HOAQ: JURNAL TEKNOLOGI INFORMASI, Volume 17 Nomor 2, Desember 2026, p259-270
p-ISSN: 2337-5280, e-ISSN: 26207427
DOI: https://doi.org/10.52972/hoaq.vol17no2

PERANCANGAN AGEN MUSUH BERBASIS KECERDASAN BUATAN PADA *PLATFORM* SCRATCH

Yoseph Satria Praka

Program Studi Teknik Multimedia Digital, Politeknik Negeri Jakarta
Jl. Prof. DR. G.A. Siwabessy, Kampus Universitas Indonesia Depok – Jawa Barat, Indonesia
Email: yoseph.satria.praka.tik23@stu.pnj.ac.id

ABSTRAK

Penelitian ini bertujuan untuk merancang dan mengimplementasikan agen musuh berbasis kecerdasan buatan pada *platform* Scratch menggunakan mekanisme pemilihan target terdekat berdasarkan jarak Euclidean. Agen dirancang menggunakan pendekatan *goal-based agent* dengan tujuan memilih dan mengumpulkan objek terdekat untuk meningkatkan skor. Penelitian menggunakan model pengembangan 4D yang terdiri atas *Define, Design, Development,* dan *Disseminate*. Subjek penelitian berupa gim *Wayang War* yang melibatkan satu *Player* dan satu agen AI yang berkompetisi mengumpulkan tiga jenis objek selama 30 detik. Pengujian dilakukan melalui *black-box testing* terhadap 19 skenario fungsional dan 50 percobaan *gameplay*. Data dianalisis menggunakan statistik deskriptif dan *paired-samples t-test*. Hasil *black-box testing* menunjukkan tingkat keberhasilan sebesar 100%. Dari 50 percobaan *gameplay*, *Player* memperoleh *win rate* sebesar 54%, AI sebesar 40%, dan *draw rate* sebesar 6%. Rata-rata skor *Player* sebesar 52,90 (SD = 8,72), sedangkan AI sebesar 52,64 (SD = 9,01). Hasil *paired-samples t-test* menunjukkan tidak terdapat perbedaan yang signifikan secara statistik antara skor *Player* dan AI ($t(49) = 0{,}233$; $p = 0{,}817$), dengan ukuran efek *Cohen's dz* sebesar 0,0329. Hasil tersebut menunjukkan bahwa agen mampu melakukan pemilihan target secara otonom berdasarkan kondisi permainan dan menjalankan perilaku musuh berbasis tujuan.

Kata kunci: Kecerdasan Buatan, Agen Cerdas, Scratch, Jarak Euclidean, Gim.

ABSTRACT

This study aims to design and implement an artificial intelligence-based enemy agent on the Scratch platform using a nearest-target selection mechanism based on Euclidean distance. The agent was designed using a goal-based agent approach to select and collect the nearest available object to increase its score. The research employed the 4D development model consisting of Define, Design, Development, and Disseminate. The research subject was a game called Wayang War, featuring one Player and one AI agent competing to collect three types of objects within a 30-second gameplay session. Testing consisted of black-box testing involving 19 functional scenarios and 50 gameplay trials. The data were analyzed using descriptive statistics and a paired-samples t-test. Black-box testing achieved a 100% success rate. Across the 50 gameplay trials, the Player achieved a 54% win rate, while the AI achieved a 40% win rate and a 6% draw rate. The Player obtained a mean score of 52.90 (SD = 8.72), while the AI obtained a mean score of 52.64 (SD = 9.01). The paired-samples t-test indicated no statistically significant difference between the Player and AI scores (t(49) = 0.233; p = 0.817), with a Cohen's dz effect size of 0,0329. These findings indicate that the agent was able to autonomously select targets based on the game state and perform goal-based enemy behavior.

Keywords: Artificial Intelligence, Intelligent Agent, Scratch, Euclidean Distance, Game.

1. PENDAHULUAN

Perkembangan kecerdasan buatan telah memperluas kemampuan permainan digital dengan memungkinkan karakter non-pemain untuk memahami kondisi permainan, mengambil keputusan, dan merespons tindakan pemain [1], [2], [3]. Dalam lingkungan permainan, agen cerdas dapat dirancang untuk menjalankan perilaku secara mandiri, seperti memilih target, menentukan pergerakan, dan berinteraksi dengan objek [4], [5]. Salah satu pendekatan yang dapat digunakan untuk menerapkan perilaku tersebut adalah pemilihan target berdasarkan jarak, yaitu agen mengevaluasi jarak antara dirinya dengan objek-objek yang tersedia sebelum menentukan target terdekat. Pendekatan ini memberikan bentuk pengambilan keputusan yang sederhana dan dapat diterapkan pada lingkungan permainan interaktif [6].

Scratch merupakan *platform* pemrograman visual yang mendukung pengembangan permainan interaktif melalui pemrograman berbasis blok. Lingkungan pemrograman visual yang dimilikinya memungkinkan

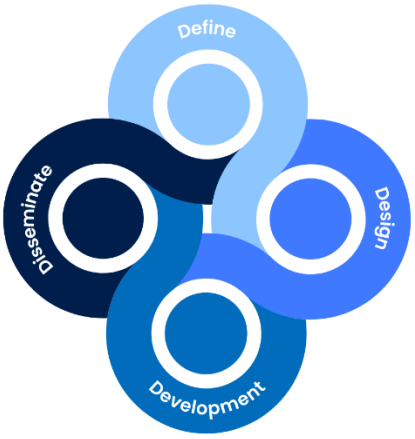
Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

mekanisme permainan dan konsep komputasi diterapkan melalui struktur pemrograman yang mudah dipahami [7], [8], [9]. Penelitian sebelumnya telah menunjukkan pemanfaatan Scratch dalam berbagai pengembangan permainan. Penelitian [10] mengeksplorasi integrasi AI dan pembelajaran berbasis permainan menggunakan Scratch dan Unity untuk mendukung pembelajaran matematika yang interaktif. Sementara itu, penelitian [11] mengembangkan permainan Space Shooter menggunakan Scratch dengan mekanisme permainan yang melibatkan interaksi dengan musuh, kemampuan bertahan pemain, dan pengumpulan skor. Penelitian lainnya memanfaatkan Scratch untuk mengembangkan permainan berbasis labirin [12], serta permainan edukatif interaktif untuk pembelajaran sejarah kerajaan Hindu-Buddha [13]. Selain untuk pengembangan permainan, Scratch juga telah digunakan dalam penerapan algoritma komputasi, seperti *Circle Collision Detection*, untuk mendukung interaksi objek permainan secara waktu nyata [14]. Berbagai penelitian tersebut menunjukkan bahwa Scratch dapat digunakan sebagai lingkungan untuk menerapkan mekanisme permainan interaktif dan pendekatan komputasi.

Berdasarkan penelitian-penelitian tersebut, Scratch telah digunakan untuk mengembangkan permainan interaktif, mekanisme *gameplay*, interaksi pengguna, konten edukatif, serta penerapan algoritma komputasional pada interaksi antarobjek. Namun, masih terdapat ruang pengembangan pada penerapan pengambilan keputusan otonom pada karakter musuh, khususnya mekanisme *goal-based agent* yang secara dinamis memilih target berdasarkan kondisi lingkungan dalam skenario permainan kompetitif. Penelitian ini mengisi ruang tersebut melalui pengembangan *Wayang War*, yaitu permainan berbasis Scratch dengan karakter musuh yang bersaing dengan *Player* untuk mengumpulkan tiga objek. Agen musuh mengamati posisi objek yang tersedia, menghitung jarak Euclidean dari posisinya ke setiap objek, kemudian memilih objek dengan jarak terpendek sebagai target. Ketika kondisi permainan berubah, seperti objek telah dikumpulkan dan berpindah ke posisi baru, agen melakukan pemilihan target kembali. Dengan mekanisme tersebut, agen tidak bergantung pada urutan target yang telah ditentukan, tetapi mengambil keputusan berdasarkan kondisi permainan pada setiap siklus. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan *goal-based agent* dengan mekanisme pemilihan target terdekat berbasis jarak Euclidean pada platform Scratch. Kontribusi utama penelitian terletak pada implementasi hubungan antara kondisi lingkungan, tujuan agen, evaluasi jarak, dan tindakan agen dalam bentuk pengambilan keputusan otonom pada karakter musuh dalam permainan kompetitif berbasis pemrograman visual.

