

**PERMAINAN PETAK UMPET DENGAN PEMANFAATAN ARTIFICIAL
INTELLIGENCE PATHFINDING A* MENGGUNAKAN
VIRTUAL REALITY**

SKRIPSI

FREDERIKO HUTAHAEAN

171402062



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

**PERMAINAN PETAK UMPET DENGAN PEMANFAATAN ARTIFICIAL
INTELLIGENCE PATHFINDING A* MENGGUNAKAN
VIRTUAL REALITY**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

FREDERIKO HUTAHAEAN

171402062



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2024

PERSETUJUAN

Judul : Permainan Petak Umpet Dengan Pemanfaatan Artificial
Intelligence Pathfinding A* Menggunakan Virtual Reality
Kategori : Skripsi
Nama Mahasiswa : Frederiko Hutahaean
Nomor Induk Mahasiswa : 171402062
Program Studi : Sarjana (S-1) Teknologi Informasi
Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Medan, 4 Juli 2024

Komisi Pembimbing :

Pembimbing 2,



Ainul Hizriadi S.Kom., M.Sc
NIP. 198510272017061001

Pembimbing 1,



Fahrurrozi Lubis B.IT., M.Sc.IT
NIP. 198610122018052001

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,


Dedy Arisandi S.T., M.Kom.
NIP. 197908312009121002

PERNYATAAN**PERMAINAN PETAK UMPET DENGAN PEMANFAATAN ARTIFICIAL
INTELLIGENCE PATHFINDING A* MENGGUNAKAN
VIRTUAL REALITY****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing – masing telah disebutkan sumbernya

Medan, 20 April 2024

A handwritten signature in black ink, consisting of stylized, overlapping loops and strokes, likely representing the initials 'FH'.

Frederiko Hutahaeen

171402062

UCAPAN TERIMA KASIH

Penulis memanjatkan puji dan syukur kehadirat Tuhan Yang Maha Esa, karena berkat kesehatan dan rahmat yang dilimpahkan, penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Permainan Petak Umpet Dengan Pemanfaatan Artificial Intelligence Pathfinding A* Menggunakan Virtual Reality” sebagai salah satu syarat untuk memperoleh gelar Sarjana di Program Studi S1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara. Dalam proses penyusunan skripsi ini tidak terlepas dari berbagai kendala yang telah dihadapi. Penyusunan skripsi ini tidak lepas dari bantuan dan dukungan dari beberapa pihak baik secara langsung maupun tidak langsung. Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Keluarga Penulis, Ayah Dharma Hutahaean, Ibu Sanna Mujurani Tampubolon, Kakak Hans Ivander P.P. Hutahaean serta Adik Mutiara Vania Hutahaean yang selalu memberikan dukungan, doa, semangat senantiasa kepada penulis,
2. Bapak Dr. Muryanto Amin, S.Sos., M.Si selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lydia, M.Sc selaku Dekan Fasilkom-TI USU.
4. Bapak Dedy Arisandi S.T., M.Kom selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Fahrurrozi Lubis B.IT, M.Sc.IT selaku Dosen Pembimbing I dan juga Bapak Ainul Hizriadi S.kom, M.SC selaku Dosen Pembimbing II yang telah meluangkan waktunya untuk memberikan bimbingan, petunjuk, dan semangat dalam proses penyelesaian skripsi ini.
6. Bapak/Ibu dosen penguji.
7. Seluruh Dosen Program Studi S1 Teknologi Informasi yang telah memberikan ilmu selama perkuliahan.

8. Para staff dan pegawai Fasilkom-TI Universitas Sumatera Utara yang telah membantu dalam segala urusan administrasi di masa perkuliahan dan penyelesaian skripsi.
9. Kakak/abang senior yang memberikan saran, kritik, maupun masukan dan bantuan kepada penulis selama masa perkuliahan maupun selama masa pengerjaan skripsi.
10. Teman-teman angkatan 2017 yang telah memberikan dukungan dan semangat selama masa perkuliahan dan juga selama masa pengerjaan skripsi yang tidak dapat penulis sebutkan satu - persatu.
11. Teman-teman di Kom B yang tidak dapat penulis sebutkan satu-persatu.
12. Para sahabat di Kom B dan Anggota GULLY yaitu Frans, Riyo, Maulana, Syarfan, Irfan, Josua, Jonathan, Moris, Aldo, Yonadab, Dicky, Alfi dan Yusman yang selalu memberikan dukungan, doa, dan semangat dimasa perkuliahan sampai selesai penyusunan skripsi ini.
13. Semua pihak yang terlibat langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu-persatu yang telah membantu dalam penyelesaian skripsi ini

ABSTRAK

Permainan petak umpet merupakan salah satu permainan yang paling populer pada kalangan anak-anak, dan sering dimainkan dalam acara rekreasi. Dalam upaya untuk menghadirkan pengalaman agar lebih menarik daya tarik, penelitian ini bertujuan untuk merancang permainan petak umpet yang menggunakan teknologi *Virtual Reality (VR)* dengan memanfaatkan kecerdasan buatan yaitu *pathfinding A**. Pada perancangan tersebut, VR digunakan sebagai menciptakan lingkungan 3D yang realistis sehingga pemain dapat merasakan sensasi seolah-olah pemain benar-benar berada di dalam ruang petak umpet, selain itu, pemanfaatan penggunaan kecerdasan buatan dengan algoritma *pathfinding A** dapat memberikan kemampuan pada karakter animasi untuk menyusun strategi yang terbaik dalam mencari atau menemukan pemain yang sedang bersembunyi. Metode pengembangan pada permainan ini melibatkan tahapan desain lingkungan *VR*, implementasi algoritma *pathfinding A**, integrasi antarmuka pada pengguna *VR*, serta uji coba terlebih dahulu kinerja dan responsivitas permainan yang telah digunakan pada target pengguna. Hasil pengujian yang telah dilakukan pada pengguna mendapatkan akumulasi respon positif sebanyak 85,49% terhadap semua aspek-aspek pertanyaan mengenai aplikasi yang telah diujikan dan dibuat.

Kata Kunci : Permainan, petak umpet, *Virtual Reality*, *Artificial Intelligence*, *Pathfinding A**

***HIDE AND SEEK GAME UTILIZING ARTIFICIAL INTELLIGENCE A*
PATHFINDING THROUGH VIRTUAL REALITY***

ABSTRACT

Hide and seek is one of the most popular games among children, and is often played in recreational events. In an effort to bring the experience to be more attractive, this research aims to design a hide-and-seek game that uses Virtual Reality (VR) technology by utilizing artificial intelligence, namely A* pathfinding. In this design, VR is used to create a realistic 3D environment so that players can feel the sensation as if the player is really in the hide and seek room, besides that, the utilization of the use of artificial intelligence with the A* pathfinding algorithm can provide the ability for animated characters to develop the best strategy in finding or finding players who are hiding. The development method in this game involves the stages of VR environment design, implementation of the A* pathfinding algorithm, interface integration for VR users, and testing the performance and responsiveness of the game that has been developed on the target user. The results of testing that have been carried out on users get an accumulated positive response of 85,49% to all aspects of questions about the application that has been tested and made.

Keywords: Game , hide and seek, Virtual Reality, Artificial Intelligence, A Pathfinding*

DAFTAR ISI

ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN.....	13
1.1 Latar Belakang.....	13
1.2 Rumusan Masalah.....	14
1.3 Batasan Masalah.....	15
1.4 Tujuan Penelitian.....	15
1.5 Manfaat Penelitian.....	15
1.6 Metodologi Penelitian.....	16
BAB 2 LANDASAN TEORI.....	18
2.1 <i>Game</i>	18
2.2 Permainan Tradisional.....	18
2.3 <i>Virtual Reality</i> (VR).....	19
2.4 Unity 3D.....	20
2.5 Blender 3D.....	20
2.6 <i>Artificial Intelligence</i>	21
2.7 <i>Pathfinding A*</i>	21
2.8 Unified Modelling Language (UML).....	24
1. <i>Use Case Diagram</i>	24
2. <i>Activity Diagram</i>	25
3. <i>Sequence Diagram</i>	26
2.9 Android.....	26
2.10 Objek 3D.....	26

2.11	Penelitian Terdahulu	27
BAB 3 ANALISIS DAN PERANCANGAN		29
3.1	Analisis Sistem	29
1.	Analisis Masalah	29
2.	Analisis Kebutuhan Sistem	29
3.	Analisis Kebutuhan <i>Game</i>	30
3.2	Arsitektur Umum	31
1.	Pengumpulan Data	32
2.	Modelling 3D (Blender)	33
3.	Unity 3D (<i>Game & AI Engine</i>)	34
4.	Output	36
3.3	Pemodelan Sistem	36
1.	<i>Use Case Diagram</i>	37
2.	<i>Activity Diagram</i>	38
3.	<i>Sequence Diagram</i>	39
3.4	Perancangan Antar Muka	40
1.	Tampilan Main Menu	40
2.	Tampilan Informasi Pengembang	41
3.	Tampilan <i>Gameplay</i> permainan	41
BAB 4 IMPLEMENTASI SISTEM		43
4.1	Implementasi Permainan	47
4.2	Tampilan Permainan Pada <i>Game Engine</i> (Unity)	47
1.	Tampilan Menu	47
2.	Tampilan Info Pengembang	48
3.	Tampilan Permainan	48
4.	Tampilan Menjalankan Misi	49
4.3	Tahapan Permainan	53

4.4	Pengujian Sistem	53
BAB 5 KESIMPULAN DAN SARAN		59
5.1	Kesimpulan.....	59
5.2	Saran.....	59
DAFTAR PUSTAKA		61

DAFTAR GAMBAR

	Hal
Gambar 2. 1 Pathfinding A*	24
Gambar 3. 1 Arsitektur Umum	31
Gambar 3. 2 Ilustrasi <i>Pathfinding A*</i>	35
Gambar 3. 3 Ilustrasi Arah <i>Pathfinding A*</i>	36
Gambar 3. 4 <i>Use Case Diagram</i>	37
Gambar 3. 5 Activity Diagram	39
Gambar 3. 6 <i>Sequence Diagram</i>	40
Gambar 3. 7 Desain Tampilan Main Menu	41
Gambar 3. 8 Desain Info Pengembang	41
Gambar 3. 9 Desain <i>Gameplay</i> Permainan	42
Gambar 3. 10 AI Mengejar Pemain	43
Gambar 3. 11 AI Menangkap Pemain	44
Gambar 3. 12 Pemain Bersembunyi	45
Gambar 3. 13 <i>Navmesh Agent AI</i>	46
Gambar 4.1 Tampilan Menu	47
Gambar 4.2 Tampilan Developer	48
Gambar 4.3 Tampilan Permainan	49
Gambar 4.4 Tampilan Menjalankan Misi	49
Gambar 4.5 Tampilan Musuh Mengejar Pemain	50
Gambar 4.6 Tampilan Musuh Menangkap Pemain	50
Gambar 4.7 Tampilan Musuh Masuk Ruangan Rahasia	51
Gambar 4.8 Tampilan Pemain Berhasil Menyelesaikan Misi	51
Gambar 4.9 Tampilan Pemain Mengejar Musuh	52

DAFTAR TABEL

	Hal
Tabel 2.1 <i>Use Case Diagram</i>	24
Tabel 2.2 <i>Activity Diagram</i>	25
Tabel 2.3 <i>Sequence Diagram</i>	26
Tabel 2.4 Penelitian Terdahulu	27
Tabel 3.1 Proses Aktor <i>Use Case</i>	37
Tabel 3.2 Proses Memulai Permainan	37
Tabel 3.3 Proses Info Pengembang	38
Tabel 3.4 Proses Keluar Aplikasi	38
Tabel 4.1 Tabel <i>Blackbox</i> Test	53
Tabel 4.2 Tabel Main Menu	54
Tabel 4.3 Tabel Menjalankan Misi	54
Tabel 4.4 Tabel Kuis	54
Tabel 4.5 Kuis Pertama	55
Tabel 4.6 Kuis Kedua	56
Tabel 4.7 Kuis Ketiga	56
Tabel 4.8 Kuis Keempat	57
Tabel 4.9 Kuis Kelima	57

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Game atau permainan dapat diartikan sebagai aktifitas baik dalam bentuk tindakan nyata ataupun di dalam suatu sistem/aplikasi yang dapat membawa kesenangan/hiburan, serta untuk melatih konsentrasi dan kemampuan berfikir yang dapat mengasah otak bagi penggunaanya (Putri dan Adisusilo, 2016). Permainan yang telah ada sejak lama atau dimainkan sejak zaman dahulu disebut dengan permainan tradisional. Permainan tradisional merupakan bentuk kegiatan yang berasal dari kebiasaan suatu masyarakat tertentu.

Salah satu permainan tradisional yaitu petak umpet. Permainan petak umpet sangat sering dimainkan pada masa kanak-kanak. Dalam permainan petak umpet, pemain terdiri dari pencari dan yang bersembunyi. Aturan bermain petak umpet cukup sederhana. Pemain yang ditunjuk sebagai pencari, umumnya karena kalah dalam undian, telah menutup mata dan menghitung sesuai kesepakatan hingga semua pemain lain telah bersembunyi. Selanjutnya, pencari telah mencari pemain yang bersembunyi. Pemain yang bersembunyi juga memiliki misi dalam permainan, yaitu sebelum diketahui pencari harus menyentuh suatu benda yang ditetapkan sebagai pusat permainan. Pemain yang gagal menyentuh pusat permainan sebelum ditemukan atau kalah, telah mengganti posisi pencari. Permainan petak umpet memiliki berbagai manfaat, diantaranya melatih kecepatan dan kecerdikan, juga melatih ketelitian dan kejelian dalam memilih tempat bersembunyi. Untuk mendukung konsep permainan ini dengan pengimplementasian AI maka akan digunakan algoritma Pathfinding A*. Pathfinding A* menyelesaikan masalah yang menggunakan graph untuk perluasan ruang statusnya. Dengan kata lain digunakan untuk menyelesaikan permasalahan yang bisa direpresentasikan dengan graph. Algoritma A* adalah sebuah algoritma yang telah dikembangkan. Dengan menerapkan fungsi heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang

sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan. Didalam algoritma A* menggunakan dua senarai yaitu *OPEN* dan *CLOSED*. Pada algoritma A* memiliki tiga kondisi dimana NPC berada di keadaan *OPEN*, berada di keadaan *CLOSED*, dan tidak berada di keduanya. Jika NPC dalam keadaan *OPEN* maka akan dilakukan pengecekan apakah perlu perubahan *parent* atau tidak tergantung pada nilai melalui *parent* lama maupun *parent* baru (Izza, 2020).

