

Penerapan Algoritma *Pathfinding A** dalam *Game Dual Legacy* berbasis *Android*

Felix Octavian^{*1}, Latius Hermawan²

^{1,2}Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Katolik Misi Charitas, Palembang 30113, Sumatera Selatan, Indonesia
Email: ¹felixwu15@gmail.com, ²tiuz.hermawan@ukmc.ac.id

Abstract. *A* Pathfinding Algorithm Implementation in Dual Legacy Game based on Android.* Games have 2 characters, the player, and the NPC (Non-Playable Character) which cannot be controlled by the player, so the NPC movements are easy to predict. A Star (A*) algorithm is a pathfinding algorithm or finding a way to a destination, in this case searching for the closest path to the player and avoiding obstacles. The enemy NPC is tasked with chasing the player, and the enemy NPC must reduce the player's health. A* algorithm calculates the distance of one of the paths and then calculates the distance of the other paths. The algorithm will choose the shortest path when all paths have been completed. Research focuses on the NPC's task of finding the shortest route. The A* in the "Dual Legacy" 2D Side-Scrolling RPG game based on Android is expected with this algorithm NPC can search for and chase players/players via the nearest path. The conclusion is that the A Star Algorithm has been successfully implemented, the NPC approaches the player through the shortest distance by avoiding obstacles.

Keywords: A Star (A*) algorithm, NPC, game, Android, 2D side-scrolling RPG

Abstrak. *Game biasanya terdapat 2 karakter yaitu player dan NPC (Non-Playable Character) yang tidak bisa dikendalikan oleh player sehingga pergerakan karakter NPC mudah ditebak. Algoritma A Star (A*) merupakan algoritma pathfinding atau mencari jalan ke tujuan, dalam kasus ini mencari jalan terdekat ke player dan menghindari rintangan yang ada. NPC musuh ini bertugas untuk mengejar player dan NPC musuh harus mengurangi darah player. Algoritme A* menghitung jarak satu jalur, menyimpannya, lalu menghitung jarak jalur lainnya. Setelah semua jalur dihitung, algoritma A* memilih jalur terpendek. Penelitian berfokus pada tugas NPC untuk pencarian rute terdekat. Menerapkan algoritma pathfinding A* pada NPC game Dual Legacy 2D Side-Scrolling RPG berbasis Android diharapkan dengan algoritma tersebut NPC dapat mencari dan mengejar pemain / player melalui jalan terdekat. Kesimpulan perancangan ini adalah Algoritma A Star berhasil diimplementasikan, NPC mendekati player melalui jarak terdekat dengan menghindari halangan yang ada.*

Kata Kunci: algoritma A Star (A*), NPC, game, Android, 2D side-scrolling RPG

1. Pendahuluan

Dewasa ini perkembangan *game* mengalami perkembangan yang cukup pesat. Beberapa pendapat menemukan bahwa cukup banyak orang yang mengenal aplikasi permainan yang ada dalam komputer mulai dari anak kecil hingga dewasa yang memainkan *game* pada perangkat sejenis *Android*, karena permainan tersebut sangat interaktif dan mudah untuk dimainkan [1]. Namun dengan berkembangnya teknologi, *game* tidak hanya digunakan untuk hiburan, kini *game* telah diperluas fungsinya, misalnya *game* dapat digunakan sebagai sarana pembelajaran, biasa disebut *game* edukasi, ajang bisnis dan bersaing dengan misalnya salah satu olahraga profesional. Saat ini ada banyak *game* aksi, pertarungan, simulasi, strategi, olahraga, musik, pendidikan, dan petualangan [2][14]. Banyak *genre game* yang berkembang saat ini contohnya seperti Permainan *platformer*. *Platformer game* dapat dibagi menjadi *puzzle* dan *action platformer* [3]. Sejak perilisannya Super Mario Bros., banyak *platformer* lain yang lahir, terutama *platformer* 2D seperti "Sonic The Hedgehog". *Platform* 2D menyertakan *game side-scrolling*, artinya *game* tersebut dilihat dari sisi sudut kamera [4].

Game “Dual Legacy” ini merupakan *game* 2D *side-scrolling* RPG yang mirip dengan *game platformer* seperti “Mario Bros”, dimana *player* harus melewati rintangan untuk lanjut ke lantai berikutnya/*stage* berikutnya. Akan tetapi *game* “Dual Legacy” ini dibuat lebih menantang lagi dengan cara pemberian kecerdasan buatan/*artificial intelligence* di musuh yang ada di platform. NPC (*Non Player Character*) musuh ini bertugas untuk mengejar *player* apabila *player* mendekati jarak tertentu dan NPC musuh juga harus membunuh/mengurangi darah *player*.

