

PENERAPAN HEURISTIK PADA ALGORITMA A-STAR UNTUK MENOPTIMALKAN PENELUSURAN LOKASI DI SATU KAWASAN (STUDI KASUS KAWASAN UNIVERSITAS DARMA PERSADA)

Herianto¹, Muhammad Akbar²

^{1,2}Program Studi Teknologi Informasi Universitas Darma Persada

*Koresponden : heri.unsada@gmail.com

ABSTRAK

Penelusuran (*searching*) lokasi di suatu kawasan sangat penting untuk mempercepat penemuan jalan menuju ke lokasi tertentu, dalam konteks penelitian ini lokasinya terbatas di satu kawasan atau lingkungan yang terbatas. Secara umum sebenarnya sudah tersedia aplikasi map semacam google map yang dapat digunakan untuk menelusuri jalan dari satu titik lokasi ke lokasi lain. Tetapi aplikasi google map atau semacamnya digunakan untuk lingkup yang sangat luas dan seringkali kurang detail untuk daerah-daerah gedung dan ruang tertentu di suatu lingkungan atau kawasan tadi. Untuk itu dibutuhkan aplikasi khusus untuk membantu tamu atau orang baru saat berkunjung ke lingkungan baru (misal kampus) agar dapat mencapai titik/tempat tertentu (gedung dan ruang) dengan jalur perjalanan (*path*) yang paling efisien. Algoritma A* (*A-star*) adalah algoritma pencarian AI (*Artificial Intelligence*) yang dapat menelusuri path terbaik berdasarkan heuristik (nilai dari pengalaman) tertentu berdasarkan kasusnya. Pada penelitian ini dilakukan kajian untuk menemukan rumusan heuristic terbaik yang diterapkan pada algoritma A* untuk menelusuri dan mendapatkan jalur (*path*) terbaik dari satu titik lokasi ke lokasi lain pada lingkungan yang terbatas, dalam hal ini kasus yang digunakan adalah lingkungan kampus Universitas Darma Persada. Dari hasil kajian diperoleh rumusan heuristic yang dapat meningkatkan kinerja sebesar 23 %. Kinerja dihitung dengan membandingkan hasil pemanfaatan heuristic dengan tanpa memanfaatkan heuristic. Pada penelitian ini juga dibangun aplikasi yang bersifat *user-friendly* menggunakan Unity 3D berupa animasi dan simulasi perjalanan dari titik awal di kawasan kampus menuju titik tujuan.

Kata Kunci : *A-star, heuristic, Kawasan terbatas, AI search*

1. PENDAHULUAN

Mengutip (Herianto, 2022) bahwa Artificial intelligence dapat digunakan untuk membantu pencarian solusi masalah atau kasus pencarian (penelusuran) jalur dari titik masuk (*start*) ke titik keluar (*goal*) yaitu menggunakan metoda pencarian ruang keadaan (*state space searching*). Metoda pencarian ruang keadaan ini secara umum ada 2 (dua) jenis yaitu pencarian buta (*blind search*) dan pencarian heuristic (*informed search*).

Pada pencarian buta tidak digunakan informasi tertentu saat proses pencarian solusi terjadi, yang digunakan hanya mekanismenya saja, apakah proses pencarian mengutamakan daerah secara melebar (*Breadth First Search, BFS*) atau mengutamakan kedalamannya (*Depth First Search, DFS*).

Pada metode pencarian heuristic, saat proses penelusuran terjadi selain dengan mekanisme seperti pada *blind search*, juga mempertimbangkan (menggunakan) informasi khusus yang diharapkan dapat mempercepat penemuan solusi yang optimal. Contoh metoda ini adalah hill climbing, best first, A algoritma, A* algorithm, dan sebagainya.

