

IMPLEMENTASI KONTROL SUARA DAN ALGORITMA A PADA GAME *PLATFORMER* 2D “KEBO BLAYON” MENGUNAKAN UNITY

IMPLEMENTATION OF VOICE CONTROL AND A ALGORITHM IN 2D *PLATFORMER* GAME “KEBO BLAYON” USING UNITY

Ahmad Dhani¹, Ellen Proborini², Fajar Husein Asy'ari³

Prodi Informatika Sekolah Tinggi Teknik Pati^{1,2,3}

e-mail: Ahmadino16@gmail.com¹, ellena@sttp.ac.id², fajarhusain@sttp.ac.id³

Abstract : *The advancement of sensor technology, particularly voice sensors, offers innovative opportunities in the development of games. This study aims to design and develop a 2D platformer game titled Kebo Blayon, which implements voice control as the main interaction method and the A* algorithm for enemy (NPC) navigation. The game is developed using Unity and targeted for Android devices, following the Game Development Life Cycle (GDLC) method consisting of six development stages, from initialization to release. Voice input is utilized to control the main character—soft sounds trigger movement while louder sounds trigger jumping actions. Meanwhile, the A* algorithm is implemented to compute the optimal path for enemy characters to reach the player's position. The testing results show that the voice control system functions responsively and the A* algorithm efficiently guides NPCs. Design and prototype validation by experts and general users both yielded feasibility ratings above 90%. Therefore, the implementation of voice control and the A* pathfinding algorithm in Kebo Blayon proves effective in enhancing interactive and inclusive gameplay within the 2D platformer genre.*

Keywords: *2D platformer game, voice control, A* algorithm, Unity, GDLC.*

Abstrak Perkembangan teknologi sensor, khususnya sensor suara, membuka peluang inovatif dalam pengembangan *game*. Penelitian ini bertujuan untuk merancang dan mengembangkan *game platformer 2D* berjudul *Kebo Blayon* yang mengimplementasikan kontrol suara sebagai media interaksi utama serta algoritma A* untuk navigasi karakter musuh (NPC). *Game* dikembangkan menggunakan *Unity* dan ditujukan untuk perangkat *Android* dengan menerapkan metode *Game Development Life Cycle (GDLC)* yang terdiri dari enam tahap pengembangan, mulai dari inisialisasi hingga rilis. Kontrol suara digunakan untuk menggerakkan karakter utama, di mana suara pelan digunakan untuk berjalan dan suara keras untuk melompat. Sementara itu, algoritma A* diimplementasikan untuk menghitung lintasan optimal karakter musuh menuju posisi pemain. Hasil pengujian menunjukkan bahwa kontrol suara berfungsi secara responsif dan algoritma A* berhasil mengarahkan NPC secara efisien. Validasi yang dilakukan terhadap aspek desain dan prototipe memperoleh tingkat kelayakan lebih dari 90%, baik dari ahli maupun responden umum. Dengan demikian, penerapan kontrol suara dan algoritma A* pada *game Kebo Blayon* terbukti efektif dalam meningkatkan pengalaman bermain yang interaktif dan inklusif dalam genre *platformer 2D*.

Kata Kunci: Game platformer 2D, kontrol suara, algoritma A*, Unity, GDLC.

PENDAHULUAN

Perkembangan teknologi digital pada beberapa dekade terakhir mengalami kemajuan pesat [1]. Teknologi sensor adalah salah satu teknologi yang mengalami perkembangan. Perkembangan teknologi sistem audio digital ini dapat mendorong kemajuan teknologi di Indonesia [2]. Sensor suara sendiri bukan teknologi yang baru, tetapi masih jarang digunakan untuk implementasi ke dalam *game*. Suara lebih banyak digunakan untuk mengisi latar belakang *game* dan juga setiap gerakan pemain di dalam *game*. Teknologi sensor suara ini membuka peluang untuk pengembangan *game* yaitu untuk menambahkan inovasi dalam interaksi pemain dengan *game*.

Game memiliki banyak *genre* seperti *adventure*, *action*, *RPG*, dan masih banyak lainnya. *Genre* yang populer dalam dunia *game* salah satunya adalah *genre platformer 2d* yang menawarkan pengalaman bermain sederhana namun menarik [3]. *Platform game* atau *platformer* adalah *video game* yang tantangan utamanya adalah mengarahkan atau mengendalikan suatu karakter melewati peron peron yang terpisah, biasanya dengan melompati peron [4]. *Genre platformer 2d* terkenal dengan *gameplay* atau permainannya, pemain harus melewati rintangan, mengambil koin atau item tertentu, hingga menghindari musuh untuk mendapatkan nilai atau skor agar dapat melewati *level* permainannya untuk membuka *level* berikutnya. *Game platformer 2d* yang populer contohnya adalah *Super Mario Bros* dan *Metal slug* [5].

Pengembangan *game platformer 2d* masih banyak menggunakan kontrol tombol atau *joystick*, hal tersebut memerlukan ketangkasan dan ketrampilan tangan. Bagi beberapa pemain kontrol tombol atau *joystick* dapat menghambat dan mengurangi pengalaman bermain, terutama bagi beberapa pemain yang memiliki keterbatasan fisik. Penerapan teknologi sensor suara dapat mengatasi kendala yang dihadapi pemain yang kesulitan menggunakan kontrol tombol atau *joystick* [6]. Sensor suara adalah teknologi yang memungkinkan pengenalan suara atau frekuensi suara sebagai *input* untuk interaksi di dalam *game* [7]. Frekuensi suara yang di-*input* melalui *microphone* akan menentukan gerak atau mengontrol karakter di dalam *game platformer 2d* seperti berjalan atau melompat untuk melewati rintangan.

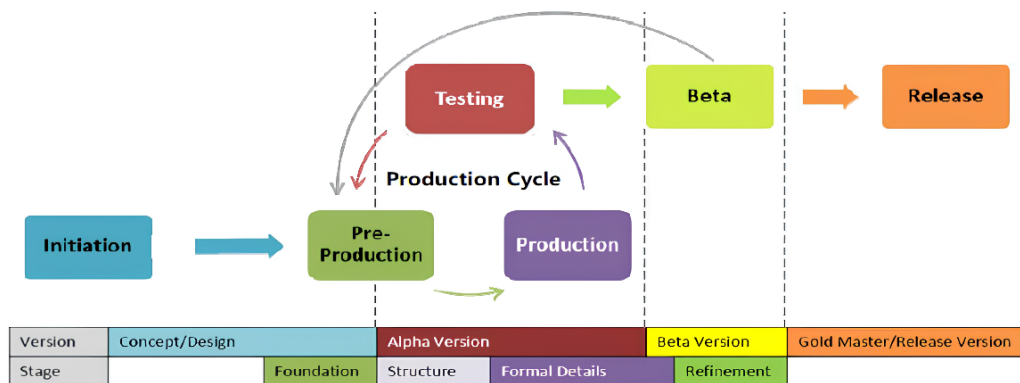
Algoritma A* (*A-Star*) merupakan algoritma pencarian jalur yang sangat signifikan dalam pengembangan *game platformer 2d*, khususnya untuk mengoptimalkan pergerakan karakter maupun *NPC (NonPlayable Character)* [8]. Algoritma A* merupakan algoritma pencarian yang menghitung rute paling efisien [9]. Algoritma ini juga lebih efisien dibandingkan dengan pencarian blind seperti BFS atau DFS karena menggunakan informasi heuristik untuk memandu pencarian [10].

