

Implementasi Algoritma Route Planning untuk Game dengan Kontrol Gerak Berbasis Klik pada Arena Permainan

Muhammad Alfansya – 13523005

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: muhammadalfansya@gmail.com , 13523005@std.stei.itb.ac.id

Abstract— Permainan online telah menjadi salah satu bentuk hiburan yang paling diminati di era digital saat ini, dengan perkembangan teknologi yang pesat. Salah satu aspek penting dalam permainan online adalah kontrol pergerakan karakter, di mana sistem berbasis klik semakin populer. Dalam sistem ini, pemain dapat menggerakkan karakter dengan mengklik tujuan di layar, yang memerlukan algoritma pathfinding yang efisien untuk memastikan navigasi yang optimal. Makalah ini membahas implementasi algoritma pathfinding, termasuk A*, Uniform Cost Search (UCS), Greedy Best-First Search (GBFS), dan Breadth-First Search (BFS), dalam konteks permainan DOTA 2. Melalui simulasi berbasis grid 2D, analisis dilakukan untuk mengevaluasi kinerja masing-masing algoritma dalam berbagai skenario pergerakan.

Keywords— *Astar, BFS, Dota, GBFS, kontrol, Path, UCS*

I. PENDAHULUAN

Permainan online telah menjadi salah satu bentuk hiburan yang paling populer di era digital saat ini. Sejak awal pengembangannya, permainan online telah mengalami perkembangan yang signifikan, baik dari segi teknologi maupun desain. Pada awalnya, permainan online sederhana hanya dapat diakses melalui koneksi dial-up, namun kini, dengan kemajuan teknologi internet dan perangkat keras, permainan online dapat diakses dengan mudah melalui berbagai platform, termasuk konsol, PC, dan perangkat mobile.

Jenis-jenis permainan online sangat beragam, mencakup berbagai genre seperti aksi, petualangan, strategi, dan role-playing games (RPG). Platform permainan juga berperan penting dalam menentukan cara pemain berinteraksi dengan permainan. Permainan berbasis konsol biasanya dikontrol menggunakan stik, sementara permainan berbasis PC umumnya menggunakan kombinasi keyboard dan mouse. Di sisi lain, permainan mobile mengandalkan kontrol berbasis sentuhan, yang memungkinkan pemain untuk berinteraksi dengan permainan melalui layar sentuh, baik dengan menggunakan analog virtual maupun kontrol sentuh lainnya.

Salah satu metode kontrol pergerakan karakter yang semakin populer dalam permainan online adalah sistem berbasis klik. Dalam sistem ini, pemain dapat menggerakkan karakter dengan

mengklik tujuan di layar, baik menggunakan mouse di PC maupun dengan menyentuh layar di perangkat mobile. Metode ini menawarkan kemudahan dan kenyamanan, karena pemain tidak perlu mengarahkan setiap gerakan karakter secara manual, seperti yang dilakukan dengan kontrol analog atau tombol arah. Sebagai gantinya, karakter akan secara otomatis mencari jalan menuju tujuan yang telah ditentukan oleh pemain.

Namun, implementasi sistem kontrol berbasis klik ini memerlukan algoritma pathfinding yang efisien untuk memastikan karakter dapat menemukan jalur yang optimal menuju tujuan tanpa terjebak atau mengalami kesulitan dalam navigasi. Dalam makalah ini, penulis akan membahas implementasi algoritma pathfinding yang dapat digunakan dalam permainan dengan kontrol gerak berbasis klik, penulis akan menggunakan contoh permainan online DOTA 2.



