

MINGGU KE-3 : BAB 4

Dasar Pemrograman Arduino

Kelas Memulai Jadi IoT Engineer Hebat



Isi dan elemen dari dokumen ini memiliki hak kekayaan intelektual yang dilindungi oleh undang-undang

Dilarang menggunakan, merubah, memperbanyak, dan mendistribusikan dokumen ini untuk tujuan komersil

A. Pengertian Arduino IDE



Arduino IDE adalah perangkat lunak *open-source* yang digunakan untuk memprogram papan Arduino. Arduino IDE ini menggunakan bahasa pemrograman yang baku yaitu bahasa C. Bahasa pemrograman ini sangat populer dan banyak digunakan oleh para *programmer* profesional. Untuk beberapa aplikasi pemrograman, biasanya menyebut hasil pengkodean dengan istilah *script*, *listing*, atau *program*. Nah, untuk Arduino IDE sendiri sebutannya bukan itu, melainkan “*Sketch*”, yang apabila diterjemahkan ke bahasa Indonesia berarti “sketsa”.

B. Struktur Dasar Program Arduino

Karakteristik penulisan kode program Arduino sangatlah sederhana. Hanya terdiri atas dua *method* utama, yaitu *void setup()* dan *void loop()*. Perlu anda ketahui bahwa *method void* pada Arduino itu dipakai untuk mendeklarasikan suatu pernyataan tanpa mengembalikan sebuah nilai.

Struktur program Arduino yang benar yaitu sebagai berikut :

```
void setup()  
{  
  pernyataan;  
}  
  
void loop()  
{  
  pernyataan;  
}
```

Keterangan :

Kedua *method* itu memiliki keterkaitan satu sama lain. *Sketch* Arduino akan menjadi *error* jika salah satu *method* tersebut tidak ada di dalamnya.



1. void setup()

Void setup() merupakan bentuk inisialisasi atau pengenalan dalam program Arduino dan hanya dieksekusi sekali sejak program dijalankan.

Umumnya *method* ini digunakan untuk pendefinisian mode pin atau memulai komunikasi serial. Perhatikan contoh *void setup()* berikut ini :

```
void setup()  
{  
  pinMode(13,OUTPUT);  
}
```

Keterangan :

Dalam perintah tersebut terdapat pernyataan (*statement*) yang berisi penginisialisasian pin 13 sebagai pin *output* (keluaran).



2. void loop()

Setelah *void setup()* dijalankan, selanjutnya program akan menjalankan *void loop()*.

Method ini akan dieksekusi terus-menerus secara berurutan hingga program berhenti dijalankan. Perhatikan contoh *void loop()* berikut ini :

```
void loop()  
{  
  digitalWrite(13, HIGH);  
  delay(1000);  
}
```

Keterangan :

Untuk beberapa kasus pemrograman yang kompleks, terkadang *programmer* juga menambahkan *void* baru untuk mempermudah pemrograman. Jadi *void* baru ini nantinya bisa diletakkan setelah atau sebelum *void loop()*.

C. Syntax Arduino

Secara umum, sintaks (*syntax*) dikenal sebagai seperangkat aturan tata bahasa pada kata atau klausa dalam membentuk kalimat. Sedangkan dalam bahasa pemrograman, sintaks diartikan sebagai suatu aturan dalam penulisan kode program agar kode tersebut bisa dimengerti komputer.

1. Titik Koma (*Semicolon*)

Adalah sintaks wajib dalam pemrograman Arduino. Biasanya sintaks ini ditempatkan pada akhir pernyataan. Contohnya seperti ini :

```
int x = 13;
```

2. Kurung Kurawal (*Curly Braces*)

Berfungsi untuk mendefinisikan awal dan akhir dari sebuah blok *method*. Contohnya seperti yang anda lihat di bawah ini :

```
void loop()
{
  isi program
}
```



3. #define

Yaitu penamaan suatu nilai konstan sebelum program dikompilasi. Misalnya saat mendeklarasikan nomor pin yang digunakan. Hal ini seperti yang ditunjukkan pada gambar berikut :

```
#define ledPin 3
```

Keterangan :

- ledPin = variabel yang digunakan.
- 3 = nilai (*value*) atau pin yang sedang digunakan.

4. #include

Untuk memanggil *library* ke dalam sebuah *sketch* agar mempermudah pengguna dalam proses pengkodean. Cara penggunaan sintaks sangatlah mudah. Anda hanya perlu menambahkan nama *library* yang ingin di-*import* ke dalam program. Contohnya seperti gambar di bawah ini :

```
#include <OneWire.h>
```




D. Komentar

1. Sintaks komentar satu baris

Untuk membuat komentar satu baris, anda dapat menggunakan sintaks garis miring dua kali. Contoh penggunaan sintaksnya seperti ini :

```
// isi komentar
```

2. Sintaks komentar lebih dari satu baris

Untuk membuat komentar lebih dari satu baris, anda harus menggunakan kombinasi sintaks (dua garis miring dan dua bintang) dimana komentar yang ingin dimasukkan itu berada tepat di bagian tengahnya. Berikut ini adalah contoh penggunaannya :

```
/* isi komentar */
```

E. Nilai Konstan

1. TRUE / FALSE

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

```
If(c==TRUE);  
{  
  perintah;  
}
```

2. HIGH / LOW

Merupakan konstanta yang digunakan untuk menentukan kondisi pada pin digital.

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



F. Perintah Pin Input/Output

1. pinMode(pin, mode)

Untuk memperjelas apakah suatu pin digunakan sebagai *INPUT* atau *OUTPUT*. Contoh penulisannya seperti berikut ini :

```
pinMode(3, INPUT);
```

2. digitalRead(pin)

Untuk membaca nilai pin digital yang dikehendaki lalu menyimpannya ke dalam suatu variabel. Berikut contoh penulisannya :

```
Baca = digitalRead(8);
```



3. **digitalWrite(pin, value)**

Untuk menuliskan suatu nilai pada pin digital. Lengkapnya, fungsi digitalWrite() pada Arduino dipakai untuk menentukan nilai logika awal yang ada pada suatu pin apakah itu *LOW* atau *HIGH*. Contoh penulisannya terlihat di bawah ini :

```
digitalWrite(9,HIGH);
```

4. **analogRead(pin)**

Untuk membaca data dari pin analog kemudian disimpan ke dalam suatu variabel dimana data hasil pembacaan tersebut berupa nilai integer dengan kisaran 0 sampai 1023. Berikut contoh penulisannya :

```
Baca = analogRead(A0);
```

5. **analogWrite(pin, value)**

Untuk mengirimkan nilai analog dengan metode PWM (*Pulse Width Modulation*) pada pin khusus yang bertanda tilde (~), yaitu melalui pin papan Arduino Uno: 3; 5; 6; 9; 10; dan 11. Berikut contoh penulisannya :

```
analogWrite(5,100);
```

Sekian Materi

Dasar Pemrograman Arduino

Sampai Jumpa di Materi Berikutnya

