

**PEMBANGKIT *MAP MAZE* UNTUK *GAME 3D* BERBASIS PENCAPAIAN
PLAYER DENGAN METODE *MARKOV CHAIN***

SKRIPSI

Oleh:
MOH. FADLI ROBBY
NIM. 200605110127



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**PEMBANGKIT *MAP MAZE* UNTUK *GAME 3D* BERBASIS PENCAPAIAN
PLAYER DENGAN METODE *MARKOV CHAIN***

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
MOH. FADLI ROBBY
NIM. 200605110127

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

HALAMAN PERSETUJUAN

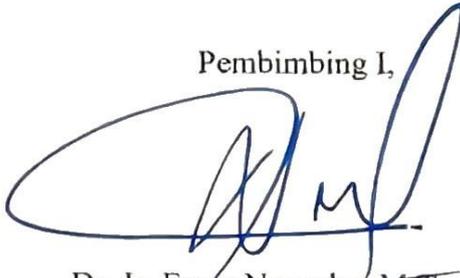
**PEMBANGKIT MAP MAZE UNTUK GAME 3D BERBASIS PENCAPAIAN
PLAYER DENGAN METODE MARKOV CHAIN**

SKRIPSI

Oleh:
Moh. Fadli Robby
NIM. 200605110087

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 02 Juni 2025

Pembimbing I,



Dr. Ir. Fresy Nugroho, M.T., IPM
NIP. 19710722 201101 1 001

Pembimbing II,



Dr. Ir. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Ir. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

PEMBANGKIT MAP MAZE UNTUK GAME 3D BERBASIS PENCAPAIAN PLAYER DENGAN METODE MARKOV CHAIN

SKRIPSI

Oleh :
Moh. Fadli Robby
NIM. 200605110127

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 17 Juni 2025

Susunan Dewan Penguji

Ketua Penguji : Dr. Yunifa Miftachul Arif, M.T.
NIP. 19830616 201101 1 004

Anggota Penguji I : Ahmad Fahmi Karami, M.Kom
NIP. 19870909 202012 1 001

Anggota Penguji II : Dr. Ir. Fresy Nugroho, M.T., IPM
NIP. 19710722 201101 1 001

Anggota Penguji III : Dr. Ir. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Ir. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Moh. Fadli Robby

NIM : 200605110127

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Pembangkit *Map Maze* Untuk Game 3D Berbasis Pencapaian
Player Dengan Metode Markov Chain

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 17 Juni 2025
Yang membuat pernyataan,



Moh. Fadli Robby
NIM.200605110127

HALAMAN MOTTO

**“Boleh Kalah
Asal
Jangan Menyerah”**

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamin, dengan penuh rasa syukur yang mendalam kepada Allah Subhanahu wa Ta'ala, karya ini saya persembahkan kepada:

Ayah dan Ummi, yang tak pernah lelah mendoakan dan mendampingi setiap langkah perjuangan ini, menjadi sumber semangat di saat lelah, dan kekuatan di saat rapuh.

Kakak saya, M. Farizal Jamil yang tak pernah lelah menasihati meski seringkali dengan nada marah namun semua itu menjadi dorongan semangat yang terus memotivasi dan mendukung saya sejak awal hingga akhir perkuliahan.

Adik saya, Ghefira Naylufarh yang dengan segala tingkah lakunya kerap mengganggu, namun justru menjadi hiburan tersendiri dalam menjalani hari-hari perkuliahan.

Dewan Asatidz Pondok Pesantren Al-Berr Pandaan, khususnya Abuya KH. Muhammad Izzuddin Muslich, atas segala ilmu, nasihat, dan bimbingan yang beliau berikan, terutama saat menghadapi ujian kehidupan. Beliau mengajarkan untuk tetap tegar dan berserah diri kepada Allah.

Para dosen dan pembimbing yang telah memberikan ilmu, bimbingan, dan arahan selama proses studi saya ini.

Semoga karya ini menjadi amal jariyah, ilmu dan memberikan manfaat bagi siapa pun yang membacanya khususnya untuk diri saya pribadi.

KATA PENGANTAR

Bismillahirrahmanirrahim,

Alhamdulillahirobbil‘alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul “Pembangkit *Map Maze* untuk Game 3D Berbasis Pencapaian *Player* dengan Metode *Markov Chain*” sebagai salah satu syarat untuk menyelesaikan studi pada Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulis ingin menyampaikan ucapan terima kasih yang besar kepada semua pihak yang selalu memberikan bantuan dan motivasi kepada penulis dalam menyelesaikan skripsi ini. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. H.M. Zainuddin, MA., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT., IPU, selaku Ketua Jurusan Teknik Informatika yang senantiasa memberikan fasilitas, program, dan motivasi untuk kelancaran penulisan skripsi.
4. Dr. Ir. Fresy Nugroho, M.T., IPM selaku Dosen Pembimbing I, yang telah memberikan motivasi, bimbingan, arahan, dan masukan yang sangat berarti dalam setiap tahapan penelitian ini.

5. Bapak Dr. Ir. Fachrul Kurniawan, M.MT, IPU selaku Dosen Pembimbing II, yang telah memberikan saran yang konstruktif, serta membantu penulis dalam memperdalam pemahaman terhadap topik penelitian.
6. Seluruh jajaran dosen dan staff Teknik Informatika yang telah membantu sekaligus memberikan ilmu selama proses kelancaran selama perkuliahan.
7. Ayah, Ummi, Kakak serta Adik saya tercinta yang selalu memberikan dukungan dan motivasi untuk terus berusaha, dan doa yang tak pernah terputus selalu disampaikan agar dapat menuntaskan perkuliahan beserta skripsi dengan lancar dan baik.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Semoga karya ini dapat menjadi bagian dari kontribusi yang berkelanjutan dalam memperkuat dan mengembangkan ilmu pengetahuan, serta menjalankan peran sebagai hamba Allah yang berkomitmen.

Malang, 17 Juni 2025

Moh. Fadli Robby

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
ABSTRAK	xiv
ABSTRACT	xv
البحث مستخلص	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	6
BAB II STUDI PUSTAKA	7
2.1 Penelitian Terkait	7
2.2 Landasan Teori.....	8
2.2.1 Game	8
2.2.2 Maze.....	9
2.2.3 Maze Generation	10
2.2.4 Kesulitan Game Adaptif.....	11
2.2.5 Tingkat Kesulitan Game	13
2.2.6 Markov Chain	13
BAB III ANALISIS DAN PERANCANGAN GAME	17
3.1 Analisis dan Perancangan	17
3.1.1 Analisis Game Maze	17
3.1.2 Perancangan Sistem Game	18
3.1.3 Perancangan Antarmuka Game.....	19
3.2 Finite State Machine (FSM).....	21
3.3 Rancangan Implementasi Metode Markov Chain.....	24
3.3.1 Desain Perhitungan Metode Markov Chain.....	24
3.3.2 Simulasi Pengujian Sistem Markov Chain	31
3.4 Desain Pengujian Sistem.....	34
BAB IV HASIL DAN PEMBAHASAN	35
4.1 Implementasi Antarmuka Game	35
4.2 Implementasi Sistem	37
4.2.1 Implementasi Sistem Maze Generator	37
4.2.2 Implementasi Sistem Markov Chain.....	40

4.2.3 Pengujian Markov Chain dengan Matlab	43
4.3 Uji Coba Game.....	44
4.4 Pengujian Sistem.....	48
4.4.1. System Usability Scale (SUS)	49
4.4.2. Game Experience Questionnaire (GEQ).....	54
4.5 Integrasi Islam.....	60
4.5.1. Muamalah Ma'Allah	60
4.5.2. Muamalah Ma'annas.....	61
4.5.3. Muamalah Ma'alam	61
BAB V KESIMPULAN DAN SARAN	63
5.1 Kesimpulan	63
5.2 Saran.....	64
DAFTAR PUSTAKA	66

DAFTAR TABEL

Tabel 2.1 Jurnal Penelitian Terdahulu	7
Tabel 3.1 Parameter Level	25
Tabel 3.2 Variabel Input	27
Tabel 3.3 Hasil Pembagian Variabel.....	28
Tabel 3.4 Pengujian Normalisasi Input Data	31
Tabel 3.5 Pengujian Normalisasi Bobot Input Data.....	32
Tabel 4.1 Perbandingan Output Markov Chain dengan Matlab	44
Tabel 4.2 Versi standar pertanyaan dari SUS	49
Tabel 4.3 Demografi Responden.....	50
Tabel 4.4 Hasil Kuisisioner SUS	51
Tabel 4.5 Hasil Total Kuisisioner SUS.....	52
Tabel 4.6 Hasil Akhir Skor SUS	53
Tabel 4.7 Pertanyaan Core Module.....	56
Tabel 4.8 Pertanyaan Post-Game Module.....	56
Tabel 4.9 Hasil Komponen Core Module	57
Tabel 4.10 Hasil Komponen Post Game Module	57
Tabel 4.11 Korelasi Kuisisioner SUS dan GEQ	58

DAFTAR GAMBAR

Gambar 2.1 Jenis-jenis Maze.	10
Gambar 3.1 Game Menu Flow.....	18
Gambar 3.2 Rancangan tampilan Menu Utama	20
Gambar 3.3 Rancangan tampilan Pemilihan Level.....	20
Gambar 3.4 Rancangan tampilan Tentang Game	21
Gambar 3.5 Desain FSM.....	23
Gambar 3.6 Desain Sistem Markov	24
Gambar 4.1 Tampilan (UI) Menu	36
Gambar 4.2 Tampilan (UI) Pemilihan Level	36
Gambar 4.3 Tampilan (UI) Seputar Game.....	37
Gambar 4.4 Beberapa Bentuk Maze Generator Level 2	45
Gambar 4.5 Tampilan Misi Sukses	46
Gambar 4.6 Tampilan Dokumen Sejarah.....	46
Gambar 4.7 Beberapa Bentuk Maze Generator Level 7	47
Gambar 4.8 Tampilan Misi Gagal.....	48

ABSTRAK

Robby, Moh. Fadli. 2025. **Pembangkit Map Maze Untuk Game 3D Berbasis Pencapaian Player Dengan Metode *Markov Chain***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) Dr. Ir. Fresy Nugroho, M.T., IPM (II) Dr. Ir. Fachrul Kurniawan, M.MT., IPU

Kata Kunci: Maze, Markov Chain, Game 3D, Tingkat Kesulitan Adaptif, GEQ

Penelitian ini merancang dan mengembangkan game edukatif 3D berjudul *The Lost Histories* yang dilengkapi dengan sistem pembangkit maze adaptif berbasis metode Markov Chain. Sistem ini secara otomatis menyesuaikan tingkat kesulitan permainan berdasarkan performa pemain, dengan mengacu pada lima parameter utama: skor, waktu penyelesaian, jumlah nyawa, total musuh yang dikalahkan, dan item yang berhasil dikumpulkan. Penilaian dilakukan menggunakan metode System Usability Scale (SUS) dan Game Experience Questionnaire (GEQ) untuk mengukur aspek kegunaan serta pengalaman bermain. Hasil pengujian menunjukkan bahwa game memiliki tingkat usability yang baik, dengan nilai rata-rata SUS sebesar 73,6 yang masuk dalam kategori “Good”. Sementara itu, evaluasi GEQ memperlihatkan bahwa pemain merasakan pengalaman bermain yang menyenangkan, menantang, dan imersif. Temuan ini membuktikan bahwa penerapan pendekatan adaptif dalam desain permainan mampu meningkatkan kenyamanan dan kepuasan pemain secara signifikan.

ABSTRACT

Robby, Moh. Fadli 2025. **Maze Map Generator for 3D Game Based on Player Achievement Using Markov Chain Method**. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisor (I) Dr. Ir. Fresy Nugroho, M.T., IPM, (II) Dr. Ir. Fachrul Kurniawan, M.MT., IPU.

This study designs and develops a 3D educational game titled The Lost Histories, featuring an adaptive maze generation system based on the Markov Chain method. The system dynamically adjusts the game's difficulty level according to player performance, using five key parameters: score, completion time, number of lives, number of defeated enemies, and collected items. Evaluation was conducted using the System Usability Scale (SUS) and the Game Experience Questionnaire (GEQ) to assess usability and gameplay experience. The test results show that the game achieved a good level of usability, with an average SUS score of 73.6, categorized as "Good." Additionally, the GEQ results indicated that players experienced an enjoyable, challenging, and immersive gameplay. These findings demonstrate that the adaptive approach effectively enhances player comfort and satisfaction throughout the game.

Keywords: Maze, Markov Chain, 3D Game, Adaptive Difficulty, GEQ

البحث مستخلص

رَبِّي، محمد فضل . ٢٠٢٥ . مولد خريطة المتاهة للعبة ثلاثية الأبعاد بناءً على إنجازات اللاعب باستخدام طريقة سلسلة ماركوف . مشروع تخرج لنيل درجة البكالوريوس . قسم هندسة المعلومات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية – مالانج . المشرف (١) الدكتور المهندس فريسي نوغروهو، إم.تي (٢) الدكتور المهندس فخر الكورنياوان إم.إم.تي .

الكلمات المفتاحية: المتاهة، سلسلة ماركوف، لعبة ثلاثية الأبعاد، صعوبة تكتيفية، GEQ

مزودة بنظام توليد متاهة، The Lost Histories تصمم هذه الدراسة وتطور لعبة تعليمية ثلاثية الأبعاد بعنوان يقوم النظام بضبط مستوى صعوبة اللعبة ديناميكياً (Markov Chain) تكتيفي يعتمد على طريقة سلسلة ماركوف استناداً إلى أداء اللاعب، معتمداً على خمسة معايير رئيسية: النقاط، وقت الإنهاء، عدد الأرواح، عدد الأعداء (System Usability Scale) المهزومين، والعناصر المجمعمة. تم إجراء التقييم باستخدام مقياس قابلية الاستخدام لقياس جوانب سهولة (Game Experience Questionnaire - GEQ) واستبيان تجربة اللعب (SUS - الاستخدام وتجربة اللعب. أظهرت نتائج الاختبار أن اللعبة حققت مستوى جيداً من قابلية الاستخدام، حيث بلغ متوسط أن اللاعبين خاضوا تجربة لعب GEQ حوالي 73.6، مما يضعها في فئة "جيدة". كما أظهرت نتائج SUS درجة ممتعة، وتحدياً، وانغماس. وتُظهر هذه النتائج أن النهج التكتيفي يُسهم بشكل فعّال في تعزيز راحة ورضا اللاعبين أثناء اللعب.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Industri permainan berkembang pesat dalam beberapa tahun terakhir, termasuk di Indonesia. Awalnya, pasar game Indonesia tergolong kecil, tetapi kini mengalami pertumbuhan signifikan (Yuwono, 2021). Permainan baru dengan grafis berkualitas, fitur inovatif, dan narasi yang menarik semakin banyak bermunculan. Perkembangan ini mencakup game tradisional seperti permainan papan hingga video game lintas platform. (Sudrajat, 2022)

Dengan berkembangnya industri *game* secara global, terutama di Indonesia, minat terhadap permainan digital semakin meningkat pesat. Indonesia telah diakui sebagai salah satu kontributor terbesar dalam industri *game* di Asia Tenggara, dengan sejumlah kurang lebih 256 game dibuat oleh *developer* lokal di platform Steam hingga bulan Januari 2024 (Katadata, 2024). Hal ini mencerminkan pertumbuhan yang signifikan dalam industri *game* di Indonesia dan menandakan potensi yang besar bagi pengembang *game* lokal untuk bersaing secara global.

Dalam industri *game* modern, pengalaman bermain yang menarik dan menantang menjadi fokus utama pengembang (Dirgantara et al., 2023). Labirin atau *maze* seringkali menjadi elemen desain yang digunakan untuk menambahkan tingkat kesulitan yang tepat serta meningkatkan tingkat ketegangan dan kepuasan *player* (Krisdiawan et al., 2022).

Genre *maze* atau labirin merupakan permainan dimana *player* harus menavigasi melalui labirin yang kompleks dengan tujuan mencapai titik keluar atau mencapai tujuan tertentu di dalamnya (Setiadharna et al., 2020). *Player* harus menghindari berbagai rintangan dan jebakan yang tersebar di sepanjang jalur labirin. Jika *player* tersandung pada rintangan atau tersesat dalam labirin, mereka mungkin akan kalah atau perlu memulai kembali dari titik awal. Tujuan utama *player* dalam *game maze* adalah menyelesaikan labirin dengan cepat dan efisien untuk mencapai skor tertinggi atau memenangkan level.

Pembuatan *map generator* labirin dalam game merupakan tantangan tersendiri. Algoritma yang digunakan harus mampu menghasilkan labirin yang bervariasi dan menantang (Susanto et al., 2020). Tingkat kesulitan labirin yang dihasilkan harus dapat disesuaikan sesuai dengan preferensi dan tingkat keterampilan *player*. Pada game 3D, labirin juga perlu disesuaikan dengan keterbatasan rendering agar tampilannya tetap menarik dan optimal (Habibullah, 2024). Selain itu, penentuan tingkat kesulitan pada *game* merupakan aspek krusial yang dapat mempengaruhi pengalaman bermain *player*.

