BAB 2
TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Pulsa Niaga adalah toko dan juga Server pulsa isi ulang kartu *GSM* dan *CDMA*, pulsa niaga juga melayani jasa pembayaran rekening listrik dan telkom, berlokasi di Jl. Karang tineung dalam No. 67c yang telah berdiri sejak 15 mei 2010 didirikan oleh bapak Dede Memed Maulana

2.1.1 Sejarah Perusahaan

Sebelum berdirinya perusahaan pulsa niaga, bapak Dede hanya melakukan penjualan isi ulang pulsa prabayar, dimana kios pulsa ini didirikan pada tahun 2008, dengan berkembangnya usaha yang dikelola bapak Dede, beliau melihat peluang bisnis yang cukup besar, sehingga pada tahun 2009 beliau mengembangkan usaha keagenan pulsanya dengan menambahkan sebuah komputer yang bisa menangani transaksi yang dilakukan kios pulsa beliau, dan seiring berjalannya waktu, kios pulsa bapak Dede mengalami kemajuan dan mempekerjakan 1 orang karyawan sebagai operator di kios beliau, dan mulai merekrut orang-orang yang ingin membangun kios pulsa sebagai agen pulsa di Servernya, pada tahun 2010, bapak Dede mendaftarkan usahanya dan membentuk perusahaan pulsa niaga, hingga saat ini pulsa niaga telah berkembang tidak hanya melayani penjualan pulsa, melainkan melayani pembayaran PPOB (*Payment Point Online Bank*) dan *Voucher Game*

2.1.2 Visi dan Misi Perusahaan

Setiap perusahaan haruslah memiliki Visi dan Misi sebagai simbol suatu perusahaan tersebut memiliki ambisi dan keinginan tertentu dalam membangun perusahaannya, hal tersebut juga dapat meyakinkan pelanggan perusahaan untuk mau bergabung dengan perusahaan yang dikelolanya, sehingga perlunya penerapan Visi dan Misi pada suatu perusahaan agar suatu perusahaan tersebut tidak diremehkan oleh pihak luar, demikian juga halnya dengan CV. Pulsa niaga yang memiliki Visi dan Misi yang jelas, adapun Visi dan Misinya adalah sebagai beriku

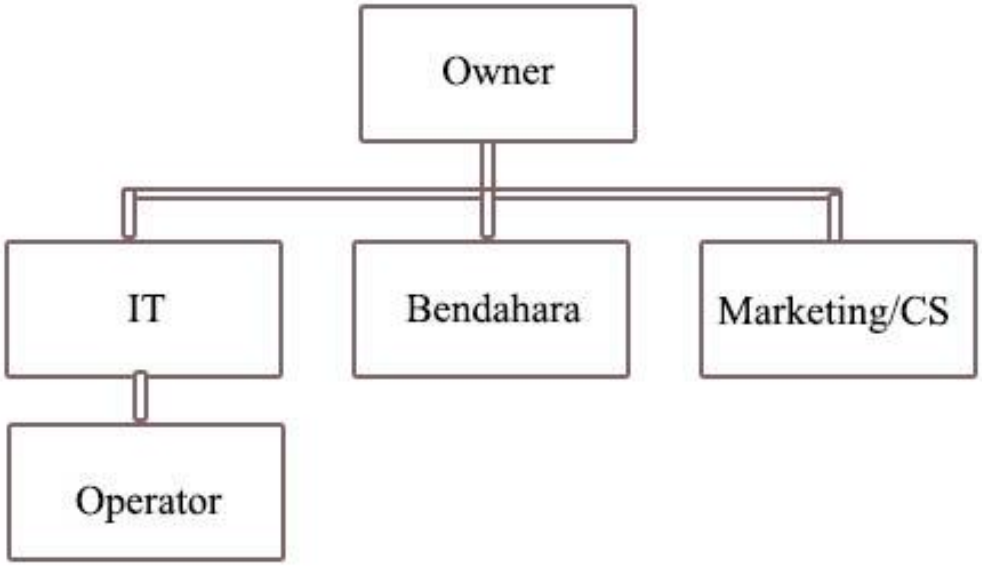
2.1.2.1 *Visi*

Menjadi mitra kerja/penyedia *1 Chip All Operator* dan *e-Payment Point Online Bank* terdepan, tangguh dan terpercaya serta selalu mengedepankan kenyamanan yang dapat menciptakan kepuasan pelanggan

2.1.2.2 *Misi*

1. Bekerja sama dengan prinsip saling menguntungkan dalam komunitas
2. berusaha dengan giat demi hasil yang terbaik
3. Kesejahteraan bersama dan kemajuan Bangsa,
4. Pelayanan 24 jam per hari tanpa hari libur

2.1.3 Struktur Organisasi dan Deskripsi Kerja



Gambar 2.1.1 Struktur Organisasi

2.1.3.1 Tugas dan Wewenang

Dalam menjalankan perusahaan, dibutuhkan yang namanya tugas dan wewenang dari setiap anggota diperusahaan tersebut, begitu juga dengan perusahaan CV. Pulsa niaga yang memiliki tugas dan wewenangnya masing-masing

a. Owner

Owner adalah seseorang yang memimpin jalannya perusahaan serta melakukan *management* terhadap perusahaan dimana tugas dan wewenangnya adalah :

1. Memimpin perusahaan
2. Menyusun strategi perusahaan
3. Pemberi keputusan perusahaan

b. IT

IT adalah divisi yang bertugas dibidang infrastruktur teknologi yang ada pada perusahaan CV. Pulsa niaga, adapun tugas dan wewenangnya adalah :

1. Melakukan perawatan Server
2. Menyusun pembelian alat operasional
3. Penanggungjawab alat inventaris

c. Bendahara

Bendahara adalah seseorang yang memegang tanggungjawab keuangan di perusahaan CV. Pulsa niaga dengan tugas dan wewenangnya adalah :

1. Membuat laporan keuangan
2. Membuat laporan inventaris

d. Marketing/CS

Marketing/CS adalah suatu divisi yang menghubungkan perusahaan dengan pelanggan, adapun tugas dan wewenangnya adalah :

1. Melakukan promosi
2. Melayani agen dan pelanggan
3. Menjaga kios pulsa niaga

e. Operator

Operator adalah jabatan yang dinaungi divisi IT, dimana tugas dan wewenangnya adalah :

1. Mengawasi Server
2. Menjaga kios pulsa niaga

2.2 Landasan Teori

Landasan teori merupakan definisi, konsep yang telah disusun secara sistematis dan dasar yang kuat dalam sebuah penelitian. Landasan teori yang digunakan dalam penyusunan sistem *monitoring* dan *auto switch* sumber daya pada komputer Server meliputi Aplikasi *Android*, Pengertian Tenaga Matahari, arus listrik, *Arduino*, sensor-sensor yang digunakan untuk mendukung kerja *Arduino*, Pemrograman Berorientasi Objek, Konsep Dasar Berorientasi Objek, Kompoonen Sistem Tenaga Surya, *Unified Modeling Language* (UML).