2. METODE PENELITIAN

Penelitian ini menggunakan model pengembangan 4D yang terdiri atas empat tahapan, yaitu *Define*, *Design*, *Development*, dan *Disseminate* [15]. Prosedur penelitian disajikan pada Gambar 1 yang menggambarkan alur kerja secara keseluruhan dalam merancang, mengimplementasikan, dan mengevaluasi agen musuh cerdas pada permainan *Wayang War*.



Gambar 1. Tahapan Penelitian dengan Model 4D

Tahap *Define* berfokus pada penentuan konsep dasar yang diperlukan dalam pengembangan permainan dan agen cerdas. Konsep permainan mencakup *gameplay*, tujuan permainan, peran pemain dan musuh, objek yang dapat dikumpulkan, serta mekanisme interaksi [16]. Sementara itu, konsep AI mencakup perilaku agen musuh, khususnya kemampuannya dalam mengidentifikasi tiga objek yang tersedia, menghitung jarak terhadap setiap objek, dan memilih objek terdekat sebagai target menggunakan jarak Euclidean. Konsep-konsep tersebut menjadi dasar untuk tahapan pengembangan selanjutnya.

Tahap *Design* menerjemahkan konsep yang telah ditentukan ke dalam rancangan sistem. *Flowchart* permainan dibuat untuk menggambarkan urutan *gameplay*, meliputi inisialisasi permainan, pengumpulan objek, perhitungan skor, dan penghentian permainan [17]. Selain itu, diagram model agen dirancang untuk menggambarkan proses persepsi, pengambilan keputusan, dan tindakan yang dilakukan oleh agen cerdas. Agen menerima informasi mengenai posisi objek yang tersedia, membandingkan jarak setiap objek terhadap musuh, memilih objek dengan jarak minimum, kemudian bergerak menuju target yang telah dipilih [6].

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Tahap *Development* meliputi proses implementasi permainan dan agen cerdas menggunakan platform Scratch. Mekanisme permainan, objek yang dapat dikumpulkan, perilaku pemain, pergerakan musuh, sistem penilaian, dan timer diimplementasikan sesuai dengan rancangan yang telah dibuat. Mekanisme AI diterapkan menggunakan blok pemrograman Scratch untuk menghitung jarak antara musuh dan setiap objek yang tersedia serta menentukan target terdekat. *Prototype* yang dihasilkan kemudian dipersiapkan untuk dilakukan evaluasi fungsional.

Tahap *Disseminate* meliputi pengujian *black-box*, pengujian *gameplay*, dan distribusi. Pengujian *black-box* dilakukan untuk memverifikasi bahwa fungsi permainan dan agen cerdas dapat berjalan sesuai dengan kebutuhan yang telah ditentukan tanpa memeriksa implementasi internal *system* [18]. Selanjutnya, pengujian *gameplay* dilakukan sebanyak 50 kali percobaan dengan durasi masing-masing permainan selama 30 detik. Setiap percobaan dimulai dari kondisi awal yang sama untuk menjaga konsistensi eksperimen. Posisi awal *Player* ditetapkan pada koordinat X = -71 dan Y = -26, sedangkan posisi awal *Enemy* atau agen AI ditetapkan pada X = 95 dan Y = -37. Tiga objek yang menjadi target permainan juga ditempatkan pada posisi awal yang telah ditentukan, yaitu Poin 1 pada koordinat X = 11 dan Y = 125, Poin 2 pada X = -34 dan Y = 104, serta Poin 3 pada X = 30 dan Y = 90. Dengan menetapkan kondisi awal yang sama pada setiap percobaan, perbedaan hasil *gameplay* dapat diamati berdasarkan interaksi dan pengambilan keputusan selama permainan, bukan akibat perbedaan konfigurasi awal.

Pada setiap percobaan, *Player* dikendalikan secara manual menggunakan kontrol permainan, sedangkan *Enemy* dikendalikan secara otomatis oleh agen cerdas menggunakan mekanisme pemilihan target terdekat berdasarkan jarak Euclidean. Pada awal permainan, agen mengevaluasi jarak antara posisi *Enemy* dan ketiga objek yang tersedia, kemudian memilih objek dengan jarak terpendek sebagai target. Ketika objek berhasil dikumpulkan dan berpindah ke posisi baru sesuai mekanisme permainan, kondisi lingkungan berubah dan agen melakukan evaluasi ulang untuk menentukan target berikutnya. Dengan demikian, meskipun kondisi awal setiap percobaan dibuat konsisten, posisi objek selama *gameplay* dapat berubah akibat interaksi antara *Player*, AI, dan objek permainan. Pada akhir setiap percobaan, skor *Player* dan AI dicatat dan digunakan untuk menentukan hasil permainan, yaitu kemenangan *Player*, kemenangan AI, atau hasil seri. Seluruh 50 percobaan menggunakan durasi permainan, aturan pengumpulan objek, mekanisme pergerakan AI, dan kondisi awal yang sama. Hasil pengujian kemudian dianalisis menggunakan statistik deskriptif untuk memperoleh nilai minimum, maksimum, rata-rata, median, standar deviasi, varians, dan distribusi hasil kemenangan.

3. HASIL DAN PEMBAHASAN

Konsep Permainan

Wayang War, merupakan permainan kompetitif yang mempertemukan pemain dengan musuh yang dikendalikan oleh AI untuk bersaing dalam mengumpulkan objek dan memperoleh skor tertinggi. *Gameplay* dirancang dengan tujuan yang sederhana agar fokus utama penelitian dapat diarahkan pada perilaku pengambilan keputusan agen musuh cerdas. Konsep umum permainan disajikan pada Tabel 1.

Tabel 1. Konsep Permainan *Wayang War*

Komponen Permainan	Deskripsi
Nama Permainan	*Wayang War*
Jenis Permainan	Permainan kompetitif berbasis pengumpulan objek
Platform	Scratch
Durasi Permainan	30 detik untuk setiap permainan
Pemain	Dikendalikan secara langsung oleh pengguna
Musuh	Dikendalikan oleh agen cerdas
Objek yang Dikumpulkan	3 jenis objek yang tersedia dalam permainan
Tujuan Utama	Mengumpulkan objek sebanyak mungkin sebelum permainan berakhir
Sistem Skor	Setiap objek yang dikumpulkan menambah skor pemain atau musuh
Kondisi Menang	Pemain atau musuh dengan skor tertinggi pada akhir 30 detik dinyatakan sebagai pemenang
Kondisi Seri	Permainan berakhir seri apabila pemain dan musuh memperoleh skor yang sama

