

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Adaptor

Adaptor adalah sebuah perangkat berupa rangkaian elektronika untuk mengubah tegangan listrik yang besar menjadi tegangan listrik lebih kecil, atau rangkaian untuk mengubah arus bolak-balik (arus AC) menjadi arus searah (arus DC). Adaptor / *power supply* merupakan komponen inti dari peralatan elektronik. Adaptor digunakan untuk menurunkan tegangan AC 22 Volt menjadi kecil antara 3 volt sampai 12 volt sesuai kebutuhan alat elektronika. Terdapat 2 jenis adaptor berdasarkan sistem kerjanya, adaptor sistem trafo *step down* dan adaptor sistem *switching*.

Dalam prinsip kerjanya kedua sistem adaptor tersebut berbeda, adaptor *step-down* menggunakan teknik induksi medan magnet, komponen utamanya adalah kawat email yang di lilit pada teras besi, terdapat 2 lilitan yaitu lilitan primer dan lilitan skunder, ketika listrik masuk kelilitan primer maka akan terjadi induksi pada kawat email sehingga akan terjadi gaya medan magnet pada teras besi kemudian akan menginduksi lilitan skunder.

Sedangkan sistem *switching* menggunakan teknik transistor maupun IC *switching*, adaptor ini lebih baik dari pada adaptor teknik induksi, tegangan yang di keluarkan lebih stabil dan komponennya suhunya tidak terlalu panas sehingga mengurangi tingkat resiko kerusakan karena suhu berlebih, biasanya regulator ini di gunakan pada peralatan elektronik digital.

Adaptor dapat dibagi menjadi empat macam, diantaranya adalah sebagai berikut :

1. Adaptor DC *Converter*, adalah sebuah adaptor yang dapat mengubah tegangan DC yang besar menjadi tegangan DC yang kecil. Misalnya : Dari tegangan 12v menjadi tegangan 6v;
2. Adaptor *Step Up* dan *Step Down*. Adaptor *Step Up* adalah sebuah adaptor yang

- dapat mengubah tegangan AC yang kecil menjadi tegangan AC yang besar. Misalnya : Dari Tegangan 110v menjadi tegangan 220v. Sedangkan Adaptor *Step Down* adalah adaptor yang dapat mengubah tegangan AC yang besar menjadi tegangan AC yang kecil. Misalnya : Dari tegangan 220v menjadi tegangan 110v.
3. Adaptor *Inverter*, adalah adaptor yang dapat mengubah tegangan DC yang kecil menjadi tegangan AC yang besar. Misalnya : Dari tegangan 12v DC menjadi 220v AC.
  4. Adaptor *Power Supply*, adalah adaptor yang dapat mengubah tegangan listrik AC yang besar menjadi tegangan DC yang kecil. Misalnya : Dari tegangan 220v AC menjadi tegangan 6v, 9v, atau 12v DC.



**Gambar 2.1 Bentuk Fisik Adaptor**

(Sumber : tecnoarnel.com.ve, diakses pada tanggal 04 April 2017, Pukul 13:34)

## **2.2. Arduino**

### **2.2.1. Pengertian Arduino**

Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC, sehingga sering disebut *single chip microcomputer*. Lebih lanjut, mikrokontroler merupakan sistem komputer

yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC (*Personal Computer*) yang memiliki beragam fungsi. Perbedaan lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dengan mikrokontroler.

Pengertian Arduino Menurut (Feri Djuandi, 2011) Arduino adalah merupakan sebuah *board minimum system* mikrokontroler yang bersifat *open source*. Didalam rangkaian board arduino terdapat mikrokontroler AVR seri ATmega 328 yang merupakan produk dari Atmel.

Menurut Sulaiman (2012:1), arduino merupakan *platform* yang terdiri dari *software* dan *hardware*. *Hardware* Arduino sama dengan mikrokontroler 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. Pemrograman Arduino tidak sebanyak tahapan mikrokontroler konvensional karena Arduino sudah didesain mudah untuk dipelajari, sehingga para pemula dapat mulai belajar mikrokontroler dengan Arduino.

Menurut Santosa (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 pengertian - pengertian yang dikemukakan diatas maka dapat disimpulkan bahwa arduino merupakan *kit* elektronik atau papan rangkaian elektronik yang didalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dan memiliki *software* pemrograman yang berlisensi *open source*.

### **2.2.2. Hardware**

*Hardware* dalam arduino memiliki beberapa jenis, yang memiliki kelebihan dan kekurangan di masing-masing papannya. Penggunaan jenis arduino disesuaikan dengan kebutuhan, hal inilah yang mempengaruhi dari jenis prosessor yang akan digunakan. Apabila semakin kompleks perancangan dan program yang akan dibuat

atau dirancang, maka harus sesuai pula jenis kontroler yang digunakan. Yang membedakan antara arduino yang satu dengan arduino jenis lainnya adalah penambahan fungsi dalam setiap board-nya dan jenis mikrokontroler yang digunakan. Dalam tugas akhir ini, jenis arduino yang digunakan adalah arduino mega 2560.

### 2.2.2.1.Arduino UNO

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.

**Tabel 2.1 Deskripsi Arduino UNO**

Mikrokontroler	ATmega328
Tegangan pengoperasian	5V
Tegangan input yang disarankan	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
Jumlah pin input analog	6
Arus DC tiap pin I/O	40 Ma
Arus DC untuk pin 3.3V	50 Ma
Memori Flash	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

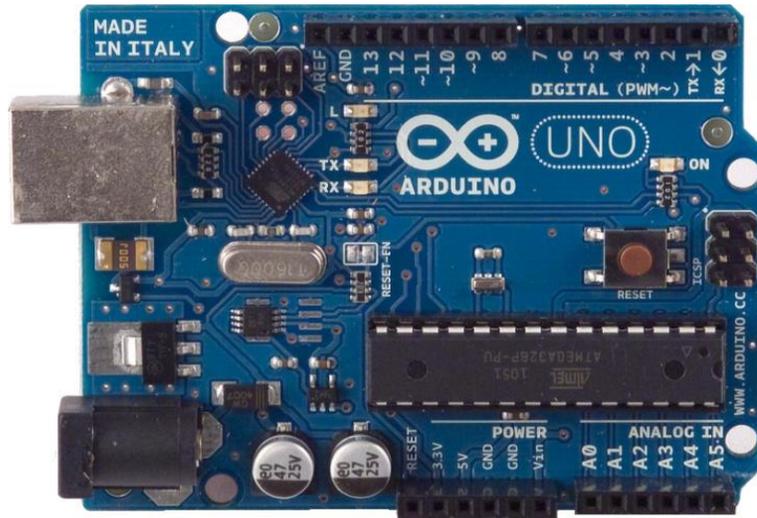
Clock Speed	16 MHz
-------------	--------

(Sumber : B. Gustomo, 2015, diakses pada tanggal 10 Juni 2017, Pukul 17:14)

