

**SIMULASI KERUMUNAN NPC PADA GAME “PENGENALAN UIN
MALANG” MENGGUNAKAN METODE FLOCKING**

SKRIPSI

**Oleh:
DICKY ARYA PRATAMA
NIM. 19650008**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**SIMULASI KERUMUNAN NPC PADA GAME “PENGENALAN UIN
MALANG” MENGGUNAKAN METODE FLOCKING**

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
DICKY ARYA PRATAMA
NIM. 19650008

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN


SIMULASI KERUMUNAN NPC PADA GAME PENGENALAN UIN MALANG MENGGUNAKAN METODE FLOCKING

SKRIPSI

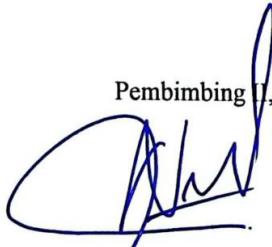
Oleh:
DICKY ARYA PRATAMA
NIM. 19650008

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal:

Pembimbing I,



Dr. Yunifa Miftachul Arif, M. T
NIP. 19830616201101 1 004

Pembimbing II,


Dr. Fresy Nugroho, M. T
NIDT. 2022077103

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




Dr. Irfanul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN





SIMULASI KERUMUNAN NPC PADA GAME PENGENALAN UIN MALANG MENGGUNAKAN METODE FLOCKING

SKRIPSI

Oleh:
DICKY ARYA PRATAMA
NIM. 19650008

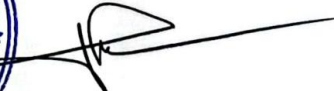
Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal:

Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. M. Faisal, M.T</u> NIP. 19740510 200501 1 007	()
Anggota Penguji I	: <u>Hani Nurhayati, M.T</u> NIP. 19780625 200801 2 006	()
Anggota Penguji II	: <u>Dr. Yunifa Miftachul Arif, M.T</u> NIP. 19830616 201101 1 004	()
Anggota Penguji III	: <u>Dr. Fresy Nugroho, M.T</u> NIP. 19710722 201101 1 001	()

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




Dr. Achrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Dicky Arya Pratama

NIM : 19650008

Fakultas / Prodi : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Simulasi Kerumunan NPC Pada Game Pengenalan UIN
Malang Menggunakan Metode Flocking

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, 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 merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang,
Yang membuat pernyataan,



Dicky Arya Pratama
NIM.19650008

MOTTO

“Lakonah lakoneh, kennengngah kennengneh”

HALAMAN PERSEMBAHAN

Dipersembahkan untuk diriku sendiri, yang telah tumbuh dan berkembang sepanjang perjalanan ini. Terima kasih kepada Ibu dan adik-adikku yang selalu memberikan dukungan tak terbatas selama masa penulisan dan perjalanan pendidikanku.

Tak lupa untuk teman-teman di grup Ahlussunnah wal Jamaah, terima kasih atas motivasi dan dukungan kalian yang tak pernah berhenti selama proses penelitian ini. Kalian adalah pendorong semangat yang membuat perjalanan ini jauh lebih berwarna.

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT, karena atas rahmat dan hidayah-Nya, skripsi ini dapat terselesaikan. Shalawat serta salam senantiasa tercurah kepada junjungan kita, Nabi Muhammad SAW, yang telah menjadi suri tauladan bagi seluruh umat manusia. Skripsi ini berjudul "Simulasi Kerumunan NPC pada Game Pengenalan UIN Malang Menggunakan Metode Flocking."

Penulisan skripsi ini bertujuan untuk menggali dan mengembangkan pemahaman mengenai simulasi kerumunan dalam dunia game, khususnya pada lingkungan UIN Malang. Terlebih lagi, metode Flocking digunakan sebagai pendekatan untuk memodelkan perilaku kerumunan non-player character (NPC) dalam game tersebut. Proses penelitian ini merupakan perjalanan yang penuh dengan tantangan, pembelajaran, dan eksplorasi konsep-konsep baru.

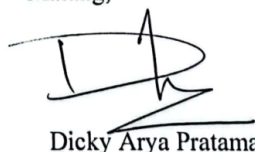
Penulis menyadari bahwa penyelesaian skripsi ini tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, dalam kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang setinggi-tingginya kepada:

1. Allah SWT, yang telah memberikan rahmat, hidayah, dan kekuatan selama proses penulisan skripsi ini.
2. Keluarga Tercinta, khususnya kepada Ibu dan keluarga yang selalu memberikan doa, dukungan, dan kasih sayang yang tak terhingga.
3. Dosen Pembimbing, atas bimbingan, arahan, dan masukan yang sangat berharga dalam menyusun skripsi ini.
4. Teman-teman Mahasiswa, yang selalu memberikan semangat, inspirasi, dan pengalaman menarik sepanjang perjalanan perkuliahan.

Semua dukungan dan kontribusi dari pihak-pihak tersebut menjadi pendorong utama penyelesaian skripsi ini. Meskipun masih jauh dari kesempurnaan, penulis berharap bahwa skripsi ini dapat memberikan kontribusi positif, terutama dalam pengembangan dunia game di lingkungan UIN Malang.

Akhir kata, penulis menyadari bahwa setiap kritik dan saran yang membangun akan sangat berarti untuk pengembangan diri. Semoga skripsi ini bermanfaat dan dapat memberikan inspirasi bagi pembaca yang tertarik dalam bidang pengembangan game. Terima kasih.

Malang,



Dicky Arya Pratama

DAFTAR ISI

HALAMAN PENGAJUAN	2
HALAMAN PERSETUJUAN.....	3
HALAMAN PENGESAHAN	4
PERNYATAAN KEASLIAN TULISAN	5
MOTTO	6
HALAMAN PERSEMBAHAN.....	7
KATA PENGANTAR.....	8
DAFTAR ISI.....	10
DAFTAR GAMBAR.....	12
DAFTAR TABEL.....	14
ABSTRAK	15
ABSTRACT	16
مستخلص البحث	17
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	6
1.3 Tujuan Penelitian	6
1.4 Batasan Masalah.....	6
1.5 Manfaat Penelitian	7
BAB II KAJIAN PUSTAKA	8
2.1 Penelitian Terkait	8
2.2 Simulasi.....	9
2.3 Game	10
2.4 NPC	12
2.5 Usability Testing	14
2.6 System Usability Scale (SUS).....	15
2.7 Flock (Kerumunan)	17
BAB III METODE PENELITIAN	23
3.1 Tahapan Penelitian	23
3.2 Storyboard	24
3.3 Desain Sistem.....	29
3.4 Rancangan Pengujian Sistem	34
3.4.1 Perbandingan Waktu Tempuh Kerumunan NPC Pada Penggunaan Metode Flocking.....	34
3.4.2 Perbandingan Kemampuan Kerumunan NPC Menghadapi Hambatan Statis Menggunakan Metode Flocking.....	36
3.4.3 Pengujian Tingkat Usability Metode Flocking Pada Kerumunan NPC	37
3.5 Perhitungan Manual	40
3.5.1 Cohesion	44
3.5.2 Alignment	45
3.5.3 Separation	46
BAB IV	
HASIL DAN PEMBAHASAN	48

4.1 Implementasi Sistem	48
4.2 Pengujian Sistem.....	54
4.2.1 Perbandingan waktu tempuh pada penggunaan metode flocking.....	54
4.2.1.1 Perbandingan waktu tempuh kerumunan NPC tanpa hambatan	54
4.2.1.2 Perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC tanpa metode flocking.....	61
4.2.1.3 Perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC dengan metode flocking.....	67
4.2.1.4 Perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC dengan metode flocking dengan FSM	74
4.2.2 Perbandingan Kemampuan Kerumunan NPC Menghadapi Hambatan Statis Menggunakan Metode Flocking.....	80
4.2.3 Pengujian Tingkat Usability Metode Flocking Pada Kerumunan NPC	74
4.3 Analisis.....	83
4.3.1 Analisis Data Perbandingan Waktu Tempuh Pada Penggunaan Metode Flocking.....	83
4.3.2 Analisis Data Kemampuan Kerumunan NPC Menghadapi Hambatan Statis Menggunakan Metode Flocking.....	84
4.3.3 Analisis Data Pengujian Tingkat Usability Metode Flocking	85
4.4 Integrasi Islam.....	87
BAB V	89
KESIMPULAN DAN SARAN	89
5.1 Kesimpulan	89
5.2 Saran.....	90
DAFTAR PUSTAKA	91

DAFTAR GAMBAR

Gambar 2.1 <i>System usability scale</i>	15
Gambar 2.2 Perilaku <i>Cohesion</i>	17
Gambar 2.3 Perilaku <i>Alignment</i>	18
Gambar 2.4 Perilaku <i>Separation</i>	19
Gambar 3.1 Alur tahapan penelitian	21
Gambar 3.2 <i>Design System</i>	25
Gambar 3.3 Alur NPC	26
Gambar 3.4 Diagram blok cara kerja <i>flocking</i>	26
Gambar 3.5 Penerapan <i>Flocking</i>	27
Gambar 3.6 FSM NPC	28
Gambar 3.7 Skema pengujian waktu tempuh kerumunan NPC	29
Gambar 3.9 Alur Pengujian <i>System Usability Scale</i>	32
Gambar 3.10 Penerimaan nilai skor SUS keseluruhan	34
Gambar 4.1 <i>Game Object Flock script</i>	42
Gambar 4.2 <i>GameObject FlockAgent script</i>	43
Gambar 4.3 <i>Flocking NPC Spawn</i>	44
Gambar 4.4 <i>Flocking Movement FSM</i>	44
Gambar 4.5 <i>Flocking Movement FSM Idle</i>	45
Gambar 4.6 <i>Flocking Movement FSM Search and Run</i>	46
Gambar 4.7 <i>Flocking Movement FSM Walk</i>	47
Gambar 4.8 Hasil Waktu tempuh kerumunan NPC GD > FH Tanpa Hambatan <i>Flock Berlawanan Arah</i>	48
Gambar 4.9 Hasil Waktu tempuh kerumunan NPC GD > Saintek Tanpa Hambatan <i>Flock Berlawanan Arah</i>	50
Gambar 4.10 Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah Tanpa Hambatan <i>Flock Berlawanan Arah</i>	52
Gambar 4.11 Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan <i>Flock Berlawanan Arah tanpa metode flocking</i>	55
Gambar 4.12 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan <i>Flock Berlawanan Arah tanpa metode flocking</i>	57
Gambar 4.13 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan <i>Flock Berlawanan Arah tanpa metode flocking</i>	59
Gambar 4.14 Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan <i>Flock Berlawanan Arah menggunakan metode flocking</i>	61
Gambar 4.15 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan <i>Flock Berlawanan Arah menggunakan metode flocking</i>	63
Gambar 4.16 Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan Hambatan <i>Flock Berlawanan Arah menggunakan metode flocking</i>	65
Gambar 4.17 Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan <i>Flock Berlawanan Arah menggunakan metode flocking FSM</i>	67
Gambar 4.18 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan <i>Flock Berlawanan Arah menggunakan metode flocking FSM</i>	69

Gambar 4.19 Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan
Hambatan *Flock* Berlawanan Arah menggunakan metode *flocking* FSM..... 71

DAFTAR TABEL

Tabel 3.2 Pertanyaan kuesioner System Usability Scale (SUS)	32
Tabel 3.3 Posisi Agen	35
Tabel 4.1 Analisa perbandingan waktu tempuh kerumunan NPC GD->FH tanpa hambatan	49
Tabel 4.2 Analisa perbandingan waktu tempuh kerumunan NPC GD->Saintek tanpa hambatan.....	51
Tabel 4.3 Analisa perbandingan waktu tempuh kerumunan NPC GD->Tarbiyah tanpa hambatan.....	53
Tabel 4.4 Analisa perbandingan waktu tempuh kerumunan NPC GD->FH dengan hambatan kerumunan NPC tanpa metode <i>flocking</i>	56
Tabel 4.5 Analisa perbandingan waktu tempuh kerumunan NPC GD-> Saintek dengan hambatan kerumunan NPC tanpa metode <i>flocking</i>	58
Tabel 4.6 Analisa perbandingan waktu tempuh kerumunan NPC GD -> Tarbiyah dengan hambatan kerumunan NPC tanpa metode <i>flocking</i>	60
Tabel 4.7 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i>	62
Tabel 4.8 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i>	64
Tabel 4.9 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i>	66
Tabel 4.10 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i> FSM.....	68
Tabel 4.11 Analisa perbandingan waktu tempuh kerumunan NPC GD -> Saintek dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i> FSM.....	70
Tabel 4.12 Analisa perbandingan waktu tempuh kerumunan NPC GD -> Saintek dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i> FSM.....	72
Tabel 4.13 Data hasil survey yang diolah menggunakan metode System Usability Scale (SUS)	74
Tabel 4.14 Hasil perhitungan skor menggunakan metode System Usability Scale	79

ABSTRAK

Pratama, D. A. 2023. **Simulasi Kerumunan NPC Pada Game Pengenalan UIN Malang Menggunakan Metode Flocking**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Yunifa Miftachul Arif, M.T (II) Dr. Fresy Nugroho, M.T .

Kata kunci: simulasi, kerumunan, NPC. *game*, pendidikan

Penelitian ini mengimplementasikan metode *flocking* untuk mensimulasikan perilaku kerumunan pada NPC dalam game "Pengenalan UIN Malang" untuk menciptakan perilaku berkerumun yang realistis. Meskipun kerumunan NPC tanpa metode memiliki waktu tempuh lebih cepat, kerumunan NPC dengan metode *flocking* memberikan perilaku berkerumun yang interaktif dengan tingkat usability yang baik. Hasil evaluasi menggunakan System Usability Testing menunjukkan skor usability yang memadai. Implementasi ini berhasil menciptakan pengalaman berinteraksi yang lebih baik, meskipun waktu tempuhnya tidak secepat NPC tanpa metode.

ABSTRACT

Pratama, D. A. 2023. **Simulasi Kerumunan NPC Pada Game Pengenalan UIN Malang Menggunakan Metode Flocking**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University, Malang. Mentor: (I) Dr. Yunifa Miftachul Arif, M.T (II) Dr. Fresy Nugroho, M.T .

Keywords: simulation, crowd, NPC. games, education

This research implements the flocking method to simulate crowd behavior on NPCs in the game "Introduction to UIN Malang" to create realistic crowding behavior. Although NPC crowding without a method has a faster travel time, NPC crowding with the flocking method provides interactive swarming behavior with a good level of usability. Evaluation results using System Usability Testing show adequate usability scores. This implementation succeeded in creating a better interaction experience, although the travel time was not as fast as NPCs without the method.

مستخلص البحث

برائما، د. أ. (2023). محاكاة حشد الشخصيات غير القابلة للعب في لعبة تعريف عن جامعة مولانا مالك إبراهيم مالانج باستخدام طريقة الحشد. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. يونيفا مفتاح العارف، الماجستير. المشرف الثاني: د. فريسي نوغروهو، الماجستير.

الكلمات الرئيسية: محاكاة، حشد، شخصية غير قابلة للعب، ألعاب، تعليم.

يطبق هذا البحث طريقة الحشد لمحاكاة سلوك الحشود على الشخصيات غير القابلة للعب في لعبة "تعريف عن UIN Malang" لخلق سلوك ازدحام واقعي. على الرغم من أن حشود NPC بدون أي طريقة كانت لها وقت السفر أسرع، إلا أن حشود NPC باستخدام طريقة الحشد قدمت سلوك سرب تفاعلي بمستوى جيد من سهولة الاستخدام. أظهرت نتائج التقييم باستخدام اختبار قابلية استخدام النظام درجة مناسبة لقابلية الاستخدام. تمكن هذا التطبيق من خلق تجربة تفاعل أفضل، على الرغم من أن وقت السفر لم يكن بنفس سرعة الشخصيات غير القابلة للعب بدون أي طريقة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengertian Simulasi sendiri adalah suatu metodologi untuk melaksanakan percobaan dengan menggunakan model dari satu sistem nyata (Siagian, 1987). Sedangkan pengertian game simulasi menurut Adams (dalam Tatiek Romlah, 2006: 118) adalah *game* untuk merefleksikan situasi-situasi dari kehidupan sebenarnya. Untuk menerapkan refleksi situasi nyata pada keseharian yang biasa terjadi pada lingkungan belajar Universitas Islam Negeri Malang pada game “Pengenalan UIN Malang”, diperlukan sekelompok NPC yang berjalan secara berkerumun untuk merefleksikan situasi nyata orang yang berlalu-lalang di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

NPC (*Non-Player Character*) adalah karakter *non-player* dalam game yang dikendalikan oleh komputer. NPC merupakan objek yang biasa ditemui di berbagai macam *genre* game, baik itu game petualangan, RPG (*Role Play Game*), maupun game simulasi. Dalam sebuah game, NPC digunakan sebagai pemandu, lawan main, atau sebagai karakter yang memberikan misi (Siswanto & Suni, 2021). Dalam setiap game, NPC memiliki perilaku yang dijalankan secara otomatis sesuai dengan tujuan pembuatan NPC tersebut. Namun, dalam konteks simulasi kerumunan pada NPC, terdapat masalah yang harus diatasi

Pemberian NPC pada game “Pengenalan UIN Malang” bertujuan untuk mensimulasikan kerumunan manusia yang berlalu-lalang di lingkungan Universitas Islam Negeri Maulana Malik Ibrahim Malang. Dalam game, NPC bergerak secara

individu dan dapat saling berinteraksi. Hal ini dapat memungkinkan terjadinya tabrakan antar NPC. Alhasil, NPC yang bergerak secara individu dapat menghasilkan pola kerumunan yang saling menabrak dan tidak beraturan yang dalam konteks ini tidak merefleksikan situasi kerumunan yang terjadi pada Universitas Islam Negeri Maulana Malik Ibrahim Malang. Maka dari itu, untuk mensimulasikan kerumunan manusia pada NPC diperlukan suatu perilaku yang dapat membuat NPC dapat bergerak secara berkerumun seperti situasi aslinya.

Dalam kasus simulasi kerumunan, penghindaran tabrakan antara objek dan rintangan (*Obstacle*) sangat penting dikembangkan. Terdapat beberapa metode yang dapat digunakan oleh peneliti sebelumnya untuk menerapkan perilaku kerumunan pada NPC. Sharmady pada penelitiannya menggunakan model seluler automata untuk mensimulasikan jama'ah tawaf dan juga menggunakan model *discrete-event* untuk simulasi perilaku dan tindakan jama'ah (Sarmady et al., 2011). Fadila pada penelitiannya (Fadila & Arif, 2020) menggunakan algoritma RVO (*Reciprocal velocity obstacle*) untuk membuat NPC dapat bergerak menghindari objek. Dalam penelitiannya, Fikri et al. (2023) menggunakan algoritma *boids* pada efek partikel untuk mengaktifkan pergerakan dinamis dan menghindari tabrakan dengan lingkungan sekitarnya. Hasil penelitian ini menunjukkan bahwa penerapan algoritma *boids* mampu mencegah terjadinya tabrakan antara efek partikel dengan karakter dalam permainan. Maulani pada penelitiannya (Maulani et al., 2019) menggunakan algoritma *boids* untuk perilaku kerumunan pada NPC. Hasil dari penelitian tersebut menunjukkan tidak terjadi tabrakan antar *boids*. Selain itu, disimpulkan juga semakin jauh jarak antar *boids* semakin besar juga kemungkinan

player menghindari kerumunan NPC (*boids*) tersebut. Sedangkan Kim, pada penelitiannya (Kim et al., 2015) untuk menghindari tabrakan menggunakan *velocity based model* dan FSM yang memungkinkan agen untuk menghindari tabrakan dengan cara mengantisipasi tabrakan dan mengambil tindakan untuk menghindari tabrakan tersebut. Djamaludin pada penelitiannya (Djamaludin, 2016) menggunakan metode *boids* dan *artificial bee colony* untuk simulasi jama'ah yang mengelilingi ka'bah.

Dalam bukunya, Daniel Thalmann menjelaskan bahwa ada banyak cara untuk mensimulasikan kerumunan, terutama cara untuk menghindari tabrakan, salah satunya adalah *flocking* (Musse et al., 2013). metode *Flocking* adalah suatu teknik simulasi dalam komputer grafik yang menirukan perilaku burung atau hewan lain dalam kelompok yang bergerak secara bersama-sama dan terorganisir. Metode *Flocking* terinspirasi dari perilaku kelompok/kawanan burung. metode *Flocking* memiliki tiga perilaku dasar untuk pola koordinasi *flocking*. Kohesi (*Cohesion*) merupakan salah satu perilaku dasar *flocking*, kohesi memaksa agen (NPC) untuk tetap berkelompok dengan kerumunannya dengan cara pergi ke titik tengah kelompok tersebut. Penyelarasan (*Alignment*) merupakan perilaku dasar kedua dari metode *flocking*. Dalam perilaku ini, tiap agen akan diselaraskan untuk bergerak ke arah yang sama. Penghindaran (*Avoidance*) merupakan perilaku dasar terakhir dari metode *flocking*. Perilaku ini berfungsi untuk mencegah tabrakan antar agen. Metode tersebut dikembangkan oleh Reynold dengan memberi beberapa perilaku kendali seperti mengikuti pemimpin (*leader following*) dan menghindari hambatan (*obstacle avoidance*) (Reynolds, 1999). Meilany dalam penelitiannya (Dewi et al.,

2011) mensimulasikan arus orang yang meninggalkan *mall* melalui pintu masuk utama menggunakan metode *flocking*. Metode *Flocking* digunakan untuk memastikan bahwa tidak terjadi tabrakan individu keluar dari *pintu mall* dengan individu lain. Hampir sama dengan konteks tersebut, (Nugroho et al., 2010) Untuk meniru pergerakan kerumunan dalam keadaan panik, peneliti menggunakan metode *flocking* untuk memungkinkan individu untuk dengan cepat keluar dari gedung ketika terjadi kebakaran. Peneliti memodelkan bagaimana menjaga jarak antar manusia untuk pergerakan individu dalam kerumunan, dimana pergerakan tersebut harus mampu memberikan aksi secara efisien dalam menghadapi kondisi untuk menuju aksi selanjutnya.

