# BAB II

# TINJAUAN PUSTAKA

## 2.1. Mikrokontroler

### 2.1.1. Pengertian Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti *processor*, memori (sejumlah kecil *Random Access Memory (RAM)*, memori program, atau keduanya), dan perlengkapan *input output*.

Mikrokontroler merupakan sebuah prosesor yang digunakan untuk kepentingan kontrol. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan *computer mainframe*, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri.

Beberapa fitur yang umumnya ada di dalam mikrokontroler adalah sebagai berikut :

*1. Random Access Memory (RAM)*

*RAM* digunakan oleh mikrokontroller untuk tempat penyimpanan variabel. Memori ini bersifat *volatile* yang berarti akan kehilangan semua datanya jika tidak mendapatkan catu daya.

2. *Read Only Memory (ROM)*

   *ROM* seringkali disebut sebagai kode memori karena berfungsi untuk tempat penyimpanan program yang akan diberikan oleh *user*.

3. *Register*

   Merupakan tempat penyimpanan nilai-nilai yang akan digunakan dalam proses yang telah disediakan oleh mikrokontroler.

4. *Special Function Register*

   Merupakan *register* khusus yang berfungsi untuk mengatur jalannya mikrokontroler. *Register* ini terletak pada *RAM*.

5. *Input* dan *Output*

   *Pininput* adalah bagian yang berfungsi sebagai penerima signal dari luar, *pin* ini dapat dihubungkan ke berbagai media *input*an seperti *keypad*, sensor, dan sebagainya. *Pin output* adalah bagian yang berfungsi untuk mengeluarkan signal dari hasil proses algoritma mikrokontroler.

6. *Interrupt*

   *Interrupt* bagian dari mikrokontroler yang berfungsi sebagai bagian yang dapat melakukan interupsi, sehingga ketika program utama sedang berjalan, program utama tersebut dapat diinterupsi dan menjalankan program interupsi terlebih dahulu.

Beberapa *interrupt* pada umumnya adalah sebagai berikut :

- *Interrupt Eksternal*

  *Interrupt* akan terjadi bila ada *input*an dari *pin interrupt.*

- *Interrupt timer*

  *Interrupt* akan terjadi bila waktu tertentu telah tercapai.

- *Interrupt serial*

  *Interrupt* yang terjadi ketika ada penerimaan data dari komunikasi serial.

### 2.1.2. Mikrokontroler ATMega 328

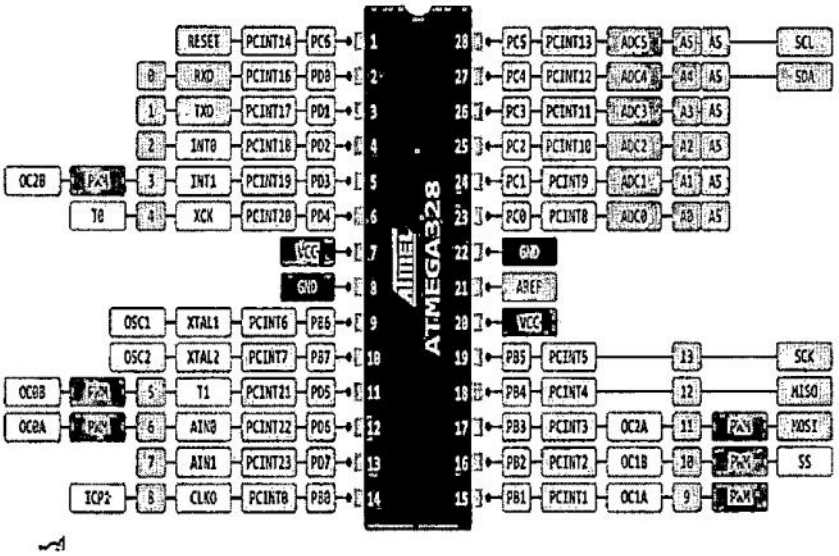
ATMega 328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur *Reduce Instruction Set Computer (RISC)* yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur *Completed Instruction Set Computer (CISC).* Mikrokontroler ini memiliki beberapa fitur antara lain :

- 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock.*
- 32 x 8-*bit register* serba guna.
- Kecepatan mencapai 16 *MIPS* dengan *clock* 16 *MHz.*
- 32 KB *Flash memory* dan pada *Arduino*memiliki *bootloader* yang dinggunakan 2 KB dari *flash memory* sebagai *bootloader.*
- Memiliki *Electrically Erasable Programmable Read Only Memory (EEPROM)* sebesar 1 KB sebagai tempat penyimpanan

data semi permanent karena *EEPROM* tetap dapat menyimpan data meskipun catu daya dimatikan.

- Memiliki *Static Random Access Memory (SRAM)* sebesar 2KB.
- Memiliki *pin I/O* digital sebanyak 14 pin 6 diantaranya *Pulse Width Modulation (PWM) output.*
- *Master/Slave SPI Serial interface.*

Konfigurasi dari *PIN* ATMega 328 adalah seperti yang ditunjukkan pada Gambar 2.1.




Gambar 2.1. Konfigurasi *Pin*ATMega 328

Pada Gambar 2.1. adalah konfigurasi dari *pin* ATmega 328 yang terdiri dari 3 buah *PORT* utama yaitu *PORTB, PORTC,* dan *PORTD* dengan total *pin input/output* sebanyak 23 *pin. PORT* tersebut dapat difungsikan sebagai *input/output digital* atau difungsikan sebagai periperal lainnya.

1. *Port* B (PB0-PB7)

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

   - *ICP1* (PB0), berfungsi sebagai *Timer Counter 1 input capture pin.*
   - OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran *PWM (Pulse Width Modulation).*
   - *MOSI* (PB3), *MISO* (PB4), *SCK* (PB5), *SS* (PB2) merupakan jalur komunikasi *SPI.*
   - Selain itu *pin* ini juga berfungsi sebagai jalur pemograman serial *(ISP).*
   - *TOSC1* (PB6) dan *TOSC2* (PB7) dapat difungsikan sebagai sumber *clock external* untuk *timer.*
   - *XTAL*1 (PB6) dan *XTAL*2 (PB7) merupakan sumber *clock* utama mikrokontroler.

2. *Port* C (PC0-PC5)

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

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

3. PC6/*RESET*

   Jika *Fuse RSTDISBL* diprogram, maka PC6 berfungsi sebagai *pin I/O* akan tetapi dengan karakteristik yang berbeda dengan PC5..PC0. Jika *Fuse RSTDISBL* tidak diprogram, maka PC6 berfungsi sebagai masukan *Reset.* Sinyal *LOW* pada *pin* ini dengan lebar minimum 1,5 mikrodetik akan membawa mikrokontroler ke kondisi *Reset,* meskipun *clock* tidak *running.*

4. *Port* D (PD0-PD7)

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

   - *USART* (*TXD* dan *RXD*) merupakan jalur data komunikasi serial dengan level sinyal *TTL. Pin TXD* berfungsi untuk mengirimkan data serial, sedangkan *RXD* kebalikannya yaitu sebagai *pin* yang berfungsi untuk menerima data serial.

