

BAB II

TINJAUAN PUSTAKA

2.1 Rumah Kost

Rumah kost adalah sejenis kamar sewa yang disewa (*booking*) selama kurun waktu tertentu sesuai dengan perjanjian pemilik kamar dan harga yang disepakati. Umumnya *booking* kamar dilakukan selama kurun waktu satu tahun. Namun demikian ada pula yang hanya menyewakan selama satu bulan, tiga bulan, dan enam bulan, sehingga sebutannya menjadi sewa tahunan, bulanan, tri bulanan, dan tengah tahunan (Irma, 2014).

2.2 E-KTP

Menurut UU No. 23 tahun 2006 tentang Administrasi Kependudukan, Definisi dari E-KTP atau kartu tanda penduduk elektronik adalah dokumen kependudukan yang memuat system keamanan / pengendalian baik dari sisi administrasi ataupun teknologi informasi dengan berbasis pada database kependudukan nasional. Penduduk hanya di perbolehkan memiliki 1 (satu) KTP yang tercantum Nomor induk Kependudukan (NIK). NIK merupakan identitas tunggal setiap penduduk dan berlaku seumur hidup. Nomor NIK yang ada di E-KTP nantinya akan dijadikan dasar dalam penerbitan paspor, Surat Izin Mengemudi (SIM), Nomor Pokok Wajib Pajak (NPWP), Polis Asuransi, Sertifikat atas Hak Tanah dan penerbitan dokumen identitas lainnya.

Pada dasarnya, teknologi yang digunakan pada E-KTP bukanlah inovasi baru. Industri perbankan dan telekomunikasi sudah memanfaatkan kartu yang dilengkapi chip sejak beberapa tahun lalu. Dalam chip tersebut tersimpan data-data pemilik kartu yang hanya bisa ditulis dan dibaca dengan menggunakan perangkat tertentu. Kemampuan chip untuk menyimpan data secara elektronik itulah yang membuatnya diberi nama *smart card* (kartu pintar).

Sebuah chip *smart card* terdiri atas dua bagian, yaitu memori dan komponen mikroprosesor . Memori yang tertanam dalam chip tersebut terbagi dalam tiga jenis, *Random Access Memory* (RAM) dengan kapasitas hingga 8 *kilobyte* dan *Read-Only*

Memory (ROM) dengan kapasitas hingga 346 *kilobyte* serta programmable ROM berkapasitas 256 kilobyte. Sedangkan mikroprosesornya memiliki kemampuan untuk menangani instruksi sepanjang 16 bit.

Chip yang dibenamkan pada E-KTP merupakan varian dari jenis kartu ISO/IEC14443. Chip tersebut diimport dalam bentuk modul seperti film negatif yang biasa digunakan pada kamera konvensional. Dalam satu baris modul terdapat tiga chip yang masing-masing sudah dilengkapi dengan RAM berkapasitas 4 kilobyte. Masing-masing chip berdimensi 1,5 x 1,5 cm tersebut nantinya dibenamkan pada kartu plastik polyvinyl chloride (PVC) berstandar ISO 7810 yang berdimensi 8,5 x 5,3 cm menggunakan mesin *pick and placer* termasuk dalam golongan *smart card* sama dengan kartu dengan standar ISO/IEC 14443.

Kepala Program Penelitian dan Perakayasa E-KTP Gembong S Wibowanto, mengungkapkan, salah satu teknologi inti dalam E-KTP adalah chip. Chip E-KTP adalah kartu pintar berbasis mikroprosesor dengan memori 8 KB. Chip dalam E-KTP berfungsi menyimpan data biodata pemilik, tanda tangan, pas foto, dan dua data sidik jari. *Default* sidik jari yang dipakai adalah telunjuk tangan kanan serta telunjuk tangan kiri. Chip E-KTP tidak tampak dari luar. Chip ini sudah memenuhi standar ISO 14443 A dan 14443 B, mendukung kerahasiaan data pemilik E-KTP. Chip juga hanya bisa dibaca oleh perangkat pembaca tertentu untuk menjamin keamanan data.

2.3 SMS Gateway

SMS (*Short Message Service*) merupakan suatu teknologi yang memungkinkan untuk mengirim dan menerima pesan antar pengguna *mobile phone*. SMS gateway merupakan sistem aplikasi untuk mengirim dan/atau menerima SMS, terutama digunakan dalam aplikasi bisnis, baik untuk kepentingan promosi, servis kepada customer, pengadaan content produk atau jasa, dan seterusnya. Karena merupakan sebuah aplikasi, maka fitur-fitur yang terdapat di dalam SMS gateway dapat dimodifikasi sesuai dengan kebutuhan (Isnawati Mulyani, Eri Satria, Asep Deddy Supriatna, 2012:2).

Beberapa fitur yang umum dikembangkan dalam aplikasi SMS *gateway* adalah:

1. *Auto-reply*

SMS *gateway* secara otomatis akan membalas SMS yang masuk. Contohnya untuk keperluan permintaan informasi tertentu (misalnya kurs mata uang atau jadwal perjalanan), di mana pengirim mengirimkan SMS dengan format tertentu yang dikenali aplikasi, kemudian aplikasi dapat melakukan *auto-reply* dengan membalas SMS tersebut, berisi informasi yang dibutuhkan.

2. Pengiriman massal

Disebut juga dengan istilah SMS *broadcast*, bertujuan untuk mengirimkan SMS ke banyak tujuan sekaligus. Misalnya, untuk informasi produk terbaru kepada pelanggan.

3. Pengiriman terjadwal

Sebuah SMS dapat diatur untuk dikirimkan ke tujuan secara otomatis pada waktu tertentu. Contohnya untuk keperluan mengucapkan selamat ulang tahun.

Untuk membuat sebuah SMS *gateway*, perlu mengenal hal-hal yang berhubungan dengan SMS *gateway* itu sendiri. Salah satu hal yang memegang peranan penting dalam pengiriman SMS adalah SMSC (*Short Message Service Center*), yang merupakan jaringan telepon selular yang menangani pengiriman SMS. Jadi, pada saat seseorang mengirimkan sebuah pesan SMS melalui ponselnya, SMSC-lah yang bertugas mengirimkan pesan tersebut ke nomor tujuan. Jika nomor tujuan tidak aktif, maka SMSC akan menyimpan pesan tersebut dalam jangka waktu tertentu. Jika SMS tetap tidak dapat terkirim sampai jangka waktu tersebut berakhir, maka SMS tersebut akan dihapus dari penyimpanan SMSC.

Di balik tampilan menu *Messages* pada sebuah ponsel sebenarnya terdapat AT Command-AT Command yang bertugas mengirim atau menerima data ke dan dari SMS *Centre*. AT Command tiap-tiap SMS device bisa berbeda-beda, tetapi pada dasarnya sama. Perintah-perintah AT Command biasanya disediakan oleh vendor alat komunikasi yang kita beli.

Beberapa AT Command yang penting untuk SMS:

1. AT+CMGS : Untuk mengirim SMS
2. AT+CMGR : Untuk memeriksa SMS
3. AT+CMGD : Untuk menghapus SMS
4. AT+CMGF : Untuk menentukan format teks

AT Command untuk SMS biasanya diikuti oleh data I/O yang diwakili oleh unit-unit PDU. Data yang mengalir ke/dari *SMSCenter* harus berbentuk PDU (*Protocol Data Unit*). PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU terdiri atas beberapa *header*, *header* untuk mengirim SMS ke *SMS-Center* berbeda dengan SMS yang diterima dari *SMSCenter*. Saat kita menerima SMS/MMS dari *handphone* (*mobile originated*) pesan tersebut tidak langsung dikirimkan ke *handphone* tujuan (*mobile terminated*), akan tetapi dikirim terlebih dahulu ke SMSC(*SMS Center*) yang biasanya berada di kantor operator telepon dan kemudian pesan tersebut diteruskan ke *handphone* tujuan.

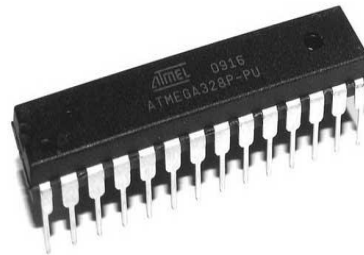
2.4 Mikrokontroler ATmega328

Mikrokontroler disebut juga MCU (Micro Chip Unit) atau μC adalah salah satu komponen elektronik atau IC yang memiliki beberapa sifat dan komponen seperti komputer, yaitu: CPU (Central Processing Unit) atau unit pemrosesan terpusat, memori kode, memori data, dan I/O (port untuk input dan output). Mikrokontroler merupakan single chip computers yang dapat digunakan untuk mengontrol sistem, disamping itu bentuknya yang kecil dan harganya yang murah sehingga dapat dicangkokkan (*embedded*) di dalam berbagai peralatan rumah tangga, kantor, industri atau robot (Listiono, 2016).

