

Optimasi Penjadwalan Maintenance Kendaraan Menggunakan Alat GPS Tracking Berbasis Android

Luqman Affandi¹, Sofyan Noor Arief², Muhammad Iko Wiryadirja³

¹Teknologi Informasi, ²Teknik Informatika, ³Politeknik Negeri Malang

¹luqman.affandi@polinema.ac.id, ²sofyan.89@gmail.com, ¹ikowirya@gmail.com

Abstract

As the primary need for daily transportation, it requires to check vehicle regularly. Vehicle in daily use causes engine performance to decrease. The mile age experience by the vehicle may cause the spare parts wear out. At this time, to arrange schedule for maintenance still recorded manually. It may cause several risk, such as oversight during recorded time for the distance per period. Therefore, it requires help from GPS Tracking to optimize the schedule for maintenance to run effectively. Using the GT06N device with a 5-second data delivery time setting the training data is calculated based on the location of latitude and longitude. While the configuration of the data transmission time is 10 seconds, it has a accuracy of 94.61% in 15 data training. The data transmission rate is greater, the greater the percentage rate of errors.

Keyword: GPS Tracking, GT06N, Maintenance Kendaraan, Representational State Transfer (REST)

1. PENDAHULUAN

Kendaraan merupakan alat transportasi yang menjadi kebutuhan primer. Peranan kendaraan berfungsi sebagai penunjang mobilitas individu maupun kelompok. Kecenderungan memakai kendaraan membuat pengendara terkadang melupakan tentang kondisi dari kendaraan itu sendiri. Kendaraan juga membutuhkan servis ringan secara periodik, ini menjadi sangat penting untuk mencegah serius masalah serius terjadi.

Salah satu indikator yang menyebabkan kondisi performa mesin menurun disebabkan dari faktor pemakaian, semakin jauh jarak tempuh yang digunakan maka efeknya akan semakin cepat suku cadang mengalami keausan. Hal ini banyak terjadi di rental kendaraan karena faktor kurangnya sistem perawatan yang baik. Perawatan mesin mempunyai peran yang penting bagi perusahaan rental kendaraan. Dalam menunjang kegiatan rental kendaraan, perusahaan harus melakukan perencanaan jadwal yang baik dalam melakukan perawatan mesin.

Pada saat ini pembuatan penjadwalan *maintenance* kendaraan di rental masih berupa pencatatan manual. Pencatatan dilakukan secara manual memiliki resiko seperti kelalaian

pengelola rental atau pemilik dalam melakukan pencatatan kilometer kendaraan.

Oleh karena itu perlu adanya sebuah sistem perawatan yang baik. Cara mudah untuk melakukan perawatan yang baik dengan melakukan pencatatan jarak tempuh berdasarkan kilometer kendaraan secara otomatis serta perhitungan dari interval waktu dan membuat notifikasi servis kendaraan secara *real-time*. Jika jarak tempuh odometer mendekati batas *maintenance* kendaraan, maka untuk penyewaan selanjutnya harus melakukan *maintenance* kendaraan terlebih dahulu. Untuk mengatasi perbedaan umur penggunaan suku cadang kendaraan, maka dapat dibuat pengaturan batas *maintenance* suku cadang dari masing-masing kendaraan yang dapat dikelola oleh pemilik atau pengelola rental. Ada beberapa teknologi yang dapat membantu untuk membuat penjadwalan servis kendaraan secara *real-time*. Teknologi yang digunakan untuk mendukung penjadwalan servis menggunakan alat *GPS Tracking*.

Dari permasalahan pencatatan odometer manual muncul sebuah ide untuk membangun sebuah sistem “Optimasi Penjadwalan *Maintenance* Kendaraan Menggunakan Alat *GPS Tracking* Berbasis *Android*”. Alat ini bekerja di *platform android* melihat banyak pengguna *android* di Indonesia yang akan

memudahkan pengguna untuk mengetahui kapan servis harus dilakukan. Pada pengembangan ini dapat menampilkan lokasi kendaraan, riwayat perjalanan kendaraan serta perhitungan jarak tempuh odometer dan perhitungan dari interval waktu untuk mendapatkan notifikasi secara *real-time* dan pemantauan dapat diakses jarak jauh.

2. LANDASAN TEORI

A. Penelitian Terdahulu

Membuat jurnal dengan judul Rancang Bangun Optimasi Penjadwalan Perawatan Mobil Pada @MayTrans di Politeknik Negeri Malang. Penelitian ini dilakukan untuk lebih meningkatkan kualitas proses bisnis perusahaan [1].

- Penerapan optimasi penjadwalan perawatan mobil di rental @MayTrans menggunakan bahasa php.
- Pada penelitian yang dilakukan Hersandi Anjar Susanto, Banni Satria Andoko, Nurudin Santoso dengan judul Rancang Bangun Optimasi Penjadwalan Perawatan Mobil Pada @MayTrans.

B. Perawatan

Perawatan atau *maintenance* adalah aktivitas agar suatu komponen atau sistem yang rusak dikembalikan atau diperbaiki dalam suatu kondisi tertentu pada periode tertentu. Telah diketahui bahwa peralatan atau mesin yang digunakan secara terus menerus meningkatkan laju kerusakan yang kemudian mempengaruhi tingkat ketersediaan peralatan atau mesin, dimana semakin tinggi laju kerusakan maka tingkat ketersediaan semakin rendah. Tujuan utama dilakukannya perawatan yaitu untuk mengupayakan agar peralatan atau mesin mampu dioperasikan secara terus menerus dalam jangka waktu tertentu sesuai dengan rencana tanpa mengalami kerusakan [2].

