

PROTOTIPE PERANGKAT PENDETEKSI LOKASI MOBIL PELAYANAN UMUM BERGERAK BERBASIS GEOLOCATION MENGGUNAKAN ALGORITMA PENCARIAN A*

Ridho Chandra Sani¹, Ilhamsyah², Suhardi³

^{1,3}Jurusan Rekayasa Sistem Komputer, ²Jurusan Sistem Informasi
Fakultas MIPA Universitas Tanjungpura
Jalan Prof. Dr. H. Hadari Nawawi Pontianak
Telp./Fax : (0561) 577963

e-mail: ridhochandrasani@yahoo.com, ilhamsyah@sisfo.untan.ac.id,
suhardi@siskom.untan.ac.id

Abstrak

Penelitian ini menerapkan algoritma pencarian A dalam pendeteksian lokasi mobil pelayanan umum bergerak menggunakan Arduino Uno, GPRS Shield, GPS, dan aplikasi mobile. Algoritma A* merupakan perbaikan dari metode best first search dengan memodifikasi fungsi heuristiknya. Algoritma A* akan meminimumkan total biaya lintasan. GPS berperan menangkap koordinat posisi dari perangkat yang terpasang di mobil. Arduino mengolah data yang diterima dari perangkat GPS dan mengirimkan ke server melalui GPRS shield yang terhubung dengan internet. Pengujian dilakukan dengan menguji data yang masuk melalui sistem serta membandingkan hasil pencarian lokasi mobil pelayanan umum terdekat menggunakan algoritma A* secara manual dengan hasil pencarian pada aplikasi antarmuka. Dalam setiap menit, data yang masuk sebanyak 1 data dengan interval setiap 50 detik, sehingga pembaharuan lokasi mobil pelayanan umum dapat diketahui dengan lebih baik. Berdasarkan hasil perbandingan pada pengujian, algoritma A* telah berhasil diterapkan pada aplikasi antarmuka berbasis Android dan dapat mendeteksi mobil pelayanan umum yang terdekat dengan pengguna..*

Kata kunci : Mobil Pelayanan Umum, Algoritma A*, GPS, Arduino

1. PENDAHULUAN

Mobil pelayanan umum adalah mobil yang saat ini digunakan oleh perbankan, dinas pemerintahan hingga *supermarket* bergerak. Dengan adanya mobil pelayanan umum ini masyarakat dapat melakukan transaksi yang dibutuhkan tanpa harus pergi ke kantornya. Oleh karena itu masyarakat membutuhkan informasi untuk mengetahui lokasi mobil pelayanan umum yang berpindah-pindah tempat dengan menggunakan *Global Positioning System* (GPS), sehingga masyarakat bisa mengetahui lokasi mobil yang terdekat dengan pengguna.

Pada penelitian sebelumnya, masyarakat dapat mengakses lokasi ATM (Anjungan Tunai Mandiri) terdekat melalui *smartphone*, sehingga pencarian lokasi

ATM bank dapat dilakukan secara praktis[1]. Kemudian terdapat penelitian pelacakan posisi kendaraan menggunakan *Global Positioning System* sehingga dapat memudahkan pengguna mengetahui koordinat posisi dari mobil pelayanan umum[2]. Penelitian tersebut menggunakan SMS (Short Message Service) dalam pengiriman dan penerimaan data namun penelitian tersebut menggunakan 2 buah *handphone* untuk mengirim dan menerima SMS. Permasalahan tersebut telah diatasi dengan menggunakan ITEAD SIM900 GPRS/GPM untuk mengirim SMS ke *user* dalam pengaturan sistem keamanan dan pemantauan lokasi mobil[3]. Namun di era saat ini penggunaan layanan SMS sudah mulai jarang digunakan dibandingkan dengan layanan internet, hal ini disebabkan

respon sistem SMS lebih lambat daripada internet.

Penelitian ini membahas penyelesaian masalah pendeteksian mobil pelayanan umum dengan menggunakan kajian yang telah dilakukan pada penelitian sebelumnya dan juga memberikan solusi terhadap saran dari penelitian tersebut. Penelitian ini menggunakan internet sebagai transmisi data dan menggunakan GPS sebagai pendeteksi posisi mobil pelayanan umum. Hasil dari penelitian ini diharapkan dapat memudahkan masyarakat dalam mendeteksi mobil pelayanan umum yang berpindah-pindah tempat.

