

BAB II

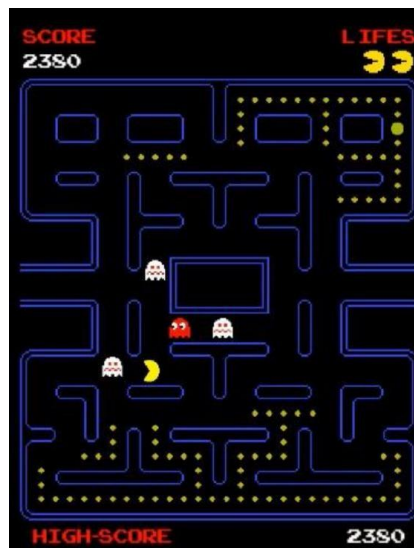
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu memuat penelitian sebelumnya yang serupa atau sejenis, dengan tujuan untuk membedakan penelitian yang sedang dilakukan dengan studi sebelumnya, serta meningkatkan atau menyempurnakan hasil-hasil sebelumnya. Hal ini bertujuan untuk menemukan inovasi atau perbaikan dalam bidang penelitian yang sedang dilakukan.

2.1.1 Penelitian Badri & Muhammad, 2020

Penelitian yang dilakukan oleh Fawaidul Badri dan Muhammad Farih Al Habib pada tahun 2020 adalah tentang bagaimana algoritma A* diterapkan pada NPC (*Non-Playable Character*) game. Karakter NPC yang digunakan pada penelitian Badri terdapat pada karakter *ghost* atau hantu pada permainan Pacman. Berikut adalah *game* yang telah dikembangkan dengan penerapan *Pathfinding* untuk mengatur pergerakan pada NPC.



Gambar 2.1 *Game* Penelitian Badri & Muhammad, 2020

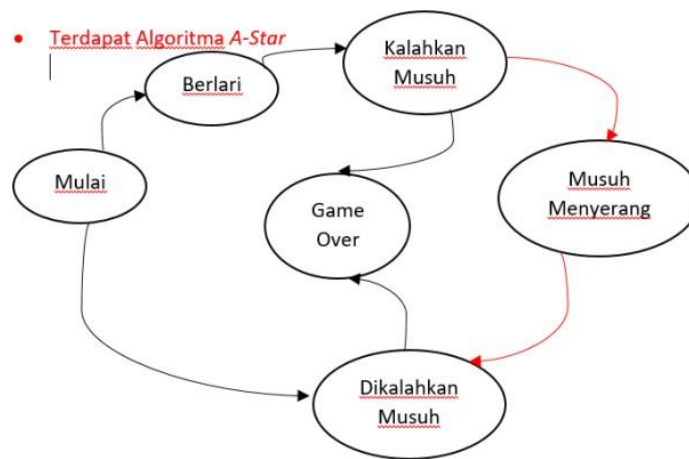
Penelitian Badri & Muhammad merupakan sebuah penelitian yang bertujuan untuk menganalisis penggunaan algoritma A* untuk menemukan jalur agar permainan terasa lebih stabil. Berikut hasil dari penelitian Fawaidul Badri dan Muhammad Farih Al Habib (Badri & Muhammad, 2020).

Tabel 2.1 Penelitian Badri & Muhammad, 2020

Judul Penelitian	
Implementasi Algoritma A* (A Star) Pada NPC (<i>Non-Playable Character</i>) Game Pacman Menggunakan <i>Game Engine Unity 5</i> Berbasis Android	
Tujuan	Ruang Lingkup
Implementasi algoritma A* agar <i>ghost</i> dapat menemukan jalur tercepat ke Pacman agar permainan terasa lebih intens.	Penelitian ini melakukan penerapan algoritma A* pada NPC <i>Ghost</i> sehingga memungkinkan NPC kejar-kejaran dengan <i>Player</i> .
Hasil Penelitian	
Dari hasil uji akurasi didapatkan rata-rata nilai 91,5%. Dari sini dapat disimpulkan bahwa implementasi algoritma A* berhasil 91,5%, dan masih memerlukan improvisasi agar akurasi mencapai nilai yang lebih tinggi.	
Perbedaan dengan Penelitian yang Dilakukan	
<ol style="list-style-type: none"> 1. Pada penelitian ini dilakukan dengan menggunakan <i>game engine Unity</i>. 2. Pada penelitian ini mengembangkan kondisi pergerakan NPC dengan efek <i>game over</i> jika bersentuhan dengan <i>Player</i> dengan akurasi 91,5%. 	

2.1.2 Penelitian Mutaqin et al, 2021

Penelitian yang dilakukan oleh Ghani Mutaqin, Juniardi Nur Fadilah dan Fresy Nugroho pada tahun 2021 adalah tentang bagaimana algoritma A* diterapkan untuk pencarian jalur terpendek pada *game*. Penelitian ini bertujuan untuk menganalisis penggunaan algoritma A* untuk membuat pergerakan NPC (*Non-Playable Character*) menjadi lebih cerdas.



Gambar 2.2 Alur Algoritma Pada Penelitian Mutaqin et al, 2021

Pada penelitian dari Mutaqin *Pathfinding* digunakan pada karakter musuh untuk melakukan aksi menyerang dan mengalahkan karakter pemain. Berikut hasil dari penelitian Ghani Mutaqin, Juniardi Nur Fadilah dan Fresy Nugroho (Mutaqin et al., 2021).

