

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian terdahulu menjadi salah satu acuan penulisan, sehingga penulis dapat memperkaya landasan teori pendahuluan yang digunakan dalam mengkaji penelitian. Beberapa rujukan Penelitian tentang Sistem Kendali Pintu Menggunakan RFID akan dijabarkan dibawah.

Rujukan penelitian yang pertama yaitu jurnal yang ditulis oleh Sinta Ariyanti, Slamet Seno Adi, dan Sugeng Purbawanto mahasiswa Universitas Negeri Semarang pada tahun 2018 dengan judul Sistem Buka Tutup Pintu Otomatis Berbasis Suara Manusia. Dalam penelitiannya peneliti menggunakan sensor suara untuk bisa membuka dan menutup pintu. Dimana ketika pengguna mengucapkan suara kata buka maka akan membuka kunci dan pintu akan terbuka, demikian juga ketika pengguna ingin menutup pintu dengan mengucapkan kata tutup maka pintu akan tertutup dan menutup kunci.

Rujukan penelitian yang kedua yaitu jurnal yang ditulis oleh Akbar Iskandar, Muhajirin, Lisah, mahasiswa STMIK AKBA pada tahun 2017 dengan judul Sistem Keamanan Pintu Berbasis Arduino. Dalam penelitiannya peneliti menggunakan sensor Fingerprint dan kamera OV7670 untuk membuka pintu dan untuk membuka pintu dari dalam ruangan menggunakan *push button*. Dimana ketika pengguna menempelkan jari ke sensor *fingerprint* maka kunci dan pintu akan terbuka dan kamera akan langsung mengambil gambar secara otomatis dan dan apa bila ingin keluar dari ruangan dengan menekan tombol *Push Button* maka kunci dan pintu akan terbuka secara otomatis.

Setelah melakukan pengamatan tentang judul dan permasalahan dari jurnal-jurnal yang berkaitan dengan kasus sistem keamanan pada pintu, penulis menemukan sejumlah judul yang hampir sama. Oleh karena itu, dalam laporan akhir ini penulis berusaha untuk mencari perbedaan dari perancangan terdahulu. Cara tersebut dapat dilakukan dengan mencari beberapa hasil penelitian terdahulu.

Terdapat beberapa persamaan dan perbedaan antara penelitian terdahulu dan penelitian yang sedang dilakukan pada saat ini. Hal ini bertujuan untuk

membuktikan bahwa penulisan laporan akhir ini asli dan bukan sebuah duplikasi dari skripsi atau laporan akhir lain.

Tabel 2.1. Perbandingan Penelitian

No	Judul Penelitian Terdahulu	Persamaan	Perbedaan
1	Sistem Buka Tutup Pintu Otomatis Berbasis Suara Manusia oleh Sinta Ariyanti, Slamet Seno Adi, dan Sugeng Purbawanto	1.Membuka dan menutup pintu secara otomatis. 2.Menggunakan papan Mikrokontroler <i>Arduino Uno</i> .	1. Menggunakan servo motor untuk membuka dan menutup pintu. 2. Menggunakan Sensor suara sebagai input.
2	Sistem Keamanan Pintu Berbasis <i>Arduino Mega</i>	1.Buka dan tutup kunci menggunakan solenoid <i>door lock</i> 2.Sistem menggunakan bahasa pemrograman C pada <i>Arduino</i> . 3.Menggunakan LCD sebagai indikator pintu bisa dibuka atau tidak	1. Menggunakan sensor <i>fingerprint</i> dan kamera untuk membuka pintu dari luar.

2.2. Mikrokontroler

Mikrokontroler adalah sebuah sistem computer yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering disebut single chip mikrokomputer. Mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik (Chamim, 2010:1).

Pada pembuatan sistem keamanan pintu perpustakaan menggunakan RFID berbasis mikrokontroler ini penulis menggunakan mikrokontroler ATmega328. Untuk penjelasan mengenai mikrokontroler ATmega328 akan dijelaskan dibawah.

1.2.1. Mikrokontroler ATmega328

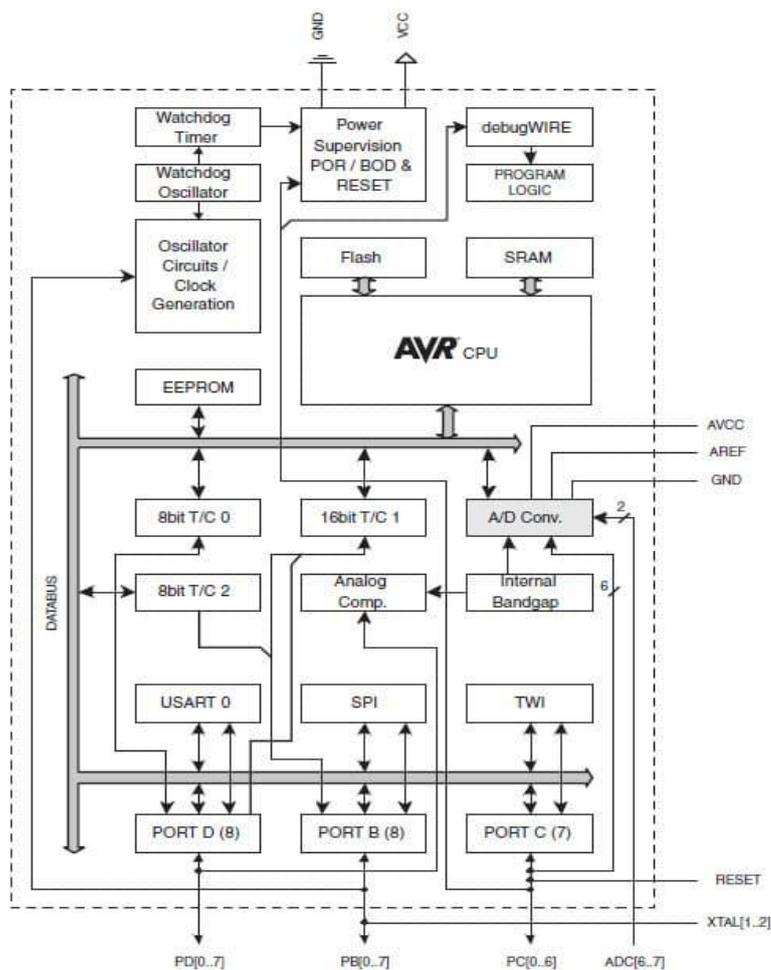
ATmega328 merupakan mikrokontroler keluarga AVR 8 bit. Beberapa tipe mikrokontroler yang sama dengan ATmega 8 ini antara lain ATmega 8535, ATmega 16, ATmega 32, ATmega 328, yang membedakan antara mikrokontroler antara lain adalah ukuran memori, banyaknya GPIO (pin *input/output*), *peripheral* (USART, *timer*, *counter*, dll). Dari segi ukuran fisik, ATmega 328 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa mikrokontroler diatas. Namun untuk segi memori dan *peripheral* lainnya ATmega 328 tidak kalah dengan yang lainnya karena ukuran memori dan *peripheral*-nya relatif sama dengan ATmega 8535, ATmega 32, hanya saja jumlah GPIO lebih sedikit dibandingkan mikrokontroler diatas. Wicaksono (2017:6)

Mikrokontroler ATmega328 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi-instruksi dapat dieksekusi dalam setiap satu siklus *clock*. 32 x 8-bit register serba guna digunakan untuk mendukung operasi pada ALU (*Arithmetic Logic unit*) yang dapat dilakukan dalam satu siklus. 6 dari register serbaguna ini dapat digunakan sebagai 3 buah register pointer 16-bit pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data.