2. LANDASAN TEORI

2.1. Algoritma A*

Algoritma A* merupakan perbaikan dari metode *best first search* dengan memodifikasi fungsi heuristiknya. Algoritma A* akan meminimumkan total biaya lintasan. Pada kondisi yang tepat, Algoritma A* akan memberikan solusi yang terbaik dalam waktu yang optimal[4]. Heuristik adalah kriteria, metoda, atau prinsip-prinsip untuk menentukan pilihan sejumlah alternatif untuk mencapai sasaran dengan efektif. Nilai heuristik digunakan untuk mempersempit ruang pencarian[5]. Metode pencarian A* menghasilkan pemilihan optimal mulai dari tempat awal kemudian melalui *graph* menuju tempat yang dituju[6]. Metode ini berdasarkan formula:

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

Keterangan:

f(n) = Solusi biaya estimasi termurah
Node awal ke Node tujuan

g(n) = Biaya sebenarnya dari Node awal ke Node tujuan

h'(n) = Biaya estimasi dari Node awal ke Node tujuan

2.2. Haversine Formula

Pada penelitian ini menggunakan pengukuran jarak berdasarkan posisi bujur dan lintang bumi untuk menentukan posisi terdekat suatu titik dengan menggunakan metode haversine. Metode tersebut digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur

(*latitude*). *Haversine formula* adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang[7]. Rumus Haversine dapat ditulis dengan persamaan sebagai berikut:

$$\begin{aligned} x &= (\text{lon}2 - \text{lon}1) \cos\left(\frac{\text{lat}1 + \text{lat}2}{2}\right); \\ y &= (\text{lat}2 - \text{lat}1); \\ d &= \sqrt{x^2 + y^2} \cdot R \end{aligned} \quad (2)$$

Keterangan:

x = *Longitude* (Lintang)

y = *Lattitude* (Bujur)

d = Jarak

R = Radius Bumi = 6371 km

1 derajat = 0.0174532925 radian

2.3. Arduino UNO

Arduino UNO adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328 (*datasheet*). Arduino UNO mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya[8]. Pada penelitian ini Arduino Uno digunakan sebagai pengolah data dan mengirim data koordinat mobil pelayanan umum ke *server*. Perangkat Arduino dapat dilihat pada Gambar 1.



Gambar 1. Arduino Uno

2.4. *Global Positioning System (GPS) Shield*

GPS adalah sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi secara instan, kecepatan dan Informasi waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca apapun[9]. Sedangkan alat untuk menerima sinyal satelit yang dapat digunakan oleh pengguna secara umum dinamakan *GPS Tracker* atau *GPS Tracking*, dengan menggunakan alat ini maka dimungkinkan *user* dapat melacak posisi kendaraan, armada ataupun mobil dalam keadaan *Real-Time*. Perangkat *GPS Shield* dapat dilihat pada Gambar 2.



Gambar 2. *GPS Shield*

2.5. *GSM Shield*

Arduino *GSM Shield* adalah sebuah *board* yang dirancang terintegrasi dengan arduino dengan fungsi untuk dapat mengirim sms, membuat *voice call* atau mengkoneksi internet dengan menggunakan *wireless network*. Untuk menjalankan *GSM shield* dengan arduino cukup hanya masukan pin dari *GSM shield* ke pin arduino uno yang telah tersedia[10]. Arduino *GSM Shield* memungkinkan arduino untuk terkoneksi dengan internet, melakukan panggilan suara dan mengirim/menerima sms dengan menggunakan radio modem M10 dari Quectel, yang dapat memungkinkan komunikasi dengan arduino dengan menggunakan *AT commands*. Pada penelitian ini *GSM Shield* digunakan sebagai koneksi internet untuk mengirim koordinat mobil pelayanan umum ke *server*. Perangkat *GSM Shield* dapat dilihat pada Gambar 3.



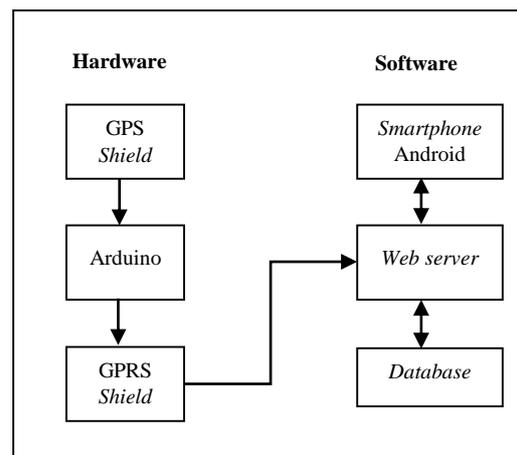
Gambar 3. *GSM Shield*

3. METODE PENELITIAN

Proses pertama dimulai dengan studi pustaka yang terkait dengan Algoritma A*, *Haversine formula*, Arduino Uno, *GPS Shield*, *GSM Shield* dan teori penunjang lainnya. Selanjutnya dilakukan analisis kebutuhan dan perancangan sistem yang kemudian diintegrasikan menjadi suatu sistem sehingga berfungsi sebagaimana mestinya. Selanjutnya dilakukan pengujian untuk mengetahui kinerja sistem. Setelah dilakukan pengujian, dilakukan analisa terhadap hasil dari pengujian dan diterapkan pada sistem pendeteksian lokasi mobil pelayanan umum bergerak setelah sistem bekerja dengan baik.

4. PERANCANGAN SISTEM

Pada perancangan sistem, digunakan diagram blok yang menjelaskan hubungan antara perangkat keras dan perangkat sistem. Diagram blok perancangan perangkat sistem dapat dilihat pada gambar 4.



