Riptek HMTE 2022

**MODUL PROGRAMMING ARDUINO**

**MINGGU 1**

# Tujuan

1. Menilik ulang dasar bahasa pemrograman C.
2. Memahami contoh kode yang diberikan.

# Perlengkapan

1. Komputer.
2. Koneksi internet.

# Rangkuman Teori dan Latarbelakang

## Dengan bahasa apa Arduino diprogram?

Framework original Arduino menggunakan bahasa C dan C++ yang disederhanakan sebagai bahasa pemrogramannya. Bahasa tersebut digunakan karena cepat dan efisien untuk mikrokontroler. C/C++ adalah bahasa pemrograman tingkat tinggi, namun dekat dengan bahasa mesin.

Dengan komunitas yang semakin berkembang, Arduino tidak hanya dapat diprogram dengan C/C++. MicroPython membawa bahasa pemrograman terfavorit ke dunia mikrokontroler, termasuk Arduino. Pengembangan dalam bahasa Java, JavaScript, Assembly juga dapat dilakukan. Namun, bahasa C/C++ tetap menjadi bahasa terbaik untuk memulai belajar.

## *Review* pemrograman Arduino

### Arduino IDE

Untuk memprogram sebuah Arduino, diperlukan IDE (*Integrated Development Environment*). IDE ini berfungsi sebagai *manager library*/*module*, berkas-berkas kode, serta *compiler* bahasa C/C++ sebelum kode diunggah ke Arduino. IDE juga memberikan petunjuk jika ada kesalahan pada kode yang kalian tulis. IDE Arduino dapat diunduh melalui website resmi mereka (<https://www.arduino.cc/en/software>). Tersedia untuk komputer berbasis Windows, Linux, dan MacOS.

### Struktur dasar

Text

Description automatically generated with medium confidence

Setiap kode Arduino memiliki dua fungsi yang wajib ada. Yaitu fungsi setup yang akan berjalan pertama kali dan hanya sekali, dan fungsi loop yang akan berjalan setelah fungsi setup dan akan berulang terus menerus.

Dalam bahasa C yang belum dimodifikasi, struktur tersebut ekuivalen dengan kode berikut:

Text, letter

Description automatically generated

Contoh kode di bawah ini dapat membantu kalian memahami perbedaan kedua fungsi tersebut.

Text

Description automatically generated

### Data dan variabel

Berikut ini adalah tipe data dalam bahasa C/C++ yang wajib kalian ketahui:

|  |  |  |
| --- | --- | --- |
| **Type** | **Keyword** | **Range** |
| Integer | int | -32768 – 32767 |
| Boolean | bool, boolean | 0/1 or true/false |
| Char | char | Alphabet, numbers, and symbols |
| Byte | byte | 0 – 255 |
| Float | float | -3.4028e+38 – 3.4028e+38 |
| Void | void | none |

Kumpulan elemen data dalam pemrograman disebut *array.* Dalam bahasa C/C++, setiap data dalam array harus memiliki tipe yang sama. Tiap elemen memiliki indeks yang dapat digunakan untuk mengakses elemen tersebut. Indeks array adalah bilangan bulat berurut yang dimulai dari 0.

Cara mendeklarasikan variable dan array pada bahasa C dalam Arduino IDE terdapat pada contoh di bawah. Kalian dapat mendeklarasikan variabel di luar maupun di dalam fungsi setup/loop.

Text

Description automatically generated

### Built-in function and library/module

Terdapat fungsi yang sudah tertanam dalam IDE Arduino dan tidak ada di bahasa C biasa. Fungsi-fungsi ini merupakan API (Application Programming Interface) untuk berinteraksi dengan fitur-fitur pada mikrokontroler, seperi pin, waktu, dan memori.

Tabel di bawah ini berisi beberapa contoh fungsi *built-in* yang akan sering kalian gunakan. Fungsi/API lainnya dapat kalian pelajari pada laman resmi Arduino (<https://arduino.cc/en/Reference>), **lampiran di halaman terakhir file ini juga berisikan beberapa fungsi tersebut**.

|  |  |  |
| --- | --- | --- |
| **API** | **Args** | **Kegunaan** |
| pinMode(pin, mode) | pin= nomor pin;  mode= keyword mode pin, antara OUTPUT, INPUT, atau INPUT\_PULLUP | Mengatur pin tertentu untuk berperilaku seperti input/output. |
| digitalRead(pin) | pin= nomor pin | Membaca nilai secara biner dari pin input. |
| digitalWrite(pin, value) | pin= nomor pin;  value= nilai pin dapat berupa 0/1, HIGH/LOW, true/false. | Mengatur nilai dari pin output secara biner. |
| analogRead(pin) | pin= nomor pin | Membaca nilai secara analog dari pin input. |
| analogWrite(pin, value) | pin= nomor pin;  value= nilai pin dalam jangkauan tertentu, seperti 0-255 (bergantung pada resolusi sinyal analog) | Mengatur nilai dari pin output secara analog. |

Modul built-in di Arduino yang akan sering kalian temui adalah sebagai berikut:

|  |  |
| --- | --- |
| **Module** | **Kegunaan** |
| Serial | Mengatur komunikasi serial dengan protocol UART. Dapat digunakan saat Arduino berkomunikasi dengan komputer atau mikrokontroler lain. Komunikasi pada modul ini terbatas pada pin TX dan RX. |
| SoftwareSerial | Mengubah pin lain menjadi TX/RX untuk komunikasi UART. Digunakan Ketika pin TX/RX default tidak cukup untuk menangani komunikasi yang diperlukan. |
| Wire | Digunakan untuk protokol serial I2C. Protokol ini dapat menghubungkan beberapa perangkat dalam satu jalur komunikasi (pin SDA/SCL). Sering kali digunakan untuk membaca banyak sensor dan mengontrol display. |
| Servo | Digunakan untuk mengontrol perilaku motor servo dengan mudah. |

# Prosedur latihan

Di akhir modul in, ada beberapa program yang harus kalian coba sebagai latihan. Kode dapat diakses melalui repositori berikut: Terdapat folder dengan nama Part 1 dan Part 2. Folder tersebut yang akan kalian kerjakan.

PART 1:

1. Pada Part 1, gunakan IDE C online untuk memudahkan kalian mengeksekusi program. IDE dapat diakses pada laman: <https://www.onlinegdb.com/online_c_compiler>. Laman web akan langsung menampilkan workspace kalian. Kegiatan koding dapat dilakukan di file main.c. Pada bagian atas terdapat action bar, gunakan Run untuk menjalankan kode, dan stop untuk menghentikan eksekusi.
2. Copy dan paste kode dengan nama “1\_deklarasi\_var.c” ke dalam main.c pada IDE online. Kerjakan sesuai petunjuk yang terdapat pada comment, lalu jalankan.
3. Ulang untuk file berikutnya secara berurut.
4. Rekam/capture hasil program kalian.

PART 2:

1. Pada part ini, akan dilakukan simulasi Arduino secara online di website <https://wokwi.com/> . Namun, kalian boleh mengimplementasikannya pada Arduino fisik, sehingga tidak perlu melakukan simulasi lagi.
2. Pada website, pilih Arduino Uno pada Start a New Project. Workspace kalian akan segera tampil.
3. Kosongkan file diagram.json pada workspace Wokwi, salin isi file diagram.json pada repositori ke diagram.json pada workspace tersebut.
4. Buka file dengan nama “1\_blink\_example.ino”, salin isinya ke sketch.ino pada workspace.
5. Lengkapi kode sesuai instruksi (jika ada), kemudian jalankan simulasi dengan menekan tombol play pada workspace.
6. Rekam/capture hasil simulasi.

Susun file-file hasil eksekusi program dan simulasi dengan rapih dan jelas. Zip (kompres) ke dalam satu file .zip/.rar. Kemudian kirim kan ke email [riptekhmteunpad@gmail.com](mailto:riptekhmteunpad@gmail.com).

A picture containing text

Description automatically generated

*(end of file)*