**SISTEM MONITORING BATERAI MOTOR BERBASIS NODEMCU UNTUK MENDETEKSI TEGANGAN, ARUS DAN PENGISIAN BATERAI**

**SKRIPSI**

**Oleh:**

**Asep Trisna Setiawan**

**15411061**

****

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BANDAR LAMPUNG  
 2019**

**SISTEM MONITORING BATERAI MOTOR BERBASIS NODEMCU UNTUK MENDETEKSI TEGANGAN, ARUS DAN PENGISIAN BATERAI**

**SKRIPSI**

(Diajukan Sebagai Salah Satu Syarat dalam memenuhi Gelar Sarjana Strata-1 (S1) pada Fakultas Ilmu Komputer, Program Studi Sistem Informasi

Universitas Bandar Lampung)

**Oleh:**

**Asep Trisna Setiawan**

**15411061**

****

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BANDAR LAMPUNG**

**2019**

# LEMBAR PENGESAHAN



|  |  |  |
| --- | --- | --- |
| Judul Skripsi | : | **SISTEM MONITORING BATERAI MOTOR BERBASIS NODEMCU UNTUK MENDETEKSI TEGANGAN, ARUS DAN PENGISIAN BATERAI** |
| Nama Mahasiswa | : | **ASEP TRISNA SETIAWAN** |
| No.Pokok Mahasiswa | : | 15411061 |
| Program Studi | : | Sistem Informasi |
| Fakultas | : | Fakultas Ilmu Komputer |

Bandar Lampung, Februari 2019

**MENGESAHKAN**

|  |  |
| --- | --- |
| Dosen Pembimbing | Ketua Program Studi  Sistem Informasi |
| **Taqwan Thamrin, S.T., M.Sc.** | **Taqwan Thamrin, S.T., M.Sc.** |

**MENGESAHKAN**

**Tim Penguji Tanda Tangan**

**Ketua :** **Fenty Ariani, S.Kom., M.Kom. ...…….....**

**Penguji Utama :** **Robby Yuli Endra, S.Kom., M.Kom. .………..**

**Penguji 1 :** **Yuthsi Aprilinda, S.Kom., M.kom. ..………..**

**Penguji 2 :** **Erlangga, S.Kom., M.Kom. ..………..**

**MENGETAHUI**

Dekan Fakultas Ilmu Komputer

**Ahmad Cucus, S.Kom., M.Kom**.

**Tanggal Lulus : 20 Februari 2019**

# ABSTRAK

**Sistem Monitoring Baterai Motor Berbasis NodeMcu Untuk Mendeteksi Tegangan, Arus Dan Pengisian Baterai**

**Oleh:**

**Asep Trisna Setiawan (15411061)**

Aki atau baterai motor/mobil merupakan salah satu komponen penting untuk kendaraan roda dua/empat sehingga perawatan dan pengontrolan sangat dibutuhkan supaya aki/baterai dapat digunakan dalam jangka waktu yang lama.

Penelitian yang dilakukan penulis bertujuan untuk mengatasi atau meminimalisir sebuah masalah yang sering dialami oleh para pemilik kendaraan roda dua/empat yaitu rusaknya aki atau baterai motor/mobil secara tiba-tiba. Penelitian yang dilakukan penulis juga bertujuan untuk mengatasi atau meminimalisir sebuah masalah yang sering dialami oleh para pemilik kendaraan roda dua/empat yaitu hilangnya kendaraan akibat banyaknya pencurian kendaraan.

Pada peneltian ini penulis menggunakan sebuah mikrokontroler *NodeMcu* sebagai pengendali, sensor tegangan dan arus sebagi input (masukan) untuk mengambil tegangan dan arus baterai, modul gps neo-6m sebagai(masukan) untuk mendeteksi latitude longitude dan modul wifi sebagai pengirim data. . Hasilnya dari penelitian ini adalah perancangan alat dan aplikasi untuk dapat menampilkan tegangan ,arusbaterai ,lokasi keberadaan kendaraan, dan anti stater yang dapat di kontrol melalui aplikasi android.

Kata kunci: NodeMcu, Sensor tegangan, Sensor Arus, modul Gps, dan baterai/aki.

***ABSTRACT***

***NodeMcu Based motorcycle Battery Monitoring System For Detecting Voltage, Current And Battery Charging***

***By:***

***Asep Trisna Setiawan (15411061)***

*Motorcycle / car battery or battery is one of the important components for two / four-wheeled vehicles so that maintenance and control is needed so that the battery / battery can be used for a long period of time.*

*Research conducted by the author aims to overcome or minimize a problem that is often experienced by the owners of two / four-wheeled vehicles that are damaged batteries or motorcycle batteries / cars suddenly. Research conducted by the author also aims to overcome or minimize a problem that is often experienced by the owners of two / four wheel vehicles, namely the loss of vehicles due to the large number of vehicle theft.*

*In this study the author uses a NodeMcu microcontroller as a controller, voltage and current sensor as input (input) to take battery voltage and current, neo-6m gps module as (input) to detect latitude longitude and wifi module as data sender. . The result of this study is the design of tools and applications to be able to display the voltage, battery current, location of the vehicle's presence, and anti-stater that can be controlled via the android application.*

*Keywords: NodeMcu, Voltage Sensor, Current Sensor, GPS module, and battery / battery.*

# RIWAYAT HIDUP

Penulis yang memiliki nama lengkap Asep Trisna Setiawan dilahirkan di Sinar Baru pada tanggal 02 September 1995, anak pertama dari empat bersaudara dari Ayah yang bernama Tatang Sudrajad Ibu bernama Jubaidah.

Pendidikan Formal yang telah ditempuh Penulis adalah **:**

1. Sekolah Dasar di SDN 02 Tanjung Durian Kecamatan Buay Pemaca Kabupaten Oku Selatan, lulus pada tahun 2009
2. Madrasah Tsanawiyah di Miftahul Huda Tanjung Rejo Kasui Way Kanan, lulus pada tahun 2012
3. Sekolah Menengah Atas di SMA Adiguna Bandar Lampung, lulus pada tahun 2015

Tercatat sebagai mahasiswa Fakultas Ilmu Komputer Universitas Bandar Lampung sebagai mahasiswa Sarjana Strata 1 dengan Program Studi Sistem Informasi sejak tahun 2015 s/d sekarang.

# MOTTO

**“Jika kamu berani jangan takut-takut, Jika kamu takut jangan berani-berani”**

**(Asep Trisna Setiawan)**

# PERSEMBAHAN

Puji syukur peneliti panjatkan kehadirat Allah SWT. Atas segala limpahan rahmat dan karunia serta hidayahNya yang telah di limpahkan kepada kita semua tak terkecuali kepada peneliti sehingga dapat terselesainkannya skripsi ini. Peneliti menyadari bahwa dalam skripsi ini tidak terlepas dari bimbingan, bantuan serta dukungan dari berbagai pihak. Dengan selesainya tugas skripsi ini, peneliti mempersembahkan kepada Ayah dan Ibu yang tidak pernah lelah memberi semangat dan do’a yang terus-menerus tanpa henti.Serta semua pihak yang telah membantu peneliti dalam menyelesaikan tugas akhir ini.

# KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Kuasa yang telah memberikan Rahmat, Taufik, Hidayah dan Inayahnya sehingga penulis dapat menyelesaikan Penulisan ini berjudul **Sistem Monitoring Baterai Motor Berbasis NodeMcu Untuk Mendeteksi Tegangan, Arus Dan Pengisian Baterai,** yang bertujuan sebagai salah satu syaratdalam memenuhi Gelar KesarjanaanStrata-1 (S1)

Penulis menyadari bahwa dalam pembuatan skripsi ini tidak terlepas dari bantuan dan bimbingan semua pihak, sehingga Penulisan ini dapat diselesaikan dengan baik. Oleh karena itu, pada kesempatan ini penulis mengucapkan banyak terima kasih kepada:

1. Ibu Dra. Hj. Sri Haryati Barusman, selaku Ketua Dewan Pembina Yayasan Administrasi Lampung.
2. Bapak Dr. Andala Rama Putra Barusman S.E., MA.Ec., selaku Ketua Yayasan Administrasi Lampung.
3. Bapak Dr. Ir. Hi. Yusuf Sulfrano Barusman, MBA., selaku Rektor Universitas Bandar Lampung.
4. Bapak Ahmad Cucus, S.Kom., M.Kom., selaku Dekan Fakultas Ilmu Komputer Universitas Bandar Lampung.
5. Bapak Taqwan Thamrin, S.T., M.Sc., selaku Ketua Program Studi Sistem Informasi pada Fakultas Ilmu Komputer Universitas Bandar Lampung dan selaku Pembimbing Utama.
6. Seluruh Bapak dan Ibu Dosen serta karyawan Fakultas Ilmu Komputer Universitas Bandar Lampung.
7. Kakak-kakak tingkat yang telah meluangkan waktu bagi penulis untuk bertanya segala hal yang berkaitan dengan Laporan Penelitian Ilmiah ini.
8. Teman-teman seperjuangan yang menempuh studi di Fakultas Ilmu Komputer Universitas Bandar Lampung.
9. Semua rekan-rekan yang telah banyak membantu penulis dalam menyusun penelitian ilmiah ini.

Serta berbagai pihak yang tidak dapat penulis sebutkan satu persatu, Akhir kata, dengan senang hati penulis memohon maaf atas segala kekurangan dalam penulisan ini, penulis megharapkan kritik, saran dan perbaikan semoga Penulisan ini, dapat berguna dan bermanfaat bagi kita semua khususnya pembaca.

|  |  |
| --- | --- |
|  | Bandar Lampung  **Penulis** |
|  |  |

# DAFTAR ISI

# Cover

[**LEMBAR PENGESAHAN** iii](#_Toc1080933)

[**ABSTRAK** v](#_Toc1080934)

[**RIWAYAT HIDUP** vii](#_Toc1080935)

[**MOTTO** viii](#_Toc1080936)

[**PERSEMBAHAN** ix](#_Toc1080937)

[**KATA PENGANTAR** x](#_Toc1080938)

[**DAFTAR ISI** xii](#_Toc1080939)

[**DAFTAR GAMBAR** xv](#_Toc1080941)

[**DAFTAR TABEL** xvii](#_Toc1080942)

[**BAB I PENDAHULUAN** i](#_Toc1080943)

[1.1. Latar Belakang i](#_Toc1080944)

[1.2. Identifikasi Masalah 2](#_Toc1080945)

[1.3. Rumusan Masalah 3](#_Toc1080946)

[1.4. Batasan Masalah 3](#_Toc1080947)

[1.5. Tujuan dan Manfaat Penelitian](#_Toc1080949)...................................................... 3

[1.6. Kerangka Pemikiran 4](#_Toc1080950)

[1.7. Sistematika Penulisan 5](#_Toc1080951)

[**BAB II LANDASAN TEORI** 6](#_Toc1080952)

[2.1. Tinjauan Pustaka 6](#_Toc1080953)

[2.2. Tinjauan Studi 8](#_Toc1080954)

[2.2.2 Definisi monitoring 8](#_Toc1080956)

[2.2.3 Mikrokontroler 8](#_Toc1080957)

