

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam melakukan penelitian, diperlukan mempelajari penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan. Penelitian-penelitian yang terkait ini, peneliti gunakan sebagai referensi untuk membantu mengembangkan penelitian, untuk mengetahui bagaimana penerapan *Algoritma* ataupun mengembangkan penelitian tersebut agar terhindar dari penelitian yang sama. Berikut adalah sumber-sumber penelitian sebelumnya yang terkait dengan pengembangan dan analisis *game* yang menerapkan *pathfinding* yang dapat dijadikan sebagai bahan acuan maupun perbandingan dalam penelitian yang akan dilakukan. Berikut adalah beberapa penelitian terdahulu.

2.1.1 Literatur 1

Penelitian yang dilakukan oleh Ade Candra, Mohammad Andri Budiman, dan Rahmat Irfan Pohan (2021), Fakultas *Computer Science and Information Technology* Universitas Sumatera Utara yang berjudul *Application of A-Star Algorithm on Pathfinding Game* membahas tentang *Pathfinding* adalah metode untuk menentukan jalur terpendek. *Pathfinding* banyak digunakan di berbagai bidang Ilmu Komputer. Salah satunya adalah pengembangan *game*. Penelitian ini akan merancang dan menganalisis *pathfinding* menggunakan Algoritma A-Star dan mengimplementasikannya pada permainan menanam pohon. Kecerdasan Buatan dalam *game* akan menemukan jalur terpendeknya sendiri dari simpul awal ke simpul tujuan. Heuristik A-Star adalah *Manhattan*

Distance yang memungkinkan pergerakan ke empat arah berbeda. Hasil menunjukkan bahwa jumlah *node* yang dikunjungi akan bertambah jika *grid* memiliki banyak *node* hambatan, dan rata-rata waktu proses *pathfinding* A-Star adalah 0,0732 detik. Lalu kelemahan dalam penelitian ini adalah pada banyaknya *obstacle* yang mempengaruhi proses untuk mencari rute[2].

2.1.2 Literatur 2

Penelitian yang dilakukan oleh Graldy Tirta Kumala, dan Wirawan Istiono (2022) Universitas Multimedia Nusantara, yang berjudul *Comparison of Flow Field and A-Star Algorithm for Pathfinding in Tower Defence Game*. Tower Defense adalah genre game yang menggunakan Algoritma pencarian jalan. Pathfinding adalah cara untuk menemukan jalur dari satu titik ke titik lainnya; pathfinding sendiri memiliki banyak varian dengan skenario yang berbeda-beda. Salah satu algoritme ini adalah algoritme A-Star, metode yang terkenal dan umum digunakan untuk pencarian jalur game. Algoritma lainnya adalah Algoritma Flow Field, yang merupakan Algoritma yang belum banyak diketahui dan menjadi topik dalam penelitian berikut ini. Penelitian ini akan dilakukan dengan membandingkan waktu yang dibutuhkan oleh kedua Algoritma untuk mencapai titik target dari titik awal, dan perbandingan waktu akan dilakukan dalam 3 skenario yang berbeda. Hasil dari penelitian adalah Algoritma Flow Field mencapai target lebih cepat dari A-Star Pathfinding pada setiap skenario yang dilakukan pada simulasi. Penelitian ini menyimpulkan bahwa Algoritma Flow Field dapat bersaing dengan Algoritma A-Star untuk mencari jalur pada game Tower Defense..

Kelemahan dari penelitian ini adalah jika map terlalu besar maka penggunaan memori akan meningkat[6].

2.1.3 Literatur 3

Penelitian yang berjudul *The Implementation of A-Star Algorithm to Determine Optimal Pursue Routes in Drug Eradication Game* yang dilakukan oleh Lukas Tommy, Yohanes Setiawan Japriadi, dan Syachriza Hilmaida Habibur Program Studi Teknik Informatika, Fakultas Teknologi Informasi, ISB Atma Luhur (2021). Narkoba memiliki beberapa dampak negatif seperti ketergantungan, kerusakan otak dan saraf, hingga kematian jika disalahgunakan. Sosialisasi mengenai bahaya narkoba di masyarakat oleh pihak kepolisian masih dilaksanakan secara konvensional sehingga kurang efektif terlebih di masa pandemi Covid-19 seperti saat ini. Pada penelitian ini akan diusulkan sebuah permainan Android yang dapat menyampaikan materi bahaya narkoba dengan menarik, interaktif, dan efektif. Pada permainan yang diusulkan, kecerdasan buatan akan diterapkan pada musuh agar dapat mengejar pemain dengan melalui rute yang optimal. Salah satu Algoritma mencari rute terdekat dari lokasi awal menuju tujuan adalah A-Star. Algoritma A-Star memanfaatkan heuristik dalam menilai simpulan yang ada pada peta berbentuk grid sehingga waktu pengerjaannya lebih singkat dibandingkan Dijkstra. Berdasarkan analisis yang sudah dilaksanakan diketahui bahwa secara keseluruhan kinerja dari Algoritma A-Star sudah baik dimana tujuh musuh dalam permainan dapat mengejar pemain secara real-time melalui rute yang optimal. Musuh namun tidak mampu untuk bekerja