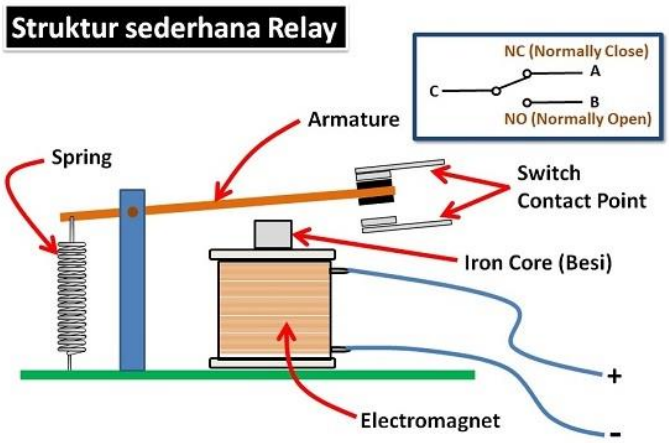
2.2.1 *Automatic Transfer Switch*

ATS merupakan singkatan dari kata *Automatic Transfer Switch*, jika dipahami berdasarkan arti kata tersebut maka ATS adalah sakelar yang bekerja secara otomatis, dengan memperhatikan sumber listrik dari PLN atau sumber utama apakah terputus, jika faktor tersebut terjadi maka sakelar akan berpindah kesumber listrik yang lainnya misalnya adalah Genset atau PLTS ataupun sumber listrik lainnya, dan ATS juga akan melakukan perpindahan sakelar ke sumber utama ketika sumber utama sudah dapat digunakan dengan baik. [2]

2.2.1.1 Relay

Relay adalah Saklar yang dioperasikan memanfaatkan listrik untuk melakukan posisi (*switch*) jalur antara sumber ke tujuan. *Relay* merupakan komponen *Electromechanical* yang terdiri dari 2 bagian utama yaitu *Coil* dan juga seperangkat kontak saklar, dimana *Coil* adalah berupa kumparan tembaga yang mampu menghasilkan medan magnet jika dialiri listrik sementara seperangkat Kontak Saklar/*Switch*, yang terdiri dari : [3]

1. *Armature* dimana *armature* berfungsi sebagai tuas penentu tujuan dari sumber apakah Kearah hidup ataupun mati
2. *Switch Contact Point* (Saklar) adalah titik tujuan yang akan dihubungkan ke sumber arus yang ditentukan oleh *armature* apakah terhubung ke dirinya atau tidak
3. *Spring* adalah sebuah per/pegas yang digunakan bertujuan untuk menentukan kondisi awal armature berada pada *switch contact point* sisi *Normaly close* atau saat *coil* tidak dialiri listrik apapun.



Gambar 2.2.1 Skema Relay

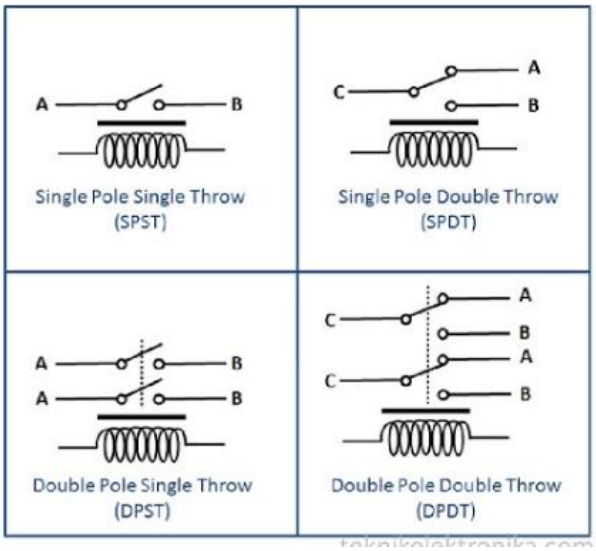
Berdasarkan gambar diatas, sebuah Besi (*Iron Core*) yang dililit oleh sebuah kumparan tembaga yang berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan tembaga diberikan arus listrik, kumparan tembaga atau *coil* akan menghasilkan gaya Elektromagnet yang dapat menarik *armature* dan menyatuh ke *iron core* dan *armature* pindah dari Posisi sebelumnya NC (*Normally close*) ke posisi baru NO (*Normally open*) sehingga dapat menjadi kondisi lain untuk melakukan pengkondisian menggunakan relay dimana arus dari sumber dapat dialirkan ke posisi barunya yaitu *normaly open*. Posisi dimana Armature tersebut berada sebelumnya (NC) akan menjadi *open* atau tidak terhubung jika *coil* tidak dialiri arus listrik, dan *armature* akan kembali lagi ke posisi awal yaitu pada *normaly close*. *Coil* yang digunakan oleh Relay untuk menarik *Contact Poin* ke Posisi *Close* pada umumnya hanya membutuhkan arus listrik yang relatif kecil, sehingga untuk melakukan *switching* tidak akan memboroskan penggunaan arus.

Terdapat banyak fungsi relay, namun ada beberapa yang umum diaplikasikan kedalam peralatan elektronika diantaranya adalah :

1. Relay digunakan untuk menjalankan Fungsi Logika (*Logic Function*)
2. Relay digunakan untuk memberikan Fungsi Penundaan Waktu (*Time Delay Function*)

3. Relay digunakan untuk mengendalikan Sirkuit Tegangan Tinggi dengan Bantuan dari signal tegangan rendah.
4. Relay berfungsi untuk melindungi Motor ataupun komponen lainnya dari kelebihan tegangan ataupun koslet (*short*).