Adanya perkembangan teknologi mempengaruhi berbagai aspek di kehidupan manusia, tak terkecuali dalam permainan. Perkembangan teknologi saat ini memengaruhi permainan menjadi digital. Namun, permainan tradisional yang lebih banyak menggunakan aktivitas fisik jika dibandingkan dengan permainan digital mengakibatkan semakin berkurangnya peminat permainan tradisional. Hal ini karena permainan tradisional dianggap kurang praktikal. Selain itu, untuk permainan petak umpet sendiri, membutuhkan tempat yang luas dan lingkungan yang sesuai sebagai tempat untuk bersembunyi. Ditambah lagi, perkembangan *Game* saat ini yang semakin mengarah padaperangkat mobile seperti tablet dan *smartphone* baik itu android maupun IOS. Sehingga, perlu adanya upaya untuk melestarikan permainan tradisional yang salah satunya petak umpet (Alam, 2022).

Dengan memasukkan konsep permainan tradisional ke dalam permainan digital, diharapkan agar pengalaman bermain yang diperoleh menjadi lebih praktikal dan juga tetap menyenangkan (Al- Farizi, 2019). Namun, *Game* atau permainan yang mudah diprediksi membuat *Game* kurang menarik. Untuk mengatasinya dapat dilakukan dengan pemanfaatan *Artificial Intelligence*. Penerapan AI dilakukan agar *Game* yang dimainkan terasa lebih realistis dan tidak membosankan karena dinilai kurang menantang. Salah satu metode yang digunakan dalam *Game* untuk kecerdasan buatanannya adalah *Pathfinding*. Metode *Pathfinding* biasanya digunakan sebagai inti pergerakan AI (Rahadiansyah et al., 2016).

1.2. Rumusan Masalah

Permainan petak umpet sebagai salah satu permainan tradisional, dianggap kurang praktikal dan menarik karena membutuhkan aktivitas fisik dan pemain yang banyak. Yang mengakibatkan pula permainan tersebut mulai tergeser dengan berkembangnya teknologi. Untuk mempertahankan permainan tradisional kepada anak-anak Indonesia generasi penerus maka dibutuhkan inovasi dalam visualisasi melibatkan teknologi dan

pemanfaatan AI, maka perlu dilakukan implementasi permainan tradisional petak umpet menjadi permainan digital berbasis Android.

1.3. Batasan Masalah

Batasan yang diterapkan untuk pengembangan permainan digital ini, diantaranya:

1. Metode yang digunakan di dalam penelitian ini adalah pemanfaatan *Artificial Intelligence* dengan menggunakan algoritma *pathfinding a**.
2. Aplikasi dibangun menggunakan teknologi *Virtual Reality*.
3. Permainan telah dibuat dengan model 3D.
4. Pemain dalam permainan hanya satu pengguna melawan NPC (*Non-Player Character*)
5. Pemain dapat memilih untuk menjadi pencari atau yang bersembunyi

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam pengembangan permainan petak umpet digital ini adalah untuk menerapkan algoritma *Pathfinding* pada permainan digital petak umpet android.

1.5. Manfaat Penelitian

Berikut merupakan manfaat yang telah diperoleh dari pengembangan permainan digital ini.

1. Meningkatkan minat agar lebih mengenal permainan tradisional, terkhususnya permainan petak umpet sebagai permainan anak-anak.
2. Memperoleh pengalaman baru dalam bermain permainan anak-anak, yaitu petak umpet.
3. Sebagai dasar untuk mengembangkan permainan digital dengan penerapan algoritma *pathfinding A** khususnya pada program studi Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

1.6. Metodologi Penelitian

Software Development Life Cycle adalah sebuah metode penelitian yang digunakan agar pengerjaan dari setai tahapan demi tahapan dari pengelolaan aplikasi dan data yang digunakan terstruktur dan melalui step atau langkah-langkah yang tepat dimulai dari analilis kebutuhan dari aplikasi, desain yang dibuat untuk memenuhi tampilan dari aplikasi, evaluasi dari desain yang telah dibuat untuk mematangkan hasil akhirnya, pemrograman sistem yang dibangun, setelah itu melakukan pengujian aplikasi, pengevaluasian aplikasi secara keseluruhan dan akhirnya aplikasi dapat dipergunakan : Agar lebih jelas kita bisa lihat dari penjelasan tiap-tiap bagian sepeti yang dijelaskan dengan singkat di bawah :

1. Analisa kebutuhan

Menganalisa tiap-tiap kebutuhan untuk kelengkapan dari aplikasi yang telah dibuat baik dari segi isi maupun tampilan yang telah dibawakan.

2. Membuat Desain

Merancang desain yang sesuai dengan tema yang telah dibawa agar menunjang kecocokan pada aplikasi.

3. Evaluasi Desain

Melakukan evaluasi untuk melihat kecocokan desain yang dibuat.

4. Mengkodekan sistem

Melakukan aktivitas programming untuk membuat sistem dapat bekerja sesuai dengan keinginan peneliti.

5. Pengujian sistem

Mencoba menjalankan aplikasi dan melakukan tes uji coba dengan berbagai pendekatan untuk melihat berjalannya sistem aplikasi.

6. Evaluasi sistem

Melakukan evaluasi secara keseluruhan pada tiap elemen aplikasi untuk bisa mendapatkan kesimpulan maupun saran kedepannya untuk kebaikan aplikasi.

7. Menggunakan sistem

Aplikasi sudah dapat didistribusikan dan dipakai.

BAB 2

LANDASAN TEORI

2.1. *Game*

Permainan tentunya sudah sangat-sangat tidak asing bagi kita. Sekarang ini sudah banyak jenis-jenis permainan baik itu permainan edukasi, kasual, komepetitif dan lain sebagainya. Permainan tentunya juga disuguhkan dalam bentuk digital dan bisa diakses secara online untuk kebanyakan permianan sekarang. *Game* juga tentunya memiliki peraturan dan selalu ada pemenang sebagai reward untuk pemain (Rozi, 2010).

Permainan dalam bentuk digital untuk peluncurannya pertama kali disuguhkan oleh Steve Russel yang pada tahun 1963 saat itu dari Amerika. *Spacewar* menjadi nama dari permainan yang dibuat dan dikembangkan juga dengan sebuah tim utama yaitu Graezt, ada juga Simson, dan yang terakhir adalah Edward.

2.2. Permainan Tradisional

Permainan dengan tema tradisional tentunya sangat banyak dan dimainkan oleh anak-anak pada jaman dahulu hingga era smartphone hadir. Tentunya permainan tradisional lebih banyak menggunakan tenaga dan kehadiran langsung daripada secara online (Kurniati, 2016). Secara sosial tentunya permainan tradisional sangat baik untuk dilestarikan dan dimainkan untuk menjaga ke aslian dari permainan tersebut.

Selain hal tersebut tentunya menjalin hubungan yang lebih dekat, interaksi dari orang ke orang yang lebih melekat ada pada permainan tradisional, jika dibandingkan dengan permainan-permainan sekarang yang minim akan sosialisasi dan banyak efek-efek negatif seperti berkata kotor, dan cenderung mengintimidasi walau tak secara langsung dilakukan. Alangkah baiknya melestarikan budaya dan permainan lokal (Rustan & Munawir, 2020).

2.3. *Virtual Reality (VR)*

Virtual Reality adalah salah satu dari banyaknya teknologi yang terus hadir di depan mata kita. VR adalah suatu teknologi yang membuat kita seolah-olah bisa merasakan langsung berada di dalam dari realitas virtual atau dunia maya, dan tentunya bisa berinteraksi didalam media tersebut sehingga permainan terasa lebih nyata dan asik (Handayani, 2022). Konsep VR sendiri pertama kali diperkenalkan pada tahun 1960-an, dengan penciptaan Telesphere Mask dan Sensorama oleh Morton (Bown, White, & Boopalan, 2017). Awalnya, konsep yang dilakukan hanya bertujuan agar pengguna merasa bagian dalam video di sekitar mereka. Ide tersebut kemudian dikembangkan kembali oleh Ivan Sutherland (Sutherland, 2001).

Dengan menggunakan VR dan sering sekali input tambahan seperti kontroler, pengguna dapat merasakan sensasi “masuk” pada lingkungan virtual tersebut. Teknologi VR ini menciptakan suatu pengalaman yang imersif dengan menggabungkan elemen visual, audio, dan interaktif sehingga target pengguna bisa merasakan kalau dirinya benar-benar berada didalam permainan tersebut. Teknologi *Virtual Reality* ini diciptakan dengan tujuan agar dapat memudahkan pekerjaan manusia di berbagai bidang seperti kedokteran, arsitek, pendidikan dan lain sebagainya (Brooks Jr., 1999).

Virtual Reality biasanya lingkungan virtual tiga dimensi yang biasanya diakses melalui komputer yang mumpuni untuk menampilkan dunia tiga dimensi tersebut (Riva G., 2015). Beberapa perangkat tambahan yang diperlukan dalam membangun *Virtual Reality* agar terasa semakin nyata diantaranya seperti headmounth, headset, walker, suit dan glove (Zega, 2018). Secara umum, VR dapat dibagi menjadi dua, yaitu non-immersive dan immersive (Wohlgenannt, Simons, & Stieglitz, 2020). VR non-immersive memanfaatkan kombinasi layar yang mengelilingi pengguna untuk menyajikan informasi virtual (Rahouti, Lovreglio, Datoussaïd, & Descamps, 2021).

Contohnya seperti simulasi penerbangan dimana pengguna duduk dikursi dengan di kelilingi banyak layar untuk memberikan persaan sedang duduk di kokpit pesawat atau kursi pengemudi tanpa tenggelam sepenuhnya. VR imersif mengacu pada penggunaan layar yang dapat dikenakan, misalnya HMD, untuk melacak pergerakan pengguna dan menyajikan informasi VR berdasarkan posisi pengguna (Slater & Sanchez-Vives, 2014), yang memungkinkan pengguna merasakan lingkungan virtual 360 derajat.

Teknologi VR yang awalnya dianggap mahal berubah setelah peluncuran *prorotype* Oculus pertama oleh Palmer Luckey, yang kemudian diakuisisi oleh Facebook di tahun 2014. Oculus dikembangkan hingga semakin populer agar mudah digunakan dalam kehidupan sehari-hari di rumah. Terdapat beragam VR yang tersedia di pasaran, seperti *HTC Vive*, *Samsung VR*, *Oculus*, *Google Cardboard*, dan masih banyak lagi.

2.4. Unity 3D

Unity 3D merupakan teknologi yang mumpuni untuk menyajikan suatu aplikasi baik multimedia maupun permainan. Fitur dan akses gratis yang diberikan tentunya sangat membantu dalam belajar mendvelop suatu aplikasi. Berbagai teknologi seperti VR, maupun AR, dan juga banyak lainnya sudah di support oleh *Game engine* ini yang membuat proses pembuatan semakin luas dan terlengkapi.

Selain itu, tentunya bisa support di berbagi media platform untuk uji dan ekspornya, seperti android dalam bentuk APK, juga IOS, Web, dan tentunya masih banyak lagi sudah bisa di support oleh *Game engine* ini selain memberikan fitur mereka juga memberikan dokumentasi yang cukup luas yang bisa digunakan untuk membantu kita (Firdaus, 2021). Unity juga menyediakan *low-level native plugin interface* yang memungkinkan *rendering multi-threaded*. Karena Unity mendukung beberapa konteks grafis, termasuk Direct3D dan OpenGL Core, kita bisa memanggil rendering OpenGL dari plugin eksternal dan menampilkannya langsung di Unity (Wheeler, et al., 2018).

2.5. Blender 3D

Dalam melakukan pembuatan model 3D ataupun animasi tentunya sangat dibutuhkan suatu aplikasi untuk bisa menciptakan visual dan objek yang bagus. Blender 3D mempunyai banyak fitur khusus dalam pembuatan model 3D dan animasi. Tentunya juga sudah support untuk ekstensi dari banyak aplikasi pihak berikutnya untuk mendvelop permainan dan lain- lain. Blender 3D juga dapat diakses secara gratis tanpa membutuhkan lisensi tambahan yang tentunya sangat membantu dalam kegiatan multimedia.

Selain Blender yang *open-source*, blender juga memiliki menyediakan *tool* untuk pemodelan yang memudahkan mengubah *environment*, dan bahkan menyediakan *plug-in* (atau add-on) untuk memperluas fungsionalitas bawaan dengan kode Python

(Reitmann, Neumann, & Jung, 2021).

2.6. *Artificial Intelligence*

Artificial intelegent atau yang dapat diartikan secara harfiah sebagai kecerdasan entitas ilmiah didefenisikan sebagai kecerdasan yang ditambahkan ke dalam proses sistem untuk diatur dalam konteks ilmiah (Zega, 2018).

Kecerdasan yang dimaksudkan dalam sistem bertujuan untuk meniru kecerdasan manusia sehingga mempermudah pekerjaan yang biasa dilakukan oleh manusia. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia.