sama dalam mengepung pemain dan cenderung berkumpul di satu titik dengan melalui rute yang sama sehingga efektivitas pengejaran tidak maksimal. Permainan yang diusulkan juga dapat menyampaikan materi mengenai jenis-jenis narkoba dan dampak negatifnya dengan menarik, interaktif, sekaligus efektif melalui menu gamepedia sekaligus karakter musuh yang ditampilkan sepanjang permainan. Kelemahan dalam penelitian ini adalah penambahan *cell* pada musuh yang tidak bisa dilewati pemain[7].

2.1.4 Literatur 4

Penelitian yang berjudul *Comparing the Efficiency of Pathfinding Algorithms for NPCs in platform games* yang dilakukan oleh Umar Affandi Shahrin Iskandar, Norizan Mat Diah, Marina Ismail, dan Azizi Abdullah (2022) Universiti Teknologi MARA dan *The National University of Malaysia*. Pathfinding telah menjadi area penelitian video game yang signifikan selama beberapa dekade. Ini biasanya digunakan sebagai inti dari setiap gerakan Kecerdasan Buatan dalam game komputer. Penelitian ini bertujuan untuk mengidentifikasi Algoritma pathfinding yang lebih cocok dan lebih efisien untuk genre video game platformer. Penelitian ini membandingkan dua Algoritma yaitu Algoritma A* dan Dijkstra. Kedua algoritme diimplementasikan dalam lingkungan permainan platform dan diuji dengan beberapa kendala berbeda untuk karakter non-pemain (NPC). Parameter yang diukur adalah waktu proses, panjang jalur yang ditempuh, dan jumlah blok/node yang dimainkan dalam proses komputasi. Untuk mengevaluasi kinerja algoritme, waktu perjalanan yang diambil, node yang dihitung, dan

jarak yang ditempuh NPC untuk mencapai tujuannya dianalisis untuk setiap algoritme. Temuan menunjukkan bahwa kedua algoritme cocok untuk kondisi tertentu di lingkungan platformer; Dijkstra tampil akurat dan berhasil menemukan jalur terpendek saat rute menuju tujuan membutuhkan lebih sedikit pergerakan vertikal, sementara A* tampil lebih efisien saat NPC diharuskan mencapai tujuan yang membutuhkan lebih banyak pergerakan vertikal. Hasilnya juga menunjukkan bahwa A* bekerja lebih baik daripada algoritme Dijkstra, karena memiliki fungsi heuristik yang meningkatkan fleksibilitasnya. Kelemahan dari penelitian ini adalah jika implementasi dilakukan *ground-based movement*[8].

2.1.5 Literatur 5

Azyan Yusra Kapi, Mohd Shahrizal Sunar, dan Zeyad Abd Alfoor (2022) Universiti Teknologi MARA, Universiti Teknologi Malaysia, dan *University of Mosul*, melakukan penelitian yang berjudul *A Comparison of Pathfinding Algorithm for Code Optimization on Grid Maps*. Ada berbagai Algoritma pencarian jalur yang dibuat dan dikembangkan selama beberapa dekade terakhir untuk membantu menemukan jalur terbaik antara dua titik. Makalah ini menyajikan perbandingan beberapa Algoritma untuk pathfinding pada peta grid 2D. Sebagai hasilnya, penelitian ini mengidentifikasi Jump Point Search Block Based Jumping (JPS (B)) sebagai Algoritma potensial dalam hal lima metrik evaluasi termasuk waktu pencarian. Perbandingan menunjukkan potensi algoritme dan pengoptimalan kode dilakukan pada algoritme JPS(B) terpilih, dan diberi nama JPS(BCO). Makalah ini juga

