

NASKAH PUBLIKASI

**PENCARIAN LOKASI BENGKEL TRUK TERDEKAT
MENGUNAKAN FORMULA HAVERSINE**

Program Studi Teknik Informatika



Disusun oleh:

Fajar Raharjo

5150411011

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019**

NASKAH PUBLIKASI

PENCARIAN LOKASI BENGKEL TRUK TERDEKAT
MENGUNAKAN FORMULA HAVERSINE

Disusun oleh:
FAJAR RAHARJO
5150411011



Dr. Erik Iman Heri U., S.T., M.Kom.

Tanggal: 28/10/2019

PENCARIAN LOKASI BENGKEL TRUK TERDEKAT MENGUNAKAN FORMULA HAVERSINE

Fajar Raharjo¹, Erik Iman Heri Ujjianto²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi & Elektro

Universitas Teknologi Yogyakarta
Jl. Ringroad Utara Jombor Sleman Yogyakarta
Email: fjrraharjo4@gmail.com

ABSTRAK

Kabupaten Klaten merupakan salah satu daerah yang menghasilkan bahan bangunan yang sangat bagus seperti pasir, batu belah dan batu bata. Adanya potensi memberikan penghasilan, masyarakat di Kabupaten Klaten banyak yang membeli truk untuk mereka gunakan sebagai kendaraan pengangkut pasir atau batu belah yang biasa disebut dengan truk galian C. Setiap hari truk galian C tersebut melintas di jalanan Kabupaten Klaten. Untuk truk luar kota yang biasanya membawa tebu untuk disetorkan ke pabrik gula, sopir akan mencari muatan kembali untuk dibawa ke kota asal. Untuk muatan yang biasa dibawa truk luar kota adalah batu bata, namun ada yang mencari pasir untuk diangkut. Namun truk tidak selamanya akan tanpa masalah. Sering terjadi truk mengalami kerusakan mesin, radiator mengalami bocor, kerusakan dinamo, ban mengalami kebocoran dan lain-lain. Sopir sangat membutuhkan informasi bengkel terdekat lokasi truk saat mengalami kerusakan. Jika truk mengalami kerusakan yang jauh dengan perkotaan maka akan kesulitan untuk sopir membeli spare part atau memanggil montir untuk memperbaiki truk. Oleh karena itu melalui perancangan dan pembuatan sebuah aplikasi untuk pencarian bengkel truk terdekat saat terjadi masalah dengan truk yang dikendarai. Aplikasi ini diharapkan dapat menampilkan informasi bengkel yang terdekat dari titik lokasi sopir dengan implementasi Formula Haversine di dalamnya. Sistem pencarian bengkel tersebut dapat memberikan kemudahan kepada pengguna khususnya sopir.

Kata kunci : Kendaraan, Formula Haversine, Bengkel

1. PENDAHULUAN

1.1 Latar Belakang

Truk sebagai alat transportasi yang digunakan untuk mengangkut barang seperti pasir, batu belah, batu bata, abu batu, tebu dan *splite* khususnya yang melewati jalanan di Kabupaten Klaten sangat banyak. Untuk truk yang mengangkut pasir dan batu belah atau yang sering disebut truk golongan C setiap hari terus beroperasi untuk memenuhi permintaan pelanggan dari Kabupaten Klaten ataupun Kabupaten yang lainnya.

Pengoperasian truk secara terus menerus tanpa adanya pengecekan mesin secara berkala dapat menimbulkan masalah kerusakan mesin yang bisa terjadi saat truk tersebut sedang beroperasi. Sopir tidak akan tahu kapan dan di mana masalah kerusakan misalnya kerusakan mesin, radiator mengalami kebocoran, kerusakan dinamo, ban mengalami kebocoran dan lain-lain akan terjadi. Jika truk mengalami kerusakan di jalan dan didaerah yang mungkin belum diketahui oleh sopir, maka akan kesulitan untuk mencari bengkel

terdekat. Jika pun sopir dapat memperbaiki sendiri, besar kemungkinan ada *spare part* yang perlu diganti.

