

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Jurnal pertama berjudul : Prototipe Sistem Presensi Kelas Berbasis RFID dan Micro SD yang ditulis oleh Nurhadi Pranoto. Fakultas Teknik Muhammadiyah Surakarta menjelaskan bahwa berdasarkan pengujian dan analisis menunjukkan bahwa ketika saklar ditekan, maka LCD 16x2 akan menyala dan menampilkan tulisan jam kosong sebagai indikasi bahwa alat menyala, mahasiswa dapat melakukan absensi dengan mendekatkan RFID *tag* ke alat. Data yang tersimpan kedalam kartu *micro SD* dengan format .TXT. Kelemahan alat ini terletak pada cara untuk melihat info seputar absensi didalamnya, data hanya dapat dilihat setelah diambil secara manual dari kartu *micro SD* yang berformat .TXT.

Jurnal kedua berjudul : Perancangan Sistem Keamanan Hak Akses Pintu Menggunakan RFID yang ditulis oleh Puguh Gambiro, Aris Triwiyatno, dan Budi Setiyono dari Jurusan Teknik Elektro, Universitas Diponegoro Semarang menjelaskan bahwa berdasarkan pengujian dan analisis, *tag ID* memiliki hak akses berdasarkan golongan dan kemampuan akses di setiap *gate* dengan lampu *LED* sebagai indikatornya, *tag ID* dapat berjalan sesuai dengan ketentuan peraturan yang dibuat dan sesuai golongannya. Tag yang digunakan saat pengecekan adalah *smartcard* dan *wristband tag*. Sistem RFID ini dapat membaca kode yang terhubung dengan *database* yang menggunakan *database FireBird* dan sekaligus dapat menampilkannya dalam sebuah tampilan *software* menggunakan Delphi. Kelemahannya adalah menggunakan kabel *LAN (Local Area Network)*.

Dari cara kerja alat kedua jurnal diatas, maka akan dibuat suatu sistem untuk mengabsensi mahasiswa dengan mengambil beberapa fitur terdahulu. Untuk alat yang akan dibuat pada laporan ini akan menggunakan koneksi *internet* agar informasi dapat diakses melalui jaringan *internet*. Teknologi RFID pada laporan ini digunakan untuk absensi mahasiswa. Dengan sistem identifikasi RFID maka identitas mahasiswa yang mengabsen dikenali dengan kartu pengenalan, nantinya perekapan data absensi dapat dilakukan tanpa perlu melalui lembaran absen seperti

biasa karena data bisa dikirim melalui jaringan *internet*. Data absensi akan masuk kedalam *database* yang terhubung ke *web*. Modul alat ini dapat mengirim data ke server dimana data yang disimpan dapat diolah lebih lanjut melalui jaringan *internet* maupun lokal dari suatu perangkat (*PC* atau *Smartphone*) yang memiliki *database* sistem ini.

2.2 Absensi

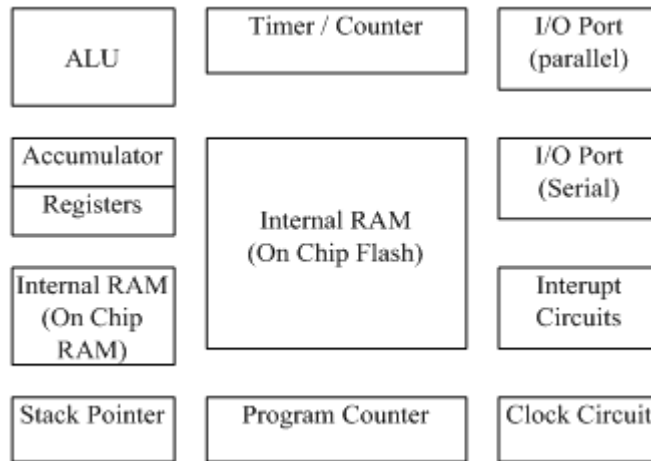
Absensi merupakan suatu aktivitas pelaporan dan pendataan kehadiran yang ada dalam sebuah institusi (Setiawan dan Kurniawan, 2015). Sistem absensi manual yang dilakukan di perguruan tinggi mengharuskan mahasiswa mengisi formulir absensi dengan tandatangan pada saat perkuliahan (Yusuf, 2016). Sistem ini memiliki beberapa kekurangan seperti adanya penitipan tandatangan, memerlukan banyak kertas dan tinta, dan membutuhkan ruang yang banyak sebagai tempat penyimpanan.

2.3 Definisi Mikrokontroller

Menurut Setiawan (2011:1) Mikrokontroller umumnya seperti komputer, mikrokontroller adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Suatu IC dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan untuk suatu kontroler sudah dikemas dalam satu keping, biasanya terdiri dari CPU (*Central Processing Unit*), RAM (*Random Access Memory*), EEPROM/EPROM/PROM/ROM, I/O, Serial & Parallel, *Timer*, *Interrupt Controller*.

Menurut Fauzi (2011:1) Mikrokontroller adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Berdasarkan definisi yang dikemukakan diatas dapat disimpulkan bahwa mikrokontroller adalah suatu IC yang didesain atau dibentuk dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan suatu kontroler sudah dikemas dalam satu keping, biasanya terdiri dari CPU (*Central Processing Unit*), RAM (*Random Access Memory*), EEPROM/EPROM/PROM/ROM, I/O, Serial & Parallel, *Timer*, *Interrupt*

Controller dan berfungsi sebagai pengontrol rangkaian elektronik serta umumnya dapat menyimpan program didalamnya.



Gambar 2.1 Blok Diagram Mikrokontroler.

