang sama. Dengan asumsi ini, RVO dapat menghasilkan gerakan yang aman dan bebas osilasi. Untuk itu, informasi yang dibutuhkan oleh setiap agen tentang agen lain adalah posisi dan kecepatannya saat ini. Bagian mendasar dari RVO adalah daripada memilih kecepatan baru untuk setiap agen yang berada di luar *velocity obstacle* agen lainnya, maka dipilihlah kecepatan baru yaitu rata-rata kecepatan arus dan kecepatan yang berada di luar *velocity obstacle* agen lainnya. Yang didefinisikan sebagai berikut:

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v}'_A \mid 2\mathbf{v}'_A - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\}.$$

Reciprocal velocity obstacle $RVO_B^A(vb, va)$ dari agen B ke agen A berisi semua kecepatan untuk agen A yang merupakan rata-rata kecepatan arus va saat ini dan kecepatan yang berada dalam $VO_B^A(vb)$ dari agen B. Yang Secara geometris

dapat diartikan sebagai *velocity obstacle* $VO_B^A(vb)$ yang diterjemahkan sedemikian rupa sehingga puncak dari RVO terletak pada $\frac{v_A + v_B}{2}$.

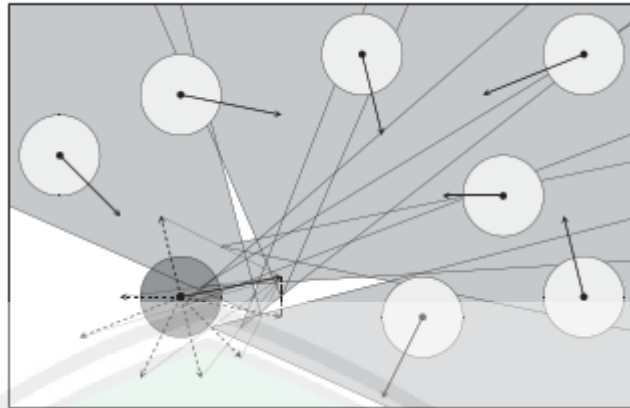


Gambar 2. 5 $RVO_B^A(vb, va)$ dari agen B ke agen A

Perbedaan mendasar antara *Velocity Obstacle* (VO) dan *Reciprocal Velocity Obstacle* (RVO) yaitu terdapat pada letak puncak kerucut dari tabrakan seperti pada gambar 3.10 Sama halnya dengan VO, konsep RVO dibangun atas matematika yang sederhana. Konsep RVO juga menangani masalah osilasi ketika kedua agen harus saling menghindar. Dengan menggunakan RVO, agen dapat diperintahkan untuk mengambil tanggung jawab manuver pengelakan yang berbeda.

2.1.4.5 Multiagen navigasi menggunakan RVO

Penjelasan mendasar mengenai konsep RVO yaitu metode ini digunakan kepada sepasang agen. Jika konsep ini diterapkan untuk agen yang bergerak di antara banyak agen lain serta adanya pergerakan pasif atau *obstacle* statis, maka kombinasi RVO untuk agen A menjadi penyatuan semua RVO yang dihasilkan oleh agen lain secara individual, yang memungkinkan adanya timbal balik dengan agen lain serta *velocity obstacle* yang dihasilkan oleh *obstacle* bergerak secara pasif (Fiorini dan Shiller 1998), seperti yang di tunjukkan pada Gambar 2.6.



Gambar 2. 6 Multi agen navigasi menggunakan RVO

2.1.5. *Tools Unity3D*

Unity merupakan salah satu *platform* dari sebuah *Engine game* dimana sangat diperlukannya sebagai alat yang terkemuka. *Unity3D* merupakan sebuah *software development* yang reintegrasi untuk menciptakan *game* yang *multi platform* dalam pembuatannya. *Unity* mempunyai perpaduan yang baik dengan aplikasi yang lebih profesional. *Software* ini juga terintegrasi untuk membuat video atau animasi dan konten lainnya seperti visualisasi arsitektur atau real-time animasi baik yang bernuansa 2D maupun 3D. *Software Unity* dapat digunakan pada *Microsoft Windows* dan *Mac OS X*. Grafis pada *Unity* dibuat dengan grafis tingkat tinggi untuk *OpenGL* dan *DirectX*. *Unity* mendukung semua format, terutama dalam format pada umumnya seperti format dari *art application*. *Unity* dapat menghasilkan yang bisa dijalankan di *Mac*, *Windows*, *Wii*, *Iphone*, *Ipad* dan *Android*. (Mutia, Djuniadi, 2015).

2.1.6. *Pariwisata*

Istilah pariwisata secara etimologi yang berasal dari Bahasa Sansekerta yang terdiri dari kata ‘*pari*’ yang berarti halus, maksudnya mempunyai tata krama

tinggi dan ‘wisata’ yang berarti kunjungan untuk melihat, mendengar, menikmati dan mempelajari sesuatu. Maka pariwisata itu berarti menyuguhkan suatu kunjungan secara bertata krama dan berbudi.

Sedangkan menurut Undang-undang RI No. 10 tahun 2009 tentang kepariwisataan menyebutkan bahwa pariwisata adalah: “Berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan oleh masyarakat, pengusaha, pemerintah dan pemerintah daerah”.

Pengertian pariwisata menurut A.J Burkat dalam Damanik (2006), pariwisata adalah perpindahan orang untuk sementara dan dalam jangka waktu pendek ke tujuan-tujuan diluar tempat dimana mereka biasa hidup dan bekerja dan juga kegiatan-kegiatan mereka selama tinggal di suatu tempat tujuan.

Menurut pendapat yang dikemukakan oleh Youti (1991:103) pariwisata berasal dari dua kata yaitu Pari dan Wisata. Pari dapat diartikan sebagai banyak, berkali-kali, berputar-putar atau lengkap. Sedangkan Wisata dapat diartikan sebagai perjalanan atau bepergian yang dalam hal ini sinonim dengan kata “*reavel*” dalam bahasa Inggris. Atas dasar itu maka kata “pariwisata” dapat juga diartikan sebagai perjalanan yang dilakukan berkali-kali atau berputar-putar dari suatu tempat ketempat yang lain yang dalam bahasa Inggris disebut juga dengan istilah “*Tour*”.

Dari beberapa definisi yang dikemukakan diatas, maka dapat disimpulkan bahwa pariwisata adalah suatu perjalanan yang dilakukan orang dari suatu tempat ke tempat lain, untuk sementara waktu dengan maksud atau tujuan tidak untuk berusaha atau mencari nafkah ataupun menetap di tempat yang dikunjungi, akan tetapi untuk menikmati perjalanan tersebut sebagai rekreasi atau untuk memenuhi

kegiatan yang beragam tanpa adanya suatu paksaan dan dilakukan perorangan maupun kelompok (Yoeti, 1996).

2.1.7. Alun-Alun Kota Batu

Perkembangan pariwisata memberikan dampak positif bagi perekonomian masyarakat. Dalam pengembangan sebuah destinasi pariwisata, berbagai aspek terkait menjadi pertimbangan untuk perencanaan yang efektif dan efisien serta tepat sasaran dengan melibatkan seluruh pemangku kepentingan terkait. Peran masyarakat local juga diperlukan untuk turut andil dalam mengembangkan, memberikan kontribusi, serta menjaga kegiatan pariwisata di daerah mereka. Selain itu, faktor sosial budaya yang juga berkaitan dengan pariwisata dapat menjadi pertimbangan dalam mengembangkan dan mengelola pariwisata agar dapat beriringan dengan nilai-nilai budaya masyarakat lokal.

Alun-alun Kota Batu sebagai ruang terbuka publik memiliki banyak manfaat untuk masyarakat. Menurut peraturan Menteri Pekerjaan Umum Nomor: 05/PRT/M/2008, manfaat ruang terbuka adalah menjaga ketersediaan lahan sebagai kawasan resapan air, menciptakan aspek planologis perkotaan melalui keseimbangan antara lingkungan alam dan lingkungan binaan yang berguna untuk kepentingan masyarakat serta meningkatkan keserasian lingkungan perkotaan sebagai sarana pengaman lingkungan perkotaan yang aman, nyaman, segar, indah, dan bersih.

2.2. Penelitian Terkait

Juniastuti Susi (2016) di dalam penelitian nya yang berjudul “*Crowd Navigation using Leader-Follower Algorithm Based Reciprocal Velocity*

Obstacles” menggunakan metode RVO untuk menciptakan gerakan yang bebas dari osilasi dan tabrakan pada navigasi multi agen. Dengan menggabungkan RVO dengan perilaku mengikuti *leader* sehingga setiap agen menavigasi untuk mengikuti pemimpin mereka untuk mencapai tujuan mereka. Hasil dari penelitian ini menunjukkan navigasi ratusan agen di lingkungan padat penduduk yang menunjukkan 18% lebih cepat dibandingkan dengan navigasi tanpa relasi untuk mencapai tujuan orang banyak.

Moch Fachri (2018) dalam penelitiannya tentang Navigasi Multiagen dengan menggunakan metode *Reciprocal Velocity Obstacles* dengan relasi *leader-follower* sehingga dapat melakukan manuver penghindaran yang agar tidak mengakibatkan kekacauan massal tanpa adanya komunikasi eksplisit antar agen. Dalam penelitian tersebut agen *follower* diharuskan mampu mengikuti *leader* kelompoknya serta memberikan agen *leader*-nya kebebasan bernavigasi, yang berarti gerakan agen *follower* sebisa mungkin tidak mengganggu *leader*-nya. Hasil yang didapatkan berupa simulasi kerumunan dengan relasi *leader-follower* yang bebas dari tabrakan, osilasi, dan kehilangan arah.

Paolo Fiorini (1998) dalam sebuah makalah yang berjudul “*Motion Planning in Dynamic Environments Using Velocity Obstacles*” menyajikan metode untuk perencanaan gerak robot dalam lingkungan yang dinamis untuk dapat melakukan manuver penghindaran untuk menghindari rintangan statis dan bergerak di ruang kecepatan, berdasarkan posisi saat ini dan kecepatan robot dan rintangan. Metode ini ditujukan untuk robot pada *obstacle* statis dan bergerak, dan untuk kendaraan otomatis dalam skenario sistem jalan raya kendaraan cerdas.

Takashi Chikayama (2000) melakukan penelitian tentang simulasi perilaku pejalan kaki dalam menghindari tabrakan. Penelitian ini menyajikan algoritma untuk memodelkan perilaku menghindari tabrakan pejalan kaki. Metode yang digunakan berdasar pada representasi lintasan dalam ruang (x, y, t) . Dimana setiap agen memilih gerakan masa depannya untuk menghindari tabrakan, meminimalkan pergerakan memutar dan variasi kecepatan. Hasil dari penelitian ini adalah simulasi dari pejalan kaki yang dapat menghindari tabrakan antar pejalan kaki secara realistis.

Muhammad Badrul (2017) melakukan penelitian tentang simulasi pengelolaan kerumunan massa demonstran menggunakan metode *Crowd Dynamic Model*. Penelitian tersebut sebagai salah satu bentuk pengembangan kejadian yang terjadi di dunia nyata yang dibutuhkan oleh para penegak hukum untuk mengatur massa demonstrasi. Hasilnya berupa visualisasi menggunakan *NetLogo*. Sehingga diharapkan dapat meningkatkan kemampuan aparat dalam mengatur kerumunan massa khususnya massa demonstran.