ATmega328 merupakan mikrokontroler 8-bit berbasis AVR-RISC buatan Atmel. Chip ini memiliki 32 KB memori ISP *flash* dengan kemampuan baca-tulis (*read write*), 1 KB EEPROM, dan 2 KB SRAM. Dari kapasitas memori *flash* nya yang sebesar 32 KB itulah chip ini diberi nama ATmega328. Chip lain yang memiliki memori 8 KB diberi nama ATmega8, dan ATmega16 untuk yang memiliki memori 16 KB.

Chip ATmega328 memiliki banyak fasilitas dan kemewahan untuk sebuah chip mikrokontroler. Chip tersebut memiliki 23 jalur *general purpose I/O (input/output)*, 32 buah register, 3 buah *timer/counter* dengan mode perbandingan, *interrupt internal* dan *external*, *serial programmable USART*, *2-wire interface serial*, serial port SPI, 6 buah channel 10-bit *A/D converter*, *programmable watchdog timer* dengan *oscilator internal*, dan lima *power saving mode*. Chip bekerja pada tegangan antara 1.8V ~ 5.5V. *Output* komputasi bisa mencapai 1 MIPS per Mhz. *Maximum operating frequency* adalah 20 Mhz.

ATmega328 menjadi cukup populer setelah chip ini dipergunakan dalam *board* Arduino. Dengan adanya Arduino yang didukung oleh *software* Arduino IDE, pemrograman chip ATmega328 menjadi jauh lebih sederhana dan mudah. Bentuk fisik mikrokontroler ATmega328 dapat dilihat pada gambar 2.1.



Gambar 2.1 Bentuk Fisik Mikrokontroler ATmega328

(Sumber: <http://ecadio.com/>, diakses tanggal 20 Januari 2017)

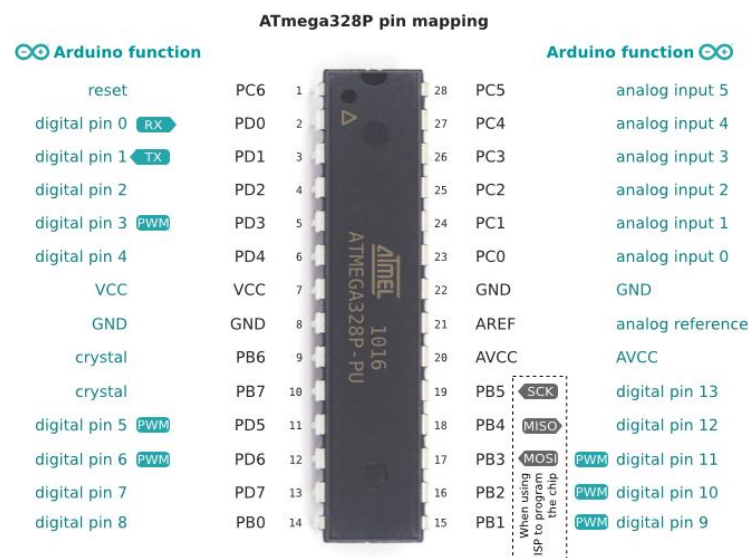
2.4.1 Fitur ATmega328

Mikrokontroler ATmega328 memiliki fitur sebagai berikut:

1. Saluran Input/Output (I/O) sebanyak 23 buah.
2. ADC *internal* sebanyak 6 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan
4. CPU terdiri atas 32 buah *register*.
5. SRAM sebesar 2 kByte.
6. Memori *Flash* sebesar 32 kByte dengan kemampuan *Read While Write*.
7. EEPROM sebesar 1 kByte yang dapat diprogram saat operasi.

8. Antarmuka komparator analog.
9. Port USART untuk komunikasi serial.
10. Port antarmuka SPI.
11. Sistem mikroprosesor 8-bit berbasis RISC dengan kecepatan maksimal 20 MHz.
12. Lima *mode Sleep* : *Idle*, *ADC Noise Reduction*, *Power-save*, *Power-down*, dan *Standby*.
13. Sumber Interupsi *External* dan *Internal*.
14. Enam buah *channels* PWM.

2.4.2 Konfigurasi Pin ATmega328



Gambar 2.2 Konfigurasi Pin ATmega328

(Sumber: <http://www.chicoree.fr/>, diakses tanggal 20 Januari 2017)

Berikut ini merupakan fungsi dari masing-masing pin pada mikrokontroler Atmega328:

1. VCC

VCC terletak pada pin 7, berfungsi untuk *supply* tegangan digital yang akan dihubungkan dengan tegangan 5V.

2. GND

GND terletak pada pin 8, berfungsi sebagai *ground* yang akan dihubungkan dengan *ground*.

3. Port B

Port B merupakan jalur data 8 bit dan memiliki 8 pin dari pin B0-B7 yang dapat difungsikan sebagai *input/output*, yaitu:

- a. PB0 berfungsi sebagai *Timer Counter 1 input capture* pin.
- b. PB1-PB3 dapat difungsikan sebagai *output PWM (Pulse Width Modulation)*.
- c. MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI.
- d. TOSC1 (PB6) dan TOSC2 (PB7) berfungsi sebagai sumber *clock external* untuk *timer*.

4. Port C

Port C merupakan jalur data 7 bit masing-masing pinter dapat *pull-up* resistor. Pin C0-C5 sebagai ADC yang berfungsi mengubah *input* analog menjadi digital. Pin C6/*Reset*, jika RSTDISBL *fuse* diprogram, maka PC6 akan berfungsi sebagai pin I/O. Namun jika RSTDISBL *fuse* tidak diprogram, maka pin ini akan berfungsi sebagai *input reset*. Namun jika tegangan yang diterima pin C6 rendah yaitu lebih rendah dari pulsa minimum, maka akan menghasilkan suatu kondisi *reset* meskipun *clock* tidak bekerja.

5. Port D

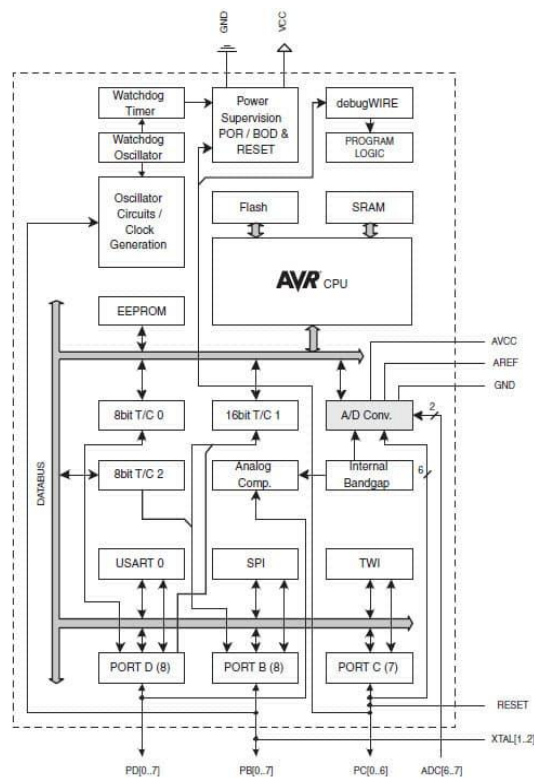
Port D merupakan jalur data 8 bit yang berfungsi sebagai I/O dengan *internal pull-up* resistor. Port D memiliki beberapa pin yaitu:

- a. PD0-PD1 (TXD dan RXD) merupakan jalur data komunikasi serial. Pin TXD berfungsi untuk mengirimkan data serial, sedangkan RXD berfungsi untuk menerima data serial.
- b. PD2-PD3 (INT0 dan INT1) berfungsi sebagai interupsi yaitu jeda dari program, pada saat program berjalan kemudian terjadi interupsi

hardware/software maka program utama akan berhenti dan akan menjalankan program interupsi.

- c. PD4 (T0) dan PD5 (T1) berfungsi sebagai masukan *counter external* untuk *timer 0* dan *timer 1*.
 - d. PD6-PD7 (AIN0 dan AIN1) keduanya merupakan masukan *input* untuk *analog comparator*.
6. AVCC
- AVCC berfungsi sebagai *supply* tegangan untuk ADC. Pin ini harus dihubungkan secara terpisah dengan VCC Karena digunakan untuk analog. Cara menghubungkan AVCC adalah melewati *low-passfilter* setelah itu dihubungkan dengan VCC.
7. AREF
- AREF merupakan pin referensi analog jika menggunakan ADC.

