**Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer**
**Volume 6, Nomor 3, November 2026**
e-ISSN: 2827-7953; p-ISSN: 2827-8135, Hal. 61-74
DOI: https://doi.org/10.55606/juisik.v6i3.2292
*Tersedia:* http://journal.sinov.id/index.php/juisik

# Implementasi *Artificial Intelligence* pada *Game* Simulasi Menggunakan *Unity*

**Fakhriza Sidhqi Wafiq Fauzi[1*], Fajar Delli Wihartiko[2], Agung Prajuhana Putra[3]**
[1,3]Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pakuan, Indonesia
[2]Departemen Ilmu Komputer, Pascasarjana, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pakuan, Indonesia
*Email: fakhriza.065121131@unpak.ac.id [1], fajardelli@unpak.ac.id [2], prajuhana.putra@unpak.ac.id [3]*
*[*]Penulis Korespondensi: fakhriza.065121131@unpak.ac.id*

***Abstract****. This research is motivated by the rapid growth of the game industry in Indonesia and the need for interactive simulation systems based on Artificial Intelligence (AI). This study aims to develop a food court management simulation game by implementing NavMesh based on the A* algorithm for character navigation, as well as Finite State Machine (FSM) and smart reading to regulate the behavior of Non-Player Characters (NPCs). The method used is the Game Development Life Cycle (GDLC), which includes the stages of concept, design, development, and evaluation. The results show that the developed system is capable of producing an intuitive user interface and an interactive 3D environment. In addition, NPCs, such as customers, waiters, and cleaning staff, are able to behave adaptively according to environmental conditions. The testing results indicate that the system operates well structurally and functionally and is responsive across various Android devices. The implementation of AI methods in the developed system supports the creation of NPC interactions that adapt to the conditions of the game environment. Thus, the integration of AI methods can enhance interactivity and realism in food court management simulation games while providing users with a more dynamic gaming experience.*

***Keywords:*** *Artificial Intelligence; Finite State Machine; Game Development Life Cycle; NavMesh; Simulation Game.*

**Abstrak**. Penelitian ini dilatarbelakangi oleh pesatnya perkembangan industri *game* di Indonesia serta kebutuhan akan sistem simulasi yang interaktif berbasis *Artificial Intelligence* (AI). Penelitian ini bertujuan untuk mengembangkan *game* simulasi manajemen pujasera dengan menerapkan NavMesh berbasis algoritma A* untuk navigasi karakter serta *Finite State Machine* (FSM) dan *smart reading* dalam mengatur perilaku *Non-Player Character* (NPC). Metode yang digunakan adalah *Game Development Life Cycle* (GDLC), yang meliputi tahap konsep, desain, pengembangan, hingga evaluasi. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu menghasilkan tampilan antarmuka yang intuitif dan lingkungan 3D yang interaktif. Selain itu, NPC, seperti pelanggan, pelayan, dan petugas kebersihan, mampu berperilaku secara adaptif sesuai dengan kondisi lingkungan. Hasil pengujian menunjukkan bahwa sistem berjalan dengan baik secara struktural dan fungsional serta responsif pada berbagai perangkat Android. Implementasi metode AI dalam sistem yang dikembangkan mampu mendukung terciptanya interaksi NPC yang adaptif terhadap kondisi lingkungan permainan. Dengan demikian, integrasi metode AI dapat meningkatkan interaktivitas dan realisme dalam *game* simulasi manajemen pujasera serta memberikan pengalaman permainan yang lebih dinamis bagi pengguna.

**Kata kunci**: *Artificial Intelligence*; *Finite State Machine*; *Game Development Life Cycle*; *Game simulasi*; NavMesh.

## 1. LATAR BELAKANG

Industri *game* mengalami perkembangan yang sangat pesat seiring dengan kemajuan teknologi informasi dan meningkatnya kebutuhan akan media hiburan interaktif. Secara global, penelitian di bidang *Artificial Intelligence* (AI) dalam *game* terus meningkat signifikan dalam beberapa tahun terakhir, menunjukkan bahwa AI menjadi komponen penting dalam pengembangan *game* modern (Bosque et al., 2025). Di Indonesia sendiri, industri *game* juga menunjukkan tren pertumbuhan yang positif sehingga mendorong pengembangan *game* yang lebih inovatif dan interaktif. Salah satu faktor utama dalam meningkatkan kualitas *game* adalah

*Naskah Masuk: 31 Juli 2026; Revisi: 20 Agustus 2026; Diterima: 23 September 2026; Terbit: 01 November 2026*

kemampuan menghadirkan pengalaman yang realistis, yang dapat dicapai melalui penerapan AI, khususnya dalam pengembangan perilaku *Non-Player Character* (NPC) yang adaptif dan responsif terhadap lingkungan.