Tabel 2.2 Penelitian Mutaqin et al, 2021

Judul Penelitian	
Implementasi Metode <i>Path Finding</i> dengan Penerapan Algoritma A-Star untuk Mencari Jalur Terpendek pada <i>Game</i> “ <i>Jumrah Launch Story</i> ”	
Tujuan	Ruang Lingkup
Implementasi algoritma A* untuk membuat pergerakan NPC menjadi lebih cerdas dalam halang rintang.	Penelitian ini melakukan penerapan algoritma A* pada NPC. Sehingga NPC untuk menemukan posisi <i>Player</i> .
Hasil Penelitian	
Dari hasil NPC yang menggunakan algoritma ini dapat mengejar <i>player</i> dan membunuh saat <i>Player</i> kurang berhati-hati. NPC juga bisa mengejar walaupun <i>Player</i> bersembunyi dan terdapat penghalang. Sedangkan NPC tanpa algoritma A-Star hanya dapat bergerak sesuai dengan program yang telah dibuat di awal.	
Perbedaan dengan Penelitian yang Dilakukan	
Pada penelitian ini mengembangkan kondisi pergerakan dengan tidak memberikan kesempatan banyak pada <i>Player</i> untuk memahami peta pada <i>game</i> dengan akurasi 89,5%.	

2.1.3 Penelitian Elvitro et al. 2022

Penelitian yang dilakukan oleh Elvitro Gumelar Agung, Dania Eridani dan Adnan Fauzi pada tahun 2022 adalah tentang bagaimana algoritma A* diterapkan pada NPC (*Non-Playable Character*) game. Penelitian ini merupakan sebuah penelitian yang bertujuan untuk menganalisis penggunaan algoritma A* untuk otomatisasi pergerakan NPC (*Non-Playable Character*).



Gambar 2.3 *Game* Penelitian Elvitro et al, 2022

Gambar diatas merupakan tampilan ketika dilakukan pengujian dengan dipasangkannya algoritma A* dan di uji apakah jalur yang dibuat sudah sesuai dan mengikuti jalur yang benar sampai ke titik akhir tanpa perhentian. Berikut hasil dari penelitian Elvitro Gumelar Agung, Dania Eridani dan Adnan Fauzi (Elvitro et al, 2022).

Tabel 2.3 Penelitian Elvitro et al, 2022

Judul Penelitian	
Implementasi Metode <i>Pathfinding</i> dengan Algoritma A* pada <i>Game Rogue-like</i> menggunakan <i>Unity</i>	
Tujuan	Ruang Lingkup
Implementasi algoritma A* untuk pencarian jalur otomatis bagi suatu unit pada <i>game</i> . Pencarian jalur terpendek.	Penelitian ini melakukan penerapan algoritma A* pada <i>enemy</i> sehingga butuh serang menyerang untuk menjalankan <i>game</i> .

Hasil Penelitian
<ol style="list-style-type: none"> 1. Cara kerja algoritma A* pada <i>game</i> adalah dengan menggunakan petak yang diberi nilai berdasarkan posisi dari titik awal dan titik akhir. Petak dengan nilai paling kecil akan digunakan sebagai jalur yang dilewati. 2. Metode A* <i>Pathfinding</i> dapat digunakan dalam <i>game</i> dengan kondisi dinamis atau berubah-ubah, karena data akan diperbarui saat <i>game</i> dimainkan.
Perbedaan dengan Penelitian yang Dilakukan
<ol style="list-style-type: none"> 1. Pada penelitian ini dilakukan dengan menggunakan <i>game engine Unity</i>. 2. Pada penelitian ini mengembangkan <i>game</i> dengan genre RPG sehingga peta yang digunakan sangat luas.

2.1.4 Penelitian Octavian & Hermawan, 2023

Penelitian yang dilakukan oleh Felix Octavian dan Latus Hermawan pada tahun 2023 adalah tentang bagaimana algoritma A* diterapkan pada NPC (*Non-Playable Character*) *game*. Penelitian ini merupakan sebuah penelitian yang bertujuan untuk menganalisis penggunaan algoritma A*.



Gambar 2.4 *Game* Penelitian Octavian & Hermawan, 2023

Pada penelitian dari Octavian, menggunakan *game* dengan genre *roguelike* dengan judul *Dual Legacy*. *Pathfinding* diterapkan kepada NPC atau musuh yang ada pada *game* untuk dapat berinteraksi dengan karakter pemain. Berikut hasil dari penelitian Fawaidul Badri dan Muhammad Farih Al Habib (Octavian & Hermawan, 2023).