Artificial intelligence (AI) membuat komputer mampu melakukan pekerjaan layaknya manusia seperti memutuskan sesuatu, berfikir logis, dll. Secara umum yang digunakan dalam adalah AI agar NPC mampu melakukan pergerakan, keputusan dan pembuatan strategi [5][13][15]. NPC tersebut dapat dikatakan sebagai *artificial intelligence* karena cara kerja NPC tersebut yang diberikan algoritma pencarian rute terdekat A* yang dapat menyelesaikan masalah seperti manusia / NPC tersebut dapat berpikir seperti manusia. Pemberian kecerdasan buatan tersebut dilakukan dengan cara memberikan algoritma *pathfinding A-star*. Prinsip dari algoritma ini melakukan perhitungan setiap simpul untuk menemukan lintasan terpendek. Algoritma A* akan menghitung jarak salah satu lintasan, lalu menyimpannya dan kemudian menghitung jarak lintasan lainnya, sehingga ditemukan lintasan terpendek secara keseluruhan [6][12].

Pemakaian algoritma A* karena algoritma ini terkenal akan keefektifitasnya untuk mencari jarak terdekat ke tujuan. Sehingga algoritma A* ini digunakan ke NPC musuh. Dibandingkan dengan algoritma *pathfinding* yang lainnya A* menggunakan fungsi heuristik yang mengutamakan *nodes* yang paling bagus dari yang lainnya sehingga proses cepat dan efektif. Menerapkan algoritma *pathfinding A** pada NP6C *game* “Dual Legacy” 2D *side-scrolling* RPG berbasis Android dimana diharapkan dengan algoritma tersebut NPC dapat mencari dan mengejar pemain/*player* melalui jalan terdekat. Dan dikarenakan *game* bersifat 2D, maka grafis *game* ini juga tidak dapat dibuat begitu tajam.

Hasil yang diharapkan dari penelitian ini yaitu menciptakan suatu *game* 2D *side-scrolling* RPG berbasis Android yang dapat menunjukkan kelebihan dan kekurangan algoritma *pathfinding A** NPC dalam mencari rute terdekat ke pemain.

2. Tinjauan Pustaka

2.1. Game (Permainan Digital)

Game (permainan digital) adalah karya seni dimana membuat pemain mampu melakukan keputusan untuk mengelola sumber daya yang dimilikinya melalui komponen yang ada di dalam *game* untuk mencapai tujuan [18][20]. *Game* dapat dimainkan menggunakan komputer/laptop, perangkat *mobile*, *console* dan lainnya. Dan juga terdapat beberapa *genre* dalam *game* yaitu aksi, petualangan, pertarungan, *first person shooter*, *third person shooter*, *role playing game*, *real time strategy*, simulasi, *sport*, *racing* dan *multiplayer* [7].

2.2. Game Dual Legacy

Pada *game* “Dual Legacy” ini *player* dan NPC musuh diberikan jumlah darah/*health bar* yang sesuai dan apabila jumlah darah mereka ini sudah habis, maka mereka akan terkalahkan. *Player* harus melewati rintangan NPC musuh dengan selamat. *Player* harus mengalahkan semua musuh yang ada agar *stage level* dapat selesai [19]. Karakter utama dari *game* “Dual Legacy” ini/*player* diberikan satu tombol serang/*attack* yang bisa digunakan kapanpun untuk mengurangi darah NPC musuh akan tetapi kerusakan/*damage* yang diberikan kepada NPC musuh berjumlah kecil sehingga harus berkali-kali menyerang mereka agar mereka dapat dikalahkan.

2.3. Algoritma A*

Algoritma *A-star* (A*) merupakan algoritma *pathfinding* yang dikembangkan dari algoritma BFS (*Breadth First Search*). Seperti hanya di BFS, A* dipandu oleh fungsi heuristik untuk mencari solusi [16]. Perbedaan antara A* dan BFS adalah bahwa algoritma ini tidak hanya menghitung biaya dari keadaan saat ini ke tujuan menggunakan fungsi heuristik, tetapi juga memperkirakan biaya jarak dari keadaan awal (lokasi tujuan) ke kondisi saat ini (ke posisi awal). Jadi jika jalur yang diambil terlalu panjang dan ada jalur lain yang lebih murah tetapi menawarkan solusi yang sama dari sudut pandang tujuan. Perhitungan yang digunakan oleh algoritma A* menggunakan rumus pada Persamaan 1 [8].

$$f(n) = g(n) + h(n) \tag{1}$$

dimana,

$f(n)$ = hasil penjumlahan yang diperoleh dari $g(x)$ dan $h(x)$,

$g(n)$ = jarak total diperoleh dari titik awal ke simpul saat ini (hasil akar dari *node* sekitar),

$h(n)$ = estimasi jarak tujuan, fungsi heuristik untuk memperkirakan berapa lama jalur menuju titik tujuan yang akan ditempuh ($|x_{Tujuan} - x_{SekitarAsal}| + |y_{Tujuan} - y_{SekitarAsal}|$)

2.4. Non-Playable Character (NPC)

Non Playable Character (NPC) merupakan sebuah agen otonom pada permainan digital dimana agen ini bergerak secara mandiri dan dikendalikan oleh program AI (*artificial intelligence*) yang diterapkan pada agen tersebut untuk mampu bertindak sesuai dengan alur permainan *game*. NPC dapat memberikan interaksi baik berupa suara, aksi, dialog percakapan, dan lain sebagainya tergantung dengan apa yang telah dilakukan oleh pemain selama ia bermain *game* tersebut [9][12][21].