2.4.3 Blok Diagram ATmega328



Gambar 2.3 Diagram Blok Atmega328

(Sumber: <http://www.mouser.co.id/>, diakses tanggal 21 Januari 2017)

Penjelasan diagram blok pada gambar 2.3 adalah sebagai berikut:

1. *Universal Asynchronous Receiver/Transmitter* (UART) adalah antar muka yang digunakan untuk komunikasi serial seperti pada RS-232, RS-422 dan RS485.
2. 2KB RAM pada memori kerja bersifat *volatile* (hilang saat daya dimatikan), digunakan oleh variabel-variabel di dalam program.
3. 32KB RAM *flash memory* bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, flash memory juga menyimpan *bootloader*. *Bootloader* adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah *bootloader* selesai dijalankan, berikutnya program di dalam RAM akan dieksekusi.
4. 1KB EEPROM bersifat *non-volatile*, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan. Tidak digunakan pada papan Arduino.
5. *Central Processing Unit* (CPU), bagian dari mikrokontroler untuk menjalankan setiap instruksi dari program.
6. Port *input/output*, pin-pin untuk menerima data (*input*) digital atau analog, dan mengeluarkan data (*output*) digital atau analog.

2.4.4 Peta Memori ATmega328

Mikrokontroler ATmega328 memiliki 3 jenis memori, yaitu memori EEPROM, memori program, dan memori data.

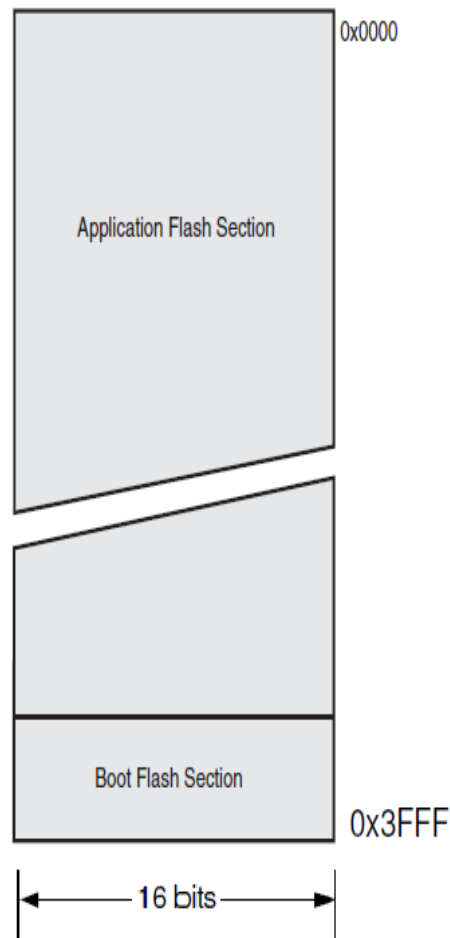
1. Memori EEPROM ATmega328

Pada memori EEPROM, data dapat ditulis/dibaca kembali dan ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM dimulai dari 0x000 hingga 0x3FF.

2. Memori Program ATmega328

ATMega328 memiliki 32K byte *On-chip In-System Reprogrammable Flash Memory* untuk menyimpan program. Memori *flash* dibagi kedalam dua

bagian, yaitu bagian program *bootloader* dan aplikasi seperti terlihat pada Gambar 2.4. *Bootloader* adalah program kecil yang bekerja pada saat sistem dimulai yang dapat memasukkan seluruh program aplikasi ke dalam memori prosesor.



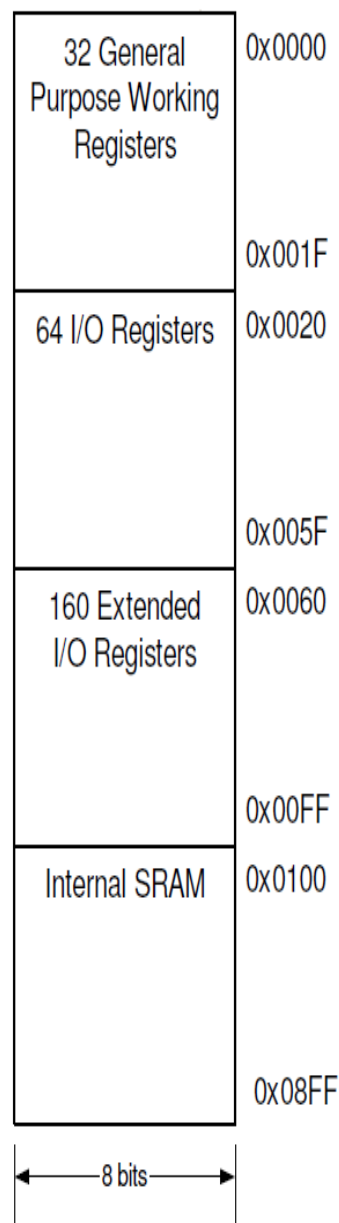
Gambar 2.4 Peta Memori Program ATmega328

(Sumber: <http://www.atmel.com/>, diakses tanggal 21 Januari 2017)

3. Memori Data ATmega328

Memori data ATmega328 terbagi menjadi 4 bagian, yaitu 32 lokasi untuk register umum, 64 lokasi untuk register I/O, 160 lokasi untuk register I/O tambahan dan sisanya 2048 lokasi untuk data SRAM internal. Register umum menempati alamat data terbawah, yaitu 0x0000 sampai 0x001F.

Register I/O menempati 64 alamat berikutnya mulai dari 0x0020 hingga 0x005F. Register I/O tambahan menempati 160 alamat berikutnya mulai dari 0x0060 hingga 0x00FF. Sisa alamat berikutnya mulai dari 0x0100 hingga 0x08FF digunakan untuk SRAM internal. Peta memori data dari ATmega 328 dapat dilihat pada Gambar 2.5.



Gambar 2.5 Gambar Peta Memori Data ATmega328

(Sumber: <http://www.atmel.com/>, diakses tanggal 21 Januari 2017)

2.5 *Radio Frequency Identification (RFID)*

Sensor *Radio Frequency Identification (RFID)* adalah teknologi yang mampu mengidentifikasi berbagai objek menggunakan gelombang radio (Akintola Kolawole dan Olutayo Kehinde, 2011:37). Sistem RFID terdiri dari 4 komponen yaitu RFID *tag (transponder)*, antena, *reader*, dan *interface software* (Miguel, et al., 2011:339).

1. RFID *tag (transponder)* memiliki *chip* yang dapat menyimpan data berupa nomor ID unik dan memiliki antena yang berfungsi untuk mentransmisikan data ke RFID *reader* melalui gelombang radio yang dipancarkan RFID *reader*.
2. Antena terdapat pada RFID *tag (tag-antena)* dan RFID *reader (reader-antena)* atau (*interogator*) yang berfungsi mentransmisikan data dari *chip* RFID *tag* ke RFID *reader* melalui gelombang radio.
3. RFID *reader* adalah perangkat yang kompatibel dengan RFID *tag*. RFID *reader* akan memancarkan gelombang radio dan menginduksi RFID *tag*, kemudian RFID *tag* akan mengirim data ID dari antena yang terdapat pada rangkaian RFID *tag* melalui gelombang radio yang dipancarkan RFID *reader*.
4. *Interface Software* yang berfungsi untuk membaca data ID dari RFID *reader* dan mengolah data tersebut sehingga dapat digunakan menjadi *password*.

2.5.1 *RFID Tag*

RFID *tag* memiliki *chip* yang didalamnya dapat menyimpan data berupa nomor ID, *transponder* atau *tag-antena* yang berfungsi untuk mengirim data melalui gelombang radio yang dipancarkan RFID *reader* dan *encapsulation* atau bungkus yang berfungsi untuk melindungi *chip* agar tidak mudah rusak (Ho Tien Dang, 2013:16). Berdasarkan itu, *tag RFID* dapat digolongkan menjadi dua :

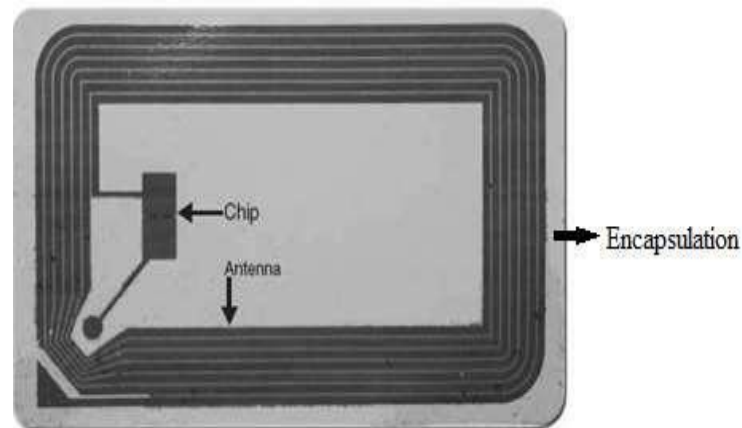
1. Tag Aktif yaitu tag yang catu dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan tag dapat

mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe tag ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh tag RFID maka rangkaianannya akan semakin kompleks dan ukurannya akan semakin besar.