- *Interrupt* (*INT0* dan *INT1*) merupakan *pin* dengan fungsi khusus sebagai interupsi *hardware*. Interupsi biasanya digunakan sebagai selaan dari program, misalkan pada saat program berjalan kemudian terjadi interupsi *hardware/software* maka program utama akan berhenti dan akan menjalankan program interupsi.
- *XCK* dapat difungsikan sebagai sumber *clock external* untuk *USART*, namun kita juga dapat memanfaatkan *clock* dari *CPU*, sehingga tidak perlu membutuhkan *external clock*.
- T0 dan T1 berfungsi sebagai masukan *counter external* untuk *timer 1* dan *timer 0*.
- AIN0 dan AIN1 keduanya merupakan masukan input untuk *analog comparator*.

5. *RESET*

   *Pin* masukan *Reset*. Sinyal *LOW* pada *pin* ini dengan lebar minimum 1,5 mikrodetik akan membawa mikrokontroler ke kondisi *Reset*, meskipun *clock* tidak *running*. Sinyal dengan lebar kurang dari 1,5 mikrodetik tidak menjamin terjadinya kondisi *Reset*.

6. *AVCC*

   *AVCC* adalah *pin* suplai tegangan untuk *ADC*, PC3..PC0, dan *ADC7..ADC6*. *Pin* ini harus dihubungkan dengan *VCC*, meskipun *ADC* tidak digunakan. Jika *ADC* digunakan, *VCC* harus

dihubungkan ke *AVCC* melalui *low-pass filter* untuk mengurangi *noise*.

7. *AREF*

*Pin* Analog *Reference* untuk *ADC*. [9]

## 2.2. *Arduino UNO*

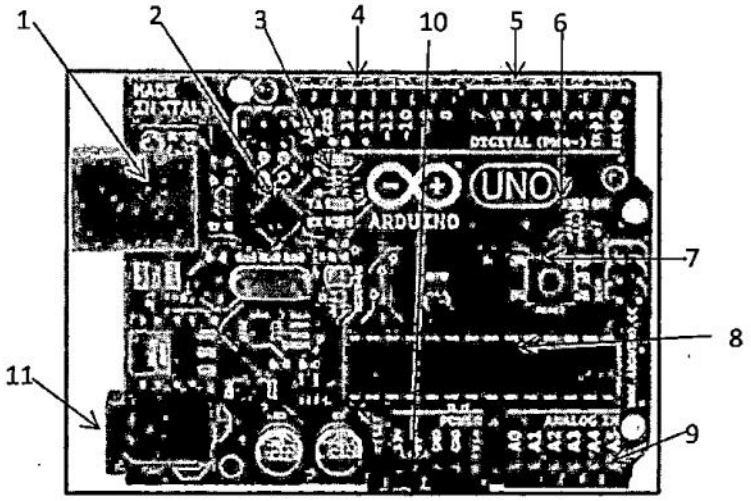
*Arduino UNO* merupakan *board Arduinorevisi* terbaru yang merupakan penerus dari *Arduino Duemilanove*. Yang membedakan antara *Arduino Uno* dan *Arduino duemilanove* yaitu tidak lagi digunakannya *chip FTDI (USB to Serial driver)* dan sebagai gantinya menggunakan ATMega8U2 yang diprogramkan untuk berfungsi sebagai *konverter USB to Serial*. Perubahan ini cukup membantu dalam instalasi *software Arduino*, terutama bagi anda yang memakai sistem operasi *windows*, karena tidak perlu menginstal *driver FTDI* untuk menghubungkan *board Arduino Uno* dengan *windows*.

*Arduino UNO* merupakan *board* mikrokontroler berbasis ATMega 328 yang memiliki 14 *pin input* dari *output* digital dimana 6 *pin input* tersebut dapat digunakan sebagai *output PWM* dan 6 *pin input* analog, 16*MHz* osilator kristal, koneksi *USB*, *jack power*, *ICSP header* dan tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan cukup hanya menghubungkan *board Arduino Uno* ke komputer dengan menggunakan kabel *USB* atau listrik dengan *AC (Alternative Current)* yang ke adaptor *DC (Direct Current)* atau baterai untuk menjalankannya.

### 2.2.1. *Hardware*

Papan *Arduino UNO* merupakan papan mikrokontroler yang berukuran kecil atau dapat diartikan juga dengan suatu rangkaian berukuran kecil yang didalamnya terdapat komputer berbentuk suatu *chip* yang kecil.

Pada Gambar 2.2. di bawah ini dapat dilihat sebuah papan *Arduino UNO* dengan beberapa bagian komponen didalamnya.




Gambar 2.2. *Board Arduino UNO*

Tabel 1.1. Bagian-Bagian Komponen *Arduino UNO*

| No | Keterangan |
|---|---|
| 1 | *Port* USB |
| 2 | *IC* Konverter Serial-*USB* |
| 3 | *LED* untuk tes*output* kaki |
| 4 | Kaki-Kaki *Input Output* Digital (D8-D13) |
| 5 | Kaki-Kaki *Input Output* Digital (D0-D7) |
| 6 | *LED* Indikator catu daya |
| 7 | Tombol *Reset* |

| | |
|---|---|
| 7 | Tombol *Reset* |
| 8 | Mikrokontroler ATMega328 |
| 9 | Kaki-Kaki *Input* Analog (A0-A5) |
| 10 | Kaki-Kaki *catu daya* (5*Volt*, *GND*) |
| 11 | Terminal *catu daya* (6-9*Volt*) |

Tabel 2.2. Ringkasan *Arduino UNO*

| Mikrokontroler | ATMega 328 |
|---|---|
| Tegangan pengoperasian | 5*Volt* |
| Tegangan *input* yang disarankan | 7-12*Volt* |
| Batas tegangan *input* | 6-20*Volt* |
| Jumlah *pin I/O* digital | 14 (6 di antaranya menyediakan keluaran *PWM*) |
| Jumlah *pininput* analog | 6 |
| Arus *DC* tiap *pin I/O* | 40 *mA* |
| Arus *DC* untuk *pin* 3.3*Voolt* | 50 *mA* |
| Memori *Flash* | 32 KB (ATmega328), sekitar 0.5 KB digunakan oleh *bootloader* |
| *SRAM* | 2*KB* (ATMega328) |
| *EEPROM* | 1 KB (ATMega328) |
| *ClockSpeed* | 16 MHz |

### 2.2.2. *Software Arduino*

*Software Arduino UNO* yang digunakan adalah *driver* dan *Integrated Development Environment (IDE)*, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan *Arduino. Integrated Development Environment (IDE)* suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan *Arduino UNO. IDE Erduino UNO*

merupakan *software* yang sangat canggih ditulis dengan menggunakan java. *IDE Arduino* terdiri dari:

1) *Editor Program*

Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

2) *Compiler*

Sebuah modul yang mengubah kode program menjadi kode biner bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.

