

Penerapan Metode *Haversine* pada Aplikasi Layanan Perbaikan Kendaraan Berbasis *Location Based Service*

(The Application of Haversine Method on Vehicle Repair Service App Using Location Based Service)

Algi Muhkamaat Abdillah¹, Rianto², Neng Ika Kurniati³

^{1,2,3}*Teknik Informatika – Universitas Siliwangi
Jl. Siliwangi No 35, Tasikmalaya, Jawa Barat*

¹147006149@student.unsil.ac.id

²rianto@unsil.ac.id

³nengikakurniati@unsil.ac.id

Abstrak — Kendaraan merupakan alat transportasi yang membantu masyarakat dalam melakukan aktivitas di kehidupan sehari-hari, tetapi selain manfaat menggunakan kendaraan bermotor, pada umumnya masyarakat juga dihadapi dengan resiko saat menggunakan kendaraan tersebut yaitu jika terjadi kerusakan pada mesin secara tiba-tiba atau biasa disebut dengan istilah mogok. Media informasi yang dapat diterapkan pada kasus kendaraan rusak secara mendadak tersebut adalah dengan membangun sebuah aplikasi android untuk memberikan informasi mengenai bengkel kendaraan yang berada di sekitar lokasi kendaraan pengguna yang mengalami kerusakan menggunakan *Location Based Service*. Tujuan dari penelitian ini adalah membangun sebuah aplikasi dengan menerapkan metode *haversine* untuk menemukan bengkel terdekat pada aplikasi layanan perbaikan kendaraan. Formula *Haversine* membutuhkan masukan *latitude* dan *longitude* lokasi pengguna dan lokasi bengkel sebagai masukan untuk menemukan bengkel terdekat. Penelitian ini didukung dengan metodologi pengembangan perangkat lunak yaitu metode *Extreme Programming* dalam melakukan analisis dan desain perancangan terhadap aplikasi yang dibangun. Hasil penelitian ini yaitu untuk membantu pengguna aplikasi yang ingin mencari informasi bengkel kendaraan terdekat dengan pencarian jarak terdekat berdasarkan lokasi pengguna saat ini ke lokasi bengkel.

Kata-kata Kunci: *Haversine*, Kendaraan, *Location Based Service*, Sistem Informasi Geografis

Abstract — *Vehicle is a tool that helps public transportation in the activities of daily life, but in addition to the benefits of using motorized vehicles, in general the community is also faced with the risk of using the vehicle that is if the*

engine suddenly broken or is called breaks down. Media information that can be applied in the case of a sudden damaged vehicle is to build an android application to provide information about vehicle workshops that are in the vicinity of damaged vehicle users using the Location Based Service. The purpose of this research is to build an application by applying the Haversine method to find the nearest workshop on the vehicle repair service application. Haversine formula requires user location and workshop location latitude and longitude as input to find the nearest workshop. This research is supported by a software development methodology, namely the Extreme Programming method in conducting design analysis and design of the applications that are built. The results of haversine testing in this application have quite good accuracy in determining the distance of the nearest workshop based on the comparison between the distance given by google direction and the calculation of the haversine formula. The results of this study are to help application users who want to find the nearest vehicle workshop information by searching for the closest distance based on the user's current location to the workshop location.

Keywords: *Haversine, extreme programming, Geographic Information System, , location based service.*

I. PENDAHULUAN

Alat transportasi merupakan kendaraan yang membantu masyarakat dalam melakukan aktivitas di kehidupan sehari-hari. Salah satu bentuk alat transportasi yang digunakan di lingkungan masyarakat adalah kendaraan bermotor. Kendaraan tersebut merupakan salah satu bentuk dari adanya kemajuan teknologi yang memberikan layanan kepada

penggunanya untuk dapat mengefisienkan waktu ketika melakukan perjalanan. Selain manfaat menggunakan kendaraan bermotor, pada umumnya masyarakat juga dihadapi dengan resiko penggunaan kendaraan tersebut yaitu jika terjadi kerusakan pada mesin secara tiba-tiba atau biasa disebut dengan istilah mogok. Kerusakan mesin pada kendaraan bermotor atau ‘mogok’ belum umum diketahui solusinya oleh setiap pengguna yang memakai kendaraan tersebut, oleh karena itu jasa montir sangat dibutuhkan untuk memperbaikinya. Kerusakan kendaraan bermotor dapat terjadi secara mendadak dimana dan kapan saja, hal tersebut membutuhkan solusi yaitu dengan menghubungi bengkel kendaraan bermotor terdekat baik mobil ataupun sepeda motor untuk segera diperbaiki dan meminta montir bengkel tersebut untuk datang ke lokasi tempat kendaraan mogok.

Media informasi yang dapat diterapkan pada kasus kendaraan rusak secara mendadak tersebut adalah dengan membangun sebuah aplikasi android untuk memberikan informasi mengenai bengkel kendaraan yang berada di sekitar lokasi kendaraan pengguna yang mengalami kerusakan menggunakan *Location Based Service*.

