

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengenalan *Software***

*Software* atau perangkat lunak adalah program komputer yang berfungsi sebagai sarana interaksi (penghubung) antara pengguna (*user*) dan perangkat keras (*hardware*). *Software* bisa juga dikatakan sebagai "penerjemah" perintah-perintah yang dijalankan pengguna komputer untuk diteruskan atau diproses oleh perangkat keras (*Hardware*). *Software* adalah program komputer yang isi intruksinya dapat diubah dengan mudah. *Software* pada umumnya digunakan untuk mengontrol perangkat keras (yang sering disebut *device driver*), melakukan proses perhitungan, berinteraksi dengan *Software* yang lain dan lebih mendasar (seperti sistem operasi, dan bahasa pemrograman), dan lain-lain[3].

#### **2.2 Karakteristik *Software***

##### 1. Berdasarkan Perbedaan Dengan *Hardware*

Untuk karakteristik dari *software* yang dilihat dari perbedaannya dengan perangkat keras atau *hardware* yang pertama adalah pada perangkat lunak sendiri mempunyai keunikan dimana *software* tersebut tidak diproduksi secara massal. Hal ini tentu saja berbeda dengan perangkat keras atau *hardware* yang diproduksi dalam jumlah banyak. Selain itu, perangkat lunak atau *software* pun tidak mengenal istilah usang.

Hal ini dikarenakan *software* terbaik tidak akan mudah untuk usang. Hal ini tentu saja berbeda dengan perangkat keras atau *hardware* yang ternyata lebih mudah untuk usang jika digunakan dalam periode waktu yang lama. Serta poin yang ketiga adalah beberapa perangkat lunak tersebut ada yang dikembangkan melalui cara *custom*. Sedangkan untuk pengembangannya sendiri melalui *Componet Based Assembly* yang digunakan pada perangkat keras.

## 2. Karakteristik Berdasarkan Fungsinya

Selain itu, berdasarkan fungsinya software dibedakan menjadi 8 jenis. Pertama adalah terkait dengan *efficiency* atau efisiensinya, dimana memiliki arti yakni tentang kemampuan dari *software* tersebut dalam menggunakan sumber daya sistem dengan cara yang efisien dan efektif ketika digunakan.

Selain itu, juga bisa dilihat dari segi *functionality* atau fungsinya sendiri yang mengacu pada tingkat kinerja dari *software* pada tujuan yang telah ditetapkan. Untuk karakteristik *software* terbaik berdasarkan fungsinya yang ketiga adalah *reliability*. Dimana kemampuan dari *software* untuk menyediakan fungsi seperti yang diinginkan ataupun di dalam kondisi tertentu. Selain itu ada pula *usability*, dimana berhubungan dengan kemampuan dari *software* untuk sejauh mana dapat dipakai dengan mudah dan baik.

Ada pula *integrity*, yang berarti berhubungan dengan kemampuan dari *software* untuk mengakses data yang mencurigakan atau tidak resmi. Selain itu ada pula karakteristik yakni *maintainability*, dimana perangkat lunak sebaiknya memiliki kemudahan untuk melakukan modifikasi. Seperti untuk memperbaiki kesalahan, memperluas fungsi dari software itu sendiri dan juga untuk memperbaiki kinerja.

*Robustness* yakni berfungsi jika perangkat mempunyai data yang tidak *valid* maka dapat dilihat sejauh mana *software* tersebut dapat bekerja. Sedangkan yang terakhir adalah *portability*, dimana memiliki arti bahwa mengacu pada kemudahan dari *software developer* atau pengembangan dari perangkat lunak itu sendiri. Dapat diartikan yakni kemampuan dari sebuah *software* untuk bisa berfungsi dengan baik tanpa adanya perubahan yang terjadi di dalamnya, baik dalam bentuk hardware maupun dalam bentuk *software*-nya.

## 3. Karakteristik Berdasarkan Sifatnya

Selain itu, karakteristik *software* terbaik juga dapat dibedakan berdasarkan sifatnya. Pertama yakni *software engineering* yang merupakan langkah yang dapat

digunakan untuk membuat *software*. Disini nantinya, akan bisa menghasilkan suatu produksi yang terkesan cukup unik dikarenakan tidak mempunyai seri produksi.

