Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.1148377
Vol. 11, No. 4, Agustus 2024, hlm. 913-928 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

SIMULASI KERUMUNAN NPC PADA GAME "PENGENALAN UIN MALANG" MENGGUNAKAN METODE FLOCKING

Dicky Arya Pratama[*1], Yunifa Mitachul Arif[2], Fresy Nugroho[3]

[1,2,3]Universitas Islam Negeri Maulana Malik Ibrahim, Malang
Email: [1]dickyarya170101@gmail.com, [2]yunif4@ti.uin-malang.ac.id, [3]fresysss@gmail.com
[*]Penulis Korespondensi

(Naskah masuk: 27 Desember 2023, diterima untuk diterbitkan: 15 Agustus 2024)

Abstrak

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.

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

SIMULATION OF NPC CROWDS IN THE GAME "INTRODUCTION TO UIN MALANG" USING THE FLOCKING METHOD

Abstract

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.

Keywords: *simulation, crowd, NPC. games, education*

1. PENDAHULUAN

Simulasi merupakan metodologi untuk melaksanakan percobaan menggunakan model dari sistem nyata (Siagian, 1987). Game simulasi, menurut Adams (dalam Tatiek Romlah, 2006), adalah permainan yang merefleksikan situasi kehidupan nyata. Dalam konteks game "Pengenalan UIN Malang," diperlukan sekelompok NPC (Non-Player Character) yang bergerak berkerumun untuk mencerminkan aktivitas sehari-hari di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

NPC adalah karakter dalam game yang dikendalikan oleh komputer dan sering digunakan sebagai pemandu, lawan main, atau pemberi misi (Siswanto & Suni, 2021). Namun, dalam simulasi kerumunan, NPC yang bergerak secara individu sering kali menghasilkan pola kerumunan yang tidak teratur dan tidak mencerminkan situasi nyata. Oleh karena itu, diperlukan perilaku yang memungkinkan NPC bergerak dalam kerumunan secara alami.

Beberapa metode telah diterapkan untuk perilaku kerumunan NPC. Misalnya, Fadila & Arif (2020) menggunakan algoritma RVO untuk menghindari objek, sementara Kim et al. (2015) menggunakan model berbasis kecepatan dan FSM untuk menghindari tabrakan. Fikri et al. (2023) menggunakan algoritma boids untuk menghindari tabrakan pada efek partikel dalam game, dan Maulani et al. (2019) juga menggunakan algoritma boids untuk perilaku kerumunan NPC.

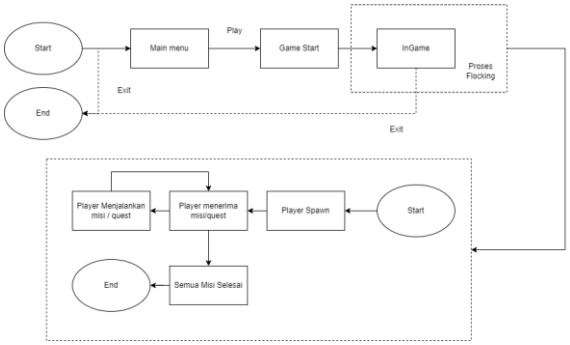
Metode *flocking*, yang dikembangkan oleh Reynold (1999), terinspirasi dari perilaku kelompok burung dan memiliki tiga perilaku dasar: kohesi, penyelarasan, dan penghindaran. Metode ini memungkinkan NPC untuk bergerak secara bersama-sama dan menghindari tabrakan dalam kerumunan (Musse et al., 2013). Metode *flocking* telah digunakan dalam berbagai penelitian, termasuk simulasi arus orang di mall (Dewi et al., 2011) dan simulasi evakuasi saat kebakaran (Nugroho et al., 2010).

Penelitian ini menerapkan metode flocking untuk mensimulasikan kerumunan NPC dalam game "Pengenalan UIN Malang," dengan tujuan menghasilkan NPC yang bergerak secara alami dan sesuai dengan situasi yang ada di lingkungan Universitas Islam Negeri Maulana Malik Ibrahim Malang.

2. METODE PENELITIAN

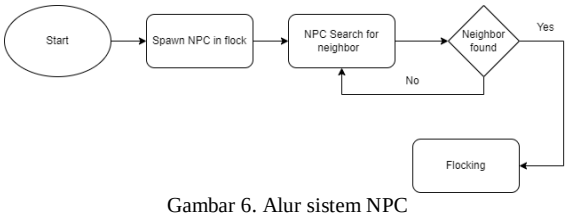
2.1 Desain Sistem

Desain sistem adalah rencana yang menggambarkan cara elemen-elemen terkait disusun menjadi satu kesatuan terstruktur. Ini berfungsi sebagai acuan bagi peneliti dalam menyelesaikan penelitiannya, memastikan semua elemen terhubung dan berfungsi dengan baik. Desain sistem bisa berupa sketsa atau ilustrasi yang menunjukkan penyusunan elemen-elemen tersebut.



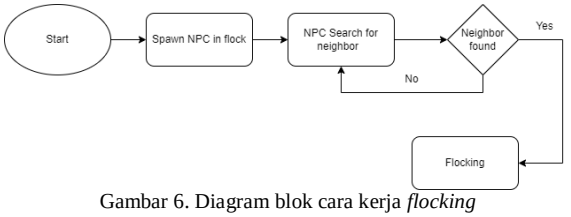
Gambar 5. *Design System Game* "Pengenalan UIN Malang"

Berdasarkan Gambar 5, perancangan sistem game dimulai dari menu utama, di mana pemain dapat memulai permainan dan masuk ke proses In-Game. Karakter pemain muncul di lokasi terakhir bermain (atau di gerbang depan untuk permainan pertama) dan menerima misi yang harus diselesaikan. Permainan dianggap selesai setelah semua misi diselesaikan, meskipun pemain dapat keluar kapan saja. Dalam proses *In-Game*, karakter *non-playable* (NPC) menggunakan metode *flocking*, dimulai dengan NPC yang muncul dalam kelompok dan melakukan pencarian kondisi tetangga.



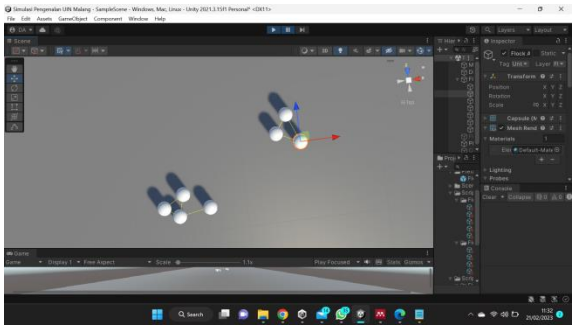
Gambar 6. Alur sistem NPC

Gambar 6 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*.