Location Based Service merupakan layanan yang mampu mendeteksi letak pengguna berada dengan memanfaatkan teknologi Global Positioning System yang terintegrasi dengan smartphone. Pengetahuan tentang posisi sangat penting bagi dunia bisnis, seperti memantau lintasan transportasi, logistic dan keperluan bisnis lainnya untuk mengetahui lokasi seseorang atau sesuatu [1]. Dengan kata lain teknologi *Location Based Service* dapat digunakan untuk menampilkan lokasi dari pelanggan yang sedang menggunakan aplikasi serta posisi bengkel dalam peta.

Proses perhitungan jarak terdekat antara lokasi pengguna dengan lokasi bengkel pada aplikasi ini diterapkan metode *haversine*, karena *haversine* biasa digunakan untuk menghitung jarak *Orthodomic*. Jarak *Orthodromic* adalah jarak terpendek antara dua titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*latitude*) sebagai variabel inputan Data [2]. *Haversine* sudah banyak digunakan untuk mempermudah pencarian lokasi terdekat, beberapa contohnya adalah untuk mencari perumahan dan fasilitas sosial terdekat [3], mencari rumah sakit dan puskesmas terdekat [4], serta untuk mendeteksi pengguna yang hadir pada sebuah acara berdasarkan lokasi pengguna [5], *Haversine* merupakan metode yang cocok digunakan untuk sistem rekomendasi karena proses *runtime* yang cepat [6]. Sistem informasi geografis pengukuran luas tanah

memanfaatkan metode *haversine formula* yang digunakan untuk mengetahui jarak antara 2 titik dengan meperhitungkan derajat kelengkungan bumi. [10].

Berdasarkan uraian tersebut, maka dapat diduga bahwa formula *haversine* dapat digunakan untuk mencari lokasi layanan perbaikan kendaraan terdekat dengan pengguna. Harapan dengan adanya sistem ini, pengguna aplikasi yang sedang mencari bengkel dapat menemukan bengkel terdekat dengan posisi saat ini dan mempersingkat proses perjalanan montir bengkel kepada pengguna aplikasi yang sedang mengalami mogok kendaraan.

II. METODE

A. Metode Penelitian

Beberapa tahapan – tahapan penyelesaian yang dilakukan dalam pembuatan aplikasi layanan perbaikan kendaraan dapat dilihat pada Gambar 1.



Gambar 1. Metode penelitian

B. Pengumpulan Data

Pengumpulan data dilakukan dengan cara melakukan pengamatan langsung ke bengkel - bengkel, data yang diperoleh dari hasil observasi adalah lokasi titik koordinat dan gambar bangunan bengkel yang akan dimasukkan ke dalam aplikasi. Titik koordinat digunakan untuk mengetahui posisi bengkel berada

yang akan ditampilkan pada peta dalam aplikasi, dan gambar bangunan bengkel digunakan untuk mempermudah pengguna untuk mencari bengkel yang sudah terdaftar pada aplikasi.

C. Analisa Formula Haversine

Teorema Haversine digunakan untuk menghitung panjang dua titik di permukaan bumi berdasarkan garis lintang dan garis bujur. Empat variabel harus disiapkan untuk menghitung dua jarak [7].

Formula *Haversine* membutuhkan *latitude* dan *longitude* lokasi pengguna dan sebuah lokasi sebagai masukan, cara mendapatkan titik koordinat dari lokasi pelanggan yaitu dengan cara memanfaatkan teknologi *Global Positioning System* yang sudah terintegrasi dengan *smartphone* dan cara mendapatkan titik koordinat dari posisi bengkel yaitu dengan cara mengambil data lokasi bengkel dari database. Data *latitude* dan *longitude* posisi pengguna dan posisi bengkel yang sudah diperoleh kemudian di masukkan kedalam formula *haversine* dan dihitung dengan rumus pada Gambar 2.

$$\begin{aligned} \Delta \text{lat} &= \text{lat}2 - \text{lat}1 \\ \Delta \text{long} &= \text{long}2 - \text{long}1 \\ a &= \sin^2(\Delta \text{lat}/2) + \cos(\text{lat}1) \cdot \cos(\text{lat}2) \cdot \sin^2(\Delta \text{long}/2) \\ c &= 2 \cdot \text{atan}2(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c \end{aligned}$$

Gambar 2. Rumus formula *haversine*

Keterangan :

R = jari-jari bumi sebesar 6371(km)

Δlat = besaran perubahan *latitude*

Δlong = besaran perubahan *longitude*

c = kalkulasi perpotongan sumbu

d = jarak (km)

Jarak yang didapatkan dari perhitungan *haversine* kemudian di sorting dari nilai yang paling kecil untuk mendapatkan daftar bengkel yang paling dekat dengan user.

D. Pengembangan Sistem

Extreme Programming (XP) merupakan suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat. Alasan menggunakan metode *Extreme Programming (XP)* karena sifat dari aplikasi yang di kembangkan dengan cepat melalui tahapan-tahapan yang ada meliputi : *Planning* (Perencanaan), *Design* (Perancangan), *Coding* (Pengkodean), dan *Testing* [8].

