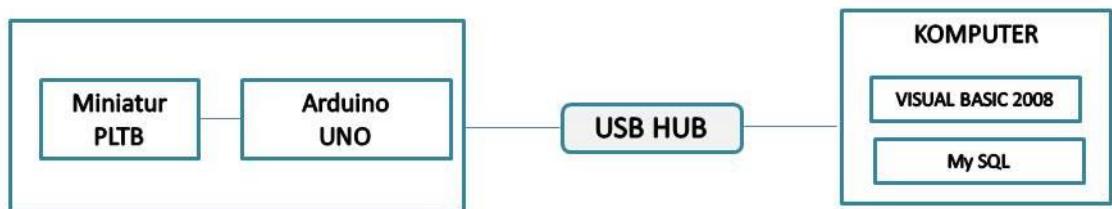


BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian yang dilakukan oleh fariz yang berjudul “*Prototype Aplikasi Logmeter Pada Pembangkit Listrik Tenaga Bayu*”, jurusan S 1 Teknik Infromatika STT-PLN, 2014, mempunyai perancangan sistem yang berjalan seperti dibawah ini :



Gambar 2.1 Perancangan Sistem Transmisi Data PLTB

Dari gambar 2.1 menjelaskan sebuah sistem yang akan bekerja untuk memonitoring data PLTB (Pembangkit Listrik Tenaga Bayu/Angin) sehingga data dapat ditransmisikan ke sistem aplikasi yang berjalan. *Prototype PLTB* sebagai sumber data tegangan dan arus, kemudian Arduino Uno akan membaca data yang didapat dari pembangkit berupa arus dan tegangan dan selanjutnya data akan dikirimkan melalui USB ke aplikasi yang dibangun menggunakan *Visual Basic 2008*. Kemudian data tersebut otomatis disimpan dalam *database* yang telah disediakan. (2014 : 43).

Dari hasil penelitian yang telah dilakukan oleh fariz diatas, Penulis melakukan analisa terhadap penelitian tersebut. Dari hasil analisa, Penulis rasa perlu adanya pengembangan pada penelitian pada sistem yang berjalan. Pengembangan di lakukan baik pada sisi *hardware* maupun

software. Dari sisi *hardware* penulis berharap pengembangan yang dilakukan dapat membaca dua pembangkit sekaligus atau disebut dengan pembangkit *hybrid* khususnya menggunakan tenaga angin dan surya. Selain itu penulis mengembangkan dalam hal jalur komunikasi data. Pada penelitian sebelumnya hanya menggunakan kabel usb dan pada penelitian yang akan dikembangkan menggunakan *GPRS Module*. Dari sisi *software* penulis mengembangkan aplikasi dalam memonitoring data pembangkit yaitu menggunakan aplikasi android. Sehingga dalam pembacaan data dapat dilakukan dimana saja.

2.2 Landasan Teori

2.2.1 Android

A. Pengertian Android

Secara umum pengertian Android adalah suatu *software* (perangkat lunak) yang berbasis Linux untuk telepon seluler dan komputer tablet yang meliputi sistem operasi, middleware dan aplikasi inti. Android SDK menyediakan alat dan API yang diperlukan untuk memulai pengembangan aplikasi pada platform. Android menggunakan bahasa pemrograman Java, yaitu kode Java yang terkompilasi dengan data dan file *resources* yang dibutuhkan aplikasi dan digabungkan oleh aapt tools menjadi paket Android. File tersebut ditandai dengan ekstensi .apk. File inilah yang didistribusikan sebagai aplikasi dan diinstall pada perangkat mobile. (Nazrudin, 2012)

B. Sejarah Android

Android Inc. adalah sebuah perusahaan *software* kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan yang berbasis IT & Communication yaitu Andy Rubin, Rich Miner, Nick

Sears dan Chris White. Hingga akhirnya pada Agustus 2005 Google Inc. membeli Android Inc. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

C. Arsitektur Android

Google mengibaratkan Android sebagai sebuah tumpukan *software*. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini susunan dari lapisan – lapisan tersebut jika dilihat dari lapisan dasar hingga lapisan teratas.

D. Linux Kernel

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android, yang mencakup memory management, security setting, power management, dan beberapa driver *hardware*. Kernel berperan sebagai abstraction layer antara *hardware* dan keseluruhan *software*. Sebagai contoh, HTC G1 dilengkapi dengan kamera. Kernel Android terdapat driver kamera yang memungkinkan pengguna mengirimkan perintah kepada *hardware* kamera.

