

PENCARIAN JALUR TERDEKAT MENUJU RUMAH SAKIT DI KOTA BOGOR DENGAN MENGGUNAKAN ALGORITMA A*

Erniyati, M. Kom¹, Mulyati, M. Kom²

^{1,2})Program Studi Ilmu Komputer, FMIPA, Universitas Pakuan,
Bogor, Indonesia

Corresponding Author: neny_erniyati@yahoo.com

Article history: received 2 Desember 2018; revised 15 Desember 2018; accepted 28 Desember 2018

Abstrak

Pencarian jalur terpendek (*shortesh path*) merupakan salah satu solusi dalam pencarian suatu lokasi yang terdekat jika terdapat beberapa jalur yang dapat dilaluinya. Seperti halnya dalam pencarian rumah sakit di kota Bogor. Kota Bogor merupakan kota yang cukup luas dan memiliki jalur yang cukup banyak. Untuk mencapai rumah sakit yang dituju dibutuhkan informasi jalur terdekat guna meminimalisir biaya dan waktu yang lebih efisien. Ada beberapa algoritma yang bisa digunakan untuk pencarian jalur terdekat diantaranya yaitu algoritma A* (*star*), algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan. Sistem ini selanjutnya dihubungkan dengan perangkat mobile yang berbasis android. Dengan menerapkan Algoritma A* (*star*) diharapkan dapat membantu dalam mencari dan menentukan lintasan terpendek sehingga didapat suatu jalur yang optimal untuk mencapai rumah sakit di Kota Bogor.

Keywords : algoritma A*, android, jalur terpendek, rumah sakit.

Abstract

The shortest path searching is one of some solutions for finding a locaton which is closest if there are several alternative paths that can be passed. It is similar with the case in searching for hospitals in the city of Bogor. Bogor is a city that is quite large and it has a number of routes. To reach the hospital destination, it needs information of the shortest path to minimize the cost and time eficiency. There are several method of algorithm can be implemented for this case, and the A* (*star*) algorithm is one of the solution. This algorithm will discard unnecessary steps which are definitely not as a solution. This system is then linked to an Android-based mobile device. By applying the A* (*star*) algorithm, it is expected to be able to help in finding and determining the shortest path with an optimal path to reach the hospital in the city of Bogor.

Keywords : algorithm A*, android, shortest path, hospital

1. Pendahuluan

Pencarian jalur terpendek (*shortesh path*) merupakan salah satu solusi dalam pencarian suatu lokasi yang terdekat jika terdapat beberapa jalur yang dapat dilaluinya. Seperti halnya dalam pencarian rumah sakit di kota Bogor. Jika sedang mengalami keadaan darurat atau butuh pertolongan karena sakit, maka pencarian rumah sakit yang terdekat sangat dibutuhkan.

Kota Bogor merupakan kota yang cukup luas dan memiliki jalur yang cukup banyak. Untuk mencapai rumah sakit yang dituju dibutuhkan informasi jalur terpendek guna meminimalisir

biaya dan waktu yang lebih efisien. Salah satu penentuan jalur lintasan terbaik adalah dengan mencari jalur terpendek. Jalur terpendek dapat dihitung dengan menggunakan teori graph.

Menurut teori graf, persoalan lintasan terpendek adalah untuk mencari lintasan antara dua buah simpul pada graf berbobot yang memiliki gabungan jumlah nilai bobot pada sisi graf yang dilalui dengan jumlah paling minimum. Persoalan lintasan terpendek paling banyak ditemui yaitu dalam bidang transportasi, seperti pada pencarian rute terpendek untuk menempuh dua kota [1].

Ada beberapa algoritma yang bisa digunakan untuk pencarian jalur terdekat diantaranya yaitu algoritma A*(a star), algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan [2].