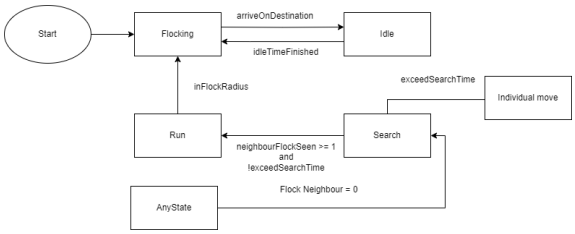
Gambar 6. Diagram blok cara kerja *flocking*

Gambar 6 adalah diagram blok yang menggambarkan perilaku *flocking*. Dalam implementasinya, algoritma ini diterapkan pada satu kelompok NPC yang berusaha menyelaraskan kecepatan dan arah gerakan mereka dengan tetangga. Tiga aspek utama dalam *flocking* adalah kohesi, penyelarasan, dan penghindaran. NPC juga dapat bergerak secara mandiri jika tidak memiliki tetangga.



Gambar 7. Penerapan *flocking*

Gambar 7 menunjukkan hasil implementasi algoritma *flocking* dalam *game*. Setiap NPC mencari tetangga terdekat dan bergerak bersama kelompoknya *(flock)*. Jika tidak menemukan tetangga, NPC akan mencari dan mendekati kelompoknya. Untuk mengoptimalkan proses ini, peneliti menggunakan FSM *(Finite-State Machine)* untuk mengatur perilaku NPC.



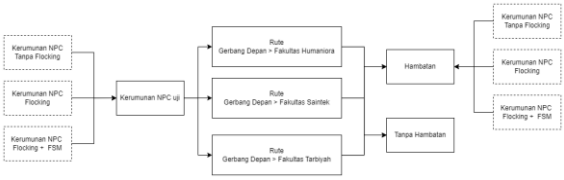
Gambar 8. FSM NPC

Gambar 8 menggambarkan implementasi *Finite-State Machine* (FSM) pada NPC. Saat permainan dimulai, NPC dipanggil bersama kelompoknya dan terlibat dalam *flocking* jika berada dalam jarak minimal dengan agen lain. Jika tujuan belum tercapai, mereka bergerak bersama. Setelah mencapai tujuan, NPC menjadi *idle*, dan setelah waktu *idle* habis, mereka kembali ke *flocking* menuju tujuan berikutnya. Jika NPC tidak memiliki tetangga, statusnya berubah menjadi pencarian *(search)* untuk menemukan kelompoknya dan bergabung kembali dalam *flocking* atau bergerak sendiri, mengoptimalkan perilaku *flocking* dan kinerja agen.

2.2 Rancangan Pengujian Sistem

Penelitian ini mengamati perbandingan waktu tempuh kerumunan NPC dengan dan tanpa metode *flocking*. Selain itu, diuji kemampuan kerumunan dalam menghadapi hambatan statis serta dinilai tingkat *usability* metode flocking pada kerumunan NPC menggunakan *System Usability Scale* (SUS).

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

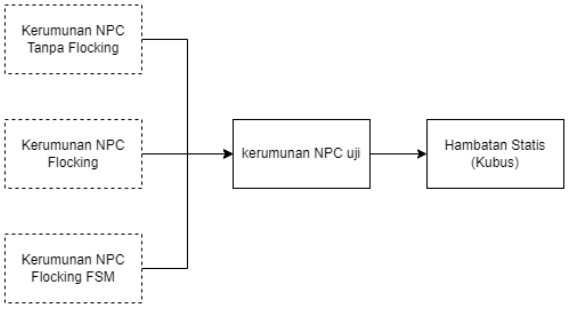


Gambar 9. Skema pengujian waktu tempuh kerumunan NPC

Gambar 9 menggambarkan skema pengujian waktu tempuh kerumunan NPC dengan jumlah agen 2, 5, 10, 15, 20, dan 25 melalui berbagai iterasi. Pengujian dilakukan dengan metode *flocking*, *flocking* dengan *Finite State Machine*, dan tanpa metode, melalui tiga rute berbeda, baik dengan maupun tanpa hambatan berupa kerumunan NPC lain. Variabel seperti kecepatan, radius tetangga, dan jarak penghindaran dijaga konstan. Tujuan penelitian ini adalah membandingkan pengaruh metode dan hambatan terhadap waktu tempuh dan koordinasi kerumunan NPC dalam simulasi *game*.

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

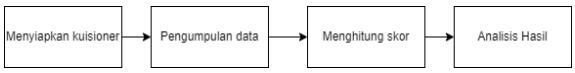
Gambar 10 menunjukkan skema pengujian untuk membandingkan kemampuan kerumunan NPC dalam menghadapi hambatan statis. Pengujian melibatkan tiga kelompok NPC: tanpa metode, menggunakan metode *flocking*, dan menggunakan metode *flocking* dengan *Finite State Machine* (FSM). Jumlah agen yang diuji adalah 2, 5, 10, 15, 20, dan 25 dalam setiap iterasi. Hambatan statis berupa kubus ditempatkan di rute lurus untuk melihat bagaimana setiap kelompok NPC memecah formasi untuk menghindari hambatan.



Gambar 10. Skema pengujian kemampuan kerumunan NPC menghadapi hambatan statis

2.2.3 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.



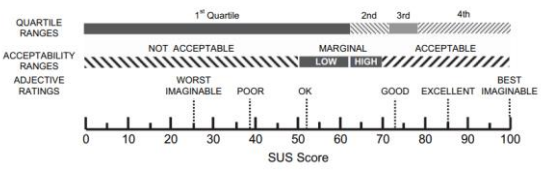
Gambar 11. Alur Pengujian *System Usability Scale*

Gambar 11 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 1. Pertanyaan kuesioner System Usability Scale (SUS)

	Pertanyaan
Q1	Saya merasa mudah untuk memahami cara kerja algoritma flocking
Q2	Saya merasa kerumunan NPC tidak secara efektif menggambarkan suasana keramaian seperti dalam kehidupan nyata
Q3	Saya merasa perilaku kerumunan NPC dalam game membuat saya merasa lebih terlibat dalam pengalaman bermain
Q4	Saya merasa metode flocking tidak dapat meningkatkan interaksi saya dengan karakter NPC dalam game
Q5	Saya merasa perilaku kerumunan NPC dalam game memiliki kegunaan yang signifikan
Q6	Saya merasa pergerakan kerumunan NPC dalam game ini tidak realistis
Q7	Saya ingin melihat lebih banyak penggunaan metode flocking dalam mengatur kerumunan NPC pada game lain
Q8	Saya merasa bahwa perilaku kerumunan NPC dalam game tidak sesuai dengan ekspektasi saya
Q9	Saya merasa pergerakan NPC dalam kerumunan menambah keseruan permainan
Q10	Saya tidak akan merekomendasikan game ini kepada teman-teman saya berdasarkan pengalaman bermain saya dengan kerumunan NPC dalam game ini

Setelah skor dihitung, nilai akhir SUS didapatkan dengan menjumlahkan semua skor dan mengalikan dengan faktor 2,5, yang mengubah rentang skor dari 0-40 menjadi 0-100, lebih representatif untuk menilai *usability*. Menurut (Bangor et.al, 2008), rentang nilai skor SUS diilustrasikan sebagai berikut.



