

## **BAB IV**

### **HASIL DAN UJI COBA**

#### **IV.1. Pengujian Rangkaian**

Pada bab ini, akan dibahas mengenai pengujian alat mulai dari pengujian alat permodul hingga pengujian alat secara keseluruhan. Pengujian tersebut akan dilakukan secara bertahap dengan urutan sebagai berikut :

1. Pengujian *Arduino UNO*
2. Pengujian Sensor Sidik Jari
3. Pengujian *DFplayer*
4. Pengujian Relay
5. Pengujian Keseluruhan

##### **IV.1.1. Pengujian Arduino UNO**

Pengujian arduino uno dilakukan bertujuan untuk mengetahui sistem board arduino uno dapat bekerja dengan baik sesuai dengan yang diharapkan. Pengujian dilakukan dengan menyalakan sebuah LED (L1) yang tersedia pada board arduino uno sendiri yang terhubung langsung dengan pin 13 (pin digital). Listing program menyalakan dan mematikan LED dapat dilihat pada tabel 4.1

Setelah rangkaian terpasang seperti pada gambar, langkah selanjutnya adalah memasukan program sederhana yang dibuat untuk menghidupkan rangkaian lampu led yang terhubung dengan pin-pin pada arduino uno tersebut dan berikut listing programnya :

```

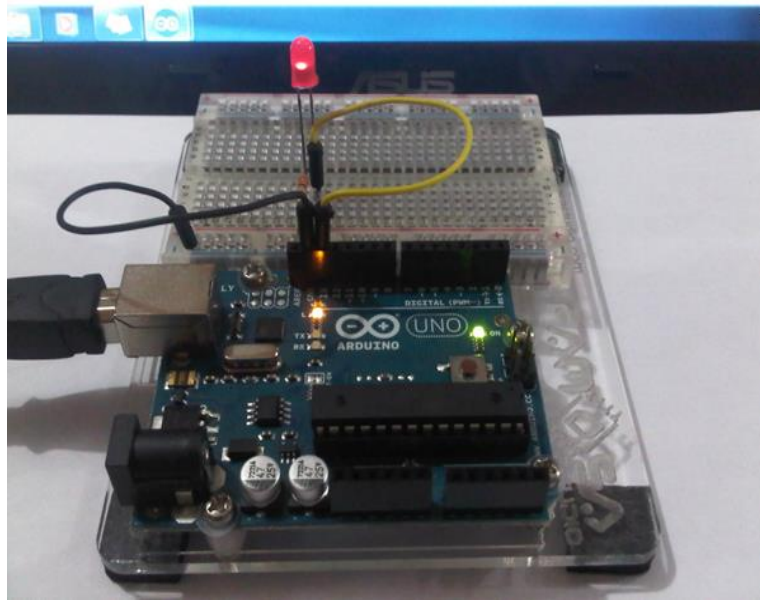
int ledPin = 13;

void setup()
{
pinMode(ledPin, OUTPUT);
}

void loop() {
digitalWrite(ledPin, HIGH);
delay(1000);
digitalWrite(ledPin, LOW);
delay(1000);
}

```

Pada tabel 4.1 di atas dapat dianalisa bahwa pada listing program tersebut dibuat sebuah variabel ledPin dengan type integer dengan nilai 13. Pada bagian void setup() variabel tersebut diatur sebagai keluaran dengan perintah pinMode (ledPin, OUTPUT). Pada bagian program utama void loop() dengan memberikan perintah digitalWrite (ledPin, HIGH) maka LED (L1) pada board arduino uno menyala. Perintah delay (1000) memberikan waktu tunda pada LED untuk bernilai HIGH selama 1000 mS. Perintah berikutnya digitalWrite (ledPin, LOW) maka LED akan padam. Perintah delay (1000) memberikan waktu tunda pada LED untuk bernilai LOW selama 1000 ms. Berdasarkan pengujian dan analisa yang telah dilakukan. Gambar menyalakan dan mematikan LED pada bord arduino uno dapat dilihat pada gambar 4.1



**Gambar IV.1. Rangkaian Arduino UNO**

#### **IV.1.2. Pengujian Sensor sidik jari**

Pengujian ini dilakukan dengan tujuan untuk mengetahui kemampuan modul fingerprint dalam scanning sidik jari. Fingerprint harus bisa membedakan sidik jari yang sudah terdaftar di EEPROM dan sidik jari yang belum terdaftar. Perintah program pendaftaran sidik jari adalah sebagai berikut