Artificial Intelligence (Kecerdasan Buatan) adalah rangkaian atau seperangkat teknologi yang bertindak layaknya manusia sehingga dapat mengambil keputusannya sendiri. Pendekatan *Artificial Intelligence* (AI) sering kali didasarkan pada teknik *machine learning* (ML) dapat digunakan untuk memahami struktur lingkungan dengan menyediakan segmentasi semantik dari titik awan 3D, yaitu deteksi dan klasifikasi berbagai objek dalam adegan (Reitmann, Neumann, & Jung, 2021).

Kecerdasan buatan memungkinkan komputer melakukan berbagai fungsi lanjutan, termasuk kemampuan untuk melihat, memahami, dan menerjemahkan bahasa lisan dan tulisan, menganalisis data, membuat rekomendasi, dan lainnya. Sehingga dapat disimpulkan bahwa kecerdasan buatan merupakan simulasi dari kecerdasan manusia yang diterapkan kepada mesin. Beberapa contoh penerapan kecerdasan buatan terdapat pada *Game*, *expert systems*, *natural language processing*, *speech recognition* and *machine vision* (Ghofur, 2016).

2.7. *Pathfinding A**

*Pathfinding A** atau pathing dapat diartikan sebagai plotting dengan menggunakan aplikasi komputer untuk menemukan rute terpendek antara dua titik. Algoritma *Pathfinding* sendiri merupakan suatu metode untuk menemukan graf yang dimulai dari satu titik dan ditarik menjauh menuju node tujuan hingga mendapatkan rute terpendek menuju node tujuan. Rute titik awal ke titik tujuan diperoleh dengan menghindari halangan-halangan yang ada dari path yang ditempuh (Cui & Shi, 2012).

Cara bekerja dari *Pathfinding* sendiri adalah dengan mengantarkan objek/musuh/pemain pada titik acuan yang telah dipilih, namun titik acuan dapat dibuat

berubah-ubah dan tetap akan dapat diikuti secara real- oleh objek tersebut sehingga dapat memutuskan area ataupun lintasan mana yang tercepat yang bisa dipakai untuk mencapai area target tersebut. *Pathfinding* dengan algoritma A* (A-star) adalah teknik yang umum digunakan dalam pengembangan *Game* untuk menemukan jalur terpendek antara dua titik.

Dalam Unity, A* dapat diimplementasikan untuk membantu karakter atau objek bergerak melalui lingkungan dengan cara yang efisien dan cerdas. A* adalah algoritma pencarian jalur yang mengkombinasikan keunggulan dari algoritma Dijkstra dan *Best-First-Search*. Algoritma ini menggunakan fungsi heuristik untuk memperkirakan jarak dari titik saat ini ke tujuan, memungkinkan pencarian yang cepat dan efisien.

Pathfinding A* menyelesaikan masalah yang menggunakan graph untuk perluasan ruang statusnya. Dengan kata lain digunakan untuk menyelesaikan permasalahan yang bisa direpresentasikan dengan graph.

Algoritma A* adalah algoritma yang dikembangkan dengan menerapkan fungsi heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan. Didalam algoritma A* menggunakan dua senarai yaitu *OPEN* dan *CLOSED*.

Pada algoritma A* memiliki tiga kondisi dimana NPC berada di keadaan *OPEN*, berada di keadaan *CLOSED*, dan tidak berada di keduanya. Jika NPC dalam keadaan *OPEN* maka akan dilakukan pengecekan apakah perlu pengubahan *parent* atau tidak tergantung pada nilai *g* melalui *parent* lama maupun *parent* baru yang semua prosesnya dilakukan secara *real-time*. (David M. Broug dan Glenn Seemann, 2004).

Notasi yang dipakai oleh algoritma A* adalah sebagai berikut:

$$f(n) = g(n) + h(n) \quad (1)$$

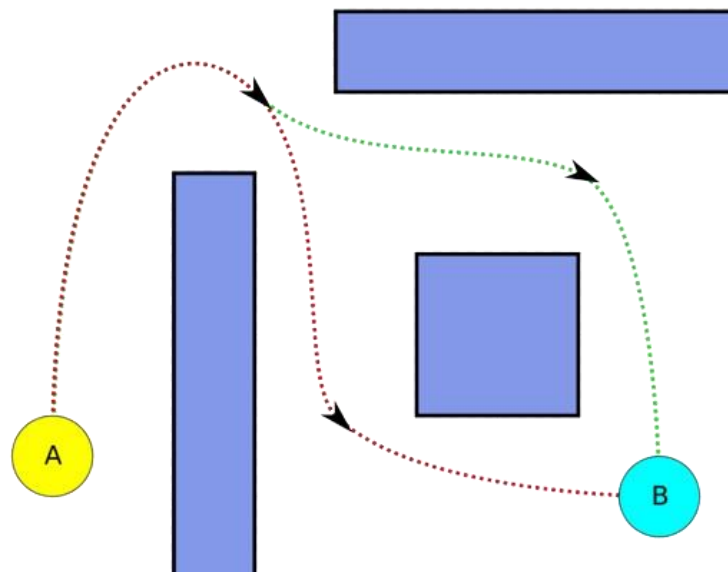
dimana

- $f(n)$ = biaya estimasi terendah
- $g(n)$ = biaya dari node awal ke node n
- $h(n)$ = perkiraan biaya dari node n ke node akhir

Adapun langkah-langkah yang dilakukan oleh algoritma A* adalah sebagai berikut:

- Inisialisasi OPEN LIST
- Letakkan simpul awal pada OPEN LIST
- Inisialisasi CLOSE LIST
- Ikuti langkah-langkah berikut sampai OPEN LIST tidak kosong:
 - Temukan simpul dengan f terkecil pada OPEN LIST dan beri nama "Q".
 - Hapus Q dari OPEN LIST.
 - Generate delapan turunan Q dan tetapkan Q sebagai induknya.
 - Untuk setiap keturunan:
 - Jika menemukan penerus adalah tujuannya, pencarian dihentikan
 - Jika tidak, hitung g dan h untuk penerusnya.
 $\text{penerus.g} = \text{q.g} + \text{jarak yang dihitung antara penerus dan q.}$
 $\text{sukesor.h} = \text{jarak terhitung antara suksesor dan tujuan.}$
 $\text{penerus.f} = \text{penerus.g ditambah penerus.h}$
 - Lewati penerus ini jika node dalam daftar OPEN dengan lokasi yang sama tetapi nilai f lebih rendah dari penggantinya.
 - Lewati penerusnya jika ada simpul dalam CLOSE LIST dengan posisi yang sama dengan penerusnya tetapi nilai f lebih rendah; jika tidak, tambahkan simpul ke ujung OPEN LIST (untuk loop).
 - Push Q ke dalam CLOSE LIST dan akhiri loop sementara.

Berikut ini adalah ilustrasi bagaimana pengambilan jalur dari algoritma *pathfinding a** agar bisa mencapai target ataupun tujuan dengan rute terbaik yang dapat dia lalui.



Gambar 2. 1 Pathfinding A*



2.8. Unified Modelling Language (UML)

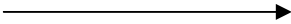
Unified Modeling Language (UML) bisa memberikan gambaran tentang apa dan alur yang bagaimana yang kita lakukan dalam mendevlop suatu aplikasi. UML memberikan gambaran jelas baik dari segi alur, aktivitas, dan sequence dari tiap scene yang dibuat (Sonata, 2019). Beberapa pembagian dari UML itu sendiri berupa :

a) *Use Case Diagram*

Use Case Diagram adalah sebuah diagram yang menunjukkan hubungan antara actors dan. Digunakan untuk analisis dan desain sebuah sistem.

Tabel 2.1 Use Case Diagram

No.	Simbol	Keterangan
1.		Aktor <i>User</i> pengguna aplikasi
2.		<i>Use Case</i> Acuan proses dari sistem



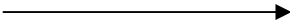



3.		<i>Unidirectional Association</i> Hubungan yang ada antara actor dan case
----	---	--


(Sumber : Simatupang & Sianturi, 2019)

b) Activity Diagram

Activity diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem.

Tabel 2.2 Activity Diagram

No.	Simbol	Keterangan
1.	 Start	Kondisi Awal Awal dari aktifitas yang akan dilakukan
2.	 End	Kondisi Akhir Akhir dari aktifitas yang dilakukan
3.		Kondisi transisi Transisi dari antar jalannya aktifitas
4.	 Swimlane	<i>Swimlane</i> Aktor dari jalannya aktifitas
5.		Aktivitas Macam-macam dari aktifitas yang ada
6.		Aktivitas Macam-macam dari aktifitas yang ada

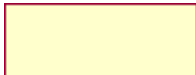


7.		Pengecekan kondisi Cek kondisi pada tiap aktifitas yang ada
----	---	--

(Sumber : Simatupang & Sianturi, 2019)

c) Sequence Diagram

Sequence Diagram adalah sebuah diagram yang menggambarkan kolaborasi dari objek-objek yang saling berinteraksi antar elemen dari suatu class.

Tabel 2.3 Sequence Diagram

No.	Simbol	Keterangan
1.	 ⋮	Objek Objek apa saja yang ada dalam sequence
2.		Pesan ke Objek sendiri Pesan dari objek yang terarah
3.		Pesan Objek Pesan yang tersampaikan dari arah yang ditentukan

(Sumber : Simatupang & Sianturi, 2019)

2.9. Android

Android adalah salah satu platform yang banyak digunakan pada *smartphone* pintar saat ini dan tentu memiliki cakupan yang lebih luas dikarenakan harga dan efisiensinya lebih mudah untuk diakses dan juga ini bersipat open source yang bisa sebagai wadah mendevelop aplikasi (Safaat, 2011). Android memiliki banyak versi sesuai dengan kebutuhan kita yang dibedakan berdasarkan nama-namayang unik dan sesuai dengan spesifikasi yang dibutuhkan.

2.10. Objek 3D

3D modeling adalah proses menciptakan representasi digital dari objek tiga dimensi

menggunakan perangkat lunak komputer. Model 3D ini digunakan di berbagai industri, termasuk film, permainan video, arsitektur, desain produk, dan animasi. 3D modeling adalah proses pembuatan representasi matematis dari permukaan objek dalam tiga dimensi melalui perangkat lunak khusus. Tujuan utamanya adalah menciptakan model digital yang dapat digunakan dalam berbagai aplikasi, seperti visualisasi, simulasi, dan produksi animasi. (Phillips, 2017).

- a. Konsep dan Desain: Memulai dengan sketsa atau konsep dasar dari objek yang telah dibuat.
- b. Pemodelan Geometri: Menggunakan perangkat lunak 3D untuk membangun bentuk dasar dari objek, sering kali dimulai dengan primitive shapes seperti kubus, bola, atau silinder.
- c. Pemodelan Detail: Menambahkan detail lebih lanjut dengan teknik seperti extrusion, beveling, dan subdivision.
- d. *Texturing*: Menerapkan tekstur pada model untuk memberikan warna, pola, dan material yang realistis.
- e. *Rigging*: Menambahkan kerangka pada model untuk memungkinkan animasi. Hal ini terutama penting untuk karakter dalam permainan video dan film.
- f. *Rendering*: Proses menghasilkan gambar atau animasi akhir dari model 3D.

2.11. Penelitian Terdahulu

Berikut ini adalah daftar penelitian terdahulu yang dipakai untuk menunjang kualitas dari penelitian yang dibuat :

Tabel 2.4 Penelitian Terdahulu

No	Peneliti	Teknologi	Keterangan
1.	Milak, A. S., Hidayat, E. W., & Aldya, A. P. (2020)	AI & 2D <i>Game</i> untuk BalapEngrang	Dengan pengujian fungsionalitas <i>User Acceptance Test (UAT)</i> didapat nilai sebesar 82,58%
2.	Hermawan, L. & Ismiati, M. B. (2022)	Augmented Reality Minimax	Permainan melawan AI memperoleh hasil kemenangan 63% untuk AI.

3.	Tamra, et. al. (2021)	Permainan edukasi denganPuzzle	Permainan ditujukan untuk anak-anak diuji dengan angketdiperoleh indeks persentase 96%.
4.	Zikky, M. (2016)	Algoritma <i>Pathfinding</i> A*dalam PacMan	A* algorithm diterapkan padalawan dalam permainan Pac- Man (three ghosts), yang jalurnya didesain menggunakan konsep NavMesh.
5	Aditya Setiyawan, Paulus Harsadi, Sri Siswanti (2019)	Penerapan Pathfinding Menggunakan Algoritma A* Pada Non Player Character (NPC) Di Game	Artificial Intelligence (AI) di game mampu membuat karakter baik player maupun NPC (<i>Non Player Character</i>) memiliki perilaku manusia yang sesungguhnya baik bergerak maupun berfikir
6	Agung, Erindani, Fauzi (2022)	Implementasi Metode Pathfinding dengan Algoritma A* pada Game Rogue-like menggunakan Unity	Pada permainan yang dibuat implementasi dari <i>pathfinding a*</i> berhasil membuat objek/NPC bisa bergerak dan menentukan arah sendiri dengan baik.

BAB 3

ANALISIS DAN PERANCANGAN

3.1. Analisis Sistem

Perancangan permainan petak umpet VR ini tentunya menggunakan teknologi terkini *Virtual Reality* untuk pengaplikasian dari kinerja aplikasinya agar seakan-akan kita memang masuk kedalam dunia permainan dan merasakannya secara langsung. Selain dari permainan petak umpet ini dibuat dengan berbagai misi, ditambahkan juga kuis disela-sela permainan agar permainan lebih menarik dengan menambahkan materi pada kuis tersebut. Dalam permainan ini ada dua mode bermain yaitu pencari dan penjaga.