Gambar 1 : DOTA 2

Sumber : (<https://www.dota2.com/hero/sniper>)

II. LANDASAN TEORI

A. Algoritma Pathfinding

Algoritma *pathfinding* adalah algoritma untuk menentukan rute optimal dari suatu titik awal menuju titik tujuan dalam suatu ruang. Secara umum, Algoritma pathfinding secara umum

dapat diklasifikasikan menjadi dua kelompok yaitu informed search dan uninformed search

1. Uninformed Search

Uninformed search, yang juga dikenal sebagai blind search, adalah penentuan rute tanpa mengetahui lokasi tujuan dan seluruh jalan yang dapat dilalui sebelum sampai ke titik yang berhubungan dengannya. Metode ini hanya mengandalkan informasi yang diperoleh selama proses pencarian berlangsung secara real-time. Karena informasi yang terbatas, uninformed search akan mencari ke berbagai arah hingga menemukan titik tujuan. Beberapa contoh algoritma uninformed search adalah :

a) Breadth-first search

Algoritma ini bekerja dengan prinsip ekspansi level per level, dimana seluruh simpul pada level tertentu akan dieksplorasi terlebih dahulu sebelum turun ke level berikutnya. Implementasi nya menggunakan struktur data *queue*. Proses dimulai dengan memasukkan simpul awal menjadi isi *queue* pertama, *queue* diproses secara iteratif dengan mengeluarkan antrian terdepan dan memasukkan tetangga simpul tersebut yang belum dikunjungi ke *queue* untuk diproses pada iterasi berikut nya. Keunggulan utama BFS adalah kemampuannya untuk selalu menemukan solusi optimal (jalur terpendek) pada graf tak berbobot, meskipun mungkin memerlukan sumber daya komputasi yang besar untuk graf dengan ukuran signifikan.

b) Uniform cost search

Merupakan perluasan dari BFS yang mampu menangani graf berbobot. Dalam UCS path dianggap sama dengan cost, menggunakan $g(n)$ yang merepresentasikan jarak dari root ke n . Mulai dengan memasukkan node start ke open set, masukkan neighbor yang memiliki cost lebih rendah ke open set, teruskan pencarian dengan node yang diproses adalah node yang di-pop dari open set di tiap iterasi. Tidak seperti BFS yang hanya menjamin optimalitas pada graf tak berbobot, UCS menjamin solusi dengan biaya total minimum dalam graf berbobot.

2. Informed Search

Berbeda dengan uninformed search, metode informed search memanfaatkan pengetahuan tambahan tentang lingkungan pencarian dalam bentuk fungsi heuristik. Fungsi ini memberikan estimasi perkiraan jarak menuju tujuan, memungkinkan algoritma untuk mengambil keputusan cerdas tentang arah pencarian yang paling menjanjikan. Hasilnya, informed search umumnya lebih efisien karena dapat memfokuskan pencarian ke arah tujuan tanpa harus mengeksplorasi seluruh ruang

pencarian. Beberapa contoh algoritma informed search adalah :

a) Greedy Best-first Search.

GBFS menerapkan pendekatan greedy dengan cara mengekspansi simpul yang terlihat lebih dekat ke tujuan, menggunakan fungsi evaluasi $f(n) = h(n)$ dengan $h(n)$ adalah biaya estimasi dari n ke tujuan, salah satu $h(n)$ yang bisa digunakan adalah garis lurus dari node n ke tujuan. Dalam konteks game, GBFS cocok untuk situasi dimana kecepatan lebih diutamakan daripada optimalitas jalur, atau ketika biaya komputasi menjadi perhatian utama.

b) A*

Algoritma ini menghindari jalur yang sudah memiliki cost mahal, menggunakan fungsi evaluasi $f(n) = g(n) + h(n)$, dengan $g(n)$ adalah biaya sejauh ini untuk mencapai simpul n , $h(n)$ adalah biaya estimasi dari simpul n ke tujuan. Algoritma A* menggabungkan algoritma UCS ($f(n) = g(n)$) dan algoritma GBFS ($f(n) = h(n)$)

B. Konsep Dasar Kontrol Permainan berbasis Klik (DOTA 2)

1. Karakter hero

Karakter yang dikontrol pemain selama permainan berlangsung disebut hero. Setiap hero dirancang dengan set kemampuan unik dan karakteristik statistik berbeda yang memungkinkan beragam strategi permainan. Dalam satu match permainan DOTA 2 terdiri dari dua tim yang masing masing tim memiliki 5 pemain dengan 5 hero pilihan nya. Pemain bisa menggerakkan hero dengan cara mengklik kanan titik tujuan di map dan hero akan mencari jalur ke titik tersebut.

