

MODUL 5B DOOR LOCK WITH FINGERPRINT SYSTEM

Nur Sekti Waskitha J (18318023)

Aditya Tresnobudi (18318024)

Asisten Percobaan: Devi Tara Avalokita

Tanggal Percobaan: 19/11/2019

EL2142-Sistem Digital dan Mikroprosesor

Laboratorium Dasar Teknik Elektro - Sekolah Teknik Elektro dan Informatika ITB



1. ALAT YANG AKAN DIBUAT

Alat yang akan kelompok kami buat adalah sebuah system kunci pintu dengan sensor sidik jari berbasis Arduino.

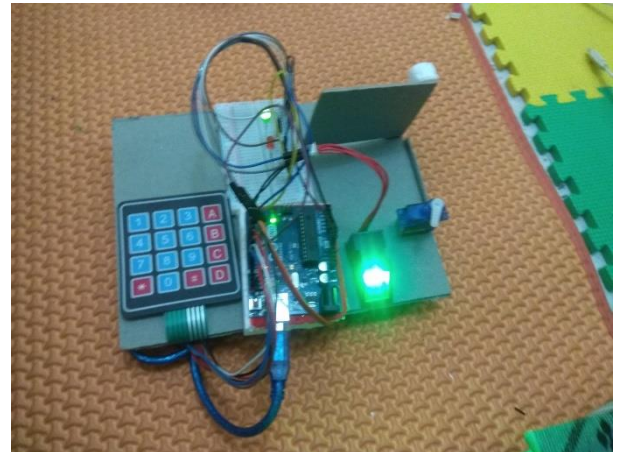
2. LATAR BELAKANG

Sistem keamanan untuk rumah atau ruangan tertentu menjadi hal yang penting dalam kehidupan sehari-hari, system keamanan tersebut selalu berkembang dari waktu ke waktu, mulai dari bentuk slot sampai kunci yang kita lihat hari ini.

Setiap jenis system keamanan yang ada punya kelebihan dan kelemahan masing-masing, mulai dari kepraktisan sampai dengan tingkat keamanan. Setiap system juga memiliki kekurangannya masing-masing, namun kekurangan yang paling utama adalah kepraktisan dan keamanan dari system tersebut, dalam hal kepraktisan terkadang kunci bisa hilang karena bentuknya yang kecil dan tidak nyaman dibawa kemana-mana, dalam hal keamanan system kunci rumah bisa dilakukan *picklocking* dengan cukup mudah.

Tujuan kami membuat alat ini adalah agar ada sebuah system keamanan untuk rumah atau ruangan yang bisa memberi solusi dari kekurangan-kekurangan system sebelumnya. Sistem fingerprint memberikan system keamanan yang spesifik (karena sidik jari yang dimiliki oleh setiap manusia spesifik) sehingga menurunkan resiko pembobolan, juga sidik jari selalu ada bersama dengan pemiliknya, sehingga bisa menghindarkan dari kesulitan dalam menggunakan system kunci pintu ini.

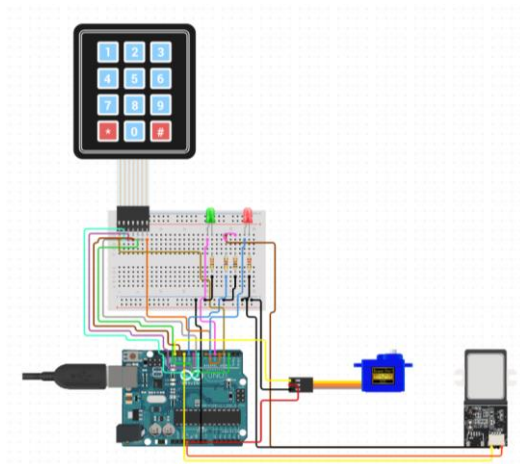
3. GAMBAR ALAT



Gambar 3-1 Door Lock with Fingerprint System

4. DESKRIPSI DAN GAMBARAN UMUM ALAT

4.1 GAMBARAN UMUM



Gambar 4.1-2 Arsitektur Rangkaian

4.2 DESKRIPSI

Alat dibuat dengan 2 buah input yaitu fingerprint sensor dan keypad, dan menggunakan 2 buah output yaitu 2 LED dan Servo

Fingerprint sensor dipasang pada pin 2 untuk Tx dan pin 3 untuk Rx arduino, pin Vcc dan pin Ground dipasang pada breadboard untuk disambung pada pin Vcc dan Ground Aduino.