Ketiga register pointer 16-bit ini disebut dengan register X (gabungan R26 dan R27), register Y (gabungan R28 dan R29), dan register Z (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki format 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit.

Selain register serba guna di atas, terdapat register lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 byte. Beberapa register ini digunakan untuk fungsi khusus antara lain sebagai register control *Timer/ Counter*, Interupsi, ADC, USART, SPI, EEPROM, dan fungsi I/O lainnya. Register-register ini menempati memori pada alamat 0x20h – 0x5Fh.

1.2.2. Blok Diagram Atmega 328



Gambar 2.1. Blok Diagram Atmega328

Penjelasan diagram blok pada gambar 2.1 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. 2KB RAM pada memori kerja bersifat *volatile* (hilang saat daya dimatikan), digunakan oleh variabel-variabel di dalam program.
2. 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.
3. 1KB EEPROM bersifat *non-volatile*, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan. Tidak digunakan pada papan Arduino.
4. *Central Processing Unit* (CPU), bagian dari mikrokontroler untuk menjalankan setiap instruksi dari program.

Port *input/output*, pin-pin untuk menerima data (*input*) digital atau analog, dan mengeluarkan data (*output*) digital atau analog. (Ichwan, 2013)

1.2.3. Peta Memori ATmega328

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

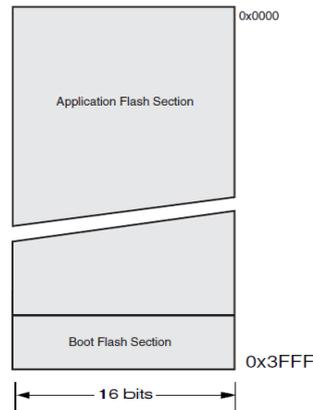
1. Memori EEPROM ATmega328

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 2.2.

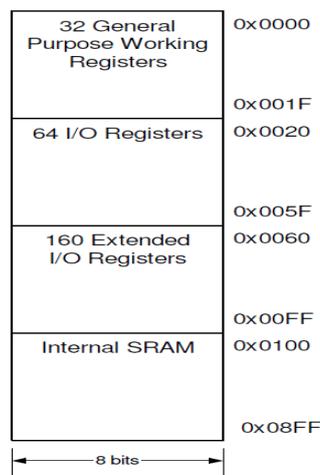
Bootloader adalah program kecil yang bekerja pada saat sistem dimulai yang dapat memasukkan seluruh program aplikasi ke dalam memori prosesor.



Gambar 2.2. Peta Memori Program ATmega328

3. Memori Data ATmega328

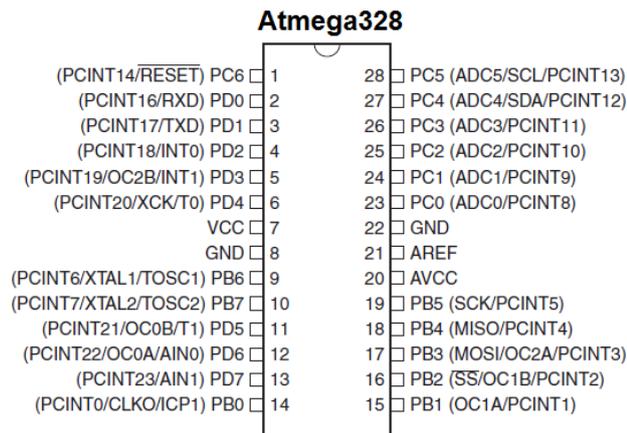
Memori data ATmega328 terbagi menjadi 4 bagian, yaitu 32 lokasi untuk register umum, 64 lokasi untuk register I/O, 160 lokasi untuk register I/O tambahan dan sisanya 2048 lokasi untuk data SRAM internal. Register umum menempati alamat data terbawah, yaitu 0x0000 sampai 0x001F. Register I/O menempati 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 2.3. Peta Memori Data ATmega328

1.2.4. Konfigurasi Pin pada Mikrokontroler Atmega 328

ATMega 328 memiliki 3 buah *PORT* utama yaitu *PORT B*, *PORT C*, dan *PORT D* dengan total pin *input/output* sebanyak 23 pin. *PORT* tersebut dapat difungsikan sebagai *input/output digital* atau difungsikan sebagai *peripheral* lainnya. (Wicaksono, 2017:7)



Gambar 2.4. Konfigurasi Pin pada Mikrokontroler ATmega 328

1. *Port B*

Port B merupakan jalur data 8 *bit* yang dapat difungsikan sebagai *input/output*. Selain itu *PORT B* juga dapat memiliki fungsi alternatif seperti di bawah ini.

- a. ICP1 (PB0), berfungsi sebagai *Timer Counter 1 input capture* pin.
- b. OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran PWM (*Pulse Width Modulation*).
- c. MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI.
- d. Selain itu pin ini juga berfungsi sebagai jalur pemrograman *serial* (ISP).
- e. TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock external* untuk *timer*.
- f. XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.

2. *Port C*

Port C merupakan jalur data 7 bit yang dapat difungsikan sebagai *input/output* digital. Fungsi alternatif *PORTC* antara lain sebagai berikut.

- a. ADC6 channel (PC0,PC1,PC2,PC3,PC4,PC5) dengan resolusi sebesar 10 bit. ADC dapat kita gunakan untuk mengubah input yang berupa tegangan analog menjadi data *digital*.
- b. I2C (SDA dan SDL) merupakan salah satu fitur yang terdapat pada *PORTC*. I2C digunakan untuk komunikasi dengan sensor atau *device* lain yang memiliki komunikasi data tipe I2C seperti sensor kompas, *accelerometer nunchuck*.

3. *Port D*

Port D merupakan jalur data 8 bit yang masing-masing pin-nya juga dapat difungsikan sebagai *input/output*. Sama seperti *Port B* dan *Port C*, *Port D* juga memiliki fungsi alternatif dibawah ini.

