Penerapan Algoritma Finite State Machine Pada Game Roguelike Moonbase Warden

1st Hafidz Ammar
S1 Teknik Informatika
Universitas Telkom Purwokerto
Purwokerto, Indonesia
hafidzammar@student.telkomuniversity.ac.id

2nd Muhamad Azrino Gustalika
S1 Teknik Informatika
Universitas Telkom Purwokerto
Purwokerto, Indonesia
azrino@telkomuniversity.ac.id

Abstrak **— *Banyak game roguelike yang menerapkan perilaku Non-Playable Character (NPC) yang sederhana dan statis sering kali tidak cukup untuk menciptakan tantangan yang optimal, sehingga membuat pemain merasa bosan. Hasil kuesioner Game Experience Questionnaire (GEQ) dengan 13 responden terhadap game roguelike serupa menunjukkan nilai rata-rata pernyataan "Aku tidak merasa bosan saat bermain" sebesar 2,54 dari skala 4, yang mengindikasikan tingginya tingkat kebosanan pemain. Penelitian ini bertujuan untuk menurunkan tingkat kebosanan bermain game roguelike melalui game "Moonbase Warden" dengan penerapan algoritma Finite State Machine (FSM) pada perilaku karakter. Hasil Kuesioner GEQ menunjukkan peningkatan nilai pada pernyataan "Aku tidak merasa bosan saat bermain" dari 2,54 terhadap game pembanding menjadi 3,31 terhadap "Moonbase Warden", yang menunjukkan penurunan tingkat kebosanan sebesar 52,73%. Penurunan tersebut terindikasi pada nilai pernyataan "Aku merasa ditantang oleh game ini" sebesar 3,77. Seluruh transisi state FSM berfungsi dengan baik berdasarkan Alpha Testing menggunakan metode Blackbox Testing, serta tingkat usability game mencapai 84,83% berdasarkan Beta Testing dengan metode Usability Testing.***

Kata kunci**— game, roguelike, perilaku, FSM, GEQ**

I. PENDAHULUAN

Video game adalah sarana hiburan yang dapat menjadi hobi atau profesi, dengan perkembangan teknologi dan inovasi yang menjadikannya salah satu industri hiburan paling menarik di dunia [1]. Game roguelike yang dikenal dengan elemen eksplorasi, permadeath, dan lingkungan yang dihasilkan secara prosedural menawarkan pengalaman yang menantang bagi pemain [2].

Banyak game menerapkan perilaku Non-Playable Character (NPC) yang sederhana. NPC tradisional yang menampilkan perilaku statis sering kali tidak cukup untuk menciptakan tantangan yang optimal bagi pemain [3], yang membuat pemain berpikir bahwa game tersebut terasa membosankan [4] dan tidak memainkan gamenya lagi [5]. Berdasarkan hasil wawancara kuesioner Game Experience Questionnaire (GEQ) dengan 13 orang responden dalam rentang usia 18 sampai 30 tahun terhadap pengalaman bermain sebuah game roguelike yang serupa, teridentifikasi bahwa pemain masih merasa bosan saat memainkan game tersebut, dengan nilai rata-rata pernyataan "Aku tidak merasa bosan saat bermain" sebesar 2.54 dari nilai maksimal 4.00.

Studi literatur menunjukkan bahwa integrasi AI pada perilaku NPC dapat memperkaya interaksi antara NPC dan pemain serta menciptakan pengalaman bermain yang berubah-ubah dan menantang [6]. Untuk mengatasi keterbatasan perilaku primitif tersebut pada game yang dikembangkan, penelitian ini menggunakan algoritma Finite State Machine (FSM) yang mampu menghasilkan respons karakter yang lebih variatif dalam berbagai studi game sebelumnya [7][8].

II. KAJIAN TEORI

Berikut adalah beberapa teori yang digunakan dalam penelitian mengenai penerapan algoritma *Finite State Machine* pada game roguelike *Moonbase Warden*.

