

# **PENGAMAN MOTOR MENGGUNAKAN SIDIK JARI**

## **PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan  
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Fatur Rivando                      NIRM : 0031510

Muhamad Reza Nafianto   NIRM : 0031519

**POLITEKNIK MANUFAKTUR NEGERI**  
**BANGKA BELITUNG**  
**TAHUN 2018**

## **LEMBAR PENGESAHAN**

### **PENGAMAN MOTOR MENGGUNAKAN SIDIK JARI**

Oleh :

Fatur Rivando / 0031510

Muhamad Reza Nafianto / 0031519

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



Ocsirendi, S.ST, M.T

Pembimbing 2

I Made Andik Setiawan, M.Eng., Ph.D

Penguji 1



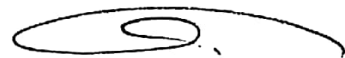
Dr. Parulian Silalahi, M.Pd.

Penguji 2



M. Iqbal Nugraha, M.Eng.

Penguji 3



Aan Febriansyah, M.T.

## **PERNYATAAN BUKAN PLAGIAT**

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Fatur Rivando

NIRM : 0031510

Nama Mahasiswa 2 : Muhamad Reza Nafianto

NIRM : 0031519

Dengan judul : Pengaman Motor Menggunakan Sidik Jari

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari nyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 20 Maret 2018

Nama Mahasiswa

Tanda Tangan

1. Fatur Rivando



.....

2. Muhamad Reza Nafianto



.....

## ABSTRAK

*Pencurian kendaraan masih marak terjadi, khususnya pada sepeda motor. Sebagian besar pencurian itu terjadi dengan cara membakar paksa kunci kontaknya. Hal ini dikarenakan sistem pengaman dari kunci kontak sepeda motor belum sepenuhnya baik sehingga perlu adanya sistem pengamanan tambahan selain pengaman pada kunci kontak. Oleh karena itu dirancang sistem pengaman motor menggunakan sidik jari yang secara teknis terhubung ke kabel kelistrikan motor sehingga menjadikannya lebih aman dari pencurian. Sistem ini menggunakan sensor sidik jari sebagai sinyal input, Arduino Mega 2560 sebagai pusat kontrolnya dan outputnya berupa relay secara teknis dapat memutus dan menyambungkan kabel kontak kelistrikan sesuai dengan perintah yang diterima dari sensor sidik jarinya. Sistem pengaman ini memiliki kelebihan yakni mudah dikendalikan, karena untuk memutus dan menyambungkan kabel kelistrikan motor dengan menempalkan sidik jari yang telah terdaftar ke sensor sidik jarinya. Pengaman motor menggunakan sidik jari ini, dilengkapi dengan menu pendaftaran dan penghapusan user serta admin. Selain mudah dikendalikan, sistem pengaman ini juga sulit untuk dilumpuhkan karena untuk melumpuhkannya harus dengan membakar kabel kelistrikan motornya. Walaupun kunci kontaknya telah dilumpuhkan oleh pencuri sepeda motor, akan tetap tidak dapat di jalankan apabila sidik jari terdaftar belum ditempelkan ke sensor sidik jarinya, karena kabel kelistrikan motor masih dalam keadaan terputus.*

*Kata Kunci: Pencurian, Sepeda Motor, Kabel Kontak Kelistrikan, Kabel Coil, Sensor Sidi Jari, Arduino Mega 2560.*

## ABSTRACT

*Vehicle theft is still rampant, especially on motorbikes. Most of these thefts occur by forcing the ignition key. This is because the safety system of the motorbike contact washers is not fully good so there needs to be an additional security system in addition to the safety on the ignition key. Therefore, a motorized safety system is designed using fingerprints that are technically connected to the motor's electrical cable so as to make it safer than theft. This system uses a fingerprint sensor as an input signal, Arduino Mega 2560 as the control center and the output in the form of a relay can technically disconnect and connect the electrical contact cable according to the order received from the fingerprint sensor. This security system has the advantage of being easy to control, because to break and connect the motor electrical cable by marking the fingerprint that has been registered to the fingerprint sensor. Motorcycle security uses this fingerprint, equipped with a registration menu and deletion of users and admin. In addition to being easy to control, this safety system is also difficult to disable because to disable it, it must be grounded in the motor's electrical cable. Even though the ignition key has been disabled by a motorcycle thief, it will still not be able to run if the registered fingerprint has not been attached to the fingerprint sensor, because the motor power cable is still disconnected.*

Keywords: Theft, Motorcycle, Electrical Contact Cable, Coil Cable, Sensor Sidi Jari, Arduino Mega 2560.