Tabel 2.4 Penelitian Octavian & Hermawan, 2023

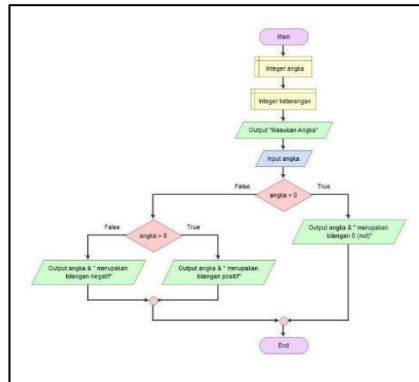
Judul Penelitian	
Penerapan Algoritma <i>Pathfinding</i> A* dalam <i>Game Dual Legacy</i> berbasis Android	
Tujuan	Ruang Lingkup
Dengan penerapan algoritma A*, NPC dapat mencari dan mengejar pemain/ <i>player</i> melalui jalan terdekat.	NPC hanya menyerang ketika berada dalam radius yang dekat saja.
Hasil Penelitian	
Algoritma A* berhasil diimplementasikan pada NPC musuh untuk mendekati <i>player</i> apabila <i>player</i> masuk ke jangkauan perhitungan musuh dan NPC musuh mendekati <i>player</i> melalui jarak terdekat dengan menghindari halangan/ <i>obstacle</i> yang ada.	
Perbedaan dengan Penelitian yang Dilakukan	
Pada penelitian ini berfokus pada halang rintang yang mengadaptasi genre <i>game Roguelike</i>	

2.2 Teori Terkait

Teori memiliki peran penting dalam penelitian ilmiah. Teori-teori yang relevan dengan topik penelitian perlu dijelaskan. Peran teori adalah sebagai dasar untuk memecahkan masalah penelitian.

2.2.1 Algoritma

Algoritma adalah urutan langkah-langkah terstruktur yang digunakan untuk menyelesaikan masalah atau mencapai tujuan tertentu. Algoritma sering digunakan dalam komputer dan ilmu komputer untuk menyelesaikan tugas-tugas seperti pengurutan data, pencarian informasi, atau pengambilan keputusan.



Gambar 2.5 Contoh Algoritma

Algoritma menjadi dasar penting dalam pengembangan *game* karena dari algoritma yang diterapkan akan menghasilkan pergerakan karakter yang memungkinkan untuk membangun *game* yang menarik (Khesya, 2022).

2.2.1.1 Pathfinding

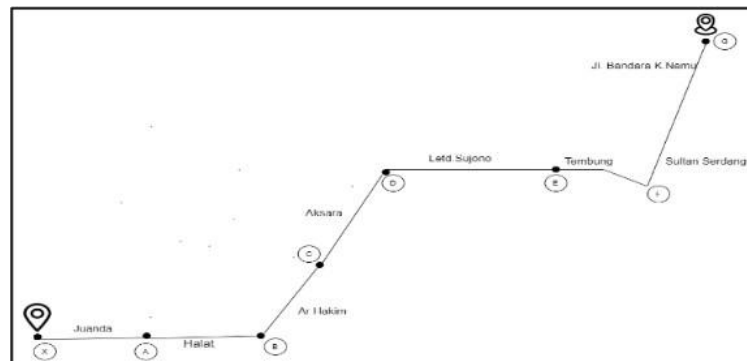
Pathfinding adalah proses mencari jalur terpendek atau optimal dari satu titik ke titik lain. Ini adalah masalah klasik dalam ilmu komputer dan sering digunakan dalam bidang seperti *game*, navigasi, dan robotika.

Gambar 2.6 Pencarian Jalur *Pathfinding*

Dalam pengembangan *game*, metode *Pathfinding* digunakan untuk membuat kondisi karakter NPC mendeteksi jalur untuk menuju target. Sehingga dari sistem yang dimiliki oleh metode *Pathfinding* ini dapat membuat pergerakan semacam patroli untuk mencari jalur target (Sugianto & Gunawan, 2021).

2.2.1.2 Algoritma A*

Algoritma A* adalah algoritma pencarian jalur yang menggabungkan pencarian heuristik dan pencarian terarah. Dengan menggunakan fungsi heuristik untuk memperkirakan jarak dari titik saat ini ke titik tujuan, A* dapat menemukan jalur terpendek atau optimal dengan efisien.

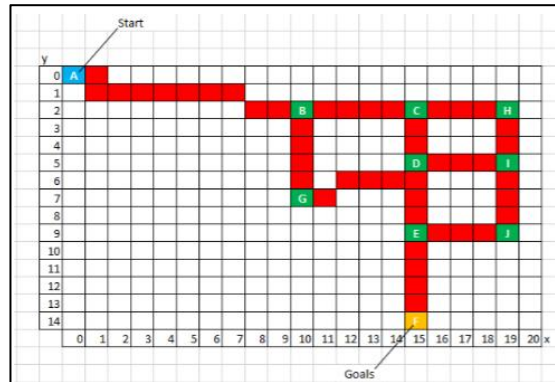


Gambar 2.7 Graf *Node* yang akan Dilalui

Algoritma A* adalah bagian dari metode *Pathfinding* yang memungkinkan pencarian atau deteksi jalur yang dilakukan mengungkap dari konsep algoritma A* yang mempertimbangkan efisiensi dari jalur yang diambil sehingga pergerakan yang dilakukan lebih optimal (Luthfita et al., 2022). Rumus utama yang digunakan dalam algoritma A* adalah $f(n) = g(n) + h(n)$, di mana $f(n)$ adalah total jarak dari *node* awal ke *node* tujuan melalui *node* (n), $g(n)$ adalah jarak sebenarnya dari *node* awal ke *node* (n), dan $h(n)$ adalah estimasi jarak terpendek dari *node* (n) ke *node* tujuan. Kombinasi dari jarak aktual dan estimasi ini memungkinkan A* untuk secara efisien menemukan jalur terpendek dengan memprioritaskan *node* yang menjanjikan berdasarkan total jarak terendah.