A. *Game*

Game berasal dari kata dalam bahasa Inggris yang berarti permainan [1]. Secara umum, game dapat dipahami sebagai media hiburan yang dapat mengasah kemampuan, memberikan pengalaman interaktif, atau berfungsi sebagai sarana pendidikan [9]. Permainan video dapat dibagi menjadi beberapa genre, seperti game strategi, roleplaying, aksi, simulasi kendaraan olahraga, hingga edukasi. Setiap genre memiliki karakteristik yang berfokus pada interaksi pemain dengan sistem permainan [1].

B. *Unity*

Saat ini Unity dikenal sebagai salah satu platform pengembangan game terbaik untuk menciptakan game 2D, 3D, simulasi, dan pengalaman virtual di berbagai perangkat, mulai dari komputer hingga konsol dan ponsel [10]. Karena ketersediaan fungsi-fungsi pokok dalam pengembangan game dan dukungan multi-platform, *Game engine* Unity menjadi standar dalam industri pengembangan game [11]. FSM Karakter diimplementasikan menggunakan *Animator Controller* yang disediakan oleh Unity yang dibuat berdasarkan desain awal penerapan FSM karakter. *Animator Controller* digunakan karena kemudahan pengelolaan transisi animasi karakter yang berhubungan dengan *state* karakter [12].

C. Finite State Machine

Finite State Machine (FSM) merupakan model perilaku yang digunakan untuk mengatur bagaimana suatu objek bereaksi terhadap kondisi dan kejadian tertentu secara

bertahap. FSM menyimpan kondisi aktif permainan dalam bentuk *state* (keadaan), sehingga keputusan yang diambil oleh sistem bergantung pada input saat ini dan keadaan sebelumnya. *State* pada FSM mewakilkan keadaan perilaku objek dalam game, seperti kondisi diam, bergerak, menyerang, atau kalah. Perpindahan antar *state* terjadi ketika suatu kondisi atau keadaan terpenuhi [13].

D. Multimedia Development Life Cycle

Multimedia Development Life Cycle (MDLC) adalah kerangka kerja sistematis untuk mengembangkan aplikasi multimedia. MDLC memiliki 6 tahap, yaitu: *Concept*, *Design*, *Collecting Content Material*, *Assembly*, *Testing*, dan *Distribution* [14].

E. User Acceptance Testing

User Acceptance Testing (UAT) merupakan tahap pengujian perangkat lunak yang dilakukan untuk memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan pengguna akhir. *Alpha testing* merupakan tahap pengujian awal yang dilakukan secara internal oleh pengembang. *Alpha testing* digunakan untuk memastikan bahwa mekanik permainan berjalan dengan baik. Pendekatan *blackbox testing* digunakan sebagai metode pengujian karena pengujian difokuskan pada perilaku sistem berdasarkan input dan output tanpa memperhatikan struktur internal program. *Beta testing* merupakan tahap pengujian yang melibatkan pengguna akhir, sehingga fokus pengujian terletak pada pengalaman penggunaan sistem secara nyata. Oleh karena itu, *usability testing* menjadi pendekatan yang tepat untuk diterapkan pada tahap beta testing [15].

F. Game Experience Questionnaire

Game Experience Questionnaire (GEQ) adalah instrumen pengukuran berbasis kuesioner yang digunakan untuk mengukur pengalaman psikologis pemain baik selama proses bermain maupun setelah permainan selesai. Kuesioner dibuat berdasarkan dimensi dan parameter pengukuran yang dibutuhkan dalam penelitian [16]. Setiap pernyataan dinilai berdasarkan skala *likert* dari nilai 0 (Tidak sama sekali) sampai nilai 4 (Sangat setuju). GEQ terdiri dari beberapa dimensi utama yang merepresentasikan pengalaman bermain game secara komprehensif, yaitu: *Immersion, Tension, Competence, Flow, Positive Affect, Negative Affect,* dan *Challenge* [17].

III. METODE

Penelitian ini menggunakan metode MDLC dalam penerapan FSM dan pengembangan game "*Moonbase Warden*".