2. Tag Pasif yaitu tag yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaianannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah tag hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca RFID harus menyediakan daya tambahan untuk tag RFID. Tag RFID telah sering dipertimbangkan untuk digunakan sebagai barcode pada masa yang akan datang. Pembacaan informasi pada tag RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang modern, maka tag RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan barcode (Bakhtiar, B. dan Susanti, R. Elektron, Vol.1: 63-64).

RFID *tag* terdiri dari 3 bagian yaitu:

1. Mikroprosesor
Mikroprosesor adalah *chip* yang terletak dalam sebuah RFID *tag* yang berfungsi sebagai penyimpan data.
2. *Metal Coil*
Metal Coil terbuat dari kawat alumunium yang berfungsi sebagai antena yang dapat beroperasi pada frekuensi 13,56 MHz. Apabila sebuah RFID *tag* masuk kedalam jangkauan *reader* maka antena akan mengirimkan data yang ada pada *tag* kepada *reader* terdekat.
3. *Encapsulating*
Encapsulating adalah bahan yang berfungsi untuk melindungi RFID *tag* dan antena terbuat dari bahan plastik atau kaca.



Gambar 2.6 RFID Tag

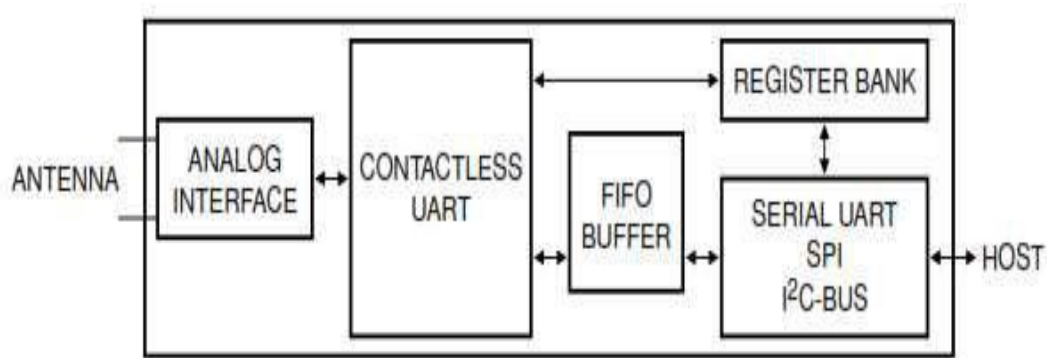
2.5.2 RFID Reader

Menurut Riza Muharrir, (2014:8) sistem RFID akan berfungsi dengan baik diperlukan RFID *reader* yang dapat membaca RFID *tag* dan mengirim data yang dibaca ke database. Sebuah *reader* menggunakan antenna untuk berkomunikasi dengan RFID *tag*. Ketika *reader* memancarkan gelombang radio seluruh RFID *tag* yang memiliki frekuensi sama dengan *reader* akan memberikan respon.

RFID *reader* memancarkan gelombang radio dan menginduksi RFID *tag*. Gelombang induksi tersebut berisi data ID dan jika dikenali oleh RFID *tag*, memori RFID *tag* (ID chip) akan terbuka. Gelombang radio yang dipancarkan oleh *reader* juga berfungsi sebagai catu daya RFID *tag* (*tag* pasif). Kemudian RFID *tag* akan mengirimkan kode yang terdapat di memori ID chip melalui antena yang terpasang di RFID *tag*. RFID *reader* akan mengirim data tersebut ke mikrokontroler untuk diproses menjadi *password*.

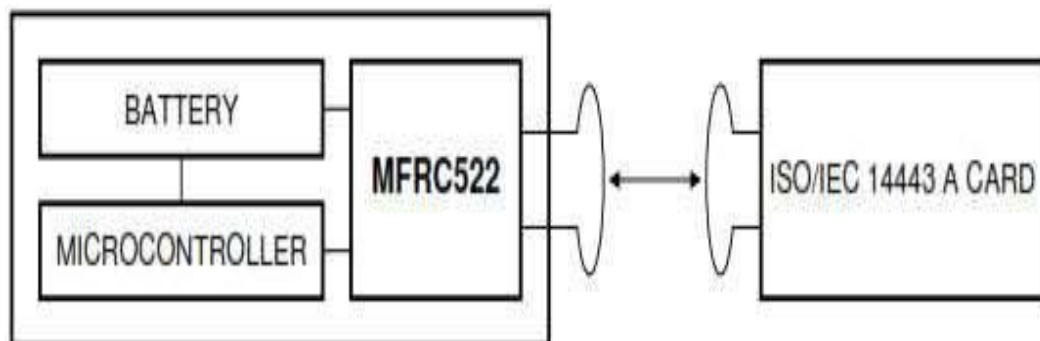
RFID *reader* memiliki antena yang berfungsi untuk memancarkan gelombang radio ke RFID *tag* dan menerima data yang dikirim oleh RFID *tag*, data tersebut berupa sinyal analog yang kemudian akan diteruskan ke *contactless* UART yang berfungsi untuk membaca data ID dari RFID *tag* kemudian data ID tersebut akan dikirim ke register bank dan FIFO buffer. Register bank mengirim data ID ke serial UART kemudian akan mengirim data ID tersebut kepada HOST (mikrokontroler). FIFO buffer berfungsi mengirim data dari *contactless* UART

kepada HOST (mikrokontroler) dan dari mikrokontroler ke *contactless* UART, data yang dikirim berupa data serial (*Datasheet RFID Reader MFRC522*). Blok diagram cara kerja RFID reader sebagai *receiver* dan transfer data dapat dilihat pada gambar 2.7.



Gambar 2.7 Cara Kerja RFID reader sebagai *receiver* dan transfer data
(Sumber: <https://www.nxp.com/>, diakses tanggal 21 Januari 2017)

RFID reader yang digunakan pada rangkaian adalah RFID reader dengan frekuensi 13.56 MHz. RFID reader 13.56 MHz dapat digunakan untuk membaca RFID tag jenis *high frequency* (HF) yang digunakan sebagai *smart card*. RFID reader 13.56 MHz dapat membaca *smart card* jenis MIFARE, ISO/IEC 14443. Data ID yang berupa nomor unik dari *smart card* akan dibaca oleh RFID reader kemudian dikirim ke mikrokontroler, *battery* digunakan untuk supply tegangan RFID dan mikrokontroler. Diagram blok sistem kerja dari mikrokontroler, RFID reader dan RFID tag (*smart card*) dapat dilihat pada gambar 2.8.



Gambar 2.8 RFID Reader Membaca Data ID Dari *Smart Card*
(Sumber: <https://www.nxp.com/>, diakses tanggal 21 Januari 2017)

2.5.3 RFID Reader RC522

RFID reader RC522 merupakan reader RFID yang mampu melakukan proses *read write* dan bekerja pada frekuensi 13.56 MHz. Tag RFID yang kompatibel dengan modul RFID ini adalah tag jenis pasif. RFID reader RC522 memiliki rintang baca kurang lebih 3 kaki. Gambar 2.9 merupakan gambar dari RFID reader RC522.

Mifare RC522 RFID Reader Module adalah sebuah modul berbasis IC Philips MFRC522 yang dapat membaca RFID dengan penggunaan yang mudah dan harga yang murah, karena modul ini sudah berisi komponen-komponen yang diperlukan oleh MFRC522 untuk dapat bekerja.

Modul ini dapat digunakan langsung oleh MCU dengan menggunakan interface SPI. MFRC522 merupakan produk NXP yang menggunakan frekuensi 13.56 MHz. MFRC522 support dengan semua varian *MIFARE Mini*, *MIFARE 1K*, *MIFARE 4K*, *MIFARE Ultralight*, *MIFARE DESFire*, *MIFARE Plus*, *MIFARE SAM AV2*, dan *MIFARE Classic*.



Gambar 2.9 RFID reader RC522

(Sumber: <http://www.elabpeers.com/>, diakses tanggal 21 Januari 2017)

Modul RFID reader ini berfungsi untuk membaca data atau nomor ID pada E-KTP karena modul RFID reader RC522 memiliki keunggulan. Adapun keunggulannya sebagai berikut:

1. Setiap Tag memiliki ID yang unik dan berbeda secara *world-wide*, tanpa tergantung *manufacturenya*. Hal ini terjadi karena adanya konsensus penomoran ID antara *manufacturers* sedunia. Dengan mendata Tags yang dipakai pada database aplikasi, maka dapat dengan mudah dan efektif

ditingkatkan aspek pengamanan dalam pembacaan (*secured/selective reading*).