:

```
void setup()
{
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\nProses Enroll/Pendaftaran Sidik Jari");
  // set the data rate for the sensor serial port
  finger.begin(57600);
```

```
if (finger.verifyPassword()) {  
    Serial.println("Sensor Fingerprint Terdeteksi!");  
} else {  
    Serial.println("Sensor Fingerprint tidak terdeteksi :(");  
    while (1) { delay(1); }  
}  
}  
  
uint8_t readnumber(void) {  
    uint8_t num = 0;  
  
    while (num == 0) {  
        while (! Serial.available());  
        num = Serial.parseInt();  
    }  
    return num;  
}  
  
void loop()          // run over and over again  
{  
    Serial.println("Pendaftaran Sidik Jari siap digunakan!");  
    Serial.println("Masukan ID yang di pilih # (Dari 1 samapai 127)");  
    Serial.println("\nAnda akan menyimpan sidik jari ini dengan ID...");  
    id = readnumber();  
    if (id == 0) { // ID #0 not allowed, try again!
```

```

    return;
}
Serial.print("Enrolling ID #");
Serial.println(id);

while (! getFingerprintEnroll() );
}
uint8_t getFingerprintEnroll() {
    int p = -1;
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                break;
            case FINGERPRINT_IMAGEFAIL:

```





### IV.1.3. Pengujian *DFpalye*

Pengujian output suara DFPlayer dilakukan untuk mengetahui proses mengidentifikasi kata-kata yang kemudian akan dikonversikan kedalam bentuk suara. Dalam pengujian ini, proses identifikasi kata tersebut digunakan untuk memutar suara yang sudah tersimpan pada DFPlayer dengan format mp3. Berikut adalah programnya.

```
#include <Adafruit_Fingerprint.h>

#include <SoftwareSerial.h>

#include "Arduino.h"

#include "SoftwareSerial.h"

#include "DFRobotDFPlayerMini.h"

SoftwareSerial mini_MP3(11, 10); // RX, TX

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

DFRobotDFPlayerMini myDFPlayer;

#define CDI 8

#define stater 9

void(*ngulang)(void)=0;

String no_jari;
```

```
String aa;

int akurasi;

byte tempel=0;

byte data_akses=0;

byte status_mesin=0;

void setup()

{

  pinMode(CDI, OUTPUT);

  pinMode(stater, OUTPUT);

  digitalWrite(CDI, HIGH);

  digitalWrite(stater, HIGH);

  Serial.begin(9600);

  mini_MP3.begin(9600);

  Serial.println(F("DFRobot DFPlayer Mini Demo"));

  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if(!myDFPlayer.begin(mini_MP3)){

    Serial.println(F("Unable to begin:"));

    Serial.println(F("1.Please recheck the connection!"));

    Serial.println(F("2.Please insert the SD card!"));

    while(true){

      delay(0);
```

```
    }  
  }  
  
  Serial.println(F("DFPlayer Mini online."));  
  
  myDFPlayer.volume(30);  
  
  while (!Serial); // For Yun/Leo/Micro/Zero/...  
  
  delay(100);  
  
  Serial.println("\nMendeteksi Fingerprint yang terdaftar");  
  
  // set the data rate for the sensor serial port  
  
  delay(1000);  
  
  mini_MP3.end();  
  
  finger.begin(57600);  
  
  if (finger.verifyPassword()) {  
    Serial.println("Sensor Terdeteksi!");  
  } else {  
    Serial.println("Sensor Tidak Terdeteksi :(");  
    while (1) { delay(1); }  
  }  
  
  finger.getTemplateCount();  
  
  Serial.print("Terdapat "); Serial.print(finger.templateCount); Serial.println(" data sidik jari  
yang terdaftar");  
  
  Serial.println("Tempatkan Jari Anda...");
```

