

EXPLORE

Jurnal Sistem Informasi & Telematika (Telekomunikasi, Multimedia & Informatika)

Robby Yuli Endra, Ahmad Cucus, Freddy Nur Affandi, M. Bintang Syahputra
DETEKSI OBJEK MENGGUNAKAN HISTOGRAM OF ORIENTED GRADIENT (HOG) UNTUK MODEL SMART ROOM

Halimah, Bobby Bachry
PEMANFAATAN MODEL ENTERPRISE ARCHITECTURE PLANNING (EAP) UNTUK PROTOTYPE E-DOCUMENT KEPEGAWAIAN (DOSEN) PADA BAGIAN SUMBER DAYA MANUSIA DI INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA

Darsin
PENERAPAN FUZZY INFERENCE SYSTEM (FIS) METODE SUGENO UNTUK MENENTUKAN KANDIDAT DOSEN TERBAIK DI UNIVERSITAS MEGOW PAK TULANG BAWANG

Ida Ayu Putu Anggie Sinthiya, Danang Kusnadi
ANALISIS EMISI GAS RUMAH KACA (GRK) DAN PEMETAAN ZONA EMISI MENGGUNAKAN GIS (GEOSPASIAL INFORMATION SYSTEM) DI KABUPATEN PRINGSEWU, LAMPUNG

Budi Usmanto, Bernadhita H.S.U
PROTOTYPE SISTEM PENDETEKSI DAN PERINGATAN DINI BENCANA ALAM DI INDONESIA BERBASIS INTERNET OF THINGS (IoT)

Oktafianto, Ponidi
SISTEM KEAMANAN GEDUNG BERBASIS SMS GATEWAY DAN MEDIA SOSIAL DENGAN MIKROKONTROLLER ATMEGA328

Pamuji Setiawan, Elisabet Yunaeti Anggraeni
PURWARUPA SISTEM PENGAIRAN SAWAH OTOMATIS DENGAN ARDUINO BERBASIS ARTIFICIAL INTELEGENT

Erlangga, Yanuarius Yanu Dharmawan
PENENTUAN PENERIMA KINERJA DOSEN AWARD MELALUI METODE TSUKAMOTO DENGAN KONSEP LOGIKA FUZZY

Yuthsi Aprilinda, Emy Sugandasari, Freddy Nur Afandi, Fenty Ariani
AUTOMATIC COUNTING MENGGUNAKAN METODE HAVERSINE UNTUK MENGHITUNG JUMLAH PENUMPANG BUS

Taqwan Thamrin, Erlangga, Wiwin Susanty
IMPLEMENTASI RUMAH LISTRIK BERBASIS SOLAR CELL



Jurnal Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia, dan Informasi)
Volume 9, Nomor 2, Oktober 2018

NO	JUDUL PENELITIAN / NAMA PENULIS	HALAMAN
1.	DETEKSI OBJEK MENGGUNAKAN HISTOGRAM OF ORIENTED GRADIENT (HOG) UNTUK MODEL SMART ROOM Robby Yuli Endra, Ahmad Cucus, Freddy Nur Affandi, M. Bintang Syahputra	99-105
2.	PEMANFAATAN MODEL ENTERPRISE ARCHITECTURE PLANNING (EAP) UNTUK PROTOTYPE E-DOCUMENT KEPEGAWAIAN (DOSEN) PADA BAGIAN SUMBER DAYA MANUSIA DI INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA Halimah, Bobby Bachry	106-113
3	PENERAPAN FUZZY INFERENCE SYSTEM (FIS) METODE SUGENO UNTUK MENENTUKAN KANDIDAT DOSEN TERBAIK DI UNIVERSITAS MEGOW PAK TULANG BAWANG Darsin	114-120
4	ANALISIS EMISI GAS RUMAH KACA (GRK) DAN PEMETAAN ZONA EMISI MENGGUNAKAN GIS (GEOSPASIAL INFORMATION SYSTEM) DI KABUPATEN PRINGSEWU, LAMPUNG Ida Ayu Putu Anggie Sinthiya, Danang Kusnadi	121-126
5	PROTOTYPE SISTEM PENDETEKSI DAN PERINGATAN DINI BENCANA ALAM DI INDONESIA BERBASIS INTERNET OF THINGS (IoT) Budi Usmanto, Bernadhita H.S.U	127-136
6	SISTEM KEAMANAN GEDUNG BERBASIS SMS GATEWAY DAN MEDIA SOSIAL DENGAN MIKROKONTROLLER ATMEGA328 Oktafianto, Ponidi	137-142
7	PURWARUPA SISTEM PENGAIRAN SAWAH OTOMATIS DENGAN ARDUINO BERBASIS ARTIFICIAL INTELEAGENT Pamuji Setiawan, Elisabet Yunaeti Anggraeni	143-151
8	PENENTUAN PENERIMA KINERJA DOSEN AWARD MELALUI METODE TSUKAMOTO DENGAN KONSEP LOGIKA FUZZY Erlangga, Yanuaris Yanu Dharmawan	152-161
9	AUTOMATIC COUNTING MENGGUNAKAN METODE HAVERSINE UNTUK MENGHITUNG JUMLAH PENUMPANG BUS Yuthsi Aprilinda, Emy Sugandasari, Freddy Nur Afandi, Fenty Ariani	162-177
10	IMPLEMENTASI RUMAH LISTRIK BERBASIS SOLAR CELL Taqwan Thamrin, Erlangga, Wiwin Susanty	178-185

Fakultas Ilmu Komputer
Universitas Bandar Lampung

JIST	Volume 9	Nomor 2	Halaman	Lampung Oktober 2018	ISSN 2087 - 2062
------	----------	---------	---------	-------------------------	---------------------

**Jurnal Manajemen Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia & Informatika)**

Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Bandar Lampung

PENANGGUNG JAWAB

Rektor Universitas Bandar Lampung

Ketua Tim Redaksi:

Ahmad Cucus, S.Kom, M.Kom

Wakil Ketua Tim Redaksi:

Marzuki, S.Kom, M.Kom

TIM PENYUNTING :

PENYUNTING AHLI (MITRA BESTARI)

Mustofa Usman, Ph.D (Universitas Lampung)

Wamiliana, Ph.D (Universitas Lampung)

Dr.Iing Lukman, M.Sc. (Universitas Malahayati)

Penyunting Pelaksana:

Robby Yuli Endra S.Kom., M.Kom

Yuthsi Aprilinda, S.Kom, M.Kom

Fenty Ariani, S.Kom., M.Kom

Pelaksana Teknis:

Wingky Kesuma, S.Kom

Elva Riana Siregar, S.Kom

Alamat Penerbit/Redaksi:

Pusat Studi Teknologi Informasi - Fakultas Ilmu Komputer

Universitas Bandar Lampung

Gedung Business Center lt.2

Jl.Zainal Abidin Pagar Alam no.26 Bandar Lampung

Telp.0721-774626

Email: explore@ubl.ac.id

PENGANTAR REDAKSI

Jurnal explore adalah jurnal yang diprakasai oleh program studi Informatika, Fakultas Ilmu Komputer Universitas Bandar Lampung, yang di kelola dan diterbitkan oleh Fakultas Ilmu Komputer / Pusat Sudi Teknologi Informasi.

Pada Edisi ini, explore menyajikan artikel/naskah dalam bidang teknologi informasi khususnya dalam pengembangan aplikasi, pengembangan machine learning dan pengetahuan lain dalma bidang rekayasa perangkat lunak, redaksi mengucapkan terima kasih dan selamat kepada penulis makalah ilmiah yang makalahnya kami terima dan di terbitkan dalam edisi ini, makalah ilmiah yang ada dalam jurnal ini memberikan kontribusi penting pada pengembangan ilmu dan teknologi.