Berdasarkan uraian tersebut, penulis bermaksud merancang dan mengembangkan *game platformer 2d* berjudul *Kebo Blayon*, yang ber-*genre platformer 2d*. Judul "*Kebo Blayon*" terinspirasi dari ciri khas atau maskot dari kota Pati sendiri yaitu hewan kerbau atau lebih dikenal dengan nama atau sebutan "*kebo landoh*".

METODE PENELITIAN

Penelitian ini menggunakan metode *Game Development Life Cycle (GDLC)* yaitu sebuah pengembangan dari sebuah *game* yang lebih mengutamakan aspek interaktif yang memiliki enam fase pengembangan, dimulai dari fase inialisasi/pembuatan konsep, *preproduction*, *production*, *testing*, *beta* dan *release* [11]. Tata cara *GDLC* dapat berbeda-beda tergantung pada metode pengembangan yang diterapkan, namun pada umumnya melibatkan tahap pra-produksi, produksi, dan pasca-produksi. Masing-masing tahap tersebut memiliki tugas dan tujuan spesifiknya sendiri untuk mencapai kesuksesan dalam penghasilan permainan [12].

Adapun ilustrasi tahapan metode *GDLC* yang digunakan dapat dilihat pada Gambar 1 sebagai berikut:



Gambar 1. GDLC

A. Initialization

Tahap pertama yaitu tahap inialisasi, penulis melakukan identifikasi kebutuhan dan pembuatan konsep game.

B. Pre-Production

Dilanjutkan dengan preproduction yang berfokus pada desain aset, mekanisme gameplay, dan perencanaan teknis.

C. Production

Tahap *production* merupakan inti dari proses pengembangan, di mana seluruh aset dan fitur *game* mulai diimplementasikan secara nyata.

D. Alpha Testing

Game selanjutnya memasuki fase *alpha testing* untuk memastikan tidak ada *bug* dan seluruh fitur berjalan sesuai rencana.

E. Beta Testing and Validation

Tahapan terakhir dilakukan untuk menguji dan memvalidasi aplikasi secara menyeluruh, guna memastikan aplikasi telah sesuai dengan kebutuhan pengguna dan tujuan pembelajaran.

HASIL DAN PEMBAHASAN

Hasil penelitian menunjukkan bahwa pembuatan prototipe *game platformer 2d "Kebo Blayon"* telah berhasil dikembangkan menggunakan metode *Game*

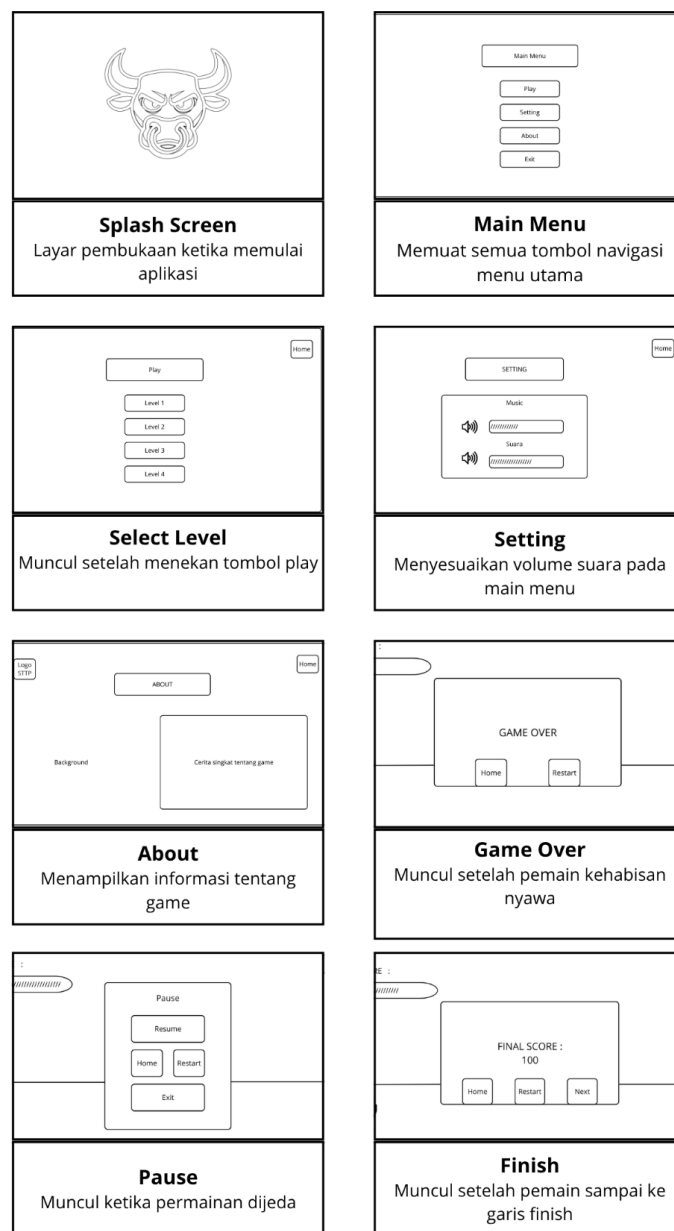
Development Life Cycle (GDLC). Setiap tahapan dalam metode *GDLC* diimplementasikan secara sistematis untuk menghasilkan aplikasi yang sesuai dengan kebutuhan.

A. Inisialisasi / Initialization

Tahap inisialisasi dalam pengembangan prototipe *game platformer 2d "Kebo Blayon"* diawali dengan penentuan rancangan desain yang akan digunakan, pemilihan jenis *game*, dan *platform* yang akan digunakan.

B. Pre-Production

Pre-Production merupakan tahap awal dalam merancang tampilan dan elemen visual aplikasi, mencakup desain UI seperti karakter, lingkungan, dan logo. Perancangan ini diwujudkan melalui storyboard berikut:



Gambar 2. Storyboard

a. *Splash Screen*

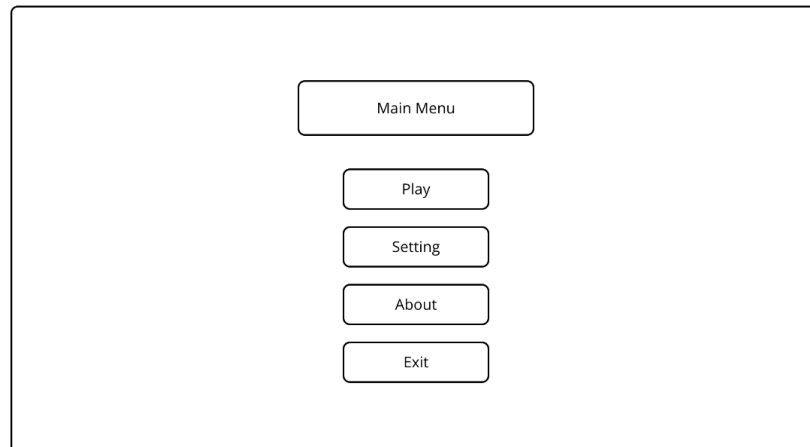
Splash screen adalah tampilan pembuka yang muncul saat pertama kali aplikasi *game* dijalankan. Pada *game* “*Kebo Blayon*”, *splash screen* menampilkan logo *game* secara visual di tengah layar dengan latar belakang sederhana.