Selain itu, pada *software* umumnya dapat diperbaiki sesuai dengan kebutuhan si penggunanya, sedangkan untuk barang dari pabrik memiliki sifat yang permanen ketika diproduksi. Selain itu, karakteristik lainnya adalah memiliki sifat *invisible* atau tidak terlihat. Tak hanya itu saja, *software* pun juga mempunyai sifat fleksibel dimana dapat dikembangkan ataupun dimodifikasi jika masih ditemukan kekurangan. Sedangkan yang terakhir adalah pada umumnya, *software* mampu dihubungkan dengan sistem komputer anda.

#### 4. Karakteristik Berdasarkan Buku Dari Ian Sommerville

Karakteristik mengenai *software* terbaik yang diambil berdasarkan buku karangan Ian Sommerville yang berjudul *software engineering* adalah terdapat empat poin yang bisa diambil. Pertama adalah *usability*, dimana penggunaan dari perangkat lunak ini haruslah sesuai dengan yang telah direncanakan. Selain itu, ada pula karakteristik lain yang disebutkan oleh Ian Sommerville yang disebutkan di dalam bukunya, yakni efisiensi.

Efisiensi sendiri mempunyai arti yakni haruslah mempunyai sifat yang efisien dalam hal penyimpanan. Untuk yang ketiga yakni *dependability*, dimana perangkat lunak atau *software* haruslah memiliki sifat yang dapat dipercaya. Sedangkan yang terakhir, adalah *maintanability* yang berarti dapat dirawat. Dimana *software* tersebut haruslah mampu untuk memenuhi perubahan dari kebutuhan penggunanya tersebut. Sehingga dapat menunjang hal yang dibutuhkan oleh pengguna.

Dengan mengetahui karakteristik dari perangkat lunak tersebut, maka anda pun juga mampu untuk lebih memahami bagaimana sifat dan kinerja dari perangkat lunak itu sendiri. *Software* merupakan salah satu komponen yang penting dan harus anda perhatikan ketika anda akan membeli laptop. Untuk itu, ketika anda akan membeli laptop atau komputer pastikan anda mengetahui terlebih

dahulu apa saja karakteristik dan pahami betul tentang *software* yang nantinya akan anda beli [4].

### 2.3 *Internet of Things*

*Internet of Things (IoT)* adalah sebuah istilah yang muncul dengan pengertian sebuah akses perangkat elektronik melalui media internet. Akses perangkat tersebut terjadi akibat hubungan manusia dengan perangkat atau perangkat dengan perangkat dengan memanfaatkan jaringan internet. Akses. perangkat tersebut terjadi karena keinginan untuk berbagi data, berbagi akses, dan juga mempertimbangkan keamanan dalam aksesnya.

*Internet of Things (IoT)* dimanfaatkan sebagai media pengembangan kecerdasan akses perangkat di dunia industri, di rumah tangga, dan beberapa sektor yang sangat luas dan beragam (contoh : sektor keamanan, dan sektor transportasi). *Internet of Things (IoT)* dapat dikembangkan dengan media perangkat elektronika yang umum seperti Arduino untuk keperluan yang spesifik (khusus). *IoT* juga dapat dikembangkan aplikasi terpadu dengan sistem operasi *Android* [5].

Selain itu juga *Internet of Things (IoT)* merupakan segala aktifitas yang pelakunya saling berinteraksi dan dilakukan dengan memanfaatkan *internet*. Dalam penggunaannya *Internet of Thing* banyak ditemui dalam berbagai aktifitas, contohnya : banyaknya *transportasi online*, *e-commerce*, pemesanan tiket secara *online*, *live streaming*, *e-learning* dan lain-lain bahkan sampai alat-alat untuk membantu dibidang tertentu seperti *remote temperature sensor*, *GPS tracking*, dan sebagainya yang menggunakan *internet* atau jaringan sebagai media untuk melakukannya. Dengan banyaknya manfaat dari *Internet of Things* maka membuat segala sesuatunya lebih mudah, dalam bidang pendidikan *IoT* sangat diperlukan untuk melakukan segala aktifitas dengan menggunakan sistem dan tertata serta sistem pengarsipan yang tepat.



