

**PENERAPAN ALGORITMA A *STAR* PADA NPC UNTUK
PENCARIAN JALUR TERDEKAT DALAM *GAME*
PETUALANGAN BERBASIS ANDROID**

SKRIPSI

**ZULFIKRI BAHRI
NIM. 71154037**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

**PENERAPAN ALGORITMA A STAR PADA NPC UNTUK
PENCARIAN JALUR TERDEKAT DALAM GAME
PETUALANGAN BERBASIS ANDROID**

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

**ZULFIKRI BAHRI
NIM. 71154037**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth.,

Dekan Fakultas Sains dan Teknologi

Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Warahmatullahi Wabarakathu

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama : Zulfikri Bahri

Nomor Induk Mahasiswa : 71154037

Program Studi : Ilmu Komputer

Judul : Penerapan Algoritma A *Star* Pada NPC Untuk Pencarian Jalur Terdekat Dalam *Game* Petualangan Berbasis Android

Dapat disetujui untuk segera *dimunaqasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Medan, 28 Januari 2021 M

15 Jumadil Akhir 1442 H

Komisi Pembimbing,

Pembimbing I,

Pembimbing II,

Dr. Mhd. Furqan, S.Si.,M.Comp.Sc.
NIP. 19800806 200604 1 003

Muhammad Ikhsan, S.T., M.Kom
NIP. 19830415 201101 1 008

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama : Zulfikri Bahri
Nomor Induk Mahasiswa : 71154037
Program Studi : Ilmu Komputer
Judul : Penerapan Algoritma A *Star* Pada NPC Untuk
Pencarian Jalur Terdekat Dalam *Game*
Petualangan Berbasis Android

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing–masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai peraturan yang berlaku.

Medan, 28 Januari 2021

Zulfikri Bahri
NIM. 71154037



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. IAIN No. 1 Medan 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url: <http://saintek.uinsu.ac.id>, Email: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor: B.037/ST/ST.V.2/PP.01.1/03/2022

Judul : Penerapan Algoritma A *Star* Pada NPC Untuk
Pencarian Jalur Terdekat Dalam *Game* Petualangan
Berbasis Android
Nama : Zulfikri Bahri
Nomor Induk Mahasiswa : 71154037
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu
Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan
dinyatakan **LULUS**.

Pada hari/tanggal : Selasa, 28 Januari 2021
Tempat/media : Via Zoom Meeting

Tim Ujian Munaqasyah,
Ketua,

Ilka Zufria, M.Kom
NIP. 198506042015031006

Dewan Penguji,
Penguji I, Penguji II,

Dr. Mhd. Furqan, S.Si., M.Comp.Sc.
NIP. 19800806 200604 1 003

Muhammad Ikhsan, S.T., M.Kom
NIP. 19830415 201101 1 008

Penguji III,

Penguji IV,

Abdul Halim Hasugian, M.Kom
NIB. 1100000113

Heri Santoso, M.Kom
NIB. 1100000114

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. Mhd. Syahnan, M.A
NIP. 196609051991031002

ABSTRAK

Google *play store* dan berbagai website memiliki banyak sekali *game mobile* berbasis android. Beberapa individu memang menghasilkan *game* dan memainkannya untuk mencari kekayaan dan sebagai bisnis. *Game* merupakan cara paling populer untuk mengisi waktu atau melepas penat setelah seharian bekerja. Pada penelitian ini, dirancang sebuah *game* bergenre *arcade* dikarenakan *game* yang bergaya ini disukai dan cocok untuk anak-anak karena melatih otak untuk memecahkan masalah dengan cepat sekaligus melatih otak, mata, dan tangan untuk berkonsentrasi lebih baik. Dimana nantinya pemain akan menghadapi NPC (*Non Playable Character*) di sepanjang jalan dan harus terus menembaki mereka untuk mengumpulkan poin sebanyak mungkin dalam batas waktu dan NPC telah diatur dengan kecerdasan buatan yang dimana NPC telah diterapkan algoritma *A star*. Algoritma *A star* akan digunakan untuk pergerakan NPC yang bergerak sesuai dengan posisi pemain. UML (*Unified Modelling Language*) merupakan bahasa pemrograman yang memanfaatkan perangkat lunak *Monodevelop* dan *Unity 3D* digunakan dalam desain *game* penelitian ini. Temuan penelitian ini telah diuji, menunjukkan bahwa menggabungkan *Unity 3D* dan menerapkan algoritma *A star* ke musuh dapat menghasilkan *game* yang sukses. Dimana NPC yang telah diterapkan algoritma *A star* dapat melacak jalur terdekat pemain.

Kata kunci : *Game arcade, Android, Kecerdasan buatan , Algoritma A star*

ABSTRACT

Google play store and various websites have lots of android based mobile games. Some individuals do produce games and play them for wealth and as a business. Games are the most popular way to pass the time or unwind after a long day at work. In this study, an arcade genre game was designed because this stylish game is liked and suitable for children because it trains the brain to solve problems quickly while training the brain, eyes, and hands to concentrate better. Where later players will face NPCs (Non Playable Characters) along the way and have to keep shooting at them to collect as many points as possible within the time limit and NPCs have been set with artificial intelligence where the NPCs have applied the A star algorithm. The A star algorithm will be used for the movement of NPCs that move according to the player's position. UML (Unified Modeling Language) is a programming language that utilizes Monodevelop and Unity 3D software used in the game design of this research. The findings of this study have been tested, showing that combining Unity 3D and applying the A star algorithm to enemies can result in a successful game. Where the NPC that has applied the A star algorithm can track the player's closest path.

Keywords: Arcade games, Android, Artificial intelligence, A star Algorithm

KATA PENGANTAR

Alhamdulillah, dengan segala puji dan rasa syukur penulis panjatkan untuk kehadirat Allah Subhanahu Wata'ala, karena hanya dengan bantuan dan pertolongannya penulis dapat menyelesaikan skripsi yang berjudul **“Penerapan Algoritma A Star Pada NPC Untuk Pencarian Jalur Terdekat Dalam Game Petualangan Berbasis Android”**, ini dapat diselesaikan untuk memenuhi salah satu persyaratan untuk mendapatkan kelulusan Program Sarjana (S1) Jurusan Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.

Penulis banyak menemui kendala dan hambatan dalam menempuh perjalanan panjang dalam proses penulisan skripsi ini, tetapi berkat kehendak-Nya penulis dapat menyelesaikan skripsi ini. Oleh karena itu, dengan rendah hati penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Syahrin Harahap, MA selaku rektor Universitas Islam Negeri Sumatera Utara.
2. Bapak Dr. Mhd. Syahnan, M.A. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
3. Bapak Ilka Zufria, M. Kom. selaku Ketua Program Studi Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
4. Bapak Rakhmat Kurniawan R, ST., M.Kom sebagai Sekretaris Program Studi Ilmu Komputer Universitas Islam Negeri Sumatera Utara.
5. Bapak Dr. Mhd. Furqan, S.Si, M.Comp.Sc. sebagai Dosen Pembimbing I yang telah membimbing, menasehati serta memberikan kontribusi dalam penyelesaian penulisan skripsi ini.
6. Bapak Muhammad Ikhsan S.T., M.Kom sebagai Pembimbing II yang telah membimbing, menasehati serta memberikan kontribusi dalam penyelesaian penulisan skripsi ini.
7. Ayah dan Bunda saya yang sudah membimbing dengan penuh kasih sayang serta membagikan makna suatu kesabaran dalam menempuh kehidupan, dan

kepada keluarga besar program studi Ilmu Komputer stambuk 2015.

8. Serta seluruh pihak yang telah membantu penulis semampunya.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari sempurna karena keterbatasan pengetahuan penulis yang lemah. Oleh karenanya, penulis sangat membutuhkan saran dan kritik yang membangun akan penulis terima dengan tulus dengan ikhlas. Semoga Allah SWT membalas seluruh kebaikan yang diterima penulis, amin ya rabbal alaimin.

Medan, 28 Januari 2021

Penulis,

ZULFIKRI BAHRI

NIM. 71154037

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
DAFTAR LAMPIRAN	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.1 Pengertian <i>Game</i>	4
2.2 Algoritma A* (<i>A Star</i>)	4
2.3 Android	5
2.4 NPC (<i>Non Playable Character</i>)	5
2.5 Unity	5
2.6 Flowchart	6
2.7 Multimedia	7
2.8 Pengujian Kotak Hitam (<i>Black Box Testing</i>)	7
2.9 Penelitian Terkait	7
BAB III METODE PENELITIAN	10
3.1 Perangkat Aplikasi	10
3.1.1 Perangkat Keras (<i>Hardware</i>)	10
3.1.2 Perangkat Lunak (<i>Software</i>)	10
3.2 Analisis <i>Game</i>	10
3.3 Perancangan <i>Unified Modeling Language</i> (UML)	11

3.4 <i>Use Case Diagram</i>	12
3.5 <i>User Interface Game</i>	13
3.5.1 <i>Tampilan Awal Game</i>	13
3.5.2 <i>Menu Utama</i>	14
3.5.3 <i>Profil Panel</i>	15
3.5.4 <i>How To</i>	15
3.5.5 <i>Game Play</i>	16
3.6 <i>Algoritma Game</i>	17
3.6.1 <i>Skema Perilaku NPC</i>	17
3.6.2 <i>Algoritma A Star</i>	17
3.6.3 <i>Probabilitas Perilaku NPC Terhadap Player</i>	18
3.6.4 <i>Penerapan Algoritma A Star</i>	19
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	26
4.1 <i>Implementasi</i>	26
4.2 <i>Implementasi Antar Muka</i>	26
4.2.1 <i>Tampilan Awal Game</i>	26
4.2.2 <i>Menu Utama</i>	27
4.2.3 <i>Profile Panel</i>	28
4.2.4 <i>How To</i>	28
4.2.5 <i>Game Play</i>	29
4.3 <i>Implementasi Algoritma A Star</i>	31
4.4 <i>Pengujian Algoritma A Star</i>	31
4.5 <i>Pengujian Kotak Hitam</i>	32
BAB V KESIMPULAN DAN SARAN	34
5.1 <i>Kesimpulan</i>	34
5.2 <i>Saran</i>	34
DAFTAR PUSTAKA	35
LAMPIRAN – LAMPIRAN	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
3.1	<i>Use Case Diagram</i>	12
3.2	Tampilan Awal <i>Game</i>	13
3.3	Tampilan Main Menu	14
3.4	Tampilan <i>Panel Profile</i>	15
3.5	Tampilan <i>How To</i>	15
3.6	Tampilan <i>Game Play</i>	16
3.7	Skema Perilaku NPC	17
3.8	Probabilitas Perilaku NPC	18
4.1	Tampilan <i>Press To Play</i>	26
4.2	Tampilan <i>Main Menu</i>	27
4.3	Tampilan <i>Profile Game</i>	28
4.4	Tampilan <i>Panel How To</i>	28
4.5	Tampilan <i>Game Play</i>	29
4.6	Tampilan Musuh Mulai Mendekati Pemain	30
4.7	Tampilan <i>Game Play Status Enemy</i>	30

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1	<i>Flowchart</i>	6
3.1	Kebutuhan <i>Hardware</i>	10
3.2	Kebutuhan <i>Software</i>	10
3.3	Deskripsi <i>Use Case Diagram</i>	12
3.4	Pergerakan NPC	19
3.5	Pergerakan NPC	20
3.6	Pergerakan NPC	21
3.7	Pergerakan NPC	22
3.8	Pergerakan NPC	23
3.9	Pergerakan NPC	24
3.10	Pergerakan NPC	25
4.1	Uji Coba Algoritma A <i>Star</i>	31
4.2	Pengujian Kotak Hitam	33

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1	Listing Program
2	Daftar Riwayat Hidup
3	Kartu Bimbingan Skripsi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Google play store dan berbagai website memiliki banyak sekali *game mobile* berbasis android. Android adalah sistem operasi seluler layar sentuh berbasis Linux yang berjalan di *smartphone* dan tablet. Android inc didirikan pada tahun 2004 dengan bantuan keuangan Google dan kemudian dikembangkan pada tahun 2005. (Kusniyati & Sitanggang, 2016)

Beberapa individu memang menghasilkan *game* dan memainkannya untuk mencari kekayaan dan sebagai bisnis. *Game* merupakan cara paling populer untuk mengisi waktu atau melepas penat setelah seharian bekerja. Hal ini menunjukkan bahwa bermain *game* boleh saja asalkan tidak dilakukan dengan cara yang ilegal, seperti perjudian. Semuanya dijelaskan dalam Surah Al - Maidah ayat 90:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِنَّمَا الْخَمْرُ وَالْمَيْسِرُ وَالْأَنْصَابُ وَالْأَزْلَامُ رَجْسٌ مِنْ عَمَلِ الشَّيْطَانِ فَاجْتَنِبُوهُ لَعَلَّكُمْ تُفْلِحُونَ

Artinya : “Hai orang-orang yang beriman, sesungguhnya minum mabuk, berjudi, (berkorban untuk) berhala, mengundi nasib dengan panah, adalah perbuatan syaitan. Maka jauhilah perbuatan-perbuatan itu agar kamu mendapat keberuntungan”.

