

**PENERAPAN ALGORITMA A-STAR (A*) PADA
KECERDASAN MUSUH DALAM GAME ANDROID 2D
BERBASIS CERITA SI KANCIL MENGGUNAKAN UNITY**

TUGAS AKHIR



Oleh:

**FARIZ IRVANSYAH
NIM : 2211500141**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2026**

**PENERAPAN ALGORITMA A-STAR (A*) PADA
KECERDASAN MUSUH DALAM GAME ANDROID 2D
BERBASIS CERITA SI KANCIL MENGGUNAKAN UNITY**

TUGAS AKHIR



Oleh:

**FARIZ IRVANSYAH
NIM : 2211500141**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2026**

**PENERAPAN ALGORITMA A-STAR (A*) PADA
KECERDASAN MUSUH DALAM GAME ANDROID 2D
BERBASIS CERITA SI KANCIL MENGGUNAKAN UNITY**

**Diajukan untuk memenuhi salah satu persyaratan memperoleh
gelar Sarjana Komputer (S.Kom)**

TUGAS AKHIR



Oleh:

**FARIZ IRVANSYAH
NIM : 2211500141**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2026**



LEMBAR PENGESAHAN



Nama : Fariz Irvansyah
Nomor Induk Mahasiswa : 2211500141
Program Studi : Teknik Informatika
Bidang Peminatan : Programming Expert
Jenjang Studi : Strata 1
Judul : PENERAPAN ALGORITMA A-STAR (A*) PADA KECERDASAN
MUSUH DALAM GAME ANDROID 2D BERBASIS CERITA SI KANCIL
MENGUNAKAN UNITY

Laporan Tugas Akhir ini telah disetujui, disahkan dan direkam secara elektronik sehingga tidak memerlukan tanda tangan tim penguji.

Jakarta, Rabu 28 Januari 2026

Tim Penguji:

Ketua : Joko Christian Chandra, S.Kom., M.Kom.
Anggota : Painem, S.Kom., M.Kom.
Pembimbing : Purwanto, S.Si, M.Kom.
Ketua Program Studi : Dr. Indra, S.Kom., M.T.I.

ABSTRAK

PENERAPAN ALGORITMA A-STAR (A*) PADA KECERDASAN MUSUH DALAM GAME ANDROID 2D BERBASIS CERITA SI KANCIL MENGGUNAKAN UNITY

Oleh : Fariz Irvansyah (2211500141)

Perkembangan teknologi *mobile* mendorong pertumbuhan industri *game* Android sebagai media hiburan sekaligus edukasi. Penelitian ini mengangkat permasalahan belum optimalnya penerapan kecerdasan buatan pada karakter musuh dalam *game* edukasi berbasis budaya lokal. Tujuan penelitian adalah mengembangkan *game* Android 2D bergenre *escape* labirin yang menerapkan algoritma A-Star (A*) sebagai kecerdasan buatan musuh dengan mengintegrasikan cerita rakyat Si Kancil sebagai elemen naratif utama. Metode pengembangan menggunakan *Game Development Life Cycle* (GDLC) yang meliputi tahap *Initiation*, *Pre-Production*, *Production*, *Testing*, *Beta*, dan *Release*. *Game* dikembangkan menggunakan *Unity Engine* dengan bahasa pemrograman C# dan terdiri dari tiga level permainan dengan tingkat kesulitan bertahap. Algoritma A* diimplementasikan untuk mengatur pergerakan musuh secara dinamis dalam mengejar pemain melalui jalur terpendek dengan mempertimbangkan struktur labirin berbasis *grid* dan fungsi *heuristik Manhattan Distance*. Hasil pengujian menunjukkan algoritma A* berhasil diterapkan dengan responsif terhadap perubahan posisi pemain dan mampu menghindari rintangan secara optimal. Pengujian fungsional *black box* menunjukkan seluruh fitur *game* berjalan sesuai rancangan. Pengujian pengguna terhadap 16 responden menggunakan skala *Likert* menghasilkan tingkat kelayakan 95,88% dengan kategori "Sangat Layak", menunjukkan *game* menarik, mudah dipahami, dan berhasil memperkenalkan cerita Si Kancil. Penelitian ini memberikan kontribusi dalam pengembangan *game* edukasi berbasis budaya lokal dengan penerapan kecerdasan buatan yang efektif dan dapat menjadi referensi pengembangan *game* serupa di masa mendatang.

Kata Kunci : Algoritma A-Star, Kecerdasan Buatan, *Game* Android 2D, Cerita Rakyat Si Kancil, *Unity Engine*, *Pathfinding*, *Game* Edukasi

xviii+132 halaman; 70 gambar; 18 tabel; 2 lampiran

SURAT PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan dibawah ini :

Nama : Fariz Irvansyah
NIM : 2211500141
Program Studi : Teknik Informatika
Bidang Peminatan : Programming Expert
Jenjang Studi : Strata 1
Fakultas : Teknologi Informasi

menyatakan bahwa TUGAS AKHIR yang berjudul:

PENERAPAN ALGORITMA A-STAR (A*) PADA KECERDASAN MUSUH DALAM
GAME ANDROID 2D BERBASIS CERITA SI KANCIL MENGGUNAKAN UNITY

Merupakan :

1. Karya tulis saya sebagai laporan tugas akhir yang asli dan belum pernah diajukan untuk mendapatkan gelar akademik apa-pun, baik di Universitas Budi Luhur maupun di perguruan tinggi lainnya.
2. Karya tulis ini bukan saduran / terjemahan, dan murni gagasan, rumusan dan pelaksanaan penelitian / implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan pembimbing di organisasi tempat riset.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Saya menyerahkan hak milik atas karya tulis ini kepada Universitas Budi Luhur, dan oleh karenanya Universitas Budi Luhur berhak melakukan pengelolaan atas karya tulis ini sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh berdasarkan karya tulis ini, serta sanksi lainnya sesuai dengan norma di Universitas Budi Luhur dan Undang-Undang yang berlaku.

Jakarta, 28 Januari 2026



Fariz Irvansyah

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayat-Nya hingga penulis dapat menyelesaikan tugas akhir dengan judul “Penerapan Algoritma A-Star (A*) Pada Kecerdasan Musuh Dalam Game Android 2D Berbasis Cerita Si Kancil Menggunakan Unity”.

Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua Orang Tua yang selalu mendoakan dan mendukung sampai penulis dapat menyelesaikan tugas akhir.
2. Bapak Purwanto, S.Si., M.Kom. selaku Dosen Pembimbing Tugas Akhir yang telah bersedia meluangkan waktu untuk membantu dan membimbing dalam penulisan tugas akhir hingga selesainya tugas akhir ini.
3. Bapak Prof. Agus Setyo Budi, M.Sc. selaku Rektor Universitas Budi Luhur.
4. Bapak Dr. Indra, S.Kom., M.T.I selaku Ketua Program Studi Teknik Informatika Budi Luhur.
5. Seluruh dosen yang telah membekali penulis dengan ilmu pengetahuan dan wawasan selama masa perkuliahan.
6. Teman-teman seperjuangan yang telah berbagi semangat, pengalaman dan bantuan selama penyusunan tugas akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki keterbatasan dan belum sepenuhnya sempurna. Penulis mengucapkan terima kasih kepada seluruh pihak yang telah berkontribusi dalam penyusunan tugas akhir ini. Semoga segala kebaikan yang diberikan mendapat balasan yang setimpal dari Allah SWT.

Akhir kata, penulis berharap penelitian ini menjadi ilmu yang bermanfaat bagi kita semua serta saran dan kritiknya untuk pengembangan permainan ini selanjutnya.

Jakarta, 18 Januari 2026

Penulis

DAFTAR TABEL

Tabel 2. 1 Studi Literatur.....	9
Tabel 3. 1 Tabel Data Responden	17
Tabel 3. 2 Tabel Perancangan Aktor Sistem.....	24
Tabel 3. 3 Tabel Perancangan Level Game	25
Tabel 4. 1 Tabel Pembuatan Desain Karakter	44
Tabel 4. 2 Tabel Pembuatan Desain Labirin.....	45
Tabel 4. 3 Tabel Pembuatan Suara.....	45
Tabel 4. 4 Tabel Pembuatan Mekanisme Permainan	46
Tabel 4. 5 Tabel Aset	47
Tabel 4. 6 Tabel Pengujian Black Box Fungsi Utama Sistem.....	101
Tabel 4. 7 Tabel Skenario Pengujian Algoritma A-Star (A*)	104
Tabel 4. 8 Tabel Hasil Pengujian Black Box Algoritma A-Star (A*).....	105
Tabel 4. 9 Tabel Kelebihan Algoritma A-Star (A*).....	106
Tabel 4. 10 Kekurangan Algoritma A-Star (A*)	107
Tabel 4. 11 Tabel Rentang Nilai Skala Likert	107
Tabel 4. 12 Tabel Hasil Kuesioner Dari Pengguna.....	107
Tabel 4. 13 Tabel Rincian Perhitungan Kuesioner	109
Tabel 4. 14 Tabel Hasil Perhitungan Persentase.....	110

DAFTAR GAMBAR

Gambar 3. 1 Rancangan Menu.....	28
Gambar 3. 2 Rancangan Layar Menu Utama	29
Gambar 3. 3 Rancangan Layar Keluar Permainan.....	29
Gambar 3. 4 Rancangan Layar Menu Pengaturan	30
Gambar 3. 5 Rancangan Layar Credits.....	31
Gambar 3. 6 Rancangan Layar Reset Data	31
Gambar 3. 7 Rancangan Layar Menu Level	32
Gambar 3. 8 Rancangan Layar Prolog Level 1	32
Gambar 3. 9 Rancangan Layar Cara Bermain	33
Gambar 3. 10 Rancangan Layar Level 1	34
Gambar 3. 11 Rancangan Layar Epilog Level 1	34
Gambar 3. 12 Rancangan Layar Prolog Level 2	35
Gambar 3. 13 Rancangan Layar Level 2	35
Gambar 3. 14 Rancangan Layar Epilog Level 2	36
Gambar 3. 15 Rancangan Layar Prolog Level 3	37
Gambar 3. 16 Rancangan Layar Level 3	37
Gambar 3. 17 Rancangan Layar Epilog Level 3	38
Gambar 3. 18 Rancangan Layar Jeda Permainan	39
Gambar 3. 19 Rancangan Layar Menang	39
Gambar 3. 20 Rancangan Layar Kalah.....	40
Gambar 3. 21 Rancangan Layar Tamat.....	41
Gambar 3. 22 Rancangan Layar Kembali Ke Menu Utama	41
Gambar 4. 1 <i>Deployment</i> Diagram.....	43
Gambar 4. 2 Flowchart Menu Utama.....	52
Gambar 4. 3 Flowchart Keluar Permainan	53
Gambar 4. 4 Flowchart Menu Pengaturan.....	54
Gambar 4. 5 Flowchart Credits	55
Gambar 4. 6 Flowchart Reset Data	56
Gambar 4. 7 Flowchart Menu Level	57
Gambar 4. 8 Flowchart Prolog Level 1	58
Gambar 4. 9 Flowchart Cara Bermain.....	59
Gambar 4. 10 Flowchart Level 1.....	60
Gambar 4. 11 Flowchart Epilog Level 1	61
Gambar 4. 12 Flowchart Prolog Level 2	63
Gambar 4. 13 Flowchart Epilog Level 2	67
Gambar 4. 14 Flowchart Prolog Level 3	68
Gambar 4. 15 Flowchart Level 3.....	70
Gambar 4. 16 Flowchart Epilog Level 3	71
Gambar 4. 17 Flowchart Jeda Permainan	73
Gambar 4. 18 Flowchart Menang.....	74
Gambar 4. 19 Flowchart Kalah	75
Gambar 4. 20 Flowchart Tamat	76
Gambar 4. 21 Flowchart Kembali Ke Menu.....	77
Gambar 4. 22 Flowchart Player	79
Gambar 4. 23 Flowchart Node.....	80

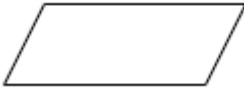
Gambar 4. 24 Flowchart Algoritma A-Star (A*).....	82
Gambar 4. 25 Flowchart Musuh	84
Gambar 4. 26 Pengujian Algoritma A-Star (A*).....	105
Gambar 4. 27 Tampilan Menu Utama.....	111
Gambar 4. 28 Tampilan Keluar Permainan	111
Gambar 4. 29 Tampilan Menu Pengaturan.....	112
Gambar 4. 30 Tampilan <i>Credits</i>	112
Gambar 4. 31 Tampilan <i>Reset Data</i>	113
Gambar 4. 32 Tampilan Menu Level	113
Gambar 4. 33 Tampilan Menu Level 2	114
Gambar 4. 34 Tampilan Prolog Level 1	114
Gambar 4. 35 Tampilan Cara Bermain.....	115
Gambar 4. 36 Tampilan Level 1	115
Gambar 4. 37 Tampilan Epilog Level 1	116
Gambar 4. 38 Tampilan Prolog Level 2	116
Gambar 4. 39 Tampilan Level 2	117
Gambar 4. 40 Tampilan Epilog Level 2	117
Gambar 4. 41 Tampilan Prolog Level 3	118
Gambar 4. 42 Tampilan Level 3	118
Gambar 4. 43 Tampilan Epilog Level 3	119
Gambar 4. 44 Tampilan Jeda Permainan.....	119
Gambar 4. 45 Tampilan Menang	120
Gambar 4. 46 Tampilan Kalah.....	120
Gambar 4. 47 Tampilan Tamat	121
Gambar 4. 48 Tampilan Kembali Ke Menu Utama	121

DAFTAR SIMBOL

Simbol Deployment Diagram

Simbol	Nama	Deskripsi
	<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih <i>node</i> .
	<i>Node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.
	<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
	<i>Link</i>	Relasi antar <i>node</i> .

Simbol Flowchart Diagram

Simbol	Nama	Deskripsi
	<i>Terminator</i>	Permulaan atau akhir program.
	<i>Flow Line</i>	Arah aliran program.
	<i>Process</i>	Proses perhitungan atau proses pengolahan data.
	<i>Input atau Output</i>	Proses <i>input</i> atau <i>output</i> data, parameter, informasi.
	<i>Connector</i>	Penghubung bagian - bagian <i>flowchart</i> yang berada pada satu halaman.
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
	<i>Off Page Connector</i>	Penghubung bagian - bagian <i>flowchart</i> yang berada pada halaman berbeda.

DAFTAR ALGORITME

Algoritme 4. 1 Algoritme Menu Utama	85
Algoritme 4. 2 Algoritme Keluar Permainan.....	85
Algoritme 4. 3 Algoritme Menu Pengaturan	85
Algoritme 4. 4 Algoritma <i>Credits</i>	86
Algoritme 4. 5 Algoritme Reset Data.....	86
Algoritme 4. 6 Algoritme Menu Level.....	87
Algoritme 4. 7 Algoritme Prolog Level 1.....	87
Algoritme 4. 8 Algoritme Cara Bermain.....	88
Algoritme 4. 9 Algoritme Level 1	88
Algoritme 4. 10 Algoritme Epilog Level 1.....	89
Algoritme 4. 11 Algoritme Prolog Level 2.....	89
Algoritme 4. 12 Algoritme Level 2.....	90
Algoritme 4. 13 Algoritme Epilog Level 2.....	91
Algoritme 4. 14 Algoritme Prolog Level 3.....	91
Algoritme 4. 15 Algoritme Level 3.....	91
Algoritme 4. 16 Algoritme Epilog Level 3.....	92
Algoritme 4. 17 Algoritme Jeda Permainan	93
Algoritme 4. 18 Algoritme Menang.....	93
Algoritme 4. 19 Algoritme Kalah	94
Algoritme 4. 20 Algoritme Tamat.....	95
Algoritme 4. 21 Algoritme Kembali Ke Menu Utama.....	96
Algoritme 4. 22 Algoritme Player.....	96
Algoritme 4. 23 Algoritme Node	97
Algoritme 4. 24 Algoritme A-Star (A*).....	98
Algoritme 4. 25 Algoritme Musuh.....	99

DAFTAR ISI

LEMBAR JUDUL.....	ii
HALAMAN PEROLEHAN GELAR	iii
LEMBAR PERSETUJUAN SETELAH SIDANG.....	iv
ABSTRAK.....	v
SURAT PERNYATAAN TIDAK PLAGIAT.....	vi
KATA PENGANTAR.....	vii
DAFTAR TABEL.....	viii
DAFTAR GAMBAR	ix
DAFTAR SIMBOL.....	xi
DAFTAR ALGORITME.....	xii
DAFTAR ISI.....	xiii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan	3
1.5. Manfaat	3
1.6. Sistematika Penulisan	4
BAB II LANDASAN TEORI.....	5
2.1. Konsep Game	5
2.2. Game Dua Dimensi (2D)	5
2.3. Game Berbasis Android.....	5
2.4. Unity Game Engine	6
2.5. Kecerdasan Buatan Dalam Game	6
2.6. Algoritma A* (A-Star).....	7
2.7. Cerita Rakyat Si Kancil	8
2.8. Studi Literatur.....	9
BAB III METODOLOGI PENELITIAN	16
3.1. Data Penelitian.....	16
3.1.1 Data Narasi Cerita Rakyat Petualangan Si Kancil.....	16
3.1.2 Data Sistem Permainan	16
3.1.3 Data Responden	16
3.2. Metode Pembanding	17

3.2.1 Pertimbangan Pemilihan Algoritma A-Star	17
3.2.2 Relevansi Studi Komparatif Dijkstra dan A-Star (A*)	18
3.2.3 Gap Penelitian dan Kontribusi Unik	19
3.2.4 Positioning Penelitian	20
3.3. Metode Penelitian	20
3.3.1 Initiation	21
3.3.2 Pre-Production	21
3.3.3 Production	21
3.3.4 Testing.....	22
3.3.5 Beta	22
3.3.6 Release	22
3.4. Rancangan Pengujian.....	23
3.4.1 Rancangan Pengujian Fungsi Utama Sistem.....	23
3.4.2 Rancangan Pengujian Algoritma A-Star (A*).....	23
3.4.3 Rancangan Pengujian Dari Pengguna	23
3.5. Perancangan Sistem Game	24
3.5.1 Gambaran Umum Sistem Game	24
3.5.2 Perancangan Aktor Sistem	24
3.5.3 Alur Permainan (Game Flow)	25
3.5.4 Perancangan Level Game.....	25
3.5.5 Perancangan Mekanisme Permainan	26
3.6. Perancangan Algoritma A* (A-Star) pada Game	26
3.6.1 Tujuan Penerapan Algoritma A*	26
3.6.2 Representasi Lingkungan Game	26
3.6.3 Penentuan Node Awal dan Node Tujuan	26
3.6.4 Perhitungan Nilai Heuristik.....	27
3.6.5 Alur Kerja Algoritma A-Star (A*) pada Musuh.....	27
3.6.6 Integrasi Algoritma A-Star (A*) dengan Sistem Game	27
3.7. Rancangan Menu	27
3.8. Rancangan Layar	28
3.8.1 Rancangan Layar Menu Utama	28
3.8.2 Rancangan Layar Keluar Permainan.....	29
3.8.3 Rancangan Layar Menu Pengaturan	30
3.8.4 Rancangan Layar Credits	30

3.8.5 Rancangan Layar Reset Data.....	31
3.8.6 Rancangan Layar Menu Level.....	31
3.8.7 Rancangan Layar Prolog Level 1.....	32
3.8.8 Rancangan Layar Cara Bermain.....	33
3.8.9 Rancangan Layar Level 1.....	33
3.8.10 Rancangan Layar Epilog Level 1.....	34
3.8.11 Rancangan Layar Prolog Level 2.....	34
3.8.12 Rancangan Layar Level 2.....	35
3.8.13 Rancangan Layar Epilog Level 2.....	36
3.8.14 Rancangan Layar Prolog Level 3.....	36
3.8.15 Rancangan Layar Level 3.....	37
3.8.16 Rancangan Layar Epilog Level 3.....	37
3.8.17 Rancangan Layar Jeda Permainan	38
3.8.18 Rancangan Layar Menang.....	39
3.8.19 Rancangan Layar Kalah	39
3.8.20 Rancangan Layar Tamat.....	40
3.8.21 Rancangan Layar Kembali Ke Menu Utama.....	41
BAB IV HASIL DAN PEMBAHASAN.....	42
4.1. Lingkungan Percobaan.....	42
4.1.1 Perangkat Keras (Hardware)	42
a. Perangkat Pengembangan.....	42
b. Perangkat Pengujian.....	42
4.1.2 Perangkat Lunak (Software).....	42
4.1.3 <i>Deployment</i> Diagram	43
4.2. Implementasi Metode	44
4.2.1 Initiation	44
4.2.2 Pre-Production	44
a. Pembuatan Desain Karakter	44
b. Pembuatan Desain Labirin	45
c. Pembuatan Suara.....	45
d. Pembuatan Mekanisme Permainan	46
e. Pembuatan Penerapan Algoritma A-Star (A*)	47
4.2.3 Production	47
4.2.4 Testing.....	50

4.2.5 Beta	50
4.2.6 Release	51
4.3. Flowchart.....	51
4.3.1 Flowchart Menu Utama.....	51
4.3.2 Flowchart Keluar Permainan.....	52
4.3.3 Flowchart Menu Pengaturan.....	53
4.3.4 Flowchart Credits.....	54
4.3.5 Flowchart Reset Data	55
4.3.6 Flowchart Menu Level	56
4.3.7 Flowchart Prolog Level 1	57
4.3.8 Flowchart Cara Bermain	58
4.3.9 Flowchart Level 1	59
4.3.10 Flowchart Epilog Level 1	60
4.3.11 Flowchart Prolog Level 2	62
4.3.12 Flowchart Level 2	63
4.3.13 Flowchart Epilog Level 2	65
4.3.14 Flowchart Prolog Level 3	67
4.3.15 Flowchart Level 3	68
4.3.16 Flowchart Epilog Level 3	70
4.3.17 Flowchart Jeda Permainan.....	72
4.3.18 Flowchart Menang	73
4.3.19 Flowchart Kalah.....	74
4.3.20 Flowchart Tamat.....	76
4.3.21 Flowchart Kembali Ke Menu	77
4.3.22 Flowchart Player	77
4.3.23 Flowchart Node.....	80
4.3.24 Flowchart Algoritma A-Star (A*).....	81
4.3.25 Flowchart Musuh	82
4.4. Algoritme	84
4.4.1 Algoritme Menu Utama	84
4.4.2 Algoritme Keluar Permainan.....	85
4.4.3 Algoritme Menu Pengaturan	85
4.4.4 Algoritme <i>Credits</i>	86
4.4.5 Algoritme Reset Data.....	86

4.4.6 Algoritme Menu Level.....	87
4.4.7 Algoritme Prolog Level 1.....	87
4.4.8 Algoritme Cara Bermain	88
4.4.9 Algoritme Level 1	88
4.4.10 Algoritme Epilog Level 1.....	89
4.4.11 Algoritme Prolog Level 2.....	89
4.4.12 Algoritme Level 2	90
4.4.13 Algoritme Epilog Level 2.....	91
4.4.14 Algoritme Prolog Level 3.....	91
4.4.15 Algoritme Level 3	91
4.4.16 Algoritme Epilog Level 3.....	92
4.4.17 Algoritme Jeda Permainan	93
4.4.18 Algoritme Menang	93
4.4.19 Algoritme Kalah	94
4.4.20 Algoritme Tamat.....	95
4.4.21 Algoritme Kembali Ke Menu Utama.....	96
4.4.22 Algoritme Player.....	96
4.4.23 Algoritme Node	97
4.4.24 Algoritme A-Star (A*)	97
4.4.25 Algoritme Musuh	99
4.5. Pengujian.....	101
4.5.1 Pengujian Fungsi Utama Sistem.....	101
4.5.2 Pengujian Algoritma A-Star (A*).....	104
4.5.3 Analisis Kelebihan Dan Kekurangan Algoritma A-Star (A*).....	106
a. Kelebihan Algoritma A-Star (A*).....	106
b. Kekurangan Algoritma A-Star (A*)	106
4.5.4 Pengujian Dari Pengguna	107
4.6. Tampilan Layar	110
4.6.1 Tampilan Layar Menu Utama.....	111
4.6.2 Tampilan Layar Keluar Permainan.....	111
4.6.3 Tampilan Layar Menu Pengaturan	111
4.6.4 Tampilan Layar <i>Credits</i>	112
4.6.5 Tampilan Layar Reset Data	112
4.6.6 Tampilan Layar Menu Level.....	113

4.6.7 Tampilan Layar Prolog Level 1	114
4.6.8 Tampilan Layar Cara Bermain	114
4.6.9 Tampilan Layar Level 1	115
4.6.10 Tampilan Layar Epilog Level 1	115
4.6.11 Tampilan Layar Prolog Level 2	116
4.6.12 Tampilan Layar Level 2	116
4.6.13 Tampilan Layar Epilog Level 2	117
4.6.14 Tampilan Layar Prolog Level 3	117
4.6.15 Tampilan Layar Level 3	118
4.6.16 Tampilan Layar Epilog Level 3	118
4.6.17 Tampilan Layar Jeda Permainan.....	119
4.6.18 Tampilan Layar Menang	119
4.6.19 Tampilan Layar Kalah.....	120
4.6.20 Tampilan Layar Tamat.....	120
4.6.21 Tampilan Layar Kembali Ke Menu Utama	121
BAB V PENUTUP.....	122
5.1. Kesimpulan.....	122
5.2. Saran	122
DAFTAR PUSTAKA.....	124
SURAT KETERANGAN RISET	126
LEMBAR KUESIONER.....	127

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi perangkat bergerak (*mobile*) khususnya pada sistem operasi Android telah mendorong pertumbuhan industri *game* secara signifikan. *Game* tidak hanya berfungsi sebagai media hiburan, tetapi juga dapat dimanfaatkan sebagai sarana edukasi, simulasi, serta media interaktif yang mampu melatih kemampuan berpikir dan pengambilan keputusan pemain. Seiring dengan meningkatnya penggunaan *smartphone* di masyarakat, pengembangan *game* berbasis Android menjadi salah satu platform yang relevan dan mudah diakses oleh berbagai kalangan.

Dalam pengembangan *game*, penerapan kecerdasan buatan (*Artificial Intelligence*) memegang peranan penting, khususnya dalam mengatur perilaku musuh (*enemy*) agar mampu memberikan tantangan yang dinamis dan tidak mudah diprediksi oleh pemain. Salah satu algoritma yang umum digunakan dalam pengembangan *game* untuk menentukan jalur pergerakan musuh adalah algoritma A* (A-Star). Algoritma A* dikenal sebagai algoritma pencarian jalur terpendek yang efisien karena mengombinasikan biaya aktual dari titik awal ke suatu *node* ($g(n)$) dengan estimasi biaya menuju tujuan ($h(n)$), sehingga menghasilkan jalur yang optimal. Oleh karena itu, algoritma A* banyak diterapkan dalam *game* bergenre labirin, strategi, maupun petualangan.

Berdasarkan beberapa penelitian sebelumnya, penerapan algoritma A* (A-Star) dalam pengembangan *game* telah banyak dilakukan, khususnya pada *game* dua dimensi (2D) dan *game* berbasis perangkat bergerak. Algoritma A* dipilih karena kemampuannya dalam menentukan jalur terpendek secara efisien pada lingkungan berbasis *grid*, sehingga sangat sesuai diterapkan pada *game* dengan peta labirin atau area terbatas.

Penerapan algoritma A* dalam pengembangan *game mobile* menggunakan *Unity Engine*. Penelitian tersebut berfokus pada pemanfaatan algoritma A* untuk meningkatkan kecerdasan pergerakan karakter dalam *game* agar mampu mencari jalur secara optimal menuju target. *Unity Engine* dipilih karena memiliki fleksibilitas tinggi, dukungan yang luas untuk pengembangan *game mobile*, serta kemudahan integrasi antara logika permainan dan kecerdasan buatan. Hasil penelitian menunjukkan bahwa algoritma A* mampu memberikan perilaku pergerakan yang lebih responsif dan dinamis dibandingkan dengan pergerakan konvensional (Oktavianto & Rizqi, 2024).

Selain itu, beberapa penelitian lain pada periode yang sama juga mengkaji implementasi algoritma A* pada *game* labirin dan *game* petualangan dengan tujuan meningkatkan tingkat tantangan dan realisme perilaku musuh. Fokus utama penelitian-penelitian tersebut umumnya terletak pada aspek teknis, seperti efisiensi pencarian jalur, optimasi jumlah *node* yang diproses, serta pengaruh algoritma terhadap performa permainan. Dengan demikian, keberhasilan algoritma A* diukur berdasarkan ketepatan jalur dan kecepatan perhitungan dalam sistem permainan.

Meskipun demikian, penelitian-penelitian tersebut masih memiliki keterbatasan dalam hal konten dan pendekatan naratif. Sebagian besar *game* yang dikembangkan menggunakan latar cerita yang bersifat umum atau fiktif, tanpa mengangkat unsur kearifan lokal atau budaya Indonesia sebagai elemen utama permainan. Cerita dalam *game* sering kali hanya berfungsi sebagai latar pendukung, bukan sebagai bagian integral yang membentuk alur permainan secara berkelanjutan dari awal hingga akhir.

Kondisi tersebut menunjukkan adanya peluang dan kebutuhan akan inovasi penelitian yang tidak hanya berfokus pada penerapan algoritma A* sebagai kecerdasan buatan musuh, tetapi juga mengintegrasikannya dengan unsur budaya lokal Indonesia. Pengintegrasian cerita rakyat sebagai narasi utama dalam *game* diharapkan dapat memberikan nilai tambah, baik dari sisi edukatif maupun pelestarian budaya, serta membedakan penelitian ini dari penelitian-penelitian sebelumnya.

Indonesia memiliki beragam cerita rakyat yang sarat akan nilai moral dan budaya, salah satunya adalah kisah Petualangan Si Kancil. Cerita ini menggambarkan tokoh Kancil sebagai hewan yang cerdik, cerdas, dan mampu menghadapi berbagai tantangan dengan kecerdasannya. Nilai-nilai tersebut relevan untuk diadaptasi ke dalam *game* petualangan dan labirin yang menuntut pemain untuk berpikir strategis dalam menyelesaikan tantangan. Melalui penggabungan algoritma A* sebagai mekanisme kecerdasan buatan dengan narasi cerita rakyat sebagai elemen utama permainan, *game* yang dikembangkan tidak hanya berfungsi sebagai media hiburan, tetapi juga sebagai sarana pengenalan dan pelestarian budaya lokal Indonesia dalam bentuk media digital yang modern dan interaktif.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk merancang dan membangun sebuah *game* Android 2D bergenre *escape* labirin yang mengangkat cerita rakyat Si Kancil dengan menerapkan algoritma A* (A-Star) sebagai dasar kecerdasan buatan musuh. Game dikembangkan menggunakan *game engine* Unity dengan bahasa pemrograman C#, yang mendukung pengembangan *game* 2D secara efektif. Melalui penerapan algoritma A* pada perilaku musuh, diharapkan *game* yang dikembangkan mampu memberikan tantangan yang adaptif sekaligus menjadi media hiburan yang mengedukasi serta memperkenalkan kembali cerita rakyat Indonesia dalam bentuk yang lebih modern dan interaktif.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini dapat dirumuskan sebagai berikut:

- a. Bagaimana menerapkan algoritma A* (A-Star) sebagai kecerdasan buatan musuh dalam *game* Android 2D bergenre *escape* labirin menggunakan Unity?
- b. Bagaimana algoritma A* dapat digunakan untuk menentukan jalur pergerakan musuh secara optimal dalam lingkungan labirin berbasis *grid*?
- c. Bagaimana perilaku musuh yang menerapkan algoritma A* dalam merespons pergerakan pemain pada setiap level permainan?

- d. Bagaimana mengintegrasikan cerita rakyat Si Kancil sebagai elemen naratif utama dalam *game* Android 2D tanpa menghilangkan fokus penelitian pada penerapan algoritma A*?

1.3. Batasan Masalah

Agar pembahasan dalam penelitian ini lebih terarah dan tidak menyimpang dari tujuan yang telah ditetapkan, maka diperlukan batasan masalah sebagai berikut:

- a. *Game* yang dikembangkan merupakan *game* dua dimensi (2D) bergenre *escape* labirin.
- b. *Game* dikembangkan untuk platform Android.
- c. Proses pengembangan *game* menggunakan *game engine* *Unity* dengan bahasa pemrograman C#.
- d. Algoritma yang digunakan sebagai kecerdasan buatan musuh adalah algoritma A* (A-Star).
- e. Penerapan algoritma A* difokuskan pada penentuan jalur pergerakan musuh dalam lingkungan labirin berbasis *grid*.
- f. *Game* terdiri dari tiga level permainan.
- g. Unsur cerita rakyat yang digunakan dalam *game* adalah cerita Petualangan Si Kancil.
- h. Penelitian ini tidak membahas perbandingan algoritma A* dengan algoritma pencarian jalur lainnya.
- i. Penelitian ini tidak membahas optimasi performa algoritma secara mendalam pada perangkat dengan spesifikasi tertentu.

1.4. Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

- a. Menerapkan algoritma A* (A-Star) sebagai kecerdasan buatan musuh dalam *game* Android 2D bergenre *escape* labirin menggunakan *Unity*.
- b. Merancang dan membangun mekanisme pergerakan musuh yang mampu menentukan jalur terpendek secara optimal dalam lingkungan labirin berbasis *grid*.
- c. Mengimplementasikan algoritma A* agar musuh dapat merespons pergerakan pemain secara dinamis pada setiap level permainan.
- d. Mengintegrasikan cerita rakyat Petualangan Si Kancil sebagai elemen naratif utama dalam *game* Android 2D sehingga dapat meningkatkan nilai edukatif dan budaya pada permainan yang dikembangkan.

1.5. Manfaat

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

- a. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan ilmu pengetahuan di bidang teknologi informasi, khususnya pada pengembangan *game* dan penerapan kecerdasan buatan. Hasil penelitian ini dapat dijadikan sebagai referensi bagi mahasiswa atau peneliti selanjutnya yang ingin mengkaji penerapan algoritma A* (A-

- Star) pada *game* Android 2D atau penelitian serupa yang mengintegrasikan algoritma kecerdasan buatan dengan unsur naratif.
- b. Penelitian ini diharapkan dapat menghasilkan sebuah *game* Android 2D bergenre *escape* labirin yang menerapkan algoritma A* sebagai kecerdasan buatan musuh. *Game* yang dikembangkan dapat memberikan pengalaman bermain yang lebih menantang dan dinamis bagi pemain, serta dapat dimanfaatkan sebagai contoh implementasi algoritma A* dalam pengembangan *game* menggunakan *Unity*.
 - c. Melalui pengintegrasian cerita rakyat Petualangan Si Kancil ke dalam *game*, penelitian ini diharapkan dapat berkontribusi dalam upaya pelestarian dan pengenalan budaya lokal Indonesia. Penyajian cerita rakyat dalam bentuk *game* digital diharapkan mampu menarik minat generasi muda untuk mengenal dan memahami nilai-nilai budaya yang terkandung dalam cerita rakyat Indonesia.

1.6. Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran secara umum mengenai isi pembahasan pada setiap bab, sehingga memudahkan pembaca dalam memahami alur penelitian secara sistematis dan logis. Penulisan tugas akhir ini terdiri dari lima bab utama yang saling berkaitan, dengan uraian sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan yang menjadi dasar dan arah dari penelitian yang dilakukan.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori pendukung yang berkaitan dengan penelitian, meliputi konsep *game*, *game* 2D, *game* Android, *Unity* sebagai *game engine*, kecerdasan buatan dalam *game*, algoritma A* (A-Star), serta kajian mengenai cerita rakyat Si Kancil dan penelitian-penelitian terdahulu yang relevan.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan dalam pengembangan *game*, tahapan pengembangan sistem, perancangan *game*, perancangan level dan karakter, serta perancangan penerapan algoritma A* sebagai kecerdasan buatan musuh.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas hasil implementasi dari perancangan yang telah dilakukan, meliputi implementasi *game* Android 2D, penerapan algoritma A* pada musuh, serta pengujian fungsional sistem untuk mengetahui apakah *game* berjalan sesuai dengan tujuan penelitian.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian serta saran yang dapat dijadikan sebagai bahan pengembangan dan penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1. Konsep Game

Game merupakan suatu bentuk aktivitas interaktif yang dirancang untuk memberikan hiburan kepada pengguna melalui aturan, tujuan, tantangan, dan mekanisme tertentu. Dalam konteks teknologi informasi, *game* dikembangkan sebagai perangkat lunak yang memungkinkan pemain untuk berinteraksi dengan sistem secara *real time*. *Game* umumnya memiliki elemen utama berupa pemain (*player*), tujuan (*goal*), aturan permainan, tantangan, serta sistem umpan balik yang memberikan respons terhadap tindakan pemain.

Pada umumnya *game* atau permainan ditujukan hanya sebagai hiburan. Akan tetapi, seiring dengan perkembangan zaman keberadaan *game* mulai dijadikan sebagai satu dari beberapa upaya yang bisa mendukung dalam memupuk motivasi atau semangat dalam belajar karena hal-hal menarik yang ada di dalamnya (Ai Tuti Kusmiati et al., 2024).

Perkembangan game digital tidak hanya terbatas pada aspek hiburan, tetapi juga mencakup aspek edukasi, simulasi, dan pelatihan. *Game* yang dirancang dengan baik mampu melatih kemampuan berpikir logis, pengambilan keputusan, serta strategi pemain dalam menyelesaikan permasalahan yang disajikan di dalam permainan.

2.2. Game Dua Dimensi (2D)

Game dua dimensi (2D) merupakan jenis *game* yang menampilkan objek permainan pada bidang dua dimensi dengan sumbu koordinat horizontal (x) dan vertikal (y). Pada *game* 2D, karakter, lingkungan, dan objek lainnya direpresentasikan dalam bentuk sprite atau gambar datar tanpa kedalaman ruang (*depth*) seperti pada *game* tiga dimensi (3D). Penggunaan *game* 2D memiliki beberapa kelebihan, antara lain kebutuhan sumber daya perangkat yang lebih rendah (Saputra & Setiawan, 2024). Proses pengembangan yang relatif lebih sederhana, serta kompatibilitas yang lebih baik pada berbagai jenis perangkat, khususnya perangkat *mobile*. Oleh karena itu, *game* 2D masih banyak digunakan dalam pengembangan *game* Android, terutama untuk genre *platformer*, petualangan, dan labirin.

2.3. Game Berbasis Android

Android merupakan sistem operasi berbasis Linux yang banyak digunakan pada perangkat *mobile* seperti *smartphone* dan tablet. Popularitas Android didukung oleh sifatnya yang terbuka (*open source*), ketersediaan perangkat dengan berbagai spesifikasi, serta dukungan ekosistem pengembangan aplikasi yang luas. *Game* berbasis Android dikembangkan dengan mempertimbangkan keterbatasan perangkat *mobile*, seperti ukuran layar, kapasitas memori, dan kemampuan pemrosesan (Octavian & Hermawan, 2023). Oleh karena itu, pengembangan *game* Android memerlukan desain yang efisien baik dari sisi grafis maupun logika permainan agar *game* dapat berjalan dengan baik pada berbagai perangkat.

2.4. Unity Game Engine

Unity merupakan salah satu *game engine* yang banyak digunakan dalam pengembangan game, baik untuk platform desktop, web, maupun *mobile*. *Unity* menyediakan lingkungan pengembangan terintegrasi (*Integrated Development Environment*) yang mendukung pembuatan *game* dua dimensi (2D) dan tiga dimensi (3D). *Unity* menggunakan bahasa pemrograman C# sebagai bahasa utama dalam pengembangan logika permainan. Kelebihan *Unity* antara lain kemudahan penggunaan, dukungan aset yang luas, serta kemampuan untuk mengintegrasikan berbagai fitur seperti animasi, fisika, dan kecerdasan buatan. Dengan dukungan *multiplatform*, *Unity* memungkinkan pengembang untuk membangun *game* Android secara lebih efisien dan terstruktur (Agung et al., 2022).

2.5. Kecerdasan Buatan Dalam Game

Kecerdasan Buatan (*Artificial Intelligence* atau AI) dalam *game* merupakan suatu teknik yang digunakan untuk mengatur perilaku karakter non-pemain (*Non-Player Character/NPC*) agar dapat bertindak secara otomatis dan menyerupai perilaku manusia. AI dalam *game* tidak bertujuan untuk menciptakan kecerdasan seperti manusia secara utuh, melainkan untuk memberikan ilusi kecerdasan sehingga permainan terasa lebih hidup, menantang, dan dinamis bagi pemain (Padila et al., 2023).

Penerapan AI dalam *game* mencakup berbagai aspek, seperti pengambilan keputusan, pergerakan karakter, penentuan strategi, hingga respons terhadap tindakan pemain. Dengan adanya AI, NPC dapat bereaksi terhadap kondisi lingkungan permainan, menyesuaikan perilakunya dengan situasi tertentu, serta memberikan tantangan yang tidak monoton (Joseph et al., 2021).

Salah satu bentuk penerapan kecerdasan buatan yang paling umum dalam *game* adalah pada sistem pergerakan musuh. Musuh yang dikendalikan oleh AI diharapkan mampu bergerak secara logis, mencari pemain, menghindari rintangan, serta menentukan jalur yang efisien untuk mencapai target. Tanpa penerapan AI, pergerakan musuh cenderung bersifat statis dan mudah ditebak, sehingga mengurangi tingkat kesulitan dan daya tarik permainan.

Dalam *game* bergenre labirin dan petualangan, kecerdasan buatan sering diterapkan dalam bentuk algoritma pencarian jalur (*pathfinding*). Algoritma *pathfinding* memungkinkan karakter musuh untuk menentukan rute terbaik dari posisi awal menuju posisi target dengan mempertimbangkan rintangan yang ada pada peta permainan. Lingkungan permainan biasanya direpresentasikan dalam bentuk *grid* atau *node*, sehingga memudahkan proses perhitungan jalur.

Beberapa algoritma *pathfinding* yang umum digunakan dalam pengembangan *game* antara lain algoritma *Breadth First Search (BFS)*, *Dijkstra*, dan *A** (*A-Star*) (Muhammad Reyhandi Akbar et al., 2024). Di antara algoritma tersebut, algoritma *A** banyak dipilih karena mampu menghasilkan jalur terpendek secara optimal dengan waktu komputasi yang relatif efisien. Oleh karena itu, algoritma *A** sering digunakan sebagai solusi kecerdasan buatan pada musuh dalam *game* 2D berbasis labirin.

Dalam penelitian ini, kecerdasan buatan difokuskan pada penerapan algoritma A* (A-Star) untuk mengatur pergerakan musuh agar dapat mengejar pemain secara optimal di dalam lingkungan labirin. Penerapan AI ini diharapkan mampu meningkatkan tingkat tantangan permainan serta memberikan pengalaman bermain yang lebih interaktif dan realistis bagi pemain.

2.6. Algoritma A* (A-Star)

Algoritma A* (A-Star) merupakan salah satu algoritma pencarian jalur (*pathfinding*) yang paling banyak digunakan dalam pengembangan *game* dan sistem kecerdasan buatan (Alif Pratama et al., 2025). Algoritma ini dikembangkan sebagai pengembangan dari algoritma *Dijkstra* dengan menambahkan fungsi *heuristik* untuk mempercepat proses pencarian jalur menuju tujuan. Tujuan utama algoritma A* adalah menemukan jalur terpendek dan paling optimal dari suatu titik awal (*start node*) menuju titik tujuan (*goal node*) dengan mempertimbangkan rintangan yang ada di lingkungan (Yohanes & Rochmawati, 2022).