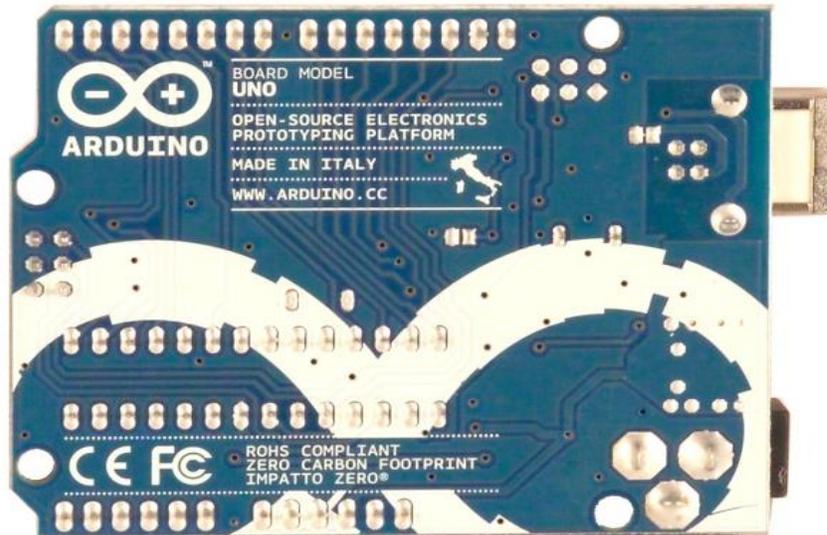
Arduino Uno berbeda dari semua *board* Arduino sebelumnya, Arduino UNO tidak menggunakan *chip driver* FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke serial. Revisi 2 dari board Arduino Uno mempunyai sebuah resistor yang 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:

- 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. Yang kedua ini merupakan sebuah pin yang tak terhubung, yang disediakan untuk tujuan kedepannya
- Sirkit RESET yang lebih kuat
- 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, untuk suatu perbandingan dengan versi sebelumnya. Tampilan Arduino UNO dapat dilihat pada gambar di bawah ini.



**Gambar 2.2 Bentuk Fisik Tampilan depan Arduino UNO**  
(Sumber : B. Gustomo, 2015, diakses pada tanggal 10 Juni 2017, Pukul 17:24)



**Gambar 2.3 Bentuk Fisik Tampilan belakang Arduino UNO**  
(Sumber : B. Gustomo, 2015, diakses pada tanggal 10 Juni 2017, Pukul 17:24)

Arduino UNO dapat disuplai melalui koneksi USB atau dengan sebuah *power* suplai eksternal. Sumber daya dipilih secara otomatis.

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

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

Adapun Pin-pin dayanya adalah sebagai berikut:

- VIN. Tegangan input ke Arduino *board* ketika *board* sedang menggunakan sumber suplai eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur). Kita dapat menyuplai tegangan melalui pin ini, atau jika penyuplaian tegangan melalui *power jack*, aksesnya melalui pin ini.
- 5V. Pin *output* ini merupakan tegangan 5 Volt yang diatur dari regulator pada *board*. *Board* dapat disuplai dengan salah satu suplai dari DC *power jack* (7-12V), USB *connector* (5V), atau pin VIN dari board (7-12). Penyuplaian tegangan melalui pin 5V atau 3,3V mem-*bypass* regulator, dan dapat membahayakan *board*. Hal itu tidak dianjurkan.
- 3V3. Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada *board*. Arus maksimum yang dapat dilalui adalah 50 mA.
- GND. Pin *ground*.

ATmega328 mempunyai 32 KB (dengan 0,5 KB digunakan untuk *bootloader*). ATmega 328 juga mempunyai 2 KB SRAM dan 1 KB *EEPROM* (yang dapat dibaca dan ditulis (RW/read and written) dengan *EEPROM library*).

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

- *Serial*: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan memancarkan (TX) serial data TTL (Transistor-Transistor Logic). Kedua pin ini dihubungkan ke pin-pin yang sesuai dari *chip* Serial Atmega8U2 USB-ke-TTL.
- *External Interrupts*: 2 dan 3. Pin-pin ini dapat dikonfigurasi untuk dipicu sebuah interrupt (gangguan) pada sebuah nilai rendah, suatu kenaikan atau penurunan yang besar, atau suatu perubahan nilai.
- *PWM*: 3, 5, 6, 9, 10, dan 11. Memberikan 8-bit PWM output dengan fungsi *analogWrite()*.
- *SPI*: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin-pin ini *men-support* komunikasi SPI menggunakan *SPI library*.
- *LED*: 13. Ada sebuah LED yang terpasang, terhubung ke pin digital 13. Ketika pin bernilai *HIGH* LED menyala, ketika pin bernilai *LOW* LED mati.

Arduino UNO mempunyai 6 input analog, diberi label A0 sampai A5, setiapnya memberikan 10 bit resolusi (contohnya 1024 nilai yang berbeda). Secara *default*, 6 input analog tersebut mengukur dari ground sampai tegangan 5 Volt, dengan itu mungkin untuk mengganti batas atas dari rangenya dengan menggunakan pin AREF dan fungsi *analogReference()*. Di sisi lain, beberapa pin mempunyai fungsi spesial:

- *I2C*: pin A4 atau SDA dan pin A5 atau SCL. Men-*support* komunikasi *I2C* dengan menggunakan *Wire library*.

Adapun sepasang pin lainnya pada board Arduino UNO yaitu:

- *AREF*. Referensi tegangan untuk *input* analog. Digunakan dengan *analogReference()*.
- *Reset*. Membawa saluran ini *LOW* untuk mereset mikrokontroler. Secara khusus, digunakan untuk menambahkan sebuah tombol reset untuk melindungi yang mem-*block* sesuatu pada *board*.

Arduino UNO mempunyai sejumlah fasilitas untuk komunikasi dengan sebuah komputer, Arduino lainnya atau mikrokontroler lainnya. Atmega 328 menyediakan serial komunikasi UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah Atmega 16U2 pada *channel board* serial komunikasinya melalui USB dan muncul sebagai sebuah port virtual ke *software* pada komputer. *Firmware* 16U2 menggunakan *driver* USB COM standar, dan tidak ada *driver* eksternal yang dibutuhkan. *Software* Arduino mencakup sebuah serial monitor yang memungkinkan data tekstual terkirim ke dan dari *board* Arduino. LED RX dan TX pada *board* akan menyala ketika data sedang ditransmit melalui *chip USB-to-serial* dan koneksi USB pada komputer (tapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *Software Serial library* memungkinkan untuk komunikasi serial pada beberapa pin digital UNO.

Atmega328 juga men-*support* komunikasi *I2C* (*I2C*) dan *SPI*. *Software* Arduino mencakup sebuah *Wire library* untuk memudahkan menggunakan bus *I2C*.

ATmega328 pada Arduino Uno hadir dengan sebuah *bootloader* yang memungkinkan kita untuk meng-*upload* kode baru ke ATmega328 tanpa menggunakan pemrogram *hardware* eksternal. ATmega328 berkomunikasi menggunakan protokol STK500 asli (referensi, file *C header*)

Kita juga dapat mem-*bypass bootloader* dan program mikrokontroler melalui kepala/header ICSP (*In-Circuit Serial Programming*).

Sumber kode *firmware* ATmega16U2 (atau 8U2 pada *board* revisi 1 dan revisi 2) tersedia. ATmega16U2/8U2 di-*load* dengan sebuah *bootloader* DFU, yang dapat diaktifkan dengan:

- Pada *board* Revisi 1: Dengan menghubungkan *jumper* solder pada belakang *board* (dekat peta Italy) dan kemudian mereset 8U2;
- Pada *board* Revisi 2 atau setelahnya: Ada sebuah resistor yang menarik garis HWB 8U2/16U2 ke *ground*, dengan itu dapat lebih mudah untuk meletakkan ke dalam mode DFU. Kita dapat menggunakan *software* Atmel's FLIP (Windows) atau pemrogram DFU (Mac OS X dan Linux) untuk me-*load* sebuah *firmware* baru. Atau kita dapat menggunakan header ISP dengan sebuah pemrogram eksternal (meng-*overwrite bootloader* DFU).