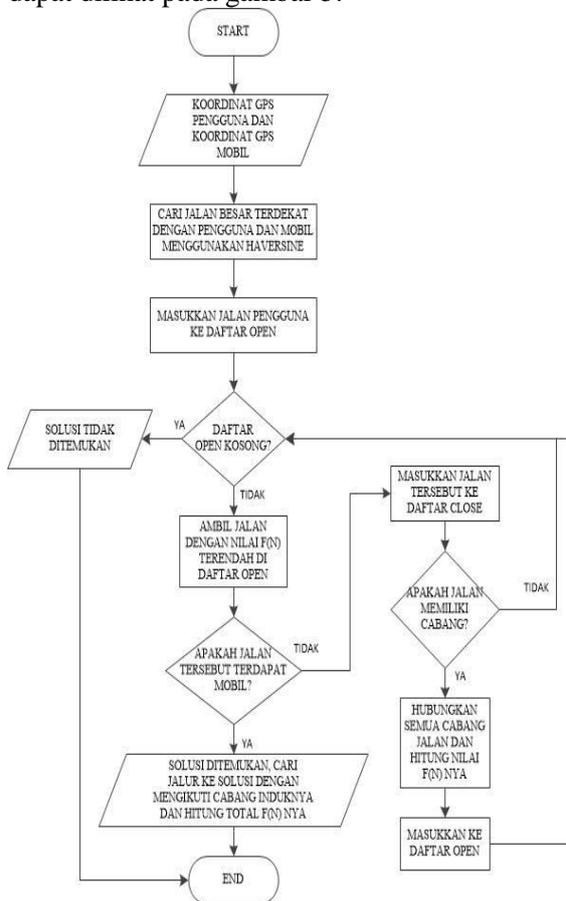
Gambar 4. Diagram Blok

Penelitian ini menggunakan perangkat pada diagram blok sebagai berikut:

- GPS Shield* menangkap koordinat posisi Mobil Pelayanan Umum.

- b. Arduino mengolah data yang dibaca GPS dan kemudian dikirim ke *Server* melalui internet menggunakan *GPRS Shield* sebagai modem. Sehingga data koordinat posisi dapat disimpan pada *server*.
- c. *Smartphone* Android mendeteksi lokasi pengguna dan mengirim posisi ke *web server*.
- d. *Database MySQL* merupakan tempat penyimpanan data dari sistem.
- e. *PHP* menyediakan layanan dan transaksi antar data sehingga memberikan data hubungan antara pengguna dan mobil pelayanan umum.

Diagram alur metode algoritma A* dapat dilihat pada gambar 5.



Gambar 5. Diagram Alur Algoritma A*

Berdasarkan Gambar 5. sistem menerima koordinat posisi pengguna dan mobil pelayanan umum serta mencari jalan besar terdekat menggunakan haversine. Jalan yang terdekat dengan pengguna akan dianggap sebagai jalan awal dan memasukkannya ke dalam daftar *OPEN*.

Kemudian daftar *OPEN* dicek apakah kosong. Jika daftar *OPEN* kosong maka solusi tidak ditemukan. Jika tidak kosong maka sistem akan mengambil jalan yang memiliki nilai $f(n)$ terendah di daftar *OPEN* berdasarkan jarak dan waktu kemacetannya. Selanjutnya dicek apakah jalan tersebut terdapat mobil pelayanan umum. Jika iya, maka solusi ditemukan dan sistem mencari jalur dari jalan awal ke solusi dengan mengikuti cabang induknya dan menghitung total $f(n)$ nya. Jika bukan tujuan, maka jalan dimasukkan ke daftar *CLOSE*. Jika jalan tersebut memiliki cabang, maka akan dihubungkan semua cabangnya dan hitung nilai $f(n)$ masing-masing cabang serta masukkan ke dalam daftar *OPEN*. Kemudian dilakukan pengulangan dengan pengecekan kembali pada daftar *OPEN* hingga solusi ditemukan ataupun tidak.

Berdasarkan penelitian yang telah dilakukan oleh Darmawan[11] dan Fendi[12] terdapat data kemacetan yang digunakan sebagai bobot tambahan pada perhitungan $f(n)$ dapat dilihat pada tabel 1.

Tabel 1. Data Kemacetan

No	Jalan	Waktu	Kategori Macet	Bobot
1.	Jl. H. M. Suwignyo	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1
2.	Jl. Husin Hamzah	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1
3.	Jl. H. Rais A. Rahman	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1
4.	Jl. Putri Candramidi	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1
5.	Jl. S.S. Abdurrahman	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1
6.	Jl. Sultan Syahrir	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1
7.	Jl. Tanjungpura	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1
8.	Jl. A. Yani	07.00 – 09.00	Macet	1
		12.00 – 14.00	Ramai Lancar	0,5
		16.00 – 18.00	Macet	1

5. IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi Perancangan Perangkat Keras

Hasil dari perancangan perangkat keras pada sistem pendeteksi lokasi mobil pelayanan umum bergerak dapat dilihat pada gambar 6.