Genre permainan *arcade* adalah salah satu yang paling populer, dan orang-orang dari segala usia dapat menikmatinya. Penulis menciptakan *game* ini karena disukai dan cocok untuk anak-anak karena melatih otak untuk memecahkan masalah dengan cepat sekaligus melatih otak, mata, dan tangan untuk berkonsentrasi lebih baik. Faktor lainnya adalah kontrol dasar *game* yang sederhana (tidak banyak tombol).

Untuk mengumpulkan *point* dan melanjutkan ke level berikutnya dalam *game* ini, pemain harus sangat berkonsentrasi dan cepat. Alhasil, *gamer* tidak akan bosan saat memainkannya. *Game* berbasis android untuk *smartphone* dan tablet.

Pemain akan menghadapi NPC (*Non Playable Character*) di sepanjang jalan dan harus terus menembaki mereka untuk mengumpulkan *point* sebanyak mungkin dalam batas waktu. *Game* ini dibagi menjadi tiga tingkatan, dengan kesulitan berbeda bagi pemain untuk berinteraksi dengan NPC tergantung pada bagaimana algoritma A star digunakan. NPC (*Non Playable Character*) adalah karakter dalam *game* seperti manusia, hewan, robot, dan makhluk lain yang tidak dapat dikendalikan oleh pemain tetapi melakukan tugas yang dikendalikan oleh kecerdasan buatan. (Safitra et al., 2020)

Algoritma A star adalah strategi pencarian jaringan yang mengunjungi simpul pohon untuk menemukan jalur dari satu keadaan ke keadaan berikutnya. Urutan pencarian ditentukan menggunakan fungsi heuristik. Algoritma ini sangat cocok untuk NPC yang mencari labirin atau ruang untuk pemain. Dalam penelitian Musfiroh, algoritma *boids* digunakan untuk mencari NPC. Algoritma *boids* terbukti tidak efisien ketika diterapkan pada target yang bergerak dalam konfigurasi tertentu, serta dalam *game* tanpa tim. (Musfiroh et al., 2014)

Berdasarkan uraian diatas, maka diangkat judul skripsi **“Penerapan Algoritma A Star Pada NPC Untuk Pencarian Jalur Terdekat Dalam Game Petualangan Berbasis Android”**

1.2 Rumusan Masalah

Berikut ini adalah rumusan masalah dalam penelitian ini, berdasarkan konteks di atas:

1. Bagaimana cara membuat permainan yang membuat pengguna lebih tertarik.
2. Bagaimana pemain dapat menyelesaikan permainan dalam waktu yang singkat.
3. Bagaimana menerapkan algoritma A star pada *game arcade*.

1.3 Batasan Masalah

Karena pengetahuan dan kemampuan penulis yang tidak memadai, penyusunan penelitian ini dibatasi. Berikut adalah batasan-batasan yang ditemui

selama proses penelitian.

1. *Game* hanya bejalan untuk sistem operasi (OS) *android*.
2. *Game* bersifat *single player*.
3. Ada tiga tahap dalam *game* ini, masing-masing dengan rintangan dan masalahnya sendiri.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini sebagaimana dinyatakan dalam latar belakang masalah di atas adalah untuk menerapkan algoritma *A star* untuk menentukan jalur terdekat dari NPC untuk menemukan pemain dalam permainan, sehingga menghasilkan pencarian yang optimal.

1.5 Manfaat Penelitian

Berdasarkan latar belakang diatas, maka manfaat dari hasil penelitian ini adalah sebagai berikut:

- 1 Berkontribusi pada pertumbuhan dunia *game* dan dapat dimanfaatkan sebagai bahan pembelajaran untuk studi masa depan.
- 2 Penelitian ini bertujuan untuk memperluas, menerapkan, dan mengamalkan ilmu yang diperoleh dalam perkuliahan di Universitas Islam Negeri Sumatera Utara Medan.
- 3 Penyelesaian penelitian ini merupakan salah satu syarat untuk memperoleh gelar sarjana di Universitas Islam Negeri Sumatera Utara Medan.

BAB II TINJAUAN PUSTAKA

2.1 Pengertian *Game*

Kata permainan berasal dari kata bahasa Inggris "*game*". Permainan adalah sesuatu yang dimainkan sesuai dengan aturan dan memiliki pemenang dan pecundang. *Game* merupakan salah satu jenis hiburan yang dapat dimanfaatkan untuk menghilangkan kepenatan mental akibat tugas-tugas rutin. (Rizal & Aryanto, 2017) *Game* bisa dikatakan adalah permainan komputer yang dibuat dengan menggunakan metode dan prosedur animasi. (Permana, 2017)

2.2 Algoritma *A star*

Algoritma *A star* adalah pendekatan komputer untuk traversal grafik dan penemuan jalur, serta menghitung jalur terbaik di sekitar *node*. (Rakhmat Kurniawan. R., ST & Yusuf Ramadhan Nasution, 2016)

Dengan menganalisis setiap *node* satu per satu, algoritma *A star* dapat menemukan jalur terpendek. Metode ini akan menghitung jarak satu jalur, menyimpannya, dan kemudian menghitung jarak jalur kedua. Algoritma *A star* akan memilih jalan terpendek ketika semua jalur telah ditemukan. (Hermanto & Dermawan, 2018)

Algoritma *A star* menggunakan notasi berikut: (Dalem, 2018)

$$f(n) = g(n) + h(n)$$

$f(n)$ = estimasi biaya terbaik

$g(n)$ = biaya dari simpul awal ke simpul n

$h(n)$ = menunjukkan biaya yang diharapkan untuk menghubungkan simpul n ke simpul akhir.

Algoritma *A star* adalah salah satu dari banyak algoritma pencarian jalur dan penerusan grafik yang banyak digunakan. Algoritma *A star* yang memimpin dari pengembangan mekanisme utama algoritma BFS (*Best First Search*). Algoritma *A star* mirip dengan algoritma BFS karena menggunakan fungsi heuristik untuk menemukan solusi. (Simanjuntak, 2019)

2.3 Android

Android adalah sistem operasi *smartphone* dan tablet yang berbasis Linux. (Rosiska, 2016) Android adalah *platform* gratis dan terbuka yang memungkinkan pengembang untuk membuat aplikasi mereka sendiri. (Ichwan & Hakiky, 2011) Android inc didirikan pada tahun 2004 dengan bantuan keuangan Google dan kemudian dikembangkan pada tahun 2005. (Kusniyati & Sitanggang, 2016)

2.4 NPC (*Non Playable Character*)

NPC (*Non Playable Character*) adalah karakter dalam *game* seperti manusia, hewan, robot, dan makhluk lain yang tidak dapat dikendalikan oleh pemain tetapi melakukan tugas yang dikendalikan oleh kecerdasan buatan. (Safitra et al., 2020)

Keberadaan NPC (*Non Playable Character*) merupakan salah satu variabel terpenting dalam menentukan menarik atau tidaknya suatu *game*. Dari sudut pandang mode permainan, jika tidak ada NPC hidup yang tidak dapat beradaptasi dan bereaksi, permainan kemungkinan besar akan dianggap membosankan dan tidak menarik sama sekali. (Muttaqin et al., 2017)

2.5 Unity

Unity 3D adalah grafis, suara, *input*, dan prosesor perangkat lunak lainnya yang dapat digunakan untuk membuat *game*, meskipun tidak selalu. Hal yang menakutkan tentang unity adalah ia memiliki kemampuan untuk membuat *game* 3D dan 2D sekaligus sangat mudah digunakan. (Nugroho & Pramono, 2017)

Unity memanfaatkan sistem navigasi gratis yang memungkinkan pengguna melihat ketiga dimensi item yang mereka buat. Ini persis sama seperti menggunakan Blender 3D. (Oktafiani, 2020)

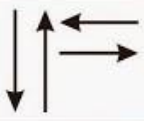






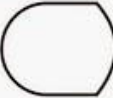







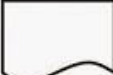
2.6 *Flowchart*

Flowchart adalah alat atau metode untuk menggambarkan tahapan yang harus diikuti dalam mengatasi suatu masalah dalam komputasi melalui penggunaan

simbol grafis tertentu. (Nuraini, 2015)

Flowchart memudahkan untuk mengatasi masalah, terutama yang memerlukan penelitian lebih lanjut. *Flowchart* adalah diagram yang digunakan untuk menunjukkan atau mempresentasikan suatu program. Akibatnya, *flowchart* harus mampu menyampaikan komponen secara mudah dimengerti. (Sutanti et al., 2020)

Flowchart menampilkan sejumlah symbol yang tercantum dalam tabel 2.1

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Tabel 2.1 simbol *flowchart* Sumber : (Sutanti et al., 2020)

2.7 Multimedia

Multimedia mengacu pada penggunaan banyak media dalam satu bentuk komunikasi. Multimedia saat ini mengacu pada penggabungan dan integrasi media ke dalam sistem komputer, seperti teks, animasi, grafik, suara, dan video. Dengan diperkenalkannya monitor komputer resolusi tinggi, teknologi video dan suara, dan upaya untuk meningkatkan pemrosesan komputer pribadi, gagasan tentang multimedia baru-baru ini semakin populer. Misalnya, komputer desktop saat ini memiliki kemampuan untuk merekam suara dan video, mengubah suara dan gambar untuk membuat efek khusus, mencampur dan menghasilkan suara dan video, membuat berbagai bentuk grafik, termasuk animasi, dan mengintegrasikan semua kemampuan ini ke dalam satu bentuk multimedia. (Surasmi, 2016)

2.8 Pengujian Kotak Hitam (*Black Box Tasting*)

Pengujian perangkat lunak dalam hal spesifikasi fungsional tanpa memeriksa desain atau kode program dikenal sebagai pengujian kotak hitam. (Parassa et al., 2017) Tujuan dari pengujian aplikasi adalah untuk melihat apakah suatu program telah beroperasi dengan baik atau ada kekurangan yang perlu diperbaiki agar program yang dikembangkan berkualitas tinggi. (Ningrum et al., 2019)

Teknik *blackbox testing* sederhana untuk diterapkan karena yang dibutuhkan hanyalah batas bawah dan atas untuk data yang diharapkan. Jumlah entri data lapangan yang akan diuji, kriteria yang harus diikuti, dan batasan maksimum dan lebih rendah yang harus dipenuhi semuanya dapat digunakan untuk menghitung perkiraan jumlah data. (Cholifah et al., 2018)

2.9 Penelitian Terkait

(Dalem, 2018) dalam jurnalnya yang berjudul penerapan algoritma A *star* menggunakan grafik untuk menghitung jarak terpendek, hasilnya menunjukkan bahwa Algoritma A *star* dapat digunakan untuk menentukan jalur optimum dan kemudian mengimplementasikannya dengan tepat. Dengan adanya hambatan yang diberikan pada setiap rute, simulasi ini dapat menemukan rute (jalur) yang optimal

dari titik awal hingga titik penyelesaian. Rute yang ditemukan merupakan rute terbaik dengan nilai $f(n)$ terkecil jika dibandingkan dengan rute lainnya, sesuai dengan hasil pengujian.