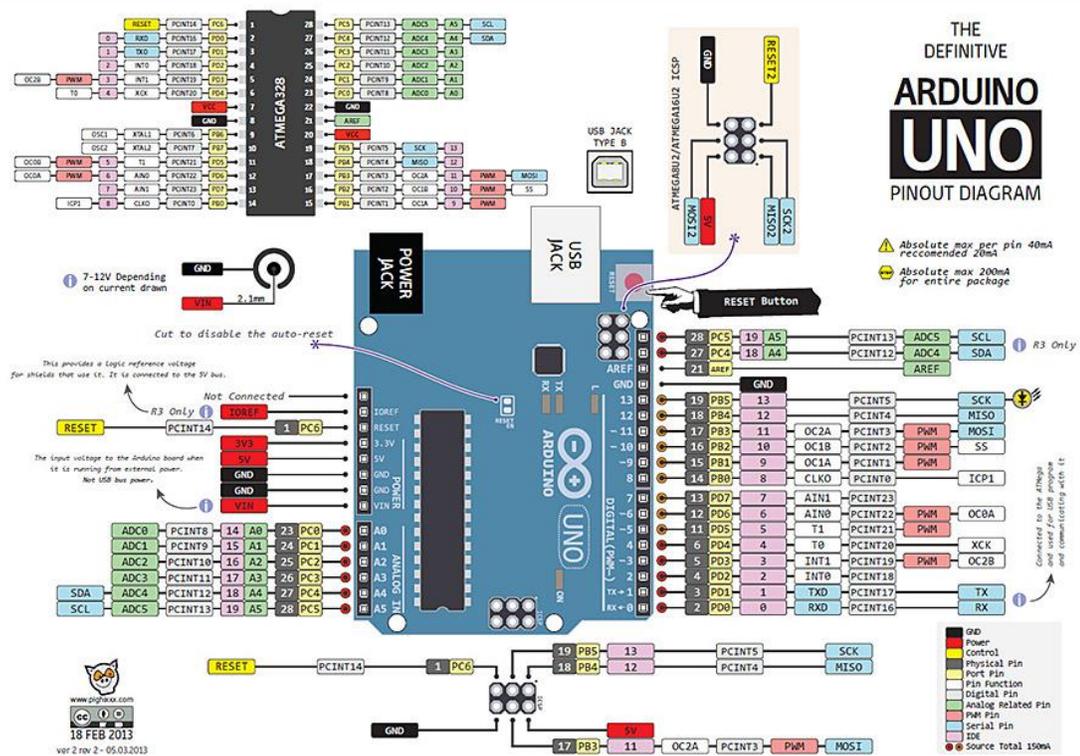
Arduino Uno berisikan sebuah jejak yang dapat dihapus untuk mencegah reset otomatis. Pad pada salah satu sisi dari jejak dapat disolder bersama untuk mengaktifkan kembali. Pad itu diberi label "RESET-RN" Kita juga dapat menonaktifkan reset otomatis dengan menghubungkan sebuah resistor 110 ohm dari tegangan 5V ke garis reset.

Arduino UNO mempunyai sebuah sekering reset yang memproteksi port USB komputer dari hubungan pendek dan arus lebih. Walaupun sebagian besar komputer menyediakan proteksi internal sendiri, sekering menyediakan sebuah proteksi tambahan. Jika lebih dari 500 mA diterima port USB, sekering secara otomatis akan memutuskan koneksi sampai hubungan pendek atau kelebihan beban hilang.

Panjang dan lebar maksimum dari PCB Arduino UNO masing-masingnya adalah 2.7 dan 2.1 inci, dengan konektor USB dan *power jack* yang memperluas dimensinya. Empat lubang sekrup memungkinkan *board* untuk dipasangkan ke sebuah permukaan atau kotak. Sebagai catatan, bahwa jarak antara pin digital 7 dan 8

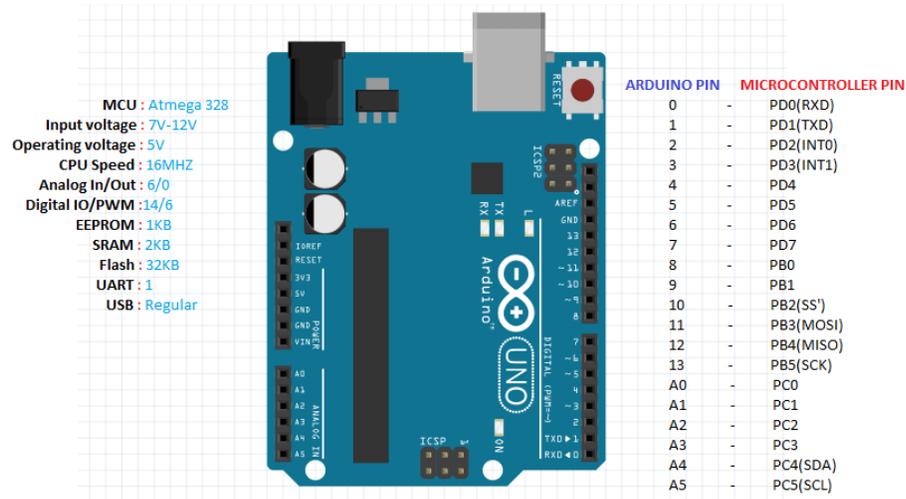
adalah 160 mil. (0.16"), bukan sebuah kelipatan genap dari jarak 100 mil dari pin lainnya.

Arduino UNO dapat diprogram dengan Arduino Software (IDE). Software IDE yang digunakan juga sama, hanya tinggal memilih board Arduino UNO pada pilihan board-nya.



**Gambar 2.4 Rangkaian PIN Diagram Arduino UNO**

(Sumber : <https://forum.arduino.cc/>, diakses pada tanggal 10 Juni 2017, Pukul 18:35)



**Gambar 2.5 Arduino UNO PIN Mapping**

(Sumber : <https://www.arduino.cc/>, diakses pada tanggal 10 Juni 2017, Pukul 18:48)

## 2.2.3. Arduino Software (IDE)

### 2.2.3.1. Pengertian Arduino Software (IDE)

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan 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.