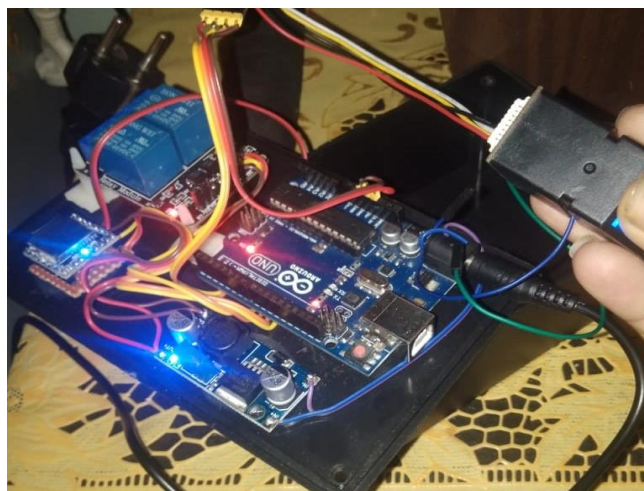
```

}

void loop()          // run over and over again
{
  //getFingerprintIDez();
  getFingerprintID();
  //Serial.println(no_jari);
  //Serial.println(akurasi);
  if(data_akses == 3){
    myDFPlayer.play(1);
    delay(300000);
    data_akses =0;
  }
}

```

Program diatas berfungsi mengaktifkan suara yang sudah di perintah kan melalui fingerprint.



**Gambar IV.4. Rangkaian System dan DFplayer**

#### IV.1.4. Pengujian Relay

Pengujian dilakukan untuk mengetahui apakah relay berfungsi atau tidaknya. berikut program nya:

```
#include <DFRobotDFPlayerMini.h>

/* 1 => blokir akses
 * 2 => welcome
 * 3 => start mesin
 * 4 => akses ditolak
 */

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mini_MP3(11, 10); // RX, TX

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

DFRobotDFPlayerMini myDFPlayer;

#define CDI 8

#define stater 9

void(*ngulang)(void)=0;

String no_jari;
```

```
String aa;

int akurasi;

byte tempel=0;

byte data_akses=0;

byte status_mesin=0;

void setup()

{

  pinMode(CDI, OUTPUT);

  pinMode(stater, OUTPUT);

  digitalWrite(CDI, HIGH);

  digitalWrite(stater, HIGH);

  Serial.begin(9600);

  mini_MP3.begin(9600);

  Serial.println(F("DFRobot DFPlayer Mini Demo"));

  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if(!myDFPlayer.begin(mini_MP3)){

    Serial.println(F("Unable to begin:"));

    Serial.println(F("1.Please recheck the connection!"));

    Serial.println(F("2.Please insert the SD card!"));

    while(true){

      delay(0);

    }

  }

}
```

```
Serial.println(F("DFPlayer Mini online."));

myDFPlayer.volume(30);

while (!Serial); // For Yun/Leo/Micro/Zero/...

delay(100);

Serial.println("\nMendeteksi Fingerpint yang terdaftar");

// set the data rate for the sensor serial port

delay(1000);

mini_MP3.end();

finger.begin(57600);

if (finger.verifyPassword()) {

  Serial.println("Sensor Terdeteksi!");

} else {

  Serial.println("Sensor Tidak Terdeteksi :(");

  while (1) { delay(1); }

}

finger.getTemplateCount();

Serial.print("Terdapat "); Serial.print(finger.templateCount); Serial.println(" data sidik jari yang
terdaftar");

Serial.println("Tempatkan Jari Anda...");

}

void loop()          // run over and over again
```

```
{  
  //getFingerprintIDez();  
  getFingerprintID();  
  //Serial.println(no_jari);  
  //Serial.println(akurasi);  
  if(data_akses == 3){  
    myDFPlayer.play(1);  
    delay(300000);  
    data_akses =0;  
  }  
  if(no_jari == "1" || no_jari == "2" || no_jari == "3"){  
    if(akurasi >= 70){  
      digitalWrite(CDI, LOW);  
      myDFPlayer.play(2);  
      no_jari ="0";  
      akurasi =0;  
      data_akses =0;  
      Start();  
    }  
  }  
  delay(50);    //don't ned to run this at full speed.  
}  
void Start(){  
  byte a=1;
```

```
while(a==1){  
    getFingerprintIDst();  
    if(no_jari == "1" || no_jari == "2" || no_jari == "3"){  
        if(akurasi >= 70){  
            if(status_mesin == 0){  
                myDFPlayer.play(3);  
                delay(2000);  
                digitalWrite(stater, LOW);  
                no_jari ="0";  
                akurasi =0;  
                status_mesin=1;  
            }  
            else if(status_mesin == 1){  
                status_mesin=0;  
                //ngulang();  
                digitalWrite(CDI, HIGH);  
                no_jari ="0";  
                akurasi =0;  
                a=0;  
            }  
        }  
    }  
}
```

```
}  
  
//=====CDI on=====  
  
uint8_t getFingerprintID() {  
  
    uint8_t p = finger.getImage();  
  
    switch (p) {  
  
        case FINGERPRINT_OK:  
  
            Serial.println("Image taken");  
  
            break;  
  
        case FINGERPRINT_NOFINGER:  
  
            //Serial.println("No finger detected");  
  
            tempel=0;  
  
            return p;  
  
        case FINGERPRINT_PACKETRECEIVEERR:  
  
            Serial.println("Communication error");  
  
            return p;  
  
        case FINGERPRINT_IMAGEFAIL:  
  
            Serial.println("Imaging error");  
  
            return p;  
  
        default:  
  
            Serial.println("Unknown error");  
  
            return p;  
  
    }  
  
}
```