Algoritma A* bekerja dengan mengevaluasi setiap *node* berdasarkan nilai fungsi evaluasi yang disebut sebagai fungsi $f(n)$. Fungsi ini merupakan hasil penjumlahan dari dua komponen utama, yaitu biaya aktual dari titik awal ke *node* saat ini ($g(n)$) dan estimasi biaya dari *node* saat ini menuju tujuan ($h(n)$). Secara matematis, fungsi A* dapat dirumuskan sebagai berikut:

$$f(n) = g(n) + h(n) \quad (2.1)$$

Nilai $g(n)$ menyatakan jarak atau biaya nyata yang telah ditempuh dari *node* awal menuju *node* ke- n , sedangkan $h(n)$ merupakan nilai *heuristik* yang memperkirakan jarak dari *node* ke- n menuju *node* tujuan. Dengan mengombinasikan kedua nilai tersebut, algoritma A* dapat menentukan *node* mana yang paling menjanjikan untuk dieksplorasi terlebih dahulu.

Dalam implementasinya, algoritma A* menggunakan dua buah daftar utama, yaitu *open list* dan *closed list*. *Open list* berisi *node-node* yang telah ditemukan namun belum dievaluasi, sedangkan *closed list* berisi *node-node* yang telah dievaluasi dan tidak akan diproses kembali. Proses pencarian jalur dimulai dengan memasukkan *node* awal ke dalam *open list*, kemudian secara berulang memilih *node* dengan nilai $f(n)$ terkecil untuk dievaluasi hingga *node* tujuan ditemukan (Setiawan et al., 2023).

Lingkungan permainan pada *game* labirin umumnya direpresentasikan dalam bentuk *grid* dua dimensi, di mana setiap sel atau tile dianggap sebagai sebuah *node*. *Node-node* tersebut dapat bersifat dapat dilewati (*walkable*) atau tidak dapat dilewati (*unwalkable*) tergantung pada keberadaan dinding atau rintangan. Pada setiap proses evaluasi, algoritma A* akan memeriksa *node* tetangga (atas, bawah, kiri, dan kanan) dari *node* yang sedang diproses untuk menentukan kemungkinan jalur yang dapat dilalui.

Keunggulan utama algoritma A* terletak pada efisiensi dan optimalisasi jalur yang dihasilkan. Dengan menggunakan fungsi *heuristik* yang tepat,

algoritma A* mampu mengurangi jumlah *node* yang perlu dievaluasi dibandingkan dengan algoritma pencarian jalur lainnya. Hal ini menjadikan A* sangat cocok diterapkan pada game, khususnya game labirin 2D, di mana perhitungan jalur perlu dilakukan secara cepat agar tidak mengganggu performa permainan (Alexander & Sahertian, 2023).

Dalam penelitian ini, algoritma A* digunakan sebagai kecerdasan buatan musuh untuk mengejar pemain di dalam lingkungan labirin. Posisi pemain dijadikan sebagai *node* tujuan (*goal*), sedangkan posisi musuh sebagai *node* awal (*start*). Dengan penerapan algoritma A*, musuh diharapkan mampu menentukan jalur terpendek secara dinamis menuju pemain dengan mempertimbangkan struktur labirin yang ada, sehingga meningkatkan tingkat tantangan dan realisme permainan.

Penerapan algoritma A* pada game Android 2D juga memiliki kelebihan dari sisi implementasi, karena struktur *grid* pada *Tilemap Unity* sangat mendukung proses *pathfinding*. Dengan demikian, algoritma A* dapat diintegrasikan secara efektif ke dalam sistem permainan menggunakan bahasa pemrograman C# (Fallo & Bulu, 2022).

2.7. Cerita Rakyat Si Kancil

Cerita rakyat merupakan salah satu bentuk warisan budaya yang berkembang dan diwariskan secara turun-temurun di masyarakat. Cerita rakyat umumnya mengandung nilai-nilai moral, sosial, dan budaya yang mencerminkan karakter serta kearifan lokal suatu daerah. Dalam konteks budaya Indonesia, cerita rakyat memiliki peran penting sebagai media pembelajaran karakter sekaligus sarana pelestarian identitas budaya bangsa.

Salah satu cerita rakyat yang sangat dikenal di Indonesia adalah cerita Petualangan Si Kancil. Si Kancil digambarkan sebagai hewan kecil yang cerdas, cerdas, dan mampu mengatasi berbagai permasalahan dengan kecerdasan serta strategi, bukan dengan kekuatan fisik. Karakteristik tersebut menjadikan Si Kancil sebagai tokoh yang dekat dengan nilai-nilai pendidikan, seperti kecerdasan, kejujuran, dan kemampuan berpikir kritis dalam menghadapi tantangan (Abiyyulian Iqbal, 2025).

Dalam berbagai versi cerita rakyat, Si Kancil sering dihadapkan pada situasi berbahaya dan konflik dengan hewan lain yang lebih kuat, seperti harimau atau buaya. Melalui kecerdasannya, Si Kancil mampu menemukan solusi dan jalan keluar dari permasalahan yang dihadapi. Pola cerita ini relevan untuk diadaptasi ke dalam game bergenre petualangan dan labirin, di mana pemain dituntut untuk berpikir strategis dalam menyelesaikan tantangan serta menghindari musuh.

Pengadaptasian cerita rakyat Si Kancil ke dalam media game digital merupakan salah satu bentuk inovasi dalam penyajian budaya lokal. Game sebagai media interaktif memungkinkan cerita tidak hanya disampaikan secara naratif, tetapi juga dialami langsung oleh pemain melalui mekanisme permainan. Dengan demikian, pemain tidak hanya membaca atau mendengar cerita, tetapi turut berperan aktif sebagai tokoh utama dalam cerita tersebut (Apriono et al., 2024).

Dalam penelitian ini, cerita Petualangan Si Kancil digunakan sebagai elemen naratif utama yang mengiringi alur permainan pada setiap level. Narasi cerita disajikan secara bertahap untuk membangun keterkaitan antara cerita dan *gameplay*, sehingga cerita tidak hanya menjadi latar belakang, tetapi menjadi bagian integral dari pengalaman bermain. Pendekatan ini diharapkan mampu meningkatkan ketertarikan pemain serta memberikan nilai edukatif yang lebih kuat.

Integrasi cerita rakyat Si Kancil (Suharto, 2002) dalam *game* Android 2D ini juga bertujuan untuk memperkenalkan kembali budaya lokal Indonesia kepada generasi muda melalui media digital yang modern. Dengan mengombinasikan unsur cerita rakyat dan teknologi *game*, penelitian ini diharapkan dapat menjadi salah satu bentuk upaya pelestarian budaya Indonesia yang relevan dengan perkembangan teknologi saat ini.

2.8. Studi Literatur

Studi literatur digunakan sebagai acuan untuk mengetahui perkembangan penelitian yang telah dilakukan sebelumnya serta untuk mengidentifikasi celah penelitian (*research gap*) yang menjadi dasar dilakukannya penelitian ini. Fokus kajian pada penelitian terdahulu umumnya berkaitan dengan penerapan algoritma A* (A-Star) dalam pengembangan *game*, khususnya *game* dua dimensi dan *game* berbasis *mobile*.

Tabel 2. 1 Studi Literatur

1	Judul	Rancang Bangun <i>Pathfinding</i> Dengan Algoritma A* (Star) Pada NPC Menggunakan <i>Unity Engine</i>
	Penulis	M. Diky Oktavianto, Maulana Rizqi
	Tahun	2024
	ISBN/ISSN	ISSN(E): 2598-9715 & ISSN(P): 2087-3921
	Deskripsi	Penelitian ini memiliki fokus untuk menerapkan algoritma A* dalam pengembangan <i>game mobile</i> melalui <i>Unity Engine</i> . Algoritma A* dipilih karena keefektifannya dalam menemukan jalur terpendek dalam lingkungan berbasis <i>grid</i> , yang umumnya ditemukan dalam <i>game</i> . Pemilihan <i>Unity Engine</i> sebagai platform pengembangan didasarkan pada popularitasnya serta kemudahan integrasinya dengan <i>game mobile</i> . Penelitian ini diharapkan mampu berkontribusi pada kemajuan pengembangan <i>game mobile</i> di Indonesia yang dapat menghadirkan pengalaman yang lebih interaktif dan dinamis. Dalam penerapan Algoritma sangat penting di era perkembangan zaman dikarenakan algoritma dapat membantu seorang developer dalam perancangan sebuah <i>game</i> yang akan dibuat. Penelitian ini menggunakan metode penelitian GDLC (<i>Game Development Life Cycle</i>) yang terdiri dari beberapa tahap seperti <i>Initiation</i> , <i>Pre-production</i> , <i>Production</i> , Pengujian, dan <i>Release</i> .

2	Judul	Penerapan Algoritma A* Dan Behaviour Trees Untuk Perilaku Non-Player Character (NPC) Pada Game “The Last Hope” Berbasis Android Menggunakan Unity 2D
	Penulis	Muhammad Luthfi Setiawan, Arbansyah, Sayekhti Harits Suryawan
	Tahun	2023
	ISBN/ISSN	p-ISSN: 2723-567X & e-ISSN: 2723-5661
	Deskripsi	Algoritma A* (A Star) merupakan suatu algoritma untuk mencari rute terdekat yang optimal dan sempurna, bisa diartikan dalam hal mencari rute terbaik dan sempurna untuk mencapai tujuan sesuai dengan yang diharapkan. Dan pohon perilaku bersifat adaptif dan spesifik untuk situasi yang berbeda. Dengan hal tersebut, NPC akan diterapkan beberapa perilaku agar lebih terstruktur dan pada saat menargetkan <i>player</i> , NPC akan memilih jalur terdekat tanpa harus menabrak objek yang ada map <i>game</i> .
3	Judul	Penerapan Metode A-Star Pada Permainan Bergenre Platformer Menggunakan Godot
	Penulis	Kevin Rio Alexander, Julian Sahertian, Ratih Kumalasari
	Tahun	2023
	ISBN/ISSN	ISSN: 2580-3336 & 2549-7952 (Online)
	Deskripsi	AI merupakan faktor terpenting dalam meningkatkan <i>gameplay</i> pada semua jenis permainan, dalam penciptaan AI ada berbagai macam algoritma yang bisa digunakan, salah satunya adalah A*. Algoritma A-Star adalah algoritma pencarian rute terpendek yang merupakan algoritma yang dituntun oleh fungsi <i>heuristiknya</i> , yang menentukan urutan simpul mana yang akan dikunjungi terlebih dahulu, Kata <i>heuristic</i> berasal dari sebuah kata kerja bahasa Yunani, <i>heuriskein</i> yang berarti ‘mencari’ atau ‘menemukan’. Dalam metode pencarian, kata <i>heuristik</i> diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi.
4	Judul	SISTEM INFORMASI GEOGRAFIS DENGAN ALGORITMA A-STAR UNTUK MENENTUKAN JALUR TERPENDEK
	Penulis	Nur Padila, Basri, Cipta Riang Sari.
	Tahun	2023
	ISBN/ISSN	eISSN: 2686–3472
	Deskripsi	Algoritma A* (<i>Admissibel Heuristic</i>) adalah strategi <i>best first search</i> dengan memakai estimasi solusi pengeluaran rendah demi mencapai tujuan dengan jarak tempuh terdekat dan Ini memiliki nilai <i>heuristik</i> yang berfungsi sebagai dasar atau prinsip untuk pencapaian tujuan yang efektif Nilai <i>heuristik</i> digunakan untuk mempersempit ruang pencarian. Metode

		pencarian A* menghasilkan jalur optimal mulai dari titik awal dan kemudian melintasi grafik ke tujuan.
5	Judul	PENERAPAN ALGORITMA A STAR (A*) PADA GAME LABIRIN
	Penulis	Diana Y.A. Fallo, Vera Rosalina Bulu
	Tahun	2022
	ISBN/ISSN	eISSN : 2621-1467
	Deskripsi	Algoritma A* merupakan algoritma yang digunakan untuk menentukan rute terpendek objek menuju ke tujuan, dengan menghitung harga yang harus dipakai untuk mencari harga terkecil yang harus dibayarkan. Algoritma diperlukan untuk membuat <i>game</i> lebih menarik. Hal ini berbanding lurus dengan jumlah pengguna, semakin menarik <i>game</i> yang dimainkan, semakin banyak orang yang akan memainkan <i>game</i> tersebut. Di dalam sebuah <i>game</i> algoritma biasa digunakan untuk menentukan level, tingkat kesulitan, skor/nilai, hingga sebagai pengambil keputusan.
6	Judul	Implementasi Metode Pathfinding dengan Algoritma A* pada Game Rogue-like menggunakan Unity
	Penulis	Elvitro Gumelar Agung, Dania Eridani, Adnan Fauzi
	Tahun	2022
	ISBN/ISSN	ISSN 2460-9056
	Deskripsi	Inisiasi (<i>Initiation</i>) Proses inisiasi merupakan pembuatan konsep awal dari <i>game</i>, <i>gameplay</i> ditentukan dan serta menentukan target pengguna dari <i>game</i>. Hasil dari tahap ini adalah sebuah konsep serta deskripsi <i>game</i> yang sederhana. Pada tahap ini ditentukan konsep awal dari <i>game</i> yang menentukan bagaimana <i>game</i> dimainkan, selain itu dilakukan juga riset untuk metode dan algoritma yang digunakan, proses ini juga memilih <i>game engine</i> yang digunakan yaitu <i>Unity</i>, <i>game engine Unity</i> dipilih karena <i>Unity</i> merupakan aplikasi gratis dan tidak membutuhkan spesifikasi yang cukup tinggi pada komputer yang digunakan untuk membuat <i>game</i>. Pra-produksi (<i>Pre-production</i>) Tahap ini merupakan pembuatan dan revisi dari <i>game</i> serta pembuatan prototipe <i>game</i>. <i>Game</i> dibuat berdasarkan genre permainan, <i>gameplay</i>, mekanik <i>game</i>, karakter, tingkat kesulitan, aspek teknis, dan dokumentasi dari <i>game</i> tersebut. Tahap ini berakhir saat revisi atau perubahan desain <i>game</i> telah disetujui [10]. Prototipe dari <i>game</i> dibuat dan masih berupa aset sederhana, tahap ini dibuat mekanik dan <i>gameplay</i>. Pada tahap ini, dibuat peta <i>grid</i> sebagai media yang digunakan untuk <i>pathfinding</i> pada <i>game</i>, peta <i>grid</i> adalah susunan yang terdiri dari kotak atau <i>node</i> yang membentuk suatu peta. Langkah pertama dalam pembuatan <i>grid</i> adalah dengan membuat <i>node</i> atau kotak yang disusun pada <i>grid</i> menggunakan <i>script</i>.

		Produksi (<i>Production</i>) Proses ini merupakan pembuatan aset dan <i>source code</i> dari <i>game</i>. Tahap ini terkait dengan penciptaan dan penyempurnaan detail, penambahan fitur baru, meningkatkan performa secara keseluruhan, dan memperbaiki <i>bug</i> yang masih terdapat di dalam <i>game</i>. Kegiatan selama penyempurnaan dilakukan untuk membuat permainan lebih interaktif, menantang, dan lebih mudah dipahami [10]. Tahap ini, <i>game</i> kemudian diberikan aset, serta peningkatan performa dan memperbaiki <i>bug</i> yang masih terdapat pada <i>game</i>. Pada proses ini, <i>script</i> dari algoritma A* dihubungkan pada <i>prefab</i> yang sudah disediakan dengan cara membuat <i>script</i> pada <i>prefab</i> untuk menghubungkan variabel pada <i>script</i> algoritma A* dan <i>script</i> di dalam <i>prefab</i>. Pengujian (Testing) Tahap ini merupakan pengujian internal yang dilakukan untuk menguji keseluruhan <i>game</i>. Pada proses ini, jika ditemukan <i>bug</i>, atau kegagalan selama pengujian, penyebab dan skenario tersebut dianalisis dan didokumentasi.. Pengujian terkait dilakukan juga untuk meningkatkan kualitas aksesibilitas yang dapat diuji melalui pengamatan perilaku penguji. Jika tester merasa sulit untuk bermain dan memahami permainan, artinya <i>game</i> tersebut tidak cukup dapat diakses. Hasil dari tahap pengujian menentukan apakah <i>game</i> dapat dilanjutkan ke tahap berikutnya atau mengulangi siklus produksi. Beta Tahap ini dilakukan pengujian oleh pihak ketiga atau. Pengujian beta masih menggunakan metode pengujian yang sama dengan metode pengujian sebelumnya. Dalam tahap ini, penguji diminta untuk menemukan <i>bug</i> supaya <i>game</i> dapat lebih disempurnakan, penguji diberi kebebasan untuk menggunakan <i>game</i>. Keluaran dari pengujian beta adalah laporan <i>bug</i> dan masukan dari penguji. Tahap beta akan dilanjutkan ke tahap selanjutnya jika jangka waktu pengujian beta berakhir atau jumlah penguji beta yang ditentukan telah memenuhi kriteria. Rilis (<i>Release</i>) Pada tahap ini, <i>game</i> telah mencapai tahap akhir dan siap untuk dirilis. Rilis melibatkan dokumentasi proyek, perencanaan untuk pemeliharaan <i>game</i> dan ekspansi dari <i>game</i>. Setelah <i>game</i> selesai dibuat, langkah terakhir adalah pemeliharaan dari <i>game</i> dan menambahkan fitur tambahan dari <i>game</i> yang dibuat.
7	Judul	Implementasi Algoritma Collision Detection dan A*(A Star) pada Non Player Character Game World Of New Normal
	Penulis	Dian Novita Yohanes dan Naim Rochmawati
	Tahun	2022
	ISBN/ISSN	ISSN : 2686-2220

	Deskripsi	Artikel utama mengidentifikasi kebutuhan kecerdasan buatan pada NPC untuk menciptakan pengalaman bermain yang dinamis dan realistis. Fenomena utama yang diteliti adalah bagaimana algoritma A* Star digunakan untuk <i>pathfinding</i> (penentuan jalur terpendek NPC menuju pemain) dan <i>Collision Detection</i> untuk mendeteksi tabrakan antar objek dalam <i>game</i>. Deskripsi fenomena ini didasarkan pada observasi bahwa NPC seperti virus COVID dan hama tikus memerlukan respons cerdas untuk mengejar karakter utama DIAN, sesuai dengan mekanisme permainan di tiga level dengan kompleksitas bertambah. Berdasarkan tinjauan pustaka dalam dokumen, penalaran deskriptif mengelompokkan studi sebelumnya ke dalam dua kategori utama: 1. <i>Pathfinding</i> dengan A Star*: Imam & Wahyu (2017) mendeskripsikan implementasi A* pada <i>game</i> labirin Android untuk mencari jalur tercepat kelinci menuju makanan, sementara Rifky & Salamun (2020) menerapkannya pada NPC penjaga di <i>game</i> Good Thief yang mengejar pemain. Kedua studi ini mengonfirmasi efektivitas A* dalam menghitung F Cost ($G + H$) untuk jalur optimal pada <i>grid</i> 2D. 1. <i>Pathfinding</i> dengan A Star*: Imam & Wahyu (2017) mendeskripsikan implementasi A* pada <i>game</i> labirin Android untuk mencari jalur tercepat kelinci menuju makanan, sementara Rifky & Salamun (2020) menerapkannya pada NPC penjaga di <i>game</i> Good Thief yang mengejar pemain. Kedua studi ini mengonfirmasi efektivitas A* dalam menghitung F Cost ($G + H$) untuk jalur optimal pada <i>grid</i> 2D. 2. <i>Collision Detection</i>: Dedi & Edy (2018) menggunakan metode bounding box untuk mendeteksi tabrakan pada <i>game</i> petualangan edukasi Si Toole, mirip dengan pendekatan pada artikel utama yang membandingkan koordinat <i>min/max</i> objek. Temuan ini digambarkan secara naratif untuk menunjukkan konsistensi algoritma dalam konteks <i>game</i> Android, dengan penekanan pada perhitungan G Cost (jarak dari start), H Cost (estimasi ke <i>goal</i>), dan deteksi irisan <i>bounding box</i>. Secara deskriptif, studi literatur mengungkap pola bahwa A* Star efektif untuk NPC musuh dinamis (seperti virus yang mengejar pemain), sedangkan <i>Collision Detection</i> memastikan realisme interaksi objek (nyawa berkurang saat tabrakan). Integrasi keduanya pada <i>game</i> World of New Normal menghasilkan tingkat kepuasan pengguna 81% dari beta testing 50 responden, mengindikasikan keberhasilan aplikasi edukasi <i>new normal</i>. Penalaran ini membatasi diri
--	-----------	---

		pada deskripsi fakta dari sumber, tanpa hipotesis kausal, sesuai karakteristik metode deskriptif.
8	Judul	Game “Finding Easter Eggs” Berbasis Augmented Reality Menggunakan Algoritma A-Star
	Penulis	Louis D.S. Joseph, Immanuela P. Saputro dan Angelia M. Adrian
	Tahun	2021
	ISBN/ISSN	ISSN: 2541-2221 & E-ISSN: 2477-8079
	Deskripsi	Penelitian serupa pernah dilakukan oleh Aulia, Nizar, Yulindon, dan Yuliana (2021) dengan judul 'Implementasi Algoritma A (A-Star) Untuk Penentuan Jarak Terpendek Pada Game The Maze'. Dalam penelitian ini, dikembangkan sebuah permainan bergenre labirin menggunakan <i>game engine Unity</i> yang menerapkan algoritma A sebagai inti dari mekanisme pencarian jalur (<i>pathfinding</i>). Fokus utama dari studi ini adalah penerapan logika pencarian rute terpendek dari titik awal (Start) menuju titik tujuan (Finish) dengan menghindari rintangan berupa tembok labirin. Metode perhitungan yang diterapkan menggunakan rumus fungsi heuristik $f(n) = g(n) + h(n)$, di mana $g(n)$ merepresentasikan biaya dari titik awal ke node saat ini, sedangkan $h(n)$ merupakan estimasi biaya dari node saat ini menuju tujuan. Mekanisme kerja sistem dirancang dengan membagi area permainan menjadi struktur <i>grid</i>. Setiap kotak pada <i>grid</i> diperiksa statusnya, apakah dapat dilalui (<i>walkable</i>) atau merupakan halangan (<i>obstacle</i>). Hasil pengujian dari penelitian ini menunjukkan bahwa algoritma A* mampu bekerja secara efektif dalam menavigasi karakter untuk menemukan jalur solusi yang paling optimal dan efisien dalam lingkungan permainan labirin 3D. Relevansi penelitian ini dengan pengembangan <i>game</i> 'Petualangan Si Kancil' terletak pada penggunaan metode algoritma yang sama serta lingkungan pengembangan berbasis <i>Unity</i>.
9	Judul	Penerapan Algoritma Pathfinding A* dalam Game Dual Legacy berbasis Android
	Penulis	Felix Octavian dan Latius Hermawan
	Tahun	2023
	ISBN/ISSN	ISSN: 2087-2534 & E-ISSN: 2089-7624
	Deskripsi	Penelitian selanjutnya yang menjadi rujukan dilakukan oleh Noviasyah, Wahyuddin, dan Mardiani (2021) dengan judul 'Penerapan Algoritma A (Star) Pada Game Petualangan Labirin Berbasis Android*'. Penelitian ini berfokus pada pengembangan aplikasi permainan bergenre petualangan labirin yang dioperasikan pada perangkat bergerak berbasis Android.

		Dalam studi ini, algoritma A* (A-Star) diimplementasikan sebagai inti dari logika kecerdasan buatan (<i>Artificial Intelligence</i>) pada karakter musuh (<i>Non-Playable Character</i>). Tujuan utama penerapan algoritma tersebut adalah untuk memungkinkan karakter musuh melakukan pelacakan posisi pemain dan menentukan rute terpendek guna melakukan pengejaran di dalam struktur labirin yang kompleks. Metode pengembangan sistem yang diterapkan dalam penelitian ini mengacu pada <i>Multimedia Development Life Cycle</i> (MDLC) yang mencakup tahapan pengonsepan, perancangan, hingga distribusi. Proses pembuatan aplikasi memanfaatkan <i>game engine Unity 3D</i> untuk visualisasi dan logika permainan. Hasil akhir dari penelitian ini menyimpulkan bahwa algoritma A* mampu bekerja secara optimal dalam menghitung estimasi biaya terkecil menggunakan fungsi <i>heuristik</i>, sehingga musuh dapat menemukan jalur solusi yang efisien untuk mencapai target pemain tanpa terjebak oleh rintangan dinding.
10	Judul	Petualangan Si Kancil
	Penulis	Prih Suharto
	Tahun	2002
	ISBN/ISSN	ISBN 979-685-234-9
	Deskripsi	Dalam dokumen naskah cerita rakyat berjudul 'Petualangan Si Kancil', terdapat segmen cerita spesifik yang bertajuk 'Kotak Ajaib'. Kisah ini dipilih sebagai referensi utama dalam pengembangan alur permainan karena memuat struktur naratif yang dinamis dan relevan dengan mekanika penyelesaian masalah (<i>problem solving</i>). Secara deskriptif, narasi ini berfokus pada kecerdikan tokoh protagonis, Si Kancil, dalam menghadapi ancaman dari tokoh antagonis (Harimau) melalui strategi manipulasi psikologis, bukan konfrontasi fisik. Objek sentral dalam cerita ini adalah sebuah 'kotak' yang diklaim oleh Si Kancil sebagai barang keramat atau ajaib milik tokoh otoritas tertinggi (Raja Sulaiman) guna mengelabui musuhnya. Dari perspektif studi literatur, kisah 'Kotak Ajaib' ini merepresentasikan model <i>trickster tale</i> klasik, di mana kecerdasan intelektual diposisikan lebih unggul dibandingkan kekuatan fisik. Struktur plot yang dibangun—dimulai dari eksposisi ancaman, komplikasi berupa interaksi dengan objek (kotak), hingga resolusi berupa keberhasilan tipu muslihat—sangat kompatibel untuk diadaptasi ke dalam logika permainan bergenre <i>puzzle-adventure</i>. Selain itu, muatan nilai moral mengenai pentingnya kebijaksanaan dan bahaya sifat tamak yang terkandung di dalamnya menjadikan naskah ini materi literasi yang edukatif bagi target pengguna anak-anak."

BAB III

METODOLOGI PENELITIAN

3.1. Data Penelitian

Dalam pengembangan *game* "Petualangan Si Kancil", data penelitian dikumpulkan dan diklasifikasikan ke dalam tiga kategori utama guna mendukung proses perancangan sistem, implementasi konten, serta pengujian. Ketiga jenis data tersebut dijabarkan sebagai berikut:

3.1.1 Data Narasi Cerita Rakyat Petualangan Si Kancil

Data narasi yang digunakan sebagai materi utama penceritaan dalam penelitian ini bersumber dari naskah cerita "Petualangan Si Kancil". Pengolahan narasi diperlukan untuk menyampaikan alur cerita petualangan Si Kancil kepada pemain, sekaligus sebagai media literasi digital yang memuat nilai-nilai moral.

Data narasi yang digunakan sebagai landasan cerita utama dalam penelitian ini adalah naskah cerita berjudul 'Kotak Ajaib' yang termuat dalam dokumen 'Petualangan Si Kancil' karya Prih Suharto. Naskah cerita ini dijadikan acuan primer dalam penyusunan skenario permainan untuk memastikan alur petualangan yang disajikan sesuai dengan sumber referensi yang digunakan.

3.1.2 Data Sistem Permainan

Data sistem mencakup seluruh elemen teknis dan aset digital yang diperlukan untuk membangun arsitektur permainan. Data ini terdiri dari:

- a. Aset Visual (*Sprite* 2D): Meliputi gambar karakter, desain lingkungan labirin, item koleksi, serta elemen antarmuka pengguna (tombol dan panel menu).
- b. Aset Audio: Kumpulan efek suara (SFX) dan musik latar (*Background Music*) untuk mendukung suasana permainan dan suara narasi untuk alur perkembangan cerita pada permainan.
- c. Logika Algoritma: Skrip pemrograman dalam bahasa C# yang memuat implementasi Algoritma A* (A-Star) untuk navigasi kecerdasan buatan (*Artificial Intelligence*) musuh, serta logika aturan permainan

3.1.3 Data Responden

Data responden merupakan data primer yang diperoleh melalui kegiatan pengujian pengguna (*User Acceptance Test*). Data ini dikumpulkan dari hasil penyebaran kuesioner kepada 16 responden yang bertindak sebagai penguji aplikasi. Informasi yang dihimpun meliputi penilaian subjektif terhadap aspek antarmuka (*user interface*), kemudahan kontrol permainan, kejelasan alur cerita, serta tingkat kepuasan pengalaman pengguna (*User Experience*) setelah memainkan aplikasi.

Tabel 3. 1 Tabel Data Responden

No.	Nama	Usia	Jenis Kelamin
1	Haidar	5	Laki-laki
2	Al	7	Laki-laki
3	Khilan	7	Laki-laki
4	Sabian	8	Laki-laki
5	Ghani	8	Laki-laki
6	Rifqi	9	Laki-laki
7	Noplin	9	Laki-laki
8	Atar	9	Laki-laki
9	Kareem	10	Laki-laki
10	Zain	21	Laki-laki
11	Khamid	21	Laki-laki
12	Chika	22	Perempuan
13	Riyadh	25	Laki-laki
14	Khalid	25	Laki-laki
15	Faqih	26	Laki-laki
16	Dwina	26	Perempuan

3.2. Metode Pembanding

Penelitian ini menggunakan pendekatan studi literatur komparatif untuk memosisikan kontribusi penelitian terhadap pengembangan *game* Android 2D dengan penerapan algoritma *pathfinding* dan integrasi nilai budaya lokal. Metode pembanding dilakukan melalui analisis penelitian-penelitian terdahulu yang relevan, khususnya yang mengkaji penerapan algoritma A-Star (A*) dalam konteks *game development* dan studi komparatif algoritma pencarian jalur.

3.2.1 Pertimbangan Pemilihan Algoritma A-Star

Pemilihan algoritma A-Star (A*) sebagai fokus utama penelitian ini didasarkan pada beberapa penelitian pembanding yang menunjukkan keunggulan A-Star (A*) dalam konteks *game* berbasis *grid* dan lingkungan labirin:

- a. Penelitian Oktavianto dan Rizqi (2024) membahas penerapan algoritma A-Star (A*) dalam pengembangan *game mobile* menggunakan *Unity Engine*. Penelitian tersebut menitikberatkan pada pemanfaatan algoritma A-Star (A*) sebagai kecerdasan buatan karakter musuh untuk menentukan jalur terpendek menuju target. Hasil penelitian menunjukkan bahwa algoritma A-Star (A*) mampu meningkatkan efektivitas pergerakan musuh sehingga permainan menjadi lebih dinamis dan menantang. Namun demikian, penelitian ini masih berfokus pada aspek teknis penerapan algoritma dan belum mengintegrasikan unsur budaya atau kearifan lokal sebagai bagian utama dari konten permainan.

- b. Penelitian lain yang mengkaji implementasi algoritma A-Star (A*) pada *game* labirin dua dimensi untuk meningkatkan tingkat kesulitan permainan juga menunjukkan hasil yang konsisten. Penelitian tersebut menggunakan lingkungan berbasis *grid* dan memanfaatkan algoritma A-Star (A*) untuk mengatur pergerakan musuh dalam mengejar pemain. Hasil pengujian menunjukkan bahwa algoritma A-Star (A*) mampu menghasilkan jalur optimal dengan waktu komputasi yang efisien. Akan tetapi, penelitian ini hanya menekankan pada performa algoritma dan mekanisme permainan, tanpa mengangkat aspek naratif atau nilai budaya sebagai elemen pendukung *game*.

3.2.2 Relevansi Studi Komparatif Dijkstra dan A-Star (A*)

Untuk memperkuat pertimbangan pemilihan A-Star (A*) dan memahami *trade-off* yang terlibat, penelitian ini merujuk pada studi komparatif yang dilakukan oleh Pratama et al. (2025) dalam jurnal "Perbandingan Algoritma Dijkstra dan A-Star dalam Pencarian Rute Terpendek Sekolah Menengah Atas (SMA) di Kota Palangka Raya Berbasis Website."

Penelitian Pratama et al. (2025) memberikan kontribusi signifikan dalam memahami karakteristik kedua algoritma melalui pengujian empiris pada 8 skenario berbeda dengan metrik evaluasi yang komprehensif meliputi:

- a. Jarak tempuh (dalam kilometer).
- b. Durasi waktu (dalam menit).
- c. Waktu eksekusi (dalam detik).
- d. Jumlah node yang diperiksa.

Hasil penelitian Pratama et al. (2025) menunjukkan pola performa yang konsisten namun kontekstual:

- a. Keunggulan A-Star pada Graf Linear/Sederhana:
 - 1) Rata-rata selisih jarak: A-Star lebih pendek 0,78 KM.
 - 2) Rata-rata selisih waktu tempuh: A-Star lebih cepat 4,71 menit.
 - 3) Rata-rata selisih waktu eksekusi: A-Star lebih cepat 0,57 detik.
 - 4) Rata-rata selisih *node* yang diperiksa: A-Star memeriksa 9 *node* lebih sedikit.
- b. Keunggulan Dijkstra pada Graf Kompleks (Percobaan 8):
 - 1) Jarak: Dijkstra lebih pendek 0,17 KM.
 - 2) Waktu tempuh: Dijkstra lebih cepat 0,21 menit.
 - 3) *Node* yang diperiksa: Dijkstra memeriksa 2 *node* lebih sedikit.
 - 4) Waktu eksekusi: Sama dengan A-Star.

Temuan Pratama et al. (2025) sangat relevan dengan konteks *game escape* labirin dalam penelitian ini karena:

- a. *Game* labirin yang dikembangkan memiliki struktur grid 2D dengan jalur yang relatif linear pada level awal dan meningkat

kompleksitasnya pada level lanjut. Berdasarkan temuan Pratama et al., A-Star terbukti lebih efisien pada struktur seperti ini karena:

- 1) *Heuristik Manhattan Distance* sesuai untuk pergerakan 4-arah (tanpa diagonal).
 - 2) Eksplorasi *node* lebih terfokus menuju target (pemain).
 - 3) *Overhead* komputasi lebih rendah untuk *real-time pathfinding*.
- b. Berbeda dengan aplikasi pencarian rute statis seperti dalam penelitian Pratama et al., *game* memerlukan *recalculation* jalur secara kontinu setiap kali pemain berpindah posisi. Keunggulan A-Star dalam hal:
- 1) Waktu eksekusi lebih cepat (rata-rata 0,57 detik).
 - 2) Jumlah *node* yang diperiksa lebih sedikit (rata-rata 9 *node*). menjadi faktor krusial untuk menjaga responsivitas AI musuh tanpa menyebabkan *lag* atau penurunan *frame rate*.
- c. Meskipun Dijkstra dapat menghasilkan jalur yang sedikit lebih pendek pada kasus tertentu (seperti ditunjukkan pada Percobaan 8), perbedaan jarak yang minimal (0,17 KM atau ~6% dari total jarak) tidak signifikan dalam konteks *gameplay*. Yang lebih penting adalah:
- 1) Konsistensi performa AI.
 - 2) Pengalaman bermain yang *smooth*.
 - 3) Efisiensi komputasi untuk mendukung elemen *game* lainnya.

3.2.3 Gap Penelitian dan Kontribusi Unik

Berdasarkan tinjauan terhadap penelitian Pratama et al. (2025) dan penelitian terdahulu lainnya, dapat diidentifikasi beberapa gap penelitian yang menjadi dasar kontribusi penelitian ini:

- a. Konteks Aplikasi
 - 1) Penelitian Pratama et al.: Aplikasi web statis untuk pencarian rute geografis.
 - 2) Penelitian ini: *Game* Android dengan AI dinamis yang memerlukan *pathfinding real-time*.
- b. Integrasi Budaya
 - 1) Penelitian sebelumnya: Fokus murni pada aspek teknis algoritma.
 - 2) Penelitian ini: Mengintegrasikan cerita rakyat Si Kancil sebagai elemen naratif utama, menjadikan *game* sebagai media pelestarian budaya digital.
- c. Target Pengguna dan Tujuan
 - 1) Penelitian sebelumnya: Utilitas navigasi untuk pengguna umum.
 - 2) Penelitian ini: Media edutainment yang menggabungkan nilai edukatif budaya dengan mekanisme *game* yang menantang.

3.2.4 Positioning Penelitian

Dengan mempertimbangkan temuan dari studi komparatif Pratama et al. (2025) dan gap yang teridentifikasi, penelitian ini diposisikan sebagai:

- a. Aplikasi Praktis dari Studi Komparatif: Mengimplementasikan rekomendasi bahwa A-Star lebih cocok untuk graf dengan struktur sederhana hingga menengah (seperti labirin *game*).
- b. Inovasi Konten: Menambahkan dimensi baru pada penerapan algoritma *pathfinding* dengan mengintegrasikan nilai budaya lokal Indonesia.
- c. Validasi Kontekstual: Menguji apakah keunggulan A-Star yang terbukti dalam konteks pencarian rute geografis juga berlaku dalam konteks *game real-time*.

Selanjutnya, penelitian lain pada periode sama yang mengembangkan *game* petualangan berbasis Android dengan kecerdasan buatan juga memanfaatkan algoritma A-Star (A*) sebagai metode pencarian jalur. Penelitian tersebut bertujuan untuk meningkatkan realisme perilaku karakter non-pemain melalui pergerakan yang adaptif terhadap lingkungan. Meskipun berhasil menerapkan algoritma A-Star (A*) secara efektif, konten cerita dalam *game* masih bersifat umum dan tidak memiliki keterkaitan dengan budaya lokal Indonesia.

Berdasarkan beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa algoritma A-Star (A*) telah banyak digunakan dan terbukti efektif dalam pengembangan *game*, khususnya untuk pengaturan pergerakan musuh dalam lingkungan labirin. Namun, sebagian besar penelitian masih berfokus pada aspek teknis dan performa algoritma, sementara integrasi unsur kearifan lokal atau budaya Indonesia sebagai konten utama permainan belum banyak dikaji secara mendalam.

Oleh karena itu, penelitian ini hadir sebagai upaya untuk mengisi celah penelitian dengan mengombinasikan penerapan algoritma A-Star (A*) sebagai kecerdasan buatan musuh dengan integrasi cerita rakyat Indonesia, yaitu Petualangan Si Kancil, sebagai elemen naratif utama dalam *game*.

Pendekatan ini diharapkan dapat memberikan kontribusi baru dalam pengembangan *game*, tidak hanya dari sisi teknis kecerdasan buatan, tetapi juga dari sisi pelestarian budaya lokal melalui media *game* digital.

3.3. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode penelitian rekayasa perangkat lunak dengan pendekatan *Game Development Life Cycle* (GDLC). GDLC merupakan metode pengembangan *game* yang mencakup tahapan perencanaan hingga pengujian, yang dirancang khusus untuk menghasilkan *game* yang berkualitas baik dari sisi teknis maupun pengalaman pengguna.

Pemilihan metode GDLC didasarkan pada kesesuaiannya dengan karakteristik penelitian ini, yaitu pengembangan *game* sebagai produk utama penelitian. Dalam pengembangan *game* "Petualangan Si Kancil", metode

pengembangan sistem diterapkan secara bertahap mulai dari perencanaan awal hingga tahap akhir peluncuran. Pendekatan ini dipilih agar proses integrasi algoritma A-Star (A*) ke dalam mekanika permainan dapat terorganisir dengan baik dan menghasilkan sistem yang stabil. Adapun rincian tahapan yang dilakukan adalah sebagai berikut:

3.3.1 Initiation

Tahap *Initiation* merupakan tahap awal dalam pengembangan *game* yang berfokus pada penentuan konsep dasar permainan. Pada tahap ini dilakukan identifikasi ide permainan, tujuan pengembangan *game*, target platform, serta genre *game* yang akan dibuat.

Tahap *Initiation* dimanfaatkan sebagai langkah fondasi untuk menetapkan batasan masalah dan tujuan akhir penelitian. Pada fase ini, analisis mendalam dilakukan terhadap materi cerita rakyat "Si Kancil" guna menentukan elemen narasi yang akan diangkat. Selain itu, konsep dasar permainan dirancang secara garis besar. Algoritma A* ditetapkan sebagai metode kecerdasan buatan (AI) karena dinilai paling efisien dalam menangani logika pencarian jalur terpendek (*pathfinding*) pada struktur labirin yang kompleks.

3.3.2 Pre-Production

Pada tahap pra-produksi, seluruh elemen teknis dan visual dirancang secara terperinci sebelum memasuki proses pemrograman. Aktivitas yang dilakukan meliputi:

- a. Penyusunan skenario permainan yang diadaptasi dari alur cerita petualangan Si Kancil.
- b. Perancangan aset visual, meliputi desain karakter (Si Kancil dan Musuh), lingkungan labirin, serta antarmuka pengguna (*User Interface*).
- c. Perancangan logika sistem *grid* yang akan digunakan sebagai area navigasi algoritma.
- d. Penyusunan skema logika pemrograman untuk pergerakan karakter dan musuh. Seluruh dokumen rancangan pada tahap ini dijadikan acuan utama untuk fase produksi.

3.3.3 Production

Tahap produksi merupakan fase realisasi dari rancangan yang telah disusun sebelumnya. Proses pembangunan aplikasi dilakukan menggunakan game engine Unity dengan bahasa pemrograman C#. Fokus implementasi pada tahap ini mencakup:

- a. Penyusunan seluruh *scene* permainan, mulai dari menu utama, level permainan, hingga tampilan akhir.
- b. Pemrograman kontrol karakter pemain agar dapat digerakkan dengan responsif.
- c. Implementasi kode algoritma A* pada karakter musuh (NPC) untuk mendeteksi dan mengejar posisi pemain secara otomatis.

- d. Penerapan rumus *heuristik Manhattan Distance* pada sistem navigasi labirin.
- e. Integrasi aset audio dan visual untuk menciptakan pengalaman bermain yang interaktif.

3.3.4 Testing

Tahap Testing bertujuan untuk memastikan bahwa *game* yang dikembangkan berjalan sesuai dengan fungsionalitas yang diharapkan. Pengujian dilakukan untuk mengidentifikasi kesalahan (*bug*) serta memastikan bahwa algoritma A* berfungsi dengan baik dalam mengatur pergerakan musuh.

Pengujian pada penelitian ini dilakukan menggunakan metode pengujian fungsional (*black box testing*), dengan fokus pada pengujian fitur permainan, perilaku musuh, serta alur permainan pada setiap level.

Tahap pengujian dilaksanakan untuk memverifikasi bahwa seluruh fungsi sistem berjalan tanpa kendala. Evaluasi dibagi menjadi dua fokus utama:

- a. Pengujian Fungsional (Black Box): Dilakukan untuk memeriksa respons tombol, transisi antar layar, serta mekanisme menang/kalah.
- b. Pengujian Algoritma: Difokuskan pada akurasi jalur yang dibentuk oleh musuh dan kecepatan respons algoritma A* saat target (pemain) berpindah posisi. Jika ditemukan *bug* atau kesalahan logika, perbaikan segera dilakukan sebelum melangkah ke tahap selanjutnya.
- c. Pengujian Dari Pengguna: Dilakukan untuk

3.3.5 Beta

Pada tahap beta, versi permainan yang telah stabil didistribusikan kepada kelompok pengguna terbatas. Tujuan tahap ini adalah:

- a. Mengumpulkan data terkait kenyamanan pengguna (*User Experience*) dan tingkat kesulitan permainan.
- b. Mendapatkan umpan balik mengenai kesesuaian elemen cerita budaya lokal yang disajikan.
- c. Melakukan revisi akhir berdasarkan data kuesioner atau saran yang diterima dari responden.