2.2.1.3 Heuristik

Heuristik adalah aturan praktis atau strategi pendekatan yang digunakan untuk memecahkan masalah atau membuat keputusan. Dalam pencarian jalur, fungsi heuristik memberikan perkiraan jarak dari titik saat ini ke titik tujuan, membantu algoritma seperti A* untuk memilih jalur yang paling kecil jaraknya.

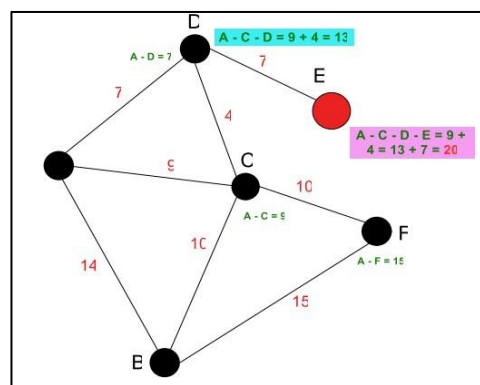


Gambar 2.8 Pengecekan Jalur Heuristik

Heuristik sendiri memiliki sebuah konsep dengan perhitungan Jarak dari setiap *node* ke *node* dapat dioptimalkan, sehingga dari hal tersebut dapat menghasilkan perhitungan yang optimal dengan waktu yang lebih efisien (Fernando et al., 2020).

2.2.1.4 Node

Dalam algoritma dan struktur data, *node* adalah titik individu dalam sebuah graf atau struktur data yang mewakili entitas atau posisi tertentu. Dalam pencarian jalur, *node* mewakili posisi dalam ruang dimana algoritma mencari jalur terpendek atau optimal.

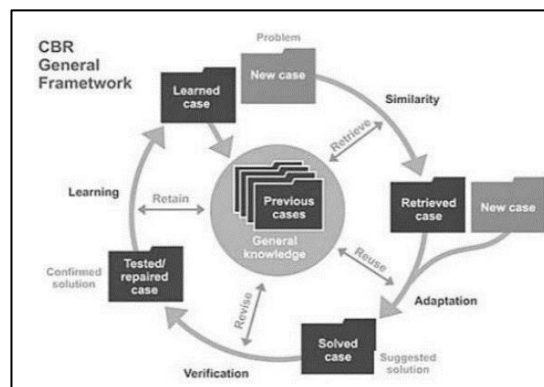


Gambar 2.9 Graf Pada Node

Sehingga *node* sendiri dapat disebut sebagai titik-titik tujuan terdekat yang dapat menghubungkan titik satu ke titik lainnya. Dari beberapa *node* yang ada nantinya akan menjadi suatu jalur atau alur yang akan digunakan untuk memulai heuristik atau algoritma A* (Fuad, 2020).

2.2.1.5 Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence* atau AI) merupakan cabang ilmu komputer yang bertujuan mengembangkan sistem yang mampu menjalankan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Ini melibatkan berbagai teknik seperti *machine learning*, logika, dan pengolahan bahasa alami, serta telah diterapkan dalam berbagai aplikasi seperti pengenalan wajah, kendaraan otonom, dan sistem rekomendasi.

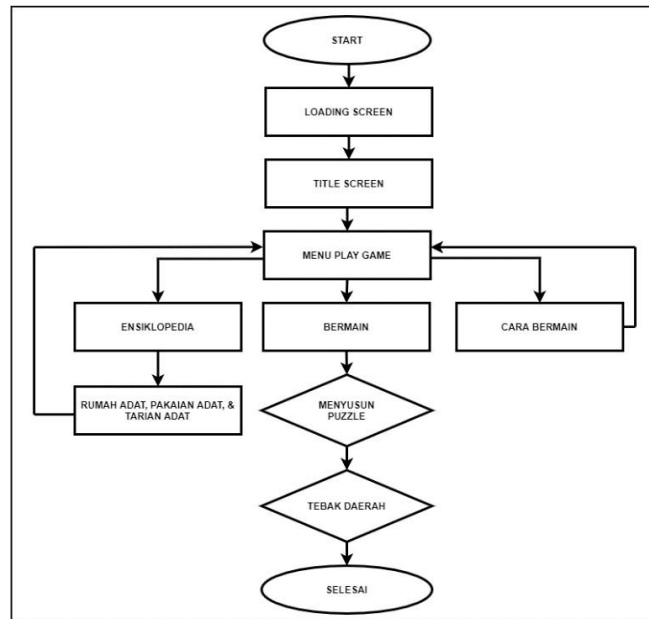


Gambar 2.10 Siklus Case Based Reasoning

Dalam pencarian jalur, AI sering digunakan untuk meningkatkan efisiensi dan kecerdasan algoritma pencarian seperti A*. Algoritma A* adalah salah satu algoritma yang digunakan karena memiliki sistem kecerdasan buatan yang memungkinkan penerapan perilaku pergerakan yang diterapkan seperti halnya yang manusia lakukan. Seperti berpatroli, dan mencari jalur terpendek, sehingga lebih menarik pergerakan dalam *game* (Devianto & Dwiasnati, 2020).