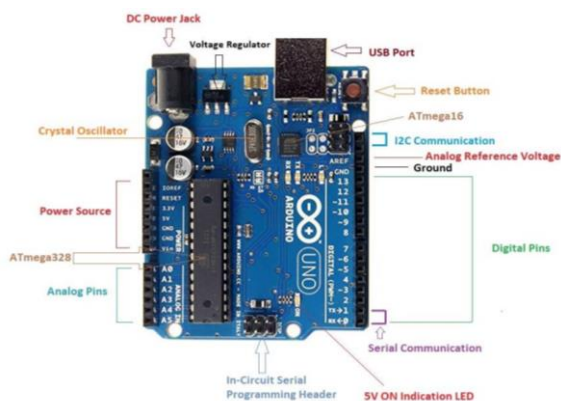
Keypad dipasang pada pin 6 sampai pin 12 pada Arduino, untuk setiap kolom dan baris.

Servo dipasang pada pin 13 sesuai dengan referensi, Vcc dipasang pada breadboard untuk disambung pada pin Vcc Arduino, begitu pula untuk pin Ground.

LED merah pada pin 9 arduino dan LED hijau pada pin 10 arduino, untuk setiap LED dipasang resistor 220Ω, Servo dipasang pada pin 12.

5. SPESIFIKASI

5.1 ARDUINO



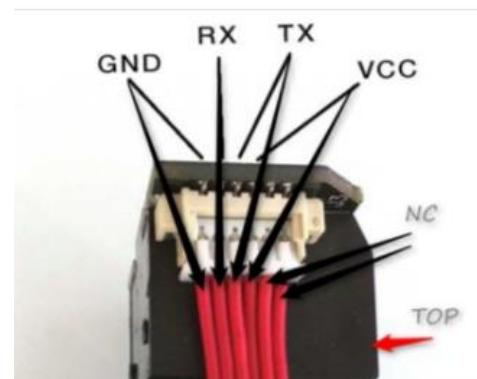
Gambar 5.1 Arduino Uno

Mikroprosesor yang digunakan pada alat yang dibuat adalah Arduino dengan seri Uno, pin-pin pada Arduino Uno bisa dilihat pada gambar 5.1.

5.2 FINGERPRINT SENSOR



Gambar 5.2-1 Fingerprint Sensor



Gambar 5.2-2 Pin pada Fingerprint Sensor

Fingerprint Sensor yang digunakan menggunakan Library Adafruit, Fingerprint sensor membutuhkan power supply sebesar 3.6-6V, fingerprint sensor berukuran 1.4x1.8 cm dan bisa menyimpan sampai 162 ID sidik jari. Fingerprint sensor bisa bekerja optimum pada suhu -20°C sampai 50°C

