| Vol. 1 | No. 1 | Halaman 51 – 65
[Maret] [2020] JASEE
e-ISSN : 2721-320X
p-ISSN : 2721-3625

SISTEM PEMANTAUAN DAN PENGINGAT WAKTU PERAWATAN KENDARAAN BERBASIS INTERNET OF THINGS (IOT)

Muchsin Attubel[1], Diky Siswanto[2], Mohammad Mukhsim[3]
[1,2,3] Fakultas Teknik, Jurusan Teknik Elektro, Universitas Widyagama Malang.
Email: sintubleee@gmail.com

Abstrak

Meningkatnya kebutuhan akan informasi, pemantauan, dan pengendalian jarak jauh terhadap Peralatan, industru, otomotif, dan bahkan kota. Hal tersebut membuat teknologi *Internet of Things* semakin dibutuhkan dan menjadi tantangan bagi penulis untuk merancang dan mengimplementasikan *Internet of Things* pada berbagai bidang. Untuk menjawab tantangan tersebut penulis akan menerapkan *Internet of things* pada kendaraan, khususnya sepeda motor, alat yang akan dirancang memiliki sistem yang menggunakan module GPS, module GSM, dan modul ACS-712 sebagai sensor arus, ketiga module tersebut akan dihubungkan ke Arduino sebagai *microcontroller*, setiap kali kendaraan hidup module ACS-712 akan mendeteksi adanya arus dari Alternator yang menandakan mesin sedang menyala, lama mesin menyala akan diakumulasi sebagai waktu ganti oli. Sedangkan module GPS akan mendeteksi lokasi kendaraan secara *realtime. Database* dan web bertugas untuk menyediakan data dan tampilan *Maps* untuk dikirim ke App android, sedangkan App android berfungsi sebagai *user interface* yaitu untuk menampilkan indikator mesin On/Off, indikator waktu ganti oli, dan menampilkan *maps* untuk memantau lokasi kendaraan.

Kata kunci: IOT, Android, *Maps*, Web Server, Database

Abstract

Increased needs for information, monitoring and remote control of equipment, industry, automotive, and even cities. This makes Internet of Things technology increasingly needed and becomes a challenge for writers to design and implement the Internet of Things in various fields. To answer this challenge the author will implement the Internet of things on vehicles, especially motorcycles, the tools that will be designed have a system that uses a GPS module, GSM module, and ACS-712 module as a current sensor, the three modules will be connected to Arduino as a microcontroller. Every time the vehicle is alive the ACS-712 module will detect the current from the Alternator which indicates the engine is running, the engine will run for a long time as an oil change time. While the GPS module will detect the location of the vehicle in real time. The formulation of the problem in this study is how to design and implement the Internet of Things on vehicles, with an android-based user interface, a microcontroller using Arduino, and Sim800L as a link to a web server.

Keywords: IOT, Android, *Maps*, Web Server, Database

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

1. PENDAHULUAN

Perkembangan teknologi di bidang Internet *of Things* yang didefinisikan sebagai segala teknologi yang memperbolehkan jaringan computer untuk berkomunikasi dengan perangkat keras lainnya melalui jaringan internet (termasuk juga perangkat keras komputer). Jaringan komputer dalam hal ini meliputi *wired* (kabel), *wireless* (nirkabel), Peer *To* Peer (P2Po, *Cloud Computing*, Bluethooth, Sensor *Network*, *Wireles* sensor *network* (WNS) dan lain-lain. Dengan adanya Internet *of Things* ini, bukan hal yang mustahil lagi bagi anda untuk bias mengendalikan rumah anda dari jarak jauh, membuat sensor kebakaran/banjir, mengendalikan pesawat terbang, mengetahui kedatangan musuh, dan lain-lain, baik pada kehidupan sehari-hari maupun militer dan perang [1]. IoT (Internet *of Things*) secara umum ialah suatu konsep yang dapat menghubungkan benda-benda di sekitar kita dengan jaringan internet yang membuatnya bisa berkomunikasi antar benda maupun dengan penggunanya. Sehingga konsep ini akan memudahkan penggunanya jika diterapkan di kehidupan nyata [2]. Selain mempermudah dan mengurangi human *error*, bagaimana memanfaatkan teknologi IoT untuk meningkatkan keamanan rumah.

Salah satu bagian penting yang harus di perhatikan dalam perawatan kendaraan bermotor adalah waktu yang tepat untuk mengganti oli, yang mana hal tersebut ditentukan dari perhitunggan jarak tempuh kendaraan namun karena sering terjadinya kelupaan oleh pengguna, dari permasalahan tersebutlah penulis terinspirasi untuk merancang sistem prediksi waktu perawatan dan pengingat.

Sementara itu, saat ini penggunaan smartphone terus meningkat bahkan dalam Laporan Internet Trends Kleiner Perkins Caufield & Byers's tersebut bahkan menyebutkan angka yang tinggi, yaitu pengguna rata-rata mengecek ponselnya 150 kali dalam sehari. Jika diakumulasi, dalam satu minggu ratarata orang bisa menggunakan smartphonenya lebih dari 1.050 kali [3]. Maka akan menjadi sangat tepat jika Sistem Monitoring Dan Sistem Pengingat Waktu Perawatan kendaraan diterapkan pada *Smarphone* sebagai *user interface.*

Dalam kasus ini penulis membandingkan dengan jurnal [4] PENGEMBANGAN SISTEM PERINGATAN GANTI OLI PADA SEPEDA MOTOR pada penelitian tersebut media pengingat yang digunakan berupa LCD yang menampilkan pesan peringatan dan lampu indikator yang diletakkan pada *dashboard* motor, hal tersebut kurang efektif dikarenakan jarangnya pengguna memperhatikan dan mengecek tampilan pada *dashboard* motor.