Tryadi Wilhelmus (2016) melakukan penelitian dengan judul “simulasi berbasis agen untuk evakuasi bencana kerumunan lapangan indoor gor x” Penerapan langkah-langkah keamanan yang tepat dapat mengurangi dampak bahaya yang terjadi ketika bencana pada kerumunan manusia di lapangan indoor GOR (Gelanggang Olah Raga) X. penelitian ini menyajikan model komputasi yang mensimulasikan evakuasi bencana, dimana manusia sebagai agent mempunyai aktivitas interaksi secara otonom atau berkelompok terhadap lingkungan yang mempunyai jalur evakuasi, susunan tepat duduk, dan sistem pendukung keputusan untuk mencari jalan keluar terdekat. Tujuan *prototipe* simulasi ini yaitu untuk

mendapatkan presentasi jumlah kapasitas penonton lapangan indoor GOR X. Berdasarkan hasil eksperimen skenario dengan waktu evakuasi terbaik ada pada skala 60-80% dimana hasil setelahnya waktu yang diperlukan cenderung konstan. Model simulasi ini bisa menjadi salah satu alternatif untuk pendukung sistem keputusan dalam memberikan rekomendasi perencanaan kapasitas penonton yang akan diambil oleh pihak manajemen atau penyelenggara kegiatan di lapangan indoor GOR X.

Jamie Snape (2011) melakukan penelitian dengan judul “*The Hybrid Reciprocal Velocity Obstacle*” penelitian tersebut bertujuan untuk navigasi bebas-tabrakan dan osilasi bebas dari beberapa *robot mobile* atau agen virtual. Dimana Setiap *robot* merasakan lingkungannya dan bertindak secara mandiri tanpa koordinasi pusat atau komunikasi dengan *robot* lain. Dalam metode ini menggunakan *reciprocal velocity obstacle hibrid* untuk navigasi beberapa *robot* bergerak atau agen virtual dalam suatu lingkungan.

Martin Funkquist (2016) membuat penelitian dengan judul “*Simulating Group Formations Using RVO*” dalam penelitian ini menerapkan formasi grup ke dalam engine game *Unity3D*. Dimana sekelompok orang bergerak dengan cara tertentu ketika menghindari kelompok dan rintangan lain. Hasil yang didapatkan dari menggunakan metode adalah Para agen tidak pernah berjalan satu sama lain, meskipun ada beberapa kebuntuan ketika jumlah agen benar-benar besar.

Fadil Muhammad (2018) melakukan penelitian simulasi kerumunan pada objek evakuasi bencana dengan judul “*Crowds Evacuation Simulation on Heterogeneous Agent Using Agent-Based Reciprocal Velocity Obstacle*” penelitian ini bertujuan untuk melihat kemampuan perilaku manusia saat mengungsi ketika sedang terjebak dalam suatu bencana dengan menggunakan metode “*used agent-based reciprocal velocity obstacle*”. Hasil percobaan menunjukkan simulasi ratusan agen dalam tiga skenario evakuasi dengan menggunakan hubungan pemimpin-pengikut menghasilkan kemungkinan tumbukan antar agen semakin yang semakin besar.



BAB III

DESAIN DAN RANCANGAN SIMULASI

3.1 Tahapan Penelitian

Tahapan penelitian yang akan dilakukan pada penelitian ini sesuai dengan diagram alur penelitian berikut ini :



Gambar 3. 1 Diagram alur penelitian

3.1.1 Studi literatur

Pada tahap ini dilakukan berbagai pengumpulan simulasi terkait beberapa hal, yaitu :

- a. Pengumpulan informasi tentang wisata
- b. Pengumpulan Informasi tentang pembuatan simulasi 3D menggunakan *Unity 3D*

- c. Pengumpulan informasi tentang metode *Reciprocal Velocity Obstacle*
- d. Pengumpulan informasi tentang penelitian terkait.

3.1.2 Desain skenario

Pada tahap ini dirancang bagaimana simulasi akan dijalankan menggunakan metode *Reciprocal Velocity Obstacle*.

3.1.3 Desain Agen dan *Enviroment*

Pada tahap ini dirancang bagaimana desain *interface* dari simulasi dan desain dari setiap agen.

3.1.4 Perancangan Simulasi

Proses perancangan simulasi dibangun dengan menggunakan *Unity engine* yang menggunakan bahasa pemrograman C#. Selain itu pembuatan *script* dilakukan menggunakan *Mono Develop Text Editor*.

3.1.5 Uji Coba Simulasi

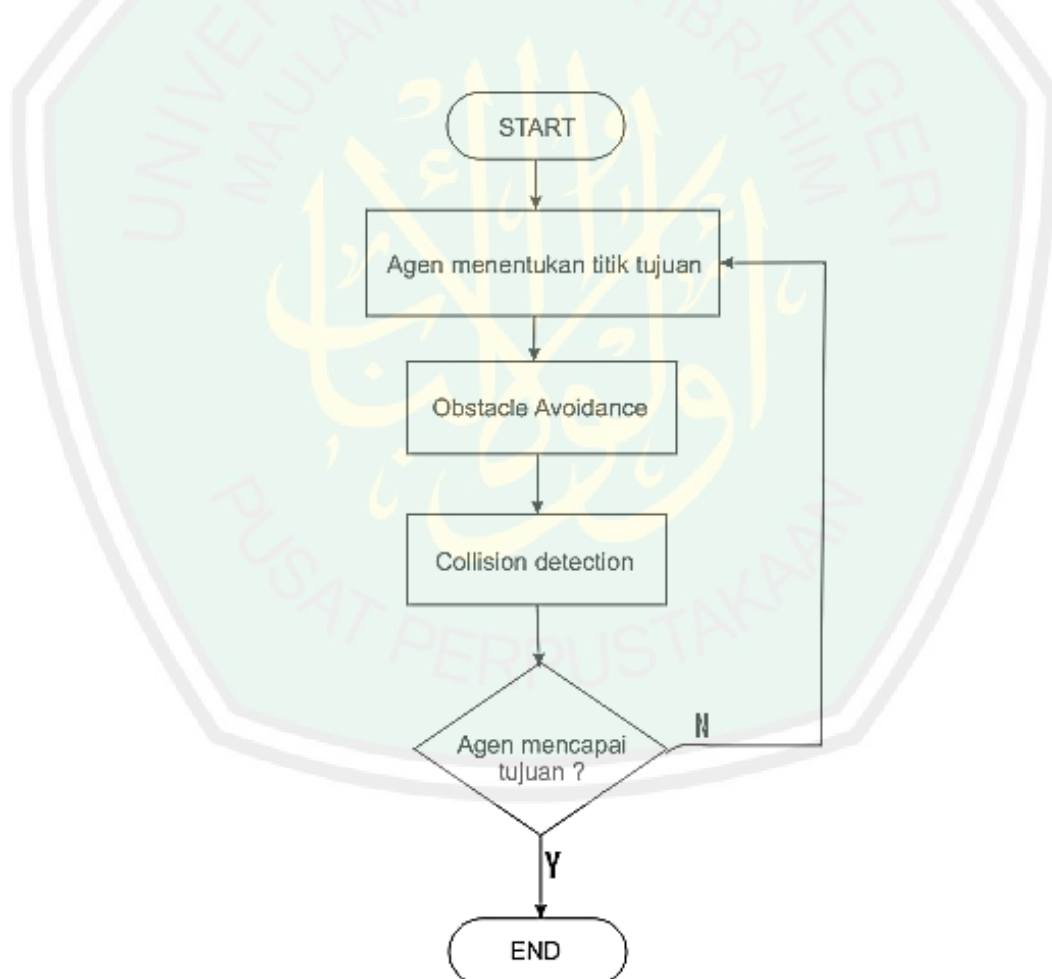
Uji coba dalam penelitian ini dilakukan pada simulasi dan karakter agen yang di implementasikan menggunakan algoritma *Reciprocal Velocity Obstacle*. Kemudian dilakukan pengecekan apakah *output* yang dihasilkan sesuai dengan yang diharapkan.

3.1.6 Penyusunan Laporan

Penyusunan laporan merupakan bagian dokumentasi dari keseluruhan pelaksanaan penelitian yang diharapkan dapat bermanfaat bagi penelitian selanjutnya.

3.2 Desain dan Perancangan Simulasi

Simulasi ini dirancang dengan menerapkan Algoritma *Reciprocal Velocity Obstacle* pada kerumunan orang yang berada di tempat wisata. Dalam simulasi ini karakter berupa agen. Agen tersebut merupakan masyarakat yang sedang berkeliling di area wisata yang memiliki perilaku cerdas dengan menerapkan metode *Reciprocal Velocity Obstacle*. Metode ini digunakan untuk memberikan perilaku pada agen saat berada dalam kerumunan orang agar dapat menghindari satu sama lain antar agen saat bergerak menuju titik tujuannya sehingga tidak terjadi tabrakan maupun osilasi.



Gambar 3. 2 *Flowchart* alur pergerakan agen

a. Agen menentukan tujuan (*seeking*)

Pada bagian ini, setiap agen akan menentukan titik tujuan atau destinasi yang diinginkan dan mendapatkan preferensi kecepatannya selanjutnya bergerak menuju destinasi.