Selain itu, sejumlah pakar yang terlibat dalam jurnal ini telah memberikan kontribusi yang sangat berharga dalam menilai makalah yang dimuat, oleh sebab itu, redaksi menyampaikan banyak terima kasih.

Pada kesempatan ini redaksi kembali mengundang dan memberikan kesempatan kepada para peneliti, di bidang pengembangan perangkat lunak untuk mempublikasikan hasil penelitiannya dalam jurnal ini.

Akhirnya redaksi berharap semoga makalah dalam jurnal ini bermanfaat bagi para pembaca khususnya bagi perkembangan ilmu dan teknologi dalam bidang perekaan perangkat lunak dan teknologi pada umumnya.

REDAKSI

AUTOMATIC COUNTING MENGGUNAKAN METODE HAVERSINE UNTUK MENGHITUNG JUMLAH PENUMPANG BUS

Yuthsi Aprilinda¹, Emy Sugandasari², Freddy Nur Afandi³, Fenty Ariani⁴

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bandar Lampung

Jl. Zainal Abidin PA no. 26, Bandar Lampung 35142, Lampung, Indonesia

Email: yuthsi.aprilinda@ubl.ac.id, emy.14421012@student.ubl.ac.id,
freddsie@yahoo.com, fenty.ariani@ubl.ac.id

ABSTRAK

Informasi tentang jumlah penumpang bus sangat berguna bagi para calon penumpang sehingga masyarakat tidak merasa resah karena jumlah penumpang yang belum pasti. Implementasi Intelligent Transport System (ITS) merupakan suatu media yang dibutuhkan di tengah meningkatnya masalah transportasi yang dihadapi di Indonesia. Informasi waktu yang dibutuhkan nantinya dapat diinformasikan melalui aplikasi sehingga dapat memberikan informasi tentang estimasi waktu perjalanan. Informasi jumlah penumpang bus yang dibutuhkan nantinya dapat diinformasikan melalui aplikasi sehingga dapat memberikan informasi tentang jumlah penumpang. Dalam penelitian ini dilakukan penerapan perhitungan formula haversine yang bertujuan untuk melakukan perhitungan seberapa jarak kita terhadap suatu objek berdasarkan latitude dan longitude keberadaannya, menggunakan GPS sebagai alat input data koordinat yang akan dikirim ke database dan di tampilkan ke dalam aplikasi android.

Kata Kunci : Haversine, GPS

1. Pendahuluan

Perkembangan teknologi terus mengalami inovasi dan perkembangan dalam berbagai bidang kehidupan. Perkembangan teknologi untuk mempermudah aktifitas manusia dalam kehidupan sehari-hari. Manusia pada saat ini memang tidak bisa terlepas dengan yang namanya teknologi, di zaman modern dan era globalisasi ini kita dituntut untuk bisa memanfaatkan perkembangan teknologi yang ada terutama dibidang *mobile* pada jasa transportasi.

Transportasi merupakan kebutuhan manusia untuk dapat meningkatkan mobilitas manusia yang cukup padat. Peran transportasi juga sangat penting untuk dapat menghubungkan satu daerah dengan daerah lain. Transportasi digolongkan menjadi tiga jenis yaitu transportasi darat, laut, dan udara. Beberapa mode transportasi darat yang ada saat ini seperti sepeda, sepeda motor, mobil pribadi, truk bahkan angkutan publik seperti kereta api, angkot, dan bus. Kebutuhan masyarakat akan jasa transportasi di Indonesia tergolong besar. Transportasi dijadikan suatu bidang usaha yang sangat potensial dikarenakan semua aspek kehidupan membutuhkan transportasi

untuk mempercepat akses kepada suatu tujuan tertentu. (Rancangan Aplikasi Web Monitoring Estimasi Kedatangan Bus Trans-Jogja Berdasarkan Lokasi Bus Dengan GPS Smartphone, 2015, Yeyen Meithia Putri Jalni & Herman Yuliansyah). Kendaraan umum yang menjadi andalan transportasi para masyarakat seperti bus sering membuat ketidaknyamanan calon penumpangnya karena jumlah penumpang yang belum pasti. Untuk mengatasi keresahan para calon penumpang kendaraan umum, maka diperlukan sebuah aplikasi yang dapat menginformasikan mengenai jumlah penumpang bus sehingga dapat mengurangi tingkat ketidaknyamanan para calon penumpang bus.

Haversine formula adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. (Gintoro, 2010). Rumus *haversine* bertujuan agar pengguna sistem dapat melakukan perhitungan seberapa jarak kita terhadap suatu objek berdasarkan *latitude* dan *longitude* keberadaannya.

1.1 Identifikasi Masalah

Berdasarkan latar belakang masalah di atas maka penulis mendapatkan beberapa masalah sebagai berikut :

- a. Kendaraan umum yang menjadi andalan transportasi masyarakat seperti bus sering membuat ketidaknyamanan calon penumpang karena tidak diketahuinya jumlah penumpang di dalam bus.
- b. Transportasi bus jarang digunakan karna masyarakat lebih memilih menggunakan transportasi online.
- c. Bagaimana pendekatan haversine dalam mengetahui radius antara bus dan penumpang.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah “Bagaimanakah membangun aplikasi informasi jumlah penumpang bus dengan menerapkan metode *haversine*?”

2. Landasan Teori

2.1 Definisi Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan sebuah tugas yang diinginkan pengguna. (I Gusti Bagus Maha Putra, 2014). Istilah aplikasi berasal dari Bahasa Inggris “*application*” yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa rodok. (<http://id.wikipedia.org/wiki/Aplikasi>)

2.2 Definisi Android

Android merupakan OS (Operating System) mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, *i-Phone OS*, *Symbian* dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka. (Hermawan,2011).

a) Jenis-jenis android berdasarkan versinya :

1. *Android 8.0 Oreo*

Android Oreo dirilis sebagai preview pengembangan kualitas pada tanggal 21 Maret 2017. Pratinju pengembangan kedua dirilis pada 17 mei 2017, dan ini dianggap sebagai kualitas *beta* dan *preview* pengembangan ketiga dirilis pada tanggal 8 juni 2017 dan menyelesaikan API. Perkembangan keempat mencakup perilaku sistem akhir dan perbaikan *bug* dan pengoptimalan terbaru. Dirilis dipublik pada tanggal 2017. (*buku pintar Gadget Android, hal:16*)

2. *Android 7.0 Nougat*

Android Nougat diperkenalkan pada ajang kumpul *developer Google I/O* pertengahan 2016, pembaruan mendasar pada versi *nougat* adalah kehadiran *google assistant* yang menggantikan *google now*. *Asistant digital* tersebut lebih bisa diandalkan untuk menjalankan berbagai fungsi. Fitur-fitur baru lainnya mencakup layar *split-screen* saat dipakai *multitasking*, serta fitur *Doze* yang telah dikenalkan diversi *android marshmallow* namun telah ditingkatkan.