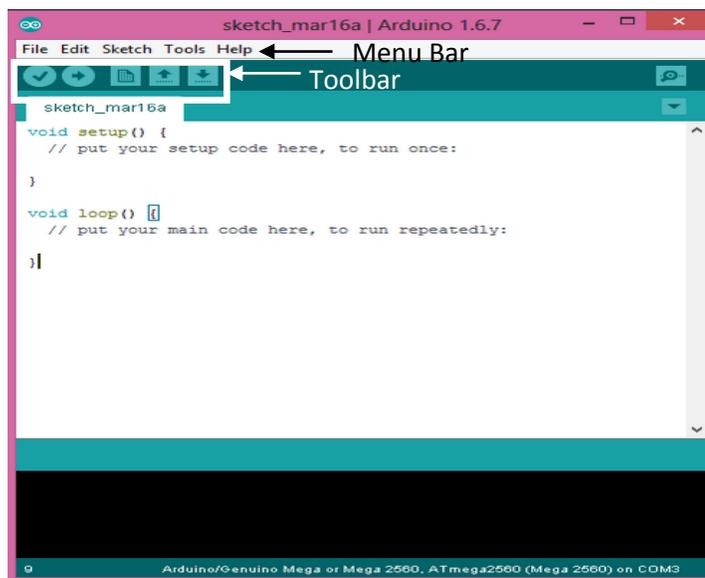
Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library C/C++* yang biasa disebut *Wiring* yang membuat operasi input dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software*

Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

### 2.2.3.2. Menulis Sketch

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-fitur seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan pengguna dalam menulis kode program.

Pada *Software* Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *Software* Arduino IDE, menunjukan *board* yang terkonfigurasi beserta *COM Ports* yang digunakan.



**Gambar 2.6** Tampilan dari Software Arduino IDE

(Sumber : <https://www.arduino.cc/>, diakses pada tanggal 05 Mei 2017, Pukul 03:20)

**Tabel 2.2 *Symbol dan fungsi dari Toolbar Software Arduino IDE***

	<p><i>Verify</i></p> <p>berfungsi untuk melakukan checking kode yang dibuat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum</p>
	<p><i>Upload</i></p> <p>Berfungsi untuk melakukan kompilasi program atau kode yang dibuat menjadi bahasa yang dapat dipahami oleh mesin alias si Arduino.</p>
	<p><i>New</i></p> <p>berfungsi untuk membuat <i>Sketch</i> baru</p>
	<p><i>Open</i></p> <p>Berfungsi untuk membuka <i>sketch</i> yang pernah dibuat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke Arduino.</p>
	<p><i>Save</i></p> <p>Berfungsi untuk menyimpan <i>Sketch</i> yang telah dibuat</p>
	<p><i>Serial Monitor</i></p> <p>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 sangat berguna sekali ketika Anda ingin 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 error.</p>

(Sumber : <http://arduino.com/>, diakses pada tanggal 05 Mei 2017, Pukul 03:28)

Berikut merupakan bagian-bagian dari menu *bar* dan penjelasannya yang terdapat pada *software* Arduino IDE.

**a. File**

- *New*, berfungsi untuk membuat membuat *sketch* baru dengan *bare* minimum yang terdiri *void setup()* dan *void loop()*.
- *Open*, berfungsi membuka *sketch* yang pernah dibuat di dalam *drive*.
- *Open Recent*, merupakan menu yang berfungsi mempersingkat waktu pembukaan file atau *sketch* yang baru-baru ini sudah dibuat.
- *Sketchbook*, berfungsi menunjukkan hirarki *sketch* yang kamu buat termasuk struktur foldernya.
- *Example*, berisi contoh-contoh pemrograman yang disediakan pengembang Arduino, sehingga pengguna dapat mempelajari program-program dari contoh yang diberikan.
- *Close*, berfungsi menutup jendela Arduino IDE dan menghentikan aplikasi.
- *Save*, berfungsi menyimpan *sketch* yang dibuat atau perubahan yang dilakukan pada *sketch*
- *Save as...*, berfungsi menyimpan *sketch* yang sedang dikerjakan atau *sketch* yang sudah disimpan dengan nama yang berbeda.
- *Page Setup*, berfungsi mengatur tampilan *page* pada proses pencetakan.
- *Print*, berfungsi mengirimkan file *sketch* ke mesin cetak untuk dicetak.
- *Preferences*, disini pengguna dapat merubah tampilan *interface* IDE Arduino.
- *Quit*, berfungsi menutup semua jendela Arduino IDE. *Sketch* yang masih terbuka pada saat tombol *Quit* ditekan, secara otomatis akan terbuka pada saat Arduino IDE dijalankan.

**b. Edit**

- *Undo/Redo*, berfungsi untuk mengembalikan perubahan yang sudah dilakukan pada *Sketch* beberapa langkah mundur dengan *Undo* atau maju dengan *Redo*.

- **Cut**, berfungsi untuk me-*remove* teks yang terpilih pada editor dan menempatkan teks tersebut pada *clipboard*.
- **Copy**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard*.
- **Copy for Forum**, berfungsi melakukan *copy* kode dari editor dan melakukan *formatting* agar sesuai untuk ditampilkan dalam forum, sehingga kode tersebut bisa digunakan sebagai bahan diskusi dalam forum.
- **Copy as HTML**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard* dalam bentuk atau format HTML. Biasanya ini digunakan agar *code* dapat di-*embedded*-kan pada halaman web.
- **Paste**, berfungsi menyalin data yang terdapat pada *clipboard*, kedalam editor.
- **Select All**, berfungsi untk melakukan pemilihan teks atau kode dalam halaman editor.
- **Comment/Uncomment**, berfungsi memberikan atau menghilangkan tanda // pada kode atau teks, dimana tanda tersebut menjadikan suatu baris kode sebagai komen dan tidak disertakan pada tahap kompilasi.
- **Increase/Decrease Indent**, berfungsi untuk mengurangi atau menambahkan indentntasi pada baris kode tertentu. Indentasi adalah “tab”.
- **Find**, berfungsi memanggil jendela *window “find and replace”*, dimana pengguna dapat menggunakannya untuk menemukan variabel atau kata tertentu dalam program atau menemukan serta menggantikan kata tersebut dengan kata lain.
- **Find Next**, berfungsi menemukan kata setelahnya dari kata pertama yang berhasil ditemukan.
- **Find Previous**, berfungsi menemukan kata sebelumnya dari kata pertama yang berhasil ditemukan.

### c. *Sketch*

- ***Verify/Compile***, berfungsi untuk mengecek apakah *sketch* yang kamu buat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang kamu buat akan di-*compile* kedalam bahasa mesin.
- ***Upload***, berfungsi mengirimkan program yang sudah dikompilasi ke Arduino Board.
- ***Uplad Using Programmer***, menu ini berfungsi untuk menuliskan *bootloader* kedalam IC Mikrokontroler Arduino. Pada kasus ini kamu membutuhkan perangkat tambahan seperti *USBasp* untuk menjembatani penulisan program *bootloader* ke IC Mikrokontroler.
- ***Export Compiled Binary***, berfungsi untuk menyimpan *file* dengan ekstensi **.hex**, dimana file ini dapat disimpan sebagai arsip untuk di *upload* ke *board* lain menggunakan *tools* yang berbeda.
- ***Show Sketch Folder***, berfungsi membuka folder *sketch* yang saat ini dikerjakan.
- ***Include Library***, berfungsi menambahkan *library* / pustaka kedalam *sketch* yang dibuat dengan menyertakan sintaks *#include* di awal kode. Selain itu kamu juga bisa menambahkan *library* eksternal dari *file .zip* kedalam Arduino IDE.
- ***Add File...***, berfungsi untuk menambahkan *file* kedalam *sketch* arduino (*file* akan dikopikan dari drive asal). File akan muncul sebagai tab baru dalam jendela *sketch*.