5.3 KEYPAD

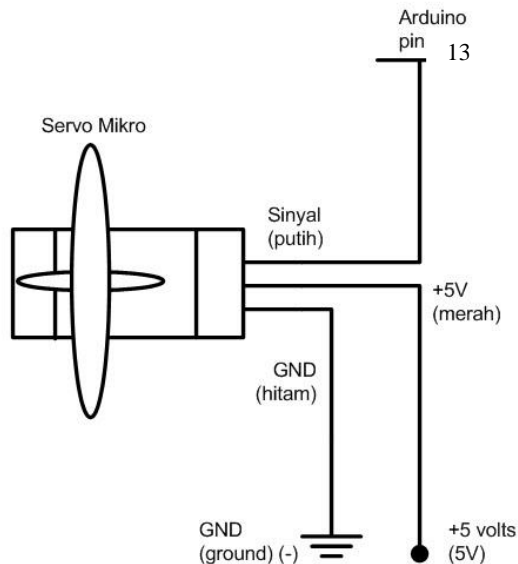


Gambar 5.3 Keypad

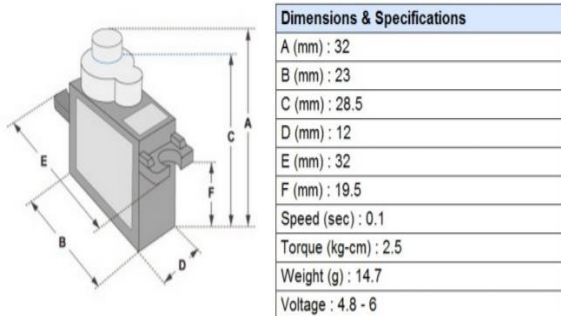
Keypad mempunyai 8 pin, dari kiri ke kanan mulai dari pin untuk kolom 4, kolom 3, kolom 2, kolom 1, baris 4, baris 3, baris 2, baris 1.

Keypad dapat digunakan secara optimum pada suhu 0°C sampai 50°C, dan berdimensi sebesar 6.9x7.6 cm.

5.4 SERVO



Gambar 5.4-1 Pin Servo



Gambar 5.4-2 Spesifikasi Servo

Servo bisa digunakan dengan power supply sebesar 4.8 sampai 6 V, mempunyai dimensi seperti yang dituliskan pada gambar 5.3-2, servo bisa diatur kecepatannya pada program, pada program servo menggunakan Library Servo.h.

6. CARA KERJA DAN FLOWCHART SISTEM

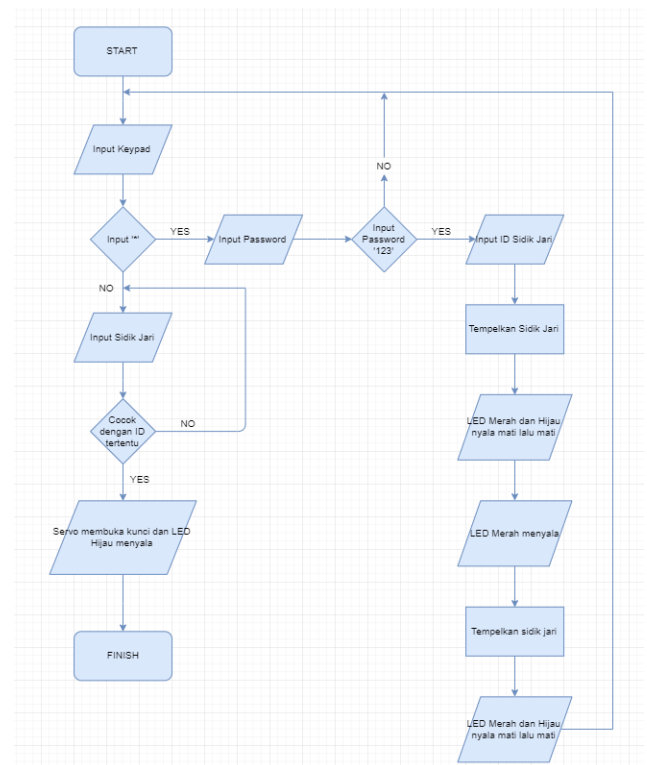
6.1 CARA KERJA

Door Lock with Fingerprint system mempunyai 2 fungsi kerja, yaitu enroll dan open the door, fungsi enroll digunakan untuk mendaftarkan sidik jari baru pada ID tertentu, dan fungsi open the door digunakan untuk membuka pintu.

Ketika fungsi enroll digunakan, maka Arduino akan digunakan untuk menyimpan data sidik jari baru, sidik jari baru bisa ditambahkan ketika password yang dimasukkan pada keypad benar.