A. Concept

Tahap *concept* menentukan tujuan pengembangan game, judul, genre, platform, jangkauan pemain, durasi, jumlah level, dan perangkat pengendalian.

B. Design

Tahap *design* merupakan proses perancangan alur permainan, antarmuka pengguna, serta rancangan FSM untuk setiap karakter. Hasil dari tahap ini berupa flowchart sistem, diagram FSM, dan wireframe antarmuka.

C. Collecting Content Material

Tahap *collecting content material* dilakukan dengan cara mengumpulkan dan membuat aset yang dibutuhkan dalam pengembangan game, seperti *background*, *tileset*, atlas karakter, dan suara karakter.

D. Assembly

Tahap *assembly* adalah proses implementasi seluruh rancangan ke dalam *game engine* Unity, termasuk penerapan algoritma FSM dan integrasi aset.

E. Testing

Tahap *testing* terdiri dari *Alpha Testing* menggunakan metode *Blackbox Testing, Beta Testing* menggunakan metode *Usability Testing* untuk mengevaluasi fungsionalitas dan tingkat kegunaan game, dan GEQ untuk mengukur pengalaman psikologis pemain, khususnya pada parameter kebosanan.

F. Distribution

Tahap *distribution* dilakukan dengan cara menyebarkan game melalui platform distribusi digital setelah dinyatakan layak berdasarkan hasil pengujian.

IV. HASIL DAN PEMBAHASAN

Penerapan algoritma FSM dan pengembangan game "*Moonbase Warden*" berhasil dilakukan menggunakan metode MDLC yang menghasilkan game *roguelike* dengan perilaku karakter yang lebih kompleks dibandingkan dengan game perbandingan.

A. Concept

Pengembangan Moonbase Warden mengambil unsur dari genre *Action*, *Platformer*, 2D, dan *Roguelike*, di mana pemain dapat melakukan aksi dengan melawan musuh, melewati rintangan yang dihadapi, dengan lingkungan yang berubah seiring berjalannya permainan.

Pengembangan game Moonbase Warden bertujuan untuk menerapkan algoritma FSM terhadap karakter dalam permainan agar pengalaman pemain menjadi tidak membosankan

Berikut merupakan konsep dasar dari game Moonbase Warden.

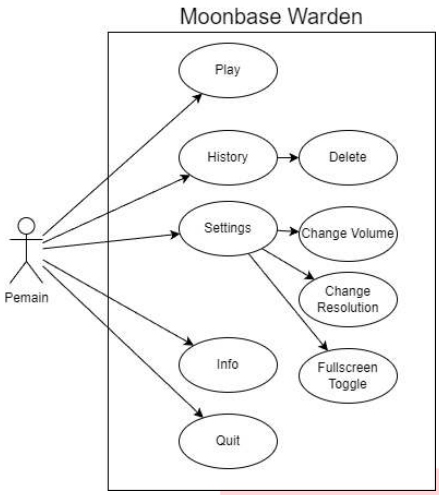
TABEL 1
(Konsep Dasar Game Moonbase Warden)

Judul	Monbase Warden
Genre	Action, Platformer, 2D, Roguelike
Platform	Windows
Jangkauan Pemain	Umum
Durasi	30 Menit, dapat diulang
Jumlah Level	3 Level dengan 2 Phase Boss Fight
Kontrol	Keyboard

B. Design

Tahap ini berfokus pada perencanaan dan perancangan elemen-elemen yang akan digunakan dalam pengembangan game "*Moonbase Warden*".

1. Diagram Use Case

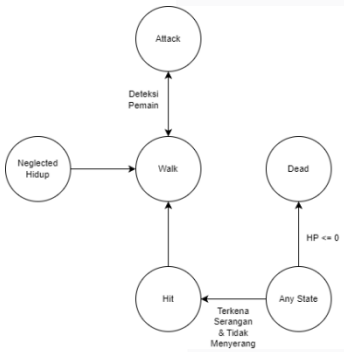


GAMBAR 1
(Use Case Diagram Moonbase Warden)

Gambar di atas menampilkan diagram terkait *Use Case* game "*Moonbase Warden*". Aktor dapat memilih opsi "*Play*", "*History*", "*Settings*", "*Info*", dan "*Quit*". Pada opsi "*History*", pemain dapat melakukan "*Delete*", dan pada opsi "*Settings*", pemain dapat melakukan "*Change Volume*", "*Change Resolution*", dan "*Fullscreen Toggle*".