Gambar 12. Penerimaan nilai skor SUS keseluruhan

Berdasarkan Gambar 12, hasil uji menggunakan *System Usability Scale* (SUS) dibagi menjadi empat kuartil. Skor dalam rentang 70 hingga 100 dianggap baik, dengan skor di atas 70 termasuk dalam kuartil ketiga, menandakan produk memiliki tingkat

usability yang baik. Produk dengan skor di atas 80 berada dalam kuartil keempat dan dianggap sangat unggul dalam *usability*.

Produk dengan skor 60 hingga 70 (kuartil kedua) memiliki tingkat *usability* yang cukup baik, tetapi masih memerlukan perbaikan. Produk dengan skor di bawah 60 (kuartil pertama) dianggap tidak dapat diterima dan perlu ditingkatkan *usability*-nya.

Tahapan akhir pengujian ini adalah analisis hasil untuk menilai tingkat *usability* metode *flocking* pada kerumunan NPC berdasarkan skor yang diperoleh.

3. KAJIAN PUSTAKA

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

3.1 Penelitian Terkait

Dalam penelitian "Pengaturan Formasi Robot Mobil Berdasarkan Pendekatan *Virtual Structure* Dengan Penghindaran Rintangan" (Apriyanto, 2018), peneliti mengembangkan sistem pengaturan formasi robot menggunakan model *flocking* dengan metode *artificial potential field* untuk menghindari tabrakan antar robot. Hasilnya menunjukkan bahwa metode ini efektif untuk penghindaran tabrakan, namun tidak dibahas pengaruhnya terhadap objek yang bergerak dinamis.

Pada penelitian "Penerapan *Flocking Algorithm* pada Game Simulasi Beternak Hewan" (Prahmana et al., 2018), algoritma *flocking* digunakan untuk mengatur kerumunan kuda dengan konsep *leader-follower*. Meskipun dijelaskan dengan jelas bagaimana metode *flocking* bekerja, penelitian ini hanya fokus pada satu jenis objek, yaitu kuda.

Penelitian "Simulasi Perilaku Jama'ah Saat Mengelilingi Ka'bah Menggunakan Metode *Flocking* dan A*" (Karim et al., 2017) mengeksplorasi penggunaan metode *flocking* dalam simulasi tawaf. Hasilnya menunjukkan bahwa metode *flocking* efektif untuk menghindari tabrakan dan dapat digunakan untuk mensimulasikan kerumunan dalam skala besar dengan konsep pemimpin untuk memandu jamaah.

3.2 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.

3.3 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 menggerakan 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.

3.4 NPC

NPC, atau *Non-Player Character*, adalah karakter dalam permainan video yang dikendalikan oleh sistem permainan, bukan oleh pemain (Warpfelt, 2016). Istilah ini sudah ada sebelum era permainan digital dan kini NPC digunakan dalam berbagai *genre* seperti RPG, petualangan, dan aksi untuk membantu pemain mencapai tujuan. NPC dapat berperan sebagai penolong, pemberi informasi, musuh, atau sekadar latar belakang, dengan perilaku yang dikendalikan oleh AI.

Dalam permainan modern, NPC semakin canggih, mampu berinteraksi dan membentuk hubungan dengan pemain, serta mempengaruhi alur cerita permainan. NPC yang lebih realistis dan interaktif meningkatkan pengalaman bermain yang lebih mendalam.

Pada penelitian game ini, NPC adalah agen dalam kerumunan (flock) yang memiliki tujuan tertentu dan bergerak bersama kelompoknya. Untuk mencegah tabrakan, NPC menggunakan metode flocking, yang mengatur perilaku mereka dalam kelompok.

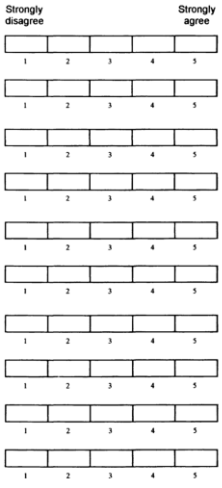
3.5 Usability Testing

Menurut Matera et al. (2006), *Usability* berasal dari kata *"usable"*, yang berarti dapat digunakan dengan baik. Perangkat dianggap efisien jika tingkat kegagalan dalam penggunaannya rendah dan memberikan hasil positif bagi penggunanya. Berdasarkan ISO 9241-11 (ISO, 1994), *usability* diukur melalui tiga aspek: *Effectiveness* (kemampuan pengguna mencapai tujuan), *Efficiency* (sumber daya yang digunakan untuk mencapai tujuan), dan *Satisfaction* (kepuasan pengguna). *Usability testing*, menurut Beny et al. (2019), adalah pengujian untuk mengidentifikasi masalah dalam penggunaan suatu sistem atau produk.

Dalam penelitian ini, peneliti menerapkan *usability testing* untuk mengukur tingkat *usability* dari metode *flocking* yang digunakan pada kerumunan NPC. Diharapkan hasilnya dapat menunjukkan bagaimana *Effectiveness*, *Efficiency*, dan *Satisfaction* dari metode *flocking* ini di mata pemain.

3.6 System Usability Scale (SUS)

System Usability Scale (SUS), dikembangkan oleh John Brooke pada 1986, memungkinkan praktisi *usability* untuk dengan cepat dan mudah menilai *usability* sebuah produk (Bangor et al., 2008). SUS memiliki banyak kelebihan, salah satunya adalah sifatnya yang tidak memihak pada teknologi tertentu, membuatnya fleksibel untuk menilai teknologi dalam berbagai konteks (Bangor et al., 2008). SUS menggunakan metode survei sederhana dengan skor 0-100, mudah dimengerti, tidak memerlukan biaya, dan dapat diterapkan pada sampel kecil (Brooke, 1996). Selain itu, metode ini sangat fleksibel dan agnostik terhadap teknologi, sehingga cocok untuk berbagai evaluasi, termasuk fitur dalam *game* (Bangor et al., 2008). Olsen et al. (2011) juga memanfaatkan SUS untuk mengukur *usability*, *playability*, dan *effectiveness* dalam sebuah *serious game*.