3.3.6 Release

Tahap rilis menandai penyelesaian siklus pengembangan. Pada fase ini, aplikasi permainan diekspor (*build*) menjadi format instalasi yang siap digunakan untuk Android. Selain itu, dokumentasi teknis dan panduan penggunaan turut disusun sebagai pelengkap laporan penelitian, menandakan bahwa sistem telah siap untuk dipresentasikan dan digunakan secara luas.

3.4. Rancangan Pengujian

Dalam penelitian ini, tahapan pengujian dirancang secara sistematis untuk memvalidasi implementasi algoritma A* (A-Star) serta memastikan fungsionalitas sistem secara menyeluruh. Serangkaian skenario pengujian disusun guna menjamin efektivitas, daya tarik visual, dan stabilitas operasional dari *game* yang dikembangkan. Adapun rancangan pengujian tersebut dibagi menjadi tiga kategori utama sebagai berikut:

3.4.1 Rancangan Pengujian Fungsi Utama Sistem

Pengujian fungsional difokuskan pada verifikasi fitur-fitur dasar permainan untuk memastikan seluruh komponen berjalan sesuai spesifikasi rancangan. Metode Black Box Testing diterapkan dalam tahap ini, di mana pengujian dilakukan terhadap fungsionalitas antarmuka tanpa memeriksa struktur kode internal. Aspek-aspek yang diuji meliputi:

- a. Responsivitas tombol navigasi dan kontrol.
- b. Kelancaran mekanisme transisi antar-*scene*.
- c. Logika perpindahan level permainan.
- d. Akurasi perhitungan sistem skor.

Hasil yang diharapkan adalah seluruh fitur permainan dapat beroperasi dengan stabil tanpa adanya kesalahan teknis (*bug* atau *error*).

3.4.2 Rancangan Pengujian Algoritma A-Star (A*)

Pengujian ini bertujuan untuk mengevaluasi efisiensi dan akurasi navigasi karakter musuh (*Non-Playable Character*/NPC) dalam menerapkan algoritma pencarian jalur. Indikator kinerja yang dievaluasi mencakup:

- a. Perilaku NPC saat posisi pemain berada di luar radius deteksi.
- b. Responsivitas NPC saat pemain memasuki area jangkauan pengejaran.
- c. Perubahan perilaku NPC saat pemain berhasil meloloskan diri dari jangkauan.

Hasil yang diharapkan adalah NPC mampu mendeteksi keberadaan pemain dan menghasilkan rute pengejaran terpendek (*shortest path*) secara optimal dan efisien.

3.4.3 Rancangan Pengujian Dari Pengguna

Pengujian tahap akhir dilakukan untuk mengukur tingkat kepuasan serta persepsi subjektif pengguna terhadap kualitas permainan. Pengumpulan data dilakukan melalui penyebaran kuesioner yang menggunakan metode pengukuran Skala Likert. Variabel penilaian yang diukur meliputi:

- a. Kualitas tampilan antarmuka (UI/UX) dan visual.
- b. Keseimbangan tingkat kesulitan permainan (*game balancing*).
- c. Tingkat ketertarikan pengguna untuk memainkan kembali (*replayability*).

- d. Tingkat kepuasan terhadap tantangan yang diberikan oleh kecerdasan buatan (AI) musuh.

Hasil yang diharapkan adalah diperoleh data yang menunjukkan bahwa mayoritas responden memberikan penilaian positif, yang mengindikasikan bahwa *game* menarik dan layak untuk dimainkan.

3.5. Perancangan Sistem Game

Perancangan sistem *game* merupakan tahap penting dalam metodologi penelitian yang bertujuan untuk menggambarkan bagaimana sistem permainan dibangun dan bagaimana komponen-komponen di dalam *game* saling berinteraksi. Pada tahap ini, dilakukan perancangan alur permainan, perancangan aktor yang terlibat, serta perancangan mekanisme permainan secara keseluruhan. Perancangan sistem ini digunakan sebagai acuan pada tahap implementasi agar pengembangan *game* berjalan sesuai dengan tujuan penelitian.

3.5.1 Gambaran Umum Sistem Game

Game yang dikembangkan dalam penelitian ini merupakan *game* Android 2D bergenre *escape* labirin dengan tokoh utama Si Kancil. Pemain berperan sebagai Kancil yang harus menyelesaikan setiap level dengan cara mengumpulkan item tertentu dan menghindari musuh yang bergerak secara dinamis menggunakan algoritma A* (A-Star).

Sistem *game* terdiri dari beberapa komponen utama, yaitu pemain (*player*), musuh (*enemy*), lingkungan labirin, item yang dapat dikumpulkan, serta sistem tujuan (*goal*) pada setiap level. Interaksi antar komponen tersebut membentuk alur permainan yang menantang dan berkesinambungan.

3.5.2 Perancangan Aktor Sistem

Aktor dalam sistem *game* ini terdiri dari dua jenis, yaitu aktor utama dan aktor pendukung. Aktor utama adalah pemain yang mengendalikan karakter Si Kancil, sedangkan aktor pendukung adalah sistem permainan yang mengatur perilaku musuh dan alur permainan.

Tabel 3. 2 Tabel Perancangan Aktor Sistem

Pemain	Pemain berperan mengendalikan karakter Si Kancil untuk bergerak di dalam labirin, mengumpulkan item, serta mencapai titik keluar pada setiap level.
Musuh	Musuh merupakan karakter non-pemain (<i>Non-Player Character/NPC</i>) yang dikendalikan oleh sistem menggunakan algoritma A* untuk mengejar pemain. Musuh akan bergerak secara otomatis

	berdasarkan posisi pemain dan struktur labirin.
Sistem <i>game</i>	Sistem <i>game</i> bertugas mengatur logika permainan, seperti perhitungan skor, pengaturan level, kondisi menang atau kalah, serta pengelolaan alur cerita.

3.5.3 Alur Permainan (Game Flow)

Alur permainan pada *game* ini dimulai dari menu utama, di mana pemain dapat memulai permainan atau keluar dari *game*. Setelah permainan dimulai, pemain akan memasuki level pertama yang diawali dengan penyajian cerita singkat sebagai pengantar permainan.

Pada setiap level, pemain harus mengumpulkan item yang telah ditentukan untuk dapat membuka akses menuju pintu keluar. Selama proses tersebut, pemain harus menghindari musuh yang bergerak menggunakan algoritma A*. Jika pemain berhasil mencapai pintu keluar, maka level dinyatakan selesai dan pemain dapat melanjutkan ke level berikutnya. Sebaliknya, jika pemain tertangkap oleh musuh, maka permainan akan berakhir dan pemain harus mengulangi level tersebut.

3.5.4 Perancangan Level Game

Game yang dikembangkan terdiri dari tiga level permainan dengan tingkat kesulitan yang meningkat secara bertahap. Setiap level dirancang dengan struktur labirin yang berbeda untuk memberikan variasi tantangan kepada pemain.

Tabel 3. 3 Tabel Perancangan Level Game

Level 1	Pemain diperkenalkan dengan mekanisme dasar permainan, seperti pergerakan karakter dan pengumpulan item serta dihadapkan pada musuh yang bergerak menggunakan algoritma A* dengan penyesuaian ringan untuk pendeteksi jarak antara musuh dengan pengguna.
Level 2	Pemain mulai dihadapkan pada musuh yang bergerak menggunakan algoritma A* dengan peningkatan pendeteksi jarak musuh dengan pengguna.
Level 3	Tingkat kesulitan ditingkatkan melalui desain labirin yang lebih kompleks dan pergerakan musuh yang lebih menantang.

3.5.5 Perancangan Mekanisme Permainan

Mekanisme permainan dirancang untuk memberikan pengalaman bermain yang menantang namun tetap mudah dipahami. Pemain dapat menggerakkan karakter menggunakan kontrol sentuh, sementara sistem *game* secara otomatis mengatur pergerakan musuh menggunakan algoritma A*.

Setiap level memiliki tujuan utama yang harus dicapai oleh pemain, yaitu mengumpulkan item dan mencapai titik keluar. Mekanisme ini dirancang agar pemain dituntut untuk berpikir strategis dalam menentukan jalur pergerakan yang aman untuk menghindari musuh.

3.6. Perancangan Algoritma A* (A-Star) pada Game

Perancangan algoritma A* (A-Star) pada *game* bertujuan untuk menggambarkan bagaimana algoritma tersebut diterapkan secara konseptual dalam sistem permainan sebelum masuk ke tahap implementasi. Tahap perancangan ini berfungsi sebagai jembatan antara landasan teori algoritma A* yang telah dibahas pada BAB II dengan implementasi teknis pada BAB IV.

3.6.1 Tujuan Penerapan Algoritma A*

Tujuan utama penerapan algoritma A* dalam *game* ini adalah untuk mengatur pergerakan musuh agar mampu menentukan jalur terpendek dan paling optimal dalam mengejar pemain di lingkungan labirin. Dengan algoritma A*, musuh diharapkan dapat bergerak secara cerdas, adaptif, dan responsif terhadap posisi pemain, sehingga permainan menjadi lebih menantang dan tidak mudah diprediksi.

3.6.2 Representasi Lingkungan Game

Lingkungan permainan direpresentasikan dalam bentuk *grid* dua dimensi yang disusun menggunakan sistem *Tilemap* pada *Unity*. Setiap *tile* pada labirin dianggap sebagai sebuah *node* yang memiliki atribut dapat dilewati (*walkable*) atau tidak dapat dilewati (*unwalkable*).

Node-node yang dapat dilewati digunakan sebagai jalur pergerakan musuh, sedangkan *node* yang tidak dapat dilewati berfungsi sebagai rintangan seperti dinding labirin. Representasi *grid* ini memudahkan algoritma A* dalam melakukan proses pencarian jalur berdasarkan hubungan antar *node*.

3.6.3 Penentuan Node Awal dan Node Tujuan

Dalam perancangan algoritma A*, posisi musuh ditetapkan sebagai *node* awal (*start node*), sedangkan posisi pemain ditetapkan sebagai *node* tujuan (*goal node*). Setiap kali terjadi perubahan posisi pemain, sistem akan memperbarui *node* tujuan sehingga jalur yang dihitung oleh algoritma A* bersifat dinamis.

Dengan pendekatan ini, musuh akan selalu berusaha mencari jalur terbaik untuk mendekati pemain berdasarkan kondisi terbaru di dalam permainan.

3.6.4 Perhitungan Nilai Heuristik

Nilai *heuristik* ($h(n)$) digunakan untuk memperkirakan jarak antara *node* saat ini dengan *node* tujuan. Pada penelitian ini, fungsi *heuristik* yang digunakan adalah jarak *Manhattan*, karena sesuai dengan karakteristik pergerakan pada *grid* dua dimensi yang hanya memungkinkan pergerakan ke atas, bawah, kiri, dan kanan.

Penggunaan *heuristik* jarak *Manhattan* membantu algoritma A* dalam mempercepat proses pencarian jalur tanpa mengurangi optimalisasi jalur yang dihasilkan.

3.6.5 Alur Kerja Algoritma A-Star (A*) pada Musuh

Alur kerja algoritma A* pada musuh dimulai ketika permainan berjalan dan posisi pemain terdeteksi oleh sistem. Musuh kemudian melakukan proses pencarian jalur dari posisinya menuju posisi pemain menggunakan algoritma A*.

Setelah jalur ditemukan, musuh akan bergerak mengikuti *node-node* pada jalur tersebut secara bertahap. Proses pencarian jalur dapat dilakukan kembali secara berkala untuk menyesuaikan pergerakan musuh dengan perubahan posisi pemain, sehingga perilaku musuh terlihat lebih realistis dan adaptif.

3.6.6 Integrasi Algoritma A-Star (A*) dengan Sistem Game

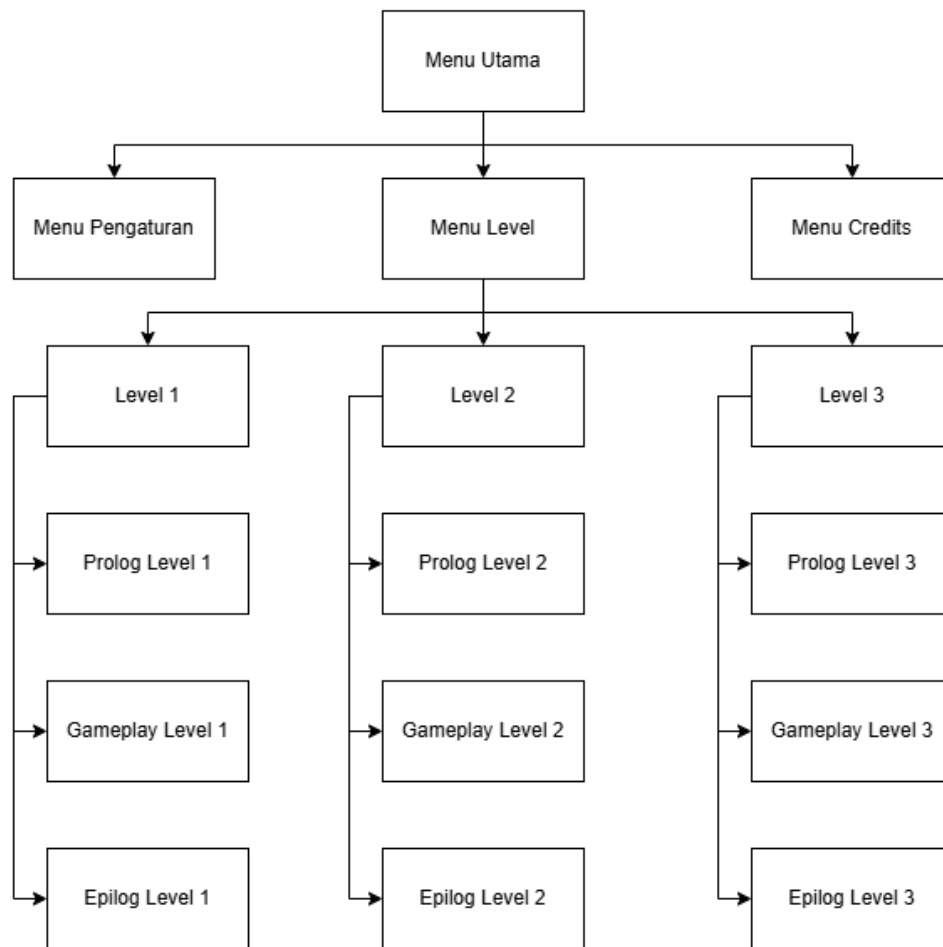
Algoritma A* diintegrasikan dengan sistem *game* menggunakan bahasa pemrograman C# pada *Unity*. Hasil perhitungan jalur digunakan sebagai dasar pergerakan musuh di dalam *game*.

Integrasi ini memungkinkan algoritma A* bekerja secara sinkron dengan sistem kontrol pemain, sistem level, serta mekanisme permainan lainnya. Dengan demikian, penerapan algoritma A* tidak hanya bersifat teoritis, tetapi benar-benar berfungsi sebagai bagian inti dari kecerdasan buatan musuh dalam *game* Android 2D yang dikembangkan.

3.7. Rancangan Menu

Struktur hierarki navigasi antarmuka (*interface navigation structure*) pada permainan dipetakan sebagaimana terlihat pada rancangan menu, di mana 'Menu Utama' ditetapkan sebagai simpul pusat (*root node*) atau gerbang akses utama. Dari titik pusat ini, akses didistribusikan ke dalam tiga modul fungsional, yaitu 'Menu Pengaturan' untuk konfigurasi sistem, 'Menu Level' untuk pemilihan babak, serta 'Menu Credits' untuk informasi pengembang.

Secara spesifik pada modul 'Menu Level', alur permainan dirancang dengan pendekatan sekuensial untuk setiap tingkatan (Level 1, 2, dan 3). Setiap sesi level disusun dalam struktur naratif yang linier, dimulai dari penayangan 'Prolog' sebagai pengantar cerita, dilanjutkan dengan fase inti 'Gameplay' (permainan labirin), dan diakhiri dengan 'Epilog'. Kendati demikian, fleksibilitas akses tetap diakomodasi melalui opsi pada 'Menu Pengaturan', di mana penayangan segmen naratif ('Prolog' dan 'Epilog') dapat dinonaktifkan apabila fokus permainan ingin diarahkan langsung pada inti *gameplay*.



Gambar 3. 1 Rancangan Menu

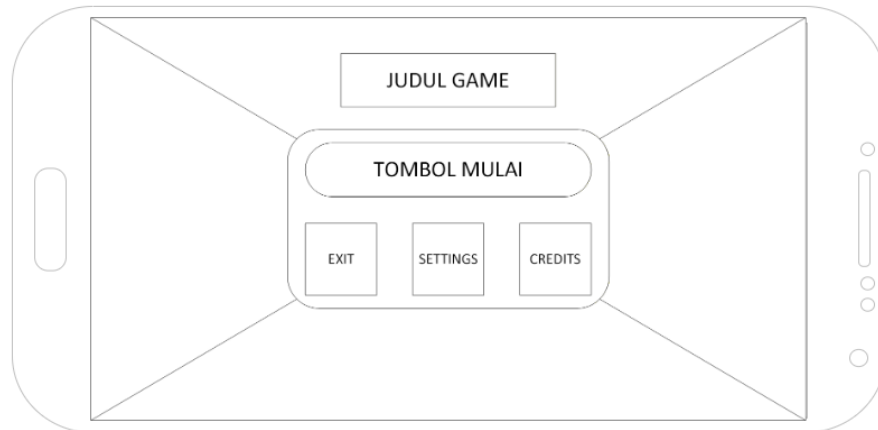
3.8. Rancangan Layar

Perancangan layar pada *game* 'Petualangan Si Kancil' disusun untuk memberikan gambaran visual mengenai antarmuka yang akan digunakan. Rancangan ini dibuat agar setiap tampilan mudah dipahami oleh pengguna, sehingga interaksi di dalam permainan dapat berjalan dengan lancar, efektif, dan efisien.

3.8.1 Rancangan Layar Menu Utama

Pada rancangan antarmuka yang divisualisasikan dalam gambar tersebut, ditampilkan untuk halaman Menu Utama yang didesain menggunakan orientasi layar lanskap (*landscape*). Elemen identitas permainan, yakni Judul *Game*, ditempatkan pada posisi *center-top* untuk memberikan informasi visual pertama kepada pengguna. Sebagai pusat interaksi, tombol 'Mulai' diposisikan secara dominan di tengah layar guna memfasilitasi akses utama menuju permainan (*gameplay*). Tepat di bawah tombol utama tersebut, disusun secara horizontal tiga tombol navigasi sekunder yang meliputi tombol '*Exit*', '*Settings*', dan '*Credits*',

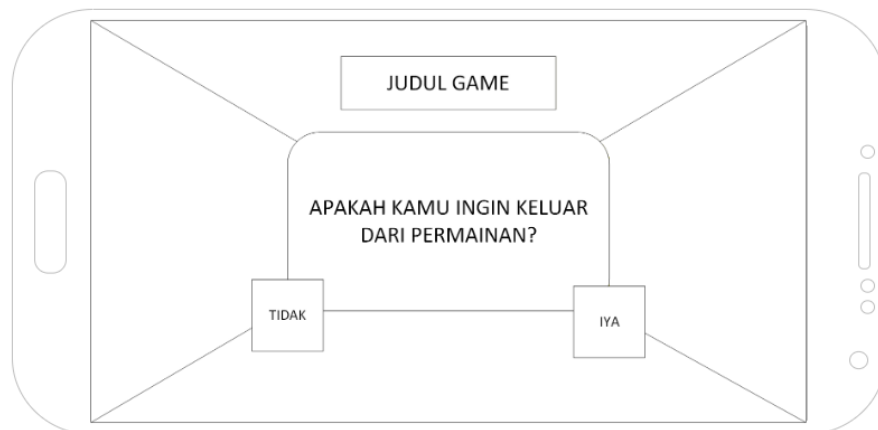
yang masing-masing difungsikan untuk keperluan keluar dari aplikasi, pengaturan sistem, serta informasi pengembang.



Gambar 3. 2 Rancangan Layar Menu Utama

3.8.2 Rancangan Layar Keluar Permainan

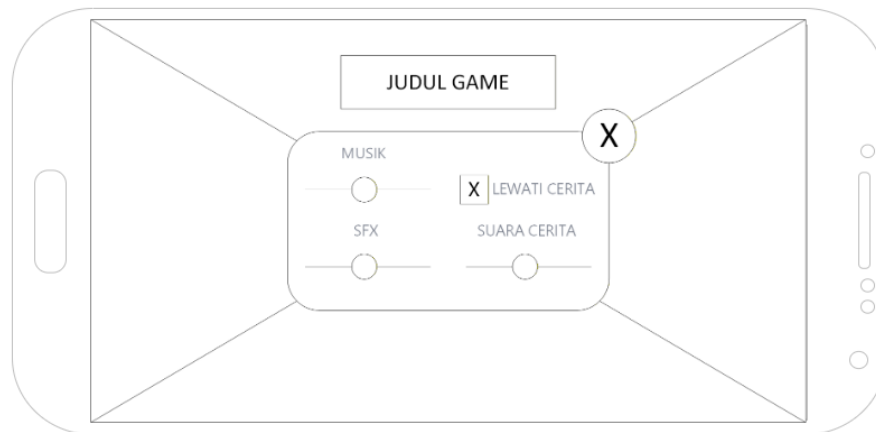
Pada rancangan antarmuka yang divisualisasikan dalam gambar tersebut, ditampilkan mekanisme konfirmasi keluar permainan melalui jendela dialog (*pop-up*) yang diposisikan secara sentral di tengah layar. Desain ini diterapkan untuk memusatkan fokus pengguna pada keputusan krusial sebelum mengakhiri sesi. Di dalam kotak dialog, dicantumkan pertanyaan verifikasi 'Apakah kamu ingin keluar dari permainan?' yang berfungsi sebagai langkah preventif terhadap penutupan aplikasi yang tidak disengaja (*accidental exit*). Guna mengakomodasi respons pengguna, disediakan dua opsi tombol interaktif yang diletakkan sejajar: tombol 'TIDAK' pada sisi kiri untuk membatalkan aksi, dan tombol 'IYA' pada sisi kanan untuk mengonfirmasi pengakhiran aplikasi.



Gambar 3. 3 Rancangan Layar Keluar Permainan

3.8.3 Rancangan Layar Menu Pengaturan

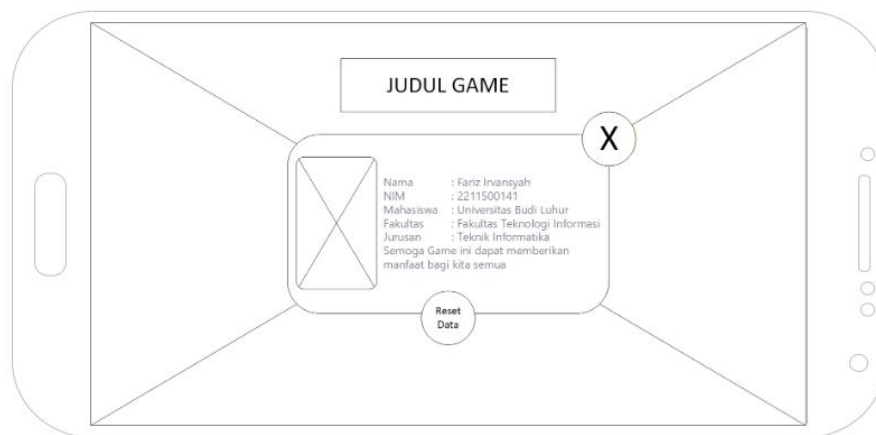
Pada rancangan antarmuka Menu Pengaturan, panel konfigurasi ditampilkan dalam format jendela *pop-up* yang terpusat di tengah layar, memungkinkan pengguna melakukan penyesuaian sistem tanpa kehilangan konteks visual utama permainan. Di dalam panel tersebut, diintegrasikan mekanisme kontrol audio berupa tiga bilah geser (*slider*) yang berfungsi untuk mengatur intensitas volume 'Musik', efek suara ('SFX'), dan 'Suara Cerita' secara independen. Selain pengaturan audio, fitur preferensi narasi diakomodasi melalui opsi 'Lewati Cerita' yang dioperasikan menggunakan elemen kotak centang (*checkbox*), memberikan fleksibilitas bagi pengguna untuk menonaktifkan segmen cerita dalam permainan. Guna menutup panel pengaturan dan kembali ke tampilan sebelumnya, sebuah tombol navigasi dengan simbol 'X' ditempatkan secara strategis pada sudut kanan atas panel.



Gambar 3. 4 Rancangan Layar Menu Pengaturan

3.8.4 Rancangan Layar Credits

Pada rancangan antarmuka Layar Kredit, informasi identitas pengembang dipresentasikan melalui jendela dialog yang ditempatkan secara terpusat pada layar. Struktur tampilan ini dibagi menjadi dua segmen utama: sisi kiri dialokasikan untuk bingkai foto profil, sedangkan sisi kanan memuat rincian biodata akademik pengembang (mencakup Nama, NIM, Institusi, Fakultas, dan Jurusan) serta pesan dedikasi. Selain informasi statis, diintegrasikan pula tombol fungsional 'Reset Data' pada bagian bawah panel yang bertujuan untuk mengembalikan kondisi permainan ke pengaturan awal. Mekanisme penutupan jendela difasilitasi oleh tombol navigasi bersimbol 'X' yang terletak pada sudut kanan atas panel.



Gambar 3. 5 Rancangan Layar Credits

3.8.5 Rancangan Layar Reset Data

Pada rancangan antarmuka untuk fitur Reset Data, mekanisme penghapusan progres permainan diamankan melalui jendela konfirmasi yang ditampilkan secara *pop-up* di tengah layar. Desain ini diterapkan untuk memastikan bahwa tindakan penghapusan data dilakukan secara sadar oleh pengguna. Di dalam kotak dialog, dicantumkan pertanyaan validasi 'Apakah kamu ingin menghapus progres saat ini?' guna mencegah risiko kehilangan data akibat ketidaksengajaan. Interaksi pengguna difasilitasi oleh dua opsi tombol yang diletakkan sejajar: tombol 'TIDAK' pada sisi kiri untuk membatalkan perintah, dan tombol 'IYA' pada sisi kanan untuk mengonfirmasi eksekusi penghapusan seluruh riwayat progres permainan.

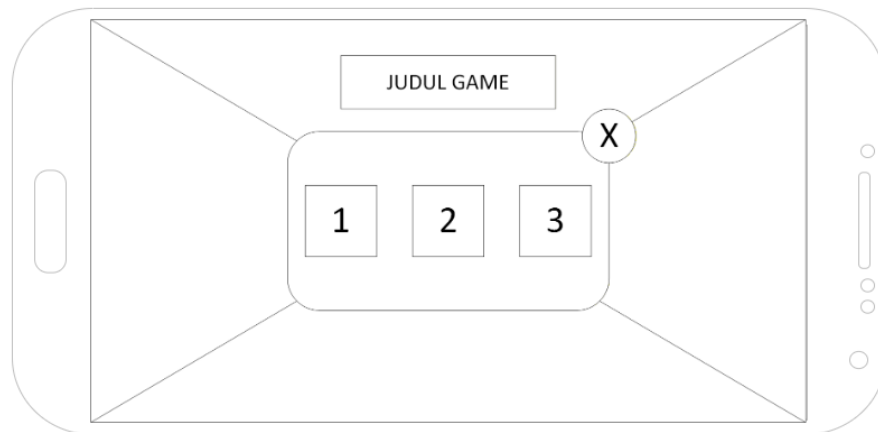


Gambar 3. 6 Rancangan Layar Reset Data

3.8.6 Rancangan Layar Menu Level

Pada rancangan antarmuka Menu Level, mekanisme pemilihan babak permainan dipresentasikan melalui jendela dialog (*pop-up*) yang muncul secara terpusat di tengah layar. Di dalam panel tersebut, disajikan tiga tombol persegi bernomor urut '1', '2', dan '3' yang disusun secara

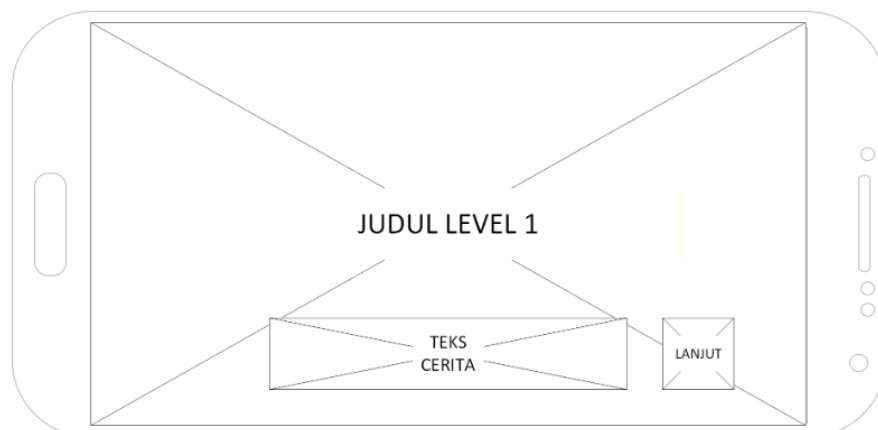
horizontal, di mana masing-masing tombol berfungsi sebagai akses pintas (*shortcut*) menuju level permainan yang dikehendaki. Identitas visual permainan tetap dipertahankan melalui penempatan label 'JUDUL GAME' pada bagian atas layar. Guna memberikan fleksibilitas navigasi bagi pengguna, tombol 'X' ditempatkan pada sudut kanan atas panel untuk menutup menu pemilihan level dan kembali ke tampilan sebelumnya.



Gambar 3. 7 Rancangan Layar Menu Level

3.8.7 Rancangan Layar Prolog Level 1

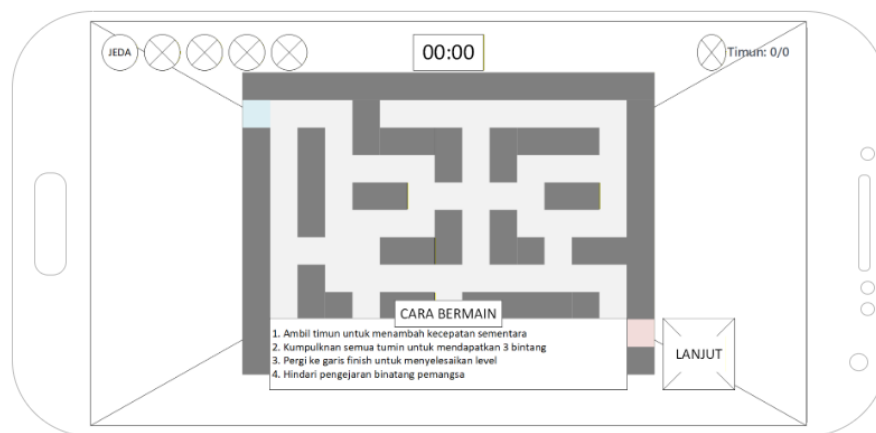
Pada rancangan antarmuka Prolog Level 1, elemen naratif diperkenalkan sebagai pengantar visual sebelum pemain memasuki sesi permainan inti. Identitas babak permainan ditampilkan secara prominen melalui label 'JUDUL LEVEL 1' yang diposisikan di tengah area layar. Mekanisme penyampaian cerita difasilitasi melalui panel teks horizontal yang terletak di bagian bawah layar (*bottom center*), yang dialokasikan khusus untuk memuat segmen 'TEKS CERITA'. Guna melanjutkan alur permainan, sebuah tombol navigasi interaktif bertuliskan 'LANJUT' ditempatkan di sisi kanan panel teks, yang berfungsi untuk memicu transisi dari sesi prolog menuju tampilan permainan utama (*gameplay*)



Gambar 3. 8 Rancangan Layar Prolog Level 1

3.8.8 Rancangan Layar Cara Bermain

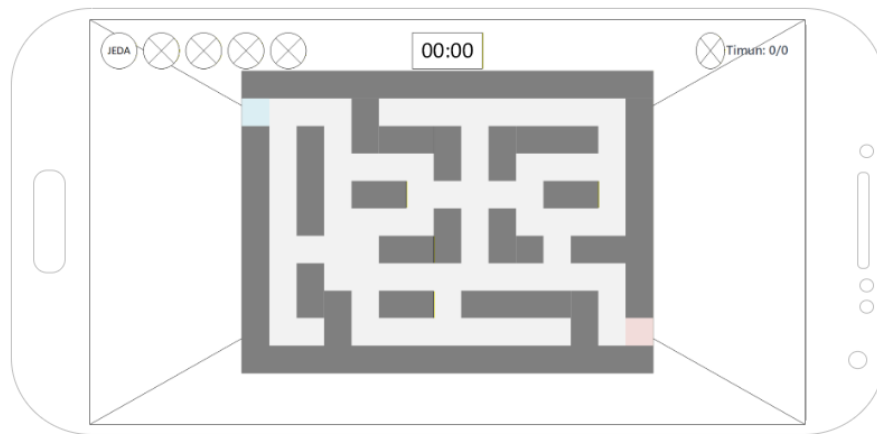
Pada rancangan antarmuka instruksi permainan, informasi panduan operasional disajikan melalui jendela *overlay* yang ditempatkan di atas tampilan area permainan (*gameplay area*). Metode ini diterapkan agar pengguna dapat memahami mekanisme permainan secara kontekstual tanpa meninggalkan lingkungan visual utama. Di dalam panel instruksi yang bertajuk 'CARA BERMAIN', dijabarkan empat poin aturan dasar secara berurutan, meliputi mekanisme pengambilan objek timun untuk peningkatan kecepatan (*power-up*), persyaratan pengumpulan timun untuk pencapaian bintang, objektif penyelesaian level pada garis finis, serta instruksi penghindaran dari kejaran predator. Setelah pemahaman instruksi tercapai, transisi menuju permainan inti difasilitasi oleh tombol navigasi 'LANJUT' yang diposisikan di sisi kanan bawah panel.



Gambar 3. 9 Rancangan Layar Cara Bermain

3.8.9 Rancangan Layar Level 1

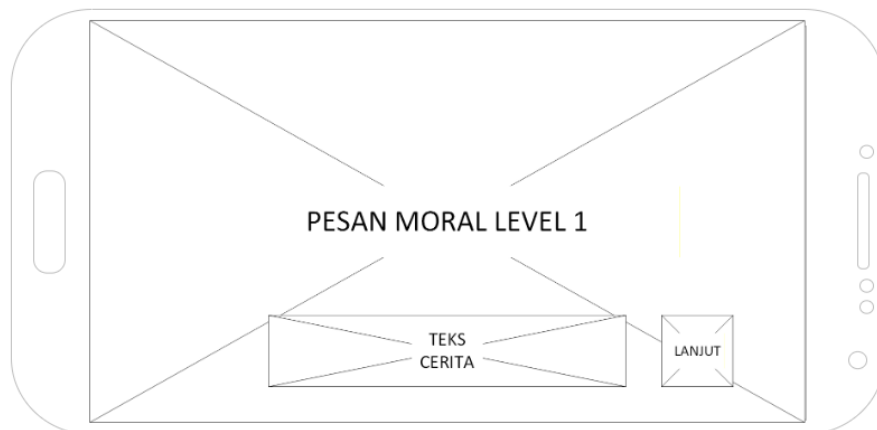
Pada rancangan antarmuka permainan Level 1, elemen visual utama difokuskan pada area labirin yang mendominasi sebagian besar layar sebagai arena interaksi pengguna. Informasi status permainan (*Heads-Up Display* atau HUD) didistribusikan secara linear di sepanjang bagian atas layar guna memudahkan pemantauan tanpa menghalangi pandangan terhadap labirin. Di sisi kiri atas, ditempatkan tombol 'JEDA' sebagai akses kontrol menu, yang diletakkan berdampingan dengan empat indikator status kesehatan (*health points*) berbentuk lingkaran. Informasi temporal dipantau melalui indikator waktu digital ('00:00') yang diposisikan secara sentral di bagian atas (*top-center*). Sementara itu, progres pencapaian objektif permainan divisualisasikan melalui penghitung item koleksi ('Timun: 0/0') yang berlokasi di sudut kanan atas layar.



Gambar 3. 10 Rancangan Layar Level 1

3.8.10 Rancangan Layar Epilog Level 1

Pada rancangan antarmuka Epilog Level 1, fase konklusi permainan disajikan melalui penyampaian nilai edukatif kepada pengguna. Fokus visual utama diarahkan pada label 'PESAN MORAL LEVEL 1' yang ditempatkan secara sentral di tengah layar sebagai penanda identitas segmen. Narasi penutup atau pesan moral tersebut dimuat dalam panel teks horizontal yang dialokasikan di bagian bawah layar (*bottom center*). Guna mengakhiri sesi level dan melanjutkan ke tahapan berikutnya, interaksi pengguna difasilitasi melalui tombol navigasi 'LANJUT' yang terletak di sisi kanan panel teks.

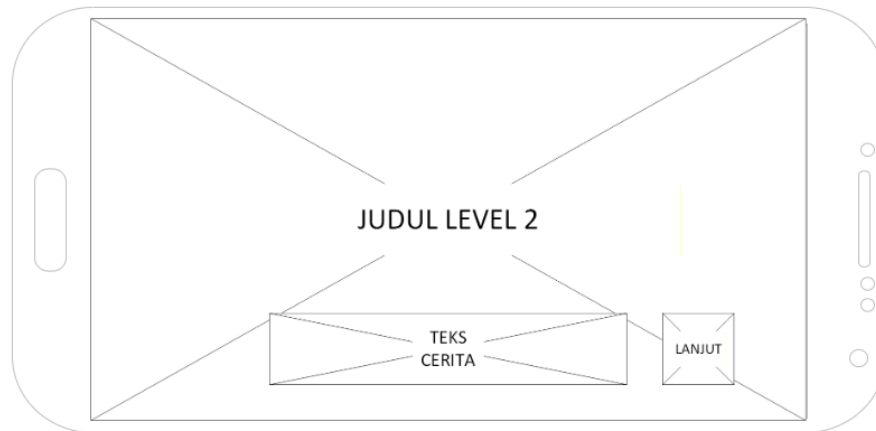


Gambar 3. 11 Rancangan Layar Epilog Level 1

3.8.11 Rancangan Layar Prolog Level 2

Pada rancangan antarmuka Prolog Level 2, struktur visual yang diterapkan memiliki konsistensi dengan pola desain prolog sebelumnya guna mempertahankan kontinuitas pengalaman pengguna. Identitas tahapan permainan ditandai melalui label 'JUDUL LEVEL 2' yang diposisikan secara terpusat di tengah layar (*center alignment*). Elemen naratif untuk kelanjutan cerita disajikan dalam panel teks horizontal yang

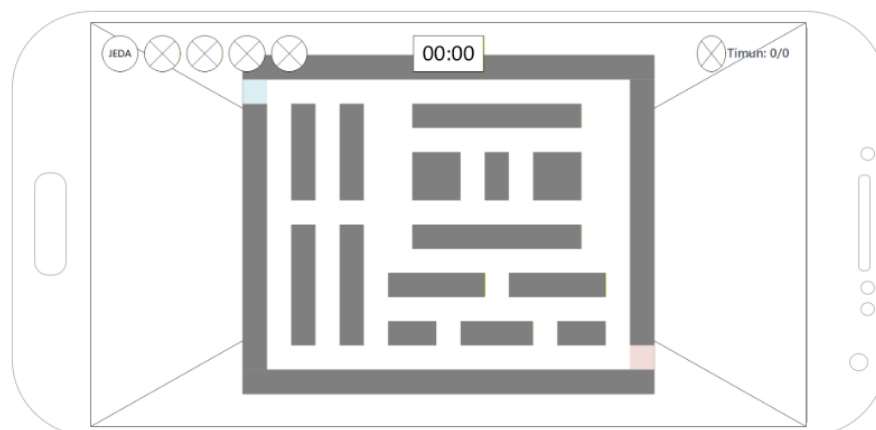
terletak di bagian bawah layar (*bottom center*). Sebagai mekanisme transisi menuju sesi permainan tingkat lanjut, tombol navigasi 'LANJUT' ditempatkan di sisi kanan panel teks untuk diakses oleh pengguna.



Gambar 3. 12 Rancangan Layar Prolog Level 2

3.8.12 Rancangan Layar Level 2

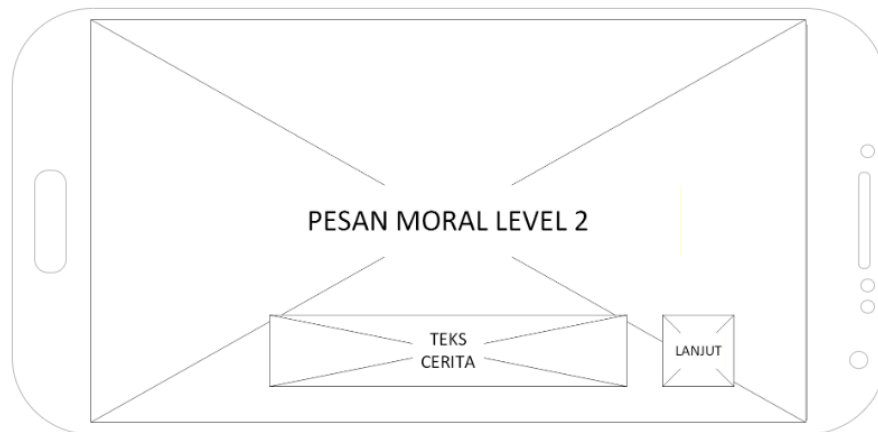
Pada rancangan antarmuka permainan Level 2, disajikan variasi struktur labirin yang berbeda dibandingkan level sebelumnya guna memberikan peningkatan tantangan spasial bagi pengguna. Konsistensi elemen visual tetap dipertahankan melalui penempatan komponen *Heads-Up Display* (HUD) di bagian atas layar, meliputi tombol navigasi 'JEDA' dan indikator kesehatan di sisi kiri, serta penghitung item koleksi 'Timun: 0/0' di sisi kanan. Pemantauan durasi permainan dilakukan melalui indikator waktu digital yang diletakkan secara sentral (*top-center*), sementara area permainan utama didominasi oleh susunan dinding labirin vertikal dan horizontal yang harus dinavigasi oleh pemain menuju titik akhir.



