

**PERANCANGAN DAN IMPLEMENTASI SISTEM
PENGLIHATAN MUSUH BERBASIS LINECASTING DALAM
GAME THE LAST LIGHT OF THE MOTHER**

SKRIPSI

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



disusun oleh

Adistian Rahman Maulana

22.82.1486

Kepada

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

**PERANCANGAN DAN IMPLEMENTASI SISTEM
PENGLIHATAN MUSUH BERBASIS LINECASTING DALAM
GAME THE LAST LIGHT OF THE MOTHER**

SKRIPSI

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



disusun oleh

Adistian Rahman Maulana

22.82.1486

Kepada

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

HALAMAN PERSETUJUAN

SKRIPSI

**PERANCANGAN DAN IMPLEMENTASI SISTEM PENGLIHATAN
MUSUH BERBASIS LINECASTING DALAM GAME THE LAST LIGHT
OF THE MOTHER**

yang disusun dan diajukan oleh

Adistian Rahman Maulana

22.82.1486

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 26 Februari 2026

Dosen Pembimbing,



Buyut Khoirul Umri M. Kom

NIK. 190302652

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

SKRIPSI

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGLIHATAN MUSUH BERBASIS LINECASTING DALAM GAME THE LAST LIGHT OF THE MOTHER

yang disusun dan diajukan oleh

Adistian Rahman Maulana

22.82.1486

Telah dipertahankan di depan Dewan Penguji
pada tanggal 26 February 2026

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Bernadhed, S.Kom., M.Kom
NIK. 190302243

Muhammad Misbahul Munir, M.Kom
NIK. 190302497

Buyut Khoirul Umri M.Kom
NIK. 190302652

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal 26 February 2026

DEKAN FAKULTAS ILMU KOMPUTER



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

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan di bawah ini,

Nama mahasiswa : Adistian Rahman Maulana
NIM : 22.82.1486

Menyatakan bahwa Skripsi dengan judul berikut:

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGLIHATAN MUSUH BERBASIS LINECASTING DALAM GAME THE LAST LIGHT OF THE MOTHER

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, 26 February 2026

Yang Menyatakan,



Adistian Rahman Maulana

KATA PENGANTAR

Puji syukur atas kehadiran Tuhan yang maha Esa yang telah memberikan rahmat, kasih sayang, serta kesempatan, sehingga penulis dapat menyelesaikan skripsi Reguler dengan judul Perancangan dan Implementasi Sistem Penglihatan Musuh Berbasis Linecasting dalam game The Last Light of the Mother. Penyusunan skripsi Reguler ini bertujuan untuk menyelesaikan salah satu mata kuliah dan kelulusan sebelum memperoleh gelar sarjana (strata satu) dari Universitas Amikom Yogyakarta. Penyusunan skripsi ini juga tidak terlepas dari berbagai pihak yang telah memberikan bantuan baik secara langsung maupun tidak langsung.