Gambar 1. *System Usability Scale* (SUS)

Gambar 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

$$Sus\ Score = Total\ Score\ 2.5 \quad (1)$$

3.7 Flock (Kerumunan)

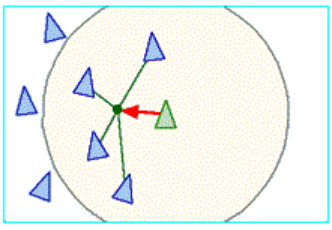
Flocking adalah fenomena di mana agen-agen bergerak secara terkoordinasi menggunakan informasi lingkungan yang terbatas dan aturan sederhana (Su et al., 2009). Dalam pengembangan game, teknik ini digunakan untuk menciptakan perilaku kelompok objek atau agen yang tampak organik dan meniru gerakan alami, seperti burung atau ikan yang bergerak bersama. Terinspirasi oleh perilaku burung dan ikan, metode *flocking* yang dikembangkan dari konsep *steering behavior* oleh Craig Reynolds memungkinkan objek dalam game, seperti NPC atau kendaraan, untuk bergerak dengan koordinasi yang cermat, menghindari tabrakan, dan mengikuti gerakan 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



Gambar 2. Perilaku Cohesion

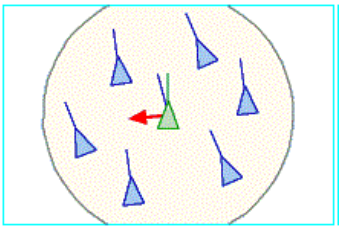
Gambar 2 merupakan ilustrasi dari perilaku *cohesion* dalam algoritma *flocking*. Perilaku *cohesion* memungkinkan agen-agennya untuk mengarahkan diri mereka ke titik tengah kelompoknya, yang membantu menjaga kelompok agar tetap bersatu. Perilaku ini memungkinkan agen untuk bergerak menuju titik tengah kelompoknya. *Cohesion* juga mengatur agar agen dapat mendekat satu sama lain ketika terlalu jauh.

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

$$d(P_x, P_b) \leq d_1 \cap \quad d(P_x, P_b) \geq d_2 \Rightarrow V_{cr} = \sum_n^x \quad (P_x - P_b) \quad (2)$$

Dalam persamaan ini, menunjukkan kecepatan yang diatur oleh aturan *cohesion*, yang dapat dilihat pada gambar 2. merepresentasikan jarak antara agen x dan tetangga b, sedangkan dan adalah nilai jarak yang telah ditentukan, adalah perhitungan arah *vector point*.

2. Alignment



Gambar 3. Perilaku Alignment

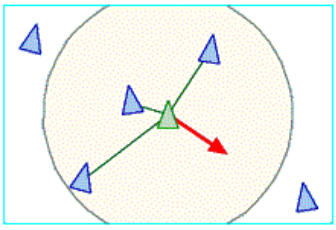
Gambar 3 merupakan ilustrasi dari perilaku alignment dalam algoritma *flocking*. Perilaku ini memungkinkan agen untuk menyesuaikan kecepatannya dengan agen lainnya, sehingga agen dapat mencocokkan kecepatannya dengan tetangganya.

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

$$d(P_x, P_b) \leq d_1 \cap \quad d(P_x, P_b) \geq d_2 \Rightarrow V_{ar} = \frac{1}{n} \sum_n^x \quad V_x \quad (3)$$

Dalam hal ini, mengacu pada kecepatan yang diatur oleh perilaku *alignment*, sebagaimana terlihat pada gambar 3, sedangkan merupakan jarak antara agen x dan tetangga b, n menunjukkan jumlah total tetangga, menunjukkan kecepatan agen x, dan adalah jarak yang sudah ditentukan.

3. Separation



Gambar 4. Perilaku Separation

Gambar 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)

$$d(P_x, P_b) \leq d_2 \Rightarrow V_{sr} = \sum_n^x \frac{(v_x + v_b)}{d(P_x, P_b)} \quad (4)$$

Perilaku separation memberikan kecepatan yang ditentukan pada gambar 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

$$d(P_x, P_b) \leq d_1 \cap \quad d(P_x, P_b) \geq d_2 \quad (5)$$

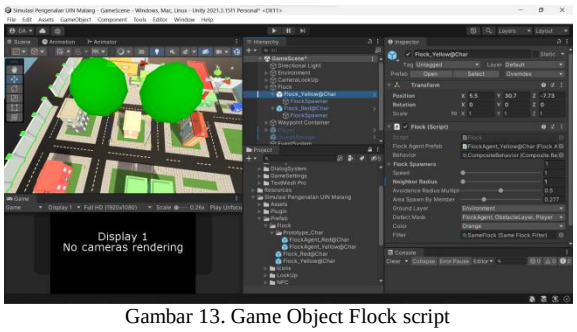
Persamaan tersebut merupakan persamaan kondisi yang menghasilkan kondisi benar ketika jarak agen x dengan tetangga b kurang dari jarak maksimal (d1) dan jarak agen x dengan tetangga b lebih dari jarak minimal (d2).

Penggunaan metode *flocking* dalam pengembangan *game* meningkatkan keberagaman perilaku agen atau objek. Dengan mengatur parameter yang berbeda, metode ini menghasilkan variasi perilaku yang membuat pengalaman bermain lebih menarik dan dinamis. Selain itu, *flocking* juga menciptakan perilaku kelompok yang lebih natural dan *realistis*, meningkatkan pengalaman bermain game.

4. HASIL DAN PEMBAHASAN

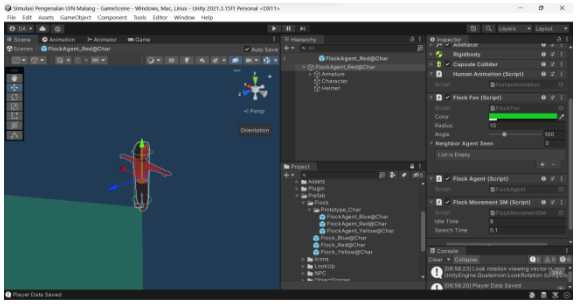
4.1 Implementasi Sistem

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.



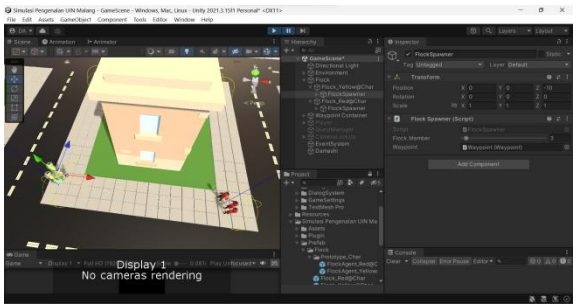
Gambar 13. Game Object Flock script

Gambar 13 menunjukkan *Game Object* 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 14. *Game Object FlockAgent script*

Gambar 14 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*.

