

**LAPORAN PRAKTIKUM
ORGANISASI SISTEM KOMPUTER**

Judul: Menjalankan Arduino



**DISUSUN OLEH
Kristian Budi Pradana Putra (M0520043)**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET**

2021

BAB I

PENDAHULUAN

1.1 Tujuan Praktikum

Tujuan dari adanya praktikum ini adalah, supaya:

1. Praktikan dapat mengetahui dan memahami struktur bagian apa saja yang ada pada arduino,
2. Praktikan dapat menggunakan IDE compiler beserta *libraries*-nya yang berbasis bahasa pemrograman C yang sudah disederhanakan,
3. Praktikan dapat mengetahui, memahami, sekaligus membuat *sketch* yang digunakan untuk menyalakan LED dan *blink*,
4. Praktikan dapat mengetahui, memahami, sekaligus membuat *sketch* yang digunakan untuk membuat lampu flip-flop,
5. Praktikan dapat mengetahui, memahami, sekaligus memperoleh *sketch* yang digunakan untuk *digital read push button*,
6. Praktikan pada akhirnya dapat menggunakan ataupun menjalankan arduino.

1.2 Dasar Teori

1.2.1 Penjelasan Umum Arduino

Awal dari pemikiran untuk menciptakan sebuah perangkat elektronik kecil, namun multifungsi ini muncul di Institut Ivrea, Italia pada tahun 2005 oleh seorang yang bernama Hernando Barragan. Hernando Barragan berencana untuk membuat alat ini agar para siswa di Institut Ivrea dapat dengan mudah dan murah, membuat perangkat desain dan interaksi. Pemikiran Barragan ini kemudian direalisasikan oleh 2 orang bernama David Cuartielles dan Massimo Banzi. Awalnya program ini diberi nama “Arduin of Ivrea”. Kemudian, oleh kedua orang tersebut, perangkat ini kemudian diganti namanya menjadi “arduino”, dan nama inilah yang membuat perangkat ini terkenal, seperti yang kita dengar sampai sekarang

Arduino merupakan sebuah perangkat elektronik yang menjadi alternatif baru dalam dunia mikro-kontroller. Mikro-kontroller sendiri merupakan suatu perangkat dasar dalam komputer yang fungsional dalam sebuah *chip*, yang dimana terbentuk dari kumpulan gerbang *Integrated Circuit* (IC). Arduino dianggap sebagai sebuah alternatif karena dalam pengoperasiannya sangat mudah untuk digunakan, serta *software* dan *hardware*-nya pun memiliki fleksibilitas yang cukup tinggi. Di dalamnya, arduino memiliki sebuah *software* pendukung yang digunakan dalam melakukan pemrograman, memiliki memori untuk penyimpanan berskala kecil, memiliki sistem berupa *open hardware*, memiliki *processor* Atmel AVR, serta memiliki *Input/Output onboard*. Oleh karena hal-hal yang dimiliki di dalamnya tersebut, arduino sebenarnya dapat berdiri sendiri, sehingga disebut sebagai komputer versi mini.

1.2.2 Keunggulan Arduino

Arduino dapat dihubungkan/dipakai ke perangkat elektronik lainnya. Sebagai contoh, arduino dapat dihubungkan dengan internet sehingga dapat mengambil data atau mengirim data. Lalu, arduino juga bisa dipakai untuk menjadi otak dalam pembuatan robotika. Selain kedua contoh tersebut, arduino juga bisa dihubungkan/dipakai dalam MP3 *Player*, televisi, lampu LED, berbagai sensor digital, GPS, dan masih banyak lagi.

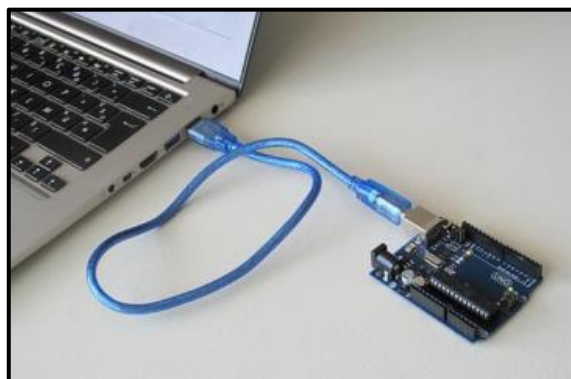
Tidak hanya karena aksesibilitas, pengoperasian yang sangat mudah, serta *software* dan *hardware* yang memiliki fleksibilitas yang cukup tinggi, arduino dipilih oleh banyak orang, perusahaan, maupun pemerintah, karena :