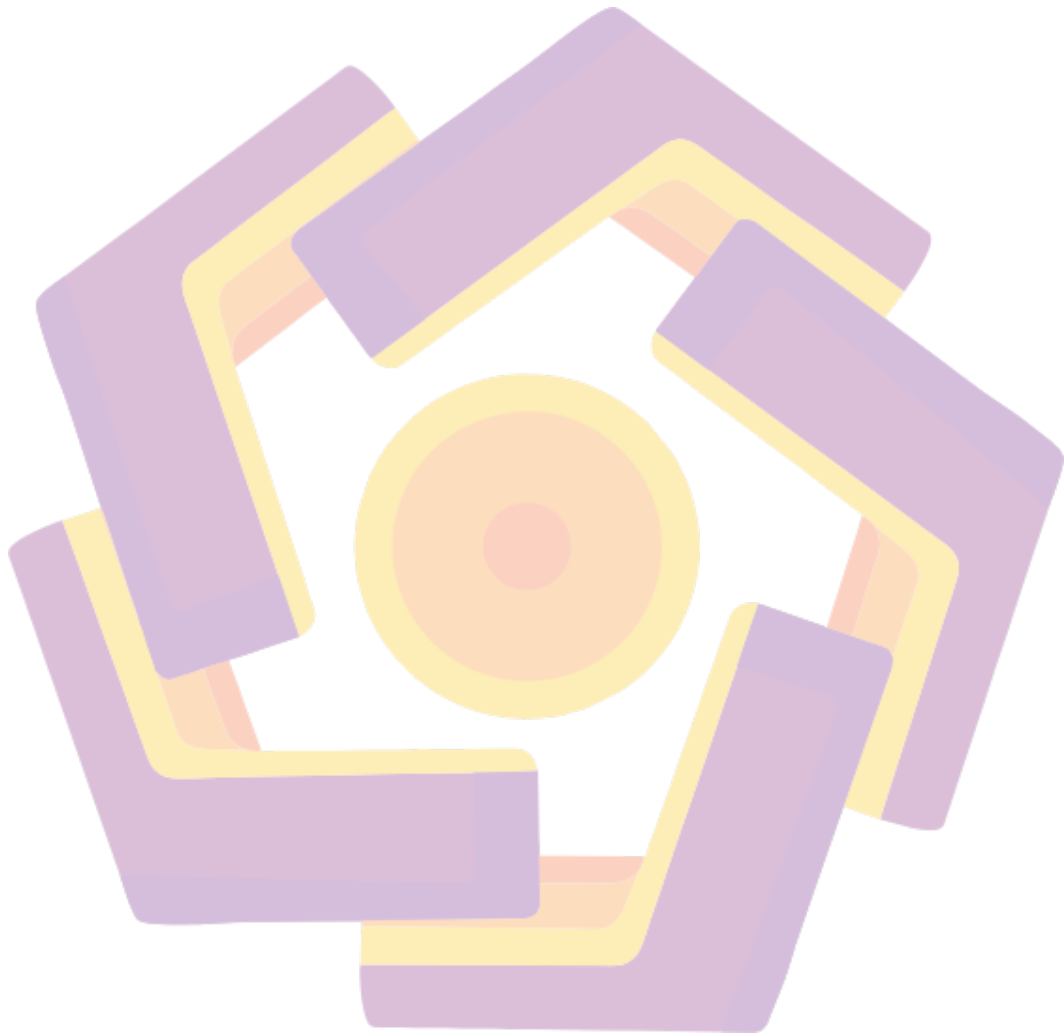
Penulis menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. M. Suyanto, M.M., selaku Rektor Universitas Amikom Yogyakarta.
2. Prof. Dr. Kusriani, M.Kom. , selaku Dekan Fakultas Ilmu Komputer Universitas Amikom Yogyakarta.
3. Bapak Agus Purwanto, M.Kom., selaku Kaprodi Studi Teknologi Informasi.
4. Bapak Buyut Khoirul Umri, M.Kom, selaku dosen pembimbing yang sudah membimbing pada penulis.
5. Ayah dan Ibu yang selalu mendoakan penulis.
6. Teman - teman dari jurusan Teknologi Informasi Universitas Amikom Yogyakarta angkatan 2022.

Semoga dengan adanya skripsi ini dapat memberikan manfaat dan sebagai informasi maupun pengetahuan bagi pembaca. Penulis menyadari adanya kekurangan dalam penulisan. Kiranya kritik dan saran dapat diberikan sebagai masukan untuk penyempurnaan dalam penulisan yang akan datang

Yogyakarta, 26 Februari 2026

Penulis



DAFTAR ISI

HALAMAN JUDUL	I
HALAMAN PERSETUJUAN	II
HALAMAN PENGESAHAN	III
HALAMAN PERNYATAAN KEASLIAN SKRIPSI	IV
KATA PENGANTAR	V
DAFTAR ISI	VII
DAFTAR TABEL	X
DAFTAR GAMBAR	XI
DAFTAR LAMPIRAN	XIV
DAFTAR ISTILAH	XV
INTISARI	XXIV
<i>ABSTRACT</i>	XXV
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 Studi Literatur	5
2.2 Dasar Teori	11
2.2.1 Konsep Posisi dan Pergerakan Objek 2D	11
2.2.2 RigidBody pada Unity Engine	11
2.2.3 <i>Collision Detection</i>	11
2.2.4 Teori <i>Linecasting</i>	12

2.2.5 Game Development Life Cycle.....	13
2.2.6 Game.....	14
2.2.7 Unity.....	14
2.2.8 Visual Studio Code.....	15
2.2.9 Black Box Testing.....	15
2.2.10 White Box Testing.....	16
2.2.11 Skala Likert.....	16
2.3 Kebutuhan Sistem.....	17
2.3.1 Kebutuhan Fungsional.....	17
2.3.2 Kebutuhan Non-Fungsional.....	18
2.4 Evaluasi.....	18
BAB III METODE PENELITIAN.....	19
3.1 Objek Penelitian.....	19
3.1.1 Program Studi Teknologi Informasi.....	19
3.1.2 Gelar Karya Mahasiswa Teknologi Informasi.....	19
3.2 Alur Penelitian.....	20
3.3 Metode Pengumpulan Data.....	20
3.3.1 Observasi.....	20
3.3.2 Wawancara.....	25
3.3.3 Studi Eksperimental.....	27
3.4 Analisis Kebutuhan.....	27
3.4.1 Kebutuhan Fungsional.....	27
3.4.2 Kebutuhan Non-Fungsional.....	28
3.5 Analisis Aspek Produksi.....	30
3.6 Pra-Produksi.....	32
3.6.1 Desain Karakter.....	32
3.6.2 Ruang Lingkup Permainan.....	34
3.6.3 Alur Permainan.....	35
3.6.4 Produksi.....	35