Gambar 3. Rancangan *Splash Screen*

b. *Menu Utama*

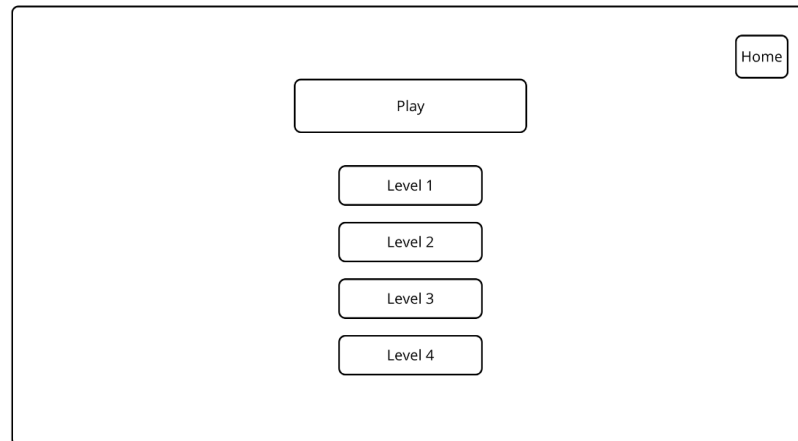
Desain *main menu* ini terdapat tulisan *main menu* di bagian tengah atas layar dan ada empat tombol dibawahnya, yaitu tombol *play*, *setting*, *about*, dan *exit*.



Gambar 4. Rancangan *Main Menu*

c. *Play*

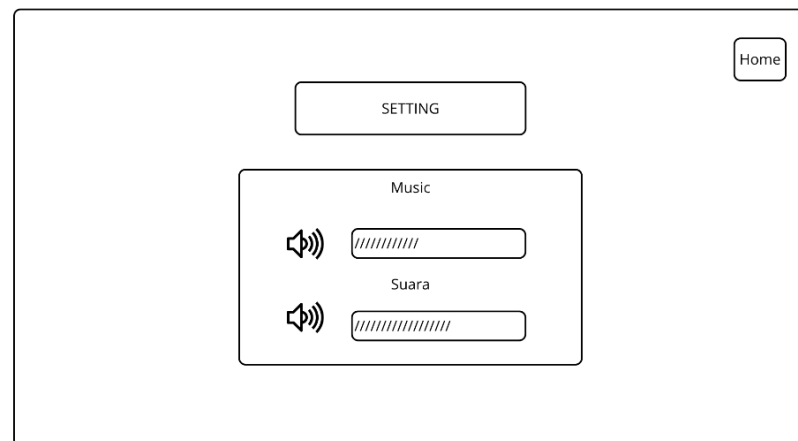
Menampilkan pilihan level yang dapat dimainkan



Gambar 5. Rancangan *Play*

d. Setting

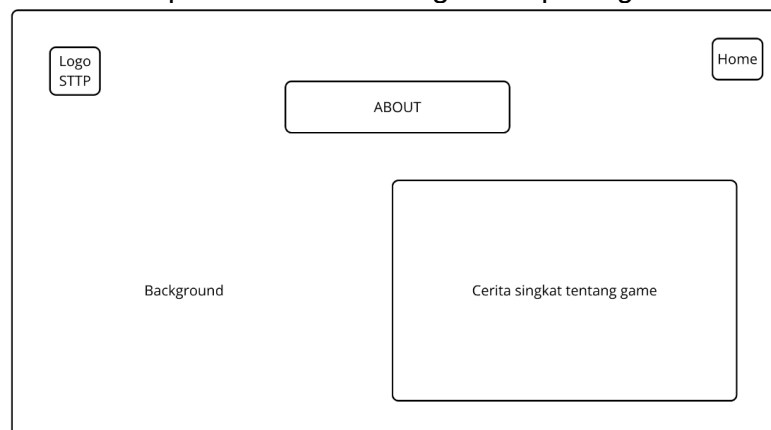
Halaman *setting* digunakan untuk mengatur volume musik latar belakang pada halaman *main* menu



Gambar 6. Rancangan *Setting*

e. About

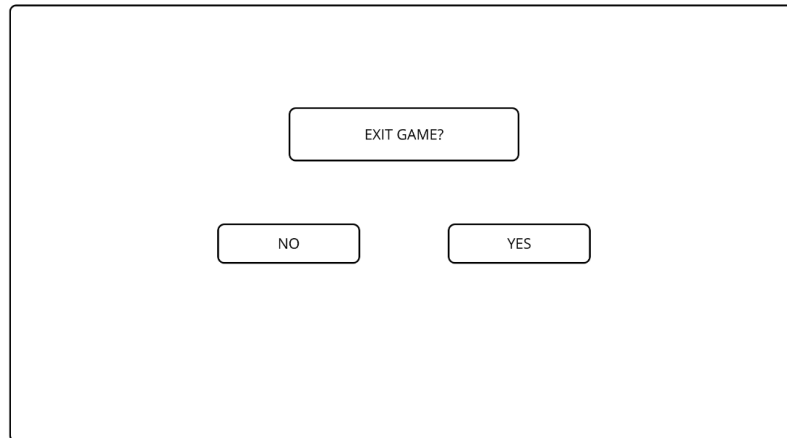
Halaman *about* menampilkan latar belakang cerita pada *game*.



Gambar 7. Rancangan *About*

f. Exit

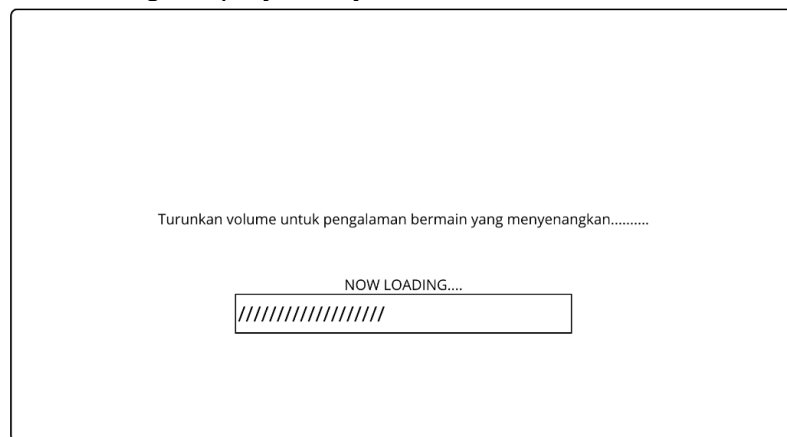
Halaman konfirmasi untuk keluar dari *game*.