2. *Operation in Hars Environment*, dimana dengan pemilihan material maupun bentuk *encapsulation* yang sesuai dengan kondisi operasional di lapangan, maka pemakaian dan pemasangan RFID Tag dimungkinkan untuk kondisi *ekstrem/hars environment*, misal: temperatur atau tekanan yang sangat tinggi.
3. Fleksibel dalam pemasangan, dengan *encapsulation* yang sesuai, RFID Tag dapat dipasang secara fleksibel dan bervariasi pada item, misalnya: Tag yang ditenamkan pada ban traktor, Tag yang di *clamped* pada *body* mobil.
4. *Reusable*, dimana RFID Tag memiliki life time yang relatif lama dan dapat dipindah-pindahkan atau dipakai kembali untuk item yang lain. Pemakaian kembali Tag tersebut dapat meningkatkan efisiensi biaya.
5. *Accuracy*, dikarenakan pembacaan dilakukan secara *device reading* dan bukan oleh indera manusia, maka tingkat akurasi menjadi sangat tinggi.
6. Dapat dilakukan secara *un-attended/automated*, dimana RFID cukup banyak diaplikasikan dengan pembacaan yang otomatis tanpa intervensi manusia, misal: aplikasi konveyor. Hal ini dimungkinkan karena RFID reader dapat langsung mendeteksi keberadaan Tag dalam area bacanya.
7. Tidak memerlukan *line of sight*, prinsip kerja RFID yang bersifat elektromagnetik dan bukan optikal memungkinkan RFID reader untuk membaca Tag walaupun Tag tersebut tidak terlihat atau tersembunyi. Misal: RFID reader dapat membaca semua Tag pada item yang berada dalam suatu peti tertutup.
8. Pembacaan yang cepat, kecepatan pembacaan juga relatif tinggi karena kemampuan membaca sekaligus informasi dari semua Tags yang berada dalam area bacanya (sebagai *simultaneous multi Tags reading*).
9. Aman, Tags bisa diberikan *password* sehingga meningkatkan faktor keamanan dimana data yang berada di Tag tidak bisa dibaca oleh setiap reader jika tidak sesuai passwordnya. Tag juga bisa dimatikan dengan *feature Killing Tag*.

Tabel 2.1. Spesifikasi Modul RFID-RC522

No	Parameter	Deskripsi
1	Supports Card	ISO/IEC14443A/MIFARE
2	Frequency	13.56 MHz
3	VDDA	3.3-5V
4	IDDA	10Ma

2.6 Arduino Uno

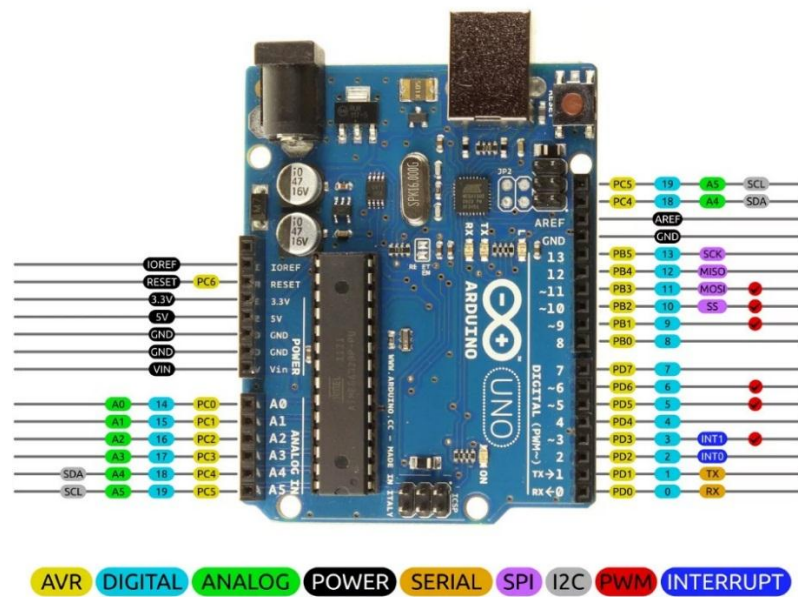
Arduino Uno adalah sebuah *board* mikrokontroler yang berbasis ATmega328. Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai *output* PWM, 6 pin sebagai input analog, 16 MHz osilator kristal, koneksi USB, jack power, kepala ICSP, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup menghubungkan *Board* Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik AC ke adaptor DC atau baterai untuk menjalankannya (Afrizal Fitriandi, Endah Komalasari, Herri Gusmed Volume 10, 2016).

Arduino Uno berbeda dari semua *board* Arduino sebelumnya, Arduino Uno tidak menggunakan chip driver FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke serial. Revisi 2 dari *board* Arduino Uno mempunyai sebuah resistor yang menarik garis 8U2 HWB ke ground, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU mode. Revisi 3 dari *board* Arduino UNO memiliki fitur-fitur baru sebagai berikut:

1. Pinout 1.0: ditambah pin SDA dan SCL yang dekat dengan pin AREF dan dua pin baru lainnya yang diletakkan dekat dengan pin RESET, IOREF yang memungkinkan shield-shield untuk menyesuaikan tegangan yang disediakan dari board. Untuk ke depannya, shield akan dijadikan kompatibel/cocok dengan board yang menggunakan AVR yang beroperasi dengan tegangan 5V dan dengan Arduino Due yang beroperasi dengan tegangan 3.3V.

2. Sirkuit RESET yang lebih kuat.
3. Atmega 16U2 menggantikan 8U2.

“Uno” berarti satu dalam bahasa Italia dan dinamai untuk menandakan keluaran (produk) Arduino 1.0 selanjutnya. Arduino UNO dan versi 1.0 akan menjadi referensi untuk versi-versi Arduino selanjutnya. Arduino UNO adalah sebuah seri terakhir dari *board* Arduino USB dan model referensi untuk papan Arduino.



Gambar 2.10 Fungsi dan Konfigurasi Arduino Uno

Berikut adalah bagian-bagian dari Arduino Uno dan fungsinya:

1. 14 pin input/output digital (0-13) berfungsi sebagai input atau output, dapat diatur oleh program.
2. Khusus untuk pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog output dimana tegangan output-nya dapat diatur. Nilai sebuah pin output analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.
3. USB, berfungsi untuk memuat program dari komputer ke dalam papan, komunikasi serial antara papan dan komputer, dan memberi daya listrik kepada papan

4. Sambungan atau jumper untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.
5. Kristal (*quartz crystal oscillator*) jika mikrokontroler dianggap sebagai sebuah otak, maka kristal adalah jantung-nya karena komponen ini menghasilkan detak-detak yang dikirim kepada mikrokontroler agar melakukan sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).
6. Tombol Reset untuk mereset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan mikrokontroler.
7. In-Circuit Serial Programming (ICSP) Port ICSP memungkinkan pengguna untuk memprogram mikrokontroler secara langsung, tanpa melalui bootloader. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.
8. Mikrokontroler Atmega328 merupakan komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.
9. Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan DC antara 7-12V.
10. 6 pin input analog (0-5), pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V

2.6.1 Spesifikasi Arduino Uno

1. Mikrokontroler ATmega328
2. Tegangan Operasi sebesar 5 V
3. Tegangan *input* sebesar 6 – 20 V tetapi direkomendasikan untuk ATmega 328 sebesar 7 – 12 V.

4. Pin digital I/O sebanyak 14 pin dimana 6 pin merupakan keluaran dari PWM.
5. Pin input analog sebanyak 6 pin.
6. Arus DC pin I/O sebesar 40 mA sedangkan Arus DC untuk pin 3.3V sebesar 50 mA.
7. *Flash memory* 328 Kb yang mana 0,5 Kb digunakan oleh *bootloader*.
8. SRAM 2 Kb.
9. EEPROM 1 Kb.
10. *Clock speed* sebesar 16MHz.

2.6.2 Komunikasi Arduino

Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran *board* ini komunikasi serial melalui USB dan muncul sebagai *com port virtual* untuk perangkat lunak pada komputer. *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Inf diperlukan.

Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi interface pada sistem.

2.6.3 Programming

Arduino UNO dapat diprogram dengan software Arduino Pilih “Arduino Uno dari menu Tools > *Board* (termasuk mikrokontroler pada *board*). ATmega328 pada Arduino Uno hadir dengan sebuah *bootloader* yang memungkinkan kita untuk mengupload kode baru ke ATmega328 tanpa menggunakan pemrogram hardware eksternal. ATmega328 berkomunikasi menggunakan protokol STK500 asli.