3.6.5 Pengujian.....	36
3.6.6 Pasca-produksi.....	36
BAB IV HASIL DAN PEMBAHASAN.....	37
4.1 Tahap Produksi.....	37
4.1.1 Pembangunan Level dan Tata Letak.....	37
4.1.2 Konfigurasi Visual.....	39
4.1.3 <i>Tilemap Collider</i> dan <i>Layer Tag</i>	41
4.1.4 Persiapan Karakter Utama.....	42
4.1.5 Implementasi Mekanik Pergerakan Karakter.....	44
4.2 Logika Orientasi Arah Hadap (<i>Facing Direction</i>).....	45
4.2.1 Pembangunan Level dan Tata Letak.....	46
4.3 Skrip Navigasi Patroli.....	47
4.3.1 Deklarasi Variabel dan Konfigurasi Patroli.....	48
4.3.2 Inisialisasi dan Validasi Komponen.....	49
4.4 Logika Deteksi Visual Hibrid.....	49
4.4.1 Implementasi Sistem Deteksi Musuh dengan Teknik <i>Linecasting</i>	49
4.4.2 Sinkronisasi Logika Program dengan Animasi.....	57
4.4.3 Skrip Navigasi Patroli.....	62
4.4.4 <i>AI Stealth</i>	69
4.5 Testing.....	81
4.5.1 Alpha Testing Black Box Testing.....	82
4.5.2 Alpha Testing White Box Testing.....	84
4.5.3 Beta Testing.....	95
4.6 Kesimpulan Evaluasi.....	102
BAB V PENUTUP.....	103
5.1 Kesimpulan.....	103
5.2 Saran.....	104
REFERENSI.....	106
LAMPIRAN.....	108

DAFTAR TABEL

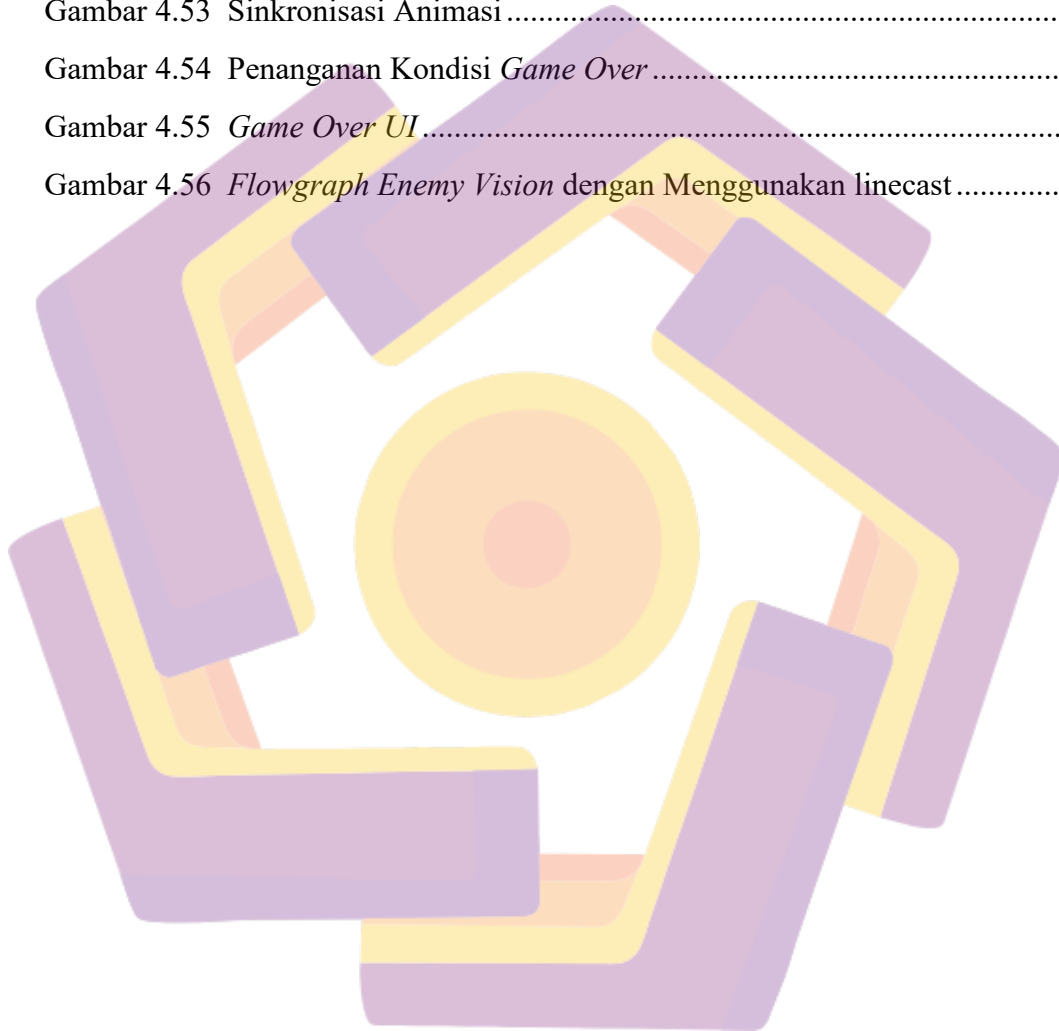
Table 2.1 Keaslian Penelitian	7
Table 2.1 Tabel Bobot Nilai	16
Table 3.1 Alur Penelitian	20
Table 3.2 Tabel <i>Hardware</i>	28
Table 3.3 Tabel <i>Software</i>	29
Table 3.4 Tabel <i>Brainware</i>	29
Table 3.5 Tabel Pemetaan Pekerjaan	30
Table 3.6 Tabel Analisis Aspek Produksi	30
Table 4.1 <i>Black Box Testing</i>	82
Table 4.2 <i>White Box Testing</i>	89
Table 4.3 Beta Testing oleh Umum	96