2.2.2 Game

Game adalah aktivitas yang melibatkan interaksi dari satu atau lebih pemain dengan tujuan tertentu, seperti hiburan, pendidikan, atau simulasi.

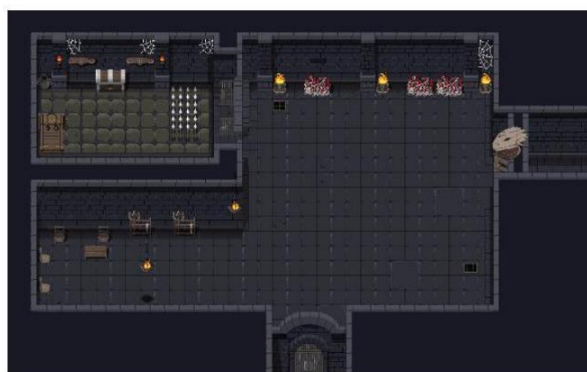


Gambar 2.11 Alur *Gameplay*

Game dapat beragam dari yang sederhana seperti permainan papan hingga yang kompleks seperti permainan video dengan grafis yang canggih dan mekanika permainan yang rumit. Bisa dikatakan bahwa *game* adalah suatu produk yang digunakan untuk membuat interaksi antara pengguna dengan sistem untuk tujuan tertentu (Ahmad et al., 2022).

2.2.2.1 *Game* Aset

Game aset adalah semua elemen yang digunakan dalam pembuatan *game*, termasuk grafik, suara, animasi, kode, dan lainnya. *Game* aset dapat mencakup karakter, lingkungan, senjata, efek suara, dan banyak lagi.



Gambar 2.12 *Mockup Preview* Aset

Komponen tersebut berkontribusi pada pengalaman bermain *game* dan bisa menjadi kunci keberhasilan sebuah *game*. *Game* asset ini juga dapat membantu pemain mendapatkan emosi saat bermain *game* sehingga permainan lebih menarik dan terasa lebih hidup (Unggul & Persada, 2023).

2.2.2.2 *Gameplay*

Gameplay mengacu pada cara bermain sebuah *game*, termasuk aturan, interaksi, tantangan, dan respons yang dialami oleh pemain saat bermain.



Gambar 2.13 Tampilan *Gameplay*

Gameplay memengaruhi pengalaman keseluruhan pemain dan bisa mencakup segala hal mulai dari kontrol karakter hingga strategi yang digunakan dalam memecahkan masalah atau mencapai tujuan (Janata et al., 2022).

2.2.2.3 RPG

RPG (*Role-Playing Game*) adalah genre *game* di mana pemain mengambil peran karakter dan mengendalikannya dalam sebuah *game*. Dalam RPG, pemain sering memiliki kontrol atas perkembangan karakter mereka, termasuk atribut, keterampilan, dan pilihan moral.



Gambar 2.14 Rancangan Karakter dan *Sprite*

Permainan semacam ini sering menekankan pada narasi yang dalam, eksplorasi dunia yang luas, dan interaksi sosial dengan karakter dalam *game*. Sehingga dalam *game* dengan genre RPG, pemain memungkinkan untuk bergerak secara bebas di dalam *game* (Sri & Andiyoko, 2019).

2.2.2.4 Map

Dalam *game*, map mengacu pada representasi *visual* dari lingkungan atau dunia permainan. Hal itu bisa berupa peta yang menampilkan wilayah geografis dalam *game* atau tata letak yang lebih abstrak dari elemen-elemen seperti bangunan, rintangan, dan fitur lainnya.



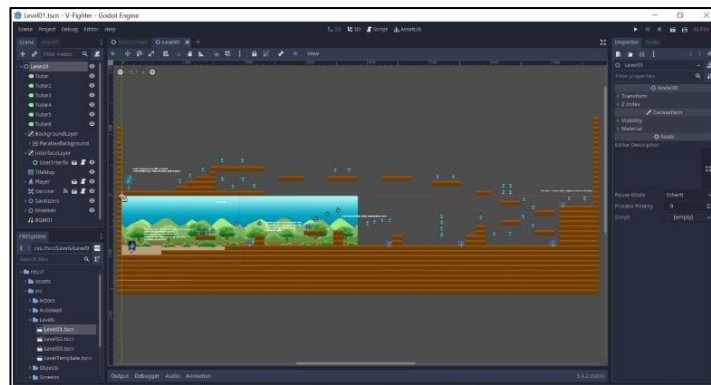
Gambar 2.15 *Sample Map*

Map membantu pemain berorientasi dalam *game* dan memberikan konteks untuk kegiatan dalam permainan. Dari map yang ada dalam *game*, dapat membuat pemain

masuk ke dalam lingkungan dunia pemain yang seolah-olah representasi dari dunia nyata (Snodgrass & Ontanon, 2021).

2.2.2.5 Tilemap

Tilemap adalah teknik dalam pengembangan *game* di mana peta atau *level game* dibangun dari serangkaian kecil gambar yang disebut *tiles* atau ubin. *Tilemap* memungkinkan pengembang untuk membuat peta dengan mudah dengan mengatur dan menggabungkan ubin-ubin ini sesuai kebutuhan.