Gambar 8. Rancangan *Exit*

g. Loading

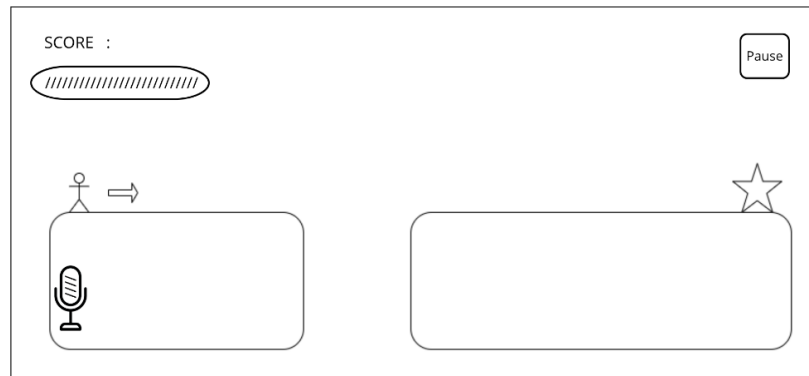
Tampilan *loading* digunakan sebagai penghubung antar *scene* atau saat transisi dari *select level* ke *gameplay* lainnya.



Gambar 9. Rancangan Kuis

h. Level

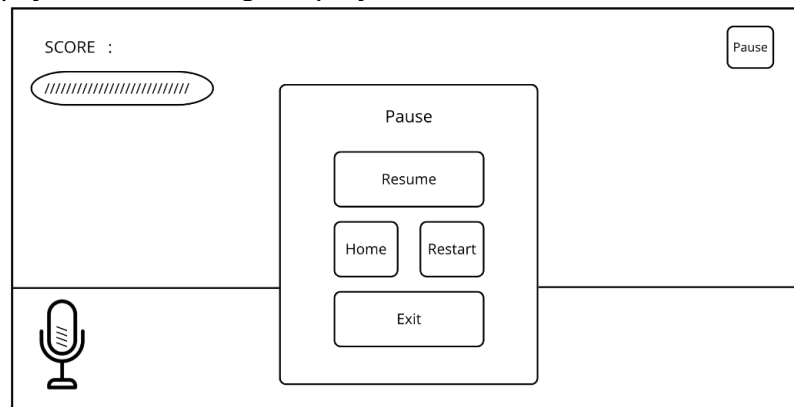
Desain *level* digunakan untuk membuat rancangan *level* yang mencakup *environment*, *ground*, *obstacle*, dan *enemy*. Setiap *level* memiliki tingkat kesulitannya masing-masing.



Gambar 10. Desain *Level*

i. Pause

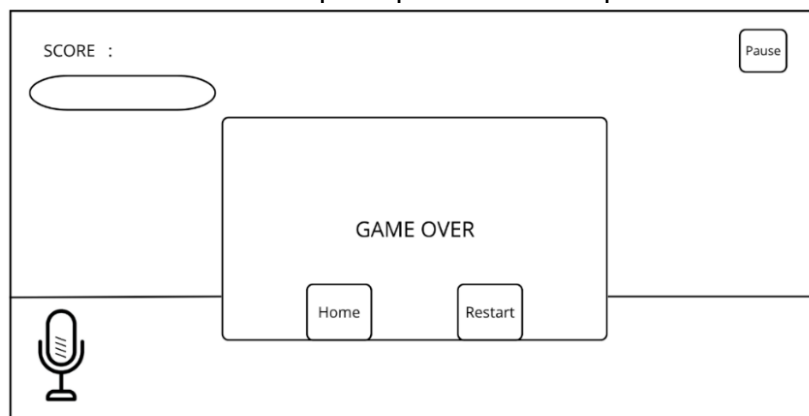
Halaman atau panel *pause* akan muncul ketika pemain menekan tombol *pause* di bagian pojok kanan atas *gameplay*.



Gambar 11. Rancangan *Pause*

j. Game Over

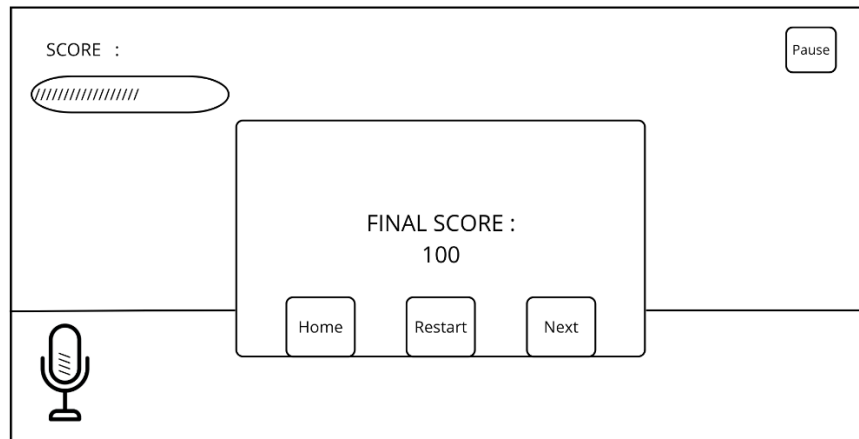
Game Over memberikan tanda kepada pemain bahwa pemain telah kalah.



Gambar 12. Rancangan *Game Over*

k. Finish

Finish muncul ketika pemain dapat melewati seluruh rintangan pada *level* dengan *bar* nyawa yang tersisa.



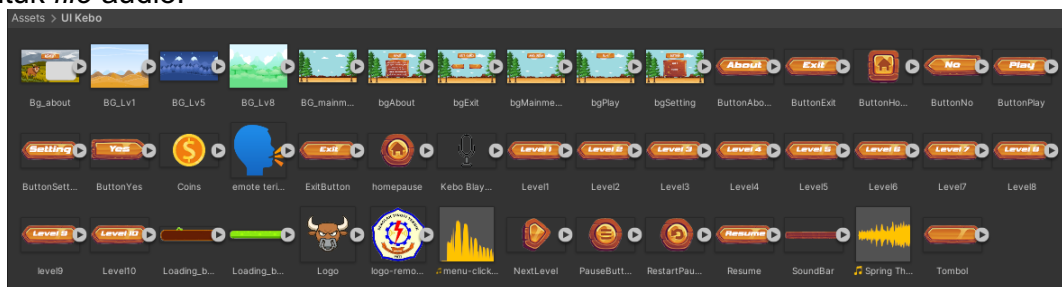
Gambar 13. Rancangan *Finish*

C. Production

Tahap produksi ini merupakan bagian terpenting dalam pembuatan *game Kebo Blayon*, dimana seluruh rancangan yang telah disiapkan pada tahap sebelumnya mulai diimplementasikan ke dalam bentuk nyata melalui *Unity*.

1) Import Assets

Assets yang telah dirancang dan diimplementasikan sebelumnya, selanjutnya akan diimpor ke dalam *Unity* dalam bentuk atau format *.png* untuk gambar dan *.mp3* untuk *file* audio.