Saat ini untuk pencarian jalur terdekat menggunakan *google map* ataupun GPS (*Global Position System*), GPS yang tersedia di aplikasi digunakan untuk menentukan jarak pengguna dengan tempat yang ingin dituju [3]. Namun di *google map* belum dapat merekomendasikan jalur mana yang terdekat menuju ke tujuan. Oleh karena dengan menerapkan Algoritma A* (star) diharapkan dapat membantu dalam mencari dan menentukan lintasan terpendek sehingga didapat suatu jalur yang optimal untuk mencapai tujuan.

Algoritma A* (*A-Star*) merupakan suatu algoritma yang termasuk pada kategori metode pencarian yang memiliki informasi (*informed search method*). Algoritma A* (*A-Star*) menggunakan estimasi jarak terdekat (*cost*/ jarak sebenarnya) untuk mencapai tujuan (*goal*) dan memiliki nilai heuristik yang digunakan sebagai dasar pertimbangan pemilihan jalur [4] Algoritma A* ini memeriksa *node* dengan menggabungkan $g(n)$, yaitu *cost* yang dibutuhkan untuk mencapai sebuah *node* dan $h(n)$ yaitu *cost* yang didapat dari *node* ketujuan [5]. Sehingga dapat dirumuskan sebagai berikut:

$$f(n) = g(n) + h(n) \tag{1}$$

$f(n)$ = perkiraan total *cost* terendah dari setiap *path* yang akan dilalui dari *node* n ke *node* tujuan.

$g(n)$ = biaya yang sudah dikeluarkan dari keadaan awal sampai keadaan n

$h(n)$ = perkiraan heuristik atau *cost* atau *path* dari *node* n ke tujuan.

Untuk menentukan nilai $h(n)$ ditunjukkan oleh persamaan :

$$h(n) = \sqrt{(Xn - Xgoal)^2 + (Yn - Ygoal)^2} \tag{2}$$

dimana :

$h(n)$ = nilai heuristik untuk *node*/ titik n

Xn = nilai koordinat X dari *node*/ titik n

Yn = nilai koordinat Y dari *node*/ titik n

$Xgoal$ = nilai koordinat X dari *node*/ titik tujuan

$Ygoal$ = nilai koordinat Y dari *node*/ titik tujuan

Adapun langkah-langkah dalam pencarian jalur terpendek dengan algoritma A* (*A-Star*) sebagai berikut [6]:

1. Masukkan *node* awal ke *openlist*.
2. Lakukan perulangan langkah-langkah berikut :
 - Cari *node* (n) dengan nilai $f(n)$ yang paling rendah dalam *openlist*. *Node* ini sekarang menjadi *current node*.
 - Keluarkan *current node* dari *open list* dan masukan ke *close list*.
 - Untuk setiap tetangga dari *current node* lakukan langkah berikut:
 1. Jika tidak dapat dilalui atau sudah ada dalam *close list*, abaikan.
 2. Jika belum ada di *open list*. Buat *current node parent* dari *node* tetangga. Simpan nilai $f(n)$, $g(n)$ dan $h(n)$ dari *node* ini.
 3. Jika sudah ada di *open list*, cek bila *node* tetangga ini lebih baik, menggunakan nilai g sebagai ukuran. Jika lebih baik ganti *parent* dari *node* ini di *open list* menjadi *current node*, lalu kalkulasi ulang nilai g dan f dari *node* ini.
 - Hentikan perulangan jika:
 1. *Node* tujuan telah ditambahkan ke *open list*, yang berarti rute telah ditemukan.



2. Belum menemukan *node goal* sementara *open list* kosong yang berarti tidak ada rute.
3. Simpan rute. Secara *backward*, urut mulai dari *node goal* ke *parent* masing-masing sampai mencapai *node* awal sambil menyimpan *node* kedalam sebuah *array*.

2. Metode Penelitian

Metode Penerapan algoritma A* (*A-Star*) untuk mencari rute Rumah Sakit di kota Bogor berbasis android ini dibuat secara bertahap menggunakan model *System Development Life Cycle* (SDLC) [7]. Model SDLC mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dengan tahapan perencanaan sistem, analisis sistem, perancangan, implementasi dan uji coba.