E. Android Runtime

Lapisan setelah Kernel Linux adalah Android Runtime. Android Runtime ini berisi *Core Libraries* dan *Dalvik Virtual Machine*. *Core Libraries* mencakup serangkaian inti *library* Java, artinya Android menyertakan satu set *library* - *library* dasar yang menyediakan sebagian besar fungsi - fungsi yang ada pada *library* - *library* dasar bahasa pemrograman Java. *Dalvik* adalah Java

Virtual *Machine* yang memberi kekuatan pada sistem Android. Dalvik VM ini di optimalkan untuk telepon seluler.

Setiap aplikasi yang berjalan pada Android berjalan pada processnya sendiri, dengan instance dari Dalvik Virtual *Machine*. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi Virtual *Machine* dengan efisien. Dalvik VM dapat mengeksekusi file dengan format Dalvik *Executable* (.dex) yang telah dioptimasi untuk menggunakan minimal memory footprint. Virtual *Machine* ini register-based, dan menjalankan class-class yang dicompile menggunakan compiler Java yang kemudian ditransformasi menjadi format .dex menggunakan "dx" tool yang telah disertakan. Dalvik Virtual *Machine* (VM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti threading dan low-level memory management.

F. Application Framework

Lapisan selanjutnya adalah application framework, yang mencakup program untuk mengatur fungsi-fungsi dasar *smartphone*. Application Framework merupakan serangkaian tool dasar seperti alokasi resource *smartphone*, aplikasi telepon, pergantian antar - proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool-tool dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

Programmer mendapatkan akses penuh untuk memanfaatkan API-API (Android Protocol Interface) yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen-komponen, setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut.

Mekanisme yang sama memungkinkan pengguna mengganti komponen-komponen yang dikehendaki. Di dalam

semua aplikasi terdapat servis dan sistem yang meliputi : Satu set Views yang dapat digunakan untuk membangun aplikasi meliputi lists, grids, text boxes, buttons, dan embeddable web browser, Content Providers yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain (misalnya Contacts), atau untuk membagi data yang dimilikinya. Resource Manager, menyediakan akses ke non-code resources misalnya localized strings, graphics, dan layout files, notification Manager yang memungkinkan semua aplikasi untuk menampilkan custom alerts pada the status bar, activity Manager yang memanage life cycle of dari aplikasi dan menyediakan common navigation backstack.

G. Application

Di lapisan teratas bercokol aplikasi itu sendiri. Di lapisan inilah anda menemukan fungsi-fungsi dasar *smartphone* seperti menelepon dan mengirim pesan singkat, menjalankan web browser, mengakses daftar kontak, dan lain-lain. Bagi rata-rata pengguna, lapisan inilah yang paling sering mereka akses.

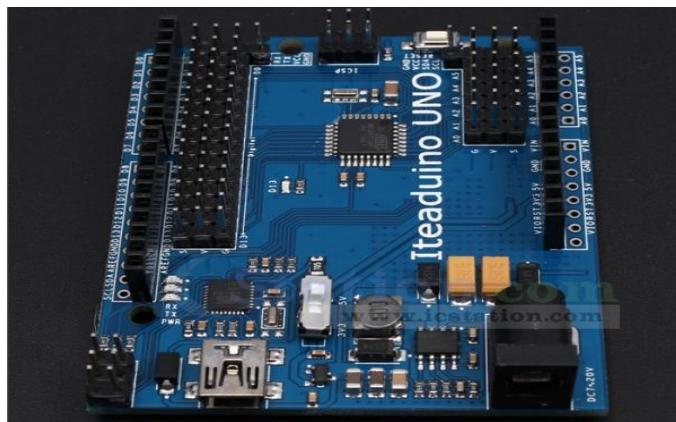
2.2.2 Arduino

A. Pengertian

Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel.

Mikrokontroler itu sendiri adalah *chip* atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi

mikrokontroler bertugas sebagai ‘otak’ yang mengendalikan *input*, proses dan *output* sebuah rangkaian elektronik.