(Hermanto & Dermawan, 2018) dalam jurnalnya yang berjudul penerapan algoritma *A star* sebagai pencari rute terpendek pada robot *hexapod* memperoleh hasil *A star* dapat diimplementasikan pada robot hexapod untuk membantu dalam menentukan jalur terpendek menuju titik tujuan. Tingkat keberhasilan robot dalam mencapai tujuan yaitu dari langkah robot tidak terhalang oleh rintangan dari arena yang digunakan, dan rata-rata hasil dari pencapaian robot hexapod untuk diproses ke titik tujuan dengan tingkat kesalahan terendah 1 *grid* dan tertinggi 3 *grid*.

(Ahmad & Widodo, 2017) dalam jurnalnya yang berjudul penerapan algoritma *A star* pada *game* petualangan labirin berbasis android didapatkan hasil bahwa heuristik *euclidean* digunakan untuk mengimplementasikan metode *A star* untuk menu bantuan dalam *game* petualangan labirin. Metode *A star* digunakan untuk menentukan jalur tercepat karakter kelinci mencapai makanannya. *Smartphone* yang menjalankan sistem operasi Android versi *Jelly Bean*, *Kitkat*, *Lollipop*, *Marshmallow*, atau *Nougat* dapat memainkan *game* petualangan labirin. Pengontrol layar sentuh dan sensor gravitasi digunakan untuk memainkan *game* petualangan labirin.

(Nuryoso et al., 2020) dalam jurnalnya yang berjudul penerapan algoritma *A star* pada pencarian rute terpendek pada rute angkutan kota di kota Sukabumi didapatkan hasil algoritma *A star* adalah program komputer yang dapat menghitung rute tercepat dan jarak tempuh yang ditempuh. Dengan menghitung jarak antar *node*, maka layak untuk mengetahui jalur terdekat pada rute angkot serta jarak tempuh dari titik awal ke tujuan dalam penelitian ini. Sistem yang dibangun dapat digunakan untuk memberikan informasi tentang rute angkot di kota Sukabumi melalui halaman *web* yang menampilkan lokasi awal rute angkot pada peta.

(Rakhmat Kurniawan. R., ST & Yusuf Ramadhan Nasution, 2016) dalam jurnalnya yang berjudul penerapan algoritma *A star* sebagai solusi pencarian rute terpendek pada *maze* didapatkan hasil algoritma *A star* dapat menentukan jalur terpendek dari *node* awal dan *node* tujuan. Karena algoritma *A star* hanya

menghitung biaya jalan yang dilaluinya, kemungkinan rute yang optimal tidak ditemukan.

(Hermawan & Setiyani, 2019) dalam jurnalnya yang berjudul implementasi algoritma *A star* pada permainan komputer *roguelike* berbasis unity didapatkan algoritma *A star* dapat diterapkan pada proses mengejar lawan melawan karakter dalam *game roguelike*. Proses pencarian jalur terpendek dalam *game* ini dapat dibuat lebih efisien dengan menggunakan algoritma *A star*. Permainan ini dapat dimainkan oleh satu pemain, dan pemain tersebut harus mampu menyusun strategi untuk mencapai garis *finish* dan mengumpulkan *point* terbanyak. Permainan ini dapat membantu peserta dalam memahami logika situasi langsung. Misalnya, bagaimana menghindari batu sandungan, bagaimana mencapai tujuan tertentu, dan sebagainya.

(Oktanugraha & Nudin, 2020) dalam jurnalnya yang berjudul implementasi algoritma *A star* dalam penentuan rute terpendek yang dapat dilalui NPC pada *game good thief* didapatkan bahwa pengujian alpha untuk menilai apakah *game* berjalan dengan lancar juga berhasil dilakukan pada penelitian di atas tentang implementasi algoritma *A star* pada rute terpendek yang dapat dilalui oleh NPC pada *game Good Thief*. Berdasarkan pengujian beta, tingkat kepuasan terhadap *game* yang dimainkan cukup memuaskan yaitu sebesar 83,22 persen, terbukti dari 30 pernyataan yang diberikan kepada 6 responden yang mengikuti pengujian *game Good Thief*.

BAB III

METODE PENELITIAN

3.1 Perangkat Aplikasi

Perangkat aplikasi diklasifikasikan menjadi dua kategori berdasarkan kegunaannya yaitu perangkat pengembang dan perangkat pengguna aplikasi.

3.1.1 Perangkat Keras (*Hardware*)

Berikut daftar perangkat keras yang digunakan dalam penelitian ini:

No	Nama Perangkat Keras
1	Prosesor AMD <i>Athlon Silver</i> 3050U dengan grafis Radeon 2.30 GHz
2	Ruang <i>Harddisk</i> 236 GB
3	RAM 4 GB

Tabel 3.1 *Kebutuhan Hardware*

3.1.2 Perangkat Lunak (*Software*)

Berikut daftar perangkat lunak yang digunakan dalam penelitian ini:

No	Nama Perangkat Lunak
1	Windows 10 64 bit
2	Microsoft <i>Office</i>
3	Adobe Photoshop
4	Unity 2017.1.3

Tabel 3.2 *Kebutuhan Software*

3.2 Analisis Game

Game yang dikembangkan bergenre *action/shooter*. *Game smartphone* ini dimainkan dengan cara menjelajahi area ilustrasi di sebuah ruangan, menembak dan mengalahkan NPC, dan mengumpulkan poin sebanyak mungkin.

Semua NPC dalam *game* ini menggunakan algoritma *A star* untuk mencari lokasi pemain dengan mengantisipasi kondisi mereka saat ini. Misalnya, ketika pemain berada jauh dari NPC maka NPC dapat memperkirakan status pemain dengan menggunakan algoritma *A star* yang ditentukan oleh programmer.

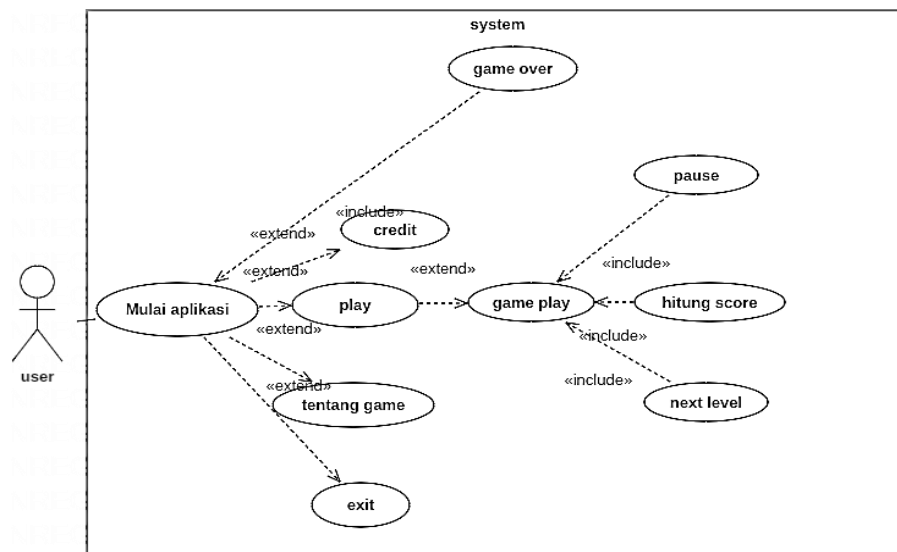
NPC akan mendapatkan halangan yang akan menghalangi pandangan dan pergerakan NPC untuk menemukan pemain, status pertama NPC akan patroli ruangan dengan jarak yang telah ditentukan, kemudian jika status pemain sudah mendekati NPC maka akan mengubah status NPC untuk mengejar pemain, dan setelah pemain berada dalam jangkauan NPC, maka akan berubah status untuk mengejar dan menembak pemain. Dan setiap kali tembakan terjadi, maka mengurangi status darah pemain dan jika status darah pemain adalah 0, permainan berakhir dan dimulai dari awal lagi. Pemain juga dapat menyerang NPC dengan menembak NPC dan setiap kali NPC terkena tembakan pemain, akan mengurangi status darah dari NPC. Jika status darah dari NPC adalah 0, NPC akan mati dan menghilang dari area tersebut.

Game ini menjadi pilihan penulis karena pada *game* nantinya pemain yang memainkan permainan nantinya harus mengandalkan ketangkasan tangan dan kecepatan untuk menyelesaikan permainan. Selain itu *game* berbasis *android* ini ditujukan untuk *smartphone* dan tablet yang sekarang sudah banyak digunakan oleh semua kalangan.

3.3 Perancangan UML (*Unified Modeling Language*)

Pemodelan UML digunakan dalam penelitian ini sebagai sistem pemodelan. *Unified Modeling Language* (UML) adalah bahasa pemodelan standar yang digunakan dalam pembuatan sistem berorientasi objek. Arsitek perangkat lunak membuat diagram UML untuk membantu pengerjaan dalam pembangunan sistem. Pada aplikasi *game* menggunakan algoritma *A star* memiliki diagram *use case* seperti pada gambar 3.1.

3.4 Use Case Diagram



Gambar 3.1 use case diagram

No	Use case	Deskripsi
1	Mulai APK	User mulai masuk mengakses aplikasi.
2	Tentang game	Tentang penggunaan game.
3	Credit	Tentang developer game.
4	Exit	Keluar dari game.
5	Play	Tampilan masuk ke game play.
6	Game play	Mulai mengerjakan misi-misi yang ada pada game.
7	Game over	Kalah ketika dalam menjalankan misi. Maka akan kembali ke awal game.
8	Hitung score	Menghitung poin dari player.
9	Pause	Proses untuk berhenti sementara.
10	Next level	Melanjutkan ke level selanjutnya.
11	Unity	Developer masuk untuk meninjau game.
12	Development	Mengembangkan game.
13	Update	Memperbarui game.

Tabel 3.3 Deskripsi use case diagram

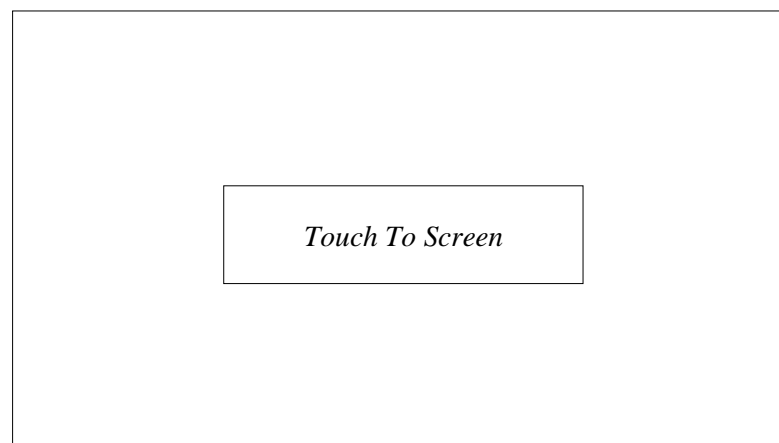
Penulis akan memaparkan kegiatan-kegiatan yang muncul sebagai akibat dari penelitian ini dalam sub bab ini. Penelitian ini merupakan produk berupa aplikasi *game* berbasis android dan dikembangkan menggunakan proses SDLC (*system development life cycle*) dengan model *waterfall*, yaitu model untuk membangun perangkat lunak yang dilakukan secara sistematis dan berurutan, mulai dari dengan analisis, desain, implementasi, pengujian, operasi, dan pengembangan. Diagram kasus akan digunakan dalam penelitian ini.