Metode Pengumpulan Data

Pada metode pengumpulan data dilakukan studi kepustakaan dan studi langsung.

1. Studi Pustaka

Pada metode ini hal yang dilakukan adalah mengumpulkan dan mempelajari buku-buku yang berhubungan dengan masalah yang dibahas dalam pengembangan aplikasi.

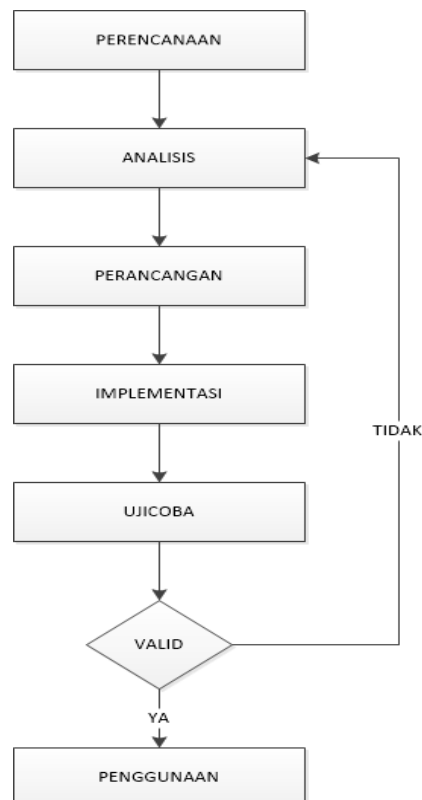
2. Studi Lapang

Pada metode ini dilakukan teknik Observasi (Pengamatan).

Pada metode observasi, dilakukan penelitian langsung ke salah satu RS di kota Bogor untuk memperoleh dan mengumpulkan data yang dibutuhkan, beserta meninjau langsung kondisi jalan yang dilewati dan kemacetan yang dialami pada saat proses observasi.

System Development Life Cycle (SDLC)

Metode yang digunakan dalam melaksanakan penelitian menggunakan pola *system development life cycle* (SDLC). Tahapan pembuatan aplikasi disajikan secara lengkap pada Gambar 1 di bawah ini:



Gambar 1 Pendekatan SDLC

Tahap Perencanaan

Tahap perencanaan sistem adalah tahap awal untuk mendapatkan *resources* yang digunakan untuk memperoleh solusi pada sistem yang akan dibuat. Sistem yang akan dibuat adalah sistem pencarian jalur terdekat Rumah Sakit di kota Bogor dengan menggunakan metode A* (*A-Star*). Sistem ini menerapkan cara melebar ke setiap jalur pada level yang sama, untuk menemukan rute terbaik dari titik awal sampai tujuan.

Tahap Analisis

Tahap analisis sistem adalah tahap setelah perencanaan dan tahap sebelum perancangan. Tahap analisis sistem merupakan tahap menganalisis permasalahan mengenai jalur terdekat dari satu titik ke titik yang lainnya dengan menggunakan metode A* (*A-Star*) guna mendapatkan jalur terbaik saat proses pengiriman barang berlangsung.

Tahap Perancangan

Tahap perancangan dilakukan guna memudahkan tahapan berikutnya yakni implementasi, tahap perancangan mencakup rancangan *flowchart view*, strukturnavigasi serta *design screen*. Didalam tahap perancangan sistem dibuat secara logika dengan menggunakan informasi-informasi yang telah terkumpul dalam bentuk data dan telah dilakukan analisis.

Tahap Implementasi

Tahap implementasi sistem merupakan suatu proses transformasi hasil perancangan kedalam bahasa pemrograman yang akan digunakan. Hal ini termasuk penulisan kode program yang dimengerti oleh mesin komputer, adapun bahasa pemrograman yang digunakan adalah *java* dan menggunakan perangkat lunak pendukung lainnya sebagai pendukung.