### d. *Tools*

- ***Auto Format***, berfungsi melakukan pengaturan format kode pada jendela editor.
- ***Archive Sketch***, berfungsi menyimpan *sketch* kedalam file **.zip**.
- ***Fix Encoding & Reload***, berfungsi memperbaiki kemungkinan perbedaan

- antara pengkodean peta karakter editor dan peta karakter sistem operasi yang lain..
- **Serial Monitor**, berfungsi membuka jendela serial monitor untuk melihat pertukaran data.
- **Board**, berfungsi memilih dan melakukan konfigurasi *board* yang digunakan.
- **Port**, memilih *port* sebagai kanal komunikasi antara *software* dengan *hardware*.
- **Programmer**, menu ini digunakan ketika pengguna hendak melakukan pemrograman *chip mikrokontroler* tanpa menggunakan koneksi *Onboard USB-Serial*. Biasanya digunakan pada proses *burning bootloader*.
- **Burn Bootloader**, mengizinkan kamu untuk mengkopikan program *bootloader* kedalam IC mikrokontroler

#### e. **Help**

Disini kamu bisa mendapatkan bantuan terhadap kegalauanmu mengenai pemrograman. Menu *help* berisikan file-file dokumentasi yang berkaitan dengan masalah yang sering muncul, serta penyelesaiannya. Selain itu pada menu *help* juga diberikan link untuk menuju Arduino Forum guna menanyakan serta mendiskusikan berbagai masalah yang ditemukan.

#### 2.2.3.3. **Sketchbook**

Arduino *Software IDE*, menggunakan konsep *sketchbook*, dimana *sketchbook* menjadi standar peletakan dan penyimpanan file program. *Sketch* yang telah dibuat dapat dibuka dengan dari *File* → **Sketchbook**, atau dengan menu **Open**.

#### 2.2.3.4. **Tabs, Multiple Files, dan Compilations**

Mekanisme ini mengijinkan pengguna untuk melakukan menejemen *sketch*, dimana lebih dari satu *file* dibuka dalam tab yang berbeda.

### **2.2.3.5.Uploading**

Merupakan mekanisme untuk mengkopikan file **.hex** atau *file* hasil kompilasi kedalam IC mikrokontroler Arduino. Sebelum melakukan *uploading*, yang perlu dipastikan adalah jenis *board* yang kamu gunakan dan *COM Ports* dimana keduanya terletak pada menu **Tools** → **Board** dan **Tools** → **Port**.

### **2.2.3.6.Library**

*Library* / Pustaka merupakan *file* yang memberikan fungsi ekstra dari *sketch* yang dibuat, agar Arduino dapat bekerja dengan *hardware* tertentu dan melakukan proses manipulasi data. Untuk menginstal *Library*, pihak ketiga yaitu *Library* bukan dari Arduino, dapat dilakukan dengan *Library Manager*, *Import file .zip*, atau *copy paste* secara manual di folder *libraries* pada *Documents* di *platform Windows*.

### **2.2.3.7.Serial Monitor**

Serial monitor merupakan suatu jendela yang menunjukkan data yang dipertukaran antara arduino dan komputer selama beroperasi, sehingga pengguna bisa menggunakan serial monitor ini untuk menampilkan nilai hasil operasi atau pesan *debugging*. Selain melihat data, pengguna juga bisa mengirimkan data ke Arduino melalui serial monitor ini, caranya dengan memasukkan data pada *text box* dan menekan tombol *send* untuk mengirimkan data. Hal penting yang harus diperhatikan adalah menyamakan *baudrate* antara serial monitor dengan *Arduino board*. Untuk menggunakan kemampuan komunikasi serial ini, pada Arduino, di bagian fungsi *void setup()*, diawali dengan instruksi **Serial.begin** diikuti dengan nilai *baudrate*.

### **2.2.3.8.Preferences**

*Preferences* mengatur tentang beberapa hal dalam penggunaan *Arduino Software IDE*, seperti ukuran font, lokasi dimana menyimpan *sketchbook*, bahasa yang digunakan pada *Arduino Software IDE*, dan masih banyak lagi. Pengguna bisa

mengatur *preferences* pada menu file yang dapat dijumpai pada *platform Windows* dan Linux.

#### **2.2.3.9. Language Support**

*Language Support* merupakan pilihan bahasa yang dapat disesuaikan pada *Software Arduino IDE*. Pada Arduino sudah terdapat pilihan Bahasa Indonesia. *Language Support* ini dapat ditemukan pada menu *file* → *preferences* atau dengan menekan **Ctrl+Comma**.

#### **2.2.3.10. Boards**

Pemilihan *board* pada *Arduino Software IDE*, berdampak pada dua parameter yaitu kecepatan CPU dan *baudrate* yang digunakan ketika melakukan kompilasi dan meng-*upload sketch*.

### **2.3. Keypad 4 X 4**

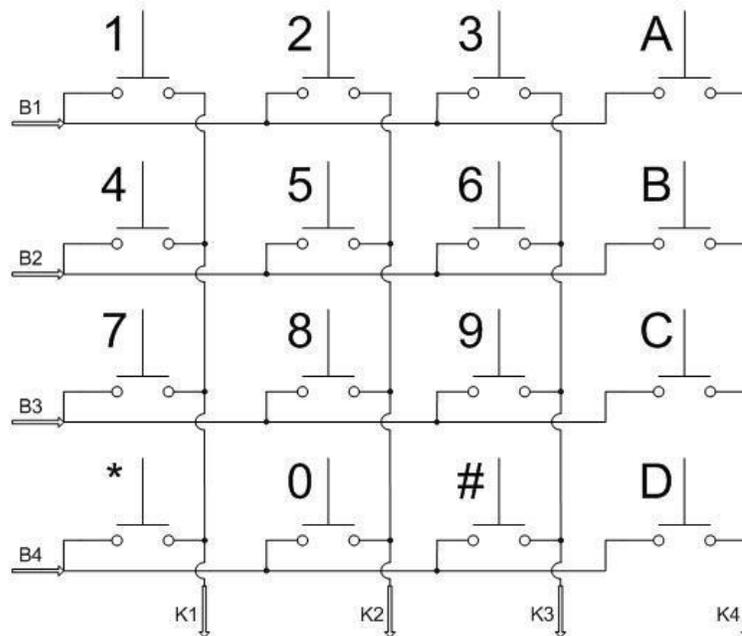
*Keypad* adalah bagian penting dari suatu perangkat elektronika yang membutuhkan interaksi manusia. *Keypad* berfungsi sebagai *interface* antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah HMI (*Human Machine Interface*).

*Keypad Matriks* adalah tombol-tombol yang disusun secara maktriks (baris x kolom) sehingga dapat mengurangi penggunaan *pin input*. Sebagai contoh, Keypad Matriks 4×4 cukup menggunakan 8 pin untuk 16 tombol. *Matrix keypad 4×4* memiliki konstruksi atau susunan yang *simple* dan hemat dalam penggunaan *port* mikrokontroler. Konfigurasi *keypad* dengan susunan bentuk *matrix* ini bertujuan untuk penghematan *port* mikrokontroler karena jumlah *key* (tombol) yang dibutuhkan banyak pada suatu sistem dengan mikrokontroler. Konstruksi *matrix keypad 4×4* untuk mikrokontroler dapat dibuat seperti pada gambar berikut. Hal tersebut dimungkinkan karena rangkaian tombol disusun secara horizontal membentuk baris dan secara vertikal membentuk kolom.



