

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penelitian sebelumnya yang dilakukan oleh Andi Widiyanto dan Nuryanto tahun 2014 dalam jurnal yang berjudul “Rancang Bangun Mobil *Remote Control* Android dengan Arduino” pada penelitian ini Android digunakan sebagai remote control untuk mengendalikan mobil dimana Arduino sebagai sistem minimumnya.

Penelitian ini bertujuan untuk memudahkan kerja manusia dalam mengendalikan remote mobil kalau kunci asli atau remotenya tertinggal atau hilang.

Cara kerja alat pada penelitian ini pada Alat yang menjadi obyek penelitian adalah sebuah mobil *Remote Control* (RC) yang selama ini dikontrol melalui *remote* dengan gelombang radio, akan dikontrol menggunakan Android melalui *bluetooth*. Rangkaian elektronik akan digantikan dengan Arduino yang akan mengontrol gerakan motor DC. Kecepatan putaran dan arah putaran motor dikendalikan oleh arduino. Hubungan antara Android dengan mobil RC dirancang Komunikasi antara Android dengan dengan mobil RC melalui koneksi *bluetooth*. Proses awal adalah *pairing* untuk menghubungkan Android dengan Arduino pada mobil RC dengan otentikasi *password* untuk pertama kali supaya hanya *smartphone* tertentu saja yang dapat digunakan. Setelah *pairing* berhasil, Arduino siap untuk menerima perintah dari Android melalui *bluetooth*. Arduino akan menerima data dari Android dan menterjemahkannya untuk menggerakkan motor DC, sehingga mobil RC dapat dikendalikan dengan Android dari jarak jauh. Proses kendali akan putus jika komunikasi dengan Android putus baik diputus atau karena diluar jangkauan mundur. Pada saat Android mengirimkan data, mikrokontroler akan memprosesnya kemudian menggerakkan motor melalui relay. Konsep kendali seperti mobil yang sebenarnya yaitu untuk menggerakkan maju atau mundur dengan menginjak pedal gas dan stir diputar ke kanan

atau kekiri. Penerapannya pedal gas menggerakkan motor belakang dan stir menggerakkan motor depan

Dari penelitian yang dilakukan, dapat di ambil kesimpulan bahwa untuk mengendalikan *remote* mobil menggunakan *Smartphone Android control* dengan media transmisi *bluetooth*.

Pada penelitian berikutnya yang di lakukan oleh Slamet Riyadi dan Bambang Eka Purnama tahun 2013 dalam jurnal “Sistem pengendalian keamanan pintu rumah berbasis sms (short message service) menggunakan mikrokontroler atmega 8535” pada penelitian ini untuk mengetahui rangkaian yang dilengkapi dengan *Handphone* digunakan sebagai pembaca sms dan Modem digunakan sebagai penerima.

Penelitian ini bertujuan untuk memudahkan kerja manusia dalam mengendalikan pintu rumah dan mengamankan sebuah rumah dalam aksi pembukaan pintu secara paksa.

Cara kerja alat pada penelitian ini Rangkaian ini terdiri dari tiga blok rangkaian utama yaitu rangkaian sistem minimum Atmega8535 (bisa untuk *download program*), rangkaian driver motor dan rangkaian *power Supply*. Dengan desain ini dimaksudkan agar bentuk fisik control pengendalian keamanan pintu rumah selain itu juga lebih mudah dan lebih efektif dalam memprogram alat tersebut, karena *downloader* tinggal pasang dan melepasnya dari *board* utamanya sehingga bisa langsung diketahui program tersebut berjalan sesuai program yang dimasukkan atau tidak.

Dari penelitian yang dilakukan, dapat di ambil kesimpulan bahwa untuk Mengendalikan keamanan pintu rumah menggunakan sms dengan media *handphone*.

2.1.1 Perbedaan dengan Penelitian Sebelumnya

Dalam laporan akhir yang berjudul “Rancang Bangun Remote Control Pintu Mobil Menggunakan Handphone Bebas Sms Gateway”. Pada rancang bangun alat ini untuk mengendalikan pintu mobil cukup menggunakan Handphone dan

sms. Alat ini menggunakan Arduino dan Sim 900 sebagai sistem minimum. Pada saat kita mengirimkan sms kepada modul sim 900 lalu arduino mendapatkan perintah yang diberikan oleh sim 900, kemudian relay aktif dan menerima perintah yang telah disesuaikan setelah relay menerima maka door lock akan terbuka.

Pada penelitian ini biaya tidak terlalu mahal dan cukup efisien dikarenakan tidak membutuhkan Smartphone Android untuk mengendalikan pintu mobil, karena dengan handphone biasa pun bisa.

2.2 Remote

Remote atau yang biasa disebut pengendali jarak jauh merupakan sebuah alat elektronik yang digunakan untuk mengoperasikan sebuah mesin dari jarak jauh. Istilah remote control juga sering disingkat menjadi remot saja. Remote juga sering mengacu pada istilah controller, donker, donker, doofer, zipper, click-buzz, box, flipper, zippity clicker or changer. Pada umumnya, pengndali jarak jauh digunakan untuk memberikan perintah dari kejauhan kepada mobil, televisi atau barang-barang elektronik lainnya seperti system stereo dan pemutar DVD. Remote untuk perangkat-perangkat ini biasanya berupa benda kecil nirkabel yang digenggam dengan sederetan tombol untuk menyesuaikan berbagai setting, seperti misalnya untuk membuka atau mengunci mobil, saluran televisi dan lain-lain.

Remote control memiliki segala control fungsi sementara perangkat yang dikendalikan itu sendiri hanya mempunyai sedikit control utama yang mendasar. Kebanyakan remote berkomunikasi dengan perangkatnya masing-masing melalui sinyal-sinyal infra merah dan beberapa saja melalui sinyal radio. Remote control biasanya menggunakan baterai AAA yang kecil atau AA sebagai satu dayanya. (Cahaya : 2014)

2.2.1 Cara Kerja Remote

Pada umumnya cara kerja remote control ini mirip dengan cara kerja sandi morse yang dikirim melalui mesin telegraf. Seorang operator pengirim mengirimkan pesan teks singkat kepada operator penerima yang berada pada jarak

tertentu. Namun pesan tersebut dikirimkan dalam bentuk pola kode-kode morse yang melambangkan huruf-huruf dalam pesan yang dikirimkannya. Mesin telegraf menggunakan kode tertentu karena tidak dapat mengirimkan data suara seperti pesawat telepon. Tetapi telegraf dapat mengirimkan arus listrik yang terhubung ke sebuah bel pada bagian penerima, sehingga operator penerima akan menerima suara dari bel dalam pola-pola tertentu yang apabila dirangkai akan dapat diterjemahkan sebagai pesan singkat.

Remote control menggunakan LED (Light Emitting Diode) infra merah yang berfungsi sebagai pengirim(transmitter) pola sinar infra merah. LED infra merah adalah sejenis lampu kecil yang memiliki dioda yang akan memancarkan cahaya infra merah apabila diberi arus.

Sinyal infra merah yang dikirimkan tidak akan dapat dilihat oleh mata kita, karena sinar infra merah tidak termasuk gelombang elektromagnetik pada spectrum cahaya tampak. Namun sinar tersebut dapat terbaca oleh receiver yang ada pada peralatan elektronik yang menerima sinyal tersebut. Receiver yang digunakan adalah sebuah foto transistor infra merah. Jika pola sinyal infra Red yang diterima bersesuaian dengan salah satu instruksi, seperti instruksi menaikkan volume suara pada pesawat televisi, maka volume suara pesawat televisi tersebut akan dinaikkan. Jika pola sinar infra merah yang dibaca tidak dapat dikenali maka pesawat televisi akan mengabaikannya. Hal ini mungkin saja terjadi jika sebuah pesawat remote control untuk peralatan lain yang berada tidak jauh dari pesawat televisi tersebut sedang digunakan. Bentuk kode sinyal tersebut untuk masing-masing tombol tergantung kepada perusahaan produsen peralatan elektronika. Pada dasarnya setiap perusahaan bebas menentukan kode sinyal untuk setiap tombol pada pesawat remote control.