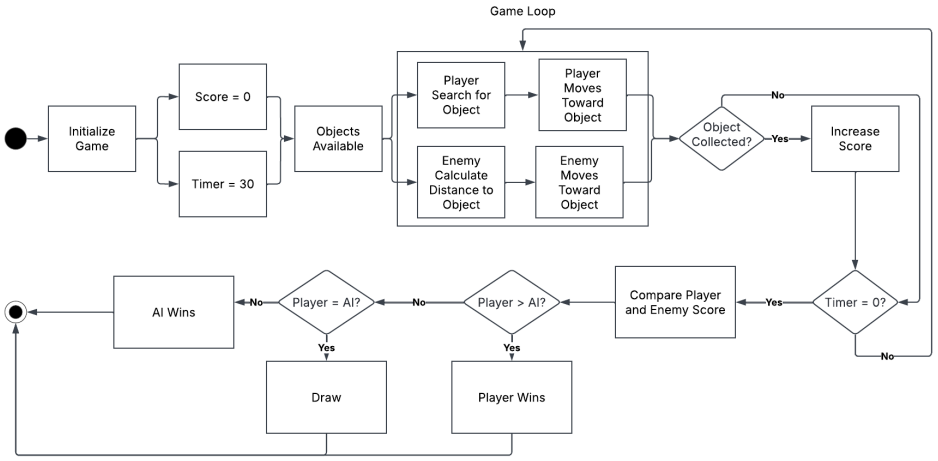
Seperti ditunjukkan pada Tabel 1, *Wayang War* dirancang sebagai permainan kompetitif berbasis pengumpulan objek dengan durasi permainan selama 30 detik. Pemain dan musuh yang dikendalikan oleh AI bersaing untuk mengumpulkan tiga jenis objek yang tersedia. Setiap objek yang berhasil dikumpulkan akan menambah skor masing-masing, sehingga jumlah objek yang dikumpulkan menjadi indikator utama performa dalam permainan. Pemain mengendalikan karakter secara langsung, sedangkan musuh menentukan objek yang akan dikejar secara mandiri berdasarkan mekanisme AI yang telah dirancang. Pada akhir permainan selama 30 detik, skor kedua pihak dibandingkan untuk menentukan pemenang atau kondisi seri.

Flowchart Permainan

Flowchart merupakan representasi grafis yang digunakan untuk menggambarkan urutan proses dan pengambilan keputusan dalam suatu sistem [19]. Dalam penelitian ini, *flowchart* menggambarkan keseluruhan alur permainan *Wayang War*, mulai dari inisialisasi permainan hingga penentuan hasil akhir. *Flowchart* juga

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

menunjukkan interaksi antara pemain dan agen musuh cerdas selama permainan berlangsung. *Flowchart Wayang War* terlampir pada Gambar 2.

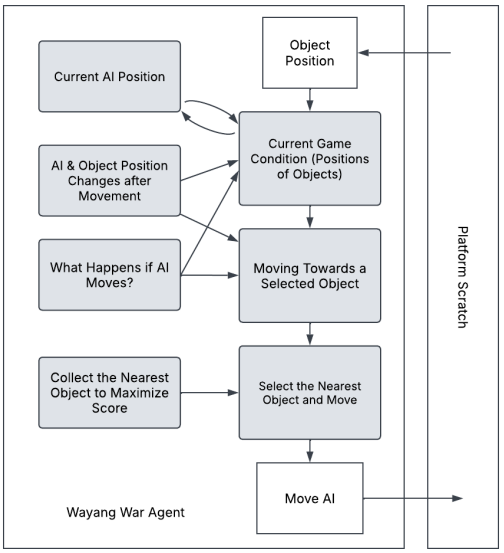


Gambar 2. *Flowchart Wayang War*

Permainan dimulai pada tahap *Start*, kemudian dilanjutkan ke tahap *Initialize*, yaitu proses mempersiapkan variabel dan pengaturan yang diperlukan, termasuk skor pemain, skor musuh, dan timer selama 30 detik. Setelah proses inisialisasi selesai, permainan memasuki tahap *Gameplay*, ketika pemain dan musuh bersaing untuk mengumpulkan objek yang tersedia. Pada tahap tersebut, AI menghitung jarak antara posisi musuh dengan ketiga objek yang tersedia. Selanjutnya, AI memilih objek dengan jarak paling pendek sebagai target. Setelah target ditentukan, musuh bergerak menuju objek yang telah dipilih, sedangkan pemain bergerak secara mandiri untuk mengumpulkan objek. Ketika suatu objek berhasil dicapai, objek tersebut dikumpulkan dan skor pemain atau musuh yang berhasil mengambilnya akan diperbarui. Setelah proses pengumpulan objek, sistem melakukan pemeriksaan *timer* untuk menentukan apakah durasi permainan selama 30 detik telah berakhir. Jika *timer* belum mencapai nol, permainan berlanjut dan AI kembali melakukan proses pemilihan target berdasarkan jarak terdekat. Sebaliknya, jika *timer* telah mencapai nol, *gameplay* berakhir dan sistem melakukan perbandingan skor antara pemain dan musuh. Pihak dengan skor lebih tinggi dinyatakan sebagai pemenang, sedangkan skor yang sama menghasilkan kondisi seri. Setelah hasil ditentukan, proses permainan berakhir pada tahap *End*.

Model Agen Permainan

Agen musuh cerdas dalam *Wayang War* dirancang menggunakan pendekatan *goal-based agent*. Agen berbasis tujuan menentukan tindakan dengan mempertimbangkan kondisi lingkungan saat ini dan menentukan tindakan yang dapat mengarahkannya pada tujuan yang telah ditetapkan. Dalam penelitian ini, tujuan agen AI adalah mengumpulkan objek yang tersedia dengan jarak paling dekat untuk memaksimalkan perolehan skor. Agen secara terus-menerus mengamati posisi objek yang tersedia, mengevaluasi jaraknya dari posisi saat ini, memilih objek terdekat, kemudian bergerak menuju target yang telah dipilih. Model agen yang diusulkan ditunjukkan pada Gambar 3.



Gambar 3. *Goal-Based Intelligent Agent* Model untuk Musuh di *Wayang War*

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Seperti ditunjukkan pada Gambar 3, *Wayang War Agent* beroperasi pada *platform* Scratch dan berinteraksi dengan lingkungan permainan melalui posisi saat ini serta posisi objek yang tersedia. *Object Position* merepresentasikan informasi yang diperoleh dari lingkungan permainan, sedangkan *Current AI Position* merepresentasikan lokasi agen musuh saat ini. Kedua informasi posisi tersebut membentuk *Current Game Condition* yang digunakan agen untuk menentukan tindakan berikutnya. Berdasarkan kondisi permainan saat ini, agen mengevaluasi objek yang tersedia dan menerapkan tujuan untuk mengumpulkan objek terdekat. Agen kemudian memilih objek dengan jarak paling pendek dan bergerak menuju objek tersebut. Target yang dipilih menentukan pergerakan yang dilakukan melalui *Move AI Actuator*. Selama AI bergerak, posisi agen akan berubah dan posisi atau ketersediaan objek juga dapat berubah setelah suatu objek berhasil dikumpulkan. Kondisi permainan yang telah diperbarui kemudian kembali dipersepsikan oleh agen sehingga membentuk siklus pengambilan keputusan yang berlangsung secara berkelanjutan selama *gameplay*. Pendekatan berbasis tujuan memungkinkan musuh mengambil keputusan berdasarkan kondisi permainan saat ini, bukan dengan mengikuti urutan pergerakan yang telah ditentukan sebelumnya. Siklus mengamati lingkungan, menentukan kondisi permainan, memilih target terdekat, dan bergerak menuju target tersebut terus berlangsung hingga permainan berakhir atau tidak terdapat lagi objek yang dapat dikumpulkan [6].

Implementasi Permainan

Permainan *Wayang War* diimplementasikan menggunakan platform pemrograman visual Scratch. Aset permainan dibuat menggunakan Canva dan kemudian diintegrasikan ke dalam Scratch sebagai elemen visual permainan. Permainan terdiri atas *Player*, *Enemy* sebagai lawan yang dikendalikan oleh AI, tiga objek yang dapat dikumpulkan (Item 1, Item 2, dan Item 3), latar belakang, serta tiga tampilan akhir permainan yang terdiri atas *Win, Lose,* dan *Draw.* Implementasi setiap *sprite* dan *Stage*, termasuk fungsi serta logika pemrogramannya, disajikan pada Tabel 2.