DAFTAR GAMBAR

Gambar 2.1 <i>Linecast terhalang</i>	12
Gambar 2.2 Rumus <i>Linecast</i>	13
Gambar 2.3 <i>Game Development Life Cycle</i>	13
Gambar 3.1 <i>Metal Gear</i>	21
Gambar 3.2 <i>Monaco What's Yours Is Mine</i>	22
Gambar 3.3 <i>Mark of the Ninja</i>	24
Gambar 3.4 Kartika	33
Gambar 3.5 Tahu Alas	34
Gambar 4.1 Labirin <i>Stage 2</i>	38
Gambar 4.2 Labirin <i>Stage 3</i>	39
Gambar 4.3 Labirin <i>Global Light Volume 0</i>	40
Gambar 4.4 Labirin <i>spot light</i>	40
Gambar 4.5 <i>tilemap collider</i>	41
Gambar 4.6 Komponen <i>Rigidbody</i>	43
Gambar 4.7 <i>Tag</i>	43
Gambar 4.8 <i>Child-Parent</i>	44
Gambar 4.9 logika <i>UpdateAnimatorParameter</i>	45
Gambar 4.10 <i>Parameter Animator</i>	46
Gambar 4.11 <i>Blendtree</i>	46
Gambar 4.12 logika <i>ApplyMovement</i>	47
Gambar 4.13 logika <i>Awake Update</i> dan <i>GatherInput</i>	47
Gambar 4.14 Variabel Komponen Hibrid	50
Gambar 4.15 Validasi <i>Trigger</i>	51
Gambar 4,16 Kodongan Fungsi Penglihatan Kerucut	51
Gambar 4,17 Menghadap Kiri Atas	52
Gambar 4.18 menghadap Kanan Bawah	53