## KATA PENGANTAR

Assalamualaikum wa rahmatullahi wabarakatuh.

Puji dan syukur penulis ucapkan kepada Allah SWT, karena berkat rahmat dan ridho-Nya penulis dapat menyelesaikan makalah proyek akhir ini tepat pada waktunya. Proyek akhir “Pengaman Motor Menggunakan Sidik Jari” merupakan salah satu syarat setiap kelompok kerja proyek akhir untuk memenuhi persyaratan pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung .

Dalam menyelesaikan makalah proyek akhir ini, penulis mendapat bimbingan dan dorongan dari berbagai pihak, baik secara dukungan maupun material. Untuk itu, pada kesempatan ini dengan segala kerendahan dan ketulusan hati penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Keluarga tercinta, khususnya ibunda tercinta yang selalu sabar membimbing, mendoakan, dan memberikan memotivasi serta menasehati penulis. Terima kasih untuk setiap peluh keringat serta kasih sayang yang tidak ternilai
2. Bapak Sugeng Ariyono, M.Eng. , Ph.D ,selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung
3. Bapak Ocsirendi, S.ST, M.T selaku dosen pembimbing 1 yang selama ini banyak membantu memberi masukan selama menjadi pembimbing kami
4. Bapak I Made Andik Setiawan, M.Eng. selaku dosen pembimbing 2 yang selama ini banyak membantu member masukan selama menjadi pembimbing kami
5. Bapak Eko Sulistyio, M.T. , selaku Kepala Prodi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung
6. Segenap dosen dan staf pengajar Politeknik Manufaktur Negeri Bangka Belitung
7. Rekan tim proyek akhir dan teman – teman seangkatan

8. Pihak – pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan tugas akhir ini yang tidak dapat disebutkan satu persatu

Penulis menyadari bahwa penulisan makalah proyek akhir ini masih terdapat banyak kekurangan baik dari segi penulisan maupun isi makalah. Oleh karena itu, penulis mengharapkan segala petunjuk, kritik, dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya. Akhir kata, penulis mohon maaf atas kekurangan dalam penulisan makalah ini dan penulis dengan senang hati menerima saran dan kritik yang membangun dari pembaca.

Semoga makalah ini dapat berguna untuk menambah wawasan dan wacana bagi rekan-rekan mahasiswa.

Walaikumsalam warahmarullahi wabarakatuh.

Sungailiat, 12 Mei 2018

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	ii
<b>PERNYATAAN BUKAN PLAGIAT</b> .....	iii
<b>ABSTRAK</b> .....	iv
<b>KATA PENGANTAR</b> .....	vi
<b>DAFTAR ISI</b> .....	viii
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR LAMPIRAN</b> .....	xii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan dan Batasan Masalah.....	3
1.3 Tujuan Proyek Akhir.....	3
<b>BAB II DASAR TEORI</b>	
2.1 Pengaman Sepeda Motor.....	5
2.2 Sidik Jari .....	6
2.2.1 Klarifikasi Pola Sidik Jari .....	7
2.3 Sepeda Motor .....	8
2.4 <i>Fingerprint</i> Sensor Adafruit.....	9
2.4 Spesifikasi <i>Fingerprint</i> Adafruit.....	10
<b>BAB III METODE PELAKSANAAN</b>	
3.1 Pengumpulan Data .....	12
3.2 Perancangan <i>Hardware</i> dan <i>Software</i> .....	12
3.3 Uji Coba .....	13



3.4 Pembuatan Laporan.....	13
----------------------------	----

## **BAB IV PEMBAHASAN**

4.1 Diagram Block .....	14
4.2 <i>Hardware</i> Pengaman Motor.....	14
4.2.1 <i>Finger Print Sensor</i> .....	15
4.2.1.1 Pengujian <i>Finger Print Sensor</i> .....	16
4.2.2 <i>Keypad 4 x 4 Matrix</i> Arduino .....	20
4.2.2.1 Pengujian <i>Keypad 4 x 4 Matrix</i> Arduino .....	21
4.2.3 LCD 16 x 2 .....	22
4.2.3.1 Pengujian LCD 16 x 2.....	23
4.2.4 Modul <i>Relay 4 Channel</i> .....	24
4.2.4.1 Pengujian Modul <i>Relay 4 Channel</i> .....	24
4.2.5 Modul <i>Stepdown</i> LM 2596 .....	25
4.2.5.1 Pengujian Modul <i>Stepdown</i> LM 2596 .....	25
4.2.6. Aki.....	26
4.3 <i>Software</i> Pengaman Motor.....	27
4.3.1 Perancangan Pembuatan Program Arduino Mega 2560 .....	27
4.3.1.1 Pembuatan Program Arduino Mega 2560.....	27
4.4 Peletakan Tempat <i>Box</i> Kontrol dan Tampilan LCD Serta Keypad.....	27
4.5 Pengujian Pengaman Motor Menggunakan Sidik Jari.....	30