Tahap Uji Coba

Uji coba sistem adalah pengujian yang dilakukan setelah pembuatan sistem selesai dibuat dengan melakukan percobaan pada *user interface*.

3. Hasil dan Pembahasan

Adapun hasil penelitian yang diperoleh adalah terbentuknya aplikasi pencarian jalur terdekat menuju rumah sakit di Kota Bogor, dalam aplikasi ini hasilnya adalah berupa rekomendasi rumah sakit yang terdekat dengan pengguna. Gambar 2 berikut ini adalah hasil tampilannya utama aplikasi:



Gambar 2. Tampilan utama Aplikasi

Halaman tampilan daftar Rumah Sakit

Halaman tampilan daftar Rumah Sakit adalah halaman yang menampilkan informasi rumah sakit dikota Bogor yang meliputi rumah sakit salak, RS Islam Bogor, Siloam, Bogor Medical



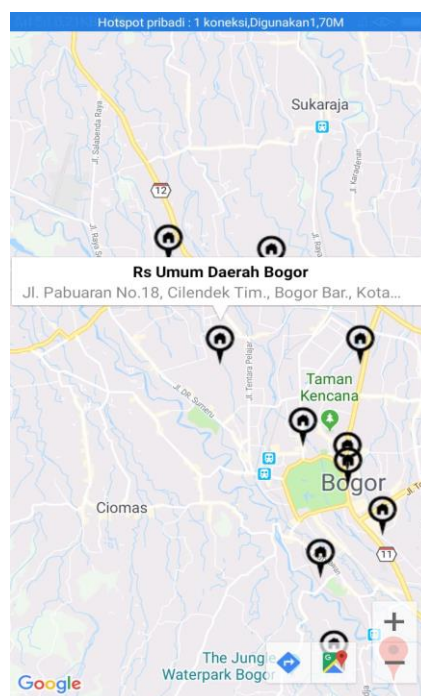
Center (BMC), RS Umum Daerah Bogor, RS PMI, RS Mulia Pajajaran dan rumah sakit Hermina, berikut tampilan halamannya pada Gambar 3.



Gambar 3. Halaman Daftar Rumah Sakit

Halaman tampilan lokasi Rumah Sakit

Halaman tampilan lokasi Rumah Sakit berisi lokasi masing-masing Rumah Sakit seperti diperlihatkan pada Gambar 4, tampilan lokasi ini diperoleh dari pencarian dengan menggunakan Google Map

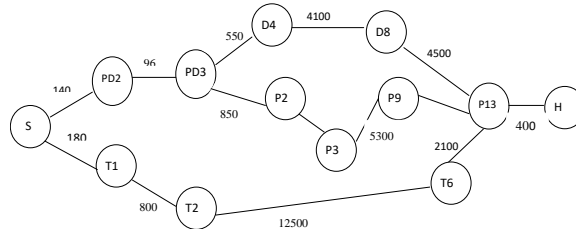


Gambar 4. Tampilan lokasi rumah sakit

Tahap berikutnya adalah uji coba sistem, pengujian sistem dilakukan untuk melihat apakah algoritma A* dapat menentukan atau mencari lintasan terpendek pada sistem. Pengujian meliputi pencarian jalur atau lintasan terpendek dari aplikasi yang telah dibuat dengan hasil perhitungan manual.

Pengujian Manual

Berikut ini adalah contoh hasil dari perhitungan manual pencarian rute terdekat toko bangunan di kota Bogor dengan menggunakan algoritma A* (A-Star). Dalam implementasi pencarian rute ini akan dicari rute terdekat dari titik awal yang menggunakan marker S sebagai symbol titik awal, menuju marker H sebagai titik tujuan seperti diperlihatkan pada Gambar 5 berikut ini:



Gambar 5. Rute rumah sakit dalam bentuk graf

Proses Pencarian Rute Terdekat

1. Menentukan jarak antara dua titik koordinat yang berhubungan. Pada tabel 2 dapat dilihat jarak antara dua titik koordinat dimana jarak tersebut diasumsikan kedalam $g(n)$.

