

**IMPLEMENTASI SISTEM STATE PLAYER
MENGUNAKAN FINITE STATE MACHINE PADA GAME
THE GREAT RIFT**

SKRIPSI

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana
Program Studi Teknologi Informasi



disusun oleh

LEON SION TURNIP

22.60.0151

Kepada

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

**IMPLEMENTASI SISTEM STATE PLAYER MENGGUNAKAN FINITE
STATE MACHINE PADA GAME THE GREAT RIFT**

SKRIPSI

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana
Program Studi Teknologi Informasi



disusun oleh

LEON SION TURNIP

22.60.0151

Kepada

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

HALAMAN PERSETUJUAN

HALAMAN PERSETUJUAN

SKRIPSI

IMPLEMENTASI SISTEM STATE PLAYER MENGGUNAKAN FINITE STATE MACHINE PADA GAME THE GREAT RIFT

yang disusun dan diajukan oleh

Leon Sion Turnip

22.60.0151

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 23 Februari 2026

Dosen Pembimbing,



Buyut Khoirul Umri, M.Kom
NIK. 190302652

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

SKRIPSI

IMPLEMENTASI SISTEM STATE PLAYER MENGGUNAKAN FINITE STATE MACHINE PADA GAME THE GREAT RIFT

yang disusun dan diajukan oleh

Leon Sion Turnip

22.60.151

Telah dipertahankan di depan Dewan Penguji
pada tanggal 23 Februari 2026

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Ibnu Hadi Purwanto, S.Kom., M.Kom.
NIK. 190302390

Nadea Cipta Laksmi, M.Kom.
NIK. 190302551

Buyut Khoirul Umri, M.Kom.
NIK. 190302652

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal 23 Februari 2026

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan di bawah ini,

Nama mahasiswa : Leon Sion Turnip
NIM : 22.60.0151

Menyatakan bahwa Skripsi dengan judul berikut:

IMPLEMENTASI SISTEM STATE PLAYER MENGGUNAKAN FINITE
STATE MACHINE PADA GAME THE GREAT RIFT

Dosen Pembimbing : Buyut Khoirul Umri, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 23 Februari 2026

Yang Menyatakan,

Leon Sion Turnip

KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan penyusunan skripsi yang berjudul “IMPLEMENTASI SISTEM STATE PLAYER MENGGUNAKAN FINITE STATE MACHINE PADA GAME THE GREAT RIFT” ini dengan baik dan tepat waktu.

Skripsi ini merupakan puncak dari proses akademik yang panjang dan disusun sebagai salah satu syarat kelulusan untuk meraih gelar Sarjana Ilmu Komputer dari Program Studi Teknologi Informasi (TI) di Universitas Amikom Yogyakarta.

Penyelesaian skripsi ini tentu tidak terlepas dari dukungan, bimbingan, serta doa dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, saya ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