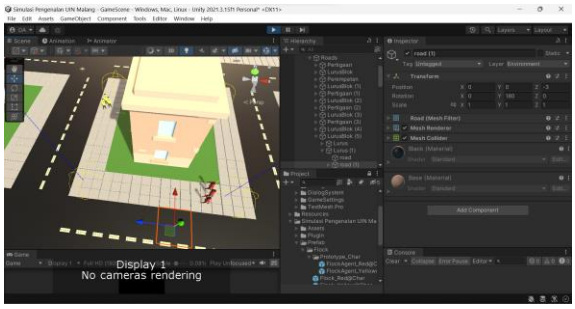


Gambar 15. *Flocking* NPC *Spawn*

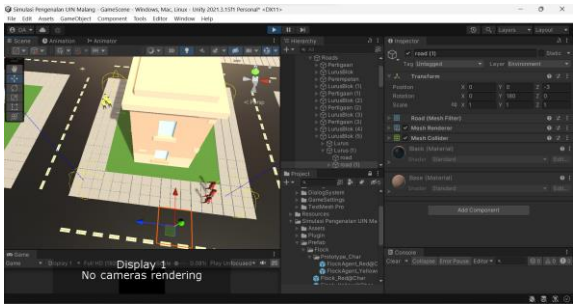
Ketika permainan dimulai, kerumunan NPC akan muncul di lokasi yang telah ditentukan dan bergerak bersama-sama dengan metode *flocking*. Mereka akan mengikuti jalur yang telah ditetapkan oleh *script Waypoint*.

Gambar 16 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.

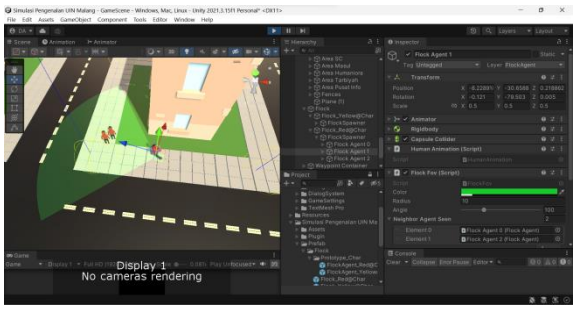


Gambar 16. *Flocking Movement* FSM



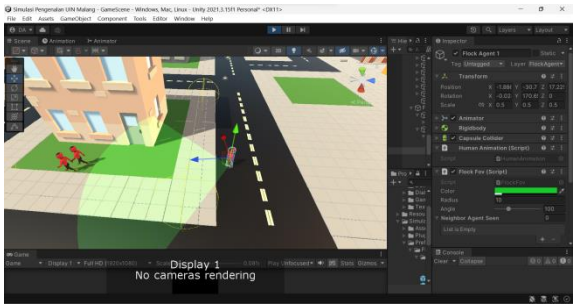
Gambar 17. *Flocking Movement FSM Idle*

Gambar 17 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 18. *Flocking Movement* FSM *Search and Run*

Gambar 18 menunjukkan agen flocking dalam *state Run*. Jika agen tidak mendeteksi tetangga di dekatnya, *Finite State Machine* akan mengalihkan agen ke *state Search* untuk mencari tetangga dalam jangkauan *Field of View* (FOV) sesuai script *FlockFOV*. Ketika agen mendeteksi tetangga, *state* akan kembali ke *Run*, di mana agen berlari menuju tetangga tersebut, dan akan kembali ke *state Flock* saat mencapai jarak *flocking* yang sesuai.



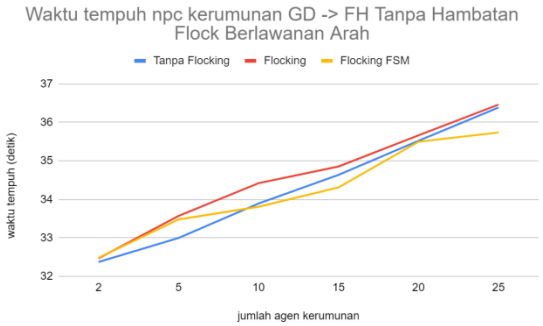
Gambar 19. *Flocking Movement* FSM *Walk*

Gambar 19 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

4.2.1 Perbandingan waktu tempuh pada penggunaan metode flocking

4.2.1.1 Perbandingan waktu tempuh pada penggunaan metode flocking



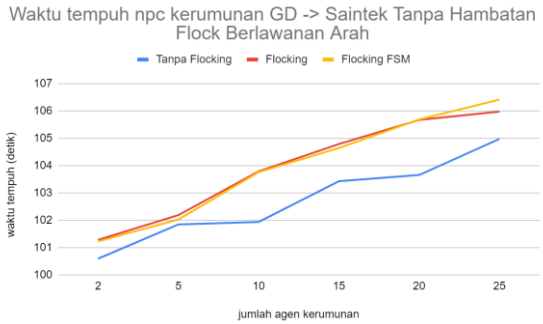
Gambar 20. Hasil Waktu tempuh kerumunan NPC GD > FH Tanpa Hambatan *Flock* Berlawanan Arah

Gambar 20 menunjukkan perbandingan waktu tempuh kerumunan NPC dari gerbang awal ke fakultas humaniora. Garis biru mewakili waktu tempuh tanpa metode *flocking*, garis merah untuk metode *flocking*, dan garis kuning untuk *flocking* dengan *Finite State Machine* (FSM). Berikut adalah analisis perbandingan waktu tempuh untuk setiap metode berdasarkan Gambar 16.

Berdasarkan data pada Gambar 16, didapatkan hasil pengamatan sebagai berikut:

- Jumlah agen dalam kerumunan NPC mempengaruhi waktu tempuh.
- Kerumunan NPC dengan metode flocking dan Finite State Machine (FSM) memiliki waktu tempuh lebih singkat untuk jumlah agen di atas 10.

- Kerumunan NPC yang menggunakan metode flocking memiliki waktu tempuh terlama, meski perbedaannya tidak signifikan.

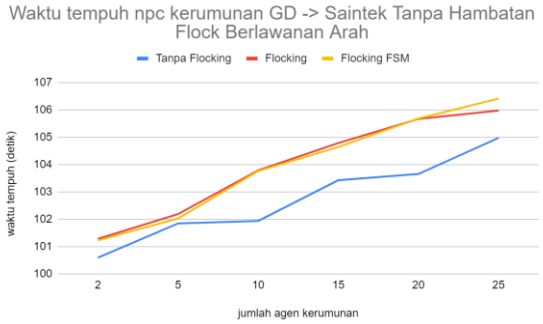


Gambar 21. Hasil Waktu tempuh kerumunan NPC GD > Saintek Tanpa Hambatan *Flock* Berlawanan Arah

Peneliti melakukan observasi tambahan untuk membandingkan waktu tempuh kerumunan NPC dari gerbang depan ke fakultas Saintek dalam *game*, menggunakan protokol pengujian yang sama. Garis biru menunjukkan waktu tempuh tanpa metode *flocking*, garis merah untuk metode *flocking*, dan garis kuning untuk *flocking* dengan *Finite State Machine* (FSM).

