

## **BAB II LANDASAN TEORI**

Dasar dari sebuah penelitian adalah landasan teori. Selain untuk mendukung pelaksanaan pada penelitian, landasan teori akan membuktikan bahwa penelitian tersebut berkualitas, dan memiliki pengetahuan yang cukup untuk melanjutkan penelitian sebelumnya, memperbaiki teori-teori pada penelitian sebelumnya, atau bahkan mematahkan teori sebelumnya.

### **2.1 Rancang Bangun**

Rancang Bangun adalah proses mengatur, melakukan atau merencanakan segala sesuatu sebelum bertindak untuk memecahkan permasalahan yang ada (Kamus Bahasa Indonesia, 2005).

### **2.2 Sistem**

Menurut (Jogiyanto, 2005) Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

### **2.3 Keamanan**

Kata "keamanan" bahasa Inggris Security, berasal dari kata Latin "se-curus". "Se" berarti "tanpa" dan "curus" berarti "kegelisahan." (Tanpa

kegelisahan, mengandung makna "keamanan"), berarti pembebasan dari kegelisahan, atau situasi damai tanpa risiko atau ancaman. Arti "keamanan" memiliki berbagai makna, termasuk "untuk merasa aman," dan "dilindungi", digunakan untuk menggambarkan situasi tanpa resiko atau *worries*.

## 2.4 Sepeda Motor

Sepeda Motor adalah kendaraan beroda dua yang digerakkan oleh sebuah mesin. Letak kedua roda sebaris lurus dan pada kecepatan tinggi sepeda motor tetap stabil disebabkan oleh gaya *giroskopik*. Sedangkan pada kecepatan rendah, kestabilan atau keseimbangan sepeda motor bergantung kepada pengaturan setang oleh pengendara. (Wikipedia, 2016).

## 2.5 SMS (*Short Message Service*)

SMS adalah sebuah layanan yang dilaksanakan dengan sebuah telepon genggam untuk mengirim atau menerima pesan-pesan pendek. Pada mulanya SMS dirancang sebagai bagian daripada GSM, tetapi sekarang sudah didapatkan pada jaringan bergerak lainnya termasuk jaringan UMTS. Sebuah pesan SMS maksimal terdiri dari 140 bytes, dengan kata lain sebuah pesan bisa memuat 140 karakter 8-bit, 160 karakter 7-bit atau 70 karakter 16-bit untuk bahasa Jepang, bahasa Mandarin dan bahasa Korea yang memakai *Hanzi* bytes ini ada data-data lain yang termasuk. Adapula beberapa metode untuk mengirim pesan yang lebih dari 140 bytes, tetapi seorang pengguna harus membayar lebih dari sekali. SMS bisa pula untuk mengirim gambar, suara dan film. SMS bentuk ini disebut MMS. Pesan-

pesan SMS dikirim dari sebuah telepon genggam ke pusat pesan sini pesan disimpan dan mencoba mengirimnya selama beberapa kali. Setelah sebuah waktu yang telah ditentukan, biasanya 1 hari atau 2 hari, lalu pesan dihapus. Seorang pengguna bisa mendapatkan konfirmasi dari pusat pesan ini.

### 2.5.1 Module SIM900

SIM900 adalah modul SIM yang digunakan pada penelitian ini. Modul SIM900 GSM/GPRS adalah bagian yang berfungsi untuk berkomunikasi antara pemantau utama dengan *Handphone*.



Gambar 2.1 Modul SIM900

## 2.6 Mikrokontroler

Menurut (Abdul Kadir, 2015) Mikrokontroler atau kadang dinamakan pengontrol tertanam (*embedded controller*) adalah suatu sistem yang mengandung masukan/keluaran, memori dan prosesor, yang digunakan pada produk seperti mesin cuci, pemutar video, dan telepon. Pada prinsipnya, mikrokontroler adalah sebuah komputer berukuran kecil yang dapat

digunakan untuk mengambil keputusan, melakukan hal-hal yang bersifat berulang, dan dapat berinteraksi dengan peranti-peranti eksternal, seperti sensor ultrasonic untuk mengukur jarak terhadap suatu objek, penerima GPS untuk memperoleh data posisi kebumian dari satelit, dan motor untuk mengontrol gerakan pada robot. Sebagai komputer yang berukuran kecil, mikrokontroler cocok diaplikasikan pada benda-benda yang berukuran kecil, misalnya sebagai pengendali pada *QuadCopter* ataupun robot.

### **2.6.1 Arduino**

Menurut (Abdul Kadir, 2012), Arduino adalah jenis papan (*board*) yang berisi mikrokontroler. Dengan perkataan lain, Arduino dapat disebut sebagai sebuah papan mikrokontroler.