3) *Uploader*

Sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan *Arduino*

Dalam bahasa pemrograman *Arduino* ada tiga bagian utama yaitu struktur, *variabel* dan fungsi.

1) Struktur Program *Arduino*

- Kerangka Program

  Kerangka program *Arduino* sangat sederhana, yaitu terdiri atas dua blok. Blok pertama adalah *void setup*() dan blok kedua adalah *void loop*().

  a. *Blok Void setup* ()

     Berisi kode program yang hanya dijalankan sekali sesaat setelah *Arduino* dihidupkan atau di-*reset*. Merupakan bagian persiapan atau instalasi program.

b. *Blok void loop()*

Berisi kode program yang akan dijalankan terus menerus. Merupakan tempat untuk program utama.

- Sintaks Program

Baik blok *void setup loop* () maupun blok *function* harus diberi tanda kurung kurawal buka "{" sebagai tanda awal program di blok itu dan kurung kurawal tutup "}" sebagai tanda akhir program seperti yang di tunjukkan pada Gambaar 2.3.

```
void setup(){
   // perintah-perintah untuk
      konfigurasi dan inisialisasi
      arduino board
 }

void loop(){
   // perintah - perintah utama
      arduino board
 }
```

Gambar 2.3. Struktur Program *Arduino UNO*

2) Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas dengan menggunakan sebuah varibel, separti yang ditinjukkan pada Gambar 2.4.

```
Type variableName = 0;
```

Gambar 2.4. Contoh Variabel *Arduino*

3) Fungsi

Pada bagian ini meliputi fungsi *input outputdigital*, *input output* analog, *advanced I/O*, fungsi waktu, fungsi matematika serta fungsi komunikasi, separti yang ditunjukkan pada Gambar 2.5.

```
Type functionName (parameters)
            {
                  // Statement;
            }
```

Gambar 2.5. Contoh Fungsi *Arduino*

- *Digital I/O*

  *Digital input* memberikan dua kondisi sinyal masukan yaitu tombol tertekan atau tombol tidak tertekan. Pada saat tombol tertekan akan memberikan tegangan 5 atau 9 *Volt* pada masukkan sedangkan sebaliknya pada saat tombol dilepas hanya memberikan tegangan 0 *Volt*. Kondisi ini disebut dengan *digital input* dengan logika 1 dan 0, dimana 1 itu untuk tegangan 5 *Volt* dan 0 untuk tegangan 0 *Volt*. Begitu juga halnya pada sisi *output*, jika hanya melibatkan dua kondisi keluaran seperti menghidupkan dan memadamkan suatu sensor jika diberi tegangan ini disebut

dengan digital *output* dimana digital 1 untuk yang diberi tegangan dan digital 0 dengan *output* tegangan 0 *Volt.*

a. *pinMode (pin, mode)*

Digunakan dalam *void setup()* untuk mengkonfigurasi *pin* apakah sebagai *Input* atau *Output. Arduino* digital *pins* secara default di konfigurasi sebagai *input* sehingga untuk merubahnya harus menggunakan *operator pinMode(pin, mode).*(lihat Gambar 2.6.)

Parameter :

✓ *pin* = angka dari *pindigital* yang akan dikonfigurasi.

✓ *Mode* = konfigurasi yang diinginkan *(Input, Input_Pullup* dan *Output).*

```
pinMode (pin, Output);      // mengset pin
sebagai output
digitalWrite (pin, High);   // pin sebagai
source voltage
```

Gambar 2.6. Contoh *pinMode*

b. *digitalRead (pin)*

Membaca nilai dari *pin* yang kita kehendaki dengan hasil *High* atau *Low*. Parameter*: pin* = angka dari *pin digital* yang akan dibaca. Gambar 2.7. adalah contoh penggunaan fungsi masukan dan keluaran digital dalam sebuah program.

```
int ledPin = 13; // LED terhubung ke pin digital 13
int inPin = 7; // pushbutton terhubung ke pin digital 7
int val = 0; // variable untuk menyimpan sebuah nilai
void setup() {
pinMode(ledPin, OUTPUT); // set pin digital 13 sebagai
keluaran
pinMode(inPin, INPUT); // set pin digital 13 sebagai masukan
}
void loop(){
val = digitalRead(inPin); // baca nilai pin input
digitalWrite(ledPin, val); // set LED sesuai dengan nilai val
}
```

Gambar 2.7. Contoh *I/O digitalRead Arduino*

c. *digitalWrite (pin, value)*

Digunakan untuk mengset *pin digital. Pindigital Arduino* mempunyai 14 ( 0 – 13 ). Sebagai contoh dalah seperti Gambar 2.8.

```
digitalWrite (pin, High);   // set
pin to High
```

Gambar 2.8. Contoh *I/O digitalWrite Arduino*

- *Analog I/O*

*Analog input output* merupakan pengolahan *input* dan *output* secara *digital* mungkin sudah memenuhi suatu kebutuhan, akan tetapi pada kondisi tertentu ada kemungkinan yang dihadapi pada kondisi *input* dan *output* yang membutuhkan besaran yang berubah-berubah dengan nilai yang kontinyu dan tidak lagi hanya dengan dua

keadaan seperti sinyal *digital*. *Arduino* dalam *analog I/O* digunakan sebagai kontroler yang mampu mengolah semua variasi tegangan keluaran dari sensor yang dihubungkan pada *pin input*nya. Perintah yang digunakan dalam *analog I/O* ini adalah:

a. *analogRead (pin)*

Membaca nilai *pin analog* yang memiliki resolusi 10 *bit*. Fungsi ini hanya dapat bekerja pada *analog pin* (0-5). Hasil dari pembacaan berupa nilai integer dengan *range* 0 sampai 1023. Contoh:

```
Value = analogRead(pin);    //      mengset
'value' sama dengan nilai analog pin
```

Gambar 2.9. Contoh *analogRead*

b. *analogWrite (pin, value)*

Mengirimkan nilai analog pada *pin analog*.

```
analogWrite(pin, value);    // menulis
ke pin analog
```

Gambar 2.10. Contoh *analogWrite*

### 2.2.3. *Power Arduino*

*Arduino* dapat diberikan *power* melalui koneksi *USB* atau *power supply*. *Power*nya di*select* secara otomatis. *Power supply* dapat menggunakan adaptor *DC* atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack* adaptor pada koneksi *portinput supply*. *Board Arduino* dapat dioperasikan menggunakan *supply* dari luar sebesar 6 -

20 *Volt*. Jika *supply* kurang dari 7 *Volt*, kadangkala *pin* 5 *Volt* akan menyuplai kurang dari 5 *Volt* dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 *Volt*, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 *Volt* sampai 12 *Volt*.