**Gambar 2.1 Ilustrasi dari Penggunaan IoT**

#### **2.4 Arduino Uno**

Arduino uno merupakan papan mikrokontroler yang didalamnya tertanam *microcontroller* dengan merk ATmega yang dibuat oleh perusahaan Atmel Corporation. Berbagai papan Arduino menggunakan tipe ATmega yang berbeda-beda tergantung dari spesifikasinya. Arduino uno sendiri merupakan kesatuan perangkat yang terdiri dari berbagai komponen elektronika dimana penggunaan alat sudah dikemas dalam kesatuan perangkat yang dibuat oleh pemroduksi untuk di perdagangkan.

Arduino dikatakan *open source* karena sebuah platform dari *physical computing*. Platform di sini adalah sebuah alat kombinasi dari *hardware*, bahasa pemrograman dan IDE (*Integrated Development Environment*) yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-upload ke dalam *memory microcontroller*.

Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) untuk bisa disambungkan dengan Arduino. *software* dan *hardware* yang sifatnya interaktif, yaitu dapat menerima rangsangan dari lingkungan dan merespon balik. Konsep untuk memahami hubungan yang manusiawi antara lingkungan yang sifat alaminya adalah analog dengan dunia *digital*, disebut dengan *physical*

*computing*. Dengan arduino uno dapat dibuat sebuah sistem atau perangkat fisik menggunakan Pada prakteknya konsep ini diaplikasikan dalam desain alat atau proyek-proyek yang menggunakan sensor dan *microcontroller* untuk menerjemahkan input analog ke dalam sistem *software* untuk mengontrol gerakan alat-alat elektro-mekanik [6].



**Gambar 2.2 Arduino Uno**

Untuk informasi lebih detail dapat melihat table spesifikasi arduino uno berikut ini.

**Tabel 2.1 Spesifikasi Arduino Uno**

| Spesifikasi                    | Detail  |
|--------------------------------|---|
| Mikrokontroler                 | Atmega328   |
| Operasi tegangan               | 5 V   |
| Tegangan Input yang Disarankan | 7-12V   |
| Jumlah Pin I/O Digital         | 14 pin digital (6 diantaranya menyediakan keluaran PWM) |
| Jumlah Pin Input Analog        | 6 Pin   |

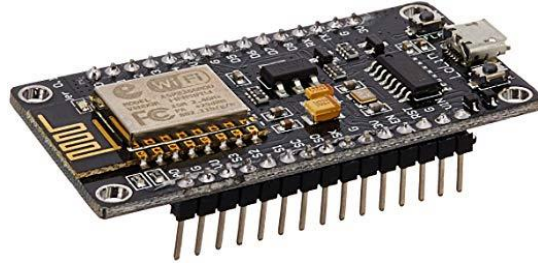
|                        |                  |
|------------------------|------------------|
| Arus DC tiap pin I/O   | 40mA             |
| Arus DC untuk Pin 3,3V | 50mA             |
| Memori <i>flash</i>    | 32 KB            |
| SRAM                   | 2 KB (Atmega328) |
| EEPROM                 | 1 KB (Atmega328) |
| <i>Clock speed</i>     | 16 MHz           |

## 2.5 Modul ESP 8266

Modul ESP8266 merupakan modul *low cost* wifi yang didukung penuh untuk penggunaan TCP/IP ataupun UDP. Esp8266 dikembangkan oleh pengembang asal Tiongkok yaitu “Espressif”. Produk ESP866 memiliki banyak varian. Pada penelitian ini digunakan ESP8266 seri ESP-01. Modul wifi ini bersifat SoC (*System on Chip*), sehingga bisa melakukan *programming* langsung ke ESP8266 tanpa memerlukan mikrokontroler tambahan. Modul Esp8266 juga menyediakan kemampuan tak tertandingi untuk menanamkan kemampuan wifi dalam sistem yang lain, atau berfungsi sebagai aplikasi *standalone* dengan biaya yang rendah dan kebutuhan ruang yang minimal [7].

NodeMCU adalah sebuah *board* elektronik yang berbasis chip ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet (WiFi). Terdapat beberapa pin I/O sehingga dapat dikembangkan menjadi sebuah aplikasi *monitoring* maupun *controlling* pada proyek IOT. NodeMCU ESP8266 dapat diprogram dengan *compiler*-nya Arduino, menggunakan Arduino IDE. Bentuk fisik dari NodeMCU ESP 8266, terdapat *port* USB (mini USB) sehingga akan memudahkan dalam pemrogramannya. NodeMCU ESP8266 merupakan modul turunan pengembangan dari modul platform IoT (*Internet of Things*) keluarga ESP8266 tipe ESP-12. Secara fungsi modul ini hampir menyerupai

dengan platform modul arduino, tetapi yang membedakan yaitu dikhususkan untuk “*Connected to Internet*“ [8].