**Gambar 2.7 Bentuk Fisik *Matrix* 4 x 4**

(Sumber : <https://depokinstruments.com/>, diakses pada tanggal 01 April 2017, Pukul 12:36)



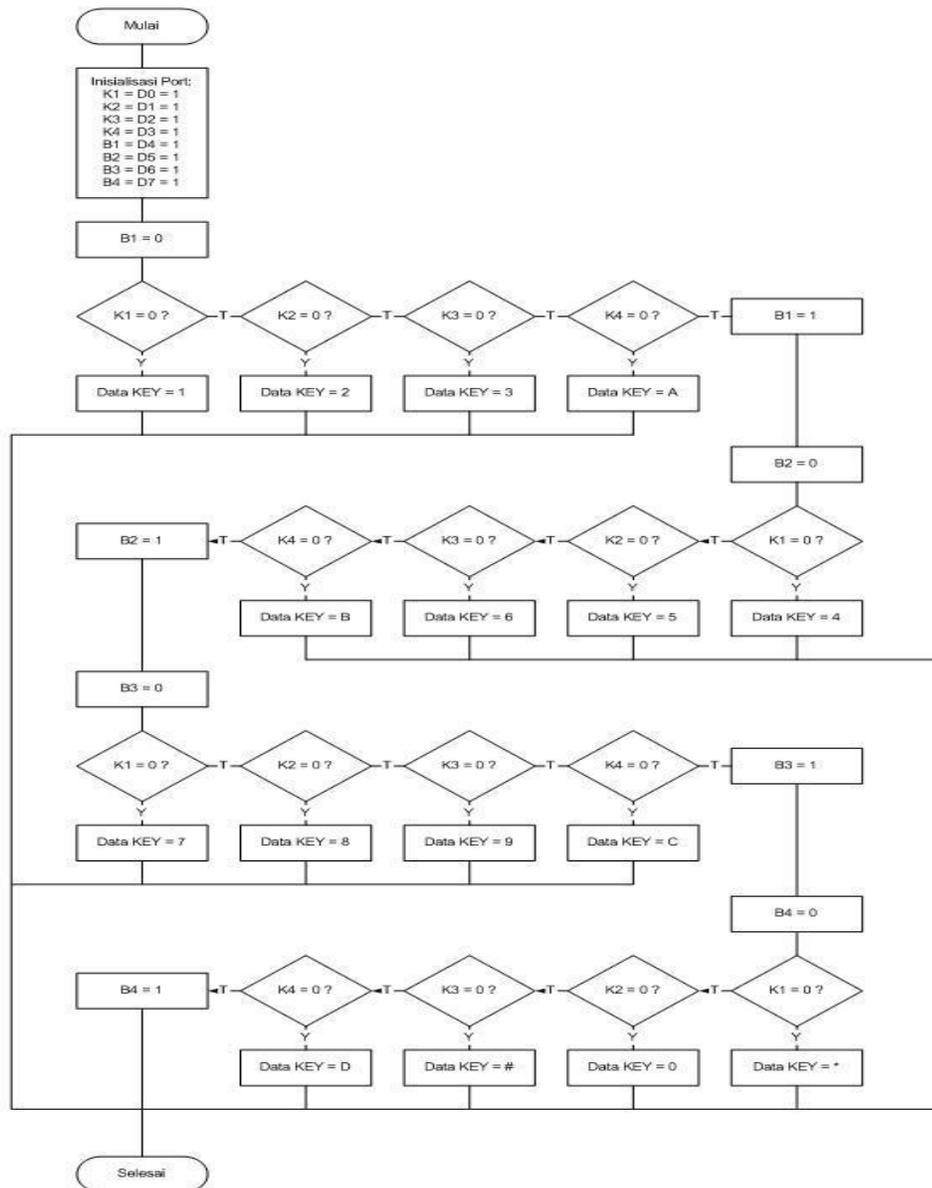
**Gambar 2.8 Rangkaian *Keypad Matrix* 4 x 4**

(Sumber : <https://depokinstruments.com/>, diakses pada tanggal 01 April 2017, Pukul 12:36)

Konstruksi *matrix keypad* 4×4 diatas cukup sederhana, yaitu terdiri dari 4 baris dan 4 kolom dengan *keypad* berupa saklar *push buton* yang diletakan disetiap persilangan kolom dan barisnya. Rangkaian *matrix keypad* diatas terdiri dari 16 saklar *push buton* dengan konfigurasi 4 baris dan 4 kolom. 8 line yang terdiri dari 4 baris dan 4 kolom tersebut dihubungkan dengan port mikrokontroler 8 bit. Sisi baris dari *matrix keypad* ditandai dengan nama B1, B2, B3, dan B4, kemudian sisi kolom ditandai dengan nama K1, K2, K3, dan K4. Sisi input atau output dari *matrix keypad* 4×4 ini tidak mengikat, dapat dikonfigurasikan kolom sebagai input dan baris sebagai *output* atau sebaliknya tergantung programernya.

Proses pengecekan dari tombol yang dirangkai secara maktriks adalah dengan teknik *scanning*, yaitu proses pengecekan yang dilakukan dengan cara memberikan umpan-data pada satu bagian dan mengecek *feedback* (umpan-balik) – nya pada bagian yang lain. Dalam hal ini, pemberian umpan-data dilakukan pada bagian baris dan pengecekan umpan-balik pada bagian kolom. Pada saat pemberian umpan-data pada satu baris, maka baris yang lain harus dalam kondisi inversinya. Tombol yang ditekan dapat diketahui dengan melihat asal data dan di kolom mana data tersebut terdeteksi.

Konfigurasi *wiring* diagram antara keypad *matrix* dan *microcontroller* menentukan bagaimana *scanning* dilakukan. *Flowhcart* berikut memperlihatkan proses *scanning* Keypad Matriks 4×4 secara lengkap:



**Gambar 2.9 Flowcart Proses Scanning pada Keypad Matrix 4 x 4**

(Sumber : <https://depokinstruments.com/>, diakses pada tanggal 01 April 2017, Pukul 12:36)

Proses *scanning* untuk membaca penekanan tombol pada *matrix keypad* 4x4 untuk mikrokontroler diatas dilakukan secara bertahap kolom demi kolom dari kolom pertama sampai kolom ke 4 dan baris pertama hingga baris ke 4. Program untuk

scanning matrix *keypad* 4×4 dapat bermacam-macam, tapi pada intinya sama. Misal kita asumsikan keypad aktif *LOW* (semua *line* kolom dan baris dipasang resistor *pull-up*) dan dihubungkan ke *port* mikrokontrolr dengan jalur kolom adalah jalur input dan jalur baris adalah jalur *output* maka proses *scanning matrix keypad* 4×4 diatas dapat dituliskan sebagai berikut :

- Mengirimkan logika *Low* untuk kolom 1 (*Col1*) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row1*) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.
- Mengirimkan logika *Low* untuk kolom 2 (*Col2*) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row1*) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.
- Mengirimkan logika *Low* untuk kolom 3 (*Col3*) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row1*) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

- Mengirimkan logika *Low* untuk kolom 4 (*Col4*) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row1*) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

Kemudian data pembacaan baris ini diolah sebagai pembacaan data penekanan tombol *keypad*. Sehingga tiap tombol pada *matrix keypad* 4×4 diatas dengan teknik scanning tersebut akan menghasilkan data penekanan tiap-tiap tombol sebagai berikut :

SW1	= 0111 0111	SW9	= 0111 1101
SW2	= 1011 0111	SW10	= 1011 1101
SW3	= 1101 0111	SW11	= 1101 1101
SW4	= 1110 0111	SW12	= 1110 1101
SW5	= 0111 1011	SW13	= 0111 1110
SW6	= 1011 1011	SW14	= 1011 1110
SW7	= 1101 1011	SW15	= 1101 1110
SW8	= 1110 1011	SW16	= 1110 1110

Data port mikrokontroler, misalkan pada SW2 = 1011 0111 tersebut terbagi dalam *nible* atas dan *nible* bawah dimana data *nible* atas (1011) merupakan data yang kita kirimkan sedangkan data *nible* bawah (0111) adalah data hasil pembacaan penekanan tombol *keypad* SW2 pada proses scanning *matrix keypad* 4×4 diatas.