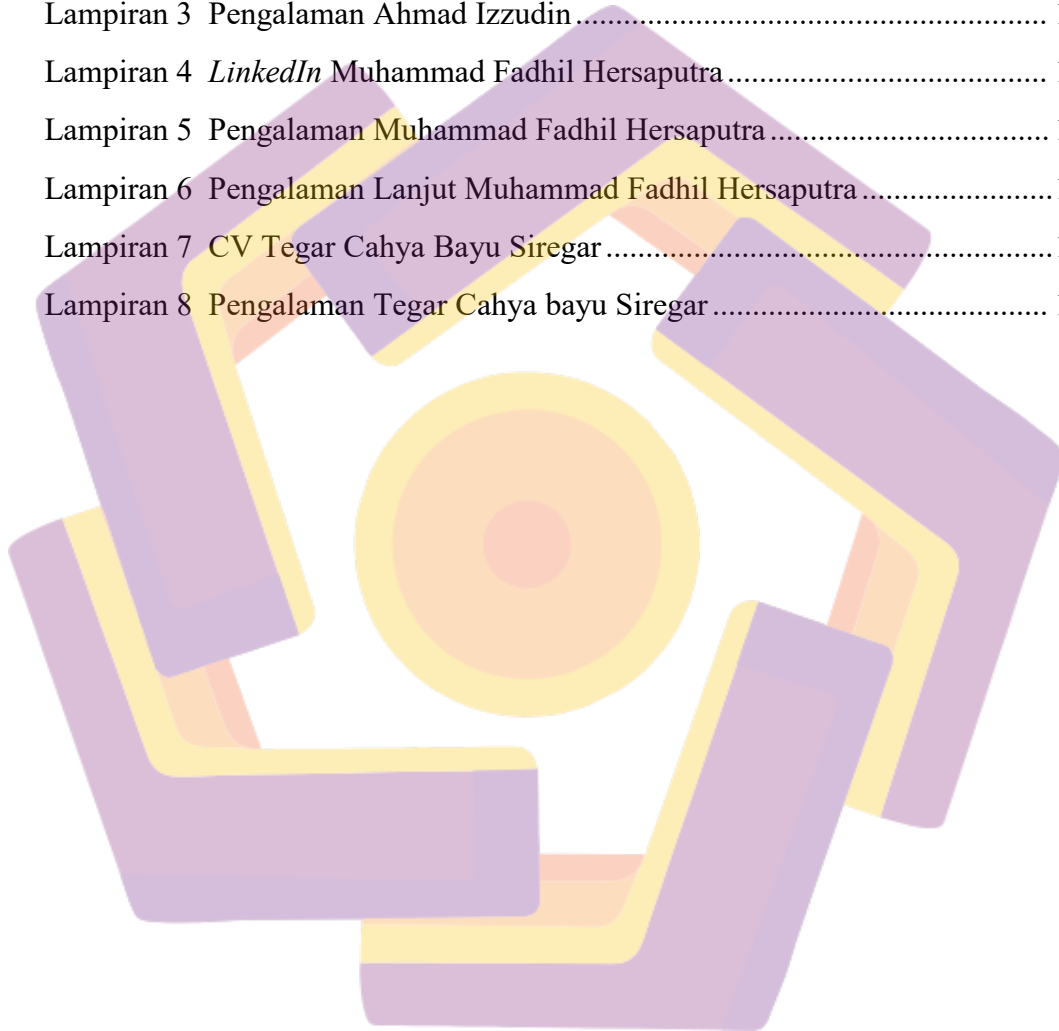
Gambar 4.19 <i>OnTrigger2D</i>	54
Gambar 4.20 <i>Linecast</i>	54
Gambar 4.21 <i>Debug True</i>	55
Gambar 4.22 <i>Debug false</i>	56
Gambar 4.23 Status Visibilitas	56
Gambar 4.24 <i>Debugging Gizmos</i>	57
Gambar 4.25 Variabel <i>enemyMovement_Simple</i>	58
Gambar 4.26 <i>Awake</i>	59
Gambar 4.27 <i>FacingDirection</i>	59
Gambar 4.28 <i>StopMovement</i>	60
Gambar 4.29 <i>FixedUpdate</i>	61
Gambar 4.30 <i>UpdateAnimatorParameter</i>	61
Gambar 4.31 <i>OnDisable</i>	62
Gambar 4.32 Variabel Patroli	63
Gambar 4.33 <i>Awake</i>	64
Gambar 4.34 <i>OnEnable</i>	64
Gambar 4.35 Penanganan Non-Aktifasi	65
Gambar 4.36 Pembaruan Patroli	65
Gambar 4.37 Pengambilan Target Patroli	66
Gambar 4.38 Mekanisme Penundaan di Titik Patroli	67
Gambar 4.39 Algoritma Navigasi Titik	68
Gambar 4.40 Eksekusi Pergerakan Fisik	69
Gambar 4.41 Struktur Finite State Machine	70
Gambar 4.42 Inisialisasi dan Konfigurasi Awal	71
Gambar 4.43 Siklus Pembaruan Logika	72
Gambar 4.44 Siklus Pembaruan Logika Lanjutan	72
Gambar 4.45 Logika Transisi Kecurigaan	73
Gambar 4.46 Visual Indikator Curiga	74
Gambar 4.47 Logika Pengejaran dan Penangkapan	74

Gambar 4.48 Visual Indikator Pengejaran	75
Gambar 4.49 Mekanisme Kehilangan Jejak dan Pencarian	76
Gambar 4.50 Manajemen Transisi <i>State</i>	77
Gambar 4.51 Manajemen Transisi <i>State</i> Lanjutan	78
Gambar 4.52 Eksekusi Pergerakan Fisik	79
Gambar 4.53 Sinkronisasi Animasi	79
Gambar 4.54 Penanganan Kondisi <i>Game Over</i>	80
Gambar 4.55 <i>Game Over UI</i>	81
Gambar 4.56 <i>Flowgraph Enemy Vision</i> dengan Menggunakan linecast	85



DAFTAR LAMPIRAN

Lampiran 1 CV Bagus Darmawan	108
Lampiran 2 CV Ahmad Izzudin	109
Lampiran 3 Pengalaman Ahmad Izzudin	110
Lampiran 4 <i>LinkedIn</i> Muhammad Fadhil Hersaputra	111
Lampiran 5 Pengalaman Muhammad Fadhil Hersaputra	112
Lampiran 6 Pengalaman Lanjut Muhammad Fadhil Hersaputra	113
Lampiran 7 CV Tegar Cahya Bayu Siregar	114
Lampiran 8 Pengalaman Tegar Cahya bayu Siregar	115



DAFTAR ISTILAH

2D	Dua dimensi; grafis yang hanya memiliki panjang dan lebar, tanpa kedalaman.
AI	Artificial Intelligence; kecerdasan buatan yang membuat musuh atau karakter dalam game dapat berpikir dan bereaksi seperti makhluk hidup.
Alpha Testing	Tahap pengujian awal yang dilakukan oleh pengembang sendiri untuk mencari bug dan memastikan fitur berjalan sesuai rencana.
Animator	Komponen di Unity yang mengatur pergantian animasi karakter berdasarkan kondisi tertentu.
Animator Controller	File di Unity yang menyimpan grafik hubungan antar status animasi dan transisinya.
Behavior Tree	Struktur logika AI yang mengatur perilaku karakter secara hierarkis, mirip cabang pohon keputusan.
Beta Testing	Tahap pengujian lanjutan yang melibatkan pengguna luar atau pemain biasa untuk menilai pengalaman bermain.
Black Box Testing	Metode pengujian perangkat lunak yang fokus pada input dan output, tanpa melihat kode program di dalamnya.
Blend Tree	Fitur di Unity Animator yang memungkinkan perpaduan animasi, misalnya antara jalan lurus dan diagonal.
Body Type	Pengaturan tipe fisik pada komponen Rigidbody, misalnya Dynamic, Kinematic, atau Static.
Brainware	Sumber daya manusia atau orang-orang yang terlibat dalam pengembangan sistem.
C#	Bahasa pemrograman yang digunakan di Unity untuk