### **2.6.2 Arduino Uno**

Arduino Uno adalah salah satu produk berlabel Arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega328 (sebuah keeping yang secara fungsional bertindak seperti sebuah komputer). Peranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks. Pengendalian LED hingga pengontrolan robot dapat diimplementasikan dengan menggunakan papan yang berukuran relatif kecil. Bahkan, dengan penambahan komponen tertentu, peranti ini biasa dipakai untuk pemantauan jarak jauh melalui internet, misalnya pengendalian alat sistem keamanan sepeda motor.

Arduino Uno mengandung mikroprosesor (berupa Atmel AVR) dan dilengkapi dengan *oscillator* 16MHz (yang memungkinkan operasi berbasis waktu dilaksanakan dengan tepat), dan regulator (pembangkit tegangan) 5 volt. Sejumlah pin tersedia di papan. Pin 0 hingga 13 digunakan untuk isyarat digital, yang hanya bernilai 0 atau 1. Pin A0-A5 digunakan untuk isyarat analog. Arduino Uno dilengkapi dengan *static random-access memory* (SRAM) berukuran 2KB untuk memegang data, *flash memory* berukuran 32KB, dan *erasable programmable read-only memory* (EEPROM) untuk menyimpan program (Abdul Kadir, 2015).



Gambar 2.2 Arduino Uno

### 2.6.3 Voltage Regulator Step Down

Voltage Regulator Step Down berfungsi untuk memberikan power supply dengan tegangan yang sesuai dengan module-module yang akan digunakan, seperti SIM900, Arduino uno, kebutuhan 5 Volt power supply pada kendaraan dan lain-lain.



Gambar 2.3 Voltage Regulator Step Down

#### 2.6.4 Relay

Relay adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen Elektromekanikal yang terdiri dari dua bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat kontak Saklar/Switch). Relay menggunakan prinsip Elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.



Gambar 2.4 Relay

## 2.7 Teori Pengembangan Sistem

Teori pengembangan sistem (*system development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan yang lama secara keseluruhan atau memperbaiki sistem yang telah ada (Jogiyanto, 2005). Adapun beberapa hal yang menyebabkan perlunya pengembangan sistem adalah adanya permasalahan-permasalahan yang timbul pada sistem yang lama, masalah yang timbul dapat berupa ketidakberesan sistem dan pertumbuhan organisasi, selain itu penyebab perlunya pengembangan sistem adalah untuk meraih kesempatan-kesempatan serta adanya intruksi-intruksi.

Beberapa prinsip dalam proses pengembangan sistem (Jogiyanto, 2005) adalah sebagai berikut:

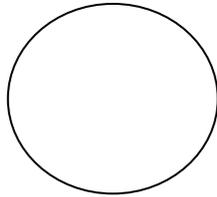
- a. Sistem yang dikembangkan adalah untuk manajemen.
- b. Sistem yang dikembangkan adalah investasi modal yang besar.
- c. Sistem yang dikembangkan memerlukan orang terdidik.
- d. Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengembangan sistem.
- e. Proses pengembangan sistem tidak harus urut.
- f. Jangna takut membatalkan proyek.
- g. Dokumentasi harus ada untuk pemedoman dalam pengembangan sistem.

### 2.7.1 Diagram Konteks (*context diagram*)

Diagram konteks (*context diagram*) adalah suatu alat yang digunakan untuk menggambarkan alir data dan interaksi sistem secara

umum. Ada tiga simbol yang digunakan pada konteks diagram untuk menggambarkan alir data, yaitu:

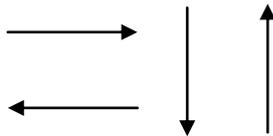
1. Proses (*process*), simbol yang digunakan:



2. Kesatuan Luar (*external entity*), simbol yang digunakan:



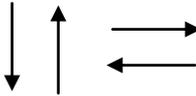
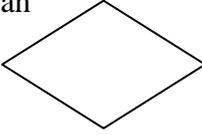
3. Arus Data (*data flow*), simbol yang digunakan:



### 2.7.2 Bagan Alir Program (program flowchart)

Bagan alir program (*program flowchart*) adalah saat yang penting bagi pengguna atau *user* untuk dapat memahami logika program secara terperinci. Bagan alir ini dibuat dari derifikasi bagan alir sistem. Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program (Jogiyanto, 2005). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah didalam program komputer secara logika. Untuk simbol-simbol program *flowchart* hampir sama dengan bagan alir dokumen, hanya terdapat perbedaan pada simbol proses, penyimpanan data *input/output* data. Adapun simbol-simbol program-program *flowchart* sebagai berikut.