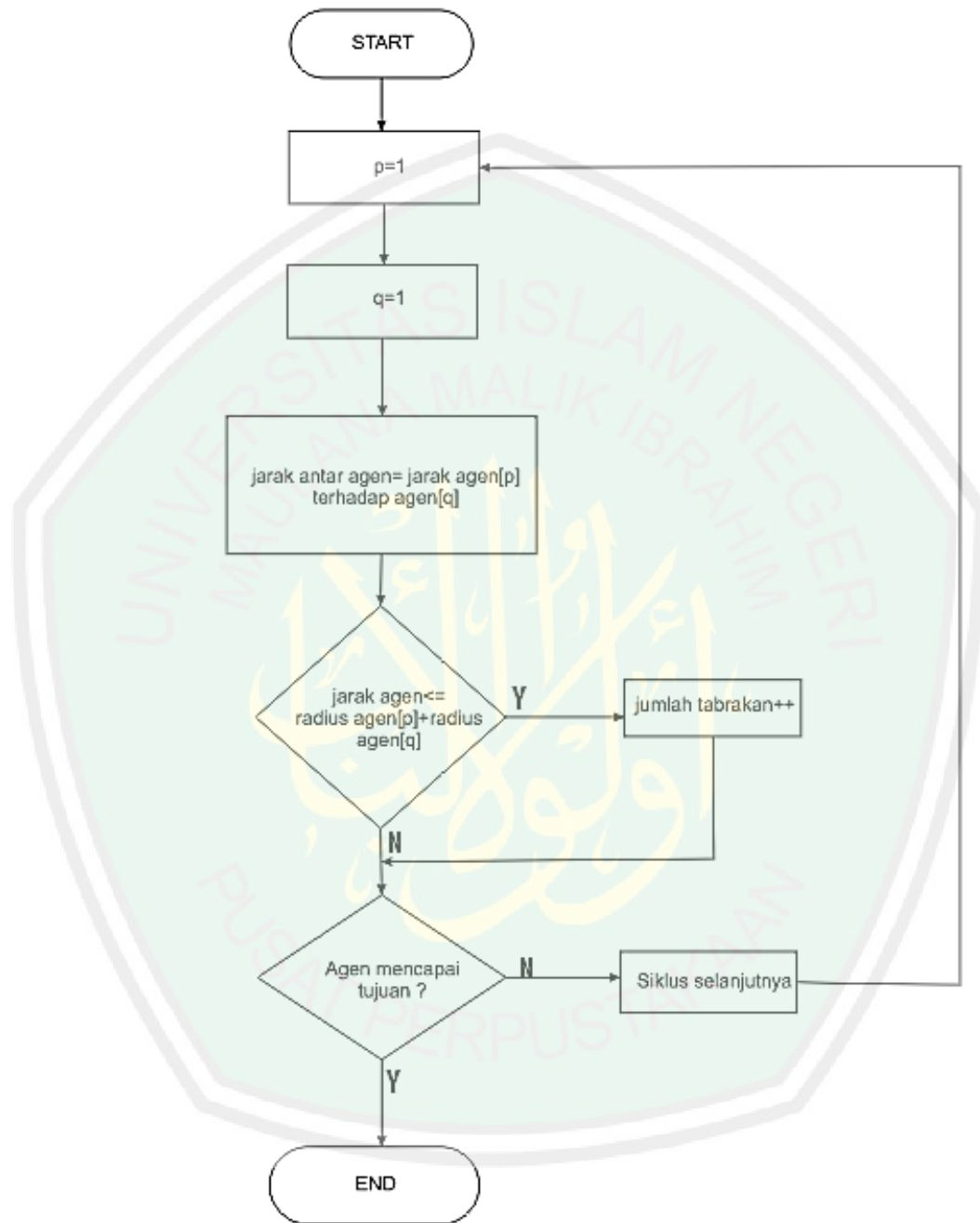
b. *Obstacle avoidance*

Pada bagian selanjutnya adalah *obstacle avoidance* yaitu mencoba untuk mencegah tabrakan terjadi terhadap hambatan statis atau dengan agen bergerak lainnya. Penghindaran dicapai dengan memilih gerakan agen yang valid melalui memilih kecepatan yang valid berdasarkan algoritma RVO. *Velocity* referensi valid bila *velocity* tersebut tidak berada dalam *collision cone* dari RVO yang artinya, *velocity* referensi tidak mengakibatkan *collision*. Bila *velocity* referensi dinyatakan valid maka *velocity* referensi telah sah menjadi *velocity* agen untuk siklus saat itu. Sebaliknya bila *velocity* referensi tersebut tidak valid yang artinya *velocity* tersebut akan mengakibatkan tabrakan dan *velocity* agen akan dipilih dari *velocity* diluar himpunan *velocity* RVO yang terdekat dengan *velocity* referensi. Output dari proses ini menentukan kecepatan agen yang akan digunakan untuk siklus saat ini.

c. *Collision detection*

Tahap terakhir setelah menentukan pergerakan agen adalah deteksi tabrakan. Algoritma pendeteksian tabrakan menggunakan deteksi

lingkaran-ke-lingkaran. Agen akan dianggap bertabrakan jika jarak antara dua agen kurang dari jumlah jari-jari kedua agen.



Gambar 3. 3 *Flowchart collision detection*

3.3 Implementasi Algoritma *Reciprocal Velocity Obstacle*

3.3.1 Parameter

Dalam simulasi ini parameter terdapat beberapa parameter yang digunakan. Parameter ini digunakan untuk menganalisa keefektifan navigasi multi agen menggunakan algoritma RVO pada simulasi kerumunan.

a. Jumlah agen

Merupakan data jumlah agen yang dipakai dalam setiap uji coba simulasi. Dimana jumlah agen dalam simulasi akan berbeda dalam setiap uji coba simulasi.

b. Waktu simulasi

Merupakan data yang merepresentasikan waktu yang diperlukan untuk melakukan simulasi. Sebelum simulasi dimulai waktu untuk melakukan simulasi telah ditentukan sebelumnya dan simulasi dianggap selesai ketika semua agen telah sampai pada titik tujuan masing-masing agen atau berhenti sesuai dengan waktu yang telah ditentukan jika terdapat agen yang tidak bisa mencapai titik tujuannya yang menyebabkan agen tersebut tidak dapat melanjutkan perjalanannya karena *stuck*. Data waktu simulasi didapatkan dengan mengambil data waktu yang telah ada pada *unity* dimana waktu simulasi dihitung ketika simulasi dimulai.

c. Jumlah *frame*

Merupakan data yang menunjukkan jumlah *frame* yang ada selama simulasi berlangsung. Dalam setiap *frame* menunjukkan *cycle* perhitungan pergerakan agen. Total *frame* dari simulasi didapat dengan pemanggilan

fungsi *update perframe* yang dapat dilihat pada *Unity* ketika simulasi berjalan.

d. *Framerate* rata-rata

Menunjukkan data rata-rata dari jumlah *frame* persatuan detik. Didapatkan dengan membagi jumlah *frame* selama simulasi berjalan dengan waktu simulasi.

$$\Sigma FPS = \frac{\Sigma Frame}{\Sigma SimulationTime}$$

e. Jumlah *collision*

Menunjukkan data jumlah tabrakan yang terjadi selama simulasi berlangsung. Data ini diambil dari jumlah *collision* yang terjadi selama simulasi berlangsung.

3.3.2 Perhitungan RVO

Pada bagian ini akan dibuktikan apakah algoritma RVO dapat digunakan untuk menghasilkan gerakan bebas tabrakan dan bebas osilasi untuk setiap agen.

a. Navigasi bebas tabrakan

Misalkan V_A merupakan kecepatan saat ini dari agen A, dan misalkan V_B merupakan kecepatan saat ini dari agen B, dan masing-masing agen A dan B memilih kecepatan baru V_A' dan V_B' diluar masing-masing RVO. Teorema berikut membuktikan bahwa kedua agen tidak akan bertabrakan dan memilih sisi yang sama untuk saling melewati.

$$\mathbf{v}_A \in VO_B^A(\mathbf{v}_B) \Leftrightarrow \mathbf{v}_B \in VO_A^B(\mathbf{v}_A)$$

$$\mathbf{v}_A \in VO_B^A(\mathbf{v}_B) \Leftrightarrow \mathbf{v}_A + \mathbf{u} \in VO_B^A(\mathbf{v}_B + \mathbf{u})$$

$$\mathbf{v}_A \notin VO_B^A(\mathbf{v}_B) \wedge \mathbf{v}'_A \notin VO_B^A(\mathbf{v}_B) \Rightarrow \\ (1 - \alpha)\mathbf{v}_A + \alpha\mathbf{v}'_A \notin VO_B^A(\mathbf{v}_B), \text{ for } 0 \leq \alpha \leq 1.$$

$$\mathbf{v}'_A \notin RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) \wedge \mathbf{v}'_B \notin RVO_A^B(\mathbf{v}_A, \mathbf{v}_B) \Rightarrow \\ \mathbf{v}'_A \notin VO_B^A(\mathbf{v}'_B) \wedge \mathbf{v}'_B \notin VO_A^B(\mathbf{v}'_A)$$

$$va' \notin RVO_B^A(vb, va) \wedge vb' \notin RVO_A^B(va, vb)$$

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v}'_A \mid 2\mathbf{v}'_A - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\}$$

Jika kita masukkan ke dalam Rumus RVO maka, menjadi :

$$\Leftrightarrow 2va' - va \notin VO_B^A(vb) \wedge 2vb' - vb \notin VO_A^B(va)$$

$$\Leftrightarrow 2va' - va \notin VO_B^A(vb) \wedge va \notin VO_B^A(2vb' - vb)$$

$$\Leftrightarrow 2va' - va - vb \notin VO_B^A(0) \wedge va = vb - 2vb' \notin VO_B^A(0)$$

$$\Leftrightarrow va' - vb \notin VO_B^A(0)$$

$$\Leftrightarrow va' \notin VO_B^A(vb')$$

$$\Leftrightarrow va' \notin VO_B^A(vb') \wedge vb' \notin VO_A^B(va')$$

b. Pengambilan sisi yang sama

RVO dapat membuat kedua agen secara otomatis memilih sisi yang sama untuk saling melewati jika masing-masing dari mereka memilih kecepatan di luar RVO agen lain yang paling dekat dengan kecepatan saat ini. Jika untuk agen A, $V_A + u$

adalah kecepatan terdekat dengan V_A di luar RVO_B , maka untuk agen B, $V_B - u$ adalah kecepatan terdekat dengan V_B di luar RVO_A .

$$v_A + u \notin RVO_B^A(v_B, v_A) \Leftrightarrow v_B - u \notin RVO_A^B(v_A, v_B)$$

Hal ini dapat dibuktikan dengan $va + u \notin RVO_B^A(vb, va)$

$$\Leftrightarrow 2(va + u) - va \notin VO_B^A(vb)$$

$$\Leftrightarrow (2vb - u) - vb \notin VO_A^B(va)$$

$$\Leftrightarrow vb - u \notin VO_A^B(va, vb)$$

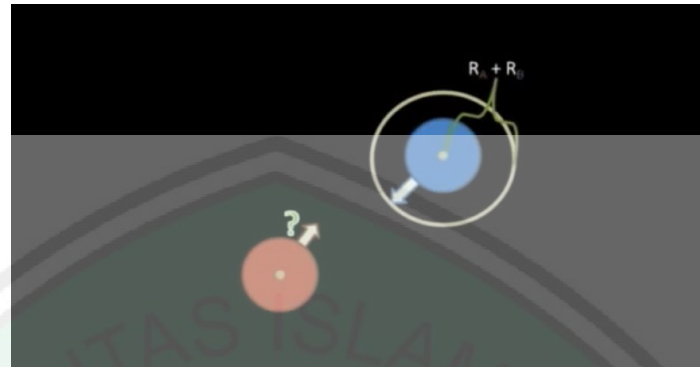
Dalam simulasi ini setiap agen diharuskan dapat menghindari tabrakan antar agen satu dengan agen yang lain. Berikut merupakan langkah-langkah dalam melakukan simulasi.



Gambar 3. 4 Dua agen yang bergerak berlawanan

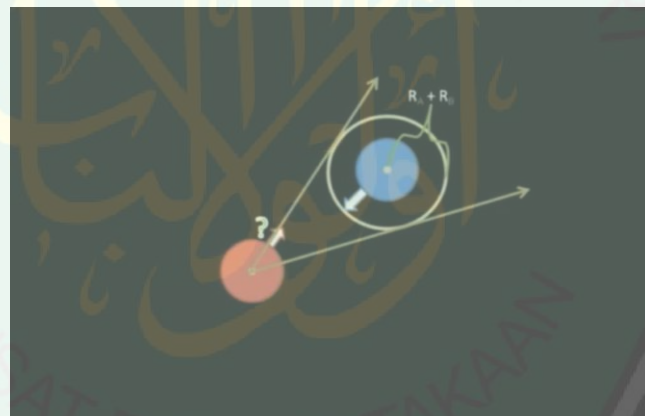
Pada Gambar 3.4 terdapat dua agen yaitu agen berwarna merah dan biru. Agen yang ditunjukkan dengan warna merah memilih salah satu agen yang paling dekat yang dianggap sebagai sebuah *obstacle* berupa agen biru. Kemudian agen

merah menghitung himpunan semua kecepatan yang akan berbenturan dengan agen itu jika tetap diam.