3.5 User Interface Game

Game ini didesain dengan desain sederhana menggunakan *photoshop*, *coreldraw* dan blender untuk objek 3D, namun cukup untuk kebutuhan permainan dan menerapkan algoritma *A star* pada NPC. Desain tampilan permainan ini adalah sebagai berikut.

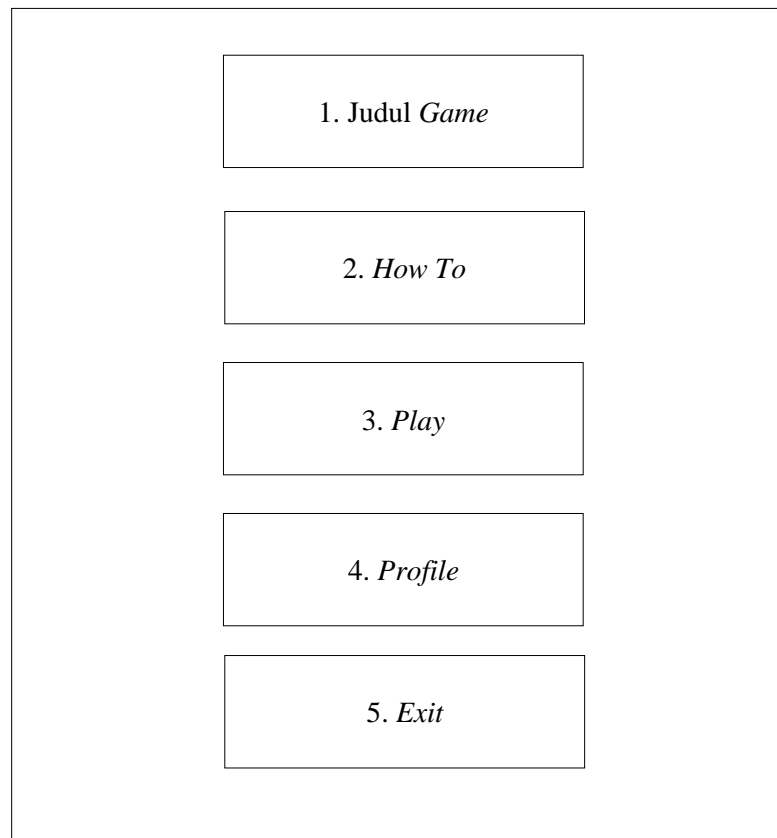
3.5.1 Tampilan Awal Game

Tampilan awal *game* yang berisi ikon *game* yang dirancang oleh penulis sendiri dan layar yang dapat disentuh untuk masuk ke *menu game*.



Gambar 3.2 Tampilan Awal *Game*

3.5.2 Menu Utama

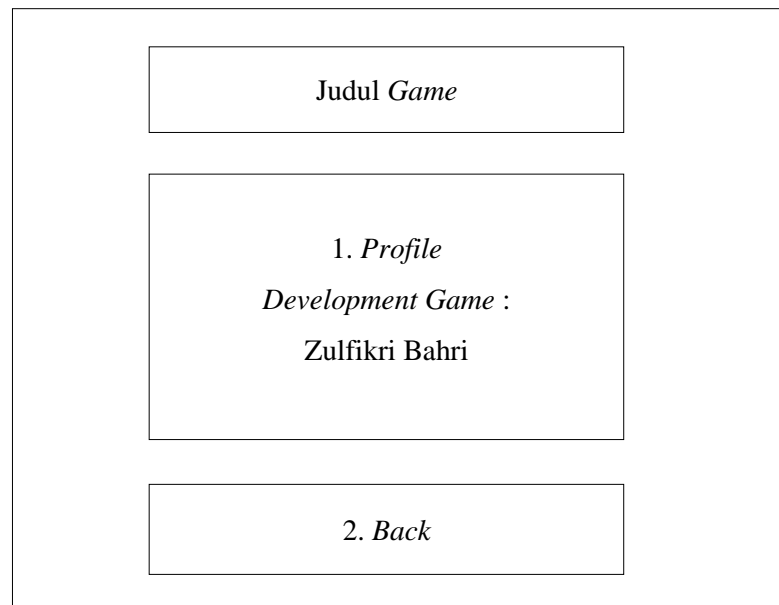


Gambar 3.3 Tampilan Menu Utama

Pada layar menu utama terdapat tombol untuk memasuki setiap adegan atau halaman yang telah disediakan.

- *Judul Game* : Judul *game* hanya sekedar gambar dari judul permainan ini.
- *Button How To* : Tombol untuk masuk ke sebuah *scene*/halaman yang berisikan sebuah video tutorial dalam menjalankan *game* ini.
- *Button Play* : Tombol untuk memulai *game*.
- *Button Profile* : Tombol untuk masuk ke *scene* tentang tim atau *developer* yang mengembangkan *game* ini.
- *Button Exit* : Untuk keluar dari *game* secara permanen.

3.5.3 Profil Panel

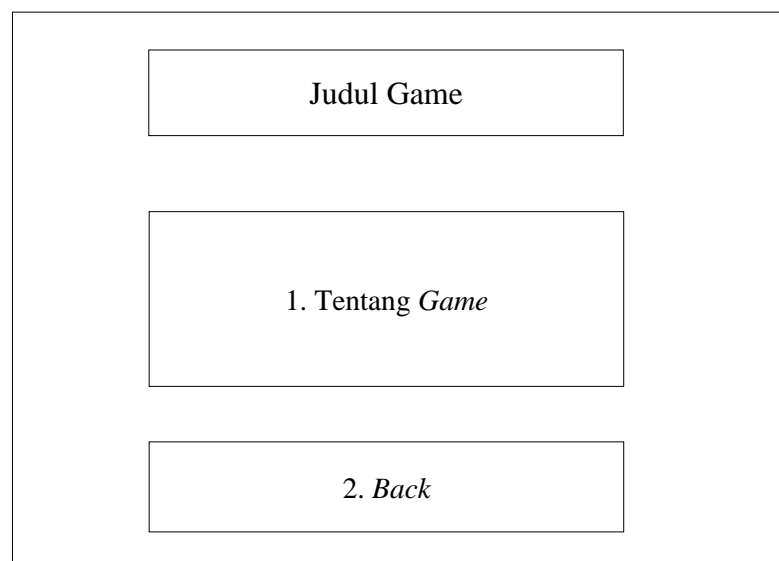


Gambar 3.4 Tampilan Profil Panel

Dalam tampilan *panel profile* ini menunjukkan orang-orang yang bekerja di dalam *game*.

- *Panel Name* : Menampilkan nama dalam tim pengembang/pembuatan *game*.
- *Button Back* : Tombol untuk kembali ke adegan permainan menu.

3.2.3 How To



Gambar 3.5 Tampilan *how to*

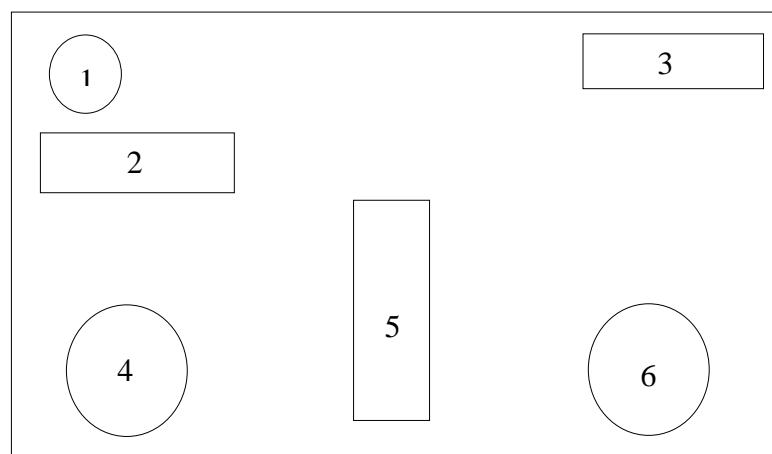
Pada tampilan tentang *game* ini menunjukkan prosedur pengoperasian dalam *game* berupa teks dan gambar.

- Panel tentang *game* : Menampilkan proses kerja dalam *game* sebagai teks dan gambar.
- *Button back* : Tombol untuk kembali ke *scene* menu permainan.

3.5.5 Game Play

Pada tampilan *game play* ini menampilkan tentang gambaran saat kita berada dalam *game*. Tampilan *game play* berisi:

- *Button Pause* : Untuk menghentikan permainan sementara dan dapat dilanjutkan kembali.
- *Health Poin (HP)* : Status darah dari pemain, jika poin darah dari pemain adalah 0 maka pemain akan mati dan permainan akan mengulang kembali.
- *Score* : Tampilan dari poin-poin yang didapatkan oleh pemain.
- *Control Analog* : Permainan dimainkan didalam versi *mobile* maka *control* arah *player* adalah *virtual analog*.
- *Player Position* : Posisi pemain di depan kamera saat permainan sedang berlangsung.
- *Button Menembak* : Untuk menembakkan senjata ke NPC yang mengarah ke titik tengah.



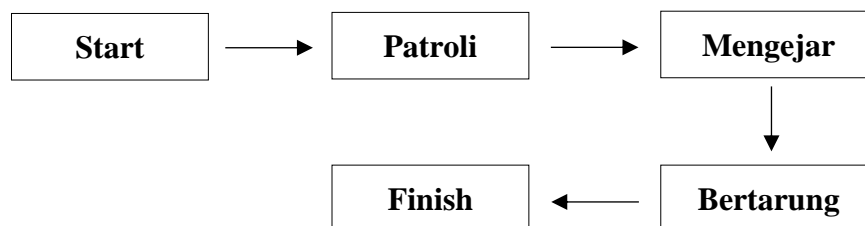
Gambar 3.6 Tampilan *game play*

3.6 Algoritma Game

Skenario yang dihasilkan dalam *game* digunakan sebagai simulasi atau tes dalam penelitian ini. *Player* (pemain) dan NPC (musuh yang menjadi subjek penelitian) adalah dua jenis karakter dalam *game*. Saat memutuskan perilaku NPC saat berhadapan dengan pemain, skenario dengan variabel *input* dan *output* dibuat untuk merujuk perilaku NPC terhadap pemain. Jarak, berpatroli, mengejar, dan menyerang adalah contoh perilaku NPC terhadap pemain.

3.6.1 Skema Perilaku NPC

State utama yang tersusun dalam FSM dapat digambarkan sebagai berikut.