2.5. Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan buatan (AI) adalah bagian ilmu komputer yang membuat perangkat komputer dapat berfikir seperti manusia. Namun seiring berkembangnya, peran perangkat digital berbasis komputer semakin mendominasi kehidupan. Komputer tidak hanya berfungsi sebagai alat komputasi, serta melakukan apapun yang dapat dilakukan oleh manusia. [10][17]. Menurut Kus, definisi kecerdasan buatan adalah “*Artificial intelligence* atau kecerdasan buatan adalah cabang ilmu komputer yang memungkinkan mesin (komputer) berfungsi seperti manusia.” [11].

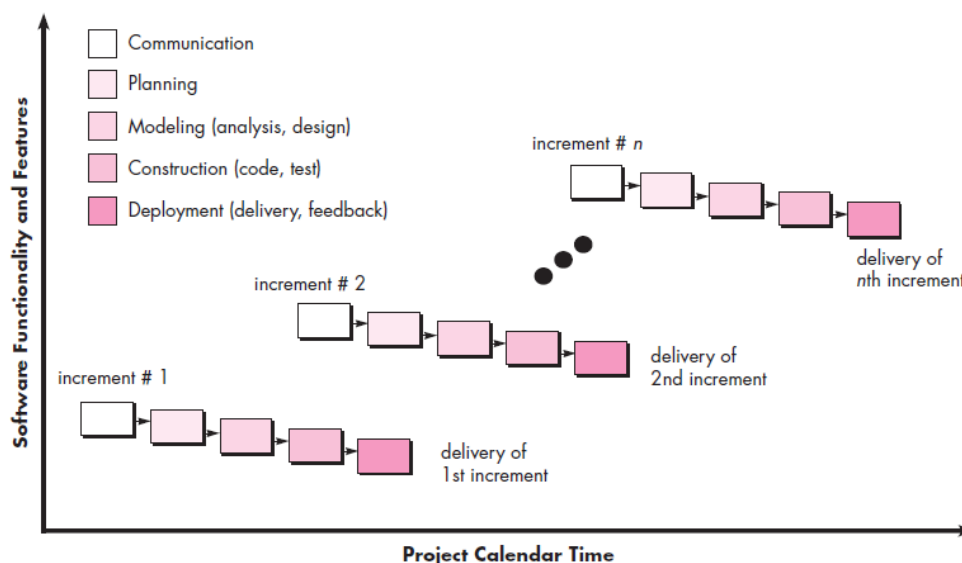
3. Metodologi Penelitian

3.1. Jenis Penelitian

Berdasarkan tujuan dan manfaat yang telah dijelaskan, di dapat bahwa penelitian ini merupakan jenis penelitian terapan. Penelitian ini disebut penelitian terapan karena dilakukan dengan menerapkan algoritma A* ke dalam NPC musuh *game* yang bertugas untuk mencari dan mengejar *player* melalui jalan terdekat.

3.2. Metodologi Pengembangan Sistem

Penelitian ini menggunakan *incremental model*. Urutan-urutan linear digunakan berulang kali dalam *incremental model* untuk membangun suatu perangkat lunak [22]. Seiring waktu, setiap perkembangan linier mengarah pada evolusi perangkat lunak yang digunakan pengguna seperti yang terlihat pada Gambar 1.



Gambar 1. *Incremental Model*

Incremental model dapat digunakan untuk proyek yang memiliki jadwal tertentu dan tidak memerlukan banyak orang untuk mengembangkannya. Berikut tahapan yang termasuk dalam model proses *incremental*:

1. *Communication* (Komunikasi)
Pada fase ini, informasi untuk pembuatan perangkat lunak dikumpulkan. Maka dari itu, pada fase ini dibuat bagian yang akan diteliti dan menentukan batas-batas masalah yang diperlukan.
2. *Planning* (Perencanaan)
Pada fase ini dibuat jadwal awal yang meliputi perkiraan waktu, tugas-tugas teknis yang harus dilakukan dan sumber daya yang dibutuhkan untuk membangun perangkat lunak.
3. *Modeling* (Pembentukan)
Pada fase ini, desain dibentuk dalam rekayasa perangkat lunak berupa desain algoritma, struktur data, desain perangkat lunak, dan desain representasi perangkat lunak.
4. *Construction* (Pembangunan)
Pada fase ini, pemrograman dilakukan berdasarkan model yang terbentuk. Setelah pemrograman selesai, kompatibilitas kebutuhan pengguna dapat diuji dan setiap kesalahan yang terjadi dapat diidentifikasi dan diperbaiki.
5. *Deployment* (Publikasi)
Pada fase ini pengembang memberikan dokumentasi tentang fitur yang dibuat dan menerima umpan balik berupa data dari *user* (pengguna) untuk memperbaiki aplikasi yang dibuat dan menambahkan fitur.