Didalam *Contact switch* relay terdapat 2 istilah umum yang dapat membedakan jenis-jenis relay yaitu *pole* dan *throw* banyaknya *pole* dan *throw* mempengaruhi banyaknya kondisi yang terdapat dalam relay, adapun jenis jenis relay berdasarkan banyaknya *pole* dan *throw* dapat dikelompokkan menjadi 4 jenis yang dapat dilihat pada gambar dibawah ini



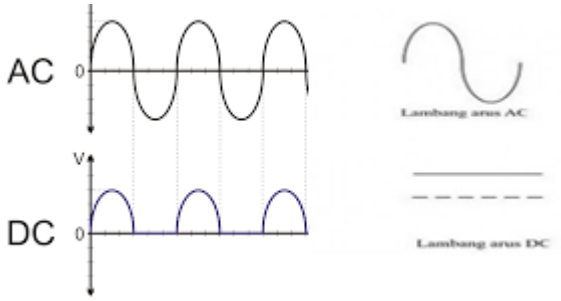
Gambar 2.2.2 Jenis Relay

2.2.1.2 Listrik

Listrik adalah salah satu jenis energi, dimana energi listrik memegang peranan penting bagi kehidupan kita. Listrik adalah suatu muatan yang terdiri dari muatan positif dan muatan negatif. Pada umumnya energy listrik dapat berjalan jika mengandung 2 unsur yaitu tegangan dan arus, dimana tegangan adalah perbedaan potensial listrik atar dua muatan listrik (negative dan positif) perbedaan tersebut yang membuat ada pergerakan muatan listrik atau arus dan Arus listrik merupakan muatan listrik yang bergerak dari tempat yang berpotensial tinggi ke tempat

berpotensial rendah, melewati suatu penghatar listrik, arus tidak dapat dihantar jika suatu penghantar tidak memiliki tegangan.

Arus listrik terdiri dari dua jenis yaitu arus listrik searah (*direct current*) dan arus listrik bolak-balik (*alternative current*). Arus listrik DC merupakan arus listrik yang mengalir secara terus menerus kesatu arah. Contoh penggunaan arus DC terdapat pada alat elektronik rumah tangga seperti komponen komputer, laptop, telepon genggam, dan perangkat elektronik lainnya yang menggunakan *adapter charger*, Sementara Arus listrik AC merupakan arus listrik yang mengalir bolak-balik. Arus AC sering digunakan untuk memasok kelistrikan dirumah atau pun kantor-kantor yang biasa dialirkan oleh PLN, penggunaan arus AC biasanya menggunakan tegangan tinggi berbeda dengan tegangan DC yang biasanya relatif rendah, untuk setiap Negara atau wilayah, penentuan standart voltase listrik AC berbeda beda, ada yang menggunakan 220V seperti di Indonesia, dan ada yang menggunakan 110V. [4]



Gambar 2.2.3 Bentuk dan Lambang dari Arus Listrik

2.2.2 Sistem Pembangkit Listrik Tenaga Surya (PLTS)

Pembangkit Listrik Tenaga Surya adalah sebuah daya alternatif, PLTS hanya akan bisa menghasilkan sumber listrik ketika sensor cahaya / solar panel mendapatkan energi matahari, semakin banyak energi matahari yang mengenai solar panel, maka akan semakin banyak arus yang dapat dihasilkan solar panel. PLTS sangat efektif jika dikembangkan dinegara atau daerah yang memiliki iklim tropis, salah satu daerah yang memiliki iklim tropis adalah Indonesia,sehingga pengembangan sistem PLTS di indonesia cukup efektif dikarenakan hampir satu tahun penuh matahari akan terus menyinari daerah indonesia dimana diketahui rata-

rata matahari bersinar di Indonesia adalah 2000jam per tahunnya sehingga diperoleh 5.48 jam/hari atau 5 jam 28.8 menit dalam sehari dengan menggunakan perhitungan 2000/365 = 5.48 jam/hari dimana 365 adalah jumlah hari dalam 1 tahun. [4]

Pengaplikasian PLTS pada tempat penelitian ini bertujuan untuk menyediakan sebuah daya alternatif pada saat terjadi pemadaman Listrik, agar monitoring tetap berjalan pada saat listrik konvensional tidak ada. Adapun komponen-komponen pendukung pembuatan PLTS adalah :

2.2.2.1 Energi Surya



Gambar 2.2.4 Matahari

Energi surya adalah energi cahaya yang dihasilkan oleh matahari yang merujuk pada radiasi energi dalam bentuk panas dan cahaya yang dipancarkan oleh matahari. Tanpa energi cahaya yang bersumber dari matahari, bumi tidak akan mampu mendukung kehidupan dan energi surya adalah bentuk energi paling berlimpah yang tersedia di bumi. Energi surya memiliki potensi besar dan banyak teknologi surya yang berkembang dengan sangat cepat.

Energi surya adalah sumber energi terbarukan yang paling penting, dikarenakan sumbernya yang tak terbatas dan hanya akan habis dikarenakan proses alamiah seperti perubahan siang ke malam dikarenakan proses alamiah rotasi bumi. Energi matahari dapat di konversi menjadi energi listrik dengan bantuan alat tenaga surya salah satu contoh tenaga surya yang dapat mengkonversi dari energi surya menjadi energi listrik adalah dengan menggunakan *photovoltaic* atau yang sering dikenal dengan nama panel surya. [5]

2.2.2.2 Solar Cell



Gambar 2.2.5 Panel surya

Sel Surya atau fotovoltaik adalah alat yang mengubah energi matahari menjadi energi listrik, dimana energi matahari yang mengenai fotovoltaik akan menghasilkan tegangan, panel surya atau sel surya terdiri dari 3 lapisan, lapisan semikonduktor p di bagian atas, lapisan pembatas di tengah, dan lapisan semikonduktor n di bagian bawah. Semikonduktor jenis n merupakan semikonduktor yang memiliki kelebihan elektron, sehingga kelebihan muatan negatif, (n = negatif). Sedangkan semikonduktor jenis p memiliki kelebihan hole, sehingga disebut dengan p (p = positif) karena kelebihan muatan positif. Caranya, dengan menambahkan unsur lain ke dalam semkonduktor, maka kita dapat mengontrol jenis semikonduktor tersebut

.

2.2.2.3 Battery



Gambar 2.2.6 *Battery* (Accu)

Dalam pembangunan sistem PLTS, baterai adalah salah satu komponen terpenting yang harus diperhatikan, dikarenakan solar panel tidak dapat menyimpan arus yang telah dihasilkannya, sehingga dibutuhkan sebuah penyimpanan. Baterai

atau *Accu* ini berfungsi sebagai sebuah wadah penyimpanan unuk energi listrik dalam bentuk kimia yang dapat menghasilkan elektron-elektron tertentu yang dapat saling berkaitan antara yang satu dengan yang lainnya dan nantinya elektron tersebutlah yang menimbulkan suatu peristiwa reaksi kimia serta mampu menciptakan arus listrik searah (arus DC) yang akan menyuplai tegangan pada beban yang terhubung pada baterai tersebut. [4]

Reaksi kimia yang membentuk elektron disebut reaksi elektrokimia. Atau dengan kata lain baterai ialah kumpulan sel-sel listrik yang dihubungkan baik secara seri maupun paralel sesuai dengan karakteristik (arus dan tegangan) beban yang terpasang.