Banyaknya potensi kerusakan pada truk saat beroperasi dan sopir yang belum mengetahui bengkel terdekat, maka Aplikasi pencarian bengkel terdekat ini tentu akan sangat membantu sopir yang truknya mengalami kerusakan di daerah Kabupaten Klaten. Berdasarkan latar belakang di atas, penulis membuat penelitian dengan judul “Pencarian Lokasi Bengkel Truk Terdekat Menggunakan Formula Haversine”.

1.2 Batasan Masalah

Dalam penelitian pembuatan Aplikasi pencarian lokasi bengkel truk terdekat yang akan dibahas dalam ruang lingkup pembahasannya, sebagai berikut:

- a. Aplikasi tentang informasi lokasi bengkel terdekat di Kabupaten Klaten.
- b. Aplikasi ini memanfaatkan Maps API dari Google yang akan digunakan untuk menampilkan titik lokasi bengkel.
- c. Aplikasi ini hanya menyediakan beberapa informasi bengkel seperti nomor telepon, alamat lengkap, jam buka tutup dan foto.

1.3 Tujuan penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini adalah:

- a. Dapat menghasilkan aplikasi untuk mengetahui lokasi penyebaran bengkel yang berada di Kabupaten Klaten.
- b. Aplikasi ini dibuat untuk mengetahui lokasi bengkel terdekat.

2. KAJIAN PUSTAKA DAN TEORI

2.1 Landasan Teori

[13] Penelitian terdahulu membahas tentang bagaimana Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal. Penelitian tersebut bertujuan agar pengguna sistem dapat melakukan perhitungan seberapa jarak kita terhadap suatu objek, kemudian dengan *Direction Service* dari Google, pengguna sistem juga dimungkinkan untuk mengetahui jalur yang bisa tempuh untuk mencapai lokasi tersebut.

[4] Penelitian terdahulu membahas tentang bagaimana Analisa Algoritma Haversine Formula Untuk Pencarian Lokasi Terdekat Rumah Sakit dan Puskesmas Provinsi Gorontalo. Penelitian tersebut

bertujuan untuk mencari lokasi terdekat rumah sakit dan puskesmas yang ada di Provinsi Gorontalo dengan menggunakan Sistem Informasi Geografis (GIS). Di samping itu juga masyarakat dapat mengetahui jarak rumah sakit dan puskesmas terdekat dengan posisi *user*.

[5] Penelitian terdahulu membahas tentang bagaimana Penerapan Metode Haversine Formula Pada Aplikasi Pencarian Lokasi Tempat Tambah Ban Kendaraan Bermotor Berbasis Mobile Android. Penelitian tersebut membahas tentang bagaimana masyarakat dapat mengetahui tempat tambah ban terdekat dengan lokasi terkini. Adanya perancangan sistem ini maka diharapkan masyarakat dapat lebih *up to date* tentang informasi dan lokasi yang dekat dengan jarak posisi pengendaraan, sedangkan platform yang digunakan berbasis sistem operasi Android

2.2 Unified Modeling Language (UML)

[10] UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2.3 Use Case Diagram

[7] Diagram *Use Case* adalah diagram yang bersifat status yang memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini memiliki 2 fungsi, yaitu mendefinisikan fitur apa yang harus disediakan oleh sistem dan menyatakan sifat sistem dari sudut pandang *user*.

2.4 Activity Diagram

[7] *Activity* diagram merupakan diagram yang bersifat dinamis. *Activity* diagram adalah tipe khusus dari diagram *state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem dan berfungsi untuk menganalisis proses.

2.5 Sequence Diagram

[12] *Sequence* diagram dibuat berdasarkan *activity* diagram dan *class* diagram. *Sequence* diagram menggambarkan aliran pesan yang terjadi antar kelas yang dideskripsikan pada *class* diagram dengan menggunakan operasi yang dimiliki kelas tersebut. Untuk aliran pesan, *sequence* diagram

merujuk pada alur sistem *activity* diagram yang telah dibuat sebelumnya.

2.6 Class Diagram

[10] Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

2.7 Software Development Kit (SDK)

[9] Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman java. Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan, namun bukan merupakan aplikasi bawaan *Handphone/Smartphone*.