Penggunaan sinyal sinar infra merah ini memang hanya cocok untuk keperluan di dalam ruang, seperti pada peralatan elektronik rumah atau kantor, karena selain memiliki keterbatasan jarak yang pendek(maksimal sekitar 10 meter), sudut pengiriman juga sangat kecil sehingga remote control harus diarahkan ke tepat ke alat elektronik tersebut. Sinar infra merah juga tidak bisa menembus dinding, sehingga harus berada di ruang.

2.3.2 Macam – macam Jenis Remote

1. Remote mobil

Pada dasarnya kunci remote hanya digunakan sebatas mengunci pintu mobil dan membukanya. Hanya menekan tombol yang ada pada remote, maka pintu mobil akan terkunci dan sebaliknya. Di remote mobil yang menggunakan baterai ini, terdapat tiga sampai empat tombol. Dua tombol bergambar gembok, satu tombol bergambar pintu bagasi, dan satu tombol lagi bergambar alarm.



Gambar 2.1 Remote Mobil

2. Remote TV

Sebuah remote control jarak dekat adalah komponen dari sebuah perangkat elektronik yang paling sering di gunakan pada televisi, DVD player, atau home theater. Remote tv ini terdapat cukup banyak tombol-tombol yang berfungsi untuk mengendalikan saluran tv, volume dan lain-lain.



Gambar 2.2 Remote TV

2.3 Handphone

Handphone merupakan alat telekomunikasi elektronik dua arah yang bisa dibawa kemana-mana dan memiliki kemampuan untuk mengirimkan pesan berupa suara. Pengertian tersebut merupakan pengertian handphone secara umum. Dalam keseharian kini manusia hampir tidak bisa lepas dari handphone. Apalagi dengan semakin berkembangnya handphone sehingga handphone memiliki berbagai fungsi sekaligus. Bukan hanya sebagai alat komunikasi saja namun telah berkembang menjadi alat dengan fungsi lainnya seperti sebagai media hiburan, media bisnis, dan sebagainya. Kini kita mengenal istilah smartphone atau ponsel pintar. Sebutan untuk handphone yang bisa digunakan untuk melakukan banyak hal. Sebelum handphone memiliki fungsi seperti sekarang ini, handphone telah mengalami perjalanan yang panjang sejak awal kemunculannya. (<https://vartikel.com/6505/pengertian-cara-kerja-handphone-dengan-ringkas/>)

2.3.1 Cara Kerja Handphone

Handphone adalah radio dua arah, seperti mengirim dan menerima sinyal radio melalui jaringan BTS. Ketika seseorang berbicara ditelepon seluler, pesan suara akan dikonversi menjadi gelombang radio. Gelombang radio berjalan memalui udara sampai mencapai BTS terdekat. BTS kemudian mengirimkan

panggilan ke jaringan komunikasi pada penerima yang dituju. BTS yang membentuk jaringan telepon dilengkapi dengan antena dan biasanya tersusun dengan bentuk struktur tinggi seperti tiang atau menara.

Handphone memperoleh daya listrik dari baterai yang dapat diisi ulang. Baterai lithium ion adalah yang sering digunakan untuk ponsel. Di bawah baterai handphone terdapat microchip yang dikenal dengan nama SIM card – Subscriber Identification Module. Chip kecil ini menyimpan konfigurasi telepon dari informasi telepon.

2.3.2 Macam – Macam Handphone

1. Handphone Monophonic

Ponsel pertama yang berkembang diseluruh negara adalah handphone monophonic. Handphone monophonic adalah ponsel pertama yang hanya menampilkan nada dasar pada frekuensi yang berbeda. Monophonic memiliki satu instrumen yang dapat memainkan nada satu ketukan.

Handphone monophonic hanya dapat digunakan untuk melakukan panggilan dan pesan singkat atau SMS. Ponsel pertama monophonic adalah yang terdapat antena pada ponselnya. Kemudian setelah itu berkembang menjadi ponsel layar kuning dan biru yang telah dilengkapi game dan memori.



Gambar 2.3 Handphone Monophonic

2. Handphone Android

Handphone Android merupakan ponsel terpopuler saat ini yang banyak dimiliki berbagai kalangan. Android merupakan besutan dari perusahaan terbesar, yaitu Google. OS Android telah digunakan oleh PDA phone layar sentuh ternama, HTC. Android memiliki teknologi yang canggih yang dapat mengalahkan Microsoft. Dengan aplikasi seperti Maps hingga Browsernya yang lebih cepat dan mudah, membuat ponsel Android banyak diminati.



Gambar 2.4 Handphone android

2.4 Sms Gateway

SMS Gateway adalah suatu platform yang menyediakan mekanisme untuk menghantar dan menerima SMS dari peralatan mobile (HP, Smartphone, dan lain-lain) yang menggunakan keyword tertentu. SMS Gateway adalah komunikasi SMS dua arah. SMS Gateway merupakan salah satu perkembangan fungsi yang dimiliki SMS.

Secara umum SMS Gateway adalah sebuah sistem yang dipergunakan untuk memudahkan seseorang atau sebuah perusahaan mengirimkan pesan SMS yang sama dalam waktu yang bersamaan pada banyak orang. Selain itu, semakin berkembangnya fungsi SMS. SMS Gateway juga dapat dimanfaatkan untuk keperluan lain seperti melakukan polling, transaksi dengan sebuah sistem,

pemantauan, dan sebagainya. (<http://informatika.web.id/mechanisme-kerja-sms.htm#more-1453>)