program di atas menghidupkan relay dengan relay dengan perintah fingerprint “jika fingerprint yang sudah terdaftar di lengket kan maka lampu pada relay akan hidup” maka relay bias di gunakan.



**Gambar IV.5. Rangkaian *Minimum System* dan Relay**

## **IV.2. Pengujian Keseluruhan**

Hasil perancangan keseluruhan strat engine sepeda motor berbasis Arduino.



**Gambar IV.6. Tampilan Alat Secara Keseluruhan**

Dari gambar IV.6 tampak bahwa alat ini terdiri dari arduino, sensor sidik jari, DFplayer, relay

#### **IV.2.1. Proses Kerja Alat**

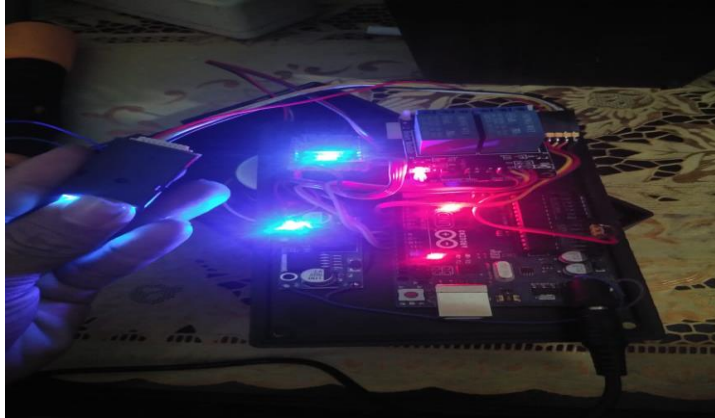
Langkah awal dari proses pengujian alat ini adalah dengan menjalankan dan melihat proses kerja alat. Langkah pertama adalah menulis program start engine. Selanjutnya setelah program di *upload* ke arduino dan semua rangkaian telah terhubung selanjutnya alat diberi daya untuk menghidupkannya, maka alat akan langsung aktif.



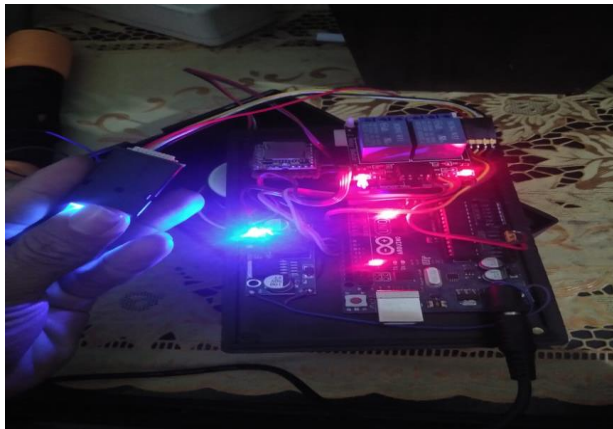
**Gambar IV.7. Tampilan Alat Setelah Aktif**

#### **IV.2.2. Pengujian Alat I**

Pada pengujian alat menggunakan sensor sidik jari yang digunakan sebagai penentu aktifnya relay yang ada 3 jenis yaitu hidup nya kunci kontak, start engine, mati nya kunci kontak. Pada pengujian I ini dilakukan pengujian dengan cara menempel sidik jari yang sudah tersimpan pada fingerprint lalu kunci kontak akan aktif setelah itu tempelkan lagi maka stater akan aktif pada fingerprint.



**Gambar IV.8. Tampilan Kunci Kontak hidup**



**Gambar IV.9. Tampilan Starter Pada Relay Hidup**