2.8 Android

[9] Android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Android umum digunakan di *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS.

2.9 Formula Haversine

[3] Formula Haversine adalah persamaan penting dalam sistem navigasi, nantinya Formula Haversine akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (*longitude*) dan garis lintang (*latitude*).

Formula Haversine adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Formula Haversine merupakan suatu metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan.

2.10 Database

[11] *Database* sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan di antaranya.

[6] *Database* adalah sekumpulan data *store* (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya.

2.11 Global Positioning System (GPS)

[1] GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit.

2.12 API Google Maps

[14] Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambah fitur Google Maps dalam web yang telah kita buat atau pada blog yang berbayar maupun gratis sekalipun dengan Google Maps API. Google Maps API adalah suatu *library* yang berbentuk JavaScript.

2.13 Latitude dan Longitude

Latitude adalah garis yang melintang di antara kutub utara dan kutub selatan, yang menghubungkan antara sisi timur dan barat bagian bumi. Garis ini memiliki posisi membentang bumi, sama halnya seperti garis *equator* (khatulistiwa), tetapi dengan kondisi nilai tertentu. Garis lintang inilah yang dijadikan ukuran dalam mengukur sisi utara-selatan koordinat suatu titik di belahan bumi.

Latitude dibedakan menjadi 2 wilayah, yaitu wilayah utara yang bisa kita sebut lintang utara dan selatan atau yang bisa disebut lintang selatan, di mana nilai koordinat di bagian utara selalu positif dan nilai koordinat di bagian selatan adalah negatif.

3. METODE PENELITIAN

3.1 Analisis

- a. Analisis Kebutuhan Perangkat Lunak
Kebutuhan perangkat lunak terdiri dari spesifikasi minimum yang dibutuhkan untuk membangun aplikasi. Berikut adalah spesifikasi minimum perangkat lunak yang digunakan dalam membangun aplikasi pencarian lokasi bengkel truk terdekat ini, yaitu:

Sistem Operasi Windows 7 32/64 bit.

Aplikasi Android Studi.

Aplikasi Visual Studio Code.

JDK versi 9.0 dan Android SDK Windows.

Xampp.

- b. Analisis Kebutuhan Perangkat Keras.

Kebutuhan perangkat keras yang diperlukan untuk membangun aplikasi yang akan dibangun berupa komputer dengan minimal spesifikasi yang dapat digunakan. Berikut spesifikasi kebutuhan perangkat keras untuk membangun aplikasi pencarian lokasi bengkel truk terdekat dengan baik, yaitu:

Prosesor : Intel Core i5 M 430 @2.27Ghz (4 CPUs). Monitor : 14 Inch

VGA : Intel ® HD Graphics Family

Harddisk : 500 Gb.

RAM : 4096 Mb.

- c. Analisis Kebutuhan Pengguna

Kebutuhan pengguna ini menjelaskan tentang spesifikasi minimum perangkat *mobile* untuk menggunakan aplikasi pencarian lokasi bengkel truk terdekat. Berikut spesifikasi kebutuhan pengguna, yaitu:

Prosesor : Quad Core 1.1 Ghz.

Versi : Android 5.0 Lollipop

Internal : 8 Gb

RAM : 1024 Mb

3.2 Desain

- a. Perancangan Sistem

Aplikasi Pencarian Lokasi Bengkel Truk Terdekat di Kabupaten Klaten menggunakan pemodelan *Unified Modeling Language* (UML) yang terdiri atas *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

- b. Perancangan Basis Data

Tabel-tabel yang akan digunakan pada basis data ini adalah:

1. Web *User*
2. Kategori Web *User*
3. Android *User*
4. Profil Aplikasi
5. Bengkel
6. Kategori Bengkel
7. Favorit

8. Menu Bantuan

9. Sub Menu Bantuan

- c. Perancangan *Desain Interface*

a. Input

Proses input meliputi *input* data lokasi, data *login user*, *register* dan nama bengkel.

b. Proses

Proses dalam aplikasi meliputi pendeteksian lokasi *user*, penentuan bengkel truk terdekat dari lokasi *user* dan penentuan bengkel truk terdekat berdasarkan kategori yang telah dipilih oleh *user*.

c. Output

Data yang diperoleh dari aplikasi adalah informasi tentang bengkel truk yang terdapat di Kabupaten Klaten dan sistem yang dapat memberikan keputusan atau opsi penentuan bengkel terdekat yang dapat dituju dengan beberapa kategori.