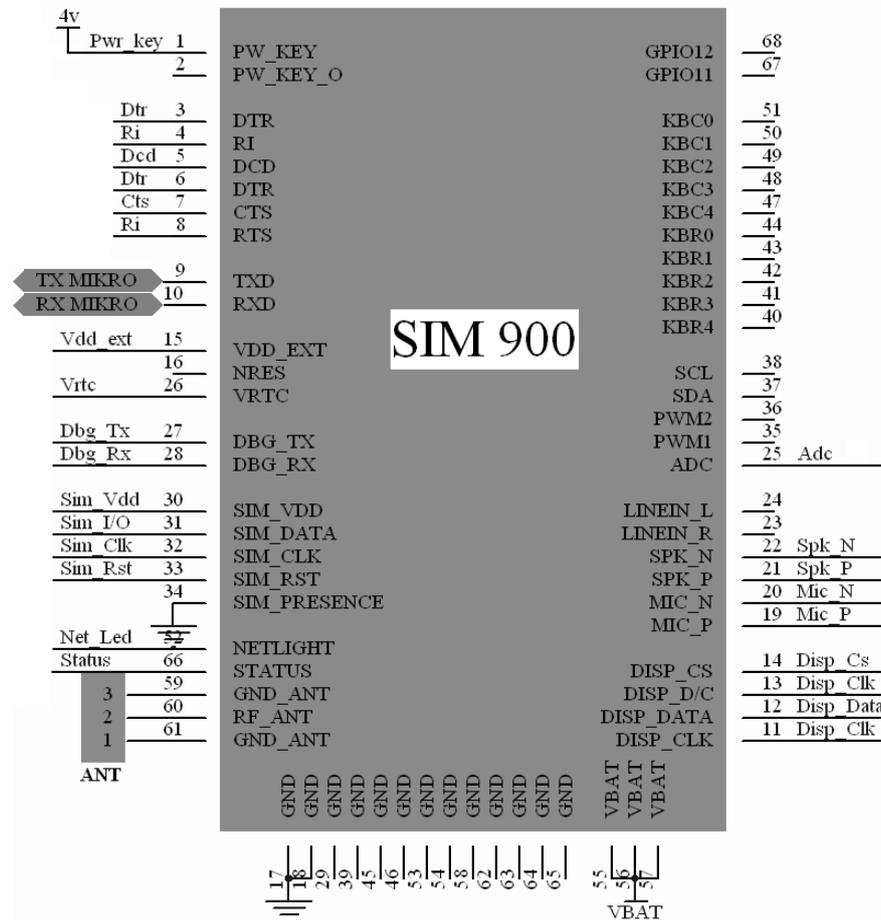
2.4.1 Cara Kerja Sms Gateway

Ketika Sms dikirim ke suatu nomor tertentu, Sms yang dikirimkan tidak akan langsung dikirimkan ke nomor tersebut, namun akan masuk terlebih dahulu ke sms center (SMSC) operator telepon yang Anda gunakan. Sms *Center* sendiri dapat diartikan sebagai sebuah *server* yang bertanggung jawab pada proses pengiriman sms dalam suatu operator. Sms yang dikirimkan dari suatu ponsel akan masuk ke SMSC ini, kemudian baru diteruskan ke nomor tujuan sms tersebut. Bila nomor yang dituju ternyata sedang mati/*offline*, SMSC ini akan menyimpan sms tersebut untuk sementara waktu, hingga nomor tujuan hidup kembali. Lamanya waktu penyimpanan sms, sangat tergantung dari lamanya waktu yang telah ditetapkan oleh operator untuk menyimpan Sms tersebut. Nomor yang telah menerima sms akan mengirimkan laporan ke SMSC bahwa Sms telah diterima. Laporan tersebut kemudian akan diteruskan kembali ke nomor pengirim sms. (<http://informatika.web.id/mechanisme-kerja-sms.htm#more-1453>)

2.4.2 Modul Sim 900A

Modul GSM SIM900 merupakan perangkat yang dapat menggantikan fungsi dari *handphone*. Untuk komunikasi data antara sistem Arduino via jaringan seluler, maka digunakan Modul GSM SIM900 yang difungsikan sebagai media pengirim dan penerima SMS (*Short Message Service*). Modem ini bertugas mengirim SMS berupa data peringatan pada user pada saat sensor dari alarm aktif. Adapun protokol komunikasi yang digunakan adalah komunikasi standard modem yaitu AT Command. (Bambang Tri : 2016)

2.4.3 Arsitektur Modul Sim 900A



Gambar 2.5 Konfigurasi Pin GSM SIM900

(Sumber : *datasheets/SIM900_AT.*)

Modul komunikasi GSM/GPRS menggunakan core IC SIM900A. Modul ini mendukung komunikasi *dual band* pada frekuensi 900 / 1800 MHz (GSM900 dan GSM1800) sehingga fleksibel untuk digunakan bersama kartu SIM dari berbagai operator telepon seluler di Indonesia. Operator GSM yang beroperasi di frekuensi *dual band* 900 MHz dan 1800 MHz sekaligus: Telkomsel, Indosat, dan XL. Operator yang hanya beroperasi pada band 1800 MHz: Axis dan Three. (Sumber : <https://store.fut-electronics.com>)

Tampilan dari konfigurasi pin GSM SIM900.



Gambar 2.6 Tampilan modul GSM SIM900.

(Sumber : <https://store.fut-electronics.com>)

Spesifikasi modul GSM SIM900A :

- a. GPRS multi-slot class 10/8, kecepatan transmisi hingga 85.6 kbps (*downlink*), mendukung PBCCH, PPP *stack*, skema penyandian CS 1,2,3,4
- b. GPRS mobile station class B
- c. Memenuhi standar GSM 2/2 +
 - o Class 4 (2 W @ 900 MHz)
 - o Class 1 (1 W @ 1800MHz)
- d. SMS (Short Messaging Service): point-to-point MO & MT, SMS cell broadcast, mendukung format teks dan PDU (*Protocol Data Unit*)
- e. Dapat digunakan untuk mengirim pesan MMS (*Multimedia Messaging Service*)
- f. Mendukung transmisi faksimili (*fax group 3 class 1*)
- g. *Handsfree mode* dengan sirkit reduksi gema (*echo suppression circuit*)
- h. Dimensi: 24 x 24 x 3 mm
- i. Pengendalian lewat perintah AT (GSM 07.07, 07.05 & SIMCOM Enhanced AT Command Set)

- j. Rentang catu daya antara 7 Volt hingga 12 Volt DC
- k. SIM Application Toolkit
- l. Hemat daya, hanya mengkonsumsi arus sebesar 1 mA pada moda tidur
- m. Rentang suhu operasional: -40 °C hingga +85 °C

2.4.4 Cara kerja sim 900

Pengolahan dan akses data dari sebuah sistem selular pada komputer biasa dapat menggunakan perangkat Modem GSM ataupun CDMA dengan menggunakan perintah AT comment. Selain menggunakan modem juga dapat menggunakan handphone type lama yang masih menggunakan perintah sistem AT coment. Berikut ini beberapa perintah “AT Command” yang biasa digunakan pada modul GSM SIM900A : (Sumber : <https://store.fut-electronics.com>)

AT+CPBF : cari no telpon

AT+CPBR : membaca buku telpon

AT+CPBW : menulis no telp di buku telpon

AT+CMGF : menyeting mode SMS text atau PDU

AT+CMGL : melihat semua daftar sms yg ada.

AT+CMGR : membaca sms.

AT+CMGS : mengirim sms.

AT+CMGD : menghapus sms.

AT+CMNS : menyeting lokasi penyimpanan ME(hp) atau SM(SIM Card)