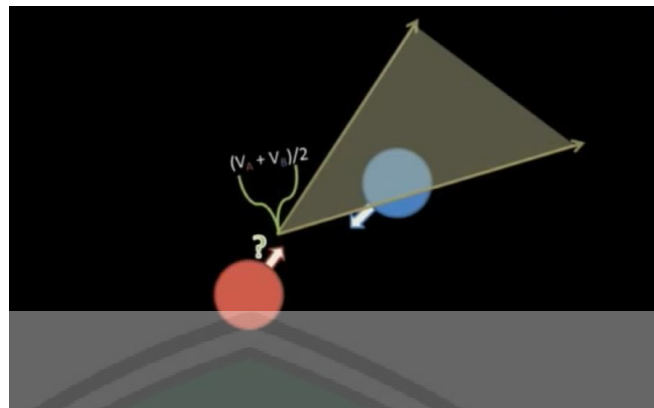
Gambar 3. 5 Agen merah menghitung wilayah *collider*

Dari Gambar 3.5 dapat dilihat bahwa agen merah menghitung wilayah terluar dari tabrakan dengan menghitung dua kali jari-jari dari agen biru.



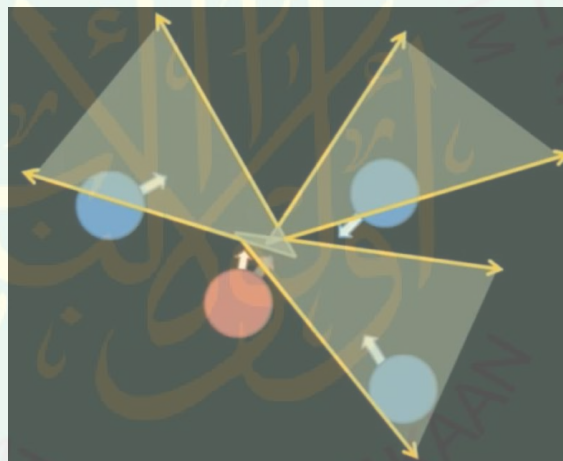
Gambar 3. 6 VO pada agen merah terhadap agen biru

Setelah menghitung himpunan dari semua kecepatan agen dan wilayah yang berpotensi terjadi tabrakan, kemudian membentuk kerucut yang berpusat pada posisi agen merah. Himpunan inilah yang disebut wilayah *velocity obstacle*.



Gambar 3. 7 RVO pada agen merah kepada agen biru

Untuk menjelaskan bahwa kedua agen akan bergerak untuk menghindari satu sama lain, dan agar tidak terjadi osilasi maka puncak kerucut itu dipindahkan sehingga puncak kerucut terletak pada kecepatan rata-rata dari dua agen.



Gambar 3. 8 RVO dari agen merah dengan beberapa agen biru

Proses ini di ulang untuk masing- masing agen dan semua kerucut dari VO yang dihasilkan disatukan bersama. Yang secara efisien digambarkan batas wilayah yang memproyeksikan kecepatan yang diinginkan agen ke titik terdekat dengan tujuan yang merupakan kecepatan aman yang bisa diambil oleh agen merah. Sehingga menghasilkan serangkaian kecepatan bebas tabrakan untuk setiap agen.

3.4 Pembuatan Lingkungan Simulasi

Lingkungan simulasi kerumunan yang dibuat disesuaikan dengan skenario simulasi. Secara garis besar objek lingkungan simulasi kerumunan dihimpun menjadi dua jenis berdasarkan kemampuan agen bernavigasi yaitu *walkable* dan *unwalkable*. Sehingga perlu ada Pendenifisian pada lingkungan simulasi bagian mana yang dapat dilalui agen dan tidak dapat dilalui untuk masing -masing scenario.

1. Cross Scenario

Pada simulasi kali ini, area dari lingkungan simulasi berbentuk jalan lurus berupa lorong yang terdapat dinding pembatas di sisi lorong seperti yang ditunjukkan pada Gambar 3.9. Dalam simulasi ini terdapat dua titik awal dan dua titik akhir. Jumlah dalam simulasi ini dibatasi maksimal 100 agen pada setiap kerumunanya.



Gambar 3. 9 Area Cross Scenario

Penempatan titik awal dan akhir masing-masing kerumunan ditunjukkan pada Tabel 3.1. berdasarkan tabel tersebut, diketahui kerumunan satu berada di titik awal pada koordinat $(-67;61)$ dan akan mencapai tujuan pada titik koordinat $(67;61)$ sedangkan untuk kerumunan agen dua akan berada di titik awal pada koordinat $(67;61)$ dan akan mencapai titik tujuan pada koordiant $(-67;61)$ Kedua kerumunan

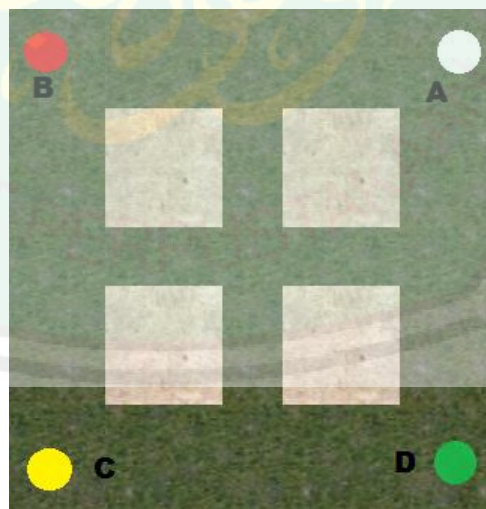
tersebut akan saling bertemu ditengah lorong yang merupakan *obstacle* pada skenario ini.

Tabel 3. 1 Titik awal dan tujuan dalam cross scenario

Kerumunan	Titik Awal	Titik Akhir
1	-67;61	67;61
2	67;61	-67;61

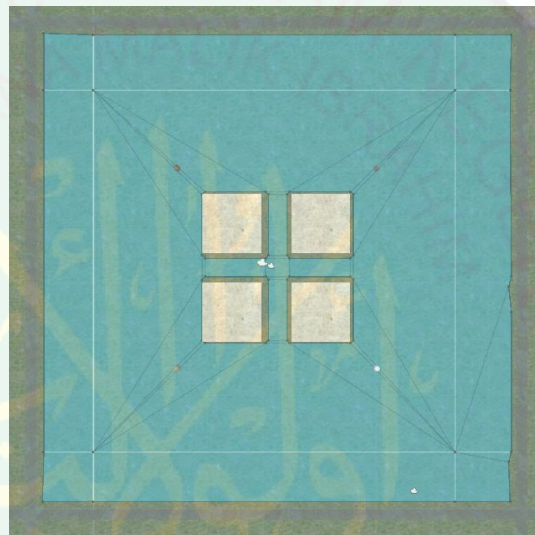
2. *Narrow Passage Scenario*

Lingkungan simulasi untuk skenario ini dibentuk pada area seluas 100x100. Gambar 3.10 menunjukkan lingkungan simulasi yang telah selesai beserta area walkable agen. Sebagaimana ditunjukkan pada gambar tersebut, terdapat 4 buah rintangan berupa empat buah kotak dengan ukuran 10x10 yang ditempatkan sedemikian rupa sehingga terbentuk jalan sempit selebar 5 unit antar rintangan tersebut yang diistilahkan dengan *narrow-passage* yang mana merupakan tantangan pada skenario ini.



Gambar 3. 10 Area simulasi *Narrow Passage Scenario*

terdapat agen yang akan ditempatkan pada simulasi seperti yang ditunjukkan pada gambar 3.10 dimana titik awal agen A berada pada titik A, agen B berada pada titik B, agen C berada pada titik C, agen D berada pada titik D. Adapun titik tujuan dari masing-masing agen adalah titik awal agen yang bersebrangan. Sehingga titik akhir agen A adalah titik awal agen C. Titik akhir agen B adalah titik awal agen D. Titik akhir agen C adalah titik awal agen A. Titik akhir agen D adalah titik awal agen B.



Gambar 3. 11 Area *walkable narrow passage scenario*

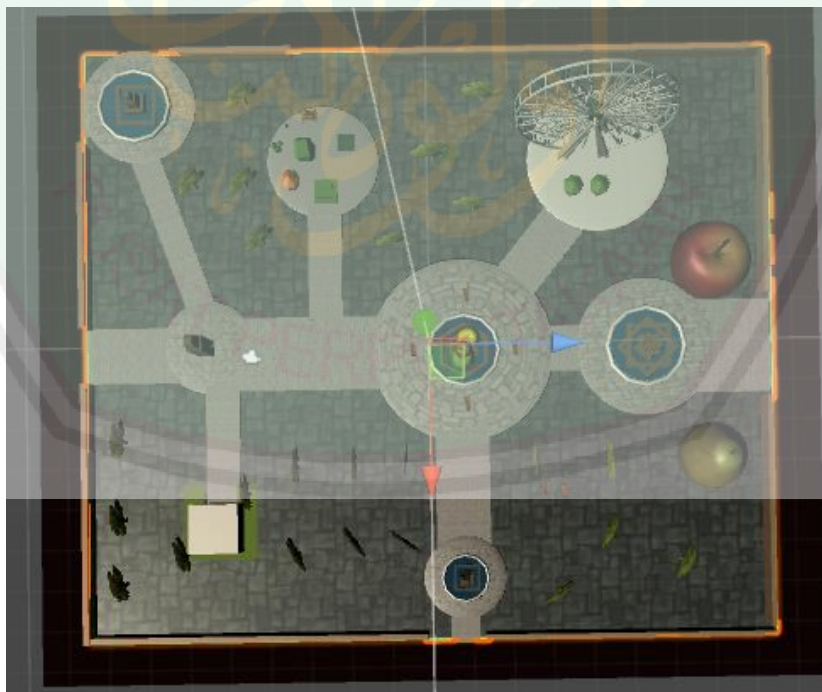
Pada tabel 3.2 dapat diketahui terdapat 4 buah titik awal dari masing masing kerumunan, yaitu kerumunan A berada pada titik awal $(-50;48)$ dan akan menuju titik tujuan yang merupakan titik awal dari kerumunan C $(50;-48)$, sedangkan kerumunan B berada pada titik awal $(-50;48)$ dan akan menuju titik tujuan $(50;48)$ yang merupakan titik awal dari kerumunan D, lalu kerumunan C berada pada titik awal $(-50;48)$ dan akan menuju titik tujuan $(50;48)$ yang merupakan titik awal dari kerumunan A, dan yang terakhir kerumunan D berada pada titik awal $(50;48)$ dan akan menuju titik tujuan $(-50;-48)$ yang merupakan titik awal dari kerumunan B.

Tabel 3. 2 Tabel titik awal dan tujuan agen

Kerumunan	Titik Awal	Titik Akhir
A	-50;48	50;-48
B	-50;-48	50;48
C	50;-48	50;48
D	50;48	-50;-48

3. Alun-alun Skenario

Lingkungan simulasi pada bagian ini berada pada area alun-alun Kota Batu, dalam skenario ini terdapat 4 titik awal yang merupakan tempat *spawn* agen dekat dengan pintu masuk berupa *gate* yang tersebar di beberapa titik. Sedangkan untuk titik akhir dari skenario ini berupa destinasi yang tersebar di dalam wilayah alun-alun dan juga terdapat di pintu keluar. Dalam skenario ini, titik akhir dari setiap kerumunan dibuat *random* dan jumlah agen hanya dibatasi maksimal 300 agen.