2.7 Bahasa Pemrograman C

Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories. Bahasa C mempunyai kemampuan lebih dibanding dengan bahasa pemrograman lain. Bahasa C merupakan bahasa pemrograman yang bersifat portable yaitu suatu pemrograman yang dibuat dengan bahasa C pada suatu komputer akan dapat dijalankan pada komputer lain dengan sedikit (atau tanpa) ada perubahan yang berarti.

Bahasa C merupakan bahasa yang biasa digunakan untuk keperluan pemrograman sistem, antara lain membuat:

1. *Assembler*
2. *Interpreter*
3. *Compiler*
4. Sistem Operasi
5. Program bantu (*utility*)
6. Editor
7. Paket program aplikasi

Dalam beberapa literatur, bahasa C digolongkan sebagai tingkat menengah (*medium level language*). Penggolongan ini bukan berarti bahasa C kurang ampuh atau lebih sulit dibandingkan dengan bahasa tingkat tinggi (*high level language* seperti *Pascal*, *Basic*, *Fortran*, *Java*, dan lain-lain). Namun untuk menegaskan bahwa bahasa C bukanlah bahasa yang berorientasi pada mesin, yang merupakan ciri dari bahasa tingkat rendah (*low level language*) yaitu bahasa mesin dan *assembly*. Pada kenyataannya, bahasa C mengkombinasikan elemen dalam bahasa tingkat tinggi dan bahasa tingkat rendah, yaitu kemudahan dalam membuat program yang ditawarkan pada bahasa tinggi dan kecepatan eksekusi dari bahasa tingkat rendah.

Program C pada hakekatnya tersusun atas sejumlah blok fungsi. Sebuah program minimal mengandung sebuah fungsi. Fungsi pertama yang harus ada dalam program C dan sudah ditentukan namanya adalah `main()`. Setiap fungsi terdiri atas satu atau beberapa pernyataan, yang secara keseluruhan dimaksudkan

untuk melaksanakan tugas khusus kurawal buka ({} dan diakhiri dengan tanda kurung kurawal tutup()). Di antara kurung kurawal itu dapat dituliskan statement-program C. Namun pada kenyataannya, suatu fungsi bisa saja tidak mengandung pernyataan sama sekali. Walaupun fungsi tidak memiliki pernyataan, kurung kurawal harus tetap ada. Sebab kurung kurawal mengisyaratkan awal dan akhir definisi fungsi (dimasandre, 2013).

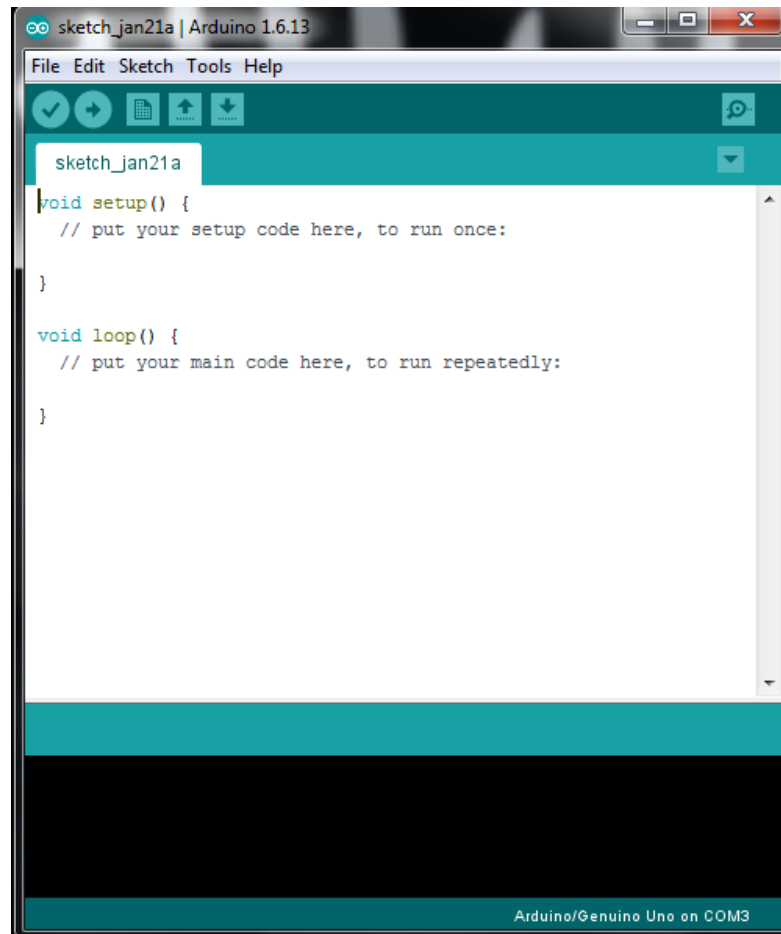
2.8 *Arduino Development Environment*

Arduino Development Environment terdiri dari editor teks untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah *toolbar* dengan tombol-tombol untuk fungsi yang umum dan beberapa menu. *Arduino Development Environment* terhubung ke arduino board untuk meng-*upload* program dan juga untuk berkomunikasi dengan arduino board.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library C/C++* yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

Perangkat lunak yang ditulis menggunakan *Arduino Development Environment* disebut *sketch*. *Sketch* ditulis pada editor teks. *Sketch* disimpan dengan file berekstensi *.ino*. area pesan memberikan informasi dan pesan error ketika kita menyimpan atau membuka *sketch*. Konsol menampilkan output teks dari *Arduino Development Environment* dan juga menampilkan pesan error ketika kita mengkompilasi *sketch*. Pada sudut kanan bawah jendela *Arduino Development Environment* menunjukkan jenis board dan port serial yang sedang digunakan. Tombol *toolbar* digunakan untuk mengecek dan meng-*upload sketch*, membuat, membuka, atau menyimpan *sketch*, dan menampilkan serial monitor.

Lingkungan *open-source Arduino* memudahkan untuk menulis kode dan meng-*upload* ke board *Arduino*. Ini berjalan pada *Windows*, *Mac OS X*, dan *Linux*. Berdasarkan Pengolahan, *avr-gcc*, dan perangkat lunak sumber terbuka lainnya. *Arduino Development Environment* dapat dilihat pada gambar 2.11.



Gambar 2.11 Tampilan Arduino IDE

Berikut ini merupakan tombol-tombol *toolbar* serta fungsinya :



Gambar 2.12 Tombol *Verify*

Tombol *Verify* berfungsi untuk mengecek error pada kode program.



Gambar 2.13 Tombol *Upload*

Tombol *Upload* berfungsi untuk meng-*compile* dan meng-*upload* program ke Arduino board.



Gambar 2.14 Tombol *New*

Tombol *New* berfungsi untuk membuat *sketch* baru.



Gambar 2.15 Tombol *Open*

Tombol *Open* berfungsi untuk menampilkan sebuah menu dari seluruh *sketch* yang berada di dalam *sketchbook*.



Gambar 2.16 Tombol *Save*

Tombol *Save* berfungsi untuk menyimpan *sketch*.

Setiap bagian dari data yang tersimpan dalam program Arduino memiliki tipe datanya masing-masing. Tergantung pada kebutuhannya, 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 1 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*.

2.9 Bahasa Pemrograman Arduino

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

Karena menggunakan bahasa pemrograman C sebagai dasarnya, bahasa pemrograman Arduino banyak sekali kemiripan, walaupun beberapa hal telah berubah.

2.9.1 Struktur

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

1. *void Setup() { }*

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

2. *void loop() { }*

Fungsi ini akan dijalankan setelah fungsi *void setup* selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya dilepaskan.

2.9.2 *Syntax*

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

1. // (komentar satu baris)

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

2. /* */ (komentar banyak baris)

Jika terdapat 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.

3. { } (kurung kurawal)

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

4. ; (titik koma)

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

2.9.3 *Variabel*

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

1. *int (integer)*

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

2. *long (long)*

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

3. *boolean (boolean)*

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

4. *float (float)*

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

5. *char (character)*

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

2.9.4 Operator Matematika

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana).

1. = Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya: $x = 10 * 2$, x sekarang sama dengan 20).
2. % Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya: $12 \% 10$, ini akan menghasilkan angka 2).
3. + Untuk penjumlahan.
4. – Untuk pengurangan.
5. * Untuk perkalian.
6. / Untuk pembagian.

2.9.5 Operator Pembandingan

Digunakan untuk membandingkan nilai logika.

1. == Sama dengan (misalnya: $12 == 10$ adalah *FALSE* (salah) atau $12 == 12$ adalah *TRUE* (benar)).
2. != Tidak sama dengan (misalnya: $12 != 10$ adalah *TRUE* (benar) atau $12 != 12$ adalah *FALSE* (salah)).