- a. USART (TXD dan RXD) merupakan jalur data komunikasi *serial* dengan level sinyal TTL. Pin TXD berfungsi untuk mengirimkan data *serial*, sedangkan RXD kebalikannya yaitu sebagai pin yang berfungsi untuk menerima data *serial*.
- b. *Interrupt* (INT0 dan INT1) merupakan pin dengan fungsi khusus sebagai interupsi *hardware*. Interupsi biasanya digunakan sebagai selaan dari program, misalkan pada saat program berjalan kemudian terjadi interupsi *hardware/software* maka program utama akan berhenti dan akan menjalankan program interupsi.
- c. XCK dapat difungsikan sebagai sumber *clock external* untuk USART, namun kita juga dapat memanfaatkan *clock* dari CPU, sehingga tidak perlu membutuhkan *external clock*.
- d. T0 dan T1 berfungsi sebagai masukan *counter external timer 1* dan *timer 0*. AIN0 dan AIN1 keduanya merupakan masukan *input* untuk *analog comparator*.

1.2.5. Fitur Mikrokontroler ATmega 328

ATmega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang mana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*) Wicaksono (2017:9).

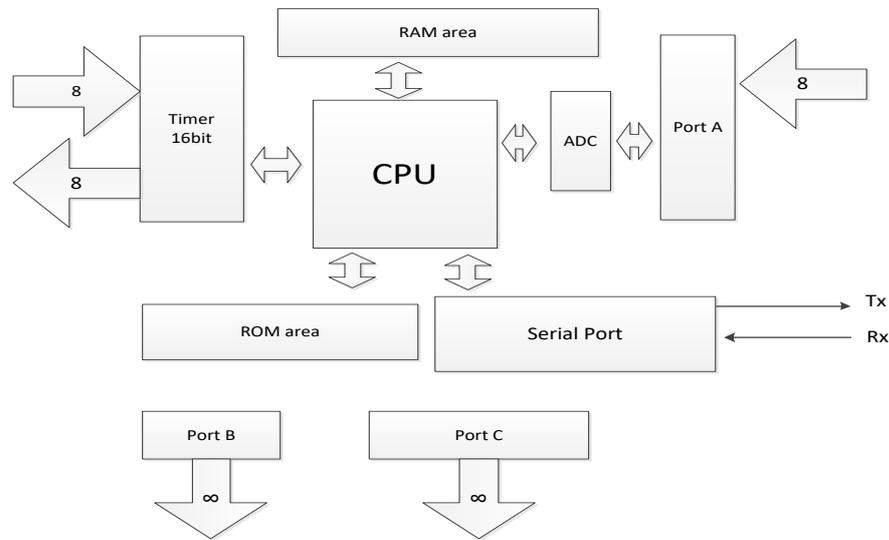
Mikrokontroler ini memiliki beberapa fitur antara lain:

1. Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1 KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
2. Memiliki SRAM (*Static Random Access Memory*) sebesar 2 KB.
3. Memiliki pin I/O *digital* sebanyak 14 pin 6 diantaranya PWM (*Pulse Width Modulation*) output.
4. 32 x 8-bit *register* serba guna.
5. Dengan *clock* 16 MHz kecepatan mencapai 16 MIPS. 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari *flash memory* sebagai *bootloader*. 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.

2.3. Arduino

Arduino adalah sebuah *platform* dari *physical computing* yang bersifat *open source*. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi Arduino adalah kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory* mikrokontroler (Djuandi, 2011:7).

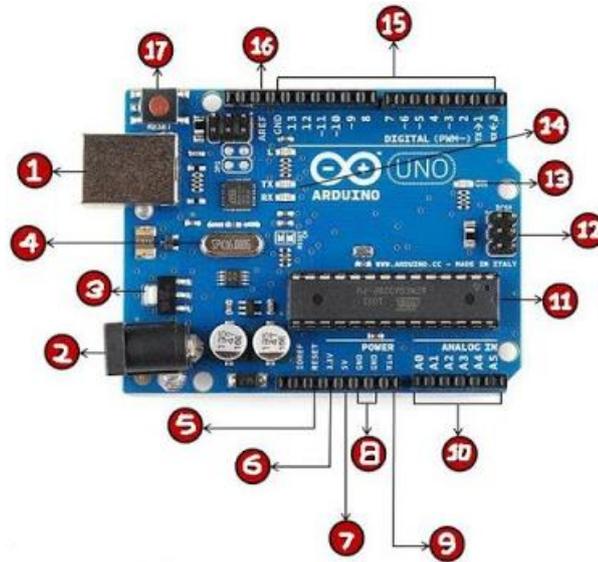
Untuk *board* Arduino ada berbagai macam seperti *Uno*, *Due*, *Mega*, *Leonardo*, *Fio*, *Lilypad*, *Nano*, *Mini*, *Ethernet*, *Esplora*, dan *Robot*. Tiap macam board Arduino berbeda fungsi, ukuran, memori, dan ada tambahan fitur-fitur yang lain tergantung sesuai kebutuhan dan pemakaian.



Gambar 2.5. Blok Diagram Arduino

Berdasarkan gambar diatas CPU berfungsi sebagai pemroses data berupa fungsi logika dan aritmatika. Untuk memori ada RAM sebagai penyimpanan data sementara dan ROM digunakan sebagai tempat penyimpanan program. *Timer* digunakan untuk menghasilkan tunda waktu dan mengukur selang waktu suatu proses. ADC berfungsi merubah sinyal masukan analog menjadi sinyal masukan digital dan dapat digunakan untuk berkomunikasi antara mikrokontroler dengan piranti-piranti eksternal (sensor) yang memiliki gelombang sinyal keluaran berupa gelombang sinyal analog (sinus). Port A, B, dan C pada adalah piranti antarmuka ke *input/output*. Dan komunikasi dengan piranti lain menggunakan komunikasi *serial*. Pada Sistem Keamanan Pintu Perpustakaan Menggunakan RFID Berbasis Mikrokontroler ini menggunakan Arduino *Uno*.

Arduino *Uno* adalah papan pengembang mikrokontroler yang berbasis *chip* ATmega328. Arduino *Uno* memiliki 14 digital *pin input/output* (dimana 14 pin diantaranya dapat digunakan sebagai output PWM antara lain pin 0 sampai 13), 6 pin input analog, menggunakan *crystal* 16 MHz, koneksi USB, port *jack*, *header* ICSP dan tombol *reset*. Hal tersebut adalah semua yang diperlukan untuk mendukung sebuah rangkaian mikrokontroler.



Gambar 2.6. Fitur arduino uno

Tabel 2.2. Penjelasan gambar fitur arduino uno

No	Nama	Deskripsi
1	<i>Power USB</i>	Digunakan untuk menghubungkan Papan Arduino dengan komputer lewat koneksi USB. sebagai supply listrik ke papan atau untuk pemrograman mikrokontroler.
2	<i>Power Jack</i>	Supply atau sumber listrik untuk Arduino dengan tipe Jack. Input DC 5 - 12 V.
3	<i>Voltage Regulator</i>	IC ini digunakan untuk menstabilkan tegangan Eksternal dari Jack No.2 menuju 5 V, tegangan aman Papan Arduino.
4	<i>Crystal Oscillator</i>	Difungsikan sebagai umpan balik pada suatu frekuensi, pada arduino uno menunjukkan 16000 kHz.
5	<i>Pin Reset</i>	Digunakan untuk mengulang program Arduino dari awal atau Reset dengan menggabungkan pin reset dengan GND.
6	Pin 3.3 V	Sumber tegangan output 3.3 Volt.