Android Nougat juga memiliki dukungan terhadap *platform virtual reality* terbaru google. (buku pintar *Gadget Android*, hal:17)

3. *Android 6 Marshmallow*

Android Marshmallow pertama kali dipertamakan mei 2015 dibawah codename “Android M”, secara resmi dirilis pada bulan oktober 2015. Fokus utama dari *Android Marshmallow* adalah pada peningkatan pengalaman pengguna secara keseluruhan untuk OS sebelumnya *android lollipop*, memperkenalkan arsitektur permissstions baru, API baru *contextual assistants* (sebuah fitur yang dikenal dengan “*Google Now On Top*”- sebuah kemampuan baru dari aplikasi google penelusuran), sebuah sistem manajemen daya baru yang mengurangi aktivitas di *background* saat perangkat tidak sedang digunakan, dukungan asli untuk pengenalan sidik jari dan konektor USB tipe-C, kemampuan untuk migrasi data kekartu microSD dan menggunakan sebagai penyimpanan utama, serta perubahan Internal lainnya (buku pintar *Gadget Android*, hal:17)

4. *Android 5.0 Lollipop*

Android Lolipop dirilis *preview* pada 25 juni 2014, menjadi tersedia dalam resmi *over-the-air (OTA) update* pada 12 November 2014, salah satu perubahan yang menonjol dalam rilis *Lollipop* adalah *user interface* didesain ulang dibangun disekitar bahasa desain yang dikenal sebagai “*material design*”. Perubahan lain termasuk perbaikan *notifications*, yang dapat diakses dari *lockscreen* dan ditampilkan dalam aplikasi *banner* dibagian atas layar. ((buku pintar *Gadget Android*, hal:17)

5. *Android 4.4 Kitkat*

Google mengumumkan *Android 4.4 Kitkat* (dinamai dengan izin dari *Nestle* dan *Hershey*) pada 3 september 2013, dirilis pada tanggal 31 oktober 2013. Keunggulannya diantaranya pembaruan antar muka dengan bar status dan navigasi transparan pada layar depan, Optimasi kinerja dengan spesifikasi perangkat yang lebih rendah, *NFC host card emulation* sebagai emulator kartu pintar, *webview* berbasis *chromium*, *sensor batching*, *stap*

detektor, dan *counter API*, peningkatan penampilan mode layar sentuh, penyeimbang audio, pemantauan audio, dan peningkatan suara audio, dukungan *bluetooth massage acces profile (MAP)*. (buku pintar *Gadget Android*, hal:20)

6. *Android 4.1 Jelly Bean*

Android jelly Been juga diluncurkan pada acara Google I/O 10 mei 2011 yang lalu. Keunggulan dan vitur baru fungsi ini, diantaranya peningkatan *input keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui *voice search* yang lebih cepat. Dilengkapi juga *Google Now* yang dapat memberikan Informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi *Android jelly Bean 4.1* pertama kali digunakan dalam produk tablet asus, yakni Google Nexus 7. (buku pintar *Gadget Android*, hal:20)

7. *Android 4.0 Ice cream sandwich*
Android ice cream sandwich diumumkan secara resmi pada 10 Mei 2011 pada ajang google IO *developer converence (sanfrenscisco)*, pihak google mengklaim *android ice creame sandwich* akan dapat digunakan baik di *smartphone* ataupun tablet. *Android ICS* membawa fitur *honeycomb* untuk *smartphone* serta ada penambahan fitur baru seperti membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan control terpadu untuk jaringan social, perangkat tambahan *fotography*, mencari *email* secara *offline*, dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy nexus. (buku pintar *Gadget Android*, hal:19)

8. *Android 3.0 Honeycomb*

Android honeycomb dirilis pada awal 2012. Versi android yang dirancang khusus untuk device dengan layar besar seperti tablet PC. fitur baru pada android honeycomb antara lain dukungan terhadap *prosesor multicore* dan grafis dengan *hardware exeleration*. *Userinterface* pada *honeycomb* juga berbeda, karena sudah didesain untuk tablet, tablet pertama yang memakai *honeycomb* adalah tablet motorola Xoom

yang dirilis bulan februari 2011. (*buku pintar Gadget Android, hal:19*)

9. Android 2.3 Gingerbread

Android gengerbread dirilis pada 6 Desember 2010, rubahan umum yang didapat dari android versi ini antara lain, peningkatan kemampuan permainan (*Gaming*), peningkatan fungsi *copy paste* layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan web N, efek audio baru (*reverb, equalizer, handphone virtualizer dan base boot*), dukungan kemampuan *near fill communicaion (NFC)* dan dukungan jumlah kamera yang lebih dari satu. (*buku pintar Gadget Android, hal:19*)

10. Android 2.2 Froyo (Frozen Yogurt)

Dirilis pada 20 Mei 2012, versi ini memiliki kecepatan kinerja dan aplikasi 2-5 kali dari versi sebelumnya. Selain itu penambahan fitur baru seperti dukungan *adobe flash 10.1*, integrasi *V8, javascript engine* yang dipakai *google chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam *SDK*, kemampuan *wifi hotspot portable* dan kemampuan *auto update* dalam aplikasi android market. (*buku pintar Gadget Android, hal:18*)

11. Android 2.0 Eclair

Android Eclair dirilis pada 3 Desember 2009. Perubahannya antara lain: pengoptimalan *hardware*, peningkatan *Google maps 3.1.2*, perubahan UI dengan browser baru dan dukungan *HTMLS*, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth 2.1*. *Android Eclair* adalah Android pertama yang mulai dipakai oleh banyak *smartphone*, fitur utama *Eclair* yaitu perubahan total struktur dan ampilan *desain interface*. (*buku pintar Gadget Android, hal:18*)

12. Android 1.6 Donut

Android Donut di rilis pada September 2009 menampilkan proses pencarian yang lebih baik dari versi-versi sebelumnya. Android Donut juga memiliki fitur-fitur tambahan seperti galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang diintegrasikan; *Text-to-speech engine*; dial kontak; teknologi *text to*

change speech, baterai indikator, dan kontrol applet VPN. (*buku pintar Gadget Android, hal:18*)

13. Android 1.5 Cupcake

Android Cupcake dirilis pada ertengahan mei 2009, masih oleh google Inc. Android ini telah dilengkapi software development kit dengan berbagai pembaharuan termasuk penambahan beberapa fitur lain, yakni kemampuan merekam dan menonton video ke youtube, upload gambar ke picasa langsung dari telepon, serta mendapat dukungan bluetooth A2DP. (*buku pintar Gadget Android, hal:17*)

2.3 Pengertian Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi Android (*Andy Juansyah : 2015*)

2.4 Pengertian Formula Haversine

Posisi di bumi dapat dipresentasikan dengan posisi garis lintang (latitude) dan bujur (longitude). Untuk menentukan jarak antara dua titik di bumi berdasarkan letak garis lintang dan bujur, ada beberapa rumusan yang digunakan. Semua rumusan yang digunakan berdasarkan bentuk bumi yang bulat dengan menghilangkan faktor bahwa bumi itu sedikit elips. Formulasi ini menggunakan rumus haversine sebagai dasar. Rumus ini dapat digunakan untuk menghitung jarak lingkaran yang jauh dari dua titik. Berikut merupakan rumus haversine formula (Veness, 2010).

$$\text{haversin}\left(\frac{d}{R}\right) = \text{haversin}(\Delta\phi) + \cos(\phi_1) \cos(\phi_2) \text{haversin}(\Delta\lambda)$$

Dimana, R adalah radius bumi (radius = 6371 km). (Paningrahi, 2014)

$$\text{jarak} = 2r \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{\text{Lat}2 - \text{Lat}1}{2} \right) + \cos(\text{Lat}2) \cdot \cos(\text{Lat}1) \cdot \sin^2 \left(\frac{\text{Lon}2 - \text{lon}1}{2} \right)} \right)$$

Gambar 2.1 Rumus Haversine

Keterangan :

Δ lat : selisih antara latitude 1 dan latitude 2
 Δ long = selisih antara longitude 1 dan longitude 2

φ = Latitude

λ = Longitude

r = Radius Bumi = 6371 km

1 derajat = 0.0174532925 radian

d = jarak yang dihasilkan

(Rofiq & Uzzy, 2014)

2.5 Global Positioning System (GPS)

Menurut Atie, dkk. (2003) dalam *e-book* yang berjudul *Sistem Informasi Geografis Untuk Pengelolaan Sumber Daya Alam*, menyatakan bahwa GPS adalah suatu jaringan satelit yang secara terus menerus memancarkan sinar radio dengan frekuensi yang sangat rendah. Alat penerima GPS secara pasif menerima sinyal ini, dengan syarat bahwa pandangan ke langit tidak boleh terhalang, sehingga biasanya alat ini hanya bekerja di ruang terbuka. Satelit GPS bekerja pada referensi waktu yang sangat teliti dan memancarkan data yang menunjukkan lokasi dan waktu pada saat itu. Operasi dari seluruh satelit GPS yang ada disinkronisasi sehingga memancarkan sinyal yang sama. Alat penerima GPS akan bekerja jika ia menerima sinyal dari sedikitnya 4 buah satelit GPS, sehingga posisinya dalam 3 dimensi bisa dihitung. Pada saat ini ada sedikitnya 24 satelit GPS yang beroperasi setiap waktu dan dilengkapi dengan beberapa cadangan. Satelit tersebut dioperasikan oleh departemen pertahanan amerika serikat, mengorbit selama 12 jam (2 orbit per hari) pada ketinggian sekitar 11.500 mil dan bergerak dengan kecepatan 2000 mil per jam.