Gambar 2.2 Arduino UNO

Mikrokontroler ada pada perangkat elektronik di sekeliling kita. Misalnya *handphone*, *MP3 player*, *DVD*, televisi, AC, dll. Mikrokontroler juga dipakai untuk keperluan mengendalikan robot. Baik robot mainan, maupun robot industri.

Karena komponen utama Arduino adalah *mikrokontroler*, maka Arduino pun dapat diprogram menggunakan komputer sesuai kebutuhan kita. (Setiawan, Sulhan, 2008).

B. Kelebihan Arduino

- Tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani upload program dari komputer.
- Sudah memiliki sarana komunikasi *USB*, sehingga pengguna Laptop yang tidak memiliki port serial/RS323 bisa menggunakannya.
- Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.
- Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board Arduino*. Misalnya *shield GPS*, *Ethernet*, *SD Card*, dll.

C. Bahasa Pemrograman Arduino

Bahasa pemrograman Arduino adalah bahasa C. Tetapi bahasa ini sudah dipermudah menggunakan fungsi-fungsi yang sederhana sehingga pemula pun bisa mempelajarinya dengan cukup mudah. Untuk membuat program Arduino dan mengupload ke dalam board Arduino, anda membutuhkan *software* Arduino IDE (*Integrated Development Enviroment*).

2.2.3 GSM/GPRS Module

Arduino GPRS Module adalah sebuah board yang dirancang terintegrasi dengan arduino dengan fungsi untuk dapat mengirim sms, membuat *voice call* atau mengkoneksi internet dengan menggunakan *wireless network*.

Arduino GPRS Module memungkinkan arduino untuk terkoneksi dengan internet, melakukan panggilan suara dan mengirim/menerima sms dengan menggunakan radio modem M10 dari Quectel, yang dapat memungkinkan komunikasi dengan arduino dengan menggunakan AT commands.

GPRS Module ini menggunakan pin digital 2 dan 3 untuk komunikasi serial *software* dengan M10. Pin 2 terkoneksi dengan M10 sehingga dapat berfungsi sebagai pin TX dan pin 3 dapat berfungsi sebagai pin RX.

M10 adalah Quadband GSM/GPRS modem yang bekerja pada frekuensi GSM850MHz, GSM900MHz, DCS1800MHz dan PCS1900MHz. M10 juga membantu TCP/UDP dan protokol HTTP melalui koneksi GPRS. Kecepatan transfer data GPRS *Downlink* dan *Uplink* maksimal adalah 85.6 kbps.



Gambar 2.3 GPRS Module

Untuk melakukan *interface* dengan jaringan seluler, *GPRS Module* membutuhkan *SIM card* yang disediakan oleh operator jaringan.

2.2.4 Pembangkit Listrik *Hybrid*

Cara kerja Pembangkit Listrik Sistim *Hybrida* Surya Angin dan Diesel sangat tergantung dari bentuk beban atau fluktuasi pemakain energi (*load profile*) yang mana selama 24 jam distribusi beban tidak merata untuk setiap waktunya. Load profil ini sangat dipengaruhi penyediaan energinya. Untuk mengatasi permasalahan tersebut maka kombinasi sumber energi antara Sumber energi terbarukan dan Diesel Generator atau disebut Pembangkit Listrik Sistem Hibrida adalah salah satu solusi paling cocok untuk sistem pembangkitan yang terisolir dengan jaringan yang lebih besar seperti jaringan PLN. (Irawan, 2012)



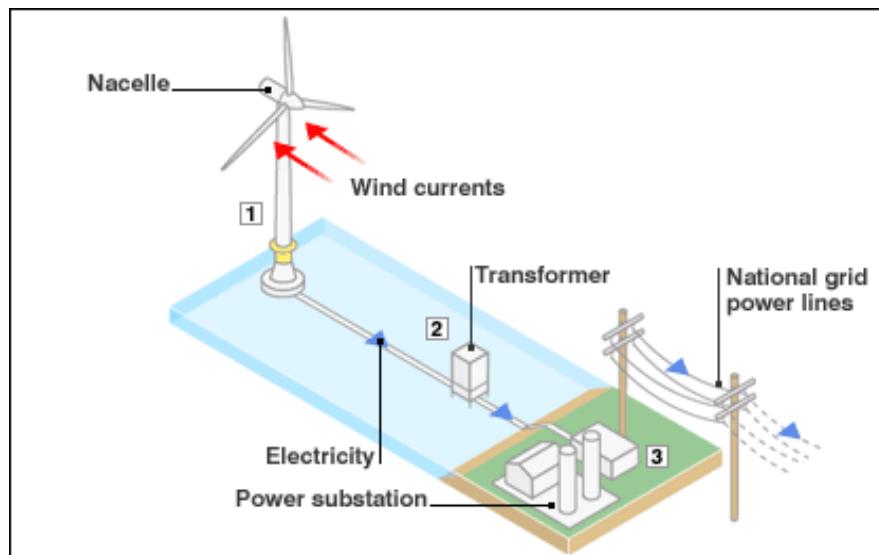
Gambar 2.4 Pembangkit Listrik *Hybrid* Tenaga Angin dan Surya