3.3 Implementasi

Berdasarkan perancangan yang telah dibuat, dilakukan implementasi dengan sistem keseluruhan. Aplikasi ini diimplementasikan langsung pada perangkat android dari berbagai versi.

3.4 Pengujian Sistem

Pada tahap ini pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Pengujian dilakukan dengan *black box testing*, yaitu pengujian atau *testing* dengan cara melihat alur kinerja dan *output* program yang dihasilkan. Pengujian ini untuk mengetahui fungsi-fungsi yang diharapkan seperti output dihasilkan secara benar dari input, dan *database* diakses serta *diupdate* secara benar dan mengujinya apakah akan menjalankan fungsi-fungsi tersebut secara tepat. Uji coba dilakukan dengan mencoba fungsi menu apakah sudah sesuai harapan yaitu menampilkan informasi yang ada pada menu, *text* dan gerakan sesuai yang diharapkan.

3.5 Pemeliharaan Sistem

Pemeliharaan sistem merupakan tahap terakhir dalam metode *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam

memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Selain itu, perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

4. ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem Yang Sedang Berjalan

Aplikasi pencarian lokasi bengkel truk terdekat di Kabupaten Klaten adalah sebuah sistem yang dibangun untuk melakukan pencarian lokasi bengkel truk terdekat secara terkomputerisasi. Dengan adanya aplikasi pencarian lokasi bengkel truk terdekat, maka seluruh data yang berhubungan dengan lokasi bengkel truk dapat disimpan dan diintegrasikan dalam proses manajemen data sehingga dapat dikontrol dengan baik dan dapat digunakan untuk pengambilan keputusan untuk menentukan lokasi bengkel truk terdekat.

Truk-truk yang melewati jalanan di Kabupaten Klaten banyak yang merupakan truk angkutan antar provinsi bahkan antar pulau. Selama ini sopir truk yang bukan berdomisili di Kabupaten Klaten, untuk mengetahui informasi keberadaan bengkel dengan cara melihat melalui aplikasi Google Maps dan tidak semua bengkel truk ada di Google Maps. Jadi jika truk mengalami kerusakan ringan atau berat dan lokasinya jauh dari pusat kota akan sangat kesulitan untuk mencari bengkel yang terdekat.

Implementasi aplikasi pencarian lokasi bengkel terdekat di Kabupaten Klaten ini diharapkan menghasilkan aplikasi yang memiliki kemampuan untuk memenuhi kebutuhan berupa:

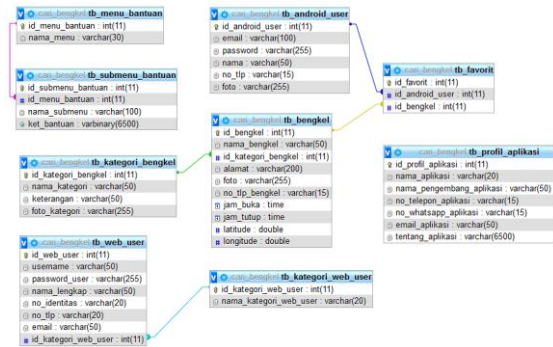
- Menentukan bengkel truk terdekat yang dapat dituju oleh sopir truk berdasarkan lokasinya.
- Menentukan bengkel truk yang dapat dituju oleh pengendara dengan filter kebutuhan akan kerusakan truk.

Pada proses analisis dari kebutuhan dan kemampuan sistem yang dirancang akan dimodelkan dengan *Unified Modelling Language* (UML).