Gambar 2.16 *Tilemap Game*

Penggunaan *tilemap* adalah pendekatan yang umum digunakan dalam *game* 2D untuk membuat lingkungan yang luas dan beragam dengan efisien. Sehingga dari satu *design tilemap* bisa menjadi map yang luas dengan kemudahan dalam melakukan perubahan dalam desain susunan map (Metayani & Tan, 2023).

2.2.2.6 Level

Dalam *game*, *level* adalah bagian atau tahap tertentu dari permainan yang terdiri dari pengaturan lingkungan, tantangan, dan objektif tertentu. Setiap *level* biasanya memiliki tingkat kesulitan yang berbeda dan mungkin memerlukan pemain untuk mencapai tujuan tertentu, seperti menyelesaikan *puzzle*, mengalahkan musuh, atau mencapai titik akhir.



Gambar 2.17 Penggunaan *Level* Pada *Game*

Sehingga dengan adanya *level* ini dapat menjadi tantangan tersendiri bagi pemain untuk mencapai perangkat tertentu dalam sebuah *game* (Yulianto & Devi, 2020).

2.2.2.7 Karakter

Karakter dalam *game* merujuk kepada entitas yang bisa dikendalikan oleh pemain atau digunakan sebagai bagian dari cerita atau pengalaman permainan.



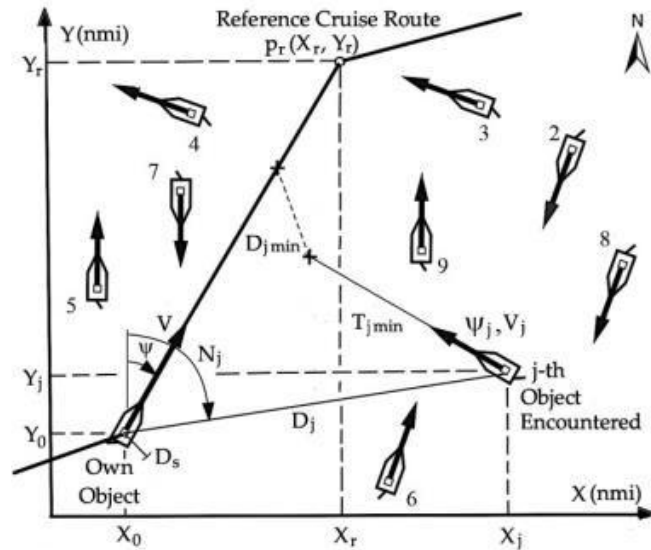
Gambar 2.18 Karakter Pada *Game*

Karakter bisa berupa manusia, makhluk fiksi, kendaraan, atau objek lainnya, dan mereka memiliki atribut dan kemampuan yang berbeda-beda. Karakter sering memiliki peran penting dalam memajukan cerita atau mencapai tujuan dalam *game* (Reno & Surahman, 2020).

2.2.2.8 Behavior

Behavior dalam *game* mengacu pada cara karakter, musuh, atau objek lainnya berperilaku atau berinteraksi dengan lingkungan dan pemain. *Behavior* bisa diprogram secara langsung oleh pengembang atau diatur oleh algoritma kecerdasan

buatan untuk memberikan respons yang realistis terhadap tindakan pemain atau perubahan dalam *game*.



Gambar 2.19 Tampilan Saat Memiliki Banyak *Object*

Perilaku yang baik dapat meningkatkan pengalaman bermain *game* dengan membuat interaksi antara elemen-elemen dalam *game* lebih dinamis dan menarik (Lisowski, 2020).

2.2.2.9 NPC

NPC (*Non-Player Character*) adalah karakter dalam sebuah permainan yang dikendalikan oleh komputer atau sistem *game*, bukan oleh pemain. Mereka sering berperan sebagai karakter dalam cerita, penjual, musuh, atau pemberi misi.

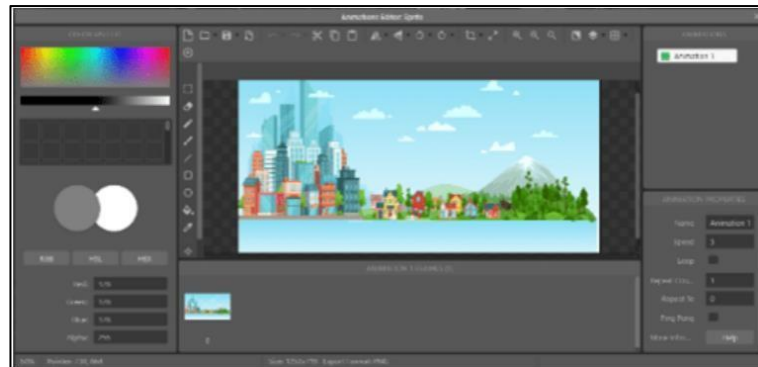
								15.7	
								NAGA 14.6	
		7.6 ↑	8.6 ↑	9.6 ↑	10.6 ↑	11.6 ↑	12.6 ↑	13.6 ↑	15.6 →
	PRAJURIT 6.5	7.5 ←	8.5 ←	9.5 ←	10.5 ←	11.5 ←	12.5 ←	13.5 ←	14.5 ←
		7.4 ↓	8.4 ↓	9.4 ↓	10.4 ↓	11.4 ↓	12.4 ↓	13.4 ↓	

Gambar 2.20 Interaksi Dengan NPC

NPC dapat menambah kedalaman dan kompleksitas dunia *game* dengan memberikan interaksi yang beragam kepada pemain. Dalam *game* NPC menjadi penting agar dari karakter pemain dapat memiliki interaksi dengan karakter lain dalam sebuah *game* (Junanto et al., 2020).