Gambar 14. Import Assets

2) Implementasi Tampilan UI

Tampilan UI yang telah dirancang dimasukkan ke dalam *Unity* sebagai antarmuka pengguna game. Implementasi ini mencakup:

1. **Splash Screen:** Menampilkan logo *game* secara visual di tengah layar dengan latar belakang sederhana.



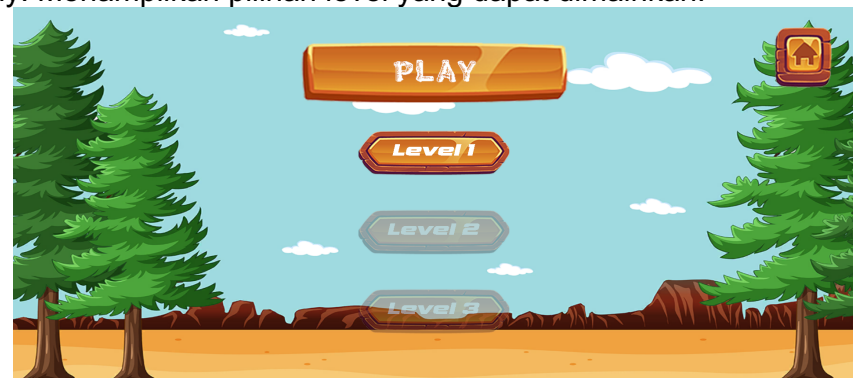
Gambar 15. Splash Screen

2. Menu Utama: Memuat tombol navigasi ke fitur-fitur utama, yaitu Play, Setting, About, dan Exit.



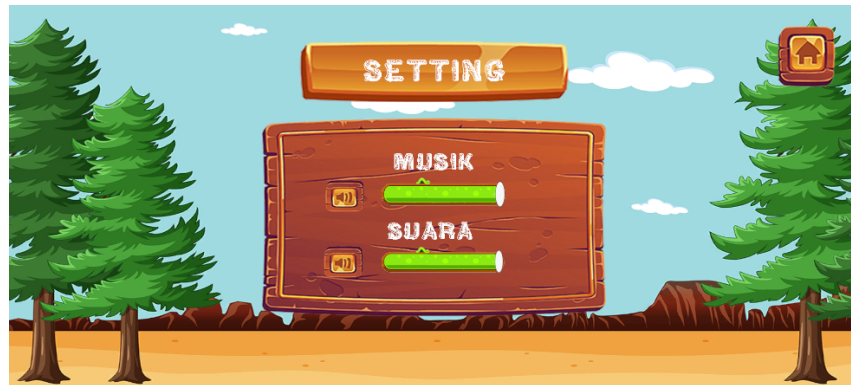
Gambar 16. Menu Utama

3. Halaman Play: Menampilkan pilihan level yang dapat dimainkan.



Gambar 17. Halaman *Play*

4. Halaman Games: Halaman untuk mengatur volume musik di latar belakang main menu.



Gambar 18. Halaman Setting

5. Halaman About: Menampilkan latar belakang cerita dalam game.



Gambar 3. Halaman About

6. Halaman Exit: Menampilkan konfirmasi untuk keluar dari game.



Gambar 20. Halaman Exit

7. Loading: halaman loading muncul ketika berpindah scene dari main menu ke gameplay.

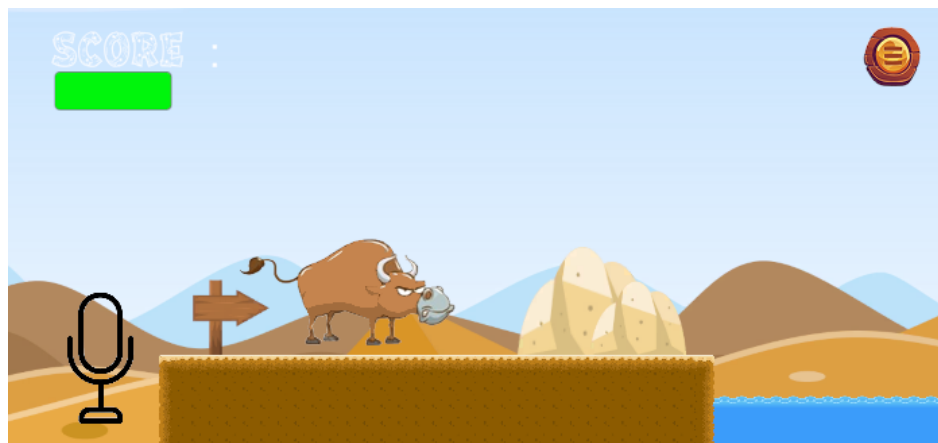


Gambar 21. Halaman Loading

3) Implementasi Gameplay

Gameplay merupakan inti dari game yang dikembangkan. Semua tindakan atau perilaku karakter dan NPC dikendalikan oleh logika masing-masing sesuai script yang terpasang.

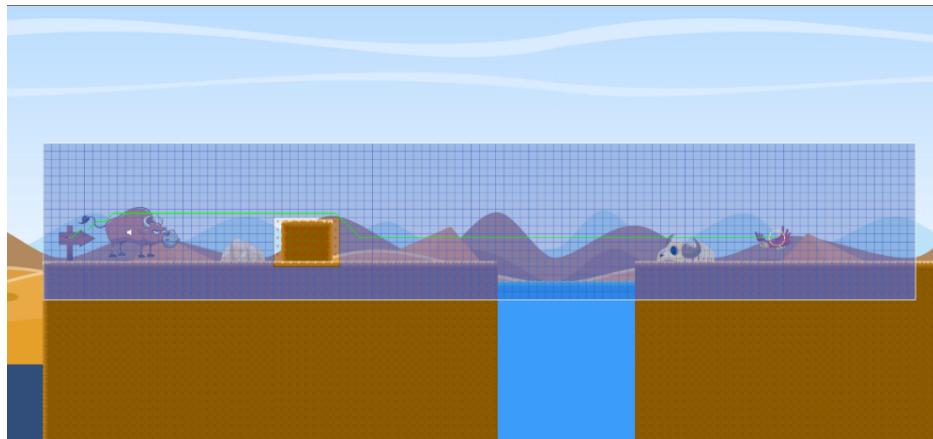
1. UI Gameplay: UI pada saat gameplay menampilkan intensitas suara dari mikrofon, skor, dan nyawa pemain.