[2.2.4 Node Mcu 9](#_Toc1080958)

[2.2.5 Sensor tegangan 8](#_Toc1080959)

[2.2.6 Sensor Arus ACS 712 8](#_Toc1080960)

[2.2.7 Global Positioning System 10](#_Toc1080961)

[2.2.8 GPS UBlox Neo 6M 10](#_Toc1080962)

[2.2.9 Arduino Software IDE 11](#_Toc1080963)

[2.2.10 Menulis Sketch 11](#_Toc1080964)

[2.2.11 Android 13](#_Toc1080965)

[2.2.12 Android 13](#_Toc1080966)

[**BAB III METODE PENELITIAN** 15](#_Toc1080967)

[3.1. Metode Pengumpulan Data 15](#_Toc1080968)

[3.1.1 Observasi 15](#_Toc1080969)

[3.1.2 Studi Pustaka 16](#_Toc1080970)

[3.2. Analisa kebutuhan perangkat keras dan Perangkat lunak 18](#_Toc1080971)

[3.2.3 Kebutuhan perangkat keras 18](#_Toc1080972)

[3.2.4 Kebutuhan perangkat lunak 18](#_Toc1080973)

[3.3. Analisa Perancangan Sistem 19](#_Toc1080974)

[3.3.5 Desain Perancangan Alat 19](#_Toc1080975)

[3.3.6 Desain Alur kerja Sistem 20](#_Toc1080976)

[3.3.7 Desain Aplikasi Android 21](#_Toc1080977)

[3.3.8 Flowchart Sistem 24](#_Toc1080978)

[3.3.9 Desain Struktur database 25](#_Toc1080979)

[**BAB IV HASIL DAN PEMBAHASAN** 26](#_Toc1080980)

[4.1. Instalasi Alat 26](#_Toc1080981)

[4.2. Aplikasi Android 37](#_Toc1080983)

[4.3. Pengujian Aplikasi 47](#_Toc1080984)

[**BAB V KESIMPULAN DAN SARAN** 52](#_Toc1080985)

[5.1. Kesimpulan 52](#_Toc1080986)

[5.2. Saran 52](#_Toc1080987)

[**DAFTAR PUSTAKA**](#_Toc1080988)

**LAMPIRAN**

# DAFTAR GAMBAR

[Gambar 2. 1 Sensor Tegangan 8](#_Toc1328099)

[Gambar 2. 2 Sensor Arus 712 9](#_Toc1328100)

[Gambar 2. 3 NodeMCU ESP826 dan Skema Pin 9](#_Toc1328101)

[Gambar 2. 4 GPS uBlox Neo 6M 11](#_Toc1328102)

[Gambar 2. 5 Arduino IDE 12](#_Toc1328103)

[Gambar 2. 6 Android Studio 14](#_Toc1328104)

[Gambar 3. 1 Desain Pearancangan Alat................................................................ 19](#_Toc1329121)

[Gambar 3. 2 Desain Alur kerja Sistem 20](#_Toc1329122)

[Gambar 3. 3 Perancangan menu login 21](#_Toc1329123)

[Gambar 3. 4 Perancangan Menu profil 21](#_Toc1329124)

[Gambar 3. 5 Perancangan Menu monitoring 22](#_Toc1329125)

[Gambar 3. 6 Perancangan menu grafik 22](#_Toc1329126)

[Gambar 3. 7 Perancangan Menu lokasi 23](#_Toc1329127)

[Gambar 3. 8 Perancangan ubah data user 23](#_Toc1329128)

[Gambar 3. 9 Perancangan Flowchart System 24](#_Toc1329129)

[Gambar 4. 1 Node Mcu Sensor tegangan............................................................. 26](#_Toc1330398)

[Gambar 4. 2 Serial monitor tegangan 28](#_Toc1330399)

[Gambar 4. 3 Rangkaian Node Mcu dan sensor asus ACS712 28](#_Toc1330400)

[Gambar 4. 4 Hasil di serial monitor arduino ide 30](#_Toc1330401)

[Gambar 4. 5 Rangakaian Node Mcu dan modul Gps Neo-6m 31](#_Toc1330402)

[Gambar 4. 6 Hasil Latitude dan longitude di serial monitor arduin ide 33](#_Toc1330403)

[Gambar 4. 7 Node Mcu dan Relay 33](#_Toc1330404)

[Gambar 4. 8 Splash Screen 38](#_Toc1330405)

[Gambar 4. 9 Menu Login 39](#_Toc1330406)

[Gambar 4. 10 Menu profil 40](#_Toc1330407)

[Gambar 4. 11 Menu utama 41](#_Toc1330408)

[Gambar 4. 12 Menu ubah data pengguna 42](#_Toc1330409)

[Gambar 4. 13 Menu Ubah Photo Profile 43](#_Toc1330410)

[Gambar 4. 14 Menu Monitoring Tegangan dan Arus Baterai 44](#_Toc1330411)

[Gambar 4. 15 Menu monitoring 45](#_Toc1330412)

[Gambar 4. 16 Menu Maps Kendaraan 46](#_Toc1330413)

[Gambar Gambar 4. 17 *Flowchart white box testing* 48](#_Toc1330414)

[Gambar 4. 18 *Flowgrap* 49](#_Toc1330415)

# DAFTAR TABEL

[Tabel 3. 1 Observasi lapangan 15](#_Toc1329358)

[Tabel 3. 2 Observasi lapangan 2 16](#_Toc1329359)

[Tabel 3. 3 Monitoring tegangan baterai motor dan mobil. 17](#_Toc1329360)

[Tabel 3. 4 Perancangan Tabel user 25](#_Toc1329361)

[Tabel 3. 5 Perancangan Tabel volt 25](#_Toc1329362)

[Tabel 3. 6 Perancangan Tabel arus 25](#_Toc1329363)

[Tabel 3. 7 Perancangan Tabel lokasi 25](#_Toc1329364)

[Tabel 4. 1 Tabel rangkaian node mcu sensor tegangan....................................... 27](#_Toc1330541)

[Tabel 4. 2 Tabel rangkaian node mcu sensor tegangan 29](#_Toc1330542)

[Tabel 4. 3 Tabel rangkaian node mcu dan modul gps neo-6m 31](#_Toc1330543)

[Tabel 4. 4 Tabel rangkaian node mcu dan Relay 34](#_Toc1330544)

[Tabel 4. 5 Tabel user 37](#_Toc1330545)

[Tabel 4. 6 Tabel volt 37](#_Toc1330546)

[Tabel 4. 7 Tabel arus 37](#_Toc1330547)

[Tabel 4. 8 Tabel lokasi 37](#_Toc1330548)

[Tabel 4. 9 *Black box testing* 47](#_Toc1330549)

[Tabel 4. 10 Tabel Pengujian validasi satu 51](#_Toc1330550)

[Tabel 4. 11 Tabel pengujian validasi dua 51](#_Toc1330551)

# PENDAHULUAN

## Latar Belakang

Seiring berjalannya waktu, berbagai teknologi terbaru bermunculan. Salah satunya dibidang outomotif berbagai kendaraan baru bermunculan dari kendaraan roda dua maupun kendaraan roda empat dari harga yang standar sampai dengan harga yang lumayan mahal. Di indonesia pengguna kendaraan sangatlah tinggi, hampir setiap orang memiliki kendaraan karena kendaraan adalah alat yang sangat membantu manusia untuk dapat menempuh perjalanan jauh dengan mudah dan dapat ditempuh dengan waktu yang rendah.

Agar Kendaraan dapat digunakan sesuai dengan kegunaannya kendaraan memerlukan sebuah perawatan yang maksimal agar kendaraan dapat digunakan dengan aman dan nyaman. Ada beberapa komponen kendaraan yang harus di perhatikan, diantaranya ban, aki/baterai dan komponen-komponen yang lainnya.

Baterai/aki merupakan salah satu komponen penting dikendaraan roda empat maupun roda dua karena semua kelistrikan semua berasal dari baterai tersebut termasuk *starter* untuk menghidupkan kendaraan itu sendiri. Pada kendaraan roda empat pada umumnya memiliki sebuah aki atau baterai dengan tegangan 12 *volt* sampai dengan 24 *volt* untuk kendaraan seperti bus, truk dan lain-lain dan memiliki kuat arus/*Ampere* yang berbeda-beda kurang lebih dari 40A-80A *(ampere*). Sedangkan pada kendaraan roda dua pada umumnya memiliki baterai pada tegangan 12 *volt* dengan arus 3 A. Aki atau baterai bisa di katakan bagus apabila dalam kondisi mesin mobil mati aki memliki tegangan 12 volt dan apabila kurang dari 12 *volt* maka bisa di katakan aki atau baterai lemah. Aki bisa di katakan bagus apabila kondisi mesin mobil hidup aki atau baterai memiliki

tegangan 13-14 *volt* apabila lebih dari 14 *volt* maka kondisi pengisian baterai *over* (melebihi kapasitas) (Shegi, 2017).

Masalah yang sering terjadi pada kendaraan roda empat adalah habisnya baterai mobil sehingga mobil tidak dapat digunakan. Masalah itu terjadi karena para pengguna kendaraan roda empat lalai dalam pengecekan baterai mobil mereka. Rusaknya atau habisnya baterai secara tiba-tiba biasanya diakibatkan oleh beberapa hal diantaranya :sistem pengisian baterai tidak normal, lupa mematikan lampu kabin, menyalakan audio saat kondisi mesin mati, kebocoran arus listrik dan lain-lain.

Selain baterai kendaraan yang perlu diperhatikan sistem keamanan tambahan pada kendaraan juga perlu diperhatikan untuk minimalisir terjadinya tindak kejahatan seperti kasus pencurian dan pengambilan paksa kendaraan bermotor semakin sering terjadi.

Melihat dari kondisi ini maka penulis melakukan sebuah penelitian yang bertujuan untuk membantu pengguna kendaraan roda empat dan roda dua dalam perawatan atau pengecekan baterainya secara berkala, dapat memonitoring keberadaan kendaraan, serta dapat mengaktifkan dan menonaktifkan anti *stater* kendaraan dari android pemilik kendaran.

## Identifikasi Masalah

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

1. Baterai motor tidak di cek secara berkala, sehingga rusak atau habisnya baterai tidak terkontrol.
2. Sistem pengisian baterai yang tidak bekerja secara normal yang menyebabkan rusaknya baterai motor.
3. Belum adanya alat bantu untuk memonitoring baterai motor yang dapat digunakan secara *real time*.
4. Diperlukankan sistem keamanan tambahan pada kendaraan untuk dapat memonitoring kendaraan .

## Rumusan Masalah

Berdasarkan masalah yang telah dijabarkan maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana cara mendesain alat yang dapat mendeteksi tegangan, dan arus pada baterai motor.
2. Bagaimana caranya mendesain sebuah aplikasi monitoring baterai motor yang dapat memberikan informasi terkait tegangan, dan arus baterai.
3. Bagaimana caranya mendesain sebuah aplikasi monitoring baterai motor yang dapat memberikan informasi terkait tegangan, arus baterai.
4. Bagaimana caranya mendesain sebuah aplikasi monitoring keberadaan kendaraan.