Ditinjau dari pemakaiannya baterai dibagi menjadi dua macam, yaitu :

1. Sel primer *(primary cell)*

   Sel listrik primer merupakan sel listrik yang kapasitasnya sudah berkurang atau sudah habis masa/umurnya pakainya, maka material aktifnya tidak bisa dinormalkan kembali seperti semula dan biasanya sel seperti ini disebut juga dengan sel listrik habis/sekali pakai.

2. Sel sekunder *(secondary cell)*

   Sel listrik sekunder merupakan sel listrik yang apabila kapasitas pemakaiannya sudah berkurang, maka material aktifnya dapat dinormalkan kembali (diisi ulang) dan biasanya sel listrik seperti itu disebut juga sebagai sistem sel listrik discharge

   Kapasitas sebuah baterai atau sel adalah banyak muatan yang tersedia dan diungkapkan dalam *Ampere-hour* (Ah). *Ampere* adalah unit satuan untuk arus listrik yang didefinisikan sebagai banyak muatan yang melalui sebuah konduktor dalam satu detik. Besarnya kapasitas baterai ini dipengaruhi oleh banyaknya material aktif, elektrolit, dan luas plat. Kapasitas baterai diukur dengan cara melepaskan muatan dengan arus konstan hingga mencapai tegangan terminalnya. Besarnya kapasitas baterai dihitung dengan perkalian arus pelepasan muatan dengan waktu yang dibutuhkan untuk mencapai tegangan terminalnya. Contoh kasus : sebuah aki memiliki kapasitas 24AH, dimana dari kapasitas tersebut dapat

diketahui maximal arus yang dapat keluar dari aki tersebut adalah 24A hanya dalam waktu maximal 1 jam, untuk mengetahui lama penggunaan aki tergantung dari beban yang dialiri oleh sumber aki, semakin sedikit arus yang dibutuhkan maka semakin lama aki dapat bertahan, sehingga dari contoh tersebut kita dapat menuliskan rumus menjadi : [4]

$$\mathbf{C = I.t}$$

Rumus 2.2.1 Mencari Kapasitas Baterai

Dimana :

C = Kapasitas Baterai

I = Besarnya arus yang mengalir

t = Waktu (dalam satuan jam)

Istilah umum lain yang sering digunakan untuk menggambarkan kemampuan baterai untuk mengirimkan arus adalah tingkat kapasitas (*rated capacity*). Namun pada umumnya, aki tidak diperbolehkan sampai pada tegangan tertentu, ada batas tegangan yang tidak boleh dilewati oleh setiap aki, kondisi melewati batas tegangan seharusnya aki disebut dengan *Over Voltage* dimana aki dalam keadaan melebihi tengangan maximal dari aki, dan *Low Voltage* dimana aki dalam keadaan aki harus di isi ulang (*recharge*) karena sudah mengalami penurunan arus lebih dari setengah kapasitas baterai.

2.2.2.4 Pengatur Pengisi Baterai (*Solar Charge Controller)*



Gambar 2.2.7 *Charge Controller*

Pengatur Pengisi Baterai (PPB) adalah peralatan elektronik yang digunakan untuk mengatur arus DC yang masuk ke baterai dan yang keluar dari baterai ke beban,*controller* juga dapat berfungsi sebagai proteksi kebocoran arus DC, dan ketika gelap dimana panel surya tidak dapat menghasilkan arus listrik, *controller* berfungsi untuk mencegah arus yang ada pada aki bergerak ke solar panel sehingga membuat aki menjadi kekurangan arus [4]

Solar charge controller mengatur *overcharging* (kelebihan pengisian - karena baterai sudah penuh) dari panel surya. Kelebihan voltase dan pengisian akan mengurangi umur baterai. Secara umum PPB memiliki 2 tipe yaitu PWM dan MPPT untuk mengatur fungsi pengisian baterai dan pembebasan arus dari baterai ke beban, dari 2 tipe PPB diatas memiliki peruntukannya masing masing sesuai skala dari PLTS yang akan dibangun, untuk sistem PLTS dengan daya rendah sampai menengah, disarankan untuk menggunakan tipe PWM, dan untuk PLTS dengan daya menengah sampai tinggi disarankan untuk menggunakan tipe MPPT

2.2.2.5 Inverter

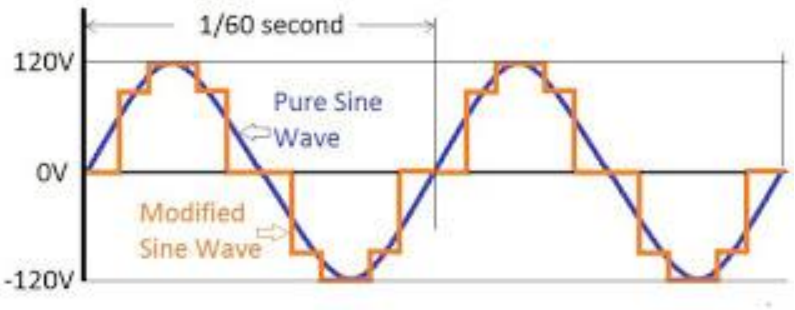


Gambar 2.2.8 *Inverter DC to AC*

Inverter adalah perangkat elektrik yang digunakan untuk mengubah arus listrik searah (DC) menjadi arus listrik bolak balik (AC). Inverter mengkonversi DC dari perangkat seperti batere, panel surya / solar cell menjadi AC. Penggunaan inverter dari dalam Pembangkit Listrik Tenaga Surya (PLTS) adalah untuk perangkat yang menggunakan AC (Alternating Current).