Berikut adalah penjelasan dari bagian-bagian dari diagram blok mikrokontroler dari gambar 2.1 diatas :

1. Register

Register merupakan suatu tempat penyimpanan (variabel) bilangan bulat yang terdiri dari 8 atau 16 *bit*. Pada umumnya *register* memiliki jumlah yang banyak, masing-masing ada yang memiliki fungsi khusus dan ada pula yang memiliki fungsi atau kegunaan secara umum. Register yang memiliki fungsi secara khusus misalnya *register timer* yang berisi data penghitungan pulsa untuk *timer*, atau register pengatur mode operasi *counter* (penghitung pulsa). Sedangkan register yang memiliki fungsi umum digunakan untuk menyimpan data sementara yang diperlukan untuk proses penghitungan dan proses operasi mikrokontroler. *Register* dengan fungsi umum sangat dibutuhkan dalam sistem mikrokontroler karena mikrokontroler hanya mampu melakukan operasi aritmetik atau logika hanya pada satu atau dua *operand* saja, sehingga untuk operasi-operasi yang melibatkan banyak variabel harus dimanipulasi dengan menggunakan variabel-variabel register umum.

2. Accumulator

Merupakan salah satu register khusus yang berfungsi sebagai *operand* umum proses aritmetika dan logika.

3. Program Counter

Merupakan salah satu register khusus yang berfungsi sebagai pencacah/penghitung eksekusi program mikrokontroler.

4. ALU (*Arithmetic and Logic Unit*)

ALU memiliki kemampuan dalam mengerjakan proses-proses aritmatika (penjumlahan, pengurangan, perkalian, pembagian) dan operasi logika (misalnya AND, OR, XOR, NOT) terhadap bilangan bulat 8 atau 16 *bit*.

5. Clock Circuits

Mikrokontroler merupakan rangkaian logika sekuensial, dimana proses kerjanya berjalan melalui sinkronisasi *clock*. Oleh karena itu diperlukan *clock circuits* yang menyediakan *clock* untuk seluruh bagian rangkaian.

6. Internal ROM (*Read Only Memory*)

Merupakan memori penyimpanan data dimana data tersebut tidak dapat diubah atau dihapus (hanya dapat dibaca). ROM biasanya diisi dengan program untuk dijalankan oleh mikrokontroler segera setelah *power* dihidupkan. Data dalam ROM tidak dapat hilang meskipun *power* dimatikan.

7. Stack Pointer

Stack merupakan bagian dari RAM yang memiliki metode penyimpanan dan pengambilan data secara khusus. Data yang disimpan dan dibaca tidak dapat dilakukan dengan cara acak karena data yang dituliskan ke dalam *stack* yang berada pada urutan yang terakhir merupakan data yang pertama kali dibaca kembali. *Stack Pointer* berisi *offset* dimana posisi data *stack* yang terakhir masuk (atau yang pertama kali dapat diambil).

8. I/O (*Input/Output*) Ports

Merupakan sarana yang digunakan oleh mikrokontroler untuk mengakses peralatan-peralatan lain di luar sistem. I/O Port berupa pin-pin yang dapat berfungsi untuk mengeluarkan data digital ataupun sebagai masukan data eksternal.

9. Interrupt Circuits

Adalah rangkaian yang memiliki fungsi untuk mengendalikan sinyal-sinyal interupsi baik internal maupun eksternal. Adanya sinyal interupsi akan menghentikan eksekusi normal program mikrokontroler untuk selanjutnya menjalankan *sub*-program untuk melayani interupsi tersebut.

Diagram blok di atas tidak selalu sama untuk setiap jenis mikrokontroler. Beberapa mikrokontroler menyertakan rangkaian ADC (*Analog to Digital Converter*) di dalamnya, ada pula yang menyertakan port I/O serial disamping port I/O *parallel* yang sudah ada.

10. Internal RAM (*Random Acces Memory*)

Merupakan memori penyimpanan data dimana data tersebut dapat diubah atau dihapus. RAM biasanya berisi data-data *variable* dan *register*. Data yang tersimpan pada RAM bersifat *volatile* yaitu akan hilang bila catu daya yang terhubung padanya dimatikan.

2.4 Arduino

Menurut Sulaiman (2012:1), arduino merupakan *platform* yang terdiri dari *software* dan *hardware*. *Hardware* Arduino sama dengan *microcontroller* pada umumnya hanya pada arduino ditambahkan penamaan pin agar mudah diingat. *Software* Arduino merupakan *software open source* sehingga dapat di download secara gratis. *Software* ini digunakan untuk membuat dan memasukkan program ke dalam Arduino.

Menurut Sulaiman (2012:1), arduino adalah *kit* elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel.

Berdasarkan dua definisi yang dikemukakan diatas dapat disimpulkan bahwa arduino merupakan *kit* elektronik atau papan rangkaian elektronik yang didalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel serta *software* pemrograman yang berlisensi *open source*.

2.5 Hardware Arduino

Menurut Sulaiman (2012:1) Arduino merupakan platform *open source* baik secara *hardware* dan *software*. Arduino terdiri dari mikrokontroler megaAVR seperti ATmega8, ATmega168, ATmega328, ATmega1280, dan ATmega 2560 dengan menggunakan Kristal osilator 16 MHz, namun ada beberapa tipe Arduino yang menggunakan Kristal osilator 8 MHz. Catu daya yang dibutuhkan untuk mensupply minimum sistem Arduino cukup dengan tegangan 5 VDC.

Port arduino Atmega series terdiri dari 20 pin yang meliputi 14 pin I/O digital dengan 6 pin dapat berfungsi sebagai output PWM (*Pulse Width Modulation*) dan 6 pin I/O analog. Kelebihan Arduino adalah tidak membutuhkan flash programmer external karena di dalam chip microcontroller Arduino telah diisi dengan *bootloader* yang membuat proses *upload* menjadi lebih sederhana. Untuk koneksi terhadap komputer dapat menggunakan RS232 to TTL *Converter* atau menggunakan Chip USB ke Serial *converter* seperti FTDI FT232.