Dalam pengembangan *game*, terutama dalam perancangan *game* labirin, peningkatan keseruan permainan memiliki dampak yang besar terhadap desain dan pembuatan *game* (Habibullah, 2024). Salah satu strategi yang banyak digunakan adalah penggunaan *gameplay* adaptif. Pendekatan ini memungkinkan interaksi yang lebih mendalam dengan *player* dengan memprediksi niat mereka berdasarkan tindakan sebelumnya, daripada sekadar merespons tindakan tingkat dasar (Soler-Dominguez et al., 2017). Untuk mempertahankan minat *player* dan memberikan

tantangan yang sesuai dengan kemampuan individu, penting untuk menentukan tingkat kesulitan yang tepat dalam permainan. Dengan begitu, pengalaman bermain akan sesuai dengan tingkat keterampilan *player*. Ini melibatkan penyesuaian dinamis dari tantangan dalam permainan untuk menciptakan aliran yang tepat, di mana *player* merasa tertantang namun tidak terlalu frustrasi (Haryanto, 2016).

Salah satu metode adaptif yang menarik dalam pembangkitan labirin adalah pendekatan Markov Chain. Metode ini menggunakan teknik probabilistik yang memodelkan perubahan keadaan sistem, di mana kondisi saat ini hanya bergantung pada kondisi sebelumnya (Azizah et al., 2019). Pendekatan ini memungkinkan labirin dibangkitkan secara acak berdasarkan model probabilitas yang telah ditentukan sebelumnya. Dengan demikian, labirin yang dihasilkan memiliki variasi struktur yang menarik bagi *player*. Metode *Markov Chain* menawarkan beberapa keunggulan, terutama dalam menangani permasalahan integral multidimensi yang rumit yang seringkali sulit untuk diselesaikan secara langsung. Selain itu, sekaligus memberikan solusi yang memadai melalui perhitungan yang dilakukan dalam rangkaian Markov (Haario et al., 2005).

Oleh karena itu, peneliti ingin mengembangkan sebuah *game* berjudul "*The Lost Histories*" yang merupakan sebuah *game* 3D yang dirancang untuk menantang *player* dengan labirin yang dihasilkan secara dinamis menggunakan metode rantai Markov serta mengandung nilai edukasi berupa peristiwa-peristiwa yang terjadi selama Kemerdekaan Indonesia. Game ini menggunakan teknologi pembangkitan maze atau labirin dengan tingkat kesulitan yang dapat disesuaikan, sehingga menciptakan pengalaman bermain yang unik setiap kali. Dengan algoritma Markov

Chain, tantangan labirin terasa segar dan tidak monoton. Selain itu, game ini juga memuat nilai edukatif berupa semangat perjuangan dan pengetahuan sejarah Indonesia yang akan diperoleh *player* setelah menyelesaikan level tertentu. Sebagai relevansi dengan penelitian ini, Al-Qur'an menyatakan dalam salah satu potongan ayat di dalam Al-Quran berikut:

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ

"Allah Swt. tidak membebani seseorang melainkan sesuai dengan kesanggupannya. Ia mendapat pahala dari (kebaikan) yang dikerjakannya dan ia mendapat siksa dari (kejahatan) yang diperbuatnya." (QS. Al-Baqarah: 286)

Menurut Prof. Dr. AG. K.H Muhammad Quraish Shihab, Lc., M.A., kitab tafsir Al-Misbah, ayat ini menjelaskan bahwasanya sejatinya Allah menetapkan aturan dan beban kewajiban sesuai dengan kapasitas manusia. Tidak ada aturan yang mustahil untuk dipatuhi, sebagaimana tidak ada ujian yang tidak mungkin dihadapi oleh hamba-Nya. Keberhasilan seseorang dalam menghadapi tantangan bergantung pada usaha dan ketekunan mereka dalam (Shihab Quraish, 2002). Dalam konteks penelitian ini, metode *Markov Chain* digunakan dalam pengembangan game untuk memungkinkan tingkat kesulitan yang adaptif, di mana tantangan dalam permainan akan selalu disesuaikan dengan kemampuan *player*. Hal ini sejalan dengan prinsip dalam ayat ini, yang mengajarkan bahwa beban atau tantangan harus sesuai dengan kapasitas seseorang. Dengan demikian, sistem ini tidak hanya memberikan pengalaman bermain yang lebih dinamis dan menarik, tetapi juga mengajarkan konsep usaha yang berkelanjutan dalam menghadapi tantangan, sebagaimana dalam kehidupan nyata. Harapannya, game ini tidak hanya

berkontribusi dalam pengembangan teknologi game, tetapi juga memberikan nilai edukatif yang bermakna bagi *player*.

1.2 Identifikasi Masalah

Berdasarkan latar belakang dan permasalahan yang telah dipaparkan diatas maka dapat ditarik kesimpulan bahwa rumusan masalah dalam penelitian ini adalah sebagai berikut.

Bagaimana mengembangkan sistem pembangkit *map* labirin untuk *game* 3D agar menghasilkan tingkat kesulitan level labirin yang adaptif dan menarik bagi *player* melalui penerapan metode *Markov Chain*?

1.3 Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan masalah dari penelitian ini sebagai berikut.

1. Ukuran grid terbatas dari ukuran 6x6 untuk level 1 dan bertambah naik hingga ukuran 15x15 untuk level 10.
2. Terdapat 5 variabel yang digunakan yaitu skor, waktu, nyawa *player*, musuh dan dokumen.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengembangkan sistem pembangkit *map* berupa labirin untuk *game* 3D edukasi yang mampu menghasilkan atau membuat

labirin yang adaptif sesuai dengan pencapaian *player* yang berbeda-beda melalui penerapan metode *Markov Chain*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat berkontribusi pada pengembangan game yang lebih dinamis dan menarik, melalui sistem pembangkitan labirin yang variatif dan menantang. Selain itu, teknologi ini juga berpotensi dikembangkan lebih lanjut dan diterapkan pada berbagai jenis game, baik 2D maupun 3D.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Hasil dari penelitian terdahulu dapat dijadikan panduan untuk mempelajari lebih jauh kelebihan dan kekurangan pada komponen tertentu. Penelitian ini akan berhubungan dengan beberapa penelitian terkait yang mencakup tingkat kesulitan dan *maze* game. Dalam Tabel 2.1 terdiri dari beberapa penelitian yang berkaitan dengan penelitian yang sedang di lakukan.

Tabel 2.1 Jurnal Penelitian Terdahulu

No	Peneliti	Judul	Persamaan	Perbedaan
1.	(Jalu Kinayun et al., 2024)	Rancang Bangun <i>Game Road Maze</i> 3D	<i>Generator Maze</i>	Menerapkan metode FSM untuk membangun sistem <i>maze</i> .
2.	(Habibullah, 2024)	<i>Procedural Maze Modeling in 3D For Game Asset</i>	<i>Generator Maze</i>	Bertujuan untuk mengembangkan alat dalam bentuk plug-in untuk game engine yang dapat membuat aset game dalam bentuk labirin.
3.	(Krisdiawan et al., 2022)	Penerapan Algoritma <i>Recursive Backtracking</i> Sebagai <i>Maze Generator</i> Pada Game Labirin Aksara Sunda	<i>Generator Maze</i>	Menerapkan algoritma <i>Recursive Backtracking</i> untuk menghasilkan level dan <i>map</i> labirin yang dinamis.
4.	(Peachey, 2022)	<i>Parameterized Maze Generation Algorithm for Specific Difficulty Maze Generation</i>	<i>Generator Maze</i>	Menggunakan kombinasi 3 metode yaitu <i>Neural Network</i> , <i>Parameterized Maze Generation</i> , dan <i>Difficulty Model</i> untuk menghasilkan <i>maze</i> dengan tingkat kesulitan tertentu

5.	(Colwell & Glavin, 2018)	<i>Colwell's castle defence: A custom game using dynamic difficulty adjustment to increase player enjoyment</i>	Level Adaptif	Menggunakan metode DDA (<i>Dynamic Diffilcuty Adaptive</i>) untuk menentukan tingkat kesulitan.
----	--------------------------	---	---------------	---

Penelitian-penelitian sebelumnya menunjukkan berbagai pendekatan dalam pengembangan maze dan sistem tingkat kesulitan dalam game. Kinayun et al. (2024) menerapkan metode Finite State Machine (FSM) untuk membangun sistem maze secara terstruktur, sementara Habibullah (2024) mengembangkan plug-in procedural untuk pembuatan aset labirin 3D dalam game engine. Krisdiawan et al. (2022) menggunakan algoritma Recursive Backtracking untuk menghasilkan maze yang dinamis. Peachey (2022) memadukan metode Parameterized Maze Generation, Neural Network, dan Difficulty Model guna menciptakan maze dengan kesulitan tertentu, dan Colwell & Glavin (2018) mengembangkan game dengan sistem Dynamic Difficulty Adjustment (DDA) untuk menyesuaikan tingkat tantangan dengan performa pemain. Meskipun pendekatan yang digunakan bervariasi, kelima penelitian tersebut sepakat bahwa adaptivitas dan variasi maze menjadi faktor penting dalam meningkatkan keterlibatan dan pengalaman pemain dalam game.

2.2 Landasan Teori

Beberapa landasan teori dalam penelitian yang akan dibuat adalah sebagai berikut.

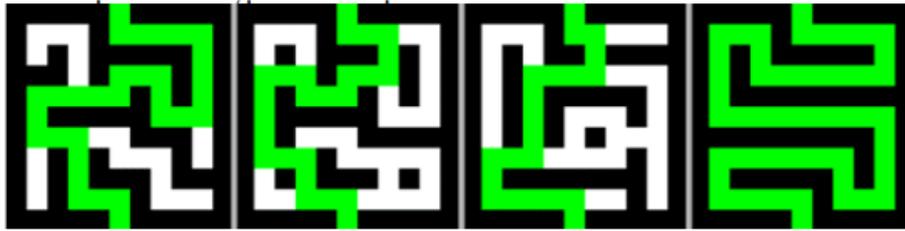
2.2.1 *Game*

Dalam kamus besar bahasa Indonesia, kata "Game" atau "Gim" berarti "permainan". Game adalah permainan interaktif yang membutuhkan perangkat

lunak atau komputer untuk bermain. Siswa diharapkan untuk menikmati permainan sehingga mereka tertarik untuk belajar dan dapat melatih kemampuan motorik anak, biasanya *game* sangat diminati oleh anak-anak remaja dan orang dewasa (Rahmawati & Leksono, 2020). *Game* merupakan salah satu bentuk hiburan yang bisa dijadikan sarana untuk bermain. Karena setiap *game* memiliki peralatan dan aturan yang spesifik. Hal ini membuat *player* perlu memanfaatkan keterampilan, strategi, kesempatan, dan keberuntungan (Irwandi et al., 2016).

2.2.2 Maze

Menurut (Setiadharna et al., 2020) maze atau labirin adalah teka-teki dengan jalur yang berliku dan banyak jalan buntu, di mana *player* harus mencari jalan keluar. Permainan labirin menuntut kemampuan navigasi dan pemecahan masalah. Variasi kompleksitas labirin mendorong *player* berpikir cepat dan beradaptasi dengan rintangan. Tantangan dalam menavigasi labirin, terkadang disertai dengan waktu terbatas atau skor, menjadikan permainan labirin sebagai pilihan yang menghibur untuk mengasah keterampilan *player* dalam mencari jalan keluar dari situasi yang rumit. Berdasarkan jenisnya, *maze* bisa dibagi menjadi 4 macam jenis seperti terlihat pada gambar berikut (Habibullah, 2024).



Gambar 2.1. Jenis-jenis *Maze*. Dari kiri ke kanan: *Perfect*, *Braid*, *Partial Braid*, dan *Unicursal*

1. *Perfect Maze*: Jenis *maze* ini menjamin hanya ada satu jalur yang menghubungkan dua titik berbeda di dalam *maze*. Tidak ada loop, sirkuit tertutup, atau area yang tidak dapat diakses, sehingga hanya ada satu solusi unik untuk menyelesaikan *maze*.
2. *Braid Maze*: *Maze* ini tidak memiliki jalan buntu dan mengandung banyak loop. Karena adanya loop, terdapat lebih dari satu jalur yang dapat ditempuh untuk mencapai suatu titik, sehingga *maze* ini memiliki banyak solusi.
3. *Partial Braid Maze*: *Maze* ini merupakan campuran antara *perfect maze* dan *braid maze*, yang memiliki kombinasi jalan buntu dan loop. Dengan demikian, terdapat beberapa jalur solusi yang memungkinkan.
4. *Unicursal Maze*: *Maze* ini hanya memiliki satu jalur tanpa persimpangan. Untuk menyelesaikannya, *player* hanya perlu mengikuti jalur yang tersedia dari awal hingga akhir, tanpa adanya opsi cabang atau jalan buntu.

2.2.3 *Maze Generation*

Maze generator adalah alat yang digunakan untuk membuat labirin secara acak dan menghasilkan labirin yang berbeda-beda pada setiap level permainan (Habibullah, 2024) Dalam sistem yang akan dikembangkan, pembuatan labirin

dimulai dengan menentukan panjang dan lebar labirin, karena semua labirin yang dibuat berbentuk persegi panjang. Selanjutnya, bagi persegi panjang tersebut menjadi kotak berukuran satu kali satu, dan letakkan sebuah titik di pusat setiap kotak. Maze generator akan membuat labirin secara acak dan menghasilkan labirin yang berbeda dengan sebelumnya ketika pemain memulai permainan baru. Ini memastikan bahwa setiap permainan menawarkan tantangan yang unik dan tidak terduga.

2.2.4 Kesulitan *Game* Adaptif

Kesulitan *Game* Adaptif merupakan mekanisme yang memungkinkan permainan untuk menyesuaikan tingkat kesulitan secara dinamis berdasarkan performa *player*. Menurut (Hunicke, 2005) pendekatan ini bertujuan untuk menciptakan pengalaman bermain yang optimal dengan menjaga keseimbangan antara tantangan dan keterampilan *player*. Ketika *player* terlihat kesulitan menyelesaikan level, sistem adaptif dapat menurunkan kesulitan, misalnya dengan mengurangi jumlah musuh atau memperpanjang waktu yang tersedia. Sebaliknya, jika *player* bermain dengan sangat baik, sistem dapat meningkatkan tantangan untuk menjaga permainan tetap menarik, seperti meningkatkan kecepatan musuh atau memperbanyak rintangan.

Adaptasi kesulitan dalam permainan dapat meningkatkan keterlibatan *player*, karena *player* merasa permainan menantang tanpa terlalu frustrasi (Hendrix et al., 2019). Dengan adanya adaptasi kesulitan, *player* dari berbagai tingkat keahlian dapat menikmati permainan yang sesuai dengan kemampuan mereka, sehingga meningkatkan kepuasan dan durasi bermain (Chen et al., 2022).

Memfaatkan prinsip *Dynamic Difficulty Adjustment*, Penelitian (Nugroho et al., 2023) menunjukkan bahwa variabel seperti ketebalan kabut sintetis dapat diubah secara real-time menyesuaikan parameter pemain untuk mempertahankan tingkat tantangan yang optimal tanpa mengakibatkan kebosanan atau frustrasi. Temuan ini mendukung bahwa pendekatan adaptif baik menggunakan Markov Chain maupun mekanisme eksternal seperti kabut mampu menghasilkan pengalaman bermain yang responsif dan seimbang.

Dalam penelitian ini, *Markov Chain* dapat digunakan untuk memprediksi performa *player* berdasarkan data dari tindakan-tindakan sebelumnya. Hal ini memungkinkan sistem untuk menentukan tingkat kesulitan yang sesuai dengan kemampuan *player* saat ini. Metode *Markov Chain* dalam adaptasi tingkat kesulitan dapat digunakan untuk mengenali pola dalam performa *player* dan kemudian menyesuaikan level permainan berikutnya agar tetap menantang namun adil (Zuin & Macedo, 2016). Dengan demikian, pendekatan berbasis *Markov Chain* dalam penelitian ini menunjukkan bagaimana model probabilistik dapat membantu dalam merespons secara adaptif terhadap perilaku *player* untuk menjaga keseimbangan dan tantangan dalam permainan (Zamith et al., 2020).