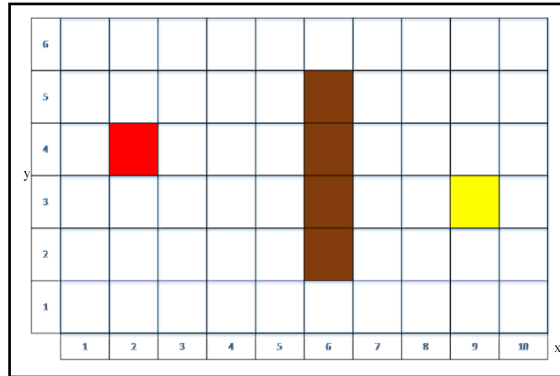
Dalam penelitian ini digunakan *model incremental process* karena model ini dikenal sebagai model yang cocok untuk digunakan ketika kita tidak memiliki banyak orang yang bekerja untuk mengembangkan sistem, serta model ini memiliki tahapan-tahapan yang harus terpenuhi terlebih dahulu sebelum mengembangkan ke tahap selanjutnya sehingga jelas nanti dimana perkembangannya terlihat.

4. Hasil dan Diskusi

4.1. Implementasi A* dan Simulasi perhitungan

Permainan yang akan dibangun memiliki tujuan menerapkan algoritma A* dalam *game* “Dual Legacy” yang dapat dimainkan di perangkat *Android*. Dalam *game* “Dual Legacy” terdapat dua jenis *player* yaitu *playable character* (PC) yang bergerak sesuai keinginan *user* dan *non-playable character* (NPC) yang bergerak tanpa perintah dari *user*. *Game* ini bergenre *survival*. *User* akan diarahkan ke tampilan menu utama dan terdapat dua pilihan yaitu mulai dan keluar. Kemudian *user* akan ditampilkan alur cerita *game* dimana terdapat seorang prajurit kerajaan (*player*) yang terperangkap di *dungeon/goa* yang berbentuk labirin, tidak hanya itu goa tersebut juga terdapat berbagai macam NPC musuh berupa monster dan *zombie*. Prajurit tersebut harus mencari jalan keluar dari goa tersebut dengan cara mengalahkan semua monster dan boss yang ada dalam goa tersebut agar dapat mengumpulkan kunci-kunci yang diperlukan untuk keluar dari goa tersebut. NPC musuh juga diterapkan algoritma A* untuk mengejar *player* apabila *player* mendekati jarak jangkauan NPC musuh yang sudah ditetapkan oleh penulis.

Terdapat satu buah *joystick* di bagian kiri layar untuk menggerakkan *player* dan satu lagi *joystick* dibagian kanan layar untuk menembakkan panah sesuai arah *joystick* yang diarahkan *player*. Terdapat juga dua buah tombol *skill* yang dapat digunakan *player* untuk membantu mengalahkan musuh. *User* dapat menghentikan *game* sementara menggunakan tombol *pause* dan memiliki beberapa pilihan seperti melanjutkan *game*, dan keluar dari *game*. *Game* akan berlangsung hingga *player* ditentukan menang atau kalah. Jika *player* berhasil mengalahkan semua musuh dan mengumpulkan kunci maka *player* dinyatakan menang seperti yang terlihat pada Gambar 2.



Gambar 2. Simulasi Perhitungan Algoritma A*

Langkah selanjutnya adalah menentukan *node* awal *enemy* (warna merah), pemain (warna kuning), dan *obstacle* (warna coklat). *Enemy* berada pada koordinat (2,4), pemain berada pada koordinat (9,3) dan *obstacle* berada pada koordinat (6,2), (6,3), (6,4), (6,5). Hitung nilai G, H, dan F untuk setiap *node* disekitar *node enemy*.