**Gambar 2.3 ESP-8266**

Untuk informasi lebih detail dapat melihat tabel spesifikasi ESP-8266 berikut ini.

**Tabel 2.2 Spesifikasi ESP-8266**

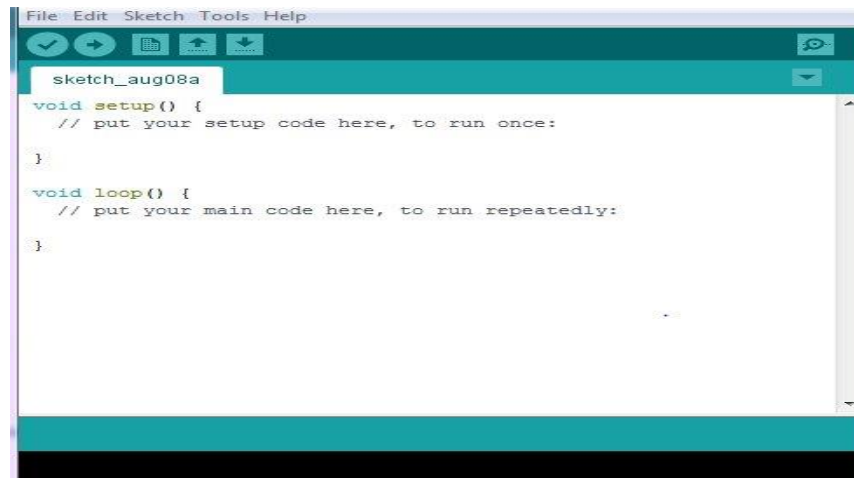
| <b>Spesifikasi</b>  | <b>Detail</b>     |
|---------------------|-------------------|
| Mikrokontroler      | ESP8266-12E       |
| Tegangan Input      | 3.3 ~ 5V          |
| GPIO                | 13 Pin            |
| Kanal PWM           | 10 Kanal          |
| 10 bit ADC Pin      | 1 Pin             |
| <i>Flash Memory</i> | 4 MB              |
| <i>Clock Speed</i>  | 40/26/24 MHz      |
| WiFi                | IEEE 802.11 b/g/n |
| Frekuensi           | 2.4 GHz – 2.5 Ghz |
| USB Port            | Micro USB         |
| USB Chip            | CH340G            |



## 2.6 Arduino IDE

Perangkat lunak Arduino IDE (*Integreted Devolepment Environment*) adalah perangkat yang membuat pengguna menjadi lebih mudah untuk merancang sistem menggunakan mikrokontroler mulai dari menuliskan *source* program, kompail program, unggah, hasil kompilasi, dan uji coba. *Arduino Development Environment* juga digunakan untuk meng-*upload* program yang sudah di *compile* ke memori program *arduino board*. *Arduino Development Environment* juga digunakan untuk meng-*upload* program, tujuan dari diciptakannya *arduino* adalah untuk mempermudah pengguna dalam membuat sistem elektronika dan mempermudah dalam melakukan pembelajaran bagi siswa yang ingin mempelajari sistem mikrokontroler. *Arduino* memiliki keunggulan tersendiri dibandingkan minimum sistem mikrokontroler lainnya karena bersifat *opensource*, dan memiliki bahasa pemrograman yang hampir sama dengan bahasa C. Selain itu *arduino* dilengkapi dengan *bootloader* yang mudah digunakan tanpa adanya penambahan komponen elektronika lagi.

*Arduino IDE* merupakan *software* yang menyerupai bahasa C dan ditulis dengan menggunakan *Java*. *Arduino IDE* terdiri dari editor program, *window* yang memungkinkan pengguna membuat dan mengedit program dalam bahasa *Processing*. *Compiler* pada *arduino* adalah sebuah modul yang mengubah kode program bahasa *Processing* menjadi kode biner. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan *Arduino*. Sebuah kode program *Arduino* umumnya disebut dengan istilah *sketch* atau dengan tipe *fine ino*. Kata *sketch* digunakan secara bergantian dengan kode program dimana keduanya memiliki arti yang sama [9].