C. GPS Tracking

Sistem pelacakan kendaraan berbasis (*GPS based vehicle tracking and security system over*

GPRS). *GPS* sendiri merupakan sebuah sistem yang dapat memberitahu posisi sebuah kendaraan secara pasti. Sistem pelacakan ini menggunakan teknologi *GPS* untuk menentukan posisi kendaraan tersebut dan perangkat *GPRS* untuk berkomunikasi dengan server di Internet melalui jaringan nirkabel telepon seluler [3].

GPS Tracking dapat diartikan suatu sistem pemantauan jarak jauh yang menggunakan Satelit *GPS* sebagai penentu lokasi kendaraan/asset bergerak dengan tepat dan akurat dalam bentuk titik koordinat yang kemudian diimplementasikan ke dalam bentuk peta digital, sehingga dapat dimengerti dengan mudah bagi penggunanya.

D. Cara Kerja GT06N

Alat yang telah dikonfigurasi akan mengirimkan sebuah packet berupa string ke dalam server [4], misalnya:

Packet:

```
(012341234123BP05000012341234123140607
A3330.4288S07036.8518W019.2230104172.3
900000000L00019C2C)
```

Jadi, string dimulai dari

```
(" Device ID = "012341234123" Command =
"BP05" ,"login_request" Custom
Data="000012341234123140607A3330.4288S
07036.8518W019.
```

```
2230104172.3900000000L00019C2C" string
akhir = ")
```

Untuk mengambil latitude dan longitude dilakukan sebuah split dan konversi menjadi float, misalnya:

```
"latitude":functions.minute_to_decimal(parseF
loat(str.substr(7,9)),str.substr(16,1)),
"longitude":functions.minute_to_decimal(parse
Float(str.substr(17,9)),str.substr(27,1)),
```

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam pembangunan aplikasi Optimasi Penjadwalan Maintenance Kendaraan Menggunakan Alat GPS Tracking Berbasis Android adalah metode

terapan. Tahapan – tahapan dalam teori terapan dilakukan dalam kegiatan berikut :

1) Mengumpulkan acuan teori maupun referensi sebagai bahan menyusun teori dasar serta pendukung dalam proses pembangunan aplikasi optimasi penjadwalan *maintenance* kendaraan dari segi *hardware* dan *software*.

2) Melakukan perancangan sistem yang meliputi:

a) Menyiapkan kebutuhan *hardware* (perangkat keras) yang dibutuhkan dalam pendukung optimasi penjadwalan *maintenance* kendaraan.

b) Melakukan instalasi *software* (perangkat lunak) yang dibutuhkan dalam pembuatan program untuk alat *GPS Tracking*, pembuatan *server*, pembuatan admin panel, *REST API*, *database* dan *android*.

c) Melakukan pengemasan sistem.

d) Melakukan pengujian sistem.

3) Melakukan pembuatan sistem yang terdiri dari instalasi perangkat keras dan pembuatan perangkat lunak. Instalasi perangkat keras dilakukan dengan memasang alat *GPS Tracking* ke dalam kendaraan. Pembuatan perangkat lunak terdiri dari pembuatan *server online GPS Tracking*, pembuatan *database*, pembuatan admin panel, *REST API* dan tampilan pada *android*.

4) Melakukan pengujian sistem yang meliputi pengujian *server GPS Tracking* dan pengujian sistem yang terdiri dari pengujian akses *server* dan akses di dalam *android*.

5) Melakukan penyusunan laporan dan kesimpulan sebagai tahap terakhir setelah melakukan pengujian sistem..

E. Metode Pengembangan

Metode pengembangan yang digunakan dalam pembangunan aplikasi Optimasi Penjadwalan Maintenance Kendaraan Menggunakan Alat *GPS Tracking* Berbasis *Android* dilakukan menggunakan *Software Development Life Cycle (SDLC) Model Waterfall*. Tahapan – tahapan dalam *SDLC Model Waterfall* yang tersusun secara sistematis adalah sebagai berikut:

1) Requirement Analysis

Dalam tahapan analisis kebutuhan dilakukan analisa kebutuhan dari sistem mulai dari kebutuhan fungsional maupun kebutuhan non fungsional. Adapun analisa kebutuhan yang dilakukan sebagai berikut:

a) Kebutuhan Fungsional

Analisa kebutuhan fungsional mendeskripsikan jenis kebutuhan yang berisikan proses-proses apa saja yang diberikan oleh sistem kepada pengguna.

b) Kebutuhan Non Fungsional

Analisa kebutuhan non fungsional mendeskripsikan batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi.

2) Design

Tahap Desain merupakan tahapan dimana pengembang menjelaskan secara rinci yang akan dilakukan dan bagaimana sebuah sistem akan dibuat. Desain yang digunakan seperti Desain Arsitektur Sistem, *Flowchart*, *Use Case Diagram*, *Scenario Use Case*, *Activity Diagram*, *Sequence Diagram*, *Physical Data Modeling* dan *Class Diagram*.