4.2 Rancangan Sistem

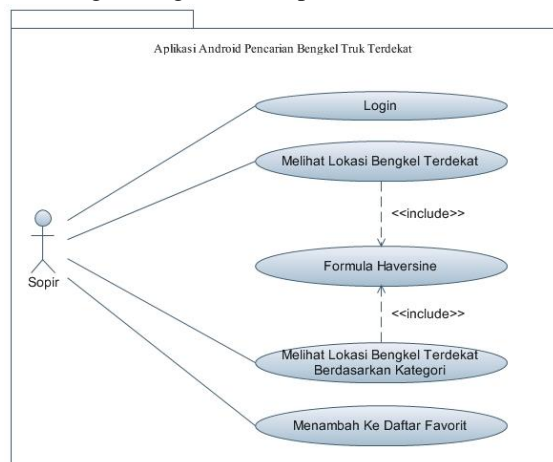
Aplikasi Pencarian Lokasi Bengkel Truk Terdekat di Kabupaten Klaten menggunakan pemodelan *Unified Modeling Language* (UML) yang terdiri atas *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*. Diagram relasi antar tabel menggambarkan primary key dan foreign key yang terdapat pada setiap tabel

dan kemudian digunakan untuk merelasikan antar tabel yang memiliki kaitan pada input dan proses. Diagram relasi antar tabel digambarkan pada Gambar 1 seperti dibawah ini:



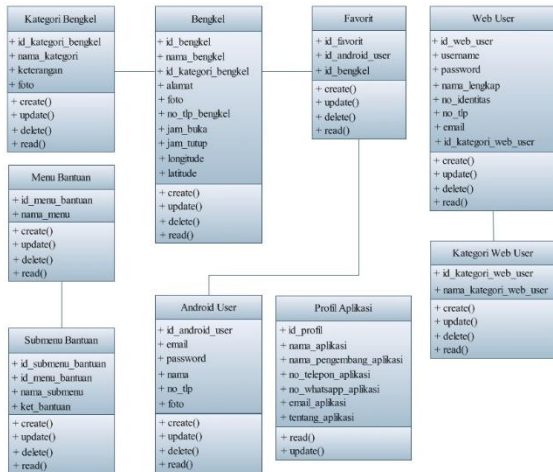
Gambar 1 Relasi Tabel

- Use Case Diagram* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Gambar di bawah ini menjelaskan interaksi antara *use case* dengan aktor terhadap sistem digambarkan menggunakan *use case diagram*. *Use Case Diagram* digambarkan pada Gambar 2.

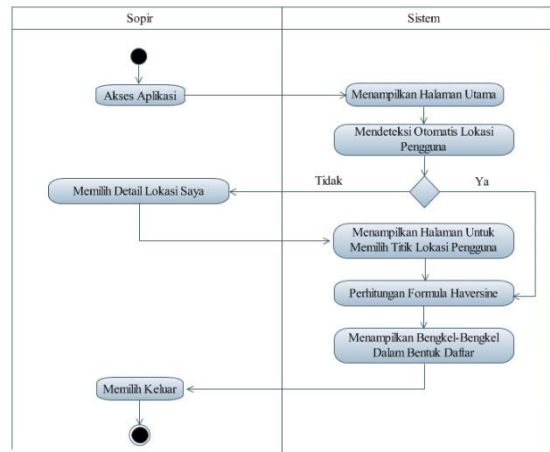


Gambar 2 Use Case Diagram

- Class Diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. *Class diagram* yang dijelaskan pada analisis ini adalah *class diagram* yang terpasang pada sistem pencarian lokasi bengkel terdekat. *Class diagram* dari sistem ini seperti terlihat pada Gambar 3.

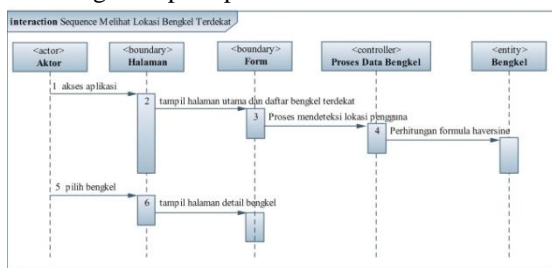


Gambar 3 Class Diagram



Gambar 5 Activity Diagram Melihat Lokasi Bengkel Terdekat

c. *Sequence* diagram digunakan untuk mengetahui tentang alur proses dan interaksi antara objek yang pada aplikasi pencarian lokasi bengkel truk terdekat. *Sequence* diagram dapat menampilkan bagaimana sistem merespons setiap kejadian atau permintaan dari *user*, dapat mempertahankan integritas internal, bagaimana data dipindah ke *user interface* dan bagaimana objek-objek diciptakan dan dimanipulasi. Adapun proses-proses yang terdapat pada saat aplikasi menampilkan daftar bengkel seperti pada Gambar 4.