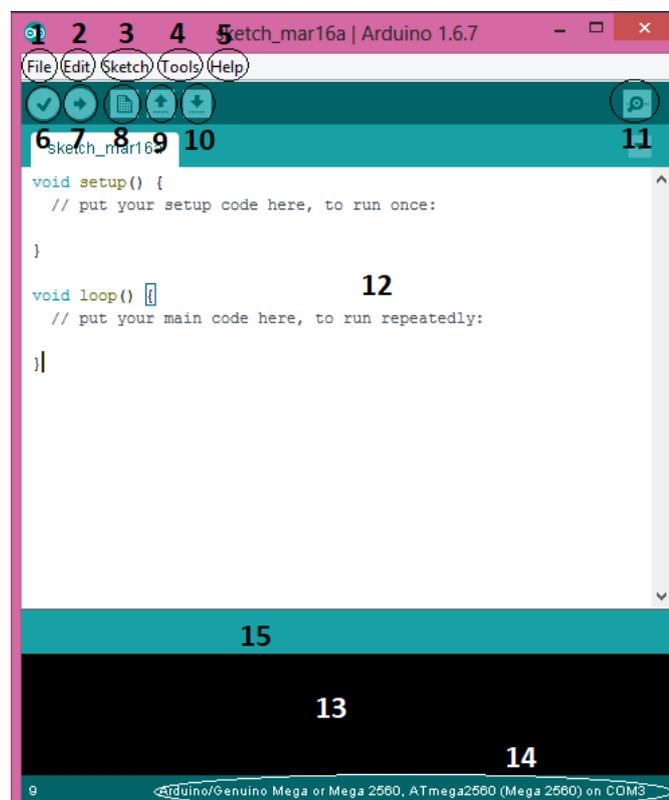
7	Pin 5V	Sumber tegangan <i>output</i> 5 Volt
8	Pin GND	<i>Ground</i> dalam sirkuit elektronik adalah akhir dari setiap jalur arus listrik.
9	Pin Vin	Pin untuk memasok listrik dari luar ke papan arduino, sekitar 5 V.
10	Analog Pins	Papan Arduino UNO memiliki enam pin analog A0 sampai A5. Digunakan untuk membaca sinyal atau sensor analog seperti sensor jarak, suhu dsb, dan mengubahnya menjadi nilai digital.
11	IC Mikrokontroler	IC atau <i>Integrated Circuit</i> , alias otak dari Papan Arduino. IC ini yang diprogram oleh papan arduino untuk mengatur pin digital (15) dan pin analog (10).
12	Pin ICSP	Sebagian besar ICSP (12) adalah untuk AVR. Dalam Arduino terdapat enam pin, MOSI, MISO, SCK, RESET, VCC, dan GND. bisa digunakan dengan <i>Bootloader</i> .
13	LED <i>Power Indicator</i>	LED akan menyala dan menandakan Papan Arduino mendapatkan <i>supply</i> listrik dengan baik. Jika tidak menyala berarti ada sesuatu yang salah dengan <i>supply</i> listrik atau papan arduinonya.
14	LED TX dan RX	TX (<i>Transmit</i>) dan RX (<i>Receive</i>), dua LED tersebut akan berkedip saat pemrograman IC atau Papan Arduino berlangsung.
15	Digital Pin I/O	Papan Arduino UNO memiliki 14 Digital Pin. Berfungsi untuk memberikan nilai logika (0 atau 1). Pin berlabel " ~ " adalah pin-pin PWM (<i>Pulse Width Modulation</i>) yang dapat digunakan untuk menghasilkan PWM. Digital Pin I / O dapat digunakan seperti saklar.

16	AREF	AREF singkatan <i>Analog Reference</i> . Dapat digunakan untuk mendapatkan sumber tegangan yang dapat diatur lewat IC. Tegangannya antara 0 sampai 5 Volt.
17	Tombol <i>Reset</i>	Digunakan untuk mengulang program Arduino dari awal atau <i>Reset</i> dengan menekan tombol.

2.4. *Integrated Development Environment (IDE) Arduino*

IDE (*Integrated Development Environment*) adalah sebuah perangkat lunak yang digunakan untuk mengembangkan aplikasi mikrokontroler mulai dari menuliskan *source* program, kompilasi, *upload* hasil kompilasi dan uji coba secara terminal *serial*. (Wicaksono, 2017:4)

Program yang ditulis dengan menggunakan IDE Arduino disebut *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *.ino*.



Gambar 2.7. IDE Arduino

Tabel 2.3. Keterangan dari gambar IDE Arduino

No	Nama	Deskripsi
1	<i>File</i>	Digunakan untuk berinteraksi dengan file yang berekstensi <i>.ino</i> .
2	<i>Edit</i>	Digunakan untuk mengedit program yang sedang ditulis pada <i>sketch</i> editor.
3	<i>Sketch</i>	Berisi fungsi untuk memverifikasi <i>sketch</i> yang telah dibuat sebelumnya
4	<i>Tools</i>	Mengandung beberapa fungsi seperti perintah untuk menentukan jenis <i>Board</i> Arduino yang akan digunakan serta port USB.
5	<i>Help</i>	Berisikan berkas-berkas dokumentasi yang berkaitan dengan masalah yang sering muncul, serta penyelesaiannya.
6	<i>Verify</i>	Berfungsi untuk melakukan pemeriksaan kode program yang dibuat.
7	<i>Upload</i>	Berfungsi untuk melakukan kompilasi program.
8	<i>New</i>	Berfungsi untuk membuat <i>sketch</i> baru.
9	<i>Open</i>	Berfungsi untuk membuka <i>sketch</i> yang pernah dibuat.
10	<i>Save</i>	Berfungsi untuk menyimpan <i>sketch</i> yang telah dibuat.
11	<i>Serial Monitor</i>	Berfungsi untuk membuka <i>serial monitor</i>
12	<i>Sketch Editor</i>	Tempat menuliskan program <i>sketch/arduino</i>
13	Konsol Log	Menampilkan status, seperti pesan <i>error</i> , <i>compile</i> , dan <i>upload</i> program Arduino.
14	Informasi <i>Board</i> dan Port	Bagian ini menginformasikan port yang dipakai oleh <i>board</i> Arduino.