Gambar 3.7 Skema perilaku NPC

3.6.2 Algoritma A star

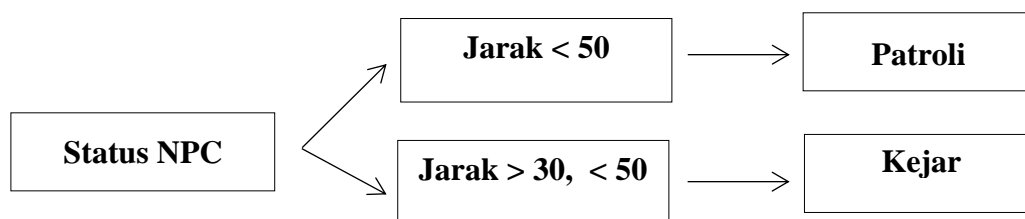
Pada tahap ini akan menggabungkan konsep algoritma *A star* ke dalam NPC (*Non Playable Character*) yang telah dirancang di atas. Proses ini kemudian menghasilkan model komputasi. Penerapan algoritma *A star* digunakan dalam *game* ini untuk membuat perilaku NPC, oleh karena itu NPC memiliki berbagai variabel sebagai parameter untuk melakukannya pada *game*.

Variabel input dan variabel output adalah dua jenis variabel yang digunakan. Status pemain saat mendekati NPC karena jarak termasuk dalam variabel input. Tindakan yang dilakukan oleh NPC terhadap *player* berupa patroli, kejar-kejaran, dan penyerangan merupakan variabel *output*. Dengan variabel-variabel tersebut prosedur penghitungan untuk menentukan perilaku NPC, seperti kapan harus berpatroli, mengejar, dan menyerang dapat dilakukan dengan mudah.

Eksekusi setiap *state* akan mengubah variabel *input*, khususnya jarak saat menggunakan algoritma *A star*. Variabel input ini nantinya akan digunakan sebagai parameter untuk mengontrol perilaku NPC terhadap pemain, seperti patroli, mengejar, dan penyerangan. NPC dilengkapi dengan algoritma *A star* yang digunakan untuk menentukan perilakunya terhadap pemain. Jarak, kesehatan, dan kecepatan adalah semua sifat yang dimiliki karakter pemain. Karakteristik ini kemudian akan digunakan sebagai nilai input untuk menentukan perilaku NPC terhadap pemain. Perilaku NPC akan ditentukan oleh tindakan pemain dengan NPC. Jika pemain jauh dari NPC maka kesehatannya masih seratus persen, oleh karena itu ia dapat bergerak tinggi dan NPC akan berpatroli daripada mengejar atau bertarung.

Karena status pemain dapat berubah kapan saja berdasarkan bagaimana permainan berlangsung. *Input* akan berubah seiring waktu, begitu pula perilaku NPC terhadap pemain. Karena nilai referensi tindakan NPC terhadap pemain selalu berupa rentang nilai, maka harus ditabulasikan sebagai nilai absolut untuk perilaku NPC terhadap parameter pemain. Untuk menghentikan NPC bertindak agresif terhadap pemain, lihat status pemain yang nilainya paling dekat dengan parameter untuk setiap nilai input.

3.6.3 Probabilitas perilaku NPC terhadap *player*



Gambar 3.8 Probabilitas perilaku NPC

Probabilitas perilaku NPC terhadap pemain adalah peluang atau prediksi perilaku yang akan dialami oleh NPC setelah menerima informasi dari tiga titik arah, yaitu jarak, kesehatan dan kecepatan.

3.6.4 Penerapan Algoritma A star

Dalam penerapan algoritma A star, berikut ini adalah contoh representasi dari fungsi matematika.

Rumus:

$$f(n) = g(n) + h(n)$$

Keterangan :

$f(n)$: fungsi dari evaluasi

$g(n)$: biaya yang telah dikeluarkan dari keadaan awal hingga *node* (n)

$h(n)$: estimasi dari biaya yang dikeluarkan dari keadaan n atau *node* (n)

7								
6								
5								
4								
3								
2								
1								
	1	2	3	4	5	6	7	8

Tabel 3.4 Pergerakan NPC

Keterangan :

= NPC/musuh

= *Player* (target)

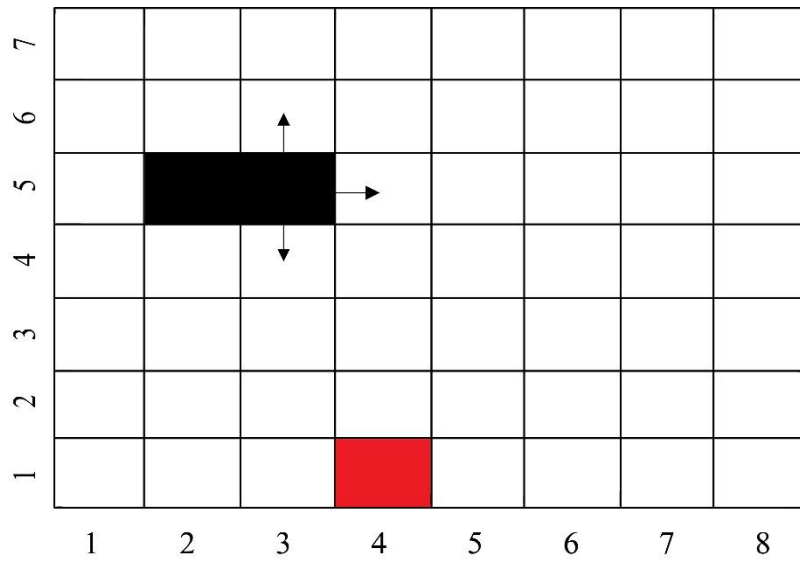
Posisi koordinat NPC adalah 2,5 dan koordinat pemain 4,1, tujuan perjalanan adalah untuk mencapai koordinat pemain yaitu 4,1. Bangkitkan 4 arah perjalanan NPC dan ambil nilai paling kecil.

7								
6								
5		←	■	→				
4			↓					
3								
2								
1				■				
	1	2	3	4	5	6	7	8

Tabel 3.5 Pergerakan NPC

- Ke Kanan = 3,5
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (3,5))$
 $= 1 + (1 + 4) = 6$
- Ke Kiri = 1,5
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (1,5))$
 $= 1 + (3 + 4) = 8$
- Ke Bawah = 2,4
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (2,4))$
 $= 1 + (2 + 3) = 6$

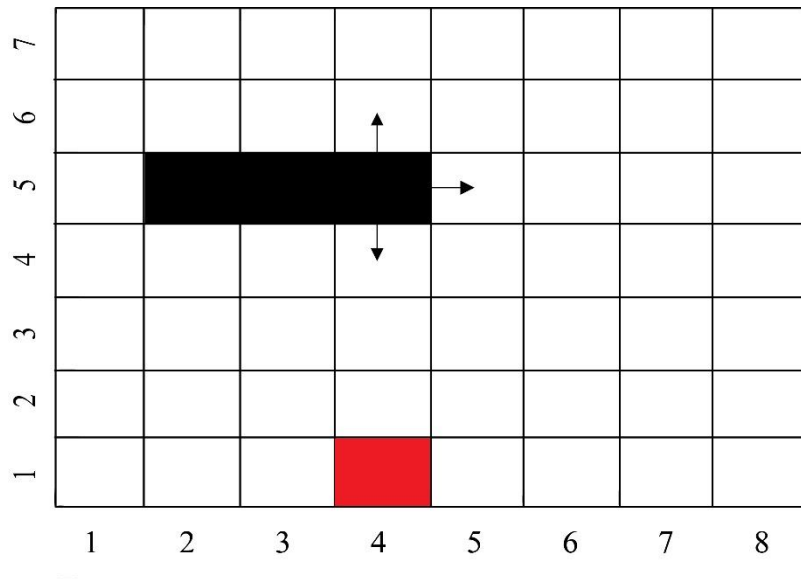
Dari tampilan tabel diumpamakan NPC yang mencari posisi pemain. Hasil perhitungan Algoritma A *star* pada NPC untuk mencari pemain didapatkan dua hasil terkecil yakni enam. Ambil salah satu, misalnya kearah kanan. Didapatkan posisi koordinat dari NPC 3,5 dengan tujuan posisi koordinatnya 4,1.



Tabel 3.6 Pergerakan NPC

- Ke Kanan = 4, 5
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (4,5))$
 $= 1 + (0 + 4) = 5$
- Ke Bawah = 3,4
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (3,4))$
 $= 1 + (1 + 3) = 5$
- Ke Atas = 3,6
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (3,6))$
 $= 1 + (1 + 5) = 7$

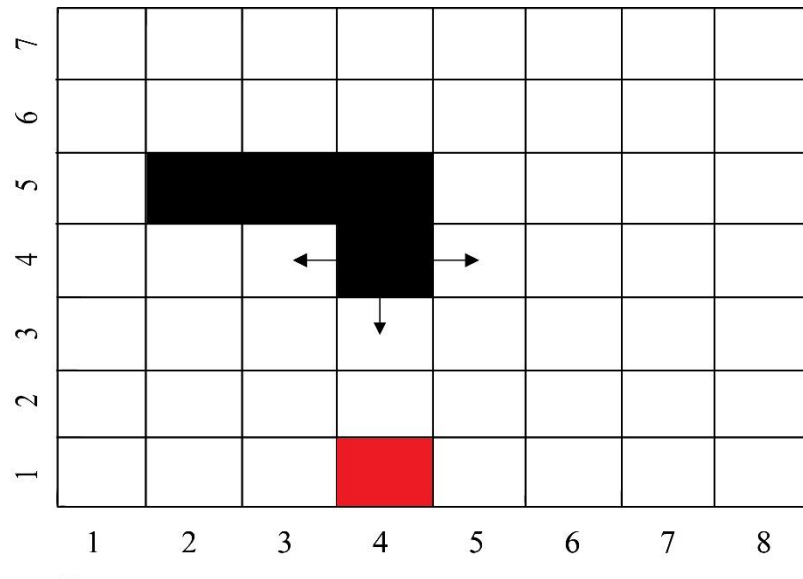
Dari perhitungan di atas, kita mendapatkan dua hasil dengan nilai terkecil yakni lima, jadi ambil salah satu hasil itu dan pindah ke dalam langkah *OPEN*. Kita ambil ke kanan. Didapatkan posisi koordinat dari NPC 4,5 dengan tujuan posisi koordinatnya 4,1.



Tabel 3.7 Pergerakan NPC

- Ke Kanan = 5,5
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (5,5))$
 $= 1 + (1 + 4) = 6$
- Ke Bawah = 4,4
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (4,4))$
 $= 1 + (0 + 3) = 4$
- Ke Atas = 4,6
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (4,6))$
 $= 1 + (0 + 5) = 6$

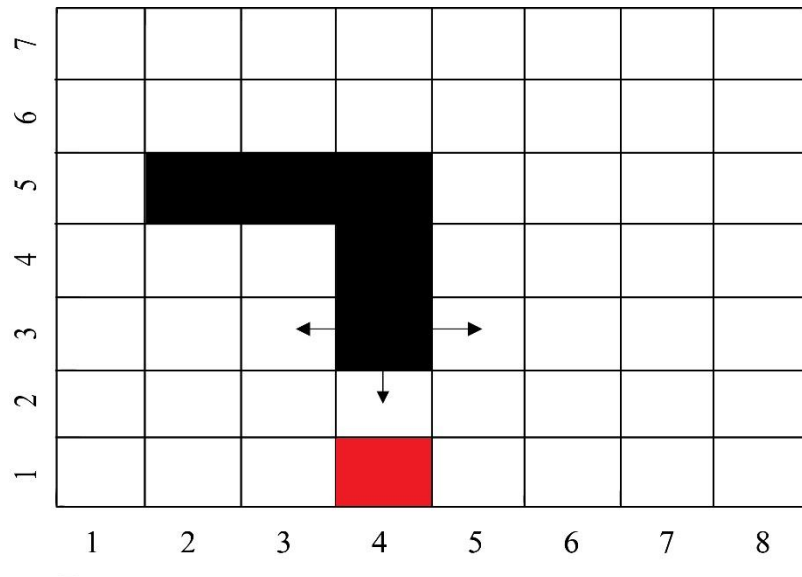
Dari perhitungan yang telah dilakukan, didapatkan nilai terkecil yaitu empat. kita ambil arah ke bawah dengan koordinat 4,4 dan semua titik masuk ke dalam *OPEN* dan yang lainnya *CLOSE*.