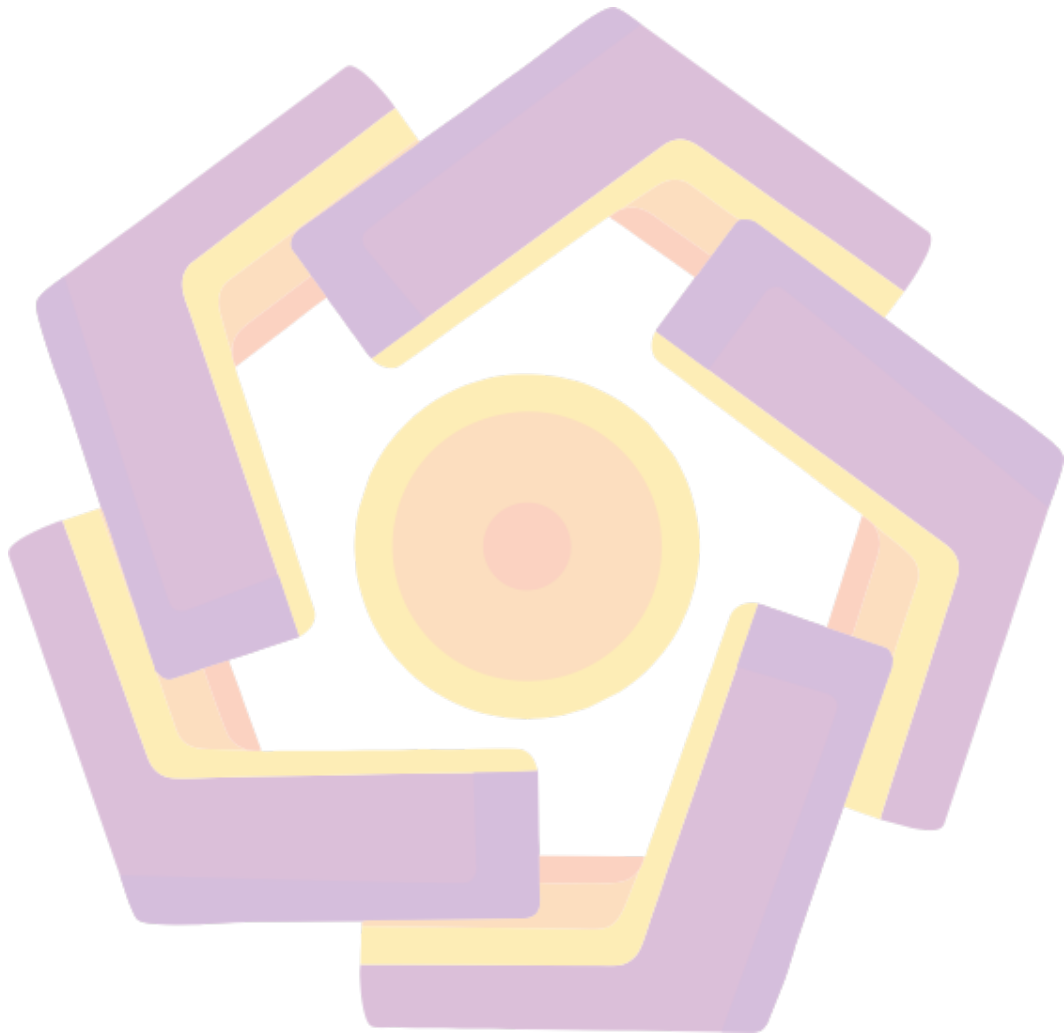
1. Kedua Orang Tua Tercinta dan keluarga besar, atas restu, dukungan material, moral, serta doa yang tiada henti mengiringi setiap langkah hingga skripsi ini selesai.
2. Bapak Buyut Khoirul Umri, M.Kom, selaku Dosen Pembimbing. Terima kasih atas kesabaran, waktu, arahan, dan bimbingan yang konstruktif selama proses penyusunan skripsi ini. Ilmu dan semangat yang Bapak berikan sangatlah berharga.
3. Bapak Agus Purwanto, M.Kom, beserta seluruh jajaran dosen dan staf administrasi di lingkungan Program Studi Teknologi Informasi. Terima kasih atas segala ilmu, fasilitas, dan kemudahan administrasi yang telah diberikan selama masa perkuliahan.
4. Teman-teman seperjuangan atas dukungan, kerja sama, dan semangat kolektif yang telah memotivasi saya dalam menyelesaikan tugas akhir ini.

Saya menyadari sepenuhnya bahwa skripsi ini masih memiliki banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, saya sangat terbuka dan menghargai segala bentuk kritik dan saran yang membangun demi perbaikan di masa mendatang. Akhir kata, saya berharap semoga karya sederhana ini dapat

memberikan manfaat dan kontribusi positif, khususnya bagi dunia pengembangan game.