Fungsi open the door membuat Fingerprint sensor digunakan untuk menerima masukan sidik jari pengguna dan bila cocok dengan sidik jari pada ID tertentu, maka akan memutar servo dan membuka kunci pintu juga menyalakan LED.

6.2 FLOWCHART



Gambar 6.2-3 Flowchart Door Lock with Fingerprint System

7. CARA PENGGUNAAN ALAT

Sambungkan system dengan power supply, bila pengguna ingin menambahkan sidik jari baru, maka pengguna harus menekan "*" pada keypad agar system memasuki fungsi enroll, setelah itu pengguna harus memasukkan password yang tepat sehingga system enroll dapat digunakan dengan benar, kebenaran password yang dimasukkan ditandai dengan LED, bila LED hijau menyala mati maka password yang dimasukkan benar, dan bila LED merah menyala mati maka password yang dimasukkan salah.

Untuk mendaftarkan sidik jari, pengguna harus memilih pada ID berapa sidik jari baru akan disimpan dengan cara memencet angka pada keypad, setelah itu pengguna menempelkan sidik jari pada Fingerprint Sensor sebanyak 2 kali dengan konfirmasi ditunjukkan oleh LED yang nyala mati.

Setelah sidik jari didaftarkan, maka pengguna akan bisa membuka pintu dengan fungsi open the door, pengguna hanya cukup menempelkan sidik jari pada Fingerprint sensor, bila sidik jari yang ditempel telah terdaftar pada ID tertentu maka servo akan berputar dan membuka kunci juga ditandai dengan LED hijau yang menyala, bila sidik jari tidak terdaftar maka tidak akan terjadi apapun.

8. HASIL DAN ANALISIS

Alat yang kami buat adalah *door lock with fingerprint system*. Alat tersebut berhasil dibuat dengan menggunakan dua input device, yaitu keypad dan sensor fingerprint, sedangkan output device-nya adalah servo yang akan berputar, menandakan pintu terbuka.

Salah satu input device yang digunakan dalam alat ini adalah keypad. Keypad disini digunakan untuk mengetikkan password jika ingin menyimpan sidik jari yang baru. Keypad bekerja dengan menginput karakter yang ditekan. Ketika karakter tersebut ditekan, maka jumper yang bersesuaian dengan baris dan kolom karakter tersebut akan teraliri arus sehingga akan diinputkan data digital yang bersesuaian dengan karakter yang diinput.

Dalam alat ini, password yang diset bisa diatur di dalam source code sehingga bisa diubah sesuka hati. Jika password yang dimasukkan salah, LED merah akan berkedip nyala mati, lalu keluar dari fungsi input password. Sementara jika password yang dimasukkan benar, maka sistem akan berlanjut ke penyimpanan data sidik jari.

Fingerprint sensor yang digunakan menggunakan prinsip optical scanning dalam bekerja. Proses scan mulaiberlangsung saat seseorang meletakkan jari pada lempengan kaca dan sebuah kamera CCD (Charge Coupled Device) mengambil gambarnya. Sensor ini dilengkapi dengan LED hijau yang digunakan untuk menyinari alur sidik jari yang akan disimpan. Sistem CCD menghasilkan gambar jari yang terbalik, area yang lebih gelap merepresentasikan lebih banyak cahaya yang dipantulkan (bagian punggung dari alur sidik jari), dan area yang lebih terang merepresentasikan lebih sedikit cahaya yang dipantulkan (bagian lembah dari alur sidik jari). Setiap titik photosite yang dihasilkan akan diakumulasikan dalam bentuk sinyal elektronis. Sinyal tersebut akhirnya dikalkulasikan dalam bentuk digital untuk akhirnya disimpan di Arduino (maksimal ada 162 data yang disimpan berdasarkan datasheet dari sensor fingerprint). Setelah data disimpan, fungsi akan kembali ke open the door.