Tabel 3.8 Pergerakan NPC

- Ke Kanan = 5,4
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (5,4))$
 $= 1 + (1 + 3) = 5$
- Ke Bawah = 4,3
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (4,3))$
 $= 1 + (0 + 2) = 3$
- Ke Kiri = 3,4
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (3,4))$
 $= 1 + (1 + 3) = 5$

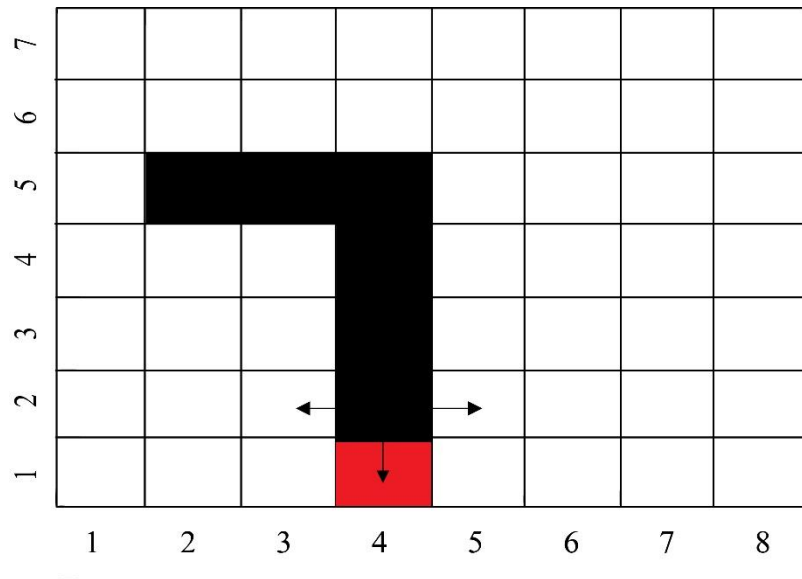
Dari perhitungan di atas, kita mendapatkan hasil nilai terkecil yakni 3. kita ambil arah ke bawah dengan koordinat 4,3 dan semua titik masuk ke dalam *OPEN*.



Tabel 3.9 Pergerakan NPC

- Ke Kanan = 5,3
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (5,3))$
 $= 1 + (1 + 2) = 4$
- Ke Bawah = 4,2
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (4,2))$
 $= 1 + (0 + 1) = 2$
- Ke Kiri = 3,3
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (3,3))$
 $= 1 + (1 + 2) = 4$

Dari perhitungan di atas, kita mendapatkan nilai terkecil yakni 2. kita ambil arah ke bawah dengan koordinat 4,2 dan semua titik masuk ke dalam *OPEN*.



Tabel 3.10 Pergerakan NPC

- Ke Kanan = 5,2
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (5,3))$
 $= 1 + (1 + 2) = 4$
- Ke Bawah = 4,1
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (4,1))$
 $= 1 + (0 + 0) = 1$
- Ke Kiri = 3,2
 $F = g + h$
 $= 1 + \text{abs}((4,1) - (3,2))$
 $= 1 + (1 + 1) = 3$

Dari perhitungan di atas, kita mendapatkan nilai terkecil yakni 1. kita ambil arah ke bawah dengan koordinat 4,1 dan semua titik masuk ke dalam *OPEN*.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Bab ini membahas mengenai implementasi rencana yang diusulkan. Selain itu, bab ini menguji simulasi untuk mengetahui apakah permainan yang dimainkan sesuai dengan tujuan penelitian yang ingin dicapai. Pengujian akan dideskripsikan dalam bentuk tabel dan gambar perbandingan keadaan musuh setelah di *input* dan sebelum di *input* algoritma *A star*.

4.2 Implementasi Antar Muka

Pada sub bab ini akan dijelaskan tata letak elemen antarmuka pengguna dan *gameplay* akan dijelaskan oleh pengguna yang dapat dilihat. Tampilan ini menggunakan desain visual yang telah dibuat secara sistematis dan melihat beberapa contoh sederhana.

4.2.1 Tampilan Awal *Game*

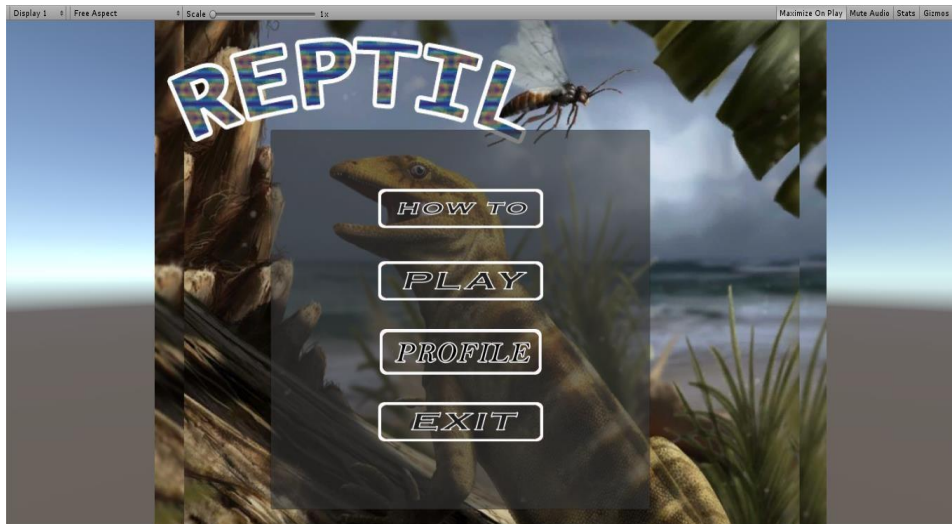
Pada tampilan awal *game* terdapat *icon game* yang dirancang oleh penulis sendiri, dan layar yang dapat ditekan untuk melanjutkan ke menu permainan. Dalam tampilan ini dapat dilihat tampilan *background reptil* yang sedang siap menerkam mangsa yang dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Awal *Game*

4.2.2 Menu Utama

Pada panel menu utama, beberapa tombol menyediakan akses ke menu berikutnya. Tampilan menu utama ini dapat dilihat pada gambar 4.2 berikut.



Gambar 4.2 Tampilan menu utama

Dalam tampilan menu utama terdapat beberapa tombol, berikut fungsi dan penjelasan pada tampilan menu utama.

- *Background* : *background* adalah tampilan yang berada di belakang papan menu berfungsi hanya sebagai penghias di dalam permainan.
- *Judul game* : judul permainan hanyalah gambar dari bacaan *reptil* yang dirancang dengan *software* photoshop.
- *Button play* : tombol yang akan membawa masuk ke *scene game play* dan mulai bermain di *level 1* dari permainan.
- *Button how to* : tombol ini berguna untuk masuk ke *panel how to* yang berisi penjelasan tentang tata cara memainkan permainan ini.
- *Button profile* : tombol ini berguna untuk masuk ke *panel profile* dan mengetahui *developer* permainan ini.
- *Button exit* : tombol ini digunakan untuk keluar dari permainan secara permanen.

4.2.3 Profil Panel

Pada tampilan profil panel ini terdapat *panel name*, *panel name* berisi nama-nama tim pengembang *game reptil* dan *button back* berfungsi untuk kembali ke menu awal *game*. Tampilan *profil game* dapat dilihat pada gambar 4.3 di bawah ini.



Gambar 4.3 Tampilan *profil panel*

4.2.4 How To

Pada *panel how to* menjelaskan tentang bagaimana petunjuk cara memainkan *game*. Berikut adalah rancangan *panel how to*.



Gambar 4.4 Tampilan *panel how to*

4.2.5 Game Play

Pada sub bab ini tampilan *game play* ini akan di jelaskan tombol-tombol yang akan pemain gunakan pada *game reptil* ini. Tombol *game play* ini berisikan antara lain.

- *Button pause* : untuk menghentikan permainan sementara dan bisa dilanjutkan kembali.
- *Control player* : *game* ini dimainkan pada *smartphone* berbasis *android* maka tombol yang akan digunakan untuk navigasi pemain adalah tombol virtual. *Button* seperti analog untuk mengarahkan pemain ke segala arah di ruang permainan.
- *Button shoot* : tombol ini digunakan Ketika pemain ingin menembak musuh yang menghalangi jalan pemain. Cara menggunakannya pemain hanya perlu menekan tombol di bagian kanan bawah.
- *Score* : adalah teks jumlah nilai yang telah berhasil diperoleh pemain dalam satu *level game*.

Dibawah ini merupakan tampilan *game play* yang disajikan pada gambar 4.5 sebagai berikut.



Gambar 4.5 Tampilan *game play*

Gambar 4.5 adalah gambar pada *level* pertama, pada *level* ke 2 akan memiliki tampilan yang sama hanya saja jumlah musuh yang berbeda kecepatan dan jumlah.



Gambar 4.6 Tampilan musuh mulai mendekati pemain

Gambar diatas merupakan tampilan pada *game* saat *enemy* mulai mendekati pemain.



Gambar 4.7 Tampilan *game play* status *enemy*

Gambar di atas adalah tampilan permainan saat musuh mulai menyerang, mengejar dan patroli untuk membunuh pemain.

4.3 Implementasi Algoritma A star

Penelitian ini akan mengimplementasikan algoritma A star pada pergerakan NPC yang bergerak tergantung dari posisi pemain, diimplementasikan dan diuji dalam sebuah tabel pengujian yang menggunakan metode *black box*.

4.4 Pengujian Algoritma A Star

Berikut adalah tabel hasil pengujian perilaku NPC untuk pemain yang telah mengimplementasikan algoritma A star. Pengujian ini terdiri dari mencari nilai dari F dan H serta penentuan algoritma dalam proses pencarian target. Pada langkah ini akan di periksa menggunakan nilai awal kordinat x, koordinat awal y, dan nilai target x dan y. Pengujian akan dilakukan dengan menggunakan metode tunggal yaitu pengujian *black box*.

Data percobaan :

X awal = 8;

Y awal = 8;

X tujuan = 10;

Y tujuan = 3; Grid = 15 x 15

Langkah	Nilai F (n)	Nilai G (n)	Nilai H (n)	X	Y
1	11.830	6	5.830	7	8
2	11.324	5	6.324	7	9
3	10.708	4	6.708	6	9
4	11.062	3	8.062	6	10
5	10.602	2	8.602	5	10
6	10.220	1	9.220	4	10
7	0	0	0	3	10

Tabel 4.1 Uji coba algoritma A star

4.5 Pengujian *Black Box*

Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian *alpha*. Metode yang digunakan dalam pengujian ini adalah pengujian *black box* yang berfokus pada persyaratan fungsional dari sistem yang dibangun pada permainan.