menulis logika permainan.

Child	Objek dalam hierarki Unity yang berada di bawah induk (parent) dan mengikuti transformasi induknya.
Collider 2D	Komponen Unity yang memberikan area tabrakan pada objek 2D, digunakan untuk mendeteksi sentuhan atau sebagai sensor.
Collision Detection	Sistem deteksi tabrakan yang menentukan kapan dua objek saling bersentuhan.
Coroutine	Fungsi di Unity yang dapat menunda eksekusi atau menjalankan proses secara bertahap tanpa menghentikan game.
Debugging	Proses mencari dan memperbaiki kesalahan dalam kode program.
Dynamic	Tipe Rigidbody yang sepenuhnya dipengaruhi oleh gaya fisika seperti gravitasi dan tumbukan.
Enemy Vision System	Sistem penglihatan musuh; mekanisme yang menentukan kapan dan bagaimana musuh dapat melihat pemain.
Entry	Titik awal dalam grafik Animator tempat animasi mulai dijalankan.
Euclidean Distance	Jarak lurus antara dua titik; digunakan untuk menghitung seberapa jauh posisi pemain dari musuh.
Event Handler	Fungsi khusus yang otomatis dipanggil saat terjadi peristiwa tertentu, misalnya OnTriggerEnter2D.
Field of View (FoV)	Sudut pandang atau area pandang musuh; seberapa lebar musuh dapat melihat ke sekeliling.
Finite State Machine (FSM)	Model logika yang membagi perilaku karakter ke dalam status-status terpisah, misalnya Patrol, Chase, dan Idle.
FixedUpdate	Fungsi di Unity yang berjalan selaras dengan siklus fisika,

	cocok untuk menggerakkan objek berbasis Rigidbody.
Flip	Membalik arah hadap sprite secara horizontal.
Fog of War	Efek visual yang menyembunyikan area peta yang tidak terlihat oleh pemain atau musuh.
Frame	Satu putaran gambar dalam game; game berjalan dalam rangkaian frame per detik.
Game Development Life Cycle (GDLC)	Siklus pengembangan game yang terdiri dari tahap pra-produksi, produksi, pengujian, dan pasca-produksi.
Game Engine	Perangkat lunak kerangka kerja untuk membuat game, seperti Unity.
Game Designer	Perancang game; orang yang bertanggung jawab merancang aturan, alur, dan pengalaman bermain.
Game Artist	Seniman game; orang yang membuat aset visual seperti karakter, lingkungan, dan antarmuka.
GameObject	Wadah kosong di Unity yang dapat diisi komponen untuk menjadi objek dalam game.
Game Over	Kondisi saat pemain kalah dan permainan berakhir.
Gizmos	Alat bantu visual di editor Unity untuk debugging, seperti menggambar garis atau lingkaran.
Global Light Volume	Komponen pencahayaan global di Unity 2D yang mengatur terang-gelapnya seluruh area.
Gravity Scale	Nilai gravitasi pada Rigidbody 2D; jika diatur 0, objek tidak akan jatuh ke bawah.
Grid	Sistem kotak-kotak untuk menempatkan objek secara rapi dan presisi di Unity.
Hardware	Perangkat keras fisik yang digunakan untuk mengembangkan atau menjalankan game.

Hashing	Teknik optimasi dengan mengubah string teks menjadi angka ID untuk mempercepat proses komputasi.
Hybrid	Gabungan dua metode atau pendekatan; dalam penelitian ini menggabungkan Collider dan Linecast.
IDE	Integrated Development Environment; aplikasi untuk menulis dan menguji kode program.
Idle	Status diam; saat karakter tidak bergerak.
IntelliSense	Fitur pelengkapan kode otomatis di Visual Studio Code yang membantu menulis kode lebih cepat dan akurat.
IndexOutOfRangeException	Kesalahan program yang terjadi saat mencoba mengakses indeks di luar batas daftar atau array.
Input	Perintah yang diberikan pemain melalui keyboard, mouse, atau kontroler.
Inspector	Jendela di Unity untuk melihat dan mengubah properti komponen pada GameObject.
Isometric Game	Game dengan perspektif miring, menciptakan ilusi 3D tetapi tetap menggunakan aset 2D.
Kinematic	Tipe Rigidbody yang tidak dipengaruhi gaya fisika, tetapi masih dapat mendeteksi tabrakan.
Layer	Lapisan atau grup objek di Unity yang digunakan untuk mengatur interaksi fisika dan collision.
LayerMask	Filter yang menentukan lapisan objek mana yang akan diabaikan atau dideteksi oleh sistem fisika.
Line of Sight (LoS)	Garis pandang; kondisi di mana musuh dapat melihat pemain secara langsung tanpa halangan.
Linecast	Metode deteksi di Unity yang menarik garis lurus dari titik A ke titik B untuk mengecek apakah ada objek penghalang di antaranya.