Berdasarkan hasil studi tentang metode *Flocking* tersebut, metode *Flocking* dapat menjalankan agen dalam kerumunan dan mensimulasikan bagaimana agen berperilaku dalam kelompok sambil mempertahankan jarak yang sesuai di antara mereka untuk mencegah tabrakan. Pada penelitian yang berfokus pada simulasi kerumunan NPC, peneliti menggunakan metode *Flocking* untuk memberikan perilaku kerumunan pada NPC pada game “Pengenalan UIN Malang” untuk menghasilkan NPC yang dapat bergerak dalam kerumunan secara alami sesuai situasi yang ada pada lingkungan Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Ayat Al-Qur'an menyatakan bahwa Allah telah menetapkan bagi setiap sesuatu dengan tepat. Hal ini sejalan dengan konsep bahwa segala sesuatu yang diciptakan Allah telah diberikan ukuran dan takdir yang telah ditetapkan-Nya.

لَّذِي لَهُ مُلْكُ السَّمَوَاتِ وَالْأَرْضِ وَلَمْ يَتَّخِذْ وَلَدًا وَلَمْ يَكُنْ لَهُ شَرِيكٌ فِي الْمُلْكِ وَخَلَقَ كُلَّ شَيْءٍ فَقْدَرَهُ
تَقْدِيرًا

“Yang kepunyaan-Nya-lah kerajaan langit dan bumi, dan Dia tidak mempunyai anak, dan tidak ada sekutu bagi-Nya dalam kekuasaan(Nya), dan dia telah menciptakan segala sesuatu, dan Dia menetapkan ukuran-ukurannya dengan serapi-rapinya” (Q.S Al-Furqan: 25)

Menurut tafsir Al-Mukhtashar di bawah pengawasan Syaikh Dr. Shalih bin Abdullah bin Humaid (Imam Masjidil Haram) dijelaskan bahwa, hanya Dia yang memiliki kerajaan langit dan bumi, Dia tidak beranak, dan tidak pula memiliki sekutu dalam kekuasaan-Nya. Dia menciptakan segala sesuatu, dan Dia menetapkan takdir dan ukuran penciptaan makhluk-Nya sesuai kehendak ilmu dan kebijaksanaan-Nya, yang masing-masing cocok dengan itu. Dari tafsir yang telah dijelaskan, dapat disimpulkan bahwa algoritma flocking merupakan upaya untuk menerapkan perilaku berkumpul burung yang telah diciptakan Allah sesuai dengan kehendak ilmu dan kebijaksanaan-Nya dapat diaplikasikan dalam pengembangan game sebagai suatu usaha untuk mencapai kebaikan.

1.2 Pernyataan Masalah

Berdasarkan latar belakang penelitian, permasalahan yang ingin dipecahkan dalam penelitian ini adalah Bagaimana menghasilkan simulasi kerumunan menggunakan metode flocking dan bagaimana tingkat usability game “Pengenalan UIN Malang”?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengukur tingkat *usability* metode *flocking* pada kerumunan NPC pada game “Pengenalan UIN Malang” menggunakan metode Sistem Usability Scale

1.4 Batasan Masalah

Agar tujuan dari penelitian dapat tercapai dengan baik, diperlukan batasan agar pembahasan dari penelitian ini berfokus pada pokok permasalahan yang ingin diselesaikan oleh penelitian ini, batasan masalah dari penelitian ini adalah sebagai berikut:

1. NPC yang menggunakan metode *flocking* pada penelitian ini hanya terbatas pada jenis NPC pejalan kaki saja.
2. Jumlah NPC yang digunakan untuk mengimplementasikan perilaku kerumunan pada penelitian ini adalah sekitar 3 hingga 5 NPC per kerumunan.

1.5 Manfaat Penelitian

1. Dapat meningkatkan kualitas permainan game "Pengenalan UIN Malang" dengan memberikan pengalaman bermain yang lebih interaktif bagi para pemain
2. Penelitian ini dapat memberikan solusi bagi masalah dalam pengembangan game yang memerlukan object (NPC) berkelompok dengan jumlah yang banyak.
3. Menambah referensi dalam optimasi perilaku multi-objek pada game

BAB II

KAJIAN PUSTAKA

Dalam kajian pustaka, peneliti akan mengulas penelitian terkait dan teori-teori yang menjadi dasar dalam pelaksanaan penelitian.

2.1 Penelitian Terkait

Dalam jurnal penelitian yang berjudul "Pengaturan Formasi Robot Mobil Berdasarkan Pendekatan Virtual Structure Dengan Penghindaran Rintangan" yang ditulis oleh (Apriyanto, 2018), dijelaskan bahwa peneliti telah membuat suatu sistem pengaturan formasi pada robot. Untuk menghindari terjadinya tabrakan antara robot ketika membentuk formasi saat bergerak menuju tujuan, peneliti menggunakan model *flocking* dengan metode *artificial potential field*. Hasil penelitian menunjukkan bahwa penggunaan metode *artificial potential field* pada model *flocking* dapat digunakan untuk penghindaran tabrakan antar robot. Namun dalam penelitian tersebut tidak dijabarkan bagaimana pengaruh metode *flocking* untuk menghindari objek yang bergerak secara dinamis.

Penelitian yang berjudul "Penerapan *Flocking Algorithm* pada Game Simulasi Beternak Hewan" (Prahmana et al, 2018) menggunakan algoritma *flocking* untuk mengatur kerumunan kuda pada game yang diuji. Konsep *leader-follower* digunakan dalam *flocking*, di mana satu objek menjadi *leader* yang mengetahui arah dan tujuan sedangkan objek lainnya menjadi *follower* yang mengikuti *leader* tersebut. Penelitian ini menjelaskan dengan jelas bagaimana metode *flocking* beroperasi dan mengatasi situasi ketika anggota kerumunan terpisah. Namun,

penelitian ini hanya mempertimbangkan satu jenis objek *flock* yaitu kuda dan tidak membahas bagaimana metode *flocking* dapat digunakan untuk mengatasi jenis *flock* yang berbeda.

Pada jurnal penelitian “Simulasi Perilaku Jama’ah Saat Mengelilingi Ka’bah Menggunakan metode *Flocking* dan A*” yang diteliti oleh (Karim et al., 2017), Penelitian ini mengeksplorasi penggunaan metode *flocking* dalam simulasi tawaf pada jama'ah dengan tujuan untuk menghindari tabrakan antara jama'ah. Dalam jurnal ini dibahas mengenai pemanfaatan metode *flocking* untuk mensimulasikan kerumunan jamaah saat tawaf. Penelitian tersebut menggunakan konsep pemimpin (*leader*) dalam setiap agen *flocking* (jama'ah) untuk memimpin mereka menuju titik rute tertentu. Hasil implementasi metode *Flocking* dalam penelitian tersebut menunjukkan perilaku jama'ah yang fleksibel terhadap kondisi lingkungan, di mana jumlah agen (jama'ah) dapat mempengaruhi waktu yang dibutuhkan oleh pemimpin dan jamaahnya dalam menyelesaikan tawaf. Dalam jurnal ini diungkapkan bahwa metode *flocking* bisa dimanfaatkan untuk mensimulasikan kerumunan dalam skala besar.

2.2 Simulasi

Menurut (Siagian, 1987) pengertian simulasi adalah suatu metodologi untuk melaksanakan percobaan dengan menggunakan model dari satu sistem. Simulasi adalah representasi atau model dari suatu sistem nyata yang dibuat menggunakan komputer atau alat lainnya untuk mengamati, menganalisis, dan memahami sistem tersebut tanpa harus melibatkan sistem yang sebenarnya. Simulasi umumnya digunakan dalam berbagai bidang seperti ilmu pengetahuan, teknik, bisnis, dan

militer. Simulasi dapat digunakan untuk menguji suatu konsep, membuat prediksi, mengidentifikasi masalah, dan membuat keputusan yang lebih baik. Dalam simulasi, suatu sistem nyata direplikasi ke dalam model matematika atau model simulasi digital yang dapat dimanipulasi untuk memeriksa hasil yang berbeda. Simulasi dapat menggunakan berbagai teknik seperti simulasi Monte Carlo, simulasi diskrit, dan simulasi kontinu, tergantung pada jenis sistem yang direpresentasikan.

2.3 Game

Menurut Clark.C dalam (Ellington et al., 1982), *Game* adalah setiap kontes (permainan) di antara musuh (pemain) yang beroperasi di bawah batasan (aturan) untuk suatu tujuan (kemenangan, pembayaran kemenangan). Menurut Wahono dalam (Agustina & Chandra, 2017) game adalah suatu aktivitas baik itu terstruktur maupun semi-terstruktur yang bertujuan sebagai sarana hiburan maupun pendidikan. *Game* dapat dimainkan oleh satu orang atau kelompok orang dengan aturan yang sudah ditentukan. Dalam *game*, pemain biasanya harus menyelesaikan tugas atau tantangan tertentu dengan cara mengikuti aturan dan menjalankan strategi yang tepat.

Dengan pesatnya perkembangan teknologi, perkembangan game pun semakin canggih dan memiliki berbagai *genre*. Menurut (Santoso, et.al, 2017) terdapat 10 kategori *game* berdasarkan genrenya, yaitu :

1. *Action – Shooting* : video *game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga *timing*.

2. *Fighting* (pertarungan) : Jenis ini memang memerlukan kecepatan refleks dan koordinasi mata-tangan, tetapi inti dari *game* ini adalah penguasaan permainan, pengenalan karakter dan *timing*.
3. *Action-Adventure*: *Game* ini sudah berkembang jauh hingga menjadi genre campuran *action beat-em up*, dan sekarang jenis ini cenderung memiliki visual 3D dan sudut pandang orang ketiga.
4. *Adventure*: Pemain dalam bergerak, berlari, melompat hingga memecut atau menembak tidak diperlukan disini. Video *game* murni petualangan lebih menekankan pada jalan cerita dan kemampuan berfikir pemain.
5. Simulasi Konstruksi dan manajemen: Video *game* ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detail berbagai faktor.
6. *Role Playing*: Video *game* cenderung bermain peran, memiliki penekanan pada tokoh atau peran perwakilan pemain di dalam permainan.
7. Strategi: Video *game* jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. *Game* jenis ini terbagi atas:
 - a. *Real time Strategy*, *game* berjalan dalam waktu sebenarnya dan serentak antara semua pihak dan pemain harus memutuskan setiap langkah yang diambil saat itu juga.
 - b. *Turn based Strategy*, *game* yang berjalan secara bergiliran, saat kita mengambil keputusan dan menggerakkan pasukan, saat itu pihak lawan menunggu, begitu pula sebaliknya.

8. *Puzzle*: Video *game* ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, hingga mendorong-dorong kota masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini.
9. Simulasi kendaraan: Video *game* ini memberikan pengalaman atau interaktivitas sedekat mungkin dengan kendaraan yang aslinya, meskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut.
10. Olahraga: Singkat padat jelas, bermain *sport* di PC atau konsol. Permainan ini dibuat dengan tingkat realistis yang tinggi.

Game juga tidak hanya dimainkan di platform konsol atau PC, namun juga di perangkat mobile seperti smartphone atau tablet. Selain sebagai bentuk hiburan, *game* juga sering digunakan sebagai alat pendidikan atau pelatihan dalam berbagai bidang seperti militer, kesehatan, dan bisnis.

2.4 NPC

NPC adalah singkatan dari *Non-Player Character*, yang dalam konteks permainan video merujuk pada karakter yang dikendalikan oleh sistem permainan dan bukan oleh pemain. Menurut (Warpfelt, 2016) penggunaan istilah NPC telah ada sebelum adanya permainan digital.. NPC hadir dalam banyak jenis permainan video, seperti RPG, petualangan, dan game aksi, dan berperan dalam membantu pemain mencapai tujuan permainan. NPC dapat memiliki peran yang berbeda-beda, seperti karakter yang membantu pemain dalam misi, memberikan informasi,

menjadi musuh atau hambatan, atau hanya sebagai latar belakang. Karakter NPC sering dikendalikan oleh sistem kecerdasan buatan (AI) dan dapat menampilkan tingkat perilaku dan interaksi yang beragam dengan pemain.

Dalam beberapa permainan video *modern*, NPC semakin berkembang dan memiliki kemampuan interaksi yang semakin canggih dengan pemain. Beberapa permainan bahkan memungkinkan pemain untuk berinteraksi dan membentuk hubungan dengan karakter NPC, seperti dalam permainan simulasi kehidupan dan permainan peran. NPC dengan kemampuan canggih dapat bereaksi terhadap tindakan pemain dan mempengaruhi alur cerita permainan. Penggunaan NPC yang canggih dan realistis juga dapat meningkatkan pengalaman bermain pemain, memberikan pengalaman yang lebih mendalam dan imersif dalam permainan.

Dalam penelitian *game* yang sedang dilakukan, NPC yang dimaksud adalah agen dalam kelompok kerumunan (*flock*). Setiap NPC memiliki tujuan tertentu yang harus dicapai dan dapat bergerak secara fleksibel dalam mencapai tujuan tersebut bersama dengan anggota kelompoknya. Untuk mencegah tabrakan dengan anggota kelompok lainnya, NPC dapat menyesuaikan diri dengan menggunakan metode *flocking*. Dengan demikian, perilaku NPC pada *game* ini bergantung pada implementasi metode Flocking dan tujuan individu dari masing-masing kelompok NPC yang disesuaikan dengan perilaku *flocking*.

2.5 Usability Testing

Menurut (Matera dkk, 2006), *Usability* berasal dari kata *usable* yang memiliki arti dapat digunakan dengan baik. Suatu perangkat dikategorikan sebagai efisien ketika tingkat kegagalan dalam penggunaannya bisa dihilangkan atau dikurangi sedikit, dan sekaligus memberikan manfaat dan hasil positif kepada penggunanya.

Menurut *International Organization for Standardization* (ISO) 9241-11 (ISO, 1994), terdapat tiga aspek yang menjadi acuan untuk mengukur seberapa bergunanya sebuah aplikasi dalam mencapai tujuan dibuatnya aplikasi tersebut. Ketiga aspek tersebut di antara lain *Effectiveness* (Kemampuan pengguna untuk berhasil mencapai tujuan mereka dalam konteks spesifik), *Efficiency* (Sumber daya yang diterapkan berkaitan dengan tingkat akurasi dan keberhasilan yang dicapai oleh pengguna dalam menyelesaikan tugas), dan *Satisfaction* (Kepuasan pengguna ketika memanfaatkan aplikasi). Usability testing adalah pengujian penggunaan terhadap sistem atau produk untuk menemukan permasalahan daya guna atau *usability* (Beny et.al, 2019).

Dalam penelitian ini, peneliti menggunakan usability testing untuk mengukur tingkat usability dari metode flocking yang digunakan pada kerumunan NPC. Hasil dari pengukuran tingkat usability metode flocking pada kerumunan NPC untuk mengatur sikap kerumunan pada NPC pada penelitian ini diharapkan dapat menjadi acuan bagaimana *Effectiveness*, *Efficiency*, dan *Satisfaction* dari penerapan metode flocking pada kerumunan NPC di mata *player*.

2.6 System Usability Scale (SUS)

System Usability Scale (SUS) dikembangkan oleh John Brooke 1986 yang dapat memungkinkan seorang praktisi usability untuk secara cepat dan mudah menilai usability dari sebuah produk (Bangor e.al, 2008). Selain dari kelebihan tersebut, metode *System Usability Scale* (SUS) memiliki banyak atribut lain yang dapat menjadikan metode ini sebagai pilihan yang tepat, salah satu dari atribut yang menjadi kelebihan paling menonjol dari metode ini adalah survey pada metode ini tidak hanya memihak pada teknologi. Dengan adanya kelebihan ini membuat metode ini bersifat lebih *flexible* untuk menilai teknologi dalam lingkup luas (Bangor e.al, 2008). Selain itu metode survey ini dapat dikalkulasikan dengan sederhana dengan nilai keluaran (*output*) berupa skor 0-100 sehingga dapat lebih mudah untuk dimengerti. Metode *System Usability Testing* (SUS) juga dikenal dengan tidak ada penggunaan biaya dalam penggunaannya serta memiliki ukuran sampel yang kecil (Brooke, 1996). Selain itu, *System Usability Testing* bersifat agnostik terhadap teknologi, sehingga penggunaan metode ini dianggap sangat fleksibel dan dapat diterapkan pada berbagai konteks (Bangor et al., 2008), termasuk dalam mengevaluasi tingkat kegunaan sebuah fitur dalam game. Dalam penelitiannya, Olsen dan rekan-rekannya (Olsen et al., 2011) mengukur aspek *usability*, *playability*, dan *effectiveness* dari sebuah serious game dengan memanfaatkan beberapa pertanyaan dari System Usability Scale (SUS).

Strongly disagree	Strongly agree
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 12345 </div>	

Gambar 2.1 *System usability scale*

Gambar 2.1 adalah ilustrasi yang memperlihatkan bagaimana skala poin pada metode *System Usability Scale* (SUS) bekerja. Metode *System Usability Scale* (SUS) merupakan sebuah kuesioner yang memiliki 10 pertanyaan dengan setiap pertanyaan memiliki 5 skala poin dimulai dari “sangat setuju” ke “sangat tidak setuju” dengan setengah dari pertanyaan tersebut berupa pertanyaan untuk respon “sangat setuju” dan yang lainnya untuk respon “sangat tidak setuju” (Brook, 1996). Hasil dari respon kuesioner tersebut kemudian akan dihitung menggunakan rumus berikut untuk mendapat nilai skor usability dari sistem tersebut

$$Sus\ Score = Total\ Score \times 2.5 \tag{2.1}$$

2.7 Flock (Kerumunan)

Flock atau kerumunan merupakan fenomena dimana sejumlah besar agen, dengan hanya menggunakan informasi lingkungan terbatas dan aturan sederhana, mengatur ke dalam gerakan terkoordinasi (Su et al., 2009). , *Flocking* merupakan suatu teknik atau metode dalam pengembangan *game* yang digunakan untuk membangun perilaku kelompok agen atau objek di dalam suatu simulasi *game*. Tujuan utama dari teknik ini adalah untuk menciptakan kelompok objek yang terlihat seperti organik dan mengalami perubahan bentuk, yang dapat meniru perilaku dalam kehidupan nyata seperti burung atau ikan yang bergerak secara berkelompok. Teknik *flocking* sering digunakan pada *game* simulasi seperti simulasi penerbangan atau simulasi dunia nyata.

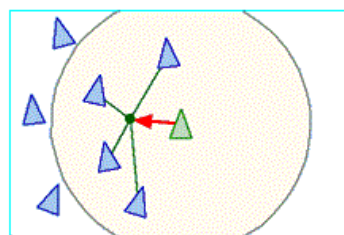
Metode *flocking* adalah sebuah inovasi yang berasal dari konsep *steering behavior* yang telah ditemukan sebelumnya oleh Craig Reynolds. Metode *flocking* terinspirasi dari perilaku burung dan ikan yang bergerak secara berkelompok dengan koordinasi yang cermat. Dalam pengembangan *game*, metode ini digunakan untuk menciptakan gerakan organik pada objek-objek dalam *game* seperti NPC, kendaraan, atau burung-burung yang terbang dalam kelompok. Konsep dasar dari metode *flocking* adalah setiap agen atau objek di dalam kelompok memiliki perilaku yang sama seperti menghindari tabrakan dengan objek lain dan mencoba mengikuti gerakan dari agen di sekitarnya.

Menurut (Reynolds, 1987) Algoritma *flocking* memiliki tiga aturan utama, yaitu :

1. *Flock Centering* : Mencoba untuk mendekat dengan anggota kerumunan terdekat.
2. *Cohesion Avoidance* : Mencegah terjadinya tabrakan dengan anggota kawanan yang berdekatan
3. *Velocity Matching* : Berusaha menyesuaikan kecepatan dengan anggota kawanan terdekat.

Dalam (Bourg & Seeman, 2004), dijelaskan bahwa peraturan-peraturan Reynolds mencakup *Cohesion*, *Separation*, dan *Alignment*. Peraturan tersebut memerintahkan setiap agen untuk memasuki kelompoknya, mengikuti arah pergerakan kelompok, dan menghindari tabrakan dengan anggota kelompoknya.

1. *Cohesion*



Cohesion

Gambar 2.2 Perilaku *Cohesion*

Gambar 2.2 merupakan ilustrasi dari perilaku *cohesion* dalam algoritma flocking. Perilaku *cohesion* memungkinkan agen-agenya untuk mengarahkan diri mereka ke titik tengah kelompoknya, yang membantu menjaga kelompok agar tetap bersatu. Perilaku ini memungkinkan agen untuk bergerak menuju titik tengah

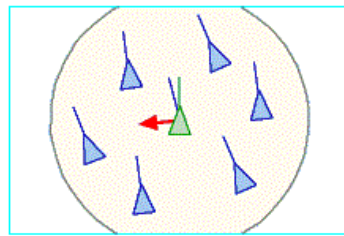
kelompoknya. Cohesion juga mengatur agar agen dapat mendekat satu sama lain ketika terlalu jauh.