Gambar 3. 13 Rancangan Layar Level 2

3.8.13 Rancangan Layar Epilog Level 2

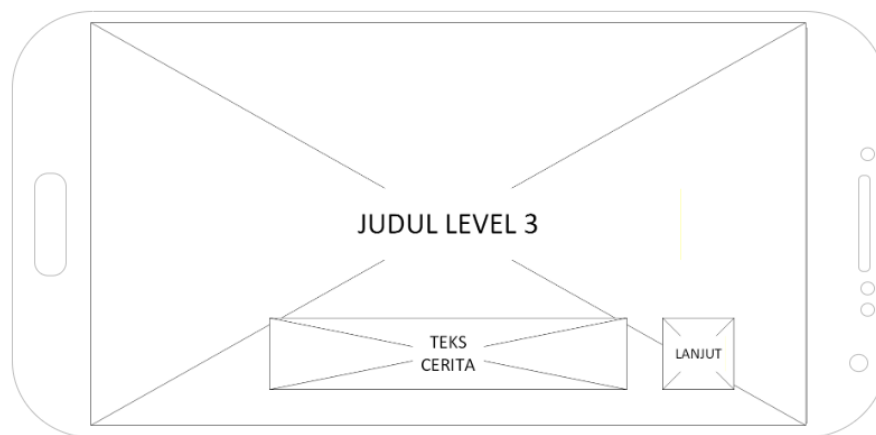
Pada rancangan antarmuka Epilog Level 2, fokus tampilan diarahkan pada penyampaian konklusi naratif yang bersifat edukatif kepada pengguna. Identitas visual segmen ini ditandai secara eksplisit melalui label 'PESAN MORAL LEVEL 2' yang ditempatkan di tengah layar sebagai pusat perhatian (*focal point*). Konten teks yang memuat nilai moral atau pesan cerita dialokasikan dalam panel horizontal di bagian bawah layar (*bottom center*). Guna memfasilitasi perpindahan menuju tahapan permainan selanjutnya atau kembali ke menu utama, tombol navigasi interaktif 'LANJUT' disediakan pada sisi kanan panel teks.



Gambar 3. 14 Rancangan Layar Epilog Level 2

3.8.14 Rancangan Layar Prolog Level 3

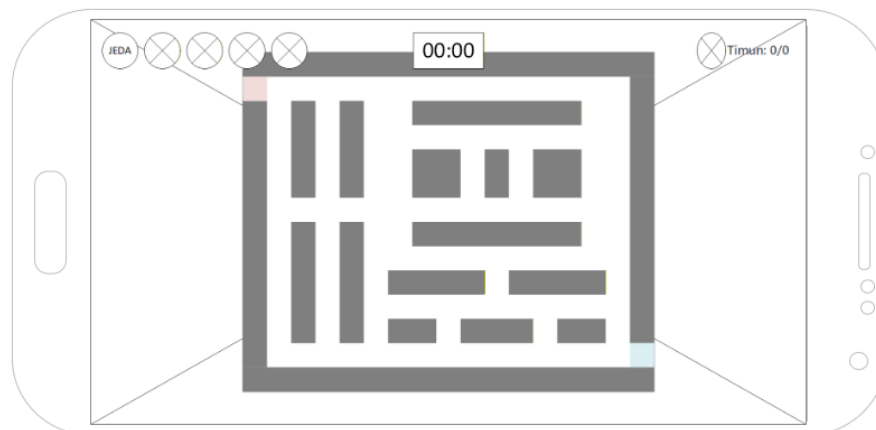
Pada rancangan antarmuka Prolog Level 3, pola tata letak visual dipertahankan konsisten dengan desain prolog pada level-level sebelumnya guna menjaga keseragaman pengalaman pengguna. Penanda identitas babak permainan divisualisasikan melalui teks 'JUDUL LEVEL 3' yang ditempatkan secara sentral di tengah layar. Media penyampaian narasi cerita dialokasikan pada panel teks horizontal yang terletak di bagian bawah layar (*bottom center*). Sebagai mekanisme transisi menuju sesi permainan tingkat lanjut, tombol navigasi 'LANJUT' ditempatkan di sisi kanan panel teks untuk diakses oleh pengguna.



Gambar 3. 15 Rancangan Layar Prolog Level 3

3.8.15 Rancangan Layar Level 3

Pada rancangan antarmuka permainan Level 3, ditampilkan konfigurasi area permainan yang didominasi oleh struktur labirin dengan kompleksitas jalur yang dirancang untuk tahap lanjut. Tata letak elemen *Heads-Up Display* (HUD) dipertahankan konsistensinya dengan level sebelumnya guna memastikan familiaritas pengguna terhadap kontrol permainan. Di bagian atas layar, instrumen pemantauan status meliputi tombol akses menu 'JEDA' dan empat indikator vitalitas karakter di sisi kiri, serta penghitung progres koleksi 'Timun: 0/0' di sisi kanan. Informasi durasi permainan disajikan melalui panel waktu digital yang terpusat di bagian atas, sementara interaksi utama pemain difokuskan pada navigasi karakter melintasi rintangan dinding labirin menuju titik target yang ditentukan.

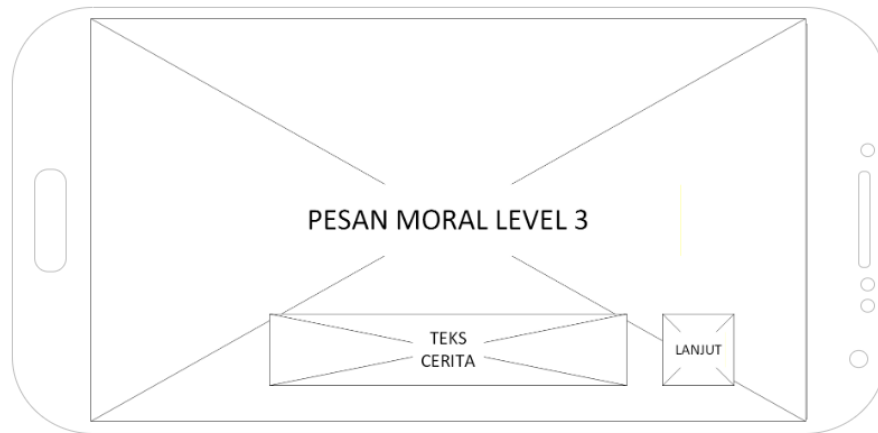


Gambar 3. 16 Rancangan Layar Level 3

3.8.16 Rancangan Layar Epilog Level 3

Pada rancangan antarmuka Epilog Level 3, tahap konklusi akhir permainan dipresentasikan melalui penyampaian pesan moral kepada pengguna. Identitas visual utama pada tampilan ini ditandai dengan label

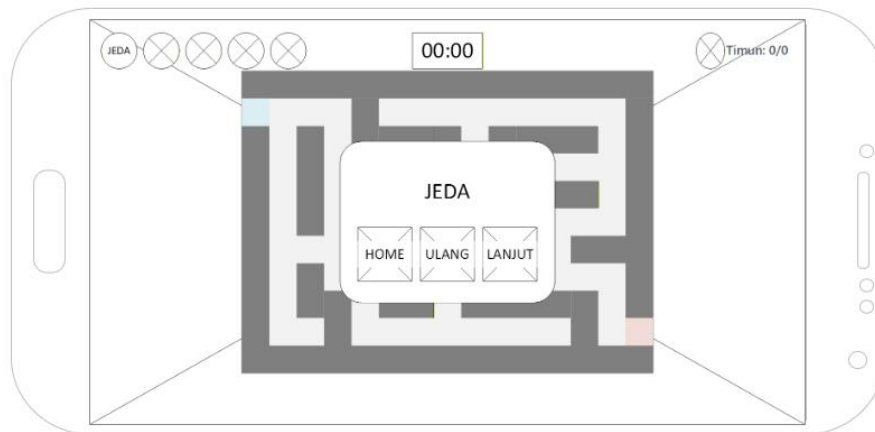
'PESAN MORAL LEVEL 3' yang diletakkan secara terpusat di tengah layar (*center alignment*). Narasi teks yang memuat nilai edukatif ditempatkan dalam panel horizontal di bagian bawah layar (*bottom center*). Guna mengakhiri alur cerita level ini dan melanjutkan ke sesi berikutnya, disediakan tombol navigasi interaktif 'LANJUT' di sisi kanan panel teks.



Gambar 3. 17 Rancangan Layar Epilog Level 3

3.8.17 Rancangan Layar Jeda Permainan

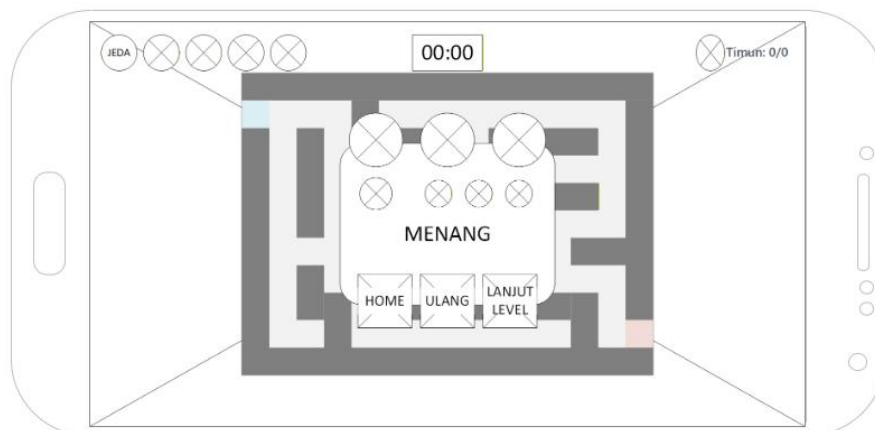
Pada rancangan antarmuka Menu Jeda, mekanisme penghentian sementara aktivitas permainan difasilitasi melalui tampilan jendela *overlay* yang diposisikan secara terpusat di tengah layar. Panel ini diberi label 'JEDA' dan dirancang untuk memberikan akses navigasi cepat tanpa menghilangkan konteks visual permainan yang sedang berlangsung. Di dalam menu tersebut, diintegrasikan tiga tombol operasional utama yang disusun secara horizontal: tombol 'HOME' di sisi kiri untuk kembali ke menu utama, tombol 'ULANG' di bagian tengah untuk memulai ulang level, dan tombol 'LANJUT' di sisi kanan untuk meneruskan sesi permainan. Selama panel ini aktif, area permainan (*maze*) tetap diperlihatkan di latar belakang guna mempertahankan orientasi visual pengguna terhadap progres level saat ini.



Gambar 3. 18 Rancangan Layar Jeda Permainan

3.8.18 Rancangan Layar Menang

Pada rancangan antarmuka Layar Menang, notifikasi keberhasilan penyelesaian level disampaikan kepada pengguna melalui jendela *overlay* yang ditempatkan secara terpusat di tengah area permainan. Status kemenangan ditekankan melalui label teks 'MENANG' yang disertai dengan visualisasi pencapaian berupa simbol bintang di bagian atas panel, yang merepresentasikan penilaian performa pemain. Guna menentukan tindak lanjut permainan, disediakan tiga opsi tombol navigasi yang disusun secara horizontal di bagian bawah panel: tombol 'HOME' untuk kembali ke menu utama, tombol 'ULANG' untuk memulai kembali level yang sama, dan tombol 'LANJUT LEVEL' untuk meneruskan progres menuju tantangan berikutnya.

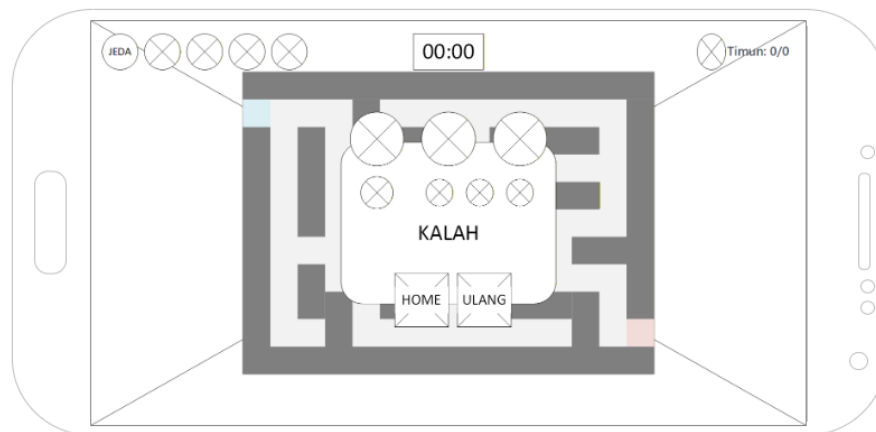


Gambar 3. 19 Rancangan Layar Menang

3.8.19 Rancangan Layar Kalah

Pada rancangan antarmuka Layar Kalah, notifikasi mengenai kegagalan dalam menyelesaikan level disampaikan kepada pengguna melalui jendela *overlay* yang muncul secara terpusat di tengah area permainan. Status kekalahan ditegaskan melalui label teks 'KALAH'

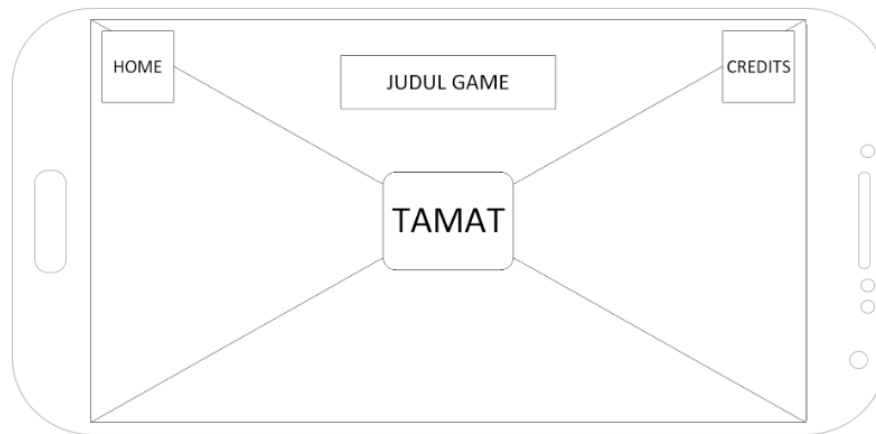
yang diposisikan di tengah panel sebagai indikator utama hasil permainan. Berbeda dengan tampilan kemenangan, pada layar ini hanya disediakan dua opsi navigasi utama untuk tindak lanjut pengguna: tombol 'HOME' di sisi kiri untuk kembali ke menu utama, dan tombol 'ULANG' di sisi kanan untuk mencoba kembali level yang gagal diselesaikan. Ketiadaan opsi untuk melanjutkan ke level berikutnya pada panel ini menegaskan batasan sistem bahwa progres hanya dapat diakses setelah objektif level terpenuhi.



Gambar 3. 20 Rancangan Layar Kalah

3.8.20 Rancangan Layar Tamat

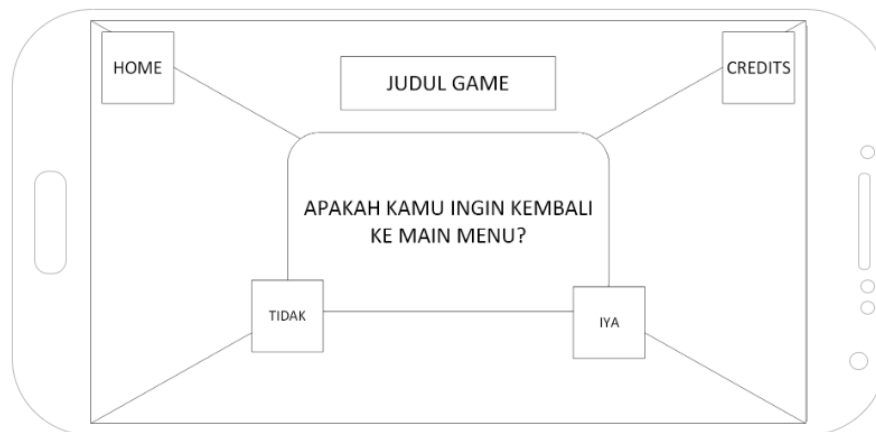
Pada rancangan antarmuka Layar Tamat, konklusi akhir dari keseluruhan alur permainan divisualisasikan secara sederhana namun jelas. Status penyelesaian permainan ditandai secara eksplisit melalui label teks 'TAMAT' yang ditempatkan dalam kotak di pusat layar (*center*), menandakan bahwa seluruh level telah berhasil diselesaikan. Identitas permainan tetap ditampilkan melalui label 'JUDUL GAME' pada bagian atas tengah layar. Guna memberikan opsi navigasi pasca-permainan, dua tombol interaktif ditempatkan pada sudut atas layar: tombol 'HOME' di sisi kiri atas untuk kembali ke menu utama, dan tombol 'CREDITS' di sisi kanan atas untuk mengakses informasi tim pengembang.



Gambar 3. 21 Rancangan Layar Tamat

3.8.21 Rancangan Layar Kembali Ke Menu Utama

Pada rancangan antarmuka konfirmasi kembali ke menu utama, mekanisme validasi navigasi diterapkan guna mencegah perpindahan halaman yang tidak disengaja oleh pengguna. Fokus interaksi diarahkan pada jendela dialog (*pop-up*) yang ditempatkan secara terpusat, yang memuat pertanyaan verifikasi 'Apakah kamu ingin kembali ke main menu?'. Di dalam panel tersebut, keputusan pengguna diakomodasi melalui dua tombol opsi: tombol 'TIDAK' di sisi kiri untuk membatalkan aksi dan tetap pada halaman saat ini, serta tombol 'IYA' di sisi kanan untuk mengonfirmasi perpindahan menuju tampilan awal permainan. Sementara itu, elemen navigasi sekunder seperti tombol 'HOME' dan 'CREDITS' tetap diperlihatkan pada sudut kiri dan kanan atas layar sebagai bagian dari struktur antarmuka latar belakang.



Gambar 3. 22 Rancangan Layar Kembali Ke Menu Utama

BAB IV

HASIL DAN PEMBAHASAN

4.1. Lingkungan Percobaan

Pada sub-bab ini, spesifikasi lingkungan percobaan yang diterapkan dalam penelitian pengembangan *game* Android 2D dengan implementasi algoritma A* (A-Star). Rincian mengenai perangkat keras (*hardware*) dan perangkat lunak (*software*). Diuraikan guna memberikan gambaran teknis mengenai kondisi pengujian dan memastikan hasil penelitian.

4.1.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan dalam penelitian ini diklasifikasikan menjadi dua kategori, yaitu perangkat pengembangan (*development*) dan perangkat pengujian (*testing*). Spesifikasi perangkat keras tersebut telah disesuaikan dengan kebutuhan komputasi pengembangan *game* 2D guna menjamin kelancaran proses implementasi dan pengujian.

a. Perangkat Pengembangan

Perangkat pengembangan difungsikan untuk proses perancangan, penulisan kode (*coding*), serta eksekusi pengujian tahap awal. Adapun spesifikasi teknis perangkat pengembangan yang digunakan adalah sebagai berikut:

- 1) Prosesor : *Intel Core i5* (Gen 8) 1.6 GHz
- 2) Memori (RAM) : 8 GB
- 3) Penyimpanan SSD : Ruang kosong tersedia minimal 50 GB
- 4) Sistem Operasi : Windows 11 (64-bit)

b. Perangkat Pengujian

Perangkat pengujian dimanfaatkan sebagai media eksekusi aplikasi hasil pengembangan (file.apk) pada lingkungan sistem operasi Android. Pengujian pada tahap ini dilakukan untuk memverifikasi kinerja dan responsivitas *game* pada perangkat seluler. Spesifikasi perangkat pengujian yang digunakan dirincikan sebagai berikut:

- 1) Perangkat : *Smartphone* Android
- 2) Memori (RAM) : 4 GB
- 3) Media Penyimpanan : Ruang kosong tersedia minimal 150 MB
- 4) Sistem Operasi : Android Versi 12
- 5) Resolusi Layar : Mendukung orientasi layar *landscape*

4.1.2 Perangkat Lunak (Software)

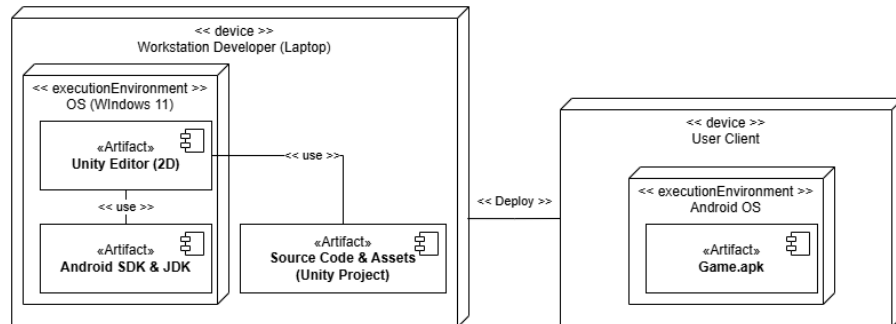
Serangkaian perangkat lunak dimanfaatkan untuk menunjang keseluruhan siklus pengembangan, mulai dari penyusunan aset, pemrograman, hingga kompilasi aplikasi Android. Perangkat lunak yang digunakan meliputi:

- a. *Unity 6.3 LTS (6000.3.2f1)*
- b. Visual Studio Code
- c. *Android Build Support*

- d. *Open JDK*
- e. *Android SDK & NDK Tools*

4.1.3 Deployment Diagram

Deployment diagram yang dirancang memvisualisasikan arsitektur implementasi sistem, mencakup spesifikasi perangkat keras dan lunak yang terlibat dalam proses pengembangan hingga distribusi ke pengguna akhir. Dalam diagram ini, dipetakan pemisahan lingkungan fisik antara stasiun kerja pengembang (*Workstation Developer*) dan perangkat pengguna (*User Client*).



Gambar 4. 1 Deployment Diagram

Lingkungan Pengembangan (*Development Environment*) Pada sisi kiri diagram, terdapat *node Workstation Developer* yang merepresentasikan perangkat keras laptop yang digunakan dalam proses produksi. *Node* ini dijalankan di atas lingkungan eksekusi (*execution environment*) sistem operasi Windows 11. Di dalamnya, terdapat komponen utama berupa *Unity Editor (2D)* yang berfungsi sebagai *engine* pengembangan permainan. Komponen ini memiliki ketergantungan (*dependency*) terhadap artefak *Android SDK & JDK* untuk keperluan kompilasi kode menjadi aplikasi Android. Selain itu, terdapat artefak *Source Code & Assets* yang merupakan kumpulan data mentah proyek *Unity* sebelum diproses menjadi aplikasi jadi.

Lingkungan Pengguna (*Client Environment*) Pada sisi kanan, didefinisikan *node User Client* yang merepresentasikan perangkat seluler yang digunakan oleh pengguna akhir (*end-user*). Lingkungan eksekusi pada *node* ini adalah *Android OS*, yang menjadi syarat utama agar aplikasi dapat beroperasi. Artefak tunggal yang terdapat pada lingkungan ini adalah *Game.apk*, yang merupakan hasil kompilasi final dari proyek yang siap dieksekusi oleh pengguna untuk memainkan permainan.

Mekanisme *Deployment* Hubungan antara kedua *node* utama dihubungkan oleh garis asosiasi dengan label *<< Deploy >>*. Hubungan ini mengindikasikan bahwa artefak *Game.apk* yang berada di perangkat pengguna merupakan hasil distribusi langsung dari proses *build* yang dilakukan pada *Workstation Developer*. Proses ini menegaskan alur implementasi sistem dari tahap pengembangan kode mentah hingga menjadi berkas aplikasi yang terinstalasi pada perangkat target.

4.2. Implementasi Metode

Tahapan dalam implementasi metode *Game Development Life Cycle* (GDLC) yang digunakan pada penelitian ini meliputi tahap *Initiation*, *Pre-Production*, *Production*, *Testing*, dan *Release*. Setiap tahapan memiliki peran penting dalam memastikan game yang dikembangkan sesuai dengan tujuan penelitian.

4.2.1 Initiation

Tahap *Initiation* merupakan tahap awal dalam pengembangan game yang berfokus pada penentuan konsep dasar permainan. Pada tahap ini dilakukan identifikasi ide permainan, tujuan pengembangan game, target platform, serta genre game yang akan dibuat.

Dalam penelitian ini, konsep *game* yang dikembangkan adalah *game* Android 2D bergenre *escape* labirin dengan tokoh utama Si Kancil. *Game* dirancang untuk memberikan tantangan kepada pemain melalui mekanisme pengumpulan item dan penghindaran musuh yang menerapkan algoritma A* sebagai kecerdasan buatan.

4.2.2 Pre-Production

Tahap *Pre-Production* merupakan tahap perancangan *game* secara detail sebelum masuk ke tahap implementasi. Pada tahap ini dilakukan perancangan alur permainan, desain level, desain karakter, serta perancangan sistem kecerdasan buatan.

Pada penelitian ini, tahap *Pre-Production* mencakup pembuatan desain labirin untuk setiap level, perancangan mekanisme permainan, serta perancangan penerapan algoritma A* sebagai kecerdasan buatan musuh. Selain itu, pada tahap ini juga dirancang alur cerita yang mengadaptasi cerita rakyat Petualangan Si Kancil sebagai elemen naratif dalam game.

a. Pembuatan Desain Karakter

Sebagai representasi visual utama, baik untuk tokoh utama maupun NPC yang berinteraksi dengan pemain, pemodelan karakter memegang peranan vital dalam permainan. Ini melibatkan pembuatan aset seperti tekstur dan desain 2D.

Tabel 4. 1 Tabel Pembuatan Desain Karakter

No.	Nama Aset	Deskripsi	Sumber
1	<i>Player</i>	Karakter Kancil yang berperan sebagai karakter utama dalam permainan	<i>Gemini Nano Banana</i>
2	<i>Enemy</i>	Karakter Harimau yang berperan sebagai musuh yang mengejar <i>Player</i> dalam permainan	<i>Gemini Nano Banana</i>

3	NPC	Karakter Rusa 3 yang berperan sebagai peranan tambahan dalam skenario level 3	<i>Gemini Nano Banana</i>
---	-----	---	---------------------------

b. Pembuatan Desain Labirin

Tahapan ini berfokus pada elemen visual dunia permainan, khususnya struktur bangunan dan dinding. Penggunaan rintangan labirin ini tidak hanya berfungsi sebagai tempat eksplorasi, tetapi juga sebagai ruang tantangan bagi pemain untuk menghindari rintangan dan menentukan strategi pelarian.

Tabel 4. 2 Tabel Pembuatan Desain Labirin

No.	Nama Aset	Deskripsi	Sumber
1	<i>Ground</i>	Aset lingkungan berupa pijakan yang dapat dilalui oleh <i>Player</i> dan <i>Enemy</i>	<i>Itchio</i>
2	<i>Wall</i>	Aset lingkungan berupa pembatas yang tidak dapat dilalui oleh <i>Player</i> dan <i>Enemy</i>	<i>Itchio</i>
3	<i>Finish</i>	Tampilan dengan teks <i>finish</i> sebagai tujuan penyelesaian level	<i>Unity</i>

c. Pembuatan Suara

Penelitian ini melalui tahapan peninjauan desain gim untuk memastikan kesesuaian antara visi artistik dan teknis. Proses ini dilakukan guna mengidentifikasi dan mendaftar seluruh kebutuhan audio, meliputi dialog, SFX, serta Foley.

Tabel 4. 3 Tabel Pembuatan Suara

No.	Nama Aset	Deskripsi	Sumber
1	BGM Menu Utama	Aset Audio pada scene menu utama	<i>Pixabay</i>
2	BGM Level 1	Aset Audio pada scene level 1	<i>Pixabay</i>
3	BGM Level 2	Aset Audio pada scene level 2	<i>Pixabay</i>
4	BGM Level 3	Aset Audio pada scene level 3	<i>Pixabay</i>
5	BGM Tamat	Aset Audio pada scene tamat	<i>Pixabay</i>
6	SFX Foley Kancil	Aset Audio untuk karakter kancil	<i>Itchio</i>

7	SFX Foley Harimau	Aset Audio untuk karakter harimau	<i>Itchio</i>
8	SFX Foley Rusa	Aset Audio untuk karakter rusa	<i>Itchio</i>
9	SFX Tombol	Aset Audio untuk tombol	<i>Itchio</i>
10	Suara Cerita Narator	Aset Audio untuk Prolog dan Epilog	<i>Elevenlabsio</i>
11	Suara Cerita Kancil	Aset Audio untuk Prolog dan Epilog	<i>Elevenlabsio</i>
12	Suara Cerita Harimau	Aset Audio untuk Prolog dan Epilog	<i>Elevenlabsio</i>
13	Suara Cerita Rusa	Aset Audio untuk Prolog dan Epilog	<i>Elevenlabsio</i>

d. Pembuatan Mekanisme Permainan

Tahapan ini membahas proses mendefinisikan aturan, sistem, dan interaksi dasar yang membentuk pengalaman bermain, meliputi pergerakan, pertarungan, teka-teki, tujuan, hingga sistem imbalan, yang bertujuan menciptakan pengalaman yang menarik, menantang, dan kohesif melalui tahapan seperti konsep, perumusan *gameplay*, prototipe, dan pengujian berulang, dengan melibatkan elemen seperti kuantitas, spasial, status, dan tindakan pemain

Tabel 4. 4 Tabel Pembuatan Mekanisme Permainan

No.	Elemen	Deskripsi
1	Aturan (<i>Rules</i>)	Batasan yang mengatur apa yang bisa dan tidak bisa dilakukan pemain
2	Aksi (<i>Actions</i>)	Tindakan yang dapat dilakukan pemain (misalnya melompat, menembak, berbicara)
3	Sistem (<i>System</i>)	Mekanisme yang lebih kompleks (misalnya sistem kesehatan, ekonomi, kemajuan level)
4	Umpan Balik (<i>Feedback</i>)	Respons permainan terhadap aksi pemain (skor, suara, visual)
5	Tujuan (<i>Goals</i>)	Apa yang harus dicapai pemain (menang, menyelesaikan level, mengumpulkan item)

e. Pembuatan Penerapan Algoritma A-Star (A*)

Pada tahap ini, dirancang logika dasar bagaimana algoritma A* akan bekerja dalam lingkungan permainan. Perancangan ini meliputi penentuan struktur data *Node* dan *Grid*, serta aturan pencarian jalur (*pathfinding*) yang akan diterapkan pada kecerdasan buatan musuh.

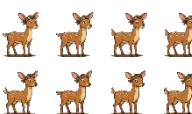
- Perancangan *Grid* dan *Node*: Lingkungan permainan dirancang sebagai matriks *grid* 2D. Setiap sel dalam *grid* didefinisikan sebagai sebuah *Node* yang memiliki atribut koordinat (x, y), status *walkable* (dapat dilewati) atau *unwalkable* (dinding/halangan), serta nilai *G-Cost*, *H-Cost*, dan *F-Cost*.
- Penentuan *Heuristik*: Metode perhitungan jarak yang dipilih untuk menghitung *H-Cost* adalah *Manhattan Distance*. Metode ini dirancang karena pergerakan musuh dalam *game* terbatas pada 4 arah utama (atas, bawah, kiri, kanan), tanpa pergerakan diagonal, sehingga *Manhattan Distance* memberikan estimasi biaya yang paling akurat dan efisien.
- Logika Pengejaran: Dirancang skema di mana posisi pemain (*Player*) akan selalu menjadi *Target Node* dan posisi musuh (*Enemy*) sebagai *Start Node*. Algoritma dirancang untuk melakukan kalkulasi ulang jalur (*re-calculate path*) secara *real-time* setiap kali pemain berpindah posisi dalam radius tertentu, memastikan musuh dapat mengejar secara dinamis.

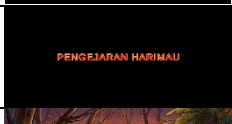
4.2.3 Production

Tahap *Production* merupakan tahap implementasi dari seluruh perancangan yang telah dilakukan sebelumnya. Pada tahap ini dilakukan pengembangan *game* menggunakan *Unity* sebagai *game engine* dengan bahasa pemrograman C#. Implementasi meliputi pembuatan *scene* permainan, penerapan *Tilemap* sebagai lingkungan labirin, pemrograman algoritma A* untuk mengatur pergerakan musuh, serta integrasi sistem kontrol pemain dan pengumpulan item. Pada tahap ini juga dilakukan pengintegrasian elemen cerita ke dalam *game* sesuai dengan rancangan yang telah dibuat. serta integrasi sistem kontrol pemain dan pengumpulan item. Proses produksi ini juga mencakup penyusunan aset visual (*sprite*) ke dalam *scene Unity*, penambahan komponen *Rigidbody2D* dan *BoxCollider2D* untuk deteksi tumbukan (*collision*), serta penulisan skrip C# untuk *Manager Script* yang mengatur alur *game loop* dari menu utama hingga *game over*.

Tabel 4. 5 Tabel Aset

No.	Aset	Keterangan	Sumber
1		Logo game	Gemini Nano Banana

No.	Aset	Keterangan	Sumber
2		Background Splash Screen	Gemini Nano Banana
3		Background Menu Utama	Gemini Nano Banana
4		Slider Musik, SFX dan Suara Cerita	Gemini Nano Banana
5		Tombol Home, Play, Restart dan Close	Gemini Nano Banana
6		Ikon Health Points dan bintang	Gemini Nano Banana
7		Ikon Start Game, Pengaturan, Credits, Exit dan Judul	Gemini Nano Banana
8		Tombol centang, close dan reset data	Gemini Nano Banana
9		Tombol jeda, 4 anak panah dan panel dialog	Gemini Nano Banana
10		Bingkai	Gemini Nano Banana
11		Item timun	Gemini Nano Banana
12		Karakter Utama Kancil	Gemini Nano Banana
13		Karakter Musuh Harimau	Gemini Nano Banana
14		Karakter NPC Rusa	Gemini Nano Banana

No.	Aset	Keterangan	Sumber
15		Prolog Level 1 Gambar 1	Unity
16		Prolog Level 1 Gambar 2	Gemini Nano Banana
17		Prolog Level 1 Gambar 3	Gemini Nano Banana
18		Epilog Level 1 Gambar 1	Gemini Nano Banana
19		Epilog Level 1 Gambar 2	Unity
20		Prolog Level 2 Gambar 1	Unity
21		Prolog Level 2 Gambar 2	Gemini Nano Banana
22		Epilog Level 2 Gambar 1	Gemini Nano Banana
23		Epilog Level 2 Gambar 2	Gemini Nano Banana
24		Epilog Level 2 Gambar 3	Gemini Nano Banana
25		Epilog Level 2 Gambar 4	Unity
26		Prolog Level 3 Gambar 1	Unity
27		Prolog Level 3 Gambar 2	Gemini Nano Banana

No.	Aset	Keterangan	Sumber
28		Epilog Level 2 Gambar 1	<i>Gemini Nano Banana</i>
29		Epilog Level 2 Gambar 2	<i>Gemini Nano Banana</i>
30		Epilog Level 2 Gambar 3	<i>Gemini Nano Banana</i>
31		Epilog Level 2 Gambar 4	<i>Unity</i>

4.2.4 Testing

Tahap Testing bertujuan untuk memastikan bahwa *game* yang dikembangkan berjalan sesuai dengan fungsionalitas yang diharapkan. Pengujian dilakukan untuk mengidentifikasi kesalahan (*bug*) serta memastikan bahwa algoritma A-Star (A*) berfungsi dengan baik dalam mengatur pergerakan musuh.

Pengujian pada penelitian ini dilakukan menggunakan metode pengujian fungsional (*black box testing*), dengan fokus pada pengujian fitur permainan, perilaku musuh, serta alur permainan pada level yang memiliki musuh.

4.2.5 Beta

Tahap Beta dilakukan setelah tahap testing internal (*Alpha*) selesai dan *bug* mayor telah diperbaiki. Pada tahap ini, versi *game* yang stabil didistribusikan kepada kelompok pengguna eksternal untuk mendapatkan umpan balik langsung mengenai pengalaman bermain.

Kegiatan yang dilakukan pada tahap Beta meliputi:

- Distribusi Aplikasi: *File* aplikasi (.apk) disebarkan kepada 16 orang responden yang terdiri dari target pengguna umum.
- Pengujian Pengguna (*User Acceptance Test*): Responden diminta untuk memainkan *game* "Petualangan Si Kancil" pada perangkat Android masing-masing untuk memastikan kompatibilitas dan kenyamanan kontrol.
- Pengumpulan Umpan Balik: Setelah bermain, responden mengisi kuesioner untuk menilai aspek visual, audio, kemudahan kontrol, dan tingkat kesulitan AI musuh. Hasil dari tahap ini dicatat sebagai data evaluasi akhir sebelum *game* memasuki tahap rilis resmi. Hasil detail dari pengujian pengguna ini dibahas lebih lanjut pada Sub-bab 4.5.4 (Pengujian Dari Pengguna).

4.2.6 Release

Tahap *Release* merupakan tahap akhir dalam metode GDLC, di mana *game* yang telah diuji dan dinyatakan berjalan dengan baik dipersiapkan untuk dirilis. Pada penelitian ini, tahap *Release* dilakukan dengan menghasilkan *file* aplikasi Android (.apk) sebagai hasil akhir penelitian.

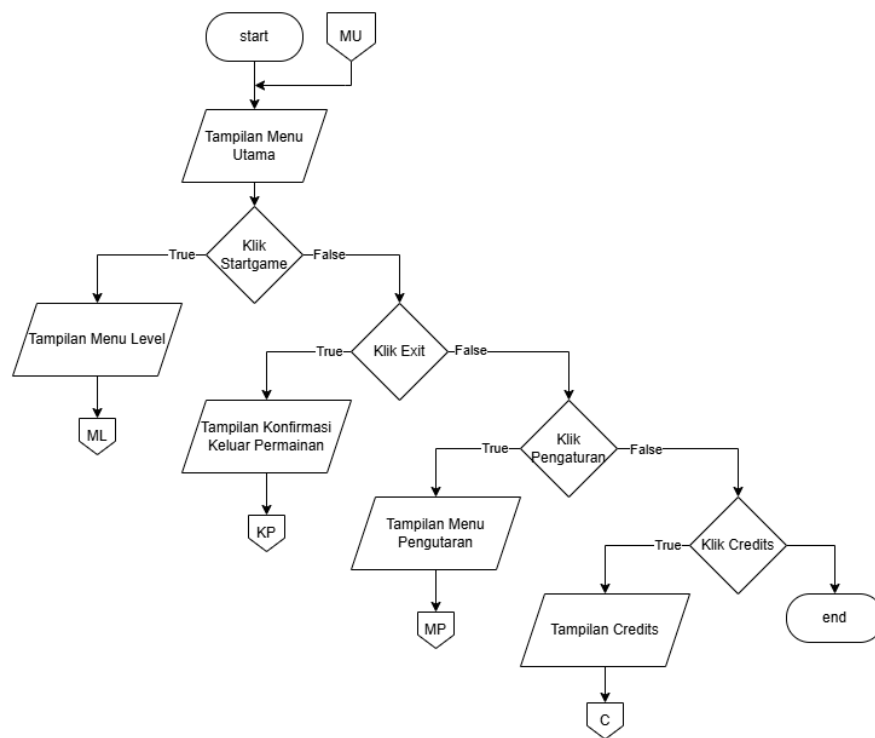
Game yang dihasilkan pada tahap ini digunakan sebagai objek evaluasi penelitian serta sebagai bukti implementasi penerapan algoritma A-Star (A*) dan integrasi cerita rakyat dalam pengembangan *game* Android 2D.

4.3. Flowchart

Flowchart untuk menggambarkan alur logika dan memberikan pemahaman yang lebih baik tentang suatu proses atau sistem secara visual. Flowchart ini bertujuan untuk memberikan pemahaman visual tentang bagaimana interaksi antara pengguna dan elemen-elemen yang ada di aplikasi.

4.3.1 Flowchart Menu Utama

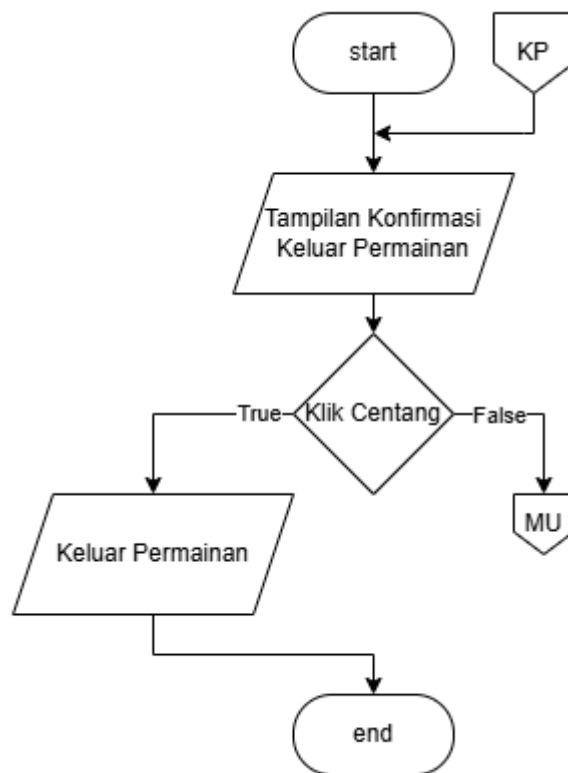
Perancangan alur kerja pada menu utama ini merepresentasikan logika navigasi antarmuka permainan Petualangan Si Kancil yang diinisial tepat setelah pengguna melewati tampilan *splash screen*. Proses dimulai dengan sistem menampilkan menu utama yang berfungsi sebagai gerbang interaksi pusat. Di dalam antarmuka ini, terdapat logika percabangan yang mendeteksi empat kemungkinan *input* dari pengguna. Pertama, apabila tombol *Start Game* ditekan, sistem akan memproses transisi menuju tampilan menu level yang ditandai dengan konektor ML. Kedua, jika pengguna memilih tombol *Exit*, sistem akan menampilkan halaman konfirmasi keluar permainan (konektor KP) terlebih dahulu sebagai langkah verifikasi sebelum menutup aplikasi. Ketiga, akses ke tombol Pengaturan akan mengarahkan sistem untuk membuka tampilan menu pengaturan (konektor MP) guna penyesuaian konfigurasi audio atau permainan. Terakhir, jika tombol *Credits* dipilih, sistem akan menyajikan informasi pengembang pada tampilan *credits* yang mengarah ke konektor C. Struktur logika ini memastikan setiap interaksi pengguna pada menu utama ditangani menuju sub-menu yang sesuai secara sistematis.



Gambar 4. 2 Flowchart Menu Utama

4.3.2 Flowchart Keluar Permainan

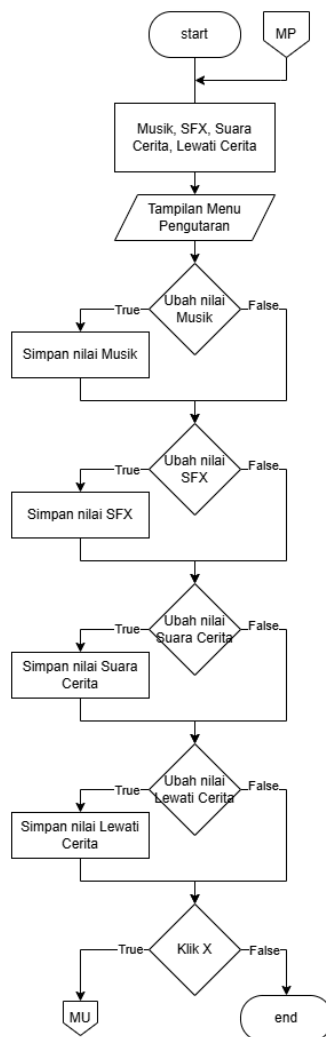
Perancangan alur logika pada tampilan konfirmasi keluar permainan ini diawali dari konektor KP yang dipicu ketika pengguna memilih opsi keluar pada menu sebelumnya. Sistem selanjutnya akan menampilkan halaman verifikasi untuk memastikan intensi pengguna sebelum benar-benar mengakhiri sesi permainan. Pada tahap ini, terjadi percabangan keputusan berdasarkan interaksi pengguna terhadap tombol centang. Apabila pengguna melakukan konfirmasi dengan menekan tombol centang, sistem akan memproses perintah untuk menutup aplikasi secara total dan mengakhiri program. Sebaliknya, jika pengguna memutuskan untuk membatalkan aksi tersebut atau jalur logika bernilai salah, sistem akan membatalkan proses terminasi dan mengarahkan navigasi kembali ke tampilan menu utama yang ditandai dengan konektor MU.