## Batasan Masalah

Supaya penelitian ini sesuai dengan tujuan yang dinginkan penulis dan dapat diselesaikan dengan tepat waktu maka penulis hanya akan membahas terkait mohnitoring baterai menggunakan *Node Mcu* untuk bisa mendeteksi *volt*( tegangan), *ampere(*arus), serta dapat mendeteksi dimana keberadaan kendaraan menggunaka modul gps NEO-6M.

## Tujuan dan Manfaat Penelitian

Adapun tujuan dari penelitian ini antara lain:

1. Untuk mendesain alat monitoring yang dapat mendeteksi tegangan, arus baterai motor dan mobil.
2. Untuk mendesain alat monitoring yang dapat mendeteksi dimana keberadaan kendaraan.
3. Untuk mendesain alat monitoring yang dapat mengaktifkan dan menonaktifkan anti stater pada kendaraan.

Adapun manfaat dari penelitian ini antara lain:

1. Peneliti dan pembaca dapat mengetahui bagaimana cara mendesain alat monitoring baterai.
2. Peneliti dan pembaca dapat mengetahui bagaimana cara kerja alat monitoring baterai.

## Kerangka Pemikiran

* Sistem pengisian baterai mobil/motor yang tidak bekerja secara normal sehingga membuat baterai dan sebagian komponen mobil rusak (Putra, 2018).
* Belum adanya alat bantu untuk memonitoring baterai mobil yang dapat digunakan secara *real time*
* Baterai mobil tidak dikontrol secara berkala.

***Approach***

***Problems***

1 kerangka pemikiran

Mikrokontroler Node Mcu

***Implementation***

***Result***

Sistem Monitoring Kendaraan untuk memonitoring tegangan baterai, arus baterai dan lokasi kendaraan.

***Development***

* Arduino IDE 1.8.1
* Sublime Teks 3
* My SQL
* Android Studio 3.0

Untuk Monitoring Kendaraan untuk memonitoring tegangan baterai, arus baterai dan lokasi kendaraan.

## Sistematika Penulisan

Adapun sistematika penulisan ini yang di bagi menjadi 5 bab adalah sebagai berikut :

**BAB I PENDAHULUAN**

Menguraikan tentang latar belakang masalah, identifikasi masalah, rumusan masalah, Batasan masalah, tujuan dan manfaat penelitian, kerangka pemikiran serta sistematika penulisan dan menguraikan dasar atau alasan mengapa penelitian tentang monitoring baterai mobil ini di lakukan, dan menjelaskan sampai mana penelitian tentang baterai mobil ini akan di lakukan.

**BAB II LANDASAN TEORI**

Menjelaskan mengenai studi pustaka terhadap teori yang mendasari dan di jadikan sebagai acuaan pada penelitian tentang monitoring baterai mobil ini. Berisi juga tentang beberapa definisi dan penjelasan mengenai mikrokontroler *NodeMcu,* Tinjauan tentang sensor tegangan, Tinjauan tentang sensor arus, dan modul gps neo-6m.

**BAB III METODE PENELITIAN**

Berisi tentang bagaimana semua data yang berkaitan dengan penelitian tentang monitoring kendaraan ini didapatkan. Berisi juga penjelasan tentang analisa kebutuhan perangkat keras dan perangkat lunak.

**BAB IV HASIL DAN PEMBAHASAN**

Menjelaskan bagaimana hasil yang di dapatkan dari penelitian tentang monitoring kendaraan ini. Berisi juga tabel percobaan alat monitoring baterai ini apakah alat monitoring baterai ini bekerja sesuai dengan tujuan awal.

**BAB V KESIMPULAN DAN SARAN**

Berisi tentang kesimpulan dari penelitian tentang monitoring kendaraan ini dari awal sampai akhir dan saran yang dapat diberikan pada hasil penelitian ini.

**DAFTAR PUSTAKA**

**LAMPIRAN**

# LANDASAN TEORI

## Tinjauan Pustaka

Guna mendukung penelitian ini digunakan beberapa tinjauan pustaka yang diambil dari beberapa jurnal penelitian yang berkaitan dengan judul penelitian serta sistem yang akan di kembangkan, adalah sebagai berikut:

1. Oleh Leonandi Agustian mahasiswa Fakultas Teknik Jurusan Teknik Elektro Fakultas Teknik Universitas Tanjungpura Pontianak yang berjudul” Rancang Bangun Sistem Monitoring Kondisi Aki Pada Kendaraan Bermotor” pada jurnal ini membahas tentang alat untuk memonitoring aki kendaraan roda dua yang menggunakan Arduino. Dimana alat tersebut dapat memberikan peringatan dalam berbentuk bunyi apabila kondisi aki berkurang tegangannya dan di tampilan di layar monitor (Agustian, 2015).
2. Oleh Rianto March Siringoringo yang berjudul “Perancangan nan Implementasi Monitoring Kendaraan Bermotor Berbasis Gps dan sms” pada penelitian ini membahas tentang monitoring kendaraan dengan menggunkan modul *bluetooth* dan modul gsm sebagia pengirim data ke android pengguna .
3. Oleh Ahmad Roihan yang berjudul “Monitoring *Location Tracker* untuk Kendaraan Berbasis Rasberry Pi” yang membahas terkait sistem monitoring kendaraan dengan *Module GPS* dan Web sebagai media akses berbasis arduino (Ahmad Roihan, 2017).
4. Oleh Dony Hendra Lesmana yang berjudul “Rancang Bangun Sistem Monitoring Kecepatan Kendaraan Berbasis Gps Dengan sms Sebagai Media Pengiriman Data“ Pada jurnal ini menjelaskan tentang tentang penelitian Sistem monitoring kecepatan kendaraan dirancang dengan memanfaatkan data posisi dan kecepatan dari GPS. Hasil pembacaan data GPS kemudian ditransmisikan ke PC *server* melalui fasilitas SMS yang kemudian akan disimpan sebagai *track record* sehari-hari pengemudi (Dony Hendra Lesmana, 2013).

## Tinjauan Studi

### Definisi monitoring

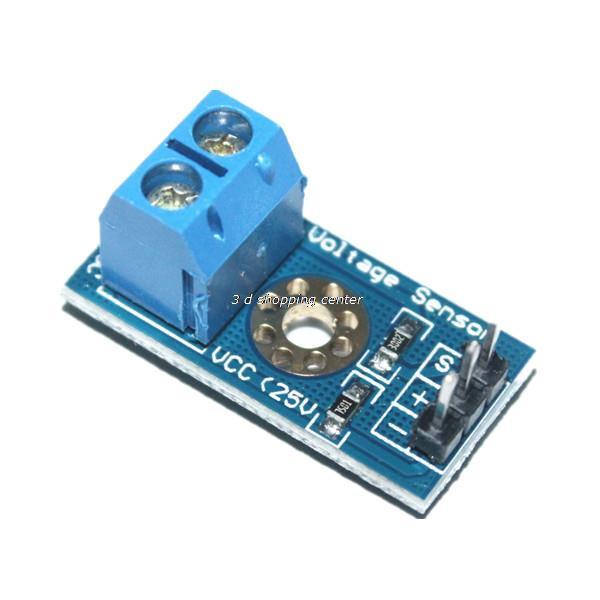
Pengertian Monitoring Menurut Cassely dan Kumar: Monitoring merupakan program yang terintegrasi, bagian penting dipraktek manajemen yang baik dan arena itu merupakan bagian integral di manajemen sehari-hari.

### Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer atau sistem komputasi yang dirancang untuk keperluan pengontrolan sistem. Mikrokontroler dilengkapi dengan CPU (Unit Pusat Pemrosesan), memori dan perangkat perantara lainnya sehingga 6 sering disebut mikro komputer serpih tunggal. Tidak seperti sistem komputer yang mampu menangani berbagai macam program aplikasi (misalnya pengolahan kata, pengolahan angka, dan lain sebagainya), mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja (hanya satu program saja yang disimpan).

### Sensor tegangan

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik disebut *Transduser.* Prinsip kerja modul sensor tegangan yaitu didasarkan pada prinsip penekanan resistansi, dan dapat membuat tegangan input berkurang hingga 5 kali dari tegangan asli. Bentuk modul sensor tegangan seperti ditunjukkan pada gambar berikut.



Gambar 2. Sensor Tegangan (Fachri, 2019)

### Sensor Arus ACS 712

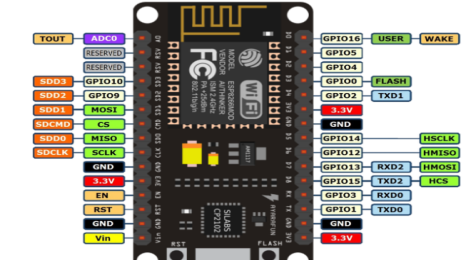
ACS712 adalah *Hall Effect* current sensor. *Hall effect allegro* ACS712 merupakan sensor yang presisi sebagai sensor arus AC atau DC dalam pembacaan arus didalam dunia industri, otomotif, komersil dan sistem-sistem komunikasi. Pada umumnya aplikasi sensor ini biasanya digunakan untuk mengontrol motor, deteksi beban listrik, switched-mode power supplies dan proteksi beban berlebih, bentuk fisik dari sensor arus ACS712 dapat dilihat pada gambar di bawah ini.



**Gambar 2. 2 Sensor Arus 712** (Fachri, 2019)

### Node Mcu

*NodeMCU* merupakan papan pengembangan produk *Internet of Things* (IoT) yang berbasiskan *Firmware* *eLua* *dan System on a Chip* (SoC) ESP8266-12E. ESP8266 sendiri merupakan chip WiFi dengan protocol stack TCP/IP yang lengkap. *NodeMCU* dapat dianalogikan sebagai board arduino-nya ESP8266. Program ESP8266 sedikit merepotkan karena diperlukan beberapa teknik wiring serta tambahan modul USB to serial untuk mengunduh program. Namun *NodeMCU* telah me-package ESP8266 ke dalam sebuah board yang kompak dengan berbagai fitur layaknya mikrokontroler + kapabilitas akses terhadap Wifi juga chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan *charging smartphone*.



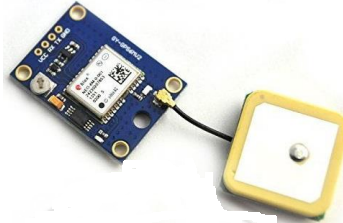
.Gambar 2. NodeMCU ESP826 dan Skema Pin

### Global Positioning System

*Global Positioning System* (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS Rusia, Galileo Uni Eropa, IRNSS India. Sistem ini dikembangkan oleh Departemen Pertahanan Amerika Serikat, dengan nama lengkapnya adalah NAVSTAR GPS (kesalahan umum adalah bahwa NAVSTAR adalah sebuah singkatan, ini adalah salah, NAVSTAR adalah nama yang diberikan oleh John Walsh, seorang penentu kebijakan penting dalam program GPS). Kumpulan satelit ini diurus oleh 50th Space Wing Angkatan Udara Amerika Serikat. Biaya perawatan sistem ini sekitar US$750 juta per tahun, termasuk penggantian satelit lama, serta riset dan pengembangan. 9 GPS Tracker atau sering disebut dengan GPS Tracking adalah teknologi AVL (Automated Vehicle Locater) yang memungkinkan pengguna untuk melacak posisi kendaraan, armada ataupun mobil dalam keadaan Real-Time. GPS Tracking memanfaatkan kombinasi teknologi GSM dan GPS untuk menentukan koordinat sebuah obyek, lalu menerjemahkannya dalam bentuk peta digital.