Yogyakarta, 23 Februari 2026

Penulis



DAFTAR ISI

IMPLEMENTASI SISTEM STATE PLAYER MENGGUNAKAN FINITE STATE MACHINE PADA GAME THE GREAT RIFT	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN SKRIPSI	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN.....	xi
DAFTAR LAMBANG DAN SINGKATAN	xii
DAFTAR ISTILAH.....	xiii
INTISARI	xv
ABSTRACT.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	7
2.1 Studi Literatur	7
2.2 Dasar Teori.....	11

2.3	Proses Produksi	13
2.4	Analisis Kebutuhan Sistem	16
2.5	Evaluasi	18
2.6	Skala Likert	18
BAB III TINJAUAN PUSTAKA		20
3.1	Objek Penelitian	20
3.2	Alur Penelitian	21
3.3	Metode Pengumpulan Data	21
BAB IV HASIL DAN PEMBAHASAN		32
4.1	Produksi	32
4.2	Pasca Produksi	51
BAB V PENUTUP		64
5.1	Kesimpulan	64
5.2	Saran	65
REFERENSI		66
LAMPIRAN		69

DAFTAR TABEL

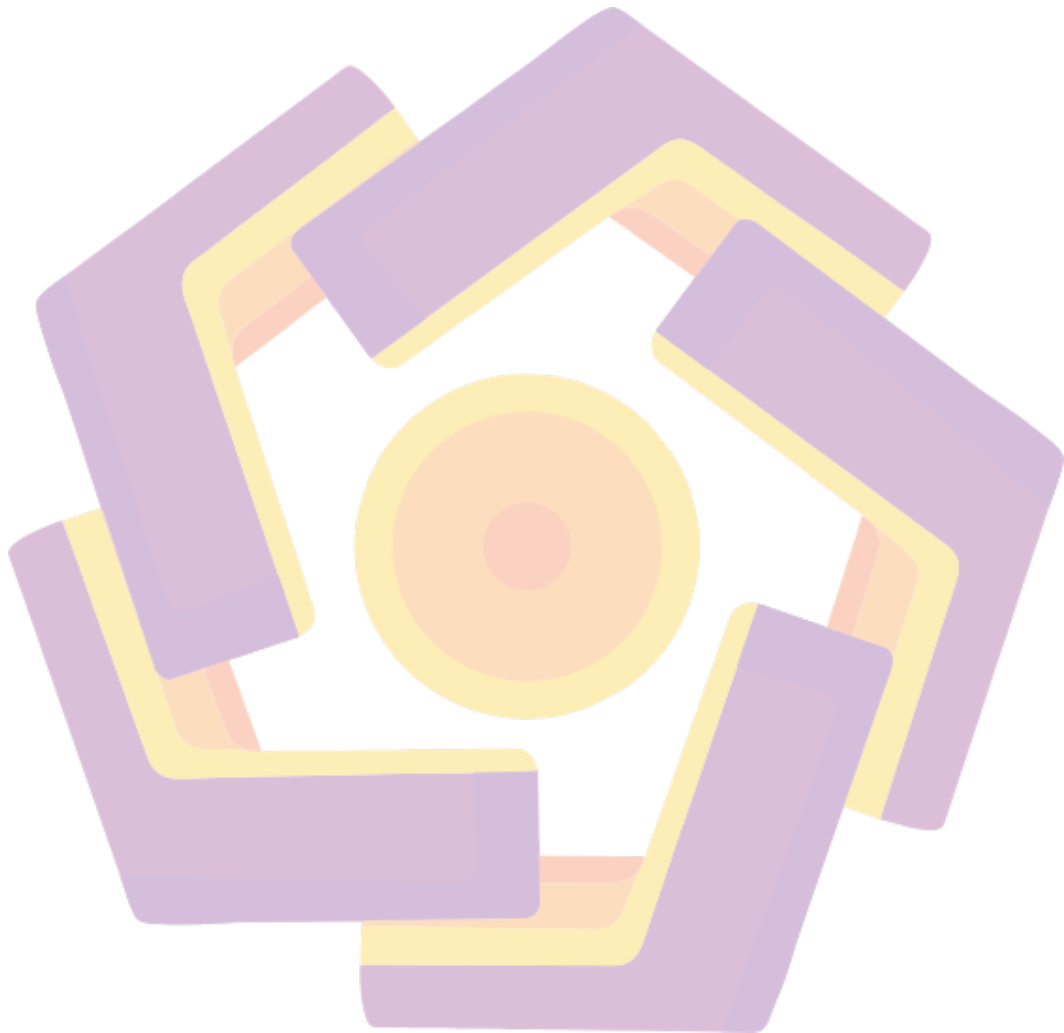
Tabel 2.1. 1 Keaslian Penelitian	9
Tabel 2.6. 1 Bobot Nilai.....	19
Tabel 2.6. 2 Persentase Nilai.....	19
Tabel 3. 1 Spesifikasi Perangkat Keras (Hardware)	25
Tabel 3. 2 Spesifikasi Perangkat Lunak (Software).....	26
Tabel 3. 3 Analisis Aspek Produksi	27
Tabel 4. 1 Hasil Pengujian Black Box Finite State Machine.....	51
Tabel 4. 2 Basis Path.....	54
Tabel 4. 3 Comprehensive State Test	55
Tabel 4. 4 Uji Kelayakan Ahli	58
Tabel 4. 5 Bobot Penilaian.....	59
Tabel 4. 6 Persentase Nilai Uji	59
Tabel 4. 7 Uji Kelayakan Umum	61
Tabel 4. 8 Bobot Nilai.....	62
Tabel 4. 9 Persentase Nilai Uji	62