Gambar 22. UI Gameplay

2. Implementasi Algoritma A*

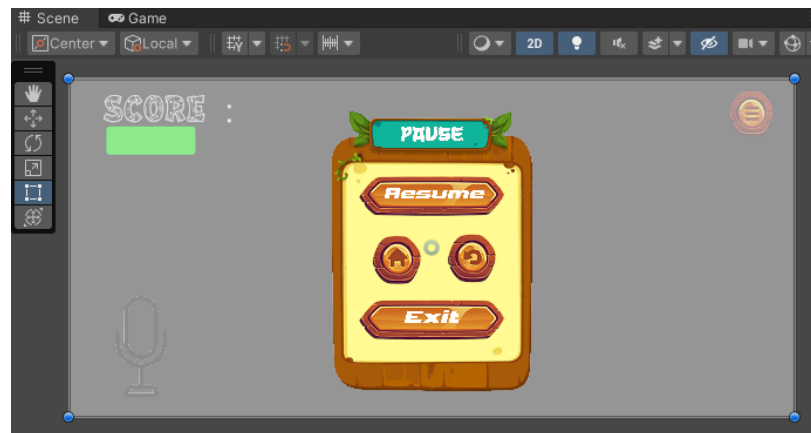
Karakter *enemy* atau burung ini dilengkapi sudah dilengkapi dengan animasi *walk* agar terlihat natural seperti burung sedang terbang. Musuh dalam *game* dikendalikan menggunakan algoritma A* yang memungkinkan mereka bergerak ke arah karakter utama secara otomatis. Musuh memiliki *script AIPath* dan *AIDestinationSetter* yang bekerja dengan komponen *Seeker* dan *GridGraph* dari *package A* Pathfinding Project*.



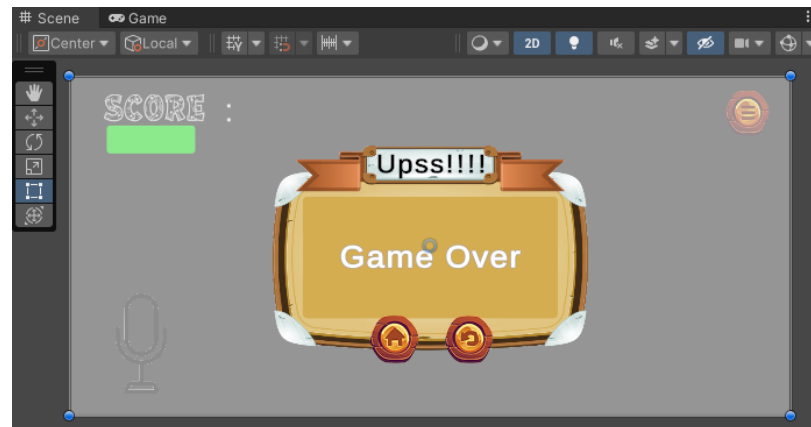
Gambar 23. Algoritma A*

3. Implementasi pause menu, game over, dan finish

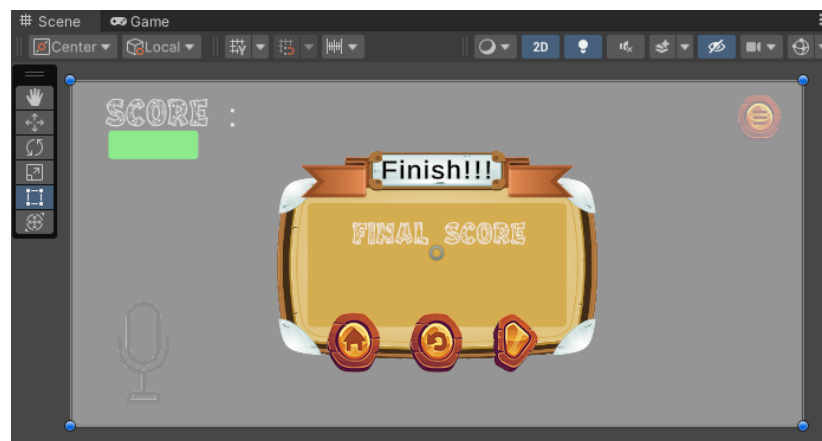
Pause menu memungkinkan pemain menghentikan sementara permainan dengan menekan tombol *pause*. Sementara itu, panel *Game Over* muncul secara otomatis ketika nyawa pemain habis akibat terkena musuh, *obstacle*, atau jatuh ke sungai. Panel ini memberikan informasi bahwa permainan telah berakhir dan menyediakan tombol untuk mengulang *level* atau kembali ke menu. Pada saat pemain berhasil menyelesaikan *level*, ditampilkan panel *Final Score* yang berisi jumlah skor dari koin yang telah dikumpulkan, serta pilihan untuk lanjut ke *level* berikutnya.



Gambar 24. Panel *Pause*



Gambar 25. Panel *Game Over*



Gambar 26. Panel *Finalscore*

D. Validation & Testing / Validasi & Pengujian

a. Alpha Testing

Alpha testing merupakan tahap awal dalam proses pengujian perangkat lunak atau *game* yang dilakukan oleh tim pengembang sebelum dirilis ke publik [13]. Pengujian ini bertujuan untuk memastikan bahwa seluruh fitur dalam *game* berfungsi sesuai dengan yang diharapkan. *Alpha testing* dalam Penelitian dan pengembangan *game Kebo Blayon* ini dilakukan menggunakan metode *blackbox testing*, yaitu metode pengujian yang berfokus pada fungsi dari sistem tanpa melihat struktur kode program [14]. Penguji hanya akan mengevaluasi apakah *output* yang dihasilkan sudah sesuai dengan *input* yang diberikan, tanpa mengetahui bagaimana proses logika di dalamnya. Metode ini sangat sesuai untuk *game* karena dapat menguji fungsionalitas tombol, kontrol suara, interaksi pemain, alur *game*, dan navigasi *level*. *Blackbox testing* dapat mengetahui apakah setiap fitur dalam *game* seperti navigasi tombol, kontrol suara, algoritma *A**, *checkpoint*, dan sistem skor berjalan dengan benar. Pengujian dilakukan dengan membuat skenario uji dan mencatat hasil dari setiap percobaan untuk dianalisis apakah fitur tersebut berhasil dijalankan dengan benar atau masih mengandung *bug*.

Tabel 1. *Blackbox test* halaman *main* menu

No	Fitur yang diuji	Langkah uji	Input	Output diharapkan	Status (Pass/Fail)
1	Tombol <i>Play</i>	Klik tombol <i>play</i> di <i>Main</i> menu	klik	Muncul halaman <i>select Level</i>	<i>Pass</i>
2	Tombol <i>Setting</i>	Klik tombol <i>setting</i> di <i>main</i> menu	klik	Muncul halaman <i>setting</i>	<i>Pass</i>
3	Tombol <i>about</i>	Klik tombol <i>about</i> di <i>main</i> menu	klik	Muncul halaman <i>about</i>	<i>Pass</i>
4	Tombol <i>exit</i>	Klik tombol <i>exit</i> di <i>main</i> menu	klik	Muncul halaman konfirmasi <i>exit</i>	<i>Pass</i>
5	Tombol <i>level</i>	Klik tmbol <i>level</i> pada halaman <i>select level</i>	klik	Muncul halaman <i>loading</i> sebelum masuk ke <i>gameplay</i>	<i>Pass</i>
6	Tombol <i>Home</i>	Klik tombol <i>home</i> pada halaman <i>select level</i>	klik	Kembali ke <i>main</i> menu	<i>Pass</i>
7	Tombol <i>Home</i>	Klik tombol <i>home</i> pada halaman <i>Setting</i>	klik	Kembali ke <i>main</i> menu	<i>Pass</i>
8	Tombol <i>Home</i>	Klik tombol <i>home</i> pada halaman <i>About</i>	klik	Kembali ke <i>main</i> menu	<i>Pass</i>
9	<i>Bar</i> Volume	Gulir <i>bar</i> volume pada halaman <i>setting</i>	Gulir	Volume suara menyesuaikan <i>bar</i>	<i>Pass</i>
10	Tombol <i>No</i>	Klik tombol <i>No</i> pada halaman konfirmasi <i>exit</i>	klik	Kembali ke <i>main</i> menu	<i>Pass</i>
11	Tombol <i>Yes</i>	Klik tombol <i>yes</i> pada halaman konfirmasi <i>exit</i>	klik	Keluar dari <i>game</i>	<i>Pass</i>