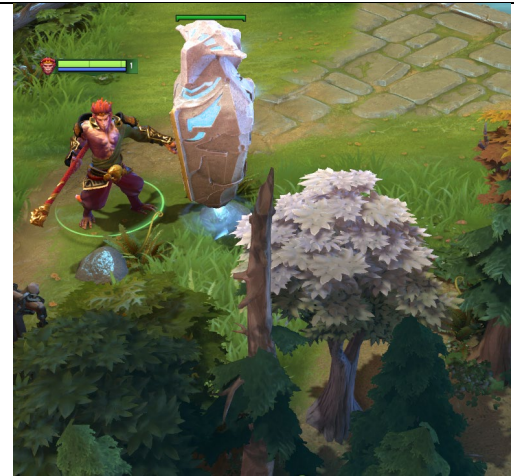
2. Hambatan dan Deteksi Tabrakan

DOTA 2 memiliki konsep collision detection yang berarti antara objek di dalam arena permainan tidak bisa saling menembus. Hero yang digerakkan pemain akan mencari jalur menuju tujuan dengan menghindari menabrak hambatan, beberapa objek yang menjadi hambatan antara lain :



Gambar 2 : Bangunan Barak di DOTA 2

Sumber : Dokumentasi penulis



Gambar 2 : Hero, Tower, dan Pohon dalam permainan DOTA 2

Sumber : Dokumentasi penulis

a) Struktur Bangunan.

Terdapat struktur bangunan di dalam permainan DOTA 2 yang menjadi objektif yang harus dihancurkan untuk memenangkan permainan. Struktur bangunan yang ada di DOTA 2 antara lain Ancient, Tower, Barak, Outpost, Watchers, dan tembok tembok sebagai ornamen arena permainan

b) Tree. Di arena permainan DOTA 2 terdapat tempat tempat dimana terdapat pohon yang tumbuh, area dengan pohon banyak disebut dengan area 'jungle'. Pemain dapat menghancurkan pohon dan bisa melewati titik tempat pohon itu tumbuh sebelum nya dan pohon tersebut akan tumbuh lagi dalam rentang beberapa waktu

c) Objek Dinamis

Selain objek objek statis tersebut, terdapat juga karakter yang bergerak secara dinamis seperti hero hero yang dikontrol pemain lain, baik sekutu maupun lawan, creep masing masing tim, dan creep netral yang terdapat di area hutan.

III. PEMBAHASAN

Untuk membahas algoritma route planning yang digunakan pada permainan DOTA 2, penulis membuat program pathfinder sederhana berbasis grid yang bisa membuat path berdasar algoritma A*, UCS, GBFS dan BFS

A. Desain Simulator

Simulator menggunakan representasi grid 2D dengan spesifikasi sebagai berikut :

- Dimensi grid 50 x 50 sel
- Setiap sel merepresentasikan area unit dalam game
- Warna sel :
 - Hijau : titik awal
 - Ungu : titik tujuan
 - Hitam : Obstacle
 - Kuning : jalur yang ditemukan
 - Merah : Area yang sudah dieksplorasi dan sudah diproses algoritma
 - Biru : Area yang sudah dieksplorasi dan belum dieksplorasi algoritma