15	Keterangan Aplikasi	Pesan-pesan yang dikerjakan aplikasi dan <i>sketch</i> .
----	---------------------	--

2.5. *Radio Frequency Identification Device (RFID)*

RFID adalah sebuah istilah umum yang digunakan untuk menjelaskan sebuah alat yang mentransmisikan identitas (dalam bentuk deretan nomor) dari sebuah objek atau manusia secara nirkabel, menggunakan gelombang radio. RFID dikelompokkan sebagai teknologi *automatic identification*.(Dwitama, 2009:7)

Teknologi identifikasi otomatis meliputi *barcode*, pembaca karakter optikal dan beberapa teknologi biometric, seperti pemindai retina. Teknologi identifikasi otomatis telah digunakan untuk mengurangi jumlah waktu dan tenaga kerja yang dibutuhkan untuk memasukkan data secara manual dan meningkatkan akurasi data.

RFID tag terdiri dari sebuah microchip yang tertanam pada sebuah antenna gelombang radio yang ditempelkan pada sebuah substrat (biasanya *silicon*). *Chip* tersebut dapat menyimpan data seukuran *2 kilobyte*. Sebagai contoh, informasi tentang sebuah barang atau tanggal pengkapalan pada sebuah pabrik dan sebagainya, dapat dituliskan pada sebuah tag.

Untuk mengambil data yang disimpan pada RFID tag, dibutuhkan sebuah *reader*. *Reader* ini adalah sebuah alat yang memiliki satu atau banyak *antenna* yang memancarkan gelombang radio dan menerima sinyal kembali dari *tag*. *Reader* kemudian melewatkan informasi tersebut dalam bentuk digital ke sistem komputer. Ada sistem RFID terdiri dari 4 Komponen yaitu:

1. RFID tag (*transponder*) memiliki *chip* yang dapat menyimpan data berupa nomor ID unik dan memiliki *antenna* yang berfungsi untuk mentransmisikan data ke RFID *reader* melalui radio yang dipancarkan RFID *reader*.
2. Antena terdapat pada RFID tag (*tag-antena*) dan RFID *reader* (*reader-antena*) 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 antenna yang terdapat pada rangkaian RFID *tag* melalui gelombang radio yang dipancarkan RFID *reader*.

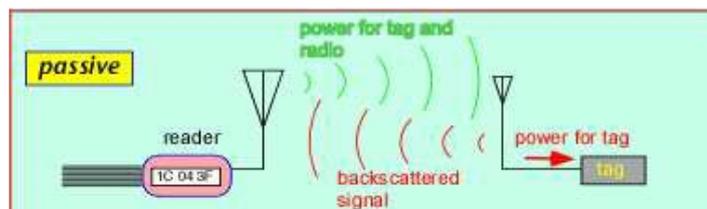
4. *Interface Software* berfungsi untuk membaca data ID dari RFID *reader* dan mengelola data tersebut sehingga dapat digunakan *password*.

2.5.1 RFID Tag

RFID *tag* adalah *device* yang dibuat dari rangkaian elektronika dan antenna yang terintegrasi didalam rangkaian tersebut. 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* yang berfungsi untuk melindungi *chip* agar tidak mudah rusak. RFID tag dibagi menjadi 2 berdasarkan catu dayanya, yaitu:

1. Tag Pasif (Passive Tags)

RFID *tag* jenis ini tidak memiliki catu daya sendiri, catu dayanya diterima dari medan elektromagnetik yang dipancarkan oleh RFID *reader*. RFID *tag* akan aktif dan dapat mengirim data hanya ketika didekatkan dengan RFID *reader*. Tag pasif dapat beroperasi atau dibaca oleh RFID *reader* dengan jarak sekitar beberapa sentimeter sampai 10m. Cara kerja RFID *tag* pasif dapat dilihat pada gambar 2.6.



Gambar 2.8. Cara Kerja RFID Tag Pasif

Berikut ini karakteristik dari RFID tag pasif yang dapat dilihat pada tabel 2.4.

Tabel 2.4. Karakteristik RFID Tag Pasif

NO	Karakteristik Tag Pasif
1	Tidak memiliki sumber tegangan sendiri
2	Modulasi akan aktif ketika tag menerima gelombang elektromagnetik dari <i>reader</i>

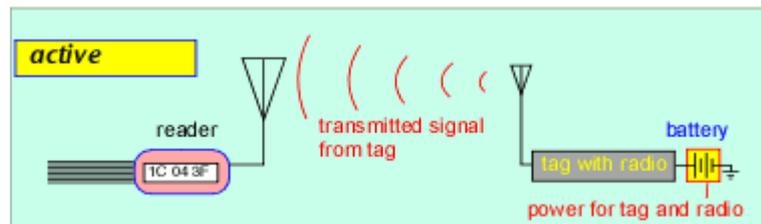
3	Jarak baca 0cm-10cm
4	Praktis dan mudah dibawa

2. Tag Aktif (Active Tags)

RFID *tag* aktif adalah RFID tag yang memiliki baterai sebagai catu dayanya sendiri dan memiliki radio *transmitters* sehingga dapat mengirimkan informasi ke RFID *reader* dengan jarak yang jauh dibandingkan dengan RFID *tag* pasif. Ada dua jenis *tag* aktif yaitu *transponder* dan *beacon*.

Transponder aktif terbangun ketika menerima sinyal dari *reader*. Ketika sebuah benda dengan *transponder* aktif mendekati objek, *reader* pada objek akan mengirim sinyal yang membangkitkan *transponder* pada benda. Kemudian *transponder* itu memancarkan waktu hidup baterai karena *tag* hanya memancarkan sinyal ketika dalam jangkauan *reader* nya.

Beacon memancarkan sinyal dengan ID uniknya pada interval yang diatur sebelumnya. Sinyal *beacon* ditangkap paling sedikit oleh 3 antena radar yang diletakkan sekitar garis keliling daerah dimana asset tersebut dilacak.



Gambar 2.9. Cara Kerja RFID Tag Aktif

Karakteristik dari RFID *tag* aktif dapat dilihat pada tabel 2.5.

Tabel 2.5. Karakteristik RFID Tag Aktif

No	Karakteristik Tag Aktif
1	Memiliki sumber tegangan sendiri (baterai)
2	Modulasi akan aktif dari <i>tag</i> sendiri
3	Harganya lebih mahal daripada <i>tag</i> pasif
4	Ukuran lebih besar dan tidak praktis

Setiap RFID *tag* terdiri dari 3 bagian yaitu:

1. Mikroprosesor

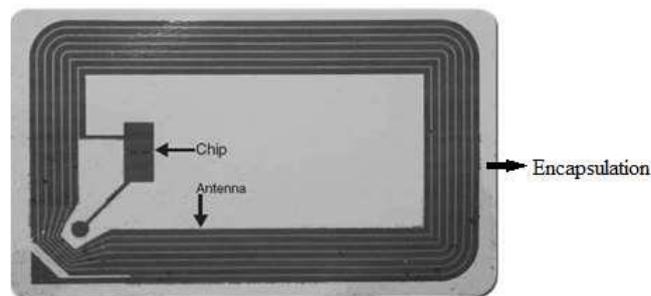
Mikroprosesor adalah chip yang terletak dalam sebuah RFID *tag* yang berfungsi sebagai penyimpan data.