Tabel 2 *Blackbox test* *gameplay*

No	Fitur yang diuji	Langkah uji	Input	Output diharapkan	Status (Pass/Fail)
1	Kontrol suara jalan	Bicara dengan volume $0 > X < 70$ dB	Suara pelan	Karakter bergerak ke kanan	<i>Pass</i>
2	Kontrol suara lompat	Bicara dengan volume ≥ 70 dB	Suara keras	Karakter melompat	<i>Pass</i>
3	Koin	Sentuh objek koin	Karakter menyentuh koin	Koin hilang, skor bertambah	<i>Pass</i>
4	<i>Obstacle</i>	Menabrak <i>obstacle</i>	Karakter menyentuh kaktus	Nyawa berkurang dan karakter berkedip merah	<i>Pass</i>
5	<i>Fall Detector</i>	Jatuh ke sungai atau jurang	Karakter jatuh	Nyawa berkurang, pemain <i>respawn</i>	<i>Pass</i>
6	<i>Checkpoint</i>	Melewati titik <i>checkpoint</i>	Karakter menyentuh	Posisi karakter disimpan	<i>Pass</i>

Jurnal EDU ELEKTROMATIKA (JEE)

ISSN: 2747-0784 (p); xxxxxx (e)

Vol 6, No. 2, Desember 2025

No	Fitur yang diuji	Langkah uji	Input	Output diharapkan	Status (Pass/Fail)
7	<i>Respawn</i>	Jatuh ke <i>fall detector</i> setelah menyentuh <i>checkpoint</i>	<i>checkpoint</i> Jatuh setelah <i>checkpoint</i>	Karakter kembali ke <i>checkpoint</i>	<i>Pass</i>
8	<i>Finish</i>	Mencapai titik akhir <i>level</i>	Karakter menyentuh rumput	Panel skor akhir muncul, waktu permainan berhenti	<i>Pass</i>
9	<i>Pause Game</i>	Klik tombol <i>pause</i>	klik	Permainan terhenti, menu <i>pause</i> muncul	<i>Pass</i>
10	<i>Resume Game</i>	Klik tombol <i>resume</i> di menu <i>pause</i>	klik	Permainan berlanjut dari posisi sebelumnya	<i>Pass</i>
11	<i>Restart Level</i>	Klik tombol <i>restart</i> pada panel <i>pause</i> , <i>game over</i> , dan <i>finish</i>	klik	<i>Level</i> dimulai ulang dari awal	<i>Pass</i>
12	Kembali ke <i>Main Menu</i>	Klik tombol <i>home</i> dari panel <i>pause</i> , <i>game over</i> , dan <i>finish</i>	Klik	Navigasi kembali ke halaman <i>main menu</i>	<i>Pass</i>
13	<i>Next Level</i>	Klik tombol <i>next level</i> dari panel skor akhir	Klik	Pindah ke <i>level</i> berikutnya	<i>Pass</i>
14	Sistem Skor	Ambil lebih dari satu koin	Ambil 2 koin	Skor bertambah sesuai jumlah koin	<i>Pass</i>
15	Sistem Nyawa	Menabrak <i>obstacle</i> sebanyak 3 kali	Sentuh 3 <i>obstacle</i>	Nyawa habis, panel <i>game over</i> muncul	<i>Pass</i>
16	Algoritma A*	Biarkan musuh berjalan mendekati target	Musuh aktif	Musuh bergerak otomatis ke arah target menggunakan jalur optimal	<i>Pass</i>
17	Musuh	Karakter menabrak musuh	Sentuh musuh	Nyawa karakter berkurang, musuh hancur	<i>Pass</i>
18	<i>UI Volume Mic Slider</i>	Karakter berjalan dan bergerak	Bersuara pelan dan keras	Volume <i>mic</i> menyesuaikan <i>input</i> suara dari mikrofon	<i>Pass</i>
19	<i>Level</i> Terkunci	Buka <i>game</i> pertama kali dan coba klik <i>level 2</i> ke atas	klik	<i>Level</i> terkunci tidak bisa dipilih	<i>Pass</i>
20	<i>Level</i> Terbuka	Selesaikan <i>level 1</i> , lalu buka halaman <i>select</i>	klik	<i>Level 2</i> terbuka setelah <i>level 1</i> selesai	<i>Pass</i>

No	Fitur yang diuji	Langkah uji <i>level</i>	Input	Output diharapkan	Status (Pass/Fail)
----	------------------	-----------------------------	-------	-------------------	--------------------

b. Beta Testing

Beta testing adalah salah satu metode pengujian yang dilakukan untuk mengesahkan atau memvalidasi suatu perangkat lunak, dalam metode ini biasa pengujian yang dilakukan yaitu dengan cara menyebarkan perangkat lunak kepada sekelompok orang yang mengerti dan paham akan media pembelajaran digital, setelah itu sekelompok orang tersebut akan diberikan sebuah kuesioner yang berisi pernyataan untuk diisi [15]. *Beta testing* pada penelitian ini ditujukan kepada dua kategori yaitu ahli IT dan mahasiswa atau masyarakat umum, dengan tujuan memvalidasi desain rancangan dan juga *game* yang sudah dikembangkan.

Validasi desain *game* "*Kebo Blayon*" dilakukan untuk memastikan bahwa rancangan *game*, baik dari segi *user interface*, mekanisme permainan, maupun alur cerita, telah sesuai dengan tujuan pengembangan dan mudah dipahami oleh pengguna. Proses validasi oleh ahli dilakukan dengan mengisi kuesioner yang memuat 25 pernyataan terkait rancangan tampilan visual, tata letak, komposisi warna, serta kenyamanan desain antarmuka *game*. Berdasarkan hasil pengisian oleh validator ahli dari bidang teknologi informasi, diperoleh skor sebesar 94,40%, yang termasuk dalam kategori "**Sangat Layak**". Uji juga dilakukan kepada 50 responden yang merupakan mahasiswa aktif Sekolah Tinggi Teknik Pati dan masyarakat umum untuk mengetahui persepsi pengguna terhadap desain *game*. Hasil rekapitulasi menunjukkan bahwa validasi desain oleh responden umum mencapai 98,32%, yang juga termasuk dalam kategori "**Sangat Layak**". Nilai tersebut mencerminkan bahwa desain *game* sangat diterima oleh pengguna dari kalangan umum dan memberikan kesan visual yang menyenangkan serta mudah dipahami.