Berbagai penelitian terbaru telah mengkaji penerapan AI dalam game, terutama pada aspek *pathfinding* dan perilaku NPC. Penggunaan algoritma A* terbukti efektif dalam menentukan jalur terpendek dan meningkatkan efisiensi navigasi karakter dalam *game* (Arifudin et al., 2025). Selain itu, integrasi *Navigation Mesh* (NavMesh) dengan algoritma A* memungkinkan pergerakan NPC yang lebih realistis dalam lingkungan yang kompleks. Di sisi lain, pendekatan *Finite State Machine* (FSM) banyak digunakan untuk mengatur perilaku NPC secara dinamis berdasarkan kondisi tertentu. Penelitian oleh Amalia et al. (2024) menunjukkan bahwa kombinasi FSM dan *pathfinding* mampu meningkatkan efisiensi rute hingga 25% dan mengurangi konflik navigasi hingga 30% dalam simulasi berbasis Unity (Amalia et al., 2024). Hal ini menunjukkan bahwa AI berbasis FSM dan A* memiliki peran penting dalam meningkatkan kualitas interaksi dalam *game*.

Namun demikian, sebagian besar penelitian masih berfokus pada implementasi satu teknik AI secara terpisah, seperti hanya pada navigasi atau hanya pada perilaku NPC. Padahal, tren penelitian terbaru menunjukkan bahwa integrasi berbagai pendekatan AI menjadi tantangan sekaligus peluang dalam pengembangan *game* modern (Xia et al., 2020). Selain itu, penelitian terkini juga menekankan pentingnya pengembangan agen yang lebih adaptif dan kontekstual dalam lingkungan virtual yang kompleks, termasuk penggunaan kombinasi rule-based system dengan teknik navigasi seperti NavMesh (Guzman & Wagner, 2026). Oleh karena itu, masih terdapat gap penelitian dalam integrasi *pathfinding*, *rule-based behaviour*, dan *smart reading* dalam satu sistem AI yang terpadu, khususnya pada *game* simulasi berbasis Android.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengimplementasikan *Artificial Intelligence* pada *game* simulasi menggunakan Unity dengan mengintegrasikan Unity NavMesh berbasis algoritma A* untuk navigasi, serta *Finite State Machine* (FSM) untuk mengatur perilaku NPC berbasis *rule-based behaviour dan smart reading*. Penelitian ini juga bertujuan untuk mengembangkan mekanisme interaksi antara pemain dan NPC dalam lingkungan simulasi pujasera, serta menguji kinerja dan responsivitas sistem pada platform Android. Dengan pendekatan ini, diharapkan dapat dihasilkan sebuah *game* simulasi yang lebih interaktif, adaptif, dan realistis, serta dapat menjadi kontribusi dalam pengembangan penelitian AI pada *game* di masa mendatang.

## 2. KAJIAN TEORITIS

*Game* merupakan media interaktif yang menggabungkan elemen hiburan, aturan, serta tujuan tertentu yang melibatkan partisipasi aktif pengguna. Dalam perkembangannya, *game* tidak hanya digunakan sebagai sarana hiburan, tetapi juga sebagai media simulasi yang mampu merepresentasikan kondisi dunia nyata secara virtual. *Game* simulasi dirancang untuk meniru aktivitas atau sistem nyata sehingga memberikan pengalaman yang realistis dan edukatif kepada pemain (Novak, 2022). Karakteristik utama dari *game* simulasi adalah adanya tingkat kompleksitas sistem dn interaksi yang tinggi, sehingga membutuhkan dukungan teknologi yang mampu menciptakan dinamika permainan yang adaptif (Sungkaew et al., 2022).

Dalam proses pengembangan *game*, penggunaan *game engine* seperti Unity menjadi salah satu faktor penting. *Unity* merupakan *game engine* multiplatform yang mendukung pengembangan *game* 2D dan 3D dengan berbagai fitur seperti *rendering, physics engine*, serta integrasi bahasa pemrograman C# (Hocking, 2022). Selain itu, Unity menyediakan sistem *Navigation Mesh* (NavMesh) yang berfungsi untuk menentukan area yang dapat dilalui *(walkable area)* oleh karakter dalam *game*. NavMesh memungkinkan agen AI melakukan navigasi secara otomatis dengan memanfaatkan algoritma *pathfinding*, seperti algoritma A*, untuk menemukan jalur terpendek secara efisien dalam lingkungan yang kompleks (Medina et al., 2023.; Gandhi & Prathamesh, 2024; Mutaqin et al., 2021).