2.2.3 Construct 3

Construct 3 adalah salah satu dari banyak perangkat lunak pembuat *game* yang memungkinkan pengguna untuk membuat *game* tanpa perlu menulis kode secara langsung yang menggunakan antarmuka grafis dan sistem perangkat lunak *drag-and-drop* untuk memungkinkan pengguna membuat *game* 2D dengan cepat dan mudah. *Construct 3* adalah editor *game* 2D berbasis HTML5 yang dikembangkan oleh Scirra Ltd.

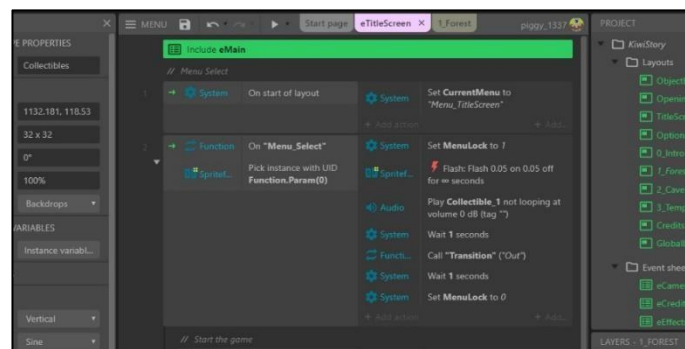


Gambar 2.21 Penggunaan *Construct*

Platform ini dirancang dengan fokus pada kemudahan penggunaan, memastikan bahwa para pengguna, terutama pengembang *game* pemula, dapat dengan cepat mempelajari dan mengoperasikan alat tersebut. Kelebihan utama *Construct 3* adalah kemampuannya untuk berjalan melalui peramban web, memungkinkan akses dan penggunaan yang fleksibel di berbagai perangkat dengan koneksi internet. Selain itu, salah satu fitur pentingnya adalah kemampuan untuk menyimpan proyek *game* baik secara daring (*online*), memungkinkan kolaborasi tim secara mudah, maupun secara luring (*offline*), memberikan kemungkinan bagi pengembang untuk bekerja tanpa ketergantungan pada koneksi internet yang stabil (Permatasari et al., 2022).

2.2.3.1 Event sheet

Dalam *Construct 3*, *Event sheet* adalah bagian dari antarmuka pengguna yang digunakan untuk membuat logika permainan atau skrip dalam *game*. Ini berfungsi sebagai daftar peristiwa yang terjadi dalam *game*, seperti *input* pengguna, tabrakan objek, atau waktu yang berakhir, dan memungkinkan pengguna untuk menentukan tindakan atau respons yang harus diambil oleh *game* dalam situasi tersebut.

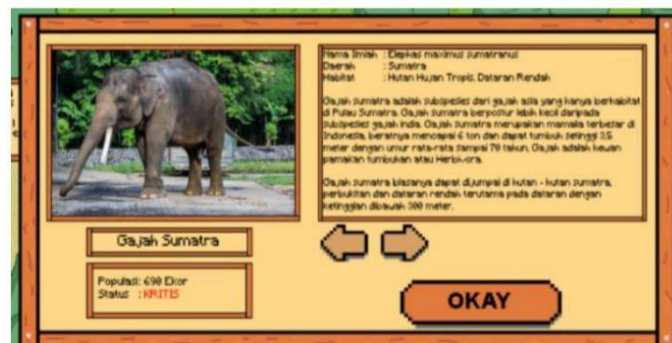


Gambar 2.22 Penyusunan *Event sheet*

Secara sederhananya *event sheet* adalah kondisi yang dibuat untuk membuat skenario dalam pergerakan *game* yang dikembangkan (Humam & Fawwaz, 2023).

2.2.4 Mobile

Dalam *game*, tampilan dapat berupa versi *mobile* atau yang bisa dimainkan di ponsel. Mengacu pada perangkat bergerak seperti *smartphone* atau tablet, dan juga ke *game* yang dirancang khusus untuk dijalankan pada perangkat-perangkat tersebut.