3) Implementation

Tahap implementasi adalah tahap pembuatan keseluruhan sistem. Sistem yang akan dibuat didasarkan pada tahap desain, dan sesuai rancangan *flowchart* serta *usecase diagram*. Kemudian untuk implementasi bahasa yang digunakan untuk pembuatan *server* dan *REST API* menggunakan bahasa pemrograman *Node.JS*, lalu pembuatan web admin menggunakan bahasa pemrograman *Framework CodeIgniter* serta pembuatan aplikasi di *android* menggunakan bahasa pemrograman *Java* dan *database* yang digunakan *MySQL*.

4) Testing

Pada tahap uji yang dilakukan menggunakan pengujian black box testing. Pengujian pada

black box testing dalam penelitian ini untuk menemukan kesalahan seperti:

- Fungsi – fungsi yang tidak benar atau tidak ada.
- Kesalahan interface
- Kesalahan dalam struktur data atau akses basis data
- Kesalahan kinerja
- Kesalahan inisialisasi dan terminasi

5) Maintenance

Tahap *maintenance* dibutuhkan untuk menangani apabila ada terjadi masalah *error* kecil yang tidak ditemukan sebelumnya atau ada penambahan fitur yang belum ada pada sistem tersebut. Pengembangan diperlukan ketika adanya perubahan seperti ketika ada pergantian *server* atau perangkat lainnya.

4. HASIL DAN PEMBAHASAN

A. Perancangan Arsitektur Sistem



Gambar 1. Desain Arsitektur Sistem *Maintenance* Kendaraan

Pada Gambar 1. menjelaskan alur kinerja dari desain sistem bahwa satelit menangkap sinyal *GPS* yang menggunakan alat *GPS Tracking* yang telah terpasang di kendaraan, kemudian perangkat *GPS Tracking* mengirim, data dari perangkat tersebut dikirim melalui *GPRS* ke pusat data server, yang kemudian disimpan oleh database server. Hasil akhir data tersebut dapat dimonitoring oleh laptop maupun smartphone dan notifikasi pengingat jadwal servis akan diterima melalui *smartphone*.

Pada sistem penjadwalan *maintenance* kendaraan data diperoleh dari alat *GPS Tracking* berupa paket bertipe data *string* yang

kemudian dilakukan *split* untuk memisah data menjadi bagian – bagian yang akan dikonversi menjadi *JSON*. Setelah itu akan diambil data titik koordinat dan dilakukan sebuah perhitungan dari dua titik koordinat untuk menghasilkan jarak tempuh suatu kendaraan. Namun pada perhitungan jarak tempuh akan terjadi selisih hasil kilometer antara perhitungan melalui titik koordinat dengan keadaan nyata kilometer kendaraan. Oleh karena itu perlu dilakukan *recalibration* kilometer oleh admin. Selain pengingat jadwal *maintenance* dari kilometer, aplikasi ini juga memiliki fitur pengingat jadwal *maintenance* kendaraan berdasarkan interval waktu dan terdapat fitur pengaturan batas servis suku cadang dari masing-masing kendaraan yang dapat dikelola oleh pemilik atau pengelola rental.



Gambar 2. Desain Arsitektur Transportasi Data

Pada Gambar 2. menjelaskan alur kinerja komunikasi data dari aplikasi optimasi penjadwalan *maintenance* kendaraan. *SSH* digunakan untuk melakukan pembaruan sistem seperti *Web Admin Panel*, *REST API*, *Web Server* yang telah terinstal aplikasi git. Pengguna dapat mengakses *SSH* menggunakan nama *domain* disertai *public port* 49915.

Admin dan Pemilik untuk mengakses data yang berhubungan dengan database server dikomunikasikan melalui *REST API* menggunakan nama *domain server* disertai *public port* 49474. Kemudian komunikasi data *REST API* ke dalam server menggunakan *private port* 8090.

Kendaraan yang sudah terpasang GT06N dapat dikonfigurasi untuk mengirim packet data *GPS*

melalui jaringan internet menggunakan nama *domain server* disertai *public port* 49497. Kemudian *packet data* diteruskan ke *server* menggunakan *local port* 8080. Berikut adalah contoh *packet data* yang dikirim ke server

Packet:

```
(012341234123BP05000012341234123140607
A3330.4288S07036.8518W019.2230104172.3
900000000L00019C2C)
```

Jadi, *string* dimulai dari

```
(" Device ID = "012341234123" Command =
"BP05" → "login_request" Custom
Data="000012341234123140607A3330.4288S
07036.8518W019.
```

```
2230104172.3900000000L00019C2C" string
akhir = ")"
```

Untuk mengambil data tersebut dilakukan sebuah *split* dan konversi menjadi bilangan, misalnya:

```
Latitude =
parseFloat(str.substr(7,9)),str.substr(16,1)
```

```
Longitude = parseFloat(str.substr(17,9)
```

```
Speed = parseInt(str.substr(30, 2), 16)
```

Kemudian dari data tersebut diubah menjadi format *JSON* dan dikirim melalui *REST API* yang telah dibuat dengan alamat url dibawah ini:

<http://www.tracknema.com->

under.cloud.revoluz.io:8090/location_history

Parameter yang akan dikirim kedalam database yaitu *imei*, *latitude*, *longitude*, *speed*.