#### 2.4. LCD 20 x 4 (Konektor I2C / TWI)

LCD (*Liquid Cristal Display*) adalah salah satu jenis *display* elektronik yang

dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari *back-lit*. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.

*Module* LCD 20 x 4 dengan konektor I2C/TWI merupakan produk dengan *high speed* serial bus. Produk ini dikhususkan bagi yang memiliki keterbatasan pin IO dalam project mikrokontroler yang dikerjakan. Dengan *module display* ini, kita cukup menggunakan 2 pin Arduino (mikrokontroler). Teks yang dihasilkan pada LCD 20 x 4 ini berwarna putih, dengan *background*-nya berwarna biru.



**Gambar 2.10 Bentuk Fisik LCD 20 x 4**

(Sumber : <https://drive.google.com/file>, diakses pada tanggal 01 April 2017, 14:49)

Modul LCD 20 x 4 mempunyai 16pin *interface*, diantaranya adalah VSS, VDD, V0, RS, R/W, E, D0, D1, D2, D3, D4, D5, D6, D7 BLA, BLK.

**Tabel 2.3 Deskripsi *Module* LCD 20 x 4**

I2C Address	0x27
Number of Characters	20 characters x 4 Lines
LED	Blue LED backlight with white char color
Adjustable	Contrast
Supply voltage	5V

Interface	IIC/TWI
View direction	Wide viewing angle
Dot size	0.55 x 0.55 mm
Dot pitch	0.60 x 0.60 mm
Character size	2.96 x 4.75 mm
Character pitch	3.55 x 5.35 mm
Size	98x60x24mm
Weight	75g

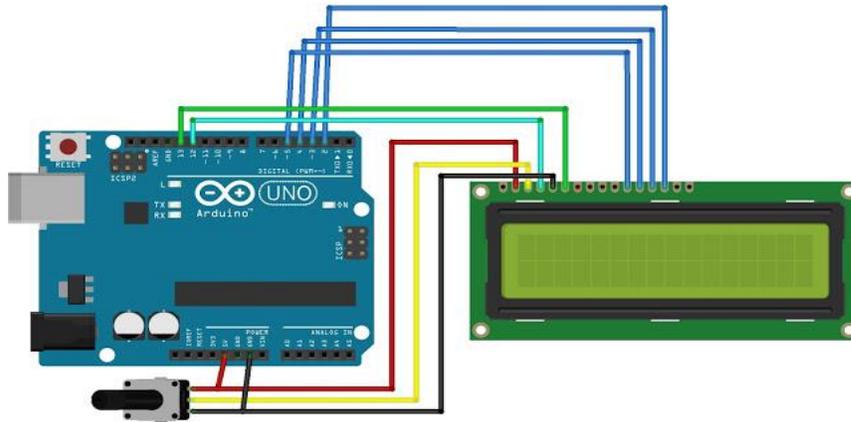
(Sumber : <https://drive.google.com/file>, diakses pada tanggal 01 April 2017, 14:49)

Berikut ini adalah pemaparan PIN, kaki atau jalur input, dan control dalam suatu LCD (*Liquid Cristal Display*) diantaranya adalah :

- 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.
- Pin RS (*Register Select*) berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukan yang masuk adalah perintah, sedangkan logika *high* menunjukan data.
- Pin R/W (*Read Write*) berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* baca data.
- Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
- 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.

Berikut adalah *Standar Wiring 20 x 4 LCD Module* tanpa *I2C Backpack* menggunakan 6 pin I/O dan Vcc Gnd total 8pin yang dihabiskan pada Arduino hanya

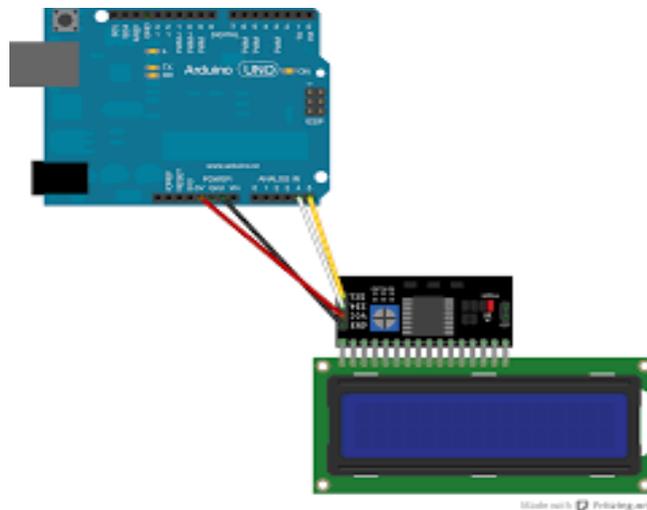
untuk pemasangan LCD, hal ini sangat tidak efisien mengingat banyak pin yang digunakan.



**Gambar 2.11 Skema Standar Wiring Modul LCD 20 x 4 tanpa I2C Backpack**

(Sumber : <https://drive.google.com/file>, diakses pada tanggal 01 April 2017, 14:49)

I2C LCD Backpack Module mempunyai 16 pin *Output* yang bisa dihubungkan dengan pin LCD 20 x 4 secara langsung (disolder permanen) dan memiliki 4pin *input* (VCC, GND, SDA, SCL). Berikut *Wiring* LCD 20 x 4 yang sudah menggunakan I2C Module ke Arduino:



(<http://forum.arduino.cc/index>, diakses pada tanggal 01 April 2017, 14:49)

## **2.5. RFID**

### **2.5.1. Pengertian RFID**

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 devais 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 devais 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.

*Radio Frequency Identification* (RFID) adalah teknologi *wireless* yang kompak yang berpotensi sangat besar untuk kemajuan perniagaan (*commerce*). RFID menggunakan *chip* yang dapat dideteksi pada range beberapa meter oleh pembaca RFID. RFID sebagai barkode generasi berikutnya dapat digunakan untuk otomatisasi *inventory control*, sehingga dapat mengurangi biaya dari pabrik ke distributor. *Tag* RFID yang telah diperbaharui mempunyai beberapa keunggulan dibandingkan dengan teknologi idetifikasi lainnya dan dapat juga digunakan untuk sistem keamanan. *Tag* RFID menawarkan solusi identifikasi dengan berbagai macam tingkat keamanan.

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.