Terlambatnya melakukan perawatan pada kendaraan bermotor dapat mengakibatkan menurunnya kinerja kendaraan bahkan bisa mengalami kerusakan. Sehingga Internet *of thing* adalah solusi tepat untuk memonitoring kendaraan dan juga sebagai sistem pengingat waktu perawatan. Jika melakukan perawatan tepat waktu tentu mencegah terjadinya kerusakan yang tidak diinginkan dan menghemat biaya perbaikan.

2. STUDI PUSTAKA

2.1 Internet *of Things*

Internet *of Thing* adalah teknologi yang memungkinkan manusia dapat mengontrol dan memantau Obyek dari jarak jauh. Tantangan utama dalam IOT adalah menjembatani kesenjangan antara dunia fisik dan dunia informasi. Misalnya mengolah data yang diperoleh dari peralatan eletronik melakui sebuah interface antara pengguna dan peralatan itu. Sensor mengumpulkan data mentah fisik dari skenario *real time* dan mengkonversikan ke dalam mesin format yang

https://doi.org/10.31328/jasee.v1i01.6

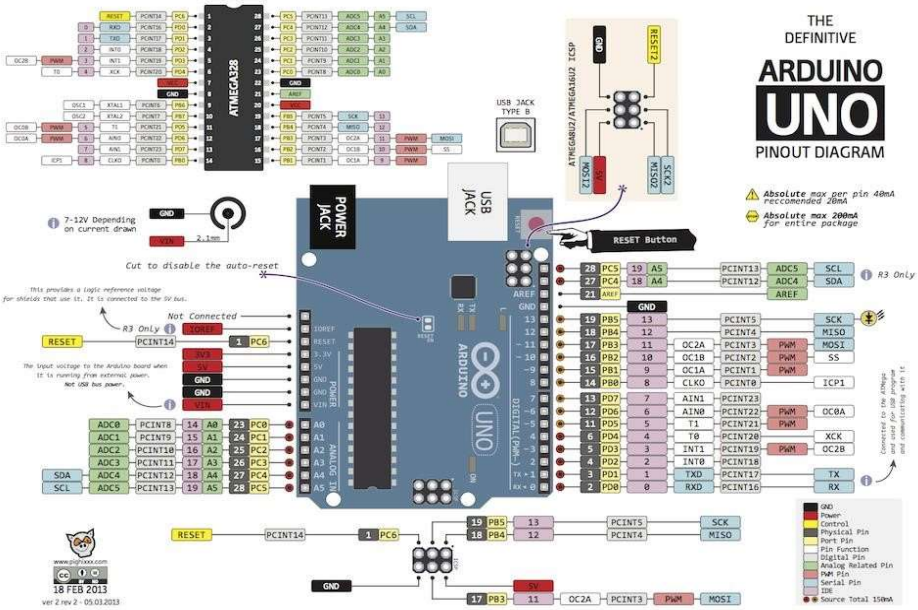
e-ISSN : 2721-320X
p-ISSN : 2721-3625

dimengerti sehingga akan mudah dipertukarkan antara berbagai bentuk format data [5]

2.2 Microcontroller

2.2.1 Arduino Uno

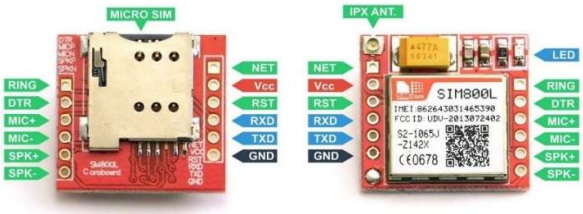
Arduino Uno adalah sebuah rangkaian yang dikembangkan dari *microcontroller* berbasis ATmega328. Arduino Uno memiliki 14 kaki digital *input / output*, dimana 6 kaki digital diantaranya dapat digunakan sebagai sinyal PWM (*Pulse Width Modulation*). Sinyal PWM berfungsi untuk mengatur kecepatan perputaran motor. Arduino Uno memiliki 6 kaki analog input, kristal osilator dengan kecepatan jam 16 MHz, sebuah koneksi USB, sebuah konektor listrik, sebuah kaki header dari ICSP, dan sebuah tombol reset yang berfungsi untuk mengulang program [6].



Gambar 2.1. Arduino Uno

2.3 Modul GSM SIM800L

SIM800L adalah modul *quad band* GSM/GPRS yang bekerja pada frekuensi GSM 850MHz, EGSM 900MHz, DCS 1800MHz dan PCS 1900 MHz. Sim800L memiliki fitur GPRS *multi slot class* 12 / *class* 10 (Optional) dan mendukung skema koding GPRS CS-1, CS-2, CS-3, dan CS-4 [7].



Gambar 2.2. Module GSM SIM800L

2.4 *Global Positioning Sistem* (GPS)

Global Positioning Sistem (GPS) adalah suatu sistem radio navigasi penentuan posisi menggunakan satelit. GPS dapat memberikan posisi suatu objek di muka bumi dengan akurat dan cepat (koordinat tiga dimensi x, y, z) dan memberikan informasi waktu serta kecepatan bergerak secara kontinyu di seluruh dunia [8].

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

Gambar. 2.3. GPS NEO-6M V2

2.5 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan Aplikasi. Android menyediakan *platfirm* terbuka bagi para pengembang untuk menciptakan aplikasi mereka [9].

Sebagai Open Source dan bebas dalam memodifikasi, di dalam Android tidak ada ketentuan yang tetap dalam konfigurasi *Software* dan Hardware. Fitur- fitur yang didapat dalam Android antara lain [8].