Inverter memiliki tipe gelombang AC yang dihasilkan, yaitu : tipe sinus dan juga tipe *modified*. Dimana perbedaan kedua tipe ini terdapat pada bentuk gelombang yang dihasilkan, jika berbentuk sinus, maka hasilnya sama seperti arus

AC dari PLN, sementara yang modified, memiliki bentuk kotak (*squere*) yang dimodifikasi menyerupai gelombang sinus, sehingga hasilnya tidak maksimal atau halus seperti gelombang sinus atau AC PLN, untuk lebih mengetahui perbedaannya, perhatikan gambar dibawah



Gambar 2.2.9 Bentuk Gelombang *Modified dan Pure Sine*

2.2.3 Mikrokontroler

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya, mikrokontroler umumnya terdiri dari CPU (*Central Processing Unit*), memori, *I/O*, dan unit pendukung seperti ADC (*Analog to Digital Converter*) yang sudah terintegrasi didalamnya. Mikrokontroler memiliki beberapa jenis yang sering digunakan di dunia industri ataupun perorangan, adapun jenis-jenisnya ialah : [6]

2.2.3.1 Mikrokontroler AVR

Mikrokonktroler *Alv and Vegard's Risc processor* atau sering disingkat AVR merupakan mikrokonktroler RISC 8 bit. Karena RISC inilah sebagian besar kode instruksinya dikemas dalam satu siklus clock.

Mikrokontroler AVR merupakan salah satu jenis arsitektur mikrokontroler yang menjadi andalan Atmel. Arsitektur ini dirancang memiliki berbagai kelebihan dan merupakan penyempurnaan dari arsitektur mikrokontroler-mikrokontroler yang sudah ada.

Berbagai seri mikrokontroler AVR telah diproduksi oleh Atmel dan digunakan di dunia sebagai mikrokontroler yang bersifat low cost dan high performance. Di Indonesia, mikrokontroler AVR banyak dipakai karena fiturnya yang cukup lengkap, mudah untuk didapatkan, dan harganya yang relatif terjangkau.

Mikrokontroler AVR ini adalah jenis mikrokontroler yang akan digunakan dalam penelitian ini, dimana yang digunakan adalah mikrokontroler AVR dengan tipe *Atmega* yang digunakan oleh *Arduino mega*



Gambar 2.2.10 Logo *Arduino*

Arduino Mega adalah papan pengembangan (development board) mikrokontroler yang berbasis chip Atmega2650. Disebut sebagai papan pengembangan karena board ini memang berfungsi sebagai arena prototyping sirkuit mikrokontroller. Dengan menggunakan papan pengembangan, anda akan lebih mudah merangkai rangkaian elektronika mikrokontroller dibanding jika anda memulai merakit *micro controller* dari awal di projectboard.

Arduino dalam penelitian ini digunakan sebagai sebuah kontrol untuk monitoring dan control relay untuk daya listrik.

2.2.3.2 Mikrokontroler MSC-51

Mikrokonktroler ini termasuk dalam keluarga mikrokonktroler CISC (*Complex Instruction Set Computer*). Sebagian besar instruksinya dieksekusi dalam 12 siklus clock.

Mikrokontroler MCS51 buatan Atmel terdiri dari dua versi, yaitu versi 20 kaki dan versi 40 kaki. Semua mikrokontroler ini dilengkapi dengan Flash PEROM (*Programmable Eraseable Read Only Memory*) sebagai media memori-program, dan susunan kaki IC-IC tersebut sama pada tiap versinya.

2.2.4 Sensor Pendukung

Untuk mempermudah pekerjaan dalam membangun sistem menggunakan mikrokontroler, kita dapat menggunakan beberapa modul sensor pendukung yang nantinya dapat mempermudah kita dalam membangun sistem, adapun

sensor sensor yang dapat digunakan dalam pembangunan sistem monitoring adalah :

2.2.4.1 ESP8266 *Module*



Gambar 2.2.11 *ESP8266 NodeMCU Mikrokontroler*

ESP8266 *Module* adalah salah satu module yang dapat menghubungkan *micro controller* kita dengan koneksi jaringan lokal (*LAN*) ataupun jaringan internet menggunakan media *wifi* untuk melakukan transper data via jaringan lokal atau pun jaringan internet.

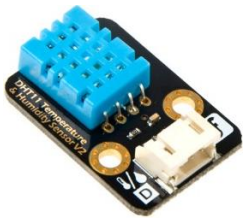
Dalam penelitian ini ESP8266 berfungsi sebagai sebuah alat untuk mengirim data dari mikrokontrol kedalam *firebase database*.

Firmware default yang digunakan oleh perangkat ini menggunakan *AT Command*, selain itu ada beberapa *Firmware SDK* yang digunakan oleh perangkat ini berbasis opensource yang diantaranya adalah sebagai berikut :

- **NodeMCU** dengan menggunakan basic programming *lua*, dan bisa juga menggunaakn bahasa C
- **MicroPython** dengan menggunakan basic programming *python*
- **AT Command** dengan menggunakan perintah perintah *AT command*

Untuk pemrogramannya sendiri kita bisa menggunakan *ESPlorer* untuk *Firmware* berbasis *NodeMCU* dan menggunakan *putty* sebagai terminal control untuk *AT Command*.

2.2.4.2 Sensor DHT 11

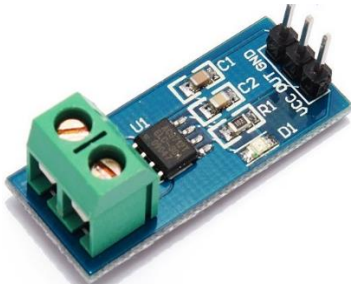


Gambar 2.2.12 *DHT11*

Sensor DHT 11 adalah sensor digital yang dapat mengukur suhu dan kelembaban udara di sekitarnya. Sensor ini sangat mudah digunakan bersama dengan *Arduino*. Memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat. Koefisien kalibrasi disimpan dalam *OTP program memory*, sehingga ketika internal sensor mendeteksi sesuatu, maka *module* ini menyertakan koefisien tersebut dalam kalkulasinya. [7]

Sensor DHT 11 digunakan untuk mengukur suhu dan kelembaban pada ruangan Server yang terdapat pada CV. Pulsa niaga, dimana suhu ruangan haruslah berada pada suhu operasional baik dari hardisk yaitu diantara 18 ℃ − 27 ℃ dengan kelembaban $40RH - 60Rh$ [8]

2.2.4.3 Sensor Arus ACS 712



Gambar 2.2.13 Sensor Arus ACS 712

Sensor Arus ACS 712 adalah sensor pengukur arus AC yang mampu mengukur besaran arus yang melewati sensor tersebut, karena didalamnya terdapat rangkaian low-offset linear Hall dengan satu lintasan yang terbuat dari tembaga. cara kerja sensor ini adalah arus yang dibaca mengalir melalui kabel tembaga yang terdapat didalamnya yang menghasilkan medan magnet yang di tangkap oleh integrated Hall IC dan diubah menjadi tegangan proporsional. Ketelitian dalam pembacaan sensor dioptimalkan dengan cara pemasangan komponen yang ada didalamnya antara penghantar yang menghasilkan medan magnet dengan hall transducer secara berdekatan. [9]

Dalam penelitian ini sensor arus ACS 712 digunakan untuk mengukur arus pada sumber daya utama yaitu PLN, sehingga dapat dilakukan keputusan otomatis untuk memindahkan sakelar ke sumber cadangan.