DAFTAR GAMBAR

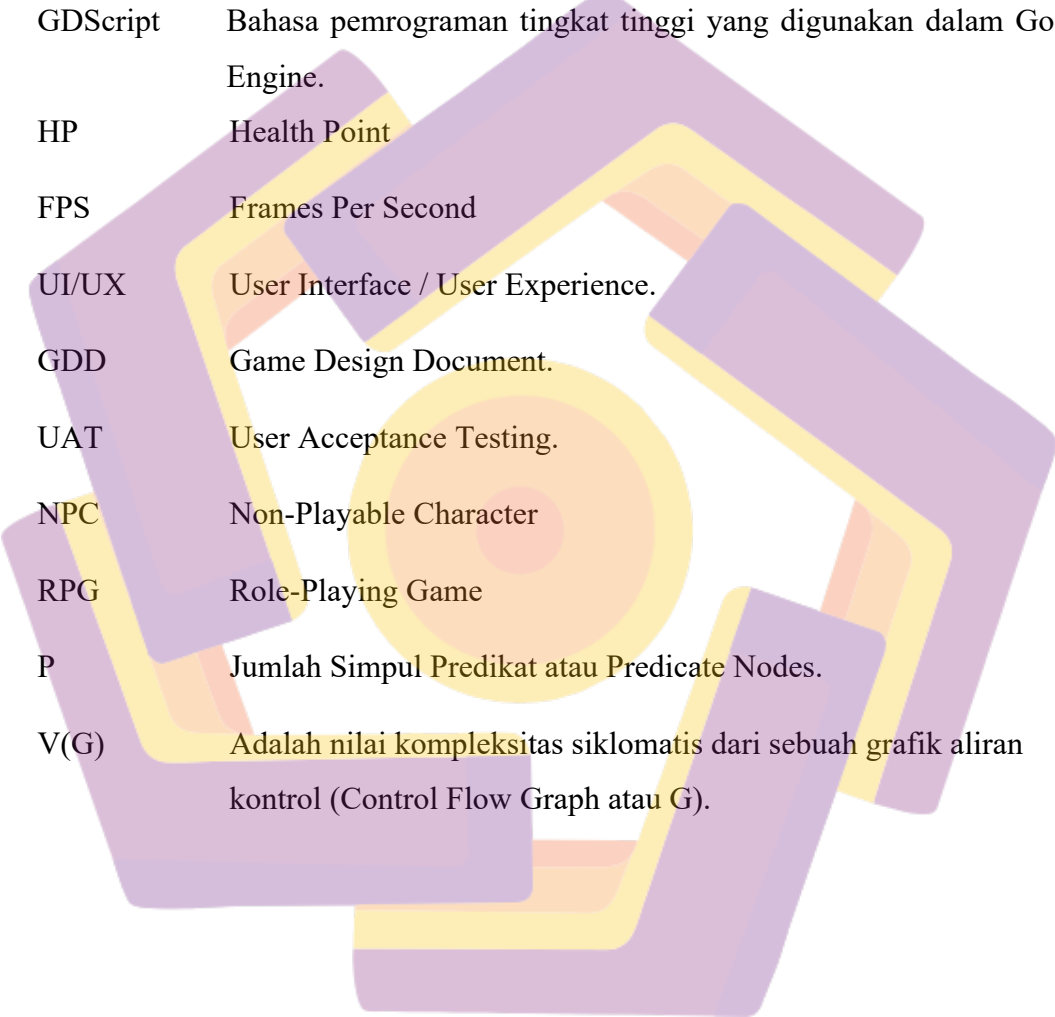
Gambar 2.2. 1 Game Agent dan Game State	11
Gambar 2.2. 2 FSM.....	12
Gambar 3. 1 Alur Penelitian	21
Gambar 3. 2 Hades.....	23
Gambar 3. 3 Hyper Light Drifter	24
Gambar 3. 4 Karakter Asset.....	29
Gambar 3. 5 Flowchart The Great Rift	31
Gambar 4. 1 StateMachine func _ready()	32
Gambar 4. 2 Child Node StateMachine	33
Gambar 4. 3 StateMachine func _initialize()	33
Gambar 4. 4 StateMachine func change_state, _physics_process, dan _unhandled_input.....	34
Gambar 4. 5 Skrip State	35
Gambar 4. 6 Implementasi Penanganan Kondisi Death	36
Gambar 4. 7 Kondisi saat Death	37
Gambar 4. 8 func take_damage	38
Gambar 4. 9 func die()	39
Gambar 4. 10 Idle State	41
Gambar 4. 11 Movement Player	42
Gambar 4. 12 Visualisasi Karakter Bergerak	43
Gambar 4. 13 Attack Player.....	45
Gambar 4. 14 Visualisasi Player Attack	45
Gambar 4. 15 Player Dash	46
Gambar 4. 16 Skill Player.....	47
Gambar 4. 17 Visualisasi Saat Player Menggunakan Skill.....	48
Gambar 4. 18 Visual Player Menggunakan Rage Mode.....	49
Gambar 4. 19 Hit Mekanisme	50
Gambar 4. 20 Visual Player Saat Terkena Damage.....	50
Gambar 4. 21 Flowchart White Testing.....	53
Gambar 4. 22 Hasil Pengujian Performa	56