Cohesion diformulasikan pada persamaan (Cui et al., 2006)

$$d(P_x, P_b) \leq d_1 \cap d(P_x, P_b) \geq d_2 \Rightarrow V_{cr} = \sum_n^x (P_x - P_b) \quad (2.2)$$

Dalam persamaan ini, \vec{V}_{cr} menunjukkan kecepatan yang diatur oleh aturan cohesion, yang dapat dilihat pada gambar 2.2. $d(P_x, P_b)$ merepresentasikan jarak antara agen x dan tetangga b, sedangkan d_1 dan d_2 adalah nilai jarak yang telah ditentukan, $(P_x \rightarrow P_b)$ adalah perhitungan arah *vector point*.

2. Alignment



Alignment

Gambar 2.3 Perilaku *Alignment*

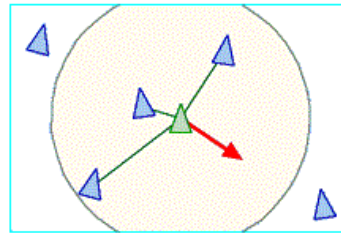
Gambar 2.3 merupakan ilustrasi dari perilaku *alignment* dalam algoritma *flocking*. Perilaku ini memungkinkan agen untuk menyesuaikan kecepatannya dengan agen lainnya, sehingga agen dapat mencocokkan kecepatannya dengan tetangganya.

Alignment diformulasikan pada persamaan (Cui et al., 2006)

$$d(P_x, P_b) \leq d_1 \cap d(P_x, P_b) \geq d_2 \Rightarrow V_{ar} = \frac{1}{n} \sum_n^x V_x \quad (2.3)$$

Dalam hal ini, \vec{V}_{ar} mengacu pada kecepatan yang diatur oleh perilaku alignment, sebagaimana terlihat pada gambar 2.3, sedangkan $d(P_x, P_b)$ merupakan jarak antara agen x dan tetangga b, n menunjukkan jumlah total tetangga, \vec{V}_x menunjukkan kecepatan agen x, d_1 dan d_2 adalah jarak yang sudah ditentukan.

3. *Separation*



Separation

Gambar 2.4 Perilaku *Separation*

Gambar 2.4 merupakan ilustrasi dari perilaku *separation* dalam algoritma flocking. Perilaku separation memungkinkan agen untuk menjaga jarak dengan agen lainnya, sehingga mencegah terjadinya kerumunan. Dengan cara ini, agen diarahkan untuk menghindari tabrakan dan mempertahankan jarak yang aman antara satu sama lain.

Separation diformulasikan dalam persamaan (Cui et al., 2006)

$$d(P_x, P_b) \leq d_2 \Rightarrow V_{sr} = \sum_n^x \frac{(v_x + v_b)}{d(P_x, P_b)} \quad (2.4)$$

Perilaku *separation* memberikan kecepatan yang ditentukan pada gambar 2.4 berdasarkan jarak antara agen x dan tetangga b, kecepatan dan dari agen x dan agen b, dan nilai jarak yang sudah ditetapkan.

Berdasarkan persamaan yang sudah dijelaskan di atas, perilaku *cohesion*, *alignment*, dan *separation* akan dijalankan apabila agen tetangga berada pada rentang jarak yang sudah ditetapkan. Hal ini dirumuskan dalam persamaan berikut

$$d(P_x, P_b) \leq d_1 \cap d(P_x, P_b) \geq d_2 \quad (2.5)$$

Persamaan tersebut merupakan persamaan kondisi yang menghasilkan kondisi benar ketika jarak agen x dengan tetangga b kurang dari jarak maksimal (d_1) dan jarak agen x dengan tetangga b lebih dari jarak minimal (d_2).

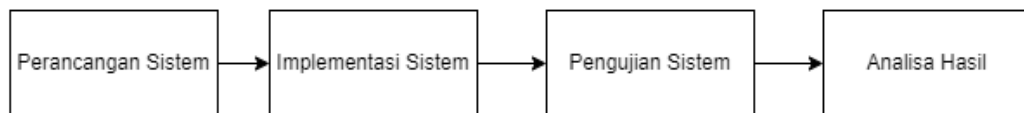
Penggunaan metode *flocking* dalam pengembangan *game* memberikan manfaat seperti meningkatkan keberagaman dalam perilaku agen atau objek di dalam *game*. Dalam penggunaannya, metode *flocking* dapat diatur dengan parameter yang berbeda sehingga dapat menghasilkan perilaku yang berbeda pada agen atau objek di dalam kelompok. Hal ini memberikan variasi yang lebih dalam perilaku agen atau objek di dalam *game* sehingga pengalaman bermain *game* menjadi lebih menarik dan dinamis. Selain itu, metode *flocking* juga membantu meningkatkan pengalaman bermain *game* karena perilaku kelompok objek yang terlihat lebih natural dan realistis.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Proses penelitian ini dimulai dengan tahap perancangan sistem, kemudian dilanjutkan dengan implementasi sistem, pengujian sistem, pengumpulan data, analisis, dan diakhiri dengan tahap pembahasan.



Gambar 3.1 Alur tahapan penelitian

Proses penelitian dimulai dengan tahapan perancangan sistem, di mana peneliti merancang desain sistem yang sedang diteliti. Kemudian, hasil desain sistem diimplementasikan pada objek penelitian berupa kerumunan NPC untuk menghasilkan perilaku berkerumun yang sesuai dengan jumlah kerumunan NPC.

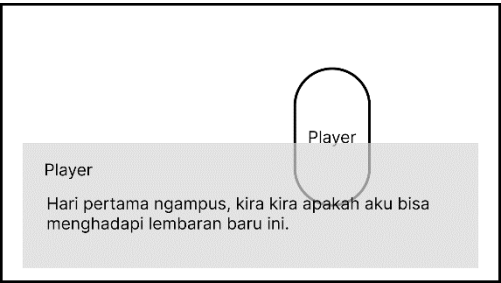
Setelah tahapan implementasi sistem, langkah berikutnya dalam penelitian adalah tahap pengujian sistem. Dalam tahap ini, peneliti akan mengevaluasi tingkat usability metode flocking pada kerumunan NPC. Untuk mendapatkan hasil evaluasi tersebut, peneliti menggunakan metode *System Usability Scale (SUS)*. Selain pengujian sistem dengan metode *System Usability Scale (SUS)*, peneliti juga melakukan observasi terhadap waktu yang dibutuhkan oleh sebuah kerumunan (*flock*) NPC untuk berpindah dari satu lokasi ke lokasi lain. Pengamatan dilakukan dalam tiga skenario: tanpa penggunaan metode *flocking*, dengan penggunaan metode *flocking*, dan dengan penggunaan metode *flocking* yang terintegrasi dengan

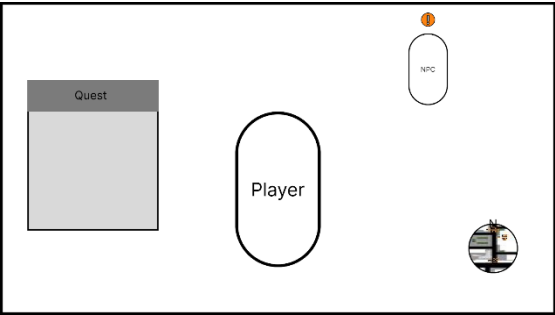
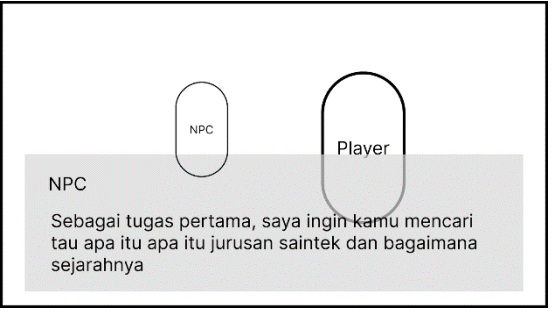
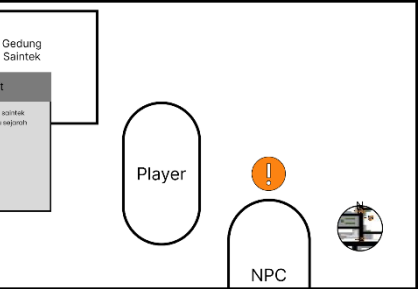
Finite State Machine (FSM). Selain itu, pengamatan juga dilakukan terhadap waktu yang diperlukan oleh kerumunan NPC untuk berinteraksi dengan kerumunan NPC lain dalam berbagai skenario yang sama. Setelah melewati tahap pengujian sistem, langkah selanjutnya dalam penelitian ini adalah Analisis. Pada tahap ini, peneliti akan menganalisa hasil pengujian yang didapatkan yang kemudian dari hasil analisis tersebut yang kemudian hasil dari analisis tersebut peneliti gunakan untuk menarik kesimpulan hasil dari penelitian ini.

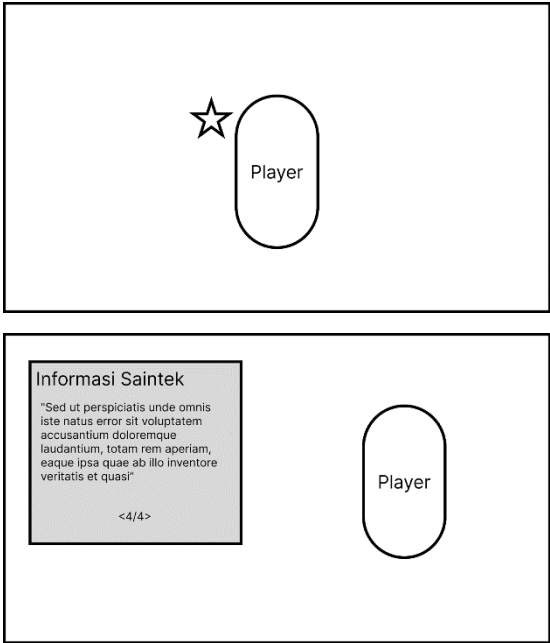
3.2 Storyboard

Pengenalan UIN Malang merupakan sebuah game yang bertujuan untuk memperkenalkan lingkungan kampus UIN Malang kepada para pemainnya. Pemain akan mengambil peran sebagai mahasiswa baru yang diberikan tugas untuk menjelajahi seluruh area kampus, dengan tujuan memahami dan mengenal fasilitas-fasilitas yang tersedia dalam lingkungan kampus, sebagai bagian dari proses orientasi mereka sebagai mahasiswa baru selama ospek.

Tabel 3. 1 Storyboard Game

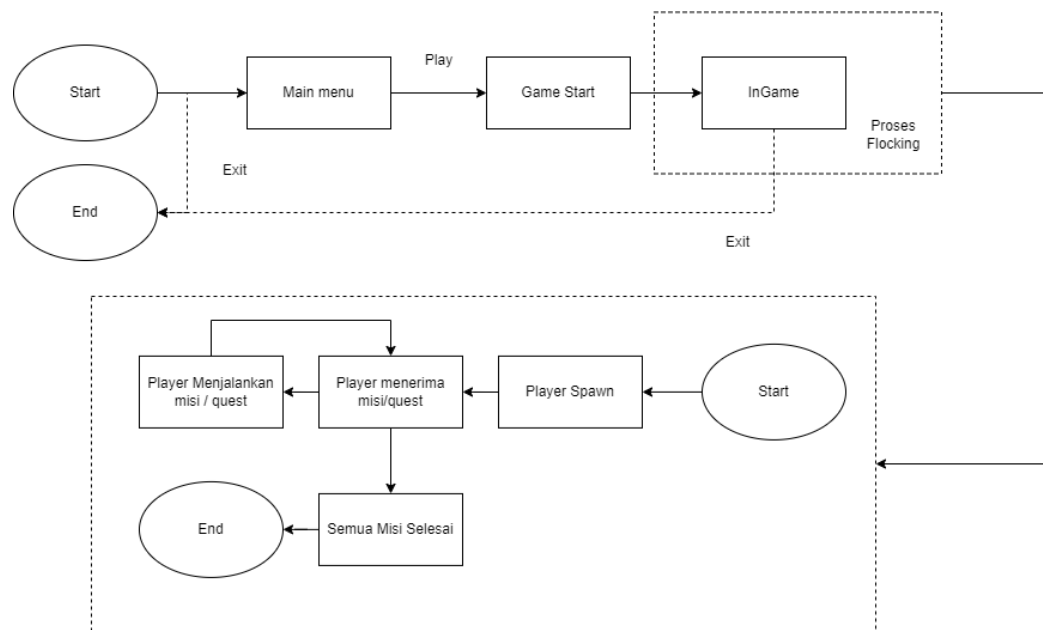
Gambar	Penjelasan
	<p>Pemain akan mengambil peran sebagai mahasiswa baru yang baru tiba di kampus UIN Malang. Mereka akan berinteraksi dengan karakter NPC (<i>Non-Player Character</i>) yang memerankan staf kampus atau mahasiswa senior. NPC akan memberikan <i>quest</i> atau tugas kepada pemain sebagai bagian dari proses orientasi mahasiswa baru, dengan tujuan agar pemain lebih mengenal Universitas Islam Negeri Maulana Malik Ibrahim Malang.</p>

Gambar	Penjelasan
	<p>Pemain akan mengambil peran sebagai mahasiswa baru yang baru tiba di kampus UIN Malang. Mereka akan berinteraksi dengan karakter NPC (<i>Non-Player Character</i>) yang memerankan staf kampus atau mahasiswa senior. NPC akan memberikan <i>quest</i> atau tugas kepada pemain sebagai bagian dari proses orientasi mahasiswa baru, dengan tujuan agar pemain lebih mengenal Universitas Islam Negeri Maulana Malik Ibrahim Malang.</p>
 <p>NPC</p> <p>Sebagai tugas pertama, saya ingin kamu mencari tau apa itu apa itu jurusan saintek dan bagaimana sejarahnya</p>	<p>Ketika berinteraksi dengan pemain, dialog antara karakter pemain dan NPC akan muncul. Melalui dialog tersebut, pemain akan mendapatkan informasi mengenai latar belakang karakter pemain, peran karakter pemain dalam permainan (sebagai mahasiswa baru yang ingin menyelesaikan misi), rincian misi yang diberikan, dan informasi-informasi relevan lainnya.</p>
 <p>Gedung Saintek</p> <p>Quest</p> <p>Sebagai tugas pertama, saya ingin kamu mencari tau apa itu apa itu jurusan saintek dan bagaimana sejarahnya</p>	<p>Setelah pemain memicu <i>quest</i>, mereka akan diarahkan ke lokasi tujuan <i>quest</i> melalui tampilan peta mini dalam permainan. Di lokasi tersebut, pemain memiliki kesempatan untuk mengeksekusi dan menyelesaikan <i>quest</i> yang telah diaktifkan sebelumnya. Pemain dapat berinteraksi dengan elemen-elemen dalam permainan yang berkaitan dengan <i>quest</i> tersebut. Dengan demikian, pemain dapat melanjutkan permainan dengan menyelesaikan tantangan-tantangan yang ada dalam <i>quest</i> yang sedang berlangsung. Proses ini memberikan pengalaman bermain yang lebih dinamis dan interaktif.</p>

Gambar	Penjelasan
 <p>The figure consists of two screenshots from a game interface. The top screenshot shows a star icon to the left of a rounded rectangle labeled 'Player'. The bottom screenshot shows a text box on the left with the title 'Informasi Saintek' and a paragraph of Latin text: "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi". Below the text is a small '<4/4>' indicator. To the right of the text box is a rounded rectangle labeled 'Player'.</p>	<p>Mekanisme dari <i>quest</i> dalam permainan mengharuskan pemain untuk mengumpulkan potongan informasi yang muncul dalam permainan. Setelah pemain berhasil mengumpulkan semua potongan informasi yang diperlukan, mereka akan memperoleh pengetahuan yang diperlukan untuk menjawab pertanyaan NPC. <i>Quest</i> akan selesai ketika pemain berhasil menjawab semua pertanyaan NPC.</p>

3.3 Desain Sistem

Desain sistem adalah sebuah rencana atau gambaran yang menggambarkan bagaimana elemen-elemen yang terkait akan disusun menjadi satu kesatuan yang terstruktur. Desain sistem ini nantinya akan menjadi acuan bagi peneliti dalam menyelesaikan penelitiannya. Desain sistem dapat berupa sketsa atau ilustrasi yang menggambarkan kumpulan elemen yang terkait dan cara penyusunannya. Hal ini dilakukan untuk memastikan bahwa semua elemen terkoneksi dengan baik dan sistem yang dirancang dapat bekerja dengan baik. Berikut merupakan untuk desain sistem game.

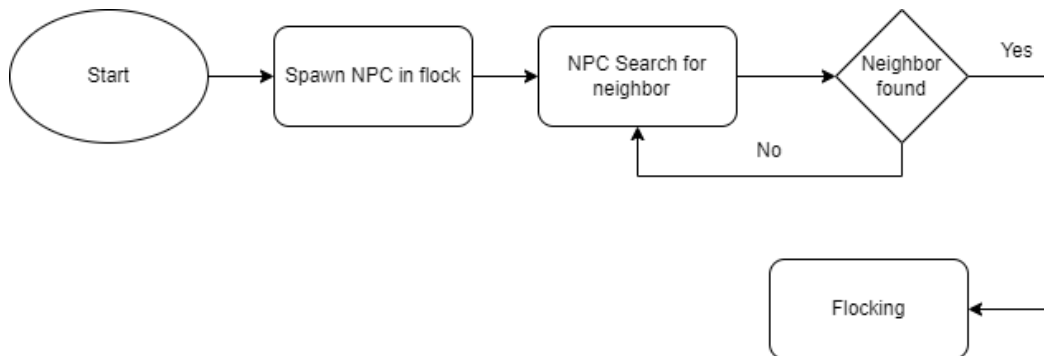


Gambar 3.2 Design System

Berdasarkan gambar desain sistem pada Gambar 3.2, perancangan sistem *game* ini dimulai dari menu utama setelah pemain menjalankan *game*. Kemudian pemain dapat memulai permainan, yang selanjutnya masuk ke dalam proses *In-Game*.

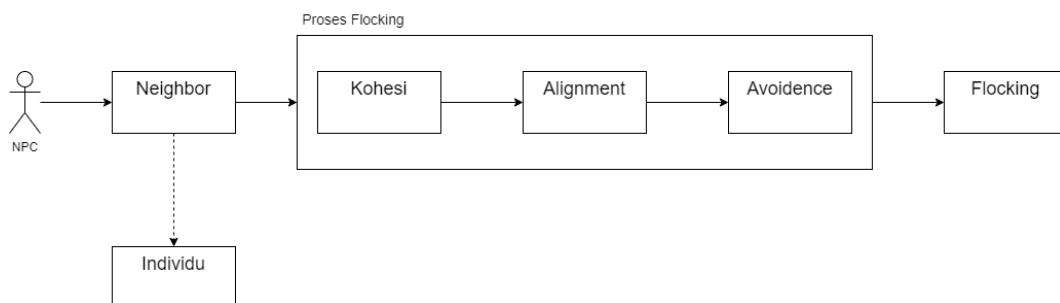
Berdasarkan gambar 3.2 pada proses *In-Game*, karakter pemain akan muncul (*spawn*) sesuai dengan lokasi terakhir pemain bermain (jika ini adalah permainan pertama pemain, karakter pemain akan muncul di gerbang depan). Setelah pemain muncul, mereka akan menerima misi yang harus diselesaikan sebagai bagian dari permainan. Permainan dianggap selesai ketika semua misi dalam permainan telah terselesaikan oleh pemain. Selama proses ini (sebelum semua misi selesai), pemain juga memiliki opsi untuk keluar dari permainan. Pada proses *In-Game*, proses *flocking* terjadi pada kerumunan karakter *non-playable* (NPC).

Cara kerja setiap karakter non-playable (NPC) dalam konteks penggunaan metode *flocking* pada permainan penelitian ini melibatkan beberapa tahap, dimulai dengan NPC muncul (*spawn*) dalam kelompok (*flock*) tertentu. Kemudian, NPC melakukan pencarian kondisi tetangga.