GPS adalah sistem yang menggunakan satelit yang di desain agar dapat menyediakan posisi secara instan, kecepatan dan informasi waktu di hampir semua tempat di muka bumi. Pada dasarnya, GPS merupakan aplikasi yang harus menunggu terlebih dahulu permintaan dari pengguna. Aplikasi ini menyediakan akurasi positioning atau penentuan posisi yang berkisar antara 100 meter juga sampai akurasi relatif pada submeter. Secara umum, semakin tinggi akurasi yang dihasilkan akan memerlukan infrastruktur

yang lebih canggih dan tentunya berhubungan dengan biaya yang akan dikeluarkan. Penggunaan GPS untuk penentuan posisi saat ini di antaranya adalah navigasi untuk kegiatan pribadi, navigasi di pesawat, survei di lepas pantai dan navigasi kapal, fleet tracking, pengendalian mesin, teknik sipil, survei daratan, GIS dan pemetaan, analisis deformasi dan sebagainya. (Hartanto, 2003)

2.6 Database

Satu Database Management Sistem (DBMS) berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi DBMS terdiri dari database dan set program pengelola untuk menambah data, menghapus data, mengambil dan membaca data.

Database adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan, instansi.

Dalam satu file terdapat record-record yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan entity yang seragam. Satu record terdiri dari field-field yang saling berhubungan untuk menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record. (Kristanto, 2007)

3. Metode Penelitian

Metode penelitian adalah cara ilmiah untuk mendapatkan data yang valid dengan tujuan dapat ditemukan, dibuktikan, dan dikembangkan suatu pengetahuan sehingga pada gilirannya dapat digunakan untuk memahami, memecahkan, dan mengantisipasi masalah. (Sugiono, 2010).

3.1 Jenis Metode Penelitian

Jenis metode penelitian yang digunakan pada tugas akhir ini adalah jenis metode penelitian deskriptif kualitatif. Metode deskriptif kualitatif bertujuan menggambarkan secara sistematis dan akurat fakta dan karakteristik mengenai bidang tertentu dengan menekankan analisisnya pada data-data yang di olah. Sugiyono (2005) menyatakan bahwa metode deskriptif adalah suatu metode yang digunakan untuk menggambarkan atau

menganalisis suatu hasil penelitian tetapi tidak digunakan untuk membuat kesimpulan yang lebih luas.

3.2 Teknik Pengumpulan Data

Untuk mendapatkan gambaran secara umum maka diperlukan beberapa teknik pengumpulan data yang antara lain meliputi Study Pustaka (*Library Reseach*) dan Pengamatan (*Observasi*).

3.2.1 Studi Pustaka (*Library Research*)

Untuk mendukung penulisan, penulis melakukan pengumpulan data melalui panduan-panduan buku referensi, paper dan literature lain yang berhubungan dengan pengetahuan teoritis mengenai masalah yang sedang diteliti. Yang kemudian dijadikan data untuk diolah lebih lanjut.

3.2.2 Pengamatan (*Observasi*)

Observasi dilakukan untuk mengumpulkan data dengan cara melalui pengamatan langsung pada objek penelitian yaitu konfigurasi data, hal tersebut dilakukan guna mendapatkan gambaran secara menyeluruh dan jelas mengenai sistem.

3.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mengetahui spesifikasi fungsi yang mampu dilakukan oleh sistem. Analisis kebutuhan adalah kebutuhan yang berasal dari pemangku kepentingan (*Stakeholder*) termasuk fungsi dan fitur dari sebuah sistem.

Aplikasi jumlah penumpang bus menggunakan formula *haversine* pada *intelligent transportation system* memiliki beberapa kemampuan, antara lain:

1. Pendeteksian dilakukan menggunakan GPS yang dikirimkan melalui smartphone, kemudian device yang terdeteksi kedalam aplikasi induk membaca setiap lokasi keberadaan dari user sebagai penumpang yang menggunakan aplikasi.
2. Aplikasi pendeteksian jumlah penumpang dengan tahapan formula *haversine* untuk menghitung jarak dan estimasi waktu dari setiap user yang terhubung ke satu pusat jaringan untuk mengetahui apakah user

tersebut dihitung sebagai penumpang atau bukan.

3.4 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional adalah kebutuhan untuk mendeskripsikan kualitas suatu sistem.

1. Tingkat keakuratan deteksi dan perhitungan radius pada aplikasi penentuan jumlah penumpang dirasa mampu mengatasi masalah perhitungan penumpang dalam satu buah kendaraan,
2. Kemampuan dari aplikasi penentuan jumlah penumpang ini mampu menghitung radius dengan menggunakan formula *haversine* untuk menentukan keberadaan user dalam satu buah jaringan terpusat.

3.5 Analisis Pengguna

Penggunaan dari perangkat lunak yang akan dikembangkan ini terdiri dari user sebagai penumpang dan supir, *user* sebagai supir memiliki aplikasi yang akan membaca lokasi terdekat dari setiap user sebagai penumpang. Aplikasi dari *user* penumpang akan mengirimkan lokasi yang akan dibaca oleh aplikasi supir serta menghitung radius dari *user* penumpang diarea supir. Lalu penentuan jarak yang terjadi apabila jarak terdekat dengan supir dan radius yang ditetapkan sebagai penentu apakah user tersebut adalah penumpang atau bukan.

3.6 Alur Sistem

Dalam aplikasi penentuan jumlah penumpang ini memiliki alur kerja sebagai berikut :

1. Aplikasi terdiri dari dua user yaitu user sebagai penumpang dan *user* sebagai supir.
2. Aplikasi dengan *user* sebagai supir akan menangkap dan menghitung jarak radius dari setiap keberadaan penumpang didekat aplikasi supir.
3. Aplikasi *user* sebagai penumpang akan mengirimkan lokasi keberadaannya melalui GPS yang akif pada *smartphone android*.
4. Setelah *user* sebagai penumpang terbaca

keberadaannya dalam area supir, *formula haversine* berjalan menentukan jarak terdekat kedua aplikasi, jika masuk kedalam area jaringan supir maka *user* tersebut dinyatakan sebagai calon penumpang yang mungkin akan memasuki kendaraan umum. Selanjutnya *formula haversine* akan menghitung jarak dengan durasi yang ditentukan sebagai indikasi apakah *user* tersebut adalah penumpang atau hanya calon penumpang, semisal jika *user* calon penumpang berada dekat selama 10 menit dengan jaringan terpusat yang dibangun aplikasi supir maka *user* tersebut dinyatakan sebagai penumpang, jika dalam kurun waktu 10 menit jaringan terputus antara kedua aplikasi maka *user* tersebut bukan dinyatakan sebagai penumpang.

- Selanjutnya aplikasi akan memuat jumlah penumpang.