Penelitian oleh (Arif et al., 2023) menunjukkan bahwa penggunaan *Finite State Machine* berbasis ANN mampu memilih skenario game adaptif sesuai preferensi pemain termasuk minat, asal, dan kebiasaan dengan tingkat akurasi yang signifikan. Hal ini memperkuat temuan bahwa metode adaptasi yang kami gunakan, meski berbasis Markov Chain, sejalan dalam memberikan pengalaman yang personal dan responsif

2.2.5 Tingkat Kesulitan Game

Teknologi ini memiliki kemampuan untuk menyesuaikan tingkat penyesuaian, adegan, dan perilaku permainan secara real-time atau dinamis sesuai dengan kemampuan dan keterampilan pemain (Zamith et al., 2020). Jika digunakan dengan benar, teknik ini dapat membantu pengalaman bermain pemain secara keseluruhan karena memberi mereka sensasi baru setiap kali bermain dan mencegah mereka bosan jika permainan terlalu sulit, terlalu mudah, atau membuat frustrasi (Colwell & Glavin, 2017).

Di sisi lain, seiring bertambahnya level dalam *game* khususnya bergenre *puzzle*, waktu yang disediakan untuk menyelesaikan setiap level semakin sedikit (Shah et al., 2017). Hal ini menambah tekanan pada *player* untuk bertindak lebih cepat dan efisien (Zuin & Macedo, 2016). Selain itu, jumlah musuh yang dihadapi dan dokumen yang perlu dikumpulkan juga bertambah, menambah tantangan yang harus dihadapi. Peningkatan bertahap ini dirancang untuk menjaga keseimbangan permainan, memberikan tantangan yang memadai, serta mendorong *player* agar terus bermain dan meningkatkan keterampilan mereka (Hendrix et al., 2019).

2.2.6 Markov Chain

Analisis Markov chain atau rantai markov adalah teknik untuk memperkirakan variabel di waktu yang akan datang dengan melihat bagaimana karakteristiknya di masa lalu. Matematikawan asal Rusia, Andrej A. Markov (1856–1922) menciptakan konsep dasar rantai Markov baru pada tahun 1907. Salah satu alat yang dapat digunakan untuk membantu dalam pengambilan

keputusan adalah analisis rantai markov. Proses stokastik adalah model probabilitas yang digunakan dalam analisis rantai Markov (Thompson et al., 1985).

a. Proses Stokastik

Proses stokastik $\{X_t, t \in T\}$ adalah himpunan variabel acak X_t , yang terdefinisi pada suatu ruang sampel, dimana t adalah satuan waktu T . Jika parameter T adalah himpunan terhingga, proses tersebut disebut proses stokastik dengan waktu diskrit sedangkan jika parameter T adalah himpunan waktu kontinu, maka proses disebut proses stokastik dengan waktu kontinu (Thompson et al., 1985).

b. Proses Markov

Proses Markov $\{X_t\}$ adalah proses stokastik dengan sifat bahwa, jika diberikan nilai X_t , maka nilai X_s untuk $s > t$ tidak dipengaruhi oleh nilai-nilai dari X_u untuk $u < t$. Dengan kata lain, peluang perilaku tertentu di masa depan dari suatu proses, ketika diketahui state saat ini, tidak dapat dipengaruhi oleh informasi tambahan di masa yang lalu (Ross, 1966).

c. Rantai Markov dan Waktu Diskrit

Rantai Markov diskrit adalah suatu proses Markov dimana ruang *state*-nya adalah terhingga atau himpunan yang terhingga dengan himpunan indeks $T = (0, 1, 2, \dots)$. Secara umum, sifat Markov adalah seperti berikut.

$$\Pr\{X_{t+1} = j | X_0 = i_0, \dots, X_t = i_t, X_t = i\} = \Pr\{X_{t+1} = j | X_t = i\} \quad (2.1)$$

Artinya, probabilitas berada di state j pada waktu $t+1$ hanya bergantung pada state i pada waktu t , dan tidak tergantung pada state sebelumnya. Kemudian untuk semua t dan semua *state* $i_0, \dots, i_{t-1}, i, j$. Ruang *state* pada rantai Markov dinyatakan dalam

bilangan bulat tak negatif $\{0, 1, 2, \dots\}$, dengan $X_t = i$ menyatakan bahwa X_t berada di *state* i (Ross, 1966).

d. Probabilitas Transisi

Peluang dari X_{t+1} dimana berada di *state* j jika diberikan X_t sedang berada di *state* i disebut peluang transisi satu langkah dan dinotasikan dengan $P_{ij}^{\wedge}(t, t + 1)$, sehingga

$$P_{ij}^{\wedge}(t, t + 1) = Pr \{X_{t+1} = j | X_t = i\} \quad (2.2)$$

Rantai markov memiliki peluang transisi stasioner ketika ada peluang transisi satu langkah independen terhadap variabel waktu t . Maka $P_{t,t+1} = P_{ij}$ independen terhadap t dan P_{ij} adalah peluang bersyarat sehingga nilai *state* melalui sebuah transisi dari i ke j dalam satu langkah (Ross, 1966). Nilai peluang transisi tersebut biasanya disusun ke dalam bentuk matriks seperti berikut.

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \dots & P_{0j} \\ P_{10} & P_{11} & P_{12} & \dots & P_{1j} \\ P_{20} & P_{21} & P_{22} & \dots & P_{2j} \\ \dots & \dots & \dots & \ddots & \dots \\ P_{i0} & P_{i1} & P_{i2} & \dots & P_{ij} \end{bmatrix} \quad (2.3)$$

Keterangan $P = (P_{ij})$ merupakan matriks peluang transisi dari suatu proses Nilai P_{ij} harus memenuhi kondisi berikut.

- a. $0 < P_{ij} < 1$ untuk seluruh $i, j = 0, 1, 2, \dots$
- b. $\sum_{j=0}^{\infty} P_{ij} = 1$ untuk seluruh $i = 0, 1, 2, \dots$

e. Normalisasi Data

Dalam beberapa konteks penerapan *Markov Chain* memerlukan normalisasi data sebelum dihitung dengan Matriks Probabilitas. Normalisasi data adalah proses mengubah nilai dari berbagai variabel menjadi skala yang seragam untuk

meningkatkan akurasi dalam pemodelan matematika dan n (Han et al., 2011). Normalisasi diperlukan ketika dataset memiliki skala variabel yang berbeda agar tidak terjadi bias dalam perhitungan. Salah satu metode normalisasi yang umum digunakan adalah Min-Max Normalization (Ching et al., 2004), yang merubah nilai ke dalam rentang 0 hingga 1 dengan rumus:

$$X_i = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.4)$$

Dalam penelitian ini, normalisasi data diterapkan sebelum perhitungan *Markov Chain*, guna memastikan bahwa semua parameter (skor, waktu, nyawa, musuh dan dokumen) memiliki skala yang sama. Hal ini bertujuan agar perhitungan probabilitas transisi dalam *Markov Chain* lebih akurat dan adil.

BAB III

ANALISIS DAN PERANCANGAN GAME

Bab ini menjelaskan tahapan desain, analisis, dan perancangan sistem yang dilakukan sebagai bagian dari metodologi penelitian, yaitu pendekatan yang berfokus pada pembangunan solusi nyata terhadap permasalahan yang diidentifikasi. Metodologi ini dimulai dari analisis kebutuhan pengguna, studi literatur terhadap teknik adaptif seperti Markov Chain, serta eksplorasi teknologi pengembangan game edukatif berbasis 3D.

Setelah analisis dilakukan, tahap perancangan mencakup pemodelan arsitektur sistem, logika permainan dengan Finite State Machine (FSM), serta perancangan antarmuka dan pengalaman pengguna (UX). Selain itu, pembangkitan maze adaptif dirancang menggunakan pendekatan matematis berbasis probabilitas (Markov Chain), yang disesuaikan dengan parameter dinamis seperti waktu, skor, dan nyawa pemain. Perancangan ini bertujuan agar game yang dikembangkan tidak hanya bersifat menghibur, tetapi juga memiliki nilai edukatif dan adaptif terhadap kemampuan pemain. Tahapan-tahapan yang disusun dalam bab ini menjadi kerangka dasar sebelum sistem diimplementasikan dan diuji pada bab selanjutnya.

3.1 Analisis dan Perancangan

3.1.1 Analisis *Game Maze*

Game “*The Lost Histories*” adalah *game maze* 3D dalam genre petualangan berbentuk *Third-Person Shooter* (TPS). Dalam permainan ini, *player* memulai perjalanan menyusuri setiap labirin di masing-masing level untuk mencari jalan keluar dan menemukan kembali potongan-potongan sejarah perjuangan bangsa

Indonesia yang menghilang. *Player* harus menemukan jalur keluar dari labirin sambil mengumpulkan item sejarah, menyelesaikan misi, dan menghadapi musuh. Fitur utama game ini adalah penggunaan metode Markov Chain, yang memungkinkan sistem secara dinamis menyesuaikan tingkat kesulitan berdasarkan hasil permainan sebelumnya. Selain itu, fitur menarik game adalah edukasi yang disajikan melalui item yang harus dikumpulkan untuk membuka dokumen berupa edukasi sejarah Kemerdekaan Indonesia yang relevan dengan pelajaran siswa di jenjang SMP/Sederajat mengutip dari (Abdurakhman; Setiawan, 2013) Dengan cara ini, setiap *player* selain mendapatkan pengalaman yang unik dan menantang, sekaligus mengenal kembali jejak sejarah bangsa Indonesia.

3.1.2 Perancangan Sistem Game

Dalam game “*The Lost Histories*”, terdapat beberapa menu pada tampilan awal menu utama seperti dapat dilihat pada gambar 3.1 berikut.



Gambar 3.1 Game Menu Flow

Diagram ini menampilkan berbagai opsi yang dapat dipilih *player* dari menu utama. Penjelasan masing-masing komponen disajikan pada Gambar 3.1. berikut.

1. Mulai Perjuangan : Opsi ini membawa *player* ke tahap pemilihan level. Setelah memilih level, *player* akan memulai gameplay.

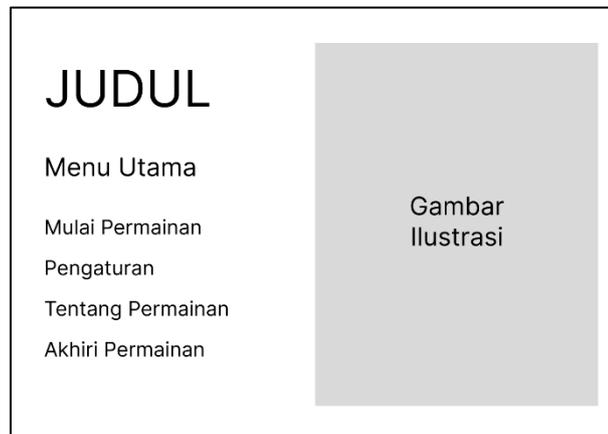
2. Pengaturan : Dalam bagian ini, *player* dapat mengatur pengaturan volume game. Opsi ini memberikan fleksibilitas kepada *player* untuk menyesuaikan pengalaman bermain sesuai dengan preferensi audio mereka.
3. Tentang *Game* : Opsi ini memberikan informasi tentang game, seperti latar belakang, tujuan, dan informasi penting lainnya yang berkaitan dengan game.
4. Akhiri Perjuangan : *Player* dapat memilih opsi ini untuk keluar dari game, yang akan menutup aplikasi.

Diagram ini secara keseluruhan menunjukkan alur navigasi dari menu utama *game* dan interaksi *player* dengan berbagai fitur yang tersedia. Ini membantu dalam memahami bagaimana *player* dapat berinteraksi dengan game dan fitur-fitur yang ada.

3.1.3 Perancangan Antarmuka Game

Antarmuka dalam game memiliki peran penting dalam memberikan pengalaman interaktif yang intuitif dan menarik bagi pemain. Perancangan antarmuka dilakukan secara terstruktur agar setiap elemen visual dapat memberikan informasi yang jelas serta mendukung navigasi dan penggunaan fitur dalam game. Pada game *The Lost Histories*, antarmuka dirancang untuk menampilkan menu utama, tampilan permainan, dan elemen-elemen pendukung lainnya secara sederhana namun fungsional. Berikut ini adalah hasil rancangan antarmuka yang telah disesuaikan dengan kebutuhan gameplay dan karakteristik pemain.

a. Tampilan Menu Utama



Gambar 3.2 Rancangan tampilan Menu Utama

Gambar 3.2 merupakan rancangan tampilan menu utama pada *game* “*The Lost Histories*”. Pada menu ini, *player* dapat memulai *game* dengan melihat berbagai opsi atau fitur yang disediakan oleh sistem.

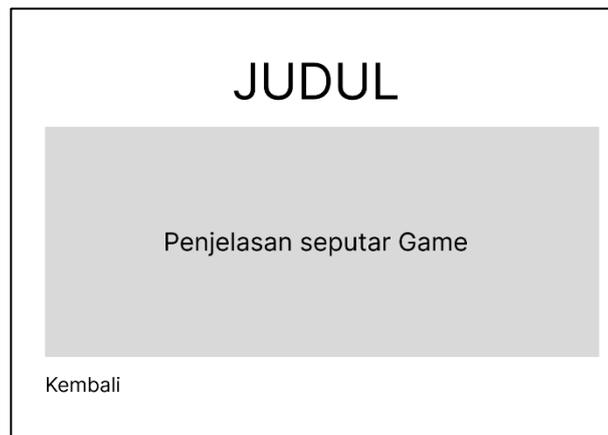
b. Tampilan Pemilihan Level



Gambar 3.3 Rancangan tampilan Pemilihan Level

Gambar 3.3 menunjukkan rancangan tampilan pemilihan level. *Player* dapat memilih level dari 1 hingga 10, masing-masing dengan tingkat kesulitan yang berbeda dengan keterangan yang ditampilkan pada bagian kiri.

c. Tampilan Tentang



Gambar 3.4 Rancangan tampilan Tentang Game

Gambar 3.4 menunjukkan rancangan *layout* tampilan tentang apa game *The Lost Histories* ini. Di dalam tampilan ini berisi deskripsi singkat tentang *game* dan metode pengembangan dibaliknya.

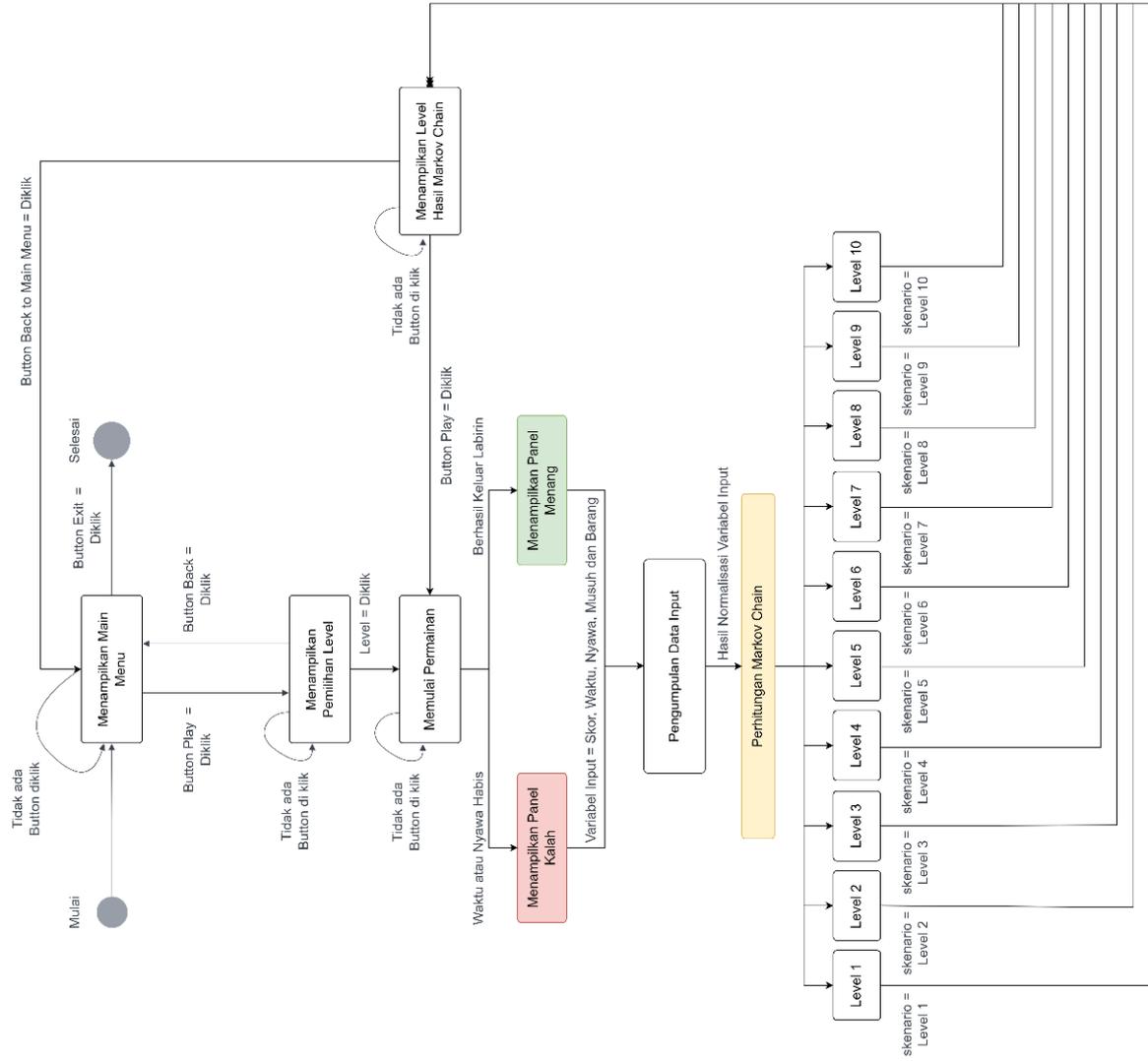
3.2 Finite State Machine (FSM)

Gambar 3.5 pada halaman berikutnya merupakan Finite State Machine (FSM) pada game "*The Lost Histories*" yang menggambarkan alur logika permainan dari awal hingga akhir. Permainan dimulai dengan menampilkan Main Menu atau Menu Utama di mana player dapat memilih untuk memulai permainan dengan mengklik tombol Mulai Perjuangan, kembali ke menu sebelumnya dengan tombol Kembali, atau mengakhiri permainan dengan tombol Akhiri Perjuangan. Setelah player memilih Mulai Perjuangan, mereka diarahkan ke menu Pemilihan Level, yang memungkinkan player memilih level dari Level 1 hingga Level 10, di mana setiap level memiliki tingkat kesulitan yang berbeda. Setelah level dipilih, permainan akan dimulai dan player harus menyelesaikan tantangan maze. Jika player berhasil keluar dari labirin, sistem akan menampilkan Panel Menang. Sementara jika waktu habis

atau nyawa player habis, maka Panel Kalah akan muncul. Kedua skenario ini berfungsi untuk mengumpulkan data input permainan yang akan digunakan dalam proses analisis adaptasi.