DAFTAR LAMPIRAN

Lampiran 1. Responden Kuisioner Uji Kelayakan Ahli	69
Lampiran 2. Responden Kuisioner Uji Pengguna Umum	77

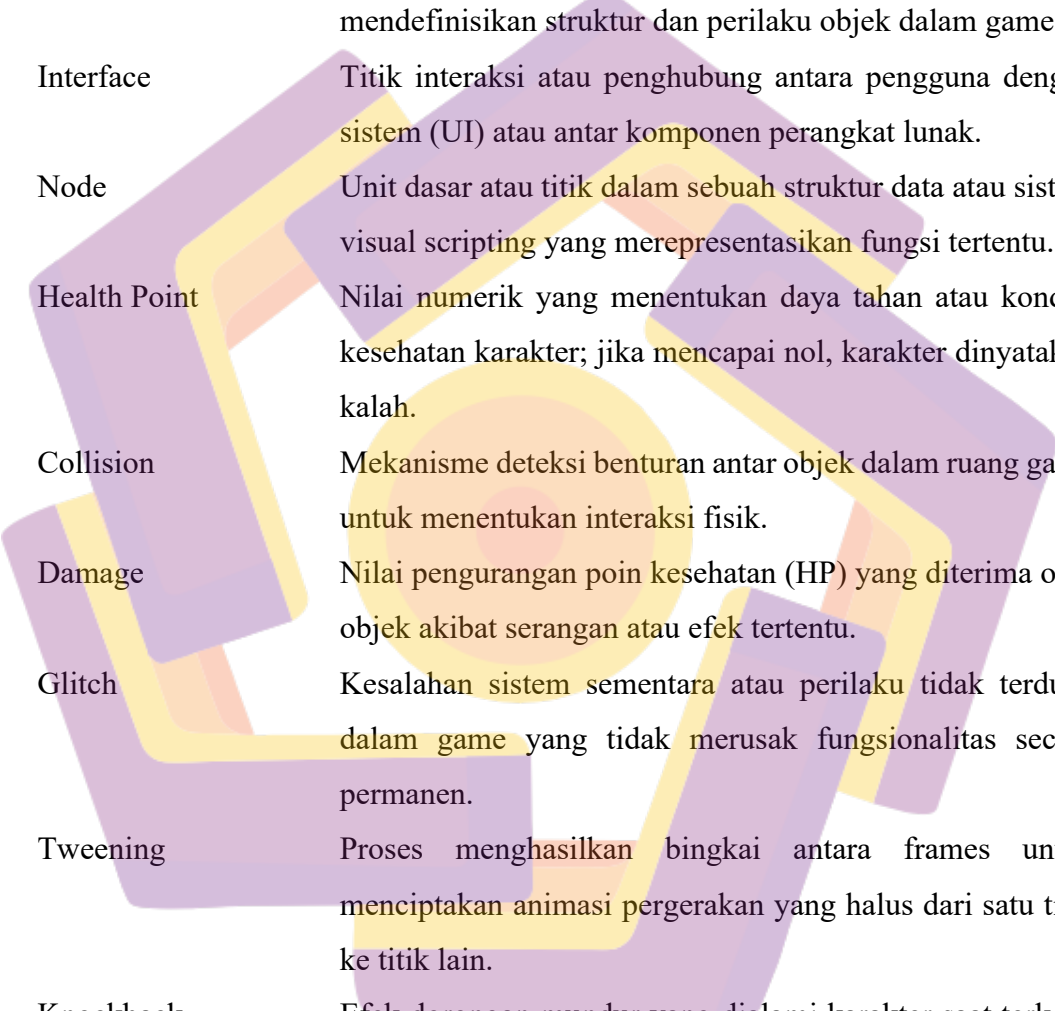


DAFTAR LAMBANG DAN SINGKATAN

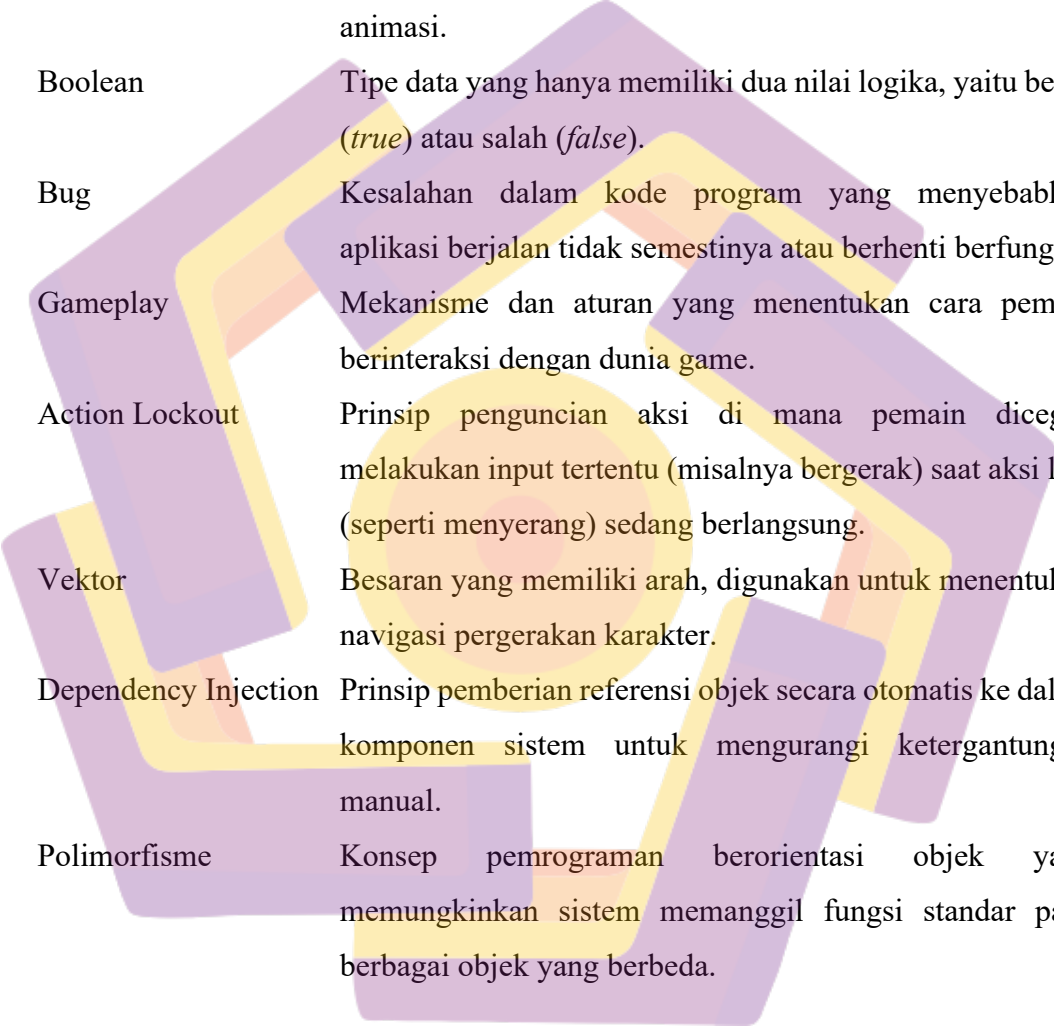


FSM	Finite State Machine
RPG	Role-Playing Game
Action RPG	Genre permainan yang menggabungkan elemen aksi dan role-playing.
GDLC	Game Development Life Cycle.
GDScript	Bahasa pemrograman tingkat tinggi yang digunakan dalam Godot Engine.
HP	Health Point
FPS	Frames Per Second
UI/UX	User Interface / User Experience.
GDD	Game Design Document.
UAT	User Acceptance Testing.
NPC	Non-Playable Character
RPG	Role-Playing Game
P	Jumlah Simpul Predikat atau Predicate Nodes.
$V(G)$	Adalah nilai kompleksitas siklomatis dari sebuah grafik aliran kontrol (Control Flow Graph atau G).

DAFTAR ISTILAH



State	Kondisi atau tingkatan aktivitas sistem di mana sistem hanya mengenali satu kondisi pada satu waktu.
Event	Kejadian atau pemicu tertentu yang menyebabkan terjadinya transisi antar <i>state</i> .
Blueprint	Kerangka kerja atau desain awal sistem yang mendefinisikan struktur dan perilaku objek dalam game.
Interface	Titik interaksi atau penghubung antara pengguna dengan sistem (UI) atau antar komponen perangkat lunak.
Node	Unit dasar atau titik dalam sebuah struktur data atau sistem visual scripting yang merepresentasikan fungsi tertentu.
Health Point	Nilai numerik yang menentukan daya tahan atau kondisi kesehatan karakter; jika mencapai nol, karakter dinyatakan kalah.
Collision	Mekanisme deteksi benturan antar objek dalam ruang game untuk menentukan interaksi fisik.
Damage	Nilai pengurangan poin kesehatan (HP) yang diterima oleh objek akibat serangan atau efek tertentu.
Glitch	Kesalahan sistem sementara atau perilaku tidak terduga dalam game yang tidak merusak fungsionalitas secara permanen.
Tweening	Proses menghasilkan bingkai antara frames untuk menciptakan animasi pergerakan yang halus dari satu titik ke titik lain.
Knockback	Efek dorongan mundur yang dialami karakter saat terkena serangan atau benturan.
Player	Subjek atau pengguna yang mengendalikan entitas di dalam game.
Character	Entitas atau representasi visual di dalam game yang memiliki atribut dan perilaku tertentu.



