Jurnal Riset Mahasiswa Matematika
Volume 4(2) (2024), Pages 82-96
p-ISSN: 2808-1552; e-ISSN: 2808-4926

Pengembangan Algoritma A-Star dengan Penyesuaian Heuristik untuk Pencarian Rute Optimal

Achmad Faiz Sukroni*, Juhari , Ach. Nasichuddin

Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia

achmadfaiz490@gmail.com*, juhari@uin-malang.ac.id, nashichuddin@gmail.com

Abstrak

Algoritma *A-Star* adalah salah satu algoritma pencarian rute terpendek yang paling populer karena keefektifannya dalam menemukan solusi optimal. Penelitian ini berfokus pada modifikasi algoritma *A-Star* untuk meningkatkan performanya dalam mengoptimalkan rute terpendek. Studi dilakukan dengan membandingkan algoritma *A-Star* standar dan versi modifikasinya berdasarkan jumlah iterasi yang diperlukan untuk mencapai solusi optimal. Penilaian dilakukan secara manual dan melalui program komputer dengan menggunakan data sampel lokasi wisata di Blitar sebagai studi kasus. Hasil analisis dalam 20 percobaan peerbandingan menunjukkan bahwa algoritma *A-Star* yang telah dimodifikasi mampu meminimalkan jumlah iterasi hingga 32% dibandingkan algoritma *A-Star* standar. Temuan ini mengindikasikan bahwa modifikasi yang dilakukan pada algoritma *A-Star* dapat meningkatkan efisiensi tanpa mengurangi akurasi hasil. Penelitian ini memberikan kontribusi penting dalam pengembangan algoritma pencarian rute terpendek, khususnya dalam penerapannya pada sektor pariwisata dan sistem navigasi.

Kata kunci: *A-Star*; Algoritma Pencarian; Rute Terpendek; Modifikasi Algoritma, Optimalisasi.

Abstract

The The A-Star algorithm is one of the most popular shortest route search algorithms due to its effectiveness in finding optimal solutions. This study focuses on modifying the A-Star algorithm to improve its performance in optimizing the shortest route. The study was conducted by comparing the standard A-Star algorithm and its modified version based on the number of iterations required to reach the optimal solution. The assessment was carried out manually and through a computer program using sample data from tourist locations in Blitar as a case study. The results of the analysis in 20 comparative experiments showed that the modified A-Star algorithm was able to minimize the number of iterations by up to 32% compared to the standard A-Star algorithm. This finding indicates that the modifications made to the A-Star algorithm can increase efficiency without reducing the accuracy of the results. This study provides an important contribution to the development of shortest route search algorithms, especially in their application to the tourism sector and navigation systems.

Keywords: A-Star; Search Algorithm; Shortest Route; Algorithm Modification; Optimization.

PENDAHULUAN

Analisis graf (*graph analysis*) merupakan salah satu cabang dari sekian banyak ilmu matematika dan ilmu komputer yang mempelajari sifat-sifat dan struktur dari graf, serta mengembangkan algoritma dan metode untuk menganalisis dan memecahkan masalah yang melibatkan graf. Analisis graf terbagi pada beberapa bidang seperti analisis konektivitas (bertujuan menentukan apakah titik-titik dalam graf terhubung atau tidak), analisis sentralitas (digunakan untuk mengidentifikasi titik-titik yang paling penting atau berpengaruh dalam suatu graf), analisis klaster (digunakan untuk mengidentifikasi kelompok-kelompok atau klaster titik

Submitted: 5 Juni 2024 Reviewed: 11 September 2024 Accepted: 18 November 2024
Doi: http://dx.doi.org/10.18860/jrmm.v4i2.31208

yang memiliki kemiripan atau kedekatan dalam suatu graf), analisis jalur (analisis yang melibatkan jalur terpendek, jalur terpanjang, atau jalur dengan kriteria tertentu antara dua titik dalam graf), dan beberapa bidang analisis yang lainya. Pada penelitian ini menerapkan analisis jalur (*path analysis*) yang mana dalam konteks optimasi rute analisis ini digunakan untuk menemukan jalur terpendek, jalur terpanjang, atau jalur dengan kriteria tertentu antara dua titik dalam graf [1]. Salah satu metode dalam analisis jalur untuk optimasi rute yaitu menggunakan algoritma *A-Star*.

Algoritma *A-Star* adalah algoritma pencarian jalur (*path finding*) yang dikembangkan pada tahun 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Algoritma *A-Star* merupakan pengembangan dari algoritma *Dijkstra* dengan penambahan fungsi heuristik untuk mengoptimalkan proses pencarian . Jika *Dijkstra* menjamin penemuan jalur terpendek namun dengan cara mengeksplorasi semua kemungkinan node hanya menggunakan $g(n)$ biaya aktual saja, *A-Star* menggunakan estimasi jarak ke tujuan untuk memprioritaskan arah pencarian yang lebih menjanjikan [2]. Pendekatan ini membuat *A-Star* lebih efisien dalam hal waktu komputasi dan penggunaan memori, terutama pada kasus pencarian jalur dalam peta yang luas dan kompleks. Kelebihan ini menjadi dasar pemilihan algoritma *A-Star* sebagai metode pencarian jalur dalam penelitian ini. Algoritma ini didasarkan pada konsep teori graf, di mana masalah pencarian rute dapat direpresentasikan sebagai graf $G = (V, E)$ dengan V sebagai himpunan titik (*node*) dan E sebagai himpunan sisi (*edge*) yang menghubungkan titik-titik tersebut. Setiap sisi memiliki bobot atau biaya yang terkait. Algoritma ini menggunakan fungsi evaluasi $f(n) = g(n) + h(n)$, di mana $g(n)$ adalah biaya aktual dari titik awal ke titik n, dan $h(n)$ adalah perkiraan heuristik biaya termurah dari titik n ke tujuan akhir. Algoritma ini dapat diterapkan pada perencanaan rute navigasi, permainan berbasis peta, robotika, kecerdasan buatan AI, jaringan transportasi, dan sebagainya [3].