Setelah data dari permainan terkumpul baik player menang maupun kalah sistem akan menggunakan perhitungan Markov Chain untuk menganalisis performa dan menentukan tingkat kesulitan level berikutnya. Proses ini memungkinkan permainan untuk beradaptasi secara dinamis terhadap performa pemain. Misalnya, jika seorang pemain secara konsisten menyelesaikan tantangan dengan baik hingga mencapai Level 10, maka sistem akan mempertahankan atau tetap merekomendasikan level yang tinggi, karena probabilitas transisi antar level merefleksikan konsistensi performa tersebut. Namun, jika performa mulai menurun, pemain dapat diturunkan ke Level 7–9 untuk menyesuaikan tantangan. Sebaliknya, jika seorang pemain gagal berulang kali di Level 1, sistem akan tetap menempatkannya di level tersebut karena probabilitas naik sangat kecil, kecuali terjadi peningkatan performa.

Mekanisme ini mencerminkan karakteristik khas dari Markov Chain, di mana status level selanjutnya hanya bergantung pada status saat ini dan probabilitas transisinya. Dengan demikian, sistem adaptif ini tidak hanya menjaga keseimbangan tantangan, tetapi juga meningkatkan motivasi dan pengalaman bermain secara personal bagi setiap pemain.

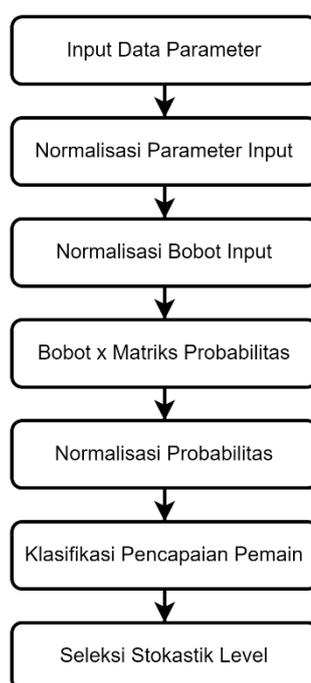


Gambar 3.5 Desain FSM

3.3 Rancangan Implementasi Metode Markov Chain

Markov Chain dirancang sebagai algoritma yang akan diimplementasikan ke dalam pengaturan level kesulitan agar metode pengaturan kesulitan level dapat bekerja lebih dinamis.

3.3.1 Desain Perhitungan Metode *Markov Chain*



Gambar 3.6 Desain Sistem Markov

Pendekatan ini menggunakan prinsip dasar *Markov Chain*, di mana transisi antar level dihitung menggunakan matriks probabilitas. Hanya saja, pembobotan parameter ditambahkan sebagai bentuk adaptasi untuk konteks game tanpa mengubah prinsip inti *Markov Chain*. Normalisasi parameter dan pembobotan input dilakukan untuk menyeimbangkan nilai agar perhitungan dengan matriks probabilitas menjadi lebih akurat dan adil. Proses kerja *Markov Chain* pada gambar 3.6 dapat dijelaskan setiap tahapnya sebagai berikut.

a. Input Data

Tahap pertama dalam proses perhitungan *Markov Chain* adalah menerima data parameter yang mencerminkan kondisi permainan pada setiap level. Parameter ini meliputi ukuran peta dalam bentuk *grid*, waktu yang tersedia, jumlah nyawa *player*, jumlah musuh yang harus dikalahkan, dan jumlah dokumen yang harus dikumpulkan. Tabel 3.1 berikut menunjukkan parameter lengkap untuk setiap level.

Tabel 3.1 Parameter Level

No.	Level	Ukuran	Waktu (detik)	Nyawa	Jumlah Musuh	Jumlah Dokumen	Total Skor
1	1	6x6	480	100	2	3	80
2	2	7x7	460	95	3	4	115
3	3	8x8	440	90	4	5	150
4	4	9x9	420	85	5	6	185
5	5	10x10	400	80	6	7	220
6	6	11x11	380	75	7	8	255
7	7	12x12	360	70	8	9	290
8	8	13x13	340	65	9	10	325
9	9	14x14	320	60	10	11	360
10	10	15x15	300	55	12	12	420

Keterangan :

- 1) Waktu : Waktu yang tersedia semakin berkurang dari 480 detik pada level pertama menjadi 300 detik pada level terakhir. Ini mencerminkan bahwa waktu yang diperlukan untuk menyelesaikan level akan berkurang seiring dengan meningkatnya kesulitan.
- 2) Nyawa *player* : Jumlah nyawa *player* akan berkurang setiap kali memulai level baru seiring meningkatnya level, dari 100 poin nyawa pada level 1 menjadi 55 poin nyawa ketika memulai level 10.

- 3) Jumlah musuh : Jumlah musuh menunjukkan bahwa semakin tinggi tingkat kesulitan, semakin banyak musuh yang harus dikalahkan oleh *player*. Setiap musuh yang dikalahkan akan bernilai 25 skor poin.
- 4) Jumlah dokumen : Sejumlah dokumen disini merupakan dokumen yang harus dikumpulkan oleh *player* jika ingin membuka dokumen sejarah. Dalam tabel 3.1 menunjukkan bahwa semakin tinggi tingkat level game, semakin banyak dokumen yang tersedia di level tersebut. Setiap dokumen yang diperoleh akan bernilai 10 skor poin.
- 5) Total Skor : Nilai total skor diperoleh melalui penjumlahan hasil jumlah musuh dikalahkan ditambah dengan jumlah dokumen yang diperoleh oleh *player* setiap menyelesaikan level.

b. Normalisasi Parameter Input

Pada tahap ini, setiap parameter input seperti skor, waktu, nyawa, musuh, dan dokumen dinormalisasi untuk memastikan bahwa semua parameter berada dalam skala yang sama. Normalisasi dilakukan dengan membagi nilai aktual parameter dengan nilai maksimum parameter tersebut:

$$inputNormalized_i = \frac{input_i}{maxInput_i} \quad (3.1)$$

Proses ini memastikan bahwa parameter dengan rentang yang lebih besar, seperti skor, tidak mendominasi hasil dibandingkan parameter lain seperti nyawa atau musuh. Normalisasi sangat penting untuk menciptakan keseimbangan dalam kontribusi setiap parameter terhadap probabilitas transisi.

c. Normalisasi Bobot Input

Setelah input dinormalisasi, langkah berikutnya adalah menghitung bobot kontribusi relatif setiap parameter. Bobot dihitung dengan membagi setiap nilai input yang telah dinormalisasi dengan total keseluruhan nilai input yang dinormalisasi:

$$inputWeight_i = \frac{inputNormalized_i}{\sum_{j=1}^n inputNormalized_j} \quad (3.2)$$

Di sini, n adalah jumlah parameter (misalnya, skor, waktu, nyawa, dll.).

Proses ini memastikan bahwa total bobot seluruh parameter adalah bernilai tidak lebih dari nilai 1.0 sehingga memungkinkan perhitungan probabilitas transisi yang proporsional. Sebagai contoh, jika skor memiliki nilai tertinggi setelah normalisasi, bobot skor akan lebih besar dibandingkan bobot parameter lainnya.

d. Bobot x Matriks Probabilitas Transisi

Tahap ini adalah inti dari perhitungan *Markov Chain*, di mana bobot input digunakan untuk menghitung probabilitas awal untuk setiap level melalui matriks probabilitas transisi. Matriks probabilitas diperoleh melalui penjumlahan variabel input berdasarkan data parameter level seperti pada tabel 3.2 berikut.

Tabel 3.2 Variabel Input

Input	State										Jumlah
	1	2	3	4	5	6	7	8	9	10	
Skor	80	115	150	185	220	255	290	325	360	420	2400
Waktu	480	460	440	420	400	380	360	340	320	300	3900
Nyawa	100	95	90	85	80	75	70	65	60	55	775
Musuh	2	3	4	5	6	7	8	9	10	12	66
Dokumen	3	4	5	6	7	8	9	10	11	12	75

Kemudian untuk menemukan matriks melalui variabel tersebut dilakukan pembagian setiap input pada setiap *state* dibagi dengan jumlah masing-masing input sebagai contoh Skor pada *State* 1 adalah 80 dibagi 2400 = 0,033 dan seterusnya sehingga akan menghasilkan nilai pada tabel 3.3 berikut.

Tabel 3.3 Hasil Pembagian Variabel

Input	State									
	1	2	3	4	5	6	7	8	9	10
Skor	0,033	0,048	0,063	0,077	0,092	0,106	0,121	0,135	0,150	0,175
Waktu	0,123	0,118	0,113	0,108	0,103	0,097	0,092	0,087	0,082	0,077
Nyawa	0,129	0,123	0,116	0,110	0,103	0,097	0,090	0,084	0,077	0,071
Musuh	0,030	0,045	0,061	0,076	0,091	0,106	0,121	0,136	0,152	0,182
Dokumen	0,040	0,053	0,067	0,080	0,093	0,107	0,120	0,133	0,147	0,160

Nilai-nilai tersebut diubah menjadi matriks probabilitas sebagai dasar perhitungan proses *Markov Chain*. Tabel tersebut akan berbentuk dalam matriks seperti berikut.

$$P = \begin{bmatrix} 0,033 & 0,048 & 0,063 & 0,077 & 0,092 & 0,106 & 0,121 & 0,135 & 0,150 & 0,175 \\ 0,123 & 0,118 & 0,113 & 0,108 & 0,103 & 0,097 & 0,092 & 0,087 & 0,082 & 0,077 \\ 0,129 & 0,123 & 0,116 & 0,110 & 0,103 & 0,097 & 0,090 & 0,084 & 0,077 & 0,071 \\ 0,030 & 0,045 & 0,061 & 0,076 & 0,091 & 0,106 & 0,121 & 0,136 & 0,152 & 0,182 \\ 0,040 & 0,053 & 0,067 & 0,080 & 0,093 & 0,107 & 0,120 & 0,133 & 0,147 & 0,160 \end{bmatrix} \quad (3.3)$$

Matriks ini menggambarkan hubungan antara setiap parameter input dan kemungkinan perpindahan ke level tertentu. Rumus pengalihan matriks probabilitas *Markov Chain* mengutip dari (Ross, 1966) adalah:

$$resultProbabilities = \sum_{j=1}^m inputWeight[j] \cdot P \quad (3.4)$$

Di sini, m adalah jumlah parameter, i adalah level yang dihitung, dan P adalah elemen matriks transisi yang menunjukkan probabilitas parameter j

memengaruhi level i . Hasilnya adalah vektor probabilitas awal untuk semua level.

e. Normalisasi Probabilitas

Probabilitas awal yang diperoleh dari matriks transisi kemudian dinormalisasi untuk memastikan total semua probabilitas adalah 1.0. Ini dilakukan dengan rumus:

$$normalizedProbabilities[i] = \frac{resultProbabilities[i]}{\sum_{i=0}^n resultProbabilities[j]} \quad (3.5)$$

Di sini, n adalah jumlah level. Normalisasi ini penting agar hasil probabilitas valid untuk seleksi stokastik.

f. Klasifikasi Pencapaian *Player*

Klasifikasi pencapaian pemain digunakan untuk menentukan tingkat keberhasilan dalam menyelesaikan suatu level. Klasifikasi ini menjadi dasar dalam menentukan rentang perubahan level selanjutnya melalui proses seleksi stokastik pada metode Markov Chain. Sistem mengevaluasi performa pemain berdasarkan lima parameter utama, yaitu skor, waktu tersisa, jumlah nyawa, jumlah musuh yang dikalahkan, dan jumlah item yang dikumpulkan. Setiap parameter terlebih dahulu dinormalisasi terhadap nilai maksimum masing-masing, kemudian diberikan bobot kontribusi sesuai tingkat kepentingannya.

Bobot ditentukan dengan mempertimbangkan seberapa besar parameter tersebut mencerminkan usaha aktif pemain dalam menjalankan misi. Parameter aktif seperti skor, jumlah musuh, dan item memiliki bobot lebih

besar dibandingkan waktu dan nyawa yang bersifat pasif. Bobot kontribusi setiap parameter adalah sebagai berikut:

- Skor : 30%
- Waktu tersisa : 10%
- Nyawa tersisa : 10%
- Musuh dikalahkan : 25%
- Item dikumpulkan : 25%

Total performa dihitung sebagai penjumlahan nilai normalisasi masing-masing parameter dikalikan dengan bobotnya. Nilai total performa tersebut diklasifikasikan menjadi tiga kategori:

1. Rendah jika total performa < 0.3
2. Sedang jika $0.3 \leq \text{performa} < 0.6$
3. Tinggi jika $\text{performa} \geq 0.6$

Kategori pencapaian ini digunakan baik ketika pemain berhasil (misi sukses) maupun gagal. Pada misi sukses, kategori akan menentukan rentang kenaikan level, sedangkan pada misi gagal, kategori digunakan untuk menentukan rentang penurunan level secara adaptif. Dengan demikian, sistem dapat memberikan penyesuaian tingkat kesulitan secara lebih adil dan proporsional, mendorong pemain untuk aktif menyelesaikan misi dengan semestinya, bukan hanya mencapai titik akhir permainan.

g. Seleksi Stokastik Level

Seleksi level berikutnya dilakukan berdasarkan probabilitas yang telah dinormalisasi. Sebuah nilai acak (*randomValue*) dihasilkan dalam rentang

[0,1], kemudian probabilitas kumulatif dihitung untuk menentukan level yang akan dipilih. Rumus probabilitas kumulatif adalah berikut.

$$cumulativeProbability[i] = \sum_{j=i}^i normalizedProbabilities[j] \quad (3.6)$$

Jika $randomValue \leq cumulativeProbability[i]$, maka level i dipilih.

Misalnya, jika probabilitas normalisasi adalah [0.2, 0.3, 0.5], maka probabilitas kumulatif adalah [0.2, 0.5, 1.0]. Jika nilai acak adalah 0.35, maka level 2 akan dipilih karena $0.35 \leq 0.5$.

3.3.2 Simulasi Pengujian Sistem Markov Chain

Pengujian sistem Markov Chain dilakukan untuk memvalidasi bahwa proses transisi antar level berdasarkan performa player bekerja sesuai dengan logika probabilistik yang dirancang. Pengujian dilakukan melalui simulasi perhitungan menyeluruh terhadap input yang diberikan pemain setelah menyelesaikan suatu level.

a. Input Data dan Normalisasi

Simulasi dilakukan berdasarkan performa pemain setelah menyelesaikan suatu level dalam pengujian ini adalah Level 4. Input yang diberikan berupa lima parameter utama, masing-masing dibandingkan dengan nilai maksimumnya, lalu dinormalisasi.