AT+CGMI : untuk mengetahui nama atau jenis ponsel

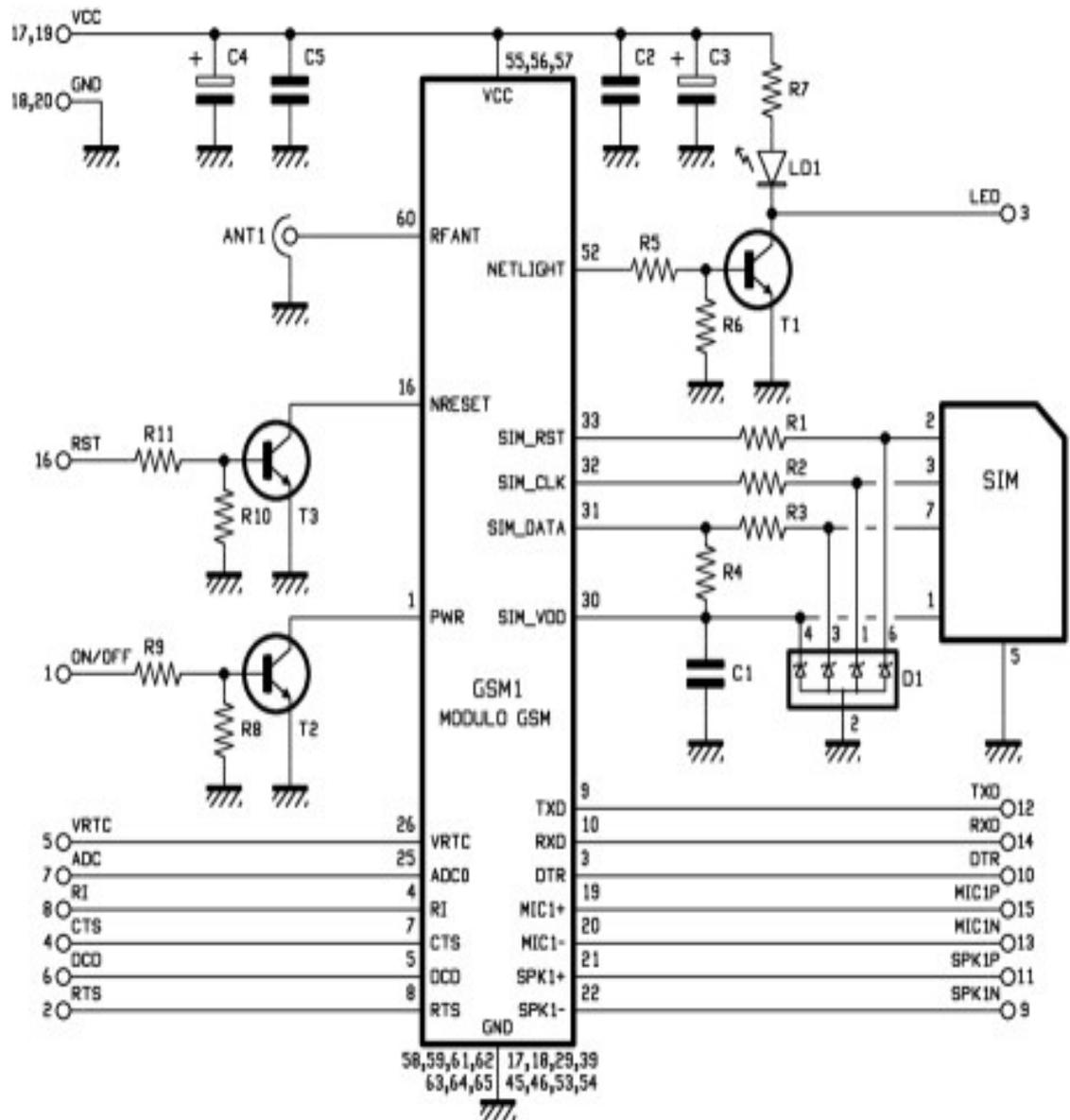
AT+CGMM : untuk mengetahui kelas ponsel

AT+COPS? : untuk mengetahui nama provider kartu GSM

AT+CBC : untuk mengetahui level baterai

AT+CSCA : untuk mengetahui alamat SMS Center

2.4.5 Blog Diagram Modul Sim 900A



Gambar 2.7 Diagram Blok Modul Sim 900A
 (Sumber : <https://store.fut-electronics.com>)

2.5 Mikrokontroler Arduino

2.5.1 Arduino

Arduino adalah sebuah platform komputasi fisik *open source* berbasis Rangkaian *input / output* sederhana (I/O) dan lingkungan pengembangan yang mengimplementasikan bahasa *Processing*. Arduino dapat digunakan untuk mengembangkan obyek interaktif mandiri atau dapat dihubungkan ke perangkat lunak pada komputer anda (seperti Flash, Pengolahan, VVVV, atau Max / MSP). Rangkaiannya dapat dirakit dengan tangan atau dibeli. IDE (*Integrated Development Environment*) Arduino bersifat *open source*. (Steven Jendri : 2016)

2.5.2 Arsitektur Arduino Uno R3

Pada perancangan dan pembuatan tugas akhir ini digunakan jenis papan arduino Uno R3. Arduino Uno adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328. (Steven Jendri : 2016)

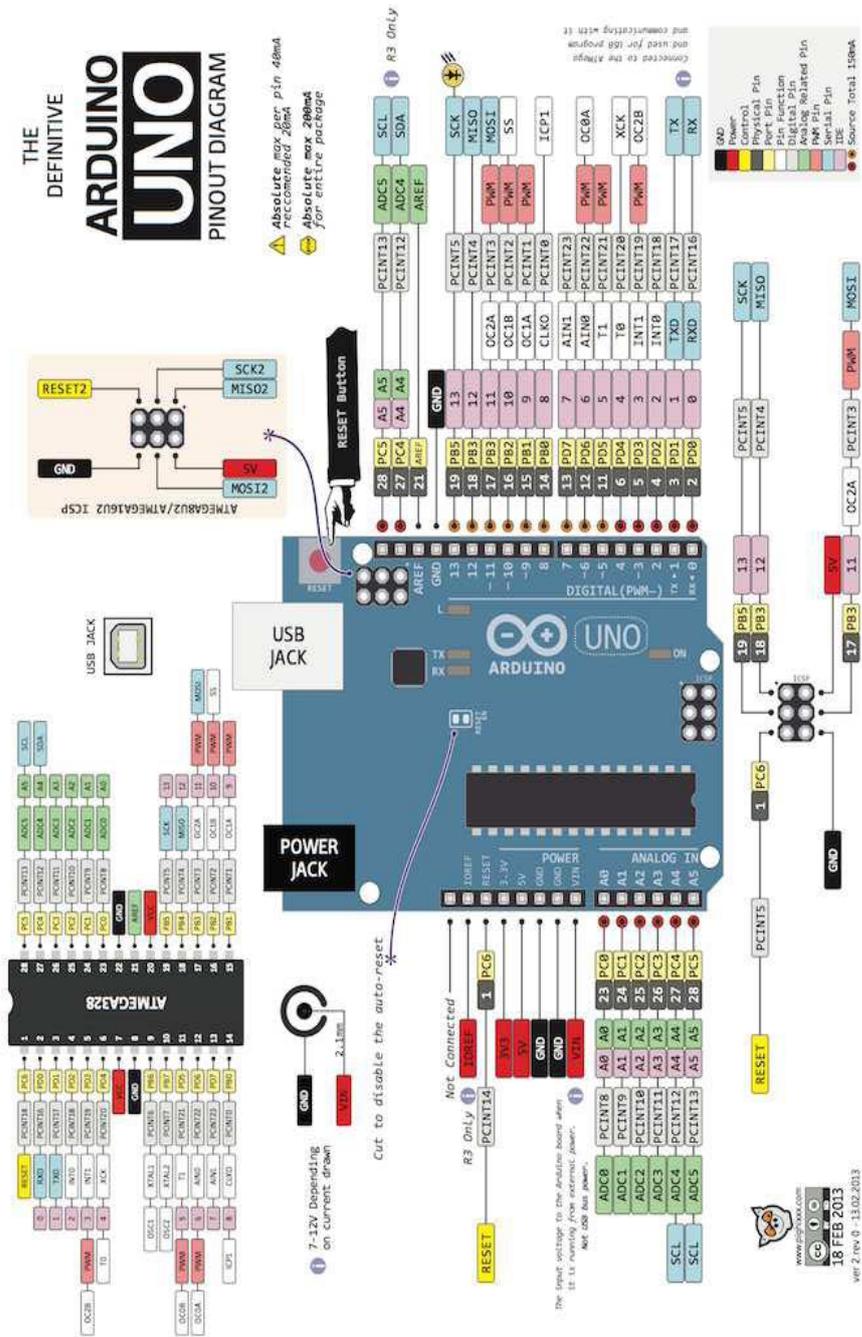
Seperti pada gambar 1. Spesifikasi Arduino Uno R3 :

- a. Mikrokontroler : ATmega328
- b. Tegangan pengoperasian : 5V Tegangan input yang
- c. disarankan: 7-12V
- d. Batas tegangan input : 6-20V
- e. Jumlah pin I/O digital : 14
- f. Jumlah pin input analog : 6
- g. Arus DC tiap pin I/O : 40 mA
- h. Arus DC untuk pin 3.3V : 50 mA
- i. Memori :32 KB (ATmega328),
- j. sekitar 0.5 KB digunakan
- k. oleh bootloader
- l. SRAM : 2 KB (ATmega328)
- m. EEPROM : 1 KB (ATmega328)
- n. Clock Speed : 16 MHz



Gambar 2.8 Papan Arduino Uno R3 (Steven Jendri : 2016)

2.5.3 Blok Diagram Arduino Uno



2.5.4 Karakteristik Arduino Uno R3

2.5.4.1 Daya (Power)

Arduino UNO dapat disuplai melalui koneksi USB atau dengan sebuah *power* suplai eksternal. Sumber daya dipilih secara otomatis. Suplai eksternal (non-USB) dapat diperoleh dari sebuah adaptor AC ke DC atau *battery*. Adaptor dapat dihubungkan dengan mencolokkan sebuah *center-positive plug* yang panjangnya 2,1 mm ke *power jack* dari *board*. Kabel lead dari sebuah *battery* dapat dimasukkan dalam *header/kepala pin Ground (Gnd)* dan pin *Vin* dari konektor POWER.