Gambar 4. 3 Flowchart Keluar Permainan

4.3.3 Flowchart Menu Pengaturan

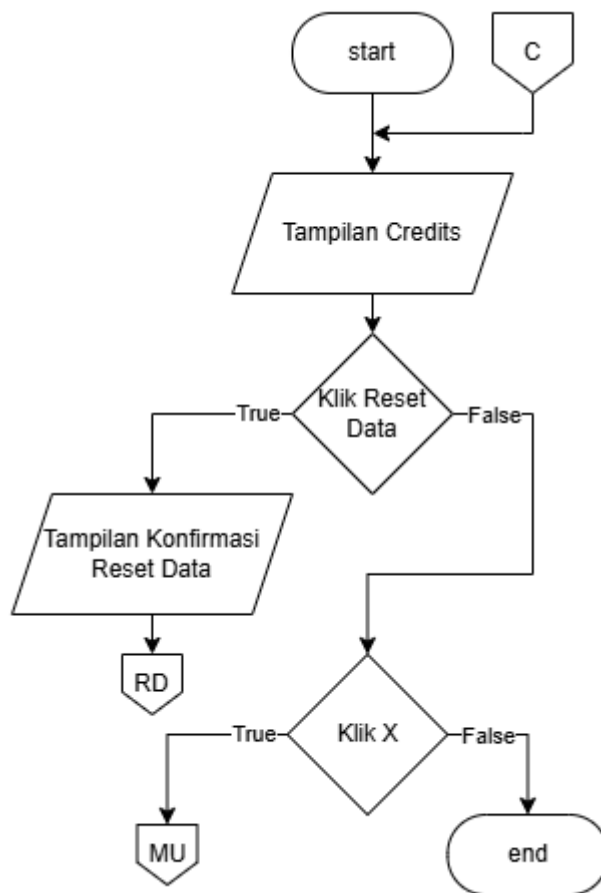
Perancangan alur logika pada tampilan menu pengaturan dimulai dari konektor MP, di mana sistem terlebih dahulu menginisialisasi parameter konfigurasi saat ini yang meliputi volume musik, efek suara (SFX), suara narasi cerita, serta status opsi melewati cerita. Setelah antarmuka pengaturan ditampilkan, sistem menjalankan mekanisme pemeriksaan interaksi pengguna secara sekuensial untuk mendeteksi perubahan nilai pada setiap elemen konfigurasi. Apabila pengguna melakukan penyesuaian pada *slider* musik, SFX, suara cerita, ataupun tombol opsi melewati cerita, sistem akan segera memperbarui dan menyimpan nilai preferensi tersebut ke dalam memori. Proses pengecekan ini bermuara pada interaksi tombol keluar (tombol X), jika pengguna menekan tombol tersebut, sistem akan mengakhiri sesi pengaturan dan mengembalikan navigasi ke tampilan menu utama melalui konektor MU, namun jika tidak ada interaksi keluar, sistem akan tetap mempertahankan tampilan pengaturan agar pengguna dapat terus melakukan penyesuaian.



Gambar 4. 4 Flowchart Menu Pengaturan

4.3.4 Flowchart Credits

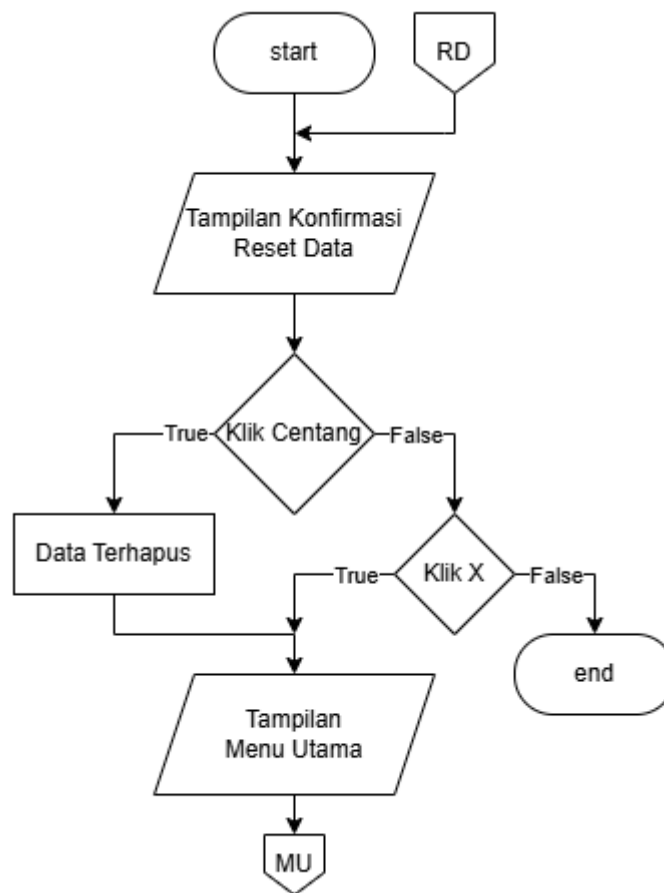
Perancangan alur logika pada tampilan *credits* ini bermula dari konektor C, di mana sistem akan menyajikan informasi terkait profil pengembang serta atribut aset permainan kepada pengguna. Dalam antarmuka ini, terdapat mekanisme percabangan untuk menangani dua kemungkinan *input* pengguna. Pertama, sistem mendeteksi interaksi pada tombol Reset Data, apabila tombol ini ditekan, alur navigasi akan dialihkan menuju halaman konfirmasi reset data yang direpresentasikan oleh konektor RD guna memverifikasi tindakan penghapusan progres permainan. Kedua, jika pengguna memilih tombol keluar (simbol X), sistem akan menutup halaman informasi ini dan mengembalikan pengguna ke tampilan menu utama melalui konektor MU. Apabila tidak ada interaksi pada kedua tombol tersebut, sistem akan tetap mempertahankan tampilan *credits* hingga pengguna memberikan perintah navigasi selanjutnya.



Gambar 4. 5 Flowchart Credits

4.3.5 Flowchart Reset Data

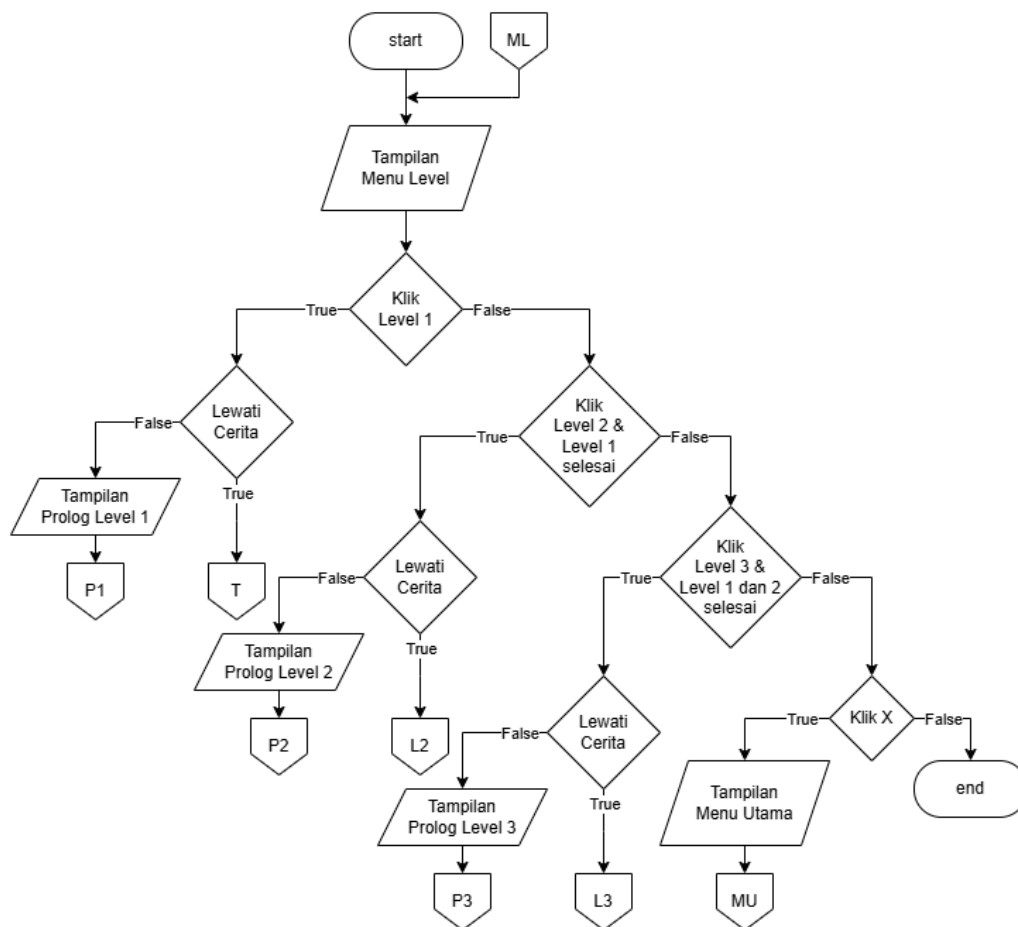
Perancangan alur logika pada tampilan konfirmasi reset data ini diawali dari konektor RD yang memicu munculnya jendela verifikasi guna memastikan keputusan pengguna. Pada antarmuka ini, sistem menangani dua kemungkinan interaksi utama terkait pengelolaan data permainan. Apabila pengguna menyetujui tindakan tersebut dengan menekan tombol centang, sistem akan segera mengeksekusi proses penghapusan data secara permanen, kemudian mengarahkan navigasi kembali ke tampilan menu utama melalui konektor MU. Sebaliknya, jika pengguna memilih untuk membatalkan operasi dengan menekan tombol simbol X, sistem akan membatalkan perintah penghapusan dan langsung mengembalikan pengguna ke menu utama tanpa melakukan perubahan apa pun pada data permainan. Mekanisme ini diterapkan untuk mencegah risiko kehilangan progres permainan akibat kesalahan *input* pengguna.



Gambar 4. 6 Flowchart Reset Data

4.3.6 Flowchart Menu Level

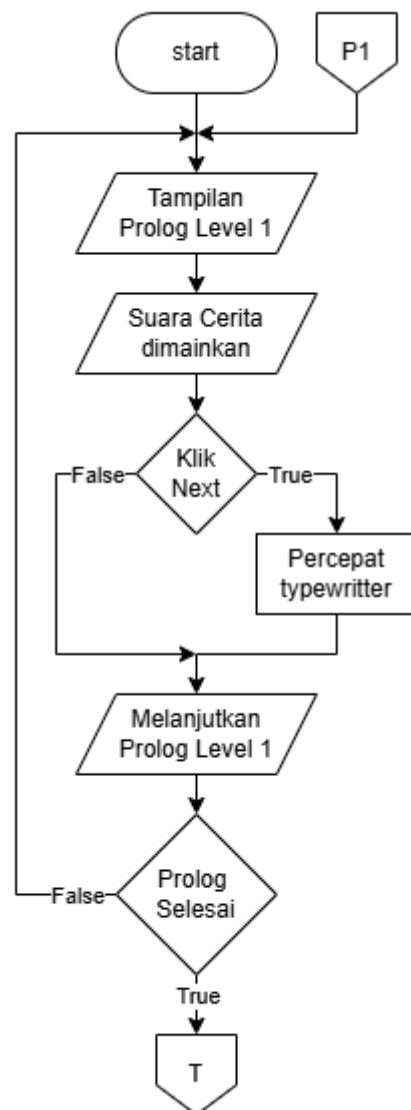
Perancangan alur kerja pada tampilan menu level ini merepresentasikan logika pemilihan tahapan permainan yang diinisialisasi melalui konektor ML. Dalam antarmuka ini, sistem menerapkan mekanisme validasi progresif di mana akses terhadap Level 2 dan Level 3 terkunci secara *default* dan hanya dapat dibuka apabila pengguna telah menyelesaikan level sebelumnya secara berurutan. Ketika pengguna memilih salah satu level yang telah terbuka, sistem akan menjalankan percabangan logika untuk memeriksa status opsi Lewati Cerita. Apabila fitur melewati cerita bernilai benar (*true*), sistem akan langsung mengarahkan navigasi ke area permainan atau tutorial yang ditandai dengan konektor T, L2, atau L3. Sebaliknya, jika fitur tersebut tidak aktif (*false*), sistem akan terlebih dahulu memuat tampilan prolog narasi yang sesuai dengan level tersebut (konektor P1, P2, atau P3). Selain itu, terdapat mekanisme navigasi keluar melalui tombol X yang akan mengembalikan pengguna ke tampilan menu utama melalui konektor MU.



Gambar 4. 7 Flowchart Menu Level

4.3.7 Flowchart Prolog Level 1

Perancangan alur logika pada tampilan prolog level 1 ini dimulai dari konektor P1, di mana sistem secara otomatis memuat antarmuka visual prolog dan memainkan audio narasi cerita secara bersamaan. Mekanisme utama dalam tampilan ini menggunakan struktur perulangan untuk menampilkan teks cerita secara bertahap (*typewriter effect*). Di dalam siklus perulangan tersebut, sistem terus memantau interaksi pengguna melalui tombol *Next*, apabila tombol ini ditekan, sistem akan mempercepat efek pengetikan teks agar cerita tampil lebih cepat, sedangkan jika tidak ada interaksi, teks akan muncul dengan kecepatan normal. Setelah teks pada segmen tersebut selesai ditampilkan sepenuhnya, sistem akan melakukan pengecekan kondisi akhir prolog. Jika seluruh rangkaian prolog telah selesai, sistem akan mengarahkan pengguna ke tampilan cara bermain melalui konektor T, namun jika prolog belum usai, logika alur akan kembali ke awal perulangan untuk memuat segmen cerita selanjutnya.

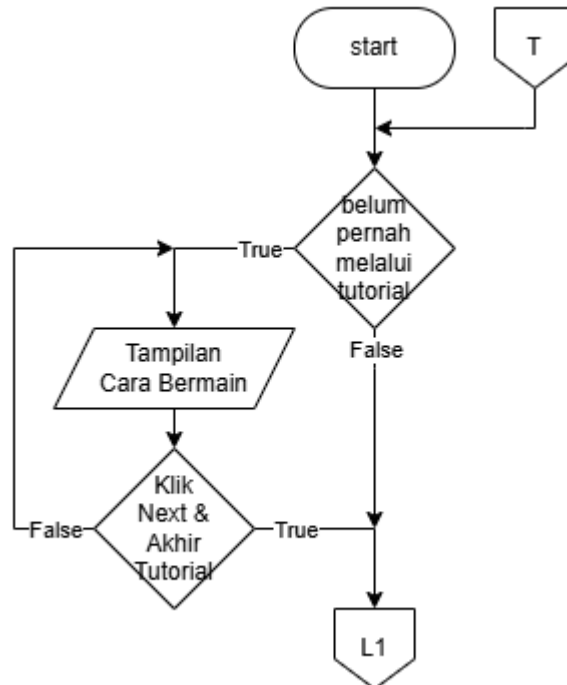


Gambar 4. 8 Flowchart Prolog Level 1

4.3.8 Flowchart Cara Bermain

Perancangan alur logika pada tampilan cara bermain ini diinisialisasi melalui konektor T yang menghubungkan antarmuka dari sesi prolog sebelumnya. Pada tahap awal, sistem melakukan validasi otomatis melalui percabangan logika untuk memeriksa riwayat aktivitas pengguna, spesifiknya apakah pengguna belum pernah menyelesaikan sesi tutorial sebelumnya. Apabila kondisi ini bernilai benar, sistem akan menyajikan antarmuka panduan permainan secara interaktif kepada pengguna. Di dalam fase ini, diterapkan mekanisme perulangan (*looping*) yang menunggu *input* pengguna pada tombol *Next* untuk menavigasi setiap halaman instruksi hingga mencapai akhir tutorial. Setelah pengguna menyelesaikan seluruh rangkaian panduan, alur navigasi akan diarahkan menuju level 1 melalui konektor L1. Sebaliknya,

jika sistem mendeteksi bahwa pengguna telah menyelesaikan tutorial sebelumnya (kondisi bernilai salah), maka tampilan panduan ini akan dilewati secara otomatis dan sistem langsung meneruskan pengguna ke permainan utama melalui konektor L1.



Gambar 4. 9 Flowchart Cara Bermain

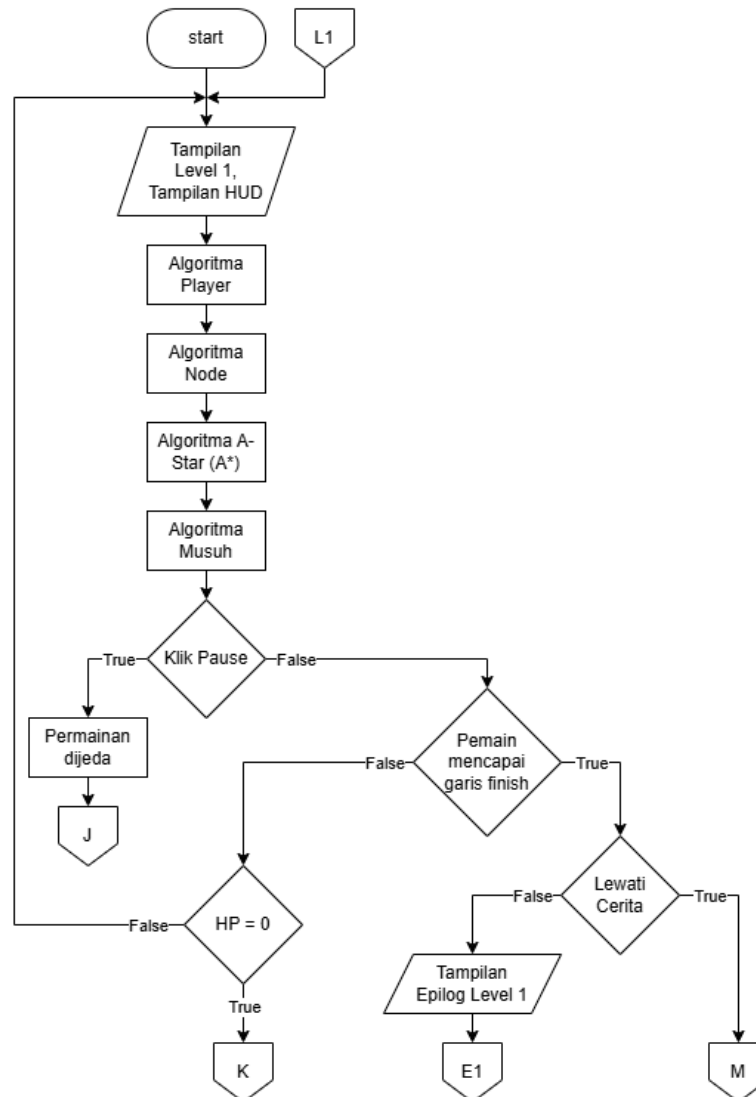
4.3.9 Flowchart Level 1

Perancangan alur logika pada permainan level 1 ini diawali dari konektor L1. Pada tahap inisialisasi, sistem akan segera memuat tampilan visual level beserta elemen antarmuka pengguna atau *Heads-Up Display* (HUD). Setelah tampilan siap, inti dari mekanisme permainan mulai berjalan dalam sebuah siklus berulang yang mengeksekusi serangkaian algoritma secara berurutan. Proses ini meliputi algoritma *player* untuk menangani kontrol karakter, algoritma *node* untuk pemetaan area, algoritma A-Star (A*) untuk pencarian jalur cerdas, serta algoritma musuh untuk mengatur perilaku pengejaran terhadap pemain.

Selama siklus permainan aktif, sistem melakukan pengecekan terhadap interaksi tombol jeda terlebih dahulu. Apabila pengguna menekan tombol pause, maka status permainan akan dihentikan sementara dan alur logika dialihkan menuju menu jeda yang ditandai dengan konektor J. Namun jika tidak ada interaksi pada tombol jeda, sistem akan melanjutkan proses ke pengecekan kondisi kemenangan karakter.

Pada tahap penentuan kondisi akhir, sistem memverifikasi apakah pemain telah berhasil mencapai garis *finish*. Jika kondisi kemenangan ini terpenuhi, sistem akan memeriksa status opsi Lewati Cerita. Apabila fitur

lewati cerita aktif, navigasi akan langsung diarahkan ke tampilan menang melalui konektor M. Sebaliknya, jika fitur tersebut tidak aktif, sistem akan menampilkan epilog level 1 terlebih dahulu melalui konektor E1. Jika pemain belum mencapai garis *finish*, sistem akan memeriksa kondisi *Health Point* (HP). Apabila nilai HP habis atau sama dengan nol, alur akan diarahkan ke kondisi kalah atau *game over* pada konektor K. Jika HP masih tersedia, siklus logika akan kembali ke awal untuk memproses bingkai permainan selanjutnya



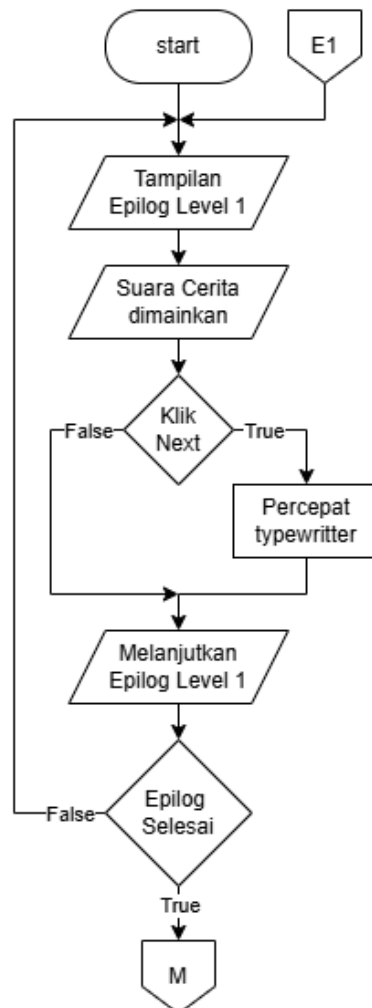
Gambar 4. 10 Flowchart Level 1

4.3.10 Flowchart Epilog Level 1

Perancangan alur logika pada tampilan epilog level 1 ini diinisialisasi melalui konektor E1, di mana sistem segera memuat antarmuka visual epilog bersamaan dengan pemutaran audio narasi

cerita. Dalam penyajian teksnya, sistem menggunakan efek animasi pengetikan (*typewriter*) yang berjalan beriringan dengan mekanisme pemantauan *input* pengguna secara *real-time*. Apabila pengguna menekan tombol *Next*, sistem akan merespons dengan mempercepat laju animasi pengetikan agar teks cerita muncul lebih instan, namun jika tidak ada interaksi, teks akan tetap ditampilkan dengan kecepatan normal mengikuti ritme narasi yang sedang berlangsung.

Siklus logika ini terus berlanjut ke tahap penampilan segmen cerita berikutnya dan diakhiri dengan proses validasi kondisi penyelesaian epilog. Pada tahap ini, sistem memeriksa apakah seluruh rangkaian narasi telah selesai ditampilkan kepada pengguna. Jika kondisi selesai belum terpenuhi, alur logika akan kembali ke awal siklus perulangan untuk memuat bagian cerita selanjutnya. Sebaliknya, apabila seluruh epilog telah usai, sistem akan menghentikan perulangan tersebut dan mengarahkan navigasi pengguna menuju tampilan menang yang ditandai dengan konektor M.



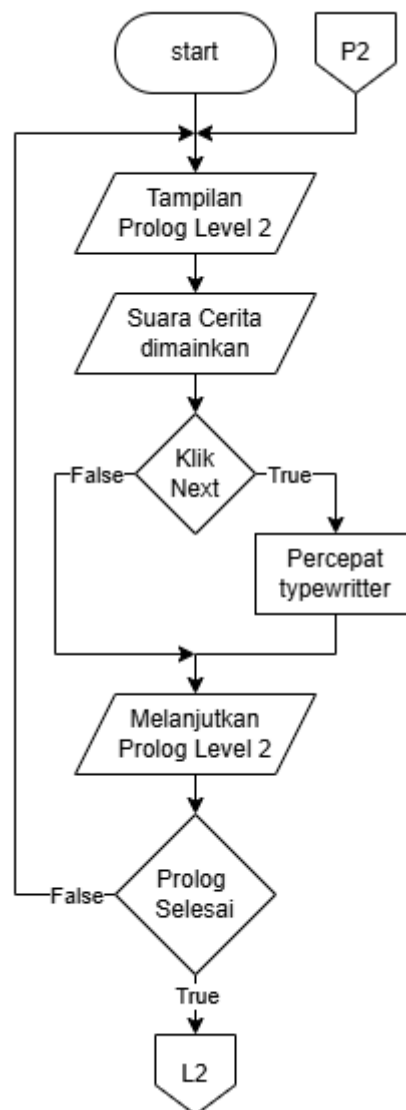
Gambar 4. 11 Flowchart Epilog Level 1

4.3.11 Flowchart Prolog Level 2

Perancangan alur logika pada tampilan prolog level 2 ini diinisialisasi melalui konektor P2. Pada tahap awal, sistem secara otomatis memuat antarmuka visual prolog yang berjalan beriringan dengan pemutaran audio narasi cerita. Mekanisme penyajian teks pada bagian ini menerapkan efek animasi pengetikan atau *typewriter effect* untuk menyelaraskan tampilan visual teks dengan suara narator yang sedang dimainkan.

Selama proses penuturan cerita berlangsung, sistem menerapkan logika perulangan yang secara aktif memantau interaksi pengguna terhadap tombol *Next*. Apabila pengguna menekan tombol tersebut, sistem akan merespons dengan mempercepat durasi animasi pengetikan agar teks tampil lebih singkat. Namun jika tidak ada interaksi pada tombol *Next*, teks akan tetap muncul dengan kecepatan normal sesuai dengan pengaturan awal.

Setelah menangani interaksi pengguna, sistem melanjutkan proses untuk menampilkan kelanjutan isi cerita pada prolog tersebut. Pada akhir setiap siklus logika, sistem melakukan validasi untuk memeriksa apakah seluruh segmen cerita prolog telah selesai ditampilkan. Jika kondisi selesai belum terpenuhi, alur logika akan kembali ke awal siklus untuk memuat kalimat berikutnya. Sebaliknya, apabila seluruh narasi telah usai, sistem akan menghentikan perulangan tersebut dan mengarahkan navigasi pengguna menuju level 2 melalui konektor L2.



Gambar 4. 12 Flowchart Prolog Level 2

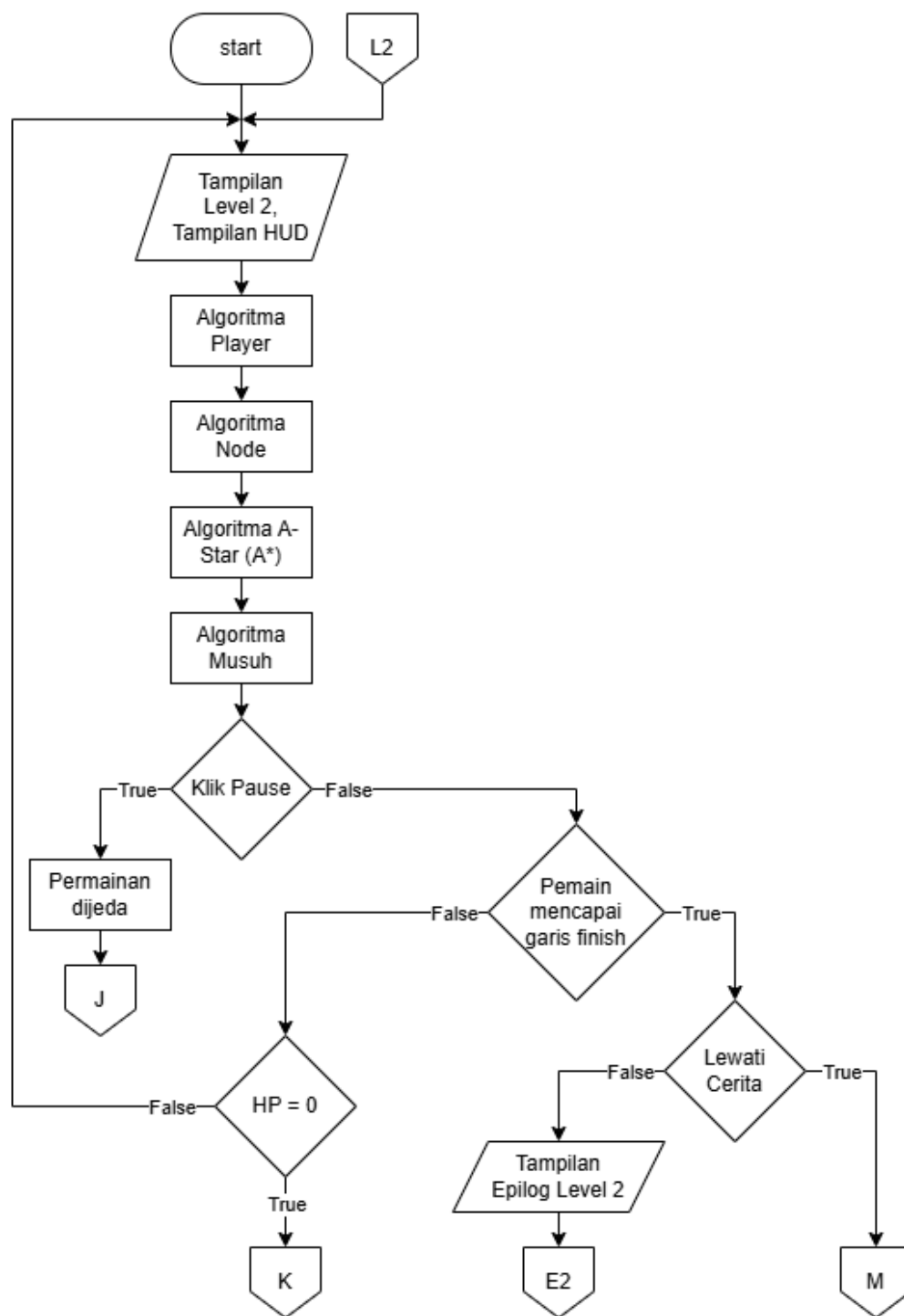
4.3.12 Flowchart Level 2

Perancangan alur logika pada permainan level 2 ini diinisialisasi melalui konektor L2. Pada tahap awal, sistem segera memuat tampilan visual lingkungan level 2 beserta elemen antarmuka pengguna atau *Heads-Up Display* (HUD). Setelah inisialisasi visual selesai, mekanisme inti permainan mulai berjalan dalam sebuah siklus berulang yang mengeksekusi serangkaian algoritma secara sekuensial. Proses ini meliputi algoritma *player* untuk kontrol karakter, algoritma *node* untuk pemetaan area, algoritma A-Star (A*) untuk navigasi cerdas, serta algoritma musuh untuk perilaku pengejaran.

Selama siklus permainan berlangsung, sistem secara aktif memantau interaksi tombol jeda dari pengguna. Apabila tombol pause ditekan, status permainan akan dihentikan sementara dan alur navigasi dialihkan

menuju menu jeda yang ditandai dengan konektor J. Namun jika tidak ada interaksi pada tombol jeda, sistem akan melanjutkan proses validasi ke tahap pengecekan kondisi kemenangan pemain.

Pada tahap penentuan kondisi akhir ini, sistem memverifikasi apakah pemain telah berhasil mencapai garis finish. Jika kondisi kemenangan terpenuhi, sistem akan memeriksa status opsi Lewati Cerita. Apabila fitur tersebut aktif, navigasi akan langsung diarahkan ke tampilan menang melalui konektor M. Sebaliknya, jika fitur lewati cerita tidak aktif, sistem akan memuat tampilan epilog level 2 terlebih dahulu melalui konektor E2. Jika pemain belum mencapai garis finish, sistem akan memeriksa kondisi Health Point (HP). Apabila nilai HP habis atau sama dengan nol, alur akan diarahkan ke kondisi kalah pada konektor K. Jika HP masih tersedia, siklus logika akan kembali ke awal untuk memproses bingkai permainan berikutnya.

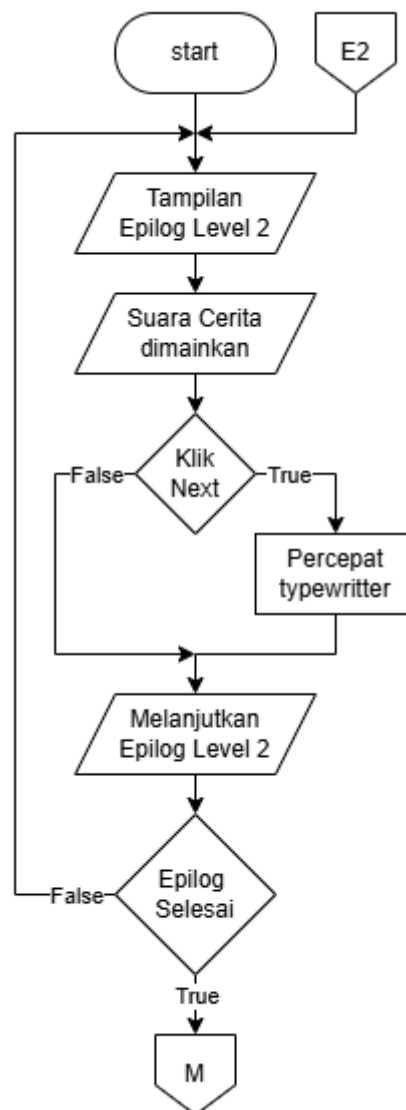


4.3.13 Flowchart Epilog Level 2

Perancangan alur logika pada tampilan epilog level 2 ini diinisialisasi melalui konektor E2. Pada tahap awal, sistem segera memuat antarmuka visual epilog yang berjalan beriringan dengan pemutaran audio narasi cerita. Mekanisme penyajian teks pada bagian

ini menerapkan efek animasi pengetikan atau *typewriter effect* untuk menyelaraskan tampilan visual teks dengan suara narator. Selama proses penuturan cerita berlangsung, sistem secara aktif memantau interaksi pengguna terhadap tombol *Next*. Apabila pengguna menekan tombol tersebut, sistem akan merespons dengan mempercepat durasi animasi pengetikan agar teks tampil lebih instan, namun jika tidak ada interaksi, teks akan tetap muncul dengan kecepatan normal.

Siklus logika ini terus berlanjut ke tahap penampilan segmen cerita berikutnya dan diakhiri dengan proses validasi kondisi penyelesaian epilog. Pada tahap ini, sistem memeriksa apakah seluruh rangkaian narasi telah selesai ditampilkan kepada pengguna. Jika kondisi selesai belum terpenuhi, alur logika akan kembali ke awal siklus perulangan untuk memuat bagian cerita selanjutnya. Sebaliknya, apabila seluruh epilog telah usai, sistem akan menghentikan perulangan tersebut dan mengarahkan navigasi pengguna menuju tampilan menang yang ditandai dengan konektor M.



Gambar 4. 13 Flowchart Epilog Level 2

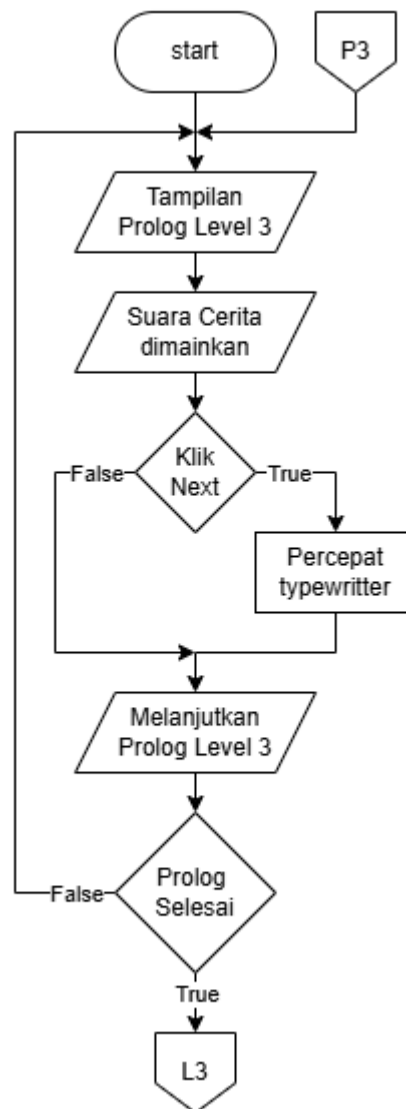
4.3.14 Flowchart Prolog Level 3

Perancangan alur logika pada tampilan prolog level 3 ini diinisialisasi melalui konektor P3. Pada tahap awal, sistem secara otomatis memuat antarmuka visual prolog yang berjalan beriringan dengan pemutaran audio narasi cerita. Mekanisme penyajian teks pada bagian ini menerapkan efek animasi pengetikan atau *typewriter effect* untuk menyelaraskan tampilan visual teks dengan suara narator yang sedang dimainkan.

Selama proses penuturan cerita berlangsung, sistem menerapkan logika perulangan yang secara aktif memantau interaksi pengguna terhadap tombol *Next*. Apabila pengguna menekan tombol tersebut, sistem akan merespons dengan mempercepat durasi animasi pengetikan agar teks tampil lebih singkat. Namun jika tidak ada interaksi pada

tombol *Next*, teks akan tetap muncul dengan kecepatan normal sesuai dengan pengaturan awal.

Setelah menangani interaksi pengguna, sistem melanjutkan proses untuk menampilkan kelanjutan isi cerita pada prolog tersebut. Pada akhir setiap siklus logika, sistem melakukan validasi untuk memeriksa apakah seluruh segmen cerita prolog telah selesai ditampilkan. Jika kondisi selesai belum terpenuhi, alur logika akan kembali ke awal siklus untuk memuat kalimat berikutnya. Sebaliknya, apabila seluruh narasi telah usai, sistem akan menghentikan perulangan tersebut dan mengarahkan navigasi pengguna menuju level 3 melalui konektor L3.



Gambar 4. 14 Flowchart Prolog Level 3

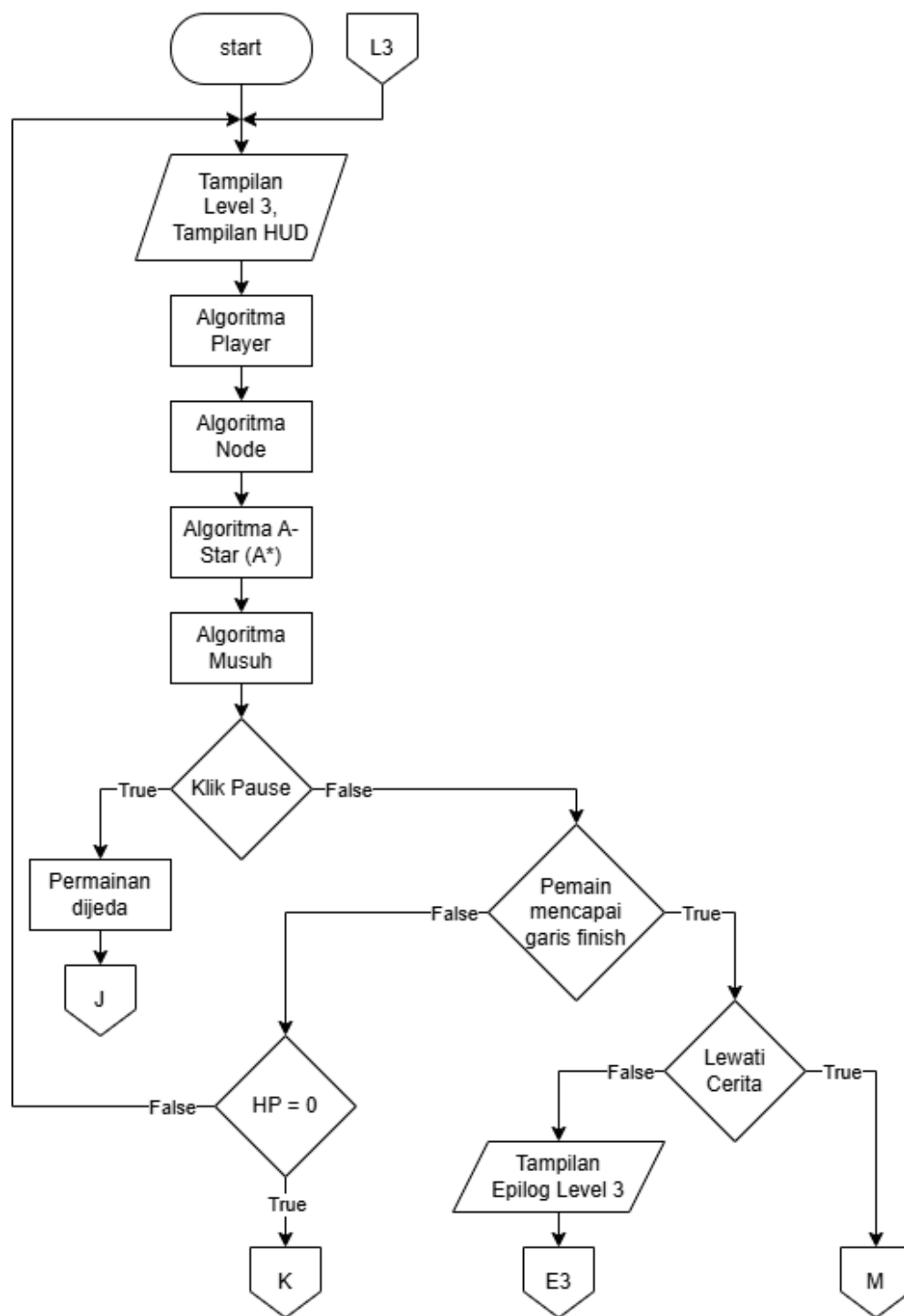
4.3.15 Flowchart Level 3

Perancangan alur logika pada permainan level 3 ini diinisialisasi melalui konektor L3. Pada tahap awal, sistem segera memuat tampilan

visual lingkungan level 3 beserta elemen antarmuka pengguna atau *Heads-Up Display* (HUD). Setelah inisialisasi visual selesai, mekanisme inti permainan mulai berjalan dalam sebuah siklus berulang yang mengeksekusi serangkaian algoritma secara sekuensial. Proses ini meliputi algoritma player untuk kontrol karakter, algoritma *node* untuk pemetaan area, algoritma A-Star (A*) untuk navigasi cerdas, serta algoritma musuh untuk perilaku pengejaran.

Selama siklus permainan berlangsung, sistem secara aktif memantau interaksi tombol jeda dari pengguna. Apabila tombol pause ditekan, status permainan akan dihentikan sementara dan alur navigasi dialihkan menuju menu jeda yang ditandai dengan konektor J. Namun jika tidak ada interaksi pada tombol jeda, sistem akan melanjutkan proses validasi ke tahap pengecekan kondisi kemenangan pemain.

Pada tahap penentuan kondisi akhir ini, sistem memverifikasi apakah pemain telah berhasil mencapai garis *finish*. Jika kondisi kemenangan terpenuhi, sistem akan memeriksa status opsi Lewati Cerita. Apabila fitur tersebut aktif, navigasi akan langsung diarahkan ke tampilan menang melalui konektor M. Sebaliknya, jika fitur lewati cerita tidak aktif, sistem akan memuat tampilan epilog level 3 terlebih dahulu melalui konektor E3. Jika pemain belum mencapai garis *finish*, sistem akan memeriksa kondisi *Health Point* (HP). Apabila nilai HP habis atau sama dengan nol, alur akan diarahkan ke kondisi kalah pada konektor K. Jika HP masih tersedia, siklus logika akan kembali ke awal untuk memproses bingkai permainan berikutnya.