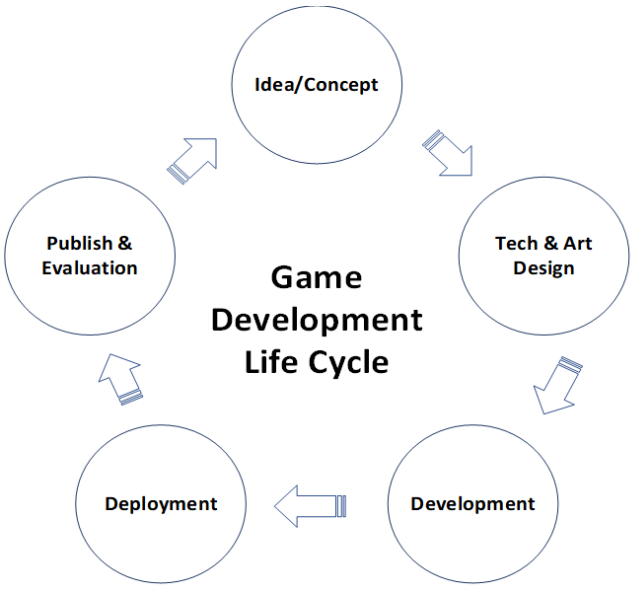
*Artificial Intelligence* (AI) dalam *game* digunakan untuk menciptakan perilaku cerdas pada karakter non-pemain atau *Non-Player Character* (NPC). Salah satu pendekatan yang umum digunakan adalah *Finite State Machine* (FSM), yaitu metode yang mengatur perilaku karakter berdasarkan sejumlah *state* yang dapat berubah sesuai kondisi tertentu (Arbansyah, 2020). FSM sering dikombinasikan dengan *rule-based behaviour*, yaitu pendekatan berbasis aturan sederhana seperti “jika–maka” untuk menentukan tindakan karakter. Selain itu, konsep *smart reading* digunakan untuk memungkinkan NPC membaca kondisi lingkungan dan merespons secara kontekstual. Kombinasi pendekatan ini mampu menghasilkan perilaku NPC yang lebih realistis, adaptif, dan responsif terhadap interaksi pemain (Karpouzis & Tsatiris, 2021; Wihartiko et al., 2023).

Beberapa penelitian terdahulu menunjukkan bahwa algoritma A* efektif digunakan dalam sistem *pathfinding* untuk menentukan jalur terpendek secara optimal. Hidayat & Qoiriah, (2022) mengimplementasikan A* dengan NavMesh pada sistem navigasi dalam ruangan dan memperoleh tingkat akurasi yang tinggi. Safira et al. (2021) juga menerapkan NavMesh pada *game* edukasi 3D dan menunjukkan hasil navigasi yang efisien. Selain itu, Lamia et al. (2022) serta Setiawan et al. (2023) menunjukkan bahwa penggunaan A* yang

dikombinasikan dengan metode perilaku seperti *behaviour tree* atau FSM mampu meningkatkan efektivitas pergerakan dan interaksi NPC dalam game. Namun, sebagian besar penelitian tersebut masih berfokus pada satu aspek AI secara terpisah. Oleh karena itu, penelitian ini mengintegrasikan NavMesh berbasis A*, FSM, *rule-based behaviour*, dan *smart reading* dalam satu sistem *game* simulasi untuk menghasilkan perilaku NPC yang lebih adaptif dan realistis.

## 3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan pengembangan sistem dengan metode. *Game Development Life Cycle* (GLDC). Metode ini dipilih karena mampu menggambarkan proses pengembangan *game* secara sistematis dari tahap perencanaan hingga evaluasi. Alur tahapan metode GDLC yang digunakan dapat dilihat pada Gambar 1 berikut.




**Gambar 1.** *Game Development Life Cycle.*

***Idea/Concept***

Tahapan *Idea* atau *Concept* merupakan tahapan identifikasi kebutuhan sistem dan perumusan konsep dasar *game*. Konsep inti *game* ini adalah simulasi manajemen Pujasera dengan batas waktu, di mana pemain mengatur tata ruang, melayani pelanggan, menjaga kebersihan, dan memperkerjakan karyawan. Ide ini merupakan adaptasi dari dua *game* referensi yaitu, mekanik pelayanan ala *Dinner Dash* dan lingkungan simulasi 3D serta sistem karyawan seperti dalam Kantin Sekolah Simulator.

### *Tech & Art Design*

Tahap *tech* dan *art design* mencakup proses perancangan teknis dan visual dari game. Pada tahap ini dijelaskan proses perancangan teknis dan visual game, yang mencakup *Game Design Document* (GDD), *Storyboard*, serta arahan desain *game* secara keseluruhan.

### *Game Design Document*

*Game Design Document* (GDD) adalah sebuah dokumen yang akan memperlihatkan konsep utama dan ide di proses pengembangan sebuah permainan. *Game Design Document game* Pujasera dapat dilihat pada Tabel 1.