Tabel 3.4 Pengujian Normalisasi Input Data

Parameter	Nilai Player	Nilai Maksimum	Normalisasi
Skor	185	420	0.440
Waktu	420	480	0.875
Nyawa	85	100	0.850
Musuh	5	12	0.417
Dokumen	6	12	0.500

Total Normalisasi	3.082
-------------------	-------

b. Perhitungan Bobot Tiap Parameter

Bobot tiap parameter dihitung dari hasil normalisasi dibagi totalnya:

Tabel 3.5 Pengujian Normalisasi Bobot Input Data

Parameter	Nilai Player	Total Normalisasi	Bobot
Skor	0.440	3.082	0.143
Waktu	0.875	3.082	0.284
Nyawa	0.850	3.082	0.276
Musuh	0.417	3.082	0.135
Dokumen	0.500	3.082	0.162
Total Bobot			1.0

c. Bobot x Matriks Probabilitas

Perhitungan dilakukan untuk tiap level (L1 sampai L10):

$$\Pr(L1) = (0.143 \times 0.033) + (0.284 \times 0.123) + (0.276 \times 0.129) + (0.135 \times 0.030) + (0.162 \times 0.040) = 0.0858$$

$$\Pr(L2) = (0.143 \times 0.048) + (0.284 \times 0.118) + (0.276 \times 0.123) + (0.135 \times 0.045) + (0.162 \times 0.053) = 0.0889$$

$$\Pr(L3) = (0.143 \times 0.063) + (0.284 \times 0.113) + (0.276 \times 0.116) + (0.135 \times 0.061) + (0.162 \times 0.067) = 0.0922$$

$$\Pr(L4) = (0.143 \times 0.077) + (0.284 \times 0.108) + (0.276 \times 0.110) + (0.135 \times 0.076) + (0.162 \times 0.080) = 0.0951$$

$$\Pr(L5) = (0.143 \times 0.092) + (0.284 \times 0.103) + (0.276 \times 0.103) + (0.135 \times 0.091) + (0.162 \times 0.093) = 0.0969$$

$$\Pr(L6) = (0.143 \times 0.106) + (0.284 \times 0.097) + (0.276 \times 0.097) + (0.135 \times 0.106) + (0.162 \times 0.107) = 0.0981$$

$$\Pr(L7) = (0.143 \times 0.121) + (0.284 \times 0.092) + (0.276 \times 0.090) + (0.135 \times 0.121) + (0.162 \times 0.120) = 0.0990$$

$$\Pr(L8) = (0.143 \times 0.135) + (0.284 \times 0.087) + (0.276 \times 0.084) + (0.135 \times 0.136) + (0.162 \times 0.133) = 0.1002$$

$$\Pr(L9) = (0.143 \times 0.150) + (0.284 \times 0.082) + (0.276 \times 0.077) + (0.135 \times 0.152) + (0.162 \times 0.147) = 0.1006$$

$$\Pr(L10) = (0.143 \times 0.175) + (0.284 \times 0.077) + (0.276 \times 0.071) + (0.135 \times 0.182) + (0.162 \times 0.160) = 0.1021$$

d. Normalisasi Probabilitas

Hasil perkalian Bobot x Matriks ditotal sebelum dinormalisasi. Maka semua nilai dapat dinormalisasi dengan membaginya terhadap total tersebut agar menjadi distribusi sah bernilai = 1.0.

e. Klasifikasi Pencapaian Pemain

Nilai total performa dihitung dengan menjumlahkan semua parameter yang telah dinormalisasi \times bobot.

- Performa < 0.3 \rightarrow Rendah (turun level)
- $0.3 \leq$ Performa $< 0.6 \rightarrow$ Sedang (naik 1–2 level)
- Performa ≥ 0.6 \rightarrow Tinggi (naik 2–4 level)

f. Seleksi Level Secara Stokastik

Sistem mengambil nilai randomValue = 0.275. Dari distribusi kumulatif:

- Cumulative L1 = 0.089
- Cumulative L2 = 0.183
- Cumulative L3 = 0.278

Karena $0.275 < 0.278$ maka Level 3 dipilih sebagai level selanjutnya.

Kesimpulan pengujian sistem Markov Chain ini membuktikan bahwa sistem Markov Chain berjalan sesuai desain perancangan melalui logika adaptif dan probabilistik dimana parameter dikonversi ke bobot proporsional kemudian probabilitas dihitung dan dinormalisasi dengan tepat. Seleksi level berikutnya dilakukan dengan sistem stokastik berbasis performa sehingga sistem tetap adil, adaptif, dan konsisten dengan rancangan algoritmik.

3.4 Desain Pengujian Sistem

Untuk menilai kualitas pengalaman pengguna di dalam memainkan *game The Lost Histories* yang menerapkan metode Markov Chain ini, peneliti melakukan pengujian sistem antara lain:

1. *System Usability Scale* (SUS): Pengujian ini bertujuan untuk mengevaluasi kemudahan penggunaan dan kepuasan pengguna terhadap sistem yang dikembangkan. SUS memberikan wawasan penting terkait seberapa mudah *player* berinteraksi dengan antarmuka game, serta seberapa nyaman dan efisien pengguna dalam menyelesaikan tugas-tugas di dalam game. Hasil dari pengujian SUS akan digunakan sebagai dasar untuk iterasi dan peningkatan lebih lanjut pada aspek *keusability* dari game. (Brooke, 1986)
2. *Game Experience Questionnaire* (GEQ): Pengujian GEQ dilakukan dengan membiarkan *player* mencoba game dalam situasi yang menyerupai kondisi nyata, sambil diamati oleh pengembang untuk menilai interaksi mereka (IJsselsteijn et al., 2013). Melalui pengujian ini, *feedback* yang diperoleh berupa pengisian kuisioner oleh para partisipan yang telah mencoba bermain *game* secara langsung terkait bagaimana *player* merasakan kesulitan, kepuasan, atau tantangan yang ada dalam game.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil implementasi game *The Lost Histories* yang telah dikembangkan sesuai dengan desain pada bab sebelumnya. Uji coba dilakukan untuk memastikan fungsi berjalan dengan baik, baik dari sisi teknis permainan, logika Markov Chain, maupun interaksi pengguna. Pembahasan mencakup pengujian validasi algoritma pembangkitan *maze*, serta evaluasi pengalaman pengguna melalui kuesioner *System Usability Scale* (SUS) dan *Game Experience Questionnaire* (GEQ). Hasil ini dianalisis untuk mengukur keberhasilan sistem dalam mencapai tujuan edukatif dan hiburan secara simultan.

4. 1. Implementasi Antarmuka *Game*

Pada tahap ini dilakukan implementasi dari perancangan antarmuka yang telah dibuat sebelumnya. Antarmuka pengguna (*User Interface*) dikembangkan menggunakan aset visual dan elemen interaktif yang sesuai dengan konsep dan mekanisme permainan *The Lost Histories*. Implementasi ini bertujuan untuk memastikan bahwa semua fitur yang dirancang dapat berfungsi secara optimal dan memberikan pengalaman pengguna yang intuitif serta mendukung alur permainan. Berikut merupakan hasil implementasi dari masing-masing bagian antarmuka game.

a. Implementasi Tampilan (UI) Menu Utama



Gambar 4.1 Tampilan (UI) Menu

Gambar 4.1 merupakan tampilan *interface* menu utama pada game *The Lost Histories*. Pada menu ini, *player* dapat melakukan memulai *game* dengan menekan tombol Mulai Perjuangan atau Akhiri Perjuangan jika *player* ingin keluar dari permainan.

b. Implementasi Tampilan Pemilihan Level



Gambar 4.2 Tampilan (UI) Pemilihan Level

Gambar 4.2 menunjukkan tampilan pemilihan level. *Player* dapat memilih level dari 1 hingga 10, masing-masing dengan tingkat kesulitan yang berbeda. Baik *player* menang atau kalah setelahnya pada level saat ini, untuk selanjutnya *player* akan memainkan *level*

yang telah dihitung melalui sistem *Markov Chain* berdasarkan pencapaian *player* pada level saat ini.

c. Implementasi Tampilan Seputar *Game*



Gambar 4.3 Tampilan (UI) Seputar *Game*

Gambar 4.3 menunjukkan tampilan Tentang *Game* atau tentang apa *game* “*The Lost Histories*” ini. Di dalam tampilan ini berisi deskripsi singkat tentang *game* dan metode pengembangan dibaliknya.

4. 2. Implementasi Sistem

Penelitian ini menghasilkan sebuah *game* 3D berjudul “*The Lost Histories*” yang memanfaatkan metode *Markov Chain* dalam pembangkitan *maze* berbasis pencapaian *player*. Implementasi dilakukan melalui aplikasi Unity dengan bahasa pemrograman C#.

4.2.1. Implementasi Sistem *Maze Generator*

Peta dasar dalam *game* adalah bertipe *maze* sesuai judul penelitian. Ukuran *maze* serta tingkat kompleksitasnya ditentukan berdasarkan parameter setiap level yang tersaji pada Tabel 3.1. Setiap level memiliki tingkat kesulitan yang berbeda, yang memengaruhi karakteristik *maze* yang

dihasilkan. Proses pembangkitan maze secara acak mengikuti *Pseudocode* 4.1, di mana algoritma dalam sistem akan menghasilkan struktur labirin yang bervariasi setiap kali level dimainkan ulang.

Pseudocode 4.1 Maze Generator

```

START
SET timeRemaining ← levelTime
START coroutine UpdateTimer

CALL GenerateCustomMaze(rows, columns)

IF playerHealth exists THEN
    SET playerHealth.CurrentHp ← MIN(initialHp, playerHealth.maxHp)
END IF

FUNCTION GenerateCustomMaze(rows, columns):
    IF rows is EVEN THEN INCREMENT rows
    IF columns is EVEN THEN INCREMENT columns

    INITIALIZE maze[rows][columns] with value 1
    CLEAR floorPositions
    DESTROY all child objects

    CALL InitializeMaze(rows, columns)
    CALL CarveMaze(startX = 1, startY = 1, rows, columns)
    CALL DrawMaze(rows, columns)
    CALL SetSpawnPoints
    CALL SpawnMazeCenter(rows, columns)

FUNCTION CarveMaze(x, y, rows, columns):
    SHUFFLE directions [UP, RIGHT, DOWN, LEFT]
    SET maze[x][y] ← 0
    ADD position (x, y) to floorPositions

    FOR EACH direction IN directions:
        COMPUTE next cell (nx, ny)
        IF cell (nx, ny) is within bounds AND maze[nx][ny] = 1 THEN
            SET passage between (x, y) and (nx, ny) ← 0
            ADD passage position to floorPositions
            RECURSIVELY CALL CarveMaze(nx, ny, rows, columns)
        END IF
    END FOR

FUNCTION DrawMaze(rows, columns):
    DESTROY all child objects in transform
    FOR EACH cell in maze:
        IF wall THEN
            INSTANTIATE wallPrefab at cell
        ELSE
            INSTANTIATE floorPrefab at cell
            INSTANTIATE roofPrefab above floor with 180° rotation
        END IF
    END FOR

FUNCTION SetSpawnPoints:
    SET playerSpawn ← floorPositions[0]
    SET finishSpawn ← last floorPosition + Y offset
    INSTANTIATE player and finish at respective positions

    CREATE availablePositions from floorPositions EXCLUDING near player
    and finish
    IF availablePositions < total object count THEN RETURN

    CALL SpawnObjects(itemPrefab, itemCount, availablePositions)
    CALL SpawnObjects(healthPrefab, healthCount, availablePositions)

```

```

CALL SpawnObjects(ammoPrefab, ammoCount, availablePositions)
CALL SpawnObjects(enemyPrefab, enemyCount, availablePositions)

FUNCTION SpawnObjects(prefab, count, positions):
  FOR count TIMES:
    SELECT random position from positions
    INSTANTIATE prefab at selected position with hover offset
    IF prefab is item/health/ammo THEN ADD RotatingItem component
    REMOVE used position from list
  END FOR
END

```

Kelas MazeGenerator bertanggung jawab untuk membangkitkan map maze secara prosedural serta menempatkan elemen gameplay di dalamnya. Proses diawali dengan inisialisasi waktu permainan (levelTime) dan memulai coroutine UpdateTimer yang berfungsi sebagai hitungan mundur hingga waktu habis. Setelah itu, pemanggilan GenerateCustomMaze() digunakan untuk membangun struktur maze berdasarkan parameter baris dan kolom.

Di dalam GenerateCustomMaze, ukuran maze divalidasi agar selalu ganjil, lalu diinisialisasi sebagai grid penuh tembok. Fungsi CarveMaze menggunakan algoritma recursive backtracking untuk membuat jalur secara acak dari titik awal. Jalur yang dibuat disimpan ke dalam floorPositions sebagai acuan tempat meletakkan objek di tahap selanjutnya.

Setelah jalur *maze* selesai, fungsi DrawMaze akan menggambar ulang seluruh grid berdasarkan data maze, di mana sel bernilai 1 diisi oleh wallPrefab, dan sisanya diisi oleh floorPrefab dan roofPrefab. Posisi pemain dan titik akhir ditentukan pada awal dan akhir daftar posisi lantai yang tersedia.

Langkah berikutnya adalah SetSpawnPoints, yang akan menghitung semua posisi yang layak digunakan untuk menempatkan objek seperti item, health, ammo, dan musuh. Fungsi SpawnObjects digunakan untuk

menyebarkan objek tersebut secara acak namun terkendali, sehingga tidak saling tumpang tindih dan tetap mempertahankan pengalaman gameplay yang adil dan seimbang.

4.2.2. Implementasi Sistem Markov Chain

Sistem pemilihan level adalah dengan mengimplementasikan metode Markov Chain setiap kali level diselesaikan oleh *player*. Cara perhitungannya berdasarkan pada desain perhitungan Markov Chain yang terlihat pada gambar 3.6 yang memiliki 7 tahap seperti pada *Pseudocode* 4.2 berikut.

Pseudocode 4.2 Markov Chain

```

START
RECEIVE input: SCORE, TIME, LIVES, ENEMIES, ITEMS, CURRENT LEVEL, and
SUCCESS STATUS

NORMALIZE each parameter using its MAXIMUM VALUE:
- SCORE divided by 420
- TIME divided by 480
- LIVES divided by 100
- ENEMIES divided by 12
- ITEMS divided by 12

CALCULATE TOTAL of all NORMALIZED VALUES

IF TOTAL equals ZERO THEN
    RETURN CURRENT LEVEL minus TWO, with a MINIMUM of ONE
END
END IF

COMPUTE WEIGHT of each parameter:
- WEIGHT = NORMALIZED VALUE divided by TOTAL

FOR each LEVEL INDEX from 1 to 10 DO
    MULTIPLY each WEIGHT by corresponding TRANSITION MATRIX VALUE
    SUM results to obtain PROBABILITY for each LEVEL
END FOR

NORMALIZE the PROBABILITY VECTOR so the SUM equals 1.0

CLASSIFY PLAYER ACHIEVEMENT as LOW, MEDIUM, or HIGH using WEIGHTED SCORE
of:
- SCORE, ENEMIES, ITEMS, TIME, and LIVES

DEFINE MINIMUM and MAXIMUM LEVEL CHANGE RANGE:
- IF LOW THEN SET RANGE = 2 to 3
- IF MEDIUM THEN SET RANGE = 1 to 2
- IF HIGH THEN
    IF SUCCESS THEN SET RANGE = 1 to 4
    ELSE SET RANGE = 1
    END IF
END IF

GENERATE RANDOM VALUE between 0 and 1

INITIALIZE CUMULATIVE PROBABILITY as 0

```

```

FOR each LEVEL INDEX from 1 to 10 DO
  ADD PROBABILITY to CUMULATIVE PROBABILITY
  IF RANDOM VALUE ≤ CUMULATIVE PROBABILITY THEN
    IF SUCCESS THEN
      RETURN CURRENT LEVEL increased WITHIN RANGE
    ELSE
      RETURN CURRENT LEVEL decreased WITHIN RANGE
    END IF
  END IF
END FOR

IF NO MATCH FOUND in PROBABILITY SELECTION THEN
  RETURN FALLBACK LEVEL:
  - INCREASE CURRENT LEVEL by 1 if SUCCESS
  - DECREASE CURRENT LEVEL by 1 if FAILURE
  - ENSURE LEVEL stays between 1 and 10

END

```

Proses diawali dengan penerimaan input, yaitu nilai-nilai yang mewakili performa pemain seperti Score, Time, Lives, Enemies, dan Items, serta status keberhasilan misi (SUCCESS STATUS) dan CURRENT LEVEL pemain saat ini. Nilai-nilai ini digunakan sebagai dasar dalam menentukan level berikutnya.

Tahapan pertama adalah NORMALIZE, di mana kelima parameter yang diterima dinormalisasi terhadap nilai maksimum masing-masing agar berada dalam skala yang sebanding. Misalnya, SCORE dibagi dengan 420, TIME dengan 480, dan seterusnya. Hasil dari proses ini akan menghasilkan lima normalized values yang merepresentasikan pencapaian relatif pemain pada tiap aspek.

Selanjutnya dilakukan TOTAL CALCULATION, yaitu penjumlahan dari semua nilai yang telah dinormalisasi. Total ini digunakan untuk memastikan apakah pemain benar-benar melakukan aktivitas dalam game. Jika totalnya adalah nol, maka sistem menganggap pemain tidak melakukan interaksi signifikan dan memberikan penalti berupa penurunan dua level melalui logika RETURN CURRENT LEVEL minus TWO.