Gambar 3.3 Alur NPC

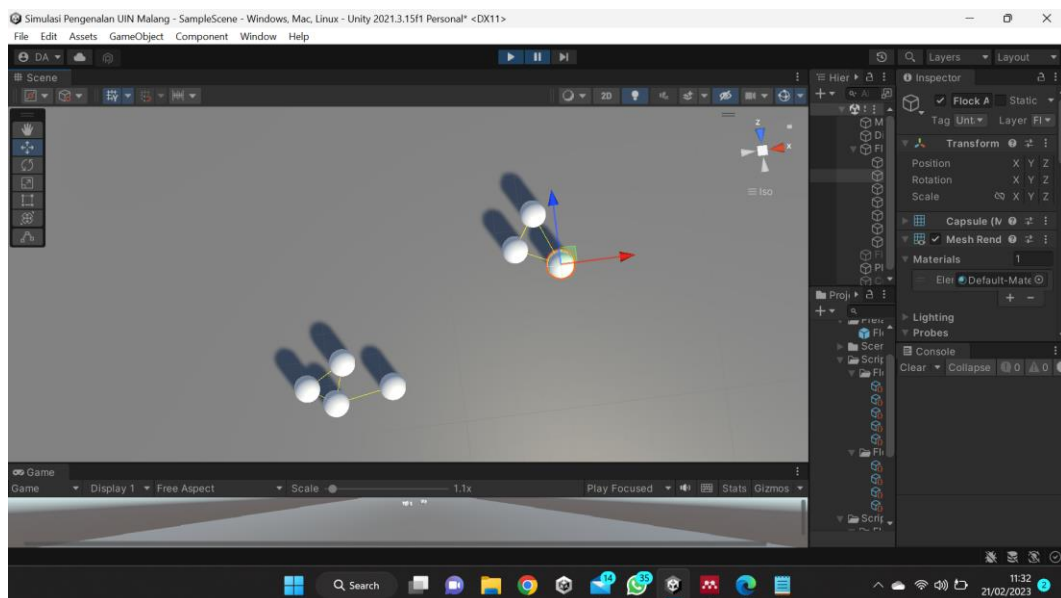
Gambar 3.3 menggambarkan alur sistem yang ditempuh oleh karakter *non-playable* (NPC). Pada tahap awal, setelah NPC muncul dalam kelompoknya, mereka akan melakukan pencarian tetangga terdekat dalam jarak tertentu. Ketika NPC berhasil menemukan tetangga-tetangga tersebut, mereka akan memulai perilaku *flocking*.



Gambar 3.4 Diagram blok cara kerja *flocking*

Gambar 3.4 adalah diagram blok yang mengilustrasikan secara sistematis bagaimana perilaku *flocking* bekerja. Dalam implementasinya, algoritma *flocking* diterapkan pada satu kelompok (*flock*) kerumunan. Saat menjalankan *flocking*, NPC

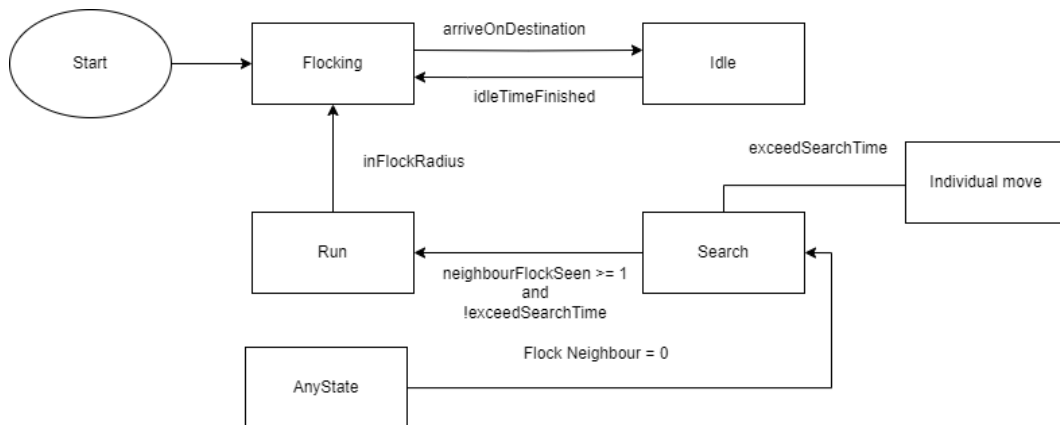
berusaha untuk menyelaraskan kecepatan dan arah gerakan mereka dengan kecepatan dan arah tetangga-tetangga yang ada. Perilaku ini mencakup tiga aspek utama dalam *flocking*, yaitu kohesi (*cohesion*), penyelarasan (*alignment*), dan penghindaran (*avoidance*). NPC juga dapat bergerak secara mandiri apabila mereka tidak memiliki tetangga.



Gambar 3.5 Penerapan *Flocking*

Gambar 3.5 menunjukkan bagaimana hasil implementasi algoritma *flocking* dalam *game*. Dalam proses *flocking*, setiap NPC akan mencari tetangga terdekat. Setelah menemukan tetangga, NPC akan menjalankan proses *flocking* dan bergerak sesuai dengan kelompoknya (*flock*). Jika NPC tidak memiliki tetangga, maka mereka akan mencari dan mendekati kelompok (*flock*) mereka.

Dalam rangka optimasi pengelolaan proses tersebut, para peneliti memutuskan untuk menggunakan FSM (*Finite-State Machine*) dalam mengatur perilaku NPC.



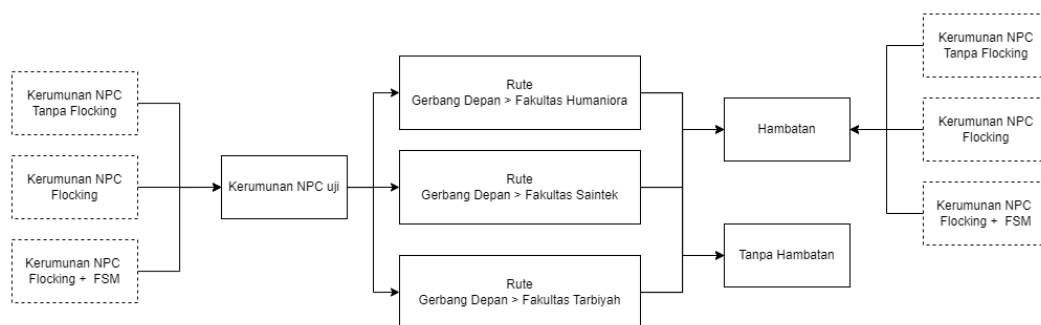
Gambar 3.6 FSM NPC

Gambar 3.6 menampilkan rancangan implementasi *Finite-State Machine* pada NPC. Saat permainan dimulai, NPC dipanggil bersama dengan anggota kelompoknya. Setelah pemanggilan, NPC akan melibatkan diri dalam proses *flocking* jika NPC tersebut berada dalam jarak minimal dengan agen *flock* lain untuk melakukan *flocking*. Jika NPC belum mencapai tujuannya, mereka akan bergerak menuju tujuan bersama kelompok mereka. Setibanya di tujuan, NPC akan memasuki keadaan *idle*. Ketika waktu *idle* habis, NPC akan kembali ke dalam proses *flocking* dan kemudian bergerak ke poin tujuan berikutnya. Dalam segala keadaan (*Any State*), jika NPC tidak memiliki tetangga, maka status NPC akan berubah menjadi pencarian (*search*) untuk mulai mencari kelompoknya. Jika NPC berhasil menemukan kelompoknya dalam waktu pencarian yang telah ditentukan, maka mereka akan bergabung kembali dalam proses *flocking*, jika tidak, maka mereka akan bergerak secara individu. Dengan adanya state pencarian, pengoptimalan perilaku *flocking* dapat tercapai, karena state pencarian memungkinkan agen *flock* untuk lebih cepat bergabung kembali dengan *flock* mereka. Hal ini akan memperkuat kinerja *flocking* dalam mengatur perilaku agen.

3.4 Rancangan Pengujian Sistem

Pada penelitian ini, dilakukan observasi perbandingan waktu tempuh pada kerumunan NPC dengan penggunaan metode *flocking* dan tanpa penggunaan metode *flocking*. Selain itu, juga dilakukan pengujian perbandingan kemampuan kerumunan NPC dalam menghadapi hambatan statis dan pengujian tingkat *usability* metode *flocking* yang diimplementasikan pada kerumunan NPC dengan menggunakan metode *System Usability Scale (SUS)* untuk mengukur tingkat *usability* dari metode *flocking* pada kerumunan NPC.

3.4.1 Perbandingan Waktu Tempuh Kerumunan NPC Pada Penggunaan Metode *Flocking*



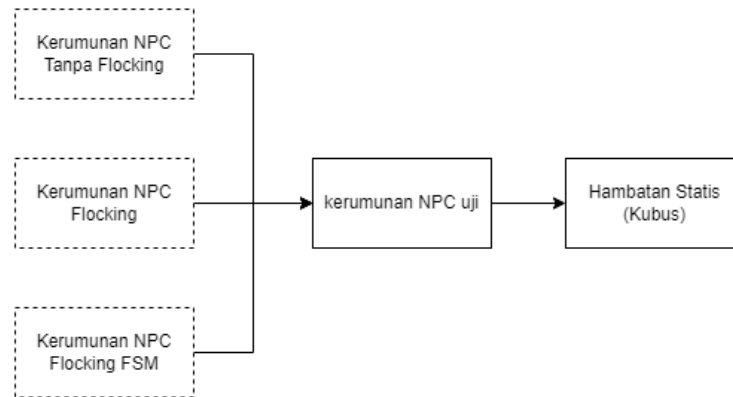
Gambar 3.7 Skema pengujian waktu tempuh kerumunan NPC

Gambar 3.7 memberikan penjelasan tentang skema pengujian waktu tempuh pada kerumunan NPC. Sejumlah agen kerumunan NPC akan diuji 2, 5, 10, 15, 20, dan 25, akan diujikan dalam setiap iterasi pengujian. Pengamatan dilakukan terhadap waktu tempuh kerumunan NPC uji ketika menggunakan metode *flocking*, metode *flocking* dengan *Finite State Machine*, dan tanpa penerapan metode tertentu, melibatkan tiga rute berbeda. Pengaruh waktu tempuh juga diamati dalam konteks menghadapi hambatan atau tanpa hambatan.

Dalam skema penelitian ini, kerumunan NPC akan dihadapkan pada hambatan berupa kerumunan NPC dengan jumlah yang sama, namun dengan implementasi yang berbeda (menggunakan *flocking*, *flocking* dengan *Finite State Machine*, dan tanpa implementasi metode). Berbagai variabel seperti kecepatan NPC, radius tetangga, dan jarak penghindaran akan dijaga konstan sepanjang proses observasi.

Pada dasarnya, pengujian ini bertujuan untuk memahami dan membandingkan performa waktu tempuh kerumunan NPC dalam kondisi yang berbeda, menggambarkan bagaimana metode dan hambatan mempengaruhi mobilitas dan koordinasi dalam suatu sistem berkerumun. Dengan mengatur variabel-variabel tertentu sebagai konstan, penelitian ini dapat memberikan wawasan yang lebih mendalam terkait efek dari metode dan hambatan terhadap perilaku kerumunan NPC dalam lingkungan simulasi *game*.

3.4.2 Perbandingan Kemampuan Kerumunan NPC Menghadapi Hambatan Statis Menggunakan Metode *Flocking*

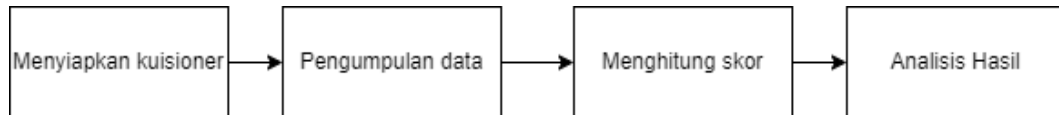


Gambar 3.8 Skema pengujian kemampuan kerumunan NPC menghadapi hambatan statis

Gambar 3.8 mengilustrasikan skema pengujian yang digunakan untuk membandingkan kemampuan kerumunan NPC ketika dihadapkan dengan hambatan statis. Berdasarkan gambar tersebut, kerumunan NPC yang akan diuji terdiri dari tiga kelompok uji, yakni yang tidak menggunakan metode, menggunakan metode *flocking*, dan menggunakan metode *flocking* dengan *Finite State Machine* (FSM) dengan jumlah agen kerumunan NPC akan diuji berjumlah 2, 5, 10, 15, 20, dan 25 dalam setiap iterasi pengujian.. Pengujian dilakukan dengan menempatkan hambatan statis berupa kubus di rute lurus yang dilewati kerumunan NPC. Tujuan dari pengujian ini adalah untuk membandingkan kemampuan kerumunan NPC dalam memecah formasi guna menghindari hambatan statis pada setiap metode uji yang diterapkan pada kerumunan NPC.

3.4.3 Pengujian Tingkat Usability Metode *Flocking* Pada Kerumunan NPC

Pengujian selanjutnya adalah pengujian tingkat *usability* metode *flocking* pada kerumunan NPC dengan menggunakan metode *System Usability Scale* (SUS). Dalam pengujian ini, ada serangkaian langkah-langkah yang harus diikuti.



Gambar 3.9 Alur Pengujian *System Usability Scale*

Gambar 3.9 menggambarkan alur dari pengujian *System Usability Scale* (SUS). Tahapan pertama dalam pengujian tingkat *usability* metode *flocking* pada kerumunan NPC adalah persiapan kuesioner dengan instruksi dan pertanyaan yang jelas.

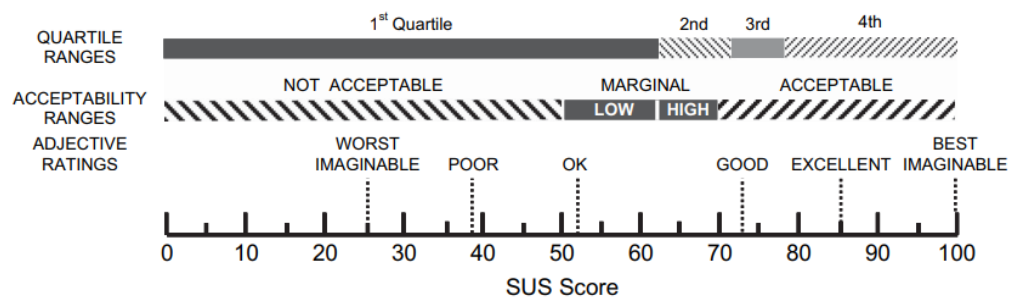
Tabel 3.2 Pertanyaan kuesioner *System Usability Scale* (SUS)

No	Pertanyaan
1	Saya merasa mudah untuk memahami cara kerja algoritma <i>flocking</i>
2	Saya merasa kerumunan NPC tidak secara efektif menggambarkan suasana keramaian seperti dalam kehidupan nyata
3	Saya merasa perilaku kerumunan NPC dalam game membuat saya merasa lebih terlibat dalam pengalaman bermain
4	Saya merasa metode <i>flocking</i> tidak dapat meningkatkan interaksi saya dengan karakter NPC dalam game
5	Saya merasa perilaku kerumunan NPC dalam game memiliki kegunaan yang signifikan
6	Saya merasa pergerakan kerumunan NPC dalam game ini tidak realistis
7	Saya ingin melihat lebih banyak penggunaan metode <i>flocking</i> dalam mengatur kerumunan NPC pada game lain
8	Saya merasa bahwa perilaku kerumunan NPC dalam game tidak sesuai dengan ekspektasi saya
9	Saya merasa pergerakan NPC dalam kerumunan menambah keseruan permainan
10	Saya tidak akan merekomendasikan game ini kepada teman-teman saya berdasarkan pengalaman bermain saya dengan kerumunan NPC dalam game ini

Tabel 3.2 merupakan tabel yang berisi kumpulan pertanyaan yang akan digunakan untuk mengukur tingkat usability metode flocking pada kerumunan NPC dengan menerapkan metode System Usability Scale (SUS). Tabel ini berperan penting dalam proses pengujian, karena pertanyaan-pertanyaan di dalamnya dirancang untuk mendapatkan pemahaman yang mendalam tentang sejauh mana metode flocking dapat memberikan pengalaman bermain yang baik kepada pemain. Dengan demikian, Tabel 3.2 menjadi alat utama dalam mengevaluasi aspek usability dari metode flocking dalam konteks permainan ini.

Kemudian, langkah kedua melibatkan pengumpulan data jawaban dari kuesioner yang kemudian diolah pada langkah ketiga, yakni perhitungan skor. Untuk menghitung skor akhir dari System Usability Scale (SUS), nilai skor dari tiap jawaban pada kuesioner diperlukan. Setiap pertanyaan berkontribusi dengan skor antara 0 hingga 4. Pertanyaan 1, 3, 5, 7, dan 9 memiliki perhitungan skor yang melibatkan pengurangan skor yang diberikan oleh responden dengan angka tetap 1. Di sisi lain, pertanyaan 2, 4, 6, 8, dan 10 mengharuskan pengurangan skor dari angka tetap 5 dengan skor yang didapat dari jawaban pertanyaan tersebut.

Setelah perhitungan skor untuk setiap pertanyaan diselesaikan, skor akhir akan dihitung menggunakan rumus *System Usability Scale* (SUS). Nilai skor dari *System Usability Scale* didapatkan dengan menjumlahkan skor dari semua pertanyaan dan menggunakannya dengan faktor 2,5. Faktor ini digunakan untuk mengubah rentang skor awal (0-40) menjadi rentang skor akhir (0-100) yang lebih representatif untuk menilai tingkat usability dari metode yang dievaluasi. Menurut (Bangor et.al, 2008), rentang nilai skor pengujian pada metode *System Usability Scale* (SUS) diilustrasikan sebagai berikut



Gambar 3.10 Penerimaan nilai skor SUS keseluruhan

Berdasarkan Gambar 3.10, skor hasil uji menggunakan *System Usability Scale* (SUS) dibagi menjadi empat kuartil. Nilai skor yang dapat dianggap baik sebagai hasil pengujian adalah skor dalam rentang 70 hingga 100. Produk dengan skor di atas 70 pada *System Usability Scale* (SUS) termasuk dalam kuartil ketiga, menandakan bahwa produk tersebut memiliki tingkat *usability* yang baik. Produk dengan skor di atas 80 dikategorikan sebagai produk yang lebih unggul dalam hal *usability* (kuartil keempat).

Sementara itu, produk dengan skor dalam rentang 60 hingga 70 (kuartil kedua) dianggap memiliki tingkat usability yang cukup baik, tetapi masih memerlukan perhatian untuk perbaikan yang berkelanjutan. Produk dengan skor di bawah 60 (kuartil pertama) dianggap tidak dapat diterima. Dengan demikian, produk dengan skor *System Usability Scale* (SUS) di atas 70 dapat dianggap baik, sedangkan produk dengan skor di bawah 60 sebaiknya ditinjau dan ditingkatkan untuk meningkatkan *usability*-nya.

Setelah perhitungan nilai skor, tahapan terakhir dalam pengujian ini adalah analisis hasil. Pada langkah ini, akan dilakukan analisis dan kesimpulan mengenai tingkat usability metode *flocking* pada kerumunan NPC berdasarkan skor yang diperoleh.

3.5 Perhitungan Manual

Sebagai variabel kontrol dalam perhitungan manual ini, peneliti menetapkan bahwa jarak antara agen dan posisi agen x (P_X) maksimal adalah 10, sesuai dengan kondisi yang diperlukan untuk membentuk sebuah *flock*.

Tabel 3.3 Posisi Agen

Agen	Posisi (x, y, z)
Posisi Agen X (P_X)	(0, 0, 0)
Posisi Agen A (P_A)	(6, 0, 8)
Posisi Agen B (P_B)	(4, 0, -3)
Posisi Agen C (P_C)	(-1, 0, 3)

Berdasarkan posisi setiap agen yang telah dijelaskan dalam Tabel 3.3, dapat dihitung jarak relatif agen (P_X) terhadap semua agen dalam flock.

1. Kalkulasi pengukuran Jarak P_X dengan P_A untuk mengetahui apakah P_X dan P_A dapat menjadi *flock* menggunakan rumus 2.5

$$\begin{aligned}
 d(P_X, P_A) &= \sqrt{(X(P_X) - X(P_A))^2 + (Y(P_X) - Y(P_A))^2 + (Z(P_X) - Z(P_A))^2} \\
 d(P_X, P_A) &= \sqrt{(0 - 6)^2 + (0 - 0)^2 + (0 - 8)^2} \\
 d(P_X, P_A) &= \sqrt{(0 - 6)^2 + (0 - 8)^2} \\
 d(P_X, P_A) &= \sqrt{6^2 + 8^2} \\
 d(P_X, P_A) &= \sqrt{36 + 64} \\
 d(P_X, P_A) &= \sqrt{100} \\
 d(P_X, P_A) &= 10
 \end{aligned}$$

Jarak maximal = $d_1 = d(P_X, P_A) = 10$, Jarak minimal = $d_2 = 0$

Syarat menjadi *flocking* $\Rightarrow d(P_X, P_A) \leq d_1 \cap d(P_X, P_A) \geq d_2$

$$d(P_X, P_A) \leq d_1 \cap d(P_X, P_A) \geq d_2 \approx 10 \leq 10 \cap 10 \geq 0$$

Berdasarkan hasil dari jarak P_X dan P_A dan syarat maximal jarak untuk menjadi *flocking*, dapat disimpulkan bahwa agen A dapat melakukan *flock* dengan agen x.

2. Kalkulasi pengukuran Jarak P_X dengan P_B untuk mengetahui apakah P_X dan P_B dapat menjadi *flock* menggunakan rumus 2.5

$$d(P_X, P_B) =$$

$$\sqrt{(X(P_X) - X(P_B))^2 + (Y(P_X) - Y(P_B))^2 + (Z(P_X) - Z(P_B))^2}$$

$$d(P_X, P_A) = \sqrt{(0 - 4)^2 + (0 - 0)^2 + (0 - (-3))^2}$$

$$d(P_X, P_A) = \sqrt{(0 - 4)^2 + (0 - (-3))^2}$$

$$d(P_X, P_A) = \sqrt{4^2 + 3^2}$$

$$d(P_X, P_A) = \sqrt{16 + 9}$$

$$d(P_X, P_A) = \sqrt{25}$$

$$d(P_X, P_A) = 5$$

$$\text{Jarak maximal} = d_1 = d(P_X, P_B) = 10, \text{ Jarak minimal} = d_2 = 0$$

$$\text{Syarat menjadi flocking} \Rightarrow d(P_X, P_B) \leq d_1 \cap d(P_X, P_B) \geq d_2$$

$$d(P_X, P_A) \leq d_1 \cap d(P_X, P_A) \geq d_2 \approx 5 \leq 10 \cap 5 \geq 0$$

Berdasarkan hasil dari jarak P_X dan P_B dan syarat maximal jarak untuk menjadi *flocking*, dapat disimpulkan bahwa agen B dapat melakukan flock dengan agen x.

