

LAMPIRAN

Berikut merupakan program dari arduino yang berisikan program pendaftaran sidik jari, mulai absensi sidik jari dan hapus sidik jari.

1. Program Pendaftaran Sidik Jari

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
uint8_t id;

void setup()
{
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Ditemukan sensor sidik jari!");
  } else {
    Serial.println("Tidak menemukan sensor sidik jari :(");
    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("Siap untuk mendaftarkan sidik jari!");
    Serial.println("Silakan ketik ID Pegawai # (dari 1 hingga 127) Anda ingin
menyimpan jari ini sebagai...");
    id = readnumber();
    if (id == 0) { // ID #0 not allowed, try again!
        return;
    }
    Serial.print("Mendaftar ID Pegawai #");
    Serial.println(id);
    while (! getFingerprintEnroll() );
}
uint8_t getFingerprintEnroll() {

    int p = -1;
    Serial.print("Menunggu jari yang valid untuk mendaftar #"); Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Gambar diambil");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Kesalahan komunikasi");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Kesalahan pencitraan");
                break;
            default:
                Serial.println("Kesalahan yang tidak diketahui");
                break;
        }
    }
}

```

```

    }
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Konversi gambar");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Gambar terlalu berantakan");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Kesalahan komunikasi");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Tidak dapat menemukan fitur sidik jari");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Tidak dapat menemukan fitur sidik jari");
    return p;
  default:
    Serial.println("Kesalahan yang tidak diketahui");
    return p;
}

Serial.println("Hapus jari");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Letakkan jari yang sama lagi");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
}

```

```

switch (p) {
case FINGERPRINT_OK:
    Serial.println("Gambar diambil");
    break;
case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Kesalahan komunikasi");
    break;
case FINGERPRINT_IMAGEFAIL:
    Serial.println("Kesalahan pencitraan");
    break;
default:
    Serial.println("Kesalahan yang tidak diketahui");
    break;
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Konversi gambar");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Gambar terlalu berantakan");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Kesalahan komunikasi");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Tidak dapat menemukan fitur sidik jari");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Tidak dapat menemukan fitur sidik jari");
    return p;
}

```

```

default:
    Serial.println("Kesalahan yang tidak diketahui");
    return p;
}

// OK converted!
Serial.print("Membuat model untuk #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Cetakan cocok!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Kesalahan komunikasi");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Sidik jari tidak cocok");
    return p;
} else {
    Serial.println("Kesalahan yang tidak diketahui");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Disimpan!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Kesalahan komunikasi");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Tidak dapat menyimpan di lokasi itu");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Kesalahan menulis ke flash");
    return p;
} else {
    Serial.println("Kesalahan yang tidak diketahui");
    return p;
}

```

```

}
}

```

2. Program Mulai Sidik Jari

```

#include <Adafruit_Fingerprint.h>
#include <DS3231.h> //mengincludekan library DS3231
DS3231 rtc(SDA, SCL); // inisialisasi penggunaan i2
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  rtc.begin();

  //setting pertama download program
  rtc.setDate(16, 7, 2018); //mensetting tanggal 07 april 2018
  rtc.setTime(6, 28, 00); //menseset jam 22:00:00
  rtc.setDOW(1); //menseset hari "Sabtu"
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Ditemukan sensor sidik jari!");
  } else {
    Serial.println("Tidak menemukan sensor sidik jari:");
    while (1) { delay(1); }
  }

  finger.getTemplateCount();

```

```

    Serial.print("Sensor contains "); Serial.print(finger.templateCount);
    Serial.println(" templates");
    Serial.println("Menunggu jari yang valid...");
}

void loop()                // run over and over again
{
    Serial.print(rtc.getDOWStr()); //prosedur pembacaan hari
    Serial.print(" ");

    Serial.print(rtc.getDateStr()); //prosedur pembacaan tanggal
    Serial.print(" -- ");

    Serial.println(rtc.getTimeStr()); //prosedur pembacaan waktu

    delay (1000); //waktu tunda 1 detik per cycle
    getFingerprintIDez();
    delay(50);      //don't ned to run this at full speed.
}

uint8_t getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Gambar diambil");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println("Tidak ada jari yang terdeteksi");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Kesalahan komunikasi");
            return p;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Kesalahan pencitraan");
            return p;
        default:
            Serial.println("Kesalahan yang tidak diketahui");
            return p;
    }
}

```

```

}

// OK success!

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Konversi gambar");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Gambar terlalu berantakan");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Kesalahan komunikasi");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Tidak dapat menemukan fitur sidik jari");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Tidak dapat menemukan fitur sidik jari");
    return p;
  default:
    Serial.println("Kesalahan yang tidak diketahui");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
  Serial.println("Menemukan pertandingan cetak!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  Serial.println("Kesalahan komunikasi");
  return p;
} else if (p == FINGERPRINT_NOTFOUND) {
  Serial.println("Tidak menemukan kecocokan");
  return p;
} else {
  Serial.println("Kesalahan yang tidak diketahui");
}

```



```

    return p;
}

// found a match!
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) return -1;

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

```

3. Program Hapus Sidik Jari

```

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
    Serial.begin(9600);
    while (!Serial); // For Yun/Leo/Micro/Zero/...

```

```

delay(100);
Serial.println("\n\nHapus Sidik Jari");

// set the data rate for the sensor serial port
finger.begin(57600);

if (finger.verifyPassword()) {
  Serial.println("Ditemukan sensor sidik jari!");
} else {
  Serial.println("Tidak menemukan sensor sidik jari :(");
  while (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("Silakan ketik ID Pegawai # (dari 1 hingga 127) yang ingin Anda
hapus...");
  uint8_t id = readnumber();
  if (id == 0) { // ID #0 not allowed, try again!
    return;
  }

  Serial.print("Hapus ID #");
  Serial.println(id);
  deleteFingerprint(id);
}

uint8_t deleteFingerprint(uint8_t id) {

```

```
uint8_t p = -1;

p = finger.deleteModel(id);

if (p == FINGERPRINT_OK) {
    Serial.println("Terhapus!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Kesalahan komunikasi");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Tidak dapat menghapus di lokasi itu");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Kesalahan menulis ke flash");
    return p;
} else {
    Serial.print("Unknown error: 0x"); Serial.println(p, HEX);
    return p;
}
}
```