### GPS UBlox Neo 6M

Pada penelitian kali ini modul GPS yang digunakan adalah berjenis uBlox Neo 6M, jenis GPS ini cukup dapat diandalkan karena memiliki keakuratan yang cukup baik dan juga beberapa fitur yang cukup menguntungkan di antaranya terdapat baterai cadangan data, built-in elektronik kompas, dan built-in antena keramik untuk menangkap sinyal dengan kuat. Kemudian untuk dapat mengkomunikasikan GPS ini dengan Arduino diperlukan sebuah library yang bernama “TinyGPS++.h”. Bentuk dari modul GPS uBlox Neo 6M dapat dilihat pada Gambar 2.4 di bawah ini:



**Gambar 2. 4 GPS uBlox Neo 6M** (Sanjaya, 2019)

### Arduino Software IDE

IDE itu merupakan singkatan dari *Integrated Developtment Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (Sketch) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama Bootlader yang berfungsi sebagai penengah antara compiler Arduino dengan mikrokontroler.

### Menulis Sketch

Program yang ditulis dengan menggunaan Arduino Software (IDE) disebut sebagai sketch. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi .ino. Teks editor pada Arduino Software memiliki fitur” seperti copy/cut/paste dan searching/replacing sehingga memudahkan dalam menulis kode program. Pada Software Arduino IDE, terdapat semacam message box berwarna hitam yang berfungsi menampilkan status, seperti pesan error, compile, 13 dan upload program. Di bagian bawah paling kanan Sotware Arduino IDE, menunjukan board yang terkonfigurasi beserta Port COM yang digunakan.



**Gambar 2. 5 Arduino IDE**

Kemudian fungsi dari beberapa tools yang terdapat pada toolbar di antaranya adalah sebagai berikut:

1. *Verify,* berfungsi untuk melakukan checking kode yang dibuat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum.

2. *Upload*, berfungsi untuk melakukan kompilasi dan mengunggah kode program yang dibuat ke dalam Bootloader Arduino untuk dijalankan.

3. *New*, berfungsi untuk membuat Sketch baru.

4. *Open*, berfungsi untuk membuka Sketch yang pernah dibuat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke Arduino.

5. *Save*, berfungsi untuk menyimpan Sketch yang telah dibuat.

6. Serial Monitor, berfungsi untuk membuka serial monitor. Serial monitor disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara Arduino dengan sketch pada port serial-nya. Serial Monitor ini berguna sekali ketika ingin membuat program atau melakukan debugging tanpa menggunakan LCD pada Arduino. Serial monitor 14 ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan error.

### Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler. Android menyediakan platform terbuka bagi para pengembang untuk meciptakan aplikasi mereka sendiri yang digunakan oleh bermacam piranti bergerak. Awalnya android dikembangkan oleh Android, Inc. dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Hansdet Alliance, konsorium dari perusahaan-perusahaan perangkat keras, perangkat lunak dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Sudah banyak platform untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), dan banyak lagi. Namun ada beberapa hal yang 12 menjadi kelebihan Android. Beberapa keungguan android dengan platform lain yaitu:

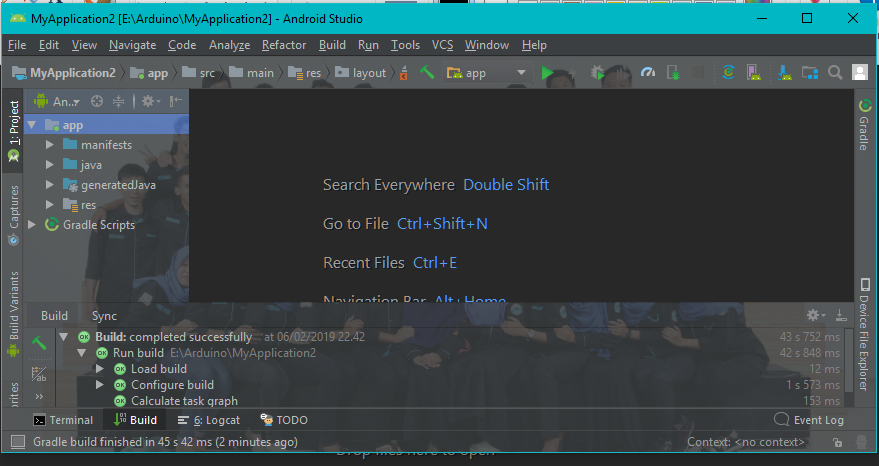
1. Keterbukaan, Android menyediakan akses ke fungsi dasar perangkat mobile menggunakan standar panggilan ke API.

2. Penghancuran perbatasan, anda dapat menggabungkan informasi dari internet ke dalam telepon, seperti informasi kontak, atau data pada lokasi geografis.

3. Cepat dan mudah perkembangannya, dalam SDK memiliki semua yang anda butuhkan untuk membuat dan menjalankan aplikasi Android, termasuk simulator dan alat debugging.

### Android Studio

Studio Android Studio adalah sebuah IDE untuk pengembangan aplikasi di platform Android. Sama seperti kombinasi antara Eclipse dan Android Developer Tools (ADT), Android Studio juga dapat di-download di situs resmi Android: http://developer.android.com/sdk/installing/studio.html. Saat ini usia Android Studio masih tergolong muda, baru versi 1.3.1 (masih early access preview). Sebelum adanya Android Studio ini, SDK sebelumnya dibundle bersama dengan Eclipse, sementara Android Studio menggunakan IntelliJ IDEA Community Edition. Beberapa pendukung IntelliJ IDEA sering mengatakan bahwa Eclipse terlalu rumit bagi pemula. Perbedaan lain nya yakni Android Studio menggunakan Gradle untuk manajamen proyeknya. Gradle adalah build automation tool yang dapat dikonfigurasikan melalui DSL berbasis Groovy. Ini yang membedakan Gradle dari Ant atau Maven yang memakai XML. Penggunaan DSL berbasis Groovy menyebabkan Gradle lebih fleksibel dan dapat diprogram dengan mudah.



**Gambar 2. 6 Android Studio**

# METODE PENELITIAN

## Metode Pengumpulan Data

Metode pengumpulan data merupakan tahap paling penting dalam sebuah penelitian. Karena pada tahap ini peneliti menjelaskan cara pengumpulan informasi serta apa yang dilakukan untuk informasi tersebut. Pada penelitian ini Ada beberapa metode yang digunakan untuk mendapatkan data/informasi diantaranya:

### Observasi

Pada metode pengumpulan data ini penulis melakukan observasi secara langsung untuk mendapat data mengenai aki atau baterai mobil dan motor. Penulis melakukan pengecekan tegangan aki baterai mobil yang masih dalam keadaan normal secara langsung. Dari observasi yang dilakukan pada tanggal 23 November 2018 didapat data sebagai berikut:

Tabel 3. 1 Observasi lapangan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | Jenis kendaraan | Waktu pengecekan tegangan baterai atau aki | |  |
| Mesin mobil menyala | Mesin mobil mati |
| 1 | Sedan/Honda jazz | 13,7 V | 12,4 V |
| 2 | Sedan/Honda civic | 14,0 V | 13,0 V |
| 3 | Mini bus/Kijang LGX | 13,8 V | 12,3 V |
| 4 | Sedan/Toyota vios | 13,8 V | 12.3 V |
| 5 | Mini bus/Avanza | 14,5 V | 12,5 V |

Dari pengecekan tegangan baterai atau aki 5 mobil pada tabel di atas maka dapat didapatkan data sebagi berikut: Aki atau baterai mobil dapat dikategorikan normal apabila mesin kendaraan/mobil dalam kondisi mati tegangannya 12-13 *Volt* dan apabila mesin kendaraan menyala tegangan baterai atau akinya 13-14 *Volt* .

Tabel 3. 2 Observasi lapangan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | Jenis kendaraan | Waktu pengecekan tegangan baterai atau aki | |  |
| Mesin motor menyala | Mesin motor mati |
| 1 | Honda beat | 13,7 V | 12,4 V |
| 2 | Mio m3 | 13,0 V | 12,0 V |
| 3 | X-ride | 13,8 V | 12,3 V |
| 4 | N-max | 13,7 V | 12.3 V |
| 5 | Mioj | 13,5 V | 12,5 V |

### Studi Pustaka

Pada metode pengumpulan data yang kedua penulis mengacu pada berbagai jurnal penelitian yang sudah pernah ada, dan beberapa artikel untuk mendapatkan sebuah data baterai/aki mobil dapat dikatakan normal, lemah dan *over*(pengisian baterai berlebihan).

1. Baterai/aki dapat dikatakan normal apabila dalam kondisi mesin mobil mati tegangan baterai/akinya 12,4.
2. Baterai/aki dapat dikatakan normal apabila dalam kondisi mesin mobil/motor menyala tegangan baterai/akinya 13,4-14,8 V.
3. Baterai/aki dikatakan normal apabila teganganya 12,8 V(*volt*) dan dan baterai dikatakan lemah apabila tegangannya kurag dari 12 V (*volt* ).
4. Baterai/aki dapat dikatakan normal apabila tegangannya 12,2-12,6 V (*volt*) dan dikatakan lemah apabila tegangannya 9-11,2V(*volt* ).
5. Baterai/aki normal apabila tegangannya 13-13,2 V (*volt*) dan lemah apabilanya tegangannya kurang dari 12 V(*volt*), dan apabila kondsis mesin menyala tegangannya 14,0-15,0 V(*volt*) ().

Dari dua metode penelitian diatas maka dapat diambil sebuah kesimpulan bahwa baterai/aki mobil dapat dikatakan lemah apabila tegangannya kurang dari 12 V (*volt).* Baterai/aki mobi dikatakan normal apabila pada saat kondisi mesin mati tegangannya 12-13 V (*volt*) dan pada saat mesin mobil menyala tegangannya 13-14 V (*volt*). Dan apabila saat kondisi mesin menyala tegangannya lebih dari 15 V(*volt*) maka kondisi pengisian baterai/aki *over* (melebihi kapasitas semestinya). Untuk lebih jelasnya seperti **Tabel 3.3** di bawah ini:

Tabel 3. 3 Monitoring tegangan baterai motor dan mobil.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Kategori monitoring | Tegangan/volt baterai/akinya | |
| Mesin motor mati | Mesin motor menyala |
| 1 | Lemah | <12 V | - |
| 2 | Normal | 12-13 V | 13-14V |
| 3 | Over (Pengisian baterai melebihi kapasitas) | - | >15 V |

## Analisa kebutuhan perangkat keras dan Perangkat lunak

### Kebutuhan perangkat keras

Pada analisa kebutuhan perangkat keras ini di butuh beberapa perangkat keras diantaranya sebagai berikut:

1. Dua buah mikrokontroler jenis Node Mcu: yang digunakan sebagai pengontrol Alat monitoring kendaraan.
2. Sensor tegangan yang berfungsi sebagai pengukur *volt* baterai yang nantinya data *volt* baterai akan di baca oleh Node Mcu.
3. Satu buah modul wifi yang berfungsi sebagia pengiriman data sensor dari *nodemcu* ke database mysql.
4. Laptop dengan spesifikasi: minimal prosesor intel inside ram 4 gb dengan sistem operasi windows 7.
5. Sensor arus ACS712 untuk :yang berguna sebagai pengukur arus baterai yang nantinya data arus aki akan di baca oleh Node Mcu.
6. Satu buah modul gps neo-6m yang berfungsi sebagai pembaca *latitude* dan *longitude.*
7. Satu buah relay 5 V yang berfungsi saklar *on/off* mesin kendaraan.