**Gambar 2.13 Bentuk Fisik RFID**

(Sumber : <http://elektronika-dasar.web.id/>, diakses pada tanggal 01 April 2017, 16:38)

### **2.5.2. Sistem RFID**

Suatu sistem RFID dapat terdiri dari beberapa komponen, seperti *tag*, *tag reader*, *tag programming station*, *circulation reader*, *sorting equipment* dan tongkat *inventory tag*. Kegunaan dari sistem RFID ini adalah untuk mengirimkan 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 bisa berisi beragam informasi, seperti ID.

Secara garis besar sebuah sistem RFID terdiri atas tiga komponen utama, yaitu *tag*, *reader* dan basis data. Secara ringkas, mekanisme kerja yang terjadi dalam sebuah sistem RFID adalah bahwa sebuah *reader* frekuensi radio melakukan *scanning* terhadap data yang tersimpan dalam *tag*, kemudian mengirimkan informasi tersebut ke sebuah basis data yang menyimpan data yang terkandung dalam *tag* tersebut.

Sistem RFID merupakan suatu tipe sistem identifikasi otomatis yang bertujuan untuk memungkinkan data ditransmisikan oleh peralatan *portable* yang disebut *tag*, yang dibaca oleh suatu *reader* RFID dan diproses menurut kebutuhan dari aplikasi tertentu. Data yang ditransmisikan oleh *tag* dapat menyediakan informasi

identifikasi atau lokasi, atau hal-hal khusus tentang produk-produk bertag, seperti harga, warna, tanggal pembelian dan lain-lain.

### 2.5.3. Prinsip Kerja RFID

RFID menggunakan sistem identifikasi dengan gelombang radio. Untuk itu minimal dibutuhkan dua buah perangkat, yaitu yang disebut *TAG* dan *READER*. Saat pemindaian data, *READER* membaca sinyal yang diberikan oleh RFID *TAG*.

#### a. RFID Tag

RFID *Tag* adalah sebuah alat yang melekat pada obyek yang akan diidentifikasi oleh RFID *Reader*. RFID *Tag* dapat berupa perangkat pasif atau aktif. Tag pasif artinya tanpa *battery* dan Tag aktif artinya menggunakan *battery*. Tag pasif lebih banyak digunakan karena murah dan mempunyai ukuran lebih kecil. RFID *Tag* dapat berupa perangkat *read-only* yang berarti hanya dapat dibaca saja ataupun perangkat *read-write* yang berarti dapat dibaca dan ditulis ulang untuk *update*.



**Gambar 2.14 Tag RFID**

(Sumber : <http://elektronika-dasar.web.id/>, diakses pada tanggal 01 April 2017, 16:38)

RFID *Tag* mempunyai dua bagian penting, yaitu:

- IC atau kepanjangan dari *Integrated Circuit*, yang berfungsi menyimpan dan memproses informasi, modulasi dan demodulasi sinyal RF, mengambil tegangan

DC yang dikirim dari RFID *Reader* melalui induksi, dan beberapa fungsi khusus lainnya.

- Antenna, yang berfungsi menerima dan mengirim sinyal RF.

RFID *Tag* tidak berisi informasi pengguna seperti nama, nomor rekening, NIK atau yang lain. RFID *Tag* hanya berisi sebuah *Tag* yang unik yang berbeda satu dengan yang lainnya. Jadi Informasi mengenai objek yang terhubung ke *tag* ini hanya terdapat pada sistem atau *database* yang terhubung pada RFID *Reader*. Saat ini RFID *Tag* bisa dibuat dengan ukuran yang sangat kecil, dan tercatat yang paling kecil adalah RFID *TAG* buatan HITACHI yang berukuran  $0.05\text{mm} \times 0.05\text{mm}$ .

#### **b. RFID Reader**

RFID *Reader* adalah merupakan alat pembaca RFID *Tag*. Ada dua macam RFID *Reader* yaitu *Reader* Pasif dan *Reader* Aktif.

*Reader* Pasif memiliki sistem pembaca pasif yang hanya menerima sinyal radio dari RFID *Tag* Aktif (yang dioperasikan dengan baterai/sumber daya). Jangkauan penerima RFID Pasif bisa mencapai 600 meter. Hal ini memungkinkan aplikasi RFID untuk sistem perlindungan dan pengawasan aset.

*Reader* Aktif memiliki sistem pembaca aktif yang memancarkan sinyal interogator ke *Tag* dan menerima balasan autentikasi dari *Tag*. Sinyal interogator ini juga menginduksi *Tag* dan akhirnya menjadi sinyal DC yang menjadi sumber daya *Tag* Pasif.

### **2.6. Motor Servo**

Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem *closed feedback* yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (*axis*) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya

dapat dikendalikan dengan memberikan variasi lebar pulsa (*duty cycle*) sinyal PWM pada bagian pin kontrolnya.

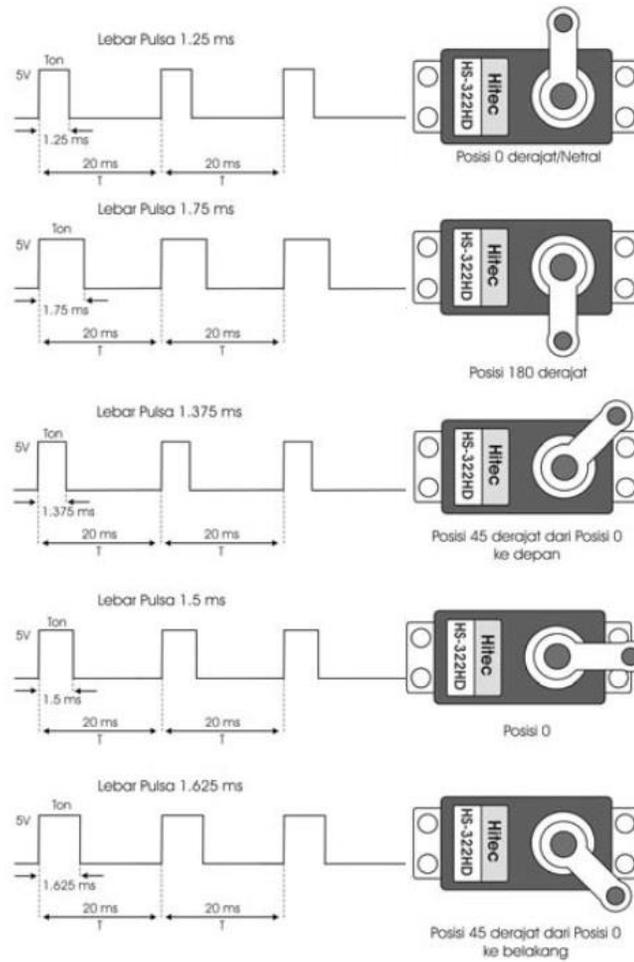


**Gambar 2.15 Bentuk Fisik Motor Servo**

(Sumber : <http://elektronika-dasar.web.id/>, diakses pada tanggal 01 April 2017, 18:14)

Motor servo dibagi menjadi 2 jenis, yaitu Motor Servo Standart 180° dan Motor Servo *Continuous*. Motor Servo Standart 180°, Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°. Motor Servo *Continuous*, Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

Operasional motor servo dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan 90°, maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati 0° dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati 180°.



**Gambar 2.16 Pulsa Kendali Motor Servo**

(Sumber : <http://elektronika-dasar.web.id/>, diakses pada tanggal 01 April 2017, 18:14)