Implementasi algoritma *A-Star* telah dilakukan pada penlitian sebelumnya. Penelitian [4] dengan tema "Implementasi Penggunaan Algortima A* pada Penentuan Jarak Terpendek dari Cilacap ke Yogyakarta" berhasil menemukan rute tercepat dari Kota Cilacap ke Kota Yogyakarta dengan jarak 4.371,3 km, dan peneltian ini algoritma *A-Star* lebih efektif dibanding dengan metode yang lainya, dikarenakan telah diuji dengan meggunakan algoritma *searching* seperti *Greedy Best First Search* dengan titik awal dan tujuan yang sama menghasilkan jarak tercepat 4.574,3 km, dengan penghematan jarak oleh algoritma *A-Star* sejauh 203 km. Penelitian selanjutnya [5] dengan tema "Penerapan Algoritma A*Star Untuk Mencari Rute Terpendek Dari Kemayoran Ke Destinasi Monumen Nasional (MONAS)" mendapat hasil jarak optimal yakni 15,070 km. Penelitian selanjutnya [6] dengan tema "Implementation Of *A-Star* Algorithm In Finfing The Shortest Route Of Cooking Oil Distribution In Karo Regency Using Graph" dalam pencarian rute terpendek pada distribusi minyak goreng algoritma *A-star* dapat menemukan rute optimal yang mana dibagi menjadi dua rute dengan rute pertama sejauh 11.29 km dan rute kedua sejauh 30.95 km. Sedangkan rute perusahaan adalah 30.95 km, sehingga presentase penghematan jarak dengan algoritma *A-star* 62.16 %.

Penelitian selanjutnya [7] dengan tema "Perbandingan Algoritma *Dijkstra* dan Algoritma *A-Star* dalam Penentuan Lintasan Terpendek dari Dinas Pendidikan Provinsi Lampung ke Beberapa Sekolah Menengah Atas (SMA) Negeri di Provinsi Lampung" pada 30 titik tujuan didapatkan hasil yang sama dalam penentuan rute terpendek. Walau demikian dari perbandingan waktu proses dengan menggunakan bahasa pemrograman *Python* dari kedua algoritma tersebut, didapat rata-rata *running time* algoritma *Dijkstra* adalah 78, 575 ms dan hasil rata-rata *running time* algoritme *A-Star* adalah 47,505 ms. Dari nilai tersebut, dapat disimpulkan bahwa rata-rata running time algoritma *A-Star* lebih cepat 31,7 ms dibandingkan dengan algoritma *Djikstra.*

Pada penelitian ini peneliti ingin mengembangkan atau memodifikasi algoritma *A-Star* dengan harapan modifikasi algoritma *A-Star* dapat lebih optimal dari algoritma *A-Star* standar pada rute destinasi wisata. Permasalahan pencarian rute terpendek menjadi aspek krusial dalam optimalisasi pengalaman wisatawan, mengingat faktor waktu dan efisiensi perjalanan sangat mempengaruhi kepuasan pengunjung. Dalam konteks ini, penerapan teknologi navigasi yang tepat menjadi semakin penting, terutama dalam menghadapi dinamika perubahan kondisi lalu lintas dan situasi di lapangan.

Algoritma *A-Star* telah terbukti efektif dalam menyelesaikan permasalahan pencarian rute terpendek karena kemampuannya menggabungkan metode pencarian terarah (heuristik) dengan perhitungan jarak sebenarnya. Namun peneliti melihat celah akan algoritma ini masih bisa dikembangkan dan lebih optimal dengan memodifikasi algoritma *A-Star* dengan mengembangkan fungsi evaluasi pada algoritma *A-Star* tradisional.

Penelitian ini menerapkan ilmu dari konsep matematika teori graf dan algoritma pencarian jalur yang diaplikasikan pada masalah nyata di bidang pencarian rute terpendek. Dengan mengombinasikan ilmu matematika dan teknologi informasi, bertujuan untuk mengoptimalkan rute terpendek sehingga dapat meningkatkan efisiensi perjalanan dan mengurangi biaya. Dengan memodifikasi algoritma *A-Star* diharapkan dapat diperoleh algoritma *A-Star* yang lebih optimal, dengan membadingkan kedua algoritma tersebut pada penentuan rute terpendek destinasi wisata Blitar.

METODE

Data dan Sumber data

Penelitian ini menggunakan sumber data sekunder yang diperoleh dari *google map* dengan mengambil titik (*node*), titik koordinat, serta jarak antar titik dengan satuan kilometer. Dengan mengambil titik (*node*) sampel pada destinasi wisata Blitar. Dalam titik-titik sampel tersebut terdiri dari titik awal, titik-titik yang akan dipilih untuk dicari rute optimal, dan titik tujuan. Dari data tersebut akan diambil tiga titik awal dan tiga titik akhir sebagai sampel penelitian untuk perbandingan antara algoritma *A-Star* tradisional dengan modifikasi algoritma *A-Star* dengan manual, dan dua puluh titik awal dan dua puluh titik akhir menggunakan program.

Tabel 1 *Data Titik Sampel Kecamatan dan Wisata Blitar*

No	Titik (*node*)	Simbol	Titik Koordinat
1	Kebun Teh Sirah Kencong	$W2$	$(-7.9761°, 112.43014°)$
2	Gumuk Sapu Angin	$W3$	$(-8.02476°, 112.4243°)$
3	Rambut Monte	$W5$	$(-8.01237°, 112.039°)$
4	Doko	$K1$	$(-8.04568°, 112.41986°)$
5	Bendungan Lahor	$W6$	$(-8.14501°, 112.45102°)$
6	Selorejo	$K2$	$(-8.13352°, 112.42316°)$
7	Kesamben	$K3$	$(-8.14563°, 112.36573°)$
8	Selopuro	$K4$	$(-8.14034°, 112.31279°)$
9	Wlingi	$K5$	$(-8.07364°, 112.3264°)$
10	Gandusari	$K6$	$(-8.04564°, 112.30376°)$
11	Garum	$K7$	$(-8.04307°, 112.23021°)$
12	Candi penataran	$W7$	$(-8.01642°, 112.20974°)$
13	Nglegok	$K8$	$(-8.1241°, 112.2495°)$

14	Talun	$K9$	$(-8.1241°, 112.2495°)$
15	Kanigoro	$K10$	$(-8.11798°, 112.20272°)$
16	Sutojayan	$K11$	$(-8.16805°, 112.21737°)$
17	Kampung Coklat	$W8$	$(-8.15605°, 112.1702°)$
18	Blitar Park	$W9$	$(-8.08099°, 112.19602°)$
19	Makam Bung Karno	$W10$	$(-8.08461°, 112.17608°)$
20	Sanan Wetan	$K12$	$(-8.09918°, 112.1781°)$
21	Kepanjen Kidul	$K13$	$(-8.10288°, 112.1639°)$
22	Sanan Kulon	$K14$	$(-8.09945°, 112.13405°)$
23	Penangkaran Rusa Maliran	$W11$	$(-8.0673°, 112.12124°)$
24	Srengat	$K15$	$(-8.0629°, 112.06911°)$
25	Ponggok	$K16$	$(-8.00251°, 112.12145°)$
26	Udanawu	$K17$	$(-8.00351°, 112.03683°)$
27	Wonodadi	$K18$	$(-8.0437°, 111.99413°)$
28	Kademangan	$K19$	$(-8.19811°, 112.10594°)$
29	Goa Luweng	$W12$	$(-8.24644°, 112.06807°)$
30	Bakung	$K20$	$(-8.29064°, 112.1092°)$
31	Tambak	$W13$	$(-8.31541°, 112.14345°)$
32	Wonotirto	$K21$	$(-8.24858°, 112.15893°)$
33	Panggung Rejo	$K22$	$(-8.26417°, 112.25334°)$
34	Air Terjun Jurug Bening	$W14$	$(-8.27248°, 112.32052°)$
35	Binangun	$K23$	$(-8.23152°, 112.33673°)$
36	SPBU Garum	$K24$	$(-8.07625°, 112.22918°)$

Tahapan Penelitian

1. Menentukan data dari *Google Maps* berupa capture jalur yang akan dilewati.
2. Menentukan dan mengumpulkan koordinat dari titik-titik sampel destinasi wisata Blitar.
3. Modifikasi algoritma *A-Star* untuk mendapatkan metode yang lebih optimal.
4. Menghitung nilai heuristik $h(n)$.
5. Melakukan eksperimen pencarian rute terpendek untuk mendapatkan hasil perbandingan algoritma.

Algoritma *A-Star* Standar

Algoritma *A-star* adalah algoritma heuristik yang sangat meningkatkan efisiensi pencarian jalur dengan memasukkan informasi tentang titik target sambil mempertahankan pencarian jalur optimal . Fungsi evaluasi adalah sebagai berikut [8]:

$$f(n) = g(n) + h(n) \quad (1)$$

Keterangan:
$f(n)$: fungsi evaluasi titik n pada setiap simpul
$g(n)$: Akumulatif biaya aktual dari titik awal hingga titik n
$h(n)$: perkiraan biaya dari titik n ke titik tujuan, juga dikenal sebagai fungsi heuristik.
$h(n)$ ditentukan dengan melakukan perhitungan dengan rumus *Heuristic Euclidean Distance* (HED) [9]:

$$h(n) = \sqrt{\left(x_{goal} - x_n\right)^2 + \left(y_{goal} - y_n\right)^2} \quad (2)$$

Keterangan:
x_n : Nilai bujur/ longitude titik n
x_{goal} : Nilai bujur/ longitude titik tujuan
y_n : Nilai lintang/ latitude titik n
y_{goal} : Nilai lintang/ latitude titik tujuan
Untuk fungsi $g(n)$ sebagai berikut [10]:

$$g(n) = e(node_{start}, node_A) + e(node_A, node_n) \quad (3)$$

Keterangan:
$node_{start}$: Titik awal
$node_A$: Titik yang sedang dijalankan
$node_n$: Titik n
$e(node_{start}, node_A)$: Jarak aktual dari titik awal menuju titik yang sedang dijalankan
$e(node_A, node_n)$: Jarak aktual dari titik yang sedang dijalankan menuju titik n.

Beberapa terminologi dasar yang terdapat pada algoritma *A-Star* adalah

1. Starting point adalah sebuah terminologi untuk posisi awal sebuah titik.
2. *A* adalah simpul yang sedang dijalankan dalam algortima pencarian rute terpendek.
3. Titik (*node*) adalah petak-petak kecil sebagai simbol dari daerah pathfinding.
4. *Parent node* adalah titik yang berada satu tingkat di atas titik lain dan terhubung langsung. Titik ini yang menjadi acuan atau induk bagi titik-titik di bawahnya
5. *Open list* adalah tempat penyimpanan data titik yang dapat diakses dari titik awal maupun titik yang sedang dijalankan. Contoh: *Open_list* = $\{n_1, n_2, n_3\}$
6. *Closed list* adalah tempat penyimpanan data titik ($node$) sampai titik *A* yang merupakan bagian dari jalur terpendek. Contoh:
   Closed_list == $\{n_1, n_2, n_3, n_A\}$
7. Harga (*F*) adalah nilai yang diperoleh dari proses penjumlahan, nilai *g* merupakan jumlah nilai tiap titik dalam jalur terpendek dari titik awal ke *A*, dan *H* adalah jumlah perkiraan nilai dari sebuah simpul ke simpul tujuan.
8. Titik tujuan yaitu simpul yang menjadi tempat akhir perjalanan[11].

Langkah-langkah algoritma *A-star* adalah sebagai berikut:

1. Mulai pencarian dari titik awal n_{start}, dan tambahkan n_{start} ke "*open_list*" sebagai titik (*node*) yang akan dicari.
2. Cari semua titik n yang terhubung dengan titik awal n_{start}, dan tambahkan ke "*open_list*". Tetapkan titik awal n_{start} sebagai *parent node* dari titik-titik ini.
3. Hapus titik awal dari "*open_list*", dan tambahkan ke "*closed_list*".
4. Hitung fungsi evaluasi $f(n)$ untuk setiap titik dalam "*open_list*" saat ini, kemudian pilih titik n_1 dengan nilai $f(n)$ terkecil. Hapus n_1 dari "*open_list*", tambahkan n_1 ke "*closed_list*", dan kosongkan "*open_list*".
5. Cari semua titik yang terhubung pada titik n_1 dan masukkan ke "*open_list*". Jika titik-titik tersebut merupakan hambatan atau titik dalam "*closed_list*", mereka tidak dipertimbangkan. Tetapkan n_1 sebagai *parent node* dari titik-titik yang terdapat pada "*open_list*", hitung fungsi evaluasi $f(n)$, kemudian pilih titik n_2 dengan nilai fungsi evaluasi terkecil. Jika pada perhitungan pertama dari titik awal atau dari perhitungan sebelum-sebelumnya terdapat nilai fungi evaluasi yang lebih kecil (n_3) maka:
   a. Kita pilih nilai fungsi evaluasi pada titik n_3.

b. Kita *update* atau kita kosongkan "*closed_list*".
c. mengisi "*closed_list*" dengan titik n_3 (titik n dengan fungsi evaluasi terkecil dari perhitungan yang lain) dan titik-titik terpilih sebelum titik n_3.
d. Tetapkan titik n_3 sebagai *parent node*.
e. Cari semua titik yang terhubung dengan titik n_3, kita masukkan ke "*open_list*", kita hitung nilai fungsi evaluasi pada titik-titik yang terdapat pada "*open_list*", kita pilih titik n dengan fungsi evaluasi terkecil, dan kita masukkan titik n dengan fungsi evaluasi terkecil pada "*closed_list*".

6. Lanjutkan perhitungan dengan memperhatikan nilai fungsi evaluasi $f(n)$ pada setiap perhitungan selanjutnya, jika terdapat nilai fungsi evaluasi lebih minimal pada perhitungan sebelumnya, maka kita *update* dan melanjutkan perhitungan pada titik dengan nilai fungsi evaluasi terkecil. Perhitungan berhenti apabila titik terpilih dengan nilai fungsi $f(n)$ terkecil yakni titik tujuan (n_{goal}) [12].

Modifikasi Algoritma *A-Star*

Modifikasi algoritma *A-Star* dilakukan guna memperoleh metode yang lebih optimal dari algoritma *A-Star* tradisional. Modifikasi ini dilakukan pada fungsi evaluasi $f(n)$ dengan menambah faktor normalisasi $\alpha(n)$ pada variabel nilai heuristik $h(n)$. Penambahan variabel $\alpha(n)$ variabel normalisasi berfungsi untuk meningkatkan efisiensi pencarian pada titik (*node*) yang ekstrem, maksud dari ekstrem yakni perbedaan jarak sisi (*edge*) yang terlalu tinggi dengan sisi yang lain dapat menyebabkan banyak iterasi dalam proses pencarian, sehingga penambahan variabel $\alpha(n)$ berfungsi untuk menemukan jalur optimal dengan iterasi yang lebih minimum. Sehingga fungsi evaluasi termodifikasi sebagai berikut :

$$f(n) = g(n) + h(n) * \alpha(n) \quad (4)$$

Keterangan:
$f(n)$: fungsi evaluasi titik n pada setiap simpul
$g(n)$: Akumulatif biaya aktual dari titik awal hingga titik n
$h(n)$: perkiraan biaya dari titik n ke titik tujuan, juga dikenal sebagai fungsi heuristik
$\alpha(n)$: Faktor normalisasi
Faktor normalisasi sebagai berikut:

$$\alpha(n) = \frac{\mu_e}{d(n)} \quad (5)$$

Keterangan:
μ_e : rata- rata dari semua jarak antar titik
$d(n)$: jarak dari titik saat ini (A) ke titik yang terhubung

Nilai heuristic $h(n)$

Perhitungan fungsi heuristik dalam suatu graf dengan menggunakan algoritma *A-Star* tradisional yakni menghitung nilai $h(n)$ pada tiap titik n menuju titik tujuan ($node_{goal}$), yang mana nilai heuristik ini sebagai bahan untuk fungsi evaluasi $f(n)$. Untuk cara perhitungan nilai heuristik $h(n)$ pada modifikasi algoritma *A-Star* masih sama yakni dengan menggunakan metode *Euclidan* (2) dan merubah satuan dari $h(n)$ dari derajat menjadi km dengan mengalikan 1 *degree earth* $\left(111.132\frac{km}{\circ}\right)$. 1 derajat bumi didapatkan dari total keliling bumi yaitu 40.075 km (empat puluh ribu tujuh puluh lima kilo meter) dibagi 360 derajat sehingga menghasilkan $\left(111.132\frac{km}{\circ}\right)$ [13].

HASIL DAN PEMBAHASAN

Implementasi Algoritma *A-Star* Standar

Penentuan Rute $K22 \rightarrow W5$. Kita cari nilai heuristik $h(n)$ pada tiap titik $node_n$ (titik selain titik awal K22 dan titik tujuan W5) menuju titik tujuan W5, hasil pencarian kita lampirkan pada Lampiran 5. K22 merupakan titik awal sehingga K22 kita masukkan kedalam *Closed_List* dengan jarak tempuh $g(K22) = 0$ km:

Closed_list= $\{K22\}$

Kemudian kita cari titik yang terhubung pada K22 dan kita masukkan ke dalam *Open_List* dan kita hitung nilai evaluasi titik tiitk tersebut:

Open_List = $\{K21, K11, W14\}$

$$f(K21) = g(K21) + h(K21) = 20 + 30.9 = 50.9$$
$$f(K11) = g(K11) + h(K11) = 17.7 + 19.8 = 37.5 \quad \text{(i)}$$
$$f(W14) = g(W14) + h(W14) = 10.7 + 25.4 = 36.1$$

Dari hasil perhitungan pertama kita peroleh bahwa W14 memiliki nilai evaluasi yang minimum sehingga W14 kita masukkan ke dalam *Closed_List* dengan jarak tempuh $g(W14) = 10.7$ km:

Closed_list= $\{K22, W14\}$

Kemudian kita cari titik yang terhubung dengan titik W14, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K23\}$

$$f(K23) = g(K23) + h(K23) = 10.7 + 6 + 20.6 = 37.3 \quad \text{(ii)}$$

Dari hasil perhitungan ke-dua kita peroleh bahwa K23 memiliki nilai evaluasi yang minimum sehingga K23 kita masukkan ke dalam *Closed_List* dengan jarak tempuh $g(K23) = 16.7$ km:

Closed_list= $\{K22, W14, K23\}$

Kemudian kita cari titik yang terhubung dengan titik W14, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K11, K4, K3\}$

$$f(K11) = g(K11) + h(K11) = 16.7 + 17.8 + 19.8 = 54.3$$
$$f(K4) = g(K4) + h(K4) = 16.7 + 14.8 + 11.2 = 42.7 \quad \text{(iii)}$$
$$f(K3) = g(K3) + h(K3) = 16.7 + 12.8 + 11.3 = 40.8$$

Karena pada perhitungan pertama masih terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada $(K22 \rightarrow K11)$ kemudian kita update *Closed_List* dengan jarak tempuh $g(K13) = 17.7$ km:

Closed_list= $\{K22, K11\}$

Kemudian kita cari titik yang terhubung dengan K11, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K21, W8, K10, K9, K23\}$

$$f(K21) = g(K21) + h(K21) = 17.7 + 16.3 + 30.9 = 64.9$$
$$f(W8) = g(W8) + h(W8) = 17.7 + 5.8 + 22.9 = 46.4$$
$$f(K10) = g(K10) + h(K10) = 17.7 + 7.1 + 17.8 = 42.6 \quad \text{(iv)}$$
$$f(K9) = g(K9) + h(K9) = 17.7 + 7.9 + 13.7 = 39.3$$
$$f(K23) = g(K23) + h(K23) = 17.7 + 17.8 + 20.6 = 56.1$$

Dari hasil perhitungan ke-empat kita peroleh bahwa K9 memiliki nilai evaluasi yang minimum sehingga K9 kita masukkan ke dalam *Closed_List* dengan jarak tempuh $g(K9) = 25.6$ km:

Closed_list= $\{K22, K11, K9\}$

Kemudian kita cari titik yang terhubung dengan titik K9, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K10, K24, K5, K4\}$

$$\begin{aligned} f(K10) &= g(K10) + h(K10) = 25.6 + 6.6 + 17.8 = 50 \\ f(K24) &= g(K24) + h(K24) = 25.6 + 7.5 + 13.4 = 46.5 \\ f(K5) &= g(K5) + h(K5) = 25.6 + 13.1 + 3.83 = 42.5 \\ f(K4) &= g(K4) + h(K4) = 25.6 + 9 + 11.2 = 45.8 \end{aligned} \quad \text{(v)}$$

Karena pada perhitungan ke-tiga masih terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada ($K22 \rightarrow W14 \rightarrow K23 \rightarrow K3$) kemudian kita update *Closed_List* dengan jarak tempuh $g(K3) = 29.5$ km:

Closed_list= $\{K22, W14, K23, K3\}$

Kemudian kita cari titik yang terhubung dengan titik K3, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K2, K1, K4, K5\}$

$$\begin{aligned} f(K2) &= g(K11) + h(K11) = 29.5 + 8.3 + 13.1 = 50.9 \\ f(K1) &= g(K4) + h(K4) = 29.5 + 13.5 + 8.18 = 51.2 \\ f(K4) &= g(K4) + h(K4) = 29.5 + 6.8 + 11.2 = 47.5 \\ f(K5) &= g(K5) + h(K5) = 29.5 + 11.7 + 3.83 = 45 \end{aligned} \quad \text{(vi)}$$

Karena pada perhitungan ke-lima masih terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada ($K22 \rightarrow K11 \rightarrow K9 \rightarrow K5$) kemudian kita update *Closed_List* dengan jarak tempuh $g(K13) = 38.7$ km:

Closed_list= $\{K22, K11, K9, K5\}$

Kemudian kita cari titik yang terhubung dengan titik K5, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K24, K6, W5, K1, K3, K4\}$

$$\begin{aligned} f(K24) &= g(K24) + h(K24) = 38.7 + 12.8 + 13.4 = 64.9 \\ f(K6) &= g(K6) + h(K6) = 38.7 + 5.8 + 4.57 = 49.1 \\ f(W4) &= g(W4) + h(W5) = 38.7 + 4.8 + 0 = 43.5 \\ f(K1) &= g(K1) + h(K1) = 38.7 + 16.9 + 8.18 = 63.8 \\ f(K3) &= g(K3) + h(K3) = 38.7 + 11.7 + 11.3 = 61.7 \\ f(K4) &= g(K4) + h(K4) = 38.7 + 8.4 + 11.2 = 58.3 \end{aligned} \quad \text{(vii)}$$

Karena pada perhitungan ke-empat masih terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada ($K22 \rightarrow K11 \rightarrow K10$) kemudian kita update *Closed_List* dengan jarak tempuh $g(K10) = 24.8$ km:

Closed_list= $\{K22, K11, K10\}$

Kemudian kita cari titik yang terhubung dengan K10, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{W9, K12, K13, W8\}$

$$\begin{aligned} f(W9) &= g(W9) + h(W9) = 24.8 + 6.1 + 30.9 = 47.7 \\ f(K12) &= g(K12) + h(K12) = 24.8 + 4.5 + 22.9 = 48.7 \\ f(K13) &= g(K13) + h(K13) = 24.8 + 5.8 + 17.8 = 51.6 \\ f(W8) &= g(W8) + h(W8) = 24.8 + 7.9 + 13.7 = 55.6 \end{aligned} \quad \text{(viii)}$$

Karena pada perhitungan ke-tiga masih terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada ($K22 \rightarrow W14 \rightarrow K23 \rightarrow K4$) kemudian kita update *Closed_List* dengan jarak tempuh $g(K4) = 31.5$ km:

Closed_list= $\{K22, W14, K23, K4\}$

Kemudian kita cari titik yang terhubung dengan titik K4, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K3, K5, K9\}$

$$f(K3) = g(K3) + h(K3) = 31.5 + 6.8 + 11.3 = 49.6$$
$$f(K5) = g(K5) + h(K5) = 31.5 + 8.4 + 3.83 = 43.7 \quad \text{(ix)}$$
$$f(K9) = g(K9) + h(K9) = 31.5 + 9 + 13.7 = 54.2$$

Karena pada perhitungan ke-tujuh terdapat nilai evaluasi yang lebih minimum dan mencapai titik tujuan maka rute terpendek telah ditemukan $(K22 \rightarrow K11 \rightarrow K9 \rightarrow K5 \rightarrow W5)$ dengan total update perubahan rute pencarian atau *Closed_List* sebanyak lima kali, total perhitungan sembilan kali, dengan rute sepanjang 43.5 km.

Closed_List $= \{K22, K11, K9, K5, W5\}$

Implementasi Modifikasi Algoritma *A-Star*

Implementasi modifikasi algoritma *A-star* untuk tahapan penerapan tetap menggunakan langkah-langkah algoritma *A-Star* Tradisional, yang membedakan hanya pada fungsi evaluasi. Adapun untuk fungsi evaluasi modifikasi Algoritma *A-Star* Sebagai berikut:

$$f(n) = g(n) + h(n) * \alpha(n); \quad \alpha(n) = \frac{\mu_e}{d(n)}$$

Sehingga kita cari terlebih dahulu nilai rata-rata semua bobot *edge* :

$$\mu_e = \frac{\sum e_i}{|E|} = \frac{jumlah\ total\ bobot\ edge}{jumlah\ total\ edge}$$

=(8+8.9+8.6+10.3+4.8+11.8+8.5+17.3+11.6+5.9+3.9+7.2+2.3+13.4+3.3+10.9+3.6+2.4+2.2+5.8+9.5+16.5+12.7+11.5+14+8.9+11.4+12.4+13.2+22.7+20+13.4+16.3+7.9+5.8+10.7+17.7+6+7.1+7.9+17.8+14.+12.8+6.6+6.1+4.2+6.7+9.2+7.2+8.5+5+12.3+12.8+7.5+13.1+9+8.4+6.8+8.3+13.5+11.7+16.9+4.8+5.8+8.9+15.7+10.8+13.+4.2+15.4+2.8+3.1+4.5+3.1)/ 74=9.57027027

Penentuan Rute $K22 \rightarrow W5$. Untuk nilai $h(n)$ sama dengan pencarian rute pada algoritma *A-Star* tradisional pada pencarian rute terpendek $K22 \rightarrow W5$. $K22$ sebagai titik awal dan $W4$ sebagai titik tujuan, maka K22 kita masukkan kedalam *Closed_List* dengan jarak tempuh $g(K22) = 0$ km:

Closed_list$= \{K22\}$

Kemudian kita cari titik yang terhubung dengan titik K22, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K21, K11, W14\}$

$$f(K21) = g(K21) + h(K21) * \alpha(K21) = 20 + 30.9 * \frac{9.57}{20} = 34.828$$
$$f(K11) = g(K11) + h(K11) * \alpha(K11) = 17.7 + 19.8 * \frac{9.57}{17.7} = 28.435 \quad \text{(i)}$$
$$f(W14) = g(W14) + h(W14) * \alpha(W14) = 10.7 + 25.4 * \frac{9.57}{10.7} = 33.482$$