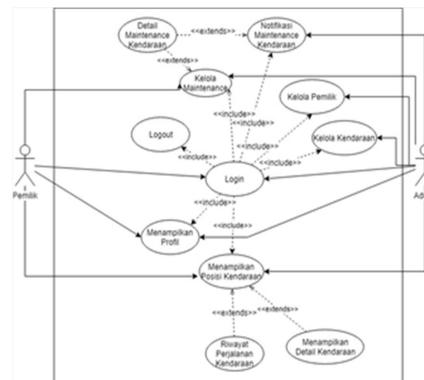
B. Perancangan Flowchart



Gambar 3. *Flowchart* Sistem *Maintenance* Kendaraan

Flowchart menjelaskan alur kerja dimulai dari melakukan cek jaringan selular jika gagal akan kembali melakukan cek kondisi jaringan selular dari alat *GPS Tracking*. Jika cek koneksi berhasil maka konfigurasi alat *GPS Tracking*. Kemudian data dikirim ke server, jika alat belum terdaftar maka kembali untuk melakukan konfigurasi alat *GPS tracking*. Jika cek data berhasil maka data dari alat *GPS Tracking* akan dilakukan perhitungan jarak tempuh dan mengirimkan notifikasi *maintenance*, lalu dilakukan validasi *speed* untuk mengurangi redundansi data sehingga dapat mengurangi resiko pada saat menampilkan *histori perjalanan kendaraan*, dan melakukan validasi jarak tempuh ini dilakukan untuk mengurangi resiko bila alat *gps tracking* telah mengalami masa usia atau rusak, kemudian dilakukan validasi apabila *odometer* atau tanggal sekarang melebihi batas maksimum maka server akan mengirim notifikasi, jika belum mencapai batas maksimum maka hanya menyimpan lokasi kendaraan.

C. Perancangan Use Case Diagram

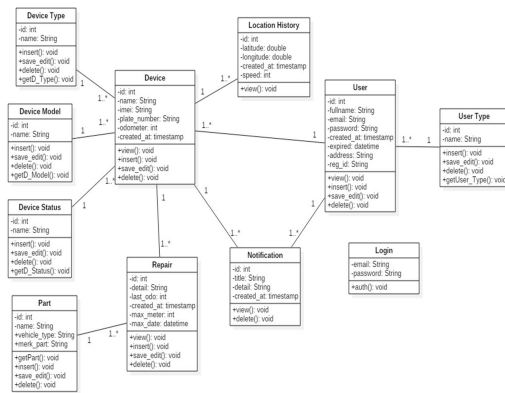


Gambar 4. *Usecase Diagram* Sistem *Maintenance* Kendaraan

Pada gambar 4. menggambarkan kegiatan – kegiatan yang akan di proses oleh sistem seperti *login*, *logout*, menampilkan profil, menampilkan posisi kendaraan, riwayat perjalanan kendaraan, menampilkan *detail*

kendaraan, kelola kendaraan, kelola pemilik, kelola maintenance, notifikasi *maintenance* kendaraan dan detail *maintenance* kendaraan.

D. Perancangan Class Diagram



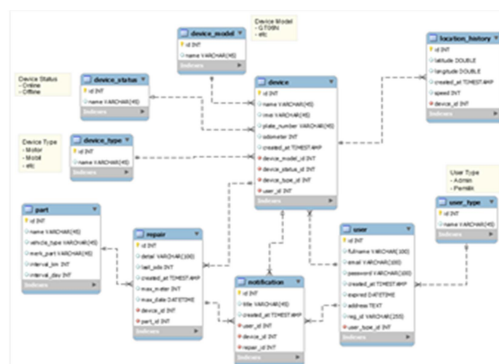
Gambar 5. Class Diagram Sistem Maintenance Kendaraan

Pada gambar 5. menunjukkan relasi antar kelas menggunakan *one to many*, menggunakan atribut *private* dan menggunakan operasi *public method*. *Method* yang digunakan pada setiap class memiliki fungsi tambah, ubah, hapus dan tampil.

E. Perancangan Physical Data Model (PDM)

Physical Data Model (PDM) merupakan model data fisik menggambarkan bagaimana model akan dibangun dalam database. Model database fisik menunjukkan semua struktur tabel, termasuk nama kolom, tipe data kolom, batasan kolom, kunci primer, kunci asing, dan hubungan antar tabel.

Berikut adalah PDM dari sistem aplikasi yang dirancang:



Gambar 6. PDM dari sistem aplikasi

Pada gambar 6. Merupakan model data fisik dari Aplikasi Optimasi Penjadwalan *Maintenance* Kendaraan. Berdasarkan gambar diatas dapat dijelaskan lebih rinci sebagai berikut:

5.1 Tabel *device_model*

Merupakan tabel yang digunakan untuk menyimpan jenis perangkat dari alat *gps tracking* seperti GT06N, TK100, dsb.

5.2 Tabel *device_status*

Merupakan tabel yang digunakan untuk menyimpan keadaan perangkat dari alat *gps tracking* seperti keadaan dalam status online atau offline.

5.3 Tabel *device_type*

Merupakan tabel yang digunakan untuk menyimpan jenis kendaraan seperti mobil, motor, dsb.

5.4 Tabel *location_history*

Merupakan tabel yang digunakan untuk keperluan menyimpan data lokasi dari alat *gps tracking* yang sedang mengirim. Dan menampilkan lokasi terakhir serta riwayat kendaraan.

5.5 Tabel *device*

Merupakan tabel yang digunakan untuk menyimpan informasi detail kendaraan seperti *odometer*, *imei*, *plat number*, dsb.