E. Pengujian Metode Haversine Pada Aplikasi

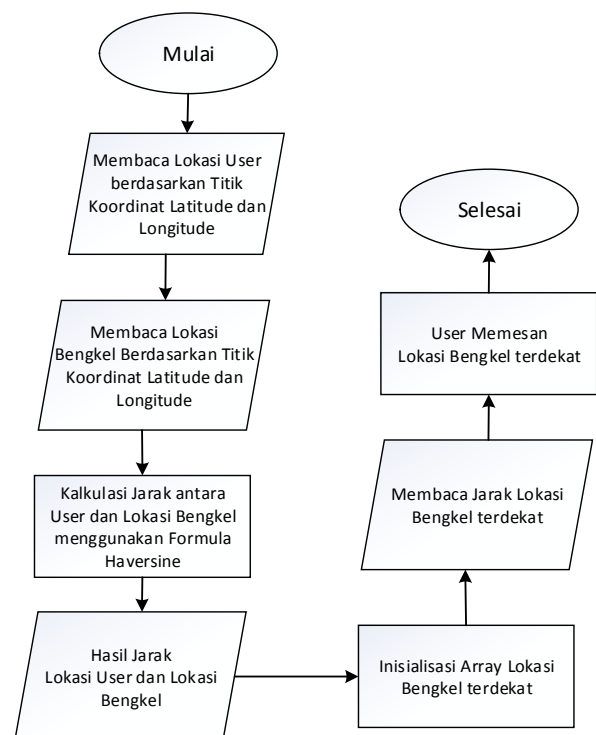
Pengujian metode *haversine* ini dilakukan dengan membandingkan titik lokasi terdekat hasil perhitungan sistem yang menggunakan formula *haversine* dengan titik lokasi terdekat hasil perhitungan dari *google direction*. Titik dari masing-masing perhitungan akan dibandingkan, apakah titik hasil perhitungan formula *haversine* yang menggunakan perhitungan garis lurus sama dengan hasil perhitungan dari *google direction* yang menggunakan perhitungan jarak jalan.

III. HASIL DAN PEMBAHASAN

A. Analisa Formula Haversine

Formula *haversine* membutuhkan titik koordinat lokasi bengkel dan pengguna untuk menemukan lokasi bengkel yang paling dekat dengan pengguna, teknologi *GPS* pada *Smartphone* pengguna dapat dimanfaatkan untuk mengetahui titik koordinat pengguna saat ini, sedangkan titik koordinat bengkel diambil dari data bengkel yang telah dikumpulkan dan disimpan pada *database*.

Sistem *flowchart* yang menggambarkan alur proses penerapan metode *haversine* pada aplikasi yang akan dibangun dapat dilihat pada Gambar 3.



Gambar 3. Flowchart algoritma *haversine* pada aplikasi

Berikut contoh analisis cara kerja Formula *haversine* dalam perhitungan jarak antara dua titik :

1) Tentukan titik koordinat lokasi bengkel dan pengguna. Titik Koordinat pertama adalah Universitas Siliwangi dengan *latitude* : -7.349952 dan *longitude* 108.222816

Titik Koordinat kedua adalah Bengkel Rama Putra dengan *latitude* : -7.344177 dan *longitude* 108.216672

2) Ubah nilai *latitude* dan *longitude* menjadi radian

$Latitude\ 1 = -7.349952 * \pi/180 = -0.1282808600$ radian

$Latitude\ 2 = -7.344177 * \pi/180 = -0.1281800673$ radian

$Longitude\ 1 = 108.222816 * \pi/180 = 1.8888444327$ radian

$Longitude\ 2 = 108.216672 * \pi/180 = 1.8887371997$ radian

3) Tentukan Δlat dan $\Delta long$

$$\Delta lat = Latitude\ 2 - Latitude\ 1$$

$$-0.1281800673 - (-0.1282808600) = 0.0001007927$$

$$\Delta long = Longitude\ 2 - Longitude\ 1$$

$$1.888737199732373 - 1.888844432759787 = -0.0001072330$$

4) Tentukan nilai a

$$\begin{aligned} & \sin^2(\Delta lat/2) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2(\Delta long/2) \\ & = \sin^2(0.0001007927/2) + \cos(-0.1282808600) * \\ & \cos(-0.1281800673) * \sin^2(-0.0001072330/2) \\ & = 0.00000000536751 \end{aligned}$$

5) Tentukan nilai perpotongan sumbu (c)

$$\begin{aligned} c &= 2 * a \sin(\sqrt{a}) \\ &= 2 * a \sin(\sqrt{0.00000000536751}) \\ &= 0.000146526584749709 \end{aligned}$$

6) Tentukan nilai jarak (d)

$$\begin{aligned} d &= R \cdot c \\ &= 6371(\text{km}) * 0.000146526584749709 \\ &= 0.933523517 \text{ km.} \end{aligned}$$

Hasil dari analisa diatas yaitu 0.933523517 km yang dihitung dari koordinat asal yaitu Universitas Siliwangi ke koordinat bengkel yaitu Bengkel Rama Motor. Hasil tersebut yang kemudian diurutkan dan kemudian dicari hasil yang memiliki nilai terkecil sebagai lokasi bengkel terdekat.

B. Pengembangan Sistem

1) *Planning* (Perencanaan)

a) *Analisis Kebutuhan Data*

Data yang dibutuhkan dalam membangun aplikasi ini dapat dilihat pada Tabel 1.