Penjelasan pada *pin power* adalah sebagai berikut:

- *Vin*

  Tegangan *input* ke *board Arduino* ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 *Volt* dari koneksi *USB* atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui *pin* ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan *pin* ini.

- 5 *Volt*

  Regulasi *power supply* digunakan untuk *power* mikrokontroler dan komponen lainnya pada *board*. 5 *Volt* dapat melalui *Vin* menggunakan *regulator* pada *board*, atau *supply* oleh *USB* atau *supply* regulasi 5 *Volt* lainnya.

- 3.3 *Volt*

  Suplai 3.3 *Volt* didapat oleh *FTDIc hip* yang ada di *board*, arus maksimumnya adalah 50 *mA*.

- *Pin Ground*

  Berfungsi sebagai jalur *ground* pada *Arduino*.

### 2.2.4. *Memory*

ATMega 328 memiliki 32 KB *flash* memori untuk menyimpan kode, juga 2 KB yang digunakan untuk *bootloader*. ATMega 328 memiliki 2 KB untuk *SRAM* dan 1 KB untuk *EEPROM*.

### 2.2.5. *Input/Output*

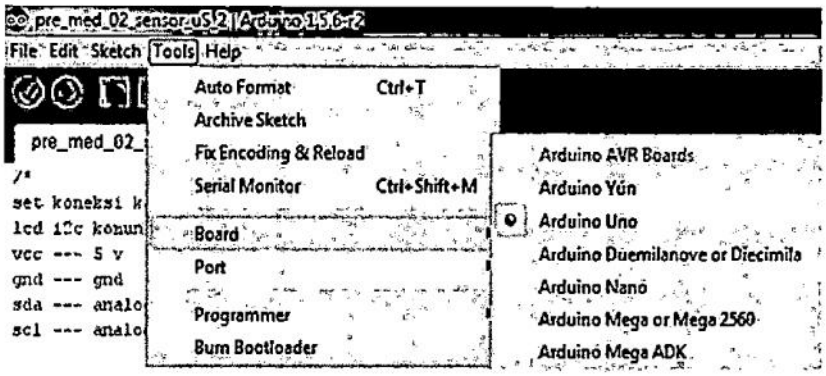
Setiap 14 *pin* digital pada *Arduino* dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode(), digitalWrite(),* dan *digitalRead(). Input/output* dioperasikan pada 5 *Volt.* Setiap *pin* dapat menghasilkan atau menerima maximum 40 *mA* dan memiliki *internal pull-upresistor* (*disconnected* oleh *default*) 20-50 *KOhm.* Beberapa *pin* memiliki fungsi sebagai berikut :

- Serial : 0 *(RX)* dan 1 *(TX).* Digunakan untuk menerima *(RX)* dan mengirim *(TX)TTL* data serial. *Pin* ini terhubung pada *pin* yang *koresponding* dari *USB FTDI* ke *TTL chip* serial.
- *Interupt eksternal* : 2 dan 3. *Pin* ini dapat dikonfigurasikan untuk *trigger* sebuah *interupt* pada *low value, rising.* atau *falling edge,* atau perubahan nilai.
- *PWM* : 3, 5, 6, 9, 10, dan 11. Mendukung 8-*bit* keluaran *PWM* dengan fungsi *analogWrite().*
- *SPI* : 10 *(SS),* 11 *(MOSI),* 12 *(MISO),* 13 *(SCK). Pin* ini men-*support* komunikasi *SPI,* yang mana masih mendukung *hardware,* yang tidak termasuk pada bahasa *Arduino.*

- *LED* : 13. Ini adalah dibuat untuk koneksi *LED* ke digital *pin* 13. Ketika *pin* bernilai *HIGH*, *LED* hidup, ketika *pin LOW*, *LED* mati.

### 2.2.6. Cara *Upload* Program

Setelah program yang disusun benar dan berhasil *compiling*, sebelumnya *Arduino Uno* tetap dalam kondisi tersambung/*connect* pada *USB Arduino.* Memilih menu *Tools* selanjutnya klik *Board* kemudian *Arduino UNO* untuk karena memakai mikrokontroler ATMega 328. Selanjutnya untuk meng-*upload* programnya pilih menu *File* kemudian *Upload* atau pada *menu bar* terdapat simbol panah kanan, tunggu sampai *Done Uploading.* Setelah sukses, maka program yang dibuat sudah terisi pada mikrokontroler yang digunakan.

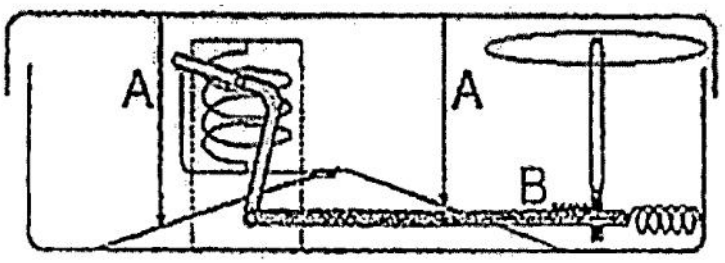



Gambar 2.11. Menentukan *Board* Yang Digunakan

## 2.3. Timbangan Badan Mekanik

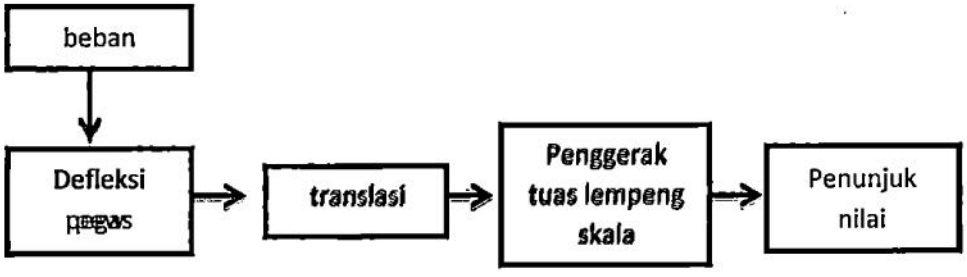
Timbangan mekanik adalah alat ukur yang digunakan untuk mengukur berat badan seseorang/pasien. Penimbangan berat badan biasanya dilakukan dengan cara menempatkan pasien dengan posisi berdiri diatas timbangan.

Timbangan analog pada bagian bawahnya memiliki kontruksi mekanik seperti pada Gambar 2.12.