Jika pemain memiliki total nilai yang valid, maka proses berlanjut ke tahap WEIGHT COMPUTATION, yaitu menghitung kontribusi relatif atau bobot dari setiap parameter terhadap total. Bobot ini mencerminkan proporsi pengaruh setiap aspek performa terhadap keseluruhan pencapaian.

Bobot-bobot tersebut kemudian digunakan dalam proses PROBABILITY COMPUTATION, yaitu menghitung peluang masing-masing level hasil dengan cara mengalikan bobot dengan nilai pada TRANSITION MATRIX. Matrix ini berisi probabilitas transisi antar level berdasarkan lima parameter performa. Setelah dijumlahkan untuk tiap kolom level, dihasilkan vektor peluang atau PROBABILITY VECTOR.

Agar akumulasi probabilitas tetap konsisten, dilakukan tahap NORMALIZATION OF PROBABILITY VECTOR, yaitu menyesuaikan nilai dalam vektor agar totalnya tepat 1.0. Ini penting untuk memastikan validitas distribusi peluang.

Berdasarkan nilai performa yang ada, pemain kemudian diklasifikasikan ke dalam kategori pencapaian melalui tahap ACHIEVEMENT CLASSIFICATION, yaitu LOW, MEDIUM, atau HIGH. Penentuan ini dilakukan dengan memberikan bobot khusus pada parameter yang dianggap paling mempengaruhi performa, misalnya Score, Enemies, dan Items.

Berdasarkan hasil klasifikasi, sistem menentukan LEVEL CHANGE RANGE. Misalnya, kategori LOW diberi kenaikan 2–3 level, MEDIUM 1–2 level, dan HIGH bisa sampai 4 level jika misi berhasil. Jika gagal, sistem

hanya memperbolehkan perubahan level ke bawah dalam rentang yang sesuai.

Tahap penting selanjutnya adalah STOCHASTIC SELECTION, yaitu pemilihan probabilistik level berdasarkan hasil PROBABILITY VECTOR. Sistem menggenerate nilai acak antara 0 dan 1, lalu membandingkannya dengan akumulasi probabilitas untuk menentukan level hasil. Jika level berhasil dipilih dari proses stokastik ini, sistem akan RETURN NEXT LEVEL yang telah disesuaikan, baik naik atau turun tergantung keberhasilan. Namun jika tidak ada probabilitas yang memenuhi, maka sistem masuk ke FALLBACK MECHANISM, yaitu secara default menaikkan level satu jika misi berhasil, atau menurunkannya satu jika gagal, dengan memastikan level tetap dalam batas 1 hingga 10.

4.2.3. Pengujian Markov Chain dengan Matlab

Untuk memastikan sistem Markov Chain yang diterapkan di dalam game “*The Lost Histories*” telah berfungsi sesuai dengan aturan yang telah dijabarkan, maka peneliti melakukan perbandingan Markov Chain dengan Matlab untuk melakukan validasi (Vrugt, 2016). Melalui 5 variabel input dari hasil percobaan setidaknya hasil level berikutnya harus sesuai dengan perhitungan Matlab. Berikut beberapa data sampel hasil perbandingan sistem Markov Chain yang telah didokumentasikan pada Tabel 4.1.

Tabel 4.1 Perbandingan Output Markov Chain dengan Matlab

No	Level Awal	Input					Output (Level)			Hasil Uji Coba
		Skor	Waktu	Nyawa	Musuh	Item	Performa	Game	Matlab	
1	3	420	300	100	12	12	Tinggi	6	6	Sesuai
2	4	400	280	90	11	11	Tinggi	7	7	Sesuai
3	5	390	260	85	10	10	Tinggi	8	8	Sesuai
4	6	385	270	95	11	12	Tinggi	9	9	Sesuai
5	7	410	290	98	12	12	Tinggi	10	10	Sesuai
6	3	250	240	75	6	7	Sedang	5	5	Sesuai
7	5	270	260	80	7	6	Sedang	7	7	Sesuai
8	6	220	230	70	5	5	Sedang	8	8	Sesuai
9	4	200	220	68	4	6	Sedang	6	6	Sesuai
10	2	260	250	76	6	8	Sedang	4	4	Sesuai
11	4	80	120	40	2	1	Rendah	2	2	Sesuai
12	6	90	100	45	3	2	Rendah	5	5	Sesuai
13	5	100	150	55	2	1	Rendah	4	4	Sesuai
14	7	110	160	50	3	2	Rendah	6	6	Sesuai
15	3	85	110	48	2	2	Rendah	2	2	Sesuai

Berdasarkan hasil perbandingan pada tabel 4.3, dapat dipastikan bahwa sistem Markov Chain telah berjalan dengan valid. Hasil pengujian mandiri konsisten dengan perhitungan Matlab, menunjukkan bahwa transisi level sesuai dengan logika probabilistik dan klasifikasi performa yang dirancang.

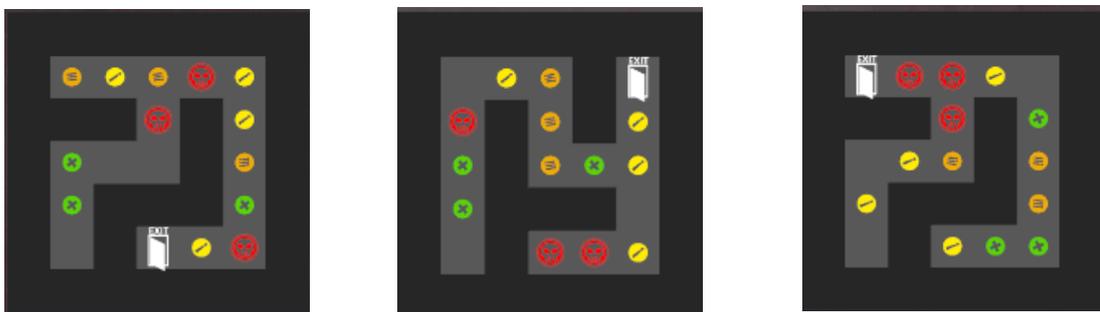
4.3. Uji Coba Game

Uji coba game dilakukan untuk memastikan bahwa implementasi *Markov Chain* berjalan sesuai desain dan menghasilkan transisi level yang dinamis berdasarkan performa *player*. Selain itu, uji coba memastikan bahwa labirin atau

maze yang dihasilkan selalu berbeda pada setiap permainan. Sistem ini menggunakan algoritma pembangkitan acak (*random*) untuk menghasilkan labirin yang berbeda di setiap level. Dengan cara ini, permainan terhindar dari pola yang monoton dan dapat menyesuaikan tantangan berdasarkan performa *player*.

1. Percobaan Pertama

Pada uji coba pertama ini penguji memulai permainan dari Level rendah yaitu Level 2 dengan parameter ukuran grid 7x7, waktu yang diberikan 450 detik (7 menit 30 detik), nyawa awal permainan adalah 95, jumlah musuh adalah 3 dan jumlah dokumen yang harus dikumpulkan adalah 4 dokumen. Pada gambar 4.4 adalah contoh simulasi 3 bentuk *maze* Level 2 yang akan selalu berbeda ketika memulai permainan.



Gambar 4.4 Beberapa Bentuk Maze Generator Level 2

Pada uji coba ini *player* mencapai pintu keluar yang bersimbol pintu *Exit* pada *map* labirin diatas dan mengumpulkan sejumlah dokumen. Sehingga tampilan yang muncul adalah “Status : SUKSES” dengan performa “Sedang” dan rekap perolehan input *player* di sebelah kiri dan hasil perhitungan *Markov Chain* menampilkan level selanjutnya di tombol sebelah kanan seperti pada gambar 4.5.



Gambar 4.5 Tampilan Misi Sukses

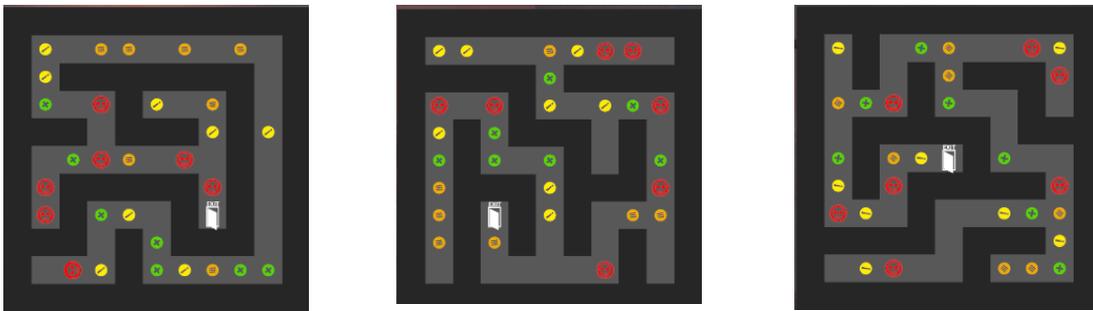
Pada gambar diatas hasil permainan diatas, pada panel bagian kanan terdapat tampilan tombol Misi Level 4 yang dihasilkan melalui proses perhitungan sistem *Markov Chain*. Selain itu, menunjukkan bahwa *player* juga berhasil menemukan seluruh dokumen yang tersebar pada labirin sehingga akan muncul tombol “DOKUMEN SEJARAH”. Jika tombol tersebut diklik maka akan tampil sebuah sejarah seperti tampak pada Gambar 4.6.



Gambar 4.6 Tampilan Dokumen Sejarah

2. Percobaan Kedua

Pada uji coba selanjutnya *player* akan memulai permainan dari Level yang lebih tinggi yaitu Level 7 dengan parameter ukuran grid 12x12, waktu yang diberikan 360 detik (6 menit 10 detik), nyawa awal permainan adalah 70, jumlah musuh adalah 8 dan jumlah dokumen yang harus dikumpulkan adalah 9 dokumen. Pada Gambar 4.7 adalah contoh simulasi 3 bentuk *maze* Level 7 yang akan selalu berbeda ketika memulai permainan di Level tersebut.



Gambar 4.7 Beberapa Bentuk Maze Generator Level 7

Pada uji coba ini *player* tidak berhasil mencapai pintu keluar karena nyawa *player* habis sebab dikalahkan oleh musuh. Sehingga tetap akan menampilkan hasil input yang sama dengan tampilan “Status : GAGAL” dengan performa “Rendah” sehingga *player* akan turun ke level dibawah dari Level 7. Pada gambar 4.8 adalah tampilan misi gagal dan karena level kali ini adalah 5 kemudian hasil perhitungan sistem adalah untuk level selanjutnya diarahkan ke level dibawahnya yaitu Level 4.



Gambar 4.8 Tampilan Misi Gagal

Pada gambar diatas, selain Status misi yang Gagal, status Dokumen bertuliskan “Tidak Ditemukan” karena tidak dapat mengumpulkan seluruh dokumen yang tersebar didalam labirin tersebut. Sehingga tidak muncul tombol “DOKUMEN SEJARAH” seperti ketika *player* berhasil mengumpulkan seluruh dokumen yang tersebar didalam labirin pada gambar 4.5.

4. 4. Pengujian Sistem

Setelah diimplementasikan dan dilakukan uji coba mandiri, tahap selanjutnya adalah pengujian sistem yang dilakukan dengan cara diujikan kepada responden untuk memainkan *game* yang telah dikembangkan dengan tujuan untuk memastikan bahwa *game The Lost Histories* telah memenuhi kriteria *Usability* dan memberikan *Experience* tersendiri bagi *player*. Pada kedua survey ini, *player* diminta untuk menilai fitur di atas dalam skala lima poin, yang menunjukkan tingkat persetujuan yang berbeda-beda. Berikut adalah justifikasi masing – masing nilai pada skala.

1. Nilai 1 mewakili jawaban sangat tidak setuju

2. Nilai 2 mewakili jawaban tidak setuju
3. Nilai 3 mewakili jawaban netral / ragu-ragu
4. Nilai 4 mewakili jawaban setuju
5. Nilai 5 mewakili jawaban sangat setuju

Setelah meminta responden untuk memainkan game *The Lost Histories*, responden diminta mengisi kuesioner untuk menilai *System Usability Scale* dan *Game Experience Questionnaire*. *System Usability Scale* (SUS) dan *Game Experience Questionnaire* (GEQ) digunakan sebagai metode evaluasi kuantitatif untuk menilai kualitas sistem dari dua sudut pandang berbeda namun saling melengkapi.

4.4.1. System Usability Scale (SUS)

System Usability Scale (SUS) merupakan sebuah sistem yang terdiri dari 10 pertanyaan yang menawarkan evaluasi subjektif terhadap kegunaan. Pertanyaan-pertanyaan ini dipilih untuk menghindari bias dan memastikan para responden memberikan perhatian yang seksama pada setiap pernyataan. SUS memiliki validitas yang sangat baik untuk mengukur kegunaan sistem karena mencakup berbagai topik yang berhubungan dengan kegunaan, termasuk dukungan, pelatihan, dan kompleksitas (Brooke, 1986). Terlihat pada tabel 4.2 terdapat 10 pertanyaan yang akan diujikan kepada responden.

Tabel 4.2 Versi standar pertanyaan dari SUS

No.	Pertanyaan
<i>Q₁</i>	<i>I think that I would like to use this system frequently</i>
<i>Q₂</i>	<i>I found the system unnecessary complex</i>
<i>Q₃</i>	<i>I thought the system was easy to use</i>
<i>Q₄</i>	<i>I think that I would need the support of a technical person to able to use this system</i>
<i>Q₅</i>	<i>I found the various functions in this system were well integrated</i>

No.	Pertanyaan
<i>Q₆</i>	<i>I thought there was too much inconsistency in this system</i>
<i>Q₇</i>	<i>I would imagine that most people would learn to use this system very quickly</i>
<i>Q₈</i>	<i>I found the system very cumbersome to use</i>
<i>Q₉</i>	<i>I felt very confident using the system</i>
<i>Q₁₀</i>	<i>I needed to learn a lot of things before I could get going with this system</i>

Menurut (Brooke, 1986), melakukan pengujian SUS setidaknya dibutuhkan data dari responden yang diuji sebanyak 20 orang. Pada penelitian ini, dilakukan pengujian pada 25 responden yang duduk di bangku SMP atau sederajat karena permainan ini mengandung edukasi tentang sejarah Indonesia yang umum diajarkan untuk siswa SMP atau sederajat. Adapun demografi responden bisa dilihat pada tabel 4.3 berikut.

Tabel 4.3 Demografi Responden

No	Variabel	Kategori	Frekuensi	Total
1	Jenis Kelamin	Laki-laki	21	25
		Perempuan	4	
2	Umur	13 tahun	9	25
		14 tahun	12	
		15 tahun	4	
3	Estimasi Bermain Game	1-2 jam	6	25
		2-4 jam	9	
		Lebih dari 4 jam	10	
4	Pernah Memainkan Game Edukasi	Ya	10	25
		Tidak	15	
5	Perangkat Untuk Memainkan Game	Smartphone (HP)	16	25
		Laptop/PC	5	
		Tablet	3	

Sebelum mengisi kuisioner, responden terlebih dahulu diminta untuk memainkan game "*The Lost Histories*" minimal sebanyak 3 level. Pemain bebas memilih level mana yang ingin dimainkan, namun diutamakan untuk

mencoba kombinasi dari level yang mudah hingga menengah atau sulit, guna mendapatkan pengalaman yang lebih menyeluruh terhadap fitur adaptif, antarmuka, serta gameplay. Hal ini bertujuan agar *feedback* yang diberikan melalui kuisisioner benar-benar merefleksikan pengalaman bermain yang cukup representatif. Kemudian responden diminta untuk mengisi kuisisioner SUS dan didapatkan hasil pada tabel 4.5 berikut.

Tabel 4.4 Hasil Kuisisioner SUS

Responde n	Pertanyaan									
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
R_1	5	2	4	3	4	3	4	2	5	2
R_2	4	2	4	3	4	4	4	1	5	2
R_3	4	2	5	3	4	2	5	2	5	1
R_4	3	2	5	3	4	2	5	2	5	1
R_5	4	3	4	3	5	3	5	2	4	1
R_6	5	2	5	3	5	3	5	2	3	2
R_7	4	2	5	2	4	3	4	2	5	2
R_8	3	2	5	2	4	2	4	2	5	1
R_9	4	4	5	2	4	2	4	1	4	2
R_{10}	4	2	5	3	5	2	5	1	4	3
R_{11}	4	2	4	3	4	2	4	3	3	3
R_{12}	4	2	4	4	5	3	4	1	3	3
R_{13}	4	3	4	3	5	2	5	1	5	1
R_{14}	4	2	4	3	4	3	3	2	3	3
R_{15}	5	3	4	3	4	3	4	2	4	3
R_{16}	5	2	4	2	4	3	5	2	4	3
R_{17}	5	4	5	3	5	4	5	3	5	3
R_{18}	4	2	5	2	4	2	4	2	5	3
R_{19}	3	2	4	2	4	1	5	2	4	1
R_{20}	4	2	4	2	4	1	5	2	3	2
R_{21}	4	2	3	4	2	2	3	3	4	3
R_{22}	4	2	4	3	3	2	4	4	5	3
R_{23}	5	1	3	2	2	3	4	4	4	2
R_{24}	4	1	5	3	3	1	3	3	4	3
R_{25}	5	1	5	2	4	3	2	3	2	2

Tabel diatas merupakan jawaban yang dikumpulkan dari responden.