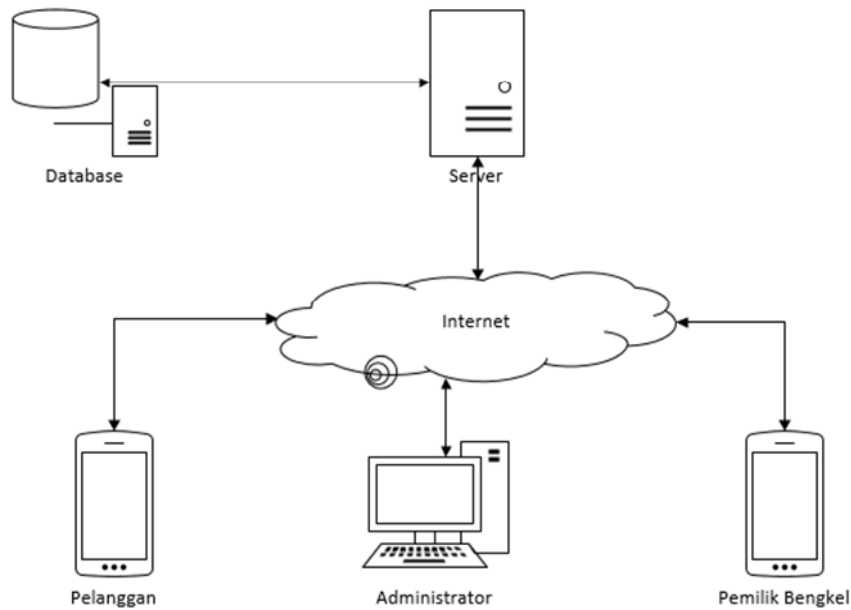
TABEL I
ANALISIS KEBUTUHAN DATA

Nama Data	Atribut
Data Pelanggan	Nama, <i>e-mail</i> , dan <i>password</i> .
Data Admin	Nama, <i>e-mail</i> , dan <i>password</i> .
Data Bengkel	Nama Bengkel, <i>e-mail</i> , <i>password</i> , alamat, hari awal, hari akhir, jam buka, jam tutup, jenis bengkel, layanan yang tersedia, nomor <i>whatsapp</i> , layanan jemputan, merek kendaraan yang diterima, proses montir, jumlah montir, <i>latitude</i> , dan <i>longitude</i> .
Data Keluhan	Subjek, Bengkel yang diminta, Merek Kendaraan, Deskripsi, Lokasi, waktu, nomor <i>whatsapp</i> bengkel, nomor <i>whatsapp</i> pengguna, <i>latitude</i> , <i>longitude</i> , dan status.
Data Laporan	Pelanggan yang melapor, bengkel yang dilaporkan, pemilik bengkel, alamat bengkel, waktu laporan, dan isi laporan.
Data Ulasan	Ulasan dan Nilai

b) *Perancangan Arsitektur Sistem*

Perancangan arsitektur sistem yang dibangun pada aplikasi pelayanan perbaikan kendaraan ini dapat dilihat pada Gambar 4, menjelaskan bahwa arsitektur sistem yang digunakan berbasis *client-server* dengan memanfaatkan *web service*

sebagai media penghubung antara sistem yang ada pada *server database*. Pelanggan, Pemilik Bengkel, dan Administrator harus terhubung ke internet untuk mengakses data yang disimpan di *server*.



