

LAMPIRAN

Lampiran. 1 Sketch Program

```
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_Fingerprint.h>
#include <Keypad.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial mySerial(11,12); //11=kuning, 12=coklat

int u=0;
uint8_t id, ID_terbaca;

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);

#define resetPin A0
#define resetLed A3
#define ledPin A1
#define relayPin A2

char key, k1,k2,k3,k4;
String jawab, kunci, ganti;
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
int hitung = 0;

char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {9,8,7,6}; //connect to the row pinouts
of the keypad
byte colPins[COLS] = {5,4,3,2}; //connect to the column
pinouts of the keypad
Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins,
```

```

ROWS, COLS );

void setup() {
  Serial.begin(9600);
  finger.begin(57600);
  pinMode(ledPin, OUTPUT);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);
  pinMode(13,OUTPUT);
  byte k0 = EEPROM.read(1);
  if(k0==255) {
    EEPROM.write(1,'1');
    EEPROM.write(2,'2');
    EEPROM.write(3,'3');
    EEPROM.write(4,'4');
  }
  baca_kunci();
  lcd.init();
  lcd.setBacklight(50);

  if(finger.verifyPassword()) {
    bacaParameter();
    lcd.clear(); lcd.print(F("Module Terdeksi "));
    lcd.setCursor(0,1);
    lcd.print(F("Sidik Ada : "));
    lcd.print(finger.templateCount);
    delay(2000);
  }
  else {
    lcd.setCursor(0,0); lcd.print(F("Koneksi Error  "));
    lcd.setCursor(0,1); lcd.print(F("Periksa Koneksi "));
    while (1);
  }
  intro();
}

void loop() {
  //check key pressed
  key = myKeypad.getKey();

  if(key=='#'){
    key = NO_KEY; while (key == NO_KEY) {key =

```

```

myKeypad.getKey();};
    if(key=='#') {
        lcd.clear(); lcd.setCursor(0, 0); lcd.print(F("Input
Password !"));
        cek_password(); if(jawab==kunci) checkKeys();
        else { lcd.clear(); lcd.setCursor(0, 0);
            lcd.print(F("Password Salah !"));
            delay(3000); intro();
        }
    }
}

if(key=='*'){
    key = NO_KEY; while (key == NO_KEY) {key =
myKeypad.getKey();};
    if(key=='*') {
        lcd.clear(); lcd.setCursor(0, 0); lcd.print(F("Enter
Old Pswd !"));
        cek_password();
        if(jawab==kunci){
            lcd.clear();
            lcd.print(F("Enter New Pswd !"));
ganti_password();
        }
        else { lcd.clear(); lcd.setCursor(0, 0);
            lcd.print(F("Password Salah !"));
            delay(3000); intro();
        }
    }
}}

if(key=='A' || key=='B'){
    if(digitalRead(relayPin)==HIGH){
        for(int i=0;i<3;i++){ //cek sidik jari 3 kali
            lcd.clear(); lcd.print(F("Letakkan Jari !!"));
delay(1000);
            getFingerprintID(); delay(500);
        }
    }
    if(digitalRead(relayPin)==LOW){
        lcd.clear(); lcd.print(F(" Kunci Enable  "));
        delay(5000);
    }
}

```

```

    }
    intro();
}

}

void intro() {
    lcd.setCursor(0,0); lcd.print(F("Security System "));
    lcd.setCursor(0,1); lcd.print(F(" by MARIANTO "));
}

void baca_kunci() {
    k1 = char(EEPROM.read(1));
    k2 = char(EEPROM.read(2));
    k3 = char(EEPROM.read(3));
    k4 = char(EEPROM.read(4));

    kunci= k1;
    kunci= kunci+k2;
    kunci= kunci+k3;
    kunci= kunci+k4;
}

void checkKeys(){
    lcd.clear();          lcd.print(F("C Untuk Merekam "));
    lcd.setCursor(0, 1); lcd.print(F("D Untuk Hapus "));

    key = NO_KEY;
    while (key == NO_KEY) {key = myKeypad.getKey();};

    if(key=='C') { //to enroll
        lcd.clear();
        lcd.print(F("Mohon Tunggu"));
        delay(1000);
        Enroll();
    }
    else if(key=='D') { //to delete
        lcd.clear();
        lcd.print(F("Mohon Tunggu"));
        delay(1000);
        delet();
    }
}