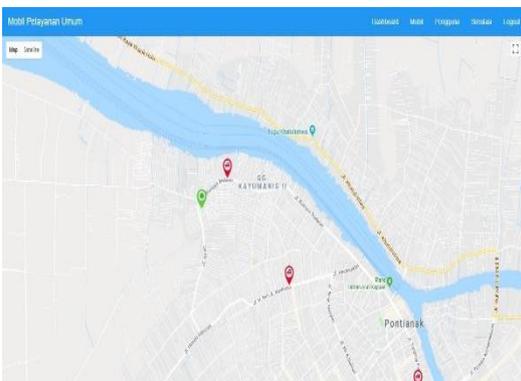


Gambar 6. Tampilan Perangkat Keras

Pada Gambar 6. dapat dilihat kabel antenna GPS, GPS *Shield* dengan *board* berwarna biru, Modem GSM dengan *board* berwarna Merah dan dikendalikan oleh Arduino Uno pada lapisan bawah Alat.

5.2. Implementasi Perangkat Lunak

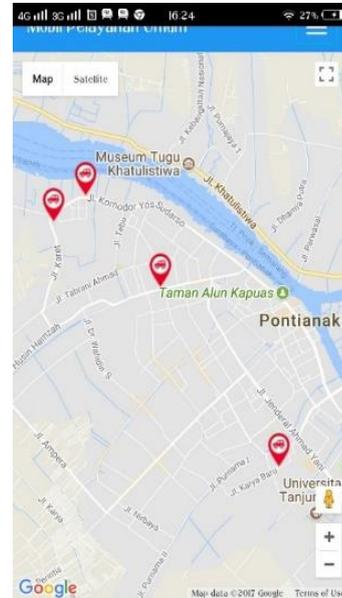
Hasil dari perancangan aplikasi berbasis *website* dapat diakses pada alamat <https://ridho.paskibra.id/admin/> akses alamat dapat menggunakan *web browser* yang hanya dapat diakses oleh *admin*. Implementasi perancangan tampilan halaman utama dapat dilihat pada gambar 7.



Gambar 7. Tampilan Halaman Utama *Admin*

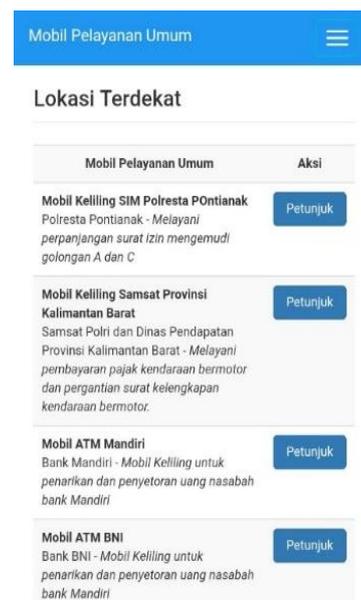
Hasil dari perancangan aplikasi berbasis Android dapat diakses oleh pengguna dengan memasang aplikasi

“Mobil Pelayanan Umum”. Aplikasi hanya dapat diakses oleh pengguna *smartphone* Android yang memiliki fitur GPS dan mendukung HTML 5. Implementasi perancangan tampilan halaman utama *user* dapat dilihat pada gambar 8.



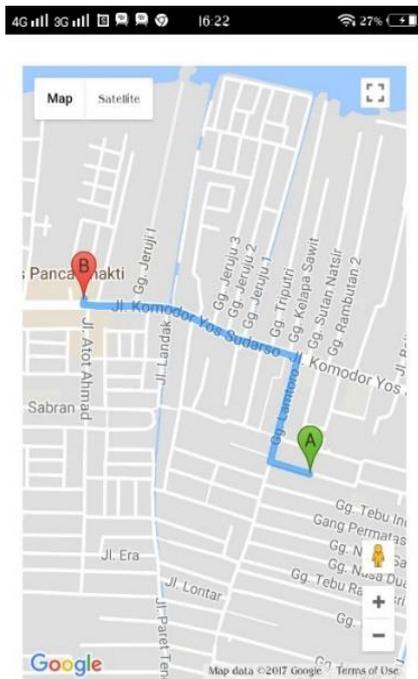
Gambar 8. Tampilan Halaman Utama *User*

Gambar 9 merupakan implementasi perancangan tampilan halaman “Lokasi Terdekat” untuk melihat mobil pelayanan umum yang terdekat, sistem menggunakan Algoritma A* untuk menentukan lokasi mobil terdekat.



Gambar 9. Tampilan Lokasi Mobil Terdekat

Gambar 10 merupakan implementasi perancangan tampilan aplikasi petunjuk arah lokasi mobil dan pengguna. saat menu petunjuk pada aplikasi ditekan, maka akan tampil rute petunjuk arah menuju lokasi mobil pelayanan umum terdekat.



Gambar 10. Tampilan Halaman Rute Terpendek

5.3. Pengujian

a. Pengujian Perangkat Keras

Dalam implementasi dapat diuji data yang masuk melalui sistem pada hubungan perangkat keras dan *server*. Dengan menggunakan MariaDB dapat dilihat *database* MySQL yang masuk ke sistem selama 5 menit adalah dengan interval selama 50 detik dan rata-rata penyimpanan data pada *server* adalah 1 data setiap menit.