Gambar 3. 12 Area skenario Alun-alun Kota Batu

Dalam alun-alun skenario titik awal dari setiap kerumunan berada pada luar wilayah are alun-alun yang merupakan gerbang masuk dari agen untuk masuk ke area simulasi. Terdapat empat buah titik awal dari alun-alun skenario yang tersebar di beberapa tempat yaitu : (1; 84,9) yang berada di barat dari area simulasi, kemudian (-60;-86) yang berada timur dari area simulasi, lalu (-1;85,3) yang berada di selatan area simulasi dan yang terakhir (73;7,93) yang berada pada utara area simulasi. Dari kerumunan agen tersebut akan menuju titik tujuan secara random yang terbagi menjadi 9 titik yang terdiri dari destinasi wisata dan area luar simulasi diantaranya (103;-73, 15,5;18,5 , 136;-13, 9,7;-46 , -18;-100, -1;85,3, -60;-86 , 73;7,93) yang bisa dilihat pada Tabel 3.3.

Tabel 3. 3 Titik awal dan destinasi dari alun-alun skenario

<i>Gate</i>	Titik awal	<i>Destination</i>
1	1; 84,9	103;-73
2	-60;-86	15,5;18,5
3	-1;85,3	136;-13
4	73;7,93	9,7;-46
		-18;-100
		1; 84,9
		-1;85,3
		-60;-86
		73;7,93

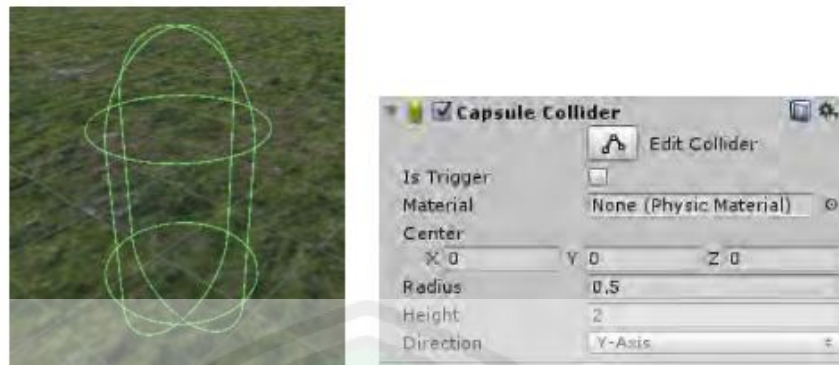
3.4.1 Deskripsi Agen



Gambar 3. 13 bentuk agen dalam simulasi

Agen dalam simulasi kerumunan ini dianggap sebagai objek *capsule* yang bertranslasi pada bidang dimensi dua. Desain agen dapat dilihat pada gambar 3.13. Pada pengujian standar dimana ukuran agen tidak bervariasi, diameter agen dibuat sebesar 1 unit. Agen ini didefinisikan sebagai *navmesh agent* pada *unity* agar agen ini mampu bernavigasi pada *navigation mesh* yang walkable di lingkungan simulasi yang telah dijelaskan sebelumnya. Dimana *navmesh agent* ini merupakan salah satu properti dari RVO untuk mendefinisikan *valid velocity* berdasarkan lebar *navmesh agent*.

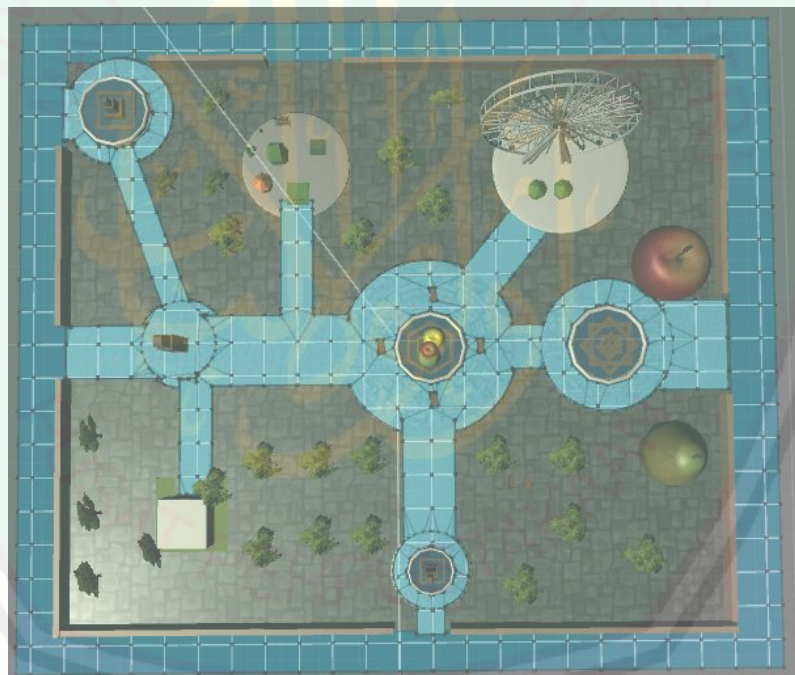
Untuk keperluan deteksi tabrakan, maka perlu ditanam *collider* yang merupakan sistem dalam *Unity* untuk *physics engine* dalam mendeteksi tabrakan. Collider yang dipakai mengikuti bentuk agen yaitu *capsule collider* dengan ukuran yang juga menyesuaikan ukuran agen. Gambar 3.14 adalah gambar *capsule collider* pada *unity* beserta propertinya. *Collider* akan mendeteksi tabrakan bila jarak antar titik pusat collider lebih rendah dari jumlah radiusnya.



Gambar 3. 14 Gambar *collider* yang berada pada tubuh agen

3.4.2 Deskripsi Alun-Alun

1. Area *walkable*



Gambar 3. 15 area *walkable*

Area *walkable* merupakan wilayah yang memungkinkan setiap agen dapat melaluinya atau berjalan melwatinya. Seperti pada gambar 3.15 di atas area

walkable yang dimaksud adalah area yang berwarna biru baik yang di dalam Alun-alun maupun yang diluar.

2. *Start position*

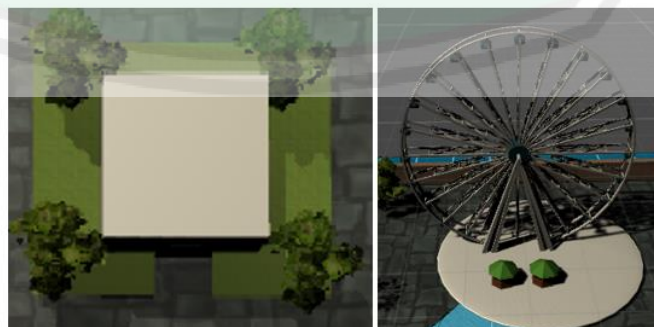
Start position merupakan titik awal dimana agen akan muncul atau *spawn*. Dalam skenario simulasi ini semua *start position* berada diluar alun-alun dekat dengan pintu masuk atau *gate* seperti yang ditunjukkan pada gambar 3.16.



Gambar 3. 16 *start point* pada simulasi alun-alun.

3. *Destination*

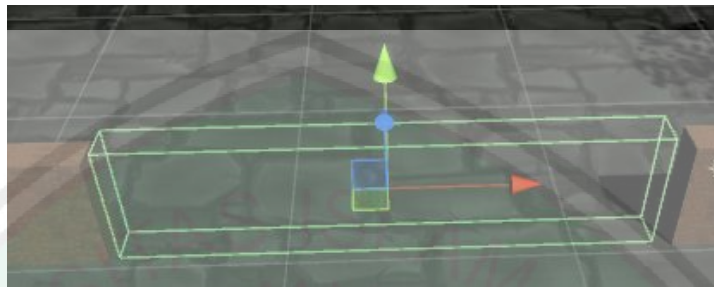
Destination merupakan tujuan atau titik akhir dari setiap agen, terdapat beberapa titik akhir dalam simulasi alun-alun ini, yaitu berupa bangunan/*building* yang berada di area alun-alun dan *spot* yang berada di luar alun-alun.



Gambar 3. 17 *Destination* pada simulasi alun-alun

4. *Gate* / pintu masuk

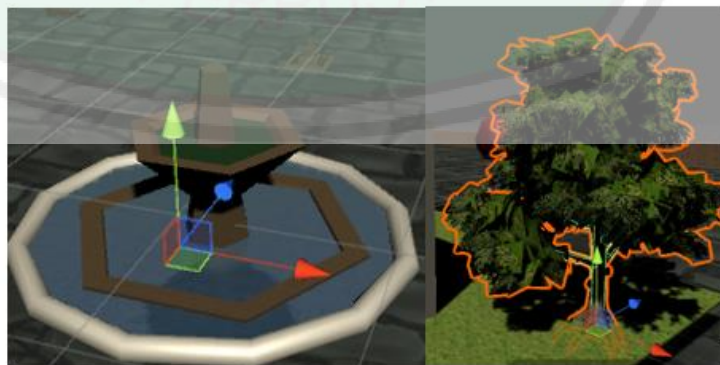
Gate / Pintu masuk merupakan tempat keluar masuknya agen dalam simulasi alun-alun ini.



Gambar 3. 18 *Gate* / pintu masuk simulasi alun-alun

5. *Obstacle*

Obstacle merupakan objek atau hambatan yang terdapat dalam simulasi ini. Setiap agen diharuskan dapat melewati setiap *obstacle* agar dapat sampai tujuan. Dalam simulasi ini terdapat dua macam *obstacle* yakni berupa statis maupun dinamis. *Obstacle* statis terdiri dari objek-objek diam seperti dinding/ *wall*, kolam air, pohon, dll yang terdapat dalam simulasi ini, sedangkan *obstacle* dinamis merupakan agen yang berada dalam simulasi ini .



Gambar 3. 19 *Obstacle* pada simulasi alun-alun

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Sistem Simulasi

Pada bagian ini akan dijelaskan pengujian dari desain sistem dan implementasi yang telah dipaparkan pada Bab 3. Pengujian akan ditujukan untuk memperlihatkan apakah sistem simulasi telah berjalan dengan baik serta implementasi RVO telah berjalan secara optimal atau tidak.

4.1.1 Perangkat Keras yang Dibutuhkan

Perangkat keras yang diperlukan untuk mengimplementasikan simulasi ini menggunakan Laptop HP 1000 Processor AMD A4-3330MX APU with Radeon™ HD Graphics (2CPUs), 2,3 GHz, dan RAM 4GB

4.1.2 Perangkat Lunak yang Dibutuhkan

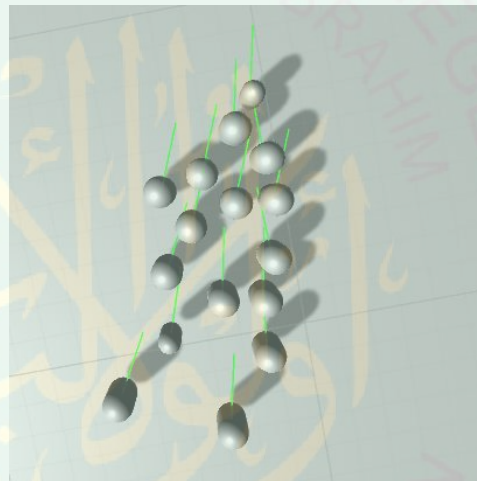
Perangkat lunak diperlukan untuk mengimplementasikan simulasi ini sebagai berikut :

- Sistem operasi : Windows 7 Ultimate 64 Bit
- Tools : Unity3d 5.3.4f1, Blender
- Konsep desain 2D : Corel Draw X7 dan Adobe Photoshop CC 2015
- Text Editor : Mono Develop

4.2 Pengujian RVO

4.2.1 *Seeking*

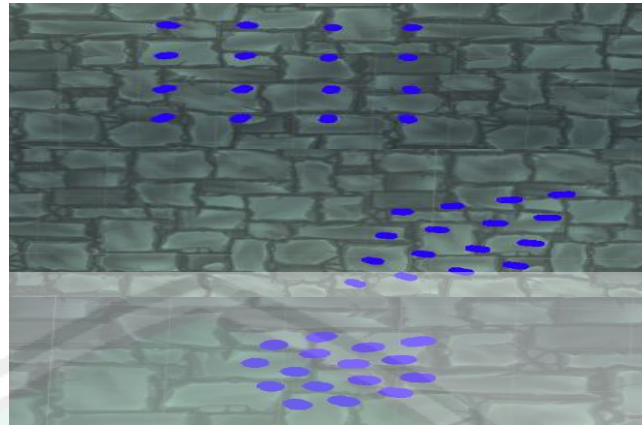
Proses pertama dalam simulasi penentuan pergerakan agen yang akan di uji adalah proses seeking atau pencarian titik tujuan. Pada bagian ini akan diperlihatkan kemampuan agen dalam mencari titik tujuannya atau destinasi. Pada Gambar 4.1 diperlihatkan hasil dari proses *seeking* dari setiap agen. Hasil dari proses *seeking* ini disimpan dalam bentuk *velocity* preferensi yang berisi kecepatan arah serta arah ke titik tujuan.