3. Kalkulasi pengukuran Jarak P_X dengan P_C untuk mengetahui apakah P_X dan P_C dapat menjadi *flock* menggunakan rumus 2.5

$$d(P_X, P_C) =$$

$$\sqrt{(X(P_X) - X(P_C))^2 + (Y(P_X) - Y(P_C))^2 + (Z(P_X) - Z(P_C))^2}$$

$$d(P_X, P_A) = \sqrt{(0 - (-1))^2 + (0 - 3)^2}$$

$$d(P_X, P_A) = \sqrt{1^2 + 3^2}$$

$$d(P_X, P_A) = \sqrt{1 + 9}$$

$$d(P_X, P_A) = \sqrt{10}$$

$$d(P_X, P_A) = 3.16$$

Jarak maximal = $d_1 = d(P_X, P_C) = 10$, Jarak minimal = $d_2 = 0$

Syarat menjadi *flocking* $\Rightarrow d(P_X, P_C) \leq d_1 \cap d(P_X, P_C) \geq d_2$

$$d(P_X, P_A) \leq d_1 \cap d(P_X, P_A) \geq d_2 \approx 3.16 \leq 10 \cap 3.16 \geq 0$$

Berdasarkan hasil dari jarak P_X dan P_B dan syarat maximal jarak untuk menjadi *flocking*, dapat disimpulkan bahwa agen B dapat melakukan flock dengan agen x.

3.5.1 Cohesion

Pada perilaku *flocking* ini, setiap agen dalam kelompok akan bergerak menuju titik tengah relatif antara posisi agen mereka dan agen lain dalam kelompoknya. Misalkan terdapat 4 agen, di mana satu agen (agen x) berperan sebagai agen acuan dengan posisi (P_x), sedangkan sisanya adalah agen dalam satu kelompok dengan agen x dan memiliki posisi (P_b).

Kalkulasi kecepatan *Cohesion* agen x dengan agen satu *flock* yang lain menggunakan rumus 2.2.

$n = 3$ (banyak agen lain selain agen x pada *flock*)

$$V_{cr} = \sum_x^n (P_x - P_b)$$

$$V_{cr} = \sum_x^n (P_x - P_b)$$

$$\begin{aligned} V_{cr} &= ((0, 0, 0) - (6, 0, 8)) + ((0, 0, 0) - (4, 0, -3)) \\ &\quad + ((0, 0, 0) - (-1, 0, 3)) \end{aligned}$$

$$V_{cr} = (-6, 0, -8) + (-4, 0, 3) + (1, 0, -3)$$

$$V_{cr} = (-6, 0, -8) + (-4, 0, 3) + (1, 0, -3)$$

$$V_{cr} = (-9, 0, -8)$$

Berdasarkan hasil pengukuran kohesi pada agen x dengan agen lainnya sebesar (3, 2.6). Artinya Agen x akan berpindah ke arah kecepatan vektor (3, 2, 6) dari posisi saat ini.

3.5.2 Alignment

Pada perilaku *flocking* ini, setiap agen dalam kelompok akan bergerak untuk menyelaraskan arah dan kecepatan vektor dengan agen lain dalam kelompoknya. Misalkan terdapat 4 agen, dengan satu agen (Agen x) sebagai contoh, yang memiliki kecepatan vektor (0, 0, 1) (bergerak ke depan/sumbu z sejauh 1 unit), dan sisanya sebagai anggota kelompok yang menyelaraskan arah dan kecepatan vektor mereka dengan Agen x dan rata-rata penambahan vektor kecepatan dari agen tetangganya.

Kalkulasi pengukuran Kecepatan *Alignment* agen x dengan agen satu *flock* yang lain menggunakan rumus 2.3.

$n = 3$ (banyak agen lain selain agen X pada *flock*)

$$V_{ar} = \frac{1}{n} \sum_x^n v_x$$

$$V_{ar} = \frac{(0, 0, 1) + (0, 0, 1) + (0, 0, 1)}{3}$$

$$V_{ar} = (0, 0, 1)$$

Berdasarkan hasil pengukuran *alignment* pada agen x dengan agen lainnya sebesar (0, 0, 1). Artinya, berdasarkan kecepatan dan banyak agen tetangga agen x, agen x akan berpindah sebesar (0, 0, 1) dari posisi saat ini.

3.5.3 Separation

Pada perilaku *flocking* ini, setiap agen dalam kelompok akan bergerak untuk menghindari benturan dengan agen lain dalam kelompoknya. Misalkan terdapat 4 agen, dengan satu agen (Agen x) sebagai contoh, yang memiliki posisi (P_x) sebagai acuan, dan sisanya merupakan anggota kelompok dengan posisi (P_b) dalam satu kelompok bersama Agen x.

Kalkulasi kecepatan *Separation* Agen x dengan Agen satu *flock* yang lain menggunakan rumus 2.4.

$$V_{sr} = \sum_n^x \frac{(v_x + v_b)}{d(P_x, P_b)}$$

$$V_{sr} = \sum_n^x \frac{P_x - P_b}{d(P_x, P_b)}$$

$$V_{sr} = \frac{P_x - P_A}{d(P_x, P_A)} + \frac{P_x - P_B}{d(P_x, P_B)} + \frac{P_x - P_C}{d(P_x, P_C)}$$

$$V_{sr} = \frac{(0,0,0) - (6,0,8)}{10} + \frac{(0,0,0) - (4,0,-3)}{5}$$

$$+ \frac{(0,0,0) - (-1,0,3)}{3.16}$$

$$V_{sr} = \frac{(-6,0,-8)}{10} + \frac{(-4,0,3)}{5} + \frac{(1,0,-3)}{3.16}$$

$$V_{sr} = (-0.6,0,-0.8) + (-0.8,0,0.6) + (0.316,0,-0.948)$$

$$V_{sr} = (-1.084,0,-1.148)$$

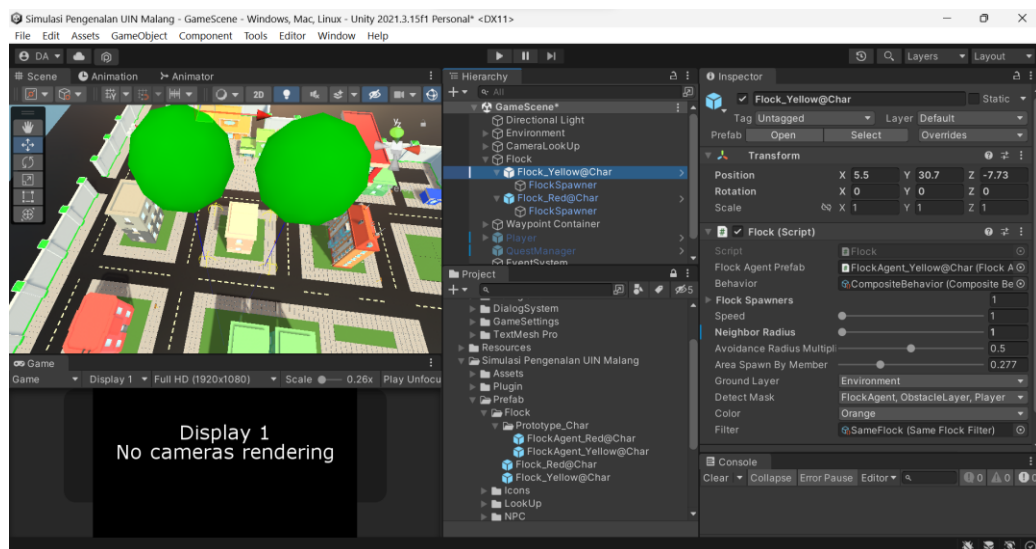
Berdasarkan hasil pengukuran *separation* pada agen x dengan agen lainnya sebesar $(-1.084, 0, -1.148)$. Artinya agen x akan bergerak dengan arah kecepatan sebesar $(-1.084, 0, -1.148)$ dari posisi sekarang.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

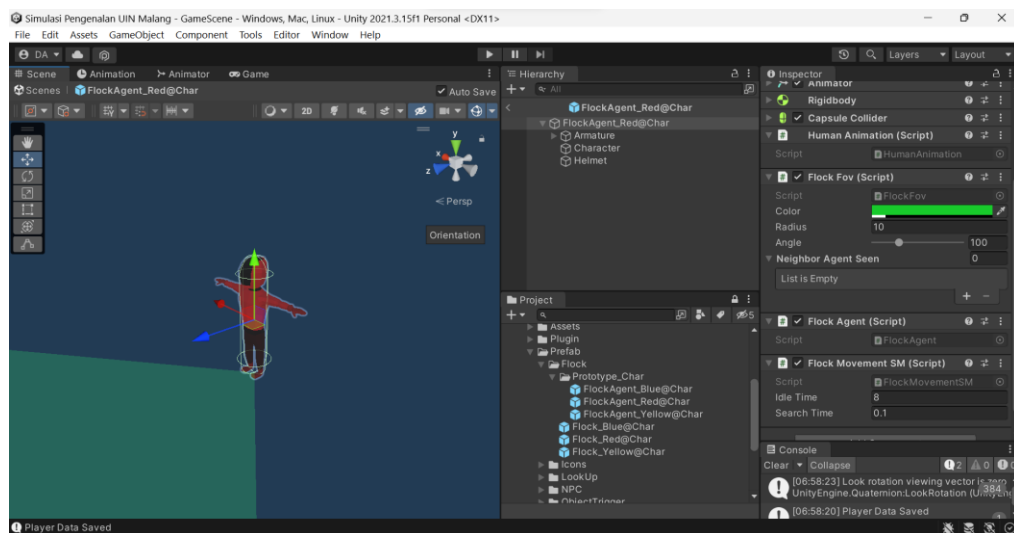
Lingkungan sistem yang diimplementasikan dalam penelitian ini dikembangkan menggunakan *game engine* Unity 3D dengan bahasa pemrograman C#. Sistem ini dijalankan pada perangkat keras dengan spesifikasi sebagai berikut: prosesor Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPU), RAM 12 GB, dan sistem operasi 64 Bit.



Gambar 4.1 *Game Object Flock script*

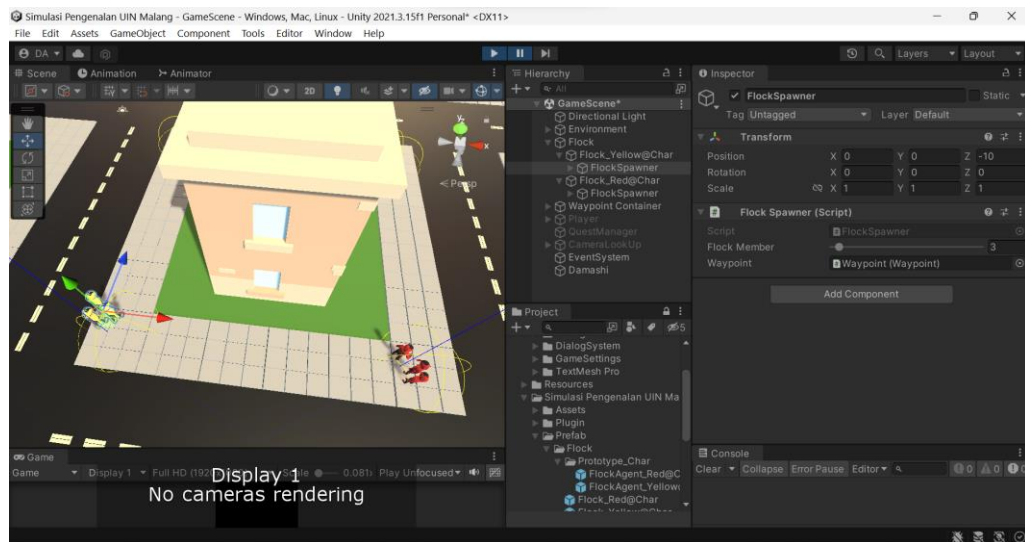
Gambar 4.1 menunjukkan *gameobject* dengan *script Flock*. Dalam sistem *flocking* pada penelitian ini, terdapat dua script utama, yaitu script *Flock* dan *FlockAgent*. Script *Flock* diintegrasikan ke dalam sebuah *Game Object* sebagai komponen utama yang mengendalikan perilaku *flocking* NPC, termasuk pengaturan kecepatan NPC, jenis agen *flocking* yang akan muncul, radius tetangga dalam *flocking*, jumlah agen yang dimunculkan (*spawn*), deteksi lapisan (*layer*) untuk

mengidentifikasi anggota *flock* dan anggota *flock* lain, rintangan, dan pemain (*player*). Di dalam *Game Object* yang menggunakan *script* ini, terdapat juga *script FlockSpawner* yang bertanggung jawab atas penempatan awal dari *flock* NPC dan pengaturan *Waypoint* pertama untuk *flock* NPC tersebut. Selain itu, *script Waypoint* digunakan sebagai tujuan dan pengaturan jalur yang akan diikuti oleh *flock* NPC.



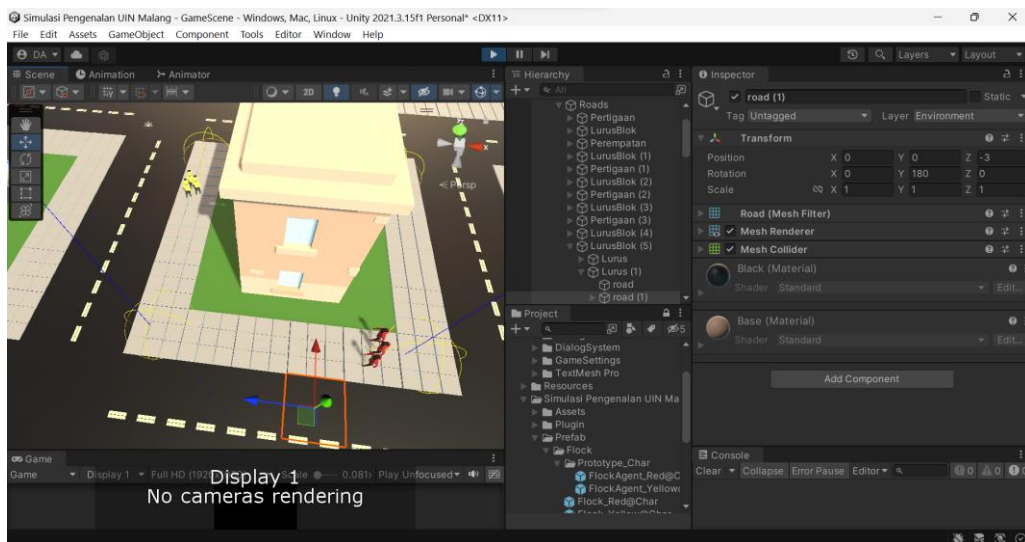
Gambar 4.2 *GameObject FlockAgent script*

Gambar 4.2 menunjukkan *script FlockAgent*, *FlockFOV*, dan *FlockMovementFSM* yang diimplementasikan pada *prefab* NPC. *Script FlockAgent* berperan sebagai agen dalam sistem *flocking* dan mengatur pergerakan agen berdasarkan perhitungan *flocking* yang didefinisikan dalam *script Flock*. Dalam *Game Object* yang sama, terdapat juga *script FlockFOV* (*Field of View*) yang bertugas untuk mengatur deteksi anggota *flock* yang terlihat, serta *script FlockMovementFSM* yang mengendalikan pergerakan agen *flock* melalui *Finite State Machine*.



Gambar 4.3 Flocking NPC Spawn

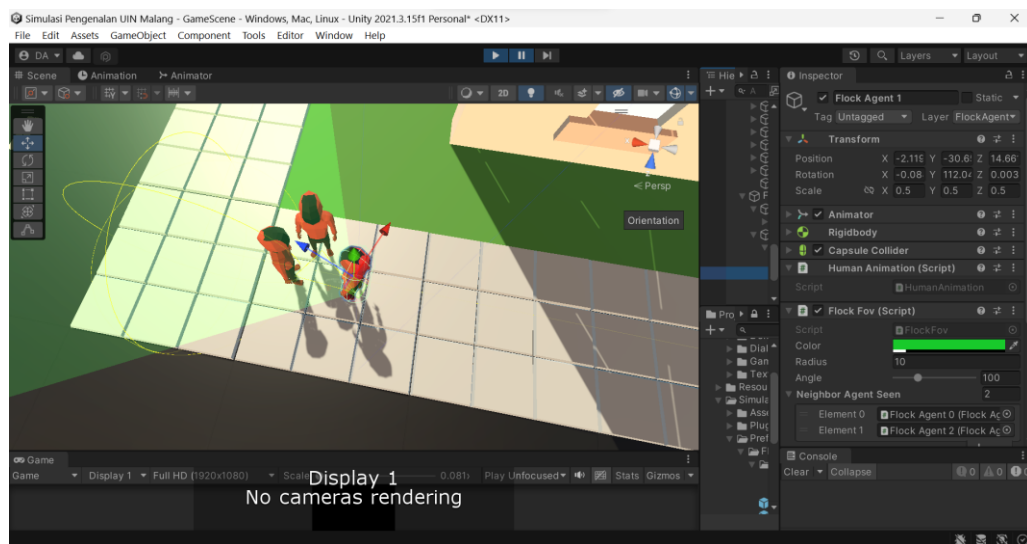
Ketika permainan dimulai, kerumunan NPC akan muncul di lokasi yang telah ditentukan dan bergerak bersama-sama dengan metode *flocking*. Mereka akan mengikuti jalur yang telah ditetapkan oleh *script Waypoint*.



Gambar 4.4 Flocking Movement FSM

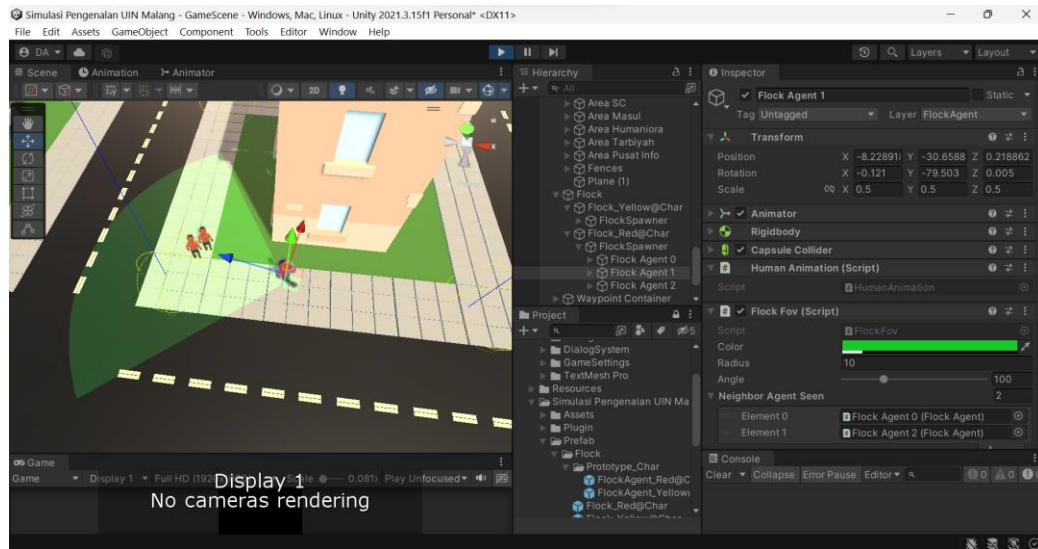
Gambar 4.4 menunjukkan bagaimana kerumunan NPC bergerak bersama menggunakan metode *flocking* dan berada dalam state *Flock*. Sesuai dengan yang dijelaskan dalam desain sistem *Finite State Machine* untuk NPC dalam sistem

flocking, setiap agen *flocking* memiliki empat *state*, yaitu *Flock*, *Idle*, *Search*, *Run*, dan *Individual Move (Walk)*. Agen *flocking* secara bawaan (*default*) akan berada dalam *state Flock*, dan untuk memasuki *state* ini, agen *flocking* setidaknya harus mendeteksi satu tetangga dalam kelompoknya dan berada dalam jangkauan untuk melakukan *flocking* dengan tetangga tersebut.