2.2.4.4 Sensor Tegangan AC



Gambar 2.2.14 Sensor Tegangan AC

Sensor tegangan AC berfungsi sebagai pengukur tegangan arus AC, dimana sensor Zmpt101b ini mampu mengukur tegangan AC 110-250 VAC

2.2.4.5 Sensor Tegangan DC



Gambar 2.2.15 Sensor Tegangan DC

Sensor tersebut berfungsi untuk menghitung tegangan DC, untuk sensor ini bisa menghitung 0-25V DC. Output dari sensor ini adalah voltase 0-5V. Sehingga 5x5=25V, jadi apabila tegangan yang dihitung adalah 5V maka keluarannya adalah 1V. Jika yang dihitung 12V, maka keluarannya adalah 2.4V.

2.2.5 Metode Pendekatan Sistem

Metode pendekatan sistem digunakan untuk mendeskripsikan sistem yang telah maupun akan dirancang dengan metode pengembangan tertentu. Dalam analisis dan perancangan ini digunakan metode pendekatan sistem yang berorientasi objek (*Object-Oriented Analysis and Design*) dan metode pembangunan sistem menggunakan metode *prototype*.

2.2.5.1 Prototype

Prototype adalah sebuah model yang mula mula (model asli) yang menjadi sebuah contoh baku, untuk digunakan sebagai contoh untuk dipelajari ataupun dikembangkan. *Prototype* adalah sebuah sistem atau alat yang akan dibuat dalam penelitian ini.

2.2.5.2 Prototyping

Prototyping adalah sebuah metode pembangunan sebuah sistem maupun perangkat atau sebuah siklus hidup menggunakan *protoyping (life cycle using prototyping)* adalah salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (working model). Tujuannya adalah mengembangkan model menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat daripada metode tradisional dan biayanya menjadi lebih rendah. Ada banyak cara untuk

memprotoyping, begitu pula dengan penggunaannya. Ciri khas dari metodologi adalah pengembang sistem *(system developer)*, klien, dan pengguna dapat melihat dan melakukan eksperimen dengan bagian dari sistem komputer dari sejak awal proses pengembangan. [1]

Definisi *Prototyping* Model

Prototyping merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan pelanggan dapatsaling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelangganhanya 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 efesiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer. Untuk mengatasi ketidakserasian antara pelanggan dan pengembang maka harus dibutuhakan 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 menyelasaikan 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.

Prototyping Model

kadang-kadang klien hanya memberikan beberapa kebutuhan umum *software* tanpa detail *input*, proses atau detail *output*. Dilain waktu mungkin dimana tim pembangun (*developer*) tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan *form user interface*. Ketika situasi seperti ini terjadi model *prototyping* sangat membantu

proses pembangunan *software*. Proses pada model *prototyping* bisa dijelaskan sebagai berikut:

Pengumpulan kebutuhan *developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan perancangan, perancangan dilakukan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *Prototype*.

Evaluasi *Prototype*: klien mengevaluasi *Prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-Prototype* dibuat untuk memuaskan kebutuhan klien dan untuk memahami kebutuhan klien lebih baik. *Prototype* yang dibuat dapat dimanfaatkan kembali untuk membangun *software* lebih cepat, namun tidak semua *Prototype* bisa dimanfaatkan.Sekalipun *Prototype* memudahkan komunikasi antar *developer* dan klien, membuat klien mendapat gambaran awal dari *Prototype*. Penjelaskan pengertian *Prototype* berdasarkan Ilmu Interaksi Manusia dan Komputer. *Prototype* merupakan alat yang digunakan untuk mensimulasikan beberapa atau tidak semua fitur dari sistem yang akan dibuat. Terdapat 3 pendekatan utama *prototyping*, yaitu:

1. *Throw-Away*

   Prototype dibuat dan dites. Pengalaman yang diperoleh dari pembuatan *prototype* digunakan untuk membuat produk akhir (final), kemudian *prototype* tersebut dibuang (tak dipakai).

2. *Incremental*

   Produk finalnya dibuat sebagai komponen-komponen yang terpisah. Desain produk finalnya secara keseluruhan haya ada satu tetapi dibagi dalam komonen-komponen lebih kecil yang terpisah (*independent*).

3. *Evolutionary*

   Pada metode ini, *Prototype*nya tidak dibuang tetapi digunakan untuk iterasi desain berikutnya. Dalam hal ini, sistem atau produk yang sebenarnya dipandang sebagai evolusi dari versi awal yang sangat terbatas menuju

produk final atau produk akhir. Di sisi manajemen, terdapat beberapa masalah potensial yang terkait dengan *prototyping*, seperti:

1. Waktu, membangun *prototype* membutuhkan waktu, sehingga seringkali *prototype* dipakai jika waktunya cepat. Hingga muncul istilah *rapid prototyping*.
2. Rencana, sebagian manajer proyek tidak memiliki pengalaman untuk menyatukan proses *prototyping* dengan keseluruhan rencana perancangan.
3. Fitur Non-Fungsional

   seringkali fitur sistem yang paling penting merupakan fitur non-fungsional seperti *safet*y dan *reliability*, tidak disertakan dalam *prototyping*.
4. Kontrak

   proses desain kadang dibatasi oleh kontrak antara desainer dengan customer yang mempengaruhi aspek teknik dan manajerial.

Teknik *Prototyping*

Terdapat beberapa terdapat beberapa teknik yang digunakan untuk membuat rapid *prototype*, seperti:

Storyboard, adalah bentuk *prototype* yang paling sederhana berupa gambaran secara grafis dari tampilan sistem yang akan dibangun tanpa fungsi dari sistem.

Simulasi fungsi terbatas, fungsi sistem disertakan pada *prototype* tidak sekadar gambar tampilannya saja.

High-level, programing support, HyperTalk adalah contoh dari *special-purpose high-level programming language* yang memudahkan desainer membuat fitur tertentu dari sebuah sistem interaktif.