```

```

    else {key = NO_KEY;}
}

void Enroll() {
    int count=0;
    lcd.clear(); lcd.print(F("Merekam Sidik "));
    lcd.setCursor(0,1); lcd.print(F("Nomor ID :"));

    while(1){
        lcd.setCursor(10,1);
        if(count<10) lcd.print("0");
        lcd.print(count);

        key = NO_KEY;
        while (key == NO_KEY) {key = myKeypad.getKey();};

        if(key=='A'){count++; if(count>25) count=0; }
        else if(key=='B') {count--; if(count<0) count=25; }
        else if(key == 'C') { //Enroll
            id=count; getFingerprintEnroll(); return;
        }
        else { key=NO_KEY; return;} //exit
    }
}

void cek_password() {
    jawab="";
    // jawaban 4 digit pertama yang masuk dipindah ke string
    key = NO_KEY;
    while (key == NO_KEY) {key = myKeypad.getKey();};
    jawab += key; lcd.setCursor(0, 1); lcd.print("");
    key = NO_KEY;
    while (key == NO_KEY) {key = myKeypad.getKey();};
    jawab += key; lcd.setCursor(1, 1); lcd.print("");
    key = NO_KEY;
    while (key == NO_KEY) {key = myKeypad.getKey();};
    jawab += key; lcd.setCursor(2, 1); lcd.print("");
    key = NO_KEY;
    while (key == NO_KEY) {key = myKeypad.getKey();};
    jawab += key; lcd.setCursor(3, 1); lcd.print("");
    //lcd.setCursor(10, 1); lcd.print(jawab);
    delay(1000);
}

```

```

}

void ganti_password() {
    cek_password();
    ganti=jawab;

    lcd.clear();
    lcd.setCursor(0, 0); lcd.print(F("ReEnter New Pswd"));

    cek_password();

    if(jawab==ganti) {
        lcd.setCursor(0, 0); lcd.print(F(" New Password  "));
        lcd.setCursor(0, 1); lcd.print(F("      OK      "));
        delay(3000);

        EEPROM.write(1,jawab[0]); //jawab adalah variabel array
6 elemen (byte)
        EEPROM.write(2,jawab[1]);
        EEPROM.write(3,jawab[2]);
        EEPROM.write(4,jawab[3]);

        baca_kunci();
    }
    else {lcd.clear();
        lcd.setCursor(0, 0); lcd.print(F(" New Password
"));
        lcd.setCursor(0, 1); lcd.print(F("  ERROR !!!
"));
        delay(3000);
    }
}

uint8_t getFingerprintEnroll() {
    int p = -1;
    lcd.clear(); lcd.print(F("Nomor ID:")); lcd.print(id);
    lcd.setCursor(0,1); lcd.print(F("Letakkan Jari "));
    delay(2000);

    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");
    lcd.clear(); lcd.print(F("Membaca Sidik "));
    break;
case FINGERPRINT_NOFINGER:
    Serial.println(".");
    lcd.clear(); lcd.print(F("Tak Ada Jari"));
    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");
    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");
    lcd.clear(); lcd.print(F("Angkat Jari Anda"));
    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");
    lcd.clear();          lcd.print(F("Letakkan Jari"));
    lcd.setCursor(0,1); lcd.print(F("  Kembali  "));

    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                lcd.clear(); lcd.print(F("Membaca Sidik "));
                break;
            case FINGERPRINT_NOFINGER:
                Serial.print(".");
                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(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;
    }

    return true;
}

int getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken"); break;
        case FINGERPRINT_NOFINGER:
            Serial.println("No finger detected");
            ID_terbaca = 0;
            finger.LEDcontrol(FINGERPRINT_LED_OFF, 0,
FINGERPRINT_LED_BLUE);
            finger.LEDcontrol(FINGERPRINT_LED_OFF, 0,
FINGERPRINT_LED_RED);
            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;
    }

    // OK success!