Board Arduino UNO dapat beroperasi pada sebuah suplai eksternal 6 sampai 20 Volt. Jika disuplai dengan yang lebih kecil dari 7 V, kiranya pin 5 Volt mungkin mensuplai kecil dari 5 Volt dan board Arduino UNO bisa menjadi tidak stabil. Jika menggunakan suplai yang lebih dari besar 12 Volt, voltage regulator bisa kelebihan panas dan membahayakan board Arduino UNO. Range yang direkomendasikan adalah 7 sampai 12 Volt. . (Steven Jendri : 2016)

2.5.4.2 Memori

Memori yang digunakan pada Aduino Uno R3 adalah ATmega328 yang mempunyai 32 KB (dengan 0,5 KB digunakan untuk *bootloader*). ATmega 328 juga mempunyai 2 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis (RW/*read and written*) dengan EEPROM *library*). (Steven Jendri : 2016)

2.5.4.3 Input dan Output

Setiap 14 pin digital pada Arduino Uno dapat digunakan sebagai *input* dan *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Fungsi-fungsi tersebut beroperasi di tegangan 5 Volt. Setiap pin dapat memberikan atau menerima suatu arus maksimum 40 mA dan mempunyai sebuah resistor pull-up (terputus secara default) 20-50 k Ω . (Steven Jendri : 2016)

2.5.4.4 Komunikasi

Arduino UNO mempunyai sejumlah fasilitas untuk komunikasi dengan sebuah komputer, Arduino atau mikrokontroler lainnya. Atmega 328

menyediakan serial komunikasi UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah Atmega 16U2 pada channel board serial komunikasinya melalui USB dan muncul sebagai sebuah port virtual ke software pada komputer. Firmware 16U2 menggunakan driver USB COM standar, dan tidak ada driver eksternal yang dibutuhkan. LED RX dan TX pada board akan menyala ketika data sedang ditransmit melalui chip USB-to-serial dan koneksi USB pada komputer (tapi tidak untuk komunikasi serial pada pin 0 dan 1). Atmega328 juga mensupport komunikasi I2C (TWI) dan SPI. (Steven Jendri : 2016)

2.5.4.5 Riset Otomatis

Dari pada mengharuskan sebuah penekanan fisik dari tombol reset sebelum sebuah penguploadan, Arduino Uno didesain pada sebuah cara yang memungkinkannya untuk direset dengan *software* yang sedang berjalan pada komputer yang sedang terhubung. (Steven Jendri : 2016)

2.5.4.6 Proteksi Arus Lebih USB

Arduino UNO mempunyai sebuah sekering reset yang memproteksi *port* USB komputer dari hubungan pendek dan arus lebih. Jika lebih dari 500 mA diterima port USB, sekering secara otomatis akan memutuskan koneksi sampai hubungan pendek atau kelebihan beban hilang. (Steven Jendri : 2016)

2.5.4.7 Karakteristik Fisik Arduino Uno R3

Panjang dan lebar maksimum dari PCB Arduino UNO masing-masingnya adalah 2.7 dan 2.1 inci, dengan konektor USB dan power jack yang memperluas dimensinya. Empat lubang sekrup memungkinkan board untuk dipasangkan ke sebuah permukaan atau kotak. Sebagai catatan, bahwa jarak antara pin digital 7 dan 8 adalah 160 mil. (0.16"), bukan sebuah kelipatan genap dari jarak 100 mil dari pin lainnya. Seperti pada gambar gambar 2.5.8 pin out diagram pada Arduino Uno R3. (Steven Jendri : 2016)

2.5.5 Bahasa Pemrograman Arduino

Arduino board merupakan perangkat yang berbasis mikrokontroler. Perangkat lunak (software) merupakan komponen yang membuat sebuah mikrokontroler dapat bekerja. Arduino board akan bekerja sesuai dengan perintah yang ada dalam perangkat lunak yang ditanamkan padanya.

Bahasa Pemrograman Arduino adalah bahasa pemrograman utama yang digunakan untuk membuat program untuk arduino board. Bahasa pemrograman arduino menggunakan bahasa pemrograman C sebagai dasarnya.

Karena menggunakan bahasa pemrograman C sebagai dasarnya, bahasa pemrograman arduino memiliki banyak sekali kemiripan, walaupun beberapa hal telah berubah. (Steven Jendri : 2016)

2.5.5.1 Struktur

Setiap program Arduino (biasa disebut sketch) mempunyai dua buah fungsi yang harus ada.

- `void setup() { }`

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

- `void loop() { }`

Fungsi ini akan dijalankan setelah setup (fungsi void setup) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (power) dilepaskan. (Steven Jendri : 2016)

2.5.5.2 Syntax

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

- `)/(komentar satu baris)`

Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang diketikkan dibelakangnya akan diabaikan oleh program.

- `/* */(komentar banyak baris)`

Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

- { }(kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

- ;(titik koma)

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan. (Steven Jendri : 2016)

2.5.5.3 Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya.

- int (integer)

Digunakan untuk menyimpan angka dalam 2 byte (16bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -32,768 dan 32,767.

- long (long)

Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -2,147,483,648 dan 2,147,483,647.

- boolean (boolean)

Variabel sederhana yang digunakan untuk menyimpan nilai TRUE (benar) atau FALSE (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.

- float (float)

Digunakan untuk angka desimal (floating point). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38.

- char (character)

Menyimpan 1 karakter menggunakan kode ASCII(misalnya 'A' = 65).
Hanya memakai 1 byte (8 bit) dari RAM. (Steven Jendri : 2016)

2.5.5.4 Operator

1. Operator Matematika

Operator yang digunakan untuk memanipulasi angka bekerja seperti matematika yang sederhana. (Steven Jendri : 2016)

- =

Membuat sesuatu menjadi sama dengan nilai yang lain(misalnya: $x = 10 * 2$,
x sekarang sama dengan 20).

- %

Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain
(misalnya: $12 \% 10$, ini akan menghasilkan angka 2).

- +

Penjumlahan

- -

Pengurangan

- *

Perkalian

- /

Pembagian

2. Operator Pembandingan

Digunakan untuk membandingkan nilai logika. (Steven Jendri : 2016)

- ==

Sama dengan (misalnya: $12 == 10$ adalah FALSE (salah) atau $12 == 12$
adalah TRUE(benar))

- !=

Tidak sama dengan (misalnya: $12 != 10$ adalah TRUE (benar) atau $12 != 12$
adalah FALSE(salah))

- <

Lebih kecil dari (misalnya: $12 < 10$ adalah FALSE (salah) atau $12 < 12$ adalah FALSE(salah) atau $12 < 14$ adalah TRUE (benar))

- >

Lebih besar dari (misalnya: $12 > 10$ adalah TRUE (benar) atau $12 > 12$ adalah FALSE(salah) atau $12 > 14$ adalah FALSE (salah))

2.5.5.5 Struktur Kondisi

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan. (Steven Jendri : 2016)

1. **if..else**, dengan format seperti berikut ini:

```
if (kondisi) { }
else if (kondisi) { }
else { }
```

Dengan struktur seperti diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisinya TRUE, dan jika tidak (FALSE) maka akan diperiksa apakah kondisi pada else if dan jika kondisinya FALSE maka kode pada else yang akan dijalankan.

2. **for**, dengan format seperti berikut ini:

```
for (int i = 0; i < #pengulangan; i++) { }
```

Digunakan bila anda ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan i++ atau ke bawah dengan i.

2.5.5.6 Digital

1. **pinMode(pin, mode)**

Digunakan untuk menetapkan mode dari suatu pin, pin adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19). Mode yang bisa digunakan adalah INPUT atau OUTPUT.

2. **digitalWrite(pin, value)**

Ketika sebuah pin ditetapkan sebagai OUTPUT, pin tersebut dapat dijadikan HIGH (ditarik menjadi 5 volts) atau LOW (diturunkan menjadi ground).