Pengujian selanjutnya digunakan pengambilan data selama 30 menit adalah dengan rata-rata interval selama 50 detik dan rata-rata penyimpanan data pada *server* adalah 1 data setiap menit, data ini dapat perbedaan karena terdapat kesalahan data dalam pengiriman koordinat pada *server*. Hal ini disebabkan karena jaringan internet yang tidak stabil. Pengujian selanjutnya digunakan pengambilan data selama 1 jam adalah dengan rata-rata

interval selama 50 detik dan rata-rata penyimpanan data pada *server* adalah 1 data setiap menit, data ini dapat perbedaan karena terdapat kesalahan data dalam pengiriman koordinat pada *server*. Hal ini disebabkan karena jaringan internet yang tidak stabil saat mengirimkan data pada Arduino ke *Server*. Pengujian perangkat dan transmisi data dapat dilihat pada tabel 2.

Tabel 2. Pengujian Perangkat dan Transmisi Data

No	Lama Pengujian	Interval Data	Rata-rata Pengiriman Permenit
1	5 Menit	50 Detik	1 Data
2	30 Menit	50 Detik	1 Data
3	1 Jam	50 Detik	1 Data

Berdasarkan pada tabel 1, maka dapat disimpulkan bahwa sistem dapat bekerja optimal setiap menit karena masih dapat mengirimkan 1 data pada setiap menit dengan interval setiap 50 detik, sehingga *update* lokasi mobil pelayanan umum dapat diketahui dengan lebih baik.

b. Pengujian Algoritma A*

Pengujian dilakukan dengan membandingkan hasil pencarian lokasi terdekat secara manual menggunakan algoritma A* dengan pengujian menggunakan aplikasi berbasis Android. Pengujian ini menggunakan 2 buah sampel titik koordinat yang berada pada ujung persimpangan masing-masing jalan serta waktu mengakses aplikasi pada pukul 16.00. Lokasi pengguna saat mengakses aplikasi berada di Mess Ketapang di Jl. Ahmad Sood dengan koordinat -0.039465, 109.334797 dan lokasi mobil pelayanan umum berada di parkir ruko Jalan Sultan Abdurrahman dengan koordinat -0.040918, 109.326797. Jalan Sultan Abdurrahman terbagi menjadi 2 yaitu Jalan Sultan Abdurrahman 1 dan Jalan Sultan Abdurrahman 2 karena terdapat jalan Putri Candramidi yang memotong jalan tersebut. Mula-mula aplikasi menggunakan haversine untuk mencari jalan utama yang terdekat dengan posisi pengguna dan mobil. Kemudian jalan utama terdekat dengan pengguna akan dianggap sebagai *node* awal dan jalan

utama terdekat dengan mobil akan dianggap sebagai tujuan. Pada pengujian ini lokasi pengguna setelah dilakukan haversine terhadap jalan utama terdekat berada pada -0.039040, 109.335615 yaitu jalan Ahmad Yani dan lokasi mobil berada pada -0.040708, 109.326859 yaitu jalan Sultan Abdurrahman.

Algoritma A* dimulai dari lokasi jalan utama terdekat pengguna yaitu jalan Ahmad Yani sebagai *node* awal. Pada jalan Ahmad Yani diambil 2 buah sampel titik koordinat yaitu -0.035505, 109.332939 dan -0.046491, 109.341994. Kemudian aplikasi akan menghitung jarak dari masing-masing koordinat tersebut ke koordinat-koordinat pada jalan lain untuk menentukan berapa banyak cabang atau simpul dari jalan Ahmad Yani. Perhitungan jarak (d) antar koordinat dengan menggunakan *formula* haversine dapat dilihat sebagai berikut:

Jalan Ahmad Yani: -0.035505, 109.332939

Lat1 = -0.035505 = -0.000619679 rad

Lon1 = 109.332939 = 1.908219764 rad

Jalan Veteran: -0.046381, 109.342251

Lat2 = -0.046381 = -0.000809501 rad

Lon2 = 109.342251 = 1.908382289 rad

$$x = (\text{lon}2 - \text{lon}1) \cos\left(\frac{\text{lat}1 + \text{lat}2}{2}\right)$$

x

$$= (1.90838228$$

$$- 1.90821976) \cos\left(\frac{-0.00061967 \pm 0.00080950}{2}\right)$$

$$x = 0.000162525$$

$$y = (\text{lat}2 - \text{lat}1)$$

$$y = (-0.000809501 - -0.000619679)$$

$$y = -0.000189822$$

$$d = \sqrt{x^2 + y^2} \cdot R$$

$$d = \sqrt{0.000162525^2 + (-0.000189822)^2} \cdot 6371$$

$$d = 1,592071393 \text{ km}$$

Perhitungan jarak antar koordinat lainnya dengan menggunakan haversine dapat dilihat pada tabel 3.