### Kebutuhan perangkat lunak

Pada analisa kebutuhan perangkat lunak ini di butuh beberapa perangkat lunak diantaranya sebagai berikut:

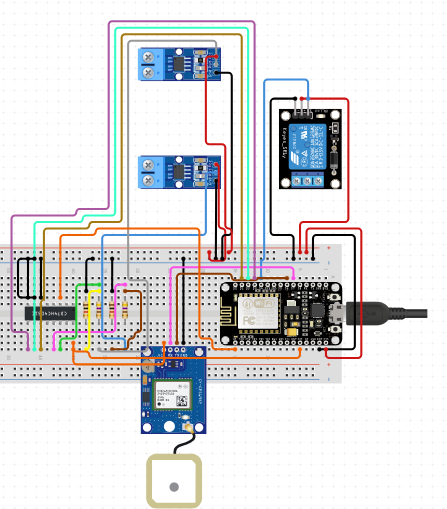
1. Arduino IDE sebagai software dimana penulis dapat mbemasukan kode perintah kedalam mikrokontroler dengan bahasa pemograman C.
2. Windows 10 sebagai sistem operasi yang digunakan untuk menginstal Arduino IDE.
3. Android IDE untuk mendesain aplikasi android.
4. Sublime text sebagai teks editor untuk membuat api android dan database mysql.
5. Xampp v3.2.1.

## Analisa Perancangan Sistem Monitoring Kendaraan

Untuk membuat sebuah sistem yang berjalan dengan baik sesuai harapan yang diinginkan maka tentunya terlebih dahulu membuat tahapan perancangan sistem yang akan di buat, berikut adalah tahapan rancangan sistem yang di bangun.

### Desain Perancangan Alat

Dibawah ini merupakan gambar perancangan alat monitoring kendaraan adapun perangkat keras yang di gunakan adalah *NodeMcu,* relay, Sensor arus dan tegangan, dan modul gps.



Gambar 3. Desain Pearancangan Alat

### Desain Alur kerja Sistem

Sensor tegangan membaca tegangan baterai , sensor arus membaca arus baterai dan Modul gps neo-6m membaca *latitude* dan *longitude* lalu *nodemcu* memproses data tersebut lalu dikirim kan ke databasa mysql. Setelah semua data sensor tegangan, sensor arus, *latitude* dan *longitude* terkirim ke database mysql lalu website dan android menampilkan data tersebut.

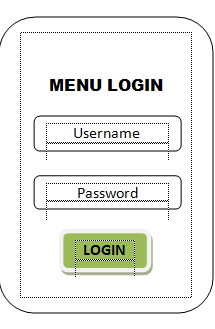
|  |  |  |
| --- | --- | --- |
| **INPUT** | **PROSES** | **OUTPUT** |
| **SENSOR ARUS**  **MODUL GPS NEO-6M**  **SENSOR TEGANGAN** | **DATABASE**  **NODE MCU** | **WEBSITE**  **ANDROID** |

Gambar 3. Desain Alur kerja Sistem

### Desain Aplikasi Android

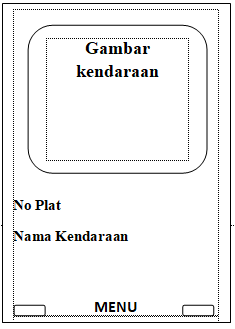
Dibawah ini adalah perancangan *interface* aplikasi mnitoring kendaraan dimana pada aplikasi tersebut memiliki beberapa menu, diantaranya sebagai berikut:

1. Pada gambar di bawah ini di merukan merukan perancangan tampilan menu login aplikasi.



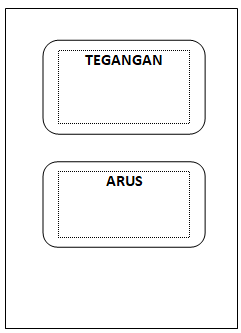
Gambar 3. Perancangan menu login

1. Pada gambar di bawah ini merukan perancangan tampilan menu setelah login berhasil.



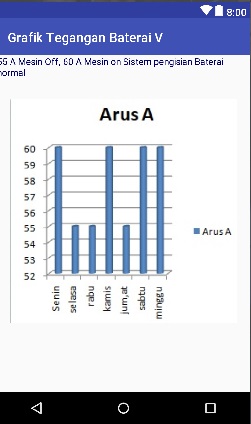
Gambar 3. Perancangan Menu profil

1. Pada gambar di bawah ini merukan perancangan tampilan menu monitoring.



Gambar 3. Perancangan Menu monitoring

1. Pada gambar di bawah ini merukan perancangan tampilan menu grafik



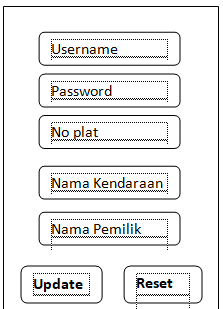
Gambar 3. Perancangan menu grafik

1. Pada gambar di bawah ini merukan perancangan tampilan menu loakai



Gambar 3. Perancangan Menu lokasi

1. Pada gambar di bawah ini merukan perancangan tampilan menu ubah data user.



Gambar 3. Perancangan ubah data user

### Flowchart Sistem

Dibawah ini merupan perancangan alur sistem aplikasi monitoring kendaraan yang di mulai dari login aplikasi, menu profil, menu grafik, menu monitoring, menu lokasi, kunci, buka kunci, menu ubah data pengguna, menu ubah foto profil. Adapun desain alurnya adalah sebagai berikut:



Gambar 3. Perancangan *Flowchart System*

### Desain Struktur database

Database yang digunakan adalah database mysql, MySQL adalah sistem manajemen database yang digunakan untuk  menyimpan data dalam tabel terpisah dan menempatkan semua data dalam satu gudang besar. Struktur database disusun dalam file fisik dioptimalkan untuk kecepatan. Model logis, dengan benda-benda seperti database, tabel, baris, dan kolom, menawarkan lingkungan pemrograman yang fleksibel. Pada database ini memiliki empat(4) tabel yaitu tabel user, tabel volt, tabel lokasi dan tabel arus.

**Tabel 3. 4 Perancangan Tabel user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| No-plat | Varchar | 12 | Input data no-plat |
| Username | Varchar | 20 | Input data username |
| password | Varchar | 10 | Input data password |
| Nama-kendaraan | Varchar | 50 | Input data nama-kendaraan |
| Nama-pemilik | Varchar | 20 | Input data nama-pemilik |
| Foto | Varchar | 50 | Input data nama foto-kendaraan |

**Tabel 3. 5 Perancangan Tabel volt**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| volt | Varchar | 12 | Input data volt baterai |
| tanggal | Date-time | 8 | Input data tanggal |

**Tabel 3. 6 Perancangan Tabel arus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| arus | Varchar | 12 | Input data arus baterai |
| tanggal | Date-time | 8 | Input data tanggal |

**Tabel 3. 7 Perancangan Tabel lokasi**

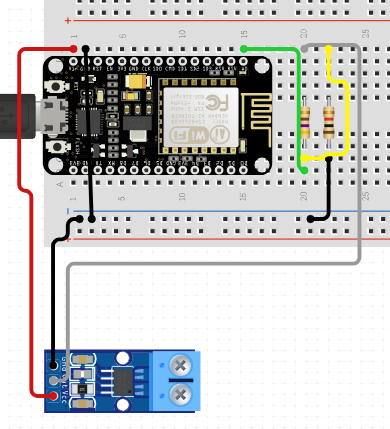
|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| Latitude | Varchar | 50 | Input data latitude |
| longitude | Varchar | 50 | Input data longitude |
| tanggal | Date-time | 8 | Input data tanggal |

# HASIL DAN PEMBAHASAN

## Instalasi Alat

### Rangkaian *Node Mcu* dan sensor tegangan

Dibawah ini merupakan rangkaian *NodeMcu* dan sensor tegangan, yang berfungsi untuk mendeteksi *volt*/tegangan baterai. Sensor tegangan mendeteksi berapa tegangan baterai lalu data tegangan baterai tersebut dibaca oleh *NodeMcu* dan *NodeMcu* memproses data tegangan tersebut untuk dikirim ke database Mysql.



Gambar 4. Node Mcu Sensor tegangan

Tabel 4. 1 Tabel rangkaian node mcu sensor tegangan

|  |  |  |
| --- | --- | --- |
| No | Pin pada sensor tegangan | Pin pada node mcu |
| 1 | Vcc | 5 V |
| 2 | Out | A0 |
| 3 | Gnd | Gnd |

### *Soure code* Node Mcu dan sensor tegangan

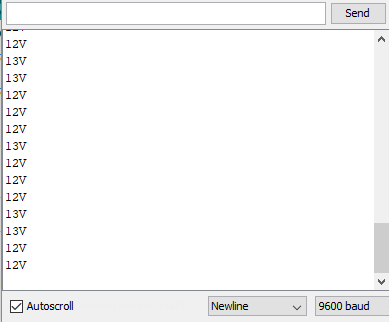
Dibawah ini merupakan *Soure code* dengan bahasa pemgraman C yang berfungsi untuk membaca tegangan baterai.

|  |
| --- |
| Int Volt1;  int Volt;  void setup()  {   Serial.begin(9600);    Serial.println("Voltage: ");   } void loop()  {       Volt1=analogRead(0);        Volt=((Volt1\*0.00489)\*5);        Serial.print(Volt);  Serial.println("V");      delay(1000); } |

### Hasil di Serial monitor Arduino IDE

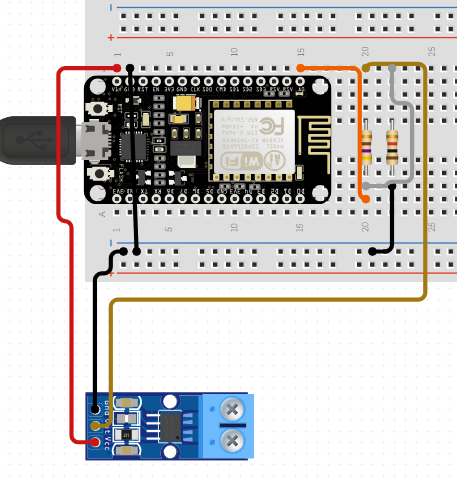
Dibawah ini adalah data tegangan baterai yang tampil pada serial monitor arduino ide.