2. Diagram FSM Karakter

Berikut adalah penerapan FSM pada karakter musuh *Neglected*



GAMBAR 2
(Penerapan FSM pada Musuh Neglected)

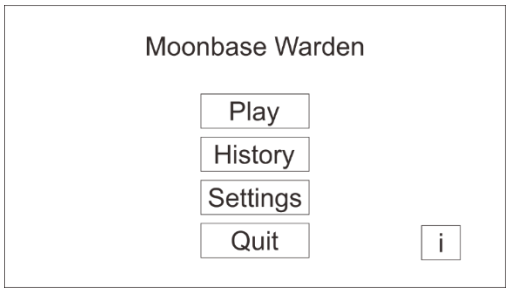
Gambar di atas merupakan diagram penerapan FSM pada musuh *Neglected*. Berikut merupakan tabel transisi *state* yang diterapkan:

TABEL 2
(Tabel Transisi State FSM pada Musuh Neglected)

No	State	Kejadian	State Transisi
1	Neglected Hidup	Kejadian: Spawn	Walk
2	Walk	Kejadian: Deteksi Pemain	Attack
3	Attack	Kejadian: Animasi Attack selesai	Walk
4	Hit	Kejadian: Animasi Hit selesai	Walk
5	Any State (state apapun)	Kejadian: Terkena Serangan Kejadian: Tidak Menyerang	Hit
		Kejadian: HP <= 0	Dead

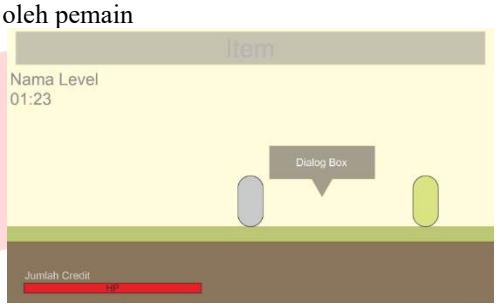
3. Wireframe

Wireframe mencakup desain awal antarmuka menu dan permainan.



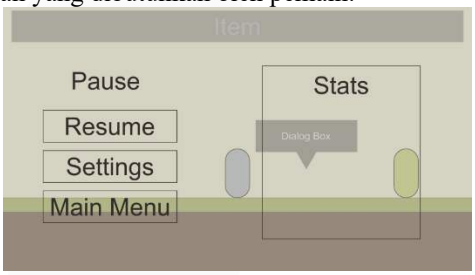
GAMBAR 3
(Wireframe Menu Utama)

Gambar di atas menampilkan desain awal antarmuka menu utama yang menyajikan beberapa use case yang dapat dipilih oleh pemain



GAMBAR 4
(Wireframe Antarmuka Permainan)

Gambar di atas menampilkan desain awal antarmuka permainan yang menampilkan elemen-elemen informasi permainan yang dibutuhkan oleh pemain.



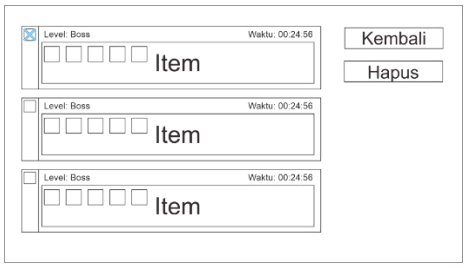
GAMBAR 5
(Wireframe Menu Jeda)

Gambar di atas menampilkan desain awal antarmuka menu Jeda yang diletakkan di atas tampilan antarmuka permainan.