2.6 Mikrokontroler Arduino Wemos

Mikrokontroler Wemos adalah sebuah mikrokontroler pengembangan berbasis modul mikrokontroler ESP 8266. Mikrokontroler Wemos dibuat sebagai solusi dari mahalnya semua sistem *wireless* berbasis mikrokontroler lainnya. Mikrokontroler Wemos ini memiliki kemampuan untuk menyediakan konektivitas WiFi serta memory yang digunakan cukup besar yaitu 4MB. (Jainrk, 2018).



Gambar 2.2 Mikrokontroler Wemos.

2.6.1 Chipset Mikrokontroler pada Mikrokontroler Wemos

Pada mikrokontroler Wemos memiliki 2 buah *chipset* yang digunakan sebagai otak kerja *platform* tersebut. Beberapa *chipset* pada mikrokontroler ini adalah :

1. Chipset ESP8266

ESP8266 adalah sebuah chip *microcontroller* yang memiliki fitur Wi-Fi yang mendukung stack TCP / IP. Diproduksi oleh produsen Cina yang berbasis di Shanghai, Espressif. Pada Agustus 2014 AI-Thinker membuat modul ESP-01 dengan menggunakan lisensi oleh Espressif. Modul ini memungkinkan *microcontroller* untuk terhubung dengan jaringan WiFi dan membuat koneksi TCP / IP. Dengan *clock* 80MHZ dengan 4 MB Eksternal RAM, mendukung format IEEE 802.11 b/g/n, enkripsi WEP dan WPA sehingga aman digunakan. *Chipset* ini memiliki 16 GPIO pin yang bekerja pada 3.3 Volt, 1 pin ADC 10 bit.

2. Chipset CH340

CH340 adalah sebuah *chipset* yang mengubah USB menjadi *serial interface*. Sebagai contohnya adalah aplikasi USB *converter to IrDA* atau aplikasi USB *converter to Printer*. Dalam mode *serial interface*, CH340 mengirimkan sinyal penghubung yang umum digunakan pada MODEM. CH340 digunakan untuk memperbesar *asynchronus serial interface* komputer atau mengubah perangkat serial interface umum untuk berhubungan dengan bus USB.

2.6.2 Sketch

Sketch adalah lembar kerja pada sistem arduino yang digunakan untuk menulis *listing* program, mengeditnya, meng-*compile* dan kemudian mengupload ke dalam mikrokontroler tersebut. *Sketch* Arduino terdiri dari bagian-bagian seperti *comments*, *setup* (), dan *loop* (). Dibawah ini akan dijelaskan secara lebih detail mengenai bagian-bagian tersebut.

2.6.3 Comments

Biasanya komentar digunakan untuk memudahkan bagi orang-orang membaca kode yang telah ditulis oleh *engineer*, untuk menjelaskan tujuan dari

dibuatnya program ini, cara kerjanya, atau mengapa program tersebut ditulis seperti itu. *Comment* pada *sketch* sangat disarankan, salah satu fungsinya adalah untuk membantu ketika adanya kode yang ingin diperbaiki. Hal ini juga dapat membantu orang lain untuk belajar atau memodifikasi kode yang sudah berjalan. Sebagai contoh seperti *comment* diatas disebut *single-line comments* ini dimulai dengan “//” dan lajut hingga dibaris akhir. *Comment* diatas digunakan untuk menjelaskan mengenai penempatan suatu kode.

2.6.4 Fungsi *Setup* ()

Ada beberapa fungsi khusus yang merupakan bagian dari *sketch* yaitu “*Setup* ()”. Fungsi *Setup* () dipanggil sekali, yaitu ketika sketsa dimulai. Fungsi ini merupakan tempat yang baik untuk melakukan pengaturan-pengaturan seperti :

- Pengaturan mode *output* pada pin *digital*.
- Inialisai *library* mikrokontroler arduino.
- Inialisasi variabel, dan lain-lain.
-

Fungsi *setup* hanya akan berjalan sekali, setelah setiap *power up* atau setelah tombol *reset* pada rangkaian modul arduino ditekan. Berikut adalah contoh dari fungsi *setup* ():

```
Void setup ()
{
  Serial, begin(9600);
  pinMode(buttonPin, INPUT);
}
```

2.6.5 Fungsi *Loop* ()

Fungsi *Loop* () adalah fungsi utama dalam *sketch* arduino ini. Fungsi ini dipanggil berulang kali oleh modul mikrokontroler untuk menjalankan program yang telah tersimpan didalamnya. Berikut adalah contoh penggunaan dari fungsi *Loop* () :


```

Void loop ()
{
If (digital Read(buttonPin) == HIGH)
serialWrite('H');
else
serialWrite('L');
delay(1000);
}

```

Fungsi *Loop* () sesuai dengan namanya, melakukan perulangan setiap *listing* program yang dituliskan, yang pada saat tertentu variabel dari program anda telah berubah sehingga sistem merespon dan menghasilkan *output* baru yang berbeda dengan hasil *output* pertama.

2.6.6 Fitur-Fitur Mikrokontroler Wemos

Berikut ini adalah fitur-fitur dari perangkat keras mikrokontroler Wemos. Perangkat keras ini dapat diprogram dengan mudah pada *sketch*. Berikut penjelasan beberapa diantaranya :

1. Karakteristik Pin *Digital Input*

Secara *default* pengaturan port *digital* adalah pengaturan untuk port masukan, sehingga mereka tidak perlu secara eksplisit dinyatakan sebagai *input* dengan pin mode (). Pin dikonfigurasi sebagai *input* sehingga pin tersebut berada dalam keadaan impedansi tinggi. Salah satu penjelasannya adalah pin input akan mengambil daya yang sangat kecil sekali pada rangkaian ketika dalam kondisi pengambilan sampel. Hal ini berarti bahwa hanya sangat sedikit arus yang digunakan untuk memindahkan kondisi input tersebut dari keadaan satu ke keadaan yang lain. Sehingga hal ini dapat membuat pin berguna untuk melakukan tugas-tugas seperti membaca sensor sentuh kapasitif, LED atau, sensor analog.