```

```

p = finger.image2Tz();
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!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
  finger.LEDcontrol(FINGERPRINT_LED_FLASHING, 25,
FINGERPRINT_LED_PURPLE, 10);
  delay(1000);
  if(u==0) { digitalWrite(ledPin,HIGH); u=1; }
  else if(u==1) { digitalWrite(ledPin,LOW); u=0; }
  digitalWrite(relayPin,LOW);
}
else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  Serial.println("Communication error");
  return p;
}
else if (p == FINGERPRINT_NOTFOUND) {
  finger.LEDcontrol(FINGERPRINT_LED_FLASHING, 25,
FINGERPRINT_LED_RED, 10);
  delay(1000);
  Serial.println("Did not find a match"); return p;
}
else {
  Serial.println("Unknown error"); return p;
}

```

```

    }

    // found a match!
    ID_terbaca=finger.fingerID;
    Serial.print("Found ID #"); Serial.print(finger.fingerID);
    Serial.print(" with confidence of ");
    Serial.println(finger.confidence);
    return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
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) {
        lcd.clear(); lcd.print(F("Anda Gagal "));
        lcd.setCursor(0,1); lcd.print(F("Coba lagi !"));
        delay(2000);
        return -1;
    }

    // found a match!
    ID_terbaca=finger.fingerID;
    Serial.print("Found ID #"); Serial.print(finger.fingerID);
    Serial.print(" with confidence of ");
    Serial.println(finger.confidence);
    return finger.fingerID;
}

void bacaParameter() {
    Serial.println(F("Reading sensor parameters"));
    finger.getParameters();
    Serial.print(F("Status: 0x"));
    Serial.println(finger.status_reg, HEX);
    Serial.print(F("Sys ID: 0x"));
    Serial.println(finger.system_id, HEX);
    Serial.print(F("Capacity: "));

```

```

Serial.println(finger.capacity);
  Serial.print(F("Security level: "));
Serial.println(finger.security_level);
  Serial.print(F("Device address: "));
Serial.println(finger.device_addr, HEX);
  Serial.print(F("Packet lengt: "));
Serial.println(finger.packet_len);
  Serial.print(F("Baud rate: "));
Serial.println(finger.baud_rate);

  finger.getTemplateCount();

  if (finger.templateCount == 0) {
    Serial.print("Sensor tidak berisi data sidik jari.
    Jalankan 'enroll'");
  }
  else {
    Serial.println("Menunggu Sidik yang valid...");
    Serial.print("Sensor Berisi ");
    Serial.print(finger.templateCount);
    Serial.println(" Sidik Jari");
  }
}

void delet(){
  int count=0;
  lcd.clear(); lcd.print(F("Hapus Sidik Jari"));
  lcd.setCursor(0,1); lcd.print(F("Nomor ID :"));

  while(1){
    lcd.setCursor(10,1);
    if(count<10) lcd.print(" "); lcd.print(count);

    key = NO_KEY;
    while (key == NO_KEY) {key = myKeypad.getKey(); };

    if(key == 'A') { count++; if(count>25) count=0; }
    else if(key=='B') { count--; if(count<0) count=25; }
    else if(key=='D') { //Delete
      id=count; deleteFingerprint(id); return;
    }
    else { key=NO_KEY; return; } //exit
  }
}

```

```

    }
}

int deleteFingerprint(uint8_t id) {
    int p = -1;
    lcd.clear();
    lcd.print(F("Mohon Tunggu"));
    p = finger.deleteModel(id);
    if (p == FINGERPRINT_OK) {
        Serial.println("Deleted!");
        lcd.clear(); lcd.print(F("Sidik Terhapus"));
        lcd.setCursor(0,1); lcd.print(F("Sukses"));
        delay(1000);
    }

    else {
        Serial.print("Something Wrong");
        lcd.clear(); lcd.print(F("Ada Kesalahan "));
        lcd.setCursor(0,1); lcd.print(F("Coba Lagi "));
        delay(2000);
        return p;
    }
}
}