3.2. Analisis Masalah

Permainan petak umpet sebagai salah satu permainan tradisional, dianggap kurang praktikal dan menarik karena membutuhkan aktivitas fisik dan pemain yang banyak. Yang mengakibatkan pula permainan tersebut mulai tergeser dengan berkembangnya teknologi. Untuk mempertahankan permainan tradisional kepada anak-anak Indonesia generasi penerus maka dibutuhkan inovasi dalam visualisasi melibatkan teknologi dan pemanfaatan AI, maka perlu dilakukan implementasi permainan tradisional petak umpet menjadi permainan digital berbasis android.

3.3. Analisis Kebutuhan Sistem

Dalam pembangunan sebuah sistem permainan tentunya dibutuhkan beberapa komponen yang membangun permainan itu sendiri menjadi lebih baik dan terpenuhi dari segala aspek penting. Kebutuhan tersebut dibagi menjadi dua bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional yang lebih jelas seperti penjelasan di bawah ini:

a) Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan utama dan performa utama yang harus dimiliki yang berisi layanan utama yang dihadirkan dalam permainan yaitu :

- g. Mampu menerapkan teknologi *Virtual Reality* dengan baik pada permainan. Mampu menampilkan objek 3D dengan jelas dan baik.
- h. Mampu menambahkan audio untuk membuat permainan lebih menyenangkan.
- i. Mampu mengimplementasikan AI dengan algoritma *Pathfinding A** pada musuh.

b) Kebutuhan Non-Fungsional

Kebutuhan selanjutnya yaitu kebutuhan non-fungsional diperlukan untuk mendukung sistem utama yang telah dibuat seperti :

1. Performa

Aplikasi mampu berjalan dengan baik sesuai dengan minimum device yang ditentukan.

2. Cost

Aplikasi gratis untuk diinstal dan tidak memerlukan biaya tambahan untuk dipergunakan.

3. Storage

Aplikasi menggunakan penyimpanan internal tanpa butuh tambahan memori yang besar.

3.4. Analisis Kebutuhan Game

Untuk membangun sebuah permainan dibutuhkan device yang memadai dengan minimum requirement agar bisa menjalankan aplikasi-aplikasi yang lumayan berat dan besar pada pembangunan permainan, maka dari itu dibutuhkan :

1. *Windows 10 x 64*

2. *Inter Core i5 and 8 GB RAM*

3. *NVIDIA 940M Graphic Card*

4. *Unity 3D*

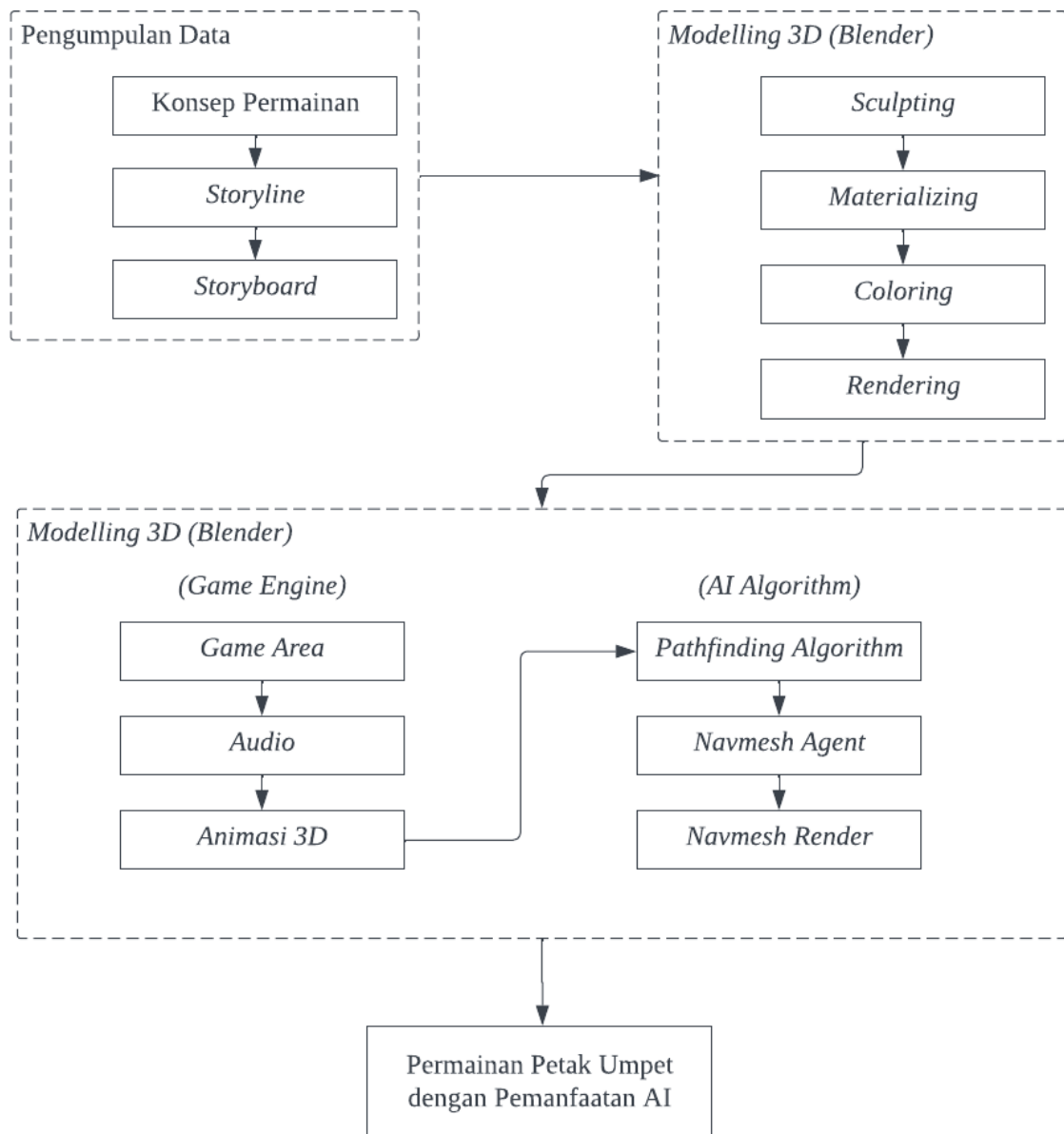
5. *Blender 3D*

Sementara itu untuk menjalankan APK Android maka dibutuhkan :

1. OS : Minimum Android v 10

3.5. Arsitektur Umum

Berikut ini adalah gambar dan penjelasan dari arsitektur aplikasi yang telah dibuat agar bisa lebih difahami dan lebih mudah untuk dimengerti.



Gambar 3. 1 Arsitektur Umum

Berdasarkan Gambar 3.1 di atas dapat dijelaskan secara rinci Arsitektur Umum sebagai berikut:

A. Pengumpulan Data

Tahap awal pembuatan permainan ini adalah menyusun konsep utama dari permainan meliputi dua hal berikut :

a) Konsep Permainan

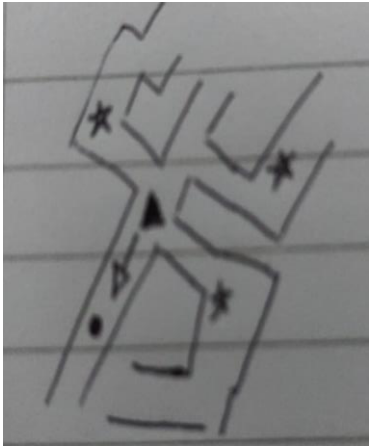
Pada awal pembuatan aplikasi kita telah membuat konsep cerita maupun tema untuk aplikasi yang telah dibuat, dimana kita telah mengangkat salah satu permainan masa kecil yaitu petak umpet dan mengimplementasikan AI (*Artificial Intelligence*) di dalamnya.




b) Storyline

Storyline merupakan cerita ataupun rentetan peristiwa yang telah dibuat di dalam permainan.

c) Storyboard

Storyboard merupakan penggambaran storyline agar terlihat lebih jelas dan bisadijadikan pedoman terarah untuk pengembangan aplikasi.

Storyboard	Storyline
	<p>Pada tipe permainan pertama akan ada mode bermain pencari. Pemain akan mencari <i>item</i> tersembunyi dalam labirin agar bisa melaju ke <i>stage</i> berikutnya tentu dengan gangguan dari musuh dengan kecerdasan <i>pathfinding a*</i></p>

	<p>Akan ada kuis jika permainan di <i>reset</i> karena pemain tertangkap dan harus menyelesaikan kuis agar bisa melanjutkan permainan</p>
	<p>Pada tipe permainan kedua pemain akan menjadi penjaga dimana akan menangkap musuh dengan kecerdasan <i>pathfinding a*</i> yang mampu berlari ketika mendeteksi pemain mendekat</p>
	<p>Setiap <i>stage</i> kecepatan berlari musuh akan meningkat baik saat menjadi pengejar maupun pelari</p>

B. Modelling 3D (Blender)

Pembuatan model telah menggunakan aplikasi Blender 3D dalam pembuatannya dan telah melalui beberapa tahapan proses yaitu :

a) *Sculpting*

Merupakan proses memahat model di tahapan awal pembuatannya dibuat semirip mungkin dengan objek referensi yang ada.

b) *Materializing*

Setelah pemahatan model selesai dilakukan proses pemberian material agar model lebih nyata dan tidak terlalu polos.

c) Coloring

Merupakan proses pemberian warna pada model agar terlihat lebih menarik.

d) Rendering

Tahap akhir dari pembuatan model adalah rendering untuk membuat model dalam hasil final dan disesuaikan formatnya untuk digunakan kembali.

C. Unity 3D (*Game & AI Engine*)

Dalam pembuatan permainan kita telah menggunakan aplikasi Unity 3D untuk membantu dalam proses developing permainan. Beberapa tahap yang telah kita lakukan yaitu :

a) Game Area

Tahapan ini adalah proses pembuatan area permainan pada setiap *stage* yang telah dimainkan.

b) Audio

Pembuatan suara pada permainan sangat penting untuk membuat permainan terasa lebih menarik dan menyenangkan.

c) Animasi 3D

Proses pembuatan animasi 3D digunakan untuk menggerakkan karakter dalam permainan.

d) Pathfinding Algorithm

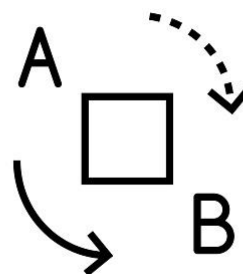
Mengangkat permainan petak umpet maka kita telah memberikan kemampuan padamusuh untuk mampu mencari kita dengan situasi dan kondisi yang ditentukan, dimana musuh akan diberikan kemampuan untuk mendeteksi target tujuan dan bisa mengejar secara ***real-time*** dan mencari rute tercepat untuk menggapai pemain. Dengan algoritma ini musuh akan memiliki kecepatan berlari dan arah target tujuan pada permainan yang telah dibuat.

Pathfinding A* menyelesaikan masalah yang menggunakan graph untuk perluasan ruang statusnya. Dengan kata lain digunakan untuk menyelesaikan permasalahan yang bisa direpresentasikan dengan graph. Algoritma A* adalah sebuah algoritma yang telah dikembangkan. Dengan menerapkan fungsi heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi

yang diinginkan.

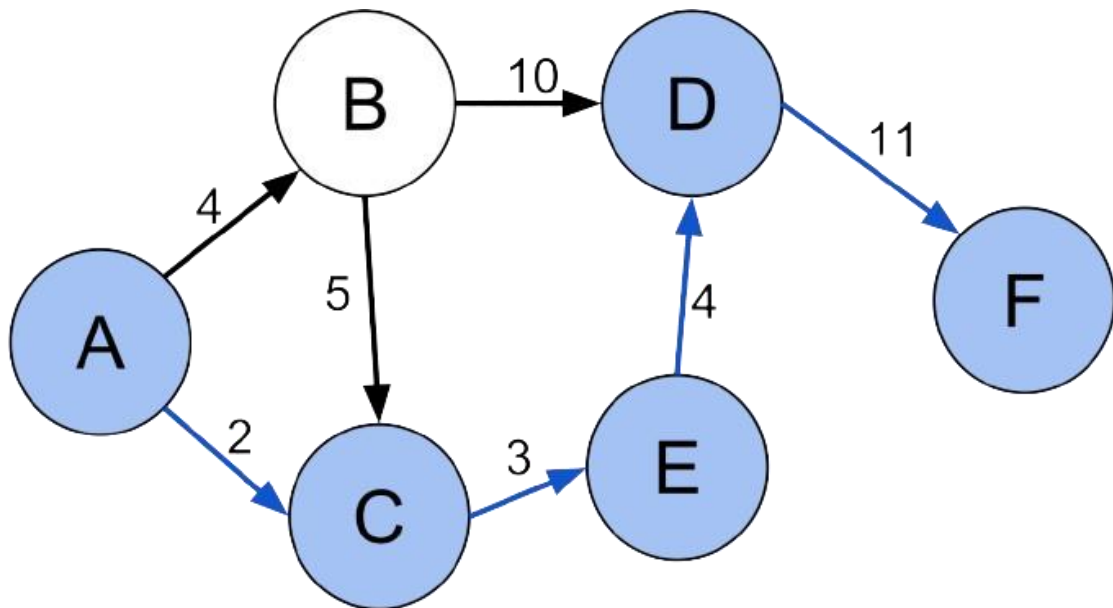
Didalam algoritma A* menggunakan dua senarai yaitu *OPEN* dan *CLOSED*. Pada algoritma A* memiliki tiga kondisi dimana NPC berada di keadaan *OPEN*, berada di keadaan *CLOSED*, dan tidak berada di keduanya. Jika NPC dalam keadaan *OPEN* maka akan dilakukan pengecekan apakah perlu pengubahan *parent* atau tidak tergantung pada nilai melalui *parent* lama maupun *parent* baru yang dimana semua proses ini dilakukan secara *real-time*.