**Gambar 2.4 Tampilan Arduino IDE**

## 2.7 Struktur Pemrograman Arduino IDE

Berikut beberapa hal yang diperlukan dalam pemrograman arduino IDE, diantaranya adalah:

### 1. Struktur

Struktur bahasa pemrograman pada arduino, terdiri dari dua bagian yaitu :

```
Void setup ()
```

```
{
```

```
//statement
```

```
}
```

```
Void loop ()
```

```
{
```

```
//statement
```

```
}
```

Void *setup* () berfungsi untuk memanggil satu kali ketika program dijalankan. Fungsi ini digunakan untuk menginisialisasi variabel, pin *mode*, memulai menggunakan *library* baru dan lainnya. Fungsi pengaturan ini dijalankan hanya sekali, setelah *power up* atau *reset* Arduino.

Sedangkan void *loop*() berfungsi untuk mengeksekusi perintah yang akan dijalankan berulang-ulang selama arduino dinyalakan. Pada fungsi *loop*() yang berturut-turut memungkinkan program untuk mengubah dan menanggapi sebuah kondisi dengan terus menerus melakukan pengulangan kondisi yang kita inginkan. Penggunaanya akan secara aktif mengontrol board Arduino [10].

## 2. *Syntax*

*Syntax* merupakan bahasa C yang dibutuhkan untuk format penulisan:

- a. // (komentar satu baris) digunakan untuk memberi catatan dari kode kode pemrograman yng telah dituliskan. Dengan menuliskan // maka apapun yang ditulis di belakangnya akan diabaikan atau tidak akan dibaca oleh program.
- b. /\*.....\*/ (komentar banyak baris) berfungsi untuk memberi catatan beberapa garis sebagai komentar
- c. {.....} atau kurawal berfungsi untuk mendefinisikan blok diagram saat mulai dan berakhir, digunakan juga pada fungsi pengulangan.
- d. ; atau titik koma berfungsi untuk mengakhiri setiap baris kode program yang ditulis.

## 3. Variabel

Variabel adalah nama yang dibuat dan disimpan dalam mikrokontroler. Variabel memiliki nilai yang berubah-ubah sewaktu-waktu saat program dijalankan sehingga perlu ditentukan jenis tipe datanya. Deklarasi variabel dapat dilakukan dengan memberi nilai awal ataupun dengan tidak memberi nilai awal. Dalam pemrograman dikenal dengan 2 maca variabel, diantaranya yaitu:

- a. Variabel global, berfungsi untuk mendeklarasikan diluar fungsi, dan berlaku secara umum dan dapat diakses dimana saja.
- b. Variabel lokal, berfungsi untuk mendeklarasikan didalam fungsi dan hanya bisa diakses oleh pernyataan yang ada di dalam fungsi.

#### 4. Tipe data

Tipe data yang digunakan dalam program ada bermacam-macam, diantaranya sebagai berikut:

- a. *Int* (integer), berfungsi untuk menyimpan angka 2 byte atau 16 bit. Tidak memiliki angka desimal, dan dapat menyimpan nilai dari -23.767 hingga 32767.
- b. *Boolean*, berfungsi untuk menyimpan nilai benar atau salah menggunakan 1 bit dari RAM.
- c. *Long*, berfungsi untuk menyimpan angka 4 byte dari RAM atau jika data integer tidak mencukupi yang mempunyai rentang -2.147.482.648 hingga 2.147.483.648.
- d. *Float*, berfungsi untuk menyimpan angka desimal 4 byte (32 bit) dari RAM yang mempunyai rentang -3.4028235E+38 sampai 3,40282335+38.
- e. *Double*, adalah tipe data sama seperti '*float*', namun double memiliki ukuran penyimpanan yang lebih besar dibandingkan '*float*'.
- f. *Char* (*character*), menyimpan 1 karakter menggunakan kode ASCII (misalnya 'A'= 65). Hanya memakai 1 byte (8 bit) dari RAM.
- e. *Array* adalah kumpulan nilai yang dapat di akses dengan *index* number, nilai yang terdapat dalam *array* dapat di panggil dengan cara menuliskan nama *array* dan *index* number. *Array* dengan *index* 0 merupakan nilai pertama dari *array*. *Array* perlu di deklarasikan dan kalau perlu di beri nilai sebelum di gunakan.