Tabel 1. Jarak Antara Dua Titik Koordinat $g(n)$

Id titik asal	Id titik tujuan	Jarak $g(n)$
S	T1	180
S	PD2	140
T1	T2	800
T2	T6	12500
T6	P13	2100
PD2	PD3	96
PD3	P2	850
P2	P3	550
P3	P9	5300
P9	P13	3400
PD3	D4	550
D4	D8	4100
D8	P13	4500
P13	H	400

2. Menentukan titik koordinat longitude dan latitude pada peta

Untuk menentukan titik koordinat suatu *node*, pada google maps pilih lokasi awal dan lokasi tujuan kemudian diambil titik-titik lokasi persimpangan jalan yang akan dilalui kemudian dari koordinat longitude latitude tersebut di konversi ke bentuk decimal. Adapun hasil penentuan titik koordinat setiap *node*/ titik adalah sebagai berikut:



Tabel 2. Titik koordinat setiap node/titik

Node	Titik koordinat longituddan latitude	Konversi ke decimal
S	-6.599549, 106.812832	696033.84 , 9275013.6
T1	.600411, 106.811906	700224.83 , 9270155.4
T2	-6.603704, 106.807732	699761.87 , 9269792.86
T6	-6.562756, 106.796167	698499.08 , 9274326.39
PD2	-6.599750, 106.813000	700346.09 , 9270228.07
PD3	-6.600454, 106.813537	700405.2 , 9270149.99
P2	-6.605036, 106.807458	699731.03 , 9269645.65
P3	-6.604463, 106.806976	699677.95 , 9269709.22
P9	-6.582802, 106.787627	697543.97, 9271386.6
P13	-6.556502, 106.777279	696412.39 , 9275025.53
D4	-6.597896, 106.817592	700854.72 , 9270431.27
D8	-6.569014, 106.809234	699941.89 , 9273629.04
H	-6.557431, 106.773897	696037.95 , 9274924.11

Tabel 3. Perhitungan g(n), h(n) dan f(n) dengan algoritma A*

Node	Titik koordinat konversi ke decimal	Jarak antar node	g(n)	h(n)	f(n)
S	T1 700224.83, 9270155.4	S-T1= 180	180	6345.90	6525.90
	T2 699761.87 , 9269792.86	T1-T2 = 800 T2-S = 980	980	6340.13	7230.13
	T6 698499.08, 9274326,39	T2-T6 = 12500 T6-S = 13480	13480	2532.67	16012.67
	P13 696412.39, 9275025.53	T6-P13 = 2100 T6-S = 15580	15580	387.93	15967.93
	D4 700854.72,9270431.27	PD3-D4 = 550 D4-S = 786	786	6586.87	7372.87
	D8 699941.89,9273629.04	D4-D8= 4280 D8-S = 5066		4113.14	9179.14
	PD2 700346.09,9270228.07	S-PD2=140	140	6372.82	6512,82
Node	Titik koordinat konversi ke decimal	Jarak antar node	g(n)	h(n)	f(n)
	PD3 700405.2,9270149.99	PD2-PD3 = 96 S-PD3 = 236	236	6470.32	6706.32

	P2 699731.03,9269645.65	PD3-P2 = 850 P2-S = 1086	1086	6442.12	7528.12
	P3 699731.03,9269645.65	P2-P3 = 550 P3-S = 1636	1636	6359.61	7995.61
	P9 697543.97,9271386.6	P3-P9 = 5300 P9-S = 6936	6936	3844.77	3844.77
	P13 696412.39,9275025.53	T6-P13 = 2100 T6-S = 15580	15580	387.93	15967.93
		P9-P13 = 3400 P13-P9-P3-P2-PD3- PD2-S= 10336	10336		10723.93
		P13-D8 = 4500 P13-D8-D4-PD3-PD2- S = 9566	9566		9953.93
	H 696037.95,9274924.11	H-P13 = 400 H-S lewat P3 = 10086	10086		10086