```

Lampiran. 2 Tampak Alat





Lampiran. 3 Dokumentasi Pengujian Alat





KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : Marianto Kurniawan Yudi	Pembimbing I : Ir. A. Abd. Jabbar, MT.
NIM : 217180044	Pembimbing II : A. Irmayani Pawelisi ST., MT.
Judul Skripsi : Rancang Bangun Sistem Pengaman Kuro Sepeda Motor menggunakan Sirkuit Jari dan e-KTP berbasis Mikrokontroler	

ARAHAN PEMBIMBING I	HARITGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARITGL & PARAF PEMBIMBING
Konsultasi 1 <i>perbaikan modul.</i>	<i>[Signature]</i>	Konsultasi 1 <i>Teori Bus II bagian dari DF</i>	<i>12/8/2024</i> <i>df</i>
Konsultasi 2 <i>Ukuran Blok Diagram</i>	<i>[Signature]</i>	Konsultasi 2 <i>Ket gbr = 2 spes di Gbr gbr</i>	<i>df</i>
Konsultasi 3 <i>gambar pengalihan pada komposisi</i>	<i>[Signature]</i>	Konsultasi 3 <i>DF urutkan ke 3 dan di 1/2</i>	<i>df</i>
Konsultasi 4 <i>Rencana foto di akhir</i>	<i>[Signature]</i>	Konsultasi 4 <i>Ace 1/ proposal</i>	<i>df</i>
Konsultasi 5 <i>ok proposal</i>	<i>[Signature]</i>	Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa setiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna merah muda dan dicetak terbalik



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Marianto Kurniawan Yudi	Pembimbing I : Ir. A. Abd. Jabbar, MT.
NIM : 21718044	Pembimbing II : A. Imayani Pawitri ST., MT.
Judul Skripsi : Rancang Bangun Sistem Pengaman Kuda Sepeda Motor menggunakan Sdk Jai dan e-KTP berbasis Mikrokontroler	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1		Konsultasi 1	
		Tampilkan hse pengujian	25/8/2024 Aji
Konsultasi 2		Konsultasi 2	
		Abstrak (Hr 6/13, 14 Um, mehe, hse	Aji
Konsultasi 3		Konsultasi 3	
		Ase / hse	Aji
Konsultasi 4		Konsultasi 4	
		Ase / ruyug	30/8/24 Aji
Konsultasi 5		Konsultasi 5	
Ok hse	26/8-24		

Lanjut ke halaman sebelah.

Perhatian :

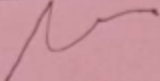
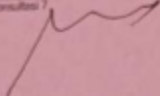
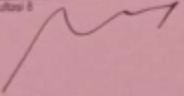
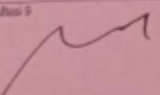
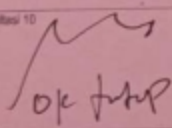
Mahasiswa wajib konsultasi minimal 5 kali

Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing

Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/jalan skripsi

Kartu ini dicetak di atas kertas karton A4 berwarna merah muda dan dicetak timbul balik

Lanjutan...

ARAHAN PEMBIMBING I	HARITOL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARITOL & PARAF PEMBIMBING
Konsultasi 6 		Konsultasi 6	
Konsultasi 7 		Konsultasi 7	
Konsultasi 8 		Konsultasi 8	
Konsultasi 9 		Konsultasi 9	
Konsultasi 10 		Konsultasi 10	

Parepare, 01 Agustus 2024Mengetahui
Ketua Program StudiAsrul, ST., MT.
NBM. 986 836

Mahasiswa

Marianito Kurniawan Yudi
NIM. 217180044

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa saat konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/jurnal skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna merah muda dan dicetak simetris