2. *MetalCoil*

MetalCoil terbuat dari kawat aluminium yang berfungsi sebagai antenna yang dapat beroperasi pada frekuensi 13.56MHz. Apabila RFID *tag* masuk ke dalam jangkauan *reader* maka antenna akan mengirimkan data yang ada pada *tag* kepada *reader* terdekat.

3. *Encapsulation*

Encapsulation adalah bahan yang berfungsi untuk melindungi RFID *tag* dan antenna terbuat dari bahan plastik atau kaca.



Gambar 2.10. Bagian RFID Tag

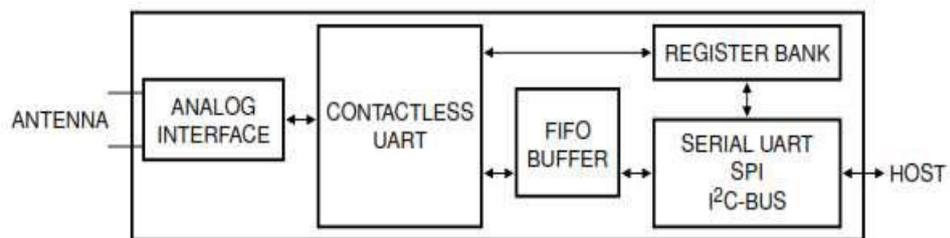
2.5.2 RFID Reader

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 (Riza Muharrir, 2014:8).

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

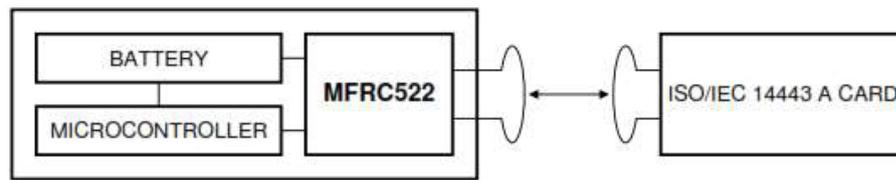
mengirimkan kode yang terdapat di memori ID *chip* melalui antena yang terpasang di RFID tag.

RFID *reader* akan mengirim data tersebut ke mikrokontroler untuk diproses menjadi *password* sebagai pengaman pintu. 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* kepada HOST (mikrokontroler) dan dari mikrokontroler ke *contactless UART*, data yang dikirim berupa data serial (*Datasheet RFID Reader MFR 522*). Blok diagram cara kerja RFID *reader* sebagai *receiver* dan *transfer* data dapat dilihat pada gambar 2.9.



Gambar 2.11. Cara kerja RFID *reader* sebagai *receiver* dan *transfer* data

RFID *reader* yang digunakan pada rangkaian adalah RFID *reader* dengan frekuensi 13.56MHz. RFID *reader* 13.56MHz dapat digunakan untuk membaca RFID *tag* jenis *high frequency* (HF) yang digunakan sebagai *smart card*. RFID *reader* 13.56MHz 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 2.14.



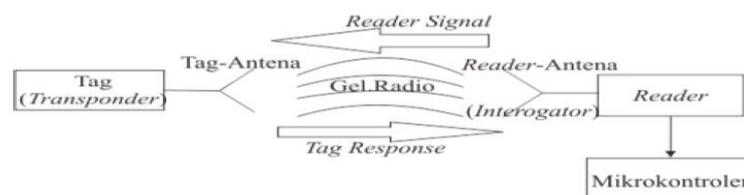
Gambar 2.12. RFID Reader Membaca Data ID Dari Smart Card

2.5.3 Cara Kerja RFID

Suatu sistem RFID dapat terdiri dari beberapa komponen seperti *tag*, *tag reader*, *tag programming station*, *circulation reader*, *sorting equipment* dan tongkat *inventory tag*. Keamanan dapat dicapai dengan dua cara. Pintu *security* dapat melakukan *query* untuk menentukan status keamanan atau RFID *tag*-nya berisi *bit security* yang bisa menjadi *on* atau *off* pada saat didekatkan ke *reader*.

Kegunaan dari sistem RFID ini adalah untuk mengirim data dari piranti *portable* yang dinamakan *tag* dan kemudian dibaca oleh RFID *reader* dan kemudian diproses oleh aplikasi komputer yang membutuhkannya. Data yang dipancarkan dan dikirimkan tadi berisi beragam informasi seperti ID, informasi lokasi atau informasi lainnya seperti harga, warna, tanggal pembelian dan lain sebagainya.

Dalam suatu sistem RFID, suatu objek dilengkapi dengan *tag* yang kecil dan murah. *Tag* tersebut berisi *transponder* dengan suatu chip memori digital yang di dalamnya berisi sebuah kode produk yang sifatnya unik. Sebaliknya, *interrogator* suatu antenna yang berisi *transceiver* dan *decoder*, memancarkan sinyal yang bisa mengaktifkan RFID *tag* sehingga dia akan membaca dan menulis data ke dalamnya. Ketika suatu RFID *tag* melewati suatu zona *electromagnetis*, maka dia akan mendeteksi sinyal aktivasi yang dipancarkan oleh *reader*. *Reader* akan *decoder* data yang ada pada *tag* dan kemudian data tadi akan diproses oleh computer.



Gambar 2.13. Diagram RFID

Tabel 2.6. *Bandwith* Frekuensi RFID

No	Frekuensi RFID	Jenis Frekuensi	Manfaat
1	125KHz-134KHz	<i>Low Frequency</i>	Menandai hewan
2	13.56MHz	<i>High Frequency</i>	<i>Smart card</i>
3	860MHz-930MHz	<i>Ultra High Frequency</i>	Membuka otomatis bagasi, identifikasi suatu barang.

1. Band LF (*Low Frequency*)

Frekuensi kerja RFID pada band LF ini terletak pada range frekuensi 125KHz – 134KHz. RFID dengan frekuensi kerja pada band LF dengan range frekuensi tersebut sering digunakan untuk keperluan penelitian *tracking* binatang dan *tracking* pengiriman suatu aset.

2. Band HF(*High Frequency*)

Frekuensi kerja RFID pada band HF terletak pada frekuensi 13,56 MHz. RFID dengan frekuensi kerja 13,56 ini digunakan di mana media data rate (TAG RFID) dan pembaca RFID (*RFID Reader*) berjarak sekitar 1,5 meter. RFID dengan frekuensi ini juga memiliki keuntungan karena tidak mengalami gangguan dari keberadaan air atau logam.

3. Band UHF (*Ultra High-Frequency*)

Range frekuensi RFID pada band UHF terletak pada range frekuensi 850 MHz – 950 MHz dan 2,4 GHz. RFID dengan frekuensi kerja pada band ISM UHF ini memiliki kecepatan pembacaan yang tinggi.