Gambar 2.12. Konstruksi Timbangan Mekanik

Secara sederhana prinsip kerja mekanik timbangan analog yaitu beban yang berada diatas timbangan akan diteruskan melalui penyangga A menuju plat tumpuan yang dihubungkan dengan pegas timbangan. Hal ini akan menyebabkan pegas dan ujung plat tumpuan akan tertarik ke bawah. Perubahan ini akan diteruskan oleh penggerak B sebagai gerakan translasi (ke kanan) yang selanjutnya akan menggerakkan tuas lempeng skala, sehingga lempeng skala akan berputar dan berhenti pada posisi tepat pada garis yang menunjukkan nilai berat badan. Gambar 2.13. adalah blok diagram dari timbangan analog adalah: [5]




Gambar 2.13. Blok Diagram Timbangan Analog

## 2.4. Penampil *LCD (Liquid Cristal Display)*

*LCD (Liquid Cristal Display)* adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. Dipasaran tampilan *LCD* sudah tersedia dalam bentuk modul yaitu tampilan *LCD* beserta rangkaian pendukungnya termasuk *ROM* dan lain-lain. *LCD* mempunyai *pin data*, kontrol catu daya, dan pengatur kontras tampilan.

*LCD* merupakan perangkat *display* yang paling umum dipasangkan dengan Mikrokontroler, mengingat ukurannya yang kecil dan kemampuannya manampilkan karakter atau grafik yang lebih dibandingkan *display seven-segment*. Pada pengembangan sistem *embedded LCD* mutlak dipelukan sebagai sumber pemberi informasi utama, misalnya alat pengukur kadar gula darah, penampil jam, penampil *counter* putaran motor industri dan lainnya.

Berdasarkan jenis tampilan, *LCD* dapat dikelompokan menjadi beberapajenis,yaitu:

- *Segment LCD*

  *LCD* ini berbentuk dari beberapa *Seven-Segment Display* atau *Sixteen Segment Display*, namun ada juga yang mengabungkan keduanya. *LCD* ini sering dipakai untuk jam digital.

- *Dot Matrix Character LCD*

  *LCD* ini terbentuk dari beberapa *Dot Matrix Display* berukuran 5x7 atau 5x9 yang membentuk sebuah matriks yang lebih besar dengan berbagai

kombinasi jumlah baris dan kolom. Kombinasi ini yang menentukan karakter yang dapat ditampilkan *LCD* tersebut. Seperti 2 baris 20 karakter atau 4 baris 20 karakter.

- *Graphic LCD*

  *LCD* jenis ini masih berkembang saat ini. *Resolasi LCD* ini bervariasi, diantaranya 128x64, 128x128. Sekarang ini *Graphic LCD* banyak dipakai pada *Handycam*, laptop, telpon seluler (*cellphone*), monitor komputer dan lain sebagainya.

*Register* yang terdapat di *LCD* adalah sebagai berikut:

- *IR (Intruction Register)*

  Digunakan untuk menentukan fungsi yang harus dikerjakan oleh *LCD* serta pengalamatan *DDRAM* atau *CGRAM*.

- *DR (Data Register)*

  Digunakan sebagai tempat data *DDRAM* atau *CGRAM* yang akan ditulis atau dibaca oleh komputer atau sistem minimum. Saat dibaca, *DR* menyimpan data *DDRAM* atau *CGRAM*, setelah itu data alamatnya secara otomatis masuk ke *DR*. Pada waktu menulis, cukup lakukan inisialisasi *DDRAM* atau *CGRAM*, kemudian untuk selanjutnya data dituliskan ke *DDRAM* atau *CGRAM* sejak awal alamat tersebut.

- *BF (Busy Flag)*

  Digunakan untuk bahwa *LCD* dalam keadaan siap atau sibuk. Apabila *LCD* sedang melakukan operasi internal, *BF* diset menjadi 1, sehingga tidak akan menerima perintah dari luar. Jadi, *BF* harus dicek apakah telah

diriset menjadi 0 ketika akan menulis *LCD* (memberi data pada *LCD*). Cara untuk menulis *LCD* adalah dengan mengeset *RS* menjadi 0 dan mengeset *R/W* menjadi 1.

- *AC (Address Counter)*

  Digunakan untuk menunjukan alamat pada *DDRAM* atau *CGRAM* dibaca atau ditulis, maka *AC* secara otomatis menunjukan alamat berikutnya. Alamat yang disimpan *AC* dapat dibaca bersamaan dengan *BF*.

- *DDRAM (Display Data Random Access Memory)*

  Digunakan sebagai tempat penyimpanan data yang sebesar 80 *byte* atau 80 karakter. *AC* menunjukan alamat karakter yang sedang ditampilkan.

- *Character Generator Read Only Memory (CGROM)*

  Pada *LCD* terdapat *ROM* untuk menyimpan karakter-karakter *American Standart Code for Interchage Intruction (ASCII)*, sehingga cukup memasukan kode *ASCII* untuk menampilkanya.

- *Character Generator Random Access Memory* (*CGRAM*)

  Sebagai data *storage* untuk merancang karakter yang dikehendaki. Untuk *CGRAM* terdapat kode *ASCII* dari 00h sampai 0Fh, tetapi hanya 8 karakter yang disediakan. Alamat *CGRAM* hanya 6 *bit*, 3 *bit* untuk mengatur tinggi karakter dan 3 *bit* tinggi menjadi 3 *bit* rendah *DDRAM* yang menunjukan karakter, sedangkan 3 *bit* rendah sebagai posisi data *CGRAM* untuk membuat tampilan baris dalam dot matriks 5x7 karakter tersebut, dimulai dari atas. Sehingga karakter untuk kode *ASCII* 00 h sama dengan 09 h sampai 07 h dengan 0 Fh. Dengan demikian untuk

perancangan 1 karakter memerlukan penulisan data ke *CGRAM* sampai 8 kali.

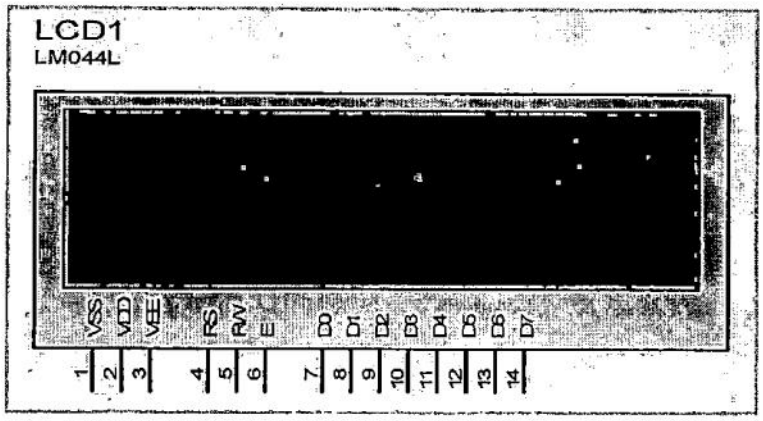
- *Cursor and Blink Control Circuit*

  Merupakan rangkaian yang menghasilkan tampilan kursor dan kondisi *blink* (berkedap-kedip).