Keunggulan dan Kelemahan *Prototyping*.

Keunggulan *prototyping* adalah:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.

2. Pelanggan berperan aktif dalam pengembangan sistem
3. Lebih menghemat waktu dalam pengembangan sistem.
4. 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.

Prototyping ini digunakan sebagai metode pembangunan sistem di penelitian ini.

2.2.5.3 Unified Modified Language (UML)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, Men-spesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berorientasi objek. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Berikut ini merupakan beberapa bagian dari UML: [10]

a. *Use Case Diagram*

Menggambarkan sejumlah eksternal aktor dan hubungannya ke *use case* yang diberikan oleh sistem. Komponen utama pembentuk *use case* antara lain:

1. Aktor (*actor*), menggambarkan suatu hal diluar sistem yang berinteraksi dengan sistem.
2. *Use case* , kegiatan/aktivitas yang disiapkan oleh sistem. Menekankan pada “apa” yang dikerjakan oleh sistem, bukan “bagaimana” sistem itu bekerja.
3. Hubungan (*link*), penjelasan tentang hubungan suatu komponen *use case* diagram dengan komponen lainnya.

Pada *use case* terdapat *use case relationship* yang menunjukkan sifat *reusable, optional*, and bahkan menspesialisasikan *use case behavior* antar tiap *use case* yang ada. Terdapat 4 jenis *Relationship* pada *use case* yaitu Asosiasi, Generalisasi, <<*include*>> dan <<*extend*>>.

1. Asosiasi

   Asosiasi digunakan untuk menggambarkan interaksi antara aktor dengan setiap *use case* tertentu. Relasi ini digambarkan sebagai garis penghubung aktor terhadap *use case* yang berelasi dengan aktor tersebut.

2. Generalisasi

   Suatu relasi antara *use case* umum (induk) dan *use case* yang lebih spesifik (anak). Relasi Generalisasi memungkinkan *use case* yang lebih spesifik memiliki perilaku (*behaviour*) yang sama dengan *use case* yang lebih umum atau bisa disebut *use case* induk. Relasi generalisasi digambarkan dengan anak panah segitiga. *Use case* yang terletak di sisi anak panah adalah *use case* induk dan yang terletak di sisi lainnya adalah *use case* yang mewarisi perilaku *use case* induk.

3. <<*include*>> *relationship*

   Relasi *include* memungkinkan terjadinya penambahan perilaku (*behaviour*) ke dalam *use case* awal yang pada dasarnya *use case* ini tidak dapat berdiri sendiri tanpa adanya penambahan *use case* , dan *use case* awal tidak akan lengkap tanpa adanya *use case* tambahan ini. *Use case* yang berada pada kepala anak panah adalah *use case* awal, dan pada sisi lain adalah *use case* penambah.

4. <<*extend*>> *relationship*

Relasi *extend* memungkinkan terjadinya penambahan beberapa *behaviour* (tingkah laku) ke dalam *use case* awal yang pada dasarnya *use case* tersebut sudah dapat berdiri sendiri tanpa adanya penambahan. *Extend* digambarkan dengan anak panah yang mempunyai garis putus-putus. *Use case* yang berada pada kepala anak panah adalah *use case* awal dan yang berada di lain sisi adalah *use case* tambahan.

b. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti *use case*.

c. *Class Diagram*

Menggambarkan struktur dan deskripsi *class*, *package* (paket) dan objek beserta hubungan satu sama lain seperti *containment* (penahanan), pewarisan, asosiasi dan lain-lain.

d. *Sequence Diagram*

Sequence diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

UML dalam penelitian ini digunakan sebagi perancangan skema relasi, *usecase* diagram, dan beberapa digram lainya di bab 3.

2.2.5.4 Object Oriented Programming

Pemrograman berorientasi objek atau *Object-Oriented Programming* disingkat OOP merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih

mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

OOP ini digunakan dalam penelitian ini sebagai metode pembanguna program *APK androidnya*.

2.2.6 Perangkat Lunak Pendukung

Penulis menggunakan beberapa perangkat lunak pendukung yang digunakan untuk pengembangan sistem, dibawah ini akan dijelaskan perangkatperangkat lunak yang penulis gunakan

2.2.6.1 Java

Java adalah salah satu bahasa pemrograman komputer yang memungkinkan *programmer* (orang yang membuat program) dapat memberi instruksi pada komputer. Java juga dapat didefinisikan sebagai suatu nama sekumpulan teknologi yang berfungsi untuk membuat dan menjalankan perangkat lunak pada komputer. Kebanyakan dari programmer lebih suka menyebut Java sebagai sebuah teknologi dibandingkan dengan hanya sebuah bahasa pemrograman, karena Java itu lebih lengkap dibandingkan dengan semua bahasa pemrograman konvensional

2.2.6.2 Javascript

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan disisi klien dan juga javascript dapat berperan sebagai Server. JavaScript klien sebagai bahasa pemrograman yang berjalan khusus untuk dibrowser atau halaman web agar halaman web menjadi lebih hidup, sementara Javascript Server berfungsi sebagai media layanan web Server, sama seperti bahasa Server lain seperti php.