- a) Harganya yang terjangkau,
- b) Di dalamnya sudah tersedia berbagai macam *library* baik yang gratis maupun yang berbayar,
- c) Menggunakan bahasa pemrograman yang sudah sederhana (bahasa C),
- d) Bersifat *open source* sehingga pemakai dapat meng-*clone* skemanya ataupun bisa juga mencari referensi-referensi skema,
- e) Sudah menggunakan *port* USB untuk pemrogramannya,
- f) Memiliki fasilitas pendukung yang lengkap (seperti memori, *pin input/output*, dll),
- g) Tersedia *bootloader* di dalam *chip* arduino,
- h) Memiliki komunitas yang banyak, dan
- i) Dapat digunakan di berbagai OS pada perangkat elektronik.

Semua ini tak lain dan tak bukan adalah agar pengguna arduino dapat dengan mudah menggunakannya, sekalipun dia adalah orang awam.

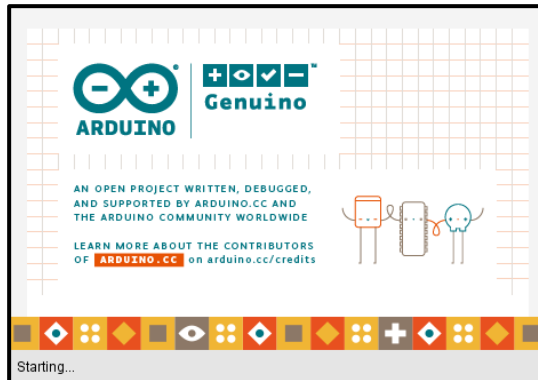
1.2.3 Langkah Awal Penggunaan Arduino

Arduino terdiri dari 2 bagian yaitu *hardware* dan *software*. Untuk *hardware*, karena ada banyak sekali jenis-jenis arduino, maka kita pilih 'Arduino Uno R3', karena kalau dilihat dari harga, dialah yang paling terjangkau, dengan fasilitas yang tercukupi, dan terdapat banyak shield yang mendukung. Pada dasarnya semua Arduino itu sama, hanya fitur dan desain saja yang membedakannya. Untuk menghubungkan Arduino Uno dengan komputer, kita dapat menggunakan kabel USB tipe A to tipe B.



Untuk *software*, apapun Arduino yang kita pakai, *software*-nya tetap memakai IDE arduino. IDE arduino adalah *software* yang digunakan untuk membuat *sketch* program arduino. Software ini dapat diinstal di Windows, Linux, ataupun MAC. Untuk mengunduh *software*-nya, kita bisa langsung mengunjungi laman resminya di

<https://www.arduino.cc/en/main/software>. Untuk versi yang terbaru, sudah tersedia driver USB, sehingga kita tinggal memilih instal bersama driver. Setelah Arduino selesai terinstal, kita bisa langsung mencoba membuka arduino, dan saat awal dibuka akan muncul tampilan seperti di bawah ini.



Setelah terbuka akan muncul jendela pemrograman Arduino, dengan antarmuka yang sangat sederhana dan mudah sekali untuk digunakan.



Pada saat pertama kali software Arduino 1.8.15 muncul, ada kode yang sudah ada di sana, yaitu void setup dan void loop. Semua kode yang dimasukkan ke dalam void setup akan dibaca sekali oleh Arduino, sedangkan kode yang dimasukkan ke void loop akan dibaca berulang.

Bahasa pemrograman dari Arduino ini sebenarnya mirip bahasa C yang digunakan pada AVR, tapi lebih disederhanakan lagi. Dengan bantuan library yang tersedia dimana-mana Arduino jadi lebih mudah untuk dipelajari dan lebih fleksibel untuk digunakan untuk membuat proyek apapun.