## 5. Operator Matematika

Operator matematika yang digunakan untuk memanipulasi angka (seperti matematika sederhana). Berikut ini merupakan contohnya.

- a. Sama dengan (=), berfungsi untuk membuat sesuatu menjadi sama dengan nilai yang lain.
- b. Persen (%), berfungsi menghasilkan sisa dari hasil pembagian suatu angka.
- c. Tambah (+), sebagai operasi penambahan.
- d. Kurang (-), sebagai operasi pengurangan.
- e. Asteris (\*), sebagai operasi perkalian.
- f. Garis miring (/), sebagai operasi pembagian

## 6. Operasi Perbandingan

Adapun contoh operasi perbandingan adalah sebagai berikut:

- a. == (sama dengan)
- b. != (tidak sama dengan)
- c. < (lebih kecil dari)
- d. > (lebih besar dari)
- e. ≤ (lebih kecil dari sama dengan)
- f. ≥ (lebih besar dari sama dengan)
- g. ! (boolean not)
- h. && (boolean and)
- i. || (boolean or)

## 7. Struktur pengatur

Adapun contoh operator perbandingan adalah sebagai berikut:

- a. *If.....else* dengan format seperti dibawah :  
*If* (kondisi) { ..... }  
*Else if* (kondisi) { ..... }

*Else {.....}*

Dengan struktur diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisi *TRUE*, dan jika tidak (*FLASE*) maka akan diperiksa apakah kondisi pada *else if* dan jika kondisinya *FALSE* maka kode pada *else* yang akan dijalankan. Program tersebut dapat digunakan untuk menentukan suatu kondisi, dan saat kondisi telah terpenuhi maka akan dilaksanakan sesuai dengan perintah yang telah ditentukan. Begitu juga saat kondisinya tidak terpenuhi.

- b. *Switch* Pernyataan *switch* yaitu sebuah variabel yang beurutan diuji oleh beberapa konstanta bilangan bulat atau karakter sintaks perintah *switch*
- c. *Looping* Yaitu pengulangan satu atau bebrapa perintah hingga mencapai keadaan tertentu. Berikut beberapa perintah *looping*, diantaranya:
  - 1) *For*..... digunakan bila anda ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan ke atas dengan *i++* atau ke bawah dengan *i--*.
  - 2) *While*.....
  - 3) *Do*.....*while*.....

## 8. Kode digital

Kode digital berfungsi untuk mengatur pin-pin digital pada arduino

- a. *PinMode* (*Pin,mode*) Kode ini berfungsi untuk mengatur mode pin. Pin disini merupakan nomor pin yang akan digunakan pada board arduino uno, yang terdapat pada pin digital 0 hingga 13, dan mode sendiri dapat berupa input ataupun output.
- b. *DigitalWrite* (*pin,value*) Kode ini berfungsi untuk pin input yang membaca nilai sensor yang ada pada pin. Nilai sebatas 1 atau 0, benar atau salah.
- c. *DigitalRead* (*pin*) Kode ini digunakan sebagai pin input, dapat menggunakan kode ini untuk mendapatkan nilai *HIGH* (+5V) atau *LOW* (*ground*)

## 9. Kode analog

Digunakan saat menggunakan pin analog pada arduino. Pin analog dimulai dari A0 hingga A5. Dan hanya dapat digunakan sebagai input. Dalam penulisan program tidak perlu menuliskan *pinMode* pada void *setup*.

- a. *analogRead* (pin) Digunakan saat pin analog ditetapkan sebagai *input*, dapat membaca keluaran voltasenya. Keluaran berupa angka 0 untuk 0V dan 1024 untuk 5V
- b. *analogWrite* (pin). Digunakan untuk beberapa pin arduino yang mendukung PWM yaitu pada pin 3, 5, 6, 9, 10, 11. Dapat merubah pin *on* atau *off* dengan cepateshingga dapat berfungsi layaknya keluaran analog. Nilai pada format kode tersebut adalah angka antara 0 (0% duty cycle ~ 0 V) dan 255 (100% duty cycle ~ 5 V).

## 10. Logic operator

Operator logical digunakan untuk membandingkan 2 ekspresi dan mengembalikan nilai balik benar atau salah tergantung dari operator yang digunakan. Terdapat 3 *operator logical AND, OR, dan NOT*, yang biasanya digunakan pada *if statement*.