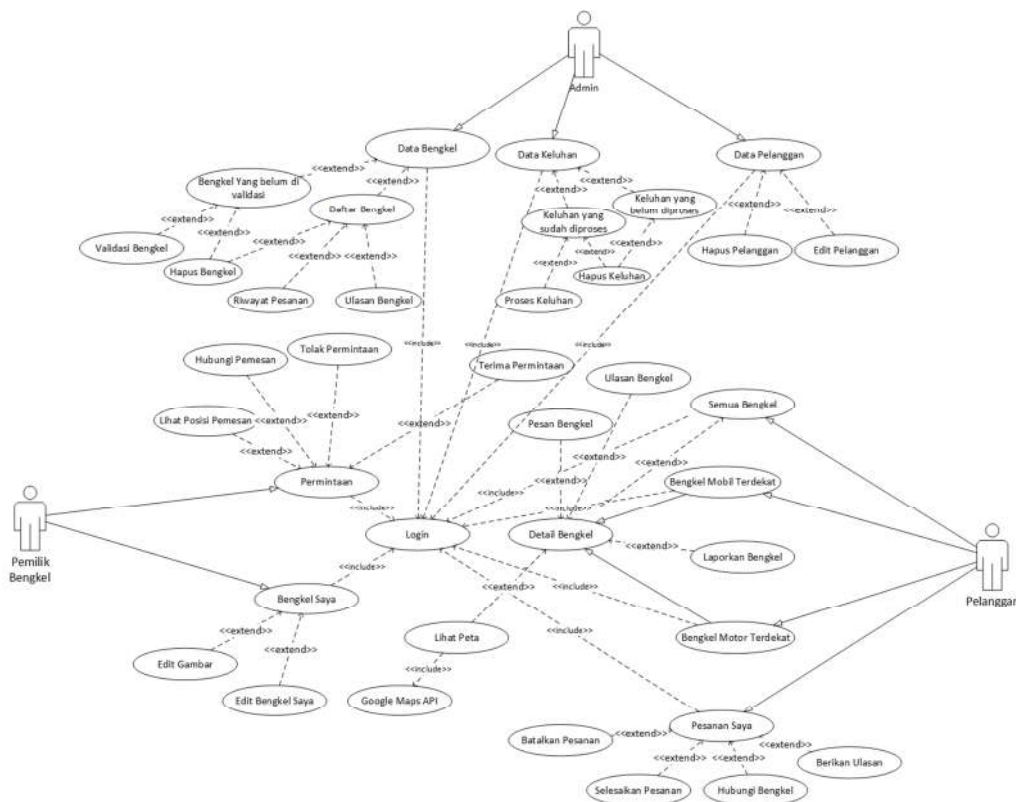
Gambar 4. Arsitektur sistem aplikasi

- 2) Design (Desain)
- a) Use Case Diagram

Desain Use Case hubungan antara pengguna dengan sistem disajikan pada Gambar 5.

Gambar 5 menunjukkan bagaimana hubungan antara pengguna dengan sistem. Pengguna

terlebih dahulu akan melakukan login yaitu apakah sebagai admin, Pelanggan, atau Pemilik Bengkel untuk menentukan halaman yang akan dimasuki.

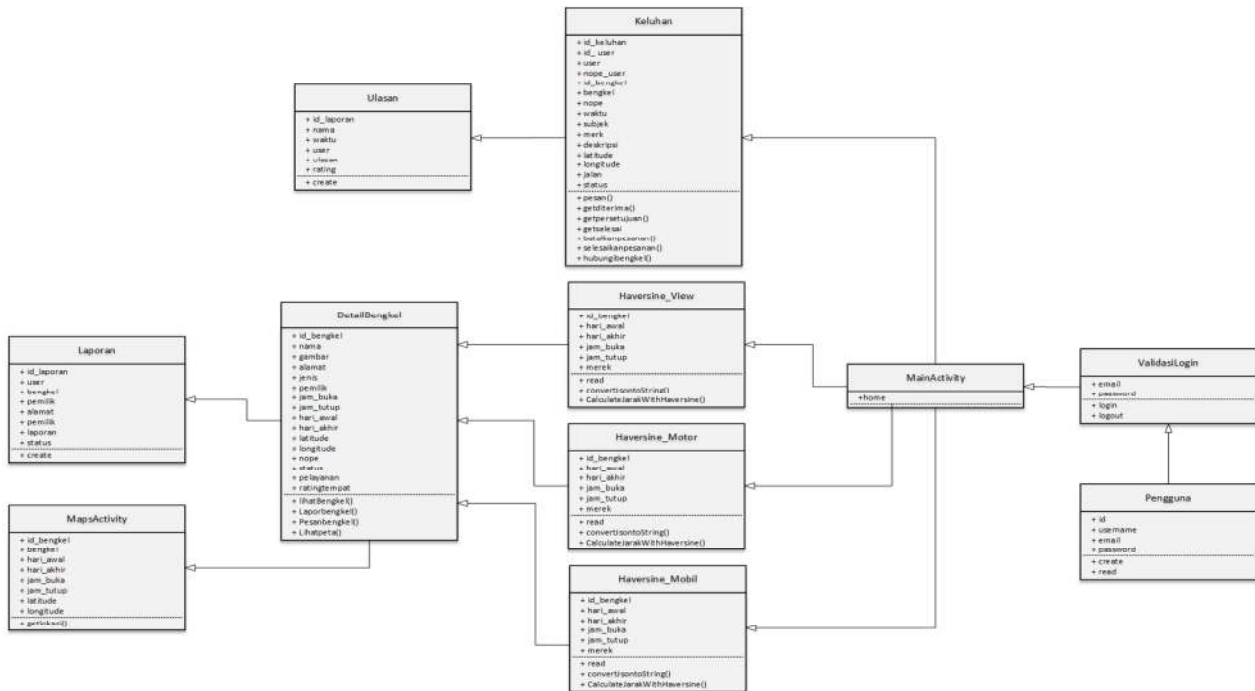


Gambar 5. Use case diagram

b) Class Diagram

Class Diagram digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem,

Desain Class Diagram dari sistem aplikasi android dari layanan perbaikan kendaraan disajikan pada Gambar 6.