Tabel 2. Implementasi Permainan *Wayang War*

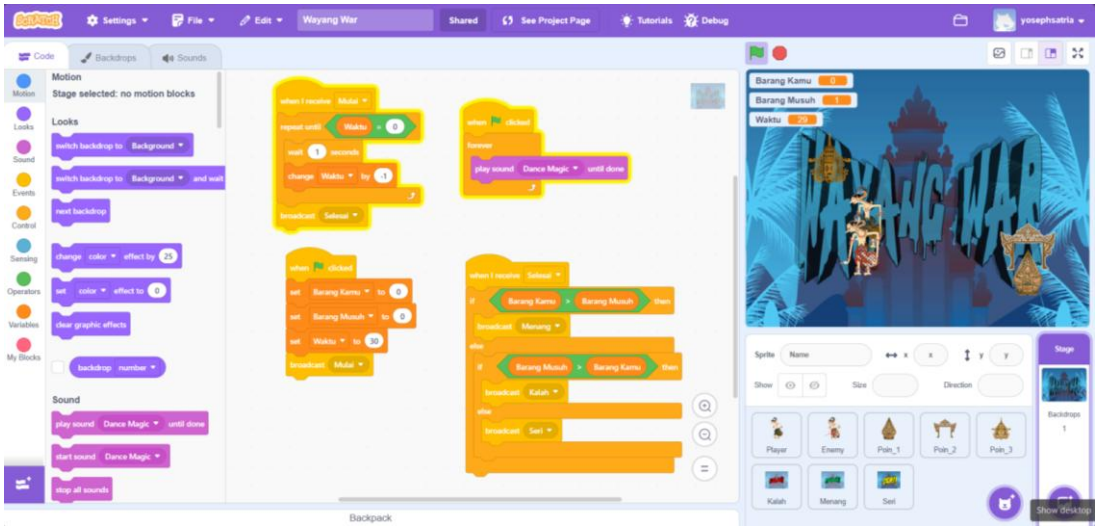
Sprite	**Kategori**	**Blok Kode / Logika**
Stage	*Game Logic*	1. *When green flag clicked*: mengatur Barang Kamu = 0, Barang Musuh = 0, dan Waktu = 30, kemudian melakukan *broadcast* Mulai. 2. *When green flag clicked*: memutar musik latar *Dance Magic* secara terus-menerus. 3. *When I receive Mulai*: mengulangi hingga Waktu = 0 → menunggu 1 detik → mengubah Waktu sebesar -1 → setelah timer selesai melakukan *broadcast* Selesai. 4. *When I receive Selesai*: membandingkan Barang Kamu dan Barang Musuh → melakukan *broadcast* Menang, Kalah, atau Seri sesuai hasil perbandingan skor.
Player	*Player Control*	1. *When green flag clicked*: memeriksa tombol panah secara terus-menerus → kanan: x + 5; kiri: x - 5; atas: y + 5; bawah: y - 5. 2. *When I receive Selesai*: menghentikan script lain pada sprite.
Enemy	*AI Movement*	1. *When I receive Mulai*: mengatur *rotation style* menjadi *don't rotate* → menghitung jarak terhadap Item 1, Item 2, dan Item 3 → memilih item dengan jarak terdekat → mengarah menuju item yang dipilih → bergerak sebanyak 3 langkah. 2. *When I receive Selesai*: menghentikan script lain pada sprite.
Item 1	*Item*	1. *When green flag clicked*: menyembunyikan sprite. 2. *When I receive Mulai*: menampilkan sprite → berpindah ke posisi acak dengan x = -200 hingga 200 dan y = -140 hingga 140. 3. *Forever*: jika menyentuh Player → Barang Kamu + 1 → memainkan suara *Coin* → berpindah ke posisi acak baru. Jika menyentuh Enemy → Barang Musuh + 1 → memainkan suara *Drop* → berpindah ke posisi acak baru.
Item 2	*Item*	Menggunakan logika yang sama dengan Item 1, meliputi penempatan secara acak, deteksi tabrakan dengan Player atau Enemy, penambahan skor, efek suara, dan perubahan posisi setelah objek dikumpulkan.
Item 3	*Item*	Menggunakan logika yang sama dengan Item 1, meliputi penempatan secara acak, deteksi tabrakan dengan Player atau Enemy, penambahan skor, efek suara, dan perubahan posisi setelah objek dikumpulkan.
Win	*End-Game Condition*	1. *When green flag clicked*: menyembunyikan sprite. 2. *When I receive Menang*: menampilkan sprite → memainkan suara *Win* hingga selesai.
Lose	*End-Game Condition*	1. *When green flag clicked*: menyembunyikan sprite. 2. *When I receive Kalah*: menampilkan sprite → memainkan suara *Lose* hingga selesai.
Draw	*End-Game Condition*	1. *When green flag clicked*: menyembunyikan sprite. 2. *When I receive Seri*: menampilkan sprite → memainkan suara *Disconnect* hingga selesai.

Tabel 2 menunjukkan implementasi permainan *Wayang War* menggunakan *platform* Scratch, termasuk fungsi dan logika pemrograman dari setiap *sprite* dan *Stage*. *Stage* berperan sebagai pengendali utama permainan

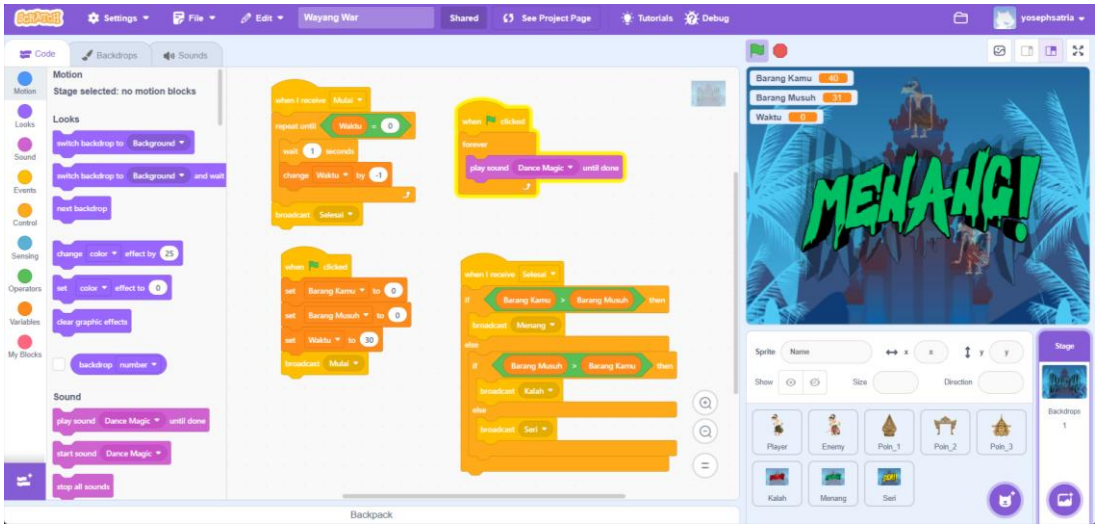
Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

dengan melakukan inisialisasi skor dan *timer* selama 30 detik, memulai *gameplay*, mengatur penghitung waktu, serta menentukan hasil akhir berdasarkan skor *Player* dan *Enemy*. *Player* menerapkan kontrol pergerakan secara langsung menggunakan tombol panah sehingga pemain dapat bergerak di dalam lingkungan permainan dan mengumpulkan objek yang tersedia. Sementara itu, *Enemy* menerapkan mekanisme agen cerdas yang diusulkan dalam penelitian ini. Selama *gameplay* berlangsung, *Enemy* mengevaluasi jarak terhadap Item 1, Item 2, dan Item 3, kemudian memilih item dengan jarak terdekat sebagai target dan bergerak menuju target tersebut. Proses ini dilakukan secara terus-menerus sehingga *Enemy* dapat menyesuaikan target berdasarkan kondisi permainan saat ini. Item 1, Item 2, dan Item 3 berfungsi sebagai objek yang dapat dikumpulkan. Setiap item ditempatkan pada posisi acak dan akan berpindah ke posisi acak lainnya setelah berhasil dikumpulkan oleh *Player* atau *Enemy*. Proses pengumpulan juga meningkatkan skor pihak yang berhasil memperoleh item dan memicu efek suara tertentu. Terakhir, *sprite Win*, *Lose*, dan *Draw* berfungsi sebagai antarmuka akhir permainan yang menampilkan hasil pertandingan setelah *Stage* membandingkan skor akhir.