Perhitungan untuk mencari nilai $f(n)$

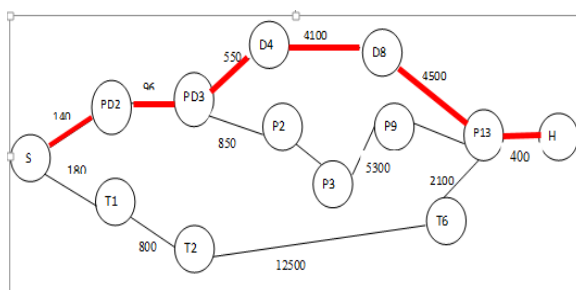
Untuk mendapatkan jalur terpendek menuju RS Hermina dengan langkah-langkah sebagai berikut:

- a) Dimulai dari start node (S = Universitas Pakuan), queue-nya adalah Queue {[S]}. Terdapat 2 node dari S yaitu node T1 dan PD2, cari nilai $f(n)$ terkecil dari node tersebut, $f(n)$ T1 merupakan $f(n)$ terkecil maka node T1 sebagai current-node. Hasil queue-nya adalah Queue {[S,T1, $g(n)$ 180 $f(n)$ 6525.90] [S, PD2, $g(n)$ 140 $f(n)$ 6512]}
- b) Lakukan tahap berikutnya seperti tahap a)
- c) Hasil Queue adalah sebagai berikut :
 - Queue {[S]}.
 - Queue { [S,T1, 180 $f(n)$ 6525.90] [S,PD2,Jarak 140 $f(n)$ 6512] }
 - Queue { [S,T1, 180 $f(n)$ 6525.90] [S,PD2,Jarak 140 $f(n)$ 6512] }
 - Queue { [S,T1, (180) $f(n)$ 6525.90] [S,PD2,**PD3** $g(n)$ =(140+96) $f(n)$ 6706.324] }
 - Queue { [S,T1, **T2** (180+800) $f(n)$ 7230.13] [S,PD2,PD3 Jarak (140+96) $f(n)$ 6706.324] }
 - Queue { [S,T1, T2 (180+800) $f(n)$ **7230.13**] [S,PD2,PD3,**D4** Jarak (140+96+550) $f(n)$ 7372.87] [S,PD2,PD3,P2 Jarak (140+96+850) $f(n)$ 7528.12] }
 - Queue { [S,T1, T2,**T6** $g(n)$ =(180+800+12500) $f(n)$ 16012.67] [S,PD2,PD3,D4 $g(n)$ =(140+96+550) $f(n)$ 7372.87213] [S,PD2,PD3,P2 $g(n)$ =(140+96+850) $f(n)$ **7528.12**] }
 - Queue { [S,T1, T2,T6 (180+800+12500) $f(n)$ 16012.67] [S,PD2,PD3,D4,D8 Jarak (140+96+550+4280) $f(n)$ 9179.144] [S,PD2,PD3,P2,**P3** Jarak (140+96+850+550=1636) $f(n)$ **7995.613**] }
 - Queue { [S,T1, T2,T6 (180+800+12500) $f(n)$ 16012.67] [S,PD2,PD3,D4,D8 Jarak (140+96+550+4280) $f(n)$ **9179.144**] [S,PD2,PD3,P2,P3,**P9** Jarak (P9-S=5300+550+850+96+140= 6936) $f(n)$ 10780.77] }
 - Queue { [S,T1, T2,T6 (180+800+12500) $f(n)$ 16012.67] [S,PD2,PD3,D4,D8,**P13** $g(n)$ =(4500+4280+550+96+140=9566) $f(n)$ **9953.93**] [S,PD2,PD3,P2,P3,P9 $g(n)$ =(P9-S=5300+550+850+96+140= 6936) $f(n)$ 10780.77] }
 - Queue { [S,T1, T2,T6 (180 + 800 + 12500) $f(n)$ 16012.67] [S, PD2, PD3, D4, D8, P13, **H** $g(n)$ =(400+4500+4280+550+96+140=10086) $f(n)$ **10086**] [S,PD2,PD3,P2,P3,P9 $g(n)$ =(5300+550+850+96+140= 6936) $f(n)$ 10780.77] }
- d) Proses dihentikan karena sudah didapatkan node tujuan dengan $f(n)$ terkecil.