2. Karakteristik Pin *Digital* ketika menjadi *Output*

Apabila pin *digital* ini dikonfigurasi sebagai *output* dengan pin mode (), maka pin ini akan berada dalam keadaan impedansi rendah. Hal ini berarti bahwa

mereka dapat menyediakan sebgaiian besar arus ke rangkaian lainnya. Hubungan pendek pada pin dapat merusak atau menghancurkan *transistor output* pada pin, atau merusak *chip* secara keseluruhan. Seringkali akan menghasilkan sebuah pin mati, akan tetapi *chip* yang masih tersisa berfungsi secara memadai. Maka baik adanya untuk menghubungkan pin *output* ke perangkat lain dengan *resistor*.

3. Pin Analog

Pin analog pada mikrokontroler memiliki 10 bit resolusi dengan nilai maksimum 3.2 Volt. Pin analog ini dapat dikonfigurasi dan digunakan seperti pin digital. Berikut karakteristik dari pin analog :

A. A/D Converter

Chips Atmega digunakan pada Arduino memiliki 1 saluran ADC (*analog to-digital converter*). ADC tersebut memiliki 10 bit resolusi dari 0 ke 1023. Sedangkan fungsi utama dari pin analog pada mikrokontroler wemos adalah untuk membaca sensor analog. Pin analog juga memiliki semua fungsi *General Purpose input / output* (GPIO) pin (sama dengan pin digital 0-13).

B. Pemetaan Pin

Pin analog dapat digunakan sama seperti pin *digital*, menggunakan penamaan A0 (untuk *input* analog 0), A1, dll. Sebagai contoh, kode berikut digunakan untuk mengatur 0 pin analog ke *output*, dan mengaturnya berlogika “*High*” :

Pin mode (A0, OUTPUT);

Digital Write(A0, HIGH);

C. Pull Up Resistor

Pin analog juga memiliki *resistor pullup*, yang bekerja sama seperti *resistor pullup* pada pin *digital*. Mereka diaktifkan dengan mengeluarkan perintah seperti dibawah ini :

Digital Write(A0, HIGH); // set pullup on analog pin 0.

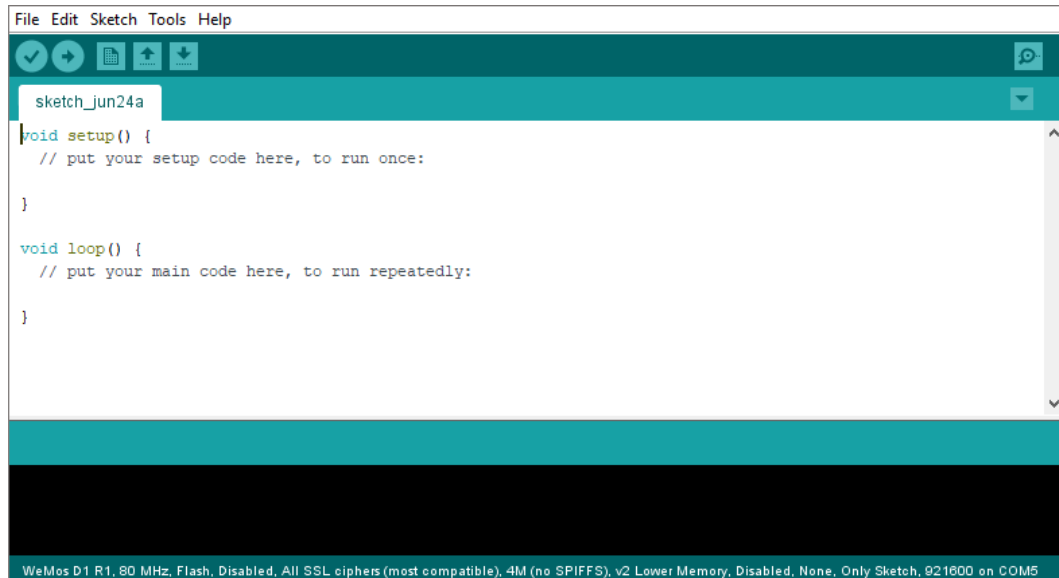
Namun mengatur *resistor pullup* akan mempengaruhi nilai yang akan diambil oleh *analog read()*. Hal tersebut dikarenakan perintah *analog read* tidak akan bekerja dengan benar jika pin sebelumnya digunakan sebagai *output*, akan tetapi apabila hal ini terjadi maka pin tersebut harus diatur kembali menjadi masukan sebelum menggunakan perintah *analog read*. Hal yang sama pula harus diterapkan jika pin telah diatur menjadi logika “*High*” sebagai *output*, *resistor pullup* harus diatur ketika beralih kembali ke *input*.

2.7 Arduino IDE

IDE merupakan kependekan dari *Integrated Development Enviroment*, atau secara sederhana merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui *sintaks* pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

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.





Program yang ditulis dengan menggunakan Arduino Software (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *.ino*. Teks editor pada Arduino Software memiliki fitur” seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan dalam menulis kode program. (Jainrk, 2018).