A. Pembangkit Listrik Tenaga Angin

Suatu pembangkit listrik dari energi angin merupakan hasil dari penggabungan dari beberapa turbin angin sehingga akhirnya dapat menghasilkan listrik.

Cara kerja dari pembangkitan listrik tenaga angin ini yaitu awalnya energi angin memutar turbin angin. Turbin angin bekerja berkebalikan dengan kipas angin (bukan menggunakan listrik untuk menghasilkan listrik, namun menggunakan angin untuk menghasilkan listrik). Kemudian angin akan memutar sudut turbin, lalu diteruskan untuk memutar rotor pada generator di bagian belakang turbin angin. Generator mengubah energi gerak menjadi energi listrik dengan teori medan elektromagnetik, yaitu poros pada generator dipasang dengan material ferromagnetik permanen. Setelah itu di sekeliling poros terdapat stator yang bentuk fisinya adalah kumparan-kumparan kawat yang membentuk loop. Ketika poros generator mulai berputar maka akan terjadi perubahan fluks pada stator yang akhirnya karena terjadi perubahan fluks ini akan

dihadirkan tegangan dan arus listrik tertentu. Tegangan dan arus listrik yang dihasilkan ini disalurkan melalui kabel jaringan listrik untuk akhirnya digunakan oleh masyarakat. Energi Listrik ini biasanya akan disimpan kedalam baterai sebelum dapat dimanfaatkan. (C. Brower, Michael, 2012).

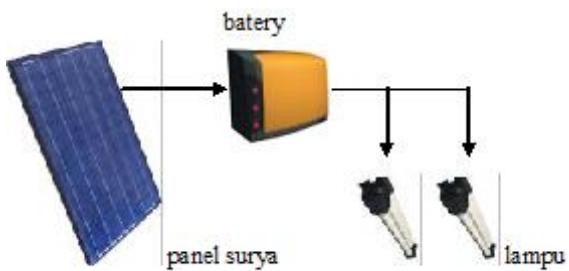


Gambar 2.5 Distribusi Listrik Kerja PLTB

B. Pembangkit Listrik Tenaga Surya

Pembangkit Listrik Tenaga Surya (PLTS) di Indonesia, paling populer digunakan untuk listrik pedesaan (terpencil), sistem seperti ini populer dengan sebutan SHS (Solar Home System). SHS umumnya berupa sistem berskala kecil, dengan menggunakan modul surya 50-100 Wp (Watt Peak) dan menghasilkan listrik harian sebesar 150-300 Wh. Karena skalanya yang kecil, sistem DC (direct current) lebih disukai, untuk menghindari losses dan self consumption akibat digunakannya inverter.

Konfigurasi SHS seperti diagram dibawah ini:



Gambar 2.6 Prinsip Kerja Panel Surya

Karena sistemnya yang kecil dan dipasang secara desentralisasi (satu rumah satu pembangkit, sehingga tidak memerlukan jaringan distribusi) SHS ideal digunakan untuk listrik di pedesaan dimana jarak rumah satu dengan lainnya berjauhan, dan keperluan listriknya relatif kecil, yakni hanya untuk memenuhi kebutuhan dasar (lampa). Meskipun secara pengertian SHS dapat saja berupa system yang besar (sejauh masih digunakan untuk listrik rumah), namun kebanyakan orang cenderung tidak menggunakan istilah SHS untuk system yang menggunakan modul lebih besar dari 100Wp (atau produksi energi harian >400Wh)

Meskipun belum ada batasan yang jelas, PLTS yang menggunakan modul surya lebih dari 100Wp (Output energi >400Wh), dan oleh karenanya lebih memungkinkan digunakan system AC (Alternating current; karena listrik yang dapat digunakan setelah dikurangi losses dan self consumption inverter masih cukup memadai), dalam tulisan ini, termasuk dalam kategori PLTS skala menengah-besar.

