

Praktikum LAB 12

Diperuntukkan untuk memenuhi salah satu tugas praktikum Mata Kuliah Aplikasi Mikrokontroler



Praktikum	: Aplikasi Mikrokontroler
Praktikum ke	: 12
Tanggal Praktikum	: Selasa, 5 Januari 2021
Tanggal Pengumpulan Laporan	: Senin, 11 Januari 2021
Nama dan NIM	: 1. Amir Husein (181344003)
Kelas	: 3-TNK
Instruktur	: 1. Ferry Satria, BSEE., M.T 2. Rahmawati Hasanah, S.ST., M.T

Politeknik Negeri Bandung
Tahun Ajaran 2020/2021

I. TUJUAN

- Mahasiswa dapat memahami prinsip penggunaan perintah If, If Else, For, While Loop dan Array pada pemrograman C Arduino
- Mahasiswa mampu melakukan koneksi antara program pada Arduino IDE dengan simulator Proteus
- Mahasiswa mampu mengonversi nilai tegangan menjadi bentuk diskrit untuk diolah pada mikrokontroler Arduino.

II. LANDASAN TEORI

1. Arduino

Arduino adalah sebuah minimum sistem mikrokontroler bersifat open-source yang banyak digunakan untuk membangun sebuah project elektronika. Platform Arduino berisi dua yaitu hardware berupa board dan sebuah software atau IDE (Integrated Development Environment) yang berjalan pada komputer, digunakan untuk menulis dan mengisikan program ke board Arduino.

Platform Arduino menjadi sangat populer bagi orang-orang yang ingin memulai belajar elektronika khususnya mikrokontroler, karena dengan menggunakan Arduino tidak lagi memerlukan hardware tambahan (sering disebut downloader) untuk mengisikan program kedalam board mikrokontroler, tapi hanya perlu kabel USB saja yang disambungkan dari komputer ke board Arduino. Bahasa pemrograman yang dituliskan pada Arduino IDE menggunakan bahasa pemrograman C++ yang telah disederhanakan, sehingga dapat lebih mudah dimengerti. Sebuah board Arduino didesain dengan standar bentuk board serta posisi dan susunan pin/port sehingga dapat lebih mudah digunakan dan diakses dengan perangkat lain.

2. Arduino IDE

IDE itu merupakan kependekan dari Integrated Development Environment, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (Sketch) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama Bootlader yang berfungsi sebagai penengah antara compiler Arduino dengan mikrokontroler.

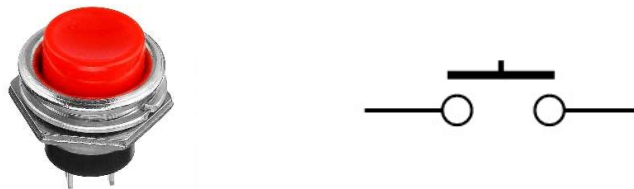
Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut Wiring yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

3. Array

Array merupakan salah satu tipe data pada pemrograman C di Arduino. Array merupakan susunan data yang terindeks, dengan indeks pertama diawali 0, serta data pada array bisa berupa int, string, bool, maupun lainnya. Data pada array bisa diinisialisasi langsung dengan menggunakan simbol kurung kurawal '{}', serta untuk melakukan pointing indeks dapat menggunakan kurung siku setelah variabel array itu sendiri. Seperti contoh terdapat array 'abjad' sebagai berikut yaitu `abjad[] = {a,c,g,j,n,d,a}`, maka untuk mendapatkan nilai dari g, dapat dipanggil dengan cara melakukan pointing dengan syntax `abjad[2]`.