Gambar 3.1 Arsitektur Sistem

3.7 Formula Haversine

3.8 Perancangan Aplikasi

Gambar 3.2 Login

Untuk menentukan jarak antara dua titik berdasarkan letak garis lintang dan bujur, ada beberapa rumusan yang digunakan. Semua rumusan yang digunakan berdasarkan bentuk bumi yang bulat dengan menghilangkan faktor bahwa bumi itu sedikit elips. Rumus ini dapat digunakan untuk menghitung jarak lingkaran yang jauh dari dua titik. Berikut merupakan rumus *haversine formula* (Veness, 2010).

Dimana, R adalah radius bumi (radius = 6371 km).

(Narayan Paningrahi, 2014)

$$jarak = 2r \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{Lat2 - Lat1}{2} \right) + \cos(Lat2) \cdot \cos(Lat1) \cdot \sin^2 \left(\frac{Lon2 - Lon1}{2} \right)} \right)$$

Keterangan :

Δlat : selisih antara latitude 1 dan latitude 2

Δlong = selisih antara longitude 1 dan longitude 2

φ = Latitude

□ = Longitude

r = Radius Bumi = 6371 km

1 derajat = 0.0174532925 radian

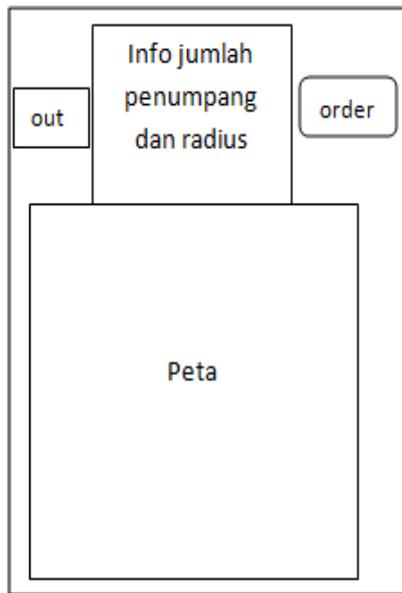
d = jarak yang dihasilkan

(M. Rofiq, 2014).

$$jarak = 2r \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{Lat2 - Lat1}{2} \right) + \cos(Lat2) \cdot \cos(Lat1) \cdot \sin^2 \left(\frac{Lon2 - Lon1}{2} \right)} \right)$$

$$\begin{aligned} \text{Jarak} &= 2.6371 \cdot \arcsin \left(\sqrt{\sin^2(0,0000000618) + \cos(-5.375845) \cdot \cos(-5.376342) \cdot \sin^2(0,000000632)} \right) \\ &= 12,742 \cdot \arcsin \left(\sqrt{(0,0000000011) + (0,9956015507) \cdot (0,995600738) \cdot (0,000000011)} \right) \\ &= 12,742 \cdot \arcsin \left(\sqrt{(0,0000000011) + (1,09034380 \times 10^{-8})} \right) \\ &= 12,742 \cdot \arcsin(\sqrt{0,000000012}) \\ &= 12,742 \cdot \arcsin(0,0001095445) \\ &= \mathbf{0,001396 \text{ km} = 1,3 \text{ m}} \end{aligned}$$

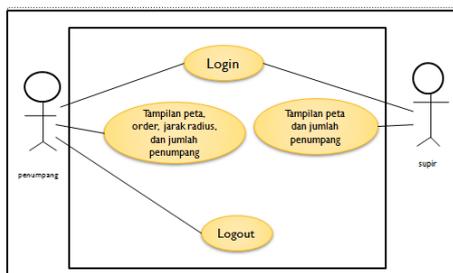
Gambar 3.3 Menu Supir



Gambar 3.4 Menu Penumpang
User melakukan regist terlebih dahulu agar dapat login masuk ke dalam aplikasi perhitungan jumlah penumpang.

3.8.1 Perancangan Sistem

Perancangan sistem menunjukkan alur aplikasi penumpang yang terdapat koordinat penumpang untuk dikirim ke server menggunakan bahasa PHP dan diubah dalam bentuk file json yang akan dikirim ke server lalu di kirim ke aplikasi supir bus untuk memberikan informasi posisi penumpang. Begitu juga aplikasi supir bus dapat mengaktifkan gps sehingga koordinat terkirim ke server untuk dikirim

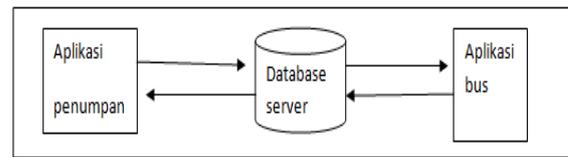


Gambar 3.10 use case diagram

3.9.4 Alur Sistem

Berikut alur sistem aplikasi yang dimulai dari tahap awal input hingga jumlah penumpang dapat terlihat di aplikasi. Pada tahap awal dilakukan input username dan password untuk dapat logi ke aplikasi. Setelah input maka akan masuk ke menu peta dimana sinyal gps penumpang dikirim

ke aplikasi penumpang sehingga penumpang dapat mengetahui posisi dan jumlah penumpang di dalam bus.



Gambar 3.5 perancangan sistem

3.9 Unified Modeling Language (UML)

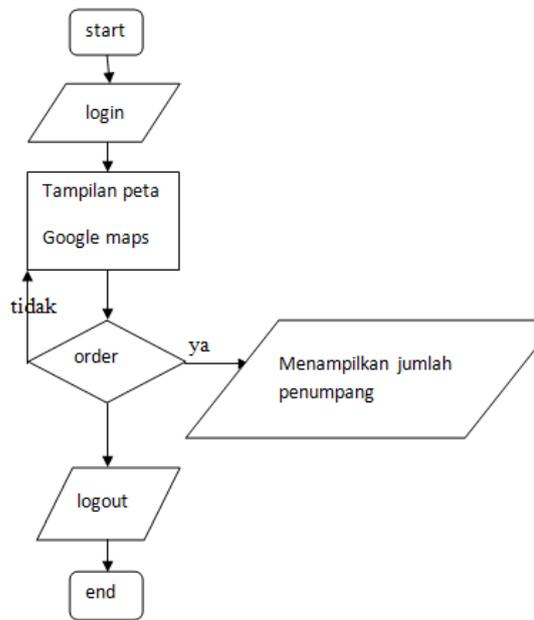
Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language (UML)*. UML muncul karna adanya kebutuhan karna adanya kebutuhan pemodelan visual untuk menspesifikasikan menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

3.9.1 Use Case Diagram

Use Case Diagram

mendesripsikan sebuah intraksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

ke server untuk diolah dan ditampilkan ke aplikasi supir. Selanjutnyaa data di masukan ke dalam aplikasi android. Kemudian data koordinat dicari perhitungan jarak radius menggunakan formula haversine, jika koordinat tidak di dapat maka kembali ke server untuk di proses kembali, kemudian kembali ke perhitungan haversine untuk di tampilan ke aplikasi.



Gambar 3.10 Flowchart Alur Sistem

3.10 Pengujian Program (*Testing*)

Pada penelitian ini tahap *testing* dilakukan setelah tahap pembuatan dengan menjalankan aplikasi dengan bertujuan untuk melakukan uji kelayakan program dan melakukan pengecekan apakah terjadi kesalahan atau tidak. Uji fungsionalitas dilakukan dengan pengujian blackbox dan whitebox.

3.10.1 Pengujian *Whitebox*

Menurut Black (2009 :2), Structural test (atau yang biasa disebut white-box tests dan glass-box tests) menemukan bug dalam elemen struktur tingkat rendah seperti yang terjadi di tingkatan code , data base scemas, chips, subassemblies dan interfaces. Pengujian struktural ini didasarkan pada bagaimana suatu sistem beroperasi. Contohnya, pengujian struktural akan mengungkapkan tempat penyimpanan database masih memiliki ruang penyimpanan username sebanyak 80 karakter, tetapi pada kenyataannya hanya memungkinkan menyimpan 40 karakter. Kasus yang sering menggunakan white box testing akan di uji dengan beberapa tahapan yaitu:

1. Pengujian seluruh keputusan yang menggunakan logikal.
2. Pengujian keseluruhan loop yang ada sesuai batasan-batasannya.