Gambar 2.3 Tampilan Software Arduino IDE

Pada aplikasi terdapat tools yang digunakan saat berjalannya aplikasi tersebut. dan dapat dilihat pada tabel dibawah ini :

Tabel 2.1 Keterangan *Tools* pada Aplikasi Arduino IDE.

| Ikon | Nama | Keterangan |
|---|---------------|---|
|  | <i>Verify</i> | Berfungsi untuk melakukan <i>checking</i> atau pengecekan kode yang dibuat apakah sudah sesuai dengan kaidah pemrograman yang ada. |
|  | <i>Upload</i> | Berfungsi untuk melakukan kompilasi program atau kode yang dibuat menjadi bahasa yang dapat dipahami oleh Arduino. |
|  | <i>New</i> | Berfungsi untuk membuat <i>Sketch</i> baru |
|  | <i>Open</i> | Berfungsi untuk membuka <i>sketch</i> yang pernah kamu buat dan membuka kembali untuk dilakukan <i>editing</i> atau sekedar <i>mengupload</i> ulang ke Arduino. |

| | | |
|---|-----------------------|---|
|  | <i>Save</i> | Berfungsi untuk menyimpan <i>Sketch</i> yang telah dibuat. |
|  | <i>Serial Monitor</i> | Berfungsi untuk membuka serial monitor. Serial monitor disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan sketch pada port serialnya. Serial Monitor ini berguna ketika membuat program atau melakukan <i>debugging</i> tanpa menggunakan LCD pada Arduino. Serial monitor ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan <i>error</i> . |

2.8 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* artinya 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 statemen-statemen program C. Namun pada kenyataannya, suatu fungsi bisa saja tidak mengandung pernyataan sama sekali. Walaupun fungsi tidak memiliki pernyataan, kurung kurawal haruslah tetap ada. Sebab kurung kurawal mengisyaratkan awal dan akhir definisi fungsi. (Heryanto, 2008).

2.9 Pengertian Database

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer serta digunakan perangkat lunak tertentu untuk memanipulasinya. Adalah salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan data bagi para pemakai.

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan. Merupakan aspek yang sangat penting dalam sistem

informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Hal ini menjadi penting karena dapat mengorganisasi, menghindari duplikasi, dan mengantisipasi hubungan antar data yang tidak jelas dan juga update yang rumit. (Hartono , 2005 : 17)

2.10 MySQL

MySQL merupakan software yang tergolong *database server* dan bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini di lengkapi oleh *source code* (kode yang di pakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode dapat di jalankan secara langsung di dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis. Hal lainnya adalah MySQL juga bersifat *multiplatform*. MySQL dapat dijalankan pada berbagai sistem operasi. (Abdul Kadir, 2009:15).

Beberapa keistimewaan dari MySQL, antara lain :

1. Portabilitas, artinya stabil pada berbagai sistem operasi seperti Windows, Linux, Mac OS.
2. Didistribusikan secara *open source* atau bebas di bawah lisensi dari GPL.
3. *Multiuser*. Dapat digunakan oleh beberapa user dalam waktu secara bersamaan tanpa mengalami masalah.
4. *Performance tuning*. Mempunyai kecepatan dalam pemrosesan query dibandingkan mesin *database* lainnya.
5. Konektivitas. Melakukan koneksi dengan klien dengan menggunakan protokol TCP/IP, Unix socket, serta *Named Pipes*.
6. Dapat menangani perubahan struktur tabel dengan perintah ALTER TABEL dibandingkan PostgreSQL maupun Oracle.
7. Mendukung penggunaan tipe kolom yang kompleks seperti *signed/unsigned integer, float, double, char, date, timestamp*, dan lain sebagainya.
8. Mempunyai beberapa lapisan keamanan seperti *level subnetmask*, nama *host*, izin akses *user* dengan sistem perijinan yang mendetail serta terenkripsi.

9. Mempunyai *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*). Hal ini akan memudahkan pengguna atau *programmer* dalam melakukan pengolahan data.

2.11 Pengertian Bagan Alir (*Flowchart*)

Flowchart adalah urutan proses kegiatan yang digambarkan dalam bentuk simbol. *Flowchart* (bagan alir) juga didefinisikan sebagai diagram yang menyatakan aliran proses dengan menggunakan anotasi- anotasi semisal persegi, panah, oval, wajik dll.

Melalui *flowchart*, kita bisa melihat langkah-langkah proses secara mendetail, lengkap dengan aktivitas yang terjadi. *Flowchart* banyak dipergunakan sebagai alat komunikasi dan dokumentasi. (Salamadian, 2017)

Menurut Hartono (2005 : 795), *flowchart* atau diagram alir merupakan bagan(*chart*) yang menunjukkan hasil (*flow*) dalam program atau prosedur sistem secara logika. Digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

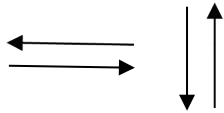
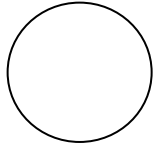
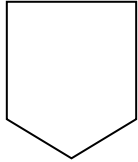
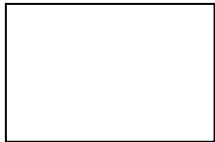
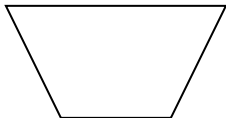
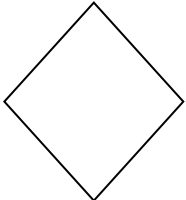
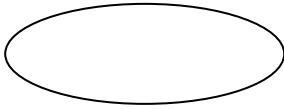
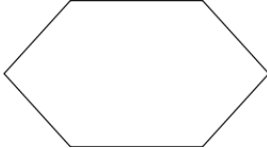
Simbol-simbol yang di pakai dalam *flowchart* dibagi menjadi 3 kelompok:


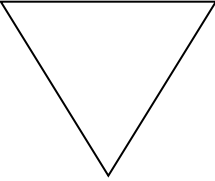

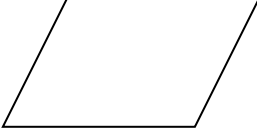
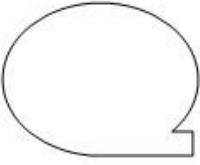


- 1) *Flow direction symbols*. Digunakan untuk menghubungkan simbol satu dengan yang lain, Disebut juga *connecting line*.
- 2) *Processing symbols*. Menunjukkan jenis operasi pengolahan dalam suatu proses/prosedur.
- 3) *Input / Output symbols*. Menampilkan jenis peralatan yang digunakan sebagai media *input* atau *output*.

