

Praktikum LAB 14

Diperuntukkan untuk memenuhi salah satu tugas praktikum Mata Kuliah Aplikasi Mikrokontroler



Praktikum	: Aplikasi Mikrokontroler
Praktikum ke	: 14
Tanggal Praktikum	: Kamis, 21 Januari 2021
Tanggal Pengumpulan Laporan	: Senin, 25 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-perintah 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.
- Mahasiswa mengerti mengenai penggunaan switch dan 7-segment pada 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







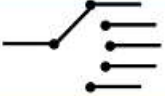

Array merupakan salah satu tipe data pada pemrograman C di Arduino. Array merupakan susunan data yang terindeks, dengan indeks pertama diawali 0, serta data pada array bisa berupa int, string, bool, maupun lainnya. Data pada array bisa diinisialisasi langsung dengan menggunakan simbol kurung kurawal '{}', 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. Saklar

Saklar atau dalam bahasa Inggris disebut Switch adalah salah satu komponen yang penting dalam setiap rangkaian atau perangkat elektronik. Seperti pada artikel yang disebutkan sebelumnya, Saklar atau Switch adalah perangkat yang digunakan untuk memutuskan atau menghubungkan aliran arus listrik. Meskipun saat ini telah banyak yang menggunakan saklar atau switch elektronik yang menggunakan sensor ataupun rangkaian yang terdiri komponen semikonduktor seperti transistor, IC dan dioda. Namun saklar mekanik atau mechanical switch masih tetap memegang peranan penting pada hampir semua perangkat atau peralatan listrik dan elektronik.

Saklar pada dasarnya merupakan perangkat mekanik yang terdiri dari dua atau lebih terminal yang terhubung secara internal ke bilah atau kontak logam yang dapat dibuka dan ditutup oleh penggunaannya. Aliran listrik akan mengalir apabila suatu kontak dihubungkan dengan kontak lainnya. Sebaliknya, aliran listrik akan terputus apabila hubungan tersebut dibuka atau dipisahkan. Selain sebagai komponen untuk menghidupkan (ON) dan mematikan (OFF) perangkat elektronik, Saklar sering juga difungsikan sebagai pengendali untuk mengaktifkan fitur-fitur tertentu pada suatu rangkaian listrik. Contohnya seperti pengatur tegangan pada pencatu daya, Sebagai pengatur Volume di Ponsel ataupun sebagai pengatur.

Berikut ini adalah jenis-jenis Saklar listrik mekanik yang digolongkan berdasarkan cara gerakan saklarnya.

Push Button Switch	Toggle Switch	Selector Switch	Limit Switch
			
			

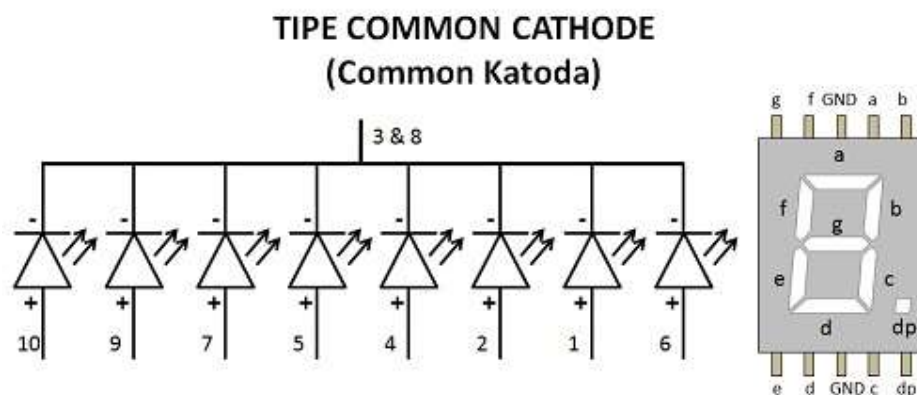
5. 7-Segment

Seven Segment Display (7 Segment Display) dalam bahasa Indonesia disebut dengan Layar Tujuh Segmen adalah komponen Elektronika yang dapat menampilkan angka desimal melalui kombinasi-kombinasi segmennya. Seven Segment Display pada umumnya dipakai pada Jam Digital, Kalkulator, Penghitung atau Counter Digital, Multimeter Digital dan juga Panel Display Digital seperti pada Microwave Oven ataupun Pengatur Suhu Digital . Seven Segment Display pertama diperkenalkan dan dipatenkan pada tahun 1908 oleh Frank. W. Wood dan mulai dikenal luas pada tahun 1970-an setelah aplikasinya pada LED (Light Emitting Diode).

Seven Segment Display memiliki 7 Segmen dimana setiap segmen dikendalikan secara ON dan OFF untuk menampilkan angka yang diinginkan. Angka-angka dari 0 (nol) sampai 9 (Sembilan) dapat ditampilkan dengan menggunakan beberapa kombinasi Segmen. Selain 0 – 9, Seven Segment Display juga dapat menampilkan Huruf Hexadecimal dari A sampai F. Segmen atau elemen-elemen pada Seven Segment Display diatur menjadi bentuk angka “8” yang agak miring ke kanan dengan tujuan untuk mempermudah pembacaannya. Pada beberapa jenis Seven Segment Display, terdapat juga penambahan “titik” yang menunjukkan angka koma decimal. Terdapat beberapa jenis Seven Segment Display, diantaranya adalah Incandescent bulbs, Fluorescent lamps (FL), Liquid Crystal Display (LCD) dan Light Emitting Diode (LED).

LED 7 Segmen Tipe Common Cathode (Katoda)

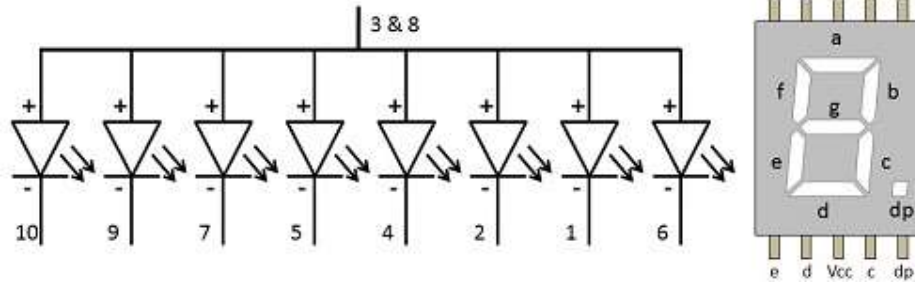
Pada LED 7 Segmen jenis Common Cathode (Katoda), Kaki Katoda pada semua segmen LED adalah terhubung menjadi 1 Pin, sedangkan Kaki Anoda akan menjadi Input untuk masing-masing Segmen LED. Kaki Katoda yang terhubung menjadi 1 Pin ini merupakan Terminal Negatif (-) atau Ground sedangkan Signal Kendali (Control Signal) akan diberikan kepada masing-masing Kaki Anoda Segmen LED.



LED 7 Segmen Tipe Common Anode (Anoda)

Pada LED 7 Segmen jenis Common Anode (Anoda), Kaki Anoda pada semua segmen LED adalah terhubung menjadi 1 Pin, sedangkan kaki Katoda akan menjadi Input untuk masing-masing Segmen LED. Kaki Anoda yang terhubung menjadi 1 Pin ini akan diberikan Tegangan Positif (+) dan Signal Kendali (control signal) akan diberikan kepada masing-masing Kaki Katoda Segmen LED.

TIPE COMMON ANODE (Common Anoda)



6.

III. SOAL

1. Buatlah rangkaian arduino yang terhubung dengan 1 buah 7 segment dan 2 buah push button. Buatlah programnya agar ketika push button 1 ditekan, 7 segment menampilkan counter 0-9 (up counter). dan ketika push button 2 ditekan, 7 segment menampilkan counter 9-0 (DOWN COUNTER).
2. Buatlah rangkaian arduino yang terhubung dengan 1 buah 7 segment dan 1 buah switch. Buatlah programnya agar ketika switch berada pada mode 0, 7 segment menampilkan angka genap (0,2,4,6,8). dan ketika switch berada pada mode 1, 7 segment menampilkan angka GANJIL (1,3,5,7,9).
3. Buatlah Program untuk counter modulo Desimal yang dikendalikan 2 saklar, dengan kemungkinan mode :
00 = RESET
01 = UP COUNTER
10 = DOWN COUNTER
11 = STOP
4. Buatlah Program untuk counter modulo Desimal yang dikendalikan 2 saklar, dengan kemungkinan mode :
00 = UP COUNTER MODULO 7
01 = UP COUNTER MODULO 10
10 = UP COUNTER MODULO 5
11 = STOP

IV. HASIL PROGRAM

1. Program :

```
#include <SevSeg.h>
SevSeg sevseg;
const int pushButton[] = {1,2};
void setup() {
  byte sevenSegments = 1;
  byte commonPins[] = {};
  byte sevenPins[] = {11,10,9,8,7,6,5,4};
  bool resistor = true;
  sevseg.begin(COMMON_CATHODE, sevenSegments, commonPins, sevenPins, resistor);
  sevseg.setBrightness(90);

  for(int i=0; i<=1; i++){
    pinMode(pushButton[i], INPUT);
    digitalWrite(pushButton[i], HIGH);
  }
}
```

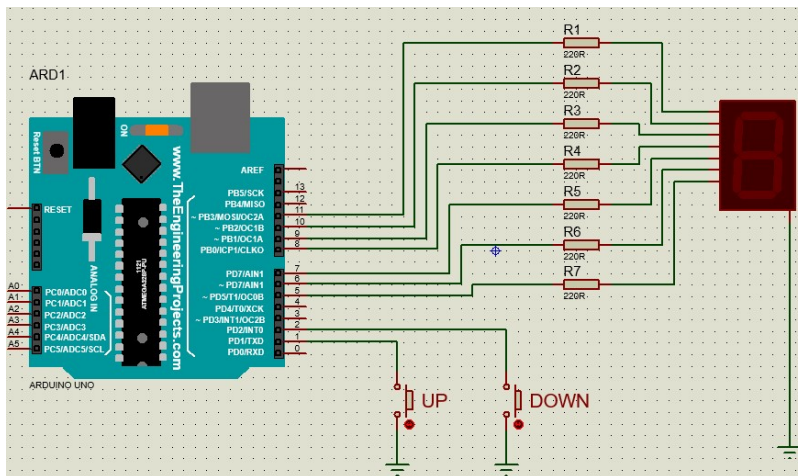
```

void loop() {
  if(digitalRead(pushButton[0]) == LOW){
    delay(100);
    for(int i=0; i<10; i++){
      setSevSeg(i);
    }
  }
  if(digitalRead(pushButton[1]) == LOW){
    delay(100);
    for(int i=9; i>=0; i--){
      setSevSeg(i);
    }
  }
}

void setSevSeg(int num){
  sevseg.setNumber(num);
  sevseg.refreshDisplay();
  delay(1000); //1 detik
}

```

Rangkaian:



2. Program:

```

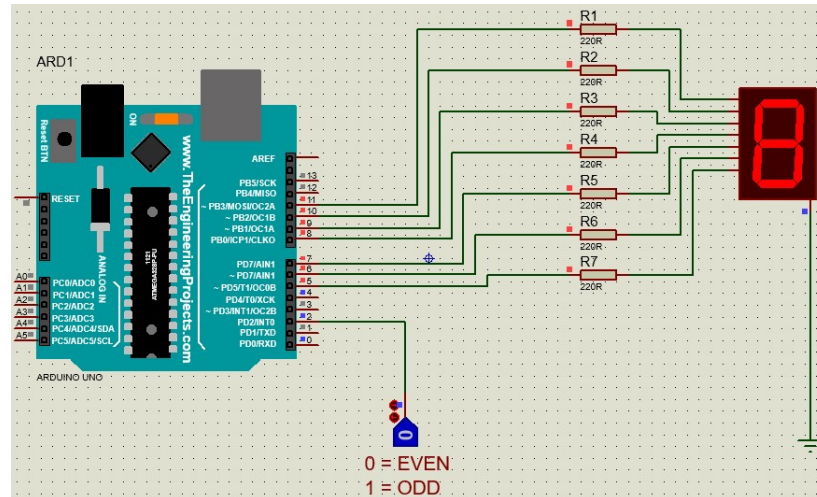
#include <SevSeg.h>
SevSeg sevseg;
const int saklar = 2;
void setup() {
  byte sevenSegments = 1;
  byte commonPins[] = {};
  byte sevenPins[] = {11,10,9,8,7,6,5,4};
  bool resistor = true;
  sevseg.begin(COMMON_CATHODE, sevenSegments, commonPins, sevenPins, resistor);
  sevseg.setBrightness(90);

  pinMode(saklar, INPUT);
}
void loop() {
  while(digitalRead(saklar) == HIGH){
    for(int i=1; i<=9; i+=2){
      setSevSeg(i);
    }
  }
  while(digitalRead(saklar) == LOW){
    for(int i=0; i<=8; i+=2){
      setSevSeg(i);
    }
  }
}

void setSevSeg(int num){
  sevseg.setNumber(num);
  sevseg.refreshDisplay();
  delay(1000); //1 detik
}

```

Rangkaian:

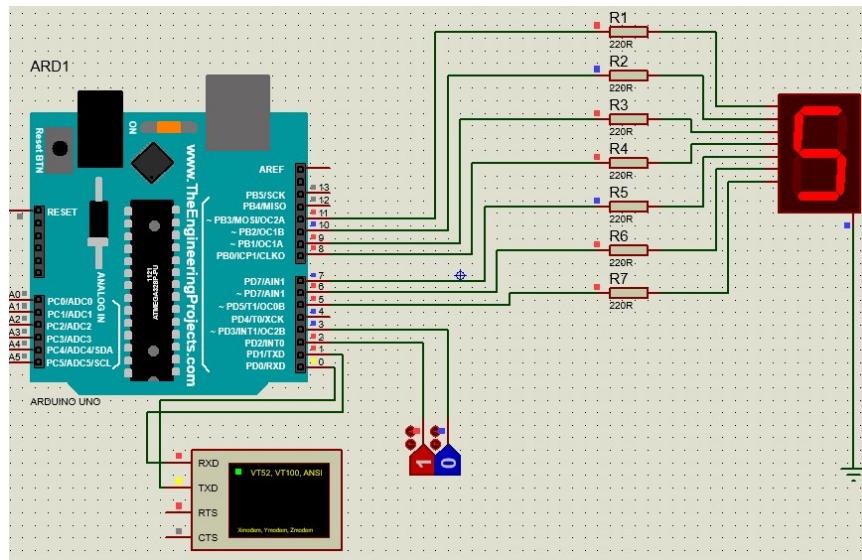


3. Program:

```
#include <SevSeg.h>
SevSeg sevseg;
const int saklar1 = 2;
const int saklar2 = 3;
void setup() {
    byte sevenSegments = 1;
    byte commonPins[] = {};
    byte sevenPins[] = {11,10,9,8,7,6,5,4};
    bool resistor = true;
    sevseg.begin(COMMON_CATHODE, sevenSegments, commonPins, sevenPins, resistor);
    sevseg.setBrightness(90);

    pinMode(saklar1, INPUT);
    pinMode(saklar2, INPUT);
    Serial.begin(9600);
}
void loop() {
    checking(digitalRead(saklar1), digitalRead(saklar2));
}
void checking(bool saklar1, bool saklar2){
    if(saklar1==LOW && saklar2==LOW){
        setSevSeg(0);
    }
    else if(saklar1==LOW && saklar2==HIGH){
        for(int i=0; i<=9; i++){
            setSevSeg(i);
        }
    }
    else if(saklar1==HIGH && saklar2==LOW){
        for(int i=9; i>=0; i--){
            setSevSeg(i);
        }
    }
    else if(saklar1==HIGH && saklar2==HIGH){
        Serial.print("Stop");
        delay(100); //Untuk Stop
    }
}
void setSevSeg(int num){
    sevseg.setNumber(num);
    sevseg.refreshDisplay();
    delay(1000); //1 detik
}
```


Rangkaian:

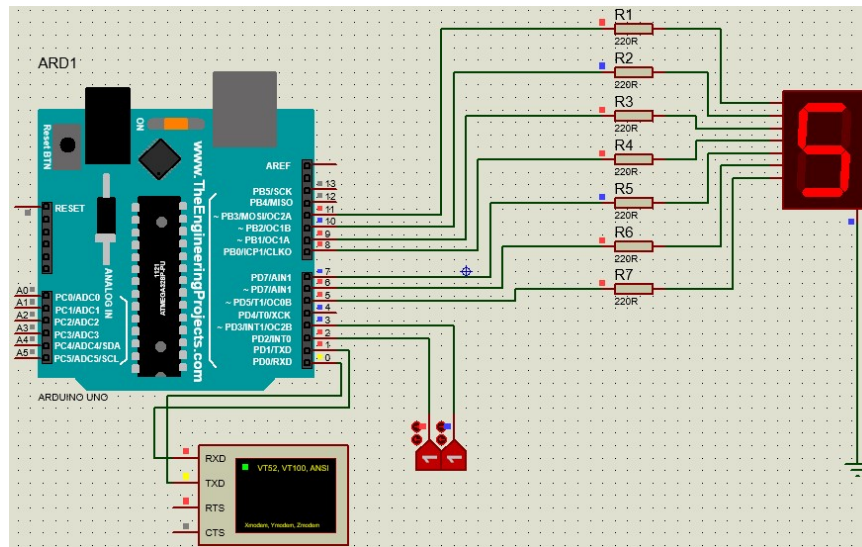


4. Program:

```
#include <SevSeg.h>
SevSeg sevseg;
const int saklar1 = 2;
const int saklar2 = 3;
void setup() {
    byte sevenSegments = 1;
    byte commonPins[] = {};
    byte sevenPins[] = {11,10,9,8,7,6,5,4};
    bool resistor = true;
    sevseg.begin(COMMON_CATHODE, sevenSegments, commonPins, sevenPins, resistor);
    sevseg.setBrightness(90);

    pinMode(saklar1, INPUT);
    pinMode(saklar2, INPUT);
    Serial.begin(9600);
}
void loop() {
    checking(digitalRead(saklar1), digitalRead(saklar2));
}
void checking(bool saklar1, bool saklar2){
    if(saklar1==LOW && saklar2==LOW){
        for(int i=0; i<7; i++){
            setSevSeg(i);
        }
    }
    else if(saklar1==LOW && saklar2==HIGH){
        for(int i=0; i<10; i++){
            setSevSeg(i);
        }
    }
    else if(saklar1==HIGH && saklar2==LOW){
        for(int i=0; i<5; i++){
            setSevSeg(i);
        }
    }
    else if(saklar1==HIGH && saklar2==HIGH){
        delay(100); //Untuk Stop
    }
}
void setSevSeg(int num){
    sevseg.setNumber(num);
    sevseg.refreshDisplay();
    delay(1000); //1 detik
}
```


Rangkaian:



V. ANALISIS

1. Pada program ini, hal pertama yang perlu dilakukan adalah melakukan impor library 7-Segment dengan sintaks `#include <SevSeg.h>` diawal program, kemudian kita implementasikan library ini dengan penulisan yang lebih mudah yaitu `sevseg`. Kemudian diinisialisasi dua buah variabel **pushButton** bertipe integer dengan bentuk array untuk pin 1 dan pin 2 pada arduino, yang dimana kedua push button ini akan diatur sebagai masukan atau INPUT.

Pada fungsi **void setup()**, terdapat variabel **sevenSegments** bertipe byte yang bernilai 1 sebagai variabel yang menyimpan ada berapa buah 7-segment yang dipakai dalam program. Kemudian terdapat variabel **commonPins** bertipe byte sebagai indikator jumlah common yang digunakan dalam program, dalam hal ini cukup dikosongkan karena hanya terdapat 1 digit 7-segment. Variabel **sevenPins** bertipe byte akan diisi oleh nilai-nilai pin yang terhubung berturut-turut ke pin A sampai dengan H pada 7-segment yang cara penginisialisasiannya menggunakan array. Variabel **resistor** bertipe bool diisi dengan true apabila terdapat resistor pada setiap kaki pin yang terhubung ke 7-segment. Selanjutnya akan dipanggil sebuah fungsi **sevseg.begin()** yang di dalamnya terdapat 4 parameter berupa 4 variabel yang sudah diinisialisasi di aal fungsi **void setup()** tadi. Fungsi **sevseg.setBrightness** digunakan untuk mengatur tingkat kecerahan dari tiap segmen di 7-segment yang nilainya berupa integer diantara 0-100.

Sebelum menuju pada fungsi **void loop()**, dibuat dahulu sebuah fungsi tambahan yang tak mengembalikan apa-apa dengan nama **setSevSeg()** dengan satu buah parameter berupa **num** bertipe integer. Fungsi ini berfungsi untuk menampilkan digit 7-segment yang telah dilengkapi dengan fungsi **sevseg.refreshDisplay()** dan delay selama 1 detik. Pada fungsi **void loop()**, akan dilakukan dua pengondisian, yang pertama adalah apabila push button pertama ditekan sehingga akan mengakibatkan level LOW, maka akan dilakukan perulangan for dengan nilai awal 0 hingga 9, yang mana index for ini akan dijadikan parameter untuk fungsi **setSevSeg()** yang nantinya akan ditampilkan pada 7-

segment. Kemudian yang kedua adalah pengondisian apabila push button kedua ditekan, maka akan dilakukan perulangan for dari 9 hingga 0 yang nantinya nilai index for tersebut digunakan pada fungsi **setSevSeg()**.

2. Pada program nomor dua ini, langkah dalam melakukan impor library dan pengaturan 7-segment sama persis seperti nomor sebelumnya. Selain itu, terdapat fungsi tambahan **setSevSeg()** yang juga sama seperti program sebelumnya. Setelah itu, dilakukan inisialisasi variabel **saklar** bertipe integer dengan value 2 (pin pada arduino), yang kemudian variabel **saklar** ini akan diatur sebagai INPUT pada fungsi **void setup()**.

Pada fungsi **void loop()**, akan dilakukan dua perulangan while, yang pertama adalah kondisi dimana selama saklar dalam kondisi HIGH, maka akan dilakukan perulangan for didalamnya dengan nilai awal 1 sampai 9 dengan step kenaikan sebesar 2 (bilangan ganjil), kemudian index dari for ini akan dijadikan parameter pada fungsi **setSevSeg()**. Perulangan while yang kedua adalah ketika dan selama saklar dalam kondisi LOW, maka akan dilakukan perulangan for didalamnya dengan nilai awal 0 hingga 8 berikut step kenaikan sebesar 2 juga (bilangan genap).

3. Pada program nomor tiga ini, langkah dalam melakukan impor library dan pengaturan 7-segment sama persis seperti nomor satu. Selain itu, terdapat fungsi tambahan **setSevSeg()** yang juga sama seperti program nomor 1. Setelah itu, dilakukan inisialisasi variabel **saklar1** bertipe integer dengan value 2 (pin pada arduino) dan **saklar2** dengan value 3, yang kemudian variabel **saklar1** dan **saklar2** ini akan diatur sebagai INPUT pada fungsi **void setup()**.

Pada fungsi **void loop()**, hanya akan dipanggil satu buah fungsi tambahan bernama **checking()** yang memiliki dua parameter yaitu kondisi **saklar1** dan **saklar2**. Pada fungsi **void checking()** ini berisi empat buah pengondisian, yang pertama apabila **saklar1** dan **saklar2** bernilai LOW, maka akan dipanggil fungsi **setSevSeg()** dengan parameter nilai 0. Kedua, apabila **saklar1** LOW dan **saklar2** HIGH, maka akan dilakukan perulangan for dengan nilai awal 0 hingga 9 dengan step kenaikan 1, kemudian dipanggil fungsi **setSevSeg()** dengan parameter berupa index perulangannya. Ketiga, adalah kondisi berkebalikan dari pengondisian kedua, maka dilakukan perulangan for yang juga berkebalikan dari perulangan pada pengondisian kedua. Keempat, adalah kondisi dimana **saklar1** dan **saklar2** sama-sama bernilai HIGH, maka hanya akan dilakukan delay selama 0.1 detik, yang berfungsi sebagai stopper.

4. Pada program nomor tiga ini, langkah dalam melakukan impor library dan pengaturan 7-segment sama persis seperti nomor satu. Selain itu, terdapat fungsi tambahan **setSevSeg()** yang juga sama seperti program nomor 1. Setelah itu, dilakukan inisialisasi variabel **saklar1** bertipe integer dengan value 2 (pin pada arduino) dan **saklar2** dengan value 3, yang kemudian variabel **saklar1** dan **saklar2** ini akan diatur sebagai INPUT pada fungsi **void setup()**.

Pada fungsi **void loop()**, hanya akan dipanggil satu buah fungsi tambahan bernama **checking()** yang memiliki dua parameter yaitu kondisi **saklar1** dan **saklar2**. Pada

fungsi **void checking()** ini berisi empat buah pengondisian, yang pertama apabila **saklar1** dan **saklar2** bernilai LOW, maka akan dilakukan perulangan for dengan nilai awal 0 hingga 6 dengan step kenaikan 1, kemudian dipanggil fungsi **setSevSeg()** dengan nilai parameter berupa index perulangan tadi. Kedua, apabila **saklar1** LOW dan **saklar2** HIGH, maka akan dilakukan perulangan for dengan nilai awal 0 hingga 9 dengan step kenaikan 1, kemudian dipanggil fungsi **setSevSeg()** dengan parameter berupa index perulangannya. Ketiga, apabila **saklar1** HIGH dan **saklar2** LOW, maka akan dilakukan perulangan for dengan nilai awal 0 hingga 4 dengan step kenaikan 1, kemudian dipanggil fungsi **setSevSeg()** dengan parameter berupa index perulangannya. Keempat, adalah kondisi dimana **saklar1** dan **saklar2** sama-sama bernilai HIGH, maka hanya akan dilakukan delay selama 0.1 detik, yang berfungsi sebagai stopper.

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. Selain itu, mikrokontroler arduino dapat digunakan untuk mengoperasikan 7-segment baik common anoda maupun katoda sesuai dengan kebutuhan, serta untuk dapat mengoperasikan 7-segment, dapat digunakan library **SevSeg** agar penggunaannya menjadi lebih mudah.

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. 2016. Teknik Elektronika. *Pengertian Seven Segment Display*. Diakses 24 Januari 2021. <https://teknikelektronika.com/pengertian-seven-segment-display-layar-tujuh-segmen/>
- Kho, Dickson. 2019. *Jenis-Jenis Saklar (Switch) dalam Rangkaian Elektronika*. Teknik Elektronika. Diakses pada 24 Januari 2021. <https://teknikelektronika.com/jenis-jenis-saklar-switch-dalam-rangkaian-elektronika/>
- Sinau. 2016. *Mengenal Arduino Software (IDE)*. Sinau Arduino. Diakses apda 11 Desember 2020. <https://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>