5.6 Tabel *user_type*

Merupakan tabel yang digunakan untuk menyimpan jenis hak akses seperti admin atau pemilik.

5.7 Tabel *user*

Merupakan tabel yang menyimpan data pengguna untuk keperluan login dan hak akses.

5.8 Tabel *part*

Merupakan tabel yang menyimpan data suku cadang untuk keperluan servis sebagai batas maksimum kendaraan.

5.9 Tabel *repair*

Merupakan tabel yang digunakan untuk menyimpan data repair sebagai pencatatan terakhir melakukan servis.

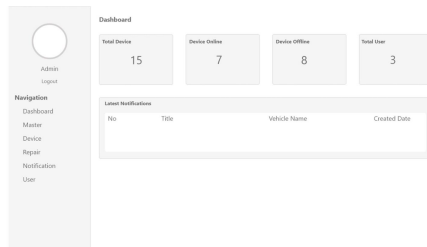
5.10 Tabel *notification*

Merupakan tabel yang digunakan untuk menyimpan data notifikasi *maintenance* kendaraan.

F. Desain Mockup

Desain *mockup* merupakan suatu media komunikasi user dengan aplikasi yang dibangun. Oleh karena itu, dalam perancangan sebuah aplikasi dibutuhkan suatu perancangan antarmuka guna memberikan gambaran umum bagi user. Perancangan antarmuka aplikasi ini dapat dilihat pada beberapa gambar berikut:

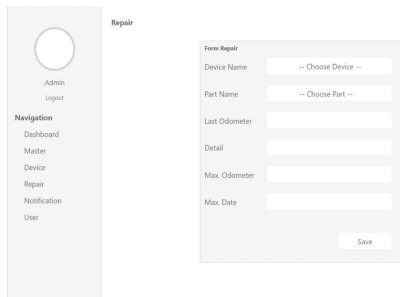
1) Halaman *Dashboard*



Gambar 7. Mockup Halaman *Dashboard*

Pada gambar 7. menampilkan halaman awal yang berisi tentang informasi notifikasi terbaru dari *maintenance* kendaraan dan menampilkan jumlah total semua *device*, total *device* yang sedang aktif, total *device* yang sedang tidak aktif dan total user.

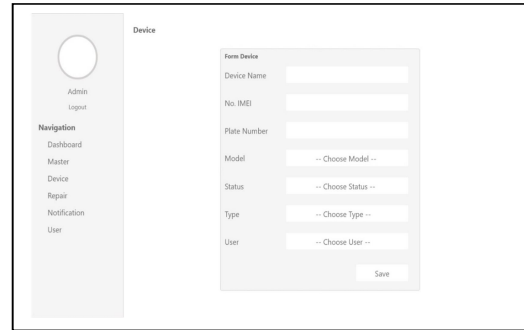
2) Halaman *Form Repair*



Gambar 8. Mockup Halaman *Form Repair*

Pada gambar 8. Menampilkan *form repair* yang berfungsi sebagai memasukan data *maintenance* kendaraan yang bertujuan untuk membuat jadwal servis kendaraan sehingga akan mendapatkan notifikasi jika melebihi batas maksimum.

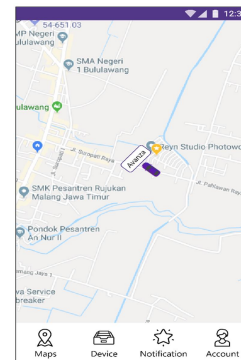
3) Halaman *Form Device*



Gambar 9. Mockup Halaman *Form Device*

Pada gambar 9. menampilkan *form device* yang berfungsi sebagai mendaftarkan kendaraan yang telah terpasang alat *gps tracking* agar lokasi kendaraan terekam kedalam riwayat lokasi kendaraan.

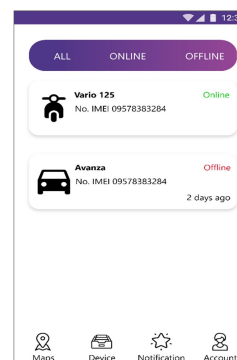
4) Menu *Maps*



Gambar 10. Mockup Menu *Maps*

Pada gambar 10. menampilkan *marker* kendaraan dengan menggunakan google maps yang sebelumnya *device* telah terdaftar oleh sistem.

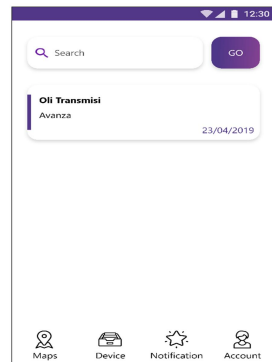
5) Menu *Daftar Device*



Gambar 11. Mockup Menu *Device*

Pada gambar 11. menampilkan daftar device menggunakan filter berdasarkan semua device, device aktif, device tidak aktif.

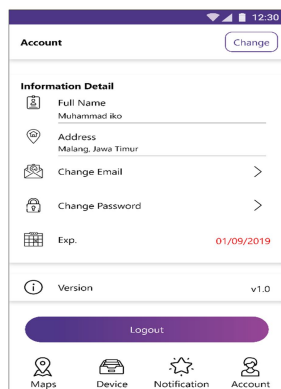
6) Menu Notifikasi



Gambar 12. *Mockup Menu Notifikasi*

Gambar 12. menampilkan daftar notifikasi maintenance kendaraan serta fitur pencarian notifikasi. Klik pada bagian daftar notifikasi untuk melihat *detail maintenance*.