Gambar 4. Serial monitor tegangan



### Rangkaian Node Mcu dan sensor asus ACS712

Dibawah ini merupakan rangkaian *NodeMcu* dan sensor arus ACS712, yang berfungsi untuk mendeteksi *ampere*/arus baterai. sensor arus ACS712 mendeteksi berapa *ampere*/arus baterai lalu data *ampere*/arus baterai tersebut dibaca oleh *NodeMcu* dan *NodeMcu* memproses data tegangan tersebut untuk dikirim ke database Mysql.



Gambar 4. Rangkaian Node Mcu dan sensor asus ACS712

Tabel 4. 2 Tabel rangkaian node mcu sensor tegangan

|  |  |  |
| --- | --- | --- |
| No | Pin pada sensor tegangan | Pin pada node mcu |
| 1 | Vcc | 5 V |
| 2 | Out | A0 |
| 3 | Gnd | Gnd |

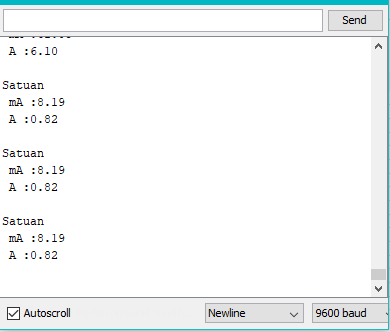
### Source code node mcu dan ACS 712

Dibawah ini merupakan *Soure code* dengan bahasa pemgraman C yang berfungsi untuk membaca arus baterai.

|  |
| --- |
| void setup() {  Serial.begin(9600); } void loop() {  //Sensor Arus max 5A  float average = 0;  for(int i = 0; i < 1000; i++) {  average = average + (0.0264 \*  analogRead(A0) -13.51);  delay(1000); } Serial.print(average); Serial.println("mA"); } |

### Hasil di Serial monitor Arduino IDE Node Mcu Node Mcu dan Sensor Arus

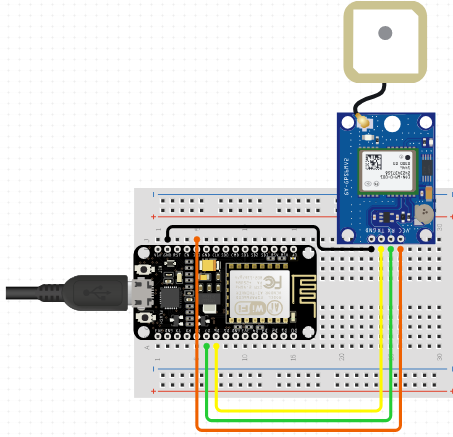
Dibawah ini adalah data tegangan baterai yang tampil pada serial monitor arduino ide.



Gambar 4. Hasil di serial monitor arduino ide

### Node Mcu dan modul Gps Neo-6m

Dibawah ini merupakan rangkaian *NodeMcu* dan modul Gps Neo-6m, yang berfungsi untuk mendeteksi *Latitude* dan *longitude*. modul Gps Neo-6m memabaci *Latitude, longitude*. lalu data *Latitude, longitude* tersebut dibaca oleh *NodeMcu* dan *NodeMcu* memproses data *Latitude, longitude* tersebut untuk dikirim ke database Mysql.



Gambar 4. Rangakaian Node Mcu dan modul Gps Neo-6m

Tabel 4. 3 Tabel rangkaian node mcu dan modul gps neo-6m

|  |  |  |
| --- | --- | --- |
| No | Pin pada modul gps neo-6m | Pin pada node mcu |
| 1 | Vcc | 3.3 V |
| 2 | Rx | D7 |
| 3 | Tx | D6 |
| 4 | Gnd | Gnd |

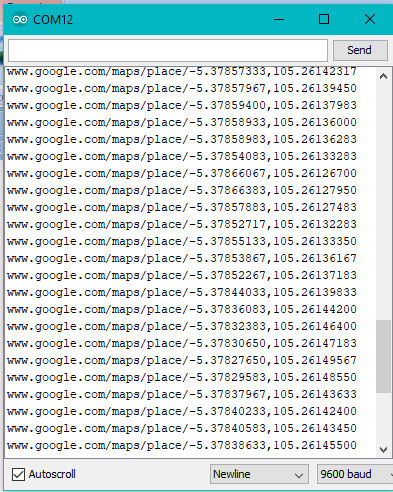
### Source code node mcu dan modul gps neo-6m

Dibawah ini merupakan *Soure code NodeMcu* dengan bahasa pemgraman C yang berfungsi untuk membaca *latitude* dan *longitude* dimana modul gps neo-6m itu berada .

|  |
| --- |
| #include <TinyGPS++.h>  #include <SoftwareSerial.h>  SoftwareSerial serial\_gps(D1, D2);  TinyGPSPlus gps;  double latitude, longitude;  void setup() {  Serial.begin(9600);  serial\_gps.begin(9600);  Serial.println("GPS Mulai");  }  void loop() {  while(serial\_gps.available()) {  gps.encode(serial\_gps.read());  }  if(gps.location.isUpdated()) {  latitude = gps.location.lat();  longitude = gps.location.lng();  String link = "www.google.com/maps/place/" + String(latitude,8) + "," + String(longitude,8) ;  Serial.println(link);  delay(5000);  }  } |

### Hasil di serial monitor Arduino IDE

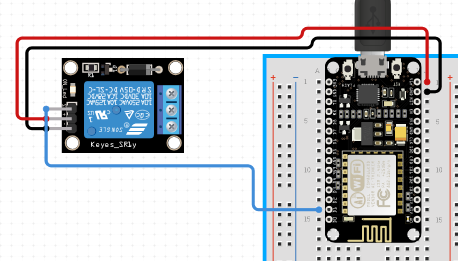
Dibawah ini adalah data *latitude* dan *longitude* yang tampil pada serial monitor arduino ide.



Gambar 4. Hasil Latitude dan longitude di serial monitor arduin ide

### Node Mcu dan Relay

Dibawah ini merupakan rangkaian *NodeMcu* dan relay, yang berfungsi sebagai saklar on/off mesin kendaraan.



Gambar 4. Node Mcu dan Relay

Tabel 4. 4 Tabel rangkaian node mcu dan Relay

|  |  |  |
| --- | --- | --- |
| No | Pin pada Relay | Pin pada node mcu |
| 1 | Vcc | 3.3 V |
| 2 | IN | D4 |
| 3 | GND | GND |

### Source code Node Mcu kontrol Relay

Dibawah ini merupakan Source code *NodeMcu* dan relay, yang berfungsi sebagai saklar on/off mesin kendaraan.

|  |
| --- |
| #include <dummy.h>  #include <ESP8266WiFi.h>  #include <ArduinoJson.h>  #include <Stepper.h>  const char\* ssid = "ARDUINO NUSANTARA";  const char\* password = "02091995";  const char\* host = "monitoringaki.000webhostapp.com"; // Your domain  String path = "/sim900/light.json";  const int pin = D4;  void setup() {  WiFi.config(ip, gateway, subnet);  delay(100);  //checking for SHIELD  if (WiFi.status() != WL\_NO\_SHIELD) {  Serial.println("WiFi shield is present");  while(true); // don't continue  }  WiFi.mode(WIFI\_STA);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  Serial.print(".");  delay(500);  }  pinMode(pin, OUTPUT);  pinMode(pin, HIGH);  Serial.begin(115200);  //delay(10);  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.begin(ssid, password);  int wifi\_ctr = 0;  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("WiFi connected");  Serial.println("IP address: " + WiFi.localIP());  }  void loop() {  Serial.print("connecting to ");  Serial.println(host);  WiFiClient client;  const int httpPort = 80;  if (!client.connect(host, httpPort)) {  Serial.println("connection failed");  return;  }  client.print(String("GET ") + path + " HTTP/1.1\r\n" +  "Host: " + host + "\r\n" +  "Connection: keep-alive\r\n\r\n");  delay(500); // wait for server to respond  // read response  String section="header";  while(client.available()){  String line = client.readStringUntil('\r');  // Serial.print(line);  // we’ll parse the HTML body here  if (section=="header") { // headers..  Serial.print(".");  if (line=="\n") {  section="json";  }  }  else if (section=="json") { // print the good stuff  section="ignore";  String result = line.substring(1);  // Parse JSON  int size = result.length() + 1;  char json[size];  result.toCharArray(json, size);  StaticJsonBuffer<200> jsonBuffer;  JsonObject& json\_parsed = jsonBuffer.parseObject(json);  if (!json\_parsed.success())  {  Serial.println("parseObject() failed");  return;  }  // Make the decision to lock and unlock  if (strcmp(json\_parsed["light"], "off") == 0) {  digitalWrite(pin, LOW);  Serial.println("LED OFF");  }  else {  digitalWrite(pin, HIGH);  Serial.println("LED ON");  }  }  }  Serial.print("closing connection. ");  } |

## Aplikasi Android

### Database

Tabel 4. 5 Tabel user

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| No-plat | Varchar | 12 | Input data no-plat |
| Username | Varchar | 20 | Input data username |
| password | Varchar | 10 | Input data password |
| Nama-kendaraan | Varchar | 50 | Input data nama-kendaraan |
| Nama-pemilik | Varchar | 20 | Input data nama-pemilik |
| Foto | Varchar | 50 | Input data nama foto-kendaraan |

Tabel 4. 6 Tabel volt

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| volt | Varchar | 12 | Input data volt baterai |
| tanggal | Date-time | 8 | Input data tanggal |

Tabel 4. Tabel arus

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| arus | Varchar | 12 | Input data arus baterai |
| tanggal | Date-time | 8 | Input data tanggal |

Tabel 4. 8 Tabel lokasi

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| Id | Int | 12 | Auto increment |
| Latitude | Varchar | 50 | Input data latitude |
| longitude | Varchar | 50 | Input data longitude |
| tanggal | Date-time | 8 | Input data tanggal |

### Menu Splash Screen

*Splash screen* merupakan tampilan tambahan yang muncul saat pertama kali ki membuka suatu aplikasi. Pada aplikasi ini *Splash screen* nya adalah seperti gambar di bawah ini:



Gambar 4. Splash Screen

### Menu Login

Menu *login* merupakan halaman yang pertama muncul pada saat aplikasi pertama dimulai. Berikut ini adalag gambar menu login. Pada gambar menu login dibawah memiliki dua *textinput, textinput* pertama berfungsi untuk memasukan nama pengguna aplikasi dan *textinput* kedua berfungsi untuk memasukan kata sandi pengguna aplikasi dan satu *button* untuk melakukan login masuk ke layout atau m enu selanjutnya.



Gambar 4. Menu Login

### Menu Profil

Menu profil merupakan halaman yang kedua muncul pada pengguna aplikasi berhasil *login*. Berikut ini adalag gambar menu profil. Pada gambar menu profil dibawah ini memiliki dua *textview, textinput* pertama berfungsi untuk menampilkan nomor plat kendaraan dan *textview* kedua berfungsi untuk menampilkan nama kendaraan pengguna dan 3 *button* berfungsi untuk beralih ke menu atau layout selanjutnya.