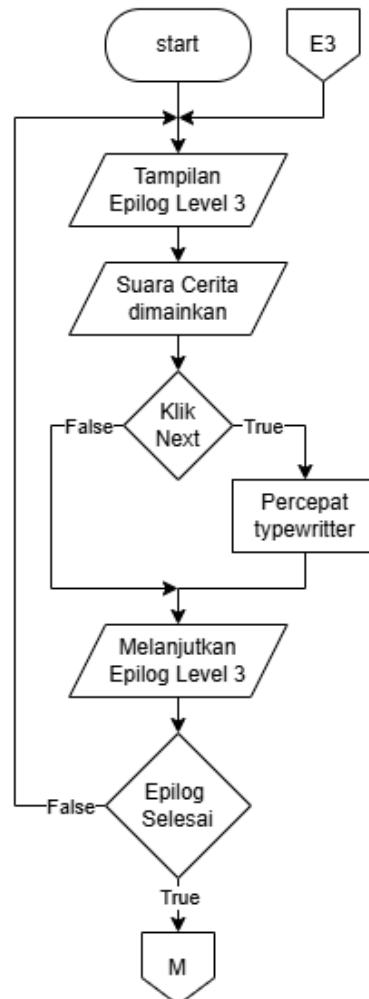
Gambar 4. 15 Flowchart Level 3

4.3.16 Flowchart Epilog Level 3

Perancangan alur logika pada tampilan epilog level 3 ini diinisialisasi melalui konektor E3. Pada tahap awal, sistem segera memuat antarmuka visual epilog yang berjalan beriringan dengan

pemutaran audio narasi cerita. Mekanisme penyajian teks pada bagian ini menerapkan efek animasi pengetikan atau *typewriter effect* untuk menyelaraskan tampilan visual teks dengan suara narator. Selama proses penuturan cerita berlangsung, sistem secara aktif memantau interaksi pengguna terhadap tombol *Next*. Apabila pengguna menekan tombol tersebut, sistem akan merespons dengan mempercepat durasi animasi pengetikan agar teks tampil lebih instan, namun jika tidak ada interaksi, teks akan tetap muncul dengan kecepatan normal.

Siklus logika ini terus berlanjut ke tahap penampilan segmen cerita berikutnya dan diakhiri dengan proses validasi kondisi penyelesaian epilog. Pada tahap ini, sistem memeriksa apakah seluruh rangkaian narasi telah selesai ditampilkan kepada pengguna. Jika kondisi selesai belum terpenuhi, alur logika akan kembali ke awal siklus perulangan untuk memuat bagian cerita selanjutnya. Sebaliknya, apabila seluruh epilog telah usai, sistem akan menghentikan perulangan tersebut dan mengarahkan navigasi pengguna menuju tampilan menang yang ditandai dengan konektor M.



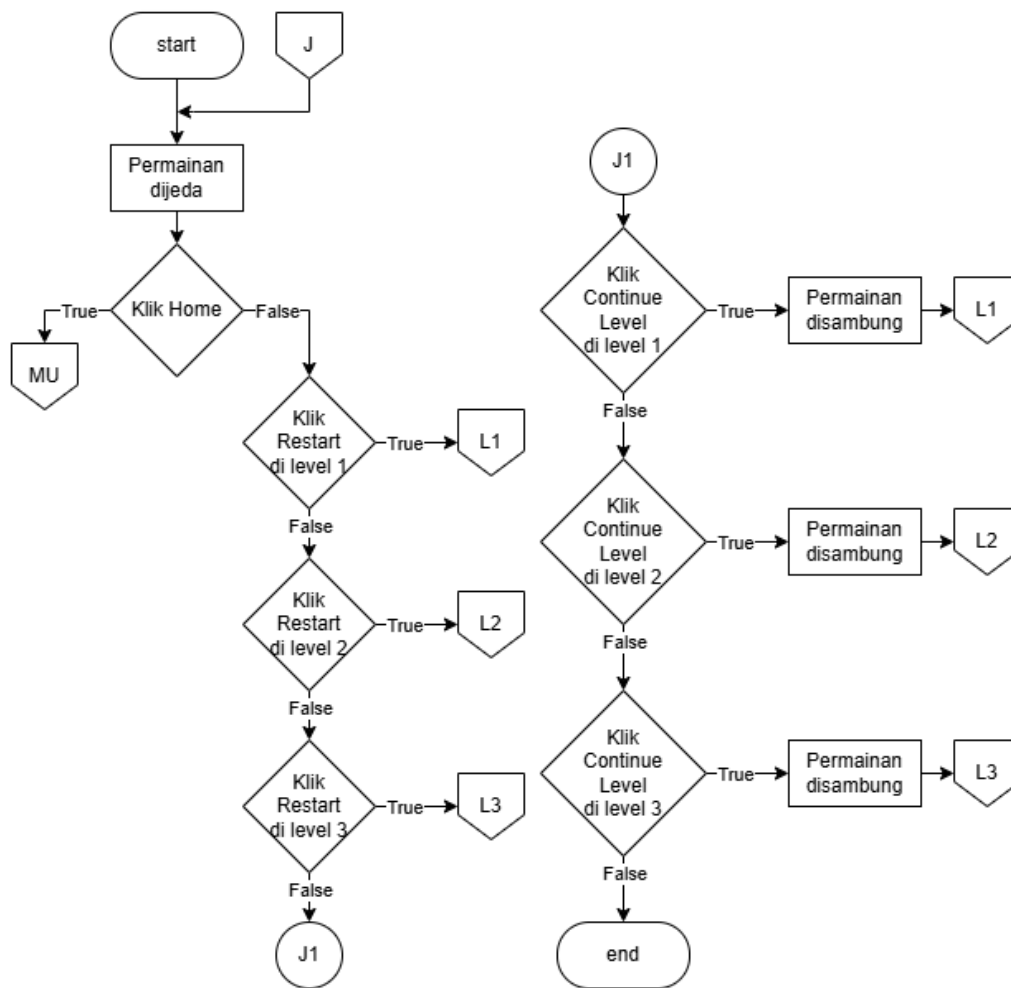
Gambar 4. 16 Flowchart Epilog Level 3

4.3.17 Flowchart Jeda Permainan

Perancangan alur logika pada tampilan menu jeda ini diinisialisasi melalui konektor J yang dipicu ketika pengguna menekan tombol pause selama permainan berlangsung. Pada kondisi ini, sistem secara otomatis membekukan status permainan sementara waktu untuk memberikan ruang interaksi kepada pengguna. Dalam antarmuka jeda tersebut, sistem menyediakan opsi navigasi utama berupa tombol *Home*. Apabila tombol ini ditekan, sistem akan menghentikan sesi permainan sepenuhnya dan mengarahkan pengguna kembali ke tampilan menu utama yang ditandai dengan konektor MU.

Selain opsi kembali ke menu utama, sistem juga menangani logika untuk memulai ulang permainan atau *restart* berdasarkan level yang sedang aktif. Mekanisme validasi dilakukan secara berurutan untuk memeriksa level mana yang sedang dimainkan oleh pengguna. Jika pengguna memilih tombol *restart* saat berada di level 1, alur akan diarahkan kembali ke inisialisasi level 1 melalui konektor L1. Logika yang sama berlaku untuk level 2 dan level 3, di mana sistem akan mengarahkan navigasi ke konektor L2 atau L3 sesuai dengan posisi level pengguna saat itu. Jika tidak ada interaksi *restart*, alur logika berlanjut ke konektor penghubung J1 untuk memproses opsi melanjutkan permainan.

Pada tahap selanjutnya setelah konektor J1, sistem memeriksa *input* untuk melanjutkan permainan atau *continue*. Serupa dengan logika *restart*, sistem melakukan verifikasi terhadap konteks level saat ini. Apabila pengguna menekan tombol *continue* di level 1, sistem akan memulihkan status permainan yang sebelumnya dibekukan dan melanjutkan sesi di level 1. Proses validasi ini berulang untuk level 2 dan level 3, yang masing-masing akan mengembalikan pengguna ke permainan melalui konektor L2 dan L3. Mekanisme ini memastikan bahwa pengguna dapat kembali ke permainan dengan status yang tepat tanpa kehilangan konteks level yang sedang dijalankan.



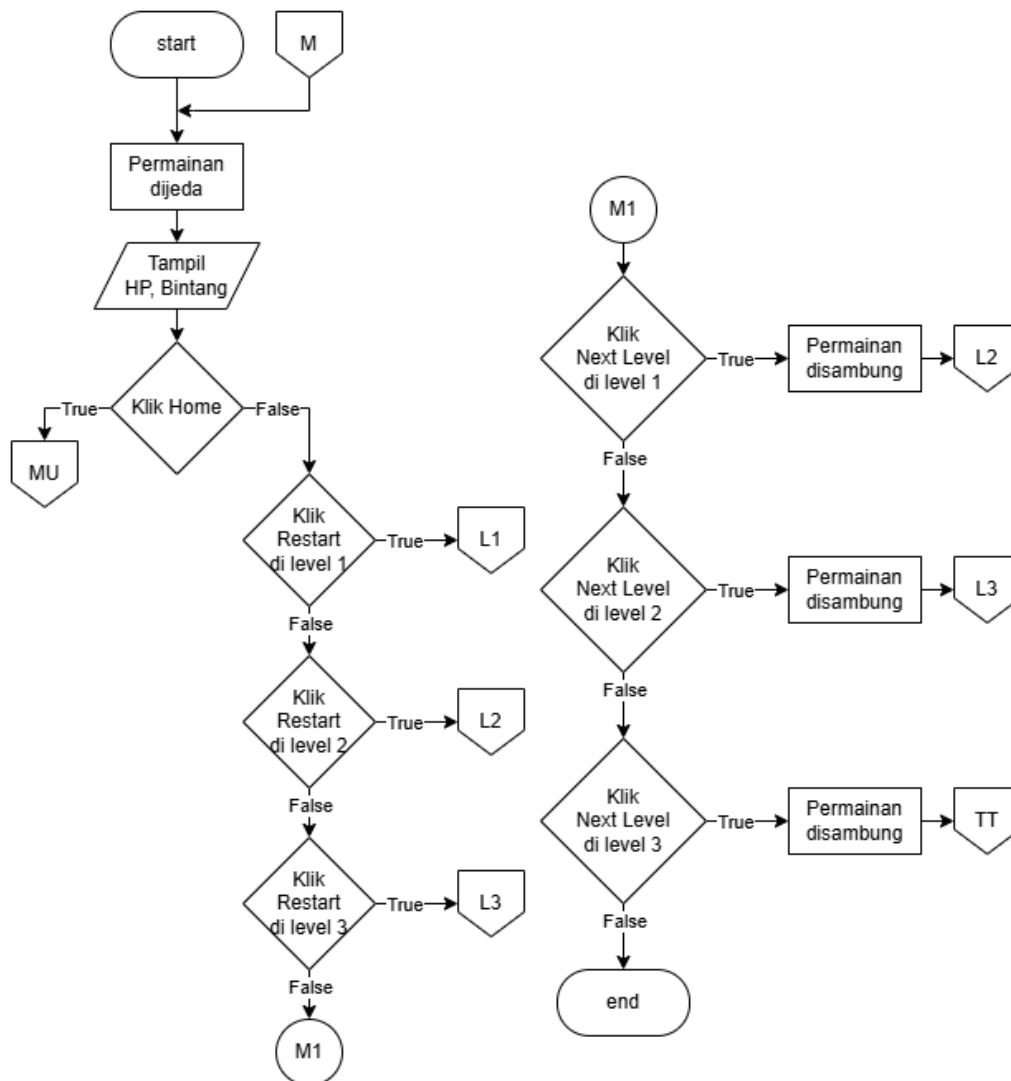
Gambar 4. 17 Flowchart Jeda Permainan

4.3.18 Flowchart Menang

Perancangan alur logika pada tampilan menang ini diinisialisasi melalui konektor M yang aktif ketika pemain berhasil menyelesaikan suatu level. Pada kondisi ini, sistem secara otomatis menjeda jalannya permainan dan menampilkan ringkasan pencapaian pemain yang meliputi sisa *Health Point* (HP) serta jumlah bintang yang diperoleh. Dalam antarmuka ini, sistem menyediakan opsi navigasi tombol *Home* yang jika ditekan akan mengarahkan pengguna kembali ke menu utama melalui konektor MU. Selain itu, terdapat mekanisme validasi untuk tombol *restart* yang menyesuaikan dengan level aktif saat itu. Apabila pengguna memilih untuk mengulang permainan di level 1, 2, atau 3, sistem akan mengarahkan alur navigasi kembali ke inisialisasi level terkait yang ditandai dengan konektor L1, L2, atau L3.

Selanjutnya, alur logika berlanjut ke konektor penghubung M1 untuk menangani interaksi pada tombol Next Level. Sistem melakukan pemeriksaan bertingkat untuk menentukan tujuan navigasi berdasarkan

level yang baru saja diselesaikan. Jika pemain menekan tombol lanjut setelah menyelesaikan level 1, sistem akan menyambungkan permainan menuju level 2 (konektor L2). Logika yang sama berlaku untuk penyelesaian level 2 yang akan diteruskan ke level 3 (konektor L3). Khusus untuk penyelesaian level 3, tombol lanjut akan mengarahkan pengguna ke tampilan tamat atau akhir permainan yang ditandai dengan konektor TT, menandakan bahwa seluruh rangkaian level telah berhasil diselesaikan.



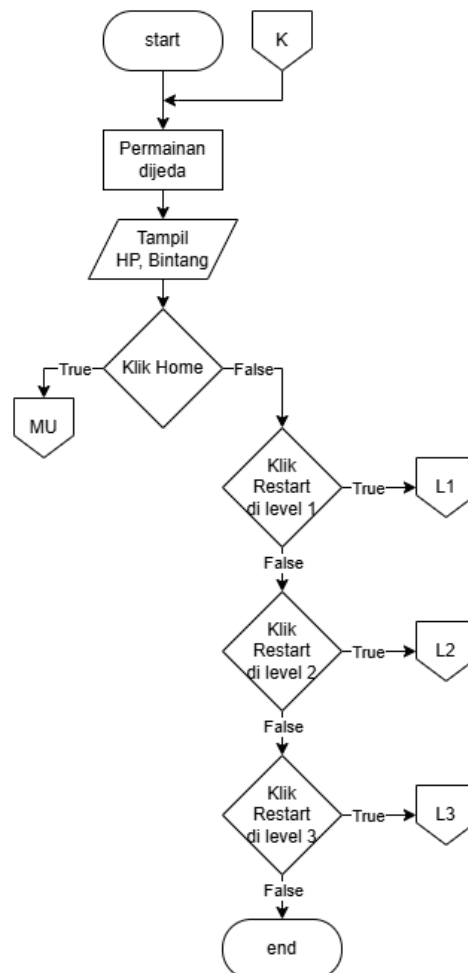
Gambar 4. 18 Flowchart Menang

4.3.19 Flowchart Kalah

Perancangan alur logika pada tampilan kalah ini diinisialisasi melalui konektor K yang aktif ketika kondisi permainan berakhir akibat habisnya poin kesehatan pemain. Pada fase ini, sistem secara otomatis menjeda jalannya permainan dan menampilkan antarmuka ringkasan

yang memuat informasi status terkini pemain. Di dalam tampilan ini, sistem menyediakan opsi navigasi utama berupa tombol *Home* yang berfungsi untuk mengembalikan pengguna ke menu awal. Apabila tombol ini ditekan, alur navigasi akan diarahkan langsung menuju tampilan menu utama yang ditandai dengan konektor MU.

Selain opsi kembali ke menu utama, sistem juga memfasilitasi pemain untuk mengulang permainan melalui tombol restart. Mekanisme validasi diterapkan untuk mendeteksi level mana yang sedang dimainkan saat kondisi kalah terjadi. Jika pengguna memilih untuk mengulang permainan pada level 1, sistem akan mengarahkan alur kembali ke inisialisasi level 1 melalui konektor L1. Logika yang setara diterapkan untuk level 2 dan level 3, di mana interaksi tombol restart akan membawa pengguna kembali ke permulaan level yang bersangkutan melalui konektor L2 dan L3. Hal ini memungkinkan pemain untuk segera mencoba kembali tantangan tanpa harus melewati menu pemilihan level lagi.

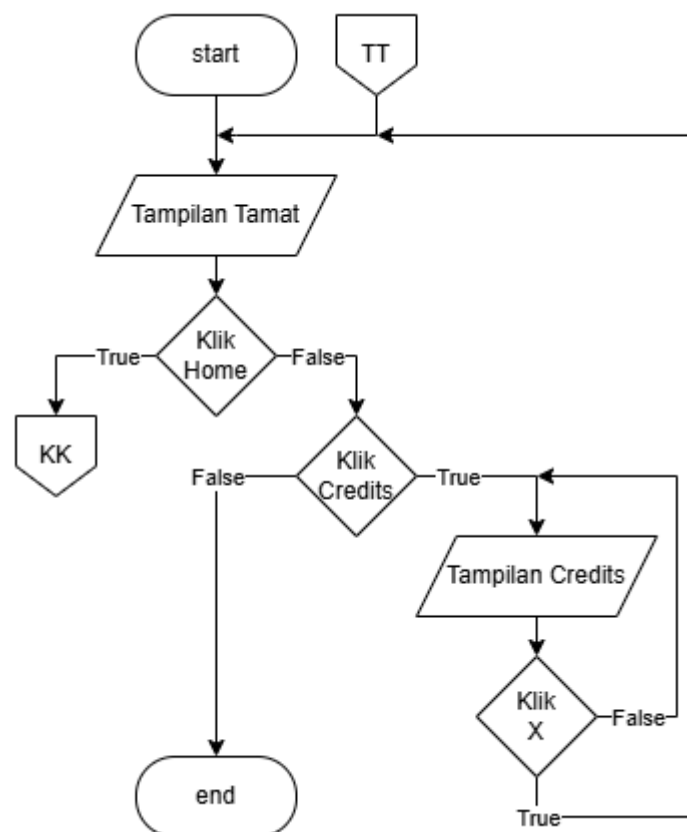


Gambar 4. 19 Flowchart Kalah

4.3.20 Flowchart Tamat

Perancangan alur logika pada tampilan tamat ini diinisialisasi melalui konektor TT yang menandakan bahwa pemain telah menyelesaikan seluruh level dalam permainan. Pada tahap ini, sistem menampilkan antarmuka tamat yang berfungsi sebagai penutup rangkaian pengalaman bermain sekaligus menyediakan opsi navigasi akhir. Mekanisme percabangan pertama berfokus pada deteksi interaksi tombol *Home*. Apabila pengguna menekan tombol tersebut, sistem akan memproses transisi navigasi menuju konektor KK untuk keluar dari halaman tamat dan kembali ke menu yang dituju.

Selain navigasi utama, sistem juga menyediakan akses informasi pengembang melalui tombol *Credits*. Jika pengguna memilih tombol ini, sistem akan memuat tampilan *credits* secara khusus. Di dalam antarmuka tersebut, terdapat mekanisme validasi interaksi pada tombol keluar atau simbol X. Apabila pengguna menekan tombol X, alur logika akan dikembalikan ke tampilan tamat awal sehingga pengguna dapat memilih opsi navigasi lainnya kembali. Namun jika tidak ada interaksi pada tombol *Credits* maupun tombol *Home*, alur logika akan mengarah ke status akhir atau *end* yang menandakan sistem menunggu *input* lebih lanjut atau menutup sesi interaksi aktif.

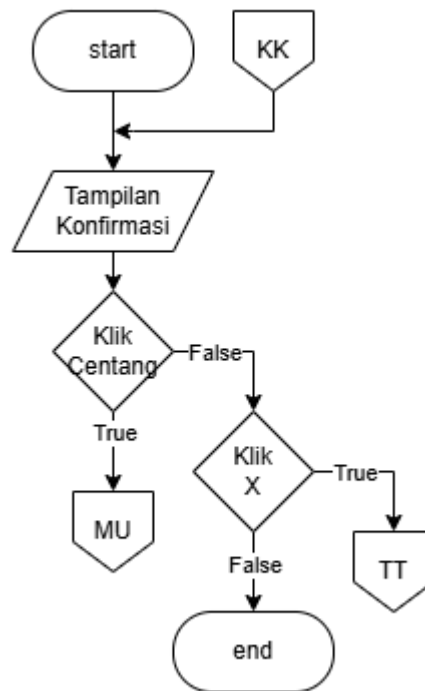


Gambar 4. 20 Flowchart Tamat

4.3.21 Flowchart Kembali Ke Menu

Perancangan alur logika pada tampilan konfirmasi ini diinisialisasi melalui konektor KK yang berfungsi sebagai mekanisme verifikasi akhir sebelum pengguna meninggalkan halaman tamat. Pada tahap ini, sistem menampilkan jendela konfirmasi *pop-up* yang meminta persetujuan pengguna untuk kembali ke menu utama. Tampilan ini bertujuan untuk mencegah perpindahan halaman yang tidak disengaja oleh pemain dan memastikan intensi navigasi mereka.

Sistem kemudian menangani dua kemungkinan interaksi navigasi utama dalam jendela konfirmasi tersebut. Apabila pengguna menyetujui untuk keluar dengan menekan tombol centang, sistem akan memproses transisi halaman menuju menu utama yang ditandai dengan konektor MU. Sebaliknya, jika pengguna memilih untuk membatalkan tindakan tersebut dengan menekan tombol simbol X, alur logika akan dikembalikan ke tampilan tamat melalui konektor TT sehingga pengguna dapat melihat kembali informasi pencapaian akhirnya.



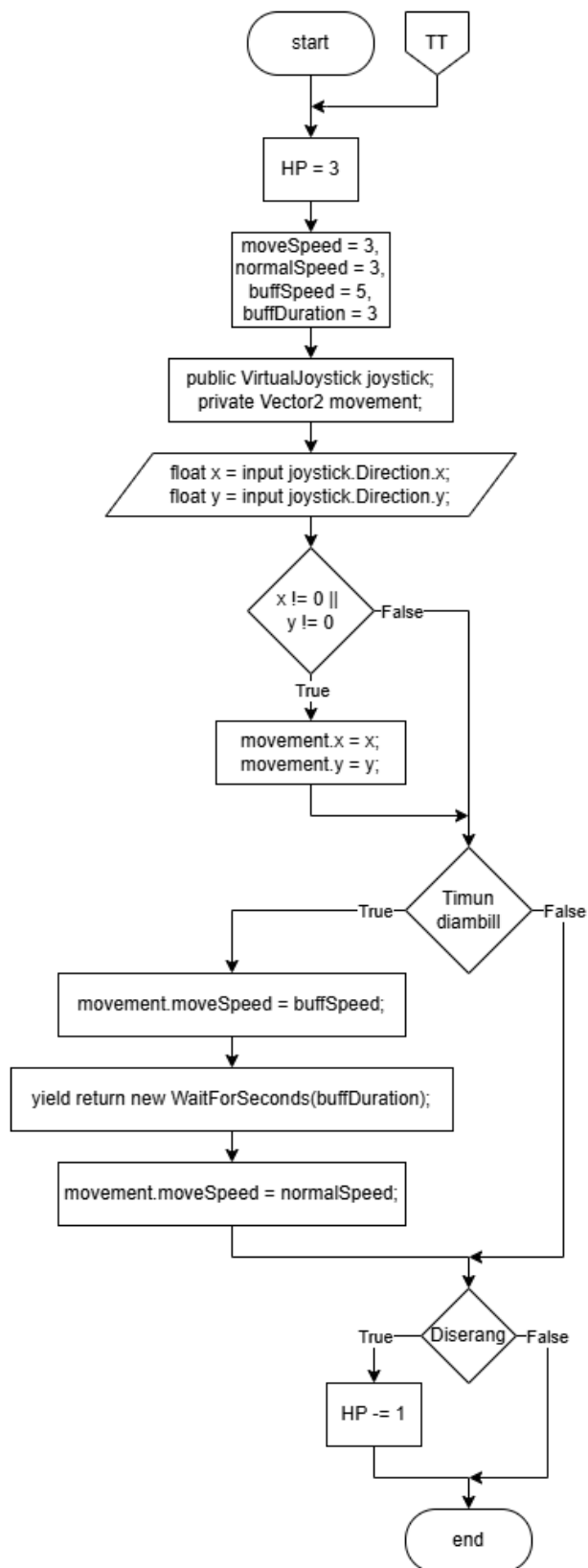
Gambar 4. 21 Flowchart Kembali Ke Menu

4.3.22 Flowchart Player

Perancangan alur logika pada algoritma *player* ini diawali dengan tahap inisialisasi variabel vital karakter. Sistem menetapkan nilai awal *Health Point* (HP) sebanyak tiga poin serta mendefinisikan parameter kecepatan gerak yang mencakup kecepatan normal dan kecepatan tambahan saat mendapatkan buff. Setelah inisialisasi selesai, sistem mulai memproses *input* navigasi dari kontrol *Virtual Joystick* dengan membaca nilai koordinat sumbu X dan Y. Apabila sistem mendeteksi

adanya *input* pergerakan yang bernilai tidak nol, vektor pergerakan karakter akan diperbarui mengikuti arah *input* tersebut agar karakter dapat berjalan pada area permainan.

Selanjutnya, alur logika memproses interaksi karakter terhadap objek lingkungan dan musuh. Sistem melakukan validasi kondisi apakah karakter telah mengambil item Timun yang berfungsi sebagai *power-up*. Jika kondisi ini terpenuhi, sistem akan meningkatkan kecepatan gerak karakter sesuai nilai *buff* selama durasi tertentu sebelum mengembalikannya ke kecepatan normal. Terakhir, sistem memeriksa kondisi pertahanan karakter untuk mendeteksi apakah terjadi serangan dari musuh. Apabila karakter terkonfirmasi menerima serangan, sistem akan mengurangi nilai HP sebanyak satu poin sebagai konsekuensi permainan sebelum mengakhiri siklus logika ini.

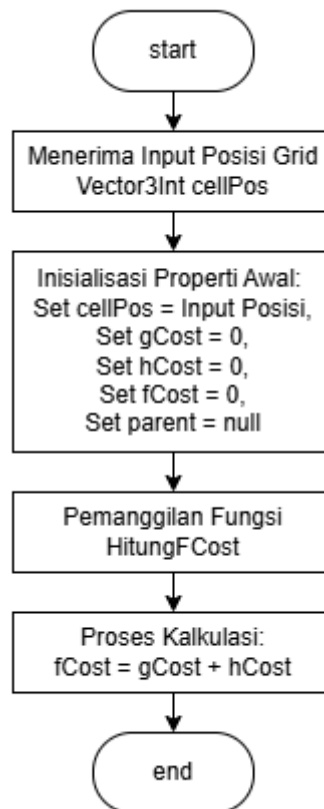


Gambar 4. 22 Flowchart Player

4.3.23 Flowchart Node

Perancangan alur logika pada algoritma *node* ini diawali dengan tahap penerimaan *input* posisi *grid* dalam format koordinat *Vector3Int*. Setelah data posisi diterima, sistem segera melakukan inisialisasi properti awal objek *node* tersebut untuk mempersiapkan variabel yang dibutuhkan dalam pencarian jalur. Proses ini mencakup penetapan nilai variabel *cellPos* sesuai *input* yang diterima serta pengaturan ulang nilai variabel biaya jalur yang meliputi *gCost*, *hCost*, dan *fCost* menjadi nol. Selain itu, referensi *parent* yang berfungsi sebagai penunjuk jejak jalur juga diatur menjadi *null* untuk memastikan status *node* dalam keadaan bersih sebelum digunakan.

Selanjutnya, alur logika beralih ke mekanisme penghitungan biaya melalui pemanggilan fungsi *HitungFCost*. Di dalam proses ini, sistem melakukan kalkulasi matematis untuk menentukan nilai total biaya atau *fCost* yang menjadi acuan utama prioritas dalam algoritma A-Star. Nilai *fCost* tersebut diperoleh dari hasil penjumlahan antara *gCost* sebagai biaya jarak dari titik awal dan *hCost* sebagai estimasi jarak menuju target. Setelah nilai *fCost* berhasil dikalkulasi dan diperbarui, rangkaian proses logika pada algoritma *node* ini berakhir.



Gambar 4. 23 Flowchart Node

4.3.24 Flowchart Algoritma A-Star (A*)

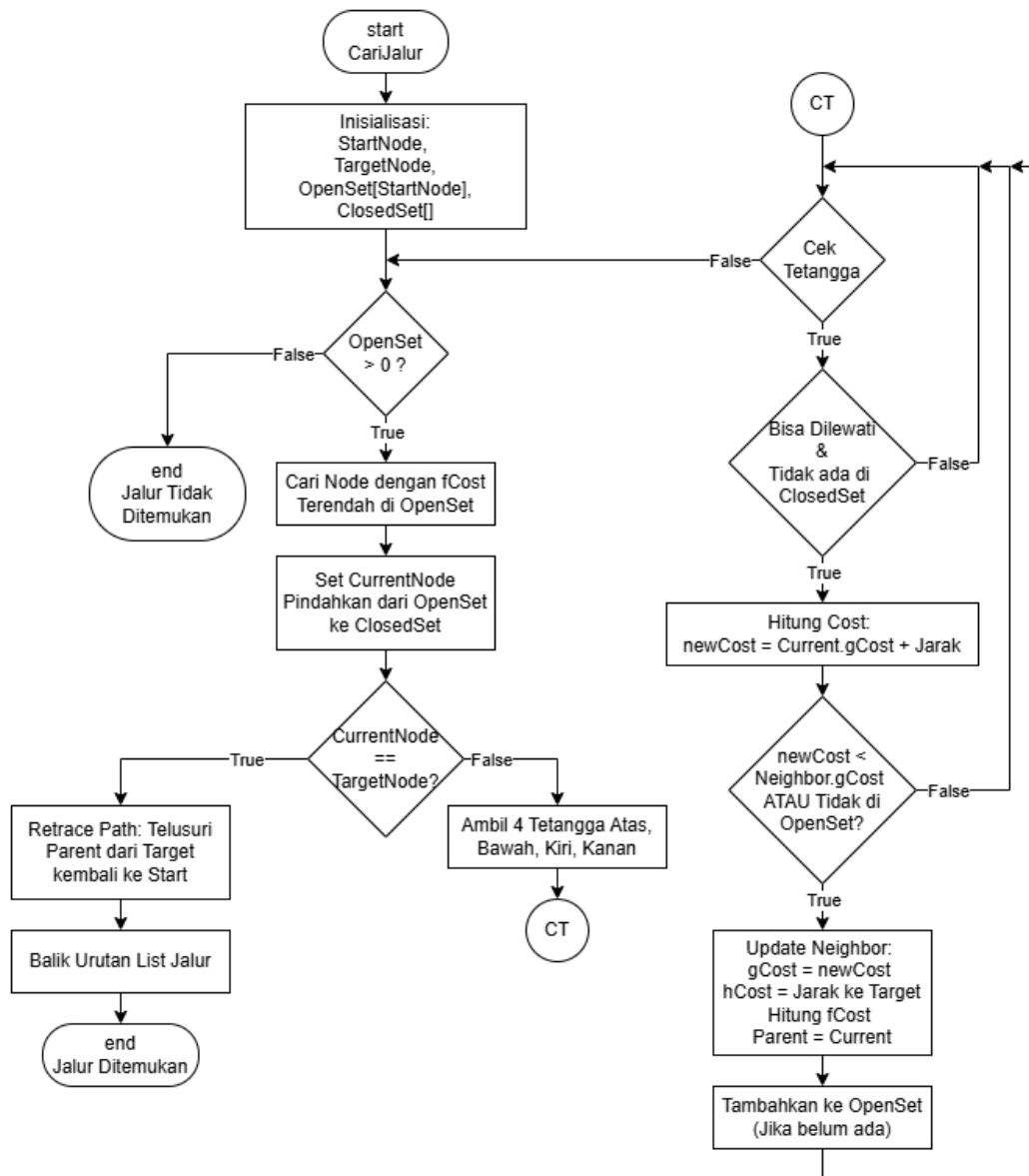
Perancangan alur logika pada algoritma pencarian jalur ini diawali dengan tahapan inisialisasi variabel utama yang dibutuhkan untuk kalkulasi. Pada tahap ini, sistem menetapkan titik awal atau *StartNode* dan titik tujuan atau *TargetNode* dalam area permainan. Selain itu, sistem mempersiapkan struktur data berupa himpunan *OpenSet* yang diisi dengan *StartNode* sebagai langkah pertama, serta himpunan *ClosedSet* yang dibiarkan kosong untuk menampung data simpul yang telah dievaluasi.

Setelah inisialisasi selesai, proses memasuki siklus perulangan utama yang memeriksa keberadaan data di dalam *OpenSet*. Apabila himpunan ini memiliki isi atau jumlah datanya lebih dari nol, sistem akan mencari dan memilih *node* yang memiliki nilai biaya total atau *fCost* terendah untuk dijadikan prioritas. *Node* yang terpilih tersebut kemudian ditetapkan sebagai *CurrentNode* dan dipindahkan posisinya dari *OpenSet* menuju *ClosedSet* sebagai tanda bahwa area tersebut sedang diproses.

Langkah krusial berikutnya adalah melakukan validasi terhadap *CurrentNode* untuk memeriksa apakah simpul tersebut merupakan target yang dituju. Jika kondisi ini bernilai benar, sistem akan menjalankan prosedur penelusuran balik atau *retrace path*. Prosedur ini bekerja dengan menelusuri jejak referensi *parent* dari titik target mundur hingga kembali ke titik awal. Urutan daftar jalur yang diperoleh kemudian dibalik agar menjadi rute yang valid dari awal ke tujuan, yang menandakan bahwa jalur telah berhasil ditemukan dan algoritma selesai.

Sebaliknya, jika *CurrentNode* bukan merupakan target, sistem akan melakukan ekspansi dengan mengambil data empat tetangga yang berada di posisi atas, bawah, kiri, dan kanan. Alur logika kemudian beralih ke sub-proses pemeriksaan tetangga yang ditandai dengan konektor CT. Pada fase ini, sistem melakukan validasi bertahap untuk memastikan bahwa tetangga tersebut merupakan area yang dapat dilewati atau *walkable* dan belum terdaftar di dalam *ClosedSet*. Jika tetangga tidak memenuhi syarat tersebut, sistem akan mengabaikannya dan beralih memeriksa tetangga lain.

Apabila tetangga memenuhi syarat validasi, sistem melanjutkan proses dengan menghitung estimasi biaya baru atau *newCost* yang merupakan penjumlahan dari *gCost* saat ini dengan jarak langkah. Nilai biaya baru ini kemudian dibandingkan dengan *gCost* milik tetangga yang sudah ada. Jika biaya baru terbukti lebih efisien atau tetangga tersebut belum pernah masuk ke dalam *OpenSet*, sistem akan melakukan pembaruan data. Pembaruan ini mencakup nilai *gCost*, *hCost* atau jarak *heuristic* ke target, perhitungan ulang *fCost*, serta penetapan *CurrentNode* sebagai *parent* dari tetangga tersebut. Terakhir, tetangga yang telah diperbarui ditambahkan ke dalam *OpenSet* untuk dievaluasi pada siklus berikutnya.



Gambar 4. 24 Flowchart Algoritma A-Star (A*)

4.3.25 Flowchart Musuh

Perancangan alur logika pada algoritma perilaku musuh ini diawali dengan tahapan *Awake* yang berfungsi untuk menginisialisasi komponen audio serta variabel pendukung lainnya saat permainan dimulai. Setelah proses inisialisasi selesai, sistem memasuki siklus perulangan utama atau *Update* yang berjalan secara terus-menerus di setiap bingkai permainan. Langkah fundamental pertama dalam siklus ini adalah melakukan perhitungan jarak matematis antara posisi musuh saat ini dengan posisi pemain untuk menentukan keputusan perilaku yang tepat.

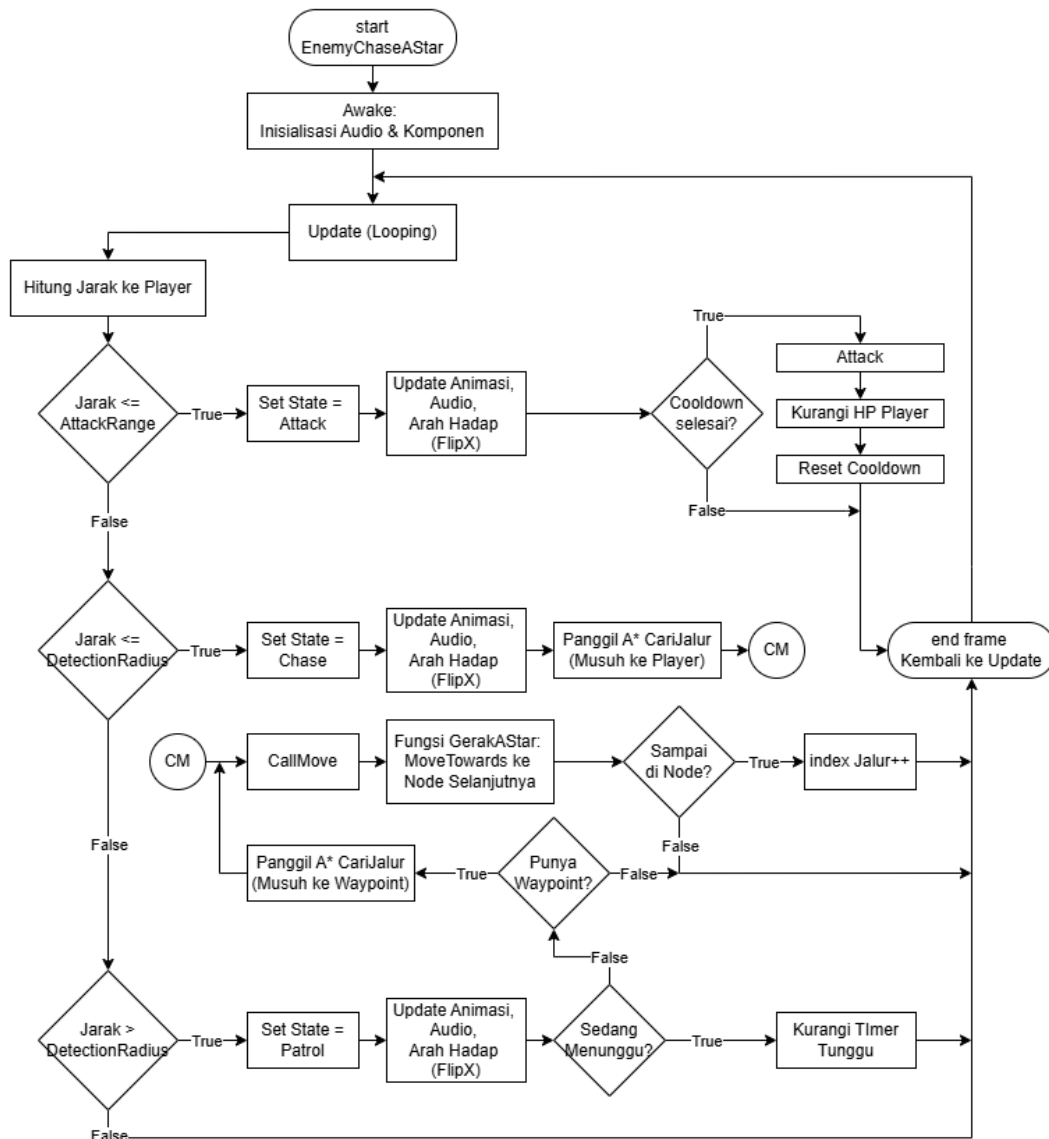
Sistem kemudian melakukan validasi kondisi jarak tersebut melalui struktur percabangan bertingkat. Prioritas pengecekan pertama berfokus pada jangkauan serangan atau *AttackRange*. Apabila jarak musuh ke

pemain terdeteksi lebih kecil atau sama dengan batas jangkauan serangan, sistem akan mengubah status musuh menjadi *Attack* serta memperbarui animasi, audio, dan arah hadap karakter. Selanjutnya, sistem memeriksa status *cooldown* serangan untuk memastikan ritme permainan yang seimbang. Jika waktu *cooldown* telah selesai, musuh akan meluncurkan serangan yang berdampak pada pengurangan *Health Point* (HP) pemain dan kemudian mereset durasi *cooldown*. Namun jika waktu tunggu belum habis, sistem akan langsung mengakhiri proses pada bingkai tersebut.

Apabila musuh tidak berada dalam jangkauan serangan, sistem melanjutkan validasi ke radius deteksi atau *DetectionRadius*. Jika jarak musuh masih berada di dalam cakupan radius ini, status musuh akan diubah menjadi Chase atau mode pengejaran. Pada kondisi ini, sistem memperbarui elemen visual dan audio, lalu segera memanggil fungsi pencarian jalur A* untuk menghitung rute optimal dari posisi musuh menuju posisi pemain saat ini. Alur logika kemudian diteruskan ke sub-sistem pergerakan yang ditandai dengan konektor CM.

Sebaliknya, jika jarak musuh terdeteksi berada di luar radius deteksi, sistem menetapkan status menjadi *Patrol*. Setelah memperbarui animasi dan arah hadap, sistem memeriksa apakah musuh sedang dalam fase menunggu. Jika kondisi menunggu bernilai benar, sistem hanya akan mengurangi variabel penghitung waktu tunggu. Namun jika musuh tidak sedang menunggu dan memiliki data titik patroli atau *waypoint*, sistem akan memanggil algoritma A* untuk mencari jalur menuju titik patroli tersebut, lalu mengarahkan alur ke konektor CM untuk eksekusi pergerakan.

Pada tahap akhir yang menangani eksekusi pergerakan melalui konektor CM, sistem menjalankan fungsi *GerakAStar* dengan metode *MoveTowards* untuk menggerakkan karakter musuh menuju *node* selanjutnya dalam jalur yang telah dikalkulasi. Sistem kemudian memvalidasi apakah musuh telah berhasil mencapai *node* tujuan tersebut. Jika posisi *node* tercapai, sistem akan meningkatkan indeks jalur agar musuh dapat bergerak ke titik berikutnya secara berurutan. Seluruh rangkaian logika ini bermuara pada penutup bingkai atau *end frame* yang mengembalikan proses ke awal siklus *Update* untuk iterasi selanjutnya.



Gambar 4. 25 Flowchart Musuh

4.4. Algoritme

Dalam tahap ini, penulis merancang dan mengimplementasikan algoritma yang diperlukan untuk memproses data dan menjalankan fungsi-fungsi sistem. Setiap algoritma memiliki peran spesifik dalam menjalankan operasi tertentu, seperti pengolahan *input*, pemrosesan data, dan penghasilan *output*. Algoritma ini menjelaskan bagaimana elemen pada permainan disusun, diproses, dan ditampilkan kepada pengguna. Tujuannya adalah untuk mengatur dan menyusun komponen secara logis dan efektif sehingga pengguna dapat dengan mudah berinteraksi dengan konten yang ditampilkan.

4.4.1 Algoritme Menu Utama

Pada tahap ini pengguna akan ditampilkan dengan beberapa pilihan tombol untuk ditekan dan dialihkan ke tampilan yang dipilih.