Sebagai bahasan berikut adalah modul *LCD* 4x20 karakter yang akan digunakan dalam Tugas Akhir ini. Salah satu alasan mengapa modul *LCD* dipakai dalam Tugas Akhir ini adalah kenyataan bahwa modul *LCD* relatif jauh lebih sedikit memerlukan daya dibandingkan dengan modul-modul *display* berbasis *LED*. Selain itu desain *LCD* lebih kompak dan dimensinya juga lebih kecil. Dengan mikrokontroler kita dapat mengendalikan suatu peralatan agar dapat bekerja secara otomatis. Untuk mengakses *LCD* 4x20 harus melakukan konfigurasi *pin* dari *LCD* dengan *pin I/O* mikrokontroler tersebut. [7]

Gambar 2.14. *LCD* 20x4

Berikut adalah diskripsi *PIN* pada *LCD*




Gambar 2.15. Konfigurasi *PinLCD* 20x4.

Lcd 4x20 memiliki 16 pin dengan fungsi yang berbeda-beda, yaitu 2 *pin* tegangan 5 *volt*, 2 *pin* tegangan 0 *Volt*, 1 *pin* pengatur kontras LCD, 3 *pin* kontrol *LCD*, dan 8 *pin data bus*. Tabel 2.3. di bawah ini adalah diskripsi dan simbol dari *pin-pin* dari *LCD* 4x20.

Tabe 2.3. Konfigurasi *Pin LCD* [8]

| No | Simbol | *I/O* | Diskripsi |
|---|---|---|---|
| 1 | *VSS* | | *Ground* |
| 2 | *VCC* | | *+5 V Power Suplay* |
| 3 | *VEE* | | *Power suplay source to control contrast* |
| 4 | *RS* | | *Register select: RS = 0 to select instruksi. Command register; RS =1 to selsct data reg.* |
| 5 | *R/W* | | *Read/Write: R/W =0 for write, R/W= 1 for read* |
| 6 | *E* | | *Enable* |
| 7 | *DB 0* | | *The 8-bit data bus* |
| 8 | *DB 1* | | *The 8-bit data bus* |
| 9 | *DB 2* | | *The 8-bit data bus* |
| 10 | *DB 3* | | *The 8-bit data bus* |
| 11 | *DB 4* | | *The 8-bit data bus* |
| 12 | *DB 5* | | *The 8-bit data bus* |
| 13 | *DB 6* | | *The 8-bit data bus* |
| 14 | *DB 7* | | *The 8-bit data bus* |
| 15 | A | | *Anoda (+)* |
| 16 | K | | *Katoda (-)* |

- *Pin* 1 dan 2

  Merupakan sambungan catu daya, *Vss* dan *Vdd*. *Pin Vdd* dihubungkan dengan tegangan positif catu daya, dan *Vss* pada 0 *Volt* atau *ground*. Meskipun data menentukan catu 5 *Volt DC* (hanya pada beberapa *mA*), menyediakan 6 *Volt* dan 4,5 *Volt* yang keduanya bekerja dengan baik, bahkan 3 *Volt* cukup untuk beberapa modul. [1]

- *Pin* 3

  *Pin*3 merupakan *pin* kontrol *Vee*, yang digunakan untuk mengatur kontras *display*. Idealnya *pin* ini dihubungkan dengan tegangan yang bisa dirubah untuk memungkinkan pengaturan terhadap tingkatan kontras *display* sesuai dengan kebutuhan, *pin* ini dapat dihubungkan dengan *variable resistor* sebagai pengatur kontras.

- *Pin* 4

  *Pin* 4 merupakan *Register Select (RS)*, masukan yang pertama dari tiga *command control input*. Dengan membuat *RS* menjadi *high*, data karakter dapat ditransfer dari dan menuju modulnya.

- *Pin* 5

  *Read/Write (R/W)*, untuk memfungsikan sebagai perintah *write* maka *R/W low* atau menulis karakter ke modul. *R/W high* untuk membaca data karakter atau informasi status dari *register*-nya.

- *Pin* 6

  *Enable (E), input* ini digunakan untuk transfer aktual dari perintah-perintah atau karakter antara modul dengan hubungan data. Ketika

menulis ke *display*, data ditransfer hanya pada perpindahan *high* atau *low*. Tetapi ketika membaca dari *display*, data akan menjadi lebih cepat tersedia setelah perpindahan dari *low* ke *high* dan tetap tersedia hingga sinyal *low* lagi.

- *Pin* 7-14

  *Pin* 7 sampai 14 adalah delapan jalur data/*data bus* (D0 sampai D7) dimana data dapat ditransfer ke dan dari *display*.

- *Pin*15 dan 16 (*Anoda* dan *Katoda*)

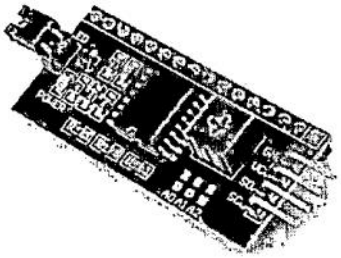
  *Pin* 16 dihubungkan ke *ground* dan *Pin*15 dihubungkan kedalam tegangan 5 *Volt* untuk memberi tegangan dan menghidupkan lampu latar/*Back Light LCD*. [1]

## 2.5. Potensiometer Geser

Potensiometer geser merupakan kembaran dari potensiometer. Perbedaannya adalah cara mengubah nilai resistansinya. Pada potensiometer, cara mengubah nilai resistansinya adalah dengan cara memutar gagang yang muncul keluar. Sedangkan, untuk potensiometer geser, cara mengubah nilai resistansinya adalah dengan cara menggeser gagang yang muncul keluar. Bentuk dari potensiometer geser dapat dilihat pada Gambar16. Pada umumnya, bahan yang digunakan untuk membuat potensiometer ini adalah karbon. Adapula yang terbuat dari kawat, namun saat ini sudah jarang digunakan karena ukurannya yang besar. Pada potensiometer geser ini, perubahan nilai resistansinya hanyalah perubahan secara linier.

memperhatikan *timing* dari seluruh komponen yang terlibat, agar tidak terjadi kesalahan dalam transaksi data. *Bus* yang cukup sering digunakan adalah *bus* bersifat paralel. Transaksi data dilakukan secara paralel sehingga transaksi data lebih cepat, akan tetapi disisi lain mahal. Jika sistem relatif tidak membutuhkan transaksi yang cepat, maka penggunaan Serial *Bus* menjadi pilihan. Salah satu pilihan sistem data bus yang sering digunakan adalah *Inter Integrated Circuit (I2C)*. Sistem *Bus I2C* pertamakali diperkenalkan oleh Firma Philips.