2.6 Alternator

Fungsi *alternator* (*Generator/Spool*) adalah mengubah energi mekanis yang didapatkan dari mesin dalam bentuk tenaga listrik. Energi mekanik dari mesin yang disalurkan oleh sebuah puli, yang memutarkan roda dan menghasilkan arus listrik bolak-balik pada stator. Arus listrik bolakbalik ini kemudian dirubah menjadi arus searah oleh dioda [10]. Arus Searah yang dihasilkan oleh *alternator* ini akan dipasangkan sensor arus untuk mengetahui bahwa mesin sedang menyala, akumulasi dari lama mesin menyala inilah yang akan dimanfaatkan untuk memprediksi waktu perawatan kendaraan.

2.7 Web Server

Web server adalah *software* yang menjadi tulang belakang dari *world wide web* (*www*) yang pertama kali tercipta sekitar tahun 1980an. Web server menunggu permintaan dari client yang menggunakan browser seperti Internet Explorer, Mozilla Firefox, dan program browser lainnya. Jika ada permintaan dari browser, maka web server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Data ini mempunyai format yang standar, disebut dengan format SGML (Standar General Markup Language) [10].

2.8 MySQL Database

MySQL merupakan sebuah *database server* yang mampu untuk memanajemen *database* dengan baik, MySQL merupakan database yang paling digemari dan paling banyak digunakan disbanding *database* lainnya MySQL merupakan *software database* yang bersifat *free* karena MySQL dilisensi dibawah GPL (*General Public License*). MySQL memiliki *query* yang telah distandarkan oleh ANSI/ISO yaitu menggunakan Bahasa SQL.

2.9 Google Maps dan Google Maps API

Google Maps memberikan sebuah jasa peta *globe virtual* gratis dan *online* dengan meneyediakan peta dan gambar setelit yang didapat diintegrasikan didalam sistem yang sebelumnya telah terdaftar [11].

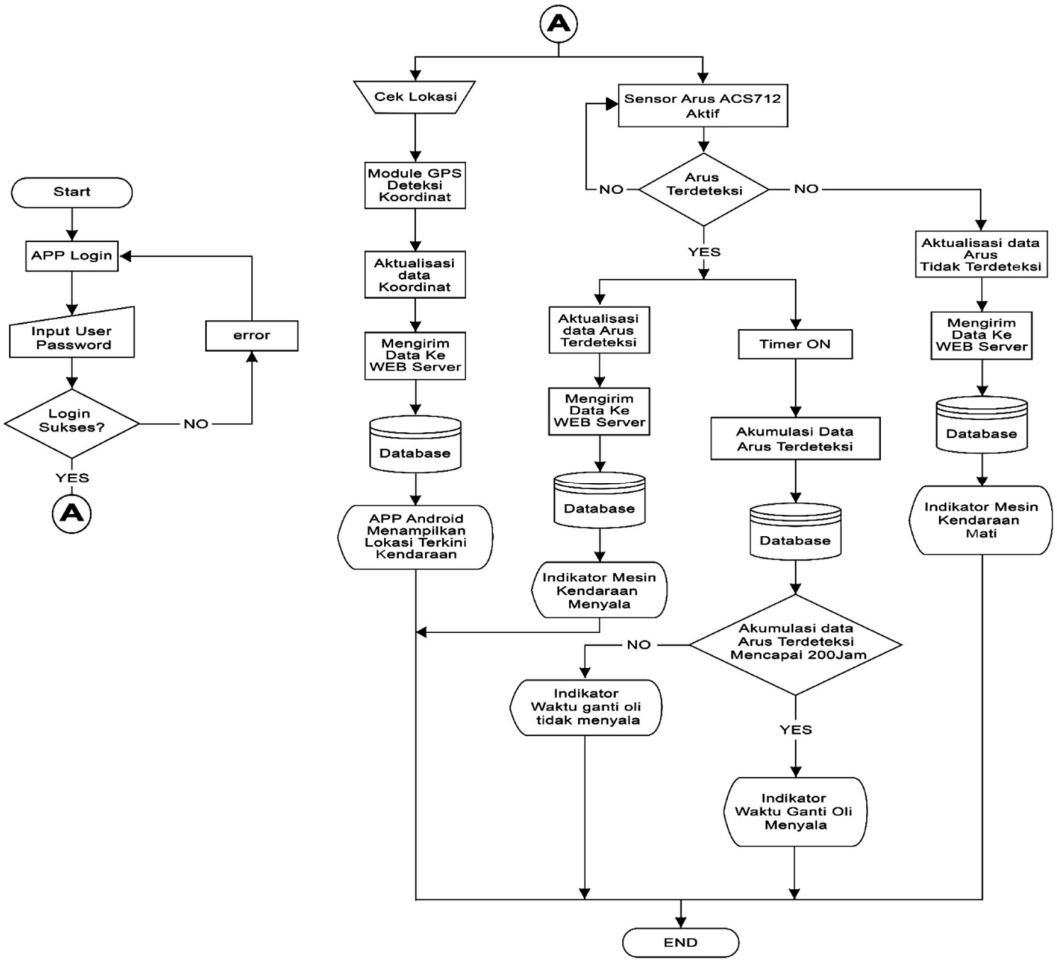
3. METODE

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

3.1 Diagram Alir

Perancangan sistem *monitoring* dan sistem pengingat waktu perawatan kendaraan ini mengikuti diagram alir (*flowchart*) berikut.