membahas masalah terkait JPS(B) dan cara mengatasinya untuk mengoptimalkan akses ke pointer indeks. Peningkatan JPS(BCO) yang disajikan mampu mencari jalur optimal lebih cepat daripada JPS(B) asli seperti yang ditunjukkan oleh temuan eksperimental. Eksperimen peta grid dengan ukuran berbeda dilakukan untuk memvalidasi kemampuan Algoritma yang diusulkan dalam hal waktu pencarian. Studi banding dengan JPS asli (B) menunjukkan peningkatan yang lebih bermanfaat pada peta grid dengan ukuran berbeda dalam hal waktu pencarian. Kekurangan penelitian ini perlunya untuk memperbaiki dan menyempurnakan JPS(B) secara signifikan[9].

2.1.6 Literatur 6

Jakub Smolka, Kamil Miszta, Maria Skublewska-Paszkowska, dan Edyta Lukasik (2019) melakukan penelitian yang berjudul *A* Pathfinding Algorithm Modification for a 3D Engine* dari Lubin, Poland. Seringkali kualitas jalur yang dikembalikan oleh algoritme pencarian jalur lebih penting daripada performa algoritme tersebut. Makalah ini menyajikan Algoritma baru, berdasarkan A*, yang lebih cocok untuk digunakan dalam mesin game 3D. Modifikasi dievaluasi dengan serangkaian tes komparatif. Algoritma standar A* digunakan sebagai benchmark dalam perbandingan. Perubahan pada algoritme terdiri dari penggunaan heuristik yang berbeda, penambahan pinalti verteks, dan pasca pemrosesan jalur. Mesin grafis 3D yang dibuat khusus digunakan sebagai lingkungan pengujian. Jalur yang dihasilkan oleh algoritme baru adalah pilihan yang lebih alami bagi manusia daripada jalur

yang dipilih oleh algoritme A* standar. Kelemahan dari penelitian ini adalah diperlukannya meningkatkan kualitas jalur untuk mengurangi penghitungan oleh Algoritma[10].

2.1.7 Literatur 7

Penelitian yang berjudul Penerapan Algoritma A-Star Dalam Mencari Jalur Tercepat dan Pergerakan *NonPlayer Character* Pada *Game* Petualangan Labirin *Tech-Edu* oleh Bagus Tegar, Dwi Irianto, Septi Andryana, dan Aris Gunaryati (2021) Program Studi Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional, Jakarta Selatan. Penelitian ini mengangkat masalah pentingnya untuk memperkuat karakter siswa pada pembelajaran tematik menggunakan media pembelajaran *board game*. Permainan merupakan sarana hiburan yang berada dalam media elektronik seperti smartphone yang dibuat semenarik mungkin sehingga pemain mendapat kepuasan batin. Perkembangan pada industri permainan berkembang cukup cepat dari mobile maupun desktop. Kategori pada permainan cukup banyak seperti aksi, strategi, olahraga, penembak, petualangan dan simulasi. Game petualangan merupakan salah satu kategori permainan yang pemainnya waspada untuk berpikir agar bisa menyelesaikan permainan tersebut. banyak hal yang perlu diperhatikan seperti visual pada peta ataupun interaksi pada NPC (*NonPlayer Character*), dengan memanfaatkan teknologi sekarang pembelajaran dapat disampaikan melalui sebuah permainan. Karakteristik pada permainan yang menghibur dan kegiatan visual yang menarik akan membuat pembelajaran semakin diminati.

Pembuatan permainan tentunya akan menggunakan Algoritma sebagai fitur ataupun sistem dalam pembuatan permainan. Algoritma a-star salah satu yang banyak digunakan dalam pembuatan permainan labirin. Algoritma a-star dapat diterapkan pada bantuan jalur tercepat ataupun pergerakan pada NPC. Tujuan dari penelitian membuat permainan edukasi labirin dengan menggunakan Algoritma a-star sebagai jalur pencarian tercepat dan pergerakan NPC. Hasil pengujian pada permainan pada tujuh perangkat mendapatkan hasil berhasil dijalankan, dalam penggunaan CPU pada perangkat mobile minimal 7% dan maksimal 57%. Penggunaan CPU pada perangkat desktop minimal sebesar 2% dan maksimal 33%. Kelemahan dalam penelitian ini adalah penggunaan pada perangkat *mobile* lebih besar dibandingkan pada perangkat *desktop*[11].