3. Pengujian pada struktur data yang sifatnya internal dan yang terjamin validitasnya.

Kelebihan Yang Terdapat Di White Box Testing:

Kesalahan Logika, Menggunakan syntax „if“ dan syntax pengulangan. Dan langkah selanjutnya metode white box testing ini akan mencari dan mendeteksi segala kondisi yang di percaya tidak sesuai dan mencari kapan suatu proses perulangan di akhiri.

Kesalahan Pengetikan, Mendeteksi dan mencari bahasa-bahasa pemrograman yang di anggap bersifat case sensitif.

Kelemahan White Box Testing:

Pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.

Pengujian White Box Testing:

Pengujian ke white box testing adalah menguji yang di dasarkan kepada pengecekan ke dalam detail rancangan, penggunaan yang di lakukan struktur control dari suatu desain pemrograman untuk dapat membagi pengujian ke beberapa kasus pengujian. Dan di dapat bahwasanya white box testing menggugurkan petunjuk untuk menghasilkan program yang di harapkan dan efisien.

Metode pengujian pada white box testing ini sering di lakukan untuk:

1. Memberikan dan membuat suatu jaminan bahwa seluruh jalur-jalur yang independen hanya menggunakan modul minimal satu kali.
2. Keputusan yang sifatnya logis dapat di gunakan di semua kondisi true (benar) atau false (salah).
3. Mengeksekusi seluruh perulangan yang ada ke pada batas nilai dan operasional di setiap situasi dan kondisi.
4. Syarat yang di lakukan dalam menjalankan strategi white box testing
5. Mendefinisikan tentang seluruh alur-alur logika yang ada.

6. Membangun dan membuat suatu kasus yang akan di gunakan untuk tahap pengujian.
7. Hasil pengujian yang telah di dapatkan akan di lakukan eveluasi kembali.
8. Pengujian yang di lakukan haruslah secara menyeluruh.

3.10.2 Pengujian *Blackbox*

Pengujian menggunakan sekumpulan aktifitas validasi, dengan pendekatan *black box testing*. Menurut Shalahuddin dan Rosa (2011), *black box testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji

desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black box testing* harus dibuat dengan kasus benar dan kasus salah. Perencanaan pengujian *blackbox* yang akan dilakukan penulis dapat dilihat pada tabel berikut :

Tabel 3.1 Perencanaan Pengujian *Blackbox*
PENGUJIAN FORM LOGIN, PENGUJIAN DATA NORMAL
(Tampilan Awal)

Data Masukan	Data Diharapkan	Pengamatan	Kesimpulan
Klik tombol “Login”	Ketik username dan password	Tampilan ke menu peta	[] Diterima [] Ditolak

PENGUJIAN FORM MENU UTAMA

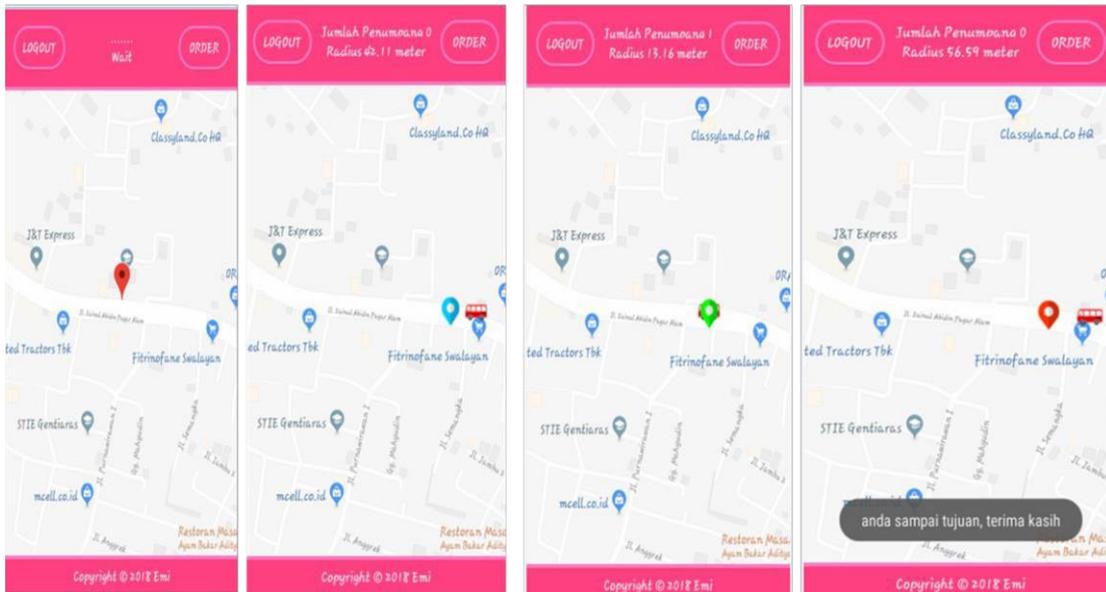
Data Masukan	Data Diharapkan	Pengamatan	Kesimpulan
Klik menu “peta”	Tampilan ke peta, jumlah penumpang, dan order.	Menampilkan posisi bus dan penumpang.	[] Diterima [] Ditolak
Klik menu “order”	Tampilan radius dan perubahan warna simbol pada penumpang	Menampilkan jumlah penumpang, dan radius antara bus dan penumpang.	[] Diterima [] Ditolak
Klik menu “logout”	Tampilan akan keluar dari aplikasi	Keluar dari aplikasi	[] Diterima [] Ditolak

4. Hasil Dan Pembahasan

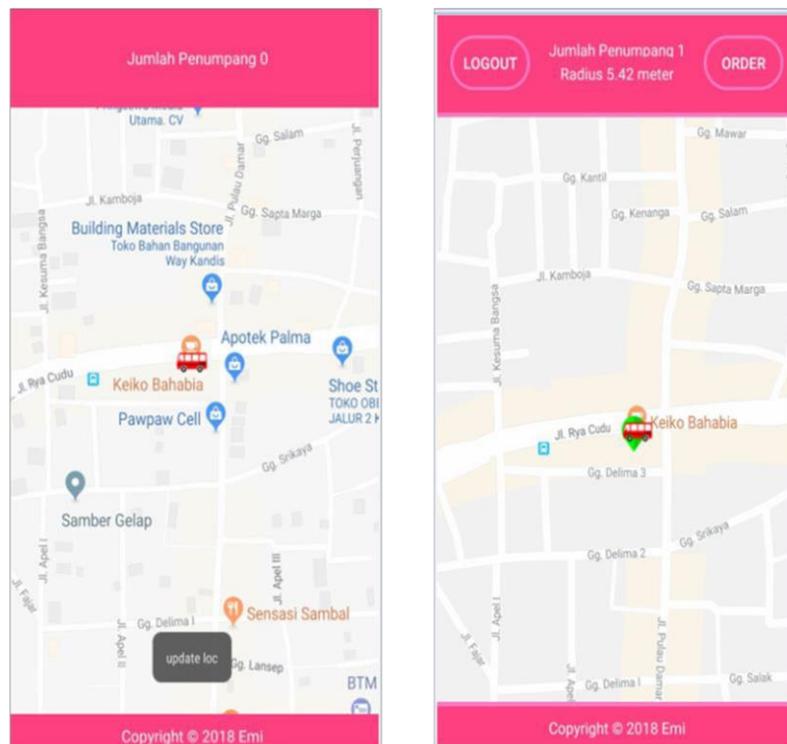
4.1 Aplikasi Penumpang

Aplikasi Penumpang untuk menjalankan Program yang telah dikirimkan koordinat supir dari server

Aplikasi penumpang pagar alam



Aplikasi penumpang sukarama



Gambar 4.1 Aplikasi Penumpang

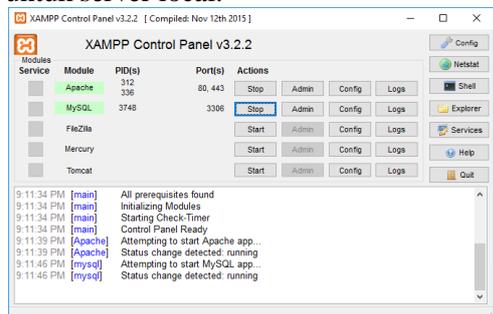
Client yang dimaksud berupa aplikasi android yang telah dirancang sesuai kebutuhan dengan menggunakan formula

haversine sebagai perhitungan dalam mencari jarak dan aplikasi dapat menampilkan jumlah penumpang. Berikut hasil dari tampilan aplikasi android. Penulis menggunakan server local untuk menjalankan aplikasi penumpang sehingga tidak ada kendala dalam menjalankan program yang telah dibuat.