Baik pada metoda blind search maupun heuristic search, struktur data yang digunakan adalah tree, dimana setiap titik dari tree disebut juga dengan node (simpul). Maka, proses pencarian adalah menelusuri setiap node dan menguji apakah node tersebut merupakan simpul tujuan (goal) atau tidak. Jika merupakan goal maka proses penelusuran berhenti, solusi ditemukan, dan jika node bukan goal maka proses penelusuran dilanjutkan dengan penentuan node berikutnya yang diuji tergantung pada pilihan algoritma yang digunakan. Jika yang dipilih adalah BFS maka node berikutnya adalah mengutamakan node yang se level di sampingnya, jika yang dipilih adalah DFS maka node berikutnya adalah yang berada di level berikut dari cabang yang sama. Metoda BFS dan DFS disebut buta karena tidak menggunakan informasi lain selain mekanisme di atas, tetapi cukup mudah diimplementasikan karena tidak perlu memikirkan fungsi heuristiknya. Sementara pada metoda pencarian heuristic (*informed search*) selain mekanisme juga dipertimbangkan nilai heuristic yang diperoleh dari rumus heuristic. Rumus heuristic ini tergantung pada kasusnya, karena kasus yang berbeda memiliki pertimbangan yang berbeda dalam menyatakan tujuan yang optimal (*optimal goal*). Misal pada kasus pencarian jarak terdekat (shortest path problem) maka kriteria optimalnya adalah nilai yang terkecil dari setiap path (jalur) yang dilalui, pada kasus perhitungan profit (keuntungan) penjualan maka kriteria optimal adalah keuntungan yang lebih besar, pada kasus travelling sales problem adalah jarak minimum mengunjungi semua node, dan begitu seterusnya untuk kasus-kasus lain. Jadi fungsi heuristic yang baik akan diperoleh dengan memahami domain permasalahan dan pilihan algoritma yang akan digunakan.

Pada sistem heuristic metoda state space search, ada 2 (dua) informasi khusus yang biasa digunakan. Yang pertama biasa disebut fungsi $g(n)$ yaitu nilai path yang sudah dilalui. Nilai ini sifatnya lebih pasti karena masa lalu atau sudah terjadi. Yang kedua adalah nilai $h(n)$, yaitu nilai prediksi, yaitu estimasi dari node saat ini berapa nilai path yang diperoleh untuk mencapai node tujuan. Algoritma state space search yang hanya mempertimbangkan nilai $g(n)$ disebut dengan dengan Uniform Cost Search (UCS), algoritma yang hanya mempertimbangkan nilai $h(n)$ disebut dengan A Algoritma, dan algoritma yang memperitimbangkan nilai $g(n)$ sekaligus $h(n)$ disebut dengan A* (dibaca A star) Algorithm.

Kasus penelusuran lokasi bertujuan memperoleh path dari satu titik lokasi ke titik lokasi lain (tujuan), tentu saja path yang dicari adalah yang paling optimal (paling kecil jarak tempuhnya). Untuk satu kawasan biasanya ada tempat-tempat yang lebih detail yang tidak ter-record pada jenis aplikasi *public-map* seperti google map dan sebagainya, sehingga ada tantangan untuk membangunnya secara mandiri. Tantangannya adalah menentukan algoritma penelusuran terbaik, mendesain rumus heuristic sesuai domain dan menyediakan user interface yang mudah digunakan oleh end-user seperti tamu atau pengunjung baru di Kawasan tersebut.

2. METODOLOGI

2.1 Struktur Tahapan Penelitian

Penelitian yang dilakukan menggunakan struktur dan tahap seperti berikut :



Gambar 1. Tahapan Penelitian

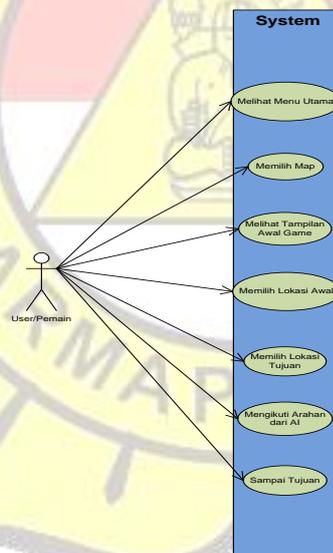
2.2 Metode Perancangan Sistem

Metode perancangan system menggunakan pemodelan UML (Unified Modeling Language) terutama Use Case Diagram dan activity diagram. Kedua diagram digunakan karena yang pertama (use case) dapat digunakan untuk merancang hubungan antara fungsional (fitur) sistem dan user yang menggunakan fitur tersebut, sedangkan yang kedua (activity) digunakan untuk memperlihatkan detail proses setiap fitur yang disediakan.