Gambar. 3.1. Bagan Alir (*Flowchart*) Sistem Monitoring Dan Sistem Pengingat Waktu Perawatan Kendaraan

Secara detail, bagan alir sistem *monitoring* dan sistem pengingat waktu perawatan kendaraan dideskripsikan sebagai berikut :

1. Proses pertama dimulai dengan *user* membuka aplikasi dan *login* dengan menginput text Username dan Password, jika username dan password sesuai yang terdaftar di Database maka menu utama akan tampil, Jika *password* dan *username* salah maka akan muncul pesan *error* dan kembali ke menu *login*.
2. Pada menu utama jika tombol cek lokasi ditekan maka GPS akan mendeteksi lokasi/koordinat kendaraan lalu dikirin ke *website*, setelah mendapat koordinat kendaraan lalu ditampilkan dalam bentuk *Maps* Google dengan pin pada lokasi terkini.
3. Pada menu utama bagian indikator mesin menyala atau mati bergantung pada sensor arus yang bekerja secara *real time*, sensor arus ini terhubung ke *output* dari *Alternator* kendaraan jika mesin menyala maka arus akan terdeteksi dan data akan dikirim ke *website* untuk selanjutnya ditampilkan sebagai kondisi mensin menyala pada *web browser* maupun APP Android, jika arus tidak terdeteksi akan ditampilkan indikator mesin mati.

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

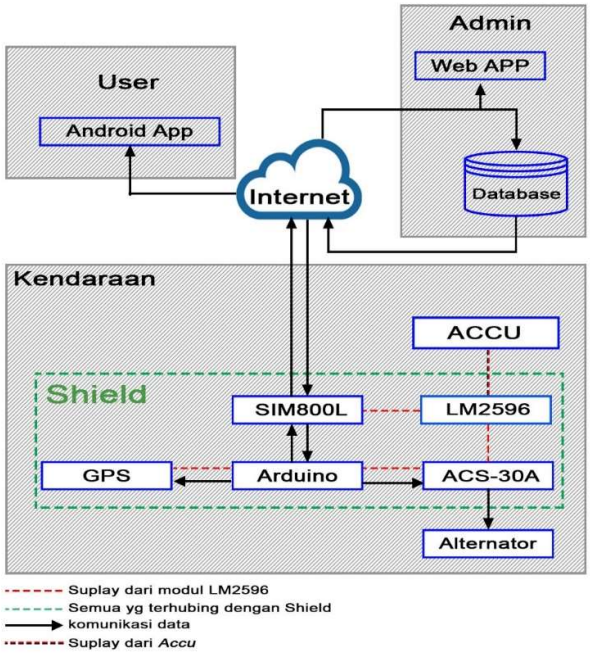
4. Pada indikator ganti oli bergantung pada akumulasi data arus terdeteksi(mesin menyala) yang dikumpulkan dari *microcontroller* lalu dikirim ke *database server* untuk di simpan, jika akumulasi data pada *database* mencapai 200jam maka indikator waktu ganti oli akan menyala, sebaliknya jika data belum mencapai 200jam makan tidak akan ditampilkan indikator ganti oli.

3.2 Rancangan Perangkat Keras

Rancangan perangkat keras terdiri dari 3 bagian yaitu pembuatan diagram blok, Pmbuatan *Shield*, dan perakitan perangkat keras.

3.2.1 **Diagram** Blok

Pada perancangan perangkat keras ini digambarkan pada Diagram Blok yang ditunjukkan pada gambar. 3.2. dideskripsikan sebagai berikut:



Gambar. 3.2. Diagram Blok Sistem

1. Pada bagian admin adalah *web server* dengan *database* yang menampung data *user* yang sudah registrasi dan akumulasi data mesin menyala, *web server* juga menyediakan tampilan *Maps* untuk ditampilkan pada App Android setiap *user* yang sudah registrasi.
2. Pada bagian *user* adalah bagian dimana pengguna android memantau kondisi dan posisi kendaraan yang ditampilkan melalui *App* Android. Desain *interface* dapat dilihat pada gambar 3.6.
3. Arduino adalah *board* dengan *microcontroller* yang akan mengakses dan mengendalikan setiap sensor yang terhubung dengan Arduino.
4. *Shield* adalah *board* yang dirancang khusus menghubungkan berbagai *sensor* dengan Arduino, pada penelitian ini board didesain khusus untuk menampung semua sensor yang dibutuhkan, gambar *board* dan *schematic* yang didesain menggunakan Eagle dapat dilihat pada gambar 3.3. dan 3.4.
5. GPS adalah penentu posisi kendaraan menggunakan satelit yang akan aktif sesuai perintah *microcontroller* pada Arduino.
6. Sim800L adalah modul GSM yang menghubungkan *microcontroller* ke internet.
7. *Module* LM2596 adalah *module Stepdown* untuk menurunkan tegangan ACCU dari 12volt menjadi 4volt lalu di-suplay ke seluruh *module.*

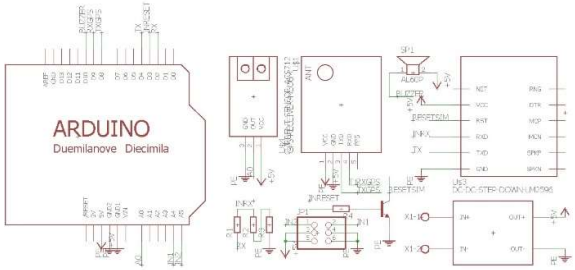
https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

8. ACS-30A adalah sensor arus untuk mendeteksi arus yang mengalir dari *alternator* ke beban yang ada pada kendaraan.
9. *Alternator* adalah alat untuk mengubah energi mekani dari putaran mesin menjadi energi listrik yang hanya akan menghasilkan listrik pada saat mesin menyala salah satu kabel output dari *alternator* akan melalui sensor ACS-30A untuk dibaca.