Pada intinya, metode *pathfinding* mencari graf dengan memulai dari satu simpul dan menjelajahi simpul yang berdekatan sampai simpul tujuan tercapai, umumnya dengan maksud untuk menemukan rute termurah. Meskipun metode pencarian grafik seperti pencarian luas pertama akan menemukan rute jika diberi waktu yang cukup, metode lain, yang "menjelajahi" grafik, akan cenderung mencapai tujuan lebih cepat. Analoginya adalah seseorang berjalan melintasi ruangan; daripada memeriksa setiap kemungkinan rute terlebih dahulu, orang tersebut umumnya akan berjalan ke arah tujuan dan hanya menyimpang dari jalur untuk menghindari halangan, dan membuat penyimpangan sekecil mungkin.



Gambar 3. 2 Ilustrasi *Pathfinding A**

Dalam menentukan arah tentunya algoritma ini sangat bisa untuk mencari rute tercepat dan ter efisien dalam mencapai target lokasi sehingga sangat cocok digunakan untuk petak umpet ini.



Gambar 3. 3 Ilustrasi Arah *Pathfinding A**

e) *Navmesh Agent*

Membuat agent atau karakter yang ditandai sebagai musuh yang bisa berkeliling di area permainan.

f) *Navmesh Render*

Merupakan proses baking pada area permainan agar pemain ataupun musuh tidak keluar dari area bermain.

D. Output

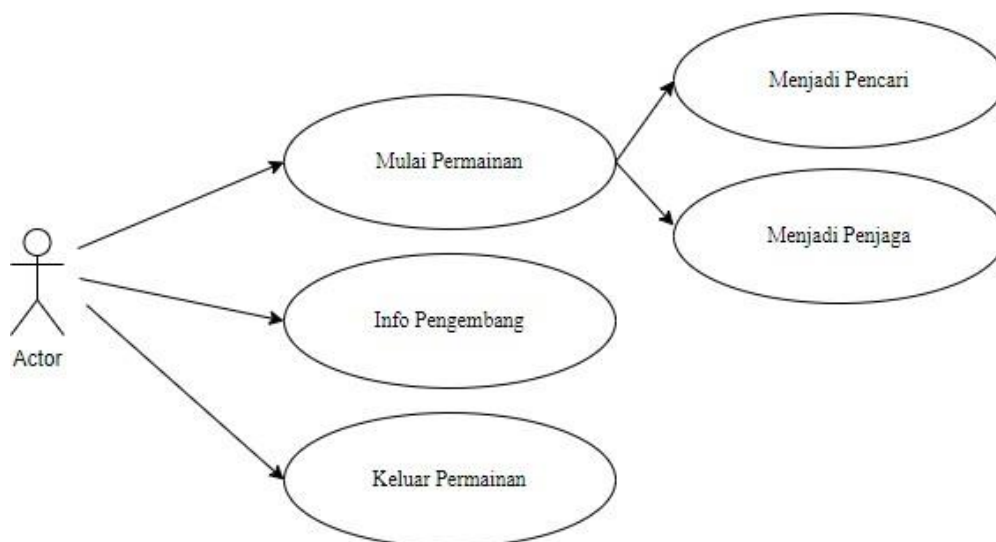
Output dari *Game* ini berupa permainan petak umpet dengan pemanfaatan (AI).

3.4. Pemodelan Sistem

Pemodelan sistem dilakukan untuk melihat bagaimana sistem telah berjalan dan apa saja aktifitas-aktifitas yang ada pada sistem. Pada bagian ini kita telah menerapkan UML ataupun *Unified Modelling Language* untuk menggambarkan secara detail sistem-sistem yang ada :

A. Use Case Diagram

Use case diagram adalah sebuah diagram yang menunjukkan hubungan antara actors dan use cases. Digunakan untuk analisis dan desain sebuah sistem dapat dilihat pada Gambar 3.2 di bawah ini :



Gambar 3. 4 Use Case Diagram

Permainan melibatkan 1 aktor yang menjadi *user*. Berikut proses dari aktor tersebut pada Tabel 3.1 di bawah ini :

Tabel 3.1 Proses Aktor Use Case

No	Aktor	Deskripsi
1	Pemain	<i>Actor</i> sebagai pemain permainan

Berikut ini skenario memulai permainan pada Tabel 3.2 :

Tabel 3.2 Proses Memulai Permainan

Identifikasi	
Nama	Permainan Petak Umpet VR
Tujuan	Menampilkan permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Halaman Utama

Aksi Aktor	Reaksi Sistem
1. Menjadi Pencari	1. Menyelesaikan Misi dan Mengalahkan Musuh
2. Menjadi Penjaga	2. Mengejar Musuh
Kondisi Akhir	Hasil

Berikut ini skenario memulai permainan pada Tabel 3.3 :

Tabel 3.3 Proses Info Pengembang

Identifikasi	
Nama	Info Pengembang
Tujuan	Menampilkan Informasi Pengembang
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Halaman Utama
Aksi Aktor	Reaksi Sistem
1. Masuk ke Menu info pengembang	2. Menampilkan Informasi
	3. Mendapat Hasil
Kondisi Akhir	Hasil

Berikut ini skenario memulai permainan pada Tabel 3.4 :

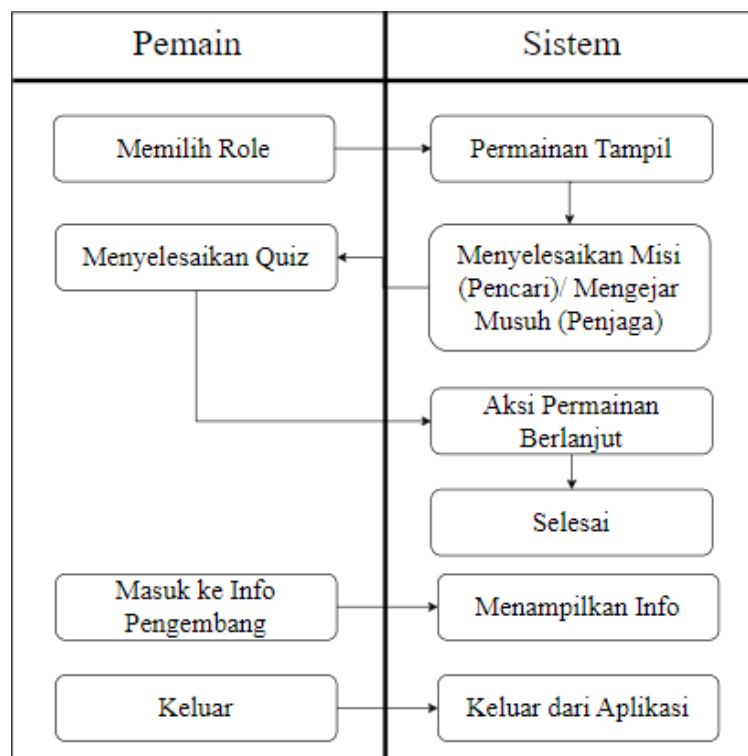
Tabel 3.4 Proses Keluar Aplikasi

Identifikasi	
Nama	Keluar
Tujuan	Keluar dari permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Halaman Utama
Aksi Aktor	Reaksi Sistem
1. Masuk ke Menu keluar aplikasi	2. Keluar dari aplikasi
	3. Mendapat hasil
Kondisi Akhir	Hasil

B. Activity Diagram

Activity Diagram adalah suatu diagram yang menggambarkan konsep aliran

data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem dapat kita lihat seperti Gambar 3.3 di bawah ini:

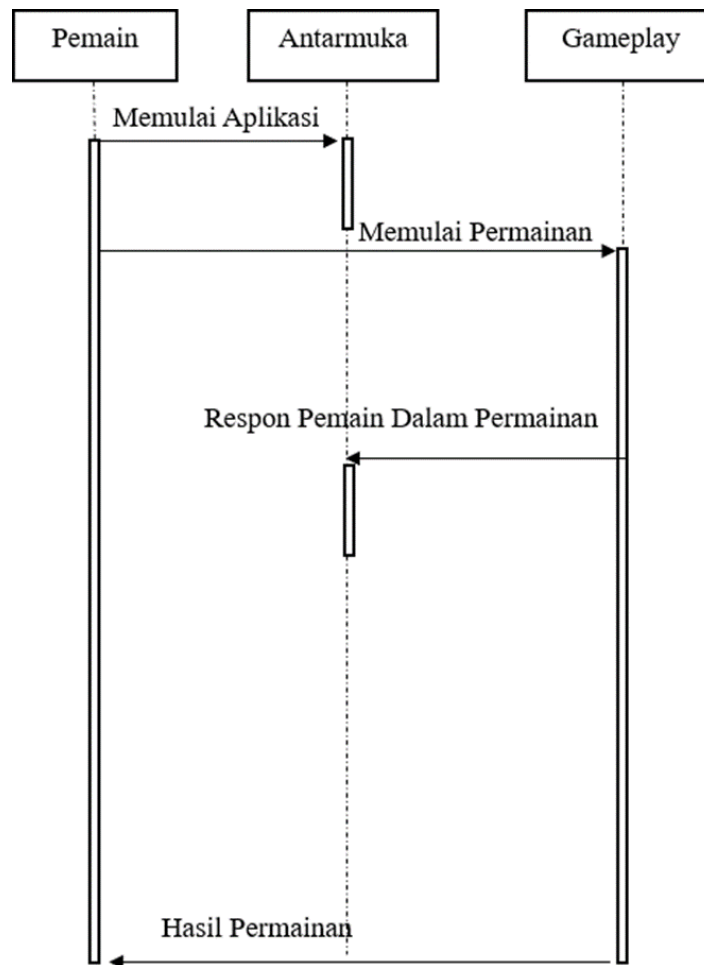


Gambar 3. 5 Activity Diagram

Pemain bisa memilih dua role bermain apakah menjadi penjaga ataupun pencari, jika menjadi pencari maka pemain harus mencari item rahasia di dalam permainan dan tentunya musuh akan menjadi penjaga yang mengganggu pemain, apabila tertangkap pemain harus menyelesaikan *quiz* agar bisa melanjutkan aksi bermain. Jika pemain menjadi penjaga maka pemain harus menangkap musuh dengan menekan bel pada badan musuh agar bisa melanjutkan ke *stage* berikutnya. Permainan terdiri dari beberapa *stage* dengan tingkat kesulitan yang berbeda beda dan *quiz* juga telah membawa materi yang semakin sulit di tiap *stage* nya.

C. Sequence Diagram

Sequence Diagram adalah sebuah diagram yang menggambarkan kolaborasi dari objek-objek yang saling berinteraksi antar elemen dari suatu class dapat kita lihat seperti Gambar 3.8 di bawah ini:



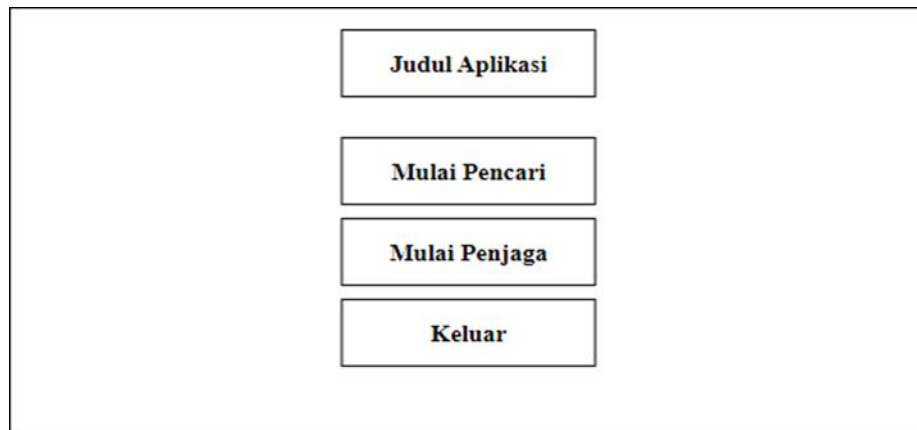
Gambar 3. 6 Sequence Diagram

3.5. Perancangan Antar Muka

Perancangan antar muka dilakukan untuk memberikan gambaran bagaimana aplikasi akan ditampilkan nantinya. Melakukan desain antar muka juga mempermudah peneliti melakukan developing pada permainan dan tentunya terarah dalam pengerjaannya.

A. Tampilan Main Menu

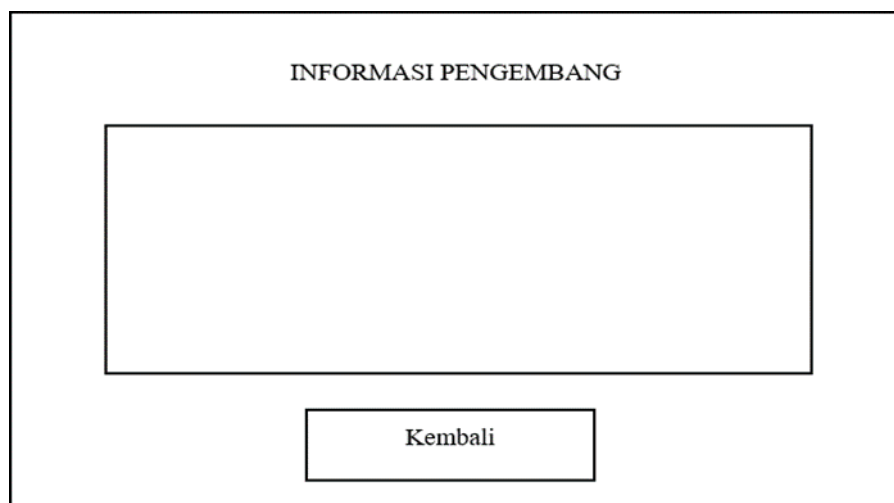
Tampilan main menu adalah tampilan awal saat pertama kali membuka permainan dimana berisi beberapa tombol yaitu Memulai Permainan, Informasi Pengembang dan Keluar dari permainan.