Keunggulan terpenting dari penggunaan PLTS adalah:

1. Tergolong kedalam sumber energy terbarukan dan ramah lingkungan.
2. Tidak memerlukan biaya maintenance dan biaya operasi.
3. Memiliki umur teknis lebih dari 30 tahun.

2.2.5 XAMPP

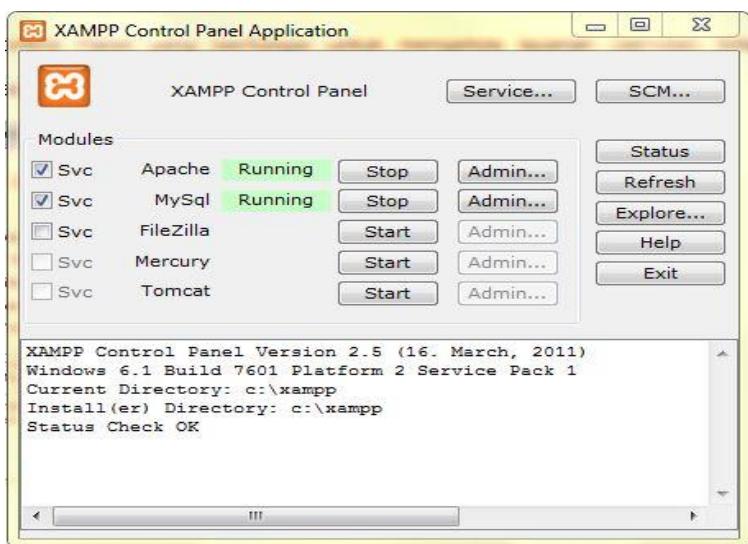
Xampp adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mendownload langsung dari web resminya.

Mengenal bagian XAMPP yang biasa digunakan pada umumnya:

1. **htdoc** adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
2. **phpMyAdmin** merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka browser lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman phpMyAdmin.
3. **Kontrol Panel** yang berfungsi untuk mengelola layanan (service) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*). (Betha Sidiq, ir. 2006)



Gambar 2.7 Xampp Control Panel

2.2.6 PHP

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi PHP adalah forum (phpBB) dan MediaWiki (*software* di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke,Xaraya, dan lain-lain. (Betha Sidiq, ir. 2006)

2.2.7 MySQL

adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. PHP MyAdmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, anda dapat membuat database, membuat tabel, menginsert, menghapus dan mengupdate

data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual. PHPMyAdmin dapat di download secara free di <http://www.phpmyadmin.net>. Karena berbasis web, maka phpmyadmin dapat di jalankan di banyak sistem operasi, selama dapat menjalankan webserver dan Mysql. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-

perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional. (R.H. Sianipar 2013)

2.2.8 JSON

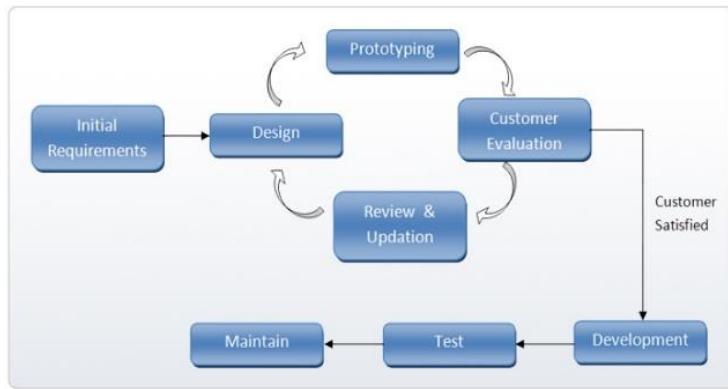
JSON adalah struktur data yang universal, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON

Format data JSON mempunyai aturan sebagai berikut : *Object* adalah satu set nama-nilai yang tidak terurut. Penulisan object di mulai dengan tanda { (*left brace*) dan di akhiri dengan tanda } (*right brace*). Setiap nama diikuti oleh tanda : (colon) dan pasangan nama-nilai dipisahkan dengan tanda , (comma). (http://www.academia.edu/10502935/92Gia_Jurnal.HTML Retrieved April 2016)