Setelah melalui proses validasi desain, dilakukan validasi terhadap *prototype game* untuk mengetahui sejauh mana sistem bekerja secara fungsional, termasuk kendali suara dan navigasi algoritma A^* pada *NPC*. Validasi *prototype* juga melibatkan dua kelompok penilai, yaitu validator ahli dan responden umum. Proses validasi oleh ahli dilakukan dengan mengisi kuesioner yang memuat 25 pernyataan terkait aspek teknis dan fungsionalitas dalam *game* yang telah dibuat. Berdasarkan hasil pengisian oleh validator ahli dari bidang teknologi informasi, diperoleh skor sebesar 96,8%, yang termasuk dalam kategori "**Sangat Layak**". Uji kelayakan juga dilakukan kepada 50 responden yang merupakan mahasiswa aktif Sekolah Tinggi Teknik Pati dan masyarakat umum untuk mengetahui persepsi pengguna terhadap *prototype game*. Hasil rekapitulasi menunjukkan bahwa validasi desain oleh responden umum mencapai 98,56%, yang juga termasuk dalam kategori "**Sangat Layak**". Nilai tersebut mencerminkan bahwa desain *game* sangat diterima oleh pengguna dari kalangan umum dan memberikan kesan visual yang menyenangkan serta mudah dipahami.

Secara keseluruhan, berdasarkan hasil validasi dari semua pihak, baik ahli, maupun responden, dapat disimpulkan bahwa *game Kebo Blayon* ini berada dalam kategori "**Sangat Layak**" untuk dilanjutkan ke tahap pengembangan lebih lanjut.

KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, dapat disimpulkan bahwa *game Kebo Blayon* berhasil dirancang dan dikembangkan dengan mengimplementasikan kontrol suara sebagai media interaksi utama serta algoritma A^* untuk navigasi karakter musuh (*NPC*). *Game* ini dibangun menggunakan *platform Unity* dan ditujukan untuk perangkat *Android*. Hasil pengujian menunjukkan bahwa kontrol suara mampu membedakan antara suara pelan untuk berjalan dan suara keras untuk melompat, serta algoritma A^* dapat mengarahkan *NPC* mengikuti jalur secara optimal menuju karakter pemain. Validasi desain dan *prototype* yang dilakukan oleh ahli dan responden umum menunjukkan bahwa *game* ini dinilai sangat layak dengan persentase kelayakan masing-masing mencapai lebih dari 90%. Dengan demikian, implementasi teknologi suara dan algoritma pencarian jalur ini terbukti efektif dalam meningkatkan pengalaman bermain dalam *game platformer 2D*.

DAFTAR PUSTAKA

- [1] R. N. Tasya Chayrina^{1*}, Awalia Rahmatia², Adek Kurniawan³, Nabila Aprilioni⁴, Vidya Nofriza⁵, Chelsy Putri⁶, "PENGUNAAN VIRTUAL REALITY PADA GAME CREED: RISE TO GLORY-CHAMPIONS EDITION DI GAME BLINK PADANG Tasya," *J. Komun.*, vol. 2, no. 1, pp. 639-652 e-ISSN:, 2024.
- [2] M. I. Tohari, J. Jamaaluddin, I. Sulistiyowati, T. Elektro, and U. M. Sidoarjo, "Sistem Pengenalan Suara Sebagai Pengendali Peralatan Audio Berbasis Arduino Uno," *Forum Pendidik. Tinggi Tek. Elektro Indones. Reg. VII*, pp. 86–90, 2021.
- [3] N. Aditiya and Diana Laily Fithri, "Analisis Game 2D 'Monkey Banana Survival' Serta Pengujian Menggunakan Metode Blackbox," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 11, no. 2, pp. 190–196, 2024, doi: 10.30656/prosisko.v11i2.8507.
- [4] Muliadi, J. Pragantha, and D. Andana Harris, "Pembuatan Game Platformer 'Beyond' Menggunakan Unity Dengan Xbox 360 Controller," *J. Ilmu Komput. dan Sist. Inf.*, vol. 4, no. 2, pp. 348–356, 2016.
- [5] B. Robyananta Saputra and R. Ciptaningrum, "Perancangan Game Kasual Pongo Tap Pixel Art 2D," *Syntax Idea*, vol. 5, no. 10, pp. 1670–1678, 2023, doi: 10.46799/syntax-idea.v5i10.2644.
- [6] Aryanti, I. Mekongga, and H. Ramadhan, "Implementasi Sensor Suara Sebagai Pengendali Gerakan Robot Penari Humanoid dengan ATMEGA 8535," *J. Penelit. Ilmu dan Teknol. Komput.*, vol. 8, no. 1, pp. 1–7, 2016.
- [7] D. Sugiharto, D. Astuti, M. Mujirudin, and ..., "Perangkat Saklar Aktivasi Melalui Pengenalan Suara Manusia," *J. Teknol. ...*, vol. 3, no. 1, 2021.
- [8] R. Rizky, "Pencarian Jalur Terdekat dengan Metode A^* (Star) Studi Kasus Serang Labuan Provinsi Banten | Prosiding Seminar Nasional Rekayasa Teknologi Informasi | SNARTISI," *Semin. Nas. Rekayasa Teknol. Inf.*, vol. 1, no. November, pp. 93–98, 2018.
- [9] M. Tryastie, L. Nelda, R. Priskila, and V. H. Pranatawijaya, "Implementasi Algoritma A^* (Star) untuk Menentukan Rute Jarak Terpendek," vol. 2, pp. 449–457, 2024.
- [10] M. Roihan, "Penerapan Algoritma A^* Untuk Pencarian Rute Terpendek Dalam Navigasi GPS," 2023.
- [11] A. A. Saputra, F. N. Putra, and R. D. R. Yusron, "Pembuatan Game Edukasi

- Pengenalan Kebudayaan Indonesia Menggunakan Metode Game Development Life Cycle (GDLC) Berbasis Android,” *J. Autom. Comput. Inf. Syst.*, vol. 2, no. 1, pp. 66–73, 2022.
- [12] D. M. Luay, Asriyanik, and W. Apriandari, “Penggunaan Metode Gdlc (Game Development Life Cycle) Untuk Mengenal Bendera Dunia,” *INFOTECH J.*, vol. 10, no. 1, pp. 40–48, 2024, doi: 10.31949/infotech.v10i1.8374.
- [13] J. Enstein, V. R. Bulu, and R. L. Nahak, “Pengembangan Media Pembelajaran Game Edukasi Bilangan Pangkat dan Akar menggunakan Genially,” *J. Jendela Pendidik.*, vol. 2, no. 01, pp. 101–109, 2022, doi: 10.57008/jjp.v2i01.150.
- [14] D. MARDIATI and Y. SAPUTRA, “Implementasi Sistem Informasi Manajemen Klinik Menggunakan Metode Black Box Testing,” *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, 2025, doi: 10.23960/jitet.v13i1.6015.
- [15] E. S. Susanto *et al.*, “Pengembangan Aplikasi Smart-Book Sebagai Media Pembelajaran,” *J. Mnemon.*, vol. 5, no. 1, pp. 64–71, 2022.