Gambar 3. 7 Desain Tampilan Main Menu

B. Tampilan Informasi Pengembang

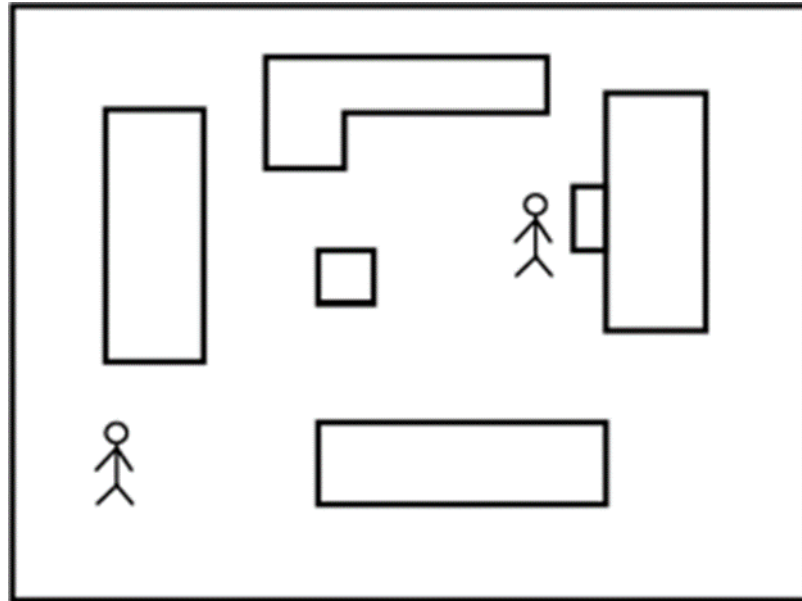
Tampilan ini berisikan informasi tentang developer yang membuat aplikasi dimana telah menrincikan informasi-informasi tentang pembuat maupun pendukung yang ada.



Gambar 3. 8 Desain Info Pengembang

C. Tampilan *Gameplay* permainan

Pada bagian ini adalah desain tampilan bagaimana di dalam permainan ataupun ketika saat bermain dimana adanya tempat seperti labirin yang jadi tempat pencari dan penjaga.



Gambar 3. 9 Desain *Gameplay* Permainan

D. Tampilan *Pathfinding A**

Pada bagian ini adalah program yang digunakan pada musuh dengan algoritma *pathfinding a** agar musuh mampu mendeteksi titik tujuan dan bisa bergerak secara *real-time* untuk mengejar pemain dan mampu mencari rute tercepat.

Pseudo-code :

Initialize variables:

- player: reference to player's position
- basePoint: reference to base point's position
- chaseDistance: distance within which agent starts chasing player
- returnToBaseDistance: distance within which agent returns to base
- agent: navigation agent
- isChasing: boolean indicating if agent is chasing player

Check if agent is chasing:

- If yes:
 - Set agent's destination to player's position
 - If agent collides with player:
 - Stop chasing
 - Reset chase distance to 0
 - Log collision event
 - If collected items are less than 3:
 - Reset collected items count to 0
 - Deactivate items 1, 2, and 3

- Activate quiz canvas
- If no:
 - If player is within chase distance:
 - Start chasing
 - Log chasing event
 - If agent is far from base:
 - Set agent's destination to base point
 - Log return to base event

Jika kita pecah berdasarkan adegan pada aplikasi saat bermain maka lebih jelasnya akan terlihat seperti rincian berikut ini :



Gambar 3. 10 AI Mengejar Pemain

Pseudo-code :

```
Jika jarakKePemain < jarakMengejar:
    MulaiMengejar()
    Cetak("Mulai mengejar pemain!")
```

Dalam keadaan tersebut maka musuh akan memulai pengejaran dan fungsi dari StartChasing akan dijalankan, sementara itu pada console sistem juga akan diberikan *command* bahwa musuh sedang mengejar pemain dimana untuk meverifikasi bahwa sistem berjalan dengan baik.

Pseudo-code :

```
StartChasing()

// Mengatur jarak pengejaran
set chaseDistance to 50

// Menandai bahwa pengejaran telah dimulai
```

```

set isChasing to true
// Mengatur jarak berhenti agent sebelum mencapai pemain
set agent.stoppingDistance to 1.5
// Menampilkan pesan debug bahwa agent mulai mengejar pemain
log "Agent mulai mengejar pemain"
// Mengaktifkan Item1
activate Item1
// Mengaktifkan Item2
activate Item2
// Mengaktifkan Item3
activate Item3
// Menonaktifkan ResetT
deactivate ResetT

```

Selanjutnya akan ada posisi dimana musuh berhasil tertangkap dan AI musuh Kembali ke pos penjaga dan pemain harus melaporkan kembali dengan menekan tombol pada pos penjaga untuk memulai kembali pencarian.

Pseudo-code :

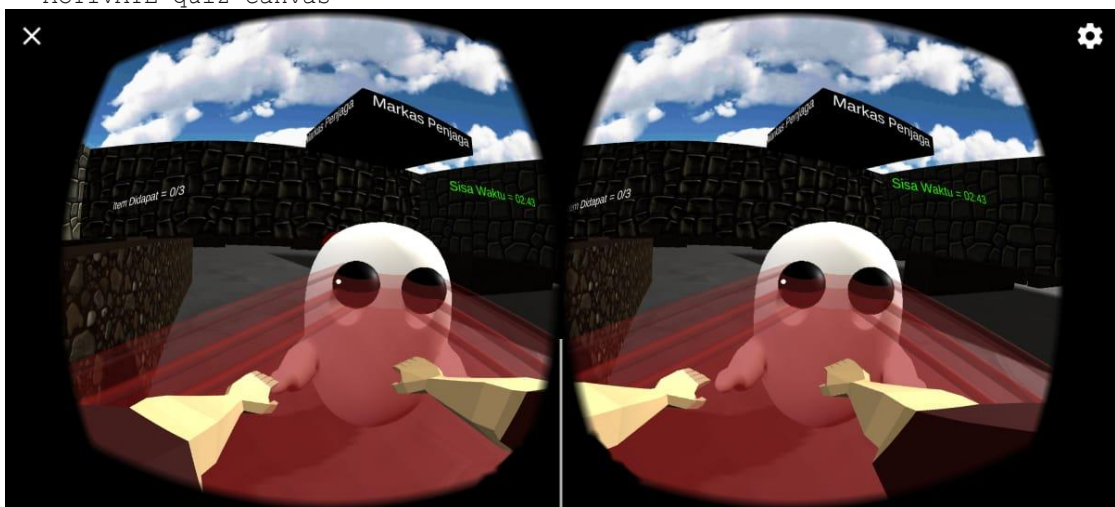
```

IF distance to player is less than 1.5
  CALL function to stop chasing
  SET chase distance to 0
  LOG "Agent collided with player! Returning to base."

IF item count is less than 3
  SET item count to 0
  DEACTIVATE item 1
  DEACTIVATE item 2
  DEACTIVATE item 3

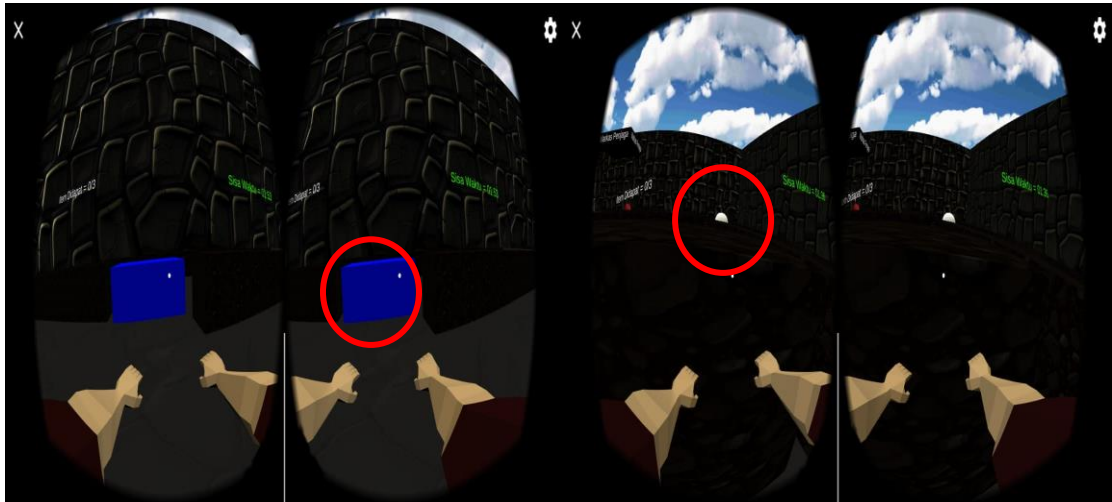
ACTIVATE quiz canvas

```



Gambar 3. 11 AI Menangkap Pemain

Selanjutnya pemain akan kembali ke pos penjaga untuk melaporkan kembali permainan baik untuk memulai Kembali setelah tertangkap maupun setelah menyelesaikan misi



Gambar 3. 12 Pemain Bersembunyi

Pada saat pemain bersembunyi di pintu ruangan rahasia dan menutupnya maka penjaga ataupun musuh akan menghentikan langkahnya karena tidak menemui ruang untuk menangkap pemain. Kode program yang berperan adalah sebagai berikut:

Pseudo-code :

```
boolean isChasing = false

if isChasing is true
    set agent's destination to player's position
```

Status dari *Chasing* ataupun sistem pelacakan akan terhenti dan kembali menjadi *false*.



Gambar 3. 13 *Navmesh Agent AI*

Pada setiap level pengaturan dari *Navmesh Agent* pada AI musuh akan di *adjust* dan ditingkatkan agar level tidak terasa monoton dan tentunya menjadikan permainan lebih menantang dan ada tantangan tersendiri.

BAB 4

IMPLEMENTASI SISTEM

4.1. Implementasi Permainan

Pada tahap akhir dari pengerjaan sistem telah ada proses yang diberi nama implementasi, dimana pada prosesi ini telah dilakukan implementasi dari tiap-tiap perencanaan yang dimuat pada bab-bab sebelumnya. Pengerjaan model tentunya dibuat dengan aplikasi Blender 3D sebagai media untuk melengkapi desain dan model, sementara itu untuk pengerjaan program telah dilakukan pada aplikasi Unity 3D yang menjadi induk aplikasi dan tempat hasil akhir aplikasi dibuat. Secara lengkap dari tiap hasil telah dijelaskan dengan rinci seperti pembahasan berikut ini.

4.2. Tampilan Permainan Pada *Game Engine* (Unity)

Pada aplikasi VR PETAK UMPET yang dibuat untuk platform android yang dibuat tentunya ditampilkan dengan baik, berikut ini tampilan aplikasi yang telah dibuat pada Unity di bawah ini :

A. Tampilan Menu

Pada tampilan dari menu utama yang dibuat telah terdiri dari tiga tombol utama yaitu memulai sebagai pencari, memulai sebagai penjaga, dan keluar dari aplikasi, seperti Gambar 4.1 di bawah ini:



Gambar 4. 1 Tampilan Menu

Keterangan Gambar 4.1 menjelaskan bahwa :

1. Terdapat beberapa tombol untuk mengisi menu utama yaitu Mulai Pencari dan Mulai Penjaga, dan Keluar.
2. Tombol Penjaga berfungsi untuk mode bermain Penjaga.
3. Tombol Pencari berfungsi untuk mode bermain Pencari.
4. Tombol Keluar berfungsi untuk keluar dari aplikasi.

B. Tampilan Info Pengembang

Pada bagian ini telah menampilkan informasi dari pengembang dan tools yang digunakan untuk membuat permainan :



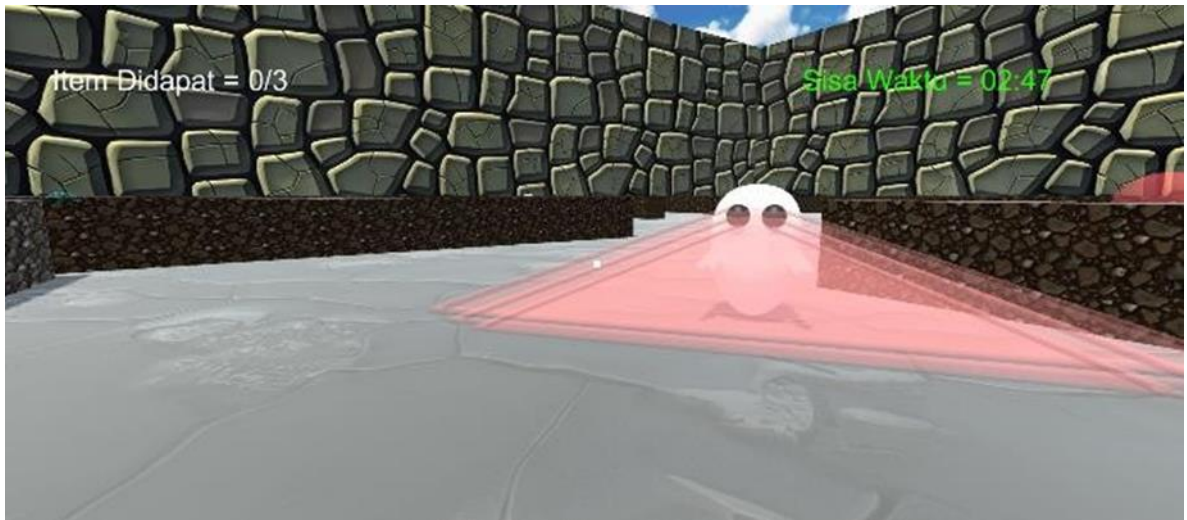
Gambar 4. 2 Tampilan Developer

Keterangan Gambar 4.2 menjelaskan bahwa :

1. Menu Info Pengembang berfungsi untuk memberikan informasi tentang pengembang aplikasi yang sedang dimainkan.
2. Tombol Kembali berfungsi untuk Kembali ke menu utama.