*I2C* merupakan bus standar yang didesain oleh Firma Philips pada awal tahun 1980-an untuk memudahkan komunikasi antar komponen yang tersebar pada papan rangkaian. *I2C* merupakan singkatan dari *Inter IC* atau komunikasi antar *IC*, sering disebut juga *IIC* atau *I2C*. Pada awalnya, kecepatan komunikasi maksimumnya diset pada 100 *kbps* karena pada awalnya kecepatan tinggi belum dibutuhkan pada transmisi data. Untuk yang membutuhkan kecepatan tinggi, ada mode 400 *kbps* dan sejak 1998 ada *mode* kecepatan tinggi 3,4 *Mbps*. *I2C* tidak hanya digunakan pada komponen yang terletak pada satu *board*, tetapi juga digunakan untuk mengkoneksikan komponen yang dihubungkan melalui kabel.



Gambar 2.17. *I2C LCD*

Karakter *I2C* :

1. *Serial Bus*

   Data dikirim serial secara per-*bit.*

2. Menggunakan dua penghantar koneksi dengan ground bersama. *I2C* terdiri dari dua penghantar :

   - *SCL (Serial Clock Line)* untuk menghantarkan sinyal *clock.*
   - *SDA (Serial Data)* untuk mentransaksikan data.

3. Jumlah Peserta *Bus* maksimal 127.

   Peserta dialamatkan melalui 7-*bit* alamat. Alamat ditetapkan kebanyakan secara hardware dan hanya sebagian kecil dapat dirubah.

4. Pengirim dan Penerima

   Setiap transaksi data terjadi antara pengirim *(Transmitter)* dan penerima *(Receiver).* Pengirim dan penerima adalah peserta bus.

5. *Master and Slave*

   *Device* yang mengendalikan operasi transfer disebut *Master,* sementara *device* yang di kendalikan oleh *Master* disebut *Slave.*

## 2.7. Komunikasi dengan *Bluetooth*

*Bluetooth* adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 *GHz unlicensed ISM* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host-host Bluetooth* dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter).

*ISM* adalah *band (Industrial Scientific and Medical)* bebas lisensi yang ditetapkan oleh *Federal Comition Comunicaton (FCC). FCC* menetapkan undang-undang yang mengatur pengoperasian piranti *LAN* nirkabel (tanpa kabel). *Band ISM* berada pada lokasi mulai dari 902 *MHz*, 2.4 *GHz* dan 5.8 *GHz* dengan lebar yang bervariasi dari sekitar 26 *MHz* hingga 150 *MHz*.

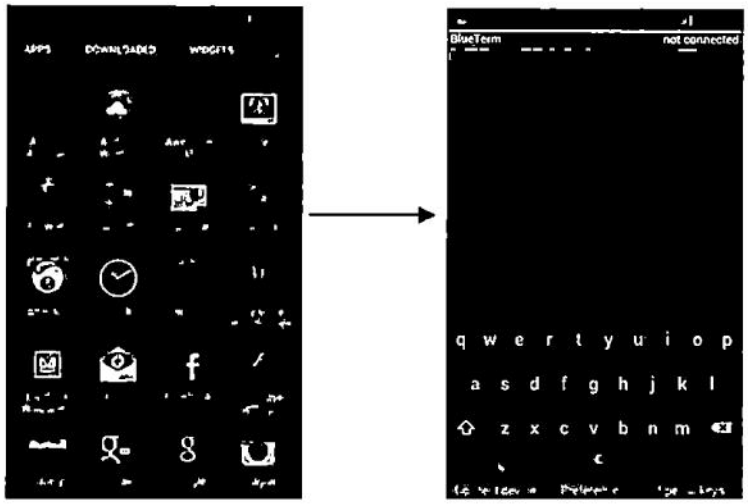
Spesifikasi *Bluetooth* menyediakan definisi link *layer* dan *application layer* sehingga mendukung aplikasi data dan suara. Teknologi *Bluetooth* juga dapat menembus benda padat dan bersifat *omni directional* sehingga tidak memerlukan posisi *line of sight* seperti pada infrared. Keamanan merupakan prioritas utama dalam pengembangan spesifikasi *Bluetooth.*

*Bluetooth* sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan *card* yang digunakan untuk *wireless local area network (WLAN)* dimana menggunakan frekuensi radio standar *IEEE* 802.11, hanya saja pada *Bluetooth*mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah. Berikut gambaran secara fisik dari *Bluetooth*:

*Android* karena operasi sistem *Android* merupakan *OS* yang sudah *Open Source* dimana pembuat *OS Android* memperbolehkan para pengguna untuk mengubah serta membuat sendiri aplikasi untuk *OS* tersebut tanpa lisensi dari pemilik *OS* selain itu *OS Android* juga memiliki tampilan yang cenderung memudahkan pengguna dalam pemakaiannya dan memiliki banyak aplikasi yang tersedia di market *Smartphone Android* tersebut dan dapat di *download* secara gratis, seperti halnya aplikasi *BlueTerm* yang kami gunakan sebagai media untak melihat hasil pengukuran yang ditampilkan pada *LCD*. [10]

## 2.9. Aplikasi *Blue Term*

Aplikasi *Blue Term* digunakan sebagai penampil melihat hasil pengukuran yang ditampilkan pada *LCD* dari *smartphone*. Aplikasi ini menggunakan *Bluetooth* sebagai media komunikasi. *Blue Term* dapat di*download* di *google play*/market pada *Smartphone Android*. Gambar 2.19. adalah tampilan dari aplikasi *Blue Term*.




Gamabar 2.19. Logo dan tampilan dalam *Blue Term*

*Blue Term* ini dapat diinstal disemua *Smartphone* beroperasi *system Android, download* di *google play* secara gratis dan sangat mudah dalam penggunaannya. [10]

## 2.10. Kalibrasi

Secara umum kalibrasi mempunyai pengertian sebagai rangkaian kegiatan membandingkan hasil pengukuran suatu alat dengan alat standar yang sesuai untuk menentukan besarnya koreksi pengukuran alat serta kesalahan relatifnya. Dalam pengertian ini alat standar yang digunakan juga harus terkalibrasi dibuktikan dengan sertifikat kalibrasi. Dengan demikian maka besarnya koreksi pengukuran alat dapat ditelusurkan ke standar nasional atau standar internasional dengan suatu mata rantai kegiatan kalibrasi yang tidak terputus.

Alat ukur yang telah dikalibrasi tidak akan secara terus menerus berlaku masa kalibrasinya, karena peralatan tersebut selama masa penggunaanya pasti mengalami perubahan spesifikasi akibat pengaruh frekuensi pemakaian, lingkungan penyimpanan, cara pemakaian, dan sebagainya. Untuk itulah selama berlakunya masa kalibrasi alat tersebut perlu dipelihara dengan baik yaitu dengan cara perawatan secara periodik.

Rangkaian kegiatan kalibrasi secara sederhana dapat digambarkan sebagai kegiatan persiapan kalibrasi, pelaksanaan kalibrasi, perhitungan data kalibrasi, penentuan ketidakpastian dan penerbitan laporan kalibrasi.