4.2 Pembahasan

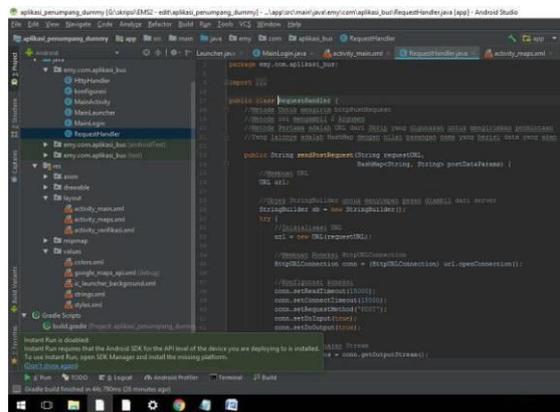
4.2.1 Menghidupkan Xampp

Langkah awal yang dilakukan dengan cara menghidupkan xamp di laptop untuk server local.



Gambar 4.2 Menghidupkan xamp

Setelah menghidupkan xamp maka jalankan program android studio seperti gambar di bawah ini:



Gambar 4.3 Program Android

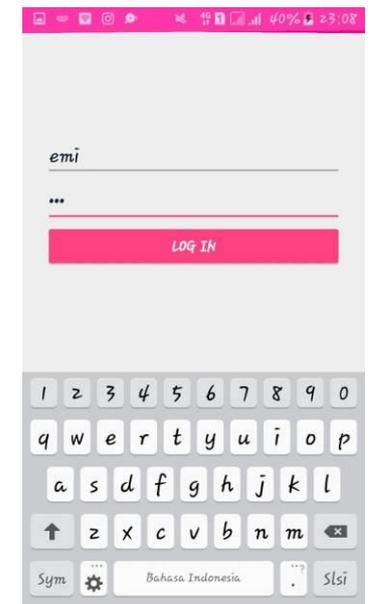
Buka browser localhost/gps/dummy.php lalu klik kanandan pilih reload sehingga menampilkan gambar seperti berikut:



Gambar 4.4 localhost

4.2.2 Proses Di Dalam Android Studio

Dalam hal ini penulis menjalankan dengan klik run pada android dan buka aplikasi seperti gambar berikut:

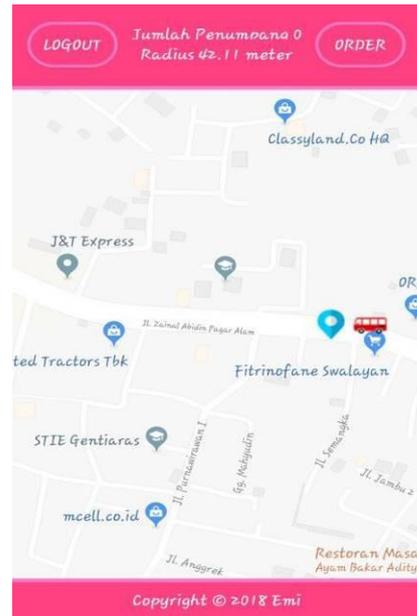


Gambar 4.5 Login

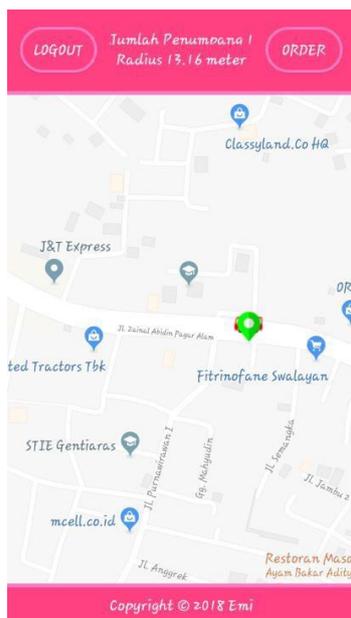
Selanjutnya menampilkan aplikasi sebelum order sebagai berikut:



Gambar 4.6 sebelum Order



Gambar 4.6 sudah order



Gambar 4.7 (c) penumpang naik bis.

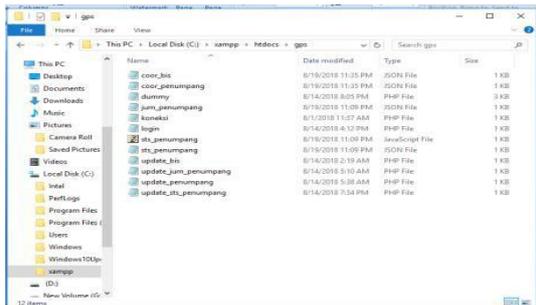


Gambar 4.7 (d) penumpang turun bis

Jika warna user pertama merah menandakan bahwa penumpang baru membuka aplikasi, jika user berwarna biru maka penumpang sudah klik order di aplikasi, jika user berwarna hijau maka penumpang telah menaiki bus, jika user kembali merah maka user sudah turun dari bus dan telah sampai tujuan.

4.2.3 File Dalam Server

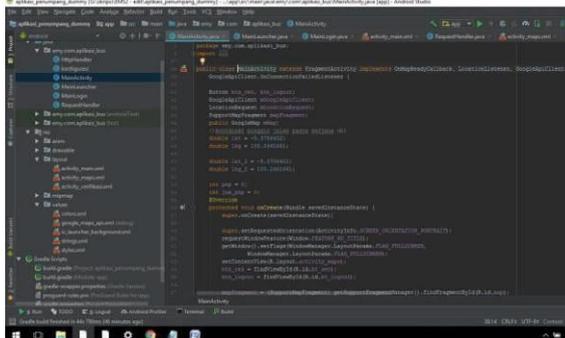
File php di dalam server digunakan untuk konfigurasi database, dan sebagai post php yang di butuhkan dalam memproses data gps penumpang yang di kirimkan dari program ahingga ke supir, begitupun sebaliknya. Berikut tampilan file:



Gambar 4.8 File dalam server

4.2.4 Android Studio

Aplikasi android berupa aplikasi penumpang telah dirancang sebagai berikut:



Gambar 4.9 Android Studio

Terdiri dari manifest, class java untuk menjalankan fungsi konfigurasi, login, splash screen, dan sebagainya sesuai dengan kebutuhan pembuatan aplikasi, lalu di buat juga tampilan desain dan penggunaan build.gradle yang disesuaikan. Pada gambar diatas terdapat perhitungan jarak yang mengukur antara penumpang dan posisi bus secara garis lurus mengikuti lengkungan bumi sehingga diterapkan *coding* sebagai berikut:

```

-----Haversine-----
-----
public static double Haversine(double
lat1,double lng1, double lat2, double lng2)
{
    double earthRadius = 6371000;
//meters
    double dLat = Math.toRadians(lat2-
lat1);
    double dLng = Math.toRadians(lng2-
lng1);
    double a = Math.sin(dLat/2) *
Math.sin(dLat/2) +
    Math.cos(Math.toRadians(lat1))
* Math.cos(Math.toRadians(lat2)) *
    Math.sin(dLng/2) *
Math.sin(dLng/2);