Node (1,3) :

• $G = \sqrt{1^2 + 3^2} = 3,16$ • $H = |9 - 1| + |3 - 3| = 8$ • $F = 3,16 + 8 = 11,16$

Node (1,4) :

• $G = \sqrt{1^2 + 4^2} = 4,12$ • $H = |9 - 1| + |3 - 4| = 9$ • $F = 4,12 + 9 = 13,12$

Node (1,5) :

• $G = \sqrt{1^2 + 5^2} = 5,09$ • $H = |9 - 1| + |3 - 5| = 10$ • $F = 5,09 + 10 = 15,09$

Node (2,3) :

• $G = \sqrt{2^2 + 3^2} = 3,60$ • $H = |9 - 2| + |3 - 3| = 7$ • $F = 3,60 + 7 = 10,60$

Node (2,5) :

• $G = \sqrt{2^2 + 5^2} = 5,38$ • $H = |9 - 2| + |3 - 5| = 9$ • $F = 5,38 + 9 = 14,38$

Node (3,3) :

• $G = \sqrt{3^2 + 3^2} = 4,24$ • $H = |9 - 3| + |3 - 3| = 6$ • $F = 4,24 + 6 = 10,24$

Node (3,4) :

• $G = \sqrt{3^2 + 4^2} = 5$ • $H = |9 - 3| + |3 - 4| = 7$ • $F = 3,16 + 8 = 12$

Node (3,5) :

• $G = \sqrt{3^2 + 5^2} = 5,83$ • $H = |9 - 3| + |3 - 5| = 8$ • $F = 3,16 + 8 = 13,83$

Jika *node* pada langkah tiga merupakan *obstacle*, maka cari *node* lain dengan nilai F paling rendah selanjutnya. Hal ini terjadi saat *enemy* sudah mencapai node (5,3). Seharusnya *enemy* bergerak ke node (6,3) dengan nilai F sebesar 9,70. Tetapi, karena node (6,3) merupakan *obstacle* maka *enemy* berpindah ke node (5,2) dengan nilai F paling rendah kedua yaitu sebesar 10,32. Proses perpindahan *enemy* dapat dilihat pada Gambar 3.

| | | | | | | | | | |
|---|------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|---|---|---|
| 6 | | | | | | | | | |
| 5 | G: 5.09 H: 10 F: 15.09 | G: 5.38 H: 9 F: 14.38 | G: 5.83 H: 8 F: 13.83 | | | | | | |
| 4 | G: 4.12 H: 9 F: 13.12 | | G: 5 H: 7 F: 12 | G: 5.65 H: 6 F: 11.65 | G: 6.40 H: 5 F: 11.40 | G: 7.21 H: 4 F: 11.21 | | | |
| 3 | G: 3.16 H: 8 F: 11.16 | G: 3.60 H: 7 F: 10.60 | G: 4.24 H: 6 F: 10.24 | G: 5 H: 5 F: 10 | G: 5.83 H: 4 F: 9.83 | G: 6.70 H: 3 F: 9.70 | | | |
| 2 | | G: 2.82 H: 8 F: 10.82 | G: 3.60 H: 7 F: 10.60 | G: 4.47 H: 6 F: 10.47 | G: 5.38 H: 5 F: 10.38 | G: 6.32 H: 4 F: 10.32 | | | |
| 1 | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Gambar 3. Simulasi Algoritma A* Terhalang Obstacle

Langkah berikutnya adalah mengulang kembali langkah 2 - 4 hingga *enemy* mencapai pemain. Langkah ini terus di ulangi hingga *enemy* mencapai *node* pemain (9,3) seperti pada Gambar 4 berikut.

| | | | | | | | | | | |
|---|------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|----|
| 6 | | | | | | | | | | |
| 5 | G: 5.09 H: 10 F: 15.09 | G: 5.38 H: 9 F: 14.38 | G: 5.83 H: 8 F: 13.83 | | | | | | | |
| 4 | G: 4.12 H: 9 F: 13.12 | | G: 5 H: 7 F: 12 | G: 5.65 H: 6 F: 11.65 | G: 6.40 H: 5 F: 11.40 | G: 7.21 H: 4 F: 11.21 | G: 8.06 H: 3 F: 11.06 | G: 8.94 H: 2 F: 10.94 | G: 9.84 H: 1 F: 10.84 | |
| 3 | G: 3.16 H: 8 F: 11.16 | G: 3.60 H: 7 F: 10.60 | G: 4.24 H: 6 F: 10.24 | G: 5 H: 5 F: 10 | G: 5.83 H: 4 F: 9.83 | G: 6.70 H: 3 F: 9.70 | G: 7.61 H: 2 F: 9.61 | G: 8.54 H: 1 F: 9.54 | G: 9.48 H: 0 F: 9.48 | |
| 2 | | G: 2.82 H: 8 F: 10.82 | G: 3.60 H: 7 F: 10.60 | G: 4.47 H: 6 F: 10.47 | G: 5.38 H: 5 F: 10.38 | G: 6.32 H: 4 F: 10.32 | G: 7.28 H: 3 F: 10.28 | G: 8.24 H: 2 F: 10.24 | G: 9.21 H: 1 F: 10.21 | |
| 1 | | | | G: 4.12 H: 7 F: 11.12 | G: 5.09 H: 6 F: 11.09 | G: 6.08 H: 5 F: 11.08 | G: 7.07 H: 4 F: 11.07 | G: 8.06 H: 3 F: 11.06 | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Gambar 4. Simulasi Algoritma A* *Enemy* Mencapai Pemain