2.2.9 Metode *Prototype*

Prototyping merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Seiring terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dikehendakinya tanpa menyebutkan secara detail *output* apa saja yang dibutuhkan, pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang memperhatikan efisiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer. Untuk mengatasi ketidakserasan antara pelanggan dan pengembang, maka harus dibutuhkan kerjasama yang baik diantara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan pelanggan dengan tidak mengesampingkan segi-segi teknis dan pelanggan akan mengetahui proses-proses dalam menyelesaikan sistem yang diinginkan. Dengan demikian akan menghasilkan sistem sesuai dengan jadwal waktu penyelesaian yang telah ditentukan.

Kunci agar model *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa *prototype* dibangun untuk mendefinisikan kebutuhan. *Prototype* akan dihilangkan sebagian atau seluruhnya dan perangkat lunak aktual aktual direkayasa dengan kualitas dan implementasi yang sudah ditentukan. (<http://www.varia.web.id/2013/06/prototyping-model.html> Retrieved April 2016).



Gambar 2.8 Prototype Model

➤ **Tahapan-tahapan *Prototyping***

Tahapan-tahapan dalam *Prototyping* adalah sebagai berikut:

1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun *prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format *output*).

3. Evaluasi *prototyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulangi langkah 1, 2 , dan 4. Mengkodekan sistem, dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai

4. Coding

Dalam pengkodean, hal yang dilakukan adalah membuat coding program yang digunakan dalam *prototype* ini.

5. Menguji sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.

6. Menggunakan sistem dan perawatan Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan dan dilakukan perawatan secara berkala guna menajaga agar system dapat berjalan dengan normal.

➤ **Keunggulan dan Kelemahan *Prototyping***

Keunggulan *prototyping* adalah:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
3. Pelanggan berperan aktif dalam pengembangan sistem
4. Lebih menghemat waktu dalam pengembangan sistem
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Kelemahan *prototyping* adalah :

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat

prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem .

3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik.

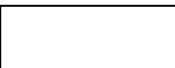
2.2.10 Bagan Alir (*Flowchart*)

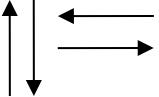
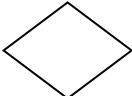
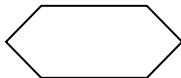
Bagan alir (*flowchart*) adalah bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir. *Flowchart* adalah bagan (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur sistem secara logika. Diagram alir popular pada awal-awal era pemrograman dengan komputer. Diagram alir lebih menggambarkan aliran instruksi didalam program secara visual daripada memperlihatkan struktur program. Notasi algoritma ini juga cocok untuk masalah kecil, tidak cocok untuk masalah besar karena akan memerlukan berlembar halaman kertas untuk menggambarkan aliran proses program yang akan dibuat. (<http://ndoware.com/diagram-alir-flowchart.html> Retrieved Juli 2016).

➤ **Simbol-simbol yang Digunakan**

Berikut ini adalah simbol-simbol *flowchart* :

Tabel 2.1 Simbol-simbol dalam *flowchart*

No	Gambar	Nama	Keterangan
1.		Simbol input/output (<i>input/output symbol</i>)	Digunakan untuk mewakili data input/output
2.		Simbol proses	Digunakan untuk mewakili suatu proses

3.		Simbol garis alir (flow line symbol)	Digunakan untuk menunjukkan arus dari proses
4.		Simbol penghubung (connector symbol)	Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lainnya
5.		Simbol keputusan (decision symbol)	Digunakan untuk suatu penyeleksian kondisi di dalam program
6.		Simbol proses terdefinisi (predefined process symbol)	Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain
7.		Simbol persiapan (preparation symbol)	Digunakan untuk memberi nilai awal suatu besaran
8.		Simbol terminal (terminal point symbol)	Digunakan untuk menunjukkan awal dan akhir dari suatu proses

2.2.11 Metode Pengujian *Black Box*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternative dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi – fungsi yang tidak benar atau hilang,
2. Kesalahan Interface,
3. Kesalahan dalam struktur data atau akses database eksternal,
4. Kesalahan Kinerja,
5. Inisialisasi dan kesalahan terminasi