Aktivitas Pengujian	Realisasi	Hasil	Kesimpulan
<i>Tap to play</i>	Masuk ke menu utama	Menampilkan menu utama	Berhasil
<i>Button panel credit</i>	Masuk ke <i>panel credit</i>	Menampilkan <i>panel credit</i>	Berhasil
<i>Button panel how to</i>	Masuk ke <i>panel how to play</i>	Menampilkan <i>panel how to play</i>	Berhasil
<i>Button exit</i>	Keluar aplikasi <i>game</i>	Keluar dari <i>game</i>	Berhasil
<i>Button play</i>	Masuk ke <i>game play</i>	Masuk ke <i>scene game play</i>	Berhasil
<i>Button pause</i>	Menampilkan <i>panel pause</i>	Menampilkan <i>panel pause</i> / menghentikan permainan sementara	Berhasil
<i>Virtual analog button</i>	<i>Control player</i>	Mengontrol <i>player</i> sesuai yang di arahkan	Berhasil

<i>Button resume</i>	<i>Disable panel pause</i>	<i>Disable panel pause</i>	Berhasil
<i>Point star</i>	Menampilkan <i>point</i> bintang	Menampilkan point bintang sesuai <i>score</i>	Berhasil
<i>Button next level</i>	<i>Masuk ke level selanjutnya</i>	<i>Menampilkan level selanjutnya</i>	Berhasil

Tabel 4.2 Tabel pengujian *black box*

Keterangan :

Tabel 4.2 merupakan tabel yang berisikan rincian uji coba terhadap fungsi *A star* tahap-pertahap hingga sampai pada objek tujuan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan dari bab 4 melalui implementasi dan pengujian, penulis dapat menarik kesimpulan sebagai berikut.

1. Perancangan *game* 3D ini menggunakan Unity 3D berhasil menerapkan algoritma *A star* (A*) pada musuh berhasil. Musuh mengejar pemain melalui jalur terpendek.
2. Unity 3D adalah alat desain *game* yang bagus, dengan implementasi algoritma *A star* yang mudah dimengerti dan dipahami.
3. *Game* ini memiliki banyak musuh dan pintar, tidak hanya itu *game* ini berfokus bertahan dalam waktu yang telah ditentukan. Hal ini dimaksudkan untuk memaksimalkan potensi pemain dalam permainan.

5.2 Saran

Tentunya penelitian ini masih memiliki banyak kekurangan dan masih perlu diteliti lebih lanjut untuk pengembangannya kembali. Agar aplikasi yang dibangun dapat berfungsi sebagaimana mestinya diperlukan masukan untuk memperbaiki kekurangan yang ada baik pada *game* yang dibangun saat ini maupun saat yang akan datang.

Adapun saran yang dapat disampaikan untuk *game* ini adalah sebagai berikut.

1. Untuk pengembangan lebih lanjut, algoritma lain dapat ditambahkan untuk membuat *game* lebih kompleks.
2. Diperlukan lebih banyak pengembangan untuk membuat *game* dibuat lebih mudah digunakan dan lebih menarik dalam cerita.
3. Perlunya pengembangan dilakukan agar *game* ini dapat dibangun dengan bahasa pemrograman berbasis *IOS*.

DAFTAR PUSTAKA

- Ahmad, I., & Widodo, W. (2017). Penerapan Algoritma A Star (A*) Pada *Game* Petualangan Labirin Berbasis Android. *Jurnal Ilmu Komputer Dan Informatika*, 3(2), 57–63. <https://doi.org/10.23917/khif.v3i2.5221>
- Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian *Black Box Testing* Pada Aplikasi *Action & Strategy* Berbasis Android Dengan Teknologi Phonegap. *Jurnal String (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206–210. <https://doi.org/10.30998/string.v3i2.3048>
- Dalem, I. B. G. W. A. (2018). Penerapan Algoritma A* (*Star*) Menggunakan *Graph* Untuk Menghitung Jarak Terpendek. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, 1(1), 41–47. <https://doi.org/10.31598/jurnalresistor.v1i1.253>
- Hermanto, D., & Dermawan, S. (2018). Penerapan Algoritma A *star* Sebagai Pencari Rute Terpendek pada Robot Hexapod. *Jurnal Nasional Teknik Elektro*, 7(2), 122. <https://doi.org/10.25077/jnte.v7n2.545.2018>
- Hermawan, H., & Setiyani, H. (2019). Implementasi Algoritma A *star* Pada Permainan Komputer *Roguelike* Berbasis Unity. *Jurnal Algoritma, Logika Dan Komputasi*, 2(1), 111–120.
- Ichwan, M., & Hakiky, F. (2011). Pengukuran Kinerja *Goodreads Application Programming Interface (API)* Pada Aplikasi *Mobile* Android (Studi Kasus Untuk Pencarian Data Buku). *Jurnal Informatika*, 2(2), 13–21.
- Kusniyati, H., & Sitanggang, N. S. P. (2016). Aplikasi Edukasi Budaya Toba Samosir Berbasis Android. *Jurnal Teknik Informatika*, 9(1), 9–18. <https://doi.org/10.15408/jti.v9i1.5573>
- Musfiroh, L., Jazuli, A., & Latubessy, A. (2014). Penerapan Algoritma *Collision Detection* Dan *Boids* Pada *Game Dokkaebi Shooter*. *Prosiding SNATIF*, 217–224. <http://jurnal.umk.ac.id/index.php/SNA/article/view/148>
- Muttaqin, Atthariq, & Mursyidah. (2017). Penerapan Algoritma *Linear Congruent Method* Terhadap *Spawning Enemy*. *Jurnal Infomedia*, 2(1), 13–18. <https://doi.org/10.30811/.v2i1.478>
- Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian *Black Box* Pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik *Equivalence Partitions*. *Jurnal Informatika Universitas Pamulang*, 4(3), 125–130. <https://doi.org/10.32493/jtsi.v3i3.5343>
- Nugroho, A., & Pramono, B. A. (2017). Aplikasi *Mobile Augmented Reality* Berbasis *Vuforia* Dan *Unity* Pada Pengenalan Objek 3D Dengan Studi Kasus Gedung M Universitas Semarang. *Jurnal Transformatika*, 14(2), 86–91. <https://doi.org/10.26623/transformatika.v14i2.442>
- Nuraini, R. (2015). Desain Algoritma Operasi Perkalian Matriks Menggunakan Metode *Flowchart*. *Jurnal Teknik Komputer Amik Bsi*, 1(1), 144–151.

- Nuryoso, Y. H., Pradjoko, & Lelah. (2020). Penerapan Algoritma A* pada Pencarian Rute Terpendek pada Rute Angkot Di Kota Sukabumi. *Jurnal Sarjana Teknik Informatika*, Vol. 8, No. 1, Februari 2020, Pp. 21-35 e-ISSN, 8(1), 21–35.
- Oktafiani, N. K. (2020). Implementasi Algoritma A* (A star) Untuk Menentukan Rute Terpendek Pada NPC Game *Healthy Food*. Universitas Islam Negeri Maulana Malik Ibrahim.
- Oktanugraha, R. F., & Nudin, S. R. (2020). Implementasi Algoritma A * (A star) Dalam Penentuan Rute Terpendek Yang Dapat Dilalui *Non Player Character* Pada Game *Good Thief*. *Journal of Informatics and Computer Science*, 02(01), 74–85.
- Parassa, Y., Pesik, M. U., & Mantala, R. (2017). Implementasi Dan Pengujian Model *Responsive Website* Objek Wisata Provinsi Sulawesi Utara. *Prosiding Seminar Nasional ASBIS 2017, December 2017*.
- Permana, B. R. (2017). Pengembangan Dan Analisis Kelayakan *Game* Pengenalan Periperal Komputer Berbasis Desktop Sebagai Sarana Pembelajaran Perakitan Komputer Dasar Siswa Kelas X SMK N 2 Yogyakarta. Universitas Negeri Yogyakarta.
- Rakhmat Kurniawan. R., ST, M. K., & Yusuf Ramadhan Nasution, M. K. (2016). Penerapan Algoritma A * (A star) Sebagai Solusi Pencarian Rute Terpendek Pada *Maze*. *Conference Paper, October*, 1–6.
- Rizal, M. M., & Aryanto, H. (2017). Perancangan *Game* “SI ALI (Aku Peduli) Melawan Korupsi” Sebagai Media Interaktif Bernilai Budaya. *Jurnal Pendidikan Seni Rupa*, 05(01), 79–86.
- Rosiska, E. (2016). Perancangan Dan Implementasi Wisata Batam Berbasis Android. *Teknologi Informasi*, 1, 12–20.
- Safitra, W., Faisol, A., & Wibowo, S. A. (2020). Penerapan Metode *Finite State Machine* Pada *Non Player Character (Npc) Game Action Strategy “Ouroboros .”* *JATI (Jurnal Mahasiswa Teknik Informatika)*, 4(2), 292–297.
- Simanjuntak, K. A. D. (2019). Perbandingan Algoritma A* Dan Algoritma *Best First Search Modification* Untuk Mencari Jalur Terpendek Pasar Swalayan Di Kota Medan. Universitas Sumatera Utara.
- Surasmi, W. A. (2016). Pemanfaatan Multimedia untuk Mendukung Kualitas Pembelajaran. *Temu Ilmiah Nasional Guru (TING) VIII, November*.
<http://repository.ut.ac.id/6555/1/TING2016ST2-32.pdf>

Sutanti, A., M. Komaruddin, M., Mustika, & Damayanti, P. (2020). Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 9(1), 1-8. <https://doi.org/10.34010/komputa.v9i1.3718>

LAMPIRAN 1
LISTING PROGRAM

SOURCE CODE ENEMY / NPC (CROCODILE)

```
using UnityEngine;
using UnityEngine.AI;
using System.Collections;
using System;

public class Enemy : MonoBehaviour
{
    [HideInInspector]
    public AgentControl[] agents;
    [HideInInspector]
    public Animator animator;
    [HideInInspector]
    string state;

    NavMeshAgent agent;
    AgentControl target;
    Transform trans;

    public float target_switching_delay = 1.66f;
    public float atk_range = 0.66f;
    public float atk_delay = 0.66f;
    public float atk_dmg = 6;
    public float dmg_delay = 0.33f;
    public float hp = 100f;
    bool is_dead;

    void Start ()
```

```

{
    agent = GetComponent<NavMeshAgent> ();
    animator = GetComponent<Animator> ();
    trans = transform;
    StartCoroutine (FindTarget ());
    state = "Move";
    StartCoroutine (Move ());
}
void Update ()
{
    animator.SetFloat ("hp", hp);
    animator.SetFloat ("speed", agent.velocity.magnitude);
    if (hp <= 0f && !is_dead) {
        is_dead = true;
        int nilai = PlayerPrefs.GetInt ("nilai") + 10;
        PlayerPrefs.SetInt ("nilai", nilai);
        StopAllCoroutines ();
        StartCoroutine (Die ());
        Debug.Log ("mati");
    }
}

```

IEnumerator Die ()

```

{
    //Debug.Log ("Dead : " + gameObject.name);
    agent.Stop ();
    Collider col = GetComponent<Collider> ();
    col.enabled = false;
    yield return new WaitForSecondsRealtime (3f);
    Destroy (this.gameObject);
}

```

```

//(Algoritma AStar)
IEnumerator FindTarget ()
{
    while (true) {
        agents = GameObject.FindObjectsOfType<AgentControl> ();
        int nearest = 0;
        float min = 100;
        for (int i = 0; i < agents.Length; i++) {
            float distance = Vector3.Distance (trans.position, agents [i].trans.position);
            if (distance < min) {
                nearest = i;
                min = distance;
            }
        }
        target = agents [nearest];
        //agent.destination = target.trans.position;
        yield return new WaitForSeconds (target_switching_delay);
    }
}

```

//script jalan

```

IEnumerator Move ()
{
    while (state == "Move") {
        agent.destination = target.trans.position;
        float distance_to_target = Vector3.Distance (trans.position,
            target.trans.position);
        if (distance_to_target < atk_range) {
            state = "Atk";
        }
    }
}

