

# **LAMPIRAN I**

No. Dok. : F-PBM-18

Tgl. Berlaku : 13 Desember 2010

No. Rev. : 00



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
**POLITEKNIK NEGERI SRIWIJAYA**

Jalan Sriwijaya Negara, Palembang 30139

Telp. 0711-353414 fax. 0711-355918

Website : [www.polisriwijaya.ac.id](http://www.polisriwijaya.ac.id) E-mail : [info@polsri.ac.id](mailto:info@polsri.ac.id)



**REKOMENDASI UJIAN LAPORAN AKHIR (LA)**

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama : Januar Riski  
NIM : 061630702198  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Judul Laporan Akhir : Rancang Bangun Pintu Menggunakan  
Sensor Sidik Jari Berbasis Mikrokontroler  
Arduino.

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Akhir (LA) pada Tahun Akademik 2018/2019.

Palembang, Juli 2019

Pembimbing I,

Pembimbing II,

Indano, S.T., M.Cs.  
NIP 197307062005011003

Ali Firdaus, S.Kom., M.Kom  
NIP 197010112001121001



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



**LEMBAR BIMBINGAN LAPORAN AKHIR**

Nama Mahasiswa : Januar Riski  
NIM : 061630701252  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Pembimbing I : Indarto, S.T., M.Cs.  
Judul Laporan Akhir : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino

No	Tanggal	Uraian	Paraf
1.	12 Juni 2019	Acc Judul & Revisi Bab I & II	
2.	14 Juni 2019	Acc Bab I dan II	
3.	18 Juni 2019	Revisi Bab III	
4.	20 Juni 2019	Acc Bab III	
5.	8 Juli 2019	Revisi Alat	
6.	9 Juli 2019	Acc Alat	
7.	10 Juli 2019	Bab IV dan V Revisi	
8.	12 Juli 2019	Acc Bab IV dan V	

Palembang, 2019

Mengetahui,  
Ketua Jurusan Teknik Komputer

Ir. A. Bahri Joni Malyan, M.Kom.  
NIP. 196007101991031001



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
POLITEKNIK NEGERI SRIWIJAYA  
Jalan Srijaya Negara, Palembang 30139  
Telp. 0711-353414 fax. 0711-355918  
Website : www.polsri.ac.id E-mail : info@polsri.ac.id



### LEMBAR BIMBINGAN LAPORAN AKHIR

Nama Mahasiswa : Januar Riski  
NIM : 061630701252  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Pembimbing II : Ali Firdaus, S.Kom., M.Kom.  
Judul Laporan Akhir : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino

No	Tanggal	Uraian	Paraf
1.	12 Juni 2019	Acc Judul dan Revisi Bab I & II	
2.	14 Juni 2019	Acc Bab I dan II	
3.	18 Juni 2019	Revisi Bab III	
4.	20 Juni 2019	Acc Bab III	
5.	8 Juli 2019	Revisi Alat	
6.	9 Juli 2019	Acc Alat	
7.	10 Juli 2019	Revisi bab IV dan V	
8.	12 Juli 2019	Acc Bab IV dan V	

Palembang, 2019  
Mengetahui,  
Ketua Jurusan Teknik Komputer

Ir. A. Bahri Joni Malyan, M.Kom.  
NIP. 196007101991031001



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Sriwijaya Negara, Palembang 30139


Telp. 0711-353414 fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



**REVISI LAPORAN AKHIR**

Ruang : 3  
Dosen Penilai : Ikhlasion Mekongga, S.T., M.Kom  
Nama Mahasiswa : Januar Riski  
NIM : 0616 3070 1252  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Judul Laporan KP : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino

No	Uraian Revisi	Paraf
		

Palembang, .....  
Dosen Penilai,

(.....)  
NIP.



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



**REVISI LAPORAN AKHIR**

Ruang : 3  
Dosen Penilai : Hartati Deviana  
Nama Mahasiswa : Januar Riski  
NIM : 0616 3070 1252  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Judul Laporan KP : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino

No	Uraian Revisi	Paraf
	Perbaiki bab III & IV	

Palembang, .....

Dosen Penilai,

(Hartati Deviana)

NIP.





KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Sriwijaya Negara, Palembang 30139

Telp. 0711-353414 fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



### REVISI LAPORAN AKHIR

Ruang : 3  
Dosen Penilai : Azwardi, S.T., M.T.  
Nama Mahasiswa : Januar Riski  
NIM : 0616 3070 1252  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Judul Laporan KP : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino

No	Uraian Revisi	Paraf
1	Rancangan melambatkan pulsi skala	
2	ok. 318 dibunt skala.	
3	sumbu.	
4	2.7 partur, gor.	
5	Kesimpulan diambil dari data percobaan.	

Palembang, .....  
Dosen Penilai,

(.....)

NIP.



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139  
Telp. 0711-353414 fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



REVISI LAPORAN AKHIR

Ruang : 3  
Dosen Penilai : Herlambang Suputra, S.Pd., M.Kom., Ph.D.  
Nama Mahasiswa : Januar Riski  
NIM : 0616 3070 1252  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Judul Laporan KP : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino

No	Uraian Revisi	Paraf
1.	Perbaiki Daftar Pustaka.	
2.	Tambahkan sumber yg belum ada	
3.	Tambahkan flowchart tentang daftar Sidik jari.	

Palembang, 17-7-2019

Dosen Penilai

*(Herlambang S.)*

NIP.





KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI  
**POLITEKNIK NEGERI SRIWIJAYA**  
Jalan Srijaya Negara, Palembang 30139  
Telp. 0711-353414 Fax. 0711-355918  
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



**PELAKSANAAN REVISI LAPORAN AKHIR (LA)**

Mahasiswa berikut,

Nama : Januar Riski  
NIM : 061630701252  
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer  
Judul Laporan Akhir : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino

Telah melaksanakan revisi terhadap Laporan Akhir (LA) yang diseminarkan pada hari Rabu tanggal 17 bulan Juli tahun 2019. Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penilai yang memberikan revisi:

No.	Komentar	Nama Dosen Penilai *)	Tanggal	Tanda Tangan
1.	OK.	Ikhthison Mekongga, S.T., M.Kom.	8/8 <sup>19</sup>	
2.	Acc	Azwardi, S.T., M.T.	29/7/19	
3.	Acc	Hartati Deviana, S.T., M.Kom.	5/8 <sup>19</sup>	
4.	Acc	Herlambang Saputra, S.Pd., M.Kom., Ph.D.	12/8-19	

Palembang, Juli 2019

Ketua Penilai \*\*)

Ikhthison Mekongga, S.T., M.Kom.  
NIP. 197705242000031002

**Catatan:**

\*) Dosen penilai yang memberikan revisi saat seminar laporan LA.

\*\*) Dosen penilai yang ditugaskan sebagai Ketua Penilai saat seminar LA.

Lembaran pelaksanaan revisi ini harus dilampirkan dalam Laporan Akhir (LA).



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI  
**POLITEKNIK NEGERI SRIWIJAYA**  
Jalan Sriwijaya Negara, Palembang 30139  
Telp. 0711-353414 Fax. 0711-355918  
Website : [www.polisriwijaya.ac.id](http://www.polisriwijaya.ac.id) E-mail : [info@polsri.ac.id](mailto:info@polsri.ac.id)



**LEMBAR PENGESAHAN LAPORAN AKHIR (LA)**

Ruang : 3 (Tiga)  
Nama : Januar Riski  
NIM : 061630701252  
Jurusan/Program Studi : Teknik Komputer  
Judul Laporan Akhir : Rancang Bangun Pintu Menggunakan Sensor Sidik Jari  
Berbasis Mikrokontroler Arduino

No	Nama Penguji	Revisi	Ket	Paraf Yang Mengesahkan	
				Pembimbing 1	Pembimbing 2
1.	Ikhthison Mekongga, S.T., M.Kom	-	Acc		
2.	Azwardi, S.T., M.T.	1. Rancangan mekanik pakai skala 2. Gambar 3.18 dibuat skala 3. Perbarui gambar 2.7 4. Kesimpulan diambil dari data percobaan	Acc		
3.	Hartati Deviana, S.T., M.Kom	Perbaiki bab III dan IV	Acc		
4.	Herlambang Saputra, S.Pd., M.Kom., Ph.D.	1. Perbaiki daftar pustaka 2. Tambahkan sumber yang belum ada 3. Tambahkan flowchart tentang daftar sidik jari	Acc		