2.6 Solenoid Door Lock

Solenoid adalah actuator yang mampu melakukan gerakan linier yaitu gerakan lurus menarik atau mendorong. Solenoid DC dapat bekerja secara elektro

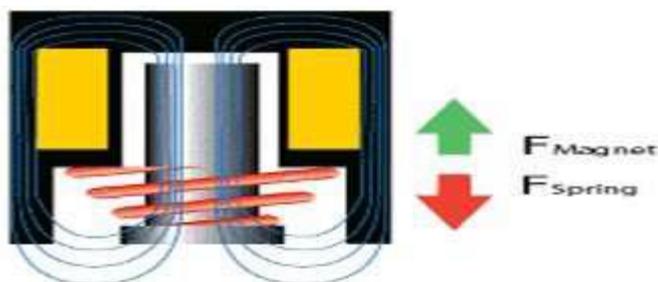
mekanis dengan memberikan sumber tegangan, maka solenoid dapat menghasilkan gaya yang linier (Budiharto Widodo, 2006).

Solenoid Door Lock ini adalah salah satu solenoid pengunci otomatis yang difungsikan khusus sebagai untuk mengunci pintu. Solenoid door lock ini membutuhkan tegangan supply 9V-12V, system kerja solenoid pengunci pintu ini adalah NC (*Normally Case*). Katup solenoid akan tertarik jika ada tegangan dan sebaliknya katup solenoid akan memanjang jika tidak ada tegangan. Tampilan Solenoid seperti gambar 2.16 dibawah ini.



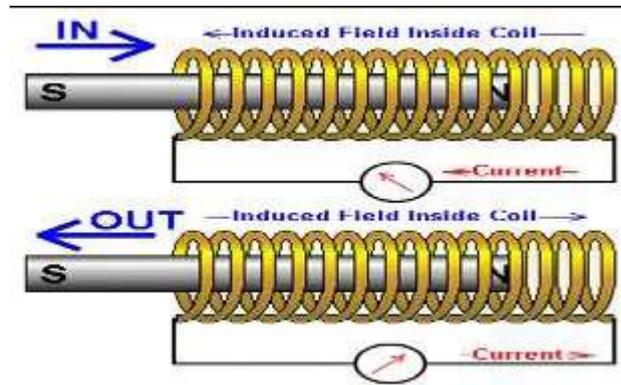
Gambar 2.14. Solenoid Door Lock

Pada Solenoid *Door Lock* memiliki kumparan yang teradapat pada intibesi. Ketika Arus listrik melalui kumparan ini, maka medan magnet yang akan menghasilkan energi sehingga dapat menarik intibesi. Poros dalam Solenoid Door Lock adalah inti besi berbentuk silinder yang disebut plunger. Medan magnet dapat membuat plunger untuk menarik atau repelling. Ketika medan magnet dimatikan, pegas kembali pada keadaan semula. Cara kerja Solenoid Door Lock dijelaskan pada gambar 2.17. dibawah.



Gambar 2.15. Cara Kerja Solenoid Door Lock

Pergerakan Solenoid Door Lock juga ditampilkan seperti gambar 2.12. dibawah, yakni pada saat lilitan arus teraliri maka inti besi akan bergerak. Gerakan pada inti besi, mengikuti dari arah arus pada lilitan (Budiharto Widodo, 2006).



Gambar 2.16. Pergerakan Solenoid Door Lock

2.7 Relay

Relay merupakan komponen output yang paling sering digunakan pada beberapa peralatan elektronika dan diberbagai bidang lainnya. Relay berfungsi untuk menghubungkan atau memutuskan aliran arus listrik yang dikontrol dengan memberikan tegangan dan arus tertentu pada koilnya (Setiawan, 21:2011).

Ada 2 macam relay berdasarkan tegangan untuk menggerakkan koilnya, yaitu AC dan DC. Ada berbagai macam relay berdasarkan *pole*-nya. Pada perancangan kali ini dipakai *Single Pole Double Throw (SPDT)* dan *Double Pole Double Throw (DPDT)* yang berfungsi untuk menghubungkan dan memutus arus untuk menggerakkan peralatan diluar rangkaian. Bentuk fisik relay seperti gambar 2.6 dibawah ini.



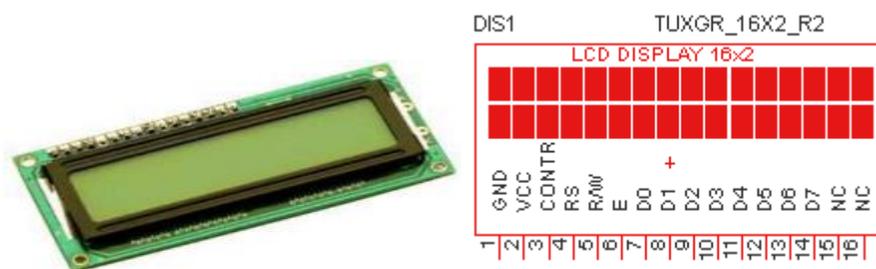
Gambar 2.19. Bentuk fisik Relay

Pada dasarnya relay adalah sebuah kumparan yang diari arus listrik sehingga kumparan mempunyai sifat sebagai magnet. Magnet sementara tersebut digunakan untuk menggerakkan suatu system saklar yang terbuat dari logam sehingga pada saat relay diari arus listrik maka kumparan akan terjadi kemagnetan dan menarik logam tersebut, saat arus listrik diputus maka logam akan kembali pada posisi semula.

2.8 *Liquid Crystal Display (LCD)*

Liquid Crystal Display (LCD) adalah satu bagian dari modul peraga yang menampilkan karakter yang diinginkan. Untuk dapat mengontrol tampilan ini diperlukan karakter generator yaitu bentuk-bentuk karakter yang dapat ditampilkan, urutan dan posisi dari karakter yang akan ditampilkan dan pergantian ke display harus disimpan dan digabungkan di RAM. (Muarofah, 2011).

LCD merupakan output yang akan menampilkan suatu tulisan, LCD merupakan output yang akan menampilkan suatu tulisan, LCD modul menerima kode-kode karakter (8 *bit* per karakter) dari suatu mikroprosesor atau komputer. Gambar LCD dan skema LCD 16x2 dapat dilihat pada gambar 2.14.



Gambar 2.18. Bentuk Fisik LCD dan Skema LCD 16x2

Tabel 2.7 Konfigurasi Pin pada LCD:

No	Nama Pin	Deskripsi	Port Mikrokontroler
1	GND	0V	PD0
2	VCC	+5V	PD1
3	VEE	Tegangan Kontras LCD	Keluaran Trimpot

4	RS	Register select, 0=register perintah, 1=register data	PD6
5	R/W	1=read, 0=write	Dihubungkan Ke Ground
6	E	Enable Clock LCD, logika 1 setiap kali pengiriman atau pembacaan data	PD7
7	D0	Data Bus 0	
8	D1	Data Bus 1	
9	D2	Data Bus 2	
10	D3	Data Bus 3	
11	D4	Data Bus 4	PC4
12	D5	Data Bus 5	PC5
13	D6	Data Bus 6	PC6
14	D7	Data Bus 7	PC7
15	Anoda	Tegangan negatif <i>backlight</i>	PC0
16	Katoda	Tegangan positif <i>backlight</i>	PC1

Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada tabel diskripsi, *interface* LCD merupakan sebuah parallel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 *nibble* data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa *clock* EN setiap nibblenya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus menset EN ke kondisi *high* “1” dan kemudian menset dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.