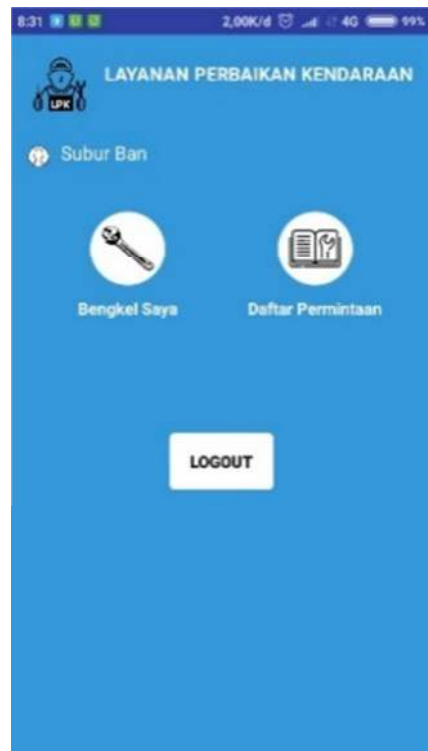
Gambar 6. Class diagram aplikasi android

3) Implementasi Antarmuka

Implementasi antar muka ini merupakan aplikasi yang telah selesai dikerjakan sesuai dengan kebutuhan data yang telah dianalisa dan dirancang sebelumnya, serta penerapan metode *haversine* dapat dijalankan untuk mencari bengkel terdekat dari posisi pelanggan.

a) Tampilan Utama Aplikasi Andoid Pemilik Bengkel

Halaman ini merupakan tampilan halaman utama untuk pemilik bengkel, pada halaman utama bengkel pemilik bengkel dapat masuk ke halaman bengkel saya atau daftar permintaan, tampilan menu utama untuk pemilik bengkel dapat dilihat pada Gambar 7.



Gambar 7. Tampilan halaman menu utama pemilik bengkel

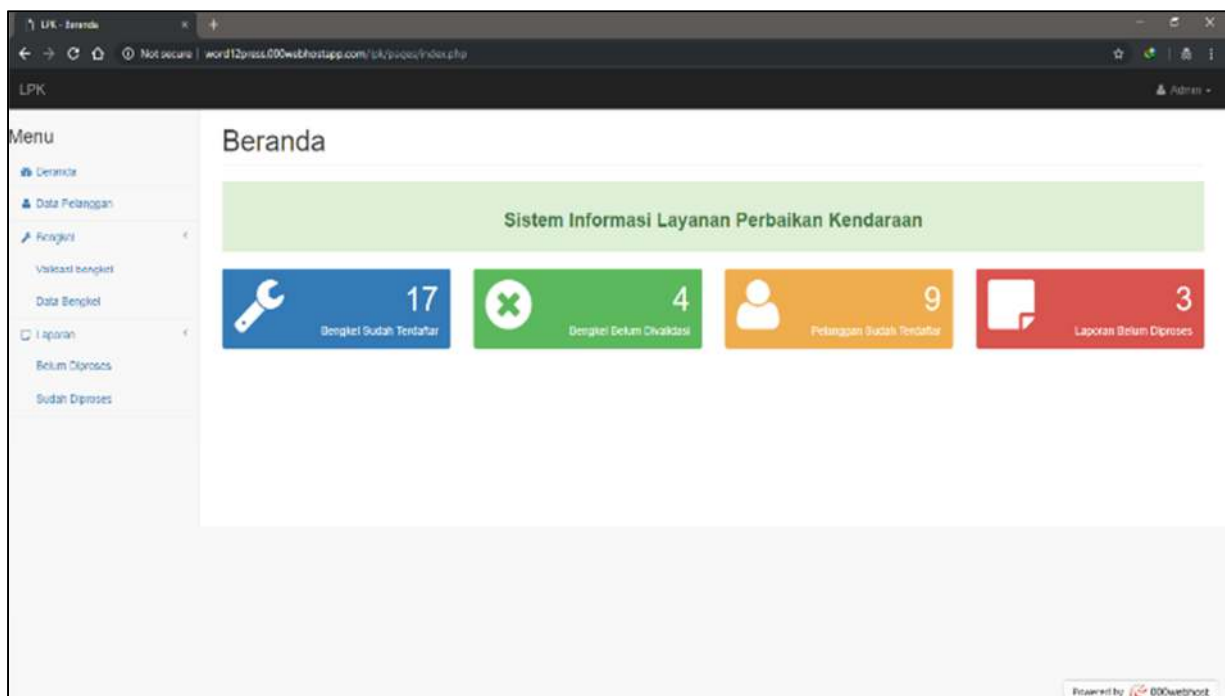
- b) Tampilan Utama Aplikasi Android Pelanggan
Merupakan tampilan halaman utama untuk pelanggan, pada halaman utama pelanggan dapat masuk ke halaman semua bengkel, bengkel mobil, bengkel motor atau pesanan saya, dan bisa memilih tombol *logout* untuk keluar sebagai pelanggan, tampilan menu pelanggan dapat dilihat pada Gambar 8.



Gambar 8. Tampilan halaman menu utama pelanggan

- c) Tampilan Aplikasi Web
Merupakan tampilan halaman utama Admin. Admin mempunyai hak akses untuk mengelola data pelanggan, bengkel, dan laporan. Pada halaman utama admin menampilkan data bengkel yang belum divalidasi, data bengkel yang sudah divalidasi, data pengguna yang sudah terdaftar, dan data laporan yang belum diproses, tampilan aplikasi web dapat dilihat pada Gambar 9.