B. Analisis Skenario Pergerakan

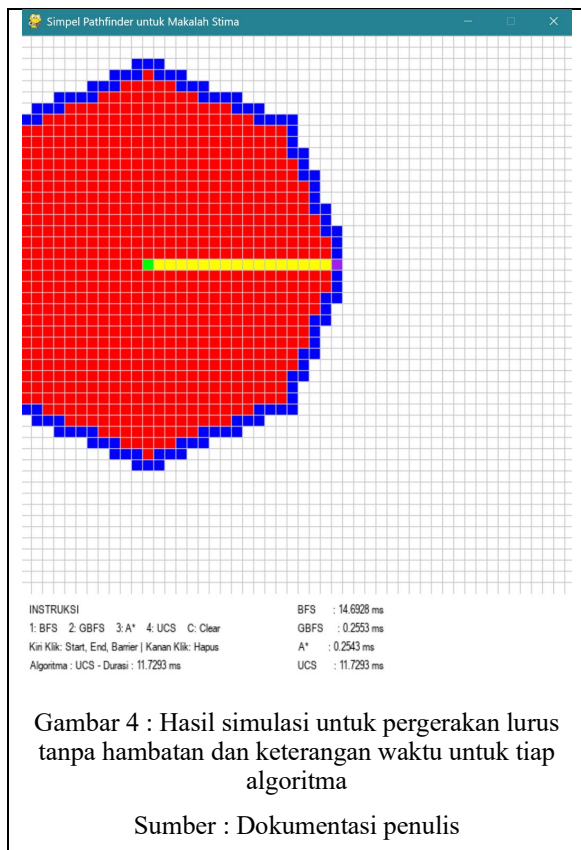
a) Pergerakan Lurus Tanpa Hambatan



Gambar 3 : Pergerakan Lurus tanpa Hambatan

Sumber : Dokumentasi penulis

Hasil Simulasi :



- Semua algoritma menemukan path lurus optimal
- A* dan GBFS paling efisien dengan waktu penjelajahan tercepat, 0.2553 ms dan 0.2543ms
- BFS dan UCS menjelajah lebih banyak node dan memakan waktu lebih lama yaitu 14.6928ms dan 11.7293ms

b) Pergerakan dengan Sedikit Hambatan



Gambar 5 : Pergerakan dengan sedikit hambatan

Sumber : Dokumentasi penulis

Hasil Simulasi :



- Semua algoritma menemukan jalur melengkung mengitari obstacle
- GBFS dan A* memiliki waktu penjelajahan lebih cepat dari BFS dan UCS tapi GBFS menempuh jalur yang terkadang lebih panjang dari A*
- BFS dan UCS menjelajah grid terlalu luas sehingga memiliki waktu penjelajahan lebih lama tapi memiliki jalur yang optimal

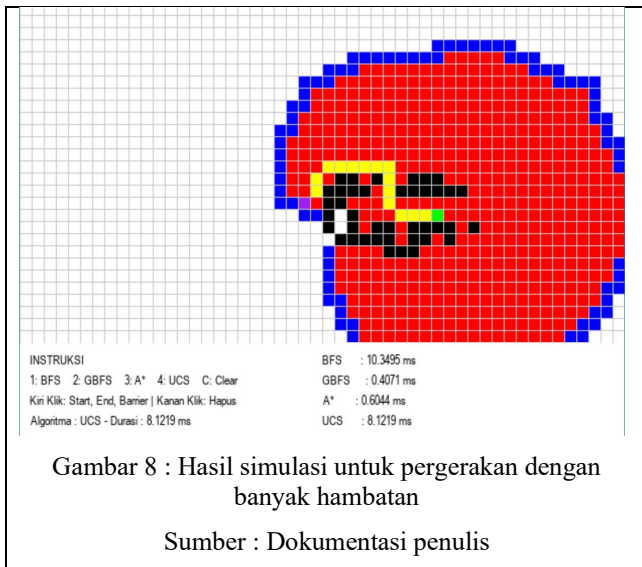
c) Pergerakan dengan Banyak Hambatan



Gambar 7 : Pergerakan dengan banyak hambatan

Sumber : Dokumentasi penulis

Hasil Simulasi :