Gambar 4. Menu profil

### Menu Utama

Pada aplikasi ini memiliki sebuah *layout* untuk menu utama yang berisisi beberapa *button*. Pada gambar menu utama dibawah memiliki 7 *button* adapun penjelesannya sebagai berikut:

1. *Button* pengguna : bergambar logo *user* cara menggunakan adalah dengan mengklik dan berfungsi untuk menuju menu ubah data pengguna .
2. Button lokasi : bergambar logo *location* cara menggunakan adalah dengan mengklik dan berfungsi untuk menuju menu lokasi kendaraan .
3. Button Kamera : bergambar logo *camera* cara menggunakan adalah dengan mengklik dan berfungsi untuk menuju menu ubah photo profile .
4. Button volt baterai : Bertulisan volt baterai cara menggunakan adalah dengan mengklik dan berfungsi untuk menuju menu cek tegangan baterai .
5. Button Kunci : Bertulisan kunci cara menggunakan adalah dengan mengklik dan berfungsi untuk mengaktifkan anti *starter* pada kendaraan.
6. Button Arus : Baterai Bertulisan arus baterai cara menggunakan adalah dengan mengklik dan berfungsi untuk menuju menu cek arus baterai .
7. Button Buka kunci : Bertulisan kunci cara menggunakan adalah dengan mengklik dan berfungsi untuk menonaktifkan anti *starter* pada kendaraan.

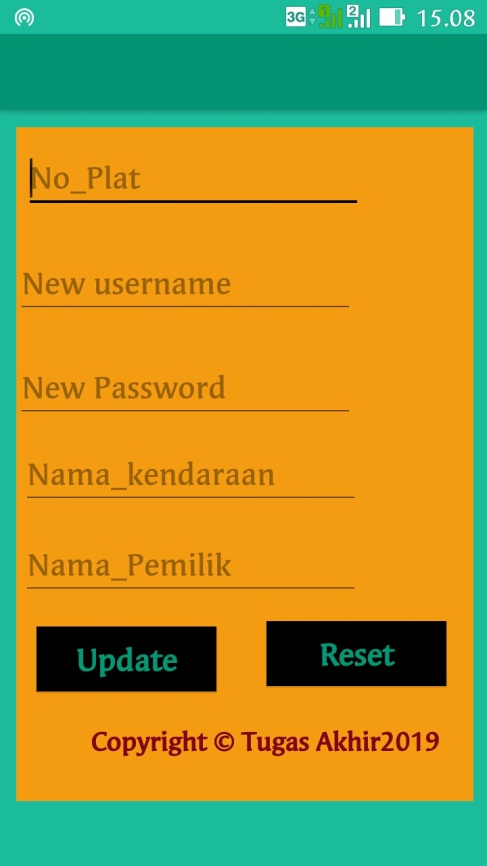
Menu utama pada aplikasi ini adalah sebagai berikut:



Gambar 4. Menu utama

### Menu Ubah Data Pengguna

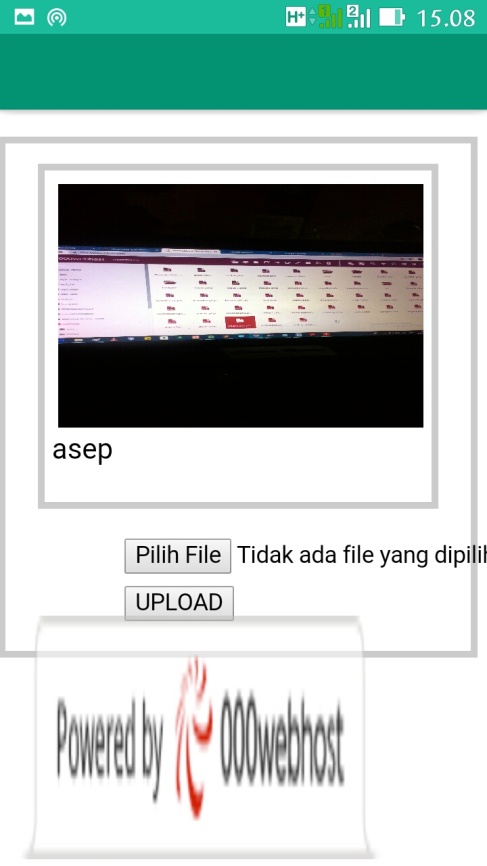
Menu ubah data pengguna adalah menu yang berfungsi untuk merubah data pengguna aplikasi seperti nama pengguna, kata sandi dan lain sebagainya. Menu ubah data pengguna dibawah Memiliki 5 *textinput* yang berfungsi untuk menginputkan data baru kendaraan maupun data baru pengguna aplikasi. Dua buah *button* yang berfungsi sebagai *button update* data dan hapus *form* menu ubah data. Adapun gambar menu ubah data pengguna adalah sebagai berikut:



Gambar 4. Menu ubah data pengguna

### Menu Ubah Photo Profile

Menu Ubah Photo Profile adalah menu untuk merubah foto profile kendaraan Pada Menu Ubah Photo Profile memiliki 3 buah komponen yakni *imageview*, *button* pilih gambar, dan *button* *upload* gambar. Adapun gambar Menu Ubah Photo Profile pada aplikasi ini adalah sebagai berikut:

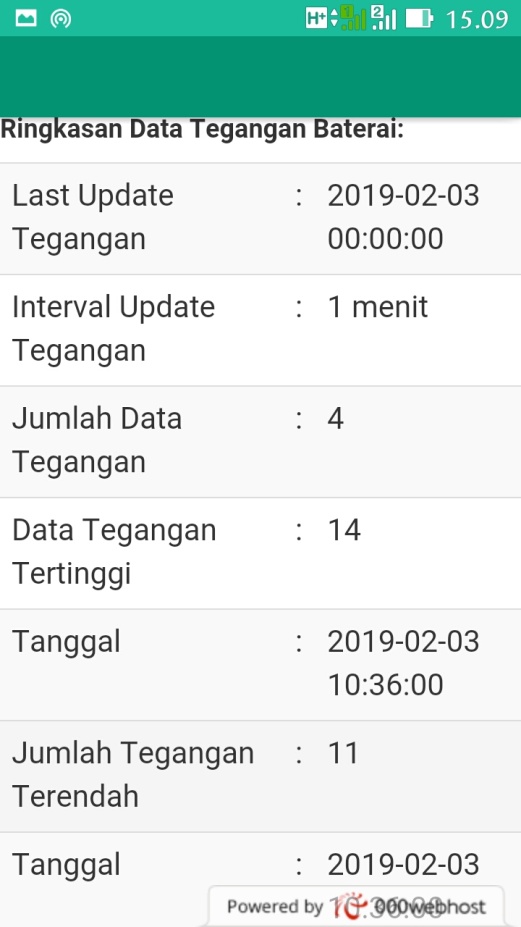
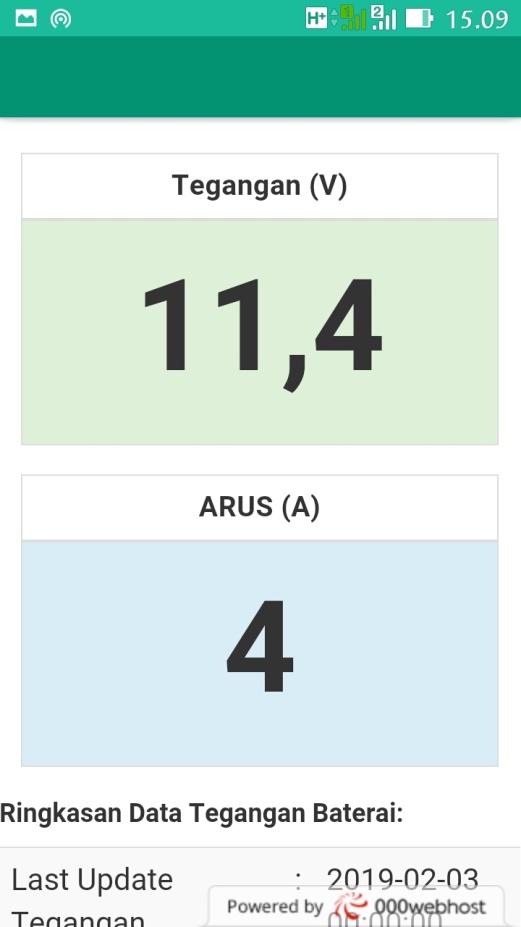


Gambar 4. Menu Ubah Photo Profile

.

### Menu Monitoring Tegangan dan Arus Baterai

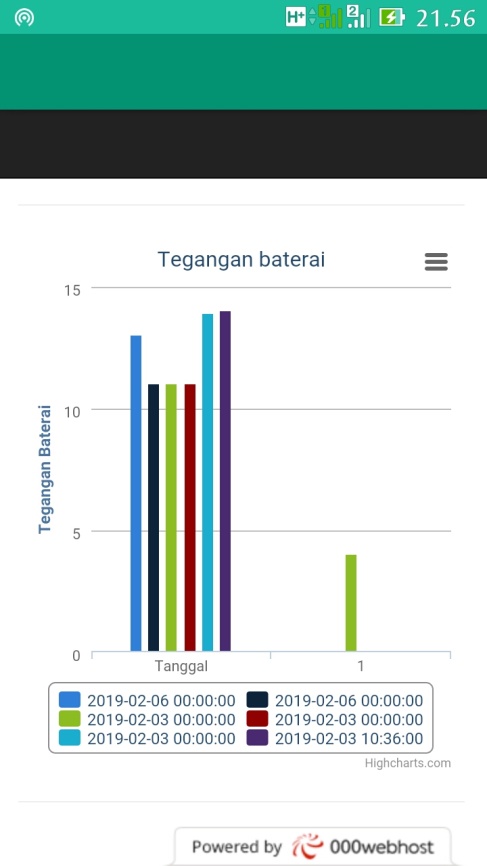
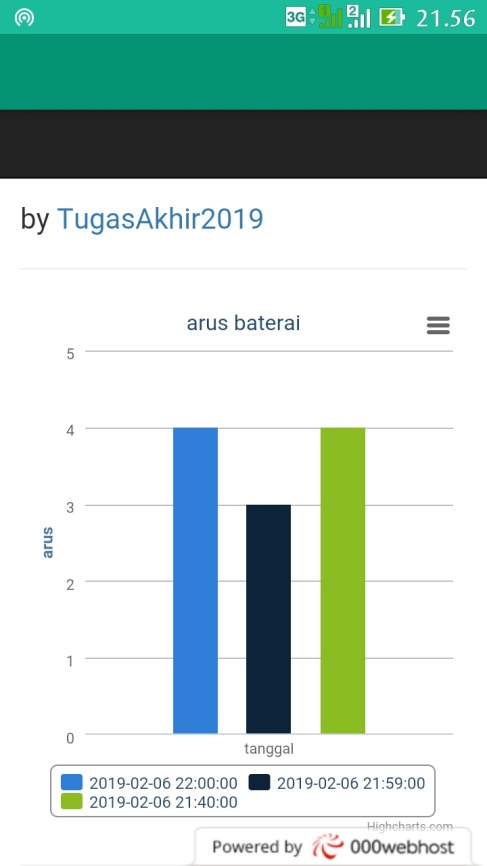
Pada menu berfungsi sebagai menu untuk melihat data tegangan, data arus dan data detail tegangan arus baterai seperti halnya kapan data terakhir masuk, kapan data tegangan dan arus tertinggi dan terendah adapun gambar Menu Monitoring Tegangan dan Arus Baterai adalah sebagai berikut:



Gambar 4. Menu Monitoring Tegangan dan Arus Baterai

### Menu Grafik

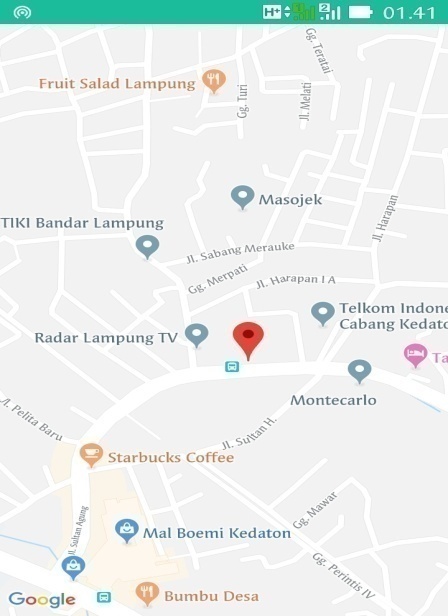
Menu grafik adalah menu untuk memonitoring berapa tegangan dan arus baterai pada jangka waktu tertentu dan untuk memonitoring bagaimana keadaan baterai pada jangka waktu terdekat. Pada menu ini juga pengguna dapat melihat analisis kondisi baterai apakah masih dalam kondisi bagus atau mulai melemah. Dibawah ini adalah gambar menu monitoring pada aplikasi monitoring kendaraan ini:



Gambar 4. Menu monitoring

### Menu Maps Kendaraan

Menu lokasi kendaraan merupakan halaman yang muncul ketika *button* lokasi yang berada di menu utama di klik. Pada gambar menu lokasi di atas terdapat satu *marker* terahir kendaraan berada. Dan pada menu ini pengguna aplikasi dapat melacak di mana letak kendaraan apabila kendaraan hilang. Di bawah ini merupakan gambar menu lokasin kendaraan.



Gambar 4. Menu Maps Kendaraan

## Pengujian Aplikasi*.*

1. Black Box Testing

Pada tahapan ini penulis melakukan pengujian sistem untuk memeriksa apakah suatu perangkat lunak yang di hasilkan sudah dapat di jalankan sesuai dengan standar tertentu. Dibawah ini merupakan tabel hasil pengujian *Black box testing* .

Tabel 4. *Black box testing*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NO** | **Skenario Pengujian** | | **Aksi** | **Hasil yang**  **Diharapkan** | **Hasil yang Didapat** | **Status** |
| 1 | Tombol LOGIN  pada menu Login masuk | | Klik | Pengguna masuk ke Menu Profile | Pengguna masuk ke Menu Profile | Berhasil |
| 2 | Tombol menu  pada menu profile | | Klik | Pengguna masuk ke Menu utama | Pengguna masuk ke Menu utama | Berhasil |
| 3 | Tombol lokasi  pada menu utama | | Klik | Pengguna masuk ke menu maps kendaraan terakhir berada | Pengguna masuk ke menu maps  kendaraan terakhir berada | Berhasil |
| 4 | Tombol user  pada menu utama | | Klik | Pengguna masuk ke menu ubah data pengguna | Pengguna masuk ke menu ubah data pengguna | Berhasil |
| 5 | Tombol kamera  pada menu utama | | Klik | Pengguna masuk ke menu ubah foto pengguna | Pengguna masuk ke menu ubah foto pengguna | Berhasil |
| 6 | Button Reset pada menu ubah user | | Klik | Menghapus form inputan pada menu ubah data pengguna | Menghapus form inputan pada menu ubah data pengguna | Berhasil |
| 7 | Tombol monitoring  pada profile | Klik | | Pengguna masuk ke menu  monitoring | Pengguna masuk ke menu monitoring | Berhasil |
| 8  9  1 | Tombol grafik  pada menu profile | Klik | | Pengguna masuk ke men grafik | Pengguna masuk ke men grafik | Berhasil |

1. White Box Testing

Pengujian yang didasarkan pada detail prosedur dan alur logika kode program. *Whitebox* testing pengujian yang dilakukan sampai kepada detail pengecekan kode program. Berikut merupakan hasil *whitebox* testing pada aplikasi yang telah dibuat :

1. Flowchart white box testingpada Aplikasi monitoring kendaraan



Gambar Gambar 4. *Flowchart white box testing*

1. Flowgraph

Pada pengujian *white box testing*



Gambar 4. *Flowgrap*

1. Cyclomatic Complexity

Berikut ini adalah hasil perhitungan dari *cyclomatic complecity* untuk melihat kompleksitas aplikasi pada penelitian ini :

Jumlah Region = 8

E = Jumlah Busur/Link

N= Jumlah Simpul

P = Jumlah Predikat

V(G) = E – N + 2

V(G) = 28 – 22+ 2 = 8

V(G) = P + 1

V(G) = 7 + 1 = 8

1. Independent Path

Independent Path digunakan untuk mengetahui jalur-jalur dari sistem pada penelitian ini. adapun alur sistem berjalan pada aplikasi ini dapat dilihat urutan angka-angka berikut ini:

|  |  |  |
| --- | --- | --- |
| Path | Node | Proses |
| 1 | 1-2-3-4-5-22 | Proses menampilkan data tegangan dan arus baterai |
| 2 | 1-2-3-4-6-7-22 | Proses menampilkan grafik tegangan dan arus baterai |
| 3 | 1-2-3-4-6-8-9-22 | Proses menampilkan lokasi kendaraan |
| 4 | 1-2-3-4-6-8-10-11-22 | Proses on anti starter kendaraan |
| 5 | 1-2-3-4-6-8-10-12-13-22 | Proses *off* anti starter kendaraan |
| 6 | 1-2-3-4-6-8-10-12-14-15-16-22 | Proses *update* data pengguna baru |
| 7 | 1-2-3-4-6-8-10-12-14-17-18-19-22 | Proses ubah foto pfofile pengguna |
| 8 | 1-2-3-4-6-8-10-12-14-17-20-21-22 | Proses keluar dari aplikasi |

## Uji Validasi Alat Monitoring Kendaraan

Pada uji validasi ini penulis langsung mengimplementasikan Alat monitoring kendaraan pada motor YAMAHA Mio M3 tahun 2015. Pada validasi ini didapatkan beberapa hasil diantaranya sebagai berikut:

Pada pengujian **Tabel 4.10** dibawah ini dilakukan pada saat kondisi mesin *off*.

Tabel 4. 10 Tabel Penngujian satu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Tegangan** | **Arus** | **Waktu uji** | |
| **Tanggal** | **Jam** |
| 1 | 12.34 V | 2.60 A | 17/02/2019 | 21.34 |
| 2 | 12.34 V | 2.60 A | 17/02/2019 | 21.36 |
| 3 | 12.33 V | 2.65 A | 17/02/2019 | 21.39 |
| 4 | 12..33 V | 2.60 A | 17/02/2019 | 21.42 |

Pada pengujian **Tabel 4.11** dibawah ini dilakukan pada saat kondisi mesin *on*.

Tabel 4. 11 Tabel pengujian validasi dua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Tegangan** | **Arus** | **Waktu uji** | |
| **Tanggal** | **Jam** |
| 1 | 13.95 V | 2.60 A | 17/02/2019 | 23.50 |
| 2 | 13.96 V | 2.60 A | 17/02/2019 | 23.54 |
| 3 | 14.02 V | 2.60 A | 17/02/2019 | 22.02 |
| 4 | 14.03 V | 2.60 A | 17/02/2019 | 22.05 |

# KESIMPULAN DAN SARAN

## Kesimpilan

Berdasarkan hasil dan pembahasan dapat disimpulkan bahwa penelitian yang penulis lakukan telah mencapai tujuan sebagai berikut:

1. Alat dapat mendeteksi tegangan, arus baterai dan data tegangan, arus dapat di lihat di android pemilik kendaraan.
2. Aplikasi dapat menampilkan letak kendaraan dalam bentuk maps.
3. Anti *stater* kendaraan dapat di aktifkan dan di nonaktifkan dari android pemilik kendaraan untuk mencegah pencurian kendaraan.

## Saran

Dari hasil pembahasan dan kesimpulan penelitian ini dapat dimbil saran yang dapat digunakan sebagai acuan untuk pengembangan berikutnya, diantaranya:

1. Alat dan aplikasi monitoring kendaraan ini masih menggunakan database mysql sehingga data yang terima oleh aplikasi android tidak *real time*, disarankan untuk memakai database seperi monggo db, firebase dan lain sebagainya.
2. Modul gps neo-6m tidak dapat mendeteksi latitude dan longitude pada lokasi yang tertutup oleh gedung.
3. Sensor arus ACS712 tidak dapat membaca arus baterai secara akurat, sehingga di sarankan untuk menggunakan sensor arus dengan tipe selain ACS712.

**DAFTAR PUSTAKA**

Agustian, L. (2015). Rancang Bangun Sistem Monitoring Kondisi Aki Pada Kendaraan Bermotor. *1*, 1-5.

Ahmad Roihan, M. S. (2017). Monitoring Location Tracker untuk Kendaraan Berbasis Rasberry Pi. *3*.

Dony Hendra Lesmana, M. R. (2013). RANCANG BANGUN SISTEM MONITORING KECEPATAN KENDARAAN BERBASIS GPS DENGAN SMS SEBAGAI MEDIA PENGIRIMAN DATA. *Jurnal Dony Hendra Lesmana,* , 1-13.

Hasanah, A. L. (2019). *NESABA MEDIA*. Dipetik january 19, 2019, dari https://www.nesabamedia.com/pengertian-gps/

Mochadnan. (2013). *Moch Adnan Blog*. Dipetik oktober 4, 2017, dari Moch Adnan Blog: http://www.mochadnan.com

Naziq, A. (2017, Oktober). *Informasi Anyar*. Dipetik january 19, 2019, dari https://sites.google.com/site/informasiterbarusekali/pengertian-mikrokontroller

Putra, A. (2018). *Mobilupdate*. Dipetik agustus 20, 2018, dari http://mobilupdate.net

Sanjaya, A. (2019). Dipetik Februari 17, 2019, dari Global Techno Solution: http://blog.adisanjaya.com/2016/12/mengakses-modul-gps-neo6mv2-neo-6m.html

Siringoringo, R. M. (2015). PERANCANGAN DAN IMPLEMENTASI MONITORING KENDARAAN BERBASIS GPS DAN SMS. *e-Proceeding of Applied Science* *, 1*, 868.

Vei, A. (2018, mei 16). *Ralali news*. Dipetik agustus 19, 2018, dari https://news.ralali.com

Wardana, K. (2018, maret). *TUTORIAL] Menggunakan Multiplekser pada ESP8266 (NodeMCU)*. Dipetik Januari 10, 2019, dari https://tutorkeren.com/artikel/tutorial-menggunakan-multiplekser-pada-esp8266-nodemcu.htm

Wardana, K. (2018, maret). *TUTORIAL] Menggunakan Multiplekser pada ESP8266 (NodeMCU)*. Dipetik Januari 10, 2019, dari

https://tutorkeren.com/artikel/tutorial-menggunakan-multiplekser-pada-esp8266-nodemcu.htm

**LAMPIRAN**