Hasil implementasi permainan *Wayang War* pada *platform* Scratch ditunjukkan pada Gambar 4 dan Gambar 5. Gambar 4 menunjukkan tampilan *gameplay* yang memperlihatkan *Player* dan *Enemy* berada dalam area permainan bersama tiga objek yang menjadi target pengumpulan. Sistem permainan juga menampilkan informasi skor dan *timer* sebagai indikator kondisi *gameplay*. Sementara itu, Gambar 5 menunjukkan tampilan akhir permainan setelah waktu permainan berakhir dan skor *Player* serta *Enemy* dibandingkan. Tampilan akhir tersebut menampilkan salah satu kondisi hasil permainan, yaitu *Win*, sesuai dengan hasil perolehan skor masing-masing pihak.



Gambar 4. Tampilan *Gameplay*



Gambar 5. Tampilan Kondisi *Win*

Black Box Testing

Pengujian *black-box* dilakukan untuk memverifikasi fungsionalitas permainan *Wayang War* berdasarkan input dan output yang dapat diamati tanpa memeriksa kode sumber internal [20]. Pengujian difokuskan pada fungsi utama permainan, meliputi inisialisasi permainan, timer, audio, perilaku item, pergerakan *Player*, perilaku AI, perubahan skor, serta kondisi *Win, Lose,* dan *Draw*. Setiap pengujian dilakukan dengan memberikan *input* atau

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

kondisi permainan tertentu, kemudian mengamati apakah perilaku yang dihasilkan sesuai dengan hasil yang diharapkan. Hasil pengujian *black-box* secara lengkap disajikan pada Tabel 3.

Tabel 3. Hasil *Black Box Testing*

No.	Komponen yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	*Green Flag* / Inisialisasi Permainan	Klik *green flag* untuk memulai permainan.	Permainan melakukan inisialisasi dengan benar, *Player* dan *Enemy* muncul, skor diatur menjadi 0, dan *timer* diatur menjadi 30 detik.	*Pass*
2	Memulai Permainan	Setelah *green flag* diklik, amati kondisi permainan.	*Event* Mulai dijalankan dan *gameplay* dimulai.	*Pass*
3	*Timer*	Memulai permainan dan mengamati *timer*.	*Timer* dimulai dari 30 detik dan berkurang satu detik hingga mencapai 0.	*Pass*
4	Audio Latar	Memulai permainan dan mengamati musik latar.	Audio Dance Magic diputar selama *gameplay*.	*Pass*
5	Pergerakan *Player*	Menekan tombol panah kanan, kiri, atas, dan bawah.	*Player* bergerak sesuai dengan arah tombol yang ditekan.	*Pass*
6	Item 1	Memulai permainan dan mengamati Item 1.	Item 1 muncul pada posisi acak di dalam area permainan.	*Pass*
7	Item 1 Dikumpulkan *Player*	Menggerakkan *Player* hingga menyentuh Item 1.	Barang Kamu bertambah 1, suara pengumpulan diputar, dan Item 1 berpindah ke posisi acak yang baru.	*Pass*
8	Item 1 Dikumpulkan *Enemy*	Membiarkan *Enemy* mencapai Item 1.	Barang Musuh bertambah 1, suara yang sesuai diputar, dan Item 1 berpindah ke posisi acak yang baru.	*Pass*
9	Item 2	Memulai permainan dan mengamati Item 2 serta interaksinya.	Item 2 muncul secara acak dan dapat dikumpulkan oleh *Player* atau *Enemy* menggunakan mekanisme yang sama dengan Item 1.	*Pass*
10	Item 3	Memulai permainan dan mengamati Item 3 serta interaksinya.	Item 3 muncul secara acak dan dapat dikumpulkan oleh *Player* atau *Enemy* menggunakan mekanisme yang sama dengan Item 1.	*Pass*
11	Pemilihan Target AI	Memulai permainan dengan tiga item yang memiliki jarak berbeda dari *Enemy*.	*Enemy* mengidentifikasi dan memilih item yang tersedia dengan jarak paling dekat sebagai target.	*Pass*
12	Pergerakan AI	Mengamati *Enemy* setelah memilih target.	*Enemy* bergerak menuju item terdekat yang telah dipilih.	*Pass*
13	Pemilihan Ulang Target AI	Mengubah kondisi permainan dengan mengumpulkan atau memindahkan suatu item.	*Enemy* mengevaluasi posisi objek yang telah diperbarui dan memilih item terdekat yang tersedia sebagai target baru.	*Pass*
14	Pengumpulan Item oleh AI	Membiarkan *Enemy* mencapai target yang telah dipilih.	*Enemy* mengumpulkan item dan Barang Musuh bertambah 1.	*Pass*
15	Akhir *Gameplay*	Membiarkan *timer* mencapai 0.	*Gameplay* berhenti dan sistem membandingkan skor *Player* dan *Enemy*.	*Pass*
16	Kondisi Menang	Mengakhiri permainan dengan kondisi Barang Kamu > Barang Musuh.	Layar *Win* ditampilkan dan suara *Win* diputar.	*Pass*
17	Kondisi Kalah	Mengakhiri permainan dengan kondisi Barang Musuh > Barang Kamu.	Layar *Lose* ditampilkan dan suara *Lose* diputar.	*Pass*
18	Kondisi Seri	Mengakhiri permainan dengan kondisi Barang Kamu = Barang Musuh.	Layar *Draw* ditampilkan dan suara Disconnect diputar.	*Pass*
19	Terminasi Permainan	Memicu salah satu hasil akhir setelah timer mencapai 0.	Permainan berhenti dan tidak terdapat pergerakan *Player* maupun *Enemy* selanjutnya.	*Pass*

Tabel 3 menunjukkan hasil pengujian *black-box* yang dilakukan pada permainan *Wayang War*. Pengujian mencakup keseluruhan fungsi *gameplay*, dimulai dari inisialisasi permainan melalui *green flag*, dilanjutkan dengan aktivasi permainan dan timer selama 30 detik. Audio latar juga diuji untuk memastikan permainan memberikan umpan balik audio sesuai dengan rancangan. Pergerakan *Player* diuji menggunakan empat tombol panah untuk memastikan karakter dapat bergerak sesuai dengan arah yang diberikan.

Ketiga objek yang dapat dikumpulkan diuji untuk memverifikasi kemunculan, penempatan secara acak, proses pengumpulan, penambahan skor, efek suara, dan perubahan posisi setelah objek dikumpulkan. Item 2 dan Item 3 menerapkan mekanisme pengumpulan yang sama dengan Item 1. Perilaku AI diuji dengan mengamati

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

kemampuannya dalam mengidentifikasi tiga objek yang tersedia, menghitung jaraknya, memilih objek terdekat sebagai target, dan bergerak menuju target tersebut. AI juga diuji setelah terjadi perubahan kondisi permainan untuk memastikan bahwa agen dapat memilih target baru berdasarkan jarak yang telah diperbarui.