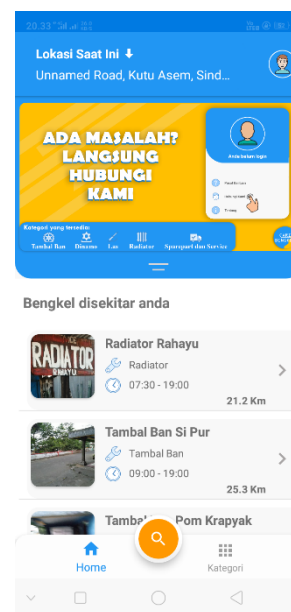


Gambar 4 Sequence Diagram Melihat Lokasi Bengkel Terdekat

d. *Activity* Diagram merupakan alur kerja pada setiap *usecase*. *Activity* diagram pada analisa ini mencakup *activity* diagram setiap *usecase*. Untuk memudahkan dalam perancangan *activity* diagram maka aplikasi pencarian lokasi bengkel truk terdekat ini akan dipecah menjadi beberapa bagian. Berikut ini adalah gambar *activity* diagram dari proses aplikasi yang terjadi. Proses-proses yang terjadi pada saat aplikasi menampilkan daftar bengkel dapat dilihat pada Gambar 5.

5. IMPLEMENTASI SISTEM

Implementasi sistem merupakan tahapan lanjutan dari tahapan perancangan sistem. Implementasi sistem bertujuan untuk menterjemahkan keperluan perangkat lunak kedalam bentuk sebenarnya yang dimengerti oleh komputer. Dalam tahap implementasi ini, akan dijelaskan mengenai perangkat keras (Hardware) dan perangkat lunak (Software) yang digunakan dalam membangun sistem, flowchart dan data yang digunakan untuk pengolahan dari hasil pemrograman perancangan tampilan interface (antarmuka).



Gambar 6 Halaman Home

Gambar 6 menunjukkan halaman utama aplikasi yang merupakan halaman yang pertama kali muncul setelah *splash screen* saat pengguna

mengakses aplikasi ini. Aplikasi akan otomatis mencari bengkel semua kategori yang ditampilkan dalam bentuk daftar dan urut dari bengkel yang paling dekat sampai ke bengkel yang paling jauh



Gambar 7 Halaman Detail Bengkel

Gambar 7 menunjukkan tampilan halaman untuk detail bengkel. Halaman detail bengkel berisi tentang informasi detail bengkel seperti titik bengkel, jam buka dan tutup dan kategori bengkel. Pada halaman ini fitur favorit berada. Halaman ini juga menampilkan tombol untuk menghubungi pihak bengkel melalui telepon atau whatsapp.

6. PENUTUP

6.1 Kesimpulan

Berdasarkan implementasi dan pembahasan aplikasi yang telah dibangun maka dapat disimpulkan sebagai berikut.

- Aplikasi ini dapat memberikan kemudahan bagi pengguna khususnya sopir untuk mencari lokasi bengkel yang terdekat.
- Aplikasi dapat memberikan informasi bengkel yang sangat dibutuhkan oleh pengguna seperti nomor telepon, jam buka dan tutup dan titik lokasi bengkel tersebut berada.
- Aplikasi ini dapat memberikan kemudahan bagi pengguna khususnya sopir untuk mencari lokasi bengkel terdekat menurut kategori atau kebutuhan akan kerusakannya.

6.2 Saran

Penulis sadar dalam pembangunan aplikasi ini masih terdapat beberapa kekurangan, maka penulis berharap bagi penulis selanjutnya dapat mempertimbangkan saran-saran berikut ini.