3. digitalRead(pin)

Ketika sebuah pin ditetapkan sebagai INPUT maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah HIGH (ditarik menjadi 5 volts) atau LOW (diturunkan menjadi ground). (Steven Jendri : 2016)

2.5.5.7 Analog

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam alam analog (menggunakan trik). Berikut ini cara untuk menghadapi hal yang bukan digital.

1. analogWrite(pin, value)

Beberapa pin pada Arduino mendukung PWM (pulse width modulation) yaitu pin 3, 5, 6, 9, 10, 11. Ini dapat merubah pin hidup (on) atau mati (off) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. Value (nilai) pada format kode tersebut adalah angka antara 0 (0% duty cycle ~ 0V) dan 255 (100% duty cycle ~ 5V).

2. analogRead(pin)

Ketika pin analog ditetapkan sebagai INPUT anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1024 (untuk 5 volts). (Steven Jendri : 2016)

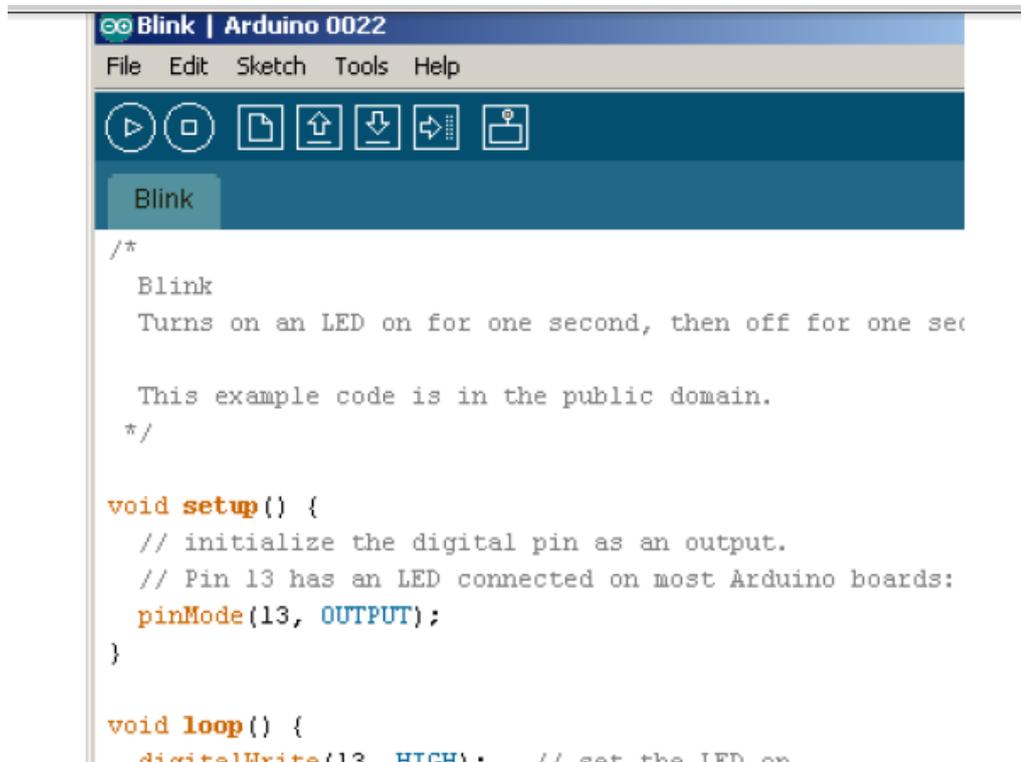
2.5.6 Software Arduino IDE

Sehubungan dengan pembahasan untuk saat ini software arduino yang akan digunakan adalah driver dan IDE, walaupun masih ada beberapa software lain yang sangat berguna selama pengembangan Arduino. (Simanjuntak : 2013, 11)

IDE arduino adalah software yang sangat canggih ditulis dengan menggunakan java. IDE Arduino terdiri dari :

- Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
- Compiler, sebuah modul yang mengubah kode program (bahasa processing) menjadi kode biner. Bagaimanapun sebuah microcontroller tidak akan bisa memahami bahasa processing. Yang bisa dipahami oleh microcontroller adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.

- Uploader, sebuah modul yang memuat kode biner dari computer ke dalam memory dalam papan arduino.



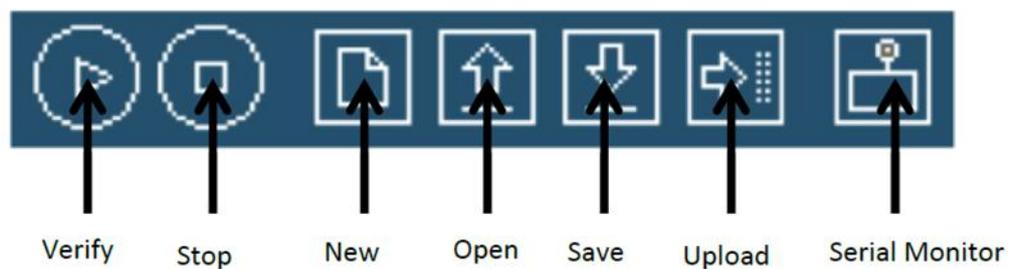
Gambar 2.10 Tampilan Arduino IDE

(Agung : 2014, 13)

Pada gambar 2.10 anda dapat melihat toolbar IDE yang memberikan akses instan ke fungsi-fungsi yang penting :

- Dengan tombol Verify, anda dapat mengkompilasi program yang saat ini di editor.
- Tombol New menciptakan program baru dengan mengosongkan isi dari jendela editor saat ini. Sebelum hal itu terjadi, IDE memberikan anda kesempatan untuk menyimpan semua perubahan belum disimpan.
- Dengan Open anda dapat membuka program yang ada dari sistem file.
- Tombol Save menyimpan program saat ini.
- Ketika anda mengklik tombol Upload, IDE mengkompilasi saat ini program dan upload ke papan Arduino yang telah anda pilih di IDE menu Tools > Serial port.

- Arduino dapat berkomunikasi dengan komputer melalui koneksi serial. Mengklik tombol serial monitor membuka jendela serial monitor yang memungkinkan anda dapat melihat data yang dikirimkan oleh arduino dan juga untuk mengirim data kembali.
- Tombol stop menghentikan serial monitor



Gambar 2.11 Toolbar Arduino IDE

(Agung : 2014, 14)

Meskipun menggunakan IDE sangat mudah, anda mungkin mengalami masalah. Dalam kasus tersebut, kita lihat menu Help. Menu Help menunjukkan banyak sumber daya yang berguna di website arduino yang menyediakan solusi cepat tidak hanya untuk semua masalah khas tetapi juga untuk referensi materi dan tutorial.

Untuk dapat memahami fitur-fitur IDE yang paling penting, kita akan membuat program-program sederhana yang membuat dioda pemancar cahaya (LED) berkedip. LED merupakan sumber cahaya murah dan efisien, dan arduino sudah dilengkapi dengan beberapa LED. Satu LED yang berkedip menunjukkan apakah Arduino saat ini memiliki daya dan dua LED lainnya berkeip saat data ditransmisikan atau diterima melalui koneksi serial. Dalam proyek kecil pertama anda akan membuat LED arduino yang berkedip. (Simanjuntak : 2013, 11)

2.5.6.1 Tipe-Tipe data dalam Arduino

Setiap bagoan dari data yang anda simpan dalam program arduino memiliki tipe datanya masing-masing. Tergantung pada kebutuhan anda, anda dapat memilih dari tipe-tipe data berikut ini :