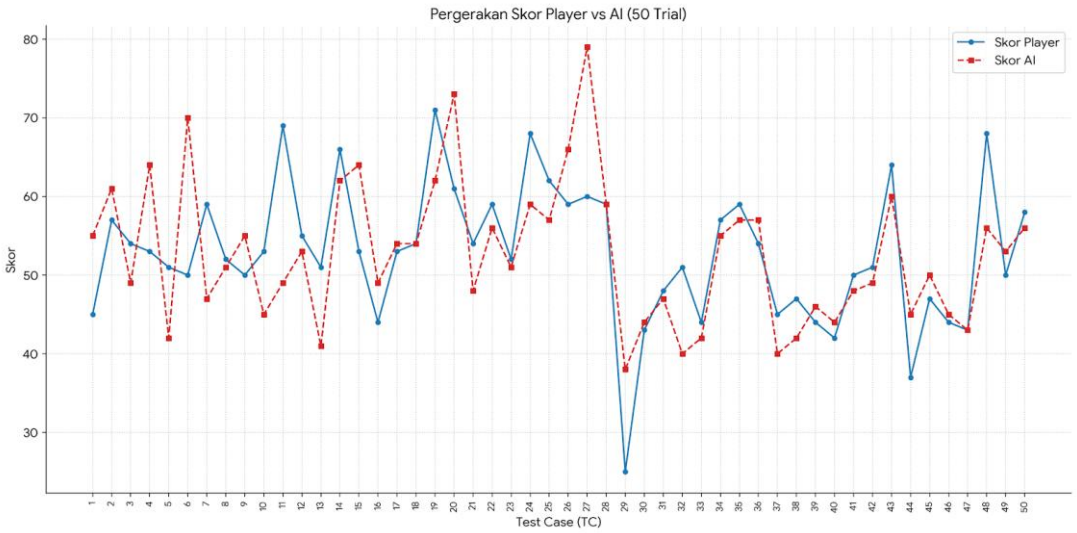
Pengujian terakhir mengevaluasi mekanisme akhir permainan, meliputi kondisi timer mencapai nol, perbandingan skor, serta tampilan *Win, Lose,* dan *Draw*. Efek suara yang terkait dengan masing-masing hasil juga diverifikasi. Dari 19 skenario pengujian, seluruh skenario menghasilkan keluaran yang sesuai dengan hasil yang diharapkan dan dikategorikan sebagai Pass. Dengan demikian, pengujian *black-box* memperoleh tingkat keberhasilan sebesar 100%, yang dihitung menggunakan persamaan berikut:

$$Success\ Rate = \frac{19}{19} \times 100\% = 100\%$$

Hasil tersebut menunjukkan bahwa seluruh fungsi *gameplay* yang diimplementasikan, termasuk mekanisme agen musuh cerdas, telah berjalan sesuai dengan rancangan dan perilaku yang diharapkan.

Gameplay Testing

Pengujian *gameplay* dilakukan untuk mengevaluasi performa agen musuh cerdas selama sesi permainan berlangsung. Pengujian dilakukan melalui 50 kali percobaan *gameplay*, dengan setiap percobaan mencatat skor akhir yang diperoleh *Player* dan AI *Enemy* setelah periode permainan selama 30 detik. Kondisi akhir pada setiap percobaan dikategorikan menjadi *Win, Lose,* atau *Draw* berdasarkan perbandingan skor antara *Player* dan AI. Hasil lengkap pengujian *gameplay* disajikan pada Gambar 6.



Gambar 6. Hasil *Gameplay Testing*

Gambar 6 menunjukkan hasil pengujian *gameplay* yang dilakukan dalam 50 kali percobaan. Setiap percobaan mencatat skor akhir yang diperoleh *Player* dan AI setelah periode permainan selama 30 detik. Berdasarkan perbandingan skor, *Player* memperoleh 27 kemenangan, 20 kekalahan, dan 3 hasil seri. Hasil tersebut menunjukkan bahwa *Player* memperoleh skor lebih tinggi pada lebih banyak percobaan, sementara AI juga mampu memperoleh beberapa kemenangan. Hal ini menunjukkan bahwa agen musuh cerdas mampu memberikan persaingan selama permainan berlangsung. Persentase setiap kondisi dihitung berdasarkan total 50 percobaan *gameplay*. Persentase kemenangan *Player* dihitung sebagai berikut:

$$Win\ Rate = \frac{27}{50} \times 100\% = 54\%$$

Persentase kekalahan *Player* adalah:

$$Loss\ Rate = \frac{20}{50} \times 100\% = 40\%$$

Sementara itu, persentase hasil seri adalah:

$$Draw\ Rate = \frac{3}{50} \times 100\% = 6\%$$

Dengan demikian, *Player* memperoleh tingkat kemenangan sebesar 54%, sedangkan 40% percobaan berakhir dengan kekalahan dan 6% berakhir seri. Tingkat kemenangan *Player* lebih tinggi 14 poin persentase dibandingkan tingkat kekalahannya. Meskipun demikian, AI masih mampu memenangkan 20 dari 50 percobaan, atau sebesar

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

40% dari keseluruhan percobaan. Hal ini menunjukkan bahwa agen musuh cerdas mampu menghasilkan kondisi permainan yang kompetitif dan tidak selalu mengalami kekalahan ketika berhadapan dengan *Player*. Tiga hasil seri yang terjadi pada percobaan ke-18, ke-28, dan ke-47 menunjukkan bahwa AI mampu memperoleh skor yang sama dengan *Player* pada beberapa sesi permainan. Hasil tersebut semakin menunjukkan bahwa mekanisme pemilihan target berdasarkan jarak dapat menghasilkan kondisi permainan yang berbeda-beda bergantung pada perubahan posisi *Player*, AI, dan objek yang dapat dikumpulkan selama *gameplay* berlangsung. Selanjutnya, skor *gameplay* dianalisis menggunakan statistik deskriptif untuk mengamati distribusi dan variasi skor *Player* dan AI. Hasil analisis tersebut disajikan pada Tabel 4.

Tabel 4. Statistik Deskriptif dan Hasil *Paired-Samples t-Test*

Statistik	***Player***	***AI***	**Perbandingan *Player–AI***
N	50	50	—
Minimum	25,00	38,00	—
Maximum	71,00	79,00	—
Range	46,00	41,00	—
Sum	2645,00	2632,00	—
Mean	52,9000	52,6400	0,26000
Std. Deviation	8,71604	9,01418	7,89422
Variance	75,969	81,256	—
Skewness	-0,314	0,693	—
Kurtosis	1,141	0,399	—
t	—	—	0,233
df	—	—	49
Sig. (2-tailed)	—	—	0,817
95% CI	—	—	-1,98351 to 2,50351
Cohen's *dz*	—	—	0,0329

Tabel 4 menunjukkan statistik deskriptif skor yang diperoleh *Player* dan AI dari 50 percobaan *gameplay*. Kedua kelompok memiliki N = 50, dengan rata-rata skor *Player* sebesar 52,9000 dan AI sebesar 52,6400, sehingga terdapat selisih mean sebesar 0,2600 poin. *Player* memperoleh total skor 2645, sedangkan AI memperoleh 2632. Dari sisi variasi, *Player* memiliki skor minimum 25 dan maksimum 71 dengan *range* 46, sedangkan AI memiliki skor minimum 38 dan maksimum 79 dengan *range* 41. Standar deviasi *Player* sebesar 8,71604 dan AI sebesar 9,01418 menunjukkan bahwa variasi skor AI sedikit lebih besar. Nilai skewness dan kurtosis juga menunjukkan adanya perbedaan pola distribusi skor antara kedua pihak. Untuk menguji apakah perbedaan rata-rata skor tersebut signifikan secara statistik, dilakukan *paired-samples t-test* karena skor *Player* dan AI berasal dari pasangan pertandingan yang sama pada setiap percobaan. Hasil pengujian menunjukkan rata-rata perbedaan skor sebesar 0,26000 dengan standar deviasi perbedaan sebesar 7,89422. Nilai $t(49) = 0,233$ dengan $p = 0,817$ dan interval kepercayaan 95% sebesar [-1,98351; 2,50351]. Karena nilai $p > 0,05$, tidak terdapat perbedaan yang signifikan secara statistik antara skor *Player* dan AI pada 50 percobaan *gameplay*. Hasil tersebut menunjukkan bahwa meskipun *Player* memiliki rata-rata skor sedikit lebih tinggi daripada AI, perbedaan tersebut belum memberikan bukti statistik yang cukup untuk menyatakan adanya perbedaan performa rata-rata antara keduanya. Temuan ini melengkapi hasil pengujian *gameplay*, yaitu *Player* memperoleh *win rate* sebesar 54%, AI sebesar 40%, dan hasil seri sebesar 6%. Dengan demikian, mekanisme *goal-based agent* dengan pemilihan target terdekat berdasarkan jarak Euclidean menghasilkan agen yang dapat berpartisipasi secara kompetitif dalam *gameplay*.