3. < Lebih kecil dari (misalnya: $12 < 10$ adalah *FALSE* (salah) atau $12 < 12$ adalah *FALSE* (salah) atau $12 < 14$ adalah *TRUE* (benar)).
4. > Lebih besar dari (misalnya: $12 > 10$ adalah *TRUE* (benar) atau $12 > 12$ adalah *FALSE* (salah) atau $12 > 14$ adalah *FALSE* (salah)).

2.9.6 Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan.

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 ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan.

2.9.7 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 dan output.

2. `digitalWrite(pin, value)`

Ketika sebuah pin ditetapkan sebagai output, pin tersebut dapat dijadikan *high* (ditarik menjadi 5v) atau *low* (diturunkan menjadi *ground*).

3. `digitalRead(pin)`

Ketika sebuah pin ditetapkan sebagai input maka dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *high* atau *low*.

2.9.8 Analog

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 atau mati dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. 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 dapat membaca keluaran voltasenya. Keluarannya berupa angka antara 0 (untuk 0v) dan 1024 (untuk 5v).

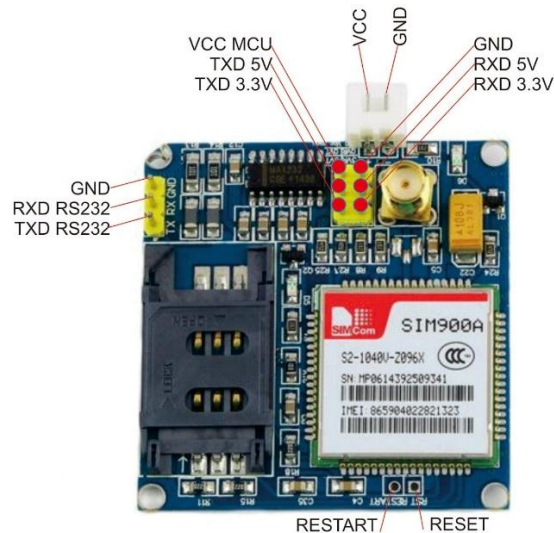
2.10 Modul GSM SIM900A

GSM (*Global System For Mobile Communication*) adalah sistem komunikasi seluler generasi kedua yang menjadi standar global komunikasi nirkabel. Teknologi GSM merupakan standar komunikasi yang lebih banyak diterapkan pada telepon genggam yang digunakan sebagai alat komunikasi bergerak (Afrizal Fitriandi, Endah Komalasari, Herri Gusmed Volume 10, 2016).

GSM merupakan standar komunikasi yang menyediakan layanan komunikasi dalam bentuk pesan pendek SMS (*Short Message Service*). Layanan komunikasi pertukaran pesan pendek antar pengguna jaringan GSM inilah yang paling banyak digunakan oleh masyarakat. Jangkauan frekuensi standar komunikasi GSM yakni 935-960 MHz untuk transmisi base dan 890-915 MHz untuk transmisi bergerak.

SIM900A adalah modul *Quad-band* GSM/GPRS berbentuk SMT terbuat dari sebuah prosesor canggih ARM926EJ-S, sehingga ukurannya kecil (24mm x 24mm x 3 mm) dan merupakan solusi yang efektif sebagai modul komunikasi.

SIM900A sudah menerapkan antarmuka standar industri dalam menyediakan fitur komunikasi GSM/GPRS 850/900/1800/1900MHz untuk *voice*, SMS, Data, dan Fax.



Gambar 2.17 Modul GSM SIM900A

(Sumber: <http://img.dxcdn.com/>, diakses tanggal 23 Januari 2017)

2.10.1 Fitur Modul GSM SIM900A

1. Memiliki 4 tingkat jaringan frekuensi 850/ 900/ 1800/ 1900 MHz.
2. Paket data GPRS kelas 10/8.
3. GPRS *mobile station* kelas B.
4. Memenuhi standar GSM 2/2+.
5. Kelas 4 (2 W @850/ 900 MHz).
6. Kelas 1 (1 W @ 1800/1900MHz).
7. SAIC (*Single Antenna Interference Cancellation*).
8. Dikontrol dengan AT *Commands* (GSM 07.07 ,07.05 dan SIMCOM *enhanced AT Commands*).
9. SIM *application toolkit*.
10. SMS (*Short Messaging Service*): point-to-point MO & MT, SMS *cell broadcast*, mendukung format teks dan PDU (*Protocol Data Unit*).
11. Mendukung transmisi faksimili (fax group 3 class 1).

2.10.2 Spesifikasi SIM900A

1. Dimensi: 24 x 24 x 3 mm
2. Berat: 3.4g
3. Rentang tegangan antara 5V-7V
4. Rentang suhu operasi -40°C to +85 °C
5. Hemat daya, hanya mengkonsumsi arus sebesar 1 mA pada *sleep mode*.

2.11 LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama, selain itu LCD juga dapat digunakan untuk menampilkan karakter atau gambar.

Dalam modul LCD terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan *register*, memori dan *register* yang digunakan adalah:

1. DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang ditampilkan.
2. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat berubah-ubah sesuai dengan keinginan.
3. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD.
4. *Register* perintah, yaitu *register* yang berisi perintah-perintah dari mikrokontroler ke panel LCD pada saat proses penulisan data atau tempat status dari panel LCD dapat dibaca pada saat pembacaan data.
5. *Register* data, yaitu register menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada *register* akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur.



Gambar 2.18 *Liquid Crystal Display 16x2*

(Sumber: <http://lh5.ggpht.com/>, diakses tanggal 23 Januari 2017)

Pin, kaki atau jalur *input* dan kontrol dalam suatu LCD diantaranya adalah:

1. Pin data adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD (*Liquid Cristal Display*) dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
2. Pin RS (*Register Select*) berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukkan yang masuk adalah perintah, sedangkan logika *high* menunjukkan data.
3. Pin R/W (*Read Write*) berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* baca data.
4. Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
5. Pin VLCD berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

2.11.1 Karakteristik

Modul LCD 16x2 memiliki karakteristik sebagai berikut:

1. Terdapat 16x2 karakter huruf yang bias ditampilkan.
2. Setiap huruf terdiri dari 5x7 dot-matrix kursor.
3. Terdapat 192 macam karakter.
4. Terdapat 80x8 bit *display* RAM (maksimal 80 karakter).
5. Memiliki kemampuan penulisan dengan 8 bit maupun dengan 4 bit.
6. Dibangun dengan osilator local.
7. Satu sumber tegangan 5 volt.

2.11.2 Spesifikasi

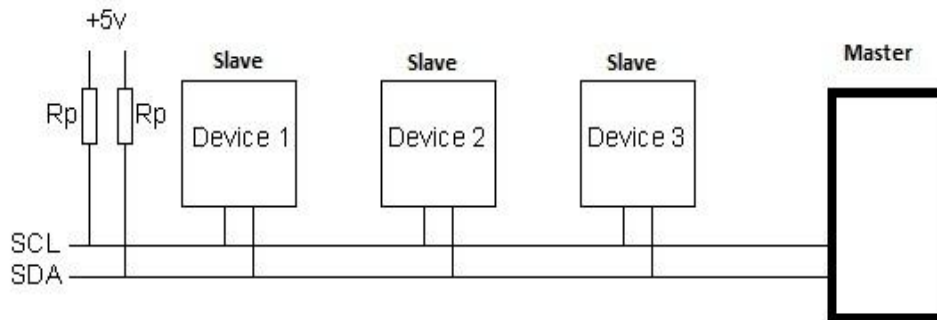
Untuk LCD 16x2 yang dilengkapi dengan modul I2C/TWI yang didesain untuk meminimalkan penggunaan pin pada saat menggunakan display LCD 16x2. Normalnya sebuah LCD 16x2 akan membutuhkan sekurang-kurangnya 8 pin untuk dapat diaktifkan. Namun LCD 16x2 jenis ini hanya membutuhkan 2 pin saja. Adapun spesifikasinya sebagai berikut:

1. I2C Address: 0x3F.
2. *Back lit (Blue with char color).*
3. Sumber tegangan 5 volt.
4. Dimensi: 82x35x18 mm.
5. Berat 40 gram.
6. *Interface: I2C.*

2.11.3 I2C Connector

I2C (*Inter Integrated Circuit*) adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C/TWI terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya serta *pull up* resistor yang digunakan untuk transfer data antar perangkat. I2C juga merupakan transmisi serial setengah duplex oleh karena itu aliran data dapat diarahkan pada satu waktu.

Tingkat transfer data mengacu pada sinyal *clock* pada SCL Bus 1/16th *slave* informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal *start*, mengakhiri transfer data dengan membentuk sinyal *stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati master, yang bertindak sebagai master disini adalah mikrokontroler. Adapun konfigurasi fisik I2C dapat dilihat pada gambar 2.19.