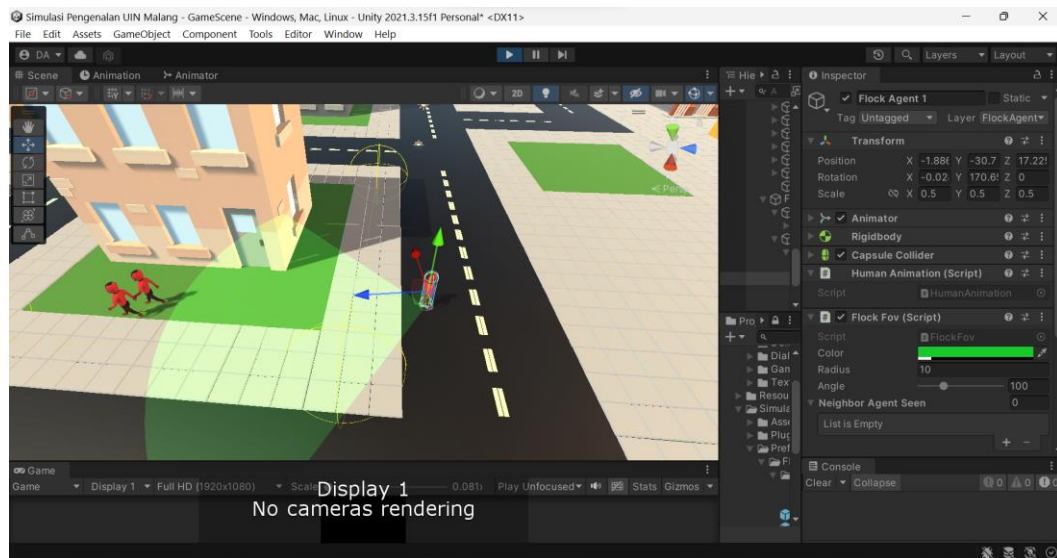
Gambar 4.5 *Flocking Movement FSM Idle*

Gambar 4.5 menunjukkan kondisi *flocking* saat berada dalam *state Idle*. Agen *flocking* akan memasuki *state Idle* ketika mereka mencapai *waypoint* tujuan. Dalam *state* ini, agen akan berdiri diam, menghadap satu sama lain, dan mungkin terlibat dalam percakapan.



Gambar 4.6 *Flocking Movement FSM Search and Run*

Gambar 4.6 menunjukkan agen *flocking* yang sedang berada dalam *state Run*. Ketika agen *flocking* tidak mendeteksi kehadiran tetangga *flocking* yang berdekatan, maka *Finite State Machine* akan mengalihkan agen ke *state Search*. Pada *state* ini, agen akan mencari tetangga dalam jangkauan *Field of View* (FOV) seperti yang ditentukan dalam script *FlockFOV*. Ketika agen *flock* mendeteksi keberadaan tetangga dalam area ini, maka *state* akan berpindah ke *state Run*, di mana agen *flock* akan berlari menuju tetangga *flock* tersebut. Mereka akan kembali ke *state Flock* ketika mencapai jarak *flocking* yang sesuai.



Gambar 4.7 *Flocking Movement FSM Walk*

Gambar 4.7 menggambarkan agen *flock* yang sedang berada dalam *state Walk*. Ketika agen *flock* tidak dapat mendeteksi tetangga sejenisnya dalam jangkauan *Field of View* (FOV), maka agen *flock* akan beralih ke *state Walk*. Dalam *state* ini, agen *flock* akan terus berjalan menuju waypoint sambil mencoba terus-menerus mendeteksi tetangga sejenisnya dalam area *Field of View*. Dengan penjelasan ini, dapat disimpulkan bahwa sistem *flocking* pada kerumunan NPC telah diimplementasikan sesuai dengan desain sistem yang direncanakan.

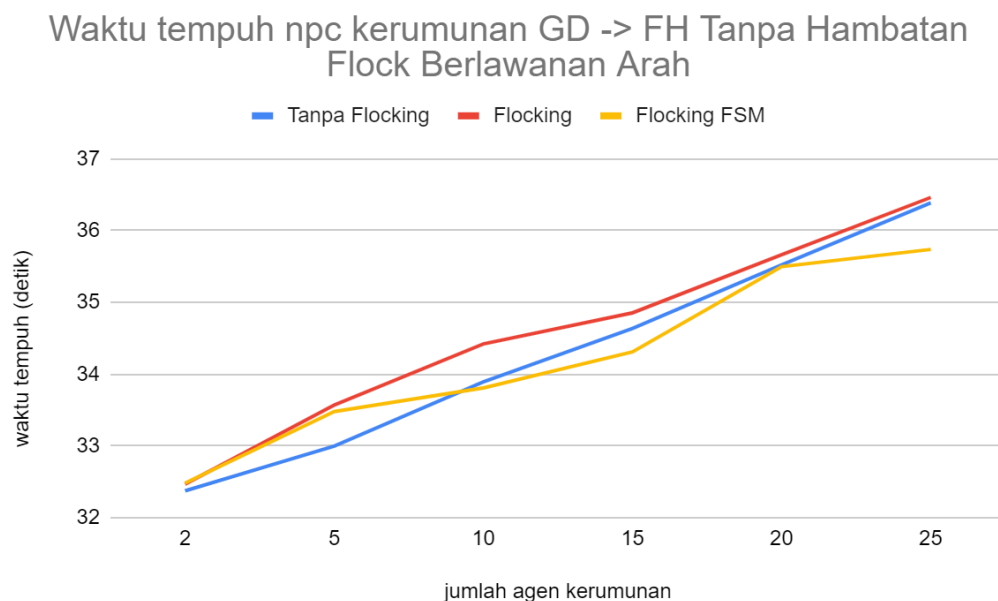
4.2 Pengujian Sistem

Pengujian sistem merupakan tahap kritis dalam evaluasi performa kerumunan NPC dalam lingkungan simulasi *game*. Pengujian ini dirancang untuk menyelidiki kemampuan kerumunan NPC dalam mengatasi hambatan, waktu tempuh, dan tingkat *usability*-nya.

4.2.1 Perbandingan waktu tempuh pada penggunaan metode *flocking*

Sub bab ini mengupas secara mendalam perbandingan waktu tempuh pada penggunaan metode *flocking* dalam simulasi kerumunan NPC. Fokus utama adalah mengevaluasi efisiensi waktu tempuh antara kerumunan NPC yang menerapkan metode *flocking*, kerumunan NPC yang menerapkan metode *flocking* dengan *Finite State Machine* dan kerumunan yang tidak menggunakan metode *flocking*. Melalui serangkaian uji coba, dilakukan pemantauan terhadap bagaimana metode *flocking* mempengaruhi kecepatan kerumunan NPC dalam berbagai konteks simulasi. Hasil analisis ini diharapkan dapat memberikan gambaran yang jelas tentang sejauh mana metode *flocking* dapat meningkatkan atau menghambat kinerja waktu tempuh kerumunan NPC dalam lingkungan simulasi *game*.

4.2.1.1 Perbandingan waktu tempuh kerumunan NPC tanpa hambatan



Gambar 4.8 Hasil Waktu tempuh kerumunan NPC GD > FH Tanpa Hambatan *Flock* Berlawanan Arah

Gambar 4.8 menunjukkan hasil perbandingan waktu tempuh kerumunan NPC dengan posisi awal merupakan referensi posisi gerbang awal pada game dan posisi akhir merupakan referensi posisi fakultas humaniora dalam game dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berikut hasil analisa perbandingan waktu tempuh kerumunan NPC untuk setiap metode berdasarkan Gambar 4.8.

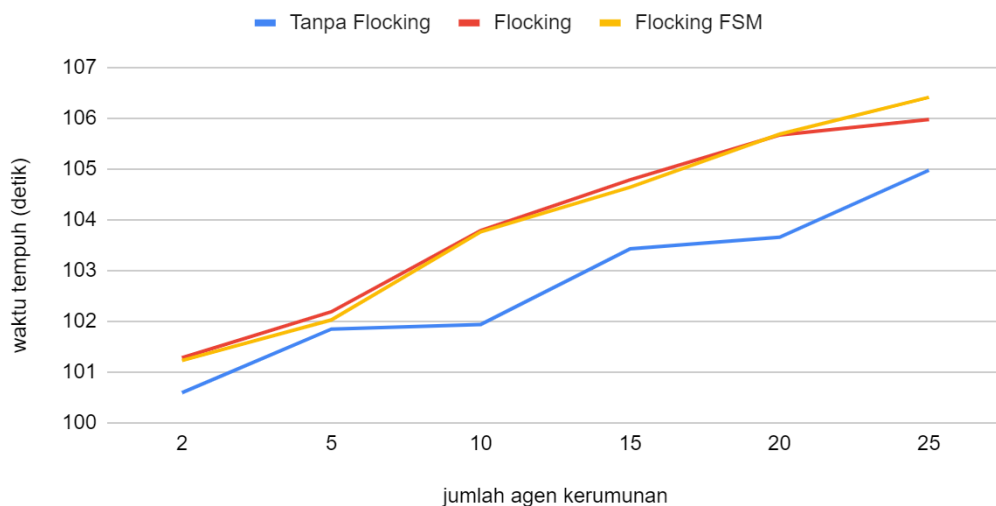
Tabel 4.1 Analisa perbandingan waktu tempuh kerumunan NPC GD->FH tanpa hambatan

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > FH tanpa hambatan (s)			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	32,374	32,468	32,482	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
5	32,998	33,57	33,478	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
10	33,892	34,422	33,808	[<i>Flocking</i> FSM, <i>Flocking</i> , Tanpa <i>Flocking</i>]
15	34,638	34,854	34,31	[<i>Flocking</i> FSM, Tanpa <i>Flocking</i> , <i>Flocking</i>]
20	35,522	35,664	35,498	[<i>Flocking</i> FSM, Tanpa <i>Flocking</i> , <i>Flocking</i>]
25	36,386	36,4625	35,738	[<i>Flocking</i> FSM, Tanpa <i>Flocking</i> , <i>Flocking</i>]

Berdasarkan data pada Tabel 4.1, didapatkan hasil pengamatan sebagai berikut:

- Banyak agen pada kerumunan NPC mempengaruhi waktu tempuh dari kerumunan NPC.
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang lebih singkat pada percobaan dengan jumlah agen di atas 10.
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu tempuh terlama pada iterasi percobaan ini meskipun dengan perbedaan waktu yang tidak terlalu signifikan.

Waktu tempuh npc kerumunan GD -> Saintek Tanpa Hambatan Flock Berlawanan Arah



Gambar 4.9 Hasil Waktu tempuh kerumunan NPC GD > Saintek Tanpa Hambatan *Flock* Berlawanan Arah

Dalam rangka pengujian tambahan, peneliti melaksanakan observasi perbandingan waktu tempuh kerumunan NPC antara titik awal yang diacu sebagai posisi gerbang depan dalam lingkungan game dan titik akhir yang diacu sebagai

posisi fakultas saintek dalam game, dengan menggunakan protokol pengujian yang serupa dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Hasil rinci dari pengujian ini dapat ditemukan pada Gambar 4.9. Hasil analisa dari grafik tersebut dijabarkan sebagai berikut

Tabel 4.2 Analisa perbandingan waktu tempuh kerumunan NPC GD->Saintek tanpa hambatan

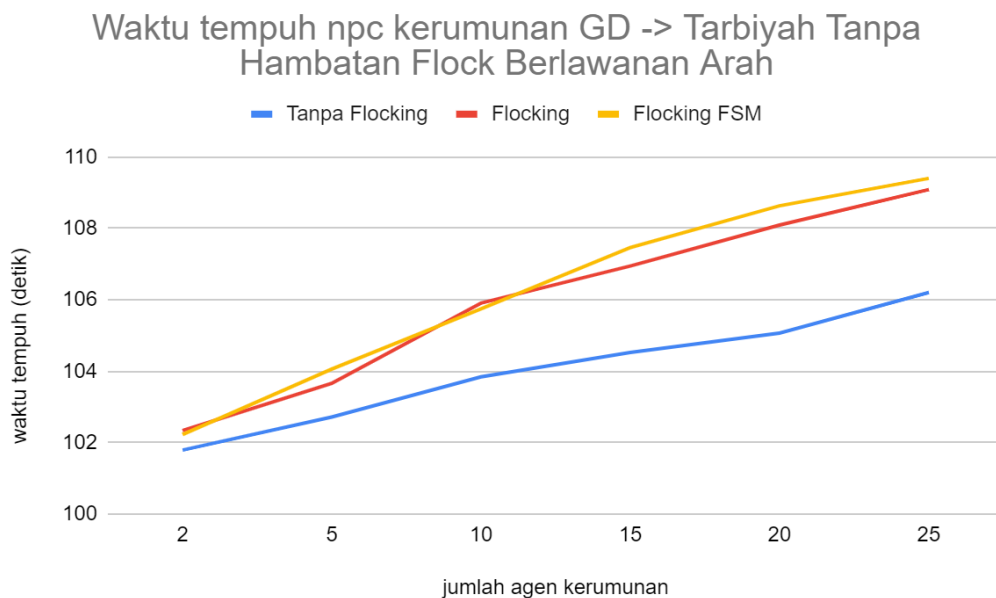
Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Saintek tanpa hambatan (s)			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	100,594	101,282	101,228	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	101,848	102,186	102,026	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
10	101,938	103,788	103,762	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
15	103,428	104,784	104,638	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	103,654	105,668	105,684	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
25	104,972	105,972	106,408	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan data pada Tabel 4.2, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa menggunakan metode *flocking* menunjukkan waktu tempuh paling singkat dalam percobaan ini, dengan perbedaan waktu yang signifikan dibandingkan dengan kerumunan NPC yang menerapkan metode

flocking dan yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).

- b. Kerumunan NPC tanpa menggunakan metode *flocking* memiliki waktu tempuh paling singkat dalam percobaan ini, dengan perbedaan waktu yang signifikan jika dibandingkan dengan kerumunan NPC yang menerapkan metode *flocking* dan yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).
- c. Kerumunan NPC dengan metode *flocking* dan *Finite State Machine* (FSM) memiliki waktu tempuh yang cenderung lebih cepat, meskipun perbedaannya tidak begitu signifikan dibandingkan dengan kerumunan NPC yang hanya menggunakan metode *flocking*.



Gambar 4.10 Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah Tanpa Hambatan *Flock* Berlawanan Arah

Pengujian serupa diterapkan pada kerumunan NPC dengan posisi awal di depan gerbang dalam permainan dan posisi akhir di dekat Fakultas Tarbiyah dalam *game* dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Gambar 4.10 menampilkan hasil perbandingan waktu tempuh kerumunan NPC yang tidak mengalami hambatan antara posisi awal di depan gerbang permainan dan posisi akhir di dekat Fakultas Saintek pada *game*. Analisis hasil grafik, seperti yang diilustrasikan pada Gambar 4.10, dapat dijabarkan sebagai berikut.

Tabel 4.3 Analisa perbandingan waktu tempuh kerumunan NPC GD->Tarbiyah tanpa hambatan

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Tarbiyah tanpa hambatan			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	101,792	102,34	102,228	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	102,716	103,662	104,062	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
10	103,844	105,906	105,746	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
15	104,528	106,946	107,462	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	105,068	108,094	108,632	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
25	106,206	109,088	109,402	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]

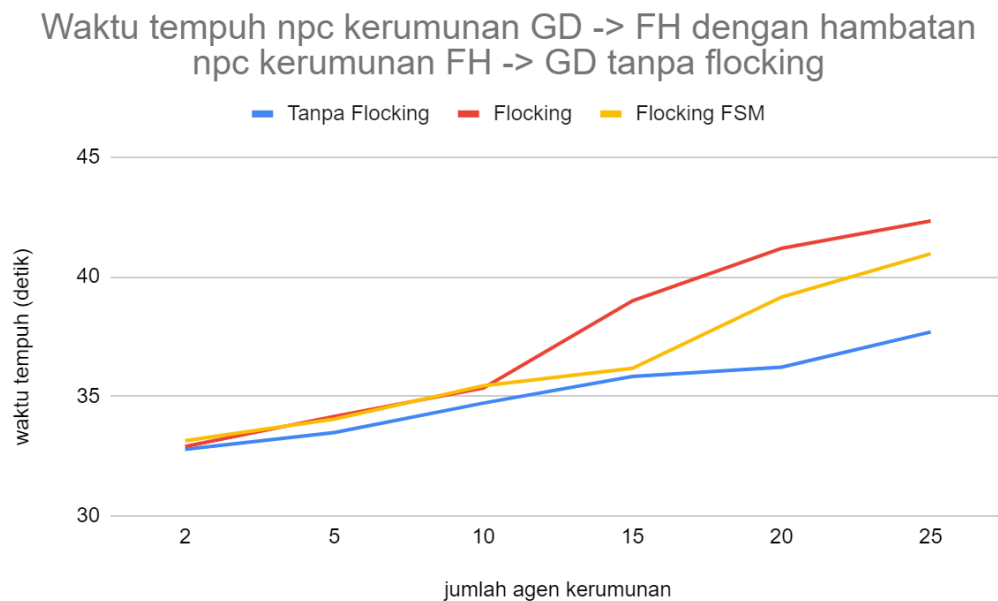
Berdasarkan data pada Tabel 4.3, didapatkan hasil pengamatan sebagai berikut:

- a. Tidak ada perbedaan signifikan antara kerumunan NPC yang menerapkan metode *flocking* dan yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM),
- b. Terdapat kecenderungan kerumunan NPC yang menerapkan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih cepat dibandingkan dengan yang hanya menggunakan metode *flocking* saja.

Berdasarkan percobaan uji waktu tempuh kerumunan NPC tanpa hambatan, didapatkan hasil sebagai berikut:

- a. Pada rute yang sama, semakin banyak jumlah agen pada kerumunan NPC pada metode manapun, waktu tempuh kerumunan NPC menjadi semakin lama
- b. Pada percobaan tanpa hambatan, kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang jauh lebih singkat dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Perbedaan waktu tempuh ini menjadi semakin besar ketika dihadapkan dengan rute yang lebih jauh.

4.2.1.2 Perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC tanpa metode *flocking*



Gambar 4.11 Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan Flock Berlawanan Arah tanpa metode *flocking*

Gambar 4.11 memperlihatkan perbandingan waktu tempuh antara kerumunan NPC yang memulai perjalanan dari gerbang awal dalam permainan dan berakhir di Fakultas Humaniora dalam permainan dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Pada pengujian ini, kerumunan NPC menghadapi hambatan berupa kerumunan NPC tanpa menerapkan metode *flocking*,

dengan jumlah agen yang sama. Hasil analisa dari grafik yang diilustrasikan pada gambar 4.11 dijabarkan sebagai berikut

Tabel 4.4 Analisa perbandingan waktu tempuh kerumunan NPC GD->FH dengan hambatan kerumunan NPC tanpa metode *flocking*

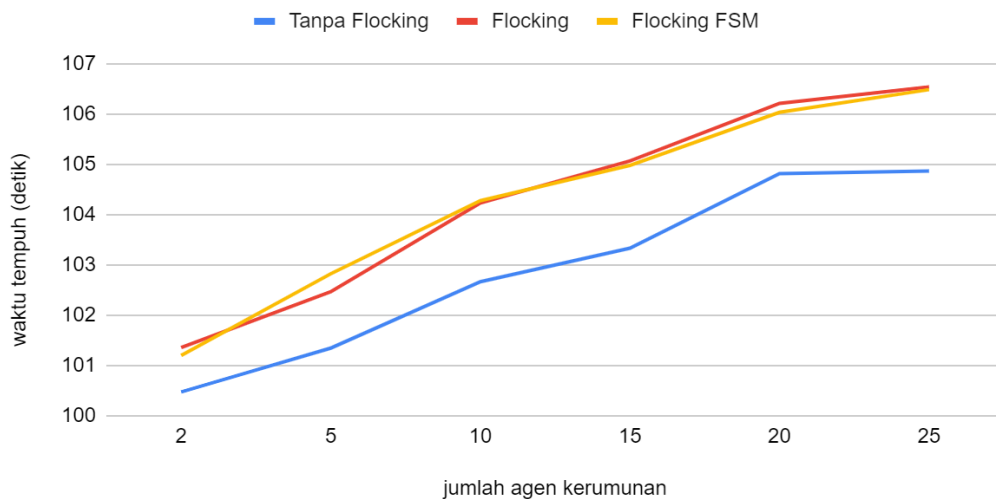
Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > FH dengan hambatan kerumunan NPC tanpa metode <i>flocking</i>			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	32,786	32,9025	33,144	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
5	33,486	34,164	34,05	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
10	34,716	35,344	35,442	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
15	35,83	38,992	36,17	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	36,214	41,184	39,145	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
25	37,688	42,328	40,954	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan data pada Tabel 4.4, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling singkat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang lebih singkat pada jumlah agen yang lebih besar (jumlah agen di atas 15) dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja.
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu yang lebih singkat dibandingkan dengan kerumunan NPC yang menggunakan

metode *flocking* dengan *Finite State Machine* (FSM) untuk jumlah agen 15 kebawah .