Distribusi

Pada tahap diseminasi, game *Wayang War* yang telah dikembangkan didistribusikan sebagai *game* Scratch berbasis web sehingga dapat diakses dan dimainkan secara langsung melalui peramban web. Permainan ini dapat diakses secara publik melalui s.pnj.ac.id/WayangWar. Distribusi ini memungkinkan pengguna untuk memainkan *Wayang War* tanpa memerlukan instalasi perangkat lunak tambahan serta mendukung penyebarluasan agen musuh cerdas yang telah diimplementasikan.

Diskusi

Hasil pengujian *gameplay* menunjukkan bahwa agen musuh cerdas mampu melakukan pemilihan target secara mandiri dan mengumpulkan objek dalam game *Wayang War*. Dari 50 percobaan, *Player* memperoleh tingkat kemenangan sebesar 54%, sedangkan AI memenangkan 40% percobaan dan menghasilkan 6% hasil seri. Statistik deskriptif menunjukkan bahwa rata-rata skor *Player* (52,90) dan AI (52,64) memiliki selisih sebesar 0,26 poin. Hasil *paired-samples t-test* menunjukkan bahwa perbedaan tersebut tidak signifikan secara statistik, dengan $t(49) = 0,233$, $p = 0,817$, dan interval kepercayaan 95% sebesar [-1,98; 2,50]. Ukuran efek berdasarkan *Cohen's d* untuk data berpasangan *Cohen's dz* sebesar 0,0329, yang menunjukkan bahwa perbedaan rata-rata skor yang diamati memiliki magnitudo yang sangat kecil. Dengan demikian, hasil pengujian tidak memberikan bukti yang cukup mengenai perbedaan performa rata-rata antara *Player* dan AI pada kondisi eksperimen yang dilakukan.

Mekanisme pemilihan target berdasarkan jarak Euclidean dalam penelitian ini memang relatif sederhana dibandingkan metode AI yang lebih kompleks. Namun, penerapannya dapat dikategorikan sebagai bentuk agen

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

berbasis tujuan (*goal-based agent*) karena agen memiliki tujuan yang jelas, yaitu mengumpulkan objek untuk memperoleh skor, mengamati kondisi lingkungan melalui posisi objek dan posisi dirinya, mengevaluasi beberapa alternatif target, kemudian menentukan tindakan berdasarkan target yang paling sesuai dengan tujuan tersebut. Dalam implementasinya, jarak Euclidean berfungsi sebagai mekanisme evaluasi untuk menentukan target terdekat, sedangkan keputusan untuk bergerak menuju target merupakan tindakan agen. Ketika objek dikumpulkan dan berpindah posisi, kondisi lingkungan berubah sehingga agen melakukan evaluasi dan pemilihan target kembali. Dengan demikian, kecerdasan pada penelitian ini tidak diklaim berasal dari kompleksitas algoritmanya, tetapi dari kemampuan agen untuk memetakan kondisi lingkungan dan tujuan permainan menjadi keputusan tindakan secara otonom.

Temuan ini memperluas penelitian sebelumnya mengenai pengembangan permainan berbasis Scratch. Penelitian [11], [12], [13] menunjukkan bahwa Scratch dapat digunakan untuk mengembangkan game interaktif yang mencakup mekanisme *gameplay*, interaksi pengguna, dan konten edukatif, sedangkan [14] menunjukkan penerapan algoritma komputasional untuk interaksi *real-time* antarobjek dalam *game.* Sementara itu, penelitian [10] secara khusus menyoroti integrasi AI dan *game-based learning* menggunakan Scratch dan Unity. Berbeda dengan penelitian tersebut, penelitian ini berfokus pada implementasi agen musuh dengan pengambilan keputusan berbasis tujuan dalam skenario permainan kompetitif. Agen tidak mengikuti urutan target yang telah ditentukan, tetapi mengevaluasi jarak terhadap beberapa objek dan memilih target terdekat berdasarkan kondisi lingkungan. Hal ini memberikan contoh implementasi sederhana *goal-based decision-making* pada Scratch untuk menghasilkan perilaku musuh secara mandiri.

Hasil penelitian memberikan kontribusi praktis dalam eksplorasi mekanisme permainan berbasis AI pada lingkungan pemrograman visual. Tingkat keberhasilan 100% pada pengujian *black-box* mengonfirmasi bahwa fungsi *game* dan perilaku AI berjalan sesuai dengan spesifikasi yang dirancang, sedangkan pengujian *gameplay* menunjukkan bahwa agen dapat berkompetisi dengan *Player*. Meskipun rata-rata skor *Player* sedikit lebih tinggi, interval kepercayaan perbedaan [-1,98; 2,50] dan ukuran efek yang sangat kecil (*Cohen's dz* = 0,0329) menunjukkan bahwa selisih 0,26 poin memiliki makna praktis yang terbatas dalam kondisi pengujian ini. Oleh karena itu, penelitian ini tidak menyatakan bahwa AI lebih unggul daripada *Player*, tetapi menunjukkan bahwa mekanisme *goal-based agent* dengan pemilihan target terdekat berbasis jarak Euclidean dapat diimplementasikan sebagai pendekatan sederhana untuk menghasilkan perilaku musuh yang otonom dan kompetitif pada Scratch.

4. SIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan agen musuh berbasis AI pada *platform* Scratch menggunakan pendekatan berbasis tujuan (*goal-based*) dengan mekanisme pemilihan target terdekat berdasarkan jarak Euclidean. Agen mampu mengamati posisi objek yang tersedia, menentukan objek dengan jarak terdekat, dan bergerak menuju objek tersebut secara mandiri selama *gameplay*. Pengujian *black-box* memperoleh tingkat keberhasilan sebesar 100% pada 19 skenario pengujian, yang menunjukkan bahwa fungsi permainan dan perilaku agen berjalan sesuai dengan spesifikasi yang telah ditentukan. Hasil dari 50 percobaan *gameplay* menunjukkan bahwa *Player* memperoleh tingkat kemenangan sebesar 54%, sedangkan AI memperoleh tingkat kemenangan sebesar 40% dan menghasilkan 6% hasil seri. Rata-rata skor *Player* sebesar 52,90 dan AI sebesar 52,64, dengan selisih rata-rata sebesar 0,26. Hasil *paired-samples t-test* menunjukkan bahwa perbedaan tersebut tidak signifikan secara statistik, dengan $t(49) = 0{,}233$ dan $p = 0{,}817$, serta interval kepercayaan 95% sebesar [-1,98; 2,50]. Ukuran efek *Cohen's dz* sebesar 0,0329 juga menunjukkan bahwa magnitudo perbedaan skor sangat kecil. Hasil tersebut menunjukkan bahwa agen yang diimplementasikan mampu menjalankan mekanisme pengambilan keputusan berbasis tujuan dan menghasilkan performa permainan yang tidak menunjukkan perbedaan rata-rata yang signifikan dibandingkan *Player* dalam kondisi pengujian yang dilakukan.