- Menambahkan fitur rute dari lokasi pengguna ke bengkel yang akan di tampilkan di halaman detail.
- Menambahkan aktor yaitu pemilik bengkel agar pemilik bengkel dapat menambahkan bengkelnya langsung lewat aplikasi.
- Meningkatkan keamanan aplikasi untuk menjaga keamanan data dari pihak yang tidak bertanggung jawab.

UCAPAN PERSEMBAHAN

Naskah Publikasi ini dapat diselesaikan tidak lepas dari segala bantuan, bimbingan, dorongan dan doa dari berbagai pihak, yang pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

- Bapak Dr. Bambang Moertono Setiawan, M.M., CA., Akt. Selaku Rektor Universitas Teknologi Yogyakarta.
- Bapak Sutarman, S.Kom., M.Kom., Ph.D. Selaku Dekan Fakultas Teknologi Informasi dan Elektro Universitas Teknologi Yogyakarta
- Ibu Dr. Enny Itje Sela, S.Si., M.Kom selaku Ketua program studi Teknik Informatika Universitas Teknologi Yogyakarta
- Bapak Dr. Erik Iman Heri U., S.T., M.Kom selaku dosen pembimbing
- Kepada kedua orang tua yang telah memberikan do'a serta dukungannya.

DAFTAR PUSTAKA

- [1]Alfeno, S. dan Devi, R.E.C. (2017), *Implementasi Global Positioning System (GPS) dan Location Based Service (LBS) Pada Sistem Informasi Kereta Api Untuk Wilayah Jabodetabek*, *Jurnal Sisfotek Global*, 7(2).
- [2]Amin, Zaenal dan Santoso (2012), *Pemodelan Sistem Informasi Persediaan Barang Pada PT. Nutech Pundi Arta*, Jakarta: Universitas Budi Luhur.
- [3]Chopde, P.N.. dan Nichat, M.. (2013), *Landmark Based Shortest Path Detection by Using A* and Haversine Formula*, *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2), 5.

- [4] Farid dan Yunus, Y. (2017), *Analisa Algoritma Haversine Formula Untuk Pencarian Lokasi Terdekat Rumah Sakit dan Puskesmas Provinsi Gorontalo*, *ILKOM Jurnal Ilmiah*, 9(3), 353–355.
- [5] Fauzi, A., Pernando, F. dan Raharjo, M. (2018), *Penerapan Metode Haversine Formula Pada Aplikasi Pencarian Lokasi Tempat Tambal Ban Kendaraan Bermotor Berbasis Mobile Android*, *Jurnal Teknik Komputer AMIK BSI*, 4(2), 56–63.
- [6] Ladjamudin, A. Bin (2013), *Analisis dan Desain Sistem Informasi*, Yogyakarta: Graha Ilmu.
- [7] Murad, Kusniawati, D.F., Asyanto, N. dan Agus (2013), *Aplikasi Intelligence Website Untuk Penunjang Laporan PAUD Pada Himpaudi Kota Tangerang*, *Jurnal CCIT*, 7(1).
- [8] Pressman, R.S. (2010), *Software Engineering : a practitioner's approach*, New York: McGraw-Hill.
- [9] Safaat, N. (2012), *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Bandung: Informatika.
- [10] Sukamto, R.A. dan Shalahuddin, M. (2013), *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Bandung: Informatika.
- [11] Sutarman (2012), *Buku Pengantar Teknologi Informasi*, Jakarta: Bumi Aksara 2.
- [12] Vidia, Danada dan Dkk (2013), *Analisis Dan Perancangan Sistem Informasi Rawat Jalan Di Rumah Sakit Hewan Universitas Airlangga Surabaya Dengan Metode Berorientasi Objek*, Surabaya: Universitas Airlangga.
- [13] Yulianto, Ramadiani dan Kridalaksana, A.H. (2018), *Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal*, *Jurnal Ilmiah Ilmu Komputer*, 13(1), 14–21.
- [14] Yusro (2013), *Pengertian Google Maps API*, (<http://www.myusro.info/2013/02/pengertian-google-maps-api.html>) akses 30 Maret 2019.