1. Tipe data boolean mengambil satu byte memori dan dapat bernilai benar atau salah.
2. Tipe data char mengambil satu byte nomor memori dan menyimpan dari 128 sampai 127. Angka-angka ini biasanya mewakili karakter yang dikodekan dalam ASCII.
3. Tipe data int (integer) membutuhkan dua byte memori. Anda dapat menggunakannya untuk menyimpan angka dari -32.768 ke 32.767. unsigned int juga menghabiskan dua byte memori tetapi menyimpan angka dari 0 sampai 65.535.
4. Untuk angka yang lebih besar, digunakan tipe data long. Mengonsumsi empat byte memori dan menyimpan nilai dari -214783648 ke 2147483647. Unsigned long juga perlu empat byte tetapi menyimpan rentang nilai dari 0 sampai 4.294.967.295.
5. Tipe data float dan double adalah tipe data yang sama. Anda dapat menggunakan jenis tipe ini untuk menyimpan angka floating-point. Keduanya menggunakan empat byte memori dan mampu menyimpan nilai-nilai dari -3.4028235E+38 untuk 3.4028235E+38.
6. Tipe data void hanya untuk deklarasi fungsi. Ini menunjukkan bahwa fungsi tersebut tidak mengembalikan nilai.
7. Array menyimpan nilai yang memiliki tipe data yang sama.
8. Sebuah string adalah sebuah array nilai char. Arduino IDE mendukung penciptaan string dengan beberapa sintaksis gula semua ini deklarasi membuat string dengan isi yang sama. (Simanjuntak : 2013, 11)

2.5.6.2 Komplikasi dan Program Uploading

Sebelum anda mengkompilasi dan meng-upload program ke arduino, anda harus mengkonfigurasi dua hal dalam IDE : jenis Arduino anda menggunakan dan port serial arduino anda terhubung ke. Mengidentifikasi jenis arduino mudah, karena dicetak di papan tulis, jenis populer adalah Uno, Duemilnove, Diecimila, Nano, Mega Mini, NG, BT, Lilypad, Pro atau pro mini. Dalam beberapa kasus anda juga harus memeriksa apa mikrokontroler arduino

anda mengguankan paling memiliki Atmega 168 atau sebuah Atmega328. Anda dapat menemukan jenis mikrokontroler dicetak pada mikrokontroler itu sendiri.

Ketika anda telah mengidentifikasi dengan tepat jenis arduino anda. Memilih dari menu tools> board. Sekarang anda harus memilih port serial arduino anda terhubung untuk dari >menu serial port tools. Pada sistem windows, Device Manager, dan mencari USB Serial Port dibawah ports (COM dan LPT) entri menu.

Biasanya port bernama COM1, COM2, atau sesuatu yang serupa. Setelah anda telah memilih port serial kanan, klik tombol verify dan anda akan melihat output berukut di daerah pesan IDE (yang arduino IDE menyebut program sketsa) : Binary ukuran sketsa : 1010 bytes (dari 32256 byte maksimum) ini berarti IDE berhasil telah menyusun kode sumber ke dalam 1.010 byte kode mesin yang kita dapat meng-upload ke Arduino.

Jika anda melihat pesan kesalahan sebagai gantinya, periksa apakah anda telah mengetik di program yang benar. Tergantung pada papan arduinoyang anda gunakan maksimum byte mungkin berbeda. Misalnya pada arduino Duemilanove biasanya 14336 byte. Dalam kasus kesalahan, periksalah apakah anda memilih jenis arduino benar dan port serial yang benar dalam menu tools.

Selama proses upload, TX dan RX LED akan berkedip selama beberapa detik. Ini adalah normal itu terjadi setiap kali Arduino dan komputer anda berkomunikasi melalui port serial. Ketika arduino mengirimkan informasi ternyata pada TX LED. Ketika mendapat beberapa bit, ternyata pada RX LED. Karena komunikasi ini cukup cepat, LED mulai berkedip dan anda tidak dapat mengidentifikasi transmisi byte tunggal.(Agung : 2014, 20)

2.6 Pengujian Perangkat Lunak

2.6.1 Pengujian Perangkat Lunak

Pengujian adalah proses eksekusi program untuk menemukan kesalahan. Pengujian perangkat lunak (*testing*) merupakan bagian terpenting dalam pengembangan perangkat lunak (*software engineering*). Pengujiannya atau uji coba tersebut bertujuan untuk mencari sebanyak mungkin kesalahan (*bug error*) dan

menemukan kesalahan yang sebelumnya tidak ditemukan, serta untuk mengurangi resiko yang terkandung dalam suatu sistem komputer. Suatu pengujian dikatakan berhasil apabila menemukan kesalahan-kesalahan yang belum terdeteksi. Secara umum pengujian dilakukan untuk menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya.

Beberapa aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak, yaitu :

- a. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
- b. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
- c. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

2.6.2 Strategi Pengujian

Strategi yang dibuat harus mengacu kepada resiko dan proses yang ada yang dapat mengurangi resiko tersebut. Dua komponen dari strategi pengujian adalah :

- a. Faktor pengujian, merupakan resiko atau isu-isu yang ada yang perlu menjadi perhatian.
- b. Fase pengujian, merupakan siklus pengembangan sistem dimana akan dilakukan proses pengujian.

2.6.3 Faktor Pengujian

Dalam merancang suatu pengujian, faktor-faktor yang menjadi resiko merupakan basis atau tujuan dari proses pengujian. Resiko-resiko yang berhubungan dengan proses pengujian ini disebut dengan faktor pengujian. Faktor-faktor pengujian tersebut merupakan kunci sukses suatu perangkat lunak :

- a. *Correctness* (keakuratan dan kelengkapan data).
- b. *File integrity* (pengelolaan *file* secara benar).
- c. *Authorization* (kesesuaian data yang diproses dengan ketentuan-ketentuan secara manajemen).

- d. *Audit trail* (kemampuan untuk merekam ulang suatu proses yang pernah terjadi).
- e. *Continuity of processing* (kemampuan untuk selalu berjalan meskipun terjadi suatu problem).
- f. *Service level* (pelayanan yang diberikan tidak mengecewakan pemakai).
- g. *Access control* (perlindungan dari modifikasi, perusakan, salah penggunaan, yang disengaja atau tidak).
- h. *Compliance* (kesesuaian sistem).
- i. *Reliability* (ketepatan dan kemampuan dari sistem).
- j. *Ease of use* (pemahaman sistem oleh pemakai).
- k. *Maintenance* (kemampuan untuk mencari dan memperbaiki kesalahan dalam pengoperasian sistem).
- l. *Portability* (kemampuan untuk mentransfer program dari satu konfigurasi perangkat keras dan atau perangkat lunak ke konfigurasi yang lain).
- m. *Coupling* (kemampuan untuk menghubungkan komponen dalam sistem aplikasi dengan semua sistem aplikasi dalam suatu lingkungan proses).
- n. *Performance* (rata-rata dari sumber daya komputasi dari kode yang diperlukan oleh sistem untuk menjalankan fungsi yang telah ditetapkan).
- o. *Ease of operation* (kemampuan untuk mengintegrasikan sistem ke dalam lingkungan operasi dan kemudian mengoperasikan sistem aplikasi, baik yang manual maupun otomatis).

2.6.4 Membuat Strategi Pengujian

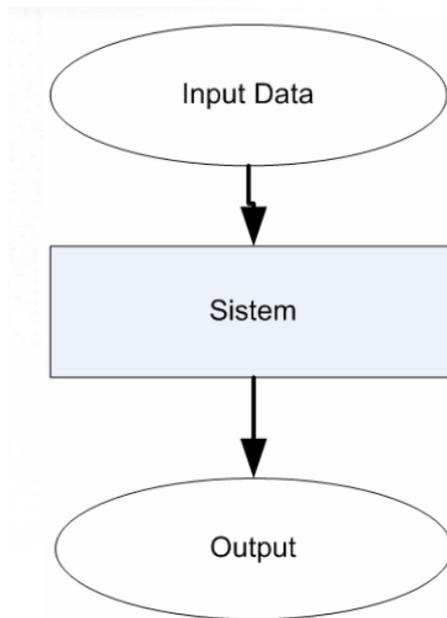
Strategi pengujian disusun berdasarkan resiko-resiko yang telah dipilih. Pada umumnya terdapat empat langkah dalam pembuatan strategi pengujian, sedangkan untuk perangkat lunak yang bersifat khusus memerlukan sedikit modifikasi. Langkah-langkah adalah :

- a. Seleksi dan perangkaan faktor pengujian. Dalam banyak kasus, dari faktor-faktor yang ada cukup diambil 3 – 7 faktor pengujian. Proses ini dilakukan dengan cara membuat daftar dalam satu matriks dan diurutkan dari faktor yang paling penting sampai yang kurang penting.