Palembang, Juli 2019  
Mengetahui,  
Ketua Jurusan Teknik Komputer

Ir. A. Bahri Joni Malyan, M.Kom  
NIP. 196007101991031001

# **LAMPIRAN II**

## Listing Program

### program.ino

```
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>

SoftwareSerial mySerial(2, 3);
SoftwareSerial serdata(4, 5);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&serdata);
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3F, 16, 2);

char state;
uint8_t id;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  mySerial.begin(9600);
  Wire.begin();
  lcd.begin();
  // Turn on the backlight and print a message.
  lcd.backlight();

  pinMode(8, OUTPUT);
  pinMode(12, INPUT_PULLUP);

  // set the data rate for the sensor serial port
  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) {
      delay(1);
    }
  }
  finger.getTemplateCount();
  Serial.print("Sensor contains ");
  Serial.print(finger.templateCount); Serial.println(" templates");
  Serial.println("Waiting for valid finger...");

  digitalWrite(8, HIGH);
```

```

    lcd.home (); // go home on LCD
    lcd.setCursor (0, 0);
    lcd.print("Door Lock System");
    lcd.setCursor (0, 1);
    lcd.print("==ruang juviter==");
    delay(1000);
    // lcd.clear();

}

void loop() {
    // put your main code here, to run repeatedly:
    lcd.home (); // go home on LCD
    lcd.setCursor (0, 0);
    lcd.print("Door Lock System");
    lcd.setCursor (0, 1);
    lcd.print("==ruang juviter==");

    getFingerprintID();

    if (Serial.available() > 0 ) {
        state = Serial.read();
        Serial.println(state);
    }
    Serial.println(analogRead(A0));
    if ( state == 'A' || digitalRead(12) == LOW || analogRead(A0) <=
20)
    {
        lcd.clear();
        lcd.setCursor (0, 0);
        lcd.print("Door Lock System");
        lcd.setCursor (0, 1);
        lcd.print("Door Open");
        digitalWrite(8, LOW);
        delay(2000);
        lcd.clear();
        lcd.print("Door Lock System");
        lcd.setCursor (0, 1);
        lcd.print("Door Close");
        digitalWrite(8, HIGH);
    }
}

```

**finger.ino**



```

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");
      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.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Door Lock System");
  }
}

```

```

    lcd.setCursor (0, 1);
    lcd.print("Door Open");
    digitalWrite(8, LOW);
    delay(2000);
    lcd.clear();
    lcd.print("Door Lock System");
    lcd.setCursor (0, 1);
    lcd.print("Door Close");
    digitalWrite(8, HIGH);
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    lcd.clear();
    lcd.print("tidak ada data");
    delay(1000);
    lcd.clear();
    return p;
} else {
    Serial.println("Unknown error");
    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;
}

uint8_t readnumber(void) {
    uint8_t num = 0;

    while (num == 0) {
        while (! Serial.available());
        num = Serial.parseInt();
    }
    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(".");
        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");
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");

```

```

while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            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;
}
}
delay(2000);
id = 0;
}

```

### **enroll.ino**

```

#include <Adafruit_Fingerprint.h>
SoftwareSerial mySerial(4, 5);
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("Found fingerprint sensor!");
} else {
  Serial.println("Did not find fingerprint sensor :(");
  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("Ready to enroll a fingerprint!");
  Serial.println("Please type in the ID # (from 1 to 127) you want
to save this finger as...");
  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:
        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");
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");
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            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;
}
}

```

#### **delete.ino**

```

#include <Adafruit_Fingerprint.h>

// On Leonardo/Micro or others with hardware serial, use those! #0
// is green wire, #1 is white
// uncomment this line:
// #define mySerial Serial1

// For UNO and others without hardware serial, we must use
// software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
// comment these two lines if using hardware serial
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\nDelete Finger");

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

```

```

    if (finger.verifyPassword()) {
        Serial.println("Found fingerprint sensor!");
    } else {
        Serial.println("Did not find fingerprint sensor :(");
        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("Please type in the ID # (from 1 to 127) you want
to delete...");
    uint8_t id = readnumber();
    if (id == 0) { // ID #0 not allowed, try again!
        return;
    }

    Serial.print("Deleting 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("Deleted!");
    } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
        Serial.println("Communication error");
        return p;
    } else if (p == FINGERPRINT_BADLOCATION) {
        Serial.println("Could not delete in that location");
        return p;
    } else if (p == FINGERPRINT_FLASHERR) {
        Serial.println("Error writing to flash");
        return p;
    } else {

```



```
Serial.print("Unknown error: 0x"); Serial.println(p, HEX);  
return p;  
}  
}
```