```

```

    double c = 2 *
Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
    double dist = earthRadius * c;
    return dist;
}

```

```

-----warna user-----
-----
public void onLocationChanged(Location
location) {
    if (location != null) {
        TextView view =
(TextView)findViewById(R.id.textView2);
        TextView view2 =
(TextView)findViewById(R.id.textView);
        // lat = location.getLatitude();
        // lng = location.getLongitude();
        new GetCoor().execute();
        new
GetCoor_penumpang().execute();
        new StatusPNP().execute();
        new JumPNP().execute();
        LatLng loc_usr = new LatLng(lat,
lng);
        LatLng loc_pnp = new

```

```

LatLng(lat_2, lng_2);
        mMap.clear();
        mMap.addMarker(new
MarkerOptions()
            position(loc_pnp).
            icon(BitmapDescriptorFactory.fromResour
ce(R.drawable.bis)).
//gambar bus
            title(String.valueOf(lat_2) + ","
+ String.valueOf(lng_2)));
        if((pnp == 1) && (jum_pnp == 1)){
//saat di bis
            mMap.addMarker(new
MarkerOptions()
                position(loc_usr).
                icon(BitmapDescriptorFactory.fromResour
ce(R.drawable.pnp_3)).
//user warna hijau
                title(String.valueOf(lat) + "," +
String.valueOf(lng))
            );}
        else if((pnp == 1) && (jum_pnp ==
0)) //order bis
        {
            mMap.addMarker(new
MarkerOptions()

```

```

        position(loc_usr).

icon(BitmapDescriptorFactory.fromResource(R.drawable.pnp_2)).
//user warna biru
        title(String.valueOf(lat) + ","
+ String.valueOf(lng))
    );
    }else //jauh dari bis
    {
        mMap.addMarker(new
MarkerOptions().
        position(loc_usr).

icon(BitmapDescriptorFactory.fromResource(R.drawable.pnp_1)).
//user warna merah
        title(String.valueOf(lat) + ","
+ String.valueOf(lng))
    );
    }
    //
mMap.moveCamera(CameraUpdateFactory
.newLatLng(loc_usr));
        update_loc(String.valueOf(lat),
String.valueOf(lng));
        double conv =
Haversine(lat,lng,lat_2,lng_2);
        if ((conv < konfigurasi.RADIUS)
&& (pnp == 1)){ // radius penumpang 15
meter
            jum_penumpang("1");
        }
        else if ((conv > 15.0) && (jum_pnp
== 1)){
            sts_penumpang("0");
            jum_penumpang("0");
        }
        else
        { }
        view.setText("Jumlah Penumpang "
+ String.valueOf(jum_pnp));
        view2.setText("Radius "
+ String.format ("%0.2f", conv) + " meter");
    }
}

-----konfigurasi-----
public class konfigurasi {
    public static final String IP_SET =
"http://192.168.43.182"; //setting ip server

```

```

    public static final String URL_ADD =
IP_SET + "/gps/update_penumpang.php";
//mengirim lokasi penumpang
    public static final String
URL_ADD_LOGIN = IP_SET +
"/gps/login.php"; //mengirim login
    public static final String
URL_ADD_STS_PNP=IP_SET +
"/gps/update_sts_penumpang.php";
//mengirim status order penumpang
    public static final String
URL_ADD_JUM_PNP=IP_SET +
"/gps/update_jum_penumpang.php";
//mengirim jumlah penumpang
    public static final String URL_CALL =
IP_SET + "/gps/coord_bis.json";
//memanggil koordinat bis
    public static final String
URL_CALL_PENUMPANG = IP_SET +
"/gps/coord_penumpang.json";
//memanggil koordinat penumpang
    public static final String URL_STS_PNP
= IP_SET + "/gps/sts_penumpang.json";
//memanggil status penumpang
    public static final String
URL_JUM_PNP = IP_SET +
"/gps/jum_penumpang.json";
//memanggil jumlah penumpang
    public static final String LAT = "lat";
//latitude bis
    public static final String LNG = "lng";
//longitude bis
    public static final String STS = "sts"; //
status bis
    public static final String JUM =
"jumlah"; // jumlah bis
    public static final String USERNAME =
"username"; //username
    public static final String PASSWORD =
"password"; //password
    public static final double RADIUS =
15.0; //radius antara penumpang dan bis }
-----Login-----
public class MainLogin extends
AppCompatActivity {
    Button btn_login;
    EditText username;
    EditText password;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```
setContentViewById(R.layout.activity_verifikasi);
    username =
    (EditText)findViewById(R.id.username);
    password =
    (EditText)findViewById(R.id.password);
    btn_login =
    findViewById(R.id.btnLogin);
    btn_login.setOnClickListener(new
    View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String cek_username =
            username.getText().toString();
            String cek_password =
            password.getText().toString();

            login(cek_username,cek_password);
        }
    });
    Login ae = new Login();
    ae.execute();
}
```

5. Kesimpulan

1. Perhitungan jarak dan radius menggunakan haversine hanya garis lurus antara bus dan penumpang yang mengikuti lengkungan bumi.
2. Ada beberapa faktor yang mempengaruhi kendala aplikasi, yaitu faktor keakuratan GPS yang digunakan, pemilihan metode yang tepat, juga kekuatan sinyal untuk menangkap koordinat dalam aplikasi.
3. Dalam penelitian menggunakan satu bus, namun dapat dikembangkan di beberapa bus lainnya.
4. User dapat melihat posisi bus dan informasi jumlah penumpang saat ini dari aplikasi yang dapat dilihat setiap saat
5. Keakuratan posisi bus tergantung jenis GPS yang digunakan, jika GPS semakin berkualitas maka posisi bus semakin akurat.

Daftar Pustaka

- [1] Andy Juansyah (2015). *Pengertian Android Studio*
- [2] Anfonious Aditya Hartanto (2003), (ebook) *Mengenal teknik dan bisnis location based service* : hal.19, ISBN 979- 20-4220-2
- [3] Atie, dkk (2003). *Global Positioning System (GPS)*
- [4] Chopde, Nitin R. Mangesh K .Nikhat, *landmark Shortest Path Deection by Using A* and Haversine Formula*. ISSN (Print) : 2320 – 9798. *Interational Journal of Innovative Research in computer and Communication Engineering*. Vol. 1, Issue 2, April 2013.
- [5] Harianto Kristanto, *Konsep dan Perancangan Database*, ISBN 979-533- 861-7
- [6] I Gusti Bagus Maha Putra (2014). *Definisi Aplikasi*
- [7] M.Rofiq, Riza Fathul. *Penentuan Jalur terpendek menuju café dikota malang menggunakan metode Bellman-ford dengan location based service berbasis android*. Vol.8 No 2
- [8] Narayan Paningrahi (2014), (e-book) *Computing In Geographic Information System* : 212-214, ISBN 978-1-4822-2314-9)
- [9] Pathil, Swati. Dan saroja M.kulkarni, *Real Time Tracking System Using GPS*. *International Journal of Innovative Research in Computer Science & Technology*. ISSN: 2347-5552, Volume-2, Issue-3,May 2014.
- [10] Yeyen meithia Putri jalni & herman yuliansyah (2015). *Rancangan Aplikasi web Monitoring Estimasi Kedatangan BUS Trans-jogja berdasarkan lokasi Bus dengan GPS smantphm*

Redaksi :
Research Of Information Technology Universitas Bandar Lampung
Gedung Business Center Lt. 2
Jl. Zainal Abidin No. 26 Bandar Lampung
Telp. 0721 - 774626
e-Mail : explorer.rit@ubl.ac.id