2.1.8 Literatur 8

Phalgun Chintala, Rolf Dornberger, dan Thomas Hanne (2022) melakukan penelitian yang berjudul *Robotic Path Planning by Q Learning and a Performance Comparison with Classical Path Finding Algorithms* dari *University of Applied Sciences and Arts Northwestern Switzerland*. Q-Learning adalah bentuk pembelajaran penguatan untuk masalah pencarian jalur yang tidak memerlukan model lingkungan. Ini memungkinkan agen untuk menjelajahi lingkungan tertentu dan pembelajaran dicapai dengan memaksimalkan hadiah untuk serangkaian tindakan yang dilakukan. Belakangan ini, pendekatan Q-Learning terbukti berhasil dalam berbagai aplikasi mulai dari sistem navigasi hingga video game. Makalah ini

mengusulkan metode berbasis Q-learning yang mendukung perencanaan jalur untuk robot. Makalah ini juga membahas pilihan nilai parameter dan menyarankan parameter yang dioptimalkan saat menggunakan metode tersebut. Kinerja algoritme pencarian jalur paling populer seperti algoritme A* dan Dijkstra telah dibandingkan dengan pendekatan Q-learning dan mampu mengungguli Q-learning sehubungan dengan waktu komputasi dan panjang jalur yang dihasilkan. Kelemahan pada penelitian ini adalah untuk mengkomparasi pada *hybrid classical path-based*[12].

2.2 Game (Permainan)

Menurut Clark *game* adalah kegiatan yang melibatkan keputusan pemain, untuk mencapai tujuan dengan dibatasi oleh ruang lingkup tertentu. Pemain dituntut untuk bisa mencapai *game over* atau permainan dinyatakan selesai dengan kondisi menang ataupun kalah, sehingga pemain harus berfikir dan membuat strategi untuk memenangkan permainan dengan waktu atau poin yang ingin dicapai[13].

Di zaman sekarang permainan sudah banyak menggunakan media elektronik, yang berbentuk multimedia, untuk dapat menarik minat dan menghibur pemain. *Game* tidak hanya dimainkan oleh anak-anak, tetapi orang dewasa sampai orang tua juga suka bermain *game*. Banyak macam *game* yang sudah beredar seiring berkembangnya zaman. *Game* juga dapat dibedakan menjadi dua jenis yaitu *game* 2D (dua dimensi) dan 3D (tiga dimensi). *Game* dua dimensi adalah *game* yang memiliki sumbu X dan Y, *game* tiga dimensi adalah *game* yang memiliki *axis* X, Y, dan Z[13].

Game memiliki banyak macam genre, contohnya: *Action* (aksi), *Adventure* (petualangan), *Strategy* (strategi), *Sport* (olahraga), *Arcade* (ketangkasan) dan lain sebagainya. *Action* adalah *game* yang biasanya banyak aksi yang dilakukan seperti bertarung, tembak menembak dan lain sebagainya. Kemudian *Adventure* adalah sebuah yang pemainnya harus berpetualang seakan masuk kedalam *game* tersebut. *Strategy* adalah genre *game* yang banyak disukai oleh *gamers* karena pemain dituntut untuk dapat berpikir dan bertindak cepat. *Sport* adalah permainan yang biasanya terdapat permainan memukul, menendang, ataupun berbalapan. *Arcade* adalah permainan yang konsep dan desain yang sederhana, dan memiliki tingkat kesulitan yang bertambah ditiap level, sehingga permainan akan bertambah seru dan menantang[13].

2.3 Pathfinding

Pathfinding adalah salah satu pencarian jalur tercepat dari titik awal ke titik tujuan dengan menghindari halangan disepanjang jalur yang dilewati, *pathfinding* sendiri memiliki banyak variasi untuk permasalahan tertentu[6]. *Pathfinding* juga menjadi salah satu teknik yang populer untuk masalah pencarian jalur, pada pengaplikasiannya seperti pada *virtual reality*, *game* komersil, dan robotik. Pada masa lampau, *pathfinding* berfokus pada pencarian rute saja, tetapi sekarang fokusnya menjadi lebih detail seperti biaya terendah, dan jalur yang teraman untuk dapat menghindari macam gangguan seperti bentuk rute, maupun *obstacles*[9].