Gambar 4. 1 *Seeking*

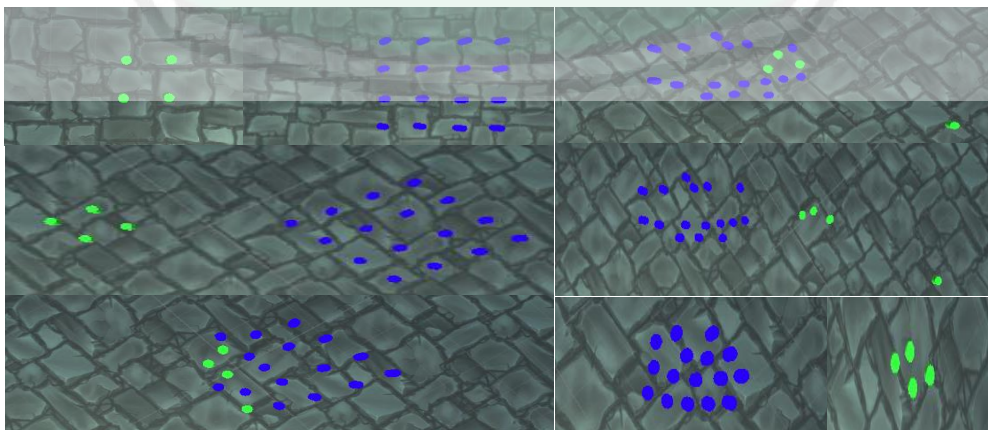
4.2.2 *Steering*

Setelah agen menentukan titik tujuannya dan telah mendapatkan *velocity* preferensi dari proses *seeking* maka proses selanjutnya yang diujikan adalah pengarahan agen menuju destinasiya yaitu *steering*.

Gambar 4. 2 *Steering*

4.2.3 *Collision Avoidance*

Pada tahap pengujian *collision avoidance*, agen yang telah mendapatkan *velocity* referensi akan melakukan proses pengujian keberhasilan dari *velocity* referensi tersebut dengan algoritma RVO. Pengujian dilakukan dengan menggerakkan kerumunan agen yang memiliki titik tujuan yang berlawanan. Sehingga akan mengakibatkan terjadinya tabrakan jika kedua agen tersebut tetap bergerak menggunakan kecepatan referensinya. Pada proses ini Proses navigasi penghindaran agen dengan RVO akan mengubah *velocity* agen diluar *velocity obstacle* yang mengakibatkan tabrakan. Hasil dari pengujian ini dapat dilihat pada Gambar 4.3.

Gambar 4. 3 Pengujian *collision avoidance*

4.2.4 Ekstraksi data

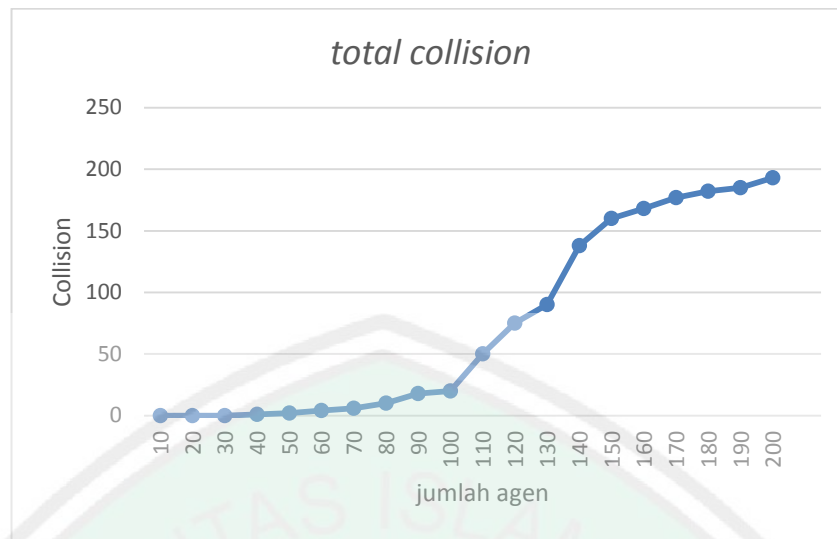
Pengujian ini dimaksudkan untuk memeriksa sistem pengambilan data dan memastikannya telah berjalan sesuai yang diinginkan. Pengujian ini dilakukan dengan mencoba seluruh fungsi dalam ekstraksi data dan memastikan apakah telah berjalan dengan baik atau tidak. Hasil dari pengujian seluruh fungsi berjalan sesuai seperti pada desain *sistem* yang dijelaskan pada bagian Bab 3. Hasil yang didapatkan dari pengujian kesesuaian sistem ini terdapat pada Tabel 4.1.

Tabel 4. 1 Ekstraksi Data

No	Nama Data	Fungsi dapat berjalan
1	Maksimal <i>frame</i>	Ya
2	Jumlah tabrakan	Ya
3	Waktu simulasi	Ya
5	Rata-rata <i>framerate</i>	Ya

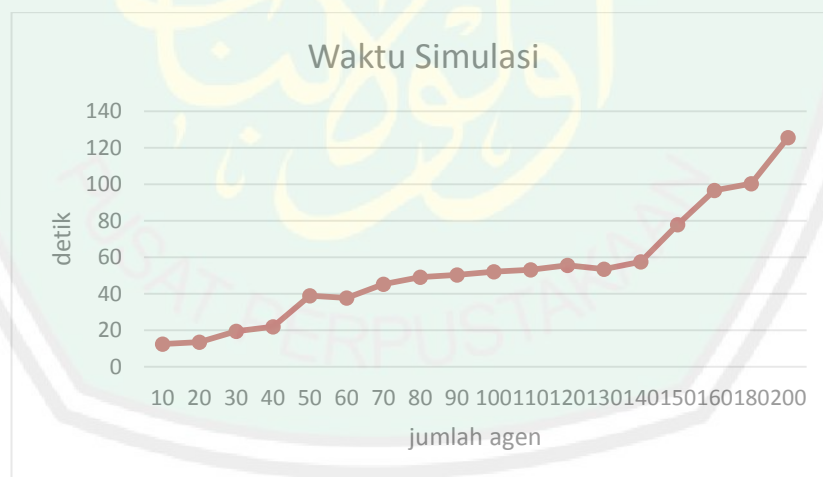
4.3.1 Pengujian Cross Scenario

Simulasi *cross scenario* mempertemukan dua kelompok kerumunan dengan arah tujuan berlawanan. Pertemuan kelompok agen ini bisa mengakibatkan tabrakan pada suatu tingkat keramaian kerumunan. Dari data yang didapat, terjadi peningkatan tabrakan secara eksponensial pada simulasi yang dapat dilihat di grafik pada Gambar 4.4. Hal ini berakibat penumpukan kerumunan karena keterbatasan lebar jalan yang bisa dilalui, dimana pada jumlah 200 agen terjadi *overlapping* antar agen karena kerumunan telah sangat berdesakan.



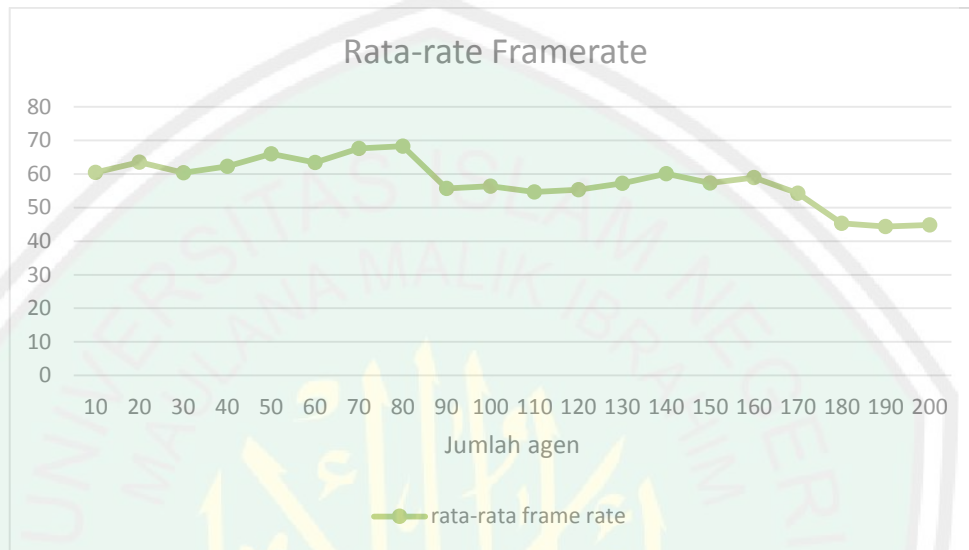
Gambar 4. 4 Grafik total *collision* pada *cross scenario*

Hasil simulasi *cross scenario* menunjukkan bahwa agen akan mengalami *collision* pada saat simulasi agen dilakukan dengan total 60 agen dan akan terus meningkat seiring dengan jumlah agen yang akan disimulasikan.



Gambar 4. 5 Grafik waktu simulasi pada *cross scenario*

Waktu simulasi yang digunakan dalam *cross scenario* cenderung lama, hal ini dikarenakan karena luas area simulasi yang digunakan relatif kecil. Seperti yang ditunjukkan pada Grafik 4.5 di atas, waktu simulasi meningkat secara signifikan pada 140 agen.