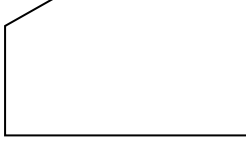
Berikut dibawah ini adalah simbol-simbol yang digunakan dalam *flowchart* disertai dengan keterangan fungsinya :

Table 2.2 Tabel *Flow Symbols*.

| SIMBOL | KETERANGAN |
|--------|------------|
|--------|------------|

| | |
|---|---|
|  | <p>Simbol arus / <i>flow</i>, menyatakan jalannya arus suatu proses.</p> |
|  | <p>Simbol <i>connector</i>, menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.</p> |
|  | <p>Simbol <i>offline connector</i>, menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.</p> |
|  | <p>Simbol proses, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.</p> |
|  | <p>Simbol <i>manual</i>, menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.</p> |
|  | <p>Simbol <i>decision</i>, yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak</p> |
|  | <p>Simbol <i>terminal</i>, menyatakan permulaan atau akhir suatu program.</p> |
|  | <p>Simbol <i>predefined process</i>, menyatakan persediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p> |

| | |
|---|---|
|  | <p>Simbol <i>keying operation</i>, menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>.</p> |
|  | <p>Simbol <i>offline-storage</i>, menunjukkan bahwa data dalam simbol ini akan disimpan ke dalam suatu media tertentu.</p> |
|  | <p>Simbol <i>manual input</i>, menyatakan data secara manual dengan menggunakan <i>online keyboard</i>.</p> |
|  | <p>Simbol <i>input / output</i>, menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.</p> |
|  | <p>Simbol <i>magnetic tape</i>, menyatakan input berasal dari pita magnetis atau output tersimpan ke dalam pita magnetis.</p> |
|  | <p>Simbol <i>disk storage</i>, menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> tersimpan ke dalam <i>disk</i>.</p> |
|  | <p>Simbol <i>document</i>, mencetak keluaran dalam bentuk dokumen (memulai <i>printer</i>).</p> |

| | |
|---|--|
|  | <p>Simbol <i>punched card</i>, menyatakan input berasal dari kartu atau output ditulis ke kartu.</p> |
|---|--|

(Sumber : Indrajani, 2015)

2.12 RFID

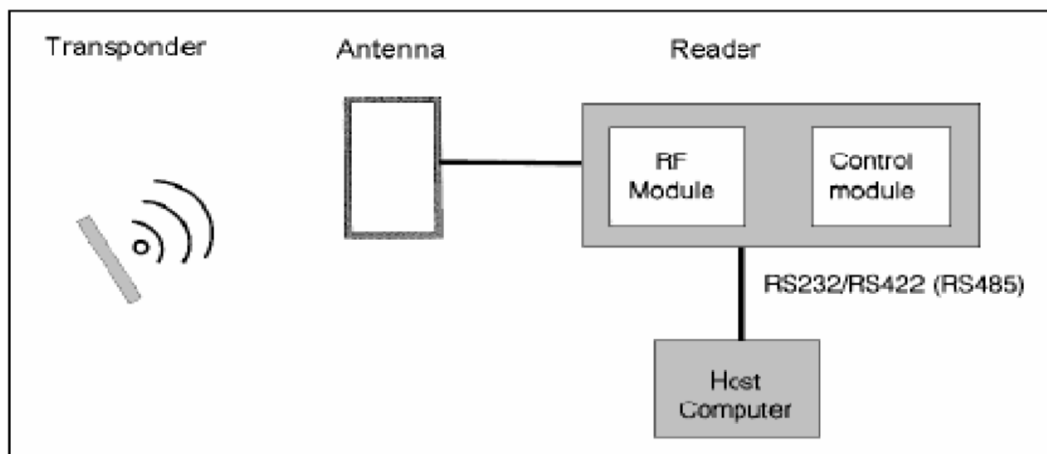
RFID adalah sebuah teknologi yang menggunakan frekuensi radio untuk mengidentifikasi suatu barang atau manusia. (Erwin, 2004 : 7). Menurut Henlia (2006 : 1), sejarah perkembangan *radio frequency identification* dimulai sejak tahun 1920, tetapi berkembang menjadi *IFF (Identification Friend or Foe) transponder* pada tahun 1939. Yang waktu itu berfungsi sebagai alat identifikasi pesawat musuh, dipakai oleh militer Inggris pada perang dunia II. Sejak tahun 1945 beberapa orang berfikir bahwa perangkat pertama RFID ditemukan oleh Leon Theremin sebagai suatu *tool spionase* untuk pemerintahan Rusia.

RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah *devais* kecil yang disebut *tag* atau *transponder (Transmitter + Responder)*. *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari *device* yang *kompatibel*, yaitu pembaca RFID (*RFID Reader*).

RFID adalah teknologi identifikasi yang *fleksibel*, mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID dapat disediakan dalam *device* yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

Pada sistem RFID umumnya, *tag* atau *transponder* ditempelkan pada suatu objek. Setiap tag membawa dapat membawa informasi yang unik, di antaranya: *serial number*, *model*, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika *tag* ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, *tag* akan mentransmisikan informasi yang ada pada *tag* kepada pembaca RFID, sehingga proses identifikasi objek dapat dilakukan.

Menurut Kania (2011 : 16), sistem RFID terbagi menjadi 3 komponen, yaitu: RFID *Tag*, RFID *Terminal Reader*, dan *Middleware*. Sedangkan untuk jenisnya RFID terbagi, berdasarkan frekuensi, berdasarkan sumber energi, dan berdasarkan bentuk. Adapun kelebihan dari sistem RFID tersebut adalah sistem *inventori* berkecepatan tinggi, proses sirkulasi yang cepat, penanganan buku-buku secara otomatis.



Gambar 2.4 Komponen Utama Sistem RFID

RFID mampu membaca suatu objek data dengan ukuran tertentu tanpa melalui kontak langsung (*contactless*) dan tidak harus sejajar dengan objek yang dibaca, selain dapat menyimpan informasi pada tag RFID sesuai dengan kapasitasnya penyimpanan.