Algoritme 4. 1 Algoritme Menu Utama

```
1. Start (MU)
2. Display Tampilan Menu Utama
3. While (True):
4.   If Tombol Startgame ditekan Then
5.     Display Tampilan Menu Level
6.     Exit While
7.   Else If Tombol Exit ditekan Then
8.     Display Tampilan Konfirmasi Keluar Permainan
9.     Exit While
10.  Else If Tombol Pengaturan ditekan Then
11.    Display Tampilan Menu Pengaturan
12.    Exit While
13.  Else If Tombol Credits ditekan Then
14.    Display Tampilan Credits
15.    Exit While
16.  End If
17. End While
18. End
```

4.4.2 Algoritme Keluar Permainan

Pada tahap ini pengguna akan ditampilkan dengan 2 pilihan tombol untuk ditekan dan dialihkan ke tampilan yang dipilih.

Algoritme 4. 2 Algoritme Keluar Permainan

```
1. Start (KP)
2. Display Tampilan Konfirmasi Keluar Permainan
3. While (True):
4.   If Tombol Centang ditekan Then
5.     Keluar Permainan (Application.Quit)
6.     Exit While
7.   Else If Tombol Silang/Batal ditekan Then
8.     Go to ke Tampilan Menu Utama (MU)
9.     Exit While
10.  End If
11. End While
12. End
```

4.4.3 Algoritme Menu Pengaturan

Pada tahap ini pengguna akan ditampilkan dengan beberapa pilihan tombol untuk ditekan dan beberapa *slider* untuk diubah nilainya.

Algoritme 4. 3 Algoritme Menu Pengaturan

```
1. Start (MP)
2. Initialize Musik, SFX, Suara Cerita, Lewati Cerita
3. Display Tampilan Menu Pengaturan
```

```

4. If Slider Musik diubah nilainya Then
5.   Simpan nilai Musik
6. End If
7. If Slider SFX diubah nilainya Then
8.   Simpan nilai SFX
9. End If
10. If Slider Suara Cerita diubah nilainya Then
11.   Simpan nilai Suara Cerita
12. End If
13. If Tombol Lewati Cerita ditekan Then
14.   Simpan nilai Lewati Cerita
15. End If
16. If Tombol X ditekan Then
17.   Go to Tampilan Menu Utama (MU)
18. End If
19. End

```

4.4.4 *Algoritme Credits*

Pada tahap ini pengguna akan ditampilkan dengan profil pembuat *game* dan terdapat tombol *reset data* untuk ditekan.

Algoritme 4. 4 Algoritma Credits

```

1. Start (C)
2. Display Tampilan Credits
3. If Tombol Reset Data ditekan Then
4.   Display Tampilan Konfirmasi Reset Data
5.   Go to RD
6. Else If Tombol X ditekan Then
7.   Go to Tampilan Menu Utama (MU)
8. End If
9. End

```

4.4.5 *Algoritme Reset Data*

Pada tahap ini pengguna akan ditampilkan dengan 2 pilihan tombol untuk ditekan dan dialihkan ke tampilan yang dipilih.

Algoritme 4. 5 Algoritme Reset Data

```

1. Start (RD)
2. Display Tampilan Konfirmasi Reset Data
3. If Tombol Centang ditekan Then
4.   Execute Data Terhapus
5.   Display Tampilan Menu Utama
6.   Go to MU
7. Else If Tombol X ditekan Then
8.   Display Tampilan Menu Utama
9.   Go to MU
10. Else

```

11. End 12. End If 13. End

4.4.6 Algoritme Menu Level

Pada tahap ini pengguna akan ditampilkan dengan beberapa pilihan tombol untuk ditekan dan dialihkan ke tampilan yang dipilih.

Algoritme 4. 6 Algoritme Menu Level

1. Start (ML) 2. Display Tampilan Menu Level 3. If Tombol Level 1 ditekan Then 4. If Lewati Cerita == True Then 5. Go to T (Tampilan Cara Bermain/Tutorial) 6. Else 7. Display Tampilan Prolog Level 1 8. Go to P1 9. End If 10. Else If Tombol Level 2 ditekan AND Level 1 selesai Then 11. If Lewati Cerita == True Then 12. Go to L2 (Level 2) 13. Else 14. Display Tampilan Prolog Level 2 15. Go to P2 16. End If 17. Else If Tombol Level 3 ditekan AND Level 1 & 2 selesai Then 18. If Lewati Cerita == True Then 19. Go to L3 (Level 3) 20. Else 21. Display Tampilan Prolog Level 3 22. Go to P3 23. End If 24. Else If Tombol X ditekan Then 25. Display Tampilan Menu Utama 26. Go to MU 27. Else 28. End 29. End If 30. End
--

4.4.7 Algoritme Prolog Level 1

Pada tahap ini pengguna akan ditampilkan dengan cerita prolog untuk level 1.

Algoritme 4. 7 Algoritme Prolog Level 1

1. Start (P1) 2. While (Prolog Belum Selesai) Do

3.	Display Tampilan Prolog Level 1
4.	Play Suara Cerita
5.	If Tombol Next ditekan Then
6.	Percepat Typewriter
7.	End If
8.	Melanjutkan Prolog Level 1 (Update Teks)
9.	If Prolog Selesai == True Then
10.	Go to T (Tampilan Cara Bermain)
11.	Exit Loop
12.	End If
13.	End While
14.	End

4.4.8 Algoritme Cara Bermain

Pada tahap ini pengguna akan dipandu dengan langkah cara bermain sesuai dengan aturan yang berlaku pada permainan.

Algoritme 4. 8 Algoritme Cara Bermain

1.	Start (T)
2.	If Belum Pernah Melalui Tutorial == True Then
3.	While (True) Do
4.	Display Tampilan Cara Bermain
5.	If Tombol Next ditekan AND Akhir Tutorial == True Then
6.	Go to L1 (Tampilan Level 1)
7.	Exit While
8.	Else
9.	Update Tampilan Cara Bermain (Halaman Selanjutnya)
10.	End If
11.	End While
12.	Else
13.	Go to L1 (Tampilan Level 1)
14.	End If
15.	End

4.4.9 Algoritme Level 1

Pada level 1 pengguna akan disajikan dengan HUD dan tombol jeda permainan. Pengguna juga ditantang untuk menghindari pengejaran musuh yang diterapkan Algoritma A* di dalamnya.

Algoritme 4. 9 Algoritme Level 1

1.	Start (L1)
2.	While (True) Do
3.	Display Tampilan Level 1 & HUD
4.	Run Algoritma Player
5.	Run Algoritma Node
6.	Run Algoritma A-Star (A*)
7.	Run Algoritma Musuh

```

8.  If Klik Pause == True Then
9.      Set Permainan dijeda
10.     Go to J (Menu Jeda)
11.     Exit Loop
12. Else If Pemain mencapai garis finish == True Then
13.     If Lewati Cerita == True Then
14.         Go to M (Menang/Level Selesai)
15.     Else
16.         Display Tampilan Epilog Level 1
17.         Go to E1
18.     End If
19.     Exit Loop
20. Else If HP == 0 Then
21.     Go to K (Kalah/Game Over)
22.     Exit Loop
23. End If
    // Jika semua kondisi False, loop kembali ke langkah 3
24. End While
25. End

```

4.4.10 Algoritme Epilog Level 1

Pada tahap ini pengguna akan ditampilkan dengan cerita epilog untuk level 1.

Algoritme 4. 10 Algoritme Epilog Level 1

```

1. Start (E1)
2. While (Epilog Belum Selesai) Do
3.     Display Tampilan Epilog Level 1
4.     Play Suara Cerita
5.     If Tombol Next ditekan Then
6.         Percepat Typewriter
7.     End If
8.     Melanjutkan Epilog Level 1 (Update Teks)
9.     If Epilog Selesai == True Then
10.        Go to M (Tampilan Menang/Level Selesai)
11.        Exit Loop
12.    End If
13. End While
14. End

```

4.4.11 Algoritme Prolog Level 2

Pada tahap ini pengguna akan ditampilkan dengan cerita prolog untuk level 2.

Algoritme 4. 11 Algoritme Prolog Level 2

```

1. Start (P2)
2. While (Prolog Belum Selesai) Do

```

```

3.  Display Tampilan Prolog Level 2
4.  Play Suara Cerita
5.  If Tombol Next ditekan Then
6.      Percepat Typewriter
7.  End If
8.  Melanjutkan Prolog Level 2 (Update Teks)
9.  If Prolog Selesai == True Then
10.      Go to L2 (Tampilan Level 2)
11.      Exit Loop
12.  End If
13. End While
14. End

```

4.4.12 Algoritme Level 2

Pada level 2 pengguna akan disajikan dengan HUD dan tombol jeda permainan. Pengguna juga ditantang untuk menghindari pengejaran musuh yang diterapkan Algoritma A* di dalamnya.

Algoritme 4. 12 Algoritme Level 2

```

1. Start (L2)
2. While (True) Do
3.  Display Tampilan Level 2 & HUD
4.  Run Algoritma Player
5.  Run Algoritma Node
6.  Run Algoritma A-Star (A*)
7.  Run Algoritma Musuh
8.  If Klik Pause == True Then
9.      Set Permainan dijeda
10.     Go to J (Menu Jeda)
11.     Exit Loop
12. Else If Pemain mencapai garis finish == True Then
13.     If Lewati Cerita == True Then
14.         Go to M (Menang/Level Selesai)
15.     Else
16.         Display Tampilan Epilog Level 2
17.         Go to E2
18.     End If
19.     Exit Loop
20. Else If HP == 0 Then
21.     Go to K (Kalah/Game Over)
22.     Exit Loop
23. End If
    // Jika semua kondisi False, loop kembali ke langkah 3
24. End While
25. End

```

4.4.13 Algoritme Epilog Level 2

Pada tahap ini pengguna akan ditampilkan dengan cerita epilog untuk level 2.

Algoritme 4. 13 Algoritme Epilog Level 2

1. Start (E2) 2. While (Epilog Belum Selesai) Do 3. Display Tampilan Epilog Level 2 4. Play Suara Cerita 5. If Tombol Next ditekan Then 6. Percepat Typewriter 7. End If 8. Melanjutkan Epilog Level 2 (Update Teks) 9. If Epilog Selesai == True Then 10. Go to M (Tampilan Menang/Level Selesai) 11. Exit Loop 12. End If 13. End While 14. End

4.4.14 Algoritme Prolog Level 3

Pada tahap ini pengguna akan ditampilkan dengan cerita prolog untuk level 3.

Algoritme 4. 14 Algoritme Prolog Level 3

1. Start (P3) 2. While (Prolog Belum Selesai) Do 3. Display Tampilan Prolog Level 3 4. Play Suara Cerita 5. If Tombol Next ditekan Then 6. Percepat Typewriter 7. End If 8. Melanjutkan Prolog Level 3 (Update Teks) 9. If Prolog Selesai == True Then 10. Go to L3 (Tampilan Level 3) 11. Exit Loop 12. End If 13. End While 14. End

4.4.15 Algoritme Level 3

Pada level 3 pengguna akan disajikan dengan HUD dan tombol jeda permainan. Pengguna juga ditantang untuk menghindari pengejaran musuh yang diterapkan Algoritma A* di dalamnya.

Algoritme 4. 15 Algoritme Level 3

1. Start (L3)

```

2. While (True) Do
3.   Display Tampilan Level 3 & HUD
4.   Run Algoritma Player
5.   Run Algoritma Node
6.   Run Algoritma A-Star (A*)
7.   Run Algoritma Musuh
8.   If Klik Pause == True Then
9.     Set Permainan dijeda
10.    Go to J (Menu Jeda)
11.    Exit Loop
12.  Else If Pemain mencapai garis finish == True Then
13.    If Lewati Cerita == True Then
14.      Go to M (Menang/Level Selesai)
15.    Else
16.      Display Tampilan Epilog Level 2
17.      Go to E3
18.    End If
19.    Exit Loop
20.  Else If HP == 0 Then
21.    Go to K (Kalah/Game Over)
22.    Exit Loop
23.  End If
    // Jika semua kondisi False, loop kembali ke langkah 3
24. End While
25. End

```

4.4.16 Algoritme Epilog Level 3

Pada tahap ini pengguna akan ditampilkan dengan cerita epilog untuk level 2.

Algoritme 4. 16 Algoritme Epilog Level 3

```

1. Start (E3)
2. While (Epilog Belum Selesai) Do
3.   Display Tampilan Epilog Level 3
4.   Play Suara Cerita
5.   If Tombol Next ditekan Then
6.     Percepat Typewriter
7.   End If
8.   Melanjutkan Epilog Level 3 (Update Teks)
9.   If Epilog Selesai == True Then
10.    Go to M (Tampilan Menang/Level Selesai)
11.    Exit Loop
12.  End If
13. End While
14. End

```

4.4.17 Algoritme Jeda Permainan

Pada tahap ini pengguna akan ditampilkan dengan beberapa pilihan tombol untuk ditekan dan dialihkan ke tampilan yang dipilih.

Algoritme 4. 17 Algoritme Jeda Permainan

```
1. Start (J)
2. Set Permainan Dijeda (Time.timeScale = 0)
3. Display Tampilan Menu Jeda
4. While (Permainan Sedang Jeda) Do
5.   If Tombol Home ditekan Then
6.     Display Tampilan Menu Utama
7.     Go to MU
8.     Exit Loop
9.   Else If Tombol Restart ditekan AND Sedang di Level 1 Then
10.    Reload Level 1
11.    Go to L1
12.   Else If Tombol Restart ditekan AND Sedang di Level 2 Then
13.    Reload Level 2
14.    Go to L2
15.   Else If Tombol Restart ditekan AND Sedang di Level 3 Then
16.    Reload Level 3
17.    Go to L3
18.   Else If Tombol Continue ditekan AND Sedang di Level 1 Then
19.    Set Permainan Disambung (Time.timeScale = 1)
20.    Go to L1 (Resume)
21.   Else If Tombol Continue ditekan AND Sedang di Level 2 Then
22.    Set Permainan Disambung (Time.timeScale = 1)
23.    Go to L2 (Resume)
24.   Else If Tombol Continue ditekan AND Sedang di Level 3 Then
25.    Set Permainan Disambung (Time.timeScale = 1)
26.    Go to L3 (Resume)
27.   End If
28. End While
29. End
```

4.4.18 Algoritme Menang

Pada tahap ini pengguna akan ditampilkan dengan beberapa pilihan tombol setelah berhasil menyelesaikan level dan dialihkan ke tampilan yang dipilih pada tombol yang tersedia.

Algoritme 4. 18 Algoritme Menang

```
1. Start (M)
2. Set Permainan Dijeda (Time.timeScale = 0)
3. Display Tampilan Menang, HP, dan Bintang
4. While (Menunggu Input Pemain) Do
5.   If Tombol Home ditekan Then
6.     Display Tampilan Menu Utama
```

```

7.    Go to MU (Menu Utama)
8.    Exit Loop
9.  Else If Tombol Restart ditekan Then
10.   If Sedang di Level 1 Then
11.     Reload Level 1
12.     Go to L1 (Level 1)
13.   Else If Sedang di Level 2 Then
14.     Reload Level 2
15.     Go to L2 (Level 2)
16.   Else If Sedang di Level 3 Then
17.     Reload Level 3
18.     Go to L3 (Level 3)
19.   End If
20.   Exit Loop
21. Else If Tombol Next Level ditekan Then
22.   Set Permainan Disambung (Time.timeScale = 1)
23.   If Sedang di Level 1 Then
24.     Load Level 2
25.     Go to L2 (Level 2)
26.   Else If Sedang di Level 2 Then
27.     Load Level 3
28.     Go to L3 (Level 3)
29.   Else If Sedang di Level 3 Then
30.     Load Tampilan Tamat
31.     Go to TT (Tampilan Tamat)
32.   End If
33.   Exit Loop
34. End If
35. End While
36. End

```

4.4.19 Algoritme Kalah

Pada tahap ini pengguna akan ditampilkan dengan beberapa pilihan tombol setelah gagal menyelesaikan level dan dialihkan ke tampilan yang dipilih pada tombol yang tersedia.

Algoritme 4. 19 Algoritme Kalah

```

1. Start (K)
2. Set Permainan Dijeda (Time.timeScale = 0)
3. Display Tampilan Kalah, HP, dan Bintang
4. While (Menunggu Input Pemain) Do
5.   If Tombol Home ditekan Then
6.     Display Tampilan Menu Utama
7.     Go to MU
8.     Exit Loop
9.   Else If Tombol Restart ditekan AND Sedang di Level 1 Then
10.    Reload Level 1

```

```

11.   Go to L1
12.   Exit Loop
13. Else If Tombol Restart ditekan AND Sedang di Level 2 Then
14.   Reload Level 2
15.   Go to L2
16.   Exit Loop
17. Else If Tombol Restart ditekan AND Sedang di Level 3 Then
18.   Reload Level 3
19.   Go to L3
20.   Exit Loop
21. End If
22. End While
23. End

```

4.4.20 Algoritme Tamat

Pada tahap ini akan menampilkan layar tamat setelah melanjutkan dari level 3 dan tersedia beberapa tombol yang dapat dipilih.

Algoritme 4. 20 Algoritme Tamat

```

1. Start (TT)
2. Display Tampilan Tamat
3. While (True) Do
4.   If Tombol Home ditekan Then
5.     Go to KK (Konfirmasi Keluar / Ke Menu Utama)
6.     Exit Loop
7.   Else If Tombol Credits ditekan Then
8.     // Masuk ke sub-menu Credits
9.     Display Tampilan Credits
10.    While (Credits Aktif) Do
11.      If Tombol X ditekan Then
12.        // Kembali ke Tampilan Tamat
13.        Display Tampilan Tamat
14.        Go to TT (Tampilan Tamat)
15.        Exit Sub-Loop
16.      Else
17.        // Looping Tampilan Credits (Jalur False)
18.        Display Tampilan Credits
19.      End If
20.    End While
21.  Else
22.    End
23.  End If
24. End While
25. End

```

4.4.21 Algoritme Kembali Ke Menu Utama

Pada tahap ini pengguna akan ditampilkan dengan 2 pilihan tombol untuk ditekan dan dialihkan ke tampilan yang dipilih.

Algoritme 4. 21 Algoritme Kembali Ke Menu Utama

```
1. Start (KK)
2. Display Tampilan Konfirmasi
3. If Tombol Centang ditekan Then
4.   Display Tampilan Menu Utama
5.   Go to MU (Menu Utama)
6. Else If Tombol X ditekan Then
7.   Display Tampilan Tamat
8.   Go to TT (Tamat)
9. Else
10.  End
11. End If
12. End
```

4.4.22 Algoritme Player

Pada tahap ini algoritma *Player* dibuat untuk mendukung algoritma Level 1, 2 dan 3.

Algoritme 4. 22 Algoritme Player

```
1. Start
2. // Inisialisasi Variabel
   Set HP = 3
   Set moveSpeed = 3
   Set normalSpeed = 3
   Set buffSpeed = 5
   Set buffDuration = 3
   Inisialisasi VirtualJoystick dan Vector2 movement

3. // Proses Input Pergerakan
   Read Input x = joystick.Direction.x
   Read Input y = joystick.Direction.y

4. If (x != 0 OR y != 0) Then
5.   Set movement.x = x
6.   Set movement.y = y
7.   Jalankan Animasi Jalan (Run)
8. End If

9. // Logika Interaksi Item (Timun)
10. If Timun diambil == True Then
11.   Set movement.moveSpeed = buffSpeed
12.   Wait/Delay selama buffDuration (Coroutine)
13.   Set movement.moveSpeed = normalSpeed
```

```

14. End If

15. // Logika Interaksi Musuh (Damage)
16. If Diserang == True Then
17.   Set HP = HP - 1
18.   Jalankan Efek Hurt/Terluka
19. End If

20. End

```

4.4.23 Algoritme Node

Pada tahap ini algoritma *Node* dibuat untuk mendukung algoritma A-Star yang diterapkan ke musuh.

Algoritme 4. 23 Algoritme Node

```

1. Define Class Node
2.  // Deklarasi Properti
   Variable cellPos   : Vector3Int (Posisi Grid)
   Variable gCost     : Integer   (Jarak dari titik Awal)
   Variable hCost     : Integer   (Estimasi jarak ke Target)
   Variable fCost     : Integer   (Total Biaya: G + H)
   Variable parent    : Node      (Node sebelumnya untuk jejak
balik)

3.  // Konstruktor: Inisialisasi awal saat Node dibuat
   Function Constructor(InputPosisi)
     Set cellPos = InputPosisi
     Set gCost = 0
     Set hCost = 0
     Set fCost = 0
     Set parent = Null
   End Function

4.  // Method: Menghitung total biaya (F Cost)
   Function HitungFCost()
     Set fCost = gCost + hCost
     Return fCost
   End Function

5. End Class

```

4.4.24 Algoritme A-Star (A*)

Pada tahap ini algoritma A-Star dibuat sebagai fungsi untuk mendukung algoritma yang diterapkan ke musuh.

Algoritme 4. 24 Algoritme A-Star (A*)

```
1. Function CariJalur(startPos, targetPos)
2.  // Inisialisasi
3.  Clear NodeCache
4.  Set startNode = CreateNode(startPos)
5.  Set targetNode = CreateNode(targetPos)
6.  Set openSet = List berisi {startNode}
7.  Set closedSet = List Kosong

8.  // Loop Utama Pencarian
9.  While (openSet Count > 0) Do
10.   Set currentNode = Node di openSet dengan fCost terendah
11.   // (Jika fCost sama, pilih hCost terendah)

12.   Remove currentNode dari openSet
13.   Add currentNode ke closedSet

14.   // Cek apakah target ditemukan
15.   If currentNode == targetNode Then
16.     Call RetracePath(startNode, targetNode)
17.     Return Path
18.   End If

19.   // Eksplorasi Tetangga (Atas, Bawah, Kiri, Kanan)
20.   For Each neighbor in GetNeighbors(currentNode) Do
21.     If (neighbor Tidak Bisa Dilewati) OR (neighbor ada di closedSet) Then
22.       Continue Loop (Skip tetangga ini)
23.     End If

24.     // Hitung biaya pergerakan baru (gCost)
25.     Set newCost = currentNode.gCost + GetDistance(currentNode,
neighbor)

26.     If (newCost < neighbor.gCost) OR (neighbor tidak ada di openSet)
Then
27.       // Update data tetangga
28.       Set neighbor.gCost = newCost
29.       Set neighbor.hCost = GetDistance(neighbor, targetNode)
30.       Call neighbor.HitungFCost()
31.       Set neighbor.parent = currentNode

32.       If neighbor tidak ada di openSet Then
33.         Add neighbor ke openSet
34.       End If
35.     End If
36.   End For
```

```

37. End While

38. Return Null (Jalur tidak ditemukan)
39. End Function

40. // Fungsi Pembantu: Mengurutkan jalur dari akhir ke awal
41. Function RetracePath(startNode, endNode)
42.   Set path = List Kosong
43.   Set currentNode = endNode
44.   While (currentNode != startNode) Do
45.     Add currentNode ke path
46.     Set currentNode = currentNode.parent
47.   End While
48.   Reverse path (Balik urutan agar dari start ke end)
49.   Return path
50. End Function

```

4.4.25 Algoritme Musuh

Algoritma ini diterapkan pada musuh sebagai kecerdasan buatan yang mengatur perilaku musuh.

Algoritme 4. 25 Algoritme Musuh

```

1. Class EnemyChaseAStar
2. // Variabel
   Variable state      : Enum (Idle, Patrol, Chase, Attack)
   Variable path       : List of Vector3 (Hasil dari A*)
   Variable currentPathIndex : Integer
   Variable attackTimer : Float

3. // Fungsi Utama Unity
   Function Update()
     // 1. Tentukan State berdasarkan Jarak
     Set distance = Distance(Enemy.position, Player.position)

     If distance <= attackRange Then
       Set state = Attack
     Else If distance <= detectionRadius Then
       Set state = Chase
     Else
       Set state = Patrol
     End If

     // 2. Update Visual
     Call UpdateAnimation(state)
     Call UpdateFacingDirection()

     // 3. Eksekusi Perilaku

```

```

    If state == Attack Then
        Call PerformAttack()
    Else If state == Chase Then
        Call PerformChase()
    Else If state == Patrol Then
        Call PerformPatrol()
    End If
End Function

4. // Perilaku: KEJAR (Chase)
Function PerformChase()
    // Hitung ulang jalur A* setiap frame agar responsif
    Set path = AStar.CariJalur(Enemy.position, Player.position)
    Set currentPathIndex = 0
    Call GerakAStar()
End Function

5. // Perilaku: PATROLI (Patrol)
Function PerformPatrol()
    If SedangMenunggu Then
        Kurangi waitTimer
        If waitTimer <= 0 Then BerhentiMenunggu()
        Return
    End If

    If path Kosong OR Sampai di Waypoint Then
        Set targetWaypoint = NextWaypoint()
        Set path = AStar.CariJalur(Enemy.position, targetWaypoint)
        Set currentPathIndex = 0
    End If

    Call GerakAStar()
End Function

6. // Perilaku: SERANG (Attack)
Function PerformAttack()
    If Time.time >= lastAttackTime + cooldown Then
        Player.TakeDamage(damageAmount)
        Set lastAttackTime = Time.time
        PlaySound(AttackSFX)
    End If
End Function

7. // Fungsi Gerak Fisik
Function GerakAStar()
    If path Kosong OR currentPathIndex >= path.Count Then Return

```

```

Set targetPos = path[currentPathIndex]

// Gerakkan musuh
MoveTowards(Enemy.position, targetPos, speed * deltaTime)

// Cek jika sudah sampai di titik node tersebut
If Distance(Enemy.position, targetPos) < 0.1 Then
    currentPathIndex = currentPathIndex + 1
End If
End Function

```

4.5. Pengujian

Metode pengujian yang digunakan dalam penelitian ini adalah pengujian fungsional dengan pendekatan *black box testing*. Metode ini digunakan untuk menguji fungsi-fungsi utama sistem berdasarkan *input* dan *output* tanpa melihat detail kode program.

4.5.1 Pengujian Fungsi Utama Sistem

Hasil dari pengujian fungsi utama sistem yang terdapat dalam *game*. Setiap pengujian dilakukan untuk memastikan bahwa sistem memberikan hasil yang sesuai dengan yang diharapkan.

Tabel 4. 6 Tabel Pengujian Black Box Fungsi Utama Sistem

No.	Pengujian	Hasil yang diharapkan	Kesimpulan
1	Klik <i>Exit</i> di menu utama	Tampilan panel konfirmasi keluar permainan	Berhasil
2	Klik centang pada panel konfirmasi keluar	Keluar dari permainan	Berhasil
3	Klik x pada panel konfirmasi keluar	Tampilan menu utama	Berhasil
4	Klik Pengaturan di menu utama	Tampilan menu pengaturan	Berhasil
5	Menggeser nilai musik di menu pengaturan	Volume musik berubah	Berhasil
6	Menggeser nilai SFX di menu pengaturan	Volume SFX berubah	Berhasil
7	Menggeser nilai suara cerita di menu pengaturan	Volume suara cerita berubah	Berhasil

No.	Pengujian	Hasil yang diharapkan	Kesimpulan
8	Klik Lewati Cerita	Prolog dan Epilog tidak akan tampil pada level 1, level 2 dan level 3	Berhasil
9	Klik x pada menu pengaturan	Tampilan menu utama	Berhasil
10	Klik credits di menu utama	Tampilan profil pembuat game	Berhasil
11	Klik Reset Data di menu credits	Tampilan panel konfirmasi untuk reset data	Berhasil
12	Klik centang pada panel konfirmasi reset data	Data terhapus dan tampil ke menu utama	Berhasil
13	Klik x pada panel konfirmasi reset data	Tampilan menu utama	Berhasil
14	Klik Start Game di menu utama	Tampilan menu level	Berhasil
15	Klik level 1 di menu level	Masuk prolog level 1 atau masuk level 1 jika melewati cerita	Berhasil
16	Klik level 2 di menu level	Dapat di klik setelah menyelesaikan level 1 dan tidak dapat di klik jika belum menyelesaikan level 1. Jika level 2 bisa di klik masuk prolog level 2 atau masuk level 2 jika melewati cerita	Berhasil
17	Klik level 3 di menu level	Dapat di klik setelah menyelesaikan level 2 dan tidak dapat di klik jika belum menyelesaikan level 2. Jika level 3 bisa di klik masuk prolog level 3 atau masuk level 3 jika melewati cerita	Berhasil
18	Klik x di menu level	Tampilan menu utama	Berhasil
19	Klik next di prolog level 1, level 2 dan level 3	Jika efek typewriter belum selesai maka tombol next berfungsi untuk menyelesaikan efek typewriter. Jika efek typewriter selesai namun suara cerita belum selesai maka tombol next berfungsi untuk melanjutkan prolog.	Berhasil
20	Klik next di epilog level 1, level 2 dan level 3	Jika efek typewriter belum selesai maka tombol next berfungsi untuk menyelesaikan efek typewriter. Jika efek typewriter selesai namun suara cerita belum selesai maka tombol next berfungsi untuk melanjutkan prolog.	Berhasil
21	Tampilan cara bermain	Menampilkan aturan permainan yang hanya berlaku sekali pada saat	Berhasil

No.	Pengujian	Hasil yang diharapkan	Kesimpulan
		pengguna masuk ke level 1 untuk pertama kalinya	
22	Tampilan kalah	Menampilkan nyawa pengguna dan perolehan bintang serta 2 tombol untuk kembali dan mengulang permainan	Berhasil
23	Tampilan Menang	Menampilkan nyawa pengguna dan perolehan bintang serta 3 tombol untuk kembali ke menu utama, mengulang permainan dan melanjutkan level	Berhasil
24	Tampilan Jeda permainan	Terdapat 3 tombol untuk kembali ke menu utama, mengulang permainan dan meneruskan permainan	Berhasil
25	Klik next level di tampilan menang level 1 dan level 2	Jika pengguna berada di level 1 maka akan dialihkan ke permainan level 2. Jika pengguna berada di level 2 maka akan dialihkan ke permainan level 3	Berhasil
26	Klik next level di tampilan menang level 3	Tampilan tamat serta animasi ucapan terima kasih kepada pengguna yang telah bermain	Berhasil
27	Pengguna menggerakkan virtual joystick	Karakter Kancil bergerak sesuai arah virtual joystick yang diterapkan oleh pengguna	Berhasil
28	Pengguna mendapatkan item mentimun	Kecepatan pada karakter Kancil bertambah untuk sementara waktu	Berhasil
29	Pengguna menuju garis finish	Syarat penyelesaian level dan akan tampil epilog jika tidak melewati cerita, lalu menampilkan tampilan menang	Berhasil
30	Pengguna mengumpulkan 100% mentimun	Bintang yang diperoleh adalah 3	Berhasil
31	Pengguna mengumpulkan 50% s.d. 99% mentimun	Bintang yang diperoleh adalah 2	Berhasil
32	Pengguna mengumpulkan <50% mentimun	Bintang yang diperoleh adalah 1	Berhasil
33	Nyawa Pengguna 0	Tampilan kalah tanpa epilog	Berhasil
34	<i>Background Music</i>	Musik yang dimainkan berbeda tiap levelnya	Berhasil

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa seluruh fungsi utama dalam gim telah beroperasi sesuai dengan

rancangan yang ditetapkan. Responsivitas kontrol pemain, serta fungsi sistem level dan kondisi permainan berjalan dengan normal.

Pada pengujian nomor 8, ketiadaan penayangan Prolog dan Epilog pada level 1, 2, dan 3 telah terkonfirmasi sesuai dengan ekspektasi. Namun, ditemukan adanya anomali pada visualisasi *Head-Up Display* (HUD), khususnya pada indikator poin kesehatan (*health points*) dan durasi permainan yang tidak tampil semestinya. Kendati demikian, sistem perhitungan poin kesehatan dipastikan tetap berfungsi secara logis meskipun terjadi kegagalan pada antarmuka visualnya.

4.5.2 Pengujian Algoritma A-Star (A*)

Pengujian terhadap implementasi algoritma pencarian jalur A* (A-Star *pathfinding*) pada karakter musuh dilakukan guna memverifikasi apakah mekanisme pergerakan tersebut telah beroperasi sesuai dengan skenario rancangan yang diharapkan.

Berikut ini adalah tabel skenario pengujian untuk algoritma A* (A-Star) yang diterapkan pada musuh:

Tabel 4. 7 Tabel Skenario Pengujian Algoritma A-Star (A*)

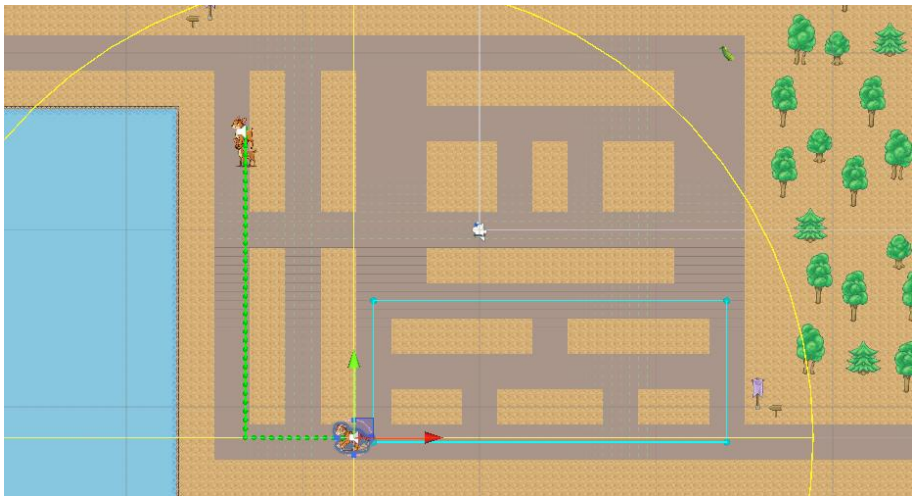
No	Skenario Pengujian	Kondisi Awal (<i>Input</i>)	Hasil yang Diharapkan (<i>Expected Output</i>)
1	Jalur Tanpa Hambatan	Musuh dan Pemain berada di satu garis lurus tanpa dinding di antaranya.	Musuh bergerak lurus langsung menuju pemain tanpa berbelok.
2	Hambatan Sederhana	Ada satu dinding lurus menghalangi jalan antara Musuh dan Pemain.	Musuh bergerak memutar ujung dinding terdekat untuk mencapai pemain (tidak menembus dinding).
3	Hambatan Bentuk U (Jebakan)	Pemain berada di balik dinding berbentuk U atau L.	Musuh tidak terjebak di dalam cekungan U, musuh mundur atau memutar jauh untuk mencari jalan masuk ke arah pemain.
4	Target Bergerak (Dinamis)	Pemain bergerak berpindah posisi saat musuh sedang mengejar.	Musuh mengubah arah jalurnya secara <i>real-time</i> mengikuti posisi terbaru pemain.
5	Target Tidak Terjangkau	Pemain berada di area tertutup total (dikelilingi dinding penuh).	Musuh berhenti pada titik terdekat yang bisa dicapai atau diam.

Pengujian terhadap penerapan algoritma A* dilakukan menggunakan metode Black Box Testing dengan pendekatan pengujian perilaku

(*behavioral testing*). Fokus pengujian diarahkan pada kemampuan agen cerdas (musuh) dalam melakukan navigasi jalur terpendek dan penghindaran rintangan (*obstacle avoidance*) di dalam arena labirin. Skenario pengujian dirancang dalam beberapa kondisi spasial yang berbeda guna memvalidasi responsivitas algoritma terhadap perubahan posisi target dan konfigurasi dinding.

Tabel 4. 8 Tabel Hasil Pengujian Black Box Algoritma A-Star (A*)

No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
1.	Pengejaran tanpa halangan	Musuh bergerak lurus ke arah pemain.	Musuh bergerak lurus mendekati pemain.	[Valid/Berhasil]
2.	Pengejaran dengan halangan tembok	Musuh mencari jalan memutar menghindari tembok.	Musuh berhasil membelok sebelum menabrak tembok.	[Valid/Berhasil]
3.	Pengejaran target bergerak	Musuh memperbarui jalur saat pemain berpindah.	Musuh mengubah arah seketika mengikuti pemain.	[Valid/Berhasil]



Gambar 4. 26 Pengujian Algoritma A-Star (A*)

Pada skenario pengujian dengan kompleksitas spasial berupa dinding penghalang (Gambar 4.25), algoritma A* diuji kemampuannya dalam melakukan navigasi penghindaran rintangan (*obstacle avoidance*). Berdasarkan tampilan layar, terlihat bahwa ketika jalur langsung menuju pemain tertutup oleh objek dinding (*unwalkable node*), sistem navigasi musuh secara otomatis melakukan kalkulasi ulang jalur (*pathfinding recalculation*). Musuh tidak menabrak dinding, melainkan bergerak menyusuri sisi rintangan untuk menemukan celah atau jalur alternatif

terdekat. Respons ini memvalidasi bahwa implementasi algoritma A* pada karakter musuh telah berfungsi baik dalam membedakan area yang dapat dilalui dan yang tidak.

Analisis hasil pengujian menunjukkan bahwa penggunaan algoritma A* mampu meningkatkan perilaku pergerakan musuh menjadi lebih dinamis dan adaptif terhadap posisi *player*. Musuh tidak bergerak secara acak, melainkan mengikuti jalur yang optimal berdasarkan struktur labirin. Dengan demikian, penerapan algoritma A* dalam *game* Android 2D *escape* labirin ini dapat dikatakan efektif dalam mendukung mekanisme permainan dan memberikan tantangan yang lebih menarik bagi pemain.

4.5.3 Analisis Kelebihan Dan Kekurangan Algoritma A-Star (A*)

Berdasarkan hasil pengujian sistem pada *game* Android 2D *escape* labirin yang telah dikembangkan, penerapan algoritma A* (A-Star) sebagai kecerdasan buatan musuh menunjukkan pengaruh yang signifikan terhadap perilaku pergerakan musuh. Analisis ini dilakukan untuk mengidentifikasi kelebihan dan kekurangan algoritma A* berdasarkan hasil implementasi dan pengujian yang telah dilakukan pada setiap level permainan.

a. Kelebihan Algoritma A-Star (A*)

Berdasarkan hasil pengujian, kelebihan algoritma A-Star (A*) dalam penelitian ini adalah sebagai berikut:

Tabel 4. 9 Tabel Kelebihan Algoritma A-Star (A*)

No.	Deskripsi
1	Algoritma A* mampu menentukan jalur terpendek secara optimal dengan mempertimbangkan biaya aktual dan estimasi menuju target, sehingga pergerakan musuh terlihat lebih cerdas dan terarah.
2	Algoritma A* bersifat adaptif terhadap perubahan posisi pemain, dibuktikan dengan kemampuan musuh untuk menghitung ulang jalur ketika pemain berpindah lokasi.
3	Algoritma A* efektif dalam menghindari rintangan pada lingkungan labirin berbasis <i>grid</i> , sehingga musuh tidak menabrak dinding atau terjebak pada jalur buntu.
4	Penerapan algoritma A* meningkatkan tingkat tantangan permainan karena musuh mampu mengejar pemain melalui jalur yang efisien dan tidak mudah diprediksi.
5	Algoritma A* dapat diimplementasikan dengan baik pada <i>game</i> 2D menggunakan <i>Unity</i> , khususnya pada peta berbasis <i>tilemap</i> .

b. Kekurangan Algoritma A-Star (A*)

Selain memiliki kelebihan, hasil pengujian juga menunjukkan beberapa keterbatasan algoritma A-Star (A*) sebagai berikut:

Tabel 4. 10 Kekurangan Algoritma A-Star (A*)

No.	Deskripsi
1	Algoritma A* memerlukan proses komputasi yang lebih kompleks dibandingkan pergerakan musuh sederhana, sehingga berpotensi memengaruhi performa pada perangkat dengan spesifikasi rendah.
2	Jumlah <i>node</i> yang diproses oleh algoritma A* meningkat seiring bertambahnya ukuran dan kompleksitas labirin, yang dapat berdampak pada efisiensi sistem.
3	Kinerja algoritma A* sangat bergantung pada pemilihan fungsi <i>heuristik</i> , <i>heuristik</i> yang kurang tepat dapat mengurangi efisiensi pencarian jalur.
4	Algoritma A* hanya berfokus pada pencarian jalur terpendek dan belum mempertimbangkan variasi perilaku kecerdasan buatan lain seperti strategi menyerang atau menghindari yang lebih kompleks.
5	Perhitungan ulang jalur secara terus-menerus pada target yang bergerak memerlukan pengelolaan yang baik agar tidak menyebabkan penurunan performa permainan.

4.5.4 Pengujian Dari Pengguna

Pengujian dilakukan melalui penyebaran kuesioner langsung kepada pihak ketiga untuk mengevaluasi fungsionalitas *game*. Data yang diperoleh dianalisis menggunakan skala Likert dengan skala 1 hingga 5.

Tabel 4. 11 Tabel Rentang Nilai Skala Likert

No.	Rentang Nilai	Keterangan
1	0% - 19%	Tidak Baik
2	20% - 39%	Kurang Baik
3	40% - 59%	Cukup
4	60% - 79%	Baik
5	80% - 100%	Sangat Baik

Tabel 4. 12 Tabel Hasil Kuesioner Dari Pengguna

No.	Pertanyaan	Tidak Baik (TB)	Kurang Baik (KB)	Cukup (C)	Baik (B)	Sangat Baik (SB)	Nilai
1	Apakah tampilan <i>game</i> “Petualangan Si Kancil” sudah cukup menarik?	-	-	-	2	14	78
2	Apakah <i>game</i> “Petualangan Si	-	-	-	1	15	79

No.	Pertanyaan	Tidak Baik (TB)	Kurang Baik (KB)	Cukup (C)	Baik (B)	Sangat Baik (SB)	Nilai
	Kancil” mudah dipahami?						
3	Apakah <i>controller player</i> pada <i>game</i> “Petualangan Si Kancil” sudah dapat digunakan dengan baik?	-	-	1	5	10	73
4	Apakah fungsi tombol pada <i>game</i> “Petualangan Si Kancil” sudah dapat digunakan dengan baik?	-	-	-	2	14	78
5	Apakah jalan cerita pada <i>game</i> “Petualangan Si Kancil” mudah dipahami?	1	-	1	1	13	73
6	Apakah <i>game</i> "Petualangan Si Kancil" ini membantu kamu memperkenalkan karakter Si Kancil?	-	-	-	3	13	77
7	Apakah suara pada <i>game</i> “Petualangan Si Kancil” berjalan dengan baik?	-	-	-	2	14	78
8	Apakah Item yang ada pada <i>game</i> “Petualangan Si Kancil” sudah bisa digunakan dengan baik?	-	-	-	2	14	78
9	Apakah musuh sudah mempunyai kecerdasan buatan mengejar saat	-	-	-	1	15	79

No.	Pertanyaan	Tidak Baik (TB)	Kurang Baik (KB)	Cukup (C)	Baik (B)	Sangat Baik (SB)	Nilai
	permainan dimulai?						
10	Apakah Anda tertarik untuk memainkannya kembali?	-	-	1	4	11	74

Berdasarkan rekapitulasi data hasil pengujian yang disajikan dalam tabel, diperoleh temuan bahwa mayoritas respons responden terdistribusi secara dominan pada kategori 'Sangat Baik' (SB) dan 'Baik' (B) di seluruh butir pertanyaan. Aspek visual, kemudahan pemahaman mekanisme permainan, serta kecerdasan buatan (AI) pada karakter musuh mendapatkan apresiasi tertinggi, di mana hampir seluruh responden memberikan penilaian maksimal. Kendati ditemukan preferensi minor pada skala 'Cukup' (C) terkait kontrol pemain dan alur cerita, serta satu respons terisolasi pada skala 'Tidak Baik' (TB) mengenai pemahaman cerita, secara agregat hasil kuesioner mengindikasikan bahwa permainan 'Petualangan Si Kancil' telah beroperasi sesuai spesifikasi dan dapat diterima dengan baik oleh target pengguna.