Scalable	Kemampuan sistem untuk menangani penambahan beban atau perubahan ukuran tanpa memengaruhi performa dasar.
Input	Perintah atau sinyal yang diberikan pengguna melalui perangkat keras (<i>keyboard, mouse, controller</i>) ke dalam sistem game.
Buffering	Teknik menyimpan input atau data sementara agar eksekusi perintah tetap berjalan lancar meski terjadi latensi atau jeda animasi.
Boolean	Tipe data yang hanya memiliki dua nilai logika, yaitu benar (<i>true</i>) atau salah (<i>false</i>).
Bug	Kesalahan dalam kode program yang menyebabkan aplikasi berjalan tidak semestinya atau berhenti berfungsi.
Gameplay	Mekanisme dan aturan yang menentukan cara pemain berinteraksi dengan dunia game.
Action Lockout	Prinsip penguncian aksi di mana pemain dicegah melakukan input tertentu (misalnya bergerak) saat aksi lain (seperti menyerang) sedang berlangsung.
Vektor	Besaran yang memiliki arah, digunakan untuk menentukan navigasi pergerakan karakter.
Dependency Injection	Prinsip pemberian referensi objek secara otomatis ke dalam komponen sistem untuk mengurangi ketergantungan manual.
Polimorfisme	Konsep pemrograman berorientasi objek yang memungkinkan sistem memanggil fungsi standar pada berbagai objek yang berbeda.

INTISARI

Pengembangan system kontrol karakter player dalam game bergenre Action RPG menghadapi tantangan kompleks dalam mengelola berbagai aksi dan state aksi secara simultan, seperti movement, dash, attack, skill, dan respon saat terkena hit musuh (seperti stun maupun knockback). Permasalahan yang sering terjadi adalah adanya konflik antara logika dan inkonsistensi state, dalam artian, saat player terkena state stun seharusnya tidak dapat melakukan state attack maupun movement yang berdampak pada game feel yang buruk dan kurangnya responsivitas yang logis, sehingga mengurangi kualitas pengalaman dalam bermain game Action RPG.

Penelitian ini bertujuan untuk menyelesaikan masalah inkonsistensi state tersebut dengan mengimplementasikan rancangan kontrol karakter yang kokoh melalui model Finite State Machine (FSM). FSM diterapkan sebagai kerangka kerja State Management dengan langkah-langkah penelitian permodelan, perancangan, dan implementasi pada script karakter player, memanfaatkan prinsip action lockout untuk memastikan hanya satu aksi utama yang aktif pada satu waktu.

Hasil implementasi menunjukkan bahwa penggunaan FSM berhasil mengelola transisi aksi secara terstruktur dan mencegah konflik state secara efektif, khususnya dalam menangani kondisi interupsi seperti Hit Stun dan aktivasi Attack. Kontribusi penelitian ini adalah menyediakan model implementasi FSM yang terperinci sebagai solusi perancangan yang efisien dan skalabel untuk mengelola kompleksitas perilaku player. Hasil dari penelitian ini dapat dimanfaatkan oleh Game Developer yang berfokus pada game Action RPG dan tertarik pada penerapan pola desain terstruktur untuk meningkatkan game feel dan pemeliharaan kode.

Kata Kunci: Fine State Machine, State Management, Kontrol Player, Action RPG, FSM

ABSTRACT

The development of the character player control system in the Action RPG (Role-Playing Game) genre faces complex challenges in simultaneously managing various actions and action states, such as movement, dash, attack, skill usage, and responding to enemy hits (like stun or knockback). A common problem is the conflict between logic and state inconsistency, meaning that when the player is inflicted with a stun status, they should ideally be unable to perform actions like attacking or moving. This inconsistency results in a poor game feel and lack of logical responsiveness, thereby diminishing the quality of the Action RPG gaming experience.

This research aims to resolve the issue of state inconsistency by implementing a robust character control architecture using the Finite State Machine (FSM) model. The FSM is applied as a State Management framework, with the research steps involving modeling, design, and implementation within the player character script, leveraging the principle of action lockout to ensure that only one primary action is active at any given time.

The implementation results demonstrate that the use of FSM successfully manages action transitions in a structured manner and effectively prevents status conflicts, especially in handling interruption conditions such as Hit Stun and attack activation. The contribution of this research is providing a detailed FSM implementation model as an efficient and scalable architectural solution for managing the complexity of player behavior. The findings of this study can be utilized by Game Developers who focus on the Action RPG genre and are interested in applying structured design patterns to enhance game feel and code maintainability.

Keywords: Finite State Machine, State Management, Player Control, Action RPG, FSM