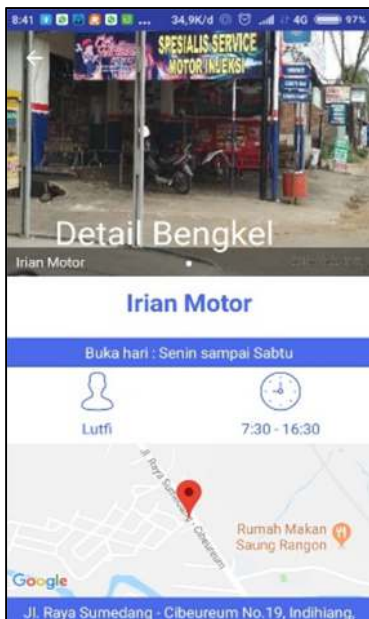
- d) Implementasi Metode *Haversine* pada Aplikasi
Hasil dan perhitungan jarak menggunakan formula *haversine* digunakan sebagai proses perhitungan untuk mencari bengkel terdekat dengan cara menampilkan hasil perhitungan titik koordinat pelanggan dengan lokasi bengkel yang ada pada *database* dan kemudian di *sorting* dari hasil yang terkecil sampai ke hasil yang terbesar, dan ditampilkan dalam *listview* pada halaman bengkel terdekat seperti dalam gambar 10 dan 11.



Gambar 9. Tampilan halaman menu utama aplikasi web



Gambar 10. Tampilan aplikasi hasil penelusuran bengkel terdekat di android



Gambar 11. Tampilan halaman detail bengkel di android

Gambar 10 merupakan tampilan halaman untuk melihat bengkel terdekat dari posisi pelanggan saat ini. Pada halaman ini berisi tentang tampilan daftar bengkel-bengkel yang dihasilkan dari penelusuran dengan metode *haversine*.

Gambar 11 merupakan tampilan untuk melihat detail informasi tentang bengkel yang dipilih, pada halaman detail bengkel pelanggan dapat melihat ulasan, melaporkan bengkel, melihat lokasi bengkel dalam bentuk peta *Google Maps*, dan melakukan pemesanan terhadap bengkel yang dipilih.

4) Pengujian Aplikasi

Hasil dari pengujian alpha menunjukkan bahwa semua tombol dan fungsi yang terdapat pada aplikasi layanan perbaikan kendaraan berfungsi sesuai dengan yang diharapkan. Pengujian beta dilakukan dengan melakukan penilaian pada aplikasi menggunakan kuesioner, Secara umum untuk penelitian jumlah sampel minimal untuk memperoleh hasil yang baik adalah 30 sampel [9].

Jawaban dari responden dapat dibuat skor tertinggi "satu" dan skor terendah "nol" untuk alternatif jawaban dalam kuesioner, dengan menetapkan kategori untuk setiap pernyataan positif, yaitu Ya = 1 dan Tidak = 0, sedangkan kategori untuk setiap pernyataan negatif, yaitu Ya = 0 dan Tidak = 1. Responden mengisi sebuah angket penilaian dengan pilihan jawaban "ya atau tidak" dari 14 pertanyaan. Dari analisis Skala Guttman, titik kesesuaian diatas 50% yaitu 93% sehingga dapat dikatakan pendapat responden mendekati sangat baik pada aplikasi layanan perbaikan kendaraan yang telah dibuat. Hasil dari Kuesioner dapat dilihat pada Tabel 2.

TABEL II
HASIL KUESIONER

No	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
2	1	1	1	1	1	1	0	1	0	0	1	1	0	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	0	1	0	0	1	1	1	1
5	1	1	1	1	1	1	0	1	0	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	0	0	0	1	0	1	1	0	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	0	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	0	1	1	0	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	0	0	1	1	1	1	0	1	0	0	1	1	1	1
24	0	0	1	1	1	1	1	1	0	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	0	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	0	1	1	1	1
28	1	1	1	1	1	1	0	1	0	1	1	1	1	1
29	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30	0	1	1	1	1	0	1	1	1	1	1	1	1	1

Keterangan : Nomor merupakan jumlah responden, sedangkan P1, P2, P3,... Pn merupakan urutan pertanyaan.

Dari responden jawaban “Ya” yang diperoleh dari kuisisioner, maka dihitung terlebih dahulu kemudian ditempatkan dalam rentang presentase sebagai berikut:

Nilai jawaban “Ya” : 1
Nilai jawaban “Tidak” : 0

Dikonversikan dalam presentase

Jawaban “ya” : $1 \times 100\% : 100\%$

Jawaban “tidak” : $0 \times 100\% : 0\%$

Perhitungan jawaban “ya” dari kuisisioner:

Jawaban “ya” rata-rata 28/30 responden $\times 100\% = 93\%$

Maka dengan demikian hasil uji aplikasi jika dilihat dari nilai aplikasi keseluruhan sebesar 93%, kemudian

hasil tersebut dimasukkan ke dalam tabel kriteria kelayakan. Hasil dari perhitungan menunjukkan bahwa kriteria kelayakan aplikasi penunjuk bengkel kendaraan terdekat sangat baik. Kriteria Presentase kelayakan dapat dilihat pada Tabel 3.