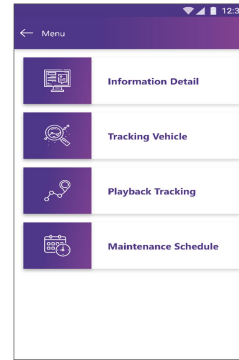
7) Menu Account



Gambar 13. *Mockup Menu Account*

Pada gambar 13. menampilkan informasi akun pengguna digunakan untuk mengubah data diri pengguna serta tombol logout kendaraan.

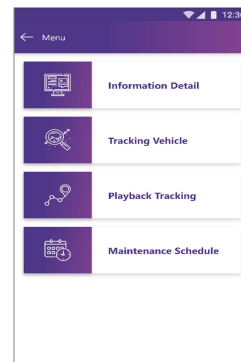
8) Menu *Fitur Device*



Gambar 14. *Mockup Menu Fitur Device*

Pada gambar 14. menampilkan daftar fitur dari *device* meliputi informasi *detail device*, *tracking detail*, riwayat perjalanan, dan penjadwalan *maintenance*.

9) Menu *Detail Maintenance*



Gambar 15. *Mockup Menu Detail Maintenance*

Gambar 15. menampilkan informasi berisi tentang detail kendaraan dan device dari alat gps tracking tersebut.

G. Implementasi Arsitektur Sistem

1) Pemasangan Alat GPS Tracker GT06N



Gambar 16. Dokumentasi Alat GT06N yang terpasang di motor Vario 125

Pada gambar 16. Merupakan gambar dari alat *GT06N* yang telah terpasang di kendaraan motor vario 125. Alat tersebut diletakkan disamping tangki motor, agar memudahkan untuk melihat indikator lampu dari alat tersebut. Dan juga lebih aman ketika cuaca hujan. Alat tersebut terhubung dengan daya aki sehingga dapat mengisi disaat baterai dari alat tersebut habis.

2) Konfigurasi GT06N

Agar terhubung dengan *server* alat tersebut harus dikonfigurasi terlebih dahulu. Berikut langkah – langkah konfigurasi alat GT06N:

- Siapkan kartu SIM Selular yang telah terisi paket data maupun pulsa secukupnya.
- Masukkan kartu SIM Selular ke dalam slot sim card alat GT06N.
- Setelah terpasang, gunakanlah HP ketik pesan dengan format dibawah ini:
Penerima: Nomer HP dari Alat GT06N
Isi Pesan:
SERVER, 1, www.tracknema.com-under.cloud.revoluz.io,49497,0#
Setelah itu alat akan membalas OK.
- Kemudian untuk mengecek apakah konfigurasi telah berhasil dapat mengirimkan pesan kembali
Penerima: Nomer HP dari Alat GT06N
Isi Pesan: SERVER#
- Hasil konfigurasi alat GT06N

H. Implementasi *Server GPS Tracker*

Tahapan implementasi server dilakukan parsing data dari alat kemudian data diolah menjadi sebuah *JSON*, lalu di kirim melalui *REST API* menggunakan *XHR*.

Berikut tahapan untuk melakukan parsing data:

- 1) Packet data berupa hexadecimal.
130604102905c800dde0210c15f3c000411d09000000000000000240600010200004f960d0a
- 2) Proses parsing packet data hexadecimal.
date = str.substr(0, 12)
latitude_raw = str.substr(14, 8)
longitude_raw = str.substr(22, 8)

speed = parseInt(str.substr(30, 2), 16)

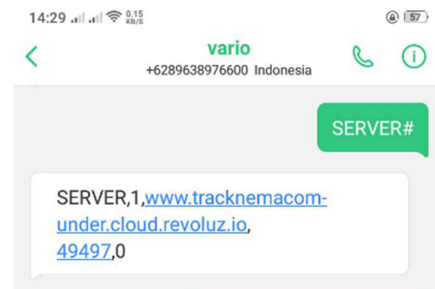
- 3) Konversi hexadecimal menjadi koordinat dengan mengubah hasil latitude_raw dan longitude_raw menjadi bilangan decimal. Kemudian hasil decimal tersebut dibagi dengan 1800000.

Latitude = parseInt(00dde021,16) / 1800000.

Longitude = parseInt(0c15f3c0,16) / 1800000.

- 4) Setelah memperoleh hasil parsing data, selanjutnya mengubah data dalam bentuk JSON

- 5) Langkah terakhir, proses pengiriman data ke dalam database menggunakan *XHR* menggunakan request post.



Gambar 17. Dokumentasi Konfigurasi ke *server GPS Tracker*

I. Implementasi *REST API*

REST API digunakan untuk menyimpan data lokasi dari alat GPS tracker, melakukan perhitungan untuk menentukan jarak tempuh kendaraan serta mengirimkan notifikasi kendaraan jika melebihi batas maksimum dari perawatan yang telah ditentukan. Berikut ini adalah *source code* dari *REST API* melakukan perhitungan jarak berdasarkan *latitude* dan *longitude*.

J. Pengujian Sistem

Pada tahapan pengujian adalah proses untuk mengetahui cara kerja dari perangkat dan menganalisa keakuratan serta kelemahan dari aplikasi dan perangkat yang digunakan. Selain itu pada tahapan pengujian berfungsi untuk mengetahui tentang cara penggunaan aplikasi agar dapat digunakan secara optimal.