Pathfinding dikategorikan menjadi dua yaitu *undirected* dan *directed*. *Undirected* dianalogikan dengan seekor semut yang terjebak didalam labirin yang berkeliaran membabi buta mencoba mencari jalan keluar, terdapat dua pendekatan

dalam penyelesaian masalah pada *pathfinding undirected* yaitu *Breadth-first search* dan *Depth-first search*. *Directed* tidak melewati labirin secara membabi buta, terdapat beberapa metode untuk menentukan semua node yang berdekatan sebelum ditentukan mana yang terbaik, menggunakan pendekatan *Uniform cost search* $g(n)$ dan *Heuristic search* $h(n)$ [14].

2.4 Algoritma A* (A Star)

Peter Hart, Nils Nilsson, dan Bertram Raphael mengusulkan Algoritma A* pada tahun 1968. Algoritma A* merupakan Algoritma *heuristic*, artinya dibutuhkan jalan pintas menuju solusi optimal [5]. Algoritma A* merupakan salah satu *pathfinding* paling populer dan *best-performing pathfinding* Algoritma. Pencipta Algoritma A* menyadari jika menjelajahi semua jalur yang kompleks maka waktu yang dibutuhkan akan semakin banyak, menggunakan *heuristic* dapat mempercepat deteksi dari jalur, tetapi memberi hasil yang kurang optimal jika terdapat banyak *obstacles*[10]. Algoritma A* adalah pengembangan dari algoritma Dijkstra dengan menetapkan biaya antara posisi awal dan posisi tujuan melalui *heuristic* untuk membuat pencarian terarah dan mempercepat pencarian[4].

Untuk menentukan jalan pintas tersebut, fungsi yang digunakan Algoritma A* sebagai berikut:

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

Dimana:

$f(n)$ = biaya estimasi terendah,

$g(n)$ = biaya dari *node* awal ke *node* n,

$h(n)$ = estimasi biaya dari *node* n untuk mencapai *node* akhir.

2.5 Algoritma Dijkstra

Algoritma Dijkstra adalah Algoritma yang digunakan untuk mencari jalur terpendek pada suatu grafik atau jalur yang terarah maupun tak terarah. Konsep Algoritma Dijkstra adalah mencari jalur terpendek dalam beberapa langkah dimana setiap langkah nantinya dipilih mana yang memiliki nilai yang kecil, dan memasukkannya kedalam susunan solusi[15]. Algoritma Dijkstra populer untuk menyelesaikan masalah pencarian jalur terdekat. Algoritma Dijkstra memiliki jalur yang mirip dengan Algoritma *greedy*; itu dimulai di titik awal dan terus menghubungkan *node* hingga mencapai titik akhir. Bilangan dibandingkan dari titik awal ke setiap *node*. Algoritma Dijkstra kemudian menemukan jalur terpendek antara dua jalur yang terkait[16]. Algoritma Dijkstra digunakan mencari jalur terpendek antara dua *node* dengan cara mematikan prosesnya ketika telah sampai ke titik tujuan. Algoritma Dijkstra juga dikenal dengan sebutan *uniform cost search*, yang tentunya sebagai contoh dari Algoritma *Best-first Search*[17].

2.6 ISO 25010

ISO (*International Organization for Standardization*) 25010 adalah standar yang memperbarui model ISO 9126 (2007). Ada delapan *sub*-fitur dan karakteristik utama. Berbasis ISO 9126, yang tujuan utamanya adalah untuk memandu dalam pengembangan produk perangkat lunak dalam hal spesifikasi dan evaluasi kebutuhan sistem[18], ilustrasi model ISO 25010 pada Gambar 2.6.

1. *Functional Suitability* menguji apakah aplikasi sudah sesuai fungsi sudah tepat dengan yang diharapkan.

2. *Performance efficiency* menguji berapa jumlah sumber daya yang digunakan dikondisi tertentu.
3. *Compability* menguji apakah aplikasi dapat bertukar informasi dengan perbedaan *hardware* atau *software* atau biasa disebut *crossplatform*.
4. *Usability* menguji apakah aplikasi dapat digunakan oleh pengguna.
5. *Reliability* menguji sejauh mana aplikasi menjalankan fungsi tertentu untuk jangka waktu tertentu.
6. *Security* menguji apakah aplikasi melindungi informasi dan data.
7. *Maintainability* menguji keefektivitas dan efisiensi aplikasi.
8. *Portability* menguji apakah aplikasi dapat berjalan di perangkat lunak yang berbeda.



Gambar 2.1 ISO 25010