Dengan mengaplikasikan teknik *black-box*, maka kita menarik serangkaian test case yang memenuhi criteria berikut ini :

1. Test case yang mengurangi, dengan harga lebih dari satu, jumlah test case tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan,
2. Test case yang member tahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada. (Pressman (2002:551))

2.2.12 Data Logger

Data logger merupakan alat yang digunakan untuk merekam data, yakni perangkat elektronik yang dipakai untuk mencatat data dari satu waktu ke waktu selanjutnya, perangkat ini dihubungkan dengan sebuah sensor. Atau bisa juga dikatakan bahwa data logger adalah alat yang digunakan untuk data logging. Apa itu data logging? Data logging merupakan sebuah

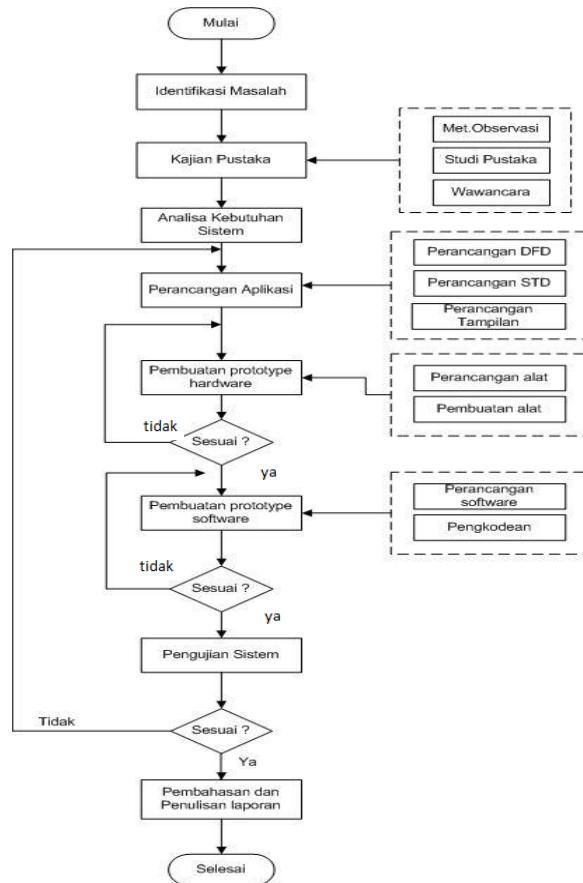
proses pengumpulan dan perekaman data yang berjalan secara otomatis dari sebuah sensor untuk kepentingan analisa maupun pengarsipan. Sensor yang digunakan sesuai dengan kebutuhan. Berbagai jenis sensor yang biasa dipakai adalah suhu, gerak, kecepatan angin, tingkat suara, kelembaban udara, pH, posisi, sudut rotasi, oksigen terlarut dan masih banyak lagi yang lainnya.

Secara fisik data logger berukuran kecil, biasanya bersifat portable dan hanya menggunakan tenaga baterai. Perangkatnya dilengkapi dengan mikroprosesor dan memori internal yang digunakan untuk mencatat dan merekam data dan sensor. Beberapa jenis *data logger* biasanya dikoneksikan dengan computer dan untuk mengaktifkannya digunakan sebuah software yang lebih simple. Pengamatan terhadap data yang terekam bisa dilakukan melalui computer. Ada juga jenis data logger yang memiliki keypad dan LCD sendiri, sehingga bisa dipakai sebagai perangkat yang terpisah dari computer. Terdapat banyak jenis data logger di pasaran yang bisa Anda beli dan sesuaikan dengan kebutuhan yang ada.

Penggunaan dari data logger tentu saja sangat membantu mempermudah proses pengamatan dan analisa tujuan tertentu. Secara otomatis, data logger mampu mengumpulkan data selama 24 jam sehingga akan sangat membantu pekerjaan Anda. Selama masa merekam Anda bisa meninggalkannya, secara otomatis alat ini akan merekam data pada periode waktu yang ditetapkan. Hal ini tentunya akan dapat memberikan gambaran yang lebih lengkap mengenai sebuah kondisi yang sedang menjadi pengamatan Anda, misalnya saja kondisi suhu atau kelembaban udara yang akan tercatat secara lengkap dari waktu ke waktu dengan menggunakan peralatan data logger tersebut. Para peneliti wajib memiliki perangkat data logger tersebut untuk dapat meringankan pekerjaan dengan hasil yang lebih akurat.