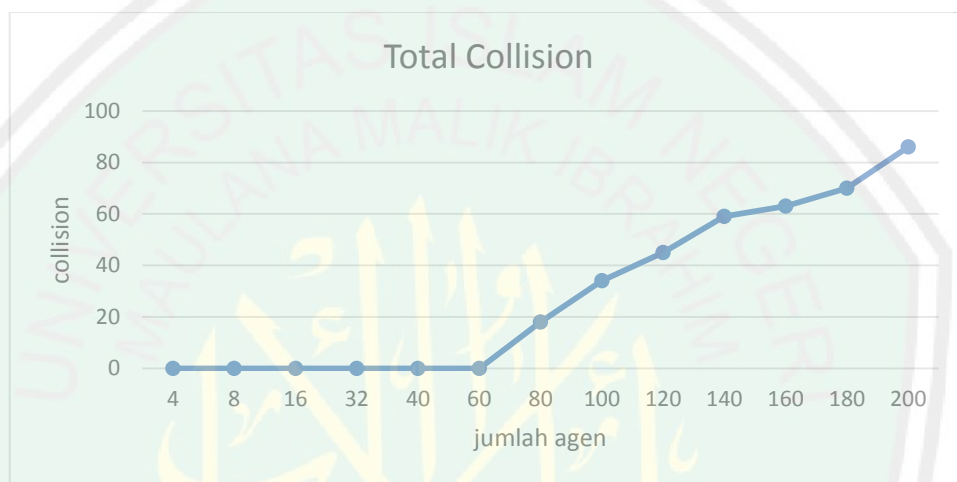


Gambar 4. 6 Grafik rata-rata *framerate*

Pada simulasi *cross* skenario rata-rata *framerate* yang dihasilkan cenderung stabil pada nilai 67.4 FPS seperti yang ditunjukkan pada Gambar 4.6 dan mengalami penurunan *framerate* yang cukup signifikan pada simulasi 90 agen. Hal ini dikarenakan semakin meningkatnya intensitas tabrakan antar agen sehingga menyebabkan penurunan *framerate*.

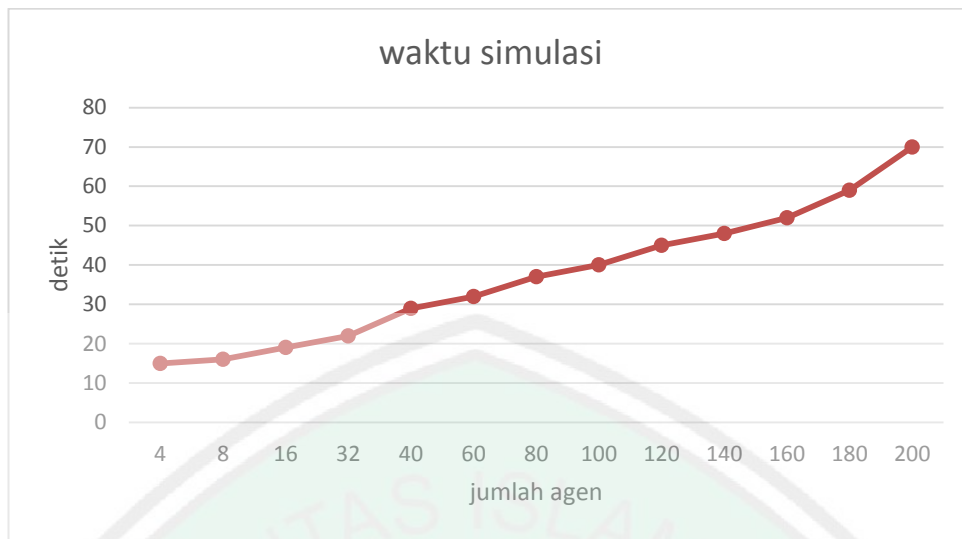
4.3.2 Pengujian Narrow Passage Scenario

Pada skenario *narrow passage*, empat kelompok di posisi pojok *environment* saling bertukar posisi. Di tengah jalur pertukaran posisi ini akan ada jalan sempit yang terbentuk dari empat buah rintangan berbentuk kotak. Tabrakan yang terjadi pada simulasi ini terjadi dimulai pada jumlah agen sebanyak 80 sebagaimana ditunjukkan pada grafik di Gambar 4.7.



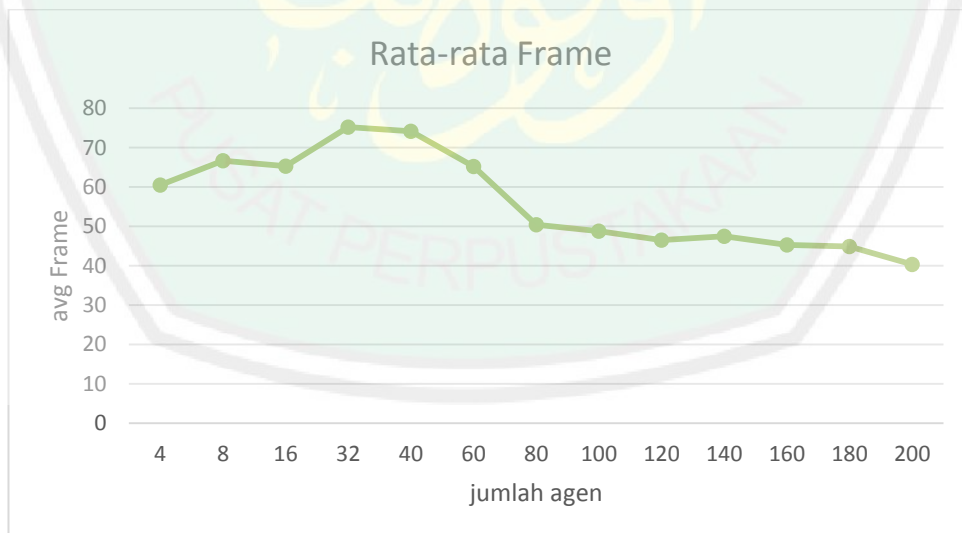
Gambar 4. 7 Grafik total collision pada Narrow Passage Scenario

Dari gambar Grafik 4.7 di atas, dapat diketahui bahwa agen akan mengalami *collision* pada saat simulasi dilakukan dengan 80 agen dan terus meningkat seiring dengan bertambahnya jumlah agen yang disimulasikan. Berbeda dengan *cross* skenario, pada *narrow passage scenario* jumlah agen yang saling bertabrakan semakin sedikit, hal ini dikarenakan wilayah area dari simulasi ini lebih luas jika dibandingkan dengan *cross* skenario.



Gambar 4. 8 Grafik waktu simulasi

Peningkatan waktu yang terjadi pada *narrow passage scenario* cenderung stabil seperti yang ditunjukkan dalam Gambar 4.8. berbeda dengan *cross scenario*, pada *narrow passage scenario* waktu yang diperlukan dalam melakukan satu periode simulasi cenderung lebih sedikit hal ini karena lebih terbukanya ruang atau area pada *narrow passage scenario*.

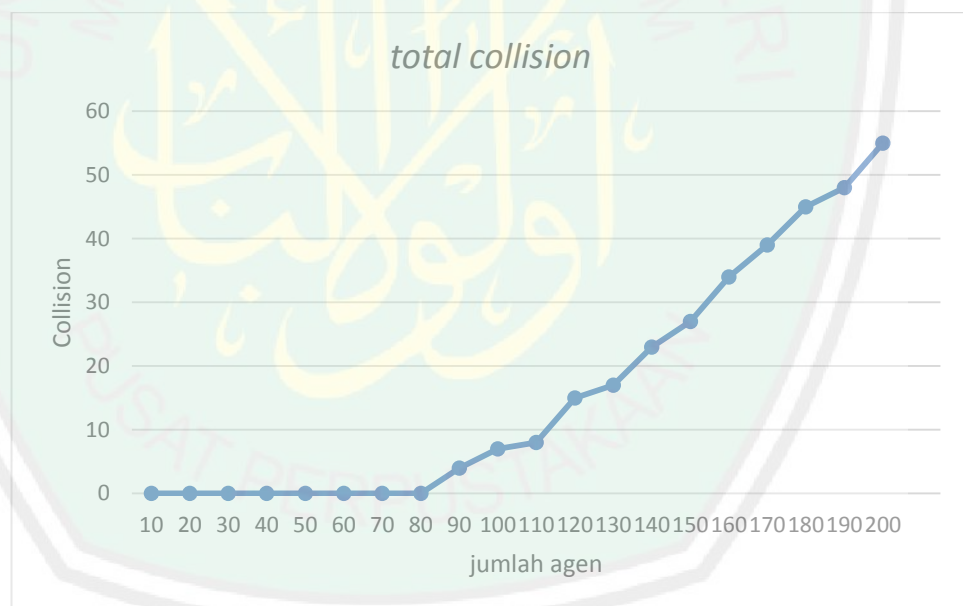


Gambar 4. 9 Grafik Rata-rata *framerate*

Pada *narrow passage scenario*, pada awal simulasi rata-rata *framerate* cenderung mengalami kenaikan secara bertahap seperti yang ditunjukkan pada gambar 4.9, namun pada simulasi 80 agen *framerate* mengalami penurunan secara signifikan hal ini dikarenakan terjadinya *collision* yang mengakibatkan *framerate* menjadi turun.

4.3.3 Pengujian Alun-alun Scenario

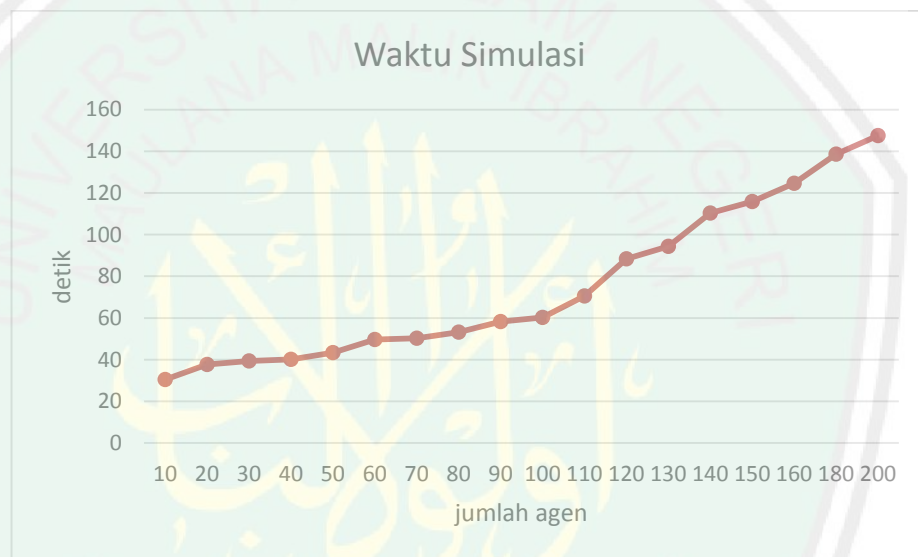
Pada skenario alun-alun, terdapat beberapa titik awal yang berada di luar wilayah simulasi sedangkan untuk titik akhir terdapat di dalam area simulasi dan luar area simulasi. Dalam penentuan titik awal dan tujuan setiap agen mendapatkan lokasi yang *random*.