Tabel 2.1 Simbol bagan alir program

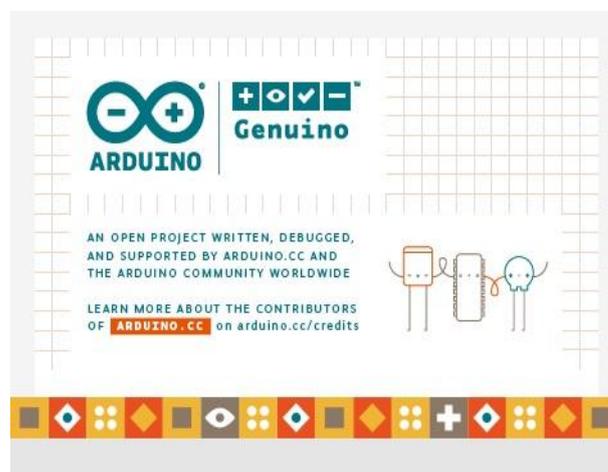
Simbol	Keterangan
<p data-bbox="389 387 560 421"><i>Input/Output</i></p> 	<p data-bbox="807 387 1299 456">Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i></p>
<p data-bbox="389 573 475 607">Proses</p> 	<p data-bbox="807 573 1326 678">Simbol proses (<i>processing symbol</i>) atau simbol pengolah yang digunakan suatu poses</p>
<p data-bbox="389 759 523 792">Garis Alir</p> 	<p data-bbox="807 759 1315 864">Simbol garis alir (<i>flow lines symbol</i>), digunakan untuk menunjukkan arus dari proses</p>
<p data-bbox="389 945 555 978">Penghubung</p> 	<p data-bbox="807 945 1337 1120">Simbol penghubung (<i>connector symbol</i>), digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang sama / dihalaman yang lain.</p>
<p data-bbox="389 1131 531 1164">Keputusan</p> 	<p data-bbox="807 1131 1278 1236">Simbol keputusan (<i>decison symbol</i>), digunakan untuk suatu penyelesaian kondisi didalam program.</p>
<p data-bbox="389 1317 616 1350">Proses terdefinisi</p> 	<p data-bbox="807 1317 1302 1422">Simbol proses terdefinisi digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukan ketempat lain.</p>
<p data-bbox="389 1503 512 1536">Terminal</p> 	<p data-bbox="807 1503 1299 1572">Simbol digunakan untuk menunjukkan awal dan akhir dar suatu program</p>

## 2.8 Perangkat Lunak Pendukung

Perangkat lunak pendukung yaitu *software* yang akan digunakan untuk menganalisa dan merancang sistem yang diusulkan, diantaranya seperti berikut:

### 2.8.1 Teori Dasar Pemrograman Arduino IDE

Arduino IDE adalah *software* yang ditujukan sebagai perangkat pengembangan *sketch* yang digunakan sebagai program di papan Arduino. IDE (*Integrated Development Environment*) berarti bentuk alat pengembangan program yang terintegrasi sehingga berbagai keperluan disediakan dan dinyatakan dalam bentuk antarmuka berbasis menu. Dengan menggunakan Arduino IDE, kita bisa menulis *sketch*, memeriksa ada kesalahan atau tidak di *sketch*, dan kemudian memunggah *sketch* yang sudah terkompilasi ke papan Arduino.

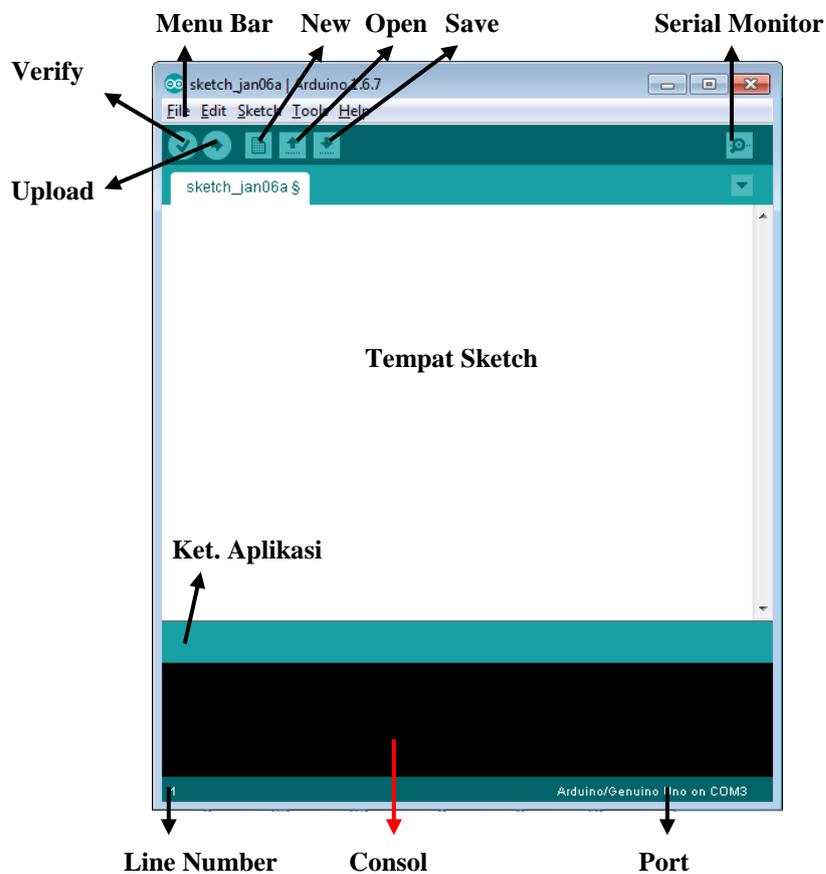


Gambar 2.5 Arduino IDE

## Membuat Program Arduino IDE

Untuk memulai Arduino IDE dilakukan dengan langkah-langkah sebagai berikut:

1. Klik *icon Start* pada sistem operasi Microsoft Windows. Pilih All Programs. Pilih Arduino.
2. Akan tampil Arduino IDE, setelah tampilan Arduino IDE muncul *sketch* baru siap dituliskan.



Gambar 2.6 Tampilan Arduino IDE

## 2.8.2 Bagian-bagian IDE Arduino

1. Menu Bar digunakan untuk memilih tugas-tugas tertentu seperti menyimpan *project*, membuka *project* dan lain-lain
2. *Verify* atau dikenal dengan istilah *Compile*. Sebelum aplikasi di-*upload* ke board Arduino, *sketch* yang dibuat diverifikasi terlebih. Jika ada kesalahan pada *sketch*, nanti akan muncul *error*. Proses *Verify/Compile* mengubah *sketch* ke *binary code* untuk diupload ke mikrokontroler.
3. *Upload* tombol ini berfungsi untuk mengupload *sketch* ke board Arduino. Walaupun kita tidak mengklik tombol *verify*, maka *sketch* akan di-*compile*, kemudian langsung di-*upload* ke board. Berbeda dengan tombol *verify* yang hanya berfungsi untuk memverifikasi *source code* saja.
4. *New Sketch* Membuka window dan membuat *sketch* baru
5. *Open Sketch* Membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file *.ino*
6. *Save Sketch* menyimpan *sketch*, tapi tidak disertai dengan meng-*compile*.
7. Serial Monitor Membuka *interface* untuk komunikasi serial.
8. Keterangan Aplikasi pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal “*Compiling*” dan “*Done Uploading*” ketika kita meng-*compile* dan mengupload *sketch* ke board Arduino

9. Konsol Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang *sketch* akan muncul pada bagian ini. Misal, ketika aplikasi meng-*compile* atau ketika ada kesalahan pada *sketch* yang kita buat, maka informasi *error* dan baris akan diinformasikan di bagian ini.
10. Baris *Sketch* bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada *sketch*.
11. Informasi Port Bagian ini menginformasikan port yang dipakai oleh board Arduino.

---

Arduino menggunakan pemrograman dengan bahasa C.

### **Struktur**

Setiap program Arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada.

Semua bahasa perograman terdiri dari

1. Ekspresi,
2. Statemen,
3. Blok statemen
4. Blok fungsi

Ekspresi adalah kombinasi operand dan operator contoh  $2+3$  ,  $X > Y$   
dst... 2,3,X dan Y adalah operand sedangkan + dan > adalah operator .

Statemen adalah instruksi lengkap dalam bahasa C diakhiri dgn tanda ; (titik koma) contoh :  $A = 2+3;$

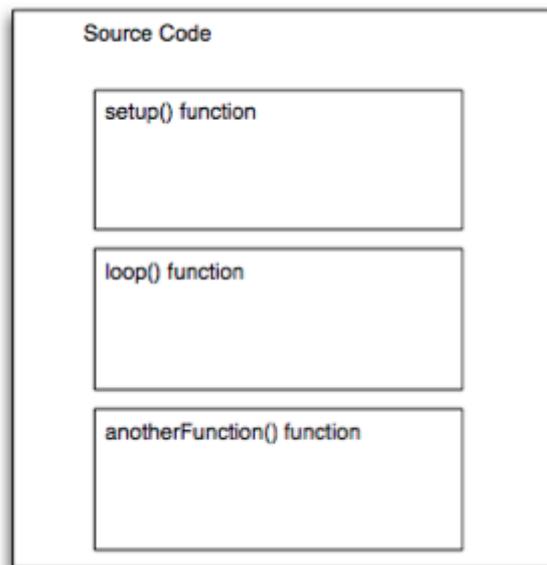
Statemen dan ekspresi C arduino indentik dengan ANSI-C , Struktur Program C Arduino minimal terdiri dari dua fungsi yaitu setup() dan loop().

- **void setup() { }**

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

- **void loop() { }**

Fungsi ini akan dijalankan setelah setup (fungsi void setup) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.



## Syntax

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

- **//(komentar satu baris)**

Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

- **/\* \*/(komentar banyak baris)**

Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

- **{ }(kurung kurawal)**

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

- **;(titik koma)**

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

## Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya.

- **int (integer)**

Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -32,768 dan 32,767.

- **long (long)**

Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -2,147,483,648 dan 2,147,483,647.

- **boolean (boolean)**

Variabel sederhana yang digunakan untuk menyimpan nilai TRUE (benar) atau FALSE (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.

- **float (float)**

Digunakan untuk angka desimal (*floating point*). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38.

- **char (character)**

Menyimpan 1 karakter menggunakan kode ASCII (misalnya 'A' = 65). Hanya memakai 1 byte (8 bit) dari RAM.

## Operator Matematika

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana).

- =

Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya:  $x = 10 * 2$ ,  $x$  sekarang sama dengan 20).

- %

Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya:  $12 \% 10$ , ini akan menghasilkan angka 2).

- +

Penjumlahan

- -

Pengurangan

- \*

Perkalian

- /

Pembagian

## Operator Pembandingan

Digunakan untuk membandingkan nilai logika.

- ==

Sama dengan (misalnya:  $12 == 10$  adalah FALSE (salah) atau  $12 == 12$  adalah TRUE (benar))

- !=

Tidak sama dengan (misalnya:  $12 != 10$  adalah TRUE (benar) atau  $12 != 12$  adalah FALSE (salah))

- <

Lebih kecil dari (misalnya:  $12 < 10$  adalah FALSE (salah) atau  $12 < 12$  adalah FALSE (salah) atau  $12 < 14$  adalah TRUE (benar))

- >

Lebih besar dari (misalnya:  $12 > 10$  adalah TRUE (benar) atau  $12 > 12$  adalah FALSE (salah) atau  $12 > 14$  adalah FALSE (salah))

## Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan (banyak lagi yang lain dan bisa dicari di internet).

1. if..else, dengan format seperti berikut ini:

```
if (kondisi) { }
```

```
else if (kondisi) { }
```

```
else { }
```

Dengan struktur seperti diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisinya TRUE, dan jika tidak (FALSE) maka akan diperiksa apakah kondisi pada else if dan jika kondisinya FALSE maka kode pada else yang akan dijalankan.

2. for, dengan format seperti berikut ini:

```
for (int i = 0; i < #pengulangan; i++) { }
```

Digunakan bila anda ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan i++ atau ke bawah dengan i--.

## **Digital**

1. pinMode(pin, mode)

Digunakan untuk menetapkan mode dari suatu pin, pin adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19).

Mode yang bisa digunakan adalah INPUT atau OUTPUT.

## 2. `digitalWrite(pin, value)`

Ketika sebuah pin ditetapkan sebagai OUTPUT, pin tersebut dapat dijadikan HIGH (ditarik menjadi 5 volts) atau LOW (diturunkan menjadi *ground*).

## 3. `digitalRead(pin)`

Ketika sebuah pin ditetapkan sebagai INPUT maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah HIGH (ditarik menjadi 5 volts) atau LOW (diturunkan menjadi *ground*).

## **Analog**

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam alam analog (menggunakan trik). Berikut ini cara untuk menghadapi hal yang bukan digital.

## 1. `analogWrite(pin, value)`

Beberapa pin pada Arduino mendukung PWM (pulse width modulation) yaitu pin 3, 5, 6, 9, 10, 11. Ini dapat merubah pin hidup (on) atau mati (off) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. Value (nilai) pada format kode tersebut adalah angka antara 0 (0% *duty cycle* ~ 0V) dan 255 (100% *duty cycle* ~ 5V).

## 2. `analogRead(pin)`

Ketika pin analog ditetapkan sebagai INPUT anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volt) dan 1024 (untuk 5 volt).