Contoh penggunaan:

### a. Logical AND

If (  $x > 0 \ \&\& \ x < 5$  ) // bernilai benar apabila kedua operator pembandingan terpenuhi

### b. Logical OR

If (  $x > 0 \ || \ y > 0$  ) // bernilai benar apabila salah satu dari operator pembandingan terpenuhi

c. *Logical NOT*

If ( !x > 0 ) // bernilai benar apabila ekspresi operator salah

## 11. Konstanta

Arduino mempunyai beberapa variable yang sudah di kenal yang kita sebut konstanta. Ini membuat memprogram lebih mudah untuk di baca. Konstanta di klasifikasi berdasarkan *group*.

a. *True/False*

Merupakan konstanta Boolean yang mendefinisikan *logic level*. *False* mendefinisikan 0 dan *True* mendefinisikan 1.

```
If( b == TRUE );
{
  //doSomething
}
```

b. *high/low*

Konstanta ini mendefinisikan aktifitas pin *HIGH* atau *LOW* dan di gunakan ketika membaca dan menulis ke digital pin. *HIGH* di definisikan sebagai 1 sedangkan *LOW* sebagai 0.

```
digitalWrite( 13, HIGH );
```

c. *input/output*

Konstanta ini digunakan dengan fungsi *pinMode()* untuk mendefinisikan mode pin digital, sebagai input atau output.

```
pinMode( 13, OUTPUT );
```

## 2.8 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android



menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android merupakan generasi baru platform mobile, platform yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan dibawah GNU, *General Public* Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah “*copyleft*” lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah *terms*.

Android didistribusikan di bawah Lisensi *ApacheSoftware* (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen *handset* khususnya) dapat memilih untuk meningkatkan platform tanpa harus memberikan perbaikan mereka ke masyarakat *open source*. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan [11].



**Gambar 2.5 Logo Android**

## **2.9 Blynk**

Aplikasi blynk berfungsi sebagai aplikasi pengontrol perangkat IoT, aplikasi ini bisa didownload secara gratis di *apps store* untuk pengguna *iphone* dan *play store* untuk pengguna android. Dalam aplikasi ini sudah disediakan *energy default* sebesar 2000 energy. setiap *widget box* mempunyai nilai *energy* yang berbeda-beda, jika *energy* yang dimiliki tidak mencukupi untuk membuat beberapa *widget box* maka diharuskan membeli *energy* tambahan .



**Gambar 2.6 Logo Blynk**

*a. Blynk Apps*

*Blynk Apps* memungkinkan untuk membuat *project interface* dengan berbagai macam komponen input output yang mendukung untuk pengiriman maupun penerimaan data serta merepresentasikan data sesuai dengan komponen yang dipilih. Representasi data dapat berbentuk visual angka maupun grafik. Terdapat 4 jenis kategori komponen yang terdapat pada Aplikasi Blynk

- 1) *Controller* digunakan untuk mengirimkan data atau perintah ke *Hardware*
- 2) *Display* digunakan untuk menampilkan data yang berasal dari *hardware* ke *smartphone*
- 3) *Notification* digunakan untuk mengirim pesan dan notifikasi.
- 4) *Interface* Pengaturan tampilan pada aplikasi Blynk dapat berupa menu ataupun *tab*.
- 5) *Others* beberapa komponen yang tidak masuk dalam 3 kategori sebelumnya diantaranya *Bridge*, *RTC*, *Bluetooth*

*b. Blynk Server*

*Blynk server* merupakan fasilitas *Backend Service* berbasis *cloud* yang bertanggung jawab untuk mengatur komunikasi antara aplikasi smart phone dengan lingkungan *hardware*. Kemampuannya untuk menangani puluhan *hardware* pada saat yang bersamaan semakin memudahkan bagi para pengembang sistem

IoT. Blynk *server* juga tersedia dalam bentuk *local server* apabila digunakan pada lingkungan tanpa internet.

b. *Blynk Library*

*Blynk Library* dapat digunakan untuk membantu pengembangan *code*. *Blynk library* tersedia pada banyak platform perangkat keras sehingga semakin memudahkan para pengembang IoT dengan fleksibilitas *hardware* yang didukung oleh lingkungan Blynk. [12].