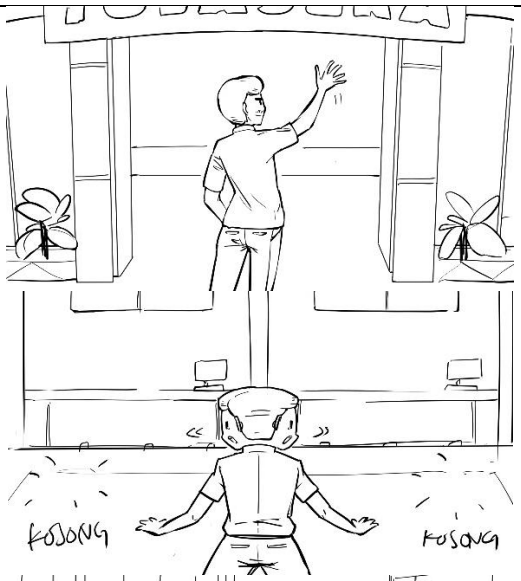
**Tabel 1.** *Game Design Document.*

| **Pendahuluan** | |
|---|---|
| Nama *Game* | Pujasera |
| Deskripsi Singkat | Pemain harus dapat mengelola pujasera agar mendapatkan hasil yang maksimal. |
| *Gendre* dan *Platform* | Simulator dan Android |
| Target Audiens | Remaja dan Dewasa |
| Konsep dan *Gameplay* | |
| Premis | Pujasera baru telah dibuka, kamu diminta oleh pamanmu untuk menjadi pengelola disana. Awalnya kamu kesulitan karena harus mengurus semuanya sendirian. Namun, pada akhirnya kamu akan dibantu oleh rekan-rekanmu untuk mengelola pujasera setelah sekian waktu. |
| Sistem kontrol | *Joystick* dan *Button* |
| Mekanik Inti | 1. Mengambil pesanan untuk pelanggan<br>2. Memberi pesanan untuk pelanggan<br>3. Menyiapkan kursi dan meja<br>4. Menambha karyawan |
| Level dan Dunia *Game* | |
| Desain Level | Lingkungan Pujasera |
| Interaksi dengan lingkungan | 1. Interaksi dengan *stand*<br>2. Interaksi dengan *shop*<br>3. Interaksi dengan pelanggan |
| Karakter | |
| Karakter utama | Pemain sebagai pengelola pujasera |
| NPC | 1. Pelayan<br>2. Pembersih sampah<br>3. Pelanggan |

***Storyboard***

*Storyboard* merupakan representasi visual yang menggambarkan alur dan *gameplay* dari sebuah game. Berikut ini Tabel 2 *storyboard* untuk *game* Pujasera

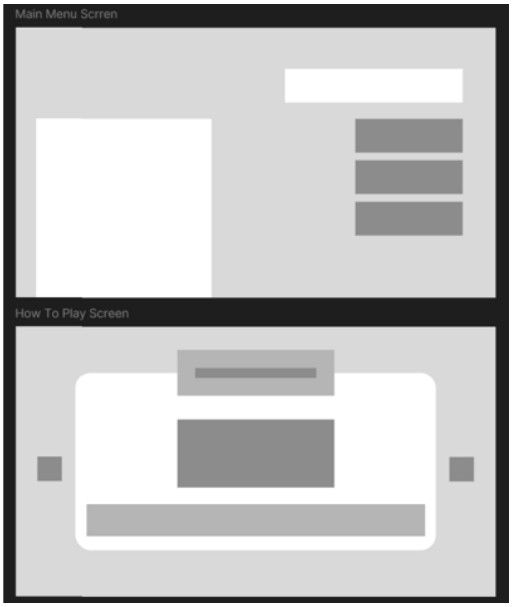
**Tabel 2.** *Storyboard Game.*

| Gambar | Penjelasan |
|---|---|
|  | Pemain berjalan menuju Pujasera dan membuka pintu Pujasera |
|   | Pemain akan memulai di Pujasera yang kosong, hanya ada bangunan *stand* yang kosong |
|   | Pemain akan menata meja, kursi dan membuka *stand* agar dapat mengundang pelanggan makan di Pujasera ini |
|  | Karena masih belum memiliki pegawai, maka pemain harus menerima pesanan dari pelanggan oleh dirinya sendiri |
|  | Setelah menerima pesanan, maka pemain akan mengunjungi *stand* untuk membawa pesanan |

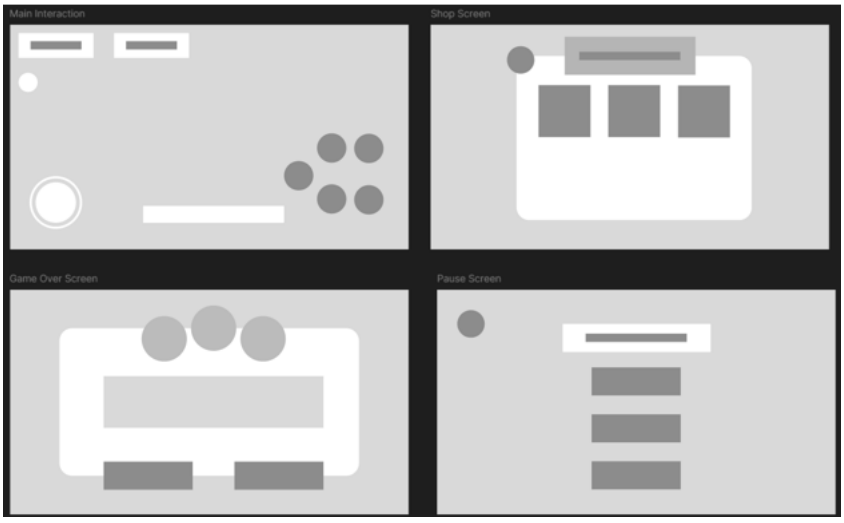
| | |
|---|---|
|  | Setelah pesanan selesai, pemain akan mengantarkan pesanan ke meja pelangggan, pelanggan akan memberikan uang dan juga nilai popularitas untuk Pujasera |
|  | Terkadang, ada pelanggan yang suka buang sampah sembarangan. Dan itu juga jadi tugas pemain untuk membersihkannya |
|  | Seiring berjalannya waktu, pemain bisa merekrut pegawai untuk membantu pekerjaan pemain |
|  | Saat uang dan popularitas sudah cukup, maka permainan akan selesai dengan memperlihatkan pemain dan karyawannya berhasil membawa Pujasera kembali ke masa kejayaannya |