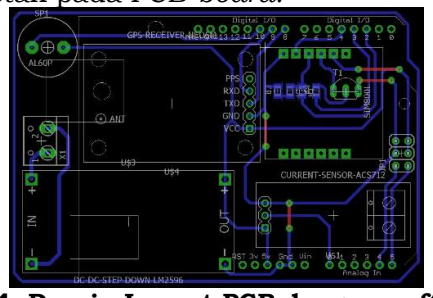
3.2.2 Pembuatan *Shield*

Pada tahap pembuatan *shield board* dimulai dengan mendesain skematik, desain skematik menggunakan *software* Eagle, berikut gambar dari skematik *shield board.*



Gambar. 3.3. Desain Skematik *Shield Biard* dengan *software* Eagle

Setelah menggambar skematik *shield board* pada Eagle, skematik diubah menjadi layout untuk dicetak pada PCB *board.*



Gambar. 3.4. Desain Layout PCB dengan *software* Eagle

3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak sistem *monitoring* dan sistem pengingat waktu perawatan kendaraan terdiri dari 4 bagian yaitu desain *user interface,* pembuatan *Blocks* coding android App, pembuatan *web server*, dan pembuatan program *microcontroller* denga Arduino IDE

3.3.1 Perancangan Android App

Desain *user interface* terdiri dari 5 *form* yaitu *form* registrasi, *form* login, *form* ganti *password*, *form* menu utama, dan *form maps view.* berikut gambar dan uraian dari masing-masing *form user interface:*

https://doi.org/10.31328/jasee.v1i01.6

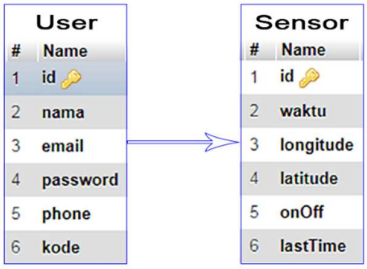
e-ISSN : 2721-320X
p-ISSN : 2721-3625

Gambar. 3.5. *Foem interface* Android

1. *Form* registrasi adalah *form* yang akan tampil pertama saat Aplikasi di buka, *form* registrasi terdiri dari 4 *textbox* yaitu, *texbox* nama, email, *password* dan nomor *hand phone*. Setelah semua *textbox* diisi dan pilih *button* registrasi maka *user* telah terdaftar dan langsung masuk ke menu login, jika *user* yang telah registrasi maka langsung pilih *button* login agara masuk ke menu login.
2. *Form login* adalah *form* yang disediakan bagi *user* yang sudah melakukan registrasi, *form login* terdiri dari 2 *textbox* dan 3 *button*, 2 *textbox* digunakan untuk mengisi ID yang sudah terdaftar saat registrasi lalu tekan *button Sign in* maka akan masuk ke menu utama, *button* registrasi digunakan jika *user* belum terdaftar, dan *button Forget Password* digunakan bagi *user* yang lupa *password* dengan mengisi ID dengan email.
3. *Form* ganti *password* disediakan bagi *user* yang lupa *password*, *form* ganti *password* terdiri dari 2 *textbox, textbox* pertama diisi dengan *code* yang telah dikirim ke *e-mail*, dan *textbox* yang kedua untuk memasukkan *password* baru.
4. *Form* menu utama adalah *form* yang berisi tampilan indikator waktu ganti oli, indikator akan menjadi warna merah jika sudah memasuki waktu ganti oli, waktu ganti oli berdasarkan lama mesin menyala. Indikator berikutnya adalah indikator mesin menyala atau mati, jika mesin menyala maka indikator akan berwarna merah. Tombol cek lokasi befungsi untuk menampilkan *form maps view*, dan *button logout* untuk keluar.
5. *Form Maps view* adalah form untuk menampilkan Maps yang menampilkan lokasi dari kendaraan terkini, dan tombol *back* untuk kembali ke menu utama.

3.3.2 Perancangan *Database Web.*

Pada Bagian Ini berisi perancangan *database* secara keseluruhan. Berikut struktur dari perancangan database.



Gambar. 3.6. Struktur *Database*

Pada perancangan *database* terdiri dari 2 tabel yaitu tabel *user* dan tabel sensor, berikut adalah rancangan dari kedua tabel tersebut:

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

3.3.2.1 Table *User*

Nama Table : tblUser
Keterangan : Berisi data pengguna
Primary Key : id

Tabel 3.1 Database User

Nama	Tipe	Keterangan
id	int	Id pengguna
Nama	Varchar	Nama pengguna
Email	Varchar	Email pengguna
Password	Varchar	*Password*
Phone	Varchar	No HP pengguna
Kode	Varchar	Kode ganti *password*

3.3.2.2 Tabel Sensor

Nama Table : tblSensor
Keterangan : Berisi data sensor
Primary Key : id

Tabel 3.2. *Database* Sensor

Nama	Tipe	Keterangan
Id	int	Id sensor
Waktu	Varchar	Akumulasi waktu mesin menyala
Longitude	Varchar	Garis bujur
Latitude	Varchar	Garis lintang
OnOff	Varchar	Kondisi mesin hidup atau mati
LastTime	Varchar	Waktu terakhir database di-*update*

3.4 Prosedur Pengujian

Pengujian dilakukan pada beberapa bagian sub-sistem, yaitu:

1. Pengujian GPS

Pengujian GPS dilakukan dengan menentukan koordinat sebagai titik acuan, pengambilan titik acuan ditentukan pada google *maps,* dengan cara memberi pin pada lokasi yang diinginkan untuk mendapatkan bujur dan lintang. Setelah mendapatkan titik acuan, kendaraan diparkir pada lokasi tersebut untuk mengambil data bujur dan lintang kendaraan, data bujur dan lintang kendaraan diambil pada database yang di *update* setiap 10 detik, setelah data didapat data akan dibandingkan tingkat keakuratannya.