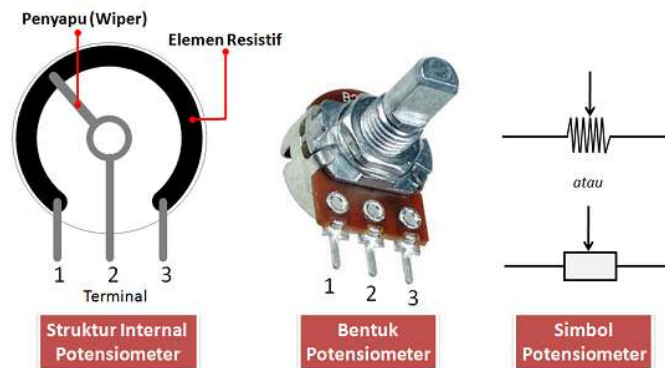
4. Push Button

Push Button merupakan salah satu komponen pasif elektronika, yang berfungsi untuk menghubungkan dua terminal sejalan untuk dapat mengalirkan arus listrik. Akan tetapi, terdapat juga push button yang dapat menghubungkan lebih dari dua terminal. Push button akan menghubungkan dua terminal secara langsung ketika ditekan, sementara ketika dilepas, push button akan kembali ke posisi semula yaitu open circuit. Dibawah ini adalah gambar dari push button dua kaki beserta simbolnya pada rangkaian.



5. Potensiometer

Potensiometer adalah salah satu jenis Resistor yang Nilai Resistansinya dapat diatur sesuai dengan kebutuhan Rangkaian Elektronika ataupun kebutuhan pemakainya. Potensiometer merupakan Keluarga Resistor yang tergolong dalam Kategori Variable Resistor. Secara struktur, Potensiometer terdiri dari 3 kaki Terminal dengan sebuah shaft atau tuas yang berfungsi sebagai pengaturnya. Gambar dibawah ini menunjukkan Struktur Internal Potensiometer beserta bentuk dan Simbolnya.



Sebuah Potensiometer (POT) terdiri dari sebuah elemen resistif yang membentuk jalur (track) dengan terminal di kedua ujungnya. Sedangkan terminal lainnya (biasanya berada di tengah) adalah Penyapu (Wiper) yang dipergunakan untuk menentukan

pergerakan pada jalur elemen resistif (Resistive). Pergerakan Penyapu (Wiper) pada Jalur Elemen Resistif inilah yang mengatur naik-turunnya Nilai Resistansi sebuah Potensiometer. Elemen Resistif pada Potensiometer umumnya terbuat dari bahan campuran Metal (logam) dan Keramik ataupun Bahan Karbon (Carbon). Berdasarkan Track (jalur) elemen resistif-nya, Potensiometer dapat digolongkan menjadi 2 jenis yaitu Potensiometer Linear (Linear Potentiometer) dan Potensiometer Logaritmik (Logarithmic Potentiometer).

III. SOAL

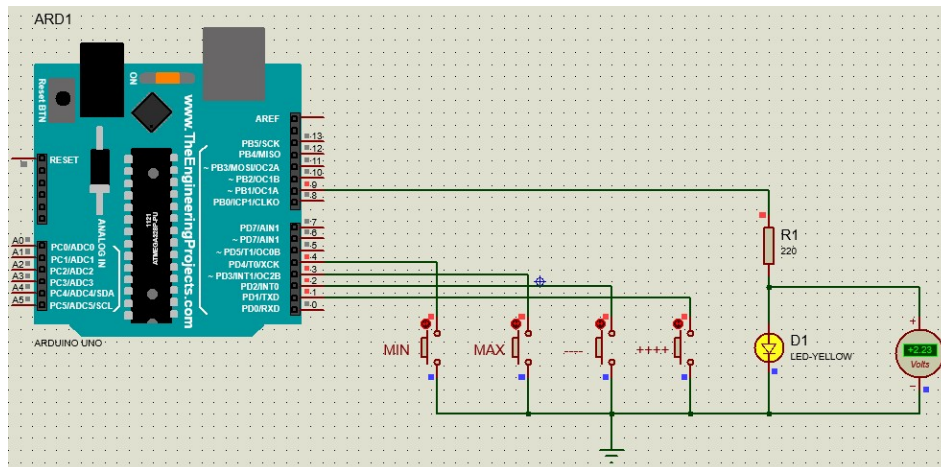
1. Buatlah rangkaian serta program arduino untuk mengatur tingkat kecerahan LED menggunakan push button.
2. Buatlah rangkaian serta program arduino untuk mengatur tingkat kecerahan LED menggunakan potensiometer.
3. Buatlah program serta rangkaian arduino dengan output 4 buah LED, yang masing-masing LED terhubung pada kaki 8, 9, 10, dan 11. Hidupkan LED secara bertahap mulai dari LED1 hingga LED4 menggunakan push button.
4. Buatlah program serta rangkaian arduino dengan output 4 buah LED, yang masing-masing LED terhubung pada kaki 8, 9, 10, dan 11. Hidupkan LED secara bertahap mulai dari LED1 hingga LED4 menggunakan potensiometer.

IV. HASIL PROGRAM

1. Program :

```
const int pinLED = 9;
const int pinButton[] = {1, 2, 3, 4};
int brightness = 0;
void setup() {
    pinMode(pinLED, OUTPUT);
    for(int i=0; i<4; i++){
        pinMode(pinButton[i], INPUT);
        digitalWrite(pinButton[i], HIGH);
    }
}
void loop() {
    if(digitalRead(pinButton[0]) == LOW){
        brightness++;
    }
    else if(digitalRead(pinButton[1]) == LOW){
        brightness--;
    }
    else if(digitalRead(pinButton[2]) == LOW){
        brightness = 255;
    }
    else if(digitalRead(pinButton[3]) == LOW){
        brightness = 0;
    }
    brightness = constrain(brightness, 0, 255);
    analogWrite(pinLED, brightness);
    delay(25);
}
```

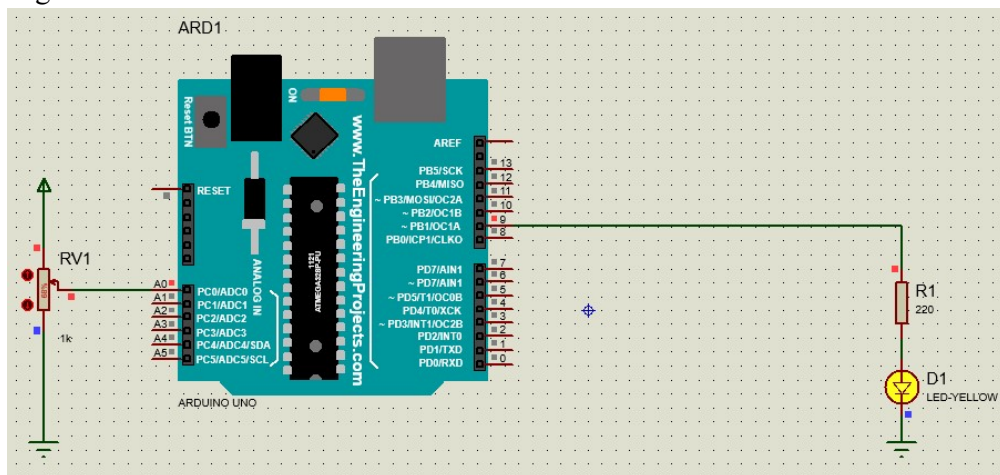
Rangkaian:



2. Program:

```
const int pinLED = 9;
const int pinPot = A0;
int sensor = 0;
int brightness = 0;
void setup() {
  pinMode(pinLED, OUTPUT);
  pinMode(pinPot, INPUT);
}
void loop() {
  sensor = analogRead(pinPot);
  brightness = map(sensor, 0, 1023, 0, 255);
  analogWrite(pinLED, brightness);
}
```

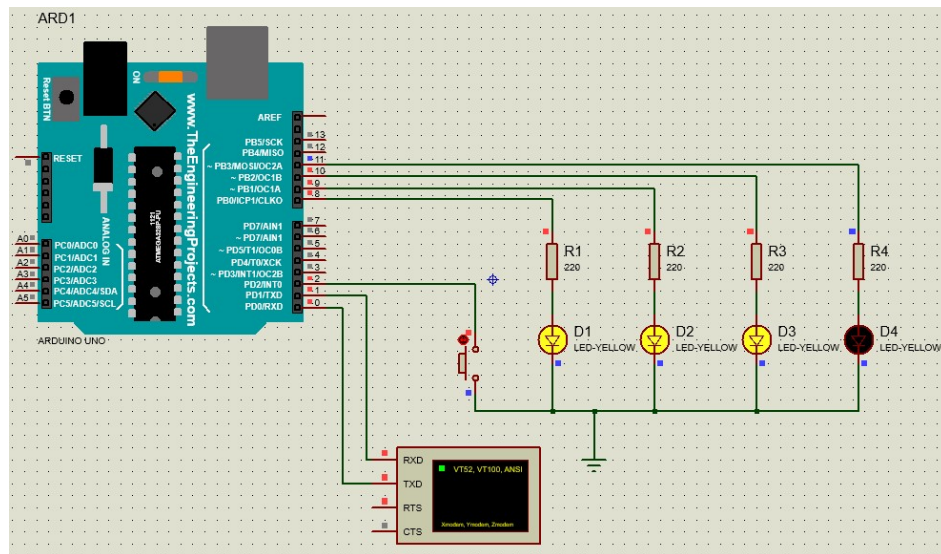
Rangkaian:



3. Program:

```
const int pinLED[] = {8, 9, 10, 11};
const int pinButton = 2;
int i = 0;
void setup() {
  for(int i=0; i<4; i++){
    pinMode(pinLED[i], OUTPUT);
  }
  pinMode(pinButton, INPUT);
  digitalWrite(pinButton, HIGH);
}
void loop() {
  buttonPressed();
}
void buttonPressed() {
  if(digitalRead(pinButton) == LOW){
    i++;
    if(i == 5){
      i = 0;
      for(int i=0; i<4; i++){
        digitalWrite(pinLED[i], LOW);
        delay(25);
      }
    }else if(i != 5){
      digitalWrite(pinLED[i-1], HIGH);
      delay(500);
    }
  }
}
```

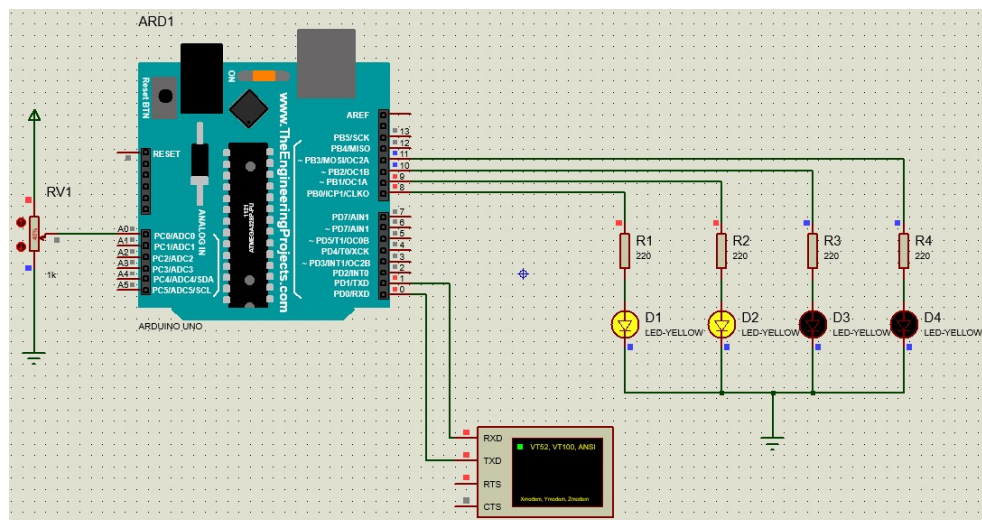
Rangkaian:



4. Program:

```
const int pinLED[] = {8, 9, 10, 11};
const int pinPot = A0;
void setup() {
    Serial.begin(9600);
    pinMode(pinPot, INPUT);
    for(int i=0; i<4; i++){
        pinMode(pinLED[i], OUTPUT);
    }
}
void loop() {
    nyala(analogRead(pinPot));
}
void nyala(int analogIndex){
    if(analogIndex < 251){
        ledConfig(HIGH, LOW, LOW, LOW);
    }
    else if(250 < analogIndex < 501){
        ledConfig(HIGH, HIGH, LOW, LOW);
    }
    else if(500 < analogIndex < 751){
        ledConfig(HIGH, HIGH, HIGH, LOW);
    }
    else if(analogIndex > 750){
        ledConfig(HIGH, HIGH, HIGH, HIGH);
    }
}
void ledConfig(bool log0, bool log1, bool log2, bool log3){
    digitalWrite(pinLED[0], log0);
    digitalWrite(pinLED[1], log1);
    digitalWrite(pinLED[2], log2);
    digitalWrite(pinLED[3], log3);
    delay(100);
}
```

Rangkaian:



V. ANALISIS

1. Pada soal ini, akan dibuat program untuk mengatur tingkat kecerahan LED menggunakan push button. Hal pertama yang perlu dilakukan adalah menginisialisasi pin untuk LED dengan nama variabel **pinLED** dengan value 9 dan pin untuk push button sebanyak 4 pin yaitu dari nomor 1 sampai 4, selain itu juga, dibuat variabel untuk menyimpan nilai brightness itu sendiri dengan nilai awal 0.

Setelah itu, pada fungsi **void setup()**, perlu diatur bahwa pin nomor untuk LED akan bekerja sebagai OUTPUT, dan 4 pin untuk push button akan bekerja sebagai masukan atau INPUT. Pada fungsi **void loop()**, akan dilakukan pengondisian sebanyak 4 kali, yang pertama adalah apabila pin push button pertama ditekan, maka brightness akan bertambah 1, push button yang kedua adalah untuk mengurangi brightness sebanyak 1, push button yang ketiga untuk mengatur brightness ke nilai maksimal (dalam hal ini adalah 255), dan push button terakhir untuk mengatur tingkat brightness ke nilai minimal (bernilai 0), untuk mendapatkan masukan dari push button ini, digunakan method **digitalRead**, dan program akan mendeteksi apabila masukannya ini bernilai LOW (pada posisi idle, pin push button akan bernilai HIGH). Setelah pengondisian ini selesai, selanjutnya adalah melakukan pembatasan nilai untuk brightness dengan method **constraint**, dan setelah itu, melakukan penulisan ke pin LED dengan method **analogWrite** dengan 2 parameter, yaitu pin LED itu sendiri dan tingkat brightness, setelahnya disertai delay selama 25ms. Perlu dicatat bahwasanya pin output yang mendukung method **analogWrite** haruslah pin PWM.

2. Pada soal ini, akan dibuat program untuk mengatur tingkat kecerahan LED menggunakan potensiometer. Pin untuk LED diisi dengan value 9, dimana pin nomor 9 pada Arduino ini adalah pin yang mendukung PWM untuk melakukan manipulasi nilai tegangan dan frekuensi untuk output. Pin untuk potensiometer berada pada pin A0, dikarenakan output dari pengaturan potensiometer merupakan sinyal analog yang perlu dikonversi ke digital, maka pin A0 digunakan sebagai pin input untuk potensiometer.

Selain itu, dilakukan inisialisasi variabel untuk menyimpan nilai keluaran dari potensiometer dengan nama **sensor** dan variabel untuk menyimpan nilai tingkat kecerahan dengan nama **brightness**, kedua variabel ini diisi dengan nilai awalan 0. Pada fungsi **void setup()**, pin LED diatur sebagai OUTPUT dan pin potensiometer sebagai INPUT. Pada fungsi **void loop()**, variabel sensor akan mendapatkan value dari nilai resistansi pada pin potensiometer, dengan menggunakan method **analogRead**, nilai yang akan didapatkan akan berkisar dari 0 hingga 1023, sedangkan pada sisi output PWM, hanya dapat menerima nilai digital dengan range 0 sampai dengan 255. Maka dari itu, diperlukan **mapping** dengan menggunakan method **map**, nilai dari variabel sensor yang didapatkan dari pin analog akan di map menjadi nilai yang berkisar antara 0 hingga 255, dan nilai mapping ini akan diisikan pada variabel **brightness**. Langkah terakhir ialah melakukan penulisan pada pin LED output dengan method **analogWrite**.

3. Pada program ini, nomor pin LED diinisialisasikan menggunakan array seperti biasa dan pin push button pun sama di pin nomor 2. Setelah itu, dibuat variabel **i** dengan nilai awal 0 sebagai variabel untuk menampung indeks push button pada LED. Pada fungsi

void setup(), pin LED dengan variabel **pinLED** diatur sebagai OUTPUT dan **pinButton** untuk push button sebagai INPUT dengan nilai awal HIGH.

Pada fungsi **void loop()**, hanya akan dipanggil fungsi tambahan yaitu **buttonPressed()**, dimana fungsi ini berisi pengondisian ketika **pushButton** ditekan (bernilai LOW), maka nilai variabel **i** akan increment dan dilakukan penngondisian, apabila **i** telah mencapai 5, maka **i** akan kembali ke 0 dan semua LED padam, sedangkan apabila belum mencapai 5, variabel **i** ini akan bertindak sebagai index untuk LEDnya itu sendiri agar menyala, misal **i** = 1, maka LED1 akan menyala, **i** = 2, LED2 akan menyala, dan seterusnya diikuti delay 500ms.

4. Pada program ini, pin LED diinisialisasikan menggunakan array dari 8 hingga 11, dan pin untuk potensiometer di pin A0. Pada fungsi **void setup()**, **pinPot** untuk potensiometer diatur sebagai INPUT dan **pinLED** untuk LED sebagai OUTPUT. Pada fungsi **void loop()**, hanya akan dipanggil fungsi **nyala**, dengan satu buah parameter yaitu **analogIndex** bertipe integer, yang nanti diisi dengan fungsi **analogRead(pinPot)**, untuk mendapatkan nilai konversi tegangan analog yang berkisar dari 0 hingga 1023. Isi dari fungsi **nyala** ini adalah berupa pengondisian, ketika **analogIndex** kurang dari 250, maka akan dipanggil fungsi **ledConfig** untuk menyalakan kombinasi LED dan hanya LED1 yang HIGH. Kenaikan kombinasi nyala LED adalah setiap perubahan 250 **analogIndex**.

VI. KESIMPULAN

Pada praktikum ini dapat diambil kesimpulan bahwa pada pemrograman Arduino menggunakan bahasa C, dapat digunakan beberapa perintah seperti halnya pemrograman C seperti biasa, yaitu seperti penggunaan IF, IF ELSE, FOR, dan WHILE LOOP. Selain itu juga, simulator Proteus dapat digunakan sebagai sarana dalam mengamati hasil compile program C Arduino tersebut, tanpa bersusah payah untuk menghubungkan perangkat Arduino langsung ke PC. Penggunaan array dalam pemrograman C Arduino memudahkan dalam melakukan inisialisasi nilai yang banyak agar meminimalisasi penggunaan syntax yang boros dan berlebihan.

VII. DAFTAR PUSTAKA

- Anonim. 2019. *Apa itu Arduino?*. Bluino. Diakses pada 11 Desember 2020. https://www.bluino.com/2019/09/apa-itu-arduino_13.html
- Kho, Dickson. 2019. *Pengertian dan Fungsi Potensiometer*. Teknik Elektronika. Diakses pada 6 Januari 2021. <https://teknikelektronika.com/pengertian-fungsi-potensiometer/>
- Sinau. 2016. *Mengenal Arduino Software (IDE)*. Sinau Arduino. Diakses apda 11 Desember 2020. <https://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>