2.3 Kerangka Pemikiran

Berikut ini adalah *flowchart* yang penulis buat sebagai tahap dalam penelitian :



Gambar 2.9 Flowchart Kerangka Pemikiran

2.3.1 Identifikasi Masalah

Pada Pembangkit *Hybrid* tenaga angin dan surya memiliki dua sistem kerja yang berbeda sehingga baik tenaga angin dan surya mempunyai output daya yang berbeda pula. Bagaimana petugas dapat mengetahui berapa daya yang dihasilkan dari masing-masing pembangkit baik angin maupun surya sehingga pembangkit hybird dapat menghasilkan daya listrik menggunakan alat dengan *data logger*. Kemudian cara alat mengirimkan hasil pembacaan daya pada pembangkit ke *smartphone android* petugas sehingga alat terintegrasi ke sistem android.

2.3.2 Tinjauan Pustaka

a. Metode observasi

Metode yang dilakukan penulis dengan cara mengadakan pengamatan langsung atau observasi terhadap komponen alat yang telah disediakan penulis untuk diteliti dan mencatat hasil penelitian guna mengumpulkan data-data yang dibutuhkan untuk membuat sistem Integrasi Pembangkit Listrik *Hybrid* Tenaga Angin dan Surya

b. Studi Pustaka

Studi literatur pustaka yang berkaitan dengan masalah pembuatan model *data logger*, Pemerograman Android, Mikrokontroller. Penulis juga melakukan pembedahan buku dan mempelajari unsur dari objek yang diteliti serta mempelajari buku petunjuk dari masing-masing komponen dan mencari berita dari internet mengenai seputar topik permasalahan yang diambil.

2.3.3 Analisa Kebutuhan Sistem

Setelah melakukan pengumpulan data, tahap selanjutnya adalah tahap analisa. Analisa yang dilakukan adalah analisa terhadap sistem dan analisa terhadap kebutuhan sistem yang akan dirancang berdasarkan hasil observasi yang didapatkan dari penelitian yang dilakukan

2.3.4 Perancangan Aplikasi

a. Perancangan DFD

Perancangan DFD (*Data Flow Diagram*) dibuat untuk menggambarkan atau menjelaskan proses kerja suatu sistem.

b. Perancangan STD

Perancangan STD (*State Transition Diagram*) dibuat untuk menggambarkan bagaimana state yang satu dihubungkan dengan

state yang lainnya pada satu waktu. Spesifikasi proses dibuat untuk mempermudah pembuatan sistem ini.

c. Perancangan Tampilan

Perancangan tampilan dari aplikasi dibuat untuk mempermudah user menggunakan aplikasi.

2.3.5 Pembuatan Sistem *Hardware*

a. Perancangan alat

Merancang alat yang digunakan untuk mencatat data daya agar dapat terintegrasi kedalam sistem android.

b. Pembuatan alat

Dari hasil rancangan alat kemudian dibuatlah alat yang dapat mencatat data energi listrik yang dihasilkan oleh Pembangkit Listrik *Hybrid* Tenaga Angin dan Surya.

2.3.5 Pembuatan sistem *software*

a. Perancangan *software*

Software yang akan dibuat harus dirancang terlebih dahulu sehingga dalam penerapan akan lebih terstruktur.

b. Pengkodean

Dalam pengkodean, hal yang dilakukan adalah membuat coding program pada Arduino, coding program Android untuk komunikasi dengan Serial arduino yang digunakan dalam sistem aplikasi ini.

2.3.7 Pengujian Sistem

Setelah aplikasi android pembacaan daya pembangkit listrik *hybrid* selesai di rancang kemudian dilakukan uji coba sistem yang telah dibuat, apakah telah sesuai dengan yang dibutuhkan atau tidak. Apabila belum sesuai, dilakukan kembali perancangan sistem

untuk memperbaiki kesalahan sehingga sistem yang dibuat sesuai dengan yang diinginkan.

2.3.8 Pembahasan dan Penulisan Laporan

Pada tahap ini penulis menjabarkan hasil dari implementasi yang telah kerjakan dan uji coba dari implementasi tersebut. Dari pembahasan dan penulisan dapat di tarik kesimpulan tentang rumusan masalah yang penulis ambil.