2.2.6.3 Database Firebase

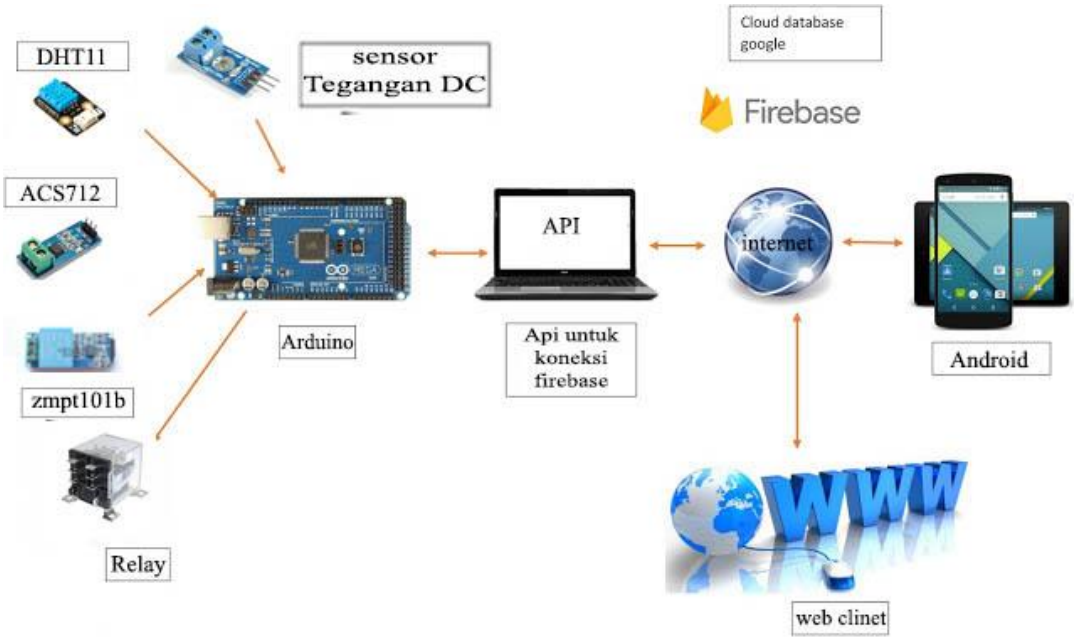
Firebase Realtime Database adalah basis data yang berbasis awan. Data disimpan bertipe JSON dan disinkronkan secara *real time* ke setiap klien yang terhubung. Ketika Anda membangun aplikasi lintas-platform dengan iOS, Android, dan JavaScript SDK, semua klien Anda berbagi satu *instance Realtime Database* dan secara otomatis menerima pembaruan dengan data terbaru

Keunggulan Database Firebase bisa dilihat pada tabe dibawah ini

Tabel 2.2.1 Keunggulan Database Firebase

Keunggulan	Keterangan
Realtime	Sebagai ganti permintaan HTTP biasa, Firebase Realtime Database menggunakan sinkronisasi data, setiap kali data berubah, setiap perangkat yang terhubung menerima pembaruan tersebut dalam milidetik. Memberikan pengalaman kolaboratif dan mendalam tanpa berpikir tentang kode jaringan.
Offline	Aplikasi Firebase tetap responsif bahkan ketika offline karena *SDK Firebase Realtime Database* mempertahankan data ke disk. Setelah konektivitas didapatkan kembali, perangkat klien akan menerima perubahan yang terlewat, menyinkronkannya dengan status Server saat ini.
Bisa Diakses dari Perangkat Klien	*Firebase Realtime Database* bisa diakses secara langsung dari perangkat seluler atau browser web; tidak diperlukan lagi Server aplikasi. Keamanan dan validasi data tersedia melalui *Firebase Realtime Database Security Rules*, aturan berbasis ekspresi yang dieksekusi ketika data dibaca atau ditulis.

Dalam menjalankan Firebase database, kita tidak perlu diharuskan membuat Server sendiri lagi, kita hanya perlu koneksi internet saja, untuk skema penggunaan firebase dalam teknologi IoT bisa dilihat pada gambar dibawah ini



Gambar 2.2.16 Skema Penggunaan Firebase dalam IoT

Firebase Database tidak berbasis Mysql, melainkan berbasis *NoSQL* (*Not Only SQL*) sehingga terdapat perbedaan dengan database relasional seperti Oracle , MySql dan yang lainnya tidak terdapat tabel,kolom,baris, itu dikarenakan Firebase menggunakan penyimpanan berbentuk JSON seperti yang sudah disinggung diatas, dimana JSON merupakan singkatan dari *JavaScript Object Notation* adalah Suatu Java script yang merepresentasikan suatu struktur data sederhana dan array asosiatif, Atau dengan kata lain , JSON adalah suatu format pengiriman data yang berbasis Javascript yang dibuat sedemikian rupa hingga mempermudah dibaca secara manusia.

Format JSON dalam 1 record ditandai dengan kurung kurawal, selanjutnya ditulis nama field dan diikuti dengan titik dua , baru dikuti dengan nilai dari kolom tersebut. Secara umum formatnya sebagai berikut

{[nama Field] : [Nilai Field] , [nama Field2] : [Nilai Field] ...}

Keunggulan dan kekurangan basis data *NoSQL* dibanding SQL adalah sebagai berikut :

Keunggulan :

1. Menggunakan OOP dalam pengaksesan atau manipulasi datanya.
2. *NoSQL* tidak mengenal schema tabel yang kaku dengan format data yang kaku. *NoSQL* sangat cocok untuk data yang tidak terstruktur, istilah singkat untuk fitur ini adalah Dynamic Schema.
3. Autosharding, istilah sederhananya, jika database *NoSQL* di jalankandi cluster Server (multiple Server) maka data akan tersebar secara otomatis dan merata keseluruh Server.
4. *NoSQL* bisa menampung data yang terstruktur, semi terstruktur dan tidak terstuktur secara efesien dalam skala besar (*big data/cloud*).

Kelemahan :

1. Biaya hostingnya mahal.
2. Sulitnya mencari hosting Cpanel yang mendukung database MongoDB atau database *NoSQL* lainnya.
3. karena bervariasinya produk dan format penyimpanan, berpindah antar satu produk database ke produk *NoSQL* lainnya perlu waktu untuk belajar.

Contohnya ketika anda pindah dari MongoDB ke Cassandra, maka anda harus belajar lagi dari awal, berbeda dengan database relasional.

4. Kebanyakan database *NoSQL* tidak melakukan transaksi *ACID* (*Atomicity, Consistency, Isolation and Durability*) , teknik mencoba dan validasi untuk memastikan bahwa data tetap konsisten di seluruh database selama transaksi.

2.2.7 Simulasi

Simulasi adalah sebuah motode pelatihan yang memeragakan sesuatau dalam bentuk tiruan yang mirip dengan keadaan sesungguhnya atau dengan kata lain penggambaran suatu sistem atau proses yang berupa pemeranan.

Simulasi digunakan dalam penelitian ini sebagai penerapan sebuah metode dalam pembangunan *Prototype* yang akan dibuat.

2.2.8 Metode Pengujian

Metode pengujian yang digunakan dalam membangun sistem ini adalah dengan menggunakan pengujian komponen fungsional utama pada sistem, dimana cara pengujian adalah dengan cara menguji fungsi-fungsi pada setiap komponen. Pengujian setiap komponen dilakukan dengan menguji setiap fungsional pada aplikasi *android*, tombol yang terdapat pada alat, serta pengukuran lama pengisian dan penggunaan sumber daya cadangan. [10]

Metode pengujian ini digunakan sebagai acuan metode pada saat pengujian alat dan *software* agar bisa ditemukan dimana kekurangan yang terdapat pada sistem ini sehingga dapat diperbaiki lagi agar menjadi lebih baik