List	Struktur data yang menyimpan kumpulan objek dalam urutan tertentu.
Loop	Pola perulangan; dalam patroli, musuh dapat bergerak memutar atau bolak-balik.
Mark of the Ninja	Game stealth 2D yang menjadi referensi penggunaan Linecast untuk deteksi musuh.
Masking	Teknik menyembunyikan atau memfilter objek tertentu agar tidak terdeteksi oleh sistem.
Mesh	Jaringan bentuk geometris yang membentuk tampilan visual objek 3D atau 2D.
Metal Gear Solid V	Game stealth 3D yang menjadi referensi sistem penglihatan dinamis dengan Raycasting.
Monaco: What's Yours Is Mine	Game stealth 2D yang menjadi referensi sistem Fog of War dan Raycasting radial.
MovePosition	Metode di Unity untuk memindahkan objek secara langsung melalui Rigidbody.
NPC	Non-Player Character; karakter dalam game yang tidak dikendalikan oleh pemain.
Null	Nilai kosong menandakan bahwa suatu variabel tidak merujuk ke objek apa pun.
OnDrawGizmosSelected	Fungsi Unity untuk menggambar gizmos hanya saat objek dipilih di editor.
OnTriggerEnter2D	Fungsi event yang dipanggil saat objek memasuki area trigger collider.
OnTriggerExit2D	Fungsi event yang dipanggil saat objek keluar dari area trigger collider.
OnTriggerStay2D	Fungsi event yang dipanggil selama objek masih berada di dalam area trigger collider.

Parent	Objek induk dalam hierarki Unity; objek anak akan bergerak mengikuti induknya.
Ping-Pong	Pola patroli bolak-balik; musuh bergerak maju lalu mundur di jalur yang sama.
Polygon Mesh	Kumpulan titik yang membentuk area visual, digunakan untuk efek Fog of War.
PolygonCollider2D	Collider 2D berbentuk poligon bebas, tidak hanya persegi; digunakan sebagai area pandang musuh.
Pre-production	Tahap awal pengembangan game, meliputi perencanaan, riset, dan desain konsep.
Production	Tahap produksi inti, tempat aset dibuat, kode ditulis, dan game dirakit.
Programmer	Orang yang menulis kode dan logika permainan.
Raycast	Metode deteksi yang menembakkan sinar tak terbatas ke satu arah untuk mengecek tabrakan.
RaycastHit2D	Struktur data yang menyimpan informasi hasil tabrakan dari Raycast atau Linecast.
Real-time	Waktu nyata; sistem yang langsung merespons input tanpa jeda berarti.
Rigidbody 2D	Komponen fisika Unity untuk objek 2D yang memungkinkan pergerakan dan deteksi tabrakan.
Runtime	Saat game sedang berjalan atau dijalankan.
Scene	Area atau level dalam game di Unity.
Scene View	Jendela editor Unity untuk melihat dan mengedit tata letak scene secara visual.
SetFloat	Perintah untuk mengirim nilai numerik ke Animator sebagai parameter.

Skala Likert	Skala penilaian untuk mengukur sikap atau pendapat, misalnya dari "Sangat Tidak Setuju" hingga "Sangat Setuju".
Software	Perangkat lunak atau program komputer.
Spot Light	Sumber cahaya yang menyorot ke satu arah tertentu.
Sprite	Gambar 2D yang digunakan sebagai tampilan visual karakter atau objek.
Sprite Renderer	Komponen Unity untuk menampilkan sprite di layar.
Standard Shader	Pengaturan material bawaan Unity untuk merender tampilan objek.
State	Status atau kondisi perilaku karakter, misalnya Patrol, Chase, atau Idle.
Stealth Game	Genre game yang menekankan pada penghindaran, bersembunyi, dan strategi, bukan pertarungan langsung.
Tag	Penanda khusus pada objek di Unity untuk memudahkan identifikasi, misalnya "Player" atau "Wall".
Testing	Tahap pengujian game untuk menemukan bug dan mengevaluasi kualitas.
Tile Palette	Alat di Unity untuk melukis tilemap dengan cepat.
Tilemap	Sistem untuk membangun level berbasis kotak-kotak (ubin) di Unity.
Tilemap Collider 2D	Komponen yang otomatis menambahkan collider pada setiap ubin di tilemap.
Time.fixedDeltaTime	Nilai waktu tetap yang digunakan dalam FixedUpdate untuk menjaga konsistensi fisika.
Top-Down	Perspektif game dari atas ke bawah, seperti melihat peta dari langit.