4.2. Implementasi Karakter

Asset karakter yang digunakan dalam game “Dual Legacy” terdapat empat yaitu karakter *player*, NPC musuh *demon*, NPC musuh *zombie*, dan NPC musuh *boss*:

1. Karakter *player*

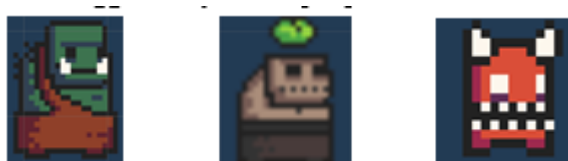
Karakter *player* yang dikontrol oleh *user* yang dapat berjalan dan menembak sesuai dengan keinginan *user*. *Asset* karakter yang digunakan dalam game “Dual Legacy” seperti Gambar 5.



Gambar 5. *Asset* Karakter *Player*

2. NPC musuh

NPC musuh tidak dapat dikendalikan oleh *player* melainkan mereka bergerak mendekati *player* sesuai dengan perhitungan algoritma A* jika *player* masuk ke dalam jarak perhitungan algoritma untuk menyerang *player*. *Asset*, NPC musuh *demon*, NPC musuh *zombie*, dan NPC musuh *boss* yang digunakan dalam game “Dual Legacy” dapat dilihat pada Gambar 6.



Gambar 6. NPC Musuh

4.3. Implementasi Antarmuka *Game*

Game “Dual Legacy” terdapat beberapa *scene*, termasuk *main menu* dan *win screen*. Berikut ini adalah antarmuka yang terdapat pada game “Dual Legacy”:

1. Antarmuka *Main Menu*

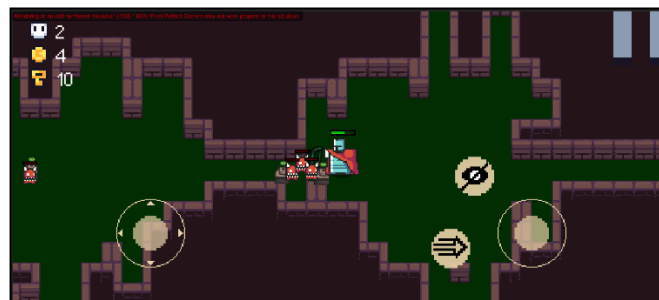
Antarmuka *main menu* ini merupakan antarmuka yang pertama kali ditampilkan ketika *user* membuka aplikasi game “Dual Legacy”, terdapat gambar latar belakang, judul game, dan tombol *play* serta *quit*. Hasil implementasi dapat dilihat pada Gambar 7.



Gambar 7. Implementasi Antarmuka Main Menu

2. Antarmuka *Game*

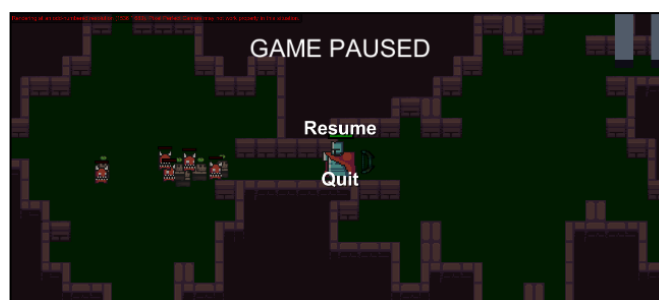
Antarmuka ini ditampilkan setelah *user* melewati antarmuka alur cerita, *user* dapat bermain dengan mengalahkan semua musuh yang ada di dalam goa dengan menggunakan kontrol yang terdapat di antarmuka *game* terdapat satu *joystick* untuk bergerak satu *joystick* untuk menembakkan panah sesuai arah yang diinginkan dua buah tombol *skill* yang dapat digunakan untuk membantu *player* dan dipojok kanan atas terdapat tombol *pause*, di pojok kiri atas terdapat tampilan status *progress game*. Hasil implementasi ada di Gambar 8.



Gambar 8. Antarmuka Scene Game

3. Antarmuka Menu *Pause*

Antarmuka *pause* ini ditampilkan apabila *user* mengklik tombol *pause* dipojok kanan atas maka *game* akan terhentikan sementara hingga *user* memilih salah satu pilihan antara *resume* untuk melanjutkan permainan atau *quit* untuk langsung keluar ke antarmuka *main menu*. Hasil implementasi dapat dilihat pada Gambar 9.