Hasil rancangan sistem yang telah dibangun berupa perangkat yang siap pakai. Perangkat ditempatkan dibawah dudukan kendaraan sepeda motor atau dibawah *dashboard* mobil. Untuk pengujian perangkat ini, terfokus pada akurasi sistem dalam menentukan nilai jarak tempuh. Selain itu fokus terhadap akurasi karena akan sangat berpengaruh pada notifikasi yang diterima oleh pengguna aplikasi. Dan juga pengujian ini juga memperhatikan hasil komunikasi data antara perangkat dan server. Karena tahap pendistribusian data merupakan satu komponen terpenting dalam hal akurasi sistem secara keseluruhan. Dimana data GPS yang diterima harus segera diproses oleh alat *GT06N* dan dikirim melalui komunikasi *GSM GPRS*. Sehingga informasi posisi kendaraan dan jarak tempuh kendaraan dapat ditampilkan kepada pengguna dalam bentuk peta yang berbasis *mobile android*.

K. Pengujian Akurasi

Pengujian akurasi adalah pengujian yang dilakukan untuk menguji tingkat keakuratan atau kesesuaian dari data testing yang didapatkan dari pakar dengan keluaran sistem. Nilai akurasi didapatkan dari perhitungan persamaan.

Pengujian ini juga memperhitungkan persentase kesalahan sistem berdasarkan persamaan berikut.

$$\frac{\text{jarak aktual} - \text{jarak tempuh}}{\text{jarak aktual}} \times 100\%$$

Jarak aktual merupakan hasil yang didapat dari odometer kendaraan. Sedangkan jarak tempuh merupakan hasil pengukuran jarak tempuh kendaraan menggunakan perangkat yang telah dibangun dan ditempatkan di kendaraan.

L. Pengujian Blackbox

Pengujian *blackbox* berfungsi untuk mengetahui apakah sistem yang telah dibangun dapat berjalan berdasarkan kebutuhan. Penulis menggunakan *blackbox* karena pengujian *blackbox* dapat melatih semua kebutuhan

fungsi suatu program dan dapat menemukan suatu kesalahan lainnya selain menggunakan metode *whitebox*. Keberhasilan dalam pengujian ditunjukkan kecocokan antara hasil yang diharapkan dan hasil yang didapatkan dari sistem. Hasil dari pengujian *blackbox* yang telah dilakukan berdasarkan daftar kebutuhan sistem ditunjukkan pada tabel 1.

Tabel 1. Hasil Pengujian Jarak Tempuh Interval 5 Detik

No	Jarak Aktual (meter)	Jarak Tempuh (meter)	Persentase Kesalahan
1	1000	967	3,3%
2		924	7,6%
3		967	3,3%
4		926	7,4%
5		998	0,2%
6	3000	2746	8,4%
7		2988	0,4%
8		2683	4,6%
9		2783	7,2%
10		2864	4,5%
11	5000	4829	3,4%
12		4861	2,8%
13		4198	4,2%
14		4802	4,0%
15		4701	6,0%

Sementara itu, dilakukan pengujian dengan konfigurasi interval 10 detik untuk membandingkan hasil pengukuran jarak tempuh memperoleh data. Adapun hasil pengujian dapat dilihat pada tabel 2.

Tabel 2. Hasil Pengujian Jarak Tempuh Interval 10 Detik

No	Jarak Aktual (meter)	Jarak Tempuh (meter)	Persentase Kesalahan
1	1000	992	0,8%
2		929	7,1%
3		953	4,7%
4		916	8,4%
5		948	5,2%
6	3000	2920	2,7%
7		2990	0,3%
8		2613	12,9%
9		2833	5,7%
10		2794	6,8%
11	5000	4899	2,0%
12		4831	3,4%
13		4288	14,2%
14		4772	4,6%
15		4901	2,0%

Setiap pengujian dapat berjalan dengan stabil, mulai dari perhitungan jarak tempuh pada *gps tracker* sampai pengiriman data ke *server*. Begitu juga dalam menentukan posisi kendaraan, sistem mampu mengkoordinasi lokasi kendaraan sesuai dengan lokasi sebenarnya. Hal ini menunjukkan bahwa konfigurasi pada alat *gps tracker GT06N*, dapat bekerja dengan baik.

Akan tetapi adanya sedikit rata-rata persentase kesalahan yang disebabkan karena adanya delay waktu pengiriman data, serta jika konfigurasi alat diatur lebih besar delay akan mempengaruhi hasil dari proses perhitungan jarak tempuh seperti faktor jalan yang berbelok – belok. Semakin besar delay pada komunikasi data akan berdampak terhadap nilai jarak tempuh semakin tidak sesuai dengan jarak aslinya. Selain itu, hal ini disebabkan oleh beberapa faktor diantaranya lokasi, cuaca dan jaringan data *GPRS* disepanjang rute jalan yang merupakan lokasi pengambilan data.

M. Pembahasan

Setiap pengujian dapat berjalan dengan stabil, mulai dari perhitungan jarak tempuh

pada perangkat *GT06N* sampai pengirim data oleh perangkat ke server. Begitu juga dalam menentukan posisi kendaraan, sistem mampu mengkoordinasi lokasi kendaraan sesuai dengan lokasi sebenarnya. Hal ini menunjukkan bahwa konfigurasi pada server penerima data dari perangkat *GT06N* dan layanan data *GPRS* dapat bekerja dengan baik.