Gambar 4. 10 Grafik total *collision* pada alun-alun skenario

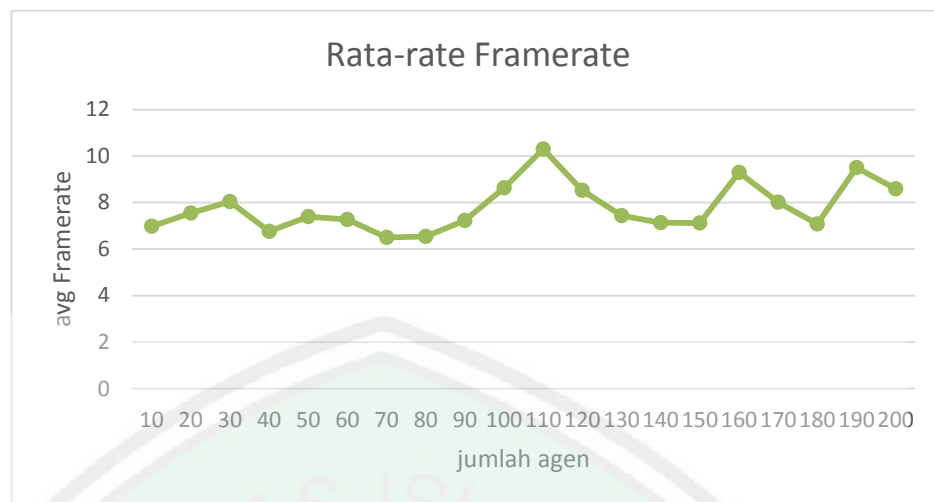
Dalam pengujian alun-alun skenario, agen mengalami tingkat tabrakan paling rendah dari dua skenario yang telah diujikan sebelumnya (*cross scenario* dan *narrow passage scenario*) hal ini dikarenakan kebebasan agen dalam bernavigasi

serta kepadatan kerumunan yang relatif rendah. Seperti pada gambar Grafik 4.10 diatas, ditunjukkan bahwa agen akan mengalami *collision* pada saat simulasi dilakukan dengan 90 agen dan terus meningkat seiring dengan bertambahnya jumlah agen yang disimulasikan. Berbeda dengan dua simulasi yang telah dilakukan sebelumnya, pada alun-alun skenario, jumlah agen yang mengalami *collision* cenderung sedikit. Hal ini dikarenakan luas dari area simulasi alun-alun yang lebih besar dari dua simulasi sebelumnya.



Gambar 4. 11 Waktu simulasi dari alun-alun skenario

Sama halnya seperti *cross scenario*, pada *narrow passage scenario* jumlah waktu yang dibutuhkan dalam simulasi juga akan bertambah lama bergantung pada jumlah agen yang akan disimulasikan. Namun pada simulasi alun-alun, jumlah waktu yang dibutuhkan setiap agen agar dapat mencapai titik tujuannya lebih lama dikarenakan area simulasi yang lebih luas dari dua simulasi sebelumnya.



Gambar 4. 12 Rata-rata *framerate* dari alun-alun simulasi

Pada skenario alun-alun, rata-rata *framerate* yang dihasilkan cenderung stabil dan tidak mengalami perubahan yang signifikan, hal ini dikarenakan pada skenario alun-alun *collision* lebih sedikit terjadi karena luas wilayah pada skenario alun-alun lebih besar sehingga agen dapat bergerak lebih bebas.

Berdasarkan uji coba yang dilakukan menunjukkan bahwa hasil percobaan dari setiap skenario simulasi hasilnya berubah-ubah. Perubahan tersebut dipengaruhi oleh beberapa faktor seperti jumlah agen dalam setiap simulasi yang berbeda-beda, jumlah *obstacle* dan luas area dari setiap skenario yang tidak sama serta waktu yang diperlukan dalam melakukan simulasi bervariasi.

Sama halnya dengan iman, iman seseorang juga dapat berubah-ubah terkadang naik dan terkadang turun hal ini juga dipengaruhi oleh beberapa faktor seperti lingkungan, sosial, budaya, ilmu dll. Naik turunnya iman dijelaskan dalam surat Al-Imron : 173 :

الَّذِينَ قَالَ لَهُمُ النَّاسُ إِنَّ النَّاسَ قَدْ جَمَعُوا لَكُمْ فَاخْشَوْهُمْ فَزَادَهُمْ إِيمَانًا وَقَالُوا حَسْبُنَا اللَّهُ

وَنِعْمَ الْوَكِيلُ

Artinya : ” (yaitu) orang-orang (yang mentaati Allah dan Rasul) yang kepada mereka ada orang-orang yang mengatakan, “Sesungguhnya manusia telah mengumpulkan pasukan untuk menyerang kamu, karena itu takutlah kepada mereka”, Maka Perkataan itu menambah keimanan mereka dan mereka menjawab: “Cukuplah Allah menjadi penolong Kami dan Allah adalah Sebaik-baik Pelindung”

Berdasarkan ayat di atas, para Ulama Ahlussunnah menjadikan ayat ini sebagai dasar mengenai bertambah dan berkurangan iman, sebagaimana pernah ditanyakan kepada Imam Sufyân bin ‘Uyainah rahimahullah, “Apakah iman itu bertambah atau berkurang? Beliau rahimahullah menjawab, “Tidakkah kalian mendengar firman Allah Azza wa Jalla yang artinya, ‘Maka Perkataan itu menambah keimanan mereka’”.

Dalam hadist Rasulullah SAW juga menjelaskan tentang naik turunnya iman yaitu :

الْإِيمَانُ بَضْعٌ وَسَبْعُونَ أَوْ بَضْعٌ وَسِتُّونَ شُعْبَةً فَأَفْضَلُهَا قَوْلُ لَا إِلَهَ إِلَّا اللَّهُ وَأَدْنَاهَا

إِمَاطَةُ الْأَدَى عَنِ الطَّرِيقِ وَالْحَيَاءُ شُعْبَةٌ مِنَ الْإِيمَانِ

Artinya: “Iman itu lebih dari tujuh puluh atau enam puluh cabang. Yang paling utama adalah ucapan, “Lâ Ilâha Illallâh” dan yang terendah adalah membersihkan gangguan dari jalanan. Dan rasa malu itu adalah satu cabang dari iman”. [Muttafaq ‘alaihi]

Hadits yang mulia ini menjelaskan bahwa iman memiliki cabang-cabang, ada yang tertinggi dan ada yang terendah . Cabang-cabang iman ini bertingkat-tingkat dan tidak berada dalam satu derajat dalam keutamaannya, bahkan sebagiannya lebih utama dari lainnya. Oleh karena itu, Imam at-Tirmidzi memuat Bab dalam sunannya, “Bab Penyempurnaan Iman, bertambah dan berkurangnya”.

Syaikh `Abdurrahmân as-Sa’di raimahullah dalam menjelaskan hadits ini menyatakan, “Ini jelas sekali menunjukkan bahwa iman itu bertambah dan berkurang sesuai dengan bertambahnya aturan syariat dan cabang-cabang iman, serta amalan hamba tersebut atau tidak mengamalkannya. Sudah dimaklumi bersama bahwa manusia bertingkat-tingkat dalam hal ini. Siapa yang berpendapat bahwa iman itu tidak bertambah dan berkurang, maka telah menyelisihi realita yang nyata di samping menyelisihi nash-nash syariat sebagaimana telah diketahui.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan yaitu dengan melakukan simulasi terhadap perilaku pergerakan kerumunan agen pada alun-alun Kota Batu, maka dapat disimpulkan bahwa algoritma *reciprocal velocity obstacle* dapat diterapkan dalam navigasi multi agen, dimana setiap agen akan menavigasi dan memilih rute secara mandiri sehingga memungkinkan untuk melakukan penghindaran tabrakan antar agen tanpa adanya komunikasi satu sama lain.

Pada umumnya dalam setiap skenario simulasi, tabrakan akan terjadi pada tingkat jumlah kerumunan tertentu. Tabrakan terjadi karena kepadatan kerumunan yang tinggi sehingga menghasilkan perhitungan yang lebih kompleks untuk navigasi agen. Hal ini terjadi karena meningkatnya jumlah agen yang perlu ditentukan pergerakannya. Sedangkan waktu yang diperlukan untuk melakukan satu periode simulasi akan terus meningkat seiring dengan banyaknya jumlah agen yang disimulasikan, selain itu luas area simulasi dan banyaknya jumlah *obstacle* juga mempengaruhi waktu yang diperlukan untuk melakukan satu periode simulasi.

Dari hasil pengujian RVO pada beberapa skenario simulasi kerumunan yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

1. Hasil simulasi *cross scenario* menggunakan RVO menghasilkan tabrakan mulai dari jumlah agen sebanyak 60 agen dan meningkat dengan tingkat kenaikan 61.5% dengan jumlah total 200 agen.

2. Hasil simulasi *narrow passage scenario* menggunakan RVO menghasilkan tabrakan mulai dari jumlah agen sebanyak 80 agen dan dan meningkat dengan tingkat kenaikan 64% dengan jumlah total 200 agen.
3. Hasil simulasi *alun-alun scenario* menggunakan RVO menghasilkan tabrakan mulai dari jumlah agen sebanyak 90 agen dan dan meningkat dengan tingkat kenaikan 24% dengan jumlah total 200 agen. Pada skenario alun-alun, tabrakan terjadi paling sedikit dibandingkan dengan *cross scenario* dan *narrow passage scenario* hal ini dikarenakan area wilayah simulasi yang lebih luas dan agen yang kemunculanya dibuat *random* dan tidak bersama-sama.

5.2 Saran

Dalam pembuatan simulasi ini masih mempunyai banyak sekali kekurangan yang diharapkan nantinya perlu untuk dilakukan pengembangan, diantaranya :

1. Pengujian simulasi dengan jumlah agen simulasi lebih banyak pada skenario dan kondisi yang berbeda menggunakan komputer dengan spesifikasi yang lebih tinggi.
2. Simulasi membutuhkan tambahan *obstacle* yang lebih bagus dan bervariasi sehingga apa yang akan disampaikan akan mudah dipahami dan terkesan lebih nyata.
3. Pembuatan dan pengembangan simulasi secara bertahap dengan level yang berbeda sehingga selain lebih bervariasi juga dapat menampilkan simulasi yang lebih baik.

Daftar Pustaka

- Badrul, Muhammad.2017.*Simulasi pengelolaan kerumunan massa demonstran dengan Crowd Dinamic Model*. Vol.4 No.3 : 1-2.
- Fahri, Moch.2016.*Simulasi Kerumunan Menggunakan Navigasi Multiagen Berbasis Reciprocal Velocity Obstacles dengan Relasi Leader Follower*. Tugas Akhir. Tidak diterbitkan. Fakultas Teknologi Industri : Surabaya.
- F, Feurtey. 2000. *Simulating the collision avoidance behavior of pedestrians*.Master's thesis. University of Tokyo.
- J, van den Berg. M, Lin. D, Manocha, 2008. *Reciprocal Velocity Obstacles For Real-time Multiagent Navigation*. USA: University of North California.
- Kurniawati, Arik dkk.2010.*Simulasi pergerakan pengunjung mall menggunakan potential field*.Jurnal Ilmiah.Vol.5 No.3: 2.
- P, Fiorini dan Z, Shiller.1998.*Motion planning in dynamic environments using velocity obstacles*. Int. Journal of Robotics Research.Vol. 17 no. 7: 760–772.
- Sarmady, Siamak. 2008. *Modeling and Simulation of Movement and Behaviors in Large Crowd Using Cellular Automata*, Master's thesis. Universiti Sains Penang: Malaysia.
- Undang-Undang Kepariwisataa No.9 Tahun 1990
- Wolinski, David. 2016. *Microscopic Crowd Simulation : Evaluation And Development Of Algorithms*. Data Structures and Algorithms. Universit'e Rennes
- Xu,Ming-Liang.2014.*Crowd Simulation and Its Applications: Recent Advances*. Journal of Computer Science and Tecnology. 29(5): 799-811.
- Y, Matsuo. Y, Abe.2001.*Collision avoidance method for multiple autonomous mobile agents by implicit cooperation*. Proc. IEEE Int. Conf. on Robotics and Automation.