Transform	Komponen yang menyimpan posisi, rotasi, dan skala objek di Unity.
Trigger	Collider yang tidak menyebabkan tabrakan fisik, hanya berfungsi sebagai sensor area.
Universal Render Pipeline (URP)	Rangkaian render di Unity untuk grafis yang lebih fleksibel dan efisien.
Update	Fungsi di Unity yang berjalan setiap frame.
Vector	Besaran matematika yang memiliki nilai dan arah; digunakan untuk pergerakan karakter.
Velocity	Kecepatan dan arah gerak suatu objek.
Visual Studio Code	Aplikasi editor kode sumber yang ringan untuk menulis program, termasuk skrip Unity.
Waypoint	Titik-titik dalam rute patroli yang harus dikunjungi musuh.
White Box Testing	Metode pengujian perangkat lunak yang melihat struktur internal kode dan logika program.
2D	Dua dimensi; grafis yang hanya memiliki panjang dan lebar, tanpa kedalaman.
AI	Artificial Intelligence; kecerdasan buatan yang membuat musuh atau karakter dalam game dapat berpikir dan bereaksi seperti makhluk hidup.
Alpha Testing	Tahap pengujian awal yang dilakukan oleh pengembang sendiri untuk mencari bug dan memastikan fitur berjalan sesuai rencana.
Animator	Komponen di Unity yang mengatur pergantian animasi karakter berdasarkan kondisi tertentu.
Animator Controller	File di Unity yang menyimpan grafik hubungan antar status animasi dan transisinya.

Behavior Tree Struktur logika AI yang mengatur perilaku karakter secara hierarkis, mirip cabang pohon keputusan.

Beta Testing Tahap pengujian lanjutan yang melibatkan pengguna luar atau pemain biasa untuk menilai pengalaman bermain.

Black Box Testing Metode pengujian perangkat lunak yang fokus pada input dan output, tanpa melihat kode program di dalamnya.

Blend Tree Fitur di Unity Animator yang memungkinkan perpaduan animasi, misalnya antara jalan lurus dan diagonal.

Body Type Pengaturan tipe fisik pada komponen Rigidbody, misalnya Dynamic, Kinematic, atau Static.

Brainware Sumber daya manusia atau orang-orang yang terlibat dalam pengembangan sistem.

C# Bahasa pemrograman yang digunakan di Unity untuk menulis logika permainan.

INTISARI

Skripsi ini membahas perancangan dan implementasi sistem penglihatan musuh yang lebih realistis dalam game 2D *The Last Light of the Mother*. Permasalahan utama yang ditemukan adalah musuh dapat mendeteksi pemain meskipun terhalang oleh dinding, sehingga mengurangi pengalaman *stealth* dan membuat permainan terasa tidak adil. Untuk mengatasi masalah tersebut, penelitian ini mengimplementasikan teknik *Linecasting*—sebuah metode deteksi garis pandang yang memastikan musuh hanya dapat melihat pemain jika tidak ada objek penghalang seperti tembok di antara mereka.

Metode penelitian menggunakan *Game Development Life Cycle (GDLC)* dengan fokus pada fase produksi dan pengujian. Sistem ini dirancang secara modular, di mana logika penglihatan musuh dipisahkan menjadi komponen sensor, kecerdasan buatan, dan animasi. Pengujian dilakukan melalui dua tahap yakni *Alpha Testing* untuk memastikan fungsionalitas teknis, dan *Beta Testing* untuk menilai pengalaman pengguna.

Hasil penelitian menunjukkan bahwa penerapan *Linecasting* berhasil mengatasi masalah deteksi yang menembus dinding. Musuh sekarang hanya bereaksi ketika benar-benar melihat pemain, dan sistem memberikan umpan balik visual berupa ikon status yang informatif.

Kata kunci: Linecasting, game 2D, kecerdasan buatan, stealth game, sistem penglihatan musuh.

ABSTRACT

*This thesis discusses the design and implementation of a more realistic enemy vision system in the 2D game *The Last Light of the Mother*. The main problem identified was that enemies could detect the player even when hidden behind walls, which reduced the stealth experience and made the game feel unfair. To solve this issue, this research implements Linecasting—a line-of-sight detection method that ensures enemies can only see the player if there are no blocking objects such as walls between them.*

The research method uses the Game Development Life Cycle (GDLC) with a focus on the production and testing phases. The system is designed modular, where the enemy's vision logic is separated into sensor, artificial intelligence, and animation components. Testing was conducted in two stages: Alpha Testing to ensure technical functionality, and Beta Testing to evaluate user experience.

The results show that the implementation of Linecasting successfully solved the wall-penetrating detection problem. Enemies now only react when they actually see the player, and the system provides informative visual feedback through status icons.

Keyword: *Linecasting, 2D game, Artificial Intelligence, stealth game, enemy vision system.*