Berdasarkan pada Gambar 21, didapatkan hasil pengamatan sebagai berikut:

- NPC tanpa metode *flocking* memiliki waktu tempuh paling singkat, berbeda signifikan dari yang menggunakan metode *flocking* atau FSM.
- NPC dengan metode *flocking* dan FSM sedikit lebih cepat, tetapi tidak signifikan dibandingkan hanya menggunakan *flocking*.



Gambar 22. Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah Tanpa Hambatan Flock Berlawanan Arah

Pengujian dilakukan pada kerumunan NPC dari gerbang ke Fakultas Tarbiyah. Garis biru menunjukkan waktu tempuh tanpa metode *flocking*, merah dengan *flocking*, dan kuning dengan *flocking* dan FSM.

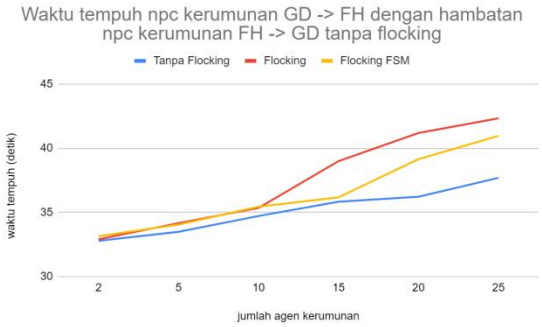
Berdasarkan gambar 22, didapatkan hasil pengamatan sebagai berikut:

- Tidak ada perbedaan signifikan antara kerumunan NPC yang menerapkan metode *flocking* dan yang menggunakan metode *flocking* dengan *Finite State Machine* (FSM),
- 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:

- Pada rute yang sama, semakin banyak agen dalam kerumunan NPC, semakin lama waktu tempuhnya.
- Tanpa hambatan, kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh yang jauh lebih singkat dibandingkan dengan yang menggunakan metode *flocking* atau *flocking* dengan *Finite State Machine* (FSM). Perbedaan ini semakin besar pada rute yang lebih jauh.

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



Gambar 23. Hasil Waktu tempuh kerumunan NPC GD > FH Tanpa Hambatan Flock Berlawanan Arah

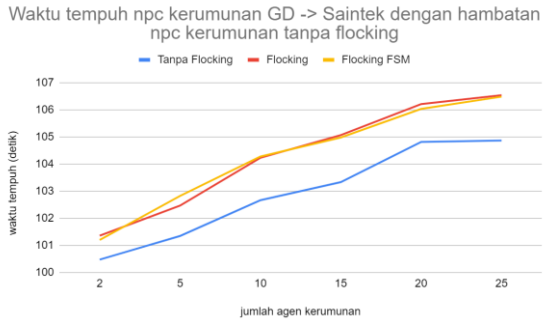
Gambar 23 menunjukkan perbandingan waktu tempuh kerumunan NPC yang bergerak dari gerbang awal ke Fakultas Humaniora. Garis biru mewakili waktu tempuh tanpa metode *flocking*, garis merah untuk metode *flocking*, dan garis kuning untuk flocking dengan *Finite State Machine* (FSM). Dalam pengujian ini, kerumunan NPC menghadapi hambatan dari kerumunan NPC tanpa *flocking* dengan jumlah agen yang sama.

Berdasarkan gambar 23, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh paling singkat.
- Kerumunan NPC dengan *flocking* dan *Finite State Machine* (FSM) lebih cepat pada jumlah agen di atas 15 dibandingkan dengan hanya metode *flocking*.
- Kerumunan NPC dengan metode *flocking* lebih cepat dibandingkan yang menggunakan FSM untuk jumlah agen 15 ke bawah.

Gambar 24 memperlihatkan perbandingan waktu tempuh kerumunan NPC yang menghadapi hambatan, dengan jumlah agen yang sama. Posisi awal adalah gerbang depan dan posisi akhir adalah Fakultas Saintek. Garis biru menunjukkan waktu tempuh NPC tanpa metode flocking, garis merah

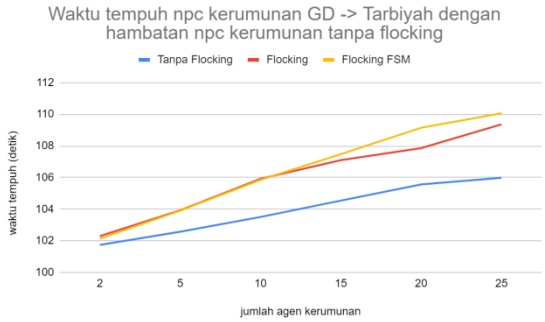
untuk NPC menggunakan metode flocking, dan garis kuning untuk NPC menggunakan flocking dengan Finite State Machine (FSM).



Gambar 24. Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan Flock Berlawanan Arah tanpa metode flocking

Berdasarkan data pada gambar 24, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh tercepat.
- Waktu tempuh kerumunan NPC dengan metode *flocking* dan yang menggunakan *Finite State Machine* (FSM) tidak berbeda signifikan.
- NPC dengan metode flocking dan FSM lebih cepat pada jumlah agen 20 ke atas dibandingkan yang hanya menggunakan *flocking*.
- NPC dengan metode *flocking* lebih cepat dibandingkan yang menggunakan FSM pada jumlah agen 15 ke bawah.



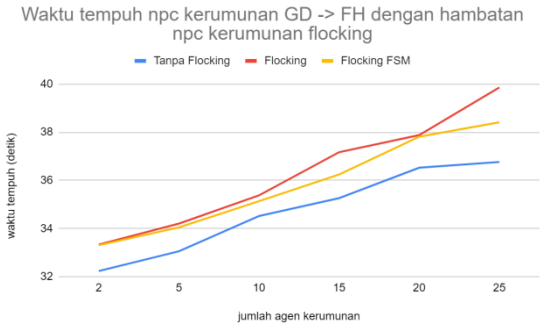
Gambar 25. Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan Hambatan Flock Berlawanan Arah tanpa metode flocking

Gambar 25 menunjukkan perbandingan waktu tempuh kerumunan NPC dari gerbang depan menuju fakultas Tarbiyah. Garis biru mewakili waktu tempuh NPC tanpa metode *flocking*, garis merah untuk NPC menggunakan metode *flocking*, dan garis kuning untuk NPC menggunakan metode *flocking* dengan *Finite State Machine* (FSM).