- e) Adapun hasil jalur dari Universitas Pakuan menuju RS Hermina yang didapatkan dengan metode algoritma A* adalah dari S-PD2-PD3-D4-D8-P13-H.

Hasil pencarian jalur terpendek menggunakan algoritma A* ditunjukkan pada Gambar 6.



Gambar 6. Hasil Analisis Jalur Terpendek pada rumah sakit Hermina Kota Bogor

4. Kesimpulan

Penerapan algoritma A* untuk pencarian jalur terpendek menuju rumah sakit di Kota Bogor dibangun dengan menggunakan *software Android Studio*, dengan bahasa pemrograman *Java* dan *XML* untuk *user interface*. Adapun tahapan penelitian dengan menggunakan model SDLC, dengan tahapannya meliputi perencanaan sistem, analisis sistem, perancangan, implementasi dan uji coba. Penerapan Algoritma A* pada aplikasi ini dapat memudahkan dalam menentukan jalur terpendek menuju rumah sakit di Kota Bogor. Aplikasi ini dijalankan pada sistem android, sehingga dapat memudahkan user untuk mengakses dimanapun dan kapanpun.

Referensi

- [1] Irsyad M dan Rasila E. 2015. Aplikasi pencarian lokasi gedung dan ruangan Universitas Islam Negeri Sultan Syarif Kasim Riau pada platform Android menggunakan algoritma A-Star(A*), *Jurnal CoreIT*, 1 (2), Desember 2015.ISSN:2460-738X.
- [2] Putra, R. D., Aswin, M., & Djurianto, W. 2012. Pencarian Rute Terdekat Pada Labirin Menggunakan Metode A*. *Jurnal EECCIS*, 6(2).
- [3] Juhara, Zamrony P. 2016. *Panduan Lengkap Pemrograman Android*. ANDI. Yogyakarta.
- [4] Imam Ahmad, Wahyu Widodo. 2017. Penerapan Algoritma A Star (A*) pada GamePetualangan Labirin Berbasis Android. *Khazanah Informatika*. 3(2): 57-62
- [5] Tosida, E, T., Walujo, A, D., Ardiansyah, D., dan Yuliani. 2018. Media Belajar Batik Berbasis Teknologi Augmented Reality. *Prosiding Kevin Prathama Nugraha1, Dr. Ir. BambangHidayat, IPM2, Leanna VidyaYovita, S.T., M.T.* 2017. PerancanganAplikasi Dan ImplementasiPencarianLokasiTerdekatPadaKawasan Telkom University Berbasis Android. *e-Proceeding of Engineering*: 4(2) Agustus 2017. ISSN: 2355-9365.
- [6] Muh. Yamin, Moh. Bandrigo Talai. 2017. Aplikasi Pencarian Jalur Terpendek Pada Rumah Sakit Umum Bahteramas Menggunakan Algoritma A* (A-Star). *Jurnal Informatika* 9(2): 1065-1078
- [7] Yoki Firmansyah, Udi. 2018. Penerapan Metode SDLC Waterfall Dalam Pembuatan Sistem Informasi Akademik Berbasis Web Studi Kasus Pondok Pesantren Al-Habi Sholeh Kabupaten Kubu Raya, Kalimantan Barat. *Jurnal Teknologi & Manajemen Informatika* 4(1): 184-191.