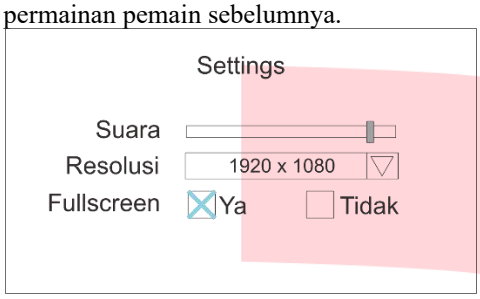
GAMBAR 6
(Wireframe Antarmuka Scoreboard)

Gambar di atas menampilkan desain awal antarmuka *scoreboard* yang menampilkan statistik permainan.



GAMBAR 7
(Wireframe Halaman Riwayat)

Gambar di atas menampilkan desain awal antarmuka halaman riwayat permainan yang menunjukkan beberapa riwayat permainan pemain sebelumnya.



GAMBAR 8
(Wireframe Halaman Pengaturan)

Gambar di atas menampilkan desain awal antarmuka halaman pengaturan yang menyajikan opsi pengaturan bagi pemain untuk diatur.

C. Collecting Content Material

Pengumpulan material terdiri dari elemen-elemen yang dibutuhkan dalam pengembangan game, yaitu *Background*, *Tileset*, dan Karakter.

1. Background

Aset *background* yang dibutuhkan dibuat untuk mendukung efek *paralax*, yaitu teknik visual yang memberikan kesan kedalaman dengan cara menggerakkan lapisan latar belakang di bagian belakang pada kecepatan berbeda dibandingkan dengan latar belakang di bagian depan.

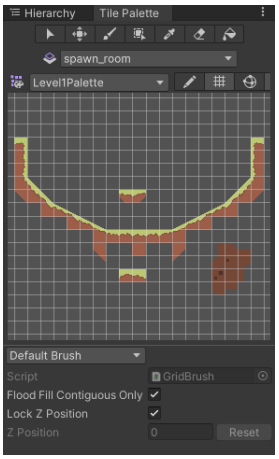


GAMBAR 9
(Aset Background)

Gambar di atas menunjukkan lapisan background yang sudah di-import ke dalam proyek Unity.

2. Tileset

Tileset digunakan dalam pembuatan lingkungan level pada game.

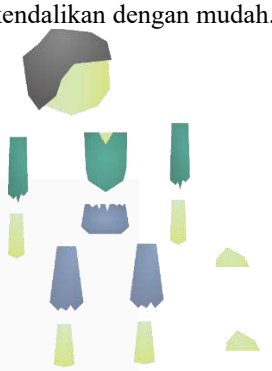


GAMBAR 10
(Tile Palette yang dibuat menggunakan Tileset)

Gambar di atas menunjukkan *Tile Palette* yang akan digunakan dalam pembuatan level.

3. Karakter

Aset karakter dibuat dalam bentuk atlas supaya animasi karakter dapat dikendalikan dengan mudah.



GAMBAR 11
(Atlas Karakter Neglected)

Gambar di atas menunjukkan aset atlas yang akan digunakan oleh karakter musuh *Neglected*.

D. Assembly

Pada tahap pembuatan, seluruh karakter menggunakan pendekatan yang sama untuk perilaku dan animasi. FSM hasil desain diimplementasikan melalui *Animator Controller* Unity dengan parameter yang dikendalikan skrip pengendali, sehingga state dan transisi animasi dikelola secara terpusat.



GAMBAR 12
(Animator Controller Musuh Neglected)

Gambar di atas menunjukkan alur logika *state machine* pada *animator controller Neglected*, yang tediri dari parameter, *state*, dan *transition*. Alur logika disusun berdasarkan diagram FSM hasil desain.

E. Testing

Pengujian transisi state karakter dalam game dilakukan dengan cara melihat perpindahan state melalui tampilan *Animator Controller*.