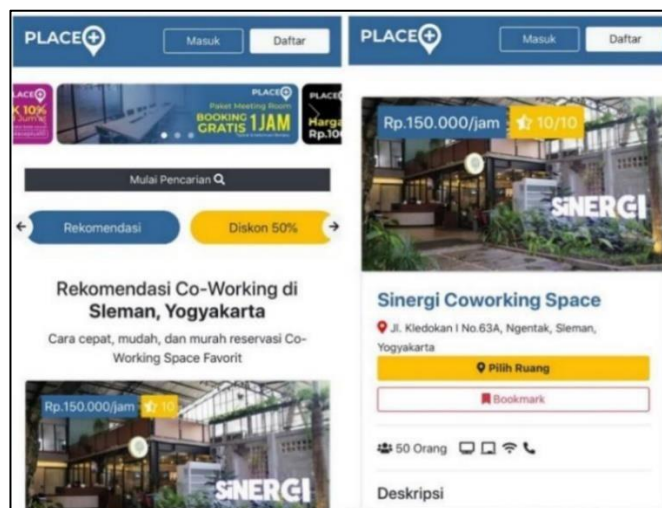


Gambar 2.23 Tampilan *Mobile Menu* Penjelasan Hewan

Game mobile sering menawarkan pengalaman bermain *game* yang portabel dan dapat dimainkan di mana saja, dan mereka sering memanfaatkan fitur-fitur khusus dari perangkat-perangkat tersebut, seperti layar sentuh atau sensor gerak. Sehingga dalam penggunaannya *game* mudah dalam penggunaan karena ada banyak fitur seperti sensor gerak yang tidak ada di tampilan desktop (Janata et al., 2022).

2.2.4.1 *User Interface*

User Interface adalah segala sesuatu yang memungkinkan interaksi antara pengguna dan sistem. Dalam *game*, ini mencakup elemen-elemen seperti *menu*, tombol, ikon, teks, dan tata letak *visual* lainnya yang memungkinkan pemain untuk berinteraksi dengan permainan.

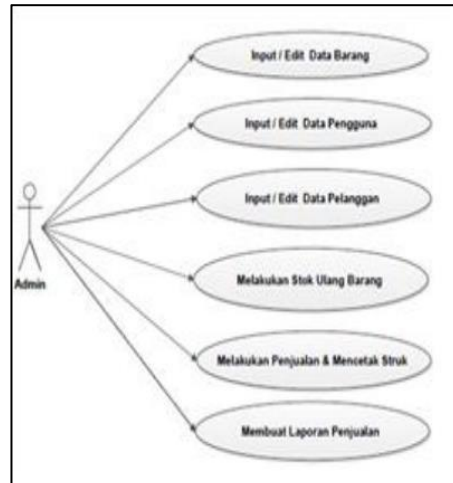


Gambar 2.24 Tampilan *User Interface*

Tujuannya adalah untuk membuat pengalaman pengguna sebaik mungkin dengan memudahkan navigasi, memberikan umpan balik yang jelas, dan memfasilitasi akses ke fitur dan informasi penting (Multazam et al., 2020).

2.2.4.2 *Prototype*

Prototype adalah versi awal atau model dari sebuah produk atau proyek yang dibuat untuk menunjukkan konsep atau fitur utama. Dalam pengembangan *game*, *prototype* sering digunakan untuk menguji mekanika permainan, tata letak *level*, atau gagasan desain lainnya sebelum mengembangkan versi final permainan.



Gambar 2.25 Use Case Admin

Prototype dapat berupa versi sederhana yang dibuat dengan cepat untuk mendapatkan umpan balik segera dari pengguna atau pihak terkait (Hermanda, 2020).

2.2.4.3 Wireframe

Wireframe adalah representasi *visual* yang sederhana dan skematis dari antarmuka pengguna atau halaman web, yang menunjukkan tata letak elemen-elemen utama tanpa detail grafis atau desain *visual*.

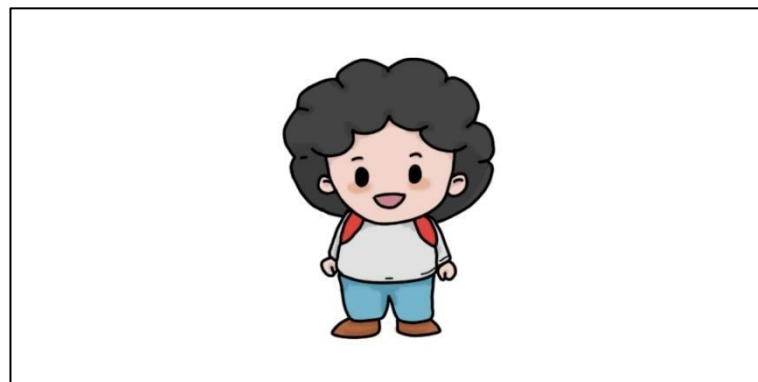


Gambar 2.26 Rancangan Wireframe

Wireframe membantu pengembang atau desainer untuk merencanakan struktur dan tata letak suatu produk dengan fokus pada fungsi dan hierarki informasi, tanpa terganggu oleh desain grafis atau estetika (Muhammad, 2022).

2.2.5 Game “*The Fabrique*”

Game “*The Fabrique*” adalah game yang dikembangkan oleh peneliti yang akan digunakan untuk mengevaluasi penggunaan metode *Pathfinding A** pada pergerakan karakter NPC (*Non-Player Character*). Game “*The Fabrique*” menceritakan cerita seorang pemuda bernama Brow Lee yang baru saja menyelesaikan pendidikannya. Kemudian Brow Lee memulai kariernya sebagai seorang karyawan di ibukota, namun bahaya sedang memantaunya. Ada preman ibukota yang cukup mengganggu dan ada pula persaingan dengan karyawan lama. Dalam game “*The Fabrique*” Brow Lee sebagai karakter utama perlu untuk mencapai bangunan target untuk menang tanpa tertangkap oleh karakter NPC.



Gambar 2.27 Karakter Brow Lee Pada *The Fabrique*