2. Pengujian konsumsi Daya *hardware*

Pengujian konsumsi daya *hardware* dilakukan untuk mengetahui daya yang dibutuhkan *hardware,* pengujian dilakukan dengan cara mengukur daya yang diserap oleh *hardware*, dan dibandingkan dengan daya yang mampu di-s*upply* oleh catu daya(aki). Agar dapat mengetahui berapa lama *hardware* dapat di-*supply* oleh aki.

3. Pengujian GSM

Pengujian GSM dilakukan untuk mengetahui apakah GPRS bekerja dengan baik, pengujian GSM dilakukan dengan cara memasukkan perintah AT *Command* pada program Arduino IDE.

4. Penujian *Database web*

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

Pengujian *database* dilakukan dengan tiga macam *integritas referensial*, yaitu penambahan *user* (*insert*) apakah database berhasil menambahkan *user* saat registrasi, penghapusan data (*delete*) apakah *database* berhasil menghapus/ mereset data lama mesin menyala, dan Peremajaan (*update*) apakah database mampu melakukan *update* data kondisi mesin hidup atau mati, akumulasi data mesin menyala, dan koordinat kendaraan.

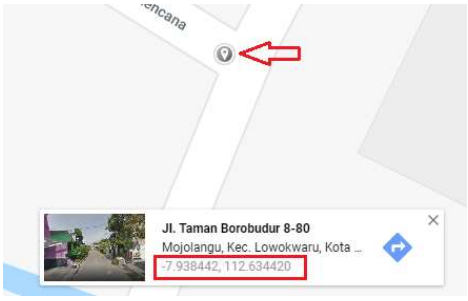
4. HASIL DAN PEMBAHASAN

4.1 Pengujian GPS

Pengujian GPS dilakukan untuk mendapatkan nilai *error* atau selisih antara titik yang sudah ditentukan, dikarenakan pada setiap setiap GPS memiliki nilai keakuratan yang berbeda. Pada tahap pengujian GPS Dimulai dengan menentukan Titik acuan pada googlemap dan mencatat koordinat pada titik yang sudah ditentukan, Penentuan titik acuan dapat dilihat pada gambar 4.1.

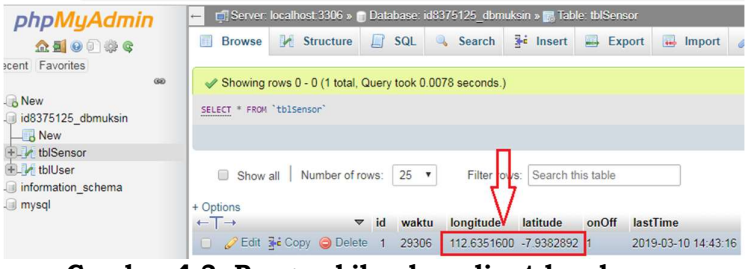
Gambar 4.1. Penentuan Titik Acuan

Gambar 4.2. menunjukkan proses pengambilan data koordinat titik acuan pada Google *Maps browser* PC



Gambar 4.2. Pengambilan data Titik Acuan

Gambar 4.3. menunjukkan proses pengambilan data koordinat kendaraan, koordinat diambil saat kendaraan berhenti pada titik acuan yang sudah ditentukan sebelumnya, koordinat diambil langsung pada *database*.



Gambar 4.3. Pengambilan koordinat kendaraan

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

Pada tabel dibawah menunjukan hasil dari pengujian GPS, yaitu perbandingan data antara titik acuan dan posisi kendaraan terdeteksi untuk mendapatkan nilai *error* atau selisih jarak antara titik acuan dan posisi kendaraan terdeteksi.

Tabel 4.1. Perbandingan hasil deteksi GPS dengan titik acuan

NO	Titik Acuan	Posisi kendaraan terdeteksi	*Error*
1	-7.938206, 112.635188	-7.9382157, 112.6351700	2.26 m
		-7.9382181, 112.6351800	1.61 m
		-7.9382248, 112.6351800	2.27 m
		-7.9382195, 112.6351700	2.49 m
		-7.9382210, 112.6351900	1.68 m
2	-7.938482, 112.634404	-7.9384973, 112.6344100	1.83 m
		-7.9384987, 112.6343900	2.41 m
		-7.9385032, 112.6343900	2.82 m
		-7.9384993, 112.6344200	2.61 m
		-7.9385010, 112.6344100	2.21 m
3	-7.937740, 112.633293	-7.9377424, 112.6332900	0.42 m
		-7.9377453, 112.6333100	1.96 m
		-7.9377440, 112.6332800	1.50 m
		-7.9377432, 112.6332700	2.56 m
		-7.9377420, 112.6332800	1.45 m
4	-7.936661, 112.634016	-7.9366613, 112.6340100	0.66 m
		-7.9366605, 112.6339900	2.86 m
		-7.9366587, 112.6340200	0.51 m
		-7.9366595, 112.6340100	0.68 m
		-7.9366607, 112.6340300	1.54 m
5	-7.937307, 112.635252	-7.9373062, 112.6352700	1.98 m
		-7.9373087, 112.6352400	1.34 m
		-7.9373056, 112.6352500	0.27 m
		-7.9373115, 112.6352300	2.47 m
		-7.9373070, 112.6352400	1.32 m
Rata-rata error			1.75 m