- b. Identifikasi fase pengembangan sistem. Biasanya sesuai dengan metodologi pengembangan sistem yang digunakan. Fase-fase yang ada kemudian ditulis menjadi salah satu komponen dalam matriks.
- c. Identifikasi resiko-resiko bisnis yang berhubungan dengan sistem yang sedang dikembangkan. Resiko-resiko tersebut kemudian dirangking dari tingkat *high, medium, low*.
- d. Menempatkan resiko yang diperoleh ke dalam matriks. Kemudian menentukan fase dan faktor yang berhubungan dengan resiko yang telah dibuat.

2.6.5 Pengujian Kotak Hitam (*Black Box Testing*)

Pengujian kotak hitam (*black box testing*) dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. *Testing* ini juga disebut sebagai *behavioral testing, specification-based testing input/output* atau *functional testing*. Pengujian kotak hitam merupakan pendekatan pengujian yang detail pengujiannya diturunkan dari spesifikasi program atau komponen. Di mana sistem merupakan “kotak hitam” yang perilakunya ditentukan dengan mempelajari *input* dan *output* yang berkaitan. Pengujian ini hanya menitikberatkan pada fungsionalitas pada *software*, berdasarkan spesifikasi kebutuhan dari *software* bukan implementasi perangkat lunak.



Gambar 2.12 Pengujian Kotak Hitam (*black box*)

Berdasarkan prosedur untuk melakukan pengujian kotak hitam, perlu dilakukan pembentukan *test case* untuk mengerjakan seluruh keperluan fungsional program.

Tujuan dari metode ini mencari kesalahan pada :

- Fungsi yang tidak benar atau tidak ada (hilang).
- Kesalahan antar-muka (*interface errors*).
- Kesalahan pada struktur data dan akses *database*.
- Kesalahan performansi (*performance errors*).
- Kesalahan inisialisasi dan terminasi.

Pengujian kotak hitam didesain untuk menjawab pertanyaan-pertanyaan mengenai bagaimana sistem tersebut bekerja. Pada umumnya pertanyaan-pertanyaan yang biasanya muncul adalah

- Bagaimana fungsi-fungsi diuji agar dapat dinyatakan *valid* ?
- *Input* seperti apa yang dapat menjadi bahan *test case* yang baik ?
- Apakah sistem sensitif terhadap *input-input* tertentu ?
- Bagaimana sekumpulan data dapat diisolasi ?
- Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?

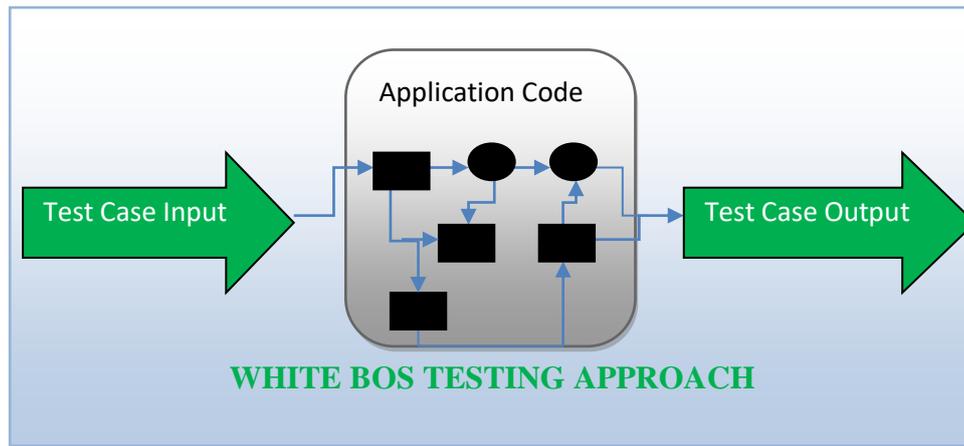
Beberapa hal yang harus diperhatikan dalam penentuan kebutuhan yang diperlukan untuk pengujian secara sistematis, yaitu :

- a. Dalam pembuatan *test case* yang efektif, harus dilakukan pengelompokan dari *test case* yang bersifat umum pada suatu sistem menjadi *test case* yang lebih khusus, sehingga tercapai *test case* individual.
- b. Dalam penentuan desain *test case*, juga digunakan mekanisme untuk memastikan bahwa *test case* yang ada telah cukup mencakup semua aspek dari sistem.
- c. Dalam pendesainan *test case* secara manual, tidak ada alat bantu otomatisasi guna menentukan *test case* yang dibutuhkan oleh sistem. Hal ini disebabkan setiap sistem memiliki perbedaan, sedangkan alat bantu tes tidak dapat mengetahui aturan benar dan salah dari suatu operasi.
- d. Pendesainan *test case* membutuhkan pengalaman, penalaran dan intuisi dari seorang *tester*.

Untuk melakukan pengujian dibutuhkan spesifikasi yang digunakan sebagai tuntunan dalam melakukan pengujian. Spesifikasi atau model sistem adalah titik awal dalam memulai desain tes, dimana spesifikasi tersebut dapat berupa 13 spesifikasi fungsional, spesifikasi kinerja, spesifikasi keamanan, spesifikasi skenario pengguna, atau spesifikasi berdasarkan pada resiko sistem. Spesifikasi tersebut menggambarkan kriteria yang digunakan untuk menentukan operasi yang benar atau dapat diterima sebagai acuan pelaksanaan tes.

2.6.6 White Box Testing

White Box Testing merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-*compile* ulang.



Gambar 2.13 *White Box Testing*

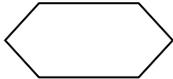
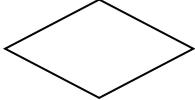
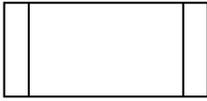
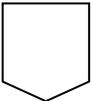
2.7 Flowchart

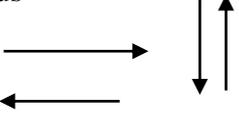
Menurut Adelia (2011:116), “Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program”. Flowchart menolong analyst dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Menurut Sulindawati (2010:8), “Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-prosedur dari suatu program”. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengopersian.

Berdasarkan beberapa pendapat yang dikemukakan di atas dapat ditarik kesimpulan flowchart atau diagram alur adalah suatu alat yang banyak digunakan untuk membuat algoritma, yakni bagaimana rangkaian pelaksanaan suatu kegiatan. Suatu diagram alur memberikan gambaran dua dimensi berupa simbol-simbol grafis. Masing-masing simbol telah ditetapkan terlebih dahulu fungsi dan artinya.

Tabel 2.1 Simbol-Simbol *Flowchart*

Simbol	Keterangan
Simbol Terminal 	Digunakan untuk menunjukkan awal dan akhir dari program.
Simbol Persiapan 	Digunakan untuk memberikan nilai awal pada suatu variable atau <i>counter</i> .
Simbol Proses 	Digunakan untuk mengolah aritmatikadan pemindahan data.
Simbol Keputusan 	Digunakan untuk mewakili operasi perbandingan logika.
Simbol Proses 	Digunakan untuk proses yang detailnya dijelaskan terpisah, misalnya dalam bentuk <i>subroutine</i> .
<i>Connector</i> 	Digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.
Simbol Penghubung 	Digunakan untuk menunjukkan hubungan arus dari suatu proses yang terputus masih dalam halaman yang berbeda.

<p>Arus</p> 	<p>Penghubung antara prosedur / proses</p>
<p><i>Document</i></p> 	<p>Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output di cetak dikertas</p>
<p><i>Input-Output</i></p> 	<p>Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya</p>
<p><i>Disk Storage</i></p> 	<p>Simbol untuk menyatakan input berasal dari disk atau ouput disimpan ke disk.</p>