Waktu tempuh npc kerumunan GD -> Saintek dengan hambatan npc kerumunan tanpa flocking



Gambar 4.12 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan Flock Berlawanan Arah tanpa metode *flocking*

Gambar 4.12 mengilustrasikan perbandingan waktu tempuh antara kerumunan NPC yang dihadapkan pada hambatan, dengan jumlah agen yang sama, dimana posisi awal mengacu pada gerbang depan dalam permainan dan posisi akhir mengacu pada Fakultas Saintek dalam permainan dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berikut merupakan penjabaran analisis berdasarkan gambar 4.12

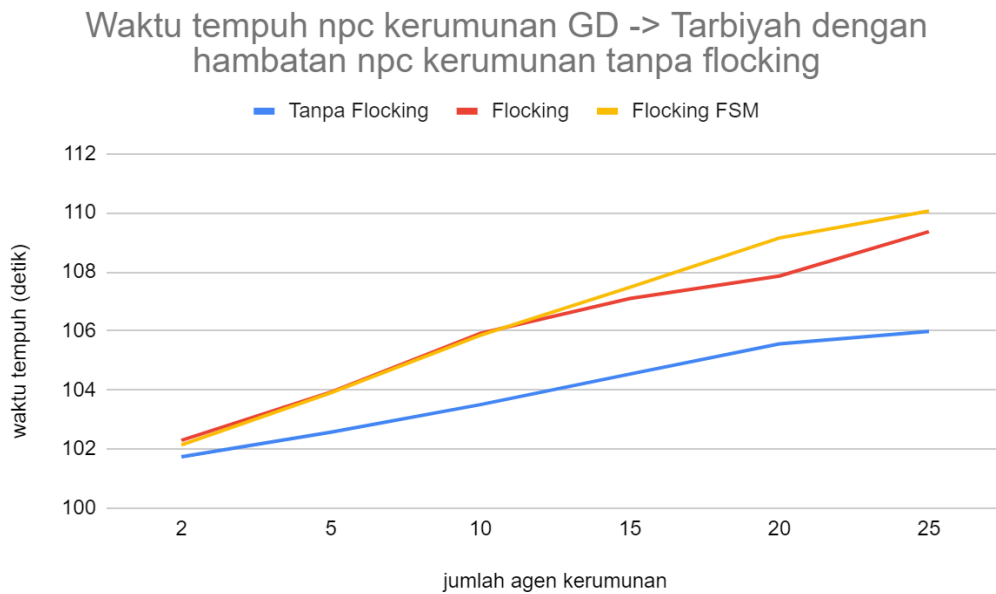
Tabel 4.5 Analisa perbandingan waktu tempuh kerumunan NPC GD-> Saintek dengan hambatan kerumunan NPC tanpa metode *flocking*

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Saintek dengan hambatan kerumunan NPC tanpa metode <i>flocking</i>			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	100,474	101,355	101,2	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	101,346	102,468	102,824	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
10	102,664	104,23	104,276	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
15	103,332	105,066	104,976	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
20	104,814	106,21	106,032	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
25	104,866	106,536	106,482	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan data pada tabel 4.5, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang hanya menggunakan metode *flocking* dengan yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) didapatkan hasil waktu tempuh yang tidak memiliki perbedaan yang signifikan
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 20 keatas dari pada kerumunan NPC yang menggunakan metode *flocking* saja
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu yang lebih singkat dibandingkan dengan kerumunan NPC yang menggunakan

metode *flocking* dengan *Finite State Machine* (FSM) untuk jumlah agen 15 kebawah.



Gambar 4.13 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan Flock Berlawanan Arah tanpa metode *flocking*

Gambar 4.13 menggambarkan perbandingan waktu tempuh kerumunan NPC yang bergerak dari posisi awal, yaitu gerbang depan dalam permainan, hingga posisi akhir, yaitu fakultas Tarbiyah dalam permainan dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berikut merupakan penjabaran analisis berdasarkan gambar 4.13

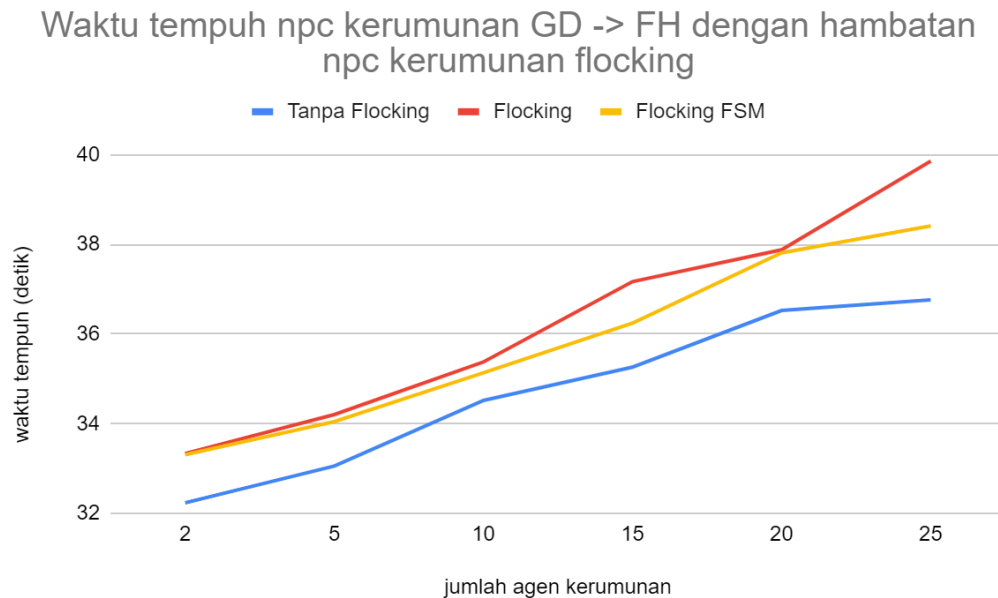
Tabel 4.6 Analisa perbandingan waktu tempuh kerumunan NPC GD -> Tarbiyah dengan hambatan kerumunan NPC tanpa metode *flocking*

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Tarbiyah dengan hambatan kerumunan NPC tanpa metode <i>flocking</i>			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	101,74	102,298	102,142	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	102,574	103,928	103,906	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
10	103,508	105,922	105,856	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
15	104,542	107,1	107,48	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
20	105,566	107,862	109,148	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
25	105,984	109,364	110,06	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]

Berdasarkan pada data tabel 4.6, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 10 ke bawah dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 15 keatas dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).

4.2.1.3 Perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC dengan metode *flocking*



Gambar 4.14 Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan Flock Berlawanan Arah menggunakan metode *flocking*

Gambar 4.14 memperlihatkan perbandingan waktu tempuh kerumunan NPC, yang berasal dari posisi awal, yaitu gerbang awal dalam permainan, menuju posisi akhir, yakni fakultas Humaniora dalam permainan dengan hambatan kerumunan NPC yang menggunakan metode *flocking* dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berikut merupakan penjabaran analisis terhadap gambar 4.14

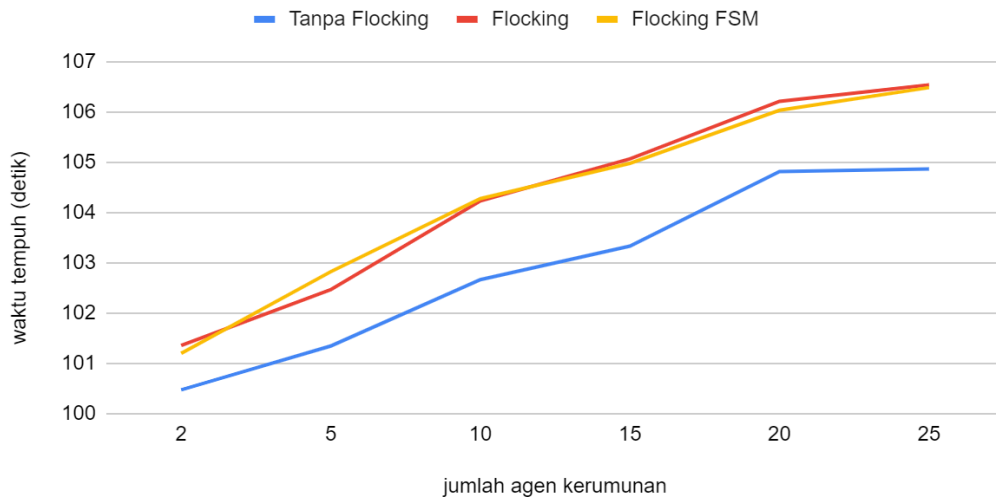
Tabel 4.7 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode *flocking*

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > FH dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i>			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	32,234	33,334	33,308	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	33,054	34,202	34,046	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
10	34,518	35,378	35,134	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
15	35,26	37,17	36,244	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	36,526	37,88	37,812	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
25	36,764	39,86	38,412	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan data pada tabel 4.7, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 15 keatas dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 10 ke bawah dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).

Waktu tempuh npc kerumunan GD -> Saintek dengan hambatan npc kerumunan tanpa flocking



Gambar 4.15 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan Flock Berlawanan Arah menggunakan metode flocking

Gambar 4.15 memperlihatkan hasil perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC menggunakan metode *flocking*. Posisi awal kerumunan mengacu pada gerbang depan dalam *game*, sementara posisi akhir merujuk pada fakultas Saintek dalam *game* dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berikut merupakan penjabaran analisis dari gambar 4.15

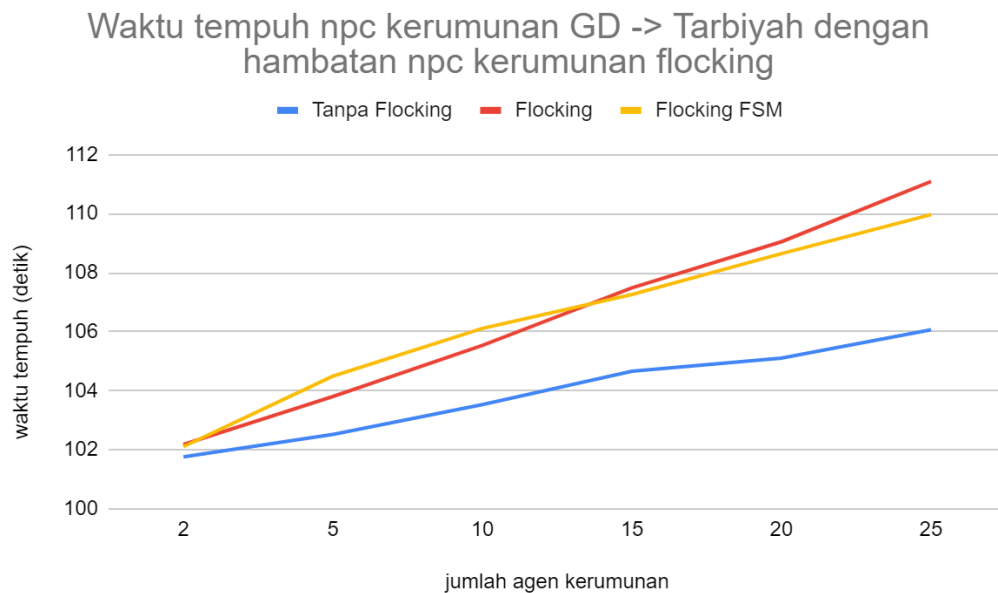
Tabel 4.8 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode *flocking*

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Saintek dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i>			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	100,62	101,432	101,194	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	101,834	103,012	102,614	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
10	102,152	104,27	104,174	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
15	103,058	105,426	105,474	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	103,782	107,798	106,586	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
25	105,038	107,56	107,136	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan data pada tabel 4.8, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 15 keatas dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 10 keatas dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).

Sedangkan untuk jumlah agen 10 kebawah kerumunan NPC yang menggunakan metode *flocking* memiliki waktu yang lebih singkat dibandingkan dengan yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).



Gambar 4.16 Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan Hambatan Flock Berlawanan Arah menggunakan metode flocking

Gambar 4.16 menggambarkan perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC yang menggunakan metode *flocking*. Pengujian dilakukan dengan posisi awal kerumunan merujuk pada gerbang depan dalam game, sedangkan posisi akhir merujuk pada fakultas tarbiyah dalam game dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC

menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berikut penjabaran analisis berdasarkan gambar 4.16

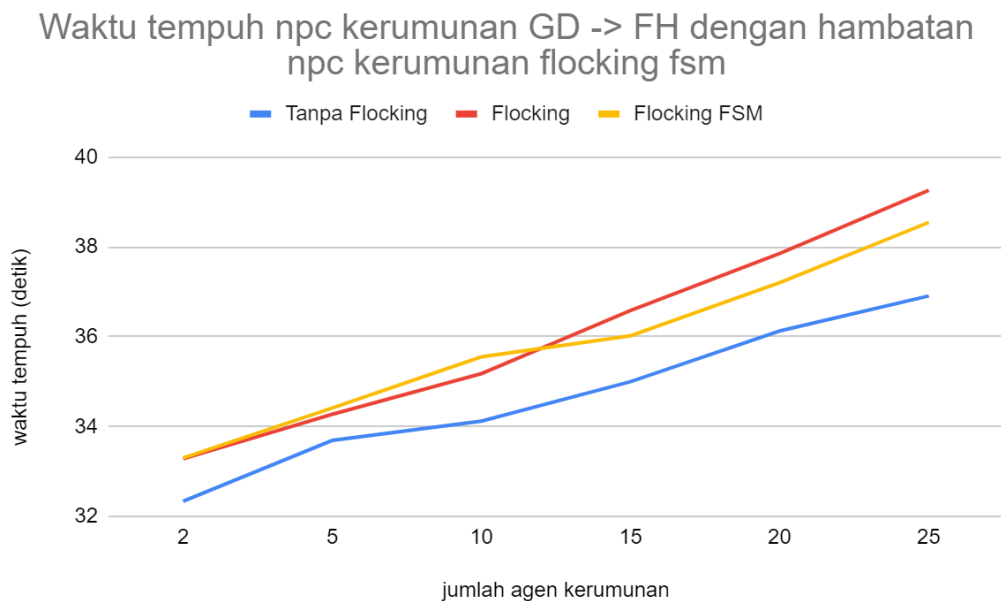
Tabel 4.9 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode *flocking*

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Tarbiyah dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i>			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	101,76	102,182	102,11	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	102,524	103,806	104,498	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
10	103,536	105,536	106,114	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
15	104,662	107,488	107,262	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	105,106	109,048	108,648	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
25	106,066	111,092	109,968	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan data pada tabel 4.9, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 15 keatas dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja.
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 10 kebawah dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).

4.2.1.4 Perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC dengan metode *flocking* dengan *FSM*



Gambar 4.17 Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan *Flock* Berlawanan Arah menggunakan metode *flocking* FSM

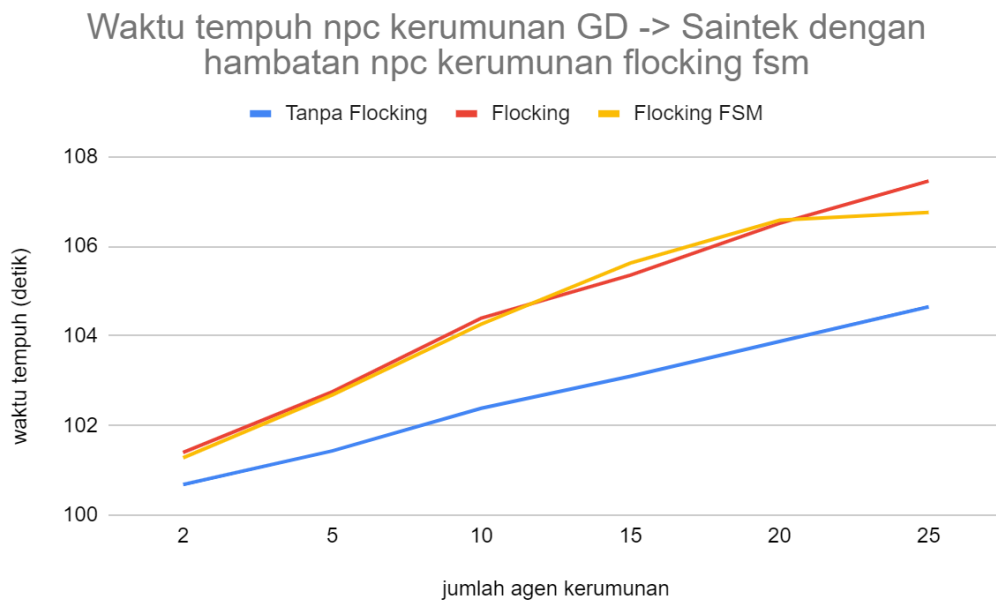
Gambar 4.17 mengilustrasikan perbandingan waktu tempuh kerumunan NPC yang bergerak dari gerbang awal dalam game menuju fakultas humaniora dalam game dengan hambatan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM).. Berdasarkan gambar 4.17, dapat dijabarkan hasil analisis sebagai berikut

Tabel 4.10 Analisa perbandingan waktu tempuh kerumunan NPC GD -> FH dengan hambatan kerumunan NPC menggunakan metode *flocking* FSM

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > FH dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i> fsm			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	32,334	33,28	33,298	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
5	33,69	34,272	34,414	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
10	34,12	35,178	35,552	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
15	34,998	36,584	36,02	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	36,128	37,852	37,202	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
25	36,908	39,26	38,544	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan data pada tabel 4.10, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 15 keatas dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja
- Kerumunan NPC yang menggunakan metode *flocking* memiliki waktu tempuh yang sedikit lebih singkat pada jumlah agen 10 kebawah dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM).



Gambar 4.18 Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan *Flock* Berlawanan Arah menggunakan metode *flocking* FSM

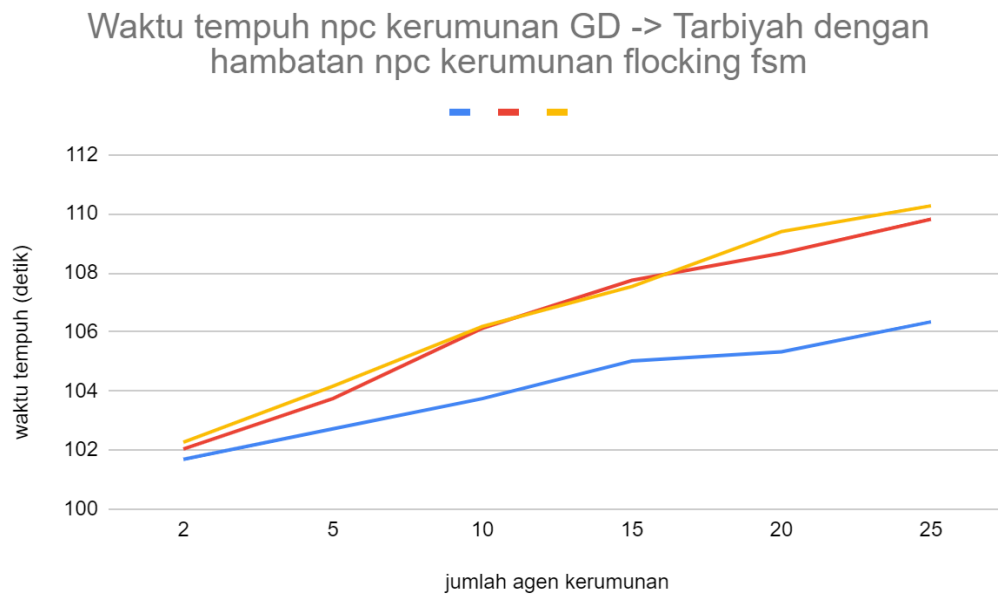
Gambar 4.18 memperlihatkan hasil perbandingan waktu tempuh kerumunan NPC dengan hambatan kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM) dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berdasarkan ilustrasi pada gambar 4.18, hasil analisa dapat dijabarkan sebagai berikut

Tabel 4.11 Analisa perbandingan waktu tempuh kerumunan NPC GD -> Saintek dengan hambatan kerumunan NPC menggunakan metode *flocking* FSM

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Saintek dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i> fsm			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa <i>Flocking</i>	<i>Flocking</i>	<i>Flocking</i> FSM	
2	100,682	101,4	101,28	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
5	101,432	102,75	102,682	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
10	102,384	104,398	104,264	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
15	103,098	105,354	105,626	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
20	103,878	106,512	106,584	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
25	104,65	107,458	106,758	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]

Berdasarkan pada data tabel 4.11, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih singkat kecuali pada iterasi percobaan dengan jumlah 15 dan 20 dengan selisih waktu tempuh yang sangat kecil dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja.



Gambar 4.19 Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan Hambatan *Flock* Berlawanan Arah menggunakan metode *flocking* FSM

Gambar 4.19 memperlihatkan perbandingan waktu tempuh kerumunan NPC yang bergerak dari posisi awal, merujuk pada gerbang depan dalam *game*, hingga posisi akhir yang merujuk pada fakultas tarbiyah dengan hambatan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) dengan garis biru pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC tanpa menggunakan metode *flocking*, garis merah pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking*, dan garis kuning pada gambar merepresentasikan waktu tempuh berbanding dengan jumlah agen kerumunan NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Berdasarkan ilustrasi pada Gambar 4.19, hasil analisis dapat dijabarkan sebagai berikut

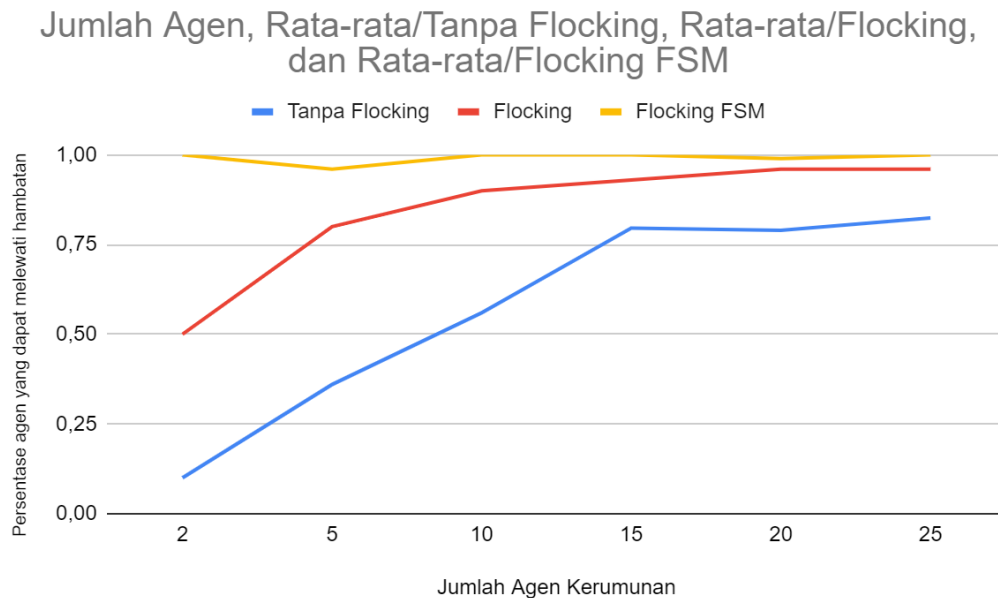
Tabel 4.12 Analisa perbandingan waktu tempuh kerumunan NPC GD -> Saintek dengan hambatan kerumunan NPC menggunakan metode *flocking* FSM

Jumlah agen	Waktu tempuh Kerumunan NPC rute GD > Tarbiyah dengan hambatan kerumunan NPC menggunakan metode <i>flocking</i> fsm			Hasil perbandingan Waktu tempuh (berdasarkan waktu tempuh tercepat)
	Tanpa Flocking	Flocking	Flocking FSM	
2	101,686	102,038	102,266	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
5	102,722	103,7425	104,166	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
10	103,744	106,126	106,186	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
15	105,02	107,75	107,54	[Tanpa <i>Flocking</i> , <i>Flocking</i> FSM, <i>Flocking</i>]
20	105,328	108,666	109,4	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]
25	106,342	109,818	110,27	[Tanpa <i>Flocking</i> , <i>Flocking</i> , <i>Flocking</i> FSM]

Berdasarkan data pada tabel 4.12, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang paling cepat
- Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki waktu tempuh yang sedikit lebih lambat selisih waktu tempuh yang sangat kecil dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja.