Ketika fingerprint sensor digunakan untuk menghidupkan servo (kunci pintu), sensor akan bekerja seperti saat menyimpan data sidik jari. Data digital sidik jari yang dihasilkan ketika kita menempelkan jari, disamakan dengan data yang telah disimpan dalam Arduino. Arduino akan mencocokkan data dari awal hingga ditemukan data digital yang sama persis. Jika hingga akhir data tidak ditemukan, servo tidak akan bekerja. Jika ada data yang sama, arus ke servo (pin 13) akan terhubung sehingga motor servo berputar sejauh 180°, lalu ditahan selama 3 detik, dilanjutkan dengan berputar kembali posisi awal.

Seperti yang kita tahu, servo yang digunakan akan bekerja jika ada sidik jari yang sesuai dengan data yang disimpan, namun kenyataanya ketika pertama kali Arduino terhubung sumber, servo akan bergerak. Gerakan servo tersebut berfungsi sebagai kalibrasi posisi motor servo agar berada di posisi yang sesuai dengan kode yang diupload ke Arduino. Selain itu, servo bergetar sesaat setelah kembali ke posisi awal. Hal tersebut karena servo akan terus bergerak ke posisi parameter sudut yang telah diberikan. Jadi, motor servo akan bergerak sedikit lalu kembali ke posisi awal (bergetar). Hal tersebut akan terus dilakukan servo selama arus terhubung.

9. LAMPIRAN

```
#include <Adafruit_Fingerprint.h>

#include <Servo.h> // menyertakan library servo
ke dalam program

#include <Keypad.h>

const byte ROWS = 4; // Four rows
const byte COLS = 3; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*', '0', '#'}
};

// Connect keypad ROW0, ROW1, ROW2 and
ROW3 to these Arduino pins.

byte rowPins[ROWS] = { 12, 11, 10, 9 };

// Connect keypad COL0, COL1 and COL2 to
these Arduino pins.

byte colPins[COLS] = { 8, 7, 6 };

// Create the Keypad
Keypad kpd = Keypad( makeKeymap(keys),
rowPins, colPins, ROWS, COLS );

Servo myservo; // variable untuk menyimpan
posisi data

int electronic_lock = 13;

// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3); // Tx, Rx

int ledRed = 4;
int ledGreen = 5;

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup()
{
  myservo.attach(13);
  pinMode(13, OUTPUT);
  pinMode(ledRed, OUTPUT);
```

```
pinMode(ledGreen, OUTPUT);

Serial.begin(9600);
while (!Serial);
delay(100);
// set the data rate for the sensor serial port
finger.begin(57600);
myservo.write(0);
Serial.println("Found fingerprint sensor");
finger.getTemplateCount();
Serial.print("Sensor contains ");
Serial.print(finger.templateCount);
Serial.println(" template(s)");
}

uint8_t readnumber(void) {
  uint8_t num = 0;

  while (num == 0) {
    num = kpd.getKey();
  }
  return num;
}

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(".");
        delay(50);
        break;
      case FINGERPRINT_PACKETRECEIVEERR:
```

```

    Serial.println("Communication error");
    break;
case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
default:
    Serial.println("Unknown error");
    break;
}
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    for (int i = 1; i < 4; i++){
        digitalWrite(ledRed, HIGH);
        digitalWrite(ledGreen, HIGH);
        delay(300);
        digitalWrite(ledRed, LOW);
        digitalWrite(ledGreen, LOW);
        delay(300); //LED merah dan hijau berkelap-
        kelip
    }
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint
    features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint
    features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
digitalWrite(ledRed, HIGH);
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
case FINGERPRINT_OK:
        Serial.println("Image taken");
        digitalWrite(ledRed, LOW);
        for (int i = 1; i < 4; i++){
            digitalWrite(ledRed, HIGH);
            digitalWrite(ledGreen, HIGH);
            delay(300);
            digitalWrite(ledRed, LOW);
            digitalWrite(ledGreen, LOW);
            delay(300); //LED merah dan hijau berkelap-
            kelip
        }
        break;
case FINGERPRINT_NOFINGER:
        Serial.print(".");
        delay(50);
        break;
case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        break;