Kriteria perhitungan skor SUS harus diikuti dalam pengolahan data yang

dikumpulkan dari kuisisioner yang telah diisi. Ada beberapa aturan untuk menentukan penilaian kuisisioner, berikut.

1. Skor jawaban responden akan dikurangi 1 untuk setiap pertanyaan ganjil.
2. Skor jawaban responden akan dikurangi 5 untuk setiap pertanyaan genap.
3. Setelah jawaban seluruh soal dijumlahkan, skor akhir SUS akan ditentukan dengan mengalikan hasilnya dengan 2,5.

Aturan diatas berlaku pada setiap 1 responden. Kemudian, skor SUS didapatkan dari mencari rata-rata skor dari masing-masing responden dengan menjumlahkan semua skor dan dibagi dengan jumlah responden. Adapun rumus 4.1 merupakan rumus menghitung skor SUS.

$$\bar{x} = \frac{\sum x}{n} \quad (4.1)$$

Dimana \bar{x} = Skor rata-rata, $\sum x$ = Jumlah skor SUS, n = Jumlah responden.

Sehingga, skor rata-rata SUS didapatkan dari jumlah dari keseluruhan responden kemudian dikali dengan 2,5. Selanjutnya akan ditentukan skor keseluruhan masing-masing responden dan ditampilkan pada tabel 4.6.

Tabel 4.5 Hasil Total Kuisisioner SUS

Responden	Pertanyaan										Total
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	
R_1	4	3	3	2	3	2	3	3	4	3	30
R_2	3	3	3	2	3	1	3	4	4	3	29
R_3	3	3	4	2	3	3	4	3	4	4	33
R_4	2	3	4	2	3	3	4	3	4	4	32
R_5	3	2	3	2	4	2	4	3	3	4	30
R_6	4	3	4	2	4	2	4	3	2	3	31
R_7	3	3	4	3	3	2	3	3	4	3	31

<i>R₈</i>	2	3	4	3	3	3	3	3	4	4	32
<i>R₉</i>	3	1	4	3	3	3	3	4	3	3	30
<i>R₁₀</i>	3	3	4	2	4	3	4	4	3	2	32
<i>R₁₁</i>	3	3	3	2	3	3	3	2	2	2	26
<i>R₁₂</i>	3	3	3	1	4	2	3	4	2	2	27
<i>R₁₃</i>	3	2	3	2	4	3	4	4	4	4	33
<i>R₁₄</i>	3	3	3	2	3	2	2	3	2	2	25
<i>R₁₅</i>	4	2	3	2	3	2	3	3	3	2	27
<i>R₁₆</i>	4	3	3	3	3	2	4	3	3	2	30
<i>R₁₇</i>	4	1	4	2	4	1	4	2	4	2	28
<i>R₁₈</i>	3	3	4	3	3	3	3	3	4	2	31
<i>R₁₉</i>	2	3	3	3	3	4	4	3	3	4	32
<i>R₂₀</i>	3	3	3	3	3	4	4	3	2	3	31
<i>R₂₁</i>	3	3	2	1	1	3	2	2	3	2	22
<i>R₂₂</i>	3	3	3	2	2	3	3	1	4	2	26
<i>R₂₃</i>	4	4	2	3	1	2	3	1	3	3	26
<i>R₂₄</i>	3	4	4	2	2	4	2	2	3	2	28
<i>R₂₅</i>	4	4	4	3	3	2	2	2	1	3	28

Setelah dilakukan perhitungan total nilai untuk tiap responden, akan dilakukan perhitungan skor SUS dan Skor rata-rata SUS seperti pada ketentuan sebelumnya.

Tabel 4.6 Hasil akhir skor SUS

Responden	Skor SUS (Total x 2.5)
<i>R₁</i>	75
<i>R₂</i>	72,5
<i>R₃</i>	82,5
<i>R₄</i>	80
<i>R₅</i>	75
<i>R₆</i>	77,5
<i>R₇</i>	77,5
<i>R₈</i>	80
<i>R₉</i>	75
<i>R₁₀</i>	80
<i>R₁₁</i>	65
<i>R₁₂</i>	67,5
<i>R₁₃</i>	82,5
<i>R₁₄</i>	62,5
<i>R₁₅</i>	67,5
<i>R₁₆</i>	75

<i>R₁₇</i>	70
<i>R₁₈</i>	77,5
<i>R₁₉</i>	80
<i>R₂₀</i>	77,5
<i>R₂₁</i>	55
<i>R₂₂</i>	65
<i>R₂₃</i>	65
<i>R₂₄</i>	70
<i>R₂₅</i>	70
Total Skor SUS	1825
Skor Rata-rata SUS	73,6

Tabel 4.6 diatas merupakan hasil akhir dari perhitungan skor SUS. Selanjutnya akan dibandingkan skor rata-rata SUS dengan penilaian SUS. Termasuk kategori mana dari hasil pengujian dengan skor rata-rata yang sudah didapatkan.

Berdasarkan hasil kuisisioner SUS yang telah diberikan kepada responden, diperoleh skor rata-rata sebesar 73,6. Menurut interpretasi standar yang dikemukakan oleh (Bangor et al., 2009), nilai SUS dengan skor di atas 70 dikategorikan dalam tingkat penerimaan yang baik (*Acceptable*) dan memperoleh predikat *Good* dalam skala penilaian persepsi kegunaan sistem. Dengan demikian, skor 73,6 menunjukkan bahwa sistem berada pada tingkat *usability* yang memuaskan dan dapat diterima oleh pengguna sejauh antarmuka atau fungsionalitasnya.

4.4.2. Game Experience Questionnaire (GEQ)

Pengujian *Game Experience Questionnaire* (GEQ) digunakan untuk mengevaluasi pengalaman *player* saat bermain game *The Lost Histories*. Metode ini terdiri dari tiga modul utama, yaitu

1. *Core Module* mengevaluasi elemen seperti kompetensi, imersi, aliran (*flow*), tantangan (*challenge*), ketegangan (*tension*), dan dampak negatif.
2. *Social Presence Module* mengukur keterlibatan sosial *player*, termasuk empati, perasaan negatif, dan keterlibatan perilaku. Modul ini digunakan jika game memiliki interaksi sosial atau *multiplayer*, menilai sejauh mana pemain merasakan kehadiran sosial pemain lain. Maka karena game yang dikembangkan adalah bertipe *single player*, modul ini diabaikan dan tidak relevan.
3. *Post-Game Module* mengevaluasi pengalaman *player* setelah bermain, seperti pengalaman positif, pengalaman negatif, rasa lelah (*tiredness*), dan kemampuan *player* untuk kembali fokus pada realitas. Instrumen ini berupa kuesioner yang digunakan untuk menilai unsur-unsur *experience* yang dirasakan oleh pengguna pada saat dan sesudah bermain, serta menjadi gambaran umum terhadap *user experience* dari sebuah game yang diteliti (Cedergren, 2018). Rumus yang digunakan untuk menghitung kusioner GEQ adalah sebagai berikut :

Rumus TCR (Tingkat Capaian Responden)

$$TCR = \frac{\text{Average Score Answer}}{5} \times 100\% \quad (4.2)$$

Keterangan :

81– 100% = Sangat baik

61 – 80% = Baik

41 – 60% = Kurang baik

21 – 40% = Tidak baik

< 20% = Sangat tidak baik

TCR dihitung untuk masing-masing komponen dari modul. Hasil TCR diinterpretasikan untuk mengevaluasi kinerja dan pengalaman *player* pada game. Berikut adalah tabel yang berisi pertanyaan-pertanyaan dari *Game Experience Questionnaire* (GEQ) sesuai dengan modul masing-masing, dilengkapi dengan kolom untuk komponen masing-masing pertanyaan.

Tabel 4.7 Pertanyaan Core Module

No	Pertanyaan	Komponen
1	Saya merasa puas dengan game ini.	<i>Competence</i>
2	Saya merasa terampil menggunakan game ini.	<i>Competence</i>
3	Saya merasa tertarik dengan cerita game ini.	<i>Immersion</i>
4	Saya merasa game ini menyenangkan.	<i>Immersion</i>
5	Game ini memberikan suasana hati yang buruk.	<i>Negative Affect</i>
6	Saya pikir game ini sulit dimainkan.	<i>Challenge</i>
7	Saya lupa dengan apa yang ada di sekitar saya.	<i>Flow</i>
8	Saya merasa bosan.	<i>Negative Affect</i>
9	Saya merasa berhasil memainkan game ini.	<i>Competence</i>
10	Saya merasa imajinatif.	<i>Immersion</i>
11	Saya merasa bahwa saya menjelajahi berbagai hal.	<i>Immersion</i>
12	Saya menikmati game ini.	<i>Competence</i>
13	Saya merasa kesal saat memainkan game ini.	<i>Tension</i>
14	Saya merasa tertantang saat bermain game ini.	<i>Challenge</i>
15	Saya merasa game ini mengesankan.	<i>Immersion</i>
16	Saya merasa sangat fokus saat bermain game ini.	<i>Flow</i>
17	Saya merasa frustrasi.	<i>Tension</i>
18	Saya merasa bahwa ini merupakan pengalaman yang berharga.	<i>Immersion</i>
19	Saya merasa kehilangan koneksi dengan luar.	<i>Flow</i>
20	Saya berusaha keras saat memainkan game ini.	<i>Challenge</i>

Tabel 4.8 Pertanyaan Post-Game Module

No	Pertanyaan	Komponen
1	Saya merasa bersemangat kembali.	<i>Positive Experience</i>
2	Saya merasa tidak nyaman.	<i>Negative Experience</i>
3	Saya merasa bingung.	<i>Negative Experience</i>
4	Saya merasa mudah memahami.	<i>Positive Experience</i>
5	Sulit dalam bermain game ini.	<i>Negative Experience</i>

6	Game ini membuat saya tidak puas.	<i>Negative Experience</i>
7	Saya menyesal memainkan game ini.	<i>Negative Experience</i>
8	Saya ingin memainkan game ini kembali.	<i>Positive Experience</i>
9	Saya merasa tertarik.	<i>Positive Experience</i>
10	Membuat saya cepat bosan.	<i>Tiredness</i>

Mirip halnya dengan pengujian SUS, responden yang telah memainkan *game* diminta untuk mengisi 30 pertanyaan tersebut dengan 5 skala *likert* yang telah dijelaskan sebelumnya. Hanya saja dalam perhitungan GEQ, setiap pertanyaan akan dihitung dengan rumus TCR berdasarkan masing-masing komponen pada setiap modulnya. Pada tabel 4.9 di bawah ini adalah hasil perhitungan TCR dari 20 pernyataan pada modul *Core Module* berdasarkan setiap komponennya.

Tabel 4.9 Hasil Komponen Core Module

Komponen	Nilai TCR	Hasil
<i>Competence</i>	83,8%	Sangat Baik
<i>Immersion</i>	79,6%	Baik
<i>Flow</i>	59,5%	Kurang Baik
<i>Tension</i>	38,4%	Tidak Baik
<i>Challenge</i>	61,3%	Baik
<i>Negative Effect</i>	39,2%	Tidak Baik

Pada tabel diatas terlihat hasil tingkat capaian responden pada komponen *Core Module* yang menunjukkan Sangat Baik pada komponen *Competence* dengan 83,8% dan kategori baik pada komponen *Immersion* dan *Challenge* serta hasil kurang bahkan tidak baik di komponen lainnya.

Tabel 4.10 Hasil Komponen Post Game Module

Komponen	Nilai TCR	Hasil
<i>Positive Experience</i>	81,2%	Sangat Baik
<i>Negative Experience</i>	43,4%	Kurang Baik
<i>Tiredness</i>	40,4%	Tidak Baik
<i>Return to Reality</i>	80%	Baik

Hasil Post Game Module dalam GEQ Tabel 4.11 menunjukkan pengalaman positif yang sangat baik, dan tingkat efek negatif yang rendah. Ini menandakan bahwa *player* merasakan pengalaman menyenangkan setelah memainkan game. Selain itu, nilai *Tiredness* yang bernilai rendah yaitu 40,4% menandakan bahwa tidak cukup melelahkan bagi *player* yang telah memainkan *game* sehingga relevan dengan komponen terakhir yang bernilai Baik berarti tidak membuat *player* begitu kecanduan untuk memainkan *game The Lost Histories*.

Pengujian SUS menilai aspek kemudahan penggunaan, efisiensi, dan kenyamanan sistem secara umum. Sedangkan pengujian GEQ, khususnya *Core Module*, menilai emosi, keterlibatan, dan tantangan saat bermain. *Post-Game Module* GEQ menilai perasaan pemain setelah bermain, seperti kepuasan dan kelelahan. Meskipun berbeda dalam fokus, beberapa kuisisioner di SUS dan GEQ memiliki tujuan yang serupa, sehingga saling menguatkan validitas satu sama lain. sebagaimana bisa disajikan pada tabel 4.11 berikut.

Tabel 4.11 Korelasi Kuisisioner SUS dan GEQ

SUS	GEQ	Komponen	Keterkaitan
Q4: <i>I think that I would need the support of a technical person to use this system</i>	Q1: Saya merasa puas dengan game ini Q9: Saya merasa berhasil memainkan game ini	<i>Competence</i>	Ketergantungan pada bantuan teknis menurunkan rasa percaya diri dan kepuasan bermain
Q5: <i>I found the various functions in this system were well integrated</i>	Q14: Saya merasa tertantang saat bermain game ini	<i>Challenge</i>	Keduanya mengevaluasi apakah sistem (game) terasa sinkron dan menantang secara fungsi

Q6: <i>I thought there was too much inconsistency in this system</i>	Q13: Saya merasa kesal saat memainkan game ini	<i>Tension</i>	Inkonistensi sistem dapat menyebabkan frustrasi atau ketegangan pada pengguna
Q10: <i>I needed to learn a lot of things before I could get going with this system</i>	Q3: Saya merasa tertarik dengan cerita game ini Q4: Saya merasa game ini menyenangkan	<i>Immersion</i>	Jika pembelajaran awal terlalu berat (SUS), akan menghambat immersiveness (GEQ)

Hasil pengujian menggunakan *System Usability Scale* (SUS) dan *Game Experience Questionnaire* (GEQ) menunjukkan hubungan yang searah. Tabel 4.11 memperlihatkan bahwa ketika nilai SUS tinggi—dalam hal ini 73,6 dengan kategori GOOD komponen dalam GEQ seperti Competence dan Challenge juga menunjukkan hasil yang positif. Artinya, kemudahan penggunaan antarmuka yang dirasakan pemain berdampak langsung pada persepsi mereka terhadap kemampuan menguasai permainan dan merasakan tantangan yang sesuai.

Komponen *Competence* dalam GEQ mencerminkan kepercayaan diri pemain terhadap kemampuan menyelesaikan misi, sementara *Challenge* menunjukkan sejauh mana tantangan dalam game terasa menantang namun tidak berlebihan. Nilai tinggi pada kedua komponen tersebut memperkuat bahwa sistem adaptif berbasis Markov Chain telah berhasil menjaga keseimbangan antara usability dan pengalaman bermain yang mendalam.

4. 5. Integrasi Islam

4.5.1. Muamalah Ma'Allah

Konsep hubungan manusia dengan Allah dalam game ini diwujudkan melalui nilai-nilai ketauhidan yang mengajarkan *player* untuk selalu mengingat kebesaran Allah dalam setiap usaha dan perjuangan mereka. Dalam game *The Lost Histories*, *player* menghadapi tantangan dalam menemukan jejak sejarah bangsa yang hilang, mencerminkan perjalanan hidup manusia yang penuh ujian, tetapi selalu dalam pengawasan dan takdir Allah.