Hasil rekapitulasi data memperoleh total skor keseluruhan sebanyak 767 dan rata-rata skor per item adalah 76.7. Rincian perhitungan yang didapat adalah sebagai berikut:

Tabel 4. 13 Tabel Rincian Perhitungan Kuesioner

No.	Rincian Perhitungan
1	$(2 \times 4) + (14 \times 5) = 8 + 70 = 78$
2	$(1 \times 4) + (15 \times 5) = 4 + 75 = 79$
3	$(1 \times 3) + (5 \times 4) + (10 \times 5) = 3 + 20 + 50 = 73$
4	$(2 \times 4) + (14 \times 5) = 8 + 70 = 78$
5	$(1 \times 1) + (1 \times 3) + (1 \times 4) + (13 \times 5) = 1 + 3 + 4 + 65 = 73$
6	$(3 \times 4) + (13 \times 5) = 12 + 65 = 77$
7	$(2 \times 4) + (14 \times 5) = 8 + 70 = 78$
8	$(2 \times 4) + (14 \times 5) = 8 + 70 = 78$
9	$(1 \times 4) + (15 \times 5) = 4 + 75 = 79$
10	$(1 \times 3) + (4 \times 4) + (11 \times 5) = 3 + 16 + 55 = 74$

Selanjutnya perhitungan persentase dilakukan dengan membandingkan Skor Aktual (jumlah nilai yang didapat) dengan Skor Maksimal (nilai tertinggi yang mungkin didapat).

$$\text{Persentase} = \left(\frac{\text{Total Skor Aktual}}{\text{Total Skor Maksimal}} \right) \times 100\% \quad (4.1)$$

Keterangan:

Jumlah Responden: 16 Orang (jumlah frekuensi per butir soal)
 Skor Maksimal per Item: 16 (responden) X 5 (skala tertinggi) = 80
 Total Skor Maksimal Keseluruhan: 80 X 10 (butir pertanyaan) = 800

Tabel 4. 14 Tabel Hasil Perhitungan Persentase

No.	Pertanyaan	Skor Aktual	Skor Maksimal	Persentase (%)	Kategori
1	Tampilan <i>game</i> menarik	78	80	97.5%	Sangat Baik
2	<i>Game</i> mudah dipahami	79	80	98.75%	Sangat Baik
3	<i>Controller player</i> responsif	73	80	91.25%	Sangat Baik
4	Fungsi tombol berjalan baik	78	80	97.5%	Sangat Baik
5	Jalan cerita mudah dipahami	73	80	91.25%	Sangat Baik
6	Memperkenalkan karakter Si Kancil	77	80	96.25%	Sangat Baik
7	Suara/Audio berjalan baik	78	80	97.5%	Sangat Baik
8	Item (Timun) bisa digunakan	78	80	97.5%	Sangat Baik
9	Kecerdasan buatan (AI) Musuh	79	80	98.75%	Sangat Baik
10	Tertarik memainkannya kembali	74	80	92.5%	Sangat Baik

Berdasarkan perhitungan di atas, diperoleh total persentase kelayakan sebesar 95,88%. Jika merujuk pada skala interpretasi kelayakan yang umum digunakan. Maka, dapat disimpulkan bahwa aplikasi *game* "Petualangan Si Kancil" masuk dalam kategori "Sangat Layak" untuk diimplementasikan dan digunakan.

4.6. Tampilan Layar

Tampilan layar menggambarkan dan menjelaskan tentang hasil implementasi dari rancangan layar permainan yang dibuat. Tampilan ini dibuat tertata dan menarik sehingga berfokus pada keseimbangan antara estetika visual dan fungsionalitas untuk meningkatkan pengalaman pengguna (UX). Hasil implementasi ini biasanya menampilkan elemen-elemen yang konsisten dengan tema permainan, terorganisir dengan rapi, dan mudah dinavigasi. Berikut adalah gambar beserta penjelasan dari tampilan layar *game* Petualangan Si Kancil.

4.6.1 Tampilan Layar Menu Utama

Menu utama di tampilkan pada saat pengguna mencoba memainkan permainan. Pada tampilan menu utama ini terdapat beberapa fitur yang dapat dinikmati oleh pengguna mulai dari tombol *start game*, tombol *exit*, tombol pengaturan dan tombol *credits*. Desain pada tampilan utama ini tidak secara keseluruhan meliputi teks sehingga mempertegas fungsi dari tombol tersebut.



Gambar 4. 27 Tampilan Menu Utama

4.6.2 Tampilan Layar Keluar Permainan

Tampilan konfirmasi untuk keluar permainan ditampilkan pada saat pengguna mencoba untuk keluar permainan dengan cara menekan tombol dengan ikon *exit* pada saat berada di tampilan menu utama. Fungsi dari tampilan ini adalah untuk mencegah pengguna keluar secara tidak sengaja. Tombol centang pada tampilan ini akan menutup permainan dan tombol x pada tampilan ini akan kembali ke tampilan menu utama.



Gambar 4. 28 Tampilan Keluar Permainan

4.6.3 Tampilan Layar Menu Pengaturan

Tampilan menu pengaturan ini terdapat 3 *slider* untuk mengubah nilai dari volume music, volume SFX dan volume suara cerita. Pengaturan audio disimpan menggunakan *PlayerPrefs* pada *Unity* sehingga

preferensi volume pemain dapat dipertahankan meskipun permainan ditutup dan dibuka kembali. Tampilan ini juga tersedia *toggle* tombol untuk melewati cerita untuk Prolog dan Epilog yang tersedia pada setiap level pada *game* Petualangan Si Kancil. Pengaturan ini juga disimpan melalui *PlayerPrefs* pada unity sehingga preferensi cara bermain pengguna dapat dipertahankan meskipun permainan ditutup dan dibuka kembali. Tombol x pada tampilan ini akan mengarahkan pengguna kembali ke tampilan menu utama.



Gambar 4. 29 Tampilan Menu Pengaturan

4.6.4 Tampilan Layar *Credits*

Tampilan tentang pembuat *game* yang muncul setelah pengguna menekan tombol dengan ikon *credits* pada tampilan menu utama. Tombol x pada tampilan ini akan mengarahkan pengguna kembali ke tampilan menu utama. Tombol *Reset Data* pada tampilan ini akan mengarahkan pengguna ke dalam tampilan untuk konfirmasi *reset data*.



Gambar 4. 30 Tampilan *Credits*

4.6.5 Tampilan Layar Reset Data

Tampilan konfirmasi untuk melakukan reset data ini akan menghapus seluruh pengalaman bermain dengan menghapus seluruh level dan menghapus bintang yang telah diperoleh seluruh levelnya. Tombol centang untuk melakukan *reset data* lalu pengguna diarahkan ke

tampilan menu utama dan tombol x untuk diarahkan kembali ke tampilan menu utama.



Gambar 4. 31 Tampilan *Reset Data*

4.6.6 Tampilan Layar Menu Level

Tampilan untuk memilih level memiliki konsep untuk melakukan penguncian untuk level yang belum seharusnya dijangkau oleh pengguna, pengguna tidak dapat mengakses level 2 sebelum pengguna menyelesaikan level 1 terlebih dahulu dan pengguna tidak dapat mengakses level 3 sebelum pengguna menyelesaikan level 2. Konsep ini diterapkan dengan tujuan mencegah pengguna melewati tutorial yang tersedia pada level 1.



Gambar 4. 32 Tampilan Menu Level

Pada tampilan ini juga menampilkan jejak dari hasil perolehan bintang yang dilakukan oleh pengguna melalui penyelesaian level.



Gambar 4. 33 Tampilan Menu Level 2

4.6.7 Tampilan Layar Prolog Level 1

Tampilan ini akan memicu suara narasi untuk menceritakan kisah awal Si Kancil ketika pengguna membuka level 1. Pada tampilan prolog akan selalu diawali dengan judul untuk level 1. Pada tampilan ini terdapat tombol *next* untuk melanjutkan membaca kisah hingga prolog selesai dilalui.



Gambar 4. 34 Tampilan Prolog Level 1

4.6.8 Tampilan Layar Cara Bermain

Cara bermain hanya ditampilkan satu kali saja pada saat pengguna telah selesai membaca prolog pada level 1. Tampilan ini akan memberitahu pengguna terkait aturan di dalam *game* sehingga pengguna dapat mengetahui cara menyelesaikan tiap level pada *game*.



Gambar 4. 35 Tampilan Cara Bermain

4.6.9 Tampilan Layar Level 1

Pengguna diharuskan menuju *finish* untuk menyelesaikan level. Pengguna disajikan dengan tampilan *Head-Up Display* (HUD), pada pojok kiri atas tampilan ini terdapat tombol jeda untuk menghentikan permainan sementara dan kemungkinan membuka menu opsi. Menampilkan ikon kepala karakter utama (si Kancil), yang berfungsi sebagai indikator visual siapa yang sedang dimainkan. Terdapat tiga ikon hati menunjukkan jumlah nyawa yang dimiliki pemain.

Pada tampilan ini di sudut pojok kanan atas terdapat ikon mentimun angka kiri menunjukkan jumlah mentimun yang sudah dikumpulkan saat ini dan angka kanan menunjukkan total target mentimun yang dapat dikumpulkan. Lalu di tengah atas tampilan terdapat penunjuk waktu berupa durasi yang telah berjalan. Di pojok kiri bawah Ini adalah *controller analog* di layar sentuh (*on-screen control*) yang digunakan pemain untuk menggerakkan karakter ke berbagai arah di dalam labirin.



Gambar 4. 36 Tampilan Level 1

4.6.10 Tampilan Layar Epilog Level 1

Kondisi akhir permainan dibedakan antara kemenangan dan kekalahan, di mana kemenangan memicu tampilan narasi epilog sebelum hasil permainan ditampilkan, sedangkan kekalahan langsung menampilkan hasil permainan. Pada akhir epilog selalu disajikan pesan

moral untuk pengalaman bermain yang menyenangkan. Pada tampilan ini terdapat tombol *next* untuk melanjutkan membaca kisah hingga epilog selesai dilalui.



Gambar 4. 37 Tampilan Epilog Level 1

4.6.11 Tampilan Layar Prolog Level 2

Tampilan ini akan memicu suara narasi untuk menceritakan kisah awal Si Kancil ketika pengguna membuka level 2. Pada tampilan prolog akan selalu diawali dengan judul untuk level 2. Pada tampilan ini terdapat tombol *next* untuk melanjutkan membaca kisah hingga prolog selesai dilalui.



Gambar 4. 38 Tampilan Prolog Level 2

4.6.12 Tampilan Layar Level 2

Pengguna diharuskan menuju *finish* untuk menyelesaikan level sekaligus menghindari pengejaran Harimau. Pengguna disajikan dengan tampilan *Head-Up Display* (HUD), pada pojok kiri atas tampilan ini terdapat tombol jeda untuk menghentikan permainan sementara dan kemungkinan membuka menu opsi. Menampilkan ikon kepala karakter utama (si Kancil), yang berfungsi sebagai indikator visual siapa yang sedang dimainkan. Terdapat tiga ikon hati menunjukkan jumlah nyawa yang dimiliki pemain.

Pada tampilan ini di sudut pojok kanan atas terdapat ikon mentimun angka kiri menunjukkan jumlah mentimun yang sudah dikumpulkan saat

ini dan angka kanan menunjukkan total target mentimun yang dapat dikumpulkan. Lalu di tengah atas tampilan terdapat penunjuk waktu berupa durasi yang telah berjalan. Di pojok kiri bawah Ini adalah *controller analog* di layar sentuh (*on-screen control*) yang digunakan pemain untuk menggerakkan karakter ke berbagai arah di dalam labirin.



Gambar 4. 39 Tampilan Level 2

4.6.13 Tampilan Layar Epilog Level 2

Kondisi akhir permainan dibedakan antara kemenangan dan kekalahan, di mana kemenangan memicu tampilan narasi epilog sebelum hasil permainan ditampilkan, sedangkan kekalahan langsung menampilkan hasil permainan. Pada akhir epilog selalu disajikan pesan moral untuk pengalaman bermain yang menyenangkan. Pada tampilan ini terdapat tombol *next* untuk melanjutkan membaca kisah hingga epilog selesai dilalui.



Gambar 4. 40 Tampilan Epilog Level 2

4.6.14 Tampilan Layar Prolog Level 3

Tampilan ini akan memicu suara narasi untuk menceritakan kisah awal Si Kancil ketika pengguna membuka level 3. Pada tampilan prolog akan selalu diawali dengan judul untuk level 3. Pada tampilan ini terdapat tombol *next* untuk melanjutkan membaca kisah hingga prolog selesai dilalui.



Gambar 4. 41 Tampilan Prolog Level 3

4.6.15 Tampilan Layar Level 3

Pengguna diharuskan menuju *finish* untuk menyelesaikan level sekaligus menghindari pengejaran Harimau. Pengguna disajikan dengan tampilan *Head-Up Display* (HUD), pada pojok kiri atas tampilan ini terdapat tombol jeda untuk menghentikan permainan sementara dan kemungkinan membuka menu opsi. Menampilkan ikon kepala karakter utama (si Kancil), yang berfungsi sebagai indikator visual siapa yang sedang dimainkan. Terdapat tiga ikon hati menunjukkan jumlah nyawa yang dimiliki pemain.

Pada tampilan ini di sudut pojok kanan atas terdapat ikon mentimun angka kiri menunjukkan jumlah mentimun yang sudah dikumpulkan saat ini dan angka kanan menunjukkan total target mentimun yang dapat dikumpulkan. Lalu di tengah atas tampilan terdapat penunjuk waktu berupa durasi yang telah berjalan. Di pojok kiri bawah Ini adalah *controller analog* di layar sentuh (*on-screen control*) yang digunakan pemain untuk menggerakkan karakter ke berbagai arah di dalam labirin.



Gambar 4. 42 Tampilan Level 3

4.6.16 Tampilan Layar Epilog Level 3

Kondisi akhir permainan dibedakan antara kemenangan dan kekalahan, di mana kemenangan memicu tampilan narasi epilog sebelum hasil permainan ditampilkan, sedangkan kekalahan langsung

menampilkan hasil permainan. Pada tampilan ini terdapat tombol *next* untuk melanjutkan membaca kisah hingga epilog selesai dilalui.



Gambar 4. 43 Tampilan Epilog Level 3

4.6.17 Tampilan Layar Jeda Permainan

Pada tampilan ini terdapat 3 tombol yaitu tombol kembali ke menu utama, tombol melakukan *restart* pada level dan tombol melanjutkan permainan.



Gambar 4. 44 Tampilan Jeda Permainan

4.6.18 Tampilan Layar Menang

Pada tampilan ini akan menampilkan status nyawa pengguna dan hasil perolehan bintang melalui pengumpulan mentimun pada tiap levelnya, bintang yang dapat diperoleh dengan ketentuan maksimal 3 bintang. Pengguna dapat kembali ke menu utama dengan menekan tombol menu utama dan melakukan *restart* untuk mencoba kembali level tersebut. Pengguna juga dapat menekan tombol *next* untuk melanjutkan ke level berikutnya.



Gambar 4. 45 Tampilan Menang

4.6.19 Tampilan Layar Kalah

Pengguna dapat kembali ke menu utama dengan menekan tombol menu utama dan melakukan *restart* untuk mencoba kembali level tersebut.



Gambar 4. 46 Tampilan Kalah

4.6.20 Tampilan Layar Tamat

Setelah pengguna melalui level 3 lalu menekan tombol *next*, maka pengguna diarahkan ke tampilan selesai menamatkan permainan Petualangan Si Kancil. Pada tampilan ini terdapat tombol untuk melihat menu *credits* dan tombol untuk kembali ke menu utama.



Gambar 4. 47 Tampilan Tamat

4.6.21 Tampilan Layar Kembali Ke Menu Utama

Tampilan untuk konfirmasi kembali ke menu utama yang hanya tersedia ketika pengguna telah menamatkan permainan.



Gambar 4. 48 Tampilan Kembali Ke Menu Utama

BAB V PENUTUP

5.1. Kesimpulan

Mengacu pada hasil analisis dan pengujian sistem yang telah dilakukan, maka kesimpulan penelitian ini dapat diuraikan sebagai berikut:

- a. Penerapan algoritma A* (A-Star) sebagai kecerdasan buatan dalam *game* ini dilakukan melalui penyusunan skrip pemrograman C# pada *game engine Unity*. Area permainan dipetakan menggunakan sistem *grid* yang terdiri dari sekumpulan *node*, di mana setiap *node* merepresentasikan titik yang dapat dilalui (*walkable*) atau rintangan (*obstacle*) untuk diproses oleh logika navigasi musuh.
- b. Penentuan jalur pergerakan musuh secara optimal dicapai melalui mekanisme perhitungan biaya jalur (*path cost*) menggunakan rumus *heuristik* $f(n) = g(n) + h(n)$. Algoritma bekerja dengan mengevaluasi *node* tetangga yang memiliki nilai biaya terendah dari posisi awal hingga target, sekaligus mengeliminasi area dinding sebagai jalur terlarang, sehingga terbentuk rute terpendek yang paling efisien.
- c. Perilaku responsif musuh terhadap pergerakan pemain dibentuk melalui mekanisme pembaruan target secara *real-time*. Ketika koordinat pemain berubah, sistem navigasi secara otomatis memicu kalkulasi ulang (*re-calculation*) jalur, sehingga musuh dapat mengubah arah pengejaran secara dinamis dan terus-menerus menyesuaikan rute untuk mengejar posisi terbaru pemain di setiap level.
- d. Integrasi cerita rakyat Petualangan Si Kancil sebagai elemen naratif utama dalam *game* memberikan nilai tambah dari sisi edukatif dan budaya, serta menjadi pembeda penelitian ini dengan penelitian sebelumnya yang hanya berfokus pada aspek teknis algoritma.
- e. Berdasarkan hasil pengujian fungsional, penerapan algoritma A* dalam *game* ini terbukti mampu meningkatkan tantangan permainan dan memberikan pengalaman bermain yang lebih interaktif.

5.2. Saran

Berdasarkan hasil penelitian serta keterbatasan sistem yang telah diidentifikasi, diajukan beberapa rekomendasi untuk pengembangan sistem maupun penelitian lanjutan di masa mendatang, yaitu:

- a. Jumlah level permainan dapat ditambah dan desain struktur labirin dirancang dengan tingkat kompleksitas yang lebih tinggi guna meningkatkan variasi serta tantangan bagi pengguna.
- b. Algoritma A* (A-Star) dapat dikombinasikan dengan metode kecerdasan buatan lain, seperti *Finite State Machine* (FSM) atau *Behavior Tree*, agar perilaku musuh yang dihasilkan menjadi lebih dinamis dan tidak monoton.
- c. Optimasi kinerja algoritma perlu dilakukan pada pengembangan selanjutnya agar aplikasi dapat beroperasi secara lebih efisien,

khususnya pada perangkat Android dengan spesifikasi perangkat keras yang terbatas (*low-end*).

- d. Cakupan konten budaya dapat diperluas dengan mengintegrasikan unsur cerita rakyat Indonesia lainnya, sehingga fungsi *game* sebagai media pelestarian budaya digital menjadi lebih komprehensif.
- e. Mekanisme permainan dapat diperkaya melalui penambahan variasi rintangan di dalam labirin, baik berupa item penghambat baru maupun modifikasi struktur peta yang lebih dinamis.

DAFTAR PUSTAKA

- Abiyyulian Iqbal. (2025). *ANALISIS NILAI KARAKTER BUKU DONGENG*. <https://repository.radenintan.ac.id/42084/2/BAB%201%202%20DAPUS.pdf>
- Agung, E. G., Eridani, D., & Fauzi, A. (2022). *Implementasi Metode Pathfinding dengan Algoritma A* pada Game Rogue-like menggunakan Unity*. <https://doi.org/10.34818/indojc.2022.7.3.677>
- Ai Tuti Kusmiati, Shinta Purnamasari, & Andinisa Rahmaniar. (2024). Analisis Pengaruh Penerapan Game dalam Pembelajaran IPA. *JURNAL PENDIDIKAN MIPA*, 14(2), 498–510. <https://doi.org/10.37630/jpm.v14i2.1595>
- Alexander, K. R., & Sahertian, J. (2023). Penerapan Metode A-Star Pada Permainan Bergenre Platformer Menggunakan Godot. In *Agustus* (Vol. 7). Online.
- Alif Pratama, F., Sahay, A. S., & Nugrahaningsih, N. (2025). Perbandingan Algoritma Dijkstra dan A-Star dalam Pencarian Rute Terpendek Sekolah Menengah Atas (SMA) di Kota Palangka Raya Berbasis Website. *JOINTECOMS (Journal of Information Technology and Computer Science) p-ISSN: 2798-284X*, 5(1), 2798–3862. <https://doi.org/10.47111/jointecom.v5i1>
- Apriono, D., Sutrisno, Mahmudah, & Moh. Mu'minin. (2024). *Pengembangan Metode Story Telling melalui Permainan Tradisional Anak Berbasis Kearifan Lokal untuk Meningkatkan Kemampuan Berbahasa Anak TK* (Vol. 7, Number 5). <https://doi.org/10.54371/jiip.v7i5.4347>
- Fallo, D. Y. A., & Bulu, V. R. (2022). PENERAPAN ALGORITMA A STAR (A*) PADA GAME LABIRIN. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, (5), 2621–1467. <https://doi.org/10.37792/jukanti.v5i1.459>
- Joseph, L. D. S., Saputro, I. P., & Adrian, A. M. (2021). “Finding Easter Eggs” An Augmented Reality Based Game Using A-Star Algorithm. *Cogito Smart Journal* |, 7(1). <https://doi.org/10.31154/cogito.v7i1.289.15-25>
- Muhammad Reyhandi Akbar, Ilka Zufria, & Aninda Muliani Harahap. (2024). Implementasi Algoritma A Star Pada Sistem Informasi Geografis Sekolah Luar Biasa di Kota Medan. *Journal of Computers and Digital Business*, 3(1), 18–25. <https://doi.org/10.56427/jcbd.v3i1.243>
- Octavian, F., & Hermawan, L. (2023). Penerapan Algoritma Pathfinding A* dalam Game Dual Legacy berbasis Android. In *Jurnal Buana Informatika* (Vol. 14, Number 1). <https://doi.org/10.24002/jbi.v14i01.6928>
- Oktavianto, M. D., & Rizqi, M. (2024). Rancang Bangun Pathfinding Dengan Algoritma A* (Star) Pada NPC Menggunakan Unity Engine. *Jurnal Ilmu Komputer Dan Bisnis*, 15(2a), 7–23. <https://doi.org/10.47927/jikb.v15i2a.831>
- Padila, N., Basri, B., & Sari, C. R. (2023). Sistem informasi geografis dengan algoritma a-star untuk menentukan jalur terdekat. *Journal Pegguruang: Conference Series*, 5(1), 370. <https://doi.org/10.35329/jp.v5i1.4063>
- Saputra, R. K., & Setiawan, K. (2024). Pengembangan Game 2D Platformer Berbasis Microbit Menggunakan Unity. *Jurnal Indonesia: Manajemen Informatika Dan Komunikasi*, 5(1), 20–25. <https://doi.org/10.35870/jimik.v5i1.420>
- Setiawan, M. L., Arbansyah, A., & Suryawan, S. H. (2023). Penerapan Algoritma A* Dan Behaviour Trees Untuk Perilaku Non-Player Character(NPC) Pada

- Game “The Last Hope” Berbasis Android Menggunakan Unity 2D. *Jurnal CoSciTech (Computer Science and Information Technology)*, 4(2), 451–460.
<https://doi.org/10.37859/coscitech.v4i2.5418>
- Suharto, P. (2002). *Petualangan Si Kancil*.
<https://repositori.kemendikdasmen.go.id/3418/>
- Yohanes, D. N., & Rochmawati, N. (2022). Implementasi Algoritma Collision Detection dan A*(A Star) pada Non Player Character Game World Of New Normal. *Journal of Informatics and Computer Science*, 03.
<https://doi.org/10.26740/jinacs.v3n03.p322-333>



**PEMERINTAH PROVINSI DAERAH KHUSUS JAKARTA
KOTAMADYA ADMINISTRASI JAKARTA SELATAN
KECAMATAN KEBAYORAN LAMA KELURAHAN GROGOL UTARA
RUKUN WARGA [RW] 12/1**

Sekretariat Jl. Uranium No.1 Telepon / Hp : 081218298767-085607418856-083895423568-081806346581

SURAT KETERANGAN RISET

Yang bertanda tangan dibawah ini:

Nama : Dedi Andriyadi
Jabatan : Ketua RT.006

Menerangkan bahwa:

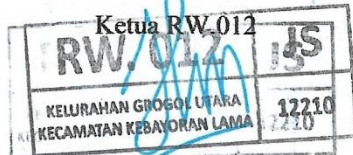
Nama : Fariz Irvansyah
NIM : 2211500141

Telah melaksanakan riset di lingkungan RW.012, Kelurahan Grogol Utara, Kecamatan Kebayoran Lama sejak 09 Desember 2025 s.d. 18 Januari 2026 dengan baik.

Demikian Surat Keterangan ini dibuat untuk dapat dipergunakan semestinya.

Dibuat di : Jakarta

Tanggal : 18 Januari 2026



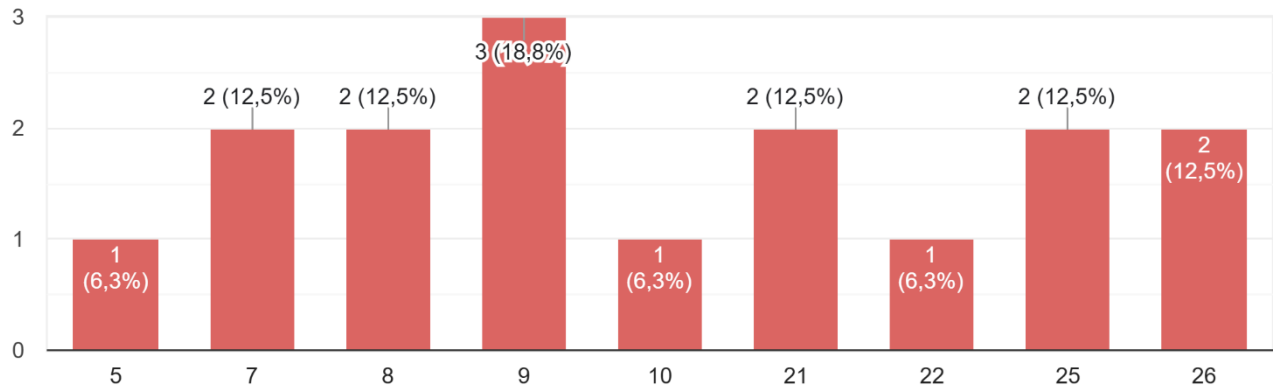
(M. Irfan Farhan Hartono)

Ketua RT.006



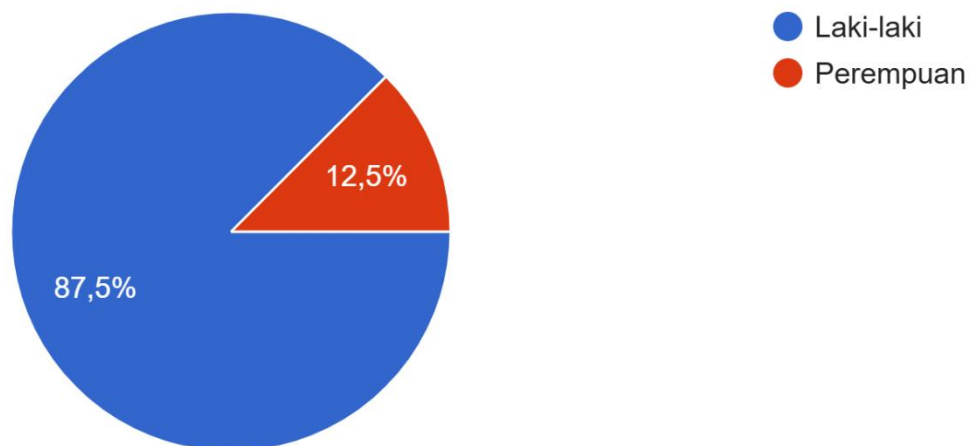
Usia

16 jawaban



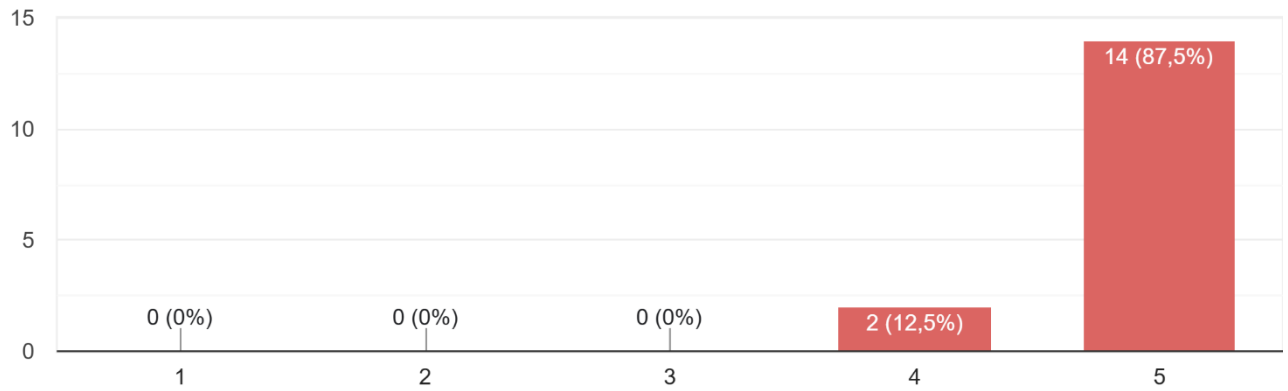
Jenis Kelamin

16 jawaban



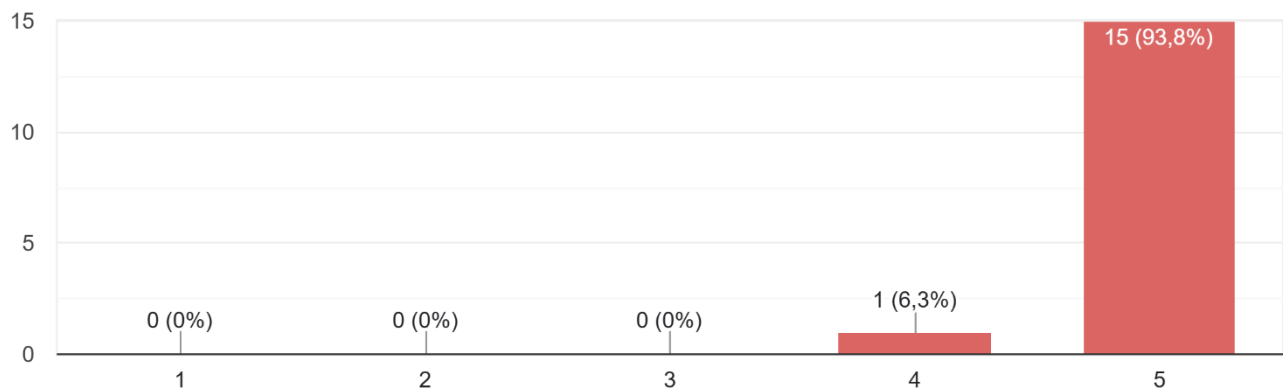
Apakah tampilan game “Petualangan Si Kancil” sudah cukup menarik?

16 jawaban



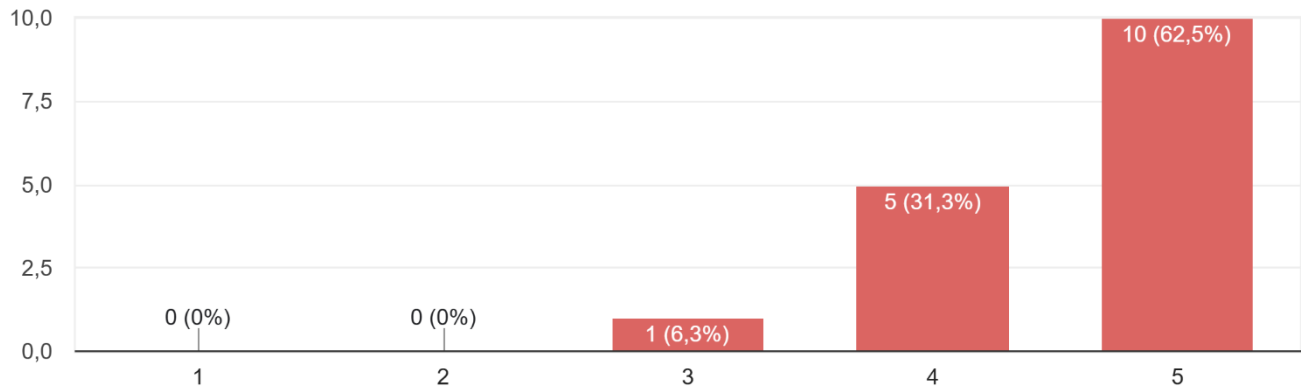
Apakah game “Petualangan Si Kancil” mudah dipahami?

16 jawaban



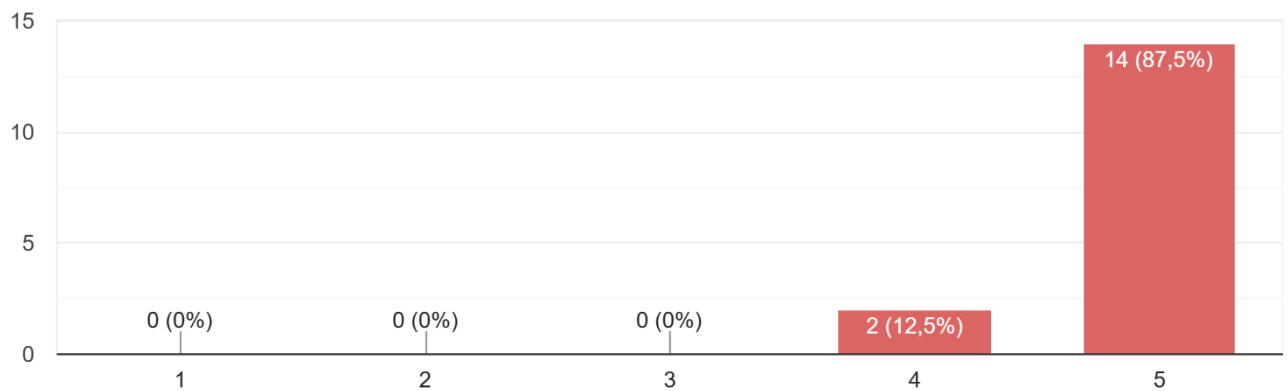
Apakah controller player pada game “Petualangan Si Kancil” sudah dapat digunakan dengan baik?

16 jawaban



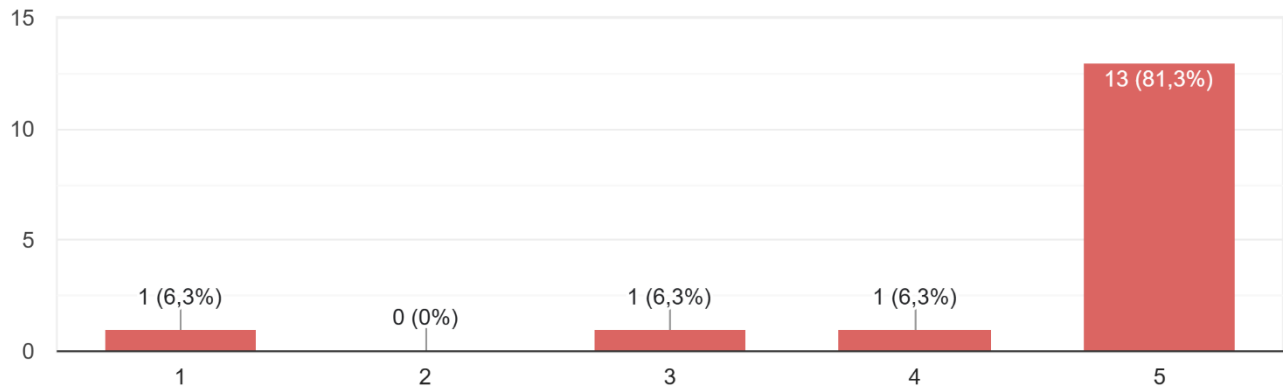
Apakah fungsi tombol pada game “Petualangan Si Kancil” sudah dapat digunakan dengan baik?

16 jawaban



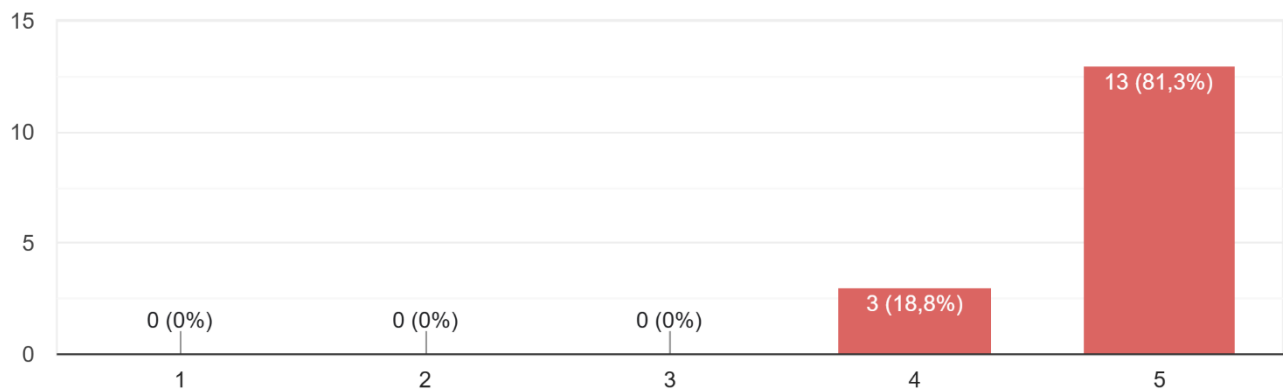
Apakah jalan cerita pada game “Petualangan Si Kancil” mudah dipahami?

16 jawaban



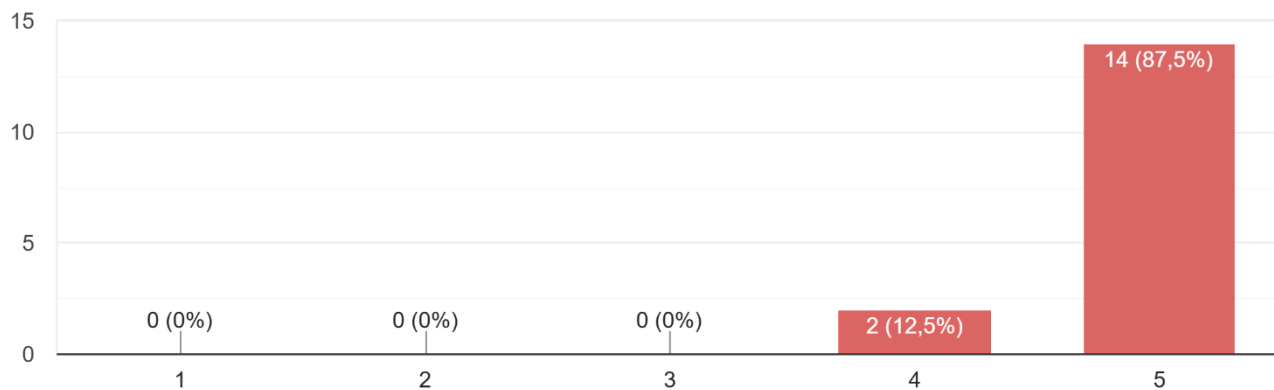
Apakah game "Petualangan Si Kancil" ini membantu kamu memperkenalkan karakter Si Kancil?

16 jawaban



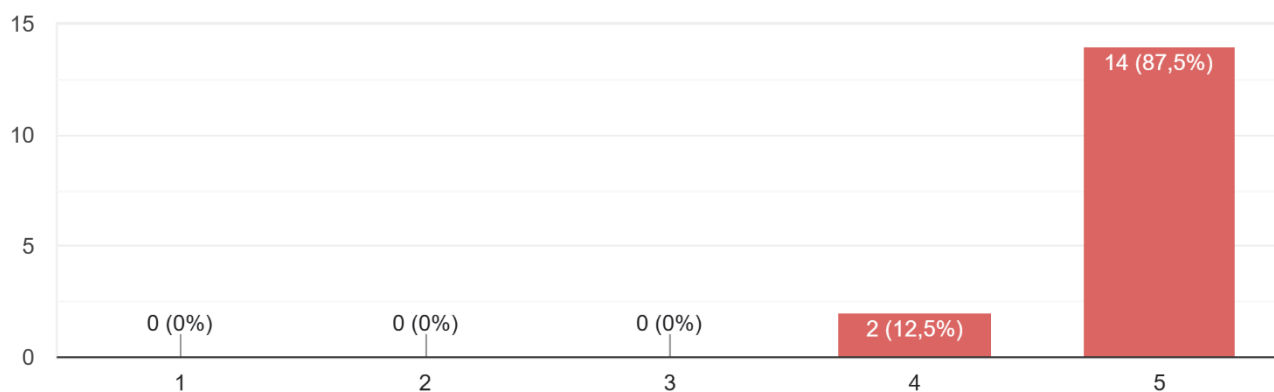
Apakah suara pada game “Petualangan Si Kancil” berjalan dengan baik?

16 jawaban



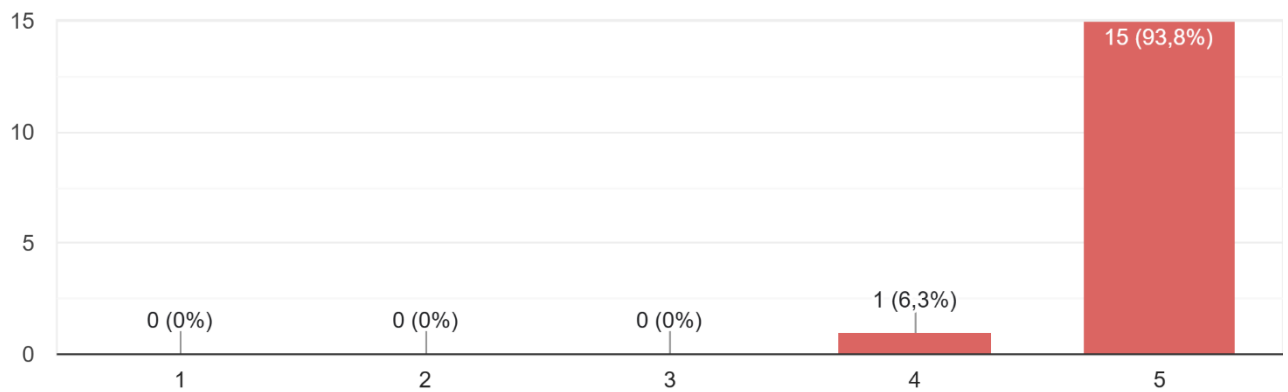
Apakah Item yang ada pada game “Petualangan Si Kancil” sudah bisa digunakan dengan baik?

16 jawaban



Apakah musuh sudah mempunyai kecerdasan buatan mengejar saat permainan dimulai?

16 jawaban



Apakah anda tertarik untuk memainkannya kembali?

16 jawaban