C. Tampilan Permainan

Pada tampilan permainan telah menampilkan dunia bermain dan juga ornament-ornamen yang ada di dalam permainan tentunya dengan karakter, musuh, environment pendukung dan hal-hal yang menunjang kinerja permainan :



Gambar 4. 3 Tampilan Permainan

Keterangan Gambar 4.3 menjelaskan bahwa :

1. Tampilan permainan berisi beberapa misi yang harus diselesaikan beserta dengan sebuah karakter yang digunakan untuk memainkan permainan.
2. Sisa waktu adalah jumlah waktu total untuk bermain.

D. Tampilan Menjalankan Misi

Pemain akan mencari berbagai item tersembunyi di are-area labirin dan harus mendapatkannya sebelum tertangkap oleh penjaga untuk dapat melanjutkan permainan, seperti Gambar 4.4 di bawah ini :

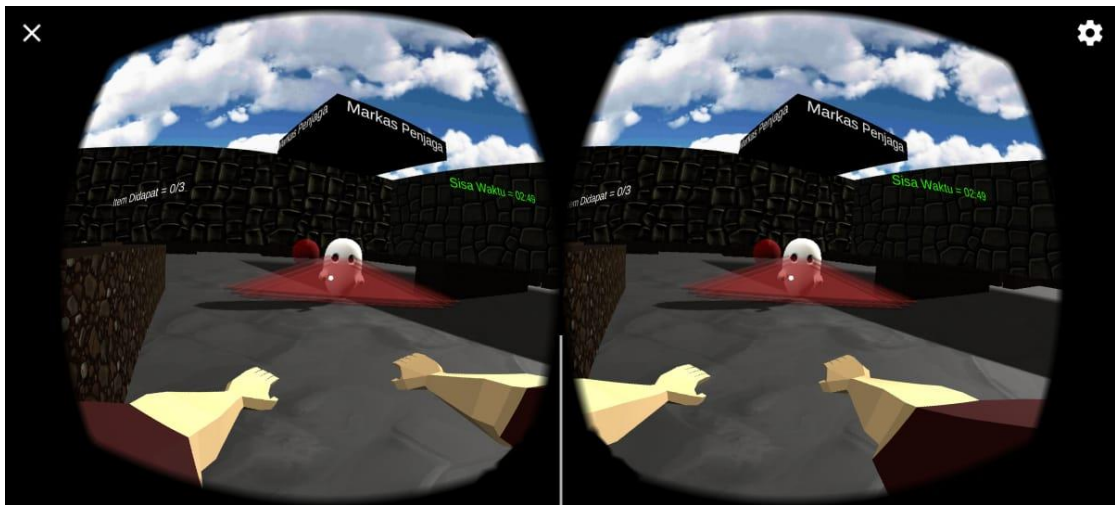


Gambar 4. 4 Tampilan Menjalankan Misi

Keterangan Gambar 4.4 menjelaskan bahwa :

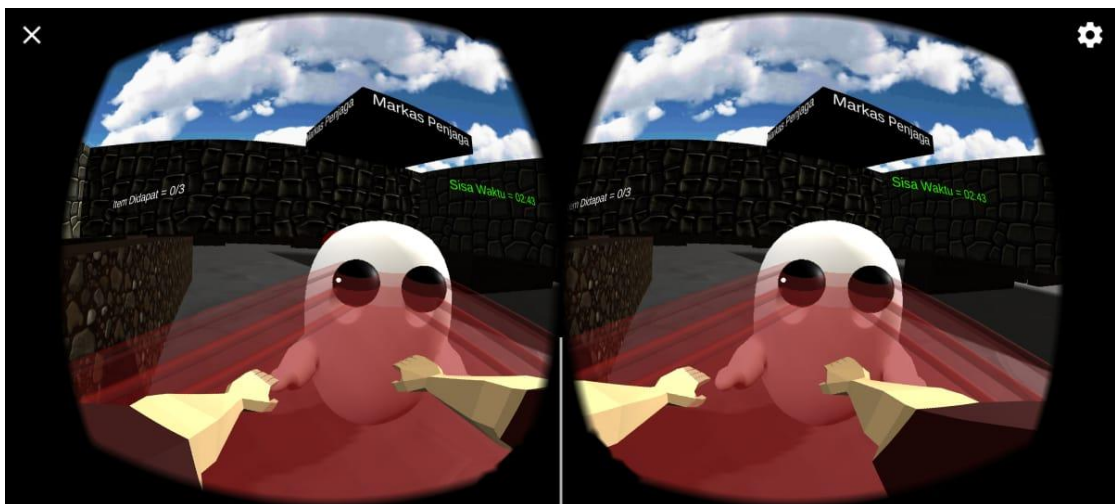
1. Saat menjalankan misi telah ada beberapa bola-bola misi yang harus diselesaikan sesuai dengan misi pada tiap *stage* agar bisa menyelesaikan permainan.

2. Tombol navigasi arah digunakan untuk berjalan menyusuri area.



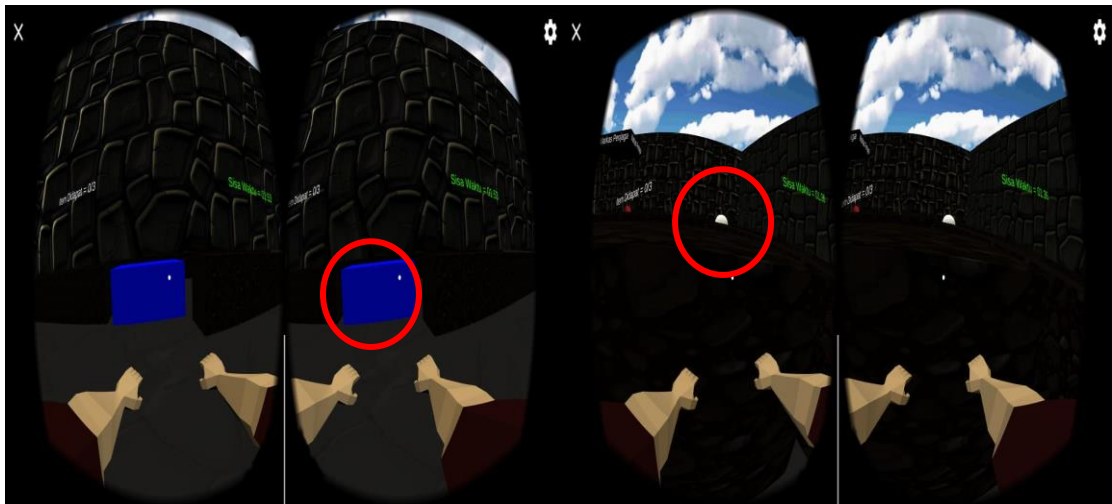
Gambar 4. 5 Tampilan Musuh Mengejar Pemain

Saat permainan dimulai *Artificial Intelligence* pada musuh secara *real-time* dengan algoritma *Pathfinding A** telah aktif dan musuh akan langsung mengejar kemanapun pemain pergi.



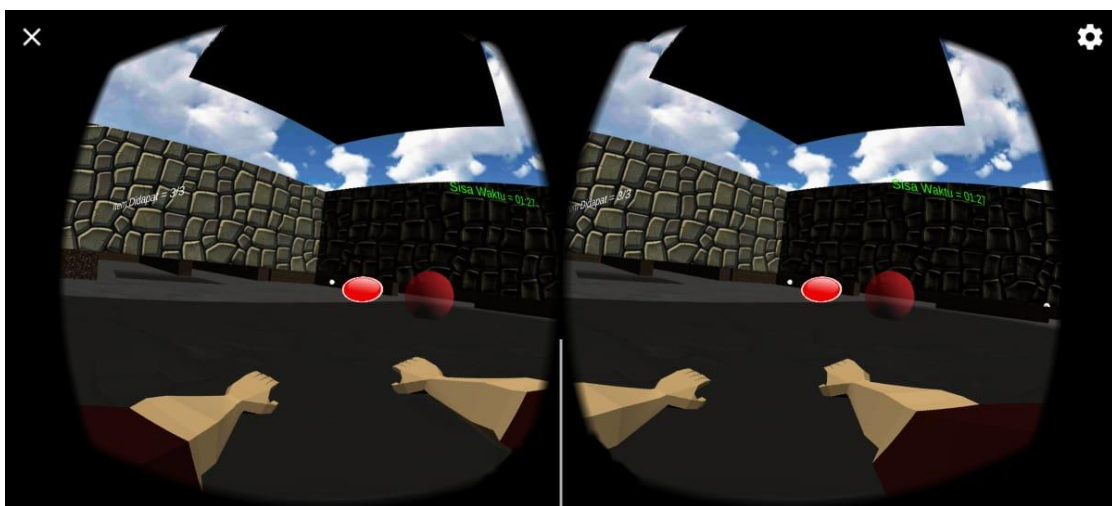
Gambar 4. 6 Tampilan Musuh Menangkap Pemain

Ketika pemain terkena sinar merah yang dipancarkan oleh musuh sebagai jarak tangkap dari musuh, maka pemain akan dikenakan hukuman pengurangan *item* pencarian dan akan di reset, pemain harus kembali lagi mengulang permainan dan mengumpulkan *item* di labirin sebelum waktu habis.



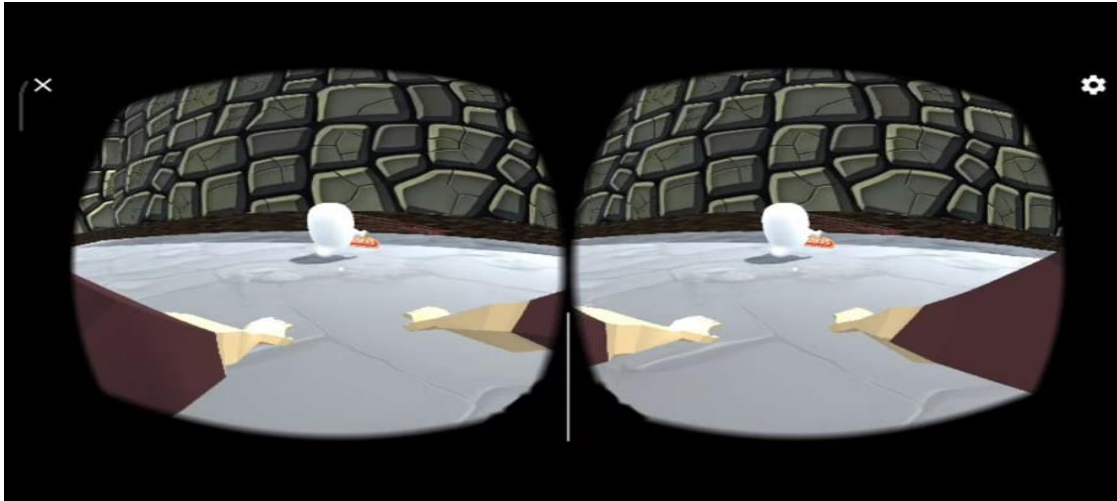
Gambar 4. 7 Tampilan Musuh Masuk Ruang Rahasia

Ketika pemain masuk kedalam ruangan rahasia di labirin dan menutup pintu maka musuh tidak bisa menemukan pemain dikarenakan semua rute menuju pemain tertutup, maka musuh akan menunggu pemain keluar dan melanjutkan pengejaran lagi, disini pemain bisa memanfaatkan waktu untuk mengingat kembali area-area yang belum ditelusuri untuk menemukan *item* misi.



Gambar 4. 8 Tampilan Pemain Berhasil Menyelesaikan Misi

Setelah mengumpulkan *item* yang dicari maka tombol untuk melanjutkan *stage* berikutnya akan muncul dan bisa segera menuju titik temu penjaga di tengah *map* untuk memencet tombol tersebut.



Gambar 4. 9 Tampilan Pemain Mengejar Musuh

Saat menjadi penjaga pemain akan melakukan hal yang berkebalikan dari sebelumnya, dimana musuh akan berlari ketika didekati dan mencoba mencari jalur menjauhi pemain, untuk memenangkannya pemain harus bisa menekan tombol di depan badan dari musuh.

Dalam mendvelop permainan ini dan tentunya juga menjalankannya di *smartphone* dibutuhkan spesifikasi yang memadai, dijelaskan dengan rinci sebagai berikut ini:

Spesifikasi device laptop pengerjaan:

- Windows 11 Pro
- 13th Gen Intel(R) Core(TM) i9-13900H 2.60 GHz
- RAM 16GB
- Storage 1TB
- 64-bit operating system, x64-based processor

Aplikasi yang digunakan:

- Unity 3D
- Blender 3D

Spesifikasi smartphone uji coba:

- Minimum Android 10 (Android Q)
- RAM 4 GB
- Storage 2GB

4.3. Tahapan Permainan

Dalam permainan Pencari yang dibuat akan dijalankan dengan beberapa tahapan permainan yaitu sebagai berikut :

1. Pemain memulai permainan dan akan tampil *stage* permainan.
2. Pemain mencari item yang tersebar di area bermain.
3. Jika tertangkap item akan direset dan waktu tetap berjalan.
4. Pemain harus menjawab kuis dengan benar untuk melanjutkan.
5. Jika berhasil mengumpulkan item sebelum waktu berakhir maka pemain langsung ke *stage* berikutnya.
6. Pada setiap *stage* musuh akan memiliki kecepatan yang meningkat untuk mengejar pemain sehingga setiap *stage* memiliki kesulitan meningkat.

Dalam permainan Penjaga yang dibuat akan dijalankan dengan beberapa tahapan permainan yaitu sebagai berikut :

1. Pemain memulai permainan dan akan tampil *stage* permainan.
2. Pemain mengejar musuh yang berlari.
3. Musuh memakai tombol di area badan harus ditekan.
4. Jika berhasil menekan maka langsung ke *stage* berikutnya.
5. Kecerdasan musuh dalam menghindari dilengkapi dengan AI dan *speed* yang berberda di tiap *stage*.