***Wireframe Interface***

*Wireframe* untuk tampilan *game* Pujasera pada scene *Main Menu* dan *Main Game* disajikan pada Gambar 2 dan 3.




**Gambar 2.** *Wireframe Main Menu.*




**Gambar 3.** *Wireframe Main Game.*

***Development***

Tahap *development* merupakan tahap implementasi dari seluruh perancangan sistem yang telah dibuat sebelumnya. Pada tahap ini, pengembangan *game* dilakukan menggunakan Unity sebagai *game engine* dengan bahasa pemrograman C#. Selain itu, digunakan *Playmaker* sebagai alat visual *scripting* untuk mempermudah implementasi logika berbasis *Finite State Machine* (FSM). Seluruh elemen permainan, seperti lingkungan pujasera, karakter pemain, serta *Non-Player Character* (NPC), mulai direalisasikan dalam bentuk objek 2D/3D sesuai dengan desain yang telah dirancang.

Implementasi *Artificial Intelligence* (AI) dilakukan dengan mengintegrasikan beberapa pendekatan, yaitu *pathfinding*, *rule-based behaviour*, dan *Finite State Machine* (FSM). Sistem *pathfinding* diterapkan menggunakan Unity NavMesh yang memungkinkan NPC bergerak

secara otomatis menuju target dengan jalur optimal. Sementara itu, FSM digunakan untuk mengatur perilaku NPC berdasarkan state tertentu, seperti *idle, walking, ordering*, dan *serving*. Setiap *state* memiliki kondisi transisi yang ditentukan menggunakan aturan berbasis logika (*rule-based*), sehingga NPC dapat merespons lingkungan secara dinamis.

Selain itu, diterapkan konsep *smart reading*, yaitu kemampuan NPC dalam membaca kondisi lingkungan untuk menentukan tindakan yang sesuai. Contohnya, NPC pelanggan akan mencari tempat duduk yang tersedia, melakukan pemesanan, serta memberikan respon terhadap pelayanan yang diberikan. Di sisi lain, NPC pegawai dirancang untuk membantu aktivitas operasional seperti mengambil pesanan dan melayani pelanggan. Seluruh interaksi ini yang disajikan pada Tabel 3 diatur melalui kombinasi FSM dan sistem navigasi berbasis NavMesh.

**Tabel 3.** *Rule* Perilaku Berdasarkan *Agent* menggunakan *Smart Reading*.

| **NPC** | ***State* Utama** | ***Rule* Perilaku** | ***Smart Reading*** |
|---|---|---|---|
| *Customer* | *Idle*, Mencari Kursi, Duduk, Memesan, Menunggu, Makan, Selesai | a. Jika tidak ada kursi, maka *customer* akan pulang.<br>b. Jika *customer* merupakan *customer* baik, maka akan membantu untuk membuang sampah. Jika bukan, maka langsung ke cek kursi.<br>c. Jika ada kursi, maka *customer* menuju kursi.<br>d. Jika kursi sudah ada yang menempati, maka cari kursi lain.<br>e. Jika tidak ada kursi lain, maka pulang.<br>f. Jika kursi kosong, maka duduk.<br>g. Jika sedang duduk, maka pilih makanan.<br>h. Jika sudah pilih makanan, maka tunggu.<br>i. Jika pesanan diterima, maka akan menikmati pesanan.<br>j. Jika sudah selesai, maka akan memberikan koin dan hati lalu pulang.<br>k. Jika sudah selesai, maka ada kemungkinan buang sampah.<br>l. Jika pesanan tidak diterima, maka akan mengurangi hati lalu pulang.<br>m. Jika ada sampah, maka pulang dan mengurangi hati. | a. Saat *customer spawn* atau datang, maka akan mencari objek kursi terdekat dengan tag "*Seat Point*".<br>b. Jika *customer* tidak menemukan kursi kosong atau kursi tidak ada maka *customer* akan pulang dengan mencari "*Customer Point*" terjauh.<br>c. Jika *customer* menemukan kursi dengan tag "*Seat Point*", maka akan menuju kursi tersebut.<br>d. Jika kursi yang *customer* pilih sudah terduduki atau tagnya berubah menjadi "*Untagged*", maka *customer* akan mencari kursi lain.<br>e. Jika kursi yang *customer* pilih masih memiliki tag "*Seat Point*", maka *customer* akan duduk disana |
| *Helper* | *Wander*, Cek Pesanan, Ambil Pesanan, Antar Pesanan | a. Jika tidak ada pesanan, maka akan berjalan-jalan.<br>b. Jika setelah jalan-jalan, maka cek pesanan. | a. Saat *helper* sedang berada pada *state wander, helper* akan terus mencari *customer* yang sudah memilih makanan. |