- Semua jalur menemukan jalur disela hambatan
- GBFS dan A* memiliki waktu penjelajahan paling singkat, tapi GBFS menempuh jalur yang kurang optimal dibanding A*\
- BFS dan UCS menjelajah grid terlalu luas sehingga memiliki waktu penjelajahan lebih lama tapi memiliki jalur yang optimal

IV. KESIMPULAN

Algoritma A* dan Greedy Best First Search (GBFS) menunjukkan kinerja yang dalam hal waktu penjelajahan, terutama dalam skenario pergerakan lurus dan dengan sedikit hambatan. Keduanya mampu menemukan jalur optimal dengan waktu yang lebih singkat dibandingkan dengan algoritma lain seperti Breadth-First Search (BFS) dan Uniform Cost Search (UCS).

Dalam skenario pergerakan dengan banyak hambatan, A* tetap menjadi pilihan terbaik karena tidak hanya cepat, tetapi juga menghasilkan jalur yang lebih optimal dibandingkan GBFS, yang terkadang memilih jalur yang lebih panjang.

Meskipun GBFS lebih cepat dalam beberapa kasus, A* selalu berhasil menemukan jalur terpendek dengan waktu yang masih terbilang sangat cepat, menjadikannya algoritma yang lebih dapat diandalkan untuk situasi di mana optimalitas jalur sangat penting.

Penggunaan A* menjadi pilihan terbaik untuk implementasi dalam sebuah permainan dengan kontrol gerak berbasis klik karena kombinasi efisiensi dan optimalitasnya.

UCAPAN TERIMAKASIH

Penulis mengucapkan terimakasih kepada Tuhan Yang Maha Esa sudah memberikan kekuatan dan daya untuk

mengerjakan tugas makalah ini dan bertahan selama ini. Penulis juga berterimakasih kepada Ibu Nur Ulfa Maulidevi sebagai dosen pengajar Strategi Algoritma yang sudah memberi pengetahuan dan pemahaman tentang materi Path Planning sehingga penulis bisa menyelesaikan makalah ini.

Penulis berterimakasih kepada orang-orang terdekat penulis, Ifalisha Rahayu Zaki Nabilla, teman dan sahabat, youtuber dan streamer DOTA 2, Rusman, yang sudah memberikan inspirasi judul untuk makalah ini.

REFRENSI

- [1] Rinaldi Munir(2021). "Penentuan Rute (Route/Path Planning Bagian 1: BFS, DFS, UCS, Greedy Best First Search". informatika.stei.itb, Rinaldi Munir, 2021. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/21-Route-Planning-\(2025\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/21-Route-Planning-(2025)-Bagian1.pdf). Diakses pada 22 Junl 2025.
- [2] Munir, Rinaldi (2021). "Penentuan Rute (Route/Path Planning) Bagian 2: Algoritma A* Bahan Kuliah IF2211 Strategi Algoritma". [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/22-Route-Planning-\(2025\)-Bagian2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/22-Route-Planning-(2025)-Bagian2.pdf). Diakses pada 22 Junl 2025.
- [3] UIN Malang. "Analisis Algoritma Pathfinding dalam Permainan DOTA 2." <http://repository.uin-malang.ac.id/13288/> Diakses pada 22 Juni 2025.
- [4] Liquipedia. "Dota 2 Heroes: Introductions." <https://liquipedia.net/dota2/Heroes/Introductions> Diakses pada 22 Juni 2025.
- [5] Liquipedia. "Dota 2 Buildings: Neutral Buildings." https://liquipedia.net/dota2/Buildings/Neutral_Buildings Diakses pada 22 Juni 2025.
- [6] Liquipedia. "Dota 2 Trees." <https://liquipedia.net/dota2/Trees> Diakses pada 22 Juni 2025.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Juni 2025

Muhammad Alfansya - 13523005