Persiapan yang dilakakan dalam kalibrasi antara lain :

1. Persiapan alat standar dan alat yang akan dikalibrasi

   Alat yang akan dikalibrasi dan alat standar dikondisikan pada kondisi yang sama sesuai metode kalibrasi, hal ini diperlukan untuk menghindarkan perbedaan hasil ukur akibat pengaruh lingkungan.

2. Metode kalibrasi

   Metode kalibrasi dapat mengacu kepada metode standar internasional maupun metode standar lainnya misalnya *text book*, jurnal, buletin, dan manual peralatan, namun perlu diperhatikan bahwa acuan tersebut harus merupakan publikasi yang diakui masyarakat luas. Selain itu dari beberapa pilihan metode kalibrasi dapat dipilih metode yang mudah dilaksanakan, karena sulitnya mengikuti metode kalibrasi dapat berakibat kesalahan dalam pengambilan data kalibrasi.

Hal-hal yang harus dilakukan saat pelaksanaan kalibrasi adalah :

1. Pengamatan awal

   Jika alat yang dikalibrasi berupa instrumen, memastikan bahwa alat tersebut dapat beroperasi normal. Jika alat berupa objek ukur memastikan bahwa alat mempunyai bentuk sempurna. Pada prinsipnya pelaksanaan kalibrasi tidak bertujuan untuk memperbaiki alat, karenanya alat yang tidak normal seyogyanya tidak boleh dikalibrasi. Alat demikian harus diperbaiki dulu oleh petugas yang khusus menangani perbaikan alat hingga alat tersebut diyakini beroperasi normal

2. Penyetelan

Penyetelan alat yang akan dikalibrasi biasanya dipertukan untuk mengbindari kesalahan titik nol. Penyetelan dapat berupa menyetel kedataran, pembersihan alat dari kotoran, menyetel titik nol, dalam hal misalnya kalibrasi neraca elektronik penyetelan dapat berupa kalibrasi internal sesuai prosedur dalam manual.

3. Pengamatan kewajaran hasil ukur

Pengamatan ini dimaksudkan untuk memastikan kewajaran penunjukan alat. Jika alat menunjukan hasil ukur yang tidak wajar mungkin perlu penyetelan kembali atau perlu dicari penyebab ketidakwajaran penunjukan alat tersebut.

4. Pengukuran

Pengukuran dilakukan pada titik ukur tertentu sepert dinyatakan dalam dokumen acuan kalibrasi sesuai kapasitas alat atau rentang ukur tertentu yang biasa digunakan oleh penggunaa alat. Jika dokumen acuan kalibrasi tidak menyatakan titik ukur, biasanya pengukuran dilakukan dalam selang 10% dari kapasitas ukur alat. Titik ukur harus dibuat mudah dibaca oleh penggunaa alat. Pada waktu pengukuran hanya melakukan pengambilan data dan tidak boleh melakukan kegiatan lainnya yang mungkin menyebabkan pembacaan atau pencatatan menjadi salah.

5. Pencatatan

Pencatatan hasil ukur harus berdasar kepada apa yang dilihat bukan kepada apa yang diraskan. Pencatatan dilakukan secara objektif dengan

menggunakan format yang telah dirancang dengan teliti sesuai dengan ketentuan metode kalibrasi. Selain data ukur hal yang perlu dicatat adalah identitas alat selengkapnya serta faktor yang mempengaruhi kalibrasi seperti suhu ruangan, kelembaban, tekanan udara dan sebagainya.

6. Perhitungan

Data kalibrasi yang diperoleh dihitung sesuai metode kalibrasi. Perhitungan biasanya melibatkan pekerjaan mengkonversi satuan, menghitung nilai maksimum-minimum, nilai rata-rata, standar deviasi, atau menentukan persamaan regresi. Hasil perhitungan akan menjadi dasar dalam penarikan kesimpulan dan penentuan ketidakpastian kalibrasi. [6]

1) Rata-rata

Rata-rata dalam perkataan sehari-hari, orang sudah menafsirkan dengan rata-rata hitung. Dan arti sebenarnya adalah bilangan yang di dapat dari hasil pembagian jumlah nilai data oleh banyaknya data dalam kumpulan tersebut. Secara matematis rata-rata dapat dirumuskkan :

$$\text{Rata} - \text{Rata } (\bar{X}) = \frac{\Sigma Xi}{n} \qquad \text{.....(1)}$$

Keterangan :

$\bar{X}$ : Rata-rata

$\Sigma Xi$ : Jumlah hasil pengukuran

n : Banyaknya pengukuran yang dilakukan

Untuk mempermudah penghitungan nilai standar deviasi dapat menggunakan *Microsoft Excel* yaitu dengn rumus =AVERAGE()

2) Simpangan *(Error)*

Merupakan selisih dari rata-rata nilai terhadap masing-masing nilai yang diukur. Rumus simpangan *(Error)* adalah :

Simpangan= $\bar{X}$ - Xn .....(2)

Keterangan :

$\bar{X}$ : Rata-rata nilai yang diukur

Xn : Nilai yang diukur

3) % *Error*

Merupakan nilai prosentase dari simpangan *(Error)* terhadap nilai yang dikehendaki. Rumus % *Error* adalah :

$$\%Error = \frac{Xn - \bar{X}}{Xn} \times 100\% \quad \text{.....(3)}$$

4) Standar Deviasi

Standart Deviasi adalah suatu nilai yang menunjukan tingkat (derajat) variasi kelompok data atau ukuran standar penyimpangan dari *mean*nya. Jika Standar Deviasinya semakin kecil maka data tersebut akan semakin presisi. Secara matematis Standar Deviasi dapat dirumuskan :

$$\mathbf{SD} = \sqrt{\frac{(X1 - \bar{X})^2 + (X2 - \bar{X})^2 + ..... + (Xn - \bar{X})^2}{n-1}} \quad \text{.....(4)}$$

Keteranagn :

SD : Standar Deviasi

Xn : Nilai pengukuran

$\bar{X}$ : Rata-rata nilai

n : Banyaknya pengukuran yang dilakukan

Untuk mempermudah penghitungan nilai standar deviasi dapat menggunakan *Microsoft Excel* yaitu dengn rumus =STDEV()

5) Ua (ketidakpastian)

Merupakan nilai perkiraan hasil pengukuran yang didalamnya terdapat nilai yang benar.Secara matematis Ua (ketidakpastian) dapat dirumuskan :

$$UA = \frac{SD}{\sqrt{n}} \qquad \text{.....(5)}$$

6) U95

Adalah nilai hasil perkalian ketidak pastian dengan 2,57. Nilai 2,57 merupakan suatu ketetapan. U95 menunjukkan data yang benar adalah 95%. Secara matematis U95 dapat dirumuskan :

U95= Ua x 2,57 .....(6)