Berdasarkan pada gambar 25, 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*



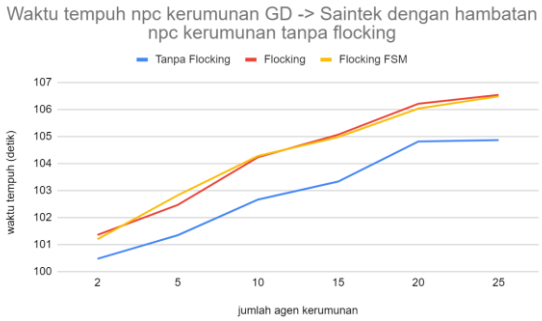
Gambar 22. Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan Flock Berlawanan Arah menggunakan metode *flocking*

Gambar 22 menunjukkan perbandingan waktu tempuh kerumunan NPC dari gerbang awal menuju fakultas Humaniora dengan hambatan. Garis biru mewakili waktu tempuh NPC tanpa metode *flocking*, garis merah untuk NPC menggunakan metode *flocking*, dan garis kuning untuk NPC dengan metode *flocking* dan *Finite State Machine* (FSM).

Berdasarkan gambar 22, 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).

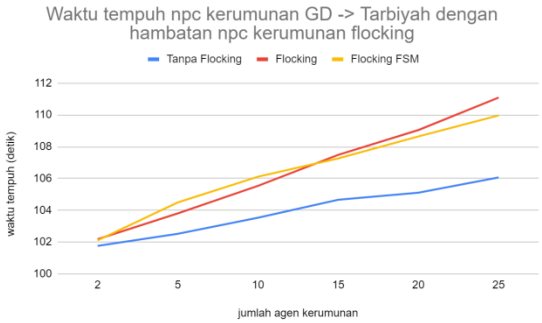
Gambar 23 menunjukkan perbandingan waktu tempuh kerumunan NPC dengan hambatan menggunakan metode *flocking*. Posisi awal kerumunan adalah gerbang depan, sedangkan posisi akhir adalah fakultas Saintek. Garis biru mewakili waktu tempuh NPC tanpa metode *flocking*, garis merah untuk NPC menggunakan metode *flocking*, dan garis kuning untuk NPC dengan metode *flocking* dan *Finite State Machine* (FSM).



Gambar 23. Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan Flock Berlawanan Arah menggunakan metode *flocking*

Berdasarkan gambar 23, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh tercepat.
- Kerumunan NPC dengan metode *flocking* dan *Finite State Machine* (FSM) memiliki waktu tempuh sedikit lebih singkat pada jumlah agen 15 ke atas dibandingkan yang hanya menggunakan metode *flocking*.
- Kerumunan NPC dengan metode *flocking* memiliki waktu tempuh lebih singkat pada jumlah agen 10 ke atas dibandingkan dengan yang menggunakan FSM, tetapi sebaliknya pada jumlah agen 10 ke bawah.



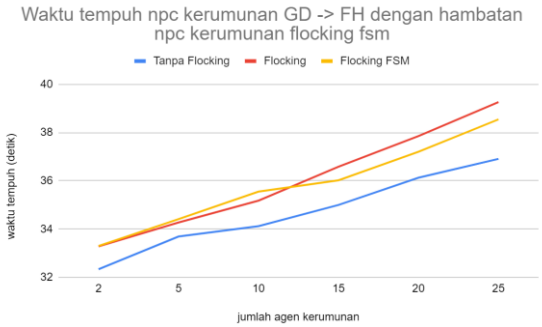
Gambar 24. Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan Hambatan Flock Berlawanan Arah menggunakan metode flocking

Gambar 24 menunjukkan perbandingan waktu tempuh kerumunan NPC dengan hambatan yang menggunakan metode *flocking*. Pengujian dilakukan dari posisi awal di gerbang depan hingga posisi akhir di fakultas tarbiyah. Garis biru menunjukkan waktu tempuh untuk NPC tanpa metode *flocking*, garis merah untuk NPC dengan metode *flocking*, dan garis kuning untuk NPC dengan metode *flocking* dan *Finite State Machine* (FSM).

Berdasarkan gambar 24, didapatkan hasil pengamatan sebagai berikut:

- Kerumunan NPC tanpa metode flocking memiliki waktu tempuh tercepat.
- Kerumunan NPC dengan metode flocking dan Finite State Machine (FSM) sedikit lebih cepat daripada yang hanya menggunakan metode flocking, mulai dari 15 agen ke atas.
- Kerumunan NPC yang menggunakan metode flocking lebih cepat dibandingkan dengan metode flocking dan FSM, pada jumlah agen 10 ke bawah.

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

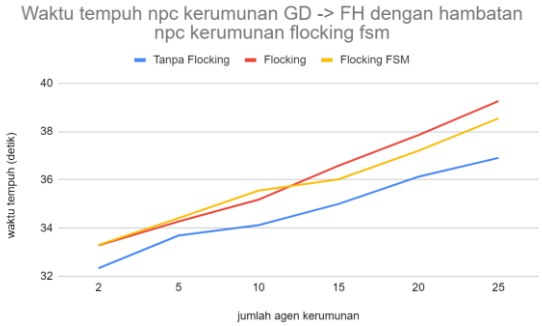


Gambar 25. Hasil Waktu tempuh kerumunan NPC GD > FH dengan Hambatan Flock Berlawanan Arah menggunakan metode flocking FSM

Gambar 25 menunjukkan perbandingan waktu tempuh kerumunan NPC dari gerbang awal menuju fakultas humaniora. Garis biru mewakili waktu tempuh agen tanpa metode *flocking*, garis merah untuk agen yang menggunakan metode *flocking*, dan garis kuning untuk agen dengan metode *flocking* dan *Finite State Machine* (FSM).

Berdasarkan gambar 25, 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).

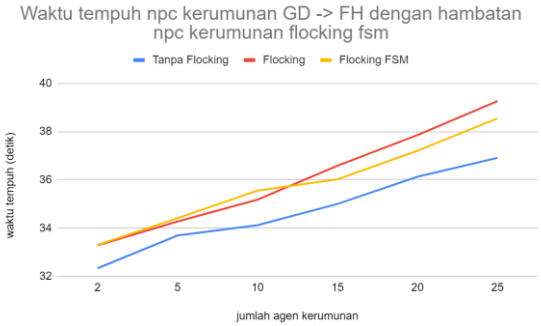


Gambar 26. Hasil Waktu tempuh kerumunan NPC GD > Saintek dengan Hambatan Flock Berlawanan Arah menggunakan metode flocking FSM

Gambar 26 menunjukkan perbandingan waktu tempuh kerumunan NPC yang menghadapi hambatan. Garis biru mewakili waktu tempuh agen tanpa metode *flocking*, garis merah untuk agen yang menggunakan metode *flocking*, dan garis kuning untuk agen dengan metode *flocking* dan *Finite State Machine* (FSM).