4.2 Pengujian Konsumsi daya Hardware

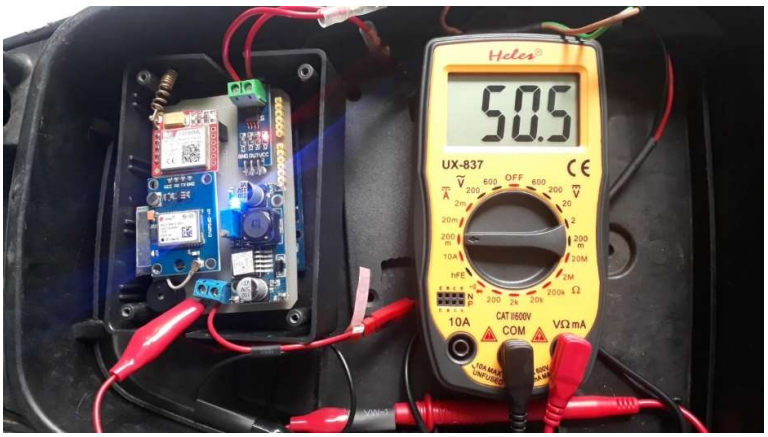
Pada pengujian ini bertujuan untuk mengetahui berapa lama Aki mampu menyuplay *hardware*, pengujian dilakukan dengan mengukur arus *hardware* lalu kapasitas Aki dibagi dengan arus yang dibutuhkan *hardware*.

Diketahui: Tegangan Aki : 12 v
Kapasitas Aki : 3.5Ah
Arus Hardware : 50mA (0.05A)

Waktu Pemakaian = Kapasitas Aki (Ah) / Amper Hardware (A)
= 3.5Ah / 0.05A = 70 Jam
Dieffisiensi Aki 20% (14Jam)
= 70 Jam – 14 Jam = 56 jam
= 2.33 Hari (2hari, 7jam, 55menit, 12 Detik)

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625



Gambar 4.4. Pengukuran arus *Hadware*

4.3 Pengujian GSM

Pengujian GSM dilakukan hanya pada mode GPRS, dikarena pada penelitian ini GSM hanya dipakai pada mode GPRS, pengujian dilakukan untuk mengetahui apakah semua fungsi dapat bekerja denga baik, hasil pengujian dapat dilihat pada gambar 4.4. Berikut penjelasan dari hasil pengujian GPRS:

```
Goodnight moon!
ÿAT
OK
AT
OK
AT
OK
AT+SAPBR=3,1,"Contype","GPRS"
OK
AT+SAPBR=1,1
OK
AT+SAPBR=2,1
+SAPBR: 1,1,"100.104.209.61"

OK
AT+HTTPINIT
OK
AT+HTTPPARA="CID",1
OK
AT+HTTPPARA="URL","muksinproject.000webhostapp.com/index.php/welcome
ERROR
AT+HTTPPARA="URL","muksinproject.000webhostapp.com/index.php/welcome
OK
AT+HTTPACTION=0
OK

+HTTPACTION: 0,200,10
```

Gambar 4.5. Hasil pengujian AT Command GPRS

1. AT+SAPBR=3,1,\"Contype\",\"GPRS\" adalah perintah untuk seting koneksi pada mode GPRS. Respon “OK” menandakan koneksi berhasil diseting pada mode GPRS.
2. AT+SAPBR=1,1 adalah perintah untuk mengaktifkan koneksi GPRS. Respon “OK” menandakan GPRS berhasil diaktifkan.
3. AT+SAPBR=2,1 +SAPBR: 1,1,“100.104.209.61” adalah perintah untuk cek IP yang didapatkan dari Operator. Respon “OK” menandakan IP sesuai.
4. AT+HTTPINIT berfungsi untuk menginisialisasi koneksi HTTP. Respon “OK” menandakan inisialisasi berhasil dilakukan.

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

5. AT+HTTPPARA=\"URL\",\"muksinproject.000webhostapp.com/index.php/welcome adalah perintah untuk mengirim data ke server. Respon “OK” menandakan pengiriman data berhasil
6. AT+HTTPACTION=0 adalah perintah untuk mengirim dan menerima balasan. Respon “OK” dan mendapat balasan 200 menandakan pengiriman sukses.

4.4 Pengujian Database web

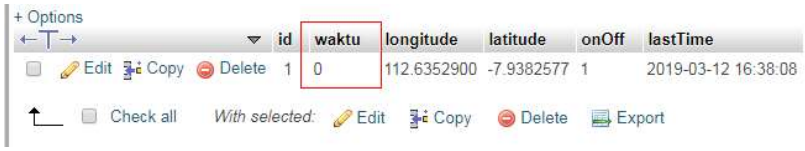
Pengujian dilakukan dengan melihat langsung pada *database server* dalam berbagai kondisi, pengujian *database web* terbagi menjadu dua bagian yaitu pengujian database sensor dan pengujian database *User.*

3.7.1 Pengujian *Database* Sensor

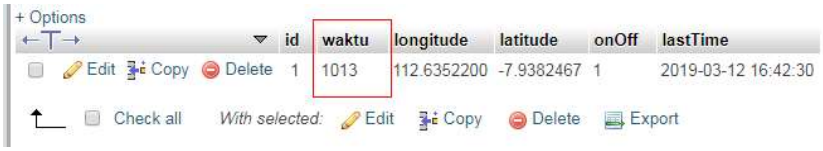
Dalam pengujian *database* sensor terbagi menjadi 4 kondisi yaitu:

1. Akumulasi kendaraan hidup.

Pada tahap pertama dilakukan pengambilan data saat kendaraan dalam kondisi mati, pengujian ini dilakukan untuk memastikan akumulasi tidak berjalan saat mesin mati. Selanjutnya data diambil pada saat kendaraan telah dihidupkan selama 1000 detik.