TABEL III
KRITERIA PRESENTASE KELAYAKAN

No	Presentase	Kriteria
1	0% - 20%	Sangat Kurang
2	21% - 40%	Kurang
3	41% - 60%	Cukup
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik

TABEL IV
PENGUJIAN METODE HAVERSINE PADA APLIKASI

Lokasi Pelanggan	Bengkel Terdekat	Jarak Haversine (Km)	Jarak Google Direction (Km)	Selisih
Masjid Al Rosyad Jl. Letnan Ibrahim Adji no 95, Indihiang	Irian Motor	0.83	1.0	0.17
	Steering Indo Jaya	0.85	1.1	0.25
	Planet Ban Rarangjami	1.05	1.3	0.25
	Netral Jaya Motor	1.98	2.2	0.22
	Indo Sarana Tasik	2.26	2.7	0.44
	Subur Ban	3.51	3.8	0.47
	Merdeka Motor	3.63	4.0	0.37

Dari pengujian yang telah dilakukan dengan posisi pelanggan berada di Masjid Al Rosyad Jl. Letnan Ibrahim Adji no 95 Indihiang, hasilnya dapat dilihat pada tabel 3 bahwa fitur pencarian bengkel terdekat pada aplikasi ini mempunyai akurasi yang cukup baik dalam menentukan jarak bengkel berdasarkan perbandingan antara jarak yang diberikan oleh *Google Direction* dengan perhitungan formula *Haversine*.

IV. PENUTUP

Kesimpulan pada penelitian yang telah dilakukan bahwa implementasi metode *haversine* memberikan solusi bagi pengguna yang ingin mencari informasi bengkel kendaraan terdekat dengan pencarian jarak terdekat berdasarkan lokasi pengguna saat ini ke lokasi bengkel. Dengan menggunakan formula *haversine* jarak yang diperoleh lebih dekat dibandingkan dengan jarak yang dihasilkan oleh *google direction* karena perhitungan formula *haversine* menggunakan perhitungan garis lurus, sedangkan perhitungan dari *google direction* menggunakan perhitungan jarak jalan berdasarkan posisi pengguna saat ini..

5) Pengujian Metode Haversine Pada Aplikasi

Pengujian metode dilakukan dengan cara menguji aplikasi untuk menampilkan bengkel terdekat dari posisi pelanggan ke lokasi bengkel dan akan dilakukan perbandingan antara hasil jarak dari perhitungan formula *haversine* dengan hasil jarak dari *Google Direction*. Hasil perbandingan antara jarak yang dihasilkan formula *haversine* dan *google direction* dapat dilihat pada Tabel 4.

DAFTAR PUSTAKA

- [1] Djafar, I., Marwan., 2016, "Pembangunan Aplikasi *Location Servise (LBS)* Kota Makassar" *SEMNAS TEKNOMEDIA* ISSN: 2302-3805, Volume 4 Nomor 1, Februari 2016, halaman 67-72.
- [2] Arfiani, I., 2016, "Pengembangan *Mobile Advertising* Sebagai Sarana Promosi Toko Roti Dan Kue Menggunakan *Location Based Service*" *Jurnal Informatika* ISSN:1978-0524, Volume 10 Nomor 2, Juli 2016, halaman 1262-1270.
- [3] Purmadipta, B., Anra, H., Irwansah, M, A., 2016, "Sistem Informasi Geografis Perumahan dan Fasilitas Sosial Terdekat dengan Metode *Haversine* Formula" *JUSTIN* ISSN: 2460-3562, Volume 4 Nomor 1, Maret 2016, halaman 104-109
- [4] Farid., Yunus, Y., 2017, "Analisa Algoritma *Haversine* Formula Untuk Pencarian Lokasi Terdekat Rumah Sakit dan Puskesmas Provinsi Gorontalo" *ILKOM* ISSN: 2087-1716, Volume 9 Nomor 3, Desember 2017, halaman 353-355.
- [5] Alam, C, N., Manaf, K., Atmadja, A, R., Aurum, D, K., 2016, "Implementation Of *Haversine* Formula For Counting Event Visitor In The Radius Based On Android Application" *4th International Conference on Cyber and IT Service Management*, ISBN: 978-1-4673-8443-8, April 2016.

- [6] Mahmoud, H., Akkari, N., 2016, "Shortest Path Calculation: A Comparative Study for Location-Based Recommender System" *World Symposium on Computer Applications & Research (WSCAR)*, ISBN: 978-1-5090-4114-5, Maret 2016.
- [7] Ganesh, L., Kumar, V, B, P., 2015 "Indoor Wireless Localization Using Haversine Formula" *International Advanced Research Journal in Science, Engineering and Technology (IARJSET)* ISSN: 2393-8021, Volume 2 Nomor 7, Juli 2015 , halaman 59–63.
- [8] Pressman, R, S., 2012, *Rekayasa Perangkat Lunak*, Andi, Yogyakarta.
- [9] Gay, L, R., Diehl, P, L., 1992, *Research Methods for Business and Management*. Macmillan Coll.
- [10] Putra, R, H, D., Sujaini, H., Safriadi, N., 2016 "Penerapan Metode Haversine Formula Pada Sistem Informasi Geografis Pengukuran Luas Tanah" *JUSTIN* ISSN: 2460-3562, Volume 4 Nomor 1, Maret 2016, halaman 157-162