Berdasarkan gambar 26, 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 27. Hasil Waktu tempuh kerumunan NPC GD > Tarbiyah dengan Hambatan Flock Berlawanan Arah menggunakan metode flocking FSM

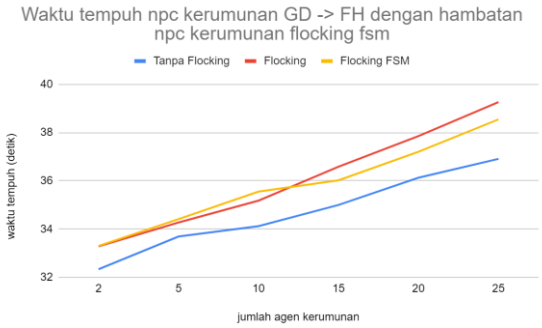
Gambar 27 menunjukkan perbandingan waktu tempuh kerumunan NPC yang bergerak dari gerbang depan ke fakultas tarbiyah. Garis biru mewakili waktu tempuh agen tanpa metode *flocking*, garis merah untuk agen yang menggunakan metode *flocking*, dan garis kuning untuk agen dengan metode *flocking* dan *Finite State Machine* (FSM).

Berdasarkan gambar 27, 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.

  Gambar 28 menunjukkan perbandingan persentase agen dalam kerumunan NPC yang berhasil melewati hambatan statis untuk setiap metode uji dan iterasi. Sumbu x mewakili jumlah agen dalam kerumunan, sementara sumbu y menunjukkan persentase yang berhasil melintasi hambatan. Garis biru mencerminkan agen tanpa metode *flocking*, garis merah untuk agen yang menggunakan metode *flocking*, dan garis kuning untuk agen dengan metode *flocking* dan *Finite State Machine*.

4.2.2 Perbandingan waktu tempuh pada penggunaan metode flocking



Gambar 28. Hasil Persentase agen yang dapa melewati hambatan statis

Hasil ini memberikan gambaran tentang efektivitas masing-masing metode dalam menghadapi hambatan statis:

- Persentase agen kerumunan NPC yang berhasil melewati hambatan statis meningkat seiring bertambahnya jumlah agen.
- Kerumunan NPC tanpa metode *flocking* memiliki persentase terendah dalam melewati hambatan statis dibandingkan dengan kerumunan yang menggunakan metode *flocking* atau *flocking* dengan *Finite State Machine* (FSM).
- Kerumunan NPC dengan metode *flocking* dan FSM menunjukkan persentase tertinggi dalam melewati hambatan statis dibandingkan metode lainnya.

4.2.3 Perbandingan waktu tempuh pada penggunaan metode *flocking*

Tabel 4.13 memuat jawaban responden terhadap kuesioner yang mengevaluasi aspek *usability* metode *flocking* dalam "*Game* Pengenalan UIN Malang," menggunakan skala penilaian 1 hingga 5. Jawaban ini akan dijumlahkan sesuai rumus *System Usability Scale* (SUS) untuk menghitung tingkat *usability* dari metode *flocking* yang diimplementasikan. Tabel ini merupakan elemen penting untuk menilai kualitas dan pengalaman pemain dalam permainan tersebut.

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:

- Kerumunan NPC tanpa metode *flocking* memiliki waktu tempuh lebih singkat dibandingkan dengan kerumunan yang menggunakan *flocking* atau flocking dengan FSM, namun mereka tidak dapat menghindari hambatan dan tidak membentuk pola kerumunan.

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
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

- Waktu tempuh antara kerumunan NPC dengan *flocking* dan *flocking* dengan FSM menunjukkan perbedaan kecil, di mana metode *flocking* dengan FSM sedikit lebih cepat, baik pada pengujian tanpa hambatan maupun dengan hambatan.
- Perbedaan waktu yang signifikan terlihat antara NPC tanpa *flocking* dan yang menggunakan *flocking* atau *flocking* dengan FSM, terutama pada rute dengan jarak jauh.
- Pada rute lurus dengan hambatan, NPC yang menggunakan *flocking* dengan FSM memiliki waktu tempuh lebih singkat dibandingkan dengan yang hanya menggunakan *flocking*. Pada rute berbelok, perbedaan waktu tempuh antara kedua metode 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:

- Kerumunan NPC tanpa metode *flocking* memiliki persentase agen yang mampu melewati hambatan statis paling rendah karena tidak memiliki mekanisme untuk menghindari halangan, sehingga lebih rentan terjebak.
- Sebaliknya, kerumunan NPC dengan metode *flocking* dan *Finite State Machine* (FSM) memiliki persentase tertinggi dalam melewati hambatan statis. Hal ini disebabkan oleh fungsi *"run"* dalam FSM yang memungkinkan NPC meningkatkan kecepatan dan lebih mudah melepaskan diri dari hambatan dibandingkan dengan metode *flocking* saja.

4.3.3 Analisis Data Pengujian Tingkat Usability Metode *Flocking*

Dalam pengujian tingkat usability menggunakan System Usability Scale (SUS), pertanyaan dengan ekspektasi respon positif ditempatkan pada urutan ganjil, sementara yang negatif pada urutan genap. Skor pertanyaan ganjil dihitung dengan mengurangi poin yang dipilih responden dari 5, sedangkan untuk pertanyaan genap, skor dihitung dengan mengurangi poin yang dipilih dari 1. Hasil ini kemudian dikalikan dengan 2.5 untuk mendapatkan skor akhir SUS. Skor ini menjadi acuan dalam menilai tingkat usability metode flocking pada kerumunan NPC dalam game "Pengenalan UIN Malang" (rumus 2.1). Hasil pengujian dari setiap responden menunjukkan nilai usability metode flocking dalam mengelola kerumunan NPC 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.

5. HASIL DAN KESIMPULAN

5.1 Kesimpulan

Penelitian ini berhasil mengimplementasikan metode *flocking* pada kerumunan NPC dalam *game*

"Pengenalan UIN Malang," yang dilengkapi dengan *Finite State Machine* (FSM) untuk mengatur perilaku agen. *Usability testing* dengan *System Usability Scale* (SUS) menghasilkan skor 68,0555, yang menunjukkan tingkat *usability* cukup baik. Meskipun kerumunan NPC dengan metode *flocking* memiliki waktu tempuh lebih lama, mereka mampu bergerak secara berkelompok, mempertahankan pola kerumunan, dan menghindari hambatan statis, sehingga berhasil mensimulasikan perilaku kerumunan yang realistis.

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:

- 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.
- 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. 201). 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.

Halaman ini sengaja dikosongkan.