Menurut Erwin (2004:07), bahwa RFID merupakan proses identifikasi seseorang atau objek dengan menggunakan proses transmisi radio. Sedangkan menurut Supriyono (2010:21) bahwa RFID (*Radio Frequency Identification*) adalah teknologi identifikasi berbasis gelombang, teknologi yang mampu mengidentifikasi berbagai objek secara tepat tanpa memerlukan kontak langsung.

Menurut Maryono Identifikasi dengan frekuensi radio adalah teknologi untuk mengidentifikasi seseorang atau objek benda menggunakan transmisi frekuensi radio, khususnya 125kHz, 13.65Mhz atau 800-900MHz.

RFID menggunakan komunikasi gelombang radio untuk secara unik mengidentifikasi objek atau seseorang. Hal ini merupakan teknologi pengumpulan data otomatis yang tercepat dalam perkembangannya. Teknologi tersebut menciptakan cara otomatis untuk mengumpulkan informasi suatu produk, tempat, waktu, atau transaksi dengan cepat, mudah tanpa *human error*. RFID menyediakan hubungan ke data dengan jarak tertentu tanpa harus melihat secara langsung, dan tidak terpengaruh lingkungan yang berbahaya seperti halnya *barcode*. Identifikasi RFID bukan sekedar kode identifikasi, sebagai pembawa data, dapat di tulis dan diperbarui data di dalamnya dalam keadaan bergerak.

Terdapat beberapa pengertian RFID menurut Supriyono yaitu :

1. RFID (*Radio Frequency Identification*) adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut label RFID atau *transponder (tag)* untuk menyimpan dan mengambil data jarak jauh.
2. Label atau *transponder (tag)* adalah sebuah benda yang bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio. Label RFID terdiri atas *microchip* silikon dan antena.

Berdasarkan catu dayanya, *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 rangkaiannya akan semakin kompleks dan ukurannya akan semakin besar.
2. *Tag* Pasif yaitu *tag* yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFId. Rangkaiannya 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. 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*.

2.12.1 Bagian Tag RFID

1. Mikroprosesor

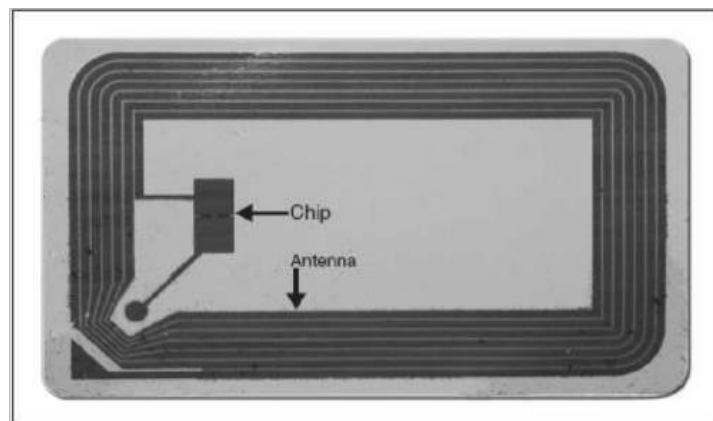
Mikroprosesor adalah chip yang terletak dalam sebuah RFID *tag* yang berfungsi sebagai penyimpan data.

2. *Metal Coil*

Metal Coil terbuat dari kawat alumunium yang berfungsi sebagai antena yang dapat beroperasi pada frekuensi 13,56MHz. 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 2.5 Tag RFID

2.12.2 RFID Reader RC522

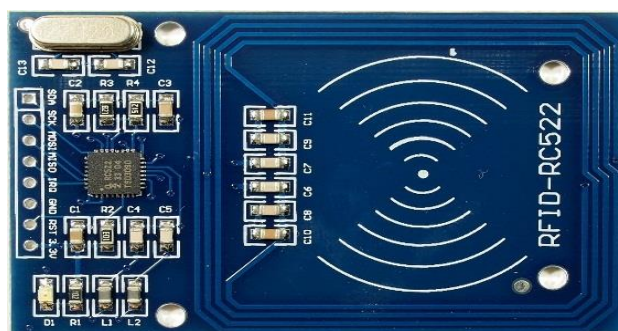
RFID reader RC522 merupakan reader RFID yang mampu melakukan proses membaca *read* dan bekerja pada frekuensi 13.56 MHz. Tag RFID yang kompatibel dengan modul RFID ini adalah tag jenis pasif. RFID reader RC522 memiliki rantang baca kurang lebih 3 kaki.

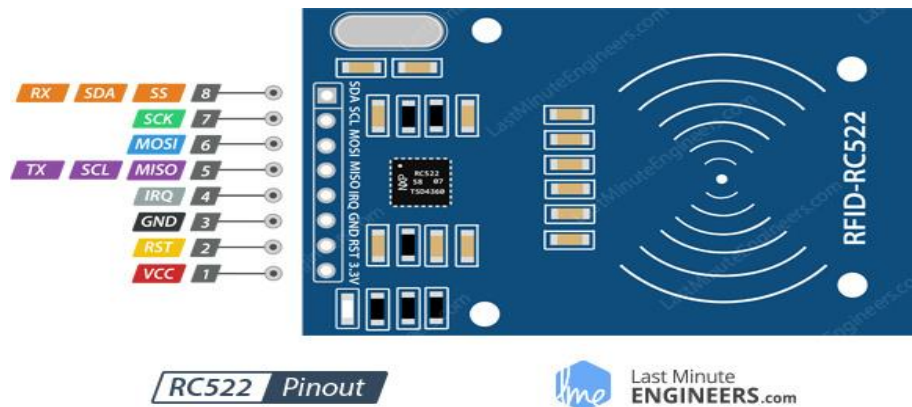
Spesifikasi dari modul ini diantaranya :

- *Working current* : 13—26mA/ DC 3.3V
- *Standby current* : 10-13mA/DC 3.3V
- *Sleeping current* : <80uA
- *Peak current* : <30mA
- Frekuensi kerja : 13.56MHz
- Protokol : SPI
- Suhu Kerja : -20 – 80⁰C
- Suhu Penyimpanan : -40 – 85⁰C
- *Max SPI speed* : 10Mbit/s
- Kecepatan komunikasi data hingga 10Mbit/s.