Rata-rata persentase kesalahan pengujian pada jarak 1000 meter, 3000 meter, dan 5000 meter tidak jauh berbeda. Secara umum, hasil pengujian pengiriman data data dari perangkat *GT06N* dapat diterima dengan baik oleh server. Kemudian dikirim melalui *REST API* dan disimpan kedalam database. Data tersebut selanjutnya ditampilkan pada aplikasi android yang sudah dibangun. Hal ini menunjukkan bahwa proses perhitungan jarak tempuh kendaraan pada *GT06N* berdasarkan data kecepatan dan waktu dapat berkerja dengan baik.

Akan tetapi adanya sedikit selisih rata-rata persentase kesalahan yang disebabkan karena adanya delay waktu dalam komunikasi data pada sistem. Hal ini disebabkan karena keterbatasan perangkat yang hanya dapat mengirimkan data minimal 5 detik. Semakin besar delay pada komunikasi data maka akan berdampak terhadap nilai jarak tempuh semakin tidak sesuai dengan jarak aslinya. Selain itu, hal ini disebabkan karena beberapa faktor lokasi, cuaca, dan jaringan data *GPRS* disepanjang rute jalan yang merupakan lokasi pengambilan data.

5. KESIMPULAN

Berdasarkan hasil pembahasan optimasi penjadwalan maintenance kendaraan ini dapat diambil adalah sebagai berikut:

Berdasarkan penelitian yang telah dilakukan oleh penulis mengenai “Optimasi Penjadwalan Maintenance Kendaraan Menggunakan Alat GPS Tracking berbasis Android” dapat ditarik kesimpulan sebagai berikut:

- 1) Sistem dapat mencatat odometer dari kendaraan yang telah terpasang perangkat *GT06N* serta terdaftar oleh sistem.

- 2) Sistem dapat mengirimkan notifikasi secara *real-time* kepada admin dan pemilik melalui aplikasi *android* untuk mengingatkan bahwa saatnya untuk melakukan *maintenance* kendaraan.
 - 3) Sistem dapat melakukan *parsing* data lokasi kendaraan yang kemudian disimpan ke dalam *database* dari perangkat yang dikirim oleh *GT06N*.
 - 4) Jika koneksi internet tidak stabil maka pengiriman data ke *server* akan lama atau gagal.
 - 5) Konfigurasi untuk mendapatkan lokasi pengiriman data yang optimal pada perangkat *GT06N* dengan mengatur 5 detik – 10 detik.
 - 6) Menggunakan perangkat *GT06N* dengan konfigurasi waktu pengiriman data 5 detik mampu memberikan hasil yang diharapkan dengan tingkat akurasi sebesar 95,51 % pada 15 data training yang telah dihitung berdasarkan lokasi *latitude* dan *longitude*. Sedangkan konfigurasi waktu pengiriman data 10 detik mampu memberikan hasil yang diharapkan dengan tingkat akurasi sebesar 94,61 % pada 15 data training. Oleh karena itu semakin besar nilai konfigurasi pengiriman data, maka tingkat persentase kesalahan juga akan semakin besar.
- Dengan adanya optimasi *maintenance* kendaraan dapat mempermudah dalam pengelola kendaraan memantau melakukan perawatan kendaraan rental serta dapat mengetahui kendaraan harus melakukan perawatan secara *realtime* dan dengan adanya alat *gps tracking* yang terpasang dalam kendaraan sangat membantu untuk memantau kendaraan dari posisi jarak jauh.

6. DAFTAR PUSTAKA

- [1] Santana, R. A., Risqiwati, D., & Sari, Z. (2017). Rancang Bangun Sistem Informasi Servis Oli Sepeda Motor Menggunakan Odometer Berbasis LBS. *Kinetik*, 2(1), 17. <https://doi.org/10.22219/kinetik.v2i1.98>
- [2] R. Neneng Khoirunnisa, Kusmaningrum, F. H. M. (2015). Penjadwalan Perawatan Pencegahan Komponen Kopling dan Rem Pada Mobil Pancar Di Dinas Pencegahan Dan Penanggulangan Kebakaran Kota Bandung, 03(01), 212–223.
- [3] Susanto, H. A., Andoko, & B. S., Santoso, N.(2015). Rancang Bangun Optimasi Penjadwalan Perawatan Mobil Pada @ MayTrans, 29–33.
- [4] Spirito. G.D, 2017,GPS Tracking Server, <https://github.com/freshworkstudio /gps-tracking-nodejs>, (di akses tanggal 10 Desember 2018).
- [5] Ayuninghemi, R., Abdillah, M. F.(2018). Simulasi Alat Notifikasi Servis Ringan Sepeda Motor, 4, 168–173.
- [6] Shenzhen. Co., L. (n.d.). User Manual GPS Vehicle tracker GT06N.
- [7] Shenzhen. Co., L. (n.d.). Buku Manual GT06N
- [8] Syahril, Rizal (2016). Pemantauan Jarak Tempuh Kendaraan Menggunakan Modul General Packet Radio Service (Gprs), Global Positioning System (Gps) Dan Arduino
- [9] Patel, K. G., Patel, H., Gupta, P., & Gamit, R. (2017). Implement Vehicle Management System Using Android, 3(3), 12–15.
- [10] Mufti, Joseph. (2015). Easy guide to Development of Google Map Android.