Gambar 4.6. Akumulasi waktu kendaraan hidup sebelum mesin On



Gambar 4.7. Akumulasi waktu kendaraan hidup selama 16 menit

2. Peremajaan Bujur dan Lintang.

Pengujian bujur dan lintang dapat dilihat pada gambar 4.6, 4.7, dan 4.8. Pada setiap gambar menampilkan derajat bujur dan lintang yang berbeda, itu menandakan bahwa peremajaan Bujur dan lintang bekerja dengan baik.

3. Kendaraan hidup atau Mati.

Pada saat kendaraan hidup Tabel *onOff* pada gambar dibawah akam menunjukkan angka 2, dan jika sebaliknya maka akan menunjukkan angka 1, jika kondisi ini terjadi menandakan *database* dan sensor arus pada kendaraan bekerja dengan baik.

Gambar 4.8. Indicator kendaraan sedang hidup

4. dan waktu terakhir kendaraan hidup.

Pada gambar 4.6, 4.7, dan 4.8 pada bagian *LastTime* menunjukkan waktu terakhir kendaraan hidup yang berbeda, hal ini menandakan setiapkali kendaraan dihidupkan database berhasil menyimpan waktu yang terakhirnya.

3.7.2 Pengujian *database User*

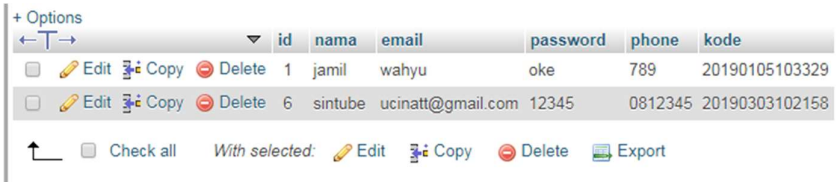
Dalam pengujian database user terbagi menjadi 2 bagian yaitu:

1. Penambahan User

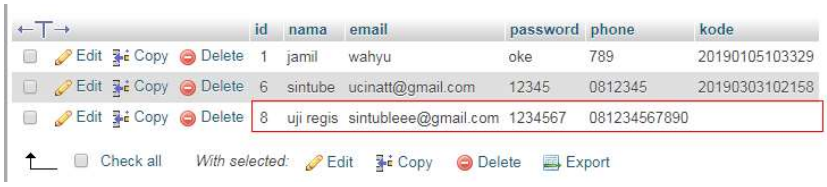
https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

Gambar 4.9 adalah gambar yang menunjukkan *database* sebelum dilakukan penambahan *user,* sedangkan pada gambar 4.10. menunjukkan *user* berhasil ditambahkan.



Gambar 4.9. Sebelum ditambahkan *User*



Gambar 4.10. Sesudah ditambahkan *User*

2. Kode untuk ganti *Password*

Pada saat *user* mengganti *password* akan diminta kode yang telah dikirim ke email, setelah kode dan *password* baru dimasukkan, maka *password user* pada *database* akan terganti. Gambar 4.10. menunjukkan kode pada *database* dan *password* yang telah tergganti.

Gambar 4.11. Kode ganti *password* dan *password* baru

5. KESIMPULAN

Berdasarkan analisis GPS, selisih jarak antara titik acuan dengan lokasi terdeteksi memiliki nilai rata-rata sebbesar 1.75m. Analisis Konsumsi daya *hardware* menunjukan kemampuan aki untuk menyuplay *hardware,* aki kendaraan dapat menyuplay *hardware* selama 2.33 Hari (2hari, 7jam, 55menit, 12 Detik), denga waktu tersebut pengguna harus menggunakan atau memanaskan kendaraan dala sehari minimal 30menit agar aki terisi kembali.

DAFTAR RUJUKAN

[1] P. A. Eka, Smart City beserta Cloud Computing, Bandung: Informatika Bandung, 2014.

[2] F. H. Putra, "Aplikasi IoT untuk Rumah Pintar dengan Fitur Prediksi Cuaca," *Ind. Journal on Computinng,* vol. 5, p. 1746, 2018.

[3] S. Gifary, "Intensitas Penggunaan Smartphone Terhadap Perilaku Komunikasi,"

https://doi.org/10.31328/jasee.v1i01.6

e-ISSN : 2721-320X
p-ISSN : 2721-3625

Jurnal Sosioteknologi, vol. 2, p. 170, 2015.

[4] A. N. Trisetiyanto, "Pengembangan Sistem Peringatan Ganti Oli Pada Sepeda Motor," *Jurnal Teknik Elektro,* pp. 1-6, 2012.

[5] G. H. Cahyono, "Internet Of Things (Sejarah,Teknologi dan Penerapannya)," *Forum Teknologi,* pp. 35-41, 2016.

[6] A. F. Silvia, "Rancang Bangun Akses Kontrol Pintu Gerbang Berbasis Arduino dan Android," *Electrans,* vol. 13, pp. 1-10, 2014.

[7] M. F. Wicaksono and H. , Mudah Belajar Mikrokontroler Arduino, Bandung: Informatika Bandung, 2017.

[8] I. . A. Saputra, "Aplikasi Layanan Bengkel Mobil Berbasis Android di Kota Bandar Lampung," *Skripsi Ilmu Komputer,* p. 15, 2017.

[9] N. Safaat, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Bandung: Informatika Bandung, 2014.

[10] F. D. Kusnandar, "Identifikasi Sistem Pengisian Pada Mobil Toyota Kijang Innova 1TR-FE," *Skripsi,* p. 6, 2015.

[11] R. Sakti, "Sistem Keamanan Motor Menggunakan Sensor MPU 6050 dan Tracking Lokasi Dengan GPS Android," *Skripsi,* p. 11, 2017.

https://doi.org/10.31328/jasee.v1i01.6