```

```
        yield return 0;
    }
    StartCoroutine (state);
}

// Script Menyerang
IEnumerator Atk ()
{
    //Debug.Log ("Atk");
    animator.SetTrigger ("atk");
    yield return new WaitForSeconds (dmg_delay);
    target.Hit (atk_dmg);
    yield return new WaitForSeconds (atk_delay);
    state = "Move";
    StartCoroutine (state);
}
}
```

SOURCE CODE PLAYER (PEMAIN)

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using UnityEngine.AI;
using UnityEngine.Events;
using System;

namespace Suriyun.MobileTPS
{
    public class AgentControl : MonoBehaviour {
        public GameCamera game_camera;
        public Animator animator;
        public BehaviourControls behaviour;
        [Header("Camera")]
        //public Camera mainCamera;
        [Header("Weapon Components")]
        public ParticleSystem muzzleflashParticles;
        public Light muzzleflashLight;
        //private Animator anim;

        [HideInInspector]
        public Transform trans;
        public float hp = 15;
        public bool is_alive = true;
        public GameObject fx_on_hit;
        public UnityEvent ActorDiedEvent;

    void Awake()
    {
        trans = transform;
    }
}
```



```

behaviour.Init(this);
}

public void Hit(float damage)
{
hp -= damage;
Instantiate(fx_on_hit, trans.position, fx_on_hit.transform.rotation);
}

private void Start ()
{
//Assign animator component
//Disable muzzleflash light at start
muzzleflashLight.enabled = false;
}

IEnumerator MuzzleflashLight ()
{
muzzleflashLight.enabled = true;
yield return new WaitForSeconds (0.02f);
muzzleflashLight.enabled = false;
}
}

[Serializable]
public class BehaviourControls
{
[HideInInspector]
public Button btn_fire;
[HideInInspector]
public Button btn_hide;
[HideInInspector]
public Button btn_right;
[HideInInspector]

```

```
public Button btn_left;
AgentControl parent;
[Header("Camera FOV Settings")]
public float zoomedFOV;
public float defaultFOV;
[Tooltip("How fast the camera zooms in")]
public float fov Speed;

[Header("Weapon Settings")]
public bool semi;
public bool auto;

//Used for fire rate
private float lastFired;
//How fast the weapon fires, higher value means faster rate of fire
[Tooltip("How fast the weapon fires, higher value means faster rate of fire.")]
public float fireRate;

[Header("Prefabs")]
public Transform casingPrefab;
public Transform bulletPrefab;
public float bulletForce;
public Transform grenadePrefab;
public float grenadeSpawnDelay;

[Header("Spawnpoints")]
public Transform casingSpawnpoint;
public Transform bulletSpawnpoint;
public Transform grenadeSpawnpoint;

[Header("Audio Clips")]
```

```

public AudioClip shootSound;

[Header("Audio Sources")]
public AudioSource shootAudioSource;
[HideInInspector]
public Transform destination;
[HideInInspector]
public int destination_index;
[HideInInspector]
public int current_index;
[HideInInspector]
public bool firing;
[HideInInspector]
public bool hiding;
[HideInInspector]
public float no_firing_delay;
[HideInInspector]
public NavMeshAgent agent;
// [HideInInspector]
public Animator animator;
public GameObject Pintu;
public Transform gun_tip;
public GameObject bullet_prefab;
Coroutine c_firing;

public void Init(AgentControl parent)
{
    this.parent = parent;
    destination = GameObject.Find("+destination").transform;
    animator = parent.GetComponent<Animator>();
    agent = parent.GetComponent<NavMeshAgent>();
}

```

```
    parent.StartCoroutine(PseudoUpdate());
}
IEnumerator Die()
{
    yield return new WaitForSecondsRealtime(2f);
}
IEnumerator PseudoUpdate()
{
    while (true)
    {
        Update();
        if (parent.hp <= 0)
        {
            parent.is_alive = false;
            Game.instance.EventGameOver.Invoke();
        }
        yield return 0;
    }
}
```

```
public void StartFiring()
{
    parent.game_camera.zoomed = true;
    //parent.game_camera.fieldOfView = Mathf.Lerp
    (parent.game_camera.fieldOfView,
    //zoomedFOV, fovSpeed * Time.deltaTime);

    if (parent.is_alive)
    {
        hiding = false;
        firing = true;
    }
}
```

```
    if (c_firing != null)
    {
        parent.StopCoroutine(c_firing);
    }
    c_firing = parent.StartCoroutine(FiringMechanism());
}
}
```

```
public void StopFiring()
{
    firing = false;
    parent.game_camera.zoomed = false;
}
```

```
IEnumerator FiringMechanism()
{
    float gun_delay = 0.1f;
    yield return new WaitForSeconds(0.16f);
    while (firing)
    {
        UnityEngine.Object.Instantiate(bullet_prefab, gun_tip.position,
gun_tip.rotation);
        yield return new WaitForSeconds(gun_delay);
    }
}
```

```
public void Hide()
{
    if (parent.is_alive)
    {
        hiding = true;
    }
}
```

```
firing = false;
no_firing_delay = 0.16f;
agent.Stop();
}
}

public void GoLeft()
{
    if (parent.is_alive)
    {
        hiding = false;
        agent.Resume();
        if (current_index < destination_index)
        {
            current_index = destination_index;
        }
        destination_index = current_index - 1;
    }
}

public void GoRight()
{
    if (parent.is_alive)
    {
        hiding = false;
        agent.Resume();
        if (current_index > destination_index)
        {
            current_index = destination_index;
        }
        destination_index = current_index + 1;
    }
}
```

```
    }  
}  
  
public void start()  
{  
    kunci = false;  
    //animator = GameObject.GetComponent<Animator> ();  
}
```

```
public void Update()  
{  
    if (kunci = true)  
    {  
        Pintu.SetActive (true);  
    }  
    else  
    {  
        Pintu.SetActive (false);  
    }  
}
```

```
UpdateCurrentPosition();  
#region :: Input Handler ::  
    if (btn_left.pressed || Input.GetKey(KeyCode.A))  
    {  
        GoLeft();  
    }  
    if (btn_right.pressed || Input.GetKey(KeyCode.D))  
    {  
        GoRight();  
    }  
    if (btn_hide.pressed || Input.GetKey(KeyCode.S))
```

```

{
Hide();
}
if (Input.GetKeyDown(KeyCode.Space))
{
//parent.game_camera.fieldOfView = Mathf.Lerp
(parent.game_camera.fieldOfView,
//zoomedFOV, fovSpeed * Time.deltaTime);
StartFiring();
}
if (Input.GetKeyUp(KeyCode.Space))
{
//parent.game_camera.fieldOfView = Mathf.Lerp
(parent.game_camera.fieldOfView,
//defaultFOV, fovSpeed * Time.deltaTime);
StopFiring();
}
#endregion
animator.SetBool("hiding", hiding);
animator.SetBool("firing", firing);
animator.SetFloat("speed", agent.velocity.magnitude);
//animator.SetFloat("vertical", Input.GetAxis("Vertical"));
//animator.SetFloat("horizontal", Input.GetAxis("Horizontal"));
destination_index = Mathf.Clamp(destination_index, 0,
Game.instance.map_data.red_move_pos.Count - 1);
destination.positionGame.instance.map_data.red_move_pos[destination_index]
.position;
agent.destination = destination.position;
animator.SetFloat("hp", parent.hp);
if (!parent.is_alive)
{

```



```

        parent.StopAllCoroutines();
    }
}
public bool kunci;
public void OnCollisionEnter (Collider other){
if (other.gameObject.name == "Kunci") {
    //Application.LoadLevel ("Intro");
    kunci =true;
    Debug.Log ("oke");
}
    if (other.gameObject.name == "Pintu") {
        //Application.LoadLevel ("Intro");
        //kunci =true;
        Debug.Log ("keok");
    }
}
private void UpdateCurrentPosition()
{
for (int i = 0; i < Game.instance.map_data.red_move_pos.Count; i++)
{
    if (Vector3.Distance(parent.trans.position,
        Game.instance.map_data.red_move_pos[i].position) < 1f)
    {
        current_index = i;
    }
}
}
}
}
}

```

SOURCE CODE UNTUK SEMUA TOMBOL BOTTON

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class buttonman : MonoBehaviour {
public GameObject FlashScreen;
public GameObject PanelMainMenu;
public GameObject PanelCredits;
public GameObject PanelHowTo;

// Use this for initialization
void Start () {
}

// Update is called once per frame
void Update () {

}

public void ButtonPlay(){
Application.LoadLevel ("Intro");
}

public void ButtonExit(){
Application.Quit ();
}

public void ButtonCredits(){
PanelCredits.SetActive (true);
PanelMainMenu.SetActive (false);
}

public void ButtonHowto(){
PanelHowTo.SetActive (true);
PanelMainMenu.SetActive (false);
```

```
}  
public void ButtonBack(){  
    PanelCredits.SetActive (false);  
    PanelMainMenu.SetActive (true);  
    PanelHowTo.SetActive (false);  
}  
public void ButtonTapto(){  
    FlashScreen.SetActive (false);  
    PanelMainMenu.SetActive (true);  
}  
}
```

LAMPIRAN 2
DAFTAR RIWAYAT HIDUP



I. Data Pribadi

Nama : Zulfikri Bahri
Nim : 71154037
Tempat, Tanggal Lahir : Tanjung Balai, 05 Mei 1997
Jenis Kelamin : Laki – Laki
Alamat : Jalan Veteran No. 18 LK II
Agama : Islam
Status Nikah : Belum Menikah
No. HP : 085362236424

II. Pendidikan Normal

2004 – 2005 : TK Kemala Bhayangkari Tanjung Balai
2006 – 2011 : SD Negeri 137697 Tanjung Balai Sumatera Utara
2011 – 2013 : SMP Negeri 1 Tanjung Balai Sumatera Utara
2013 – 2015 : SMA Negeri 1 Tanjung Balai Sumatera Utara
2015 – 2020 : UINSU Medan Sumatera Utara

LAMPIRAN 3

KARTU BIMBINGAN SKRIPSI

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik /

Nama : Zulfikri Bahri	Pembimbing I : Dr. Mhd. Furdan, S.Si., M. CO
NIM : 71154037	Pembimbing II : Muhammad Ikhsan, S.T., M.
Prog. Studi : ILMU KOMPUTER	SK Pembimbing :
Judul Skripsi : Penerapan Algoritma A Star untuk Pencarian Jalur terdekat NPC Non Player Terhadap NPC (Non Player Character) pada Game AdvenTuras Berbasis Android	


P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I		Bimbingan Proposal Skripsi BAB I			Bimbingan Proposal Skripsi BAB I	
II		Bimbingan Proposal Skripsi BAB II			Bimbingan Proposal Skripsi BAB II	
III		Bimbingan Proposal Skripsi BAB III			Bimbingan Proposal Skripsi BAB III	
IV		Bimbingan Revisian Proposal BAB III & ACC			Bimbingan Revisian Proposal BAB III & ACC	
V		Bimbingan Skripsi BAB I			Bimbingan Skripsi BAB I	

VI	Bimbingan Skripsi BAB II	Furqan	Bimbingan Skripsi BAB II	P
VII	Bimbingan Skripsi BAB III	Furqan	Bimbingan Skripsi BAB III	P
VIII	Bimbingan Skripsi BAB IV	Furqan	Bimbingan Skripsi BAB IV	P
IX	Bimbingan Skripsi BAB IV & Program	Furqan	Bimbingan Skripsi BAB IV & Program	P
X	Bimbingan Skripsi BAB V & ACC Skripsi	Furqan	Bimbingan Skripsi BAB V & ACC Skripsi	P

Medan, 2020

An. Dekan

Ketua Jurusan/Program Studi



NIP.

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing