

Pencarian Stasiun Kereta Terdekat dengan Algoritma A Star Berbasis Android di Area Stasiun Wilayah Bekasi

Ikhsan Dwikurniawan¹, Herlawati^{1*}, Robertus Suraji¹

¹Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Bhayangkara Jakarta Raya, Indonesia

Email: ikhsan.dwi17@mhs.ubharajaya.ac.id, herlawati@dsn.ubharajaya.ac.id, robertus.suraji@dsn.ubharajaya.ac.id

INFORMASI ARTIKEL

Histori artikel:

Naskah masuk, 13 Nopember 2021

Direvisi, 18 Nopember 2021

Diiterima, 20 Nopember 2021

Kata Kunci:

Android,
Algoritma A STAR,
Stasiun.

ABSTRAK

Abstract- Transportation has become one of the most important needs in daily activities in social life. Advances in information technology that exists today, can be used as a means to improve public services, one of which is in the railway sector. With advances in information technology, it can make it easier for people to find information quickly and easily, but there are still some obstacles, namely the lack of information about the nearest station route. This study aims to create an android-based nearest station search application with the shortest route to the destination station using the A-STAR Algorithm. The A-STAR algorithm is an algorithm that looks for the shortest route to reach the expected destination. The stages are 1) Entering the initial node into the open list. 2) Do a loop. 3) Save the route backward, sort from the goal node to its parent to the start node while saving the node into an array. In this study Kranji Station, Bekasi Station, East Bekasi Station, Tambun Station, Cibitung Station, Telaga Murni Station, Cikarang Station. Tests in this study were carried out by black-box testing and comparison testing between the A-STAR algorithm and Google Map, the results obtained showed that the A-STAR algorithm was successful with the shortest distance, although there was the A-STAR Algorithm with the same results as Google Maps, and the A-STAR algorithm. There is also A-STAR that shows a longer distance than Google Maps.

Abstrak- Transportasi telah menjadi salah satu kebutuhan yang sangat penting dalam kegiatan sehari-hari di kehidupan bermasyarakat. Kemajuan teknologi informasi yang ada saat ini, dapat digunakan sebagai sarana untuk meningkatkan pelayanan umum, salah satunya adalah di bidang perkeretaapian. Dengan adanya kemajuan teknologi informasi dapat memudahkan masyarakat untuk mengetahui informasi secara cepat dan mudah, tetapi masih ada beberapa kendala yaitu kurangnya informasi mengenai rute stasiun terdekat. Penelitian ini bertujuan untuk membuat aplikasi pencarian Stasiun terdekat berbasis android dengan rute terpendek menuju Stasiun tujuan dengan menggunakan Algoritma A-STAR. Algoritma A-STAR ialah algoritma yang mencari rute terpendek untuk mencapai tujuan yang diharapkan. Tahapannya yaitu 1) Memasukkan node awal ke open list. 2) Melakukan looping. 3) Simpan rute secara backward, urutkan mulai dari node goal ke parent-nya sampai ke node awal bersamaan menyimpan node-nya ke dalam sebuah array. Pada penelitian ini Stasiun Kranji, Stasiun Bekasi, Stasiun Bekasi Timur, Stasiun Tambun, Stasiun Cibitung, Stasiun Telaga Murni, Stasiun Cikarang. Pengujian pada penelitian ini dilakukan dengan pengujian black box testing dan pengujian perbandingan antara algoritma A-STAR dengan Google Map, hasilnya diperoleh menunjukkan lebih banyak algoritma A-STAR berhasil dengan jarak terpendek, walaupun ada Algoritma A-STAR yang hasil sama dengan Google Maps, dan algoritma A-STAR ada juga yang menunjukkan jarak yang lebih jauh dibandingkan Google Maps.

Copyright © 2021 LPPM - STMIK IKMI Cirebon
This is an open access article under the CC-BY license

Penulis Korespondensi:

Herlawati

Program Studi Informatika,
Universitas Bhayangkara Jakarta Raya
Jl. Raya Perjuangan Bekasi Utara, Kota Bekasi, Jawa Barat 17121, Indonesia.
Email: herlawati@dsn.ubharajaya.ac.id

1. Pendahuluan

Stasiun Kereta Api merupakan sebuah tempat pemberhentian untuk naik dan turunnya penumpang atau pengguna transportasi kereta api. Angkutan kereta api sangat diminati oleh masyarakat karena harganya yang murah dan juga waktu yang efisien dibandingkan dengan transportasi lainnya, masyarakat Bekasi adalah salah satunya. Masyarakat Bekasi banyak yang menggunakan jasa angkutan ini karena alasan tersebut. Adanya hal tersebut harus sebanding dengan meningkatnya fasilitas stasiun yang memadai, untuk melayani penumpang secara merata.

Pencarian rute terpendek adalah usaha untuk mencari rute yang paling dekat dari posisi awal hingga akhir dengan beban paling ringan atau sedikit dibandingkan dengan seluruh rute yang ada. Terdapat banyak algoritma yang dapat digunakan dalam pencarian rute terpendek. Salah satunya adalah Algoritma A-STAR.

Algoritma A-STAR merupakan salah satu algoritma pencarian graph terbaik yang mampu menemukan jalur dengan biaya pengeluaran (*cost*) paling sedikit dari titik permulaan yang diberikan sampai ke titik tujuan yang diharapkan. [1] [2]

Berdasarkan latar belakang yang dikemukakan, maka permasalahan diidentifikasi sebagai berikut: 1) Kurangnya informasi mengenai rute terdekat menuju stasiun. 2) Menghabiskan jarak yang lebih jauh dan waktu yang cukup lama untuk mengetahui rute terdekat menuju Stasiun. 3) Belum adanya sistem informasi yang menyediakan rute terdekat yang harus ditempuh untuk menuju Stasiun Kereta.

Adapun tujuan dari penelitian ini adalah: 1) Membuat Aplikasi Android mengenai rute terdekat menuju stasiun. 2) Penerapan Algoritma A-STAR berbasis Android dengan membuat aplikasi dalam pencarian rute terdekat dalam menemukan stasiun. 3) Membandingkan Algoritma A-STAR dengan Google Maps untuk mengetahui hasil untuk mencari rute terdekat agar mencapai titik tujuan lebih cepat.

Beberapa penelitian sebelumnya yang telah menggunakan Algoritma A-STAR: (i) Agar

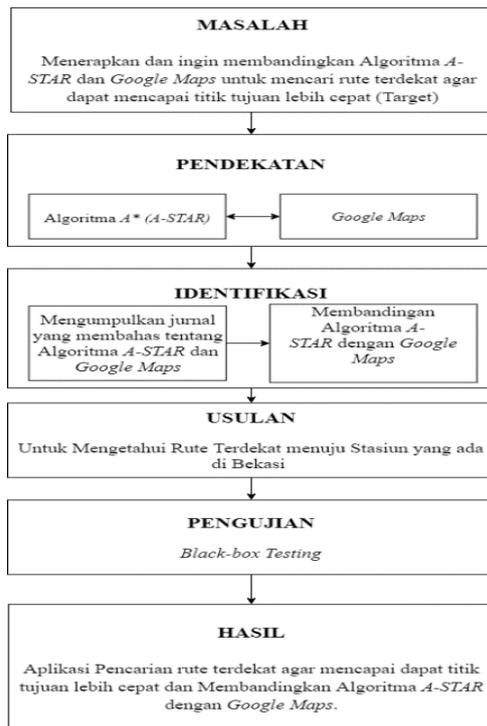
mempermudah menemukan lokasi gedung dan ruangan Universitas dengan algoritma A-STAR, [3] (ii) Pencarian jalur terpendek menuju Rumah Sakit Umum Bahterama, [4] penelitian sebelumnya objeknya adalah Rumah Sakit sedangkan peneliti menggunakan objek stasiun, (iii) Untuk mencari lokasi puskesmas terdekat dengan algoritma a-star di provinsi DKI Jakarta, sedangkan yang membedakan pada penelitian ini dengan penelitian sebelumnya adalah objek penelitiannya dan juga wilayah penelitian, [1] (iv) Untuk pencarian rute terpendek puskesmas rawat inap di banyumas, sedangkan yang membedakan pada penelitian sebelumnya adalah objek dan juga wilayah, penelitian sebelumnya objeknya adalah puskesmas di Banyumas sedangkan peneliti menggunakan objek stasiun. [5]

Jalur terpendek (*shortest path*) adalah jalur optimum yang dapat diselesaikan dengan menggunakan graph. Jalur ini biasanya ditentukan oleh rute yang memiliki total biaya perjalanan yang paling kecil atau murah. Jika diaplikasikan dengan graph maka setiap garis pada simpul titik memiliki bobot berupa nilai dan apabila dijumlahkan bobot dari garis yang dilalui maka memiliki nilai yang minimal [6].

Pencarian rute terpendek (*shortest path problem*) merupakan suatu permasalahan optimasi mencari rute minimum yang diperlukan untuk mencapai tempat tujuan berdasarkan beberapa jalur alternatif yang tersedia [7].

2. Metode Penelitian

Kerangka pemikiran pada dasarnya konsep yang akan diamati atau diukur melalui penelitian yang akan dilakukan. Menurut kerangka teori yang ada, kerangka penelitian yang akan digunakan dapat dilihat pada gambar 1.



Gambar 1. Kerangka Pemikiran

3. Hasil dan Pembahasan

Algoritma adalah sistem kerja komputer memiliki *brainware*, *hardware*, dan *software*. Tanpa salah satu dari ketiga sistem tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada software komputer. Software terbangun atas susunan program) dan syntax (cara penulisan/pembuatan program). Untuk menyusun program atau syntax, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software. Maka, algoritma berperan penting dalam penyusunan program atau syntax tersebut [8].

Algoritma A-STAR merupakan salah satu algoritma pencarian graph terbaik yang mampu menemukan jalur dengan biaya pengeluaran (cost) paling sedikit dari titik permulaan yang diberikan sampai ke titik tujuan yang diharapkan [1] [9].

Algoritma ini menggunakan fungsi jarak (*distance*) ditambah biaya (*cost*) yang biasanya dinotasikan dengan $f(x)$ untuk menentukan urutan kunjungan pencarian node di dalam tree. Perhitungan jarak didapatkan dari fungsi $path - cost$ yang selalu dinotasikan dengan $g(x)$, dimungkinkan bernilai heuristik ataupun tidak. Sedangkan, biaya didapatkan dari sebuah kemungkinan penerimaan atas "perkiraan heuristik" jarak ke titik tujuan yang dinotasikan dengan $h(x)$. [1][10], [11]

Fungsi $g(x)$ adalah jumlah biaya yang harus dikeluarkan dari node awal menuju node yang akan

dituju. Dengan $h(x)$ bagian dari fungsi $f(x)$ yang harus dapat heuristik, yang mana tidak diperbolehkan untuk terlalu jauh memperkirakan jarak ke arah tujuan. Oleh karena itu untuk aplikasi seperti routing, $h(x)$ mungkin mewakili garis lurus jarak ke titik tujuan, karena hal ini secara nyata dimungkinkan adanya jarak terpendek di antara dua titik yang dapat dirumuskan sebagai berikut:

Perhitungan Perkiraan Biaya secara Heuristik:

$$h(x) = (x - x_1) + (y - y_1) \quad (1)$$

Dimana x adalah koordinat x dari node awal, x_1 adalah koordinat x dari node lokasi ke n , dan y adalah koordinat y dari node awal, serta y_1 adalah koordinat y dari node lokasi ke n .

Jadi algoritma A-STAR dapat dirumuskan seperti berikut:

Perhitungan Algoritma A-STAR:

$$f(n) = g(n) + h(n) \quad (2)$$

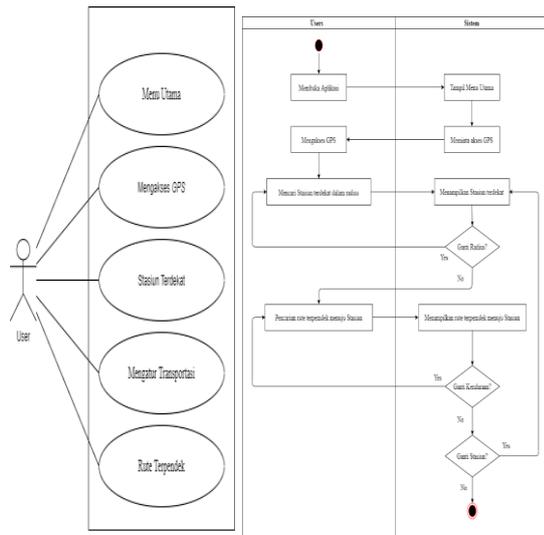
dimana: $f(n)$ = jumlah yang didapatkan dari $g(x)$ dan $h(x)$, $g(n)$ = total jarak didapat dari vertex awal ke vertex sekarang, $h(n)$ = jarak estimasi vertex tujuan, sebuah fungsi heuristik untuk membuat perkiraan seberapa jauh lintasan yang akan diambil ke vertex tujuan.

Dimana $f(x)$ merupakan biaya estimasi terendah, $g(x)$ adalah biaya dari node awal ke node n , dan $h(x)$: perkiraan biaya dari node awal ke node akhir [1]

UML adalah sebuah bahasa pemodelan untuk, menspesifikasikan, membangun dan memvisualisasikan sebuah sistem pengembangan perangkat lunak berbasis Obyek (*Object Oriented programming*). [3]

Diagram UML (*Unified Modelling Language*) digunakan untuk penggambaran proses-proses yang ada pada perancangan Aplikasi yang akan dibuat, seperti *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*.

Komponen use case Diagram terdiri: aktor dan use case. Dimana aktor merupakan pengguna, sedangkan use case adalah apa yang dilakukan aktor dengan relation sebagai penunjuknya. Pada gambar 2 merupakan use case diagram:

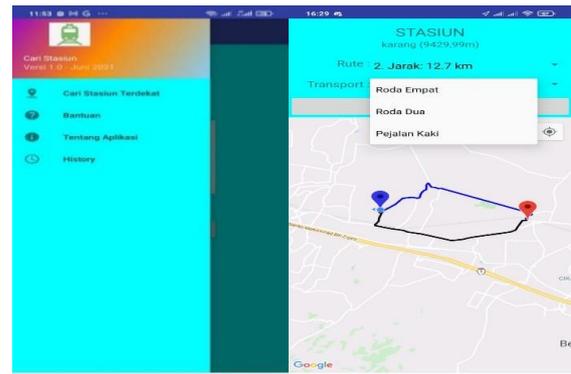


Gambar 2. Usecase Diagram dan Activity Diagram Mencari Stasiun

Pada Gambar 2. dijelaskan tentang activity diagram untuk menjalankan aplikasi pencarian stasiun terdekat, proses tampilan halaman yang digunakan oleh users. Pada activity diagram menggambarkan proses sebagai berikut: a) Users Membuka Aplikasi; b) Jika users berhasil membuka aplikasi, maka akan muncul menu utama aplikasi pencarian stasiun terdekat; c) Setelah itu, sistem akan meminta akses GPS, users akan mengakses GPS; d) Users mencari stasiun terdekat yang tersedia di aplikasi; e) Setelah itu, sistem menampilkan stasiun terdekat dari lokasi pengguna yang tersedia dalam radius. Namun jika ingin mengubah radius maka akan kembali pada proses mencari stasiun terdekat; f) Jika tidak ingin mengubah radius users akan melakukan pencarian rute terpendek; g) Setelah itu, sistem akan menampilkan rute terdekat menuju stasiun. Namun jika users ingin mengganti kendaraan maka sistem akan kembali pada proses pencarian rute terdekat. Jika tidak ingin mengganti kendaraan maka users akan melanjutkan proses; h) Jika users ingin mengganti stasiun maka sistem akan kembali pada proses menampilkan stasiun terdekat, jika tidak ingin mengganti maka users akan menemukan lokasi stasiun terdekat; i) Selesai.

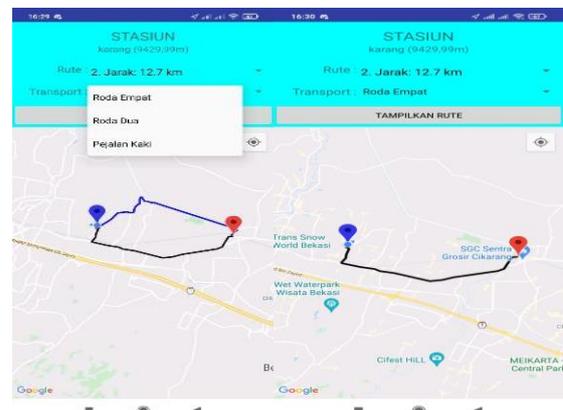
Tampilan interface

Pada tampilan interface terdapat tampilan menu utama, tampilan beberapa stasiun terdekat. Pada Gambar 3 merupakan tampilan sidebar yaitu menampilkan beberapa opsi dari aplikasi ini yaitu Cari Stasiun Terdekat, Bantuan, Tentang Aplikasi, dan History. Gambar 3 juga menunjukkan tampilan rute terpendek yang akan dipilih untuk menuju tujuan.



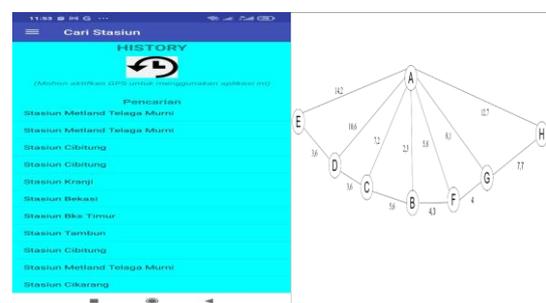
Gambar 3. Tampilan Slidebar dan Tampilan Rute Terpendek

Pada Gambar 4 merupakan tampilan users memilih kendaraan yang ingin digunakan untuk menuju tujuan. Tampilan Hasil Pencarian Rute Terpendek, juga dijelaskan tampilan hasil pencarian rute terpendek pada aplikasi ini.



Gambar 4. Tampilan Memilih Kendaraan dan Tampilan Pencarian Rute Terpendek

Pada Gambar 5 merupakan tampilan hasil history pada aplikasi ini.



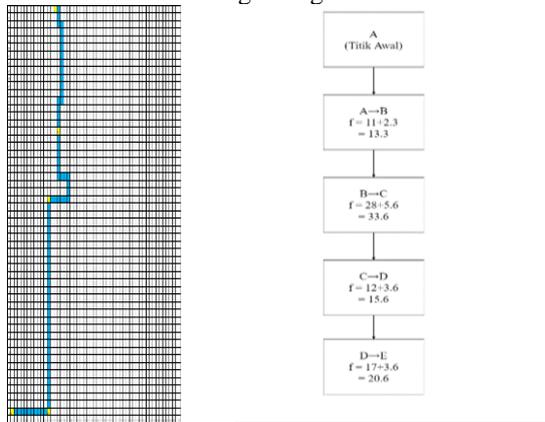
Gambar 5. Tampilan Hasil History Pencarian dan Graph Stasiun

Pada gambar 5 juga dijelaskan graph untuk melakukan pencarian rute terpendek menggunakan Algoritma A-STAR:

Pada huruf A, B, C, D, E, F, G, H, menjelaskan tentang node yang didapatkan oleh

titik koordinat (nilai dari latitude dan longitude) atau lokasi suatu objek yang akan dituju.

Pada Gambar 6 merupakan perhitungan heuristik dan Perhitungan Algoritma A-STAR.



Gambar 6. Perhitungan Heuristik dan Perhitungan Algoritma A-STAR

Setiap Indeks mewakili jarak 200 meter dan setiap node menunjukkan titik koordinat tujuan. A = Rumah (0,0), B = St. Tambun (0,11), C = St. Bekasi Timur (28,11), D = St. Bekasi (37,14), E = St. Kranji (53,13)

Perhitungan Perkiraan Biaya secara Heuristik dijelaskan sebagai berikut:

$$h(x) = (x-x_1) + (y - y_1)$$

dimana x = node awal, x_1 = node lokasi tujuan, y = node awal, y_1 = node lokasi tujuan.

- A. Heuristik A (0,0) ke B (0,11)
 $h(y-x) + (y_1+x_1) = (0-0) + (11-0)$
 $h(a, b) = (0+11)$
 $h = 11$
- B. Heuristik B (0,11) ke C (28,11)
 $h(y-x) + (y_1+x_1) = (28-0) + (11-11)$
 $h(b, c) = (28+0)$
 $h = 28$
- C. Heuristik C (28,11) ke D (37,14)
 $h(y-x) + (y_1+x_1) = (37-28) + (14-11)$
 $h(c, d) = (9+3)$
 $h = 12$
- D. Heuristik D (37,14) ke E (53,13)
 $h(y-x) + (y_1+x_1) = (53-37) + (14-13)$
 $h(d, e) = (16+1)$
 $h = 17$

Pada Tabel 1 merupakan hasil dari perhitungan heuristik.

Tabel 1. Perhitungan Heuristik

No.	Titik Indeks	Hasil
1.	A (0,0) ke B (0,11)	11
2.	B (0,11) ke C (28,11)	28
3.	C (28,11) ke D (37,14)	12
4.	D (37,14) ke E (53,13)	17

Pada Tabel 2 merupakan jarak relasi antar titik

Tabel 2. Jarak Antar Titik

No.	Relasi Titik	Jarak
1.	A→B	2,3 KM
2.	B→C	5,6 KM
3.	C→D	3,6 KM
4.	D→E	3,6 KM

Setelah mendapatkan nilai heuristik lalu akan mencari $f(n)$ menggunakan rumus algoritma A-STAR:

$h(n)$ = Nilai heuristik antar Koordinat

$g(n)$ = Jarak Koordinat ke titik tujuan

Pada Gambar 6 merupakan langkah-langkah pencarian, setelah menghitung heuristik dan juga melakukan langkah-langkah pencarian menggunakan algoritma A STAR, maka $f(n)$ total yang didapat adalah: Jalur ini adalah 83.1 Karena satu titik koordinat mewakili 200 meter maka jarak yang sebenarnya: $83.1 \times 200 = 16620$ Meter, dalam KM = 16.62 KiloMeter. KM jalur yang dilalui adalah A-B-C-D-E.

Tabel 3. Black-box Testing

Input	Proses	Output	Hasil
Klik Aplikasi	Masuk Ke Aplikasi	Menampilkan Menu Utama	Berhasil
Klik Stasiun Terdekat	Cari Maps	Menampilkan beberapa stasiun terdekat dengan radius 3000m	Berhasil
Klik Rute	Cari Masuk Ke pilihan Rute	Menampilkan Rute terdekat dan transportasi yang akan digunakan.	Berhasil
Slidebar	Masuk Ke beberapa opsi menu	Menampilkan Menu Cari Stasiun Terdekat, Menu Bantuan, Menu Tentang Aplikasi	Berhasil

Pengujian perbandingan

Dari percobaan antara A-STAR dan Google Maps dapat dilihat hasil dan rata-rata jarak yang harus ditempuh untuk mencari titik tujuan, hasil pengujian aplikasi dapat dilihat pada tabel 4, yaitu :

Tabel 4. Hasil Perbandingan Algoritma A STAR dengan Google Maps

No.	Titik Awal	Titik Tujuan	Aplikasi	Google Maps	Hasil
1.	Bumi Lestari H3/13	Stasiun Kranji	14,2 KM	14,2 KM	Sama
2.	Bumi	Stasiun	10,6	11 KM	Sesuai

No.	Titik Awal	Titik Tujuan	Aplikasi	Google Maps	Hasil
	Lestari H3/13	Bekasi	KM		
3.	Bumi Lestari H3/13	Stasiun Bekasi Timur	7,2 KM	7,3 KM	Sesuai
4.	Bumi Lestari H3/13	Stasiun Tambun	2,3 KM	2,2 KM	Tidak Sesuai
5.	Bumi Lestari H3/13	Stasiun Cibitung	5,8 KM	5,9 KM	Sesuai
6.	Bumi Lestari H3/13	Stasiun Metland Telaga Murni	8,1 KM	8,7 KM	Sesuai
7.	Bumi Lestari H3/13	Stasiun Cikarang	12,7 KM	12,8 KM	Sesuai

Dari hasil uji pada aplikasi pencarian rute terdekat menuju stasiun di Kota dan Kabupaten Bekasi menggunakan Algoritma A STAR dibandingkan dengan Aplikasi Google Maps dengan tujuh kali percobaan, dari delapan percobaan itu Algoritma A-STAR empat kali berhasil dengan jarak terpendek, satu kali Algoritma A-STAR hasil sama dengan Google Maps, dan satu kali algoritma A-STAR menunjukkan jarak yang lebih jauh dibandingkan Google Maps.

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan pada Pencarian Stasiun Kereta Terdekat Dengan Algoritma A-STAR Berbasis Android Di Bekasi. Dapat diambil kesimpulan, yaitu: 1) Dapat diterapkan Algoritma A-STAR pada pencarian lokasi stasiun terdekat berbasis Android. 2) Pengujian blackbox testing maka dapat disimpulkan aplikasi berjalan dengan baik. 3) Pada pengujian perbandingan yang dilakukan dari tujuh kali pengujian dapat disimpulkan menunjukkan lima kali algoritma A STAR berhasil jarak terpendek, satu kali Algoritma A STAR hasil sama dengan Google Maps, dan satu kali algoritma A-Star menunjukkan jarak yang lebih jauh dibandingkan Google Maps.

Daftar Pustaka

[1] V. Budiman, Y. S. H. Lesmono, and H. Agung, "Aplikasi Berbasis Android Untuk Mencari Lokasi Puskesmas Terdekat Dengan Algoritma a-Star Di Provinsi Dki Jakarta," *J. Sist. Informasi, Teknol. Inform. dan Komput.*, vol. Vol.9, no. No.1, pp. 39–48, 2018.

[2] W. Bismi, W. Gata, A. Anton, and T. Asra, "Penerapan Algoritma Hybrid Dalam Menentukan Rute Terpendek Antara Cabang Kampus," *Ultim. Comput. J. Sist. Komput.*, vol. 13, no. 1, pp. 1–9, 2021.

[3] M. Irsyad and E. Rasila, "Aplikasi

Pencarian Lokasi Gedung dan Ruangan Universitas Islam Negeri Sultan Syarif Kasim Riau pada Platform Android Menggunakan Algoritma A-Star (A *)," *J. CoreIT*, vol. 1, no. 2, pp. 90–95, 2015.

[4] M. Yamin and M. B. Talai, "Aplikasi Pencarian Jalur Terpendek Pada Rumah Sakit Umum Bahteramas Menggunakan Algoritma a* (a-Star)," *J. Inform.*, vol. 9, no. 2, pp. 1065–1078, 2015,

[5] M. A. Arsyad, D. Supriyadi, A. Veronica, L. N. Hidayah, and D. P. Pratiwi, "Penerapan Algoritma A Star Untuk Pencarian Rute Terpendek Puskesmas Rawat Inap Di Banyumas," *Conf. Electr. Eng. Telemat. Ind. Technol. Creat. Media 2019*, pp. 74–82, 2019, [Online].

[6] Z. Ramadhan, M. Zarlis, S. Efendi, and A. P. U. Siahaan, "Perbandingan Algoritma Prim dan Algoritma Floyd-Warshall dalam Menentukan Lintasan Terpendek (Shortest Path Problem)," *J. Ris. Komput.*, vol. 5, no. 2, pp. 136–139, 2018.

[7] P. Y. Utami, C. Suhery, and I. Ilhamsyah, "Aplikasi Pencarian Rute Terpendek Menggunakan Algoritma Genetika (Studi Kasus: Pencarian Rute Terpendek Untuk Pemadam Kebakaran Di Wilayah Kota Pontianak)," *J. Coding Sist. Komput. Univ. Tanjungpura*, vol. 02, no. 1, pp. 19–25, 2014.

[8] G. G. Maulana, "Pembelajaran Dasar Algoritma dan Pemrograman Menggunakan El-Goritma Berbasis Web," *J. Tek. Mesin*, vol. 6, pp. 69–73, 2017.

[9] R. Rizky, T. Hidayat, A. H. Nugroho, and Z. Hakim, "Implementasi Metode A*Star Pada Pencarian Rute Terdekat Menuju Tempat Kuliner di Menes Pandeglang Banten," *Geodika J. Kaji. Ilmu dan Pendidik. Geogr.*, vol. 4, no. 1, pp. 85–94, 2020, doi: 10.29408/geodika.v4i1.2068.

[10] S. Susilawati, R. Rizky, S. Setiyowati, and A. G. Pratama, "Penerapan Metode A*Star Pada Pencarian Rute Tercepat Menuju Destinasi Wisata Cagar Budaya Menes Pandeglang," *Geodika J. Kaji. Ilmu dan Pendidik. Geogr.*, vol. 4, no. 2, pp. 192–199, 2020.

[11] E. Erniyati and M. Mulyati, "Pencarian Jalur Terdekat Menuju Rumah Sakit Di Kota Bogor Dengan Menggunakan Algoritma a*," *Komputasi J. Ilm. Ilmu Komput. dan Mat.*, vol. 16, no. 1, pp. 245–253, 2019.