1.3 Peralatan atau Perangkat yang Digunakan

Peralatan dan perangkat yang saya gunakan selama mengeksplorasi serta memahami setiap bagian arduino, sekaligus mengerjakan laporan praktikum kali ini antara lain:

1. Set komputer,
2. Microsoft Office Word,

3. Mozilla Firefox,
4. Adobe Acrobat Reader DC, dan
5. IDE arduino.

BAB II

PEMBAHASAN

2.1 Langkah-Langkah Praktikum

2.1.1 Menyalakan LED dan *Blink*

- a) Menjalankan program arduino, bisa dari file *master* arduino yang kita simpan ataupun dari ikon di 'computer desktop'. Kemudian akan muncul program arduino IDE dengan tampilan *sketch* sesuai dengan tanggal dan bulan pada saat program dijalankan.
- b) Masuklah ke program arduino dan buat sketch untuk menyalakan LED dan *BLINK* dengan cara klik FILE >> EXAMPLES >> BASICS >> BLINK.
- c) Program yang baru saja ditulis pada menu editor dapat disimpan kedalam file yaitu dengan cara klik pada menu file >> save atau save as, pilih folder yang akan digunakan sebagai tempat menyimpan file, ketikkan nama file misal "Coba 1 Led" kemudian klik save.
- d) Setelah menyimpan *sketch*, kita bisa langsung menjalankan program yaitu dengan cara mengklik simbol *verify* atau dari menu *sketch*, pilih 'Verify/Compile' bisa juga dengan *shortcut* Ctrl + R. Apabila *verify* sukses, akan terdapat tulisan 'Done Compiling'.
- e) Setelah selesai diverifikasi, *sketch* sudah bisa langsung di-*upload* ke arduino yaitu dengan cara mengklik simbol *upload* atau dari menu file pilih 'upload', bisa juga dengan *shortcut* Ctrl + U.

2.1.2 Membuat Lampu Flip-Flop

- a) Menjalankan program arduino, bisa dari file *master* arduino yang kita simpan ataupun dari ikon di 'computer desktop'. Kemudian akan muncul program arduino IDE dengan tampilan *sketch* sesuai dengan tanggal dan bulan pada saat program dijalankan.
- b) Mengetikkan source code pada IDE arduino seperti berikut:

► Source Code :

```
void setup() {  
  // menginisialisasi pin digital 2 sebagai output.  
  pinMode(2, OUTPUT);  
}  
  
// fungsi loop yang berjalan secara terus menerus  
void loop() {  
  digitalWrite(2, HIGH); // menyalakan led  
  delay(100); // delay pada saat nyala  
  digitalWrite(2, LOW); // mematikan led  
  delay(100); // delay pada saat mati  
  digitalWrite(3, HIGH); // menyalakan led  
  delay(100); // delay pada saat nyala  
  digitalWrite(3, LOW); // mematikan led  
  delay(100); // delay pada saat mati  
}
```

- c) Program yang baru saja ditulis pada menu editor dapat disimpan kedalam file yaitu dengan cara klik pada menu file >> save atau save as, pilih folder yang akan digunakan sebagai tempat menyimpan file, ketikkan nama file misal “Coba 1 Led” kemudian klik save.
- d) Setelah menyimpan *sketch*, kita bisa langsung menjalankan program yaitu dengan cara mengklik simbol *verify* atau dari menu *sketch*, pilih ‘Verify/Compile’ bisa juga dengan *shortcut* Ctrl + R. Apabila *verify* sukses, akan terdapat tulisan ‘Done Compiling’.
- e) Setelah selesai diverifikasi, *sketch* sudah bisa langsung di-upload ke arduino yaitu dengan cara mengklik simbol *upload* atau dari menu file pilih ‘upload’, bisa juga dengan *shortcut* Ctrl + U.

2.1.3 Digital Read Push Button

- a) Menjalankan program arduino, bisa dari file *master* arduino yang kita simpan ataupun dari ikon di ‘computer desktop’. Kemudian akan muncul program arduino IDE dengan tampilan *sketch* sesuai dengan tanggal dan bulan pada saat program dijalankan.
- b) Memperoleh *sketch* melalui sampel pada FILE => EXAMPLES => DIGITAL => BUTTON.
- c) Meng-*compile* program untuk menjalankan program pada rangkaian.

2.2 Pembahasan

2.2.1 Menyalakan LED dan *Blink*

Setelah mengikuti langkah-langkah yang disediakan, bentuk *source code* dari percobaan yang pertama yaitu:



```

Coba_1_Led | Arduino 1.8.15
File Edit Sketch Tools Help

Coba_1_Led
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Done compiling.

Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

13 Arduino Uno

```

Pada fungsi ‘void setup’, terdapat perintah ‘pinMode(LED_BUILTIN, OUTPUT);’ yang membuat pin LED_BUILTIN menjadi *output* dari arduino, yang

mana pin ini terhubung dengan pin positif LED. Karena terhubung dengan pin positif LED, *output* dari Arduino akan berupa perubahan kondisi dari LED. Perintah ini tidak perlu untuk diulang, karena termasuk dalam bentuk perintah utama yang hanya akan dibaca pertama kali saat program berjalan, sehingga dimasukkan dalam fungsi ‘void setup’.

Fungsi ‘void loop’ yang akan dibaca berulang, merupakan perintah yang akan menentukan bentuk *output* yang akan dikeluarkan. Perintah ‘digitalWrite(LED_BUILTIN, HIGH);’ berarti pin LED_BUILTIN mengeluarkan *output* untuk membuat LED dalam keadaan *high* atau menyala, sedangkan ‘digitalWrite(LED_BUILTIN, LOW);’ memerintahkan yang sebaliknya. Setelah perintah ‘digitalWrite’, terdapat perintah ‘delay(1000);’ yang merupakan sebuah perintah untuk menunda perintah selanjutnya berjalan. Jadi, perintah selanjutnya tersebut akan dapat dijalankan setelah 1000 mili-detik atau 1 detik.

Dilihat dari *source code* di atas, maka *output* pada arduino board diperkirakan adalah LED akan menyala, lalu mati setelah satu detik, kemudian akan menyala-mati lagi secara berulang kali seperti berkelap-kelip. Hal ini akan sesuai dengan tujuan dari percobaan 1, yaitu menyalakan LED dan blink.

2.2.2 Membuat Lampu Flip-Flop

Setelah mengikuti langkah-langkah yang disediakan, bentuk *source code* dari percobaan yang kedua yaitu:



```
Coba_IL_FF | Arduino 1.8.15
File Edit Sketch Tools Help

Coba_IL_FF
void setup() {
  // menginisialisasi pin digital 2 sebagai output.
  pinMode(2, OUTPUT);
}

// fungsi loop yang berjalan secara terus menerus
void loop() {
  digitalWrite(2, HIGH); // menyalakan led
  delay(100); // delay pada saat nyala
  digitalWrite(2, LOW); // mematikan led
  delay(100); // delay pada saat mati
  digitalWrite(3, HIGH); // menyalakan led
  delay(100); // delay pada saat nyala
  digitalWrite(3, LOW); // mematikan led
  delay(100); // delay pada saat mati
}

Done compiling.

Sketch uses 958 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

15 Arduino Uno
```

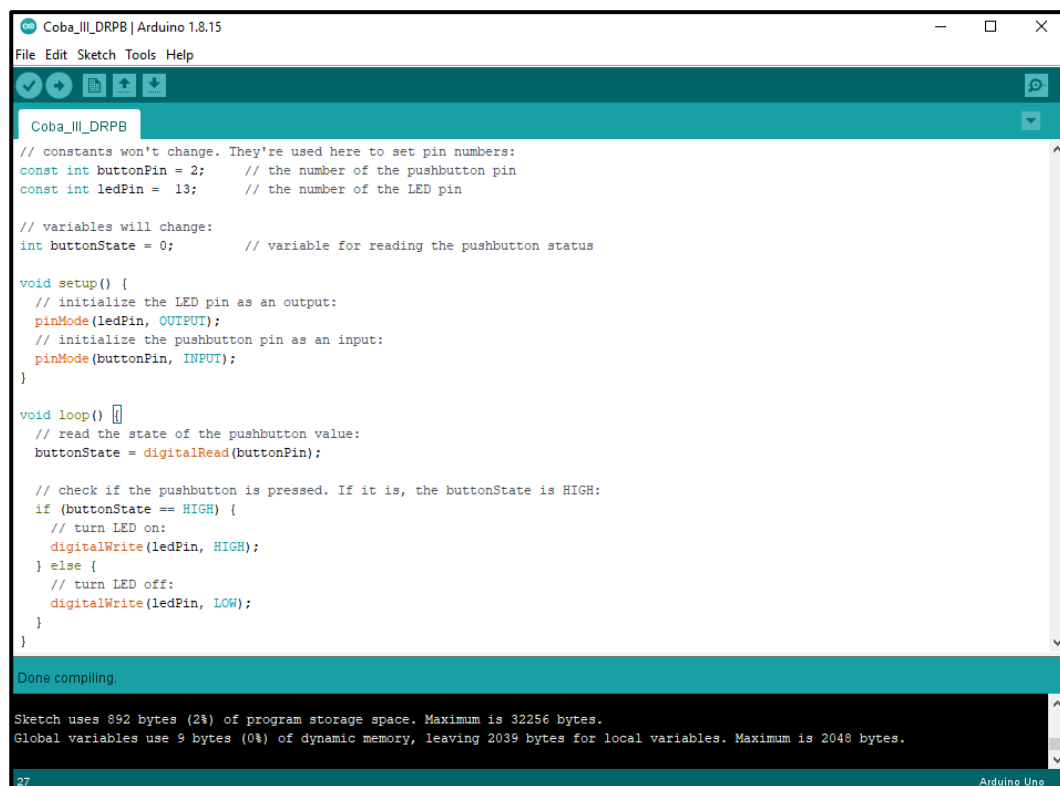
Pada fungsi ‘void setup’, terdapat perintah ‘pinMode(2, OUTPUT);’ yang membuat pin 2 menjadi *output* dari arduino, yang mana pin ini terhubung dengan pin positif LED. Karena terhubung dengan pin positif LED, *output* dari arduino akan berupa perubahan kondisi dari LED. Perintah ini tidak perlu untuk diulang, karena termasuk dalam bentuk perintah utama yang hanya akan dibaca pertama kali saat program dijalankan, sehingga dimasukkan pada fungsi ‘void setup’.

Fungsi ‘void loop’ yang akan dibaca berulang, merupakan perintah yang akan menentukan bentuk *output* yang akan dikeluarkan. Perintah ‘digitalWrite(2, HIGH);’ berarti pin 2 mengeluarkan *output* untuk membuat LED dalam keadaan *high* atau menyala, sedangkan ‘digitalWrite(2, LOW);’ memerintahkan yang sebaliknya. Pada perintah ‘digitalWrite(3, HIGH);’ dan ‘digitalWrite(3, LOW);’, juga memerintahkan hal yang sama, namun terhadap pin 3. Setelah perintah ‘digitalWrite’, terdapat ‘delay(100);’, yang merupakan sebuah perintah untuk menunda perintah selanjutnya berjalan. Jadi, perintah selanjutnya tersebut akan dapat dijalankan setelah 100 mili-detik atau 0.1 detik.

Dilihat dari *source code* di atas, maka *output* pada arduino board diperkirakan adalah LED pada pin 2 akan menyala, lalu mati setelah 0.1 detik. Kemudian setelah 0.1 detik selanjutnya, pin 3 akan menyala, lalu mati setelah 0.1 detik lagi. Setelah itu, akan menyala lagi pada pin 2 dan pin 3 secara berulang. Kedua lampu LED (dua *output*) yang dapat menyala dan mati (dua kondisi stabil) secara bergantian ini merupakan penerapan dari rangkaian flip-flop. Hal ini sesuai dengan tujuan dari percobaan 2, yaitu membuat lampu flip-flop.

2.2.3 Digital Read Push Button

Setelah mengikuti langkah-langkah yang disediakan, bentuk *source code* dari percobaan yang kedua yaitu:



```
Coba_III_DRPB | Arduino 1.8.15
File Edit Sketch Tools Help

Coba_III_DRPB

// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;      // the number of the LED pin

// variables will change:
int buttonState = 0;        // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}

Done compiling.

Sketch uses 892 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

27 Arduino Uno
```

Sebelum menuju ke fungsi ‘void setup’, program menggunakan ‘const int buttonPin = 2;’ dan ‘const int ledPin = 13;’ untuk mendefinisikan pin 2 sebagai variabel buttonPin dan pin 13 sebagai variabel ledPin.

Pada fungsi ‘void setup’, terdapat perintah ‘pinMode(ledPin, OUTPUT);’ yang membuat pin 13 menjadi *output* dari arduino, yang mana pin ini terhubung dengan pin positif LED. Kemudian terdapat juga perintah ‘pinMode(buttonPin, INPUT);’

yang membuat pin 2 menjadi *input* dari arduino. *Output* dari arduino akan berupa perubahan kondisi dari LED yang ditentukan berdasarkan *input* dari *push button* pada pin 2. Perintah ini tidak perlu untuk diulang, karena termasuk dalam bentuk perintah utama yang hanya akan dibaca pertama kali saat program dijalankan, sehingga dimasukkan pada fungsi 'void setup'.

Fungsi 'void loop' yang akan dibaca berulang, merupakan perintah yang akan menentukan bentuk *output* yang akan dikeluarkan. Perintah 'digitalRead(buttonPin);' yang digunakan untuk membaca *input* dari *button*, didefinisikan sebagai 'buttonState'. Fungsi 'if (buttonState == HIGH)' memiliki arti bahwa jika *button* bernilai *high* atau 1, maka akan menjalankan perintah 'digitalWrite(ledPin, HIGH);'. Perintah tersebut berarti mengeluarkan *output* bahwa LED pada pin 13 akan bernilai *high* atau menyala. Fungsi 'else' membaca *input* selain kondisi pada fungsi 'if', lalu menjalankan perintah sebaliknya dari fungsi 'if' tersebut.

Dilihat dari *source code* di atas, *output* pada arduino board diperkirakan adalah, LED pada pin 13 akan hidup saat *push button* pada pin 2 bernilai 1 atau *high*, dan akan mati jika *push button* bernilai sebaliknya. Nyala-matinya lampu LED didasarkan pada nilai *input* yang dibaca berdasarkan nilai dari *push button* ini sesuai dengan tujuan percobaan 3, yaitu *digital read push button*.

BAB III

PENUTUP

3.1 Kesimpulan

Secara garis besar, dapat disimpulkan bahwa arduino adalah sebuah perangkat elektronik yang menjadi alternatif baru dalam dunia mikro-kontroller. Mikro-kontroller sendiri merupakan suatu perangkat dasar dalam komputer yang fungsional dalam sebuah *chip*, yang dimana terbentuk dari kumpulan gerbang *Integrated Circuit* (IC).

Arduino terdiri dari 2 bagian yaitu Hardware dan Software. Selain itu, arduino juga memiliki banyak kelebihan dibandingkan dengan produk lain, salah satunya adalah mudah dipelajari. Bahasa pemrograman untuk arduino sendiri adalah bahasa C yang sudah disederhanakan, sehingga akan sangat memudahkan para pemula. Bahkan, orang yang tidak mempunyai background dari teknik elektro maupun programmer sekalipun, pasti bisa melakukan pemrograman arduino.

Contoh dari proyek sederhana yang bisa dibuat oleh pemula adalah menyalakan LED dan *blink*, membuat lampu flip-flop, dan *digital read push button*. Dalam pembuatan *sketch*, proyek-proyek tersebut menggunakan dua fungsi utama yaitu *setup* dan *loop*. Semua kode yang dimasukkan ke dalam fungsi ‘void setup’ akan dibaca sekali oleh arduino, sedangkan kode yang dimasukkan ke fungsi ‘void loop’ akan dibaca berulang kali.

3.2 Daftar Pustaka

1. arduino.cc. (2018, 5 Februari). What is Arduino?. Diakses pada 23 Juni 2021, dari <https://www.arduino.cc/en/Guide/Introduction>.
2. id.wikipedia.org. (2020, 24 Mei). Arduino. Diakses pada 23 Juni 2021, dari <https://id.wikipedia.org/wiki/Arduino>.
3. Khaerani, Dinda. (2021, 18 Juni). Menjalankan Arduino. Diakses pada 24 Juni 2021, dari <https://classroom.google.com/u/2/c/MzA3ODk5ODYxMTc4/a/MzY0Mjk0NTU5MTUy/details>.
4. Rahman, A. B. (2020, 19 Desember). Pengenalan Arduino. Diakses pada 24 Juni 2021, dari <https://classroom.google.com/u/2/c/MTc4MzkyNDc5NjUy>.