Saat jalur lainnya sudah siap, EN harus diset ke “0” dan tunggu beberapa saat (tergantung pada *datasheet* LCD), dan set EN kembali ke *high* “1”. Ketika jalur RS

berada dalam kondisi *low* “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi *high* atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan di layar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diset ke “1”. Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi *high* “1”, maka program akan melakukan *query* (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu *Get LCD status* (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diset ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara parallel baik 4-bit atau 8-bit merupakan 2 mode operasi primer. Untuk membuat sebuah aplikasi *interface* LCD, menentukan mode operasi merupakan hal yang paling penting.

Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4bit minimal hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroler dan LCD. Jika *bit* ini di set ($RS = 1$), maka *byte* pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di reset ($RS = 0$), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca.

2.9 Bahasa C

Lahirnya bahasa pemrograman diawali oleh terbentuknya bahasa *assembly* yang dikembangkan oleh IBM dalam tahun 1956-1963. Bahasa ini termasuk dalam bahasa tingkat rendah (*low level language*). Pada tahun 1957 sebuah tim yang dipimpin oleh John W. Backus berhasil mengembangkan sebuah bahasa pemrograman baru yang diarahkan untuk proses analisa numerik. Bahasa pemrograman tersebut dinamai dengan bahasa FORTRAN (*Formula Translation*). Setahun kemudian, yaitu pada 1958, para ilmuwan komputer dari Eropa dan

Amerika yang tergabung dalam sebuah komite menciptakan bahasa pemrograman baru yang lebih bersifat struktural dan dinamakan dengan bahasa ALGOL (*Algorithmic Language*). Kemudian pada tahun 1964, IBM kembali menciptakan bahasa pemrograman baru dengan nama PL/I (*Programming Language I*) yang lebih ditujukan untuk keperluan bisnis dan penelitian (Raharjo, 2006; 7).

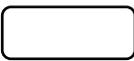
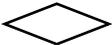
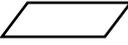
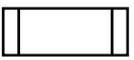
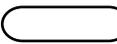
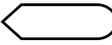
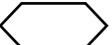
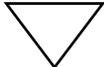
Tahun 1969 laboratorium Bell AT&T di Murray, New Jersey menggunakan bahasa *assembly* untuk mengembangkan sistem operasi Unix yang bertujuan untuk membuat program antar muka yang bersifat *programmer friendly*. Setelah Unix berjalan, lahirlah bahasa pemrograman baru yang ditulis oleh Martin Richard dengan nama bahasa BCPL (*Basic Combined Programin*). Kemudian pada tahun 1970, seorang pengembang sistem dari laboratorium tersebut yang bernama Ken Thompson membuat bahasa B yang akan digunakan untuk menulis ulang sistem operasi Unix. Nama 'B' ini konon diambil dari huruf pertama dalam kata BCPL. Karena alasan bahwa bahasa B masih terkesan lambat, maka pada tahun 1971 seorang pengembang sistem bernama Dennis Ritchie, yang juga bekerja di laboratorium yang sama, menciptakan bahasa baru dengan nama C yang bertujuan untuk menulis ulang dan menutupi kelemahan-kelemahan yang ada pada sistem operasi Unix sebelumnya. Menurut sumber yang ada, nama 'C' ini juga konon diambil dari huruf kedua dalam kata BCPL.

Sejak itu bahasa C terus digunakan untuk memelihara sistem operasi Unix, Sampai akhirnya pada tahun 90-an, bahasa C ini digunakan untuk mengembangkan sistem operasi Windows dan sekarang ini digunakan untuk mengembangkan sistem operasi Linux. Selain untuk menulis program yang merupakan *embedded system*, di kalangan industri hiburan, bahasa C juga banyak digunakan dalam mengembangkan perangkat lunak untuk permainan (*game*) menerima hal-hal inilah yang menyebabkan bahasa C menjadi bahasa yang sangat populer dikalangan industri perangkat lunak.

2.10 Flowchart

Badan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. (Suryantara, 2009)

Tabel 2.8. Simbol *Flowchart*

No.	Simbol	Nama Simbol	Keterangan
1.		<i>Alternate Process</i>	Menyatakan segala jenis operasi yang diproses dengan menggunakan mesin yang memiliki keyboard
2.		<i>Decision</i>	suatu penyelesaian kondisi dalam program
3.		<i>Data</i>	Mewakilik data <i>input</i> atau <i>output</i>
4.		<i>Predefined Process</i>	Suatu operasi yang rinciannya di tunjukkan di tempat lain
5.		<i>Document</i>	Document <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer
6.		<i>Terminator</i>	Untuk menunjukkan awal dan akhir dari suatu proses
7.		<i>Process</i>	Kegiatan proses dari operasi program komputer
8.		<i>Manual Input</i>	<i>Input</i> yang menggunakan <i>online keyboard</i>
9.		<i>Conector</i>	Penghubung ke halaman yang masih sama
10.		<i>Off-Page Connector</i>	Penghubung ke halaman lain
11.		<i>Display</i>	<i>Output</i> yang ditampilkan di monitor
12.		<i>Delay</i>	Menunjukkan penundaan
13.		<i>Preparation</i>	Memberi nilai awal suatu besaran
14.		<i>Manual Operation</i>	Pekerjaan manual
15.		<i>Card</i>	<i>Input</i> atau <i>output</i> yang menggunakan kartu
16.		<i>Punch Tape</i>	<i>Input</i> atau <i>output</i> menggunakan pita kertas berlubang
17.		<i>Merge</i>	Penggabungan atau penyimpanan beberapa proses atau informasi sebagai salah satu

18.		<i>Dirrect Access Storage</i>	<i>Input</i> atau <i>output</i> menggunakan drum magnetik
19.		<i>Magnetic Disk</i>	<i>Input</i> atau <i>output</i> menggunakan <i>hard disk</i>
20.		<i>Sequential Access Storage</i>	<i>Input</i> atau <i>output</i> menggunakan pita magnetik
21.		<i>Sort</i>	Proses pengurutan data di luar komputer
22.		<i>Stored Data</i>	<i>Input</i> atau <i>output</i> menggunakan <i>diskette</i>
23.		<i>Extract</i>	Proses dalam jalur paralel
24.		<i>Arrow</i>	Menyatakan jalan atau arus suatu proses
25.		<i>Summing Junction</i>	Untuk berkumpul beberapa cabang sebagai proses tunggal
26.		<i>Or</i>	Proses menyimpang dalam dua proses