```

```

case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
default:
    Serial.println("Unknown error");
    break;
delay(500);
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint
features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint
features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
Serial.print("Creating model for #");
Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p ==
FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p ==
FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
} else if (p ==
FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}
}

uint8_t saveFinger(){
    for (int i = 1; i <4; i++){
        digitalWrite(ledRed, HIGH);
        digitalWrite(ledGreen, HIGH);
    }
}

```



```

    delay(300);
    digitalWrite(ledRed, LOW);
    digitalWrite(ledGreen, LOW);
    delay(300); //LED merah dan hijau berkelap-
kelip
}
digitalWrite(ledRed, HIGH);
char pass1 = readnumber();
Serial.println(pass1);
char pass2 = readnumber();
Serial.println(pass2);
char pass3 = readnumber();
Serial.println(pass3);
//set sebagai password
if (pass1 == '1'){
    if (pass2 == '2'){
        if (pass3 == '3'){ //set sebagai password
            digitalWrite(ledRed, LOW);
            for (int i = 1; i <4; i++){
                digitalWrite(ledGreen, HIGH);
                delay(300);
                digitalWrite(ledGreen, LOW);
                delay(300);} //LED hijau berkelap-kelip
            Serial.println("Ready to enroll a
fingerprint!");
            Serial.println("Please type in the ID # (from
1 to 127) you want to save this finger as...");
            delay(500);
            id = readnumber();
            Serial.print("Enrolling ID #");
            Serial.println(id);
            for (int i = 1; i <4; i++){
                digitalWrite(ledGreen, HIGH);
                delay(300);
                digitalWrite(ledGreen, LOW);
                delay(300);} //LED hijau berkelap-kelip
            getFingerprintEnroll();
        }
    }
}
else{
    for (int i = 1; i <4; i++){
        digitalWrite(ledRed, HIGH);
        delay(300);
        digitalWrite(ledRed, LOW);
        delay(300);}
    }
}
else{
    if (pass2 == '0' or pass2 == '1' or pass2 == '2'
or pass2 == '3' or pass2 == '4' or pass2 == '5' or pass2
== '6' or pass2 == '7' or pass2 == '8' or pass2 == '9'){
        for (int i = 1; i <4; i++){
            digitalWrite(ledRed, HIGH);
            delay(300);
            digitalWrite(ledRed, LOW);
            delay(300);}
        }
    }
}
else {
    if (pass2 == '0' or pass2 == '1' or pass2 == '2' or
pass2 == '3' or pass2 == '4' or pass2 == '5' or pass2
== '6' or pass2 == '7' or pass2 == '8' or pass2 == '9'){
        if (pass3 == '0' or pass3 == '1' or pass3 == '2'
or pass3 == '3' or pass3 == '4' or pass3 == '5' or pass3
== '6' or pass3 == '7' or pass3 == '8' or pass3 == '9'){
            for (int i = 1; i <4; i++){
                digitalWrite(ledRed, HIGH);
                delay(300);
                digitalWrite(ledRed, LOW);
                delay(300);}
            }
        }
    }
}
}

int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;
}

```



```

p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;

// found a match!
Serial.print("Found          ID          #");
Serial.print(finger.fingerID);

Serial.print("    with confidence    of    ");
Serial.println(finger.confidence);

myservo.write(180);
digitalWrite(ledRed, LOW);
digitalWrite(ledGreen, HIGH);
Serial.println("Unlocked");
delay(4000);
myservo.write(0); // membuka pintu
digitalWrite(ledRed, HIGH);
digitalWrite(ledGreen, LOW);
delay(1000);
return finger.fingerID;
}

void loop()          // run over and over again
{
    char key = kpd.getKey();
    Serial.println(key);
    Serial.println("Waiting for valid finger...");
    digitalWrite(ledRed, HIGH);
    getFingerprintIDez(); // mengecek sidik jari
    untuk membuka pintu (servo)
    delay(1);
    if (key == '*') {
        saveFinger(); // menyimpan sidik jari baru
    }
}

```