4.4. Pengujian Sistem

Blackbox testing merupakan suatu tes yang harus kita gunakan guna mengetahui apakah sistem kita sudah melakukan aksi sebagaimana diharapkan sesuai dengan fungsinya, untuk detail uji dapat kita perhatikan sesuai dengan komponen- komponen uji pada tabel 4.1.

Tabel 4.1 Tabel *Blackbox Test*

No	Hal Uji	Rincian
1	Tampilan Halaman Main Menu	Pengujian tampilan dan button
2	Tampilan Halaman Menjalankan Misi	Pengujian tampilan, VR, dan interaksi
3	Tampilan Halaman Kuis	Pengujian tampilan dan

1. Pengujian *Blackbox* Halaman Main Menu Berikut adalah hasil uji pada halaman MainMenu.

Tabel 4.2 Tabel Main Menu

No	Uji	Harapan uji	Hasil uji
1	Uji tampilan main menu	Menampilkan semua tombol	Berhasil
2	Uji button mulai	Memulai permainan berdasarkan role	Berhasil
3	Uji Panel Info	Menampilkan info pengembang	Berhasil
4	Uji button keluar	Keluar dari aplikasi	Berhasil

2. Pengujian *Blackbox* Halaman Menjalankan Misi

Tabel 4.3 Tabel Menjalankan Misi

No	Uji	Harapan uji	Hasil uji
1	Uji tampilan awal	Menampilkan semua tombol dan objek 3D	Berhasil
2	Uji interaksi	Berjalan menyusuri area bermain	Berhasil
3	Uji objek 3D	Menampilkan sirkuit dan objek 3D	Berhasil
4	Uji <i>Virtual Reality</i>	Teknologi berhasil diterapkan	Berhasil
5	Uji <i>Stage</i>	Menampilkan <i>Stage</i>	Berhasil

3. Pengujian *Blackbox* Halaman Kuis

Tabel 4.4 Tabel Kuis

No	Uji	Harapan uji	Hasil uji
1	Uji tampilan kuis	Menampilkan soal dan pilihan jawaban	Berhasil
2	Uji button jawaban	Memberikan hasil jawaban	Berhasil

Selanjutnya telah dilakukan pengujian pengalaman pengguna. Kita telah melakukan pengujian secara acak pada 20 orang dan melihat apakah respon mereka terhadap aplikasi yang sudah kita bangun. Kuisisioner telah memuat beberapa pertanyaan mengenai kinerja dan tampilan aplikasi yang kita buat.

Kuisisioner tersebut telah dijawab dengan menyatakan beberapa pilihan yaitu Setuju maupun tidak dengan skala yang sudah ditentukan pada tiap tiap pertanyaan. Kemudian hasil dari kuisisioner tersebut dihitung dengan mengaplikasikan perhitungan *Likert*, rumusnya yaitu :

$$P(S) = S/N \times 100\% \quad (2)$$

Keterangan :

P(S) : Persentase

S : Jumlah skor

N : Jumlah skor akumulasi

Berikut adalah hasil dari jawaban pertanyaan dihitung memakai skala *Likert* :

1. Tampilan dari aplikasi yang diuji sangat menarik.

Hasil uji kuis 1 dirincikan melalui tabel 4.5.

Tabel 4.5 Kuis Pertama

Pertanyaan	Pilihan	Nilai	Responden	Nilai X Responden
1	SS	4	13	52
	S	3	15	45
	TS	2	2	4
	STS	0	0	0
Penjumlahan			30	101

$$P(S) = S/N \times 100\%$$

$$P(S) = 101/120 \times 100\% = 84,167\%$$

Dari persentase diatas kita simpulkan 84,167% dari responden menyukai tampilan dari *Game* yang telah dibuat.

2. Objek 3D yang di desain sudah tampil dengan baik.

Hasil uji kuis 2 dirincikan melalui tabel 4.6.

Tabel 4.6 Kuis Kedua

Pertanyaan	Pilihan	Nilai	Responden	Nilai X Responden
2	SS	4	10	40
	S	3	17	51
	TS	2	3	6
	STS	0	0	0
Penjumlahan			30	97

$$P(S) = S/N \times 100\%$$

$$P(S) = 97/120 \times 100\% = 80,83\%$$

Dari persentase diatas kita simpulkan 80.83% dari responden menyukai desain objek 3D yang dibuat.

3. Permainan yang dimainkan mudah untuk dimengerti.

Hasil uji kuis 3 dirincikan melalui tabel 4.7.

Tabel 4.7 Kuis Ketiga

Pertanyaan	Pilihan	Nilai	Responden	Nilai X Responden
3	SS	4	15	60
	S	3	14	42
	TS	2	1	2
	STS	0	0	0
Penjumlahan			30	104

$$P(S) = S/N \times 100\%$$

$$P(S) = 104/120 \times 100\% = 86.67\%$$

Dari persentase diatas kita simpulkan 86.67% skala penilaian terhadap kemudahan permainan untuk dimengerti oleh responden.

4. Aplikasi telah memberikan pengetahuan baru pada pemain.

Hasil uji kuis 4 dirincikan melalui tabel 4.8.

Tabel 4.8 Kuis Keempat

Pertanyaan	Pilihan	Nilai	Responden	Nilai X Responden
4	SS	4	15	60
	S	3	15	45
	TS	2	0	0
	STS	0	0	0
Penjumlahan			30	105

$$P(S) = S/N \times 100\%$$

$$P(S) = \frac{105}{120} \times 100\% = 87,5\%$$

Dari persentase diatas kita simpulkan 87,5% dari responden menganggap mendapatkan pengetahuan baru.

5. Teknologi *Virtual Reality* sudah berjalan dengan baik.

Hasil uji kuis 5 dirincikan melalui tabel 4.9.

Tabel 4.9 Kuis Kelima

Pertanyaan	Pilihan	Nilai	Responden	Nilai X Responden
5	SS	4	18	72
	S	3	10	30
	TS	2	2	4
	STS	0	0	0
Penjumlahan			30	106

$$P(S) = S/N \times 100\%$$

$$P(S) = \frac{106}{120} \times 100\% = 88,33\%$$

Dari persentase diatas kita simpulkan 88,33% dari responden menganggap *Virtual Reality* berjalan dengan baik.

Untuk mendapatkan hasil akumulasi respon positif digunakanlah rumus sebagai berikut:

$$\bar{X} P(S) = \frac{P(S)_{tot}}{N} = \frac{(84,167 + 80,83 + 86,67 + 87,5 + 88,33)\%}{5} = 85,49\%$$

Dengan demikian hasil pengujian yang telah dilakukan pada pengguna mendapatkan akumulasi respon positif sebanyak 85,49% terhadap semua aspek-aspek pertanyaan mengenai aplikasi yang telah diujikan dan dibuat.

BAB 5 KESIMPULAN DAN SARAN

5.1. Kesimpulan

Adapun kesimpulan dari penelitian *Virtual Reality* petak umpet yang dibuat adalah sebagai berikut :

1. Berhasil menerapkan teknologi VR (*Virtual Reality*) pada *Game* petak umpet dan permainan menjadi lebih interaktif.
2. *Pathfinding* berhasil diterapkan dan musuh menjadi pintar dan bisa melakukan aksinya sendiri dengan baik.
3. Pada mode bermain berhasil menerapkan versi penjaga dan versi pencari.
4. Dari persentase uji kita simpulkan 87,5% dari responden menganggap mendapatkan pengetahuan baru.
5. Dari persentase uji kita simpulkan 86,67% dari responden menganggap permainan mudah dimengerti.
6. Dari persentase uji kita simpulkan 80,83% dari responden menyukai desain objek 3D yang dibuat.
7. Dari persentase uji kita simpulkan 84,167 % dari responden menyukai tampilan dari *Game* yang telah dibuat.
8. Dari persentase uji kita simpulkan 88,33% dari responden menganggap *Virtual Reality* berjalan dengan baik
9. Dari persentase uji disimpulkan akumulasi respon positif sebanyak 85,49% terhadap semua aspek-aspek pertanyaan mengenai aplikasi yang telah diujikan dan dibuat.

5.2. Saran

Permainan ini tentunya masih memiliki banyak kekurangan seperti :

1. Masih ada Bug pada mode penjaga dimana musuh kadang tersangkut pada *obstacle*
2. Aplikasi yang dibuat menggunakan labirin, bisa dikembangkan menjadi *real world* agar lebih nyata.

DAFTAR PUSTAKA

- Alam, A. (2022). "Pendekatan Pembelajaran Berbasis *Game* Digital untuk Transaksi Kurikulum yang Efektif untuk Pembelajaran-Pengajaran Kecerdasan Buatan dan Pembelajaran Mesin" . ICSCDS, Erode,India, pp. 69-74,
- Al-farizi, S. (2019). Penerapan *Artificial Intelligence* Pada Non Player Character Menggunakan Algoritma *Pathfinding* A* (A-Star) Dan Collision Detection Pada *Game* 2d “Underwater Shooter” Berbasis Android. Sarjana thesis, Universitas Siliwangi.
- E G. Agung, D. Eridani, and A. Fauzi, "Implementasi Metode Pathfinding dengan Algoritma A* pada Game Rogue-like menggunakan Unity," Jurnal Teknik Komputer, vol. 1, no. 3, pp. 81-89, Des. 2022. <https://doi.org/10.14710/jtk.v1i3.36700>
- Firdaus, M. H. (2021). Algoritma *Pathfinding* A* pada *Game* Edukasi Penyelamatan Diri dari VirusCovid-19 Berbasis Android (Skripsi). Sekolah Tinggi Manajemen Informatika dan Komputer. Palangkaraya.
- Faizal, I. M., Nurhasana, N., & Rahmawati, E. (2019). Digitalisasi Permainan Tradisional Galah Melalui Media *Game*. Sekolah Tinggi Teknologi Garut. INTEGRATED (Information Technology and Vocational Education) Volume 1 No. 1 | April 2019: 15-1.
- Ghofur, A. 2016. Implementasi Metode *Pathfinding* dengan Menggunakan Algoritma A* (A Bintang) dan Grid Graph pada Kecerdasan Buatan Non-Player Character dalam *Game* Action Puzzle.Thesis. Universitas Pendidikan Indonesia.
- Hasanah, U. 2019. Rancang Bangun *Game* Tradisional Berbasis Android. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Alauddin Makassar.

- Handayani, F. F. & Munastiwi, E. 2022. Implementasi Permainan Tradisional di Era Digital dan Integrasinya dalam Pendidikan Anak Usia Dini. *Jurnal Pendidikan Islam Anak Usia Dini*. Volume 5 Nomor 2.
- Izza, F. (2020). Pengembangan Permainan Tradisional Dam-Daman dalam Bentuk Application Programming Interface. Yogyakarta: Universitas Islam Indonesia.
- Phillips, L. (2017). 3D Modeling for Beginners: Learn everything you need to know about 3D Modeling!. CreateSpace Independent Publishing Platform.
- Sunarya, P. A. (2022). Machine learning and *Artificial Intelligence* as educational Games. University of Raharja, Tangerang, Indonesia. *International Transactions on Artificial Intelligence (ITALIC)* Vol. 1 No. 1, November 2022, pp. 129~138
- Tjiptoherijanto, P. wantipres.go.id. 2016, 21 Desember. Petak Umpet.
- Bown, J., White, E., & Boopalan, A. (2017). Boundaries of Self and Reality Online. *12*, pp. 239–259. Cambridge, MA, USA: Academic Press. Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/B9780128041574000128?via%3Dihub>
- Brooks Jr., F. P. (1999). What's Real About. *IEEE Computer Graphics and Application*.
- Cui, X., & Shi, H. (2012). "An Overview of *Pathfinding* in Navigation Mesh". *IJCSNS International Journal of Computer Science and Network Security*.
- Harley, D. (2019). Palmer Luckey and the rise of contemporary *Virtual Reality*. *Convergence*, 26(5-6), 1144–1158. doi:10.1177/1354856519860237
- Rahouti, A., Lovreglio, R., Datoussaïd, S., & Descamps, T. (2021). Prototyping and Validating a Non-immersive *Virtual Reality* Serious Game for Healthcare Fire Safety Training. *Fire Technol*, 3041–3078. doi:10.1007/s10694-021-01098-x
- Reitmann, S., Neumann, L., & Jung, B. (2021). BLAINDER—A Blender AI Add-On for Generation of Semantically Labeled Depth-Sensing Data. (S. L. Hegarat-Masclé, Ed.) *Sensors (Basel)*, 21(6), 2144. doi:10.3390/s21062144

- Riva G., B. C.-P. (2015). Immersed in Media. *Presence-Inducing Media for Mental Health Applications*, pp. 283–332.
- Slater, M., & Sanchez-Vives, M. V. (2014). Transcending the Self in Immersive *Virtual Reality*. *Computer*, 24-30. doi:10.1109/MC.2014.198
- Sutherland, I. E. (2001). The Ultimate Display. 65(2), pp. 506-508. New York: IFIP.
Retrieved from
https://www.researchgate.net/publication/30875394_The_Ultimate_Display
- Wheeler, G., Deng, S., Toussaint, N., Pushparajah, K., Schnabel, J. A., Simpson, J. M., & Gomez, A. (2018). Unity, Virtual interaction and visualisation of 3D medical imaging data with VTK and. *Healthc Technol Lett*, 5(5), 148–153. doi:10.1049/htl.2018.5064
- Wohlgenannt, I., Simons, A., & Stieglitz, S. (2020). *Virtual Reality*. *Bus Inf Syst Eng* 62, 455–461. doi:10.1007/s12599-020-00658-9
- Zega, V. S. (2018). *Pathfinding Menggunakan Algoritma Dijkstra pada Game 'Aviar' Multiplayer Berbasis Virtual Reality*. Surabaya.