| | | | |
|---|---|---|---|
| | | c. Jika tidak ada pesanan, maka kembali jalan-jalan.<br>d. Jika ada pesanan, maka cek pesanan apa.<br>e. Jika sudah mengecek pesanan, maka menuju ke tenant.<br>f. Jika sudah mencapai tenant, maka buat pesanan.<br>g. Jika pesanan sudah dibuat, maka ambil.<br>h. Jika sudah mengambil pesanan, maka berikan pada *customer*.<br>i. Jika sudah memberikan pesanan pada *customer*, maka kembali jalan-jalan. Jika pelanggan sudah pulang namun pesanan belum diberikan, maka cari pelanggan lain. | b. Ketika *customer* yang sudah dipilih berubah tag nya menjadi "*Untagged*", maka *helper* akan mencari *customer* lainnya.<br>c. *Customer* akan menuju tenant berdasarkan menu pilihan *customer* dan membawa pesanan tersebut kepada customer.<br>d. Ketika *customer* yang sudah dipilih berubah tag nya menjadi "*Untagged*" atau *customer* sudah pergi, maka *helper* akan mencari *customer* lain yang memilih pesanan yang sama |
| *Cleaner* | *Wander*, Menuju Sampah, Membersihkan | a. Jika tidak ada sampah, maka akan berjalan-jalan.<br>b. Jika ada sampah, maka akan menuju sampah.<br>c. Jika sudah sampai pada sampah, maka akan dibersihkan.<br>d. Jika sudah ada yang membersihkan sampah, maka akan berjalan-jalan.<br>e. Jika sudah selesai membersihkan sampah, maka akan berjalan-jalan. | a. Saat *cleaner* sedang berada pada state *wander*, *cleaner* akan mencari objek sampah terdekat yang memiliki tag "*Trash*" dan akan menuju sampah tersebut.<br>b. Jika saat *cleaner* menuju sampah, sampah tersebut berubah tag nya menjadi "*Untagged*" maka *cleaner* akan kembali jalan-jalan.<br>c. Jika saat *cleaner* menuju sampah, sampah tersebut tag nya masih "*Trash*" maka *tag* sampah akan diubah menjadi "*Untagged*" dan membersihkan sampahnya. |

## 4. HASIL DAN PEMBAHASAN

### Hasil Implementasi Tampilan Sistem

Hasil implementasi menunjukkan bahwa tampilan sistem *(user interface) game* simulasi Pujasera telah berhasil merepresentasikan keseluruhan fungsi dan metode yang digunakan dalam penelitian. Antarmuka yang dikembangkan tidak hanya menampilkan elemen visual, tetapi juga menggambarkan hasil penerapan metode *Artificial Intelligence* seperti NavMesh, *Finite State Machine* (FSM), dan *smart reading* secara langsung dalam lingkungan permainan.

Pada tampilan awal, pemain diarahkan untuk memasuki area Pujasera yang masih dalam kondisi kosong. Selanjutnya, pemain dapat melakukan pengaturan lingkungan seperti penempatan meja, kursi, serta pembukaan *stand* untuk memulai operasional. Tampilan utama permainan menampilkan interaksi secara *real-time* antara pemain dan NPC, termasuk aktivitas pemesanan, pelayanan, serta pembersihan lingkungan.

Selain itu, implementasi NavMesh dapat terlihat dari pergerakan NPC yang mampu berpindah dari satu lokasi ke lokasi lain secara otomatis dengan jalur yang optimal. Perilaku NPC tersebut diatur menggunakan FSM yang mengontrol transisi antar *state*, seperti *idle*, *walking*, *ordering*, dan *serving*, sehingga setiap NPC memiliki alur perilaku yang terstruktur.

Penerapan *smart reading* juga dapat diamati melalui tampilan sistem, di mana NPC mampu merespons kondisi lingkungan secara dinamis. Contohnya, NPC *customer* dapat mencari kursi yang tersedia, *helper* dapat mendeteksi pesanan pelanggan, serta *cleaner* dapat menemukan dan membersihkan sampah secara otomatis. Hal ini menunjukkan bahwa tampilan sistem tidak hanya bersifat visual, tetapi juga menjadi representasi langsung dari kecerdasan sistem yang dibangun.




**Gambar 4.** Tampilan Antarmuka dan Implementasi *Smart Reading* pada *Game* Simulasi Pujasera.

Gambar 4 menunjukkan bagaimana antarmuka sistem menampilkan interaksi antara pemain dan NPC secara *real-time*. Selain itu, terlihat bahwa pergerakan dan perilaku NPC merupakan hasil integrasi NavMesh, FSM, dan *smart reading*, sehingga menghasilkan sistem yang interaktif dan adaptif.