Tabel 3. Perhitungan Nilai d dari Jl. Ahmad Yani

No	Nama Jalan		Koordinat		Nilai D (km)
	Asal	Tujuan	Asal	Tujuan	
1.	Jl. Ahmad Yani	Jl. Veteran	-0.035505, 109.332939	-0.041415, 109.345323	1,525 8095
2.	Jl. Ahmad Yani	Jl. KHA. Dahlan	-0.035505, 109.332939	-0.035279, 109.332801	0,029 4446

Tabel 3. Perhitungan Nilai d dari Jl. Ahmad Yani (Lanjutan)

No	Nama Jalan		Koordinat		Nilai D (km)
	Asal	Tujuan	Asal	Tujuan	
3.	Jl. Ahmad Yani	Jl. KHA. Dahlan	-0.035505, 109.332939	-0.027543, 109.327769	1,055 6044
4.	Jl. Ahmad Yani	Jl. Sultan Abdurrahman 1	-0.035505, 109.332939	-0.035536, 109.332769	0,019 2148
5.	Jl. Ahmad Yani	Jl. Sultan Abdurrahman 1	-0.035505, 109.332939	-0.039521, 109.328162	0,323 6640
6.	Jl. Ahmad Yani	Jl. Putri Candramidi	-0.035505, 109.332939	-0.039495, 109.328037	0,702 8160
7.	Jl. Ahmad Yani	Jl. Putri Candramidi	-0.035505, 109.332939	-0.032360, 109.322902	1,169 5695
8.	Jl. Ahmad Yani	Jl. Sultan Abdurrahman 2	-0.035505, 109.332939	-0.039609, 109.328062	0,708 7569
9.	Jl. Ahmad Yani	Jl. Sultan Abdurrahman 2	-0.035505, 109.332939	-0.042710, 109.324565	1,228 3686
10.	Jl. Ahmad Yani	Jl. Veteran	-0.046491, 109.341994	-0.046381, 109.342251	0,031 0846
11.	Jl. Ahmad Yani	Jl. Veteran	-0.046491, 109.341994	-0.041415, 109.345323	0,674 9816
12.	Jl. Ahmad Yani	Jl. KHA. Dahlan	-0.046491, 109.341994	-0.035279, 109.332801	1,612 2119
13.	Jl. Ahmad Yani	Jl. KHA. Dahlan	-0.046491, 109.341994	-0.027543, 109.327769	2,634 5859

Dari perhitungan tersebut, terdapat 3 buah jalan yang memiliki jarak kurang dari 50 meter yaitu jalan KHA Dahlan, jalan Sultan Abdurrahman1 dan jalan Veteran. Sehingga dapat disimpulkan bahwa 3 buah jalan tersebut merupakan 3 buah simpul dari jalan Ahmad Yani. Proses selanjutnya, dihitung nilai f(n) dari masing-masing jalan tersebut. Perhitungan nilai f(n) dengan menggunakan haversine dan algoritma A* dapat dilihat sebagai berikut:

Jalan Danau Sentarum: -0.041137,

109.313633

Jalan Danau Sentarum: -0.055368,

109.302957

g(n) = 1,978201983 km

Karena jalan Danau Sentarum tidak terdapat kemacetan pada pukul 16.00, maka:

$$h'(n) = 0$$

$$f(n) = g(n) + h'(n)$$

$$f(n) = 1,978201983 + 0$$

$$f(n) = 1,978201983$$

Jalan Dr. Wahidin: -0.041106, 109.313628

Jalan Dr. Wahidin: -0.024328, 109.302972

$g(n) = 2,210099750$ km
Karena jalan Dr. Wahidin tidak terdapat kemacetan pada pukul 16.00, maka:

$$h'(n) = 0$$

$$f(n) = g(n) + h'(n)$$

$$f(n) = 2,210099750 + 0$$

$$f(n) = 2,210099750$$

Jalan Dr. Sutomo: -0.041074, 109.313720

Jalan Dr. Sutomo: -0.047821, 109.319138

$$g(n) = 0,962184518$$
 km

Karena jalan Dr. Sutomo tidak terdapat kemacetan pada pukul 16.00, maka:

$$h'(n) = 0$$

$$f(n) = g(n) + h'(n)$$

$$f(n) = 0,962184518 + 0$$

$$f(n) = \mathbf{0,962184518}$$

Berdasarkan perhitungan tersebut, didapat nilai $f(n)$ terkecil dimiliki oleh jalan Sultan Abdurrahman1. Selanjutnya dicek apakah terdapat mobil di jalan ini. Karena tidak, maka dilakukan perhitungan jarak antara koordinat jalan Sultan Abdurrahman1 dengan koordinat jalan lainnya selain jalan yang telah didapat nilai $f(n)$ -nya untuk mencari cabang atau simpulnya.

Perhitungan jarak antar koordinat dengan menggunakan haversine dapat dilihat pada tabel 5.