Gambar 9. Implementasi Antarmuka Menu Pause

4. Antarmuka Hasil *Game*

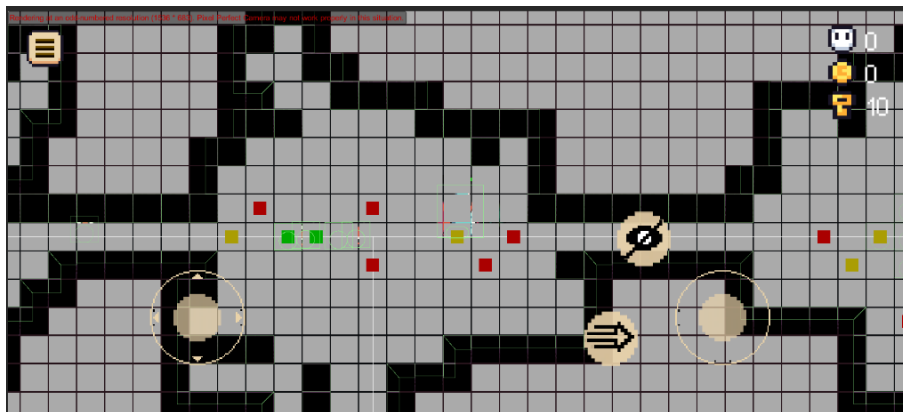
Antarmuka hasil *game* terdiri dari dua tergantung dari kondisi hasil *player* bermain, apabila berhasil mengalahkan semua musuh berarti *player* menang dan yang akan tampil adalah antarmuka hasil *game* kondisi menang dimana tertampil *win screen* dan ada pilihan *quit* untuk keluar langsung ke *main menu*. Apabila *player* kalah dengan artian

darah habis maka akan tampil antarmuka kalah dimana terdapat dua pilihan *play again* untuk bermain lagi atau *quit* untuk langsung keluar ke *main menu*. Hasil implementasi dapat dilihat pada Gambar 10.



Gambar 10. Implementasi Antarmuka Hasil Game (Kondisi Menang)

5. Antarmuka Implementasi A*
Antarmuka ini menampilkan ke *developer* tentang pembuktian bahwa algoritma *A Star* bekerja dengan seharusnya. Hasil implementasi dapat dilihat pada Gambar 11.



Gambar 11. Implementasi Antarmuka A Star

4.4. Pengujian Statistik

Pengujian ini menggunakan pengujian deskriptif. Pengujian ini dilakukan secara mandiri dengan kondisi akhir pemain dapat menang dengan cara menyelesaikan semua tugas yang harus dilakukan untuk menang dan apabila pemain tidak dapat menyelesaikan semua tugas sebelum darah pemain habis maka ia akan kalah seperti yang terlihat pada Gambar 12.

Hasil Pengujian Permainan Dual Legacy



Gambar 12. Hasil Pengujian Permainan Dual Legacy

Berdasarkan pengujian statistik yang dilakukan dapat dikatakan dalam 50 kali permainan yang diuji permainan tidak selalu berakhir dengan kondisi menang. Dimana pemain berhasil memenangkan sebanyak 29 kali sedangkan NPC berhasil memenangkan permainan sebanyak 21 kali. Hal ini membuktikan bahwa NPC mampu mengalahkan pemain dengan menggunakan algoritma yang telah diterapkan.

5. Kesimpulan dan Saran

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan dapat ditarik kesimpulan dari *game* “Dual Legacy” adalah algoritma A* berhasil diimplementasikan pada NPC musuh untuk mendekati *player* apabila *player* masuk ke jangkauan perhitungan musuh dan NPC musuh mendekati *player* melalui jarak terdekat dengan menghindari halangan/*obstacle* yang ada. Berdasarkan pengujian statistik yang dilakukan dapat dikatakan dalam 50 kali permainan yang diuji permainan tidak selalu berakhir dengan kondisi menang. Dimana pemain berhasil memenangkan sebanyak 29 kali sedangkan NPC berhasil memenangkan permainan sebanyak 21 kali. Hal ini membuktikan bahwa NPC mampu mengalahkan pemain dengan menggunakan algoritma yang telah diterapkan. Setelah penelitian dan pembangunan *game* ini dilakukan dan ditarik kesimpulan, penelitian ini masih bisa dikembangkan. Berikut ini adalah saran yang dapat disampaikan yaitu menambah beberapa tipe musuh lagi dan mungkin menambahkan algoritma lainnya agar NPC musuh tidak bertabrakkan serta membuat NPC musuh dapat menyerang *player* dengan jarak jauh.