Tabel 2.3 Konfigurasi Pin Modul Reader MFRC522 RFID.

| NO | Pin |
|----|------|
| 1. | VCC |
| 2. | RST |
| 3. | GND |
| 4. | IRQ |
| 5. | MISO |
| 6. | MOSI |
| 7. | SCK |
| 8. | SDA |





Gambar 2.6 Modul dan Konfigurasi Pin RFID RC-522

Modul RFID *reader* ini berfungsi untuk membaca data atau nomor *ID* pada RFID tag 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 ekstrim / *hars environment*, misal temperatur atau tekanan yang tinggi.
3. Fleksibel dalam pemasangan, dengan *encapsulation* yang sesuai, RFID *Tag* dapat dipasang secara fleksibel dan bervariasi pada item, misalnya *Tag* yang dibenamkan pada ban traktor, *Tag* yang di *clamped* pada *body* mobil.
4. *Reusable*, dimana RFID *Tag* memiliki *lifetime* 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*. (Henlia, 2006).

2.13 Catu Daya

Pengertian Catu Daya adalah bagian dari setiap perangkat elektronika yang berfungsi sebagai sumber tenaga. Catu daya sebagai sumber tenaga dapat berasal dari : baterai , *accu* , *solar cell* dan adaptor. Komponen ini akan mencatu tegangan sesuai dengan tegangan yang diperlukan oleh rangkaian elektronika. Jadi dapat disimpulkan bahwa, catu daya merupakan bagian yang terpenting dalam elektronika yang berfungsi sebagai sumber tenaga listrik misalnya pada baterai atau *accu*, dan merupakan suatu rangkain elektronik yang mengubah arus listrik bolak balik menjadi arus listrik searah sederhana yang baik. (Rohmatullah, 2015).

Catu daya yang baik terdiri dari transformator (*trafo*), penyearah arus listrik (*rectifier*), penyaring (*filter*), dan *voltage regulator*. *Trafo* berfungsi untuk mengubah besarnya tegangan listrik sehingga sesuai dengan yang dibutuhkan, *trafo* yang digunakan adalah *trafo step-down*, yaitu *trafo* yang mengubah tegangan listrik

220 volt menjadi tegangan output 5 volt. Penyearah (*rectifier*) adalah komponen yang mampu mengubah arus listrik *alternating current* (AC) menjadi arus listrik *direct current* (DC). Penyaring (*filter*) digunakan untuk meratakan denyut (*ripple*) gelombang pada arus DC. Kapasitor merupakan komponen yang biasa digunakan sebagai *filter* pada suatu *power supply*. *Voltage Regulator* berfungsi untuk mempertahankan nilai tegangan DC keluaran *power supply* walaupun masukan tegangan DC atau beban berubah-ubah.

2.14 *Liquid Crystal Display (LCD)*

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan diberbagai bidang misalnya alal–alat elektronik seperti televisi, kalkulator, atau pun layar komputer. LCD berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat. LCD sekarang semakin banyak digunakan, dari yang berukuran kecil, seperti LCD pada sebuah MP3 *player* sampai yang berukuran besar seperti monitor PC atau televisi. Warna yang dapat ditampilkan bisa bermacam–macam, dari yang 1 warna (*monochrome*) sampai yang 65.000 warna. Pola (*pattern*) LCD juga bisa bervariasi, dari pola yang membentuk *display* 7 segmen (misalnya LCD yang dipakai untuk jam tangan) sampai LCD yang bisa menampilkan karakter/teks dan LCD yang bisa menampilkan gambar. (Aris, 2012).



Gambar 2.7 Bentuk Fisik LCD

Tabel 2.4 Fungsi – Fungsi PIN pada LCD.

| PIN | Nama | Fungsi |
|-----|------|--------|
| 1 | Vss | GND |
| 2 | Vcc | +5v |
| 3 | Vee | LCD |

| | | |
|----|------|---|
| 4 | RS | 1 = <i>Input data</i> , 0 = <i>Input Intruksi</i> |
| 5 | R/W | 1 = <i>Read</i> , 0 = <i>Write</i> |
| 6 | E | <i>Enable</i> |
| 7 | D0 | Data 0 |
| 8 | D1 | Data 1 |
| 9 | D2 | Data 2 |
| 10 | D3 | Data 3 |
| 11 | D4 | Data 4 |
| 12 | D5 | Data 5 |
| 13 | D6 | Data 6 |
| 14 | D7 | Data 7 |
| 15 | VBL+ | 4 – 4.2 <i>volt</i> |
| 16 | VBL- | GND |

Pada LCD yang bisa menampilkan karakter (LCD karakter) dan LCD yang bisa menampilkan gambar (LCD grafik), diperlukan memori untuk membangkitkan gambar CGROM (*Character Generator ROM*) dan juga RAM untuk menyimpan data (teks atau gambar) yang sedang ditampilkan (DDRAM atau *Display Data RAM*). Diperlukan pula pengendali (*controller*) untuk berkomunikasi dengan mikrokontroler.

LCD karakter adalah LCD yang bisa menampilkan karakter ASCII dengan format dot matriks. LCD jenis ini bisa dibuat dengan berbagai ukuran, 1 sampai 4 baris, 16 sampai 40 karakter per baris dan dengan ukuran *font* 5x7 atau 5x10. LCD ini biasanya dirakit dengan sebuah PCB yang berisi pembangkit karakter dan IC pengendali serta *driver*-nya. Walaupun ukuran LCD berbeda-beda, tetapi IC pengendali yang digunakan biasanya sama sehingga protokol komunikasi dengan IC juga sama.