**Hasil Pengujian Sistem**

Dalam mendukung analisis, hasil pengujian sistem dilakukan dengan menggunakan uji coba struktural, fungsional, validasi, responsivitas, dan kuisioner. Hasil pengujian disajikan dalam Tabel 4.

**Tabel 4.** Hasil Pengujian Sistem.

| No | Jenis Pengujian | Komponen | Skenario | Hasil |
|---|---|---|---|---|
| 1 | Struktural | *Player State* | Transisi *state player* | Berhasil |
| 2 | Struktural | *Customer State* | Perilaku *customer* | Berhasil |
| 3 | Struktural | *Helper State* | Pelayanan *helper* | Berhasil |
| 4 | Struktural | *Cleaner State* | Pembersihan sampah | Berhasil |
| 5 | Fungsional | *Main Menu* | Navigasi menu | Berhasil |
| 6 | Fungsional | *Game Scene* | Interaksi *gameplay* | Berhasil |
| 7 | Validasi | Sistem | Kesesuaian dengan desain | Valid |
| 8 | Responsivitas | Aplikasi | Performa perangkat | Responsif |
| 9 | Kuisioner | Pengguna | Tingkat kepuasan | Baik |

Berdasarkan hasil uji coba yang telah dilakukan, dibuat ringkasan hasil pengujian sistem sebagaimana ditunjukkan pada Tabel 4. Tabel ini merangkum seluruh hasil pengujian yang terdiri dari uji coba struktural, fungsional, validasi, responsivitas, dan kuisioner.

## 5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengembangan *game* pujasera, dapat disimpulkan bahwa algoritma A* *Pathfinding*, *Unity Navmesh*, *rule-based behaviour*, dan *smart reading* berhasil diimplementasikan dan direalisasikan menggunakan *Unity* serta *Playmaker visual scripting*. Penelitian ini juga menunjukkan bahwa kode C# dan visual *scripting* dapat dilakukan secara bersamaan, asalkan tidak terjadi konflik atau tumpang tindih antar fungsi.

Konfigurasi nilai pada *Navmesh Surface*, *Navmesh Obstacle*, dan *Navmesh Agent* merupakan faktor kritis yang menentukan perilaku pergerakan *agent* di dalam area yang telah diproses *(baked)*. Semakin kompleks dan banyak objek dalam sebuah *scene*, waktu yang dibutuhkan untuk proses kalkulasi *baking mesh* akan semakin lama.

Hingga tahap penulisan ini, tidak ditemukan *error* kritis yang mengganggu performa game. Pengujian dilakukan terhadap *smartphone* para responden kuisioner dengan hasil yang stabil dan lancar. Pengujian akan terus dilakukan hingga *game* siap untuk dipublikasikan.

*Game* Pujasera ini dirancang untuk kompatibel dengan *smartphone* yang mendukung minimal API Level 23 (Android 6.0 *"Marshmallow"*) dan memiliki ruang penyimpanan internal tersedia sekitar 300 MB dengan ukuran file *game* sekitar 85 MB.

Penelitian ini menggunakan *Unity* 6 dan berjalan lancar tanpa kendala berarti. Implementasi yang sama tetap memungkinkan pada versi *Unity* yang berbeda, meski mungkin memerlukan langkah tambahan atau penyesuaian teknis karena perbedaan fitur antar versi.

Untuk penelitian berikutnya, pengembangan bisa difokuskan pada penambahan sistem level dan alur cerita. Kedua elemen ini dapat membuat pemain lebih nyaman bermain karena merasa memiliki tujuan yang jelas dan keterikatan dengan perkembangan dalam *game*. Contoh

nyata penerapannya bisa dilihat pada berbagai *game* simulasi karya Akhir Pekan Studio, yang sering kali menyajikan progresi permainan melalui cerita sederhana namun tetap imersif.

Penelitian lanjutan juga dapat memperkaya pengalaman bermain dengan menghadirkan lebih banyak variasi, misalnya jenis pelanggan dengan karakter berbeda, menu makanan yang lebih beragam, atau opsi dekorasi tempat. Dari segi teknis, penerapan AI yang lebih cerdas untuk mengatur perilaku NPC agar mendapatkan pengalaman bermain yang lebih menyenangkan.

Penyesuaian tempat agar lebih mirip dan terasa realistis juga menjadi saran pengembangan yang dapat dilakukan ke depannya. Dengan adanya hal tersebut maka akan meningkatkan pengalaman bermain para pemain.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing atas bimbingan dan arahannya selama penelitian ini. Ucapan terima kasih juga disampaikan kepada pihak institusi, tim pengembang, serta para responden yang telah berkontribusi dalam proses pengembangan dan pengujian game.

Penulis juga mengapresiasi semua pihak yang telah memberikan dukungan sehingga penelitian ini dapat terselesaikan dengan baik.