Referensi

- [1] Hermawan, L and M. B. Ismiati, “Penerapan Augmented reality Berbasis Minimax Algorithm pada *Game* Papan Cerdas,” *J. Buana Informatika*, pp. 21–30, 2022.
- [2] W. Widodo and I. Ahmad, “Penerapan Algoritma A Star (A*) pada *Game* Petualangan Labirin Berbasis Android,” *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 3, no. 2, p. 57, 2018, doi: 10.23917/khif.v3i2.5221.
- [3] T. Bhosale, S. Kulkarni, and S. N. Patankar, “2D Platformer *Game* in Unity Engine,” *Int. Res. J. Eng. Technol.*, pp. 3021–3024, 2018, [Online]. Available: www.irjet.net
- [4] A. Darmawan, E. Muhammad, A. Jonemaro, and T. Afirianto, “Penerapan Dynamic Audio Pada 2D Endless Runner *Game* Menggunakan Pure Data,” vol. 3, no. 5, pp. 4142–4147, 2019.
- [5] T. Stanley *et al.*, “PEMBUATAN *GAME* FIGHTING,” vol. 2, pp. 177–187, 2019.
- [6] Pamungkas. dkk, “Penerapan Algoritma A* (A Star) Pada *Game* Edukasi The Maze Island Berbasis Android,” *Penerapan Algorith.*, no. x, pp. 1–11, 2014.
- [7] B. Setiawan and L. Hermawan, “Edukasi Protokol Kesehatan Berbasis *Game*,” *Jurnal Informatika Polinema*, vol.8, no.1, pp. 25–30, 2021.
- [8] A. Pane and M. Darwis Dasopang, “Belajar Dan Pembelajaran,” *FITRAH Jurnal Kaji. Ilmu-ilmu Keisl.*, vol. 3, no. 2, p. 333, 2017, doi: 10.24952/fitrah.v3i2.945.
- [9] Winata H and Liliana, “Meningkatkan Variasi Tindakan Non-Playable Character Pada *Game* Survival Menggunakan Metode Markov”, *Jurnal INFRA*, vol.8, no.2, 2021.
- [10] Jaya, Hendra and Sabran, "Kecerdasan Buatan". *Fakultas MIPA Universitas Negeri Makassar*, pp. 1-315, 2018.
- [11] I. P. W. Merta, I. M. G. Sunarya, and I. K. R. Arthana, “Handgesture To Text Dengan Metode *Artificial intelligence* KNN (K-Nearest Neighbour),” *KARMAPATI (Kumpulan Artik. Mhs. Pendidik. Tek. Inform.*, vol. 4, no. 1, pp. 18–27, 2015, [Online]. Available: <https://ejournal.undiksha.ac.id/index.php/KP/article/view/19789>
- [12] Zhu, X. "Behavior tree design of intelligent behavior of non-player character (NPC) based on Unity3D". *Journal of Intelligent and Fuzzy Systems*, 37(5), 6071–6079, 2019.
- [13] Jeon. "An Implementation of Othello *Game* Player Using ANN based Records Learning and Minimax Search Algorithm". *The Transactions of The Korean Institute of Electrical Engineers*. 2018.

- [14] Hidayat, E. W. "Penerapan Finite State Machine pada Battle Game Berbasis Augmented Reality". *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 5(1), 54, 2019.
- [15] Mustofa, S. and Rahmawati, E. "Penerapan Finite State Machine Untuk Pengendalian Animasi Pada Video Game RPG Nusantara Legacy". *Jurnal Sistem Komputer Musi Rawas*, vol. 03, no. 01, 2018.
- [16] Sunandar U., Raharjo AS. "Perancangan Permainan Ular Tangga Berbasis Virtual Reality Dengan Algoritma Random Number Generator". *e-Proceeding of Engineering : Vol.8, No.5 Oktober 2021*
- [17] P. L. Ekawati and A. Z. Fallani, "Pemanfaatan Teknologi Game Untuk Pembelajaran Ragam Budaya Indonesia Berbasis Android," *Jurnal Link*, pp. 5-30, 2015.
- [18] Hermawan, L, Malla M., Ismiati MB. "Penerapan Algoritma PRNG Dalam Permainan Snack and Ladders Berbasis Digital". *Jurnal Informatika Upgris Vol 8, No 2, 2022.*
- [19] Calvinus W. "Implementasi Algoritma Minimax Dalam *Game Othello*". *JTIUST*, Volume 02 Nomor 01, Juni 2017.
- [20] Hermawan L, Hariadi M, Basuki RS. "Koordinasi Nonplayer Character Follower Menggunakan Algoritma Potential Fields Berbasis Multibehaviour". *Jurnal Cyberku Volume 13 No 1. 2017*
- [21] Minsong Liu. "An Improved Minimax-Q Algorithm Based on Generalized Policy Iteration to Solve a Chaser-Invader *Game*". Auckland University of Technology. IEEE Xplore. 2020.
- [22] K. Rokoyah, Y. I. Chandra, and S. Lukman, "PENERAPAN MODEL INCREMENTAL DALAM MERANCANG APLIKASI PENGENALAN BENTUK DAN FUNGSI GIGI PADA MANUSIA BERBASIS WEB," 2022.