Tabel 5. Perhitungan Nilai d dari Jl. Sultan Abdurrahman 1

No	Nama Jalan		Koordinat		Nilai d (km)
	Asal	Tujuan	Asal	Tujuan	
1.	Jl. Sultan Abdurrahman 1	Jl. Putri Candramidi	-0.035536, 109.332769	-0.039495, 109.328037	0,68604201
2.	Jl. Sultan Abdurrahman 1	Jl. Putri Candramidi	-0.035536, 109.332769	-0.032360, 109.322902	1,15259659
3.	Jl. Sultan Abdurrahman 1	Jl. Sultan Abdurrahman 2	-0.035536, 109.332769	-0.039609, 109.328062	0,69213968
4.	Jl. Sultan Abdurrahman 1	Jl. Sultan Abdurrahman 2	-0.035536, 109.332769	-0.042710, 109.324565	1,21183014
5.	Jl. Sultan Abdurrahman 1	Jl. Putri Candramidi	-0.039521, 109.328162	-0.039495, 109.328037	0,01419685
6.	Jl. Sultan Abdurrahman 1	Jl. Putri Candramidi	-0.039521, 109.328162	-0.032360, 109.322902	0,98799373
7.	Jl. Sultan Abdurrahman 1	Jl. Sultan Abdurrahman 2	-0.039521, 109.328162	-0.039609, 109.328062	0,01481189
8.	Jl. Sultan Abdurrahman 1	Jl. Sultan Abdurrahman 2	-0.039521, 109.328162	-0.042710, 109.324565	0,53452412

Dari perhitungan tersebut, terdapat 2 buah jalan yang memiliki jarak kurang dari 50 meter yaitu jalan Putri Candramidi dan jalan Sultan Abdurrahman2. Sehingga dapat

disimpulkan bahwa 2 buah jalan tersebut merupakan 2 buah simpul dari jalan Sultan Abdurrahman1.

Proses selanjutnya, dihitung nilai $f(n)$ dari masing-masing jalan tersebut. Perhitungan nilai $f(n)$ dengan menggunakan haversine dan algoritma A* dapat dilihat pada tabel 6.

Jalan Putri Candramidi: -0.039495, 109.328037

Jalan Putri Candramidi: -0.032360, 109.322902

$$g(n) = 0,97748145$$
 km

Karena jalan Putri Candramidi terdapat kemacetan pada pukul 16.00, maka:

$$h'(n) = 1$$

$$f(n) = g(n) + h'(n)$$

$$f(n) = 0,97748145 + 1$$

$$f(n) = 1,97748145$$

Jalan Sultan Abdurrahman2: -0.039609, 109.328062

Jalan Sultan Abdurrahman2: -0.042710, 109.324565

$$g(n) = 0,519712331$$
 km

Karena jalan Sultan Abdurrahman2 tidak terdapat kemacetan pada pukul 16.00, maka:

$$h'(n) = 1$$

$$f(n) = g(n) + h'(n)$$

$$f(n) = 0,519712331 + 0$$

$$f(n) = \mathbf{0,519712331}$$

Setelah mendapatkan nilai $f(n)$ pada jalan Putri Candramidi dan jalan Sultan Abdurrahman2, kemudian dibandingkan nilai $f(n)$ tersebut dengan nilai $f(n)$ pada jalan lainnya yang telah didapat sebelumnya. Setelah dibandingkan, didapatkan jalan Sultan Abdurrahman2 memiliki nilai $f(n)$ terkecil. Selanjutnya cek apakah terdapat mobil di jalan ini. Karena iya, maka tujuan telah ditemukan dan pencarian dihentikan. Kemudian aplikasi akan mencari rute yang dilalui sebelumnya sampai ke tempat tujuan dan menghitung nilai $f(n)$ totalnya.

Posisi Pengguna: -0.039040, 109.335615

Jalan Ahmad Yani: -0.035505, 109.332939

$$f(n)_{\text{orang ke jalan}} = 0,492998704$$
 km

$$f(n)_{\text{total}} = f(n)_{\text{ahmad yani}} + f(n)_{\text{sultan abdurrahman1}} +$$

$$f(n)_{\text{sultan abdurrahman2}}$$

$$f(n)_{\text{total}} = 0,492998704 + 0,677328312 +$$

$$0,519712331$$

$$f(n)_{\text{total}} = 1,690039347$$
 km

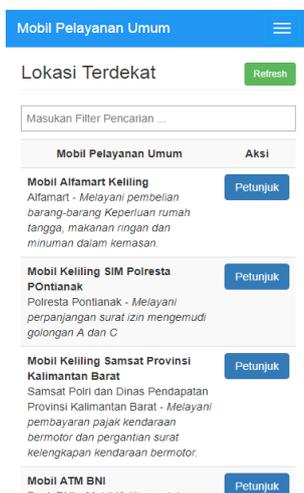
Hasil perhitungan manual algoritma A* melalui cabang jalan KHA Dahlan dan jalan Sultan Syarif Abdurrahman1 dapat dilihat pada tabel 7.

Tabel 7. Hasil Perhitungan Manual Algoritma A*

No.	Jalan Asal	Jalan Cabang	f(n) total (km)
1.	Jalan Ahmad Yani	Jalan Sultan Syarif Abdurrahman1	1,690039347
2.	Jalan Ahmad Yani	Jalan KHA. Dahlan	4,054436089

Berdasarkan tabel 6. didapat hasil f(n) total melalui cabang jalan Sultan Syarif Abdurrahman1 lebih kecil yaitu sebesar 1,69 km. Hal ini dikarenakan algoritma pencarian A* memilih rute dengan membandingkan jarak dan bobot kemacetan jalan.