TABEL 3
(Hasil Uji Coba Transisi State Pada Animator Controller Neglected)

No	State	Kejadian /Parameter	State Transisi dan Behaviour	Hasil
1	Entry	Kejadian: State Awal	neg_walk	Sesuai
2	neg_walk	hasTarget = True attackCD < 0.01	neg_attack slowMove = True attacking = True	Sesuai
3	neg_attack	Animation Exit	neg_walk slowMove = False attacking = False attackCD = 0.2	Sesuai
4	Any State	hit Trigger Set isAlive = True attacking = False	neg_kb canMove = False lockVelocity = True	Sesuai
		isAlive = False	neg_dead canMove = False	Sesuai
5	neg_kb	Animation Exit	neg_walk canMove = True lockVelocity = False	Sesuai

Kemudian *Blackbox Testing* dilakukan oleh seorang ahli di dalam bidang pengembangan game melalui sesi *Alpha Testing* untuk mengetahui kualitas fungsionalitas penerapan FSM karakter.

TABEL 4
(Hasil Blackbox Testing Terhadap Perilaku Karakter Neglected)

No	Fitur	Test Case	Hasil yang diharapkan	Hasil Pengujian
1	Neglected 1	Pemain berada di depan Neglected	Neglected melancarkan serangan	Sesuai
	Neglected 2	Pemain menyerang Neglected saat tidak menyerang	Neglected terpukul mundur	Sesuai
	Neglected 3	Pemain menyerang Neglected saat menyerang	Neglected tetap menyerang	Sesuai
	Neglected 4	Healthbar Neglected kosong	Neglected mati	Sesuai

Setelah itu dilakukan *Usability Testing* terhadap game secara keseluruhan yang dilakukan oleh 24 orang responden target demografik "*Moonbase Warden*" melalui sesi *Beta Testing*.

TABEL 5
(Hasil Uji Usability Testing)

Variabel	Indikator	Penilaian				
		STS X0	TS X1	N X2	S X3	SS X4
Learnability	Menu Utama mudah dipahami	0	0	1	4	19
	Kontrol Pemain dalam permainan mudah dikendalikan	0	1	3	5	15
	Perilaku musuh mudah dipelajari	0	0	4	8	12
Efficiency	Level dapat diselesaikan dengan cepat	1	0	8	6	9
	Tutorial pada level pertama diberikan secara singkat dan padat	0	0	1	7	16
Memorability	Tutorial pada level pertama mudah diingat	0	0	2	9	13
	Kontrol Pemain mudah diingat	1	0	2	7	14
Error	Tidak ada musuh dengan perilaku yang tidak terduga	0	2	5	5	12
	Kontrol Pemain berfungsi dengan baik	0	0	3	5	16
Satisfaction	Anda menyukai artstyle game ini	0	0	2	8	14
	Tingkat kesulitan game ini cocok dengan anda	0	1	5	7	11
	Anda merasa puas saat menyelesaikan game ini	0	0	1	6	17

Untuk menghitung persentase nilai variabel, dibutuhkan rumus berikut:

$$\bar{p}_i = \frac{\sum \bar{x}_i}{4 \times n_i} \times 100\% \quad (1)$$

Keterangan
$\bar{p}_i$: Persentase rata-rata variabel
$\sum \bar{x}_i$: Total nilai rata-rata variabel
n_i : Jumlah pertanyaan variabel

Dengan 24 responden, berikut adalah persentase nilai variabel *usability*:

TABEL 6
(Nilai Rata-rata Soal)

Variabel	No	Total	Rata-rata	Persentase
Learnability	1	90	3,75	87.50%
	2	82	3,42	
	3	80	3,33	
Efficiency	4	70	2,92	81.77%
	5	87	3,63	
Memorability	6	83	3,46	85.42%
	7	81	3,38	
Error	8	75	3,13	83.33%
	9	85	3,54	
Satisfaction	10	84	3,50	86.11%
	11	76	3,17	
	12	88	3,67	
Nilai Keseluruhan				84.83%

GEQ digunakan terhadap "*Moonbase* Warden" untuk mengetahui perubahan nilai parameter penelitian yang di ambil dari 13 orang responden berusia 18-30 tahun.