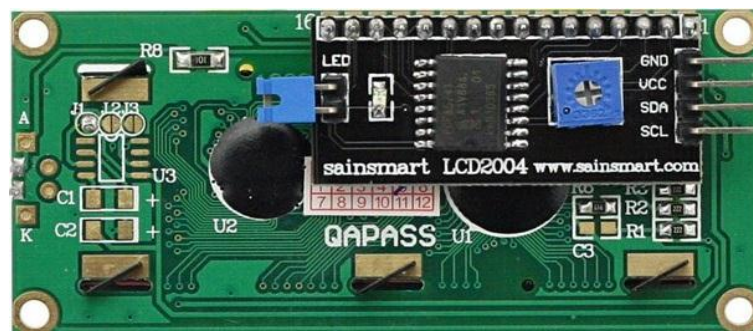
Gambar 2.19 Konfigurasi fisik I2C

(Sumber: <https://pccontrol.files.com/>, diakses tanggal 11 April 2017)

2.11.4 Interface Komunikasi I2C

Pada LCD 16x2 yang dilengkapi dengan I2C sistem komunikasi hanya memerlukan 4 buah pin yang dihubungkan, yaitu:

1. Ground.
2. VCC 5 volt.
3. Pin SDA.
4. Pin SCL.



Gambar 2.20 Komunikasi Pin I2C

(Sumber: <https://www.sainsmart.com/>, diakses tanggal 11 April 2017)

Pada modul I2C juga dilengkapi dengan potensiometer yang dapat digunakan untuk menyesuaikan kontras cahaya dengan memutar searah jarum jam untuk mendapatkan tampilan yang diinginkan.

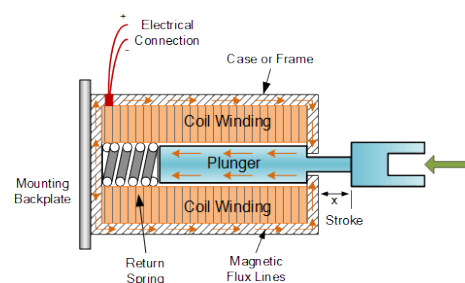
2.12 Solenoid Door Lock

Solenoid adalah aktuator yang mampu melakukan gerakan linier yaitu gerakan lurus menarik atau mendorong. Solenoid DC dapat bekerja secara elektromekanis dengan memberikan sumber tegangan, maka solenoid dapat menghasilkan gaya yang linier (Pratama, 2014).

Secara ideal, solenoid memiliki panjang lilitan yang tak berhingga dengan lilitan dari kabelnya yang rapat saling berhimpit satu sama lainnya. Maka akan menghasilkan medan elektromagnet yang sama dan konstan yang bersifat paralel terhadap inti besi yang menjadi sumbunya.

Apabila kita alirkan listrik kepada batang besi yang kita tempatkan di tengah lilitan, maka batang besi tersebut akan mendapatkan induksi magnet dan akhirnya dapat menjadi magnet. Dengan penempatan sebagian batang besi tersebut berada di dalam solenoid dan sebagiannya lagi di sebelah luarnya. Batang besi yang terinduksi magnet tersebut akan menarik masuk benda berbahan logam ke dalam solenoid. Hal ini yang dimanfaatkan untuk menggerakkan tuas, menutup dan mengunci pintu, atau menggerakkan slot kunci pintu. Prinsip kerja dari sebuah solenoid DC cukup mirip dengan sebuah solenoida AC, keduanya dirancang khusus dan menghasilkan medan elektromagnet.

Inti besi yang berbentuk bulat dan kerucut itu, salah satu ujungnya memiliki kutub positif. Ketika inti besi tersebut dimasukkan ke tengah kumparan yang penuh dengan medan magnet, maka permukaan ujung yang satunya lagi memiliki kutub negatif. Sementara di bagian bawahnya terdapat area yang cukup luas untuk menyalurkan aliran fluks magnet tersebut.



Gambar 2.21 Solenoid Door Lock

(Sumber: http://mechatronics.mech.northwestern.edu/design_ref/actuators/solenoids.html, diakses 23 Januari 2017)

2.13 Buzzer

Buzzer adalah komponen elektronika yang berfungsi mengubah energi listrik menjadi suara. Buzzer terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan akan tertarik ke dalam atau keluarsesuai arah arus dan polaritas magnetnya, karena diafragma dalam kumparan maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar dan menghasilkan suara (Pratama, 2014).



Gambar 2.22 Buzzer

(Sumber: <http://p.globalsources.com/>, diakses tanggal 23 Januari 2017)

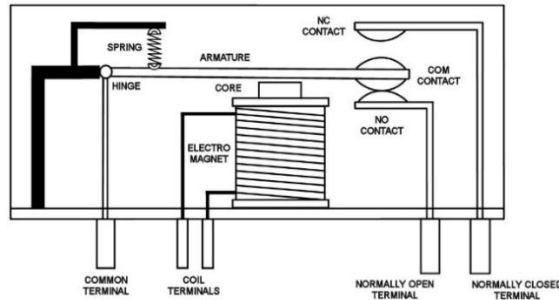
2.14 Relay

Relay adalah saklar mekanik yang dikendalikan atau dikontrol secara elektronik (elektro magnetik). Saklar pada *relay* akan terjadi perubahan posisi OFF ke ON pada saat diberikan energi elektro magnetik pada armatur relay tersebut. *Relay* pada dasarnya terdiri dari 2 bagian utama yaitu saklar mekanik dan sistem pembangkit elektromagnetik (induktor inti besi). Saklar atau kontaktor relay dikendalikan menggunakan tegangan listrik yang diberikan ke induktor pembangkit magnet untuk menarik armatur tuas saklar atau kontaktor *relay*.

Relay elektro mekanik memiliki kondisi saklar atau kontaktor dalam 3 posisi. Ketiga posisi saklar atau kontaktor relay ini akan berubah pada saat *relay* mendapat tegangan sumber pada elektromagnetnya. Ketiga posisi saklar *relay* tersebut adalah :

1. Posisi *Normally Open* (NO) , yaitu posisi saklar *relay* yang terhubung ke terminal NO (*Normally Open*). Kondisi ini akan terjadi pada saat *relay* mendapat tegangan sumber pada elektromagnetnya.

2. Posisi *Normally Colse* (NC), yaitu posisi saklar *relay* yang terhubung ke terminal NC (*Normally Close*). Kondisi ini terjadi pada saat *relay* tidak mendapat tegangan sumber pada elektromagnetnya.
3. Posisi *Change Over* (CO), yaitu kondisi perubahan armatur saklar *relay* yang berubah dari posisi NC ke NO atau sebaliknya dari NO ke NC.



Gambar 2.23 Relay

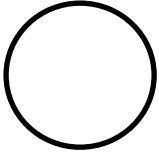

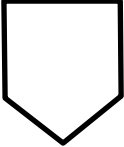

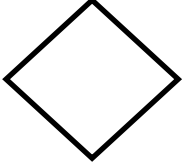



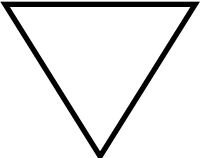
(Sumber: <http://www.glolab.com/>, diakses tanggal 23 Januari 2017)

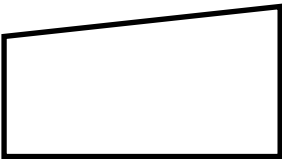

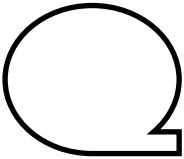



2.15 Flowchart

Flowchart atau diagram alir merupakan representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri dari sekumpulan simbol, dimana masing-masing simbol mempresentasikan suatu kegiatan tertentu. *Flowchart* diawali dengan menerima input, pemrosesan input, dan diakhiri dengan menampilkan output (Agus, 2014).

Tabel 2.2 Simbol-simbol dan keterangan *Flowchart*

No	Simbol	Keterangan
1.		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu program.

2.		Simbol <i>connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
3.		Simbol <i>manual</i> , yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
4.		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
5.		Simbol <i>process</i> , menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
6.		Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya/tidak.
7.		Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program.
8.		Simbol <i>predefined process</i> , yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
9.		Simbol <i>keying operation</i> , menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.
10.		Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.

11.		Simbol <i>manual input</i> , memasukkan data secara manual dengan menggunakan online keyboard.
12.		Simbol <i>input/output</i> , menyatakan proses masukan atau keluaran tanpa tergantung jenis peralatannya.
13.		Simbol <i>magnetic tape</i> , menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis.
14.		Simbol <i>disk storage</i> , menyatakan input berasal dari disk atau output disimpan ke disk.
15.		Simbol <i>document</i> , mencetak keluaran dalam bentuk dokumen.
16.		Simbol <i>display</i> , mencetak keluaran dalam layar monitor.