4.2.2 Perbandingan Kemampuan Kerumunan NPC Menghadapi Hambatan Statis Menggunakan Metode Flocking



Gambar 4.20 Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan Hambatan Flock Berlawanan Arah menggunakan metode flocking FSM

Gambar 4.20 memvisualisasikan perbandingan presentase agen dalam kerumunan NPC yang mampu melewati hambatan statis pada setiap metode uji dan iterasi percobaan. Sumbu x pada gambar merepresentasikan jumlah agen dalam kerumunan NPC pada setiap iterasi percobaan. Sumbu y pada gambar 4.20 menggambarkan persentase agen dalam kerumunan yang berhasil melintasi hambatan statis. Garis biru pada gambar mencerminkan persentase agen dalam kerumunan NPC tanpa menggunakan metode flocking yang berhasil melewati hambatan statis, berbanding dengan jumlah agen dalam kerumunan. Garis merah pada gambar mencerminkan persentase agen dalam kerumunan NPC yang menggunakan metode *flocking* dan berhasil melewati hambatan statis, dengan perbandingan terhadap jumlah agen dalam kerumunan. Garis kuning pada Gambar

4.20 mencerminkan persentase agen dalam kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* dan berhasil melewati hambatan statis, dengan perbandingan terhadap jumlah agen dalam kerumunan. Hasil yang diilustrasikan oleh Gambar 4.20 dapat dijelaskan sebagai berikut:

- a. Persentase agen kerumunan NPC yang dapat melewati hambatan statis meningkat selaras dengan jumlah agen kerumunan NPC pada tiap metode uji
- b. Kerumunan NPC tanpa metode *flocking* memiliki persentase agen kerumunan yang dapat melewati hambatan yang paling rendah dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM)
- c. Kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* memiliki persentase agen yang dapat melewati hambatan statis paling besar diantara kerumunan NPC dengan metode uji yang lain.

4.2.3 Pengujian Tingkat Usability Metode *Flocking* Pada Kerumunan NPC

Tabel 4.13 Data hasil survey yang diolah menggunakan metode System Usability Scale (SUS)

No	Responden	Skor Responden									
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Responden 1	3	2	4	1	4	2	4	2	5	2
2	Responden 2	4	2	4	4	4	2	4	2	4	2
3	Responden 3	3	3	3	2	3	3	3	2	4	2
4	Responden 4	3	3	3	2	4	3	4	4	2	4
5	Responden 5	1	4	2	4	3	4	1	3	1	5
6	Responden 6	3	3	3	3	3	3	3	3	3	3
7	Responden 7	4	3	4	2	5	3	3	2	4	1
8	Responden 8	4	3	4	2	5	3	3	2	4	1
9	Responden 9	3	4	4	2	5	2	4	1	5	1

No	Responden	Skor Responden									
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
10	Responden 10	5	5	5	2	5	1	2	2	5	5
11	Responden 11	4	2	3	4	4	3	4	3	3	4
12	Responden 12	3	1	4	5	1	1	1	5	5	1
13	Responden 13	5	5	5	5	5	1	5	2	5	1
14	Responden 14	5	1	5	1	5	1	3	1	5	1
15	Responden 15	5	1	5	1	5	5	1	5	1	5
16	Responden 16	5	1	5	1	5	1	5	1	5	1
17	Responden 17	4	1	4	2	3	2	4	2	5	2
18	Responden 18	5	1	5	1	5	1	5	1	5	1

Tabel 4.13 adalah tabel yang memuat daftar jawaban yang diberikan oleh para responden terhadap pertanyaan kuesioner yang diajukan dalam pengujian. Setiap pertanyaan dalam kuesioner memiliki skala penilaian dari 1 hingga 5, yang digunakan untuk mengevaluasi berbagai aspek *usability* metode *flocking* pada permainan "Game Pengenalan UIN Malang." Hasil dari jawaban para responden akan dijumlahkan sesuai dengan rumus yang telah ditetapkan dalam metode *System Usability Scale* (SUS). Dengan proses perhitungan ini, kita akan mendapatkan tingkat *usability* dari metode *flocking* yang diimplementasikan dalam permainan, yang menjadi tolok ukur penting dalam penilaian kualitas permainan dan pengalaman pemain. Dengan demikian, Tabel 4.2 menjadi elemen kunci dalam menyimpulkan, merangkum, dan menghitung skor yang akhirnya memberikan informasi tentang sejauh mana metode *flocking* meningkatkan *usability* permainan.

4.3 Analisis

4.3.1 Analisis Data Perbandingan Waktu Tempuh Pada Penggunaan Metode *Flocking*

Berdasarkan data yang dihasilkan dari pengujian perbandingan waktu tempuh kerumunan NPC, ditemukan hasil sebagai berikut:

- a. Secara keseluruhan, kerumunan NPC yang tidak menggunakan metode *flocking* memiliki waktu tempuh yang lebih singkat dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* dan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine*. Namun kerumunan NPC yang tidak menggunakan metode *flocking* tidak dapat melakukan penghindaran dan tidak memiliki pola kerumunan seperti pada kerumunan NPC yang menggunakan metode *flocking*.
- b. Didapatkan hasil waktu tempuh yang menunjukkan sedikit perbedaan waktu tempuh antara kerumunan NPC yang menggunakan metode *flocking* dan yang menggunakan metode *flocking* dengan *Finite State Machine* baik pada pengujian tanpa hambatan maupun dengan hambatan dengan kecenderungan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* sedikit lebih cepat dibandingkan dengan yang hanya menggunakan metode *flocking* saja..
- c. Terdapat perbedaan waktu yang signifikan antara kerumunan NPC yang tidak menggunakan *flocking* dengan kerumunan NPC yang menggunakan metode *flocking* maupun dengan yang menggunakan metode *flocking* dengan *Finite*

State Machine (FSM). Perbedaan waktu ini semakin terlihat pada percobaan dengan rute yang memiliki jarak yang jauh.

- d. Pada percobaan dengan hambatan, didapatkan hasil yang menunjukkan pada rute lurus (rute gerbang depan - fakultas humaniora) yang memungkinkan kerumunan NPC untuk bertumbuk dengan hambatan secara langsung, kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* memiliki waktu tempuh yang sedikit lebih singkat dibandingkan dengan kerumunan NPC yang menggunakan metode *flocking* saja. Pada rute yang memiliki belokan, cenderung tidak terjadi tabrakan secara langsung antara kerumunan NPC dengan hambatan sehingga perbedaan waktu tempuh yang dihasilkan antara kerumunan NPC yang menggunakan metode *flocking* dengan kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) perbedaannya waktu tempuh yang didapatkan tidak terlalu signifikan.

4.3.2 Analisis Data Kemampuan Kerumunan NPC Menghadapi Hambatan Statis Menggunakan Metode Flocking

Berdasarkan data yang dihasilkan dari pengujian perbandingan waktu tempuh kerumunan NPC, ditemukan hasil sebagai berikut:

- a. Secara keseluruhan, kerumunan NPC tanpa metode *flocking* memiliki persentase agen yang dapat melewati hambatan statis paling rendah dibandingkan dengan kerumunan NPC dengan metode uji lain. Hal ini dikarenakan kerumunan NPC tanpa metode *flocking* tidak memiliki handle untuk menghindari halangan sehingga membuat agen pada kerumunan NPC

tanpa metode *flocking* memiliki kemungkinan yang tinggi untuk *stuck* pada hambatan statis

- b. Berdasarkan hasil data yang didapatkan, kerumunan NPC yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM) memiliki persentase agen kerumunan NPC yang dapat melewati hambatan statis paling tinggi dibandingkan dengan kerumunan NPC dengan metode uji lain. Hal ini dikarenakan fungsi pada *state run* dalam *Finite State Machine* (FSM) pada kerumunan NPC yang memungkinkan NPC untuk berlari (menambah kecepatan gerak) ke arah agen tetangga yang terlihat. Kemampuan ini membuat agen kerumunan NPC metode *flocking* dengan *Finite State Machine* (FSM) yang *stuck* pada hambatan statis dengan mudah melepaskan diri dibandingkan dengan kerumunan NPC yang hanya menggunakan metode *flocking* saja.

4.3.3 Analisis Data Pengujian Tingkat Usability Metode *Flocking*

Dalam pengujian tingkat *usability* menggunakan metode *System Usability Scale* (SUS), pertanyaan dengan ekspektasi respon positif akan ditempatkan pada urutan ganjil, sementara pertanyaan dengan ekspektasi respon negatif akan ditempatkan pada urutan genap. Pada setiap pertanyaan ganjil, skor dihitung dengan mengurangi poin maksimal (5 poin) dari poin yang dipilih oleh responden. Sedangkan pada pertanyaan genap, skor dihitung dengan mengurangi poin yang dipilih oleh responden dari poin minimal (1 poin). Hasil perhitungan tersebut kemudian dikalikan dengan 2.5. Hasil perkalian ini merupakan skor akhir dari metode *System Usability Scale* (SUS) yang akan digunakan sebagai referensi penilaian tingkat *usability* metode *flocking* dalam mengelola kerumunan NPC pada

game pengenalan UIN Malang (rumus 2.1). Hasil perhitungan respon dari setiap responden dalam pengujian tingkat *usability* metode *flocking* menggunakan metode *System Usability Scale* adalah sebagai berikut:

Tabel 4.14 Hasil perhitungan skor menggunakan metode *System Usability Scale*

Responden	Skor
Responden 1	77,5
Responden 2	70
Responden 3	60
Responden 4	50
Responden 5	20
Responden 6	50
Responden 7	72,5
Responden 8	72,5
Responden 9	77,5
Responden 10	67,5
Responden 11	55
Responden 12	52,5
Responden 13	77,5
Responden 14	95
Responden 15	50
Responden 16	100
Responden 17	77,5
Responden 18	100
Rata-rata	68,05555556

Berdasarkan tabel 4.14, hasil perhitungan skor dari setiap responden ditemukan bahwa nilai rata-rata skor adalah sebesar 68,05. Skor ini mengindikasikan bahwa tingkat *usability* metode *flocking* pada kerumunan NPC berada dalam kategori kuartil kedua, seperti yang terlihat pada Gambar 3.8.

4.4 Integrasi Islam

Dalam kitabnya yang berjudul Ta'lim Muta'allim, Syaikh Az-Zamruji menjelaskan bahwa setiap individu Muslim memiliki kewajiban untuk mencari pengetahuan setiap saat sesuai dengan kebutuhan yang relevan dengan kondisinya. Hal ini sejalan dengan tujuan penelitian ini, yaitu untuk mencari solusi dalam mengatur perilaku *flocking* pada kerumunan NPC dalam konteks "Game Pengenalan UIN Malang," sebagai implementasi dari sikap mencari pengetahuan dan menerapkannya pada situasi yang relevan, sesuai dengan prinsip yang diajarkan dalam kitab tersebut

Dijelaskan juga pada Al-Qur'an pada Surah Sad ayat 27

وَمَا خَلَقْنَا السَّمَاءَ وَالْأَرْضَ وَمَا بَيْنَهُمَا بَطْلًا ۖ ذَٰلِكَ ظَنُّ الَّذِينَ كَفَرُوا ۖ ۖ فَوَيْلٌ لِلَّذِينَ كَفَرُوا مِنَ النَّارِ

“Dan kami tidak menciptakan langit dan bumi dan apa yang ada di antara keduanya tanpa hikmah. Yang demikian adalah anggapan orang-orang kafir, maka celakalah orang-orang kafir itu karena mereka akan masuk neraka” (Q.S Sad: 27)

Menurut Tafsir Al-Mukhtashar / Markaz Tafsir Riyadh, di bawah pengawasan Syaikh Dr. Shalih bin Abdullah bin Humaid, Imam Masjidil Haram, ayat tersebut menjelaskan bahwa Kami tidak menciptakan langit dan bumi karena iseng. Itu hanyalah dugaan orang-orang kafir. Celakalah orang-orang kafir yang menduga dengan dugaan demikian, mereka mendapatkan azab Neraka pada hari Kiamat bila mereka mati di atas kekufuran dan prasangka buruk kepada Allah. Dari tafsir ini, dijelaskan bahwa pada setiap ciptaan Allah di langit dan bumi, semuanya terdapat hikmah yang bisa kita ambil. Dalam konteks penelitian ini, peneliti dapat mengambil hikmah dari perilaku kerumunan pada hewan dan menerapkan hal tersebut untuk dipelajari dan diimplementasikan. Dengan demikian, penelitian ini menjadi sebuah upaya untuk menggali dan menerapkan pengetahuan yang memiliki akar dalam tafsir ayat tersebut serta penjelasan dari Syaikh Az-Zarnuji dalam kitabnya, sebagai landasan utama yang memberikan legitimasi dan alasan kuat dalam melakukan penelitian ini.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian ini, metode *flocking* berhasil diimplementasikan pada kerumunan NPC pada game “Pengenalan UIN Malang” yang dilengkapi dengan *Finite State Machine* (FSM) untuk mengatur perilaku agen kerumunan pada tiap kondisi pada *state*-nya. Berdasarkan pengujian tingkat *usability* metode *flocking* pada kerumunan NPC menggunakan metode *System Usability Testing* (SUS), didapatkan hasil skor *usability* dengan nilai 68,0555. Berdasarkan hasil skor tersebut, dapat dikatakan bahwa tingkat *usability* dari implementasi metode *flocking* pada kerumunan NPC dinilai cukup baik berdasarkan pada rentang nilai skor *usability* pada gambar 3.9. 1

Berdasarkan hasil skor tingkat *usability* , hasil uji perbandingan waktu tempuh dan hasil uji kemampuan kerumunan NPC dalam menghadapi hambatan statis, dapat disimpulkan bahwa walaupun kerumunan NPC yang menggunakan metode *flocking* memiliki waktu tempuh yang lebih lama, namun kerumunan NPC yang menggunakan metode *flocking* dapat bergerak secara berkerumunan, mempertahankan pola kerumunannya sekaligus dapat melakukan penghindaran terhadap hambatan statis yang membuat kerumunan NPC tersebut terlihat dapat mensimulasikan pola perilaku kerumunan yang sebenarnya .

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diterima untuk mengembangkan fitur pada yang diteliti pada penelitian ini menjadi lebih baik lagi:

- a. Pada hasil implementasi, seringkali terjadi *glitch* animasi berjalan pada agen kerumunan NPC dikarenakan animasi yang dijalankan bergantung pada nilai gerak yang dipengaruhi oleh kalkulasi gerak *flocking* pada NPC yang nilainya sewaktu-waktu dapat berubah dalam waktu singkat. Penambahan fitur *procedural animation* untuk mengatur gerak animasi berjalan pada NPC diharapkan dapat memberikan solusi terhadap masalah ini.
- b. Menambahkan algoritma penghindaran seperti RVO (*Reciprocal velocity obstacle*) khusus untuk penghindaran terhadap objek selain agen satu kerumunan pada NPC yang dapat meng-*override* metode *flocking* ketika akan terjadi tabrakan dapat menjadi solusi untuk NPC yang tidak dapat menghindari objek yang dapat bergerak cepat seperti player.

DAFTAR PUSTAKA

- Agustina, R., & Chandra, A. (2017). *Analisis Implementasi Game Edukasi “The Hero Diponegoro” Guna Meningkatkan Hasil Belajar Siswa Di Mts. Attaroqie Malang, 1(8)*, 1-84.
- Apriyanto, A. (2018). *Pengaturan Formasi Robot Mobil Berdasarkan Pendekatan Virtual Structure*, (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6), 574-594.
- Beny, B., Yani, H., & Ningrum, G. M. (2019). Evaluasi Usability Situs Web Kemenkumham Kantor Wilayah Jambi dengan Metode Usability Test dan System Usability Scale. *RESEARCH: Journal of Computer, Information System & Technology Management*, 2(1), 30-34.
- Brooke, J. (1996). Sus: a “quick and dirty” usability. *Usability evaluation in industry*, 189(3), 189-194.
- Bourg, D. M., & Seeman, G. (2004). *AI For Game Developers*. O'Reilly Media Inc
- Cui, X., Gao, J., & Potok, T. E. (2006). *A Flocking Based Algorithm For Document Clustering Analysis*, 52(8-9), 505-515.
- Dewi, M., Hariadi, M., & Purnomo, M. H. (2011, November). *Simulating the movement of the crowd in an environment using flocking*, In *2011 2nd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*.
- Djamaludin, H. S. (2016). *Pergerakan NPC menggunakan algoritma boids dan artificial bee colony pada simulasi mengelilingi Ka'bah (thawaf)*, (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).
- Ellington, H., Addinall, E., & Percival, F. (1982). *A handbook of game design*, Kogan Page.
- Fadila, J. N., & Arif, Y. M. (2020). *Implementasi Algoritma RVO sebagai Sistem Kendali Gerombolan NPC pada Permainan Action RPG. MATICS*, 12(1), 87.
- Fikri, R. (2023). *Pergerakan partikel untuk efek aura pada karakter berbasis algoritma boids* (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).

- ISO. (1994). Ergonomic requirements for office work with visual display terminals. Part 11 : Guidance on usability. *ISO No 924111*, 2008 (February 9), 22.
- Karim, S., Murti, D. H., & Kuswardayan, I. (2017). *SIMULASI PERILAKU JAMA'AH SAAT MENGELILINGI KA'BAH MENGGUNAKAN ALGORITMA FLOCKING DAN A**, 12(3), 17-26.
- Kim, S., Guy, S. J., Hillesland, K., Zafar, B., Gutub, A. A. A., & Manocha, D. (2015). *Velocity-based modeling of physical interactions in dense crowds*, *The Visual Computer*.
- Matera, M., Rizzo, F., & Carughi, G. T. (2006). Web usability: Principles and evaluation methods. *Web engineering*, 143-180.
- Maulani, M. (2019). Perilaku kelompok NPC dalam *Game* petualangan Bahasa Arab menggunakan algoritma *boids* (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).
- Musse, S. R., Thalman, D., & Raupp, S. (2013). Crowd Rendering. *Crowd Simulation Second Edition*.
- Nugroho, S. M. S., Wulandari, D. P., Christyowidiasmoro, Hariadi, M., & Purnomo, M. H. (2010). *JAVA Journal of Electrical and Electronics Engineering*.
- Olsen, T., Procci, K., & Bowers, C. (2011). Serious games usability testing: How to ensure proper usability, playability, and effectiveness. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice: First International Conference, DUXU 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part II 1* (pp. 625-634). Springer Berlin Heidelberg.
- Prahmana, A., Samsuryadi, S., & Arsalan, O. (2018). *Penerapan flocking algorithm pada game simulasi berternak hewan*, (Doctoral dissertation, Sriwijaya University).
- Reynolds, C. W. (1987). *Flocks, Herds, and Schools : A Distributed Behavioral Model*, 21(4), 25-34.
- Romlah, T. (2006). *Teori dan Teknik Bimbingan Kelompok*, Malang: UNM.
- Santoso, E., Budhi, G. S., & Intan, R. (2017). *Pembuatan Game dengan Menerapkan Metode Decision Tree: UCB1, untuk Menentukan Pemilihan Strategy dalam AI. Jurnal Infra*, 5(1).
- Sarmady, S., Haron, F., & Thalib, A. Z. (2011). *A cellular automata model for circular movements of pedestrians during Tawaf*, *Simulation Modelling Practice and Theory*, 19(3).

- Siagian, P. (1987). *Penelitian Operasional: Teori dan Praktek*, Penerbit Universitas Indonesia (UI Press).
- Siswanto, E., & Suni, F. A. (2021). *Aksi Penyerangan Non-Player Character (Npc) Menggunakan Metode Naïve Bayes Pada Shooter Game*, *urnal Teknologi Informasi dan Ilmu Komputer*, 8(6).
- Su, H., Wang, X., & Lin, Z. (2009). *Flocking of Multi-Agents With a Virtual Leader*, 54, 293-307.
- Warpfelt, H. (2016). *The Non-Player Character*, Sockholm : Department of Computer and Systems Sciences, Stockholm University.