Pengujian pada aplikasi antarmuka berbasis Android dilakukan dengan menggunakan koordinat posisi pengguna yang sama yaitu -0.039040, 109.335615. Hasil yang diperoleh pada aplikasi dari pengguna menuju mobil pelayanan umum terdekat dapat dilihat pada Gambar 11.



Gambar 11. Hasil Pencarian Lokasi Mobil Terdekat

Berdasarkan Gambar 11, hasil yang ditunjukkan pada aplikasi antarmuka berbasis Android sama dengan melakukan pencarian manual menggunakan algoritma A* yaitu menemukan mobil pelayanan umum yang berada di -0.040805, 109.326952 terlebih dahulu.

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dengan menerapkan algoritma A* untuk mengetahui posisi mobil pelayanan umum yang terdekat dengan posisi pengguna, dapat disimpulkan bahwa:

1. Telah berhasil dibuat sistem untuk mengetahui posisi pengguna dan mobil pelayanan umum menggunakan GPS dan aplikasi android untuk mempermudah pencarian mobil pelayanan umum.
2. Telah diterapkan algoritma A* dalam mencari rute terbaik dari pengguna menuju mobil pelayanan umum berdasarkan jarak yang didapat dari *Haversine formula* dan bobot kemacetan jalan.
3. Algoritma pencarian A* telah berhasil diterapkan pada aplikasi antarmuka berbasis Android dan dapat menemukan mobil pelayanan umum yang terdekat dengan pengguna. Hal ini didasarkan pada hasil pengujian yang ditunjukkan pada aplikasi antarmuka berbasis Android sama dengan melakukan pencarian manual menggunakan algoritma A* yaitu menemukan mobil pelayanan umum yang berada di koordinat -0.040805, 109.326952 terlebih dahulu.

6.2. Saran

1. Algoritma A* untuk menentukan mobil pelayanan umum ini masih menggunakan sampling koordinat jalan, untuk meningkatkan akurasi dibutuhkan data koordinat posisi jalan dan data koordinat gang untuk menghasilkan akurasi terbaik.
2. Pada hasil akhir yang menunjukkan rute direction (petunjuk arah) dari pengguna menuju mobil pelayanan umum masih menggunakan google *Maps* api direction, hal ini disebabkan karena pada penelitian ini tidak tersedia data koordinat jalan dan gang beserta data kemacetan di beberapa jalan. Oleh karena itu diharapkan untuk menentukan rute direction secara mandiri menggunakan data koordinat jalan atau gang dan data

kemacetan yang lebih lengkap di kota pontianak.

DAFTAR PUSTAKA

- [1] Yustika. 2014. *Rancang Bangun Aplikasi Android Untuk Pencarian Lokasi Atm Terdekat Di Kota Malang*. PTIIK.
- [2] Khalim, A. 2010. *Rancang Bangun Sistem Pelacak Posisi Kendaraan Menggunakan Global Positioning System (GPS)*. Jurnal ITS, Hal 1.
- [3] Pradata, R. 2013. *Pengaturan Sistem Keamanan dan Pemantauan Lokasi Mobil Dengan Fasilitas SMS*. Jurnal Universitas Brawijaya, 1.
- [4] Kusumadewi, S. 2003. *Artificial Intelligence (teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- [5] Pasaribu, N.T.Br. 2008. *Simulasi Pencarian Rute Terpendek Menggunakan Algoritma A* (A-Star)*. Jurnal Informatika Mulawarman Vol. 9 No. 5 Edisi September 2013.
- [6] Russel, S. dan Peter, N. 1995. *Artificial Intelligence A Modern Approach Second Edition*. New Jersey: Pearson Education Inc.
- [7] Putra, R. H. 2015. *Penerapan Metode Haversine formula pada Sistem Informasi Geografis Pengukuran Luas Tanah*. Jurnal Sistem dan Teknologi Informasi, Vol 1, No 1, Hal 1-6.
- [8] Kadir, A. 2013. *Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino*. Yogyakarta: Andi.
- [9] Aditya, A.H. 2003. *Mengenal Aspek Teknis dan Bisnis Location Based Services*. Jakarta: Elex Media Komputindo.
- [10] Adriansyah, A dan Hidyatama, O. 2013. *Rancang Bangun Prototipe Elevator Menggunakan Mikrokontroler Arduino Atmega 328P*. Jurnal Teknik Elektro UMB Vol. 4 No. 3.
- [11] Darmawan, E. 2014. *Identifikasi Titik Kemacetan Di Kecamatan Pontianak Utara, Pontianak Barat Dan Pontianak Kota*. Jurnal Teknik Sipil Universitas Tanjungpura Vol. 3. No.2.
- [12] Fendi, G.S. 2014. *Identifikasi Titik Kemacetan Dan Alternatif*

Penanganannya Di Kecamatan Pontianak Timur, Pontianak Selatan, Pontianak Tenggara. Jurnal Teknik Sipil Universitas Tanjungpura Vol. 2. No.2.