## DAFTAR REFERENSI

Amalia, S., Abidullah, M. D. F., Marcellino, F., Rabani, D. D., Azzahra, F. F., & Abdiansah, A. (2024). AI-Driven Traffic Simulation using Unity: Implementing Finite State Machines for Adaptive NPC Behaviour. *Journal of Intelligent Computing & Health Informatics*, *5*(2), 72. https://doi.org/10.26714/jichi.v5i2.14595

Arbansyah. (2020). *Implementasi finite state machine (FSM) pada agent permainan game lost animal at borneo berbasis android* .

Arifudin, D., Fransjaya, M., El Fakhry, Y., & Syahrizaldy, H. A. (2025). Optimization of Player Experience and Enemy AI using A* Algorithm in Game. *Sinkron*, *9*(1), 434–443. https://doi.org/10.33395/sinkron.v9i1.14349

Berrezueta-Guzman, S., & Wagner, S. (2026). Next-Gen orientation: supporting international students with generative AI NPCs in VR. *Frontiers in Computer Science*, *8*. https://doi.org/10.3389/fcomp.2026.1799323

Carlos Medina, R. S., Shane Gittabao, H. B., & Agustin, V. A. (2023). *Modifying JPS Algorithm Using Navmesh Data Structure Applied in 3D Using Unity*.

del Bosque, A., Fernández-Arias, P., Lampropoulos, G., & Vergara, D. (2025). The Role of Artificial Intelligence in Gaming. *Applied Sciences*, *15*(23), 12358. https://doi.org/10.3390/app152312358

Gandhi, A., & Punde Prathamesh. (2024). A* Algorithm and Unity for Augmented Reality-based Indoor Navigation. *Grenze International Journal of Engineering & Technology*, *10*(1).

Hidayat, M. I., & Qoiriah, A. (2022). *Implementasi Pathfinding dengan Algoritma A* pada Aplikasi Indoor Navigation Menggunakan Unity Navmesh*. https://doi.org/10.26740/jinacs.v3n03.p334-342

Jeannie Novak. (2022a). *Game Development Essentials: An Introduction (4th Edition)* (4th ed.). NOVY Unlimited.

Jeannie Novak. (2022b). *Game Development Essentials: An Introduction (4th Edition)* (4th ed.). NOVY Unlimited.

Jeannie Novak. (2022c). *Game Development Essentials: An Introduction (4th Edition)* (4th ed.). NOVY Unlimited.

Joseph Hocking. (2022). *Unity in Action, Third Edition: Multiplatform Game Development in C#* (B. Whitney (ed.); Vol. 1). Manning Publications.

Junaidi, A., & Yunus, A. (2021). *Implementasi Behavior Tree pada Perilaku Npc di Game Sidescroller*. https://doi.org/10.33479/kurawal.v4i2.459

Karpouzis, K., & Tsatiris, G. (2021). *AI in (and for) Games*. http://arxiv.org/abs/2105.03123

Lamia, K., Lumenta, A., & Sugiarso, B. (2022). *Implementation of the A* (A Star) Algorithm in the Minahasa Culture 3D Game*. *11*, 55–66. https://doi.org/10.35793/jtek.11.2.2022.40153

Mutaqin, G., Fadilah, J. N., & Nugroho, F. (2021). Implementasi Metode Path Finding dengan Penerapan Algoritma A-Star untuk Mencari Jalur Terpendek pada *Game* "Jumrah Launch Story." *Walisongo Journal of Information Technology*, *3*(1), 43–48. https://doi.org/10.21580/wjit.2021.3.1.7042

Safira, L., Harsadi, P., & Harjanto, S. (2021). Penerapan Navmesh dengan Algoritma A Star Pathfinding Pada *Game* Edukasi 3d Go Green. *Jurnal Teknologi Informasi Dan Komunikasi (TIKomSiN)*, *9*(1), 17. https://doi.org/10.30646/tikomsin.v9i1.540

Setiawan, M. L., Arbansyah, A., & Suryawan, S. H. (2023). Penerapan Algoritma A* dan Behaviour Trees untuk Perilaku Non-Player Character(NPC) pada *Game* "The Last Hope" Berbasis Android Menggunakan Unity 2D. *Jurnal CoSciTech (Computer Science and Information Technology)*, *4*(2), 451–460. https://doi.org/10.37859/coscitech.v4i2.5418

Sungkaew, K., Lungban, P., & Lamhya, S. (2022). *Game* development software engineering: digital educational *game* promoting algorithmic thinking. *International Journal of Electrical and Computer Engineering*, *12*(5), 5393–5404. https://doi.org/10.11591/ijece.v12i5.pp5393-5404

Wihartiko, F. D., Nurdiati, S., & Budiono, A. (2023). *Solving Agricultural Price Recommendation Problem Using Smart Reading Algorithms*. https://doi.org/10.1016/j.procs.2022.12.275

Xia, B., Ye, X., & Abuassba, A. O. . (2020). Recent Research on AI in Games. *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 505–510. https://doi.org/10.1109/IWCMC48107.2020.9148327