TABEL 7
(Pernyataan Game Experience Questionnaire Terhadap Game Moonbase Warden)

Dimensi	Pernyataan	Nilai rata-rata
Challenge	Aku merasa ditantang oleh game ini	3.77
	Perilaku musuh membuatku berpikir dan bereaksi	3.46
Flow	Aku tidak merasa bosan saat bermain	3.31
	Aku terlalu fokus terhadap permainan	3.38
Competence	Aku dapat mengantisipasi perilaku musuh	3.23
	Aku merasa mahir saat mengalahkan musuh	3.23
Immersion	Musuh terasa hidup	3.46
	Saya merasa terhubung dengan karakter pemain	3.23
Possitive Affect	Aku menikmati permainannya	3.69
	Aku merasa puas dengan permainannya	3.77

Hasil GEQ menunjukkan bahwa secara umum responden tidak merasa bosan, dengan nilai rata-rata total pada pernyataan nomor 3 sebesar 3.31 dari nilai maksimal 4.00.

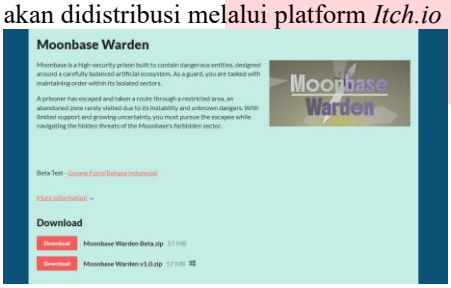
Hasil penurunan tingkat kebosanan dapat dihitung dengan rumus berikut:

$$Penurunan = \frac{Nilai\ akhir - Nilai\ awal}{4 - Nilai\ awal} \times 100\% \qquad (2)$$

Hasil menunjukkan penurunan tingkat kebosanan bermain game sebesar 52.73% dibandingkan wawancara kuesioner terhadap game sebelumnya yang hanya menghasilkan nilai rata-rata pada pernyataan nomor 3 sebesar 2.54 dari nilai maksimal 4.00.

F. Distribution

Setelah tahap pengujian selesai, game "*Moonbase Warden*" akan didistribusi melalui platform *Itch.io*



GAMBAR 13
(Halaman Itch.io Moonbase Warden)

V. KESIMPULAN

Kuesioner GEQ terhadap 13 responden menunjukkan pernyataan "Aku tidak merasa bosan saat bermain" bernilai 3,31, meningkat dari 2,54 pada game serupa, sehingga tingkat kebosanan menurun 52,73%. Penurunan ini dipengaruhi meningkatnya tantangan, tercermin dari nilai "Aku merasa ditantang oleh game ini" sebesar 3,77, akibat perilaku musuh yang lebih kompleks melalui penerapan FSM pada Animator Controller Unity. Seluruh transisi state berfungsi 100% (Alpha Testing), dengan usability 84,83% (Beta Testing).

Namun, beberapa responden masih kesulitan mengantisipasi musuh, merasa mahir, dan terhubung dengan karakter pemain (nilai terendah 3,23). Oleh karena itu, penelitian selanjutnya disarankan menambahkan indikator visual/audio perubahan state, meningkatkan kompleksitas perilaku musuh, serta memperkuat umpan balik perilaku karakter pemain agar pola perilaku lebih mudah dipahami tanpa menurunkan tantangan dan tingkat ketidakbosanan.

REFERENSI

[1] N. C. Laksmita, M. F. Filza, B. Setiaji, and Akbar, "Game Bertema Cerita Rakyat 'Si Kerudung Merah Dan Sang Serigala' dengan Metode Pengembangan Game Development Life Cycle," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 317–322, Aug. 2023, doi: 10.30865/klik.v4i1.1127.

[2] Y. Udjaja, "Orbital Dungeon Generation: A Method for Creating Dungeons Quickly and Dynamically," *Journal of Information and Communication Convergence Engineering*, vol. 20, no. 1, pp. 41–48, Mar. 2022, doi: 10.6109/jicce.2022.20.1.41.