Dari hasil perhitungan pertama kita peroleh bahwa K11 memiliki nilai evaluasi yang minimum sehingga K11 kita masukkan ke dalam *Closed_List* dengan jarak tempuh $g(K11) = 17.7$ km:

Closed_list$= \{K22, K11\}$

Kemudian kita cari titik yang terhubung dengan titik K11, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:

Open_List = $\{K21, W8, K10, K9, K23\}$

$$f(K21) = g(K21) + h(K21) * \alpha(K21) = 17.7 + 16.3 + 30.9 * \frac{9.57}{16.3} = 52.193 \quad \text{(ii)}$$

$$f(W8) = g(W8) + h(W8) * \alpha(W8) = 17.7 + 5.8 + 22.9 * \frac{9.57}{5.8} = 61.392$$
$$f(K10) = g(K10) + h(K10) * \alpha(K10) = 17.7 + 7.1 + 17.8 * \frac{9.57}{7.1} = 48.86$$
$$f(K9) = g(K9) + h(K9) * \alpha(K9) = 17.7 + 7.9 + 13.7 * \frac{9.57}{7.9} = 42.243$$
$$f(K23) = g(K23) + h(K23) * \alpha(K23) = 17.7 + 17.8 + 20.6 * \frac{9.57}{17.8} = 46.606$$

Karena pada perhitungan pertama terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada ($K22 \rightarrow W14$) kemudian kita update *Closed_List* dengan jarak tempuh $g(W14) = 10.7$ km:
Closed_list= $\{K22, W14\}$

Kemudian kita cari titik yang terhubung dengan titik W14, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:
Open_List = $\{K23\}$

$$f(K23) = g(K23) + h(K23) * \alpha(K23) = 10.7 + 6 + 20.6 * \frac{9.57}{6} = 49.65 \quad \text{(iii)}$$

Karena pada perhitungan pertama masih terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada ($K22 \rightarrow K21$) kemudian kita update *Closed_List* dengan jarak tempuh $g(W10) = 20.7$ km:
Closed_list= $\{K22, K21\}$

Kemudian kita cari titik yang terhubung dengan titik K21, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:
Open_List = $\{K19, W8, K11\}$

$$f(K19) = g(K19) + h(K19) * \alpha(K19) = 20.7 + 11.5 + 31.6 * \frac{9.57}{11.5} = 57.871$$
$$f(W8) = g(W8) + h(W8) * \alpha(W8) = 20 + 15.4 + 22.9 * \frac{9.57}{15.4} = 49.671 \quad \text{(iv)}$$
$$f(K11) = g(K11) + h(K11) * \alpha(K11) = 20 + 16.3 * \frac{9.57}{16.3} = 47.958$$

Karena pada perhitungan ke-dua masih terdapat nilai evaluasi yang lebih minimum maka kita update pencarian rute pada titik dengan nilai evaluasi minimum yaitu pada ($K22 \rightarrow K11 \rightarrow K9$) kemudian kita update *Closed_List* dengan jarak tempuh $g(K9) = 25.6$ km:
Closed_list= $\{K22, K11, K9\}$

Kemudian kita cari titik yang terhubung dengan titik K9, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:
Open_List = $\{K10, K24, K5, K4\}$

$$f(K10) = g(K10) + h(K10) * \alpha(K10) = 25.6 + 6.1 + 17.8 * \frac{9.57}{7.1} = 58.083$$
$$f(K24) = g(K24) + h(K24) * \alpha(K24) = 25.6 + 7.5 + 13.4 * \frac{9.57}{7.5} = 50.247$$
$$f(K5) = g(K5) + h(K5) * \alpha(K5) = 25.6 + 13.1 + 3.83 * \frac{9.57}{17.7} = 41.505 \quad \text{(v)}$$
$$f(K4) = g(K4) + h(K4) * \alpha(K4) = 25.6 + 9 + 11.2 * \frac{9.57}{9} = 46.543$$

Dari hasil perhitungan ke-lima kita peroleh bahwa K5 memiliki nilai evaluasi yang minimum sehingga K5 kita masukkan ke dalam *Closed_List* dengan jarak tempuh $g(K5) = 38.7$ km:
Closed_list= $\{K22, K11, K9, K5\}$

Kemudian kita cari titik yang terhubung dengan titik K9, kita update *Open_List* dengan memasukkan titik-titik tersebut, dan kita hitung nilai evaluasi pada masing-masing titik:
Open_List = $\{K24, K6, W5, K1, K3, K4\}$

$$f(K24) = g(K24) + h(K24) * \alpha(K24) = 38.7 + 12.8 + 13.4 * \frac{9.57}{12.8} = 61.547$$
$$f(K6) = g(K6) + h(K6) * \alpha(K6) = 38.7 + 5.8 + 4.57 * \frac{9.57}{5.8} = 52.062$$
$$f(W5) = g(W5) + h(W5) *\propto (W5) = 38.7 + 4.8 + 0 * \frac{9.57}{4.8} = 43.5$$
$$f(K1) = g(K1) + h(K1) * \alpha(K1) = 38.7 + 16.9 + 3.83 * \frac{9.57}{16.9} = 60.245$$
$$f(K3) = g(K3) + h(K3) * \alpha(K3) = 38.7 + 11.7 + 8.18 * \frac{9.57}{11.7} = 58.678$$
$$f(K4) = g(K4) + h(K4) * \alpha(K4) = 38.7 + 8.4 + 11.3 * \frac{9.57}{8.4} = 59.896$$ (vi)

Karena pada perhitungan ke-delapan terdapat nilai evaluasi yang lebih minimum dan mencapai titik tujuan maka rute terpendek telah ditemukan $(K22 \rightarrow K11 \rightarrow K9 \rightarrow K5 \rightarrow W4)$ dengan total update perubahan rute pencarian atau *Closed_List* sebanyak tiga kali, total perhitungan enam kali, dengan rute sepanjang 43.5 km
Closed_List = $\{K22, K11, K9, K5, W4\}$

Analisis dan Representasi Hasil

Berikut hasil pencarian rute terpendek algoritma *A-Star* tradisional dan modifikasi algoritma *A-Star* menggunakan program *Jupyter Notebook*:

Tabel 2 *Hasil Pencarian Rute Terpendek Algoritma A-Star Standar*

Rute	Update Pencarian	Rute Terpendek	Jarak	Jumlah Iterasi
$W10 \rightarrow W8$	6	$W10 \rightarrow K13 \rightarrow W8$	29.2 km	8
$K18 \rightarrow W8$	5	$K18 \rightarrow K15 \rightarrow W11 \rightarrow K14 \rightarrow K13 \rightarrow W8$	37.6 km	10
$W7 \rightarrow W8$	19	$W7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	43.5 km	23
$W4 \rightarrow W8$	27	$W4 \rightarrow K6 \rightarrow K7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	38.1 km	30
$K6 \rightarrow W8$	5	$K6 \rightarrow K7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	29.2 km	28
$K15 \rightarrow W8$	7	$K15 \rightarrow W11 \rightarrow K14 \rightarrow K13 \rightarrow W8$	27.3 km	6
$K7 \rightarrow W8$	13	$K7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	20.7 km	16
$K20 \rightarrow W10$	2	$K20 \rightarrow K19 \rightarrow K13 \rightarrow W10$	34.1 km	5
$K2 \rightarrow W10$	3	$K2 \rightarrow K3 \rightarrow K4 \rightarrow K9 \rightarrow K10 \rightarrow K12 \rightarrow W10$	37.6 km	9
$W4 \rightarrow W10$	5	$W4 \rightarrow K6 \rightarrow K7 \rightarrow W7 \rightarrow W10$	35.5 km	8
$K4 \rightarrow W10$	0	$K4 \rightarrow K9 \rightarrow K10 \rightarrow K12 \rightarrow W10$	22.5 km	4
$K17 \rightarrow W10$	2	$K17 \rightarrow K15 \rightarrow W11 \rightarrow W10$	28.5 km	6

Rute	Update Pencarian	Rute Terpendek	Jarak	Jumlah Iterasi
$K11 \rightarrow W10$	0	$K11 \rightarrow K10 \rightarrow K12 \rightarrow W10$	14 km	3
$K6 \rightarrow W10$	5	$K6 \rightarrow K7 \rightarrow W7 \rightarrow W10$	26.6 km	7
$K5 \rightarrow W10$	6	$K5 \rightarrow K9 \rightarrow K10 \rightarrow K12 \rightarrow W10$	26.6 km	10
$K18 \rightarrow W10$	1	$K18 \rightarrow K15 \rightarrow W11 \rightarrow W10$	28.5 km	4
$K22 \rightarrow W5$	5	$K22 \rightarrow K11 \rightarrow K9 \rightarrow K5 \rightarrow W5$	43.5 km	9
$K2 \rightarrow W5$	3	$K2 \rightarrow K3 \rightarrow K5 \rightarrow W5$	24.8 km	5
$K4 \rightarrow W5$	0	$K4 \rightarrow K5 \rightarrow W5$	13.2 km	2
$K11 \rightarrow W5$	2	$K11 \rightarrow K9 \rightarrow K5 \rightarrow W5$	25.8 km	4

Tabel 3 *Hasil Pencarian Rute Terpendek Modifikasi Algoritma A-Star*

Rute	Update Pencarian	Rute Terpendek	Jarak	Jumlah Iterasi
$W10 \rightarrow W8$	1	$W10 \rightarrow K13 \rightarrow W8$	29.2 km	3
$K18 \rightarrow W8$	6	$K18 \rightarrow K15 \rightarrow W11 \rightarrow K14 \rightarrow K13 \rightarrow W8$	37.6 km	10
$W7 \rightarrow W8$	1	$W7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	43.5 km	4
$W4 \rightarrow W8$	1	$W4 \rightarrow K6 \rightarrow K7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	38.1 km	6
$K6 \rightarrow W8$	1	$K6 \rightarrow K7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	29.2 km	5
$K15 \rightarrow W8$	2	$K15 \rightarrow W11 \rightarrow K14 \rightarrow K13 \rightarrow W8$	27.3 km	6
$K7 \rightarrow W8$	0	$K7 \rightarrow W9 \rightarrow K10 \rightarrow W8$	20.7 km	2
$K20 \rightarrow W10$	2	$K20 \rightarrow K19 \rightarrow K13 \rightarrow W10$	34.1 km	5
$K2 \rightarrow W10$	1	$K2 \rightarrow K3 \rightarrow K4 \rightarrow K9 \rightarrow K10 \rightarrow K12 \rightarrow W10$	37.6 km	8
$W4 \rightarrow W10$	2	$W4 \rightarrow K6 \rightarrow K7 \rightarrow W7 \rightarrow W10$	35.5 km	5
$K4 \rightarrow W10$	0	$K4 \rightarrow K9 \rightarrow K10 \rightarrow K12 \rightarrow W10$	22.5 km	4
$K17 \rightarrow W10$	3	$K17 \rightarrow K15 \rightarrow W11 \rightarrow W10$	28.5 km	5
$K11 \rightarrow W10$	0	$K11 \rightarrow K10 \rightarrow K12 \rightarrow W10$	14 km	3
$K6 \rightarrow W10$	2	$K6 \rightarrow K7 \rightarrow W7 \rightarrow W10$	26.6 km	4
$K5 \rightarrow W10$	1	$K5 \rightarrow K9 \rightarrow K10 \rightarrow K12 \rightarrow W10$	26.6 km	5

Rute	Update Pencarian	Rute Terpendek	Jarak	Jumlah Iterasi
$K18 \rightarrow W10$	0	$K18 \rightarrow K15 \rightarrow W11 \rightarrow W10$	28.5 km	3
$K22 \rightarrow W5$	4	$K22 \rightarrow K11 \rightarrow K9 \rightarrow K5 \rightarrow W5$	43.5 km	7
$K2 \rightarrow W5$	1	$K2 \rightarrow K3 \rightarrow K5 \rightarrow W5$	24.8 km	4
$K4 \rightarrow W5$	0	$K4 \rightarrow K5 \rightarrow W5$	13.2 km	2
$K11 \rightarrow W5$	0	$K11 \rightarrow K9 \rightarrow K5 \rightarrow W5$	25.8 km	3

Dari hasil pengujian pencarian rute terpendek sebanyak dua puluh kali didapatkan hasil bahwa modifikasi algoritma *A-Star* lebih cepat dalam menentukan rute terpendek kita lihat dalam banyak iterasi pada masing-masing algoritma. Presentase minimum iterasi/ efektifitas pencarian rute didapatkan melalui [14]:

$$nilai\ efektifitas = \frac{ut - um}{ut} \times 100\%$$

Di mana variabel ut mewakili jumlah iterasi pencarian rute terpendek algoritma *A-Star* tradisional dan um mewakili jumlah iterasi pencarian rute terpendek modifikasi algoritma *A-Star.* Berikut hasil perhitungan presentase nilai efektifitas:

Tabel 4 *Presentase Hasil Efektifitas Minimum Iterasi*

Pengujian ke-	Perhitungan	Hasil
1	$\frac{8-3}{8} \times 100\%$	62.5%
2	$\frac{10-10}{10} \times 100\%$	0%
3	$\frac{5-3}{5} \times 100\%$	82.6%
4	$\frac{30-6}{30} \times 100\%$	80%
5	$\frac{28-5}{28} \times 100\%$	82.1%
6	$\frac{6-6}{6} \times 100\%$	0%
7	$\frac{16-2}{16} \times 100\%$	87.5%
8	$\frac{5-5}{5} \times 100\%$	0%
9	$\frac{9-8}{9} \times 100\%$	11.1%
10	$\frac{8-5}{8} \times 100\%$	37.5%
11	$\frac{4-4}{4} \times 100\%$	0%
12	$\frac{6-5}{6} \times 100\%$	16.7%
13	$\frac{3-3}{3} \times 100\%$	0%

Pengujian ke-	Perhitungan	Hasil
14	$\frac{7-4}{7} \times 100\%$	42.9%
15	$\frac{10-5}{10} \times 100\%$	50%
16	$\frac{4-3}{4} \times 100\%$	25%
17	$\frac{9-7}{9} \times 100\%$	22.2%
18	$\frac{5-4}{5} \times 100\%$	20%
19	$\frac{2-2}{2} \times 100\%$	0%
20	$\frac{4-3}{4} \times 100\%$	25%
Rata-rata		32.3%

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa modifikasi algoritma *A-Star* memungkinkan menggantikan algoritma *A-Star* tradisional. Algoritma *A-Star* memiliki keunggulan dalam pencarian rute terpendek hampir selalu dapat menemukan jarak minimum. Tetapi dalam prosesnya terkadang terlalu banyak proses iterasi sehingga peneliti memodifikasi algoritma *A-Star* dengan data sampel titik lokasi wisata Blitar dengan menggunakan manual sebagai langkah-langkah penerapan algoritma, dan menggunakan program sebagai uji perbandingan kedua algoritma tersebut. Dengan melakukan sebanyak 20 kali pengujian pencarian rute terpendek didapatkan hasil rute dan jarak yang sama akan tetapi modifikasi algoritma *A-Star* lebih hemat 32.3% dalam proses iterasi pencarian dibanding algoritma *A-Star* tradisional

DAFTAR PUSTAKA

[1] R. Munir, "Graf," *Revisi Buku Mat. Disk. 2012*, no. Update, 2023.

[2] T. Liao, F. Chen, Y. Wu, H. Zeng, S. Ouyang, and J. Guan, "Research on Path Planning with the Integration of Adaptive A-Star Algorithm and Improved Dynamic Window Approach," *Electron.*, vol. 13, no. 2, pp. 1–20, 2024, doi: 10.3390/electronics13020455.

[3] N. N. Sania and I. Sari, "Implementasi Rencana Perjalanan Wisata Di Kota Bogor Menggunakan Algoritma Greedy Berbasis Website," *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 2, pp. 114–130, 2019, doi: 10.35760/tr.2019.v24i2.2390.

[4] A. S. R. Maula, Tundo, S. Adrianto, Kastum, and N. Sutisna, "Implementasi Penggunaan Algoritma A* pada Penentuan Jarak Terpendek dari Cilacap ke Yogyakarta," *J. Ilm. Inform. Komput.*, vol. 29, no. 1, pp. 73–82, 2024, doi: https://doi.org/10.35760/ik.2024.v29i1.10661.

[5] E. Sumantri and S. Hidayattullah, "Penerapan Algoritma A*Star untuk Mencari Rute Terpendek dari Kemayoran ke Destinasi Monumen Nasional (MONAS)," *J. Sains dan Teknol.*, vol. 5, no. 2, pp. 673–680, 2023, doi: 10.55338/saintek.v5i2.1432.

[6] Ropiqoh and R. S. Lubis, "Implementation of a-Star Algorithm in Finding the Shortest Route of Cooking Oil Distribution in Karo Regency Using Graph," *Mathline*, vol. 8, no. 2, pp. 725–738, 2023, [Online]. Available: http://doi.org/10.31943/mathline.v8i2.442

[7] S. R. Aulia, W. Wamiliana, A. Asmiati, and N. Notiragayu, "Perbandingan Algoritme Dijkstra dan Algoritme A* (A-Star) dalam Penentuan Lintasan Terpendek dari Dinas Pendidikan Provinsi Lampung ke Beberapa Sekolah Menengah Atas (SMA) Negeri di Provinsi Lampung," *J. Pepadun*, vol. 4, no. 2, pp. 183–190, 2023, doi: 10.23960/pepadun.v4i2.177.

[8] M. H. P. Yudha, S. Supian, and H. Napitupulu, "Optimalization Route to Tourism Places in West Java Using A-STAR Algorithm," *CAUCHY J. Mat. Murni dan Apl.*, vol. 7, no. 3, pp. 464–473, 2022, doi: 10.18860/ca.v7i3.17032.

[9] S. J. Russel and P. Norvig, *Artificial intelligence : Moder Approach*, Third edit., vol. 4. United States of America: Pearson Education, Inc., Upper Saddle River, New Jersey, 2010. doi: 10.1109/ICCAE.2010.5451578.

[10] F. Desiderio, "Development Of A Modified A-Star Algorithm For Path Planning," *Politec. milano 1863*, pp. 1–63, 2022, [Online]. Available: https://www.politesi.polimi.it/retrieve/c90d6703-d52d-42a9-a250-7a4fec04282a/Msc_Thesis_Desiderio_finale.pdf

[11] R. N. B. Sitepua and G. N. A. C. Putra, "Penentuan Rute Terpendek Menggunakan Algoritma A Star," *J. Nas. Teknol. Inf. dan Apl.*, vol. 1, no. November, pp. 431–440, 2022.

[12] Y. Fernando, M. A. Mustaqov, and D. A. Megawaty, "Penerapan Algoritma A-Star pada Aplikasi Pencarian Lokasi Fotografi di Bandar Lampung Berbasis Android," *J. Teknoinfo*, vol. 14, no. 1, pp. 27–34, 2020, doi: 10.33365/jti.v14i1.509.

[13] S. S. GANNETT, "Department of the Interior United States Geological Survey," *Pubs.Usgs.Gov*, vol. 19, p. 388, 1916, [Online]. Available: http://pubs.usgs.gov/wri/1985/4075/report.pdf

[14] W. F. Anshoriy, "Penerapan Metode Algoritma Clarke and Wright Savings pada Penentuan Rute Terpendek (Studi Kasus di Kantor Pos Pemeriksa Kabupaten Blitar)," Universitas Islam Negeri Maulana Malik Ibrahim Malang, 2023.