3. HASIL

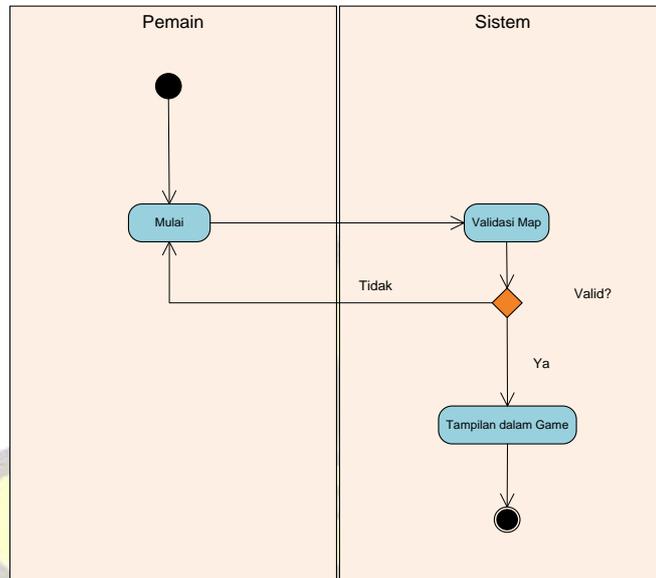
3.1 Perancangan Sistem

Rancangan Use Case Diagram

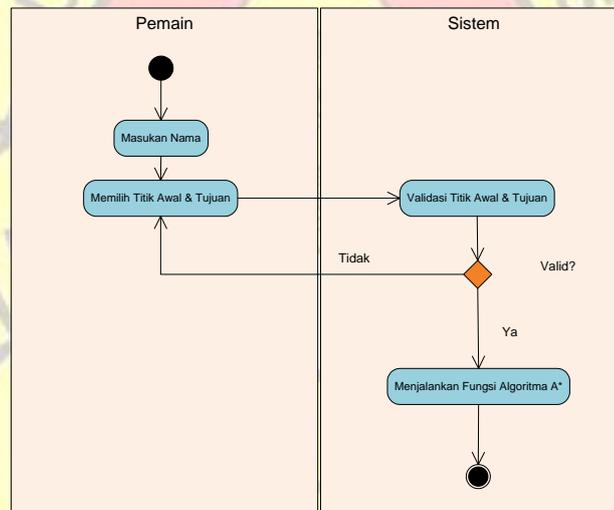


Gambar 2. Use Case Diagram

Rancangan Activity Diagram



Gambar 3. Activity Diagram Fitur Pengguna



Gambar 4. Activity Diagram Fitur Penelusuran

3.2 Interface Aplikasi

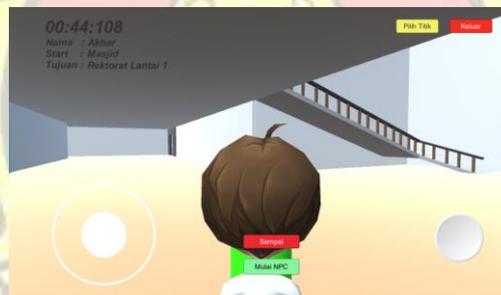
Interface utama Aplikasi :



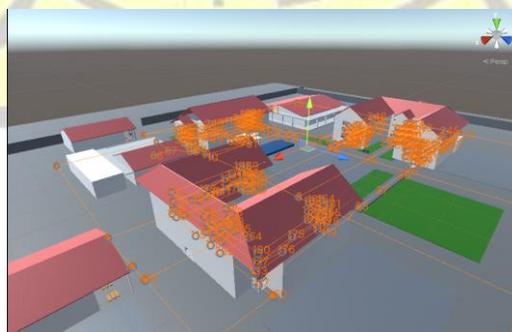
Gambar 5. Tampilan Menu Awal Fitur Penelusuran oleh User



Gambar 6. Tampilan GUI Saat Pengguna berada di titik Awal



Gambar 7. Tampilan GUI Saat Pengguna berada di titik tujuan



Gambar 8. Tampilan GUI Denah Kampus Unsada dalam satu kawasan

4. PEMBAHASAN

Sistem heuristic yang dirancang dapat dijelaskan melalui flow chart berikut :



Gambar 9. Flow chart Perhitungan Heuristik

Dari flow chart tampak bahwa system dimulai dengan menghitung jarak antar goal, kemudian masing-masing jarak tersebut ditambahkan dengan metode permutasi dari goal-goal yang ada sehingga menghasilkan susunan arah yang memiliki total jarak minimum. Setelah mendapatkan masing-masing susunannya sesuai dengan masing-masing goal, ditambahkan dengan jarak garis lurus dari node suksesor ke masing-masing goal. Setelah itu dipilih jarak yang paling kecil untuk di jadikan nilai heuristik. Untuk menentukan mana jalur yang selanjutnya akan dipilih adalah jalur yang berada di array Open dan memiliki f-cost minimum. Penghitungan f-cost dilakukan dengan mengkombinasikan jarak garis lurus dengan gambaran seperti berikut :

1	8 ↑ 2 3 5	8 ↑ 3 4 4	8 → 4 5 3	8 → 5 6 2	8 → 6 7 1	7
10 ↑ 8 3 7	6 ↑ 9 2 4	6 ↑ 10 3 3	11	8 ↓ 12 7 1	8 Tujuan 13 8 0	14
8 ↑ 15 2 6	6 ↑ 16 1 5	6 ↑ 17 2 4	18	19	20	21
8 ← 22 1 7	NPC 23	6 → 24 1 5	25	26	27	28
29	30	8 ↓ 31 2 6	32	33	34	35
36	37	38	39	40	41	42

Gambar 10. Perhitungan Grid

Keterangan Gambar 10 :

- g = cost titik npc ke kotak sebelah (dapat bergerak ke atas, ke kiri, ke kanan, dan kebawah).
- h = perkiraan cost dari kotak sebelah ke titik tujuan dengan mengabaikan obstacle. Posisi nilai F , G , dan H

Pengujian Nilai Penemuan Path

Dari hasil membangun sistem di atas dilakukan beberapa pengujian seperti berikut :

Pengujian yaitu dengan mencatat waktu tempuh dan banyak node yang dilalui dari salah satu titik awal (misal dari titik Pos Satpam ke Lantai 1 Teknik/Kelautan). Sebagai pembandingan dan melihat pengaruh heuristic maka dilakukan dua kali pencatatan yaitu tanpa heuristic dan dengan heuristic. Hasil pengujian tersebut adalah sebagai berikut :

Tabel 1. Tabel hasil pengujian pengaruh heuristic

Percobaan Ke-	Rute	Tanpa Heuristik	Dengan heuristic	Selisih	% Kinerja
1	Pos Satpam ke Lantai 1 Teknik/Kelautan	13	11	2	18 %
2	Kantin Sastra ke Kantin Teknik	15	12	3	25 %
3	Lantai 1 Rektorat ke Perpustakaan	10	10	0	0 %
4	Lantai 1 Teknik/Kelautan ke Ruang Dekan Teknik	9	6	3	50 %
5	Lantai 1 Teknik/Kelautan ke Grha Wira Bakti	8	6	2	33,3 %
6	Lantai 1 Teknik/Kelautan ke Kajur Teknik Perkapalan	10	6	4	66,6 %
7	Ruang Dekan Teknik ke Pasca Sarjana	11	9	3	33,3 %
8	Ruang MKU Lantai 1 Sastra Ekonomi ke Ruang Dekan Sastra	11	9	3	33,3 %

Dari tabel di atas tampak bahwa untuk semua pengujian diperoleh bahwa hasil dengan pemanfaatan heuristic selalu lebih optimal (jaraknya lebih kecil). Dari 7 (tujuh) kali percobaan rata-rata kinerja dari heuristic adalah total % kinerja dibagi dengan jumlah percobaan = 23%. Artinya penggunaan heuristic pada algoritma A-star untuk kasus pencarian jalur terdekat lokasi Kawasan kampus universitas Darma Persada dapat meningkatkan kinerja sebesar 23%.

5. KESIMPULAN

Algoritma A-star yaitu algoritma dengan heuristic dapat diimplementasikan untuk mengoptimalkan hasil penelusuran jalur dari satu titik ke titik lain di satu Kawasan terbatas

dalam hal ini lingkup kampus Universitas Darma Persada. Hasil pengujian dan perhitungan peningkatan kinerja untuk kasus ini sebesar 23%. Peningkatan kinerja penggunaan heuristik diperoleh berdasarkan domain permasalahan pada penelitian ini yang jika diterapkan ke domain permasalahan lain akan memiliki peluang besar juga terjadi peningkatan kinerja.

DAFTAR PUSTAKA

1. Fiqih Fauzan Ar-rafi, Achmad Udin Zailani, 2021, **Rancang Bangun Game Edukasi Sejarah Berbasis Android Menggunakan Game Engine Unity 3D Android Based Historical Educational Game Design With Unity 3D Game Engine**, Prosiding Seminar Nasional Informatika Dan Sistem Informasi 3:277–86.
2. Ginting, Budi Serasi, and Fajar Ramadhan, 2018, **Perancangan Game Become a King Berbasis**, Manajemen Informatika & Komputerisasi Akuntansi 2(1):12–21.
3. H Subakti, I Romli, ST Nur Syamsiyah, AA Budiman, H herianto, 2022, **Artificial Intelligence**, Media Sains Indonesia
4. H Subakti, M Kraugusteeliana, RH Purabaya, S Pd Herianto, 2022, **Interaksi Manusia dan Komputer**, Media Sains Indonesia
5. Kustriawan, Noval, 2018, **Pembuatan Game Bamboo Menggunakan Aplikasi Unity 3D**, Journal Of Applied Multimedia and Networking Vol 2 No 2:3–7.
6. Rohmawati, Indah, 2019, **Pengembangan Game Edukasi Tentang Budaya Nusantara 'Tanara' Menggunakan Unity 3D Berbasis Android**, Jurnal SITECH : Sistem Informasi Dan Teknologi 2(2):173–84. doi: 10.24176/sitech.v2i2.3907.
7. Safira, Linda, Paulus Harsadi, and Sri Harjanto, 2021, **Penerapan Navmesh Dengan Algoritma A Star Pathfinding Pada Game Edukasi 3d Go Green**, Jurnal Teknologi Informasi Dan Komunikasi (TIKoSIN) 9(1):17. doi: 10.30646/tikomsin.v9i1.540.
8. Safitra, Wahyu, Ahmad Faisol, and Suryo Adi Wibowo, 2020, **Application of the Finite State Machine Method to Non Player Character (NPC) Action Strategy Game 'Ouroboros'**, JATI (Jurnal Mahasiswa Teknik Informatika) 4(2):292–97. doi: 10.36040/jati.v4i2.2828.
9. Supardi, Reno, 2017, **Pembuatan Game Balap Kelinci Dengan Unity Berbasis Android**. Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi 7(1):10.
10. Zebua, Taronisokhi, Berto Nadeak, and Soni Bahagia Sinaga, 2020, **Pengenalan Dasar Aplikasi Blender 3D Dalam Pembuatan Animasi 3D**, Jurnal ABDIMAS Budi Darma 1(1):18–21