قُلْ هُوَ اللَّهُ أَحَدٌ ۝ ١ اللَّهُ الصَّمَدُ ۝ ٢ لَمْ يَلِدْ وَلَمْ يُولَدْ ۝ ٣ وَلَمْ يَكُنْ لَهُ كُفُوًا أَحَدٌ ۝ ٤

"Katakanlah (Muhammad), Dialah Allah, Yang Maha Esa. Allah tempat bergantung segala sesuatu. Dia tidak beranak dan tidak diperanakkan. Dan tidak ada sesuatu yang setara dengan-Nya." (QS. Al-Ikhlâs: 1-4)

Ayat ini menegaskan konsep tauhid yang menjadi dasar utama dalam kehidupan seorang Muslim. Dalam konteks game "*The Lost Histories*", *player* diajak untuk memahami dan mengingat sejarah perjuangan bangsa sebagai bagian dari jihad intelektual, yang sejatinya adalah bentuk penghambaan kepada Allah. Mengenang perjuangan masa lalu menjadi bentuk refleksi untuk mengambil pelajaran, sekaligus pengakuan bahwa Allah adalah satu-satunya sumber kekuatan dalam setiap perjuangan umat manusia (Suryana, 2024).

4.5.2. Muamalah Ma'annas

Dalam dunia *The Lost Histories*, *player* tidak hanya berjuang secara fisik, tapi juga dituntut menjaga nilai keadilan, kejujuran, dan tanggung jawab sosial. Moral *player* diuji dalam situasi seperti membela yang tertindas, meskipun tidak menguntungkan dirinya, atau menegakkan keadilan meskipun bertentangan dengan kelompoknya sendiri.

يَا أَيُّهَا الَّذِينَ آمَنُوا كُونُوا قَوَّامِينَ بِالْقِسْطِ شُهَدَاءَ لِلَّهِ وَلَوْ عَلَىٰ أَنفُسِكُمْ أَوِ الْوَالِدِينَ وَالْأَقْرَبِينَ

"Wahai orang-orang yang beriman! Jadilah kamu penegak keadilan, menjadi saksi karena Allah, meskipun terhadap dirimu sendiri atau ibu bapak dan kaum kerabatmu." (QS. An-Nisa: 135)

Menurut Ibnu Katsir, ayat ini merupakan seruan langsung kepada orang-orang beriman agar menjadi penegak keadilan, yakni seseorang yang terus-menerus membela keadilan dalam segala kondisi, dan bukan hanya ketika menguntungkan diri sendiri. Teguran dalam ayat ini sangat tegas bahwa keadilan tidak boleh dikalahkan oleh ikatan keluarga, kekayaan, atau kedudukan. Bahkan jika keadilan itu harus ditegakkan melawan diri sendiri, maka itu adalah bagian dari iman yang sejati. Penegakan keadilan juga menjadi tanggung jawab sosial untuk menjaga keseimbangan masyarakat dan melindungi hak-hak orang lain. (Deni Kamaludin, 2015)

4.5.3. Muamalah Ma'alam

Islam mengajarkan manusia sebagai khalifah di bumi yang bertanggung jawab untuk menjaga keseimbangan dan tidak berbuat kerusakan. Dalam

game ini, konsep ini tercermin dalam bagaimana *player* harus bertanggung jawab terhadap lingkungan sekitarnya dalam memahami sejarah.

فَكُلًّا أَخَذْنَا بِذَنبِهِ فَمِنْهُمْ مَّنْ أَرْسَلْنَا عَلَيْهِ حَاصِبًا وَمِنْهُمْ مَّنْ أَخَذْنَا الصَّيْحَةَ وَمِنْهُمْ مَّنْ خَسَفْنَا بِهِ الْأَرْضَ وَمِنْهُمْ مَّنْ أَغْرَقْنَا وَمَا كَانَ اللَّهُ لِيظْلِمَهُمْ وَلَكِنْ كَانُوا أَنْفُسَهُمْ يَظْلِمُونَ

"Maka masing-masing (mereka itu) Kami siksa disebabkan dosanya. Di antara mereka ada yang Kami timpakan kepadanya hujan batu kerikil, dan di antara mereka ada yang disambar suara keras yang mengguntur, dan di antara mereka ada yang Kami benamkan ke dalam bumi, dan di antara mereka ada yang Kami tenggelamkan. Dan Allah tidaklah berbuat zalim kepada mereka, tetapi merekalah yang menzalimi diri mereka sendiri." (QS. Al-'Ankabut: 40)

Ayat ini menjelaskan bagaimana umat-umat terdahulu dihancurkan karena kezaliman dan kerusakan yang mereka lakukan. Dalam konteks game, *player* dapat melihat bagaimana peperangan dan konflik di masa lalu menghancurkan peradaban dan budaya, serta bagaimana pentingnya menjaga nilai-nilai luhur agar sejarah tidak terulang dengan cara yang sama. Alur petualangan dalam game ini mengajarkan bahwa memahami sejarah bukan sekadar mengenang, tetapi juga sebagai peringatan agar kesalahan masa lalu tidak terulang kembali (Riyandi, 2020).

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini berhasil mengembangkan sistem pembangkit peta labirin atau *map maze* adaptif pada game 3D *The Lost Histories* menggunakan metode Markov Chain. Sistem ini mampu menyesuaikan tingkat kesulitan berdasarkan performa pemain, dengan mempertimbangkan variabel seperti skor, waktu penyelesaian, jumlah nyawa, musuh, dan dokumen yang diperoleh. Hasil implementasi menunjukkan bahwa sistem dapat menghasilkan labirin yang bervariasi dan dinamis, serta memberikan transisi level yang adil dan logis sesuai prinsip probabilistik. Pengujian logika Markov memperlihatkan bahwa transisi antar state berlangsung sesuai harapan, dengan distribusi probabilitas yang mencerminkan capaian pemain secara real-time.

Dari sisi evaluasi *usability*, metode *System Usability Scale* (SUS) digunakan untuk mengukur kemudahan penggunaan sistem. Hasil dari 25 responden menunjukkan nilai rata-rata 73,6, yang menurut skala interpretasi dari (Bangor et al., 2009) termasuk dalam kategori "Good". Skor ini menandakan bahwa sistem memiliki kegunaan yang baik, mudah dipahami, serta mendukung pengalaman interaksi yang efisien dan tidak membingungkan bagi pemain.

Selain itu, hasil pengujian menggunakan *Game Experience Questionnaire* (GEQ) mendukung temuan dari SUS, terutama pada aspek core module seperti *immersion* dan *competence*, serta post-game module yang menunjukkan adanya

positive affect dari pemain. Kesesuaian hasil antara SUS dan GEQ memperkuat bahwa sistem tidak hanya unggul secara teknis dalam penerapan logika adaptif berbasis Markov Chain, tetapi juga mampu memberikan pengalaman bermain yang menyenangkan, menantang, dan melibatkan secara emosional. Hal ini membuktikan bahwa sistem yang dikembangkan layak digunakan dan memiliki potensi pengembangan lebih lanjut.

5.2. Saran

Berdasarkan temuan penelitian dan hasil pengujian, terdapat beberapa hal yang dapat ditingkatkan untuk menyempurnakan pengalaman bermain dalam game *The Lost Histories*, antara lain:

1. Pengembangan visual dapat ditingkatkan melalui penambahan efek grafis, *cutscene* sejarah interaktif, serta sistem animasi karakter yang lebih dinamis..
2. Untuk meningkatkan tantangan dan variasi permainan, bentuk labirin dapat dibuat lebih kompleks dan tematik, misalnya labirin berbentuk simbol sejarah atau daerah ikonik Indonesia. Obstacle juga bisa divariasikan, seperti jebakan atau zona beracun. Penambahan jenis rintangan akan membuat setiap level terasa unik dan tidak monoton.
3. Game dapat ditingkatkan dengan menambahkan fitur pemilihan karakter dengan kemampuan atau latar belakang yang berbeda misalnya karakter rakyat, prajurit dan semisalnya. Selain memberikan variasi *gameplay*, fitur ini juga memperkuat unsur narasi dan imersi.

Dengan melakukan perbaikan pada aspek-aspek tersebut, diharapkan *The Lost Histories* dapat menghadirkan pengalaman bermain yang lebih mendalam, menantang, serta sarat akan nilai edukasi sejarah.

DAFTAR PUSTAKA

- Abdurakhman ; Setiawan, A. (2013). *Atlas Sejarah Indonesia Berita Proklamasi Kemerdekaan* (Vol. 01, Issue 01).
- Arif, Y. M., Nurhayati, H., Karami, A. F., Nugroho, F., Kurniawan, F., Rasyid, H. A., Aini, Q., Diah, N. M., & Garcia, M. B. (2023). An Artificial Neural Network-Based Finite State Machine for Adaptive Scenario Selection in Serious Game. *International Journal of Intelligent Engineering and Systems*, *16*(5), 488–500. <https://doi.org/10.22266/ijies2023.1031.42>
- Azizah, A., Welastika, R., Falah, A. N., Ruchjana, B. N., & Abdullah, A. S. (2019). An Application of Markov Chain for Predicting Rainfall Data at West Java using Data Mining Approach. *IOP Conference Series: Earth and Environmental Science*, *303*(1). <https://doi.org/10.1088/1755-1315/303/1/012026>
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean; adding an adjective rating. *Journal of Usability Studies*, *4*(3), 114–123.
- Brooke, J. (1986). SUS: A “Quick and Dirty” Usability Scale. *Usability Evaluation In Industry*, November 1995, 207–212. <https://doi.org/10.1201/9781498710411-35>
- Cedergren, J. E. (2018). *Evaluating the User Experience and Usability of Virtual Reality Locomotion Techniques An Empirical Comparison*.
- Chen, V. H. H., Koek, W. J. D., Ho, S. T. J., & Yu, V. (2022). *Balancing Fun and Seriousness Serious: Game Design Considerations*. July. <https://doi.org/10.1037/tms0000026>
- Ching, W. K., Fung, E. S., & Ng, M. K. (2004). Higher-order Markov chain models for categorical data sequences. *Naval Research Logistics*, *51*(4), 557–574. <https://doi.org/10.1002/nav.20017>
- Colwell, A. M., & Glavin, F. G. (2017). Colwell’s castle defence: A custom game using dynamic difficulty adjustment to increase player enjoyment. *CEUR Workshop Proceedings*, *2086*, 275–282.
- Deni Kamaludin, Y. (2015). *NORMA-NORMA HUKUM HAK ASASI MANUSIA DALAM AL-QURAN:STUDI KRITIS ATAS TAFSIR AL-QURAN SURAT AL-NISÁ’AYAT 135*. 6.
- Dirgantara, H. B., Marselino, T. L., & Kurniawati, Y. E. (2023). *Kajian Literatur Kurikulum E-sport dan Perkembangan Industri Game*. *10*(1).

- Haario, H., Saksman, E., & Tamminen, J. (2005). Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, 20(2), 265–273. <https://doi.org/10.1007/BF02789703>
- Habibullah, A. I. (2024). *Procedural Maze Modeling in 3D For Game Asset. 020048*.
- Haryanto, H. (2016). Reward Dinamis dalam Skenario Adaptif Menggunakan Metode Finite State Machine pada Game Edukasi Dynamic Reward in Adaptive Scenario Using Finite State Machine for Education Game. *Journal of Applied Intelligent System*, 1(2), 144–153.
- Hendrix, M., Bellamy-Wood, T., McKay, S., Bloom, V., & Dunwell, I. (2019). Implementing adaptive game difficulty balancing in serious games. *IEEE Transactions on Games*, 11(4), 320–327. <https://doi.org/10.1109/TG.2018.2791019>
- Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. *ACM International Conference Proceeding Series*, 265(August), 429–433. <https://doi.org/10.1145/1178477.1178573>
- IJsselsteijn, W. A., de Kort, Y. A. W., & Poels, K. (2013). *The Game Experience Questionnaire. 2013*.
- Irwandi, P., Erlansari, A., & Effendi, R. (2016). Perancangan Game First Person Shooter (Fps) “Boar Hunter” Berbasis Virtual Reality. *Rekusif*, 14(1), 68–79.
- Jalu Kinayun, S., Santi Wahyuni, F., & Xaverius Ariwibisono, F. (2024). Rancang Bangun Game Road Maze 3D. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(5), 2966–2927. <https://doi.org/10.36040/jati.v7i5.7563>
- Katadata, D. (2024). *Indonesia Jadi Kontributor Game Terbanyak di Platform Steam Se-Asia Tenggara. Databooks Katadata*. <https://databoks.katadata.co.id/datapublish/2024/01/11/indonesia-jadi-kontributor-game-terbanyak-di-platform-steam-se-asia-tenggara>
- Krisdiawan, R. A., Fitriani, A., & Budianto, H. (2022). Penerapan Algoritma Recursive Backtracking Sebagai Maze Generator Pada Game Labirin Aksara Sunda. *Media Jurnal Informatika*, 14(1), 31. <https://doi.org/10.35194/mji.v14i1.2326>
- Nugroho, F., Basid, P. M. N. S. A., Bahtiar, F. S., & Buditjahjanto, I. G. P. A. (2023). Dynamic Difficulty Adjustment of Serious-Game Based on Synthetic Fog using Activity Theory Model. *International Journal of Advanced Computer Science and Applications*, 14(6), 564–573. <https://doi.org/10.14569/IJACSA.2023.0140660>

- Peachey, L. (2022). Parameterized Maze Generation Algorithm for Specific Difficulty Maze Generation. In *Proceedings of CS488 Senior Capstone* (Vol. 1, Issue 1). Association for Computing Machinery.
- Rahmawati, I., & Leksono, I. P. (2020). Pengembangan Game Petualang untuk Pembelajaran Berhitung. *Jurnal Kajian Teknologi Pendidikan*, 5(1), 11–23.
- Riyandi, Y. (2020). Varietas Azab Di Dunia Dalam Al-Qur'an (Penafsiran Tematik QS. Al-Ankabut: 40). *Al-Dzikra: Jurnal Studi Ilmu al-Qur'an Dan al-Hadits*, 14(1), 79–98. <https://doi.org/10.24042/al-dzikra.v14i1.6314>
- Ross, S. M. (1966). Stochastic Processes Second Edition. In *University of California, Berkeley, United States of America*. https://doi.org/10.1142/9789813148963_0008
- Setiadharna, E., Husniah, L., Kholimi, A. S., & Backtracking, R. (2020). *Algoritma Maze Generator Recursive Backtracking Untuk Membuat Prosedural Labirin Pada Game Petualangan Labirin 3D*. 2(3), 373–384.
- Shah, S. H., Mohite, J. M., Musale, A. G., & Borade, J. L. (2017). Survey Paper on Maze Generation Algorithms for Puzzle Solving Games. *International Journal of Scientific & Engineering Research*, 8(2), 1064–1067.
- Shihab Quraish. (2002). Tafsir Al-Misbah Jilid 11. *Analytical Biochemistry*, 11(1), 561.
- Soler-Dominguez, J. L., Camba, J. D., Contero, M., & Alcañiz, M. (2017). A proposal for the selection of eye-tracking metrics for the implementation of adaptive gameplay in virtual reality based games. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10280, 369–380. https://doi.org/10.1007/978-3-319-57987-0_30
- Sudrajat, M. A. (2022). Media Karya Mahasiswa Komunikasi dan Desain JOURNAL OF DIGITAL COMMUNICATION AND DESIGN (JDCODE) PERANCANGAN VIDEO GAME ACTION RPG FANTASY AIOUS UNTUK PERANGKAT KOMPUTER. *Journal of Digital Communication and Design (JDCODE)*, 22–29.
- Suryana, E. (2024). NILAI-NILAI PENDIDIKAN TAUHID DALAM AL-QUR'AN SURAT AL-IKHLAS AYAT 1 SAMPAI 4 MENURUT TAFSIR IBNU KATSIR. 4(2), 83–91. <https://e-journal.uin-al-azhaar.ac.id/index.php/idaroh/article/view/700/498>
- Susanto, E. K., Fachruddin, R., Diputra, M. I., Herumurti, D., & Yunanto, A. (2020). *Maze Generation Based on Difficulty using Genetic Algorithm with Gene Pool*. 554–559.

- Thompson, W. A., Taylor, H. M., & Karlin, S. (1985). An Introduction to Stochastic Modeling. In *Journal of the American Statistical Association* (Vol. 80, Issue 390). <https://doi.org/10.2307/2287941>
- Vrugt, J. A. (2016). Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Environmental Modelling and Software*, 75, 273–316. <https://doi.org/10.1016/j.envsoft.2015.08.013>
- Yuwono, A. I. (2021). *EKSISTENSI DEVELOPER GAME INDEPENDEN INDONESIA (Studi Kasus Eksistensi Developer Game Independen Agate Studio , Creacle Studio , dan Digital Happiness Dalam Perspektif Ekonomi Politik Komunikasi)*. 2, 22–39.
- Zamith, M., da Silva, J. R., Joselli, M., & Clua, E. W. G. (2020). Applying hidden Markov model for dynamic game balancing. *Brazilian Symposium on Games and Digital Entertainment, SBGAMES, 2020-Novem*, 38–46. <https://doi.org/10.1109/SBGames51465.2020.00016>
- Zuin, G. L., & Macedo, Y. P. A. (2016). Attempting to discover infinite combos in fighting games using hidden markov models. *Brazilian Symposium on Games and Digital Entertainment, SBGAMES, 0*, 80–88. <https://doi.org/10.1109/SBGames.2015.15>