Penelitian ini memiliki keterbatasan pada penggunaan mekanisme pemilihan target yang relatif sederhana dan hanya didasarkan pada jarak Euclidean, tanpa mempertimbangkan faktor lain seperti posisi *Player*, nilai objek, selisih skor, maupun kondisi permainan pada masa mendatang. Penelitian ini juga belum menggunakan konfigurasi *baseline* berupa agen dengan strategi pengambilan keputusan yang lebih sederhana, sehingga kontribusi spesifik mekanisme pemilihan target berdasarkan jarak Euclidean belum dapat dibandingkan secara langsung dengan strategi lainnya. Meskipun penelitian telah menerapkan analisis statistik inferensial, jumlah percobaan masih terbatas pada 50 *gameplay* sehingga generalisasi hasil perlu dilakukan secara hati-hati. Penelitian selanjutnya dapat meningkatkan kemampuan agen dengan mempertimbangkan variabel kondisi permainan tambahan, menerapkan metode pengambilan keputusan atau algoritma *pathfinding* yang lebih lanjut, serta menambahkan beberapa konfigurasi agen sebagai *baseline*. Penambahan jumlah percobaan juga dapat memberikan dasar evaluasi yang lebih luas terhadap perilaku dan performa agen.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

DAFTAR PUSTAKA

[1] G. Zeng, "A review of AI-Based Game NPCs Research," *Applied and Computational Engineering*, vol. 15, no. 1, pp. 155–159, Oct. 2023, doi: 10.54254/2755-2721/15/20230827.

[2] R. Firoz and Nikita, "A Comprehensive Review of AI-Driven NPC Intelligence and Procedural Content Generation in Video Games," in *2026 9th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, Apr. 2026, pp. 920–927. doi: 10.1109/ICOEI68323.2026.11541480.

[3] R. R. Karthigasu, M. B. Jasser, S. M. Ajibade, S. Abdipoor, and H. N. Chua, "A Review on Artificial Intelligence Techniques Employed for Enhancing NPC Behavior in Video Games," in *2026 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, IEEE, Jun. 2026, pp. 221–226. doi: 10.1109/I2CACIS69435.2026.11600010.

[4] V. Levytskyi, M. Tsiutsiura, A. Yerukaiev, N. Rusan, and T. Li, "The Working Principle of Artificial Intelligence in Video Games," in *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, IEEE, May 2023, pp. 246–250. doi: 10.1109/SIST58284.2023.10223491.

[5] N. Abdullasim, "Adaptive AI Behaviour in Stealth NPCs: A Narrative Review of Current Approaches and Research Directions," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 15, no. 01, pp. 320–331, Jun. 2026, doi: 10.17576/apjitm-2026-1501-20.

[6] Y. S. Praka, C. R. T. Prameswari, N. Wahyuni, and A. Maria, "Implementation of a Pathfinding-Based Intelligent Agent for Enemy Behavior in the 'Run to BIPOL' Maze Tournament Game," *Journal of Games Game Art and Gamification*, vol. 11, no. 2, p. 38, Aug. 2026, doi: 10.21512/jggag.v11i2.16155.

[7] P. Dúo-Terrón, "Analysis of Scratch Software in Scientific Production for 20 Years: Programming in Education to Develop Computational Thinking and STEAM Disciplines," *Education Sciences*, vol. 13, no. 4, p. 404, Apr. 2023, doi: 10.3390/educsci13040404.

[8] B. Batni, S. N. Junaini, J. Sidi, W. A. Mustafa, and Z. I. Awang Ismail, "Current Research Trends of Scratch Block based Programming for K-12: A Systematic Review," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 51, no. 2, pp. 138–152, Sep. 2024, doi: 10.37934/araset.51.2.138152.

[9] G. C. Winarko and A. N. Cahyono, "Project-Based Learning with Scratch to Improve Students' Creative Thinking Ability: Systematic Literature Review," *Griya Journal of Mathematics Education and Application*, vol. 4, no. 2, pp. 190–196, Jun. 2024, doi: 10.29303/griya.v4i2.440.

[10] U. Chotijah, H. D. Bhakti, D. Saputra, M. V. S. Pratama, and M. D. Anwar, "Integrasi Artificial Intelligence dan Game-Based Learning dalam Pembelajaran Matematika Menggunakan Unity dan Scratch," *SAMBARA: Jurnal Pengabdian Kepada Masyarakat*, vol. 4, no. 2, pp. 405–420, May 2026, doi: 10.58540/sambarapkm.v4i2.1702.

[11] J. Juanda, Z. Zikrillah, and N. Nasrullah, "Building a Space Shooter Game: Utilizing Scratch for an Engaging Action Game," *Gameology and Multimedia Expert*, vol. 1, no. 4, pp. 90–94, Oct. 2024, doi: 10.29103/game.v1i4.18857.

[12] A. F. A. A. Kusuma and F. Syah, "Maze Game Design Using Scratch 3 Android Based," *Jurnal Dinamika Informatika*, vol. 11, no. 1, pp. 100–113, 2022, doi: 10.31316/jdi.v11i1.139.

[13] R. A. E. Perkasa and J. Wantoro, "Game Edukasi Interaktif Sejarah Kerajaan Hindu-Buddha menggunakan Platform Scratch," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 1, pp. 36–45, Jun. 2024, doi: 10.29408/edumatic.v8i1.25161.

[14] F. Ardiansyah, A. Kharisma Hidayah, A. Wijaya, and M. Imanullah, "Pengembangan Game Up 2 Down Birdie Melalui Integrasi Algoritma Circle Collision Detection Menggunakan Platform Scratch," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 10, no. 3, pp. 4353–4358, May 2026, doi: 10.36040/jati.v10i3.18028.

[15] A. R. Pangestu, S. Sugiyanto, T. Pinandita, and D. K. Hakim, "Game Edukasi Ilmu Pengetahuan Sosial (IPS) Berbasis Android menggunakan Metode R&D 4D di SMP Negeri 2 Rawalo," *Jurnal Teknologi Informasi dan Multimedia*, vol. 8, no. 2, pp. 382–393, May 2026, doi: 10.35746/jtim.v8i2.990.

[16] I. Sonjaya, M. Rosalina, T. D. Ariyanti, A. S. Azmi, and D. A. Martono, "Development Of Educational Media Based On Virtual Reality Multiplayer Games For Basic Communication Learning For Students With Intellectual Disabilities," in *2025 8th International Conference of Computer and Informatics Engineering (IC2IE)*, IEEE, Sep. 2025, pp. 1–6. doi: 10.1109/IC2IE67206.2025.11283385.

[17] N. Marcheta and L. K. P. H. Hartanto, "Pengembangan Game Edukasi 3D 'MathRoom' Sebagai Media Pembelajaran Bilangan Pecahan Matematika Siswa Kelas 5 Sekolah Dasar Menggunakan Unity Engine," *MULTINETICS*, vol. 10, no. 1, pp. 21–30, Jul. 2024, doi: 10.32722/multinetics.v10i1.5100.

[18] M. F. R. Fauzian, R. A. Krisdiawan, and N. A. Asikin, "Gamified English Vocabulary Puzzle Game with FSM-Based Emotional Feedback and Fisher–Yates Shuffle," *Media Jurnal Informatika*, vol. 18, no. 1, pp. 75–89, Jun. 2026, doi: 10.35194/mji.v18i1.6402.

[19] I. Al Mahdi, S. Bukhori, and M. A. Furqon, "Implementation of Finite State Machine in Flowchart-Based Visual Programming Game," *Journal of Games, Game Art, and Gamification*, vol. 10, no. 3, pp. 93–105, Dec. 2025, doi: 10.21512/jggag.v10i3.12133.

[20] I. Benawan, D. M. K. Nugraheni, B. Noranita, and G. Aryotejo, "Digital Education Game for TK-A Level Students Using Multimedia Development Life Cycle Method," *INTENSIF: Jurnal Ilmiah Penelitian dan*

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Penerapan Teknologi Sistem Informasi, vol. 7, no. 1, pp. 68–83, Feb. 2023, doi: 10.29407/intensif.v7i1.18671.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi