

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring kemajuan teknologi, mikrokontroler seperti Arduino telah menjadi solusi untuk berbagai aplikasi, termasuk bidang musik. Arduino adalah *platform* elektronik yang mudah digunakan yang dapat diprogram untuk mengontrol berbagai komponen, memberikan *fleksibilitas* dalam mengembangkan perangkat seperti Metronom digital. Metronom berbasis Arduino tidak hanya memungkinkan musisi mengatur tempo dengan lebih tepat, namun juga meningkatkan visualisasi dengan LED yang berkedip sesuai irama. Fitur ini diharapkan dapat membuat latihan menjadi lebih efektif, khususnya bagi musisi yang masih dalam tahap belajar.

Tempo yang stabil sangat penting dalam permainan musik agar setiap nada terdengar harmonis dan presisi. Musisi, baik pemula maupun profesional, membutuhkan alat untuk menjaga tempo saat berlatih. Alat yang umum digunakan adalah Metronom, yang berfungsi sebagai pemandu ritme. Namun, Metronom tradisional yang ada di pasaran sering kali memiliki keterbatasan yaitu pengaturan tempo yang terbatas, kemampuan visualisasi yang kurang, dan harga yang relatif mahal.

Metronom ini lebih murah untuk dikembangkan dibandingkan Metronom digital yang tersedia secara komersial dengan menggunakan komponen sederhana seperti *buzzer*, LED, potensiometer, dan tombol. Selain itu, Metronom berbasis

Arduino dapat dikembangkan lebih lanjut dengan menyertakan fitur tambahan seperti Pemilihan suara, pengaturan volume, dan integrasi dengan perangkat lain.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan Metronom digital berbasis Arduino dengan fungsi pengaturan tempo dan visualisasi. Kami berharap dengan menggunakan alat ini, musisi dapat lebih mudah berlatih, menjaga tempo, dan meningkatkan kemampuan bermusiknya.

Melalui penelitian ini, diharapkan dapat memberikan kontribusi dalam bidang musik dengan mengembangkan alat Metronom digital. Maka dari itu peneliti mencoba untuk mengajukan sebuah judul tugas akhir yaitu **“RANCANG BANGUNG METRONOM DIGITAL BERBASIS MIKROKONTROLER DENGAN FITUR PENGATURAN TEMPO DAN VISUALISASI UNTUK MENINGKATKAN EFEKTIVITAS LATIHAN MUSIK”**.

1.2 Perumusan Masalah

Perumusan masalah adalah bagian yang mengidentifikasi inti permasalahan yang akan diteliti dan biasanya disusun dalam bentuk pertanyaan untuk memberikan arah penelitian. Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian yang dilakukan ini adalah sebagai berikut:

1. Bagaimana Metronom digital berbasis mikrokontroler dapat membantu dalam meningkatkan efektivitas latihan musik?
2. Bagaimana *switch on/off*, *Slide* Potensiometer, dan *button* dapat bekerja sebagai *input* pada sistem?

3. Bagaimana cara kerja dari *DFPlayer* dapat mengeluarkan *output* berupa suara, serta modul PAM sebagai pengatur dari volume alat, dan peran *speaker* dalam menghasilkan suara tersebut?
4. Bagaimana OLED dapat menampilkan berapa nilai tempo dalam bentuk angka?
5. Bagaimana LED dapat digunakan untuk memvisualisasikan ketukan suara dari Metronom, sehingga mendapatkan umpan balik visual sesuai nilai tempo?

1.3 Batasan Masalah

Batasan masalah adalah ruang lingkup masalah atau membatasi ruang lingkup masalah yang terlalu luas/lebar sehingga penelitian lebih bisa fokus untuk dilakukan. Batasan masalah agar penelitian tidak menyimpang dari tujuan penelitian, maka ini difokuskan pada masalah yang berkaitan dengan hal-hal sebagai berikut:

1. Menggunakan Arduino mega 2560 sebagai pengontrol sistem.
2. Menggunakan *slide* potensiometer untuk mengatur tingkat tempo.
3. Menggunakan *button* untuk mengubah suara.
4. Menggunakan modul PAM untuk mengatur volume suara.
5. Menggunakan *DFPlayer* dan *speaker* sebagai *output* berupa suara pada alat.
6. Menggunakan OLED sebagai menampilkan informasi nilai tempo.
7. Menggunakan LED sebagai visualisasi dari nilai tempo

1.4 Hipotesa

Hipotesis adalah jawaban sementara yang hendak di uji kebenarannya melalui penelitian. Berdasarkan pada perumusan masalah di atas, penulis dapat mengambil beberapa hipotesis, yaitu:

1. Alat Metronom digital berbasis mikrokontroler dapat membantu dalam meningkatkan efektivitas latihan musik.
2. Penggunaan *switch on/off*, *slide* Potensiometer, dan *button* dapat bekerja sebagai *input* pada sistem.
3. Penggunaan *DFPlayer* dapat mengeluarkan *output* berupa suara, serta modul PAM sebagai pengatur dari volume alat, dan peran *speaker* dalam menghasilkan suara tersebut.
4. Penggunaan OLED dapat menampilkan berapa nilai tempo dalam bentuk angka.
5. Penggunaan LED dapat digunakan untuk memvisualisasikan ketukan suara dari Metronom, sehingga mendapatkan umpan balik visual sesuai nilai tempo.

1.5 Tujuan Penelitian

Penelitian ini dilakukan untuk mendapatkan tujuan penelitian tentang hal-hal yang dibahas sebagai berikut:

1. Memahami konsep kerja dari Arduino Mega 2560 sehingga dapat diterapkan pada perancangan alat Metronom digital.
2. Membangun mekanisme pengaturan tempo yang akurat dan intuitif menggunakan *slide* potensiometer, sehingga memudahkan pengguna untuk menyesuaikan tempo sesuai dengan kebutuhan latihan.

3. Mengimplementasikan fungsi tombol (*button* 1 dan *button* 2) untuk memungkinkan pengguna mengubah jenis suara metronom.
4. Mengintegrasikan Modul PAM sebagai pengatur tingkat rendahnya volume suara alat sesuai yang diinginkan pengguna.
5. Mengintegrasikan modul *DFPlayer* dan *speaker* untuk menghasilkan suara metronom yang dapat diatur jenisnya, sesuai dengan tempo dan pengaturan suara yang dipilih pengguna.
6. Mengembangkan tampilan OLED untuk menampilkan informasi *real-time* terkait tempo sehingga memudahkan pengguna memonitor pengaturan metronom.
7. Mengintegrasikan LED sebagai visualisasi ketukan metronom, sehingga memberikan umpan balik visual yang sinkron dengan tempo yang diatur.

1.6 Manfaat Penelitian

Manfaat penelitian adalah nilai atau kontribusi yang dihasilkan dari penelitian, baik secara teoritis maupun praktis. Adapun manfaat dari penelitian ini adalah sebagai berikut:

A. Bagi Penulis

1. Penelitian ini memberikan penulis kesempatan untuk mengembangkan keterampilan teknis dalam bidang mikrokontroler, pemrograman, dan elektronika.
2. Melalui penelitian ini, penulis akan meningkatkan kemampuan dalam mengidentifikasi masalah, merumuskan solusi, dan menerapkan metode ilmiah untuk mengatasi tantangan teknis.

3. Sebagai syarat untuk mendapatkan gelar di jenjang Pendidikan Strata 1 (S1).

B. Bagi Program Studi

1. Menambah jumlah alat berbasis arduino yang dimiliki oleh laboratorium sistem komputer.
2. Dapat menambah referensi akademis dalam berkarya dan menjadikan sebagai motivasi untuk dikembangkan ke alat yang kompleks penggunaannya.
3. Dapat dijadikan sebagai pedoman bagi mahasiswa selanjutnya untuk matakuliah yang berhubungan dengan hasil akhir penelitian.

C. Bagi Musisi

1. Metronom digital membantu musisi menjaga tempo secara konsisten saat berlatih, yang penting untuk penguasaan irama, terutama pada karya dengan tempo yang kompleks.
2. Musisi dapat mengatur tempo sesuai kebutuhan mereka, dengan opsi untuk mengubah tempo secara halus atau cepat menggunakan potensiometer. Metronom digital ini juga memungkinkan pengaturan irama dan subdivisi ketukan, mendukung beragam genre musik.

BAB IV

ANALISA DAN HASIL

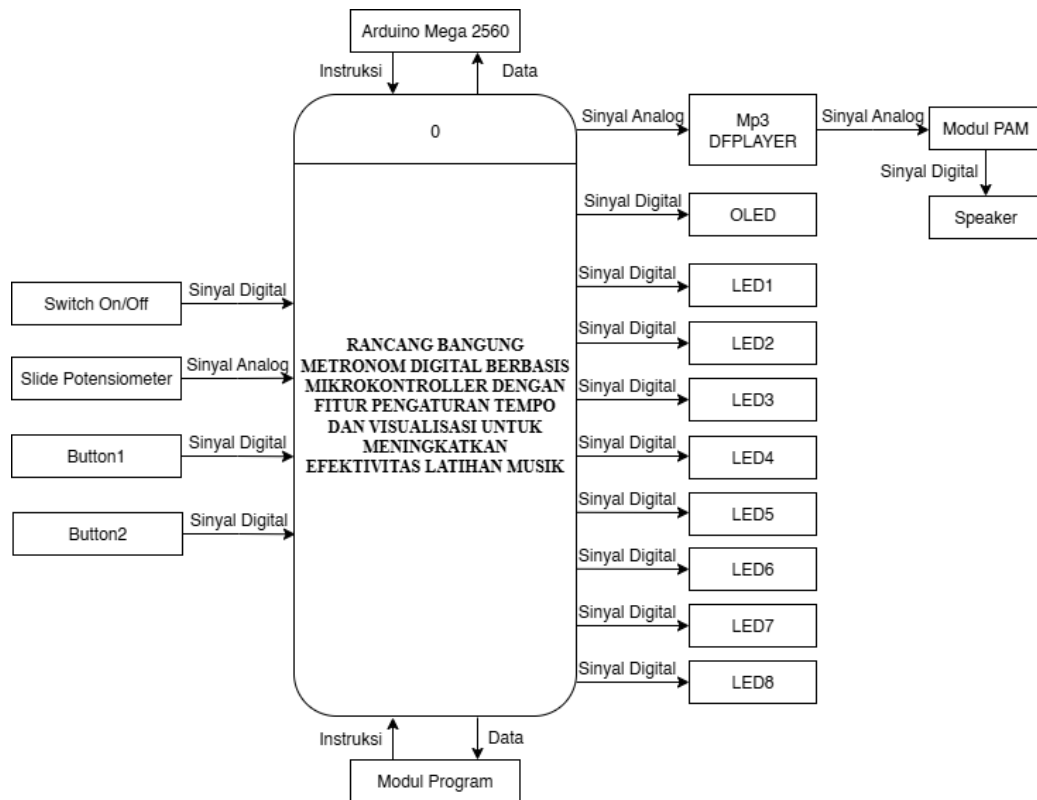
4.1 Desain Sistem Secara Umum

Desain dari sistem yang dibuat merupakan gambaran dari sistem secara keseluruhan. Dengan adanya desain ini maka prinsip kerja dari sistem serta komponen-komponen dari sistem yang digunakan akan dapat dilihat dengan jelas.

4.1.1 Context Diagram

Sub bab ini merupakan penjabaran setiap *external entity* secara keseluruhan yang digambarkan melalui *Context Diagram*. *Context Diagram* merupakan pendefenisian terhadap sistem yang akan dirancang yang bersifat menyeluruh. *Context Diagram* ini digunakan untuk memudahkan dalam proses penganalisaan sistem yang dirancang secara keseluruhan.

Context Diagram berfungsi sebagai media, yang terdiri dari suatu proses dan beberapa buah *External Entity*. *Context Diagram* yang dimaksud dapat dilihat pada Gambar 4.1.



Gambar 4.1 Context Diagram

Berdasarkan *Context Diagram* di atas, masing-masing *entity* yang digunakan akan dijelaskan sebagai berikut:

1. *Arduino Mega 2560*

Arduino Mega 2560 berfungsi sebagai pusat pengolahan seluruh data dan intruksi yang keluar dan masuk.

2. *Slide Potensiometer*

Slide potensiometer berfungsi sebagai mengatur kecepatan atau tempo.

3. *Button*

Button berfungsi sebagai memilih *output* suara.

4. *MP3 DFPlayer*

MP3 DFPlayer berfungsi sebagai pemutar suara dari *SD card*.

5. Modul PAM

Modul PAM berfungsi sebagai pengatur volume suara.

6. *Speaker*

Speaker berfungsi sebagai output berupa audio.

7. OLED

OLED berfungsi sebagai menampilkan informasi.

8. LED

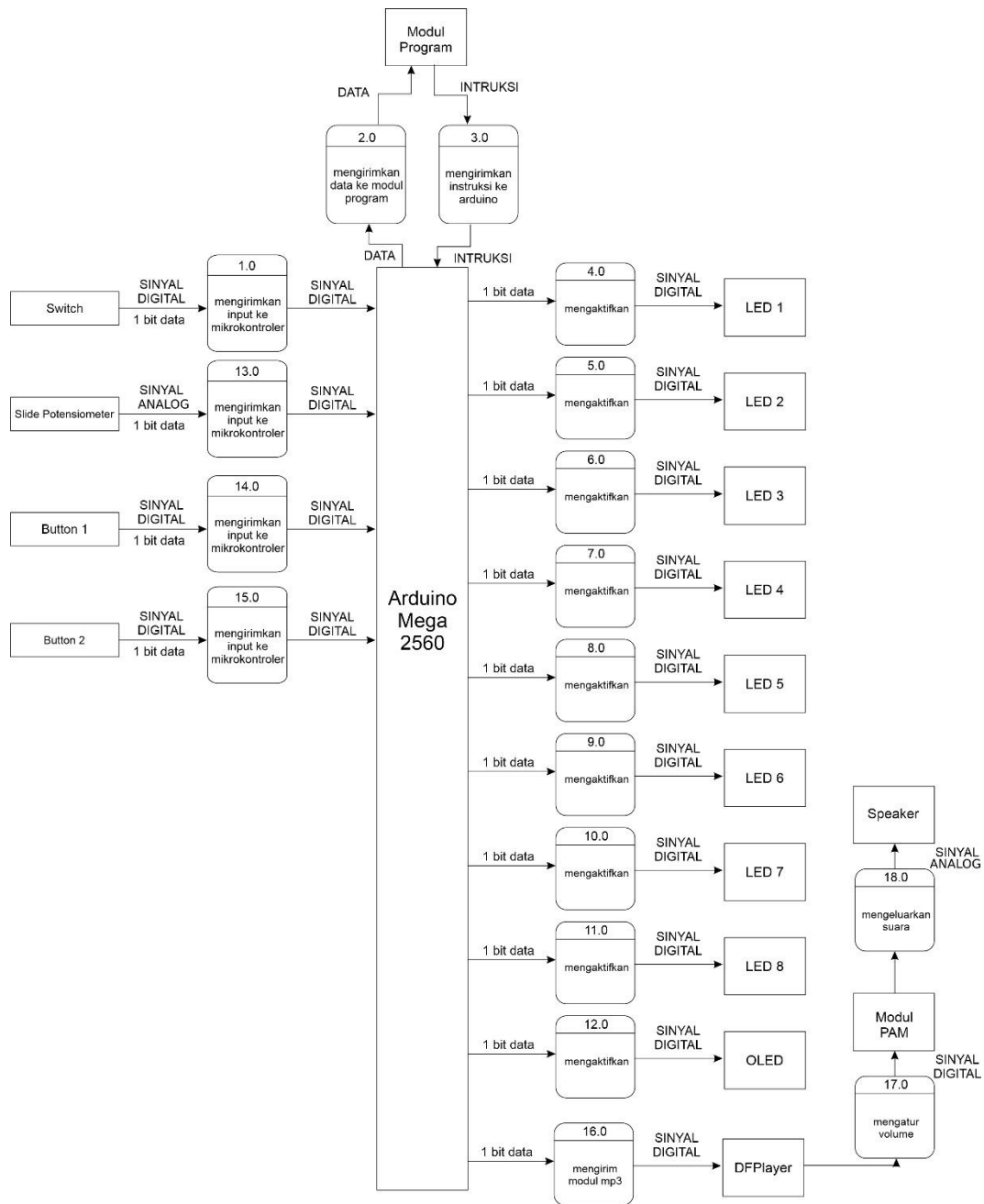
LED berfungsi sebagai menampilkan ketukan secara visual.

9. *Switch On/Off*

Switch on/off berfungsi untuk mengaktifkan perangkat.

4.1.2 Data Flow Diagram (DFD)

Data Flow Diagram adalah gambaran yang lebih rinci dari alat yang dirancang. *Data Flow Diagram* diuraikan berdasarkan *Context Diagram* yang telah dijabarkan sebelumnya. Gambar *Data Flow Diagram* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Data Flow Diagram

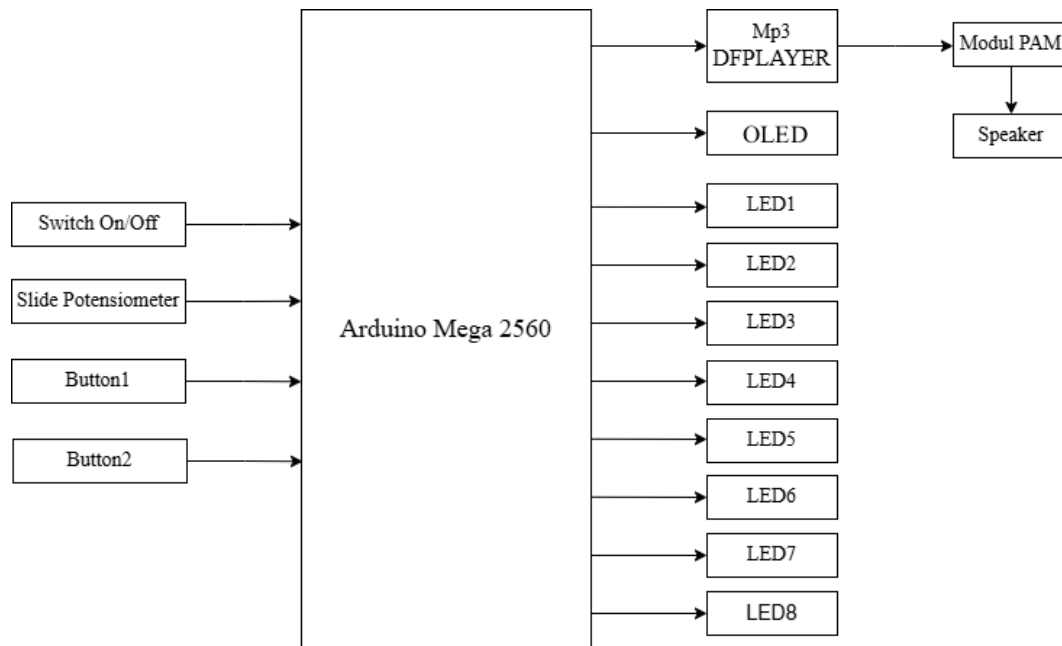
Berdasarkan Data Flow Diagram di atas dapat dijelaskan bahwa sistem ini memiliki beberapa urutan proses instruksi sebagai berikut:

1. Hidupkan *switch on/off* untuk mengirimkan input ke mikrokontroler (1.0)

2. Arduino Mega 2560 akan mengirimkan data ke modul program untuk memproses data (2.0), hasil dari modul program modul program dikirim ke Arduino Mega 2560 (3.0).
3. Hasil proses diteruskan oleh Arduino Mega 2560 untuk mengaktifkan LED 1 (4.0) sampai LED 8 (11.0) secara bergantian sesuai dengan tempo yang telah diatur untuk menunjukkan ketukan metronome.
4. Kemudian OLED akan menerima perintah dari Arduino Mega 2560 (12.0) untuk menampilkan informasi.
5. *Slide* potensiometer sebagai pengatur tempo ketukan, nilai *slide* potensiometer dikirim ke Ardunino Mega 2560 sebagai *input* (13.0), lalu diproses untuk mengatur kecepatan LED dan suara.
6. Pengguna dapat menekan *button* 2 untuk mengubah suara metronom (15.0), dan menekan *button* 1 untuk kembali ke suara sebelumnya (14.0).
7. Arduino mengirimkan perintah ke *DFPlayer* untuk memilih jenis suara metronom yang akan dimainkan (16.0).
8. *DFPlayer* kemudian mengirimkan sinyal digital ke modul PAM untuk mengatur volume suara metronom (17.0).
9. Modul PAM akan mengaktifkan *speaker* untuk mengeluarkan suara metronom sesuai jenis suara dan tempo yang dipilih (18.0).

4.1.3 Blok Diagram

Acuan dari Data *Flow* Diagram digunakan untuk mengetahui komponen-komponen dalam sistem. Informasi tersebut dapat dilihat secara lebih jelas pada Blok Diagram yang ditampilkan pada Gambar 4.3.



Gambar 4.3 Blok Diagram

Dari gambar 4.3 dapat dilihat bahwa sistem terdiri atas input, proses, dan output. Mikrokontroler Arduino Mega 2560 berperan sebagai pusat pemrosesan dan pengontrolan. *Slide Potensiometer*, *button 1*, *button 2*, dan *switch on/off* sebagai *input*. Sedangkan *output* dari sistem adalah OLED, Speaker, LED 1 sampai LED 8.

4.2 Prinsip Kerja Alat

Prinsip kerja sistem adalah cara kerja sistem yang digunakan oleh sebuah sistem dalam mencapai tujuannya. Berdasarkan hasil pemnuatan sistem yang telah penulis lakukan, prinsip kerja dari alat metronome digital yang telah dibuat ini adalah sebagai berikut:

1. Pengguna menghidupkan alat dengan menekan tombol *switch on/off*.
2. Setelah menyala delapan LED akan menyala secara berurutan dari ujung ke ujung dan *speaker* akan mengeluarkan suara ketika LED sudah bedara pada

setiap ujung dan OLED akan menampilkan tempo sesuai *Slide* Potensiometer berada.

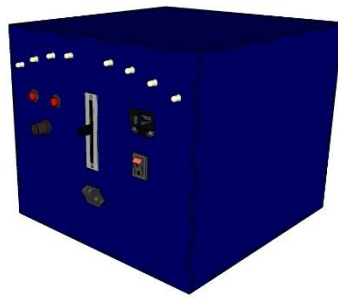
3. Kalau dirasa temponya terlalu rendah atau tinggi pengguna bisa mengatur tempo yang diinginkan dengan menggeser *Slide* Potensiometer.
4. Jika bosan mendengar suara metronom yang begitu saja pengguna bisa menekan *button 2* jika ingin kembali ke suara sebelumnya pengguna bisa menekan *button 1*.
5. Kalau suara volume metronom terlalu besar atau kecil pengguna bisa mengatur suara dengan memutar potensiometer yang ada pada Modul PAM.

4.3 Rancangan Fisik Alat

Perancangan alat ini merupakan tahap awal dari pemasangan dan menganalisa permasalahan yang dihadapi berdasarkan literatur yang menunjang perancangan alat, rancangan fisik dari alat digambarkan menggunakan software *google sketchup*. Rancangan fisik alat yang dibuat dapat dilihat pada Gambar 4.4 dan 4.4.



(a) Tampak Depan



(b) Tampak Serong

Gambar 4.4 Rancangan Fisik Alat

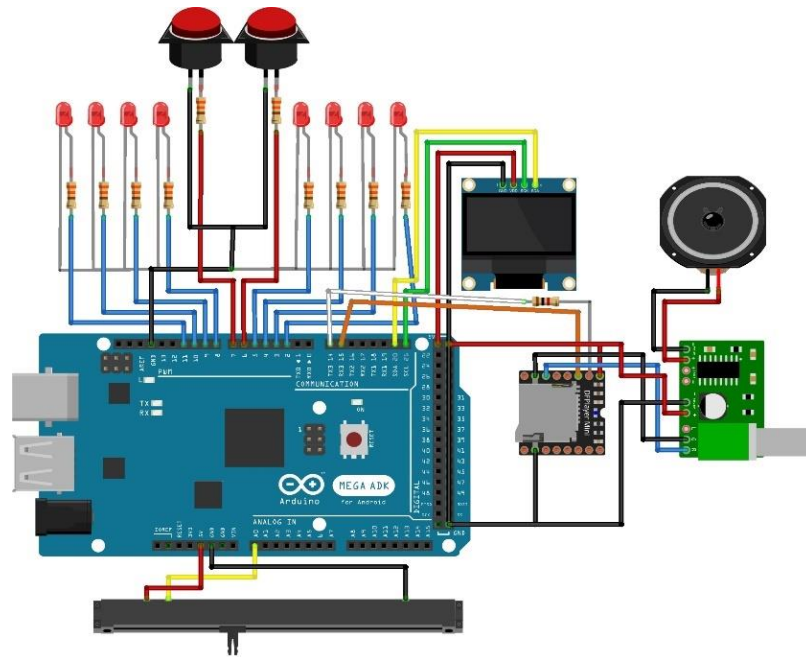
Metronom digital ini memiliki panel depan dengan deretan 8 buah LED sebagai indikator ketukan, dua *button* berfungsi memilih *output* suara, *slide* potensiometer untuk mengatur tempo BPM, modul PAM untuk mengatur volume, layar OLED untuk menampilkan informasi, *switch on/off* untuk menyalakan atau mematikan sistem, dan *speaker* sebagai *output* suara.

4.4 Desain Sistem Terperinci

Desain pada sistem adalah gambaran dari sistem secara keseluruhan. Dengan dilihat dengan sangat jelas.

4.4.1 Rangkaian Mikrokontroler Arduino Mega 2560

Pada sistem ini Mikrokontroler Arduino Mega 2560 digunakan sebagai pengontrol atau pengendali dari sistem. Arduino Mega 2560 menggunakan *chip* Atmega 2560 yang memiliki pin I/O digital 54 buah (15 pin PWM), 16 pin analog *input* dan 4 pin UART. *Board* ini sudah sangat lengkap untuk memenuhi kebutuhan sistem yang dibuat pada penelitian kali ini. Rangkaian Arduino Mega 2560 dapat dilihat pada gambar 4.5 berikut.



Gambar 4.5 Rangkaian Keseluruhan Arduino Mega 2560

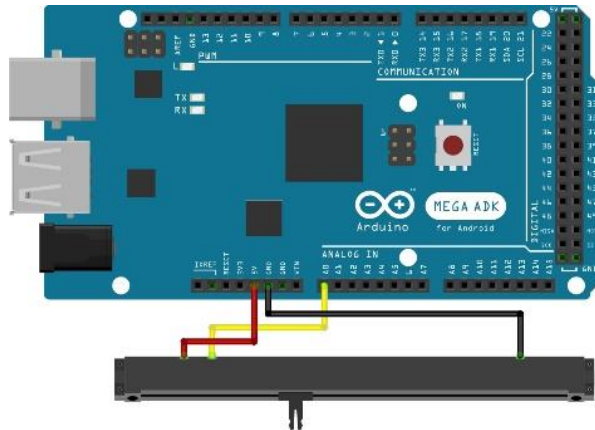
Berdasarkan gambar rangkaian di atas, konfigurasi pin yang digambarkan pada rangkaian Arduino Mega 2560 adalah sebagai berikut:

1. Pin 2, 3, 4, 5, 8, 9, 10, 11 pada Arduino Mega 2560 terhubung pada LED 1 sampai LED 8.
2. Pin 6 dan 7 pada Arduino Mega 2560 terhubung pada *button* 1 dan *button* 2.
3. Pin A0 pada Arduino Mega 2560 terhubung pada *slide* potensiometer.
4. Pin SDA dan SCL pada Arduino Mega 2560 terhubung pada OLED.
5. Pin TX3 dan RX3 pada Arduino Mega 2560 terhubung pada *DFPlayer*.

4.4.2 Rangkaian Slide Potensiometer

Slide potensiometer digunakan untuk mengatur kecepatan atau tempo.

Berikut ini adalah rangkaian komponen *slide* potensiometer.



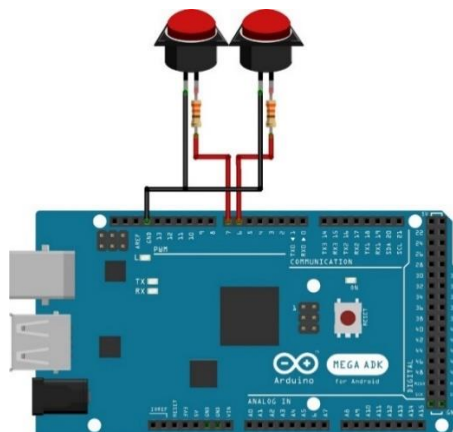
Gambar 4.6 Rangkaian *Slide* Potensiometer

Berdasarkan gambar rangkaian di atas, pin yang digunakan pada Arduino Mega 2560 adalah sebagai berikut:

1. Pin GND pada Arduino Mega 2560 terhubung ke GND *slide* potensiometer.
2. Pin 5V pada Arduino Mega 2560 terhubung ke 5V *slide* potensiometer
3. Pin A0 pada Arduino Mega 2560 terhubung ke *slide* potensiometer

4.4.3 Rangkaian Button

Pada sistem menggunakan 2 buah *button* berfungsi sebagai kontrol pemilihan suara pada sistem metronom. Berikut ini adalah rangkaian komponen 2 buah komponen *button*.



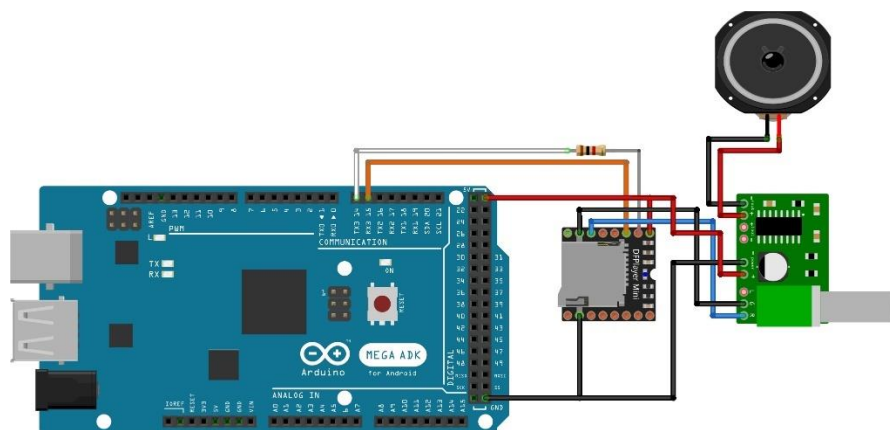
Gambar 4.7 Rangkaian *Button*

Berdasarkan gambar rangkaian di atas, pin yang digunakan pada Arduino Mega 2560 adalah sebagai berikut:

1. Pin GND pada Arduino Mega 2560 terhubung ke GND *button* 1 dan *button* 2.
2. Pin 6 pada Arduino Mega 2560 terhubung ke *button* 1.
3. Pin 7 pada Arduino Mega 2560 terhubung ke *button* 2.

4.4.4 Rangkaian DFPlayer, Modul PAM dan Speaker

DFPlayer berperan sebagai pemutar file audio dalam sistem. Modul PAM berfungsi sebagai pengatur volume suara. *Speaker* berfungsi sebagai komponen *output* suara yang menghasilkan suara mertonom. Berikut ini adalah rangkaian komponen *DFPlayer*, Modul PAM dan *speaker*.



Gambar 4.8 Rangkaian *DFPlayer*, Modul PAM, dan *Speaker*

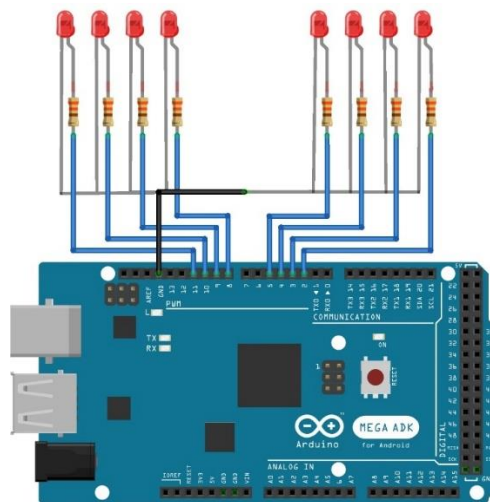
Berdasarkan gambar rangkaian di atas, pin yang digunakan pada Arduino Mega 2560 adalah sebagai berikut:

1. Pin GND pada Arduino Mega 2560 terhubung ke pin GND *DFPlayer*.
2. Pin 5V pada Arduino Mega 2560 terhubung ke pin VCC *DFPlayer*.
3. Pin TX3 pada Arduino Mega 2560 terhubung ke pin TX *DFPlayer*.
4. Pin RX3 pada Arduino Mega 2560 terhubung ke pin RX *DFPlayer*.

5. Pin DAC_R dan DAC_L pada *DFPlayer* terhubung ke pin input L dan R pada Modul PAM.
6. Pin GND pada *DFPlayer* terhubung ke GND *input* Modul PAM.
7. Pin OUT L dan OUT R pada Modul PAM terhubung ke kabel *speaker*.

4.4.5 Rangkaian LED

Pada sistem menggunakan 8 buah komponen LED berfungsi sebagai indikator visual ritme atau ketukan metronom. Berikut ini adalah rangkaian 8 buah komponen LED.



Gambar 4.9 Rangkaian LED

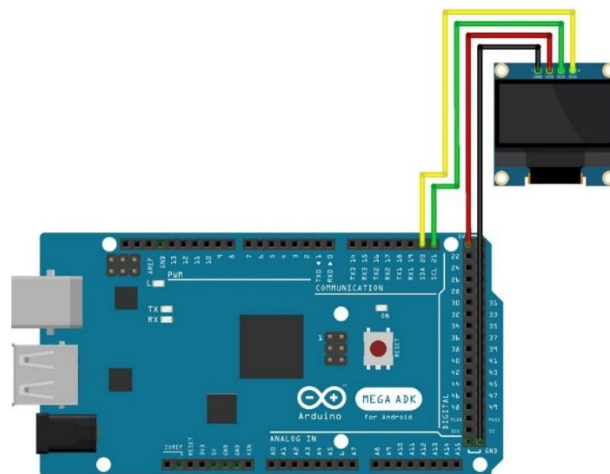
Berdasarkan gambar rangkaian di atas, pin yang digunakan pada Arduino Mega 2560 adalah sebagai berikut:

1. Pin 2 pada Arduino Mega 2560 terhubung ke pin Anode LED 1.
2. Pin 3 pada Arduino Mega 2560 terhubung ke pin Anode LED 2.
3. Pin 4 pada Arduino Mega 2560 terhubung ke pin Anode LED 3.
4. Pin 5 pada Arduino Mega 2560 terhubung ke pin Anode LED 4.
5. Pin 8 pada Arduino Mega 2560 terhubung ke pin Anode LED 5.

6. Pin 9 pada Arduino Mega 2560 terhubung ke pin Anode LED 6.
7. Pin 10 pada Arduino Mega 2560 terhubung ke pin Anode LED 7.
8. Pin 11 pada Arduino Mega 2560 terhubung ke pin Anode LED 8.
9. Pin GND pada Arduino Mega 2560 terhubung ke pin Katode LED 1 sampai LED 8.

4.4.6 Rangkaian OLED

OLED berfungsi untuk menampilkan berbagai data penting agar pengguna dapat melihat status alat secara langsung. Berikut ini adalah rangkaian komponen OLED.



Gambar 4.10 Rangkaian OLED

Berdasarkan gambar rangkaian di atas, pin yang digunakan pada Arduino Mega 2560 adalah sebagai berikut:

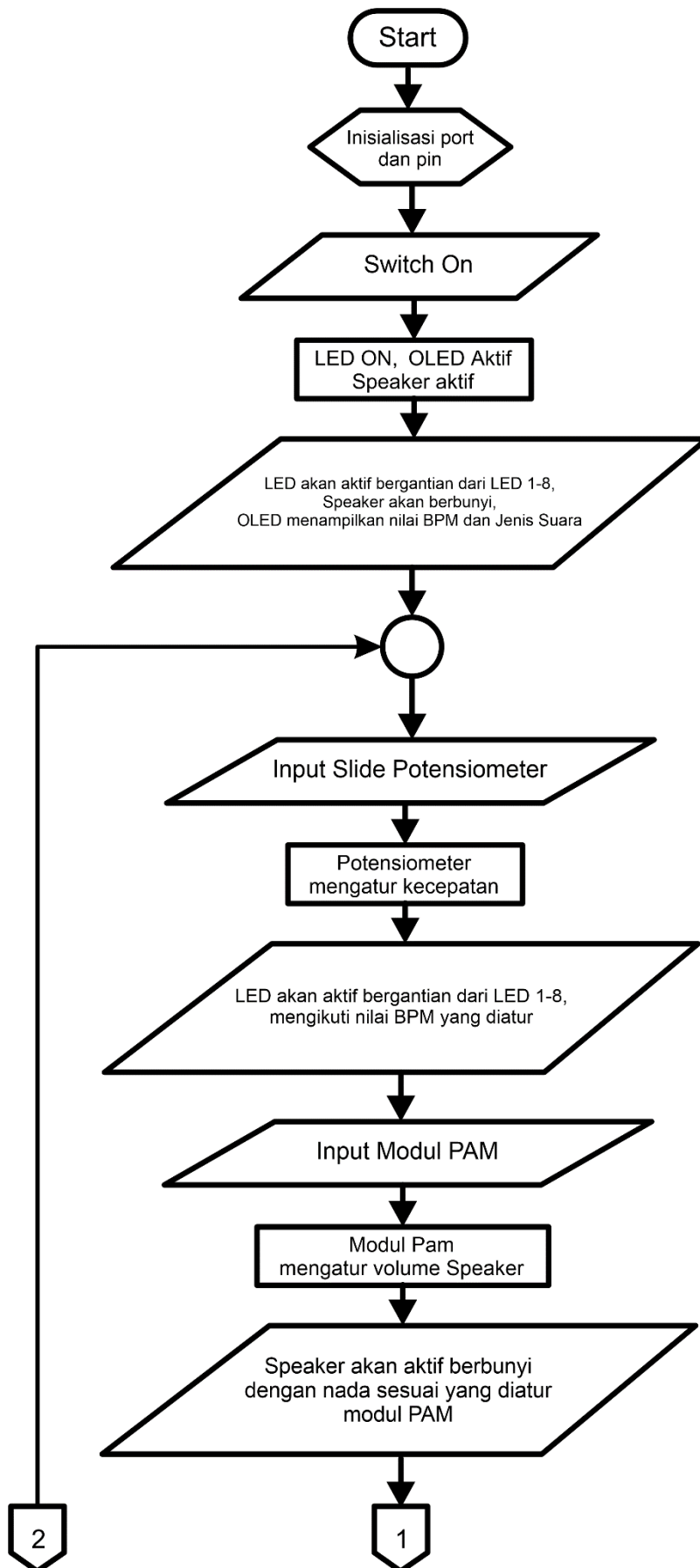
1. Pin GND pada Arduino Mega 2560 terhubung ke pin GND OLED.
2. Pin 5V pada Arduino Mega 2560 terhubung ke pin VCC OLED.
3. Pin 20 pada Arduino Mega 2560 terhubung ke pin SDA OLED.
4. Pin 21 pada Arduino Mega 2560 terhubung ke pin SCL OLED.

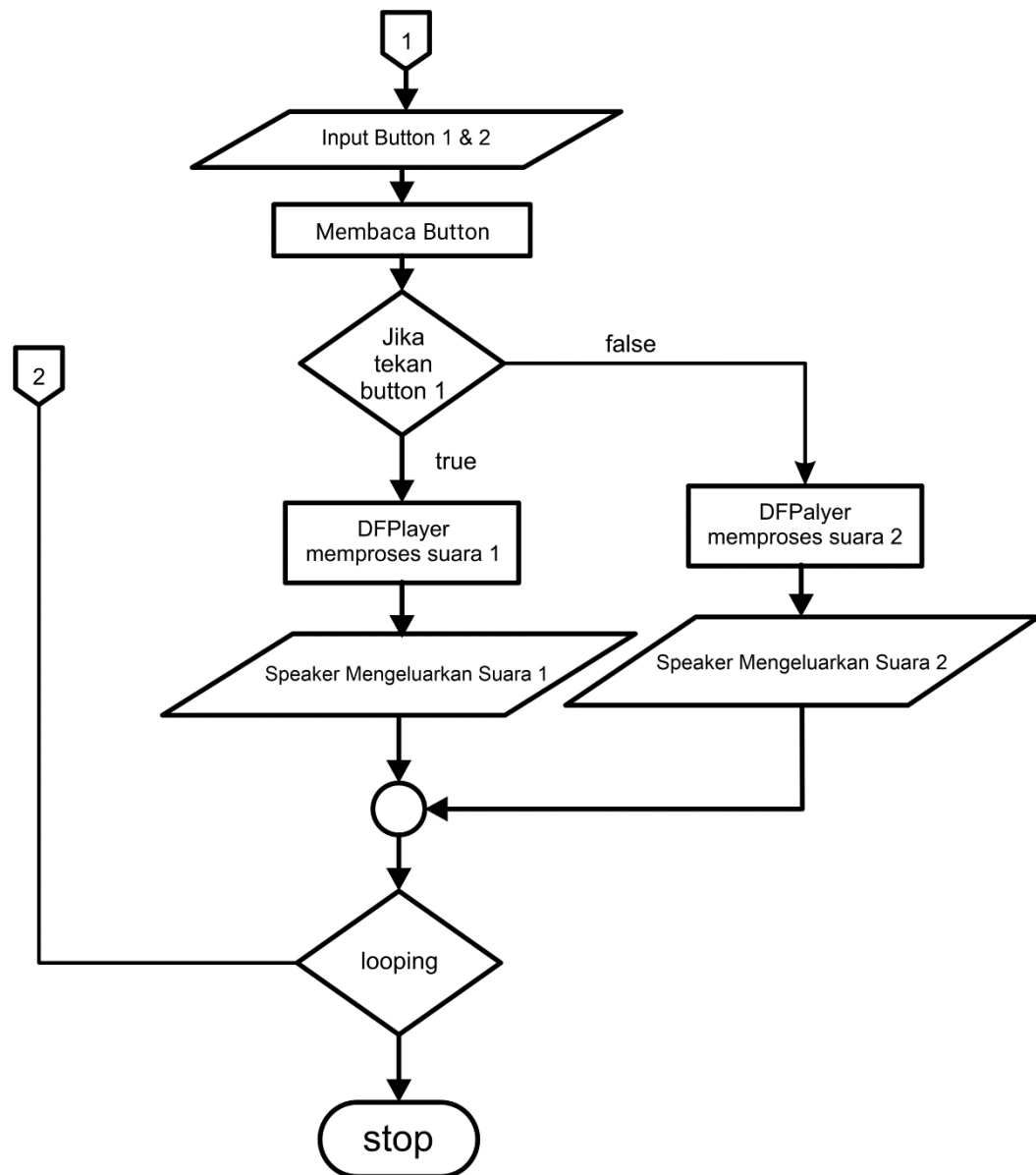
4.5 Rangkaian Modul Program

Pada rancangan modul program berisi tentang penjelasan modul program yang digunakan pada sistem yang telah dirancang. Rancangan modul program terbagi menjadi 2, yakni *flowchart* dan *listing program*.

4.5.1 *Flowchart*

Modul program dirancang agar dapat mudah dimengerti, maka dari itu sebelum membuat listing Program perlu diawali dengan penentu logika pada program. Logikanya digambarkan menggunakan flowchart seperti gambar berikut ini:





Gambar 4.11 Flowchart

4.5.2 Listing Program

Listing program pada sub bab ini akan diuraikan mengenai kode dari modul program yang digunakan untuk membuat alat metronome digital menggunakan software Arduino IDE pada Arduino Mega 2560. Beberapa modul program yang digunakan adalah sebagai berikut:

4.5.2.1 Program OLED

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Konfigurasi ukuran layar OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// Inisialisasi objek display OLED
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  // Inisialisasi komunikasi serial untuk debugging
  Serial.begin(9600);

  // Inisialisasi layar OLED dengan alamat I2C 0x3C
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("OLED tidak ditemukan!"));
    while (true); // Berhenti jika tidak ditemukan
  }

  // Bersihkan layar dan atur tampilan awal
  display.clearDisplay();
  display.setTextSize(2);          // Ukuran teks
  display.setTextColor(SSD1306_WHITE); // Warna teks

  // Menampilkan pesan sambutan
  display.setCursor(15, 10);
  display.println("Welcome!");
  display.setCursor(10, 35);
  display.println("Metronome");
  display.display();

  delay(3000); // Tunda 3 detik
}

void loop() {
  // Menampilkan informasi dinamis (contoh BPM dan suara)
  int bpm = 120;
  int sound = 1;

  display.clearDisplay();
  display.setCursor(0, 10);
  display.print("BPM: ");
  display.println(bpm);
  display.setCursor(0, 35);
```

```

display.print("Sound: ");
display.println(sound);
display.display();

delay(1000); // Tunda update tiap 1 detik
}

```

4.5.2.2 Program LED1 Sampai LED8

```

const int ledPins[8] = {2, 3, 4, 5, 8, 9, 10, 11}; // Pin LED dari kiri ke kanan

int ledIndex = 0;      // Posisi LED aktif saat ini
int ledDirection = 1;  // Arah pergerakan LED (1 = maju, -1 = mundur)
unsigned long previousMillis = 0;
unsigned long interval = 150; // Waktu antar perpindahan LED (ms)

void setup() {
  // Inisialisasi semua pin LED sebagai output
  for (int i = 0; i < 8; i++) {
    pinMode(ledPins[i], OUTPUT);
    digitalWrite(ledPins[i], LOW);
  }
}

void loop() {
  // Mengatur jeda antara tiap perpindahan LED
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    // Matikan semua LED terlebih dahulu
    for (int i = 0; i < 8; i++) {
      digitalWrite(ledPins[i], LOW);
    }

    // Nyalakan LED saat ini
    digitalWrite(ledPins[ledIndex], HIGH);

    // Update posisi LED selanjutnya
    ledIndex += ledDirection;

    // Ubah arah jika mencapai ujung
    if (ledIndex == 7 || ledIndex == 0) {
      ledDirection *= -1;
    }
  }
}

```

4.5.2.3 Program Slide Potensiometer

```
const int potPin = A0; // Pin input potensiometer slide
int bpm = 0;          // Nilai BPM yang dihitung dari potensiometer

void setup() {
  Serial.begin(9600); // Inisialisasi komunikasi serial
}

void loop() {
  int potValue = analogRead(potPin); // Baca nilai dari potensiometer (0–1023)
  bpm = map(potValue, 0, 1023, 40, 218); // Konversi ke rentang BPM 40–240

  Serial.print("Nilai Potensiometer: ");
  Serial.print(potValue);
  Serial.print(" -> BPM: ");
  Serial.println(bpm);

  delay(500); // Update setiap 0.5 detik
}
```

4.5.2.4 Program Button1 dan Button2

```
const int buttonNext = 6; // Tombol untuk suara berikutnya
const int buttonPrev = 7; // Tombol untuk suara sebelumnya

int currentSound = 1; // Suara aktif saat ini
const int maxSound = 3; // Jumlah total suara
unsigned long lastNextButtonTime = 0;
unsigned long lastPrevButtonTime = 0;
const int debounceDelay = 200; // Penundaan untuk debounce tombol

void setup() {
  Serial.begin(9600);
  pinMode(buttonNext, INPUT_PULLUP); // Tombol aktif LOW
  pinMode(buttonPrev, INPUT_PULLUP);
}

void loop() {
  // Cek tombol Next
  if (digitalRead(buttonNext) == LOW && millis() - lastNextButtonTime >
  debounceDelay) {
    currentSound++;
    if (currentSound > maxSound) currentSound = 1;
    lastNextButtonTime = millis();
  }
```

```

    Serial.print("Tombol Next ditekan. Sound: ");
    Serial.println(currentSound);
}

// Cek tombol Previous
if (digitalRead(buttonPrev) == LOW && millis() - lastPrevButtonTime >
debounceDelay) {
    currentSound--;
    if (currentSound < 1) currentSound = maxSound;
    lastPrevButtonTime = millis();

    Serial.print("Tombol Prev ditekan. Sound: ");
    Serial.println(currentSound);
}

delay(50); // Delay kecil untuk menghindari beban loop
}

```

4.5.2.5 Program DFPlayer, Modul PAM, dan Speaker

```

#include <SoftwareSerial.h>
#include <DFRobotDFPlayerMini.h>

// Inisialisasi SoftwareSerial untuk DFPlayer (gunakan pin sesuai board Anda)
SoftwareSerial mySerial(20, 21); // RX, TX

// Objek DFPlayer
DFRobotDFPlayerMini myDFPlayer;

void setup() {
    // Inisialisasi komunikasi serial
    mySerial.begin(9600);
    Serial.begin(9600);

    // Inisialisasi DFPlayer
    if (!myDFPlayer.begin(mySerial)) {
        Serial.println("Tidak dapat menemukan DFPlayer Mini");
        while (true); // Berhenti jika tidak terdeteksi
    }

    // Atur volume (0 - 30)
    myDFPlayer.volume(20);

    // Tampilkan informasi awal
    Serial.println("DFPlayer Siap. Memainkan file MP3 nomor 1...");
}

void loop() {

```

```
myDFPlayer.play(1); // Mainkan file MP3 nomor 1  
delay(5000);        // Tunggu 5 detik sebelum memutar ulang  
}
```