[3] T. S. Marchelputra, H. Haryanto, K. Hastuti, and R. Kadiasti, "Non-Playable Character (NPC) based on Behaviour Tree for Enhanced Immersive Experience in the Serious Game 'Warik's Adventure,'" *Journal of Applied Informatics and Computing (JAIC)*, vol. 7, no. 2, pp. 284–290, Dec. 2023, [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[4] R. Duncan, "Using Reinforcement Learning to Train In-game Non-Player Characters (NPCs)," Apr. 2024. Accessed: Jan. 20, 2026. [Online]. Available: https://loop.cs.mtu.edu/wp-content/uploads/2024/04/Infinite_Loop_v1_Duncan_2024_Using_ReinforcementLearning_to_Train_In_game_Non_Player_Characters.pdf

[5] Q. Mi and T. Gao, "Engagement-Oriented Dynamic Difficulty Adjustment," *Applied Sciences (Switzerland)*, vol. 15, no. 10, May 2025, doi: 10.3390/app15105610.

[6] A. Junaidi, A. Yunus, and A. S. Wiguna, "Implementasi Behavior Tree Pada Perilaku NPC di Game Sidescroller," *KURAWAL Jurnal Teknologi, Informasi dan Industri*, vol. 4, no. 2, Aug. 2021.

[7] H. Jiang, "Applications of Artificial Intelligence in Game Algorithms: History, Current Status, and Future Prospects," in *Proceedings of the 2024 International Conference on Artificial Intelligence and Communication (ICAIC 2024)*, Atlantis Press, 2024, pp. 421–432. doi: 10.2991/978-94-6463-512-6_45.

[8] A. Y. Rangan and Hendy, "Penerapan Line of Sight dan Finit State Machine pada Game Platformer 'RUN!,'" *Digital Transformation Technology (Digitech) | e*, vol. 3, no. 1, Mar. 2023, doi: 10.47709/digitech.v3i1.2745.

[9] I. Rohmawati, Sudargo, and I. Menarianti, "Pengembangan Game Edukasi Tentang Budaya Nusantara 'TANARA' Menggunakan Unity 3d Berbasis Android," *Jurnal Sistem Informasi Dan Tenologi(SITECH)*, vol. 2, no. 2, 2019, [Online]. Available: http://www.jurnal.umk.ac.id/sitech

[10] T. K. Mohd, F. Bravo-Garcia, L. Love, M. Gujadhur, and J. Nyadu, "Analyzing Strengths and Weaknesses of Modern Game Engines," *International Journal of Computer Theory and Engineering*, vol. 15, no. 1, pp. 54–60, Feb. 2023, doi: 10.7763/IJCTE.2023.V15.1330.

[11] K. Strodick and T. Schattkowsky, "Visual Scripting in Unity: A Comparative Analysis of Existing Frameworks," in *Proceedings - 2024 IEEE/ACM 8th International Workshop on Games and Software Engineering, GAS 2024*, Association for Computing Machinery, Inc, Apr. 2024, pp. 44–49. doi: 10.1145/3643658.3643921.

[12] Unity Technologies, "Unity - Manual: Animator Controllers," https://docs.unity3d.com/2022.3/Documentation//Manual/AnimatorControllers.html. Accessed: Jan. 18, 2026. [Online]. Available: https://docs.unity3d.com/2022.3/Documentation//Manual/AnimatorControllers.html

[13] Ferdinand. Wagner, *Modeling software with finite state machines : a practical approach*. Auerbach, 2006.
[14] A. C. Luther, *Authoring Interactive Multimedia*. AP Professional, 1994.
[15] B. Hambling and P. Van Goethem, *User Acceptance Testing, A step-by-step guide*. BCS, 2013.
[16] P. . Markopoulos, B. de. Ruyter, W. A. . IJsselsteijn, and D. Andrew. Rowland, *Fun and Games : second international conference, Eindhoven, the Netherlands, October 20-21, 2008 : proceedings*. Springer, 2008.
[17] K. Poels, Y. A. W. De Kort, and W. A. Ijsselsteijn, "028765 FUGA The fun of gaming: Measuring the human experience of media enjoyment," Jan. 2007. [Online]. Available: http://www.hse.fi/fuga