## **BAB V PENUTUP**

5.1 Kesimpulan .....	36
5.2 Saran.....	36

<b>DAFTAR PUSTAKA</b> .....	37
-----------------------------	----

<b>LAMPIRAN</b> .....	38
-----------------------	----

## DAFTAR TABEL

Tabel	Halaman
2.1 Spesifikasi <i>Fingerprint</i> Adafruit .....	8
2.2 Spesifikasi Arduino Mega 2560.....	10
2.3 Konfigurasi Pin Arduino.....	11
4.1 Hasil Pendaftaran Sidik Jari .....	29
4.2 Hasil Penghapusan <i>Fingerprint</i> Sensor.....	30
4.3 Pendaftaran Sidik Jari .....	41
4.4 Pengujian untuk Menghidupkan Kelistrikan Motor.....	41
4.5 Pengujian Sidik Jari Pengguna A untuk Tangan Kanan .....	41
4.6 Pengujian Sidik Jari Pengguna A untuk Tangan Kiri .....	42
4.7 Pengujian Sidik Jari Pengguna B untuk Tangan Kanan .....	42
4.8 Pengujian Sidik Jari Pengguna B untuk Tangan Kiri .....	42
4.9 Pengujian Sidik Jari Pengguna C untuk Tangan Kanan .....	42
4.10 Pengujian Sidik Jari Pengguna C untuk Tangan Kiri.....	42
4.11 Pengujian Sidik Jari Pengguna D untuk Tangan Kanan .....	43
4.12 Pengujian Sidik Jari Pengguna D untuk Tangan Kiri .....	43
4.13 Pengujian Sidik Jari Pengguna E untuk Tangan Kanan.....	43
4.14 Pengujian Sidik Jari Pengguna E untuk Tangan Kiri.....	43
4.15 Pengujian Sidik Jari Pengguna F untuk Tangan Kanan .....	43
4.16 Pengujian Sidik Jari Pengguna F untuk Tangan Kiri .....	44
4.17 Pengujian Sidik Jari Pengguna G untuk Tangan Kanan .....	44
4.18 Pengujian Sidik Jari Pengguna G untuk Tangan Kiri .....	44
4.19 Pengujian Sidik Jari Pengguna H untuk Tangan Kanan .....	44
4.20 Pengujian Sidik Jari Pengguna H untuk Tangan Kiri .....	44
4.21 Pengujian Sidik Jari Pengguna I untuk Tangan Kanan .....	45
4.22 Pengujian Sidik Jari Pengguna I untuk Tangan Kiri.....	45
4.23 Pengujian Sidik Jari Pengguna J untuk Tangan Kanan.....	45
4.24 Pengujian Sidik Jari Pengguna J untuk Tangan Kiri.....	45

## DAFTAR GAMBAR

Gambar	Halaman
2.1 Pola sidik jari (a) <i>Whorl/Swirl</i> , (b) <i>Loop</i> , (c) <i>Arch</i> .....	7
2.2 Pola-Pola Sidik Jari .....	7
2.3 <i>Coil Motor</i> .....	9
2.4 <i>Fingerprint</i> .....	9
3.1 <i>Flowchart</i> Metode Pelaksanaan .....	11
4.1 Blok Diagram Pengaman Motor Menggunakan Sidik Jari .....	14
4.2 desain peletakan alat pada motor .....	15
4.3 Pemasangan Pin <i>Fingerprint Sensor</i> .....	25
4.4 Hasil Pendaftaran Sidik Jari .....	19
4.5 Hasil Penghapusan <i>Fingerprint Sensor</i> .....	20
4.6 Pemasangan Pin keypad 4 x 4 ke Arduino Mega 2560 .....	21
4.7 Tampilan input char keypad 4 x 4 ke LCD 16 x 2 .....	22
4.8 Pemasangan Pin pada Lcd 16 x 2 .....	22
4.9 Tampilan Menu Pada Lcd 16 x 2 .....	23
4.10 Pemasangan Pin Pada Modul Relay 4 Channel .....	24
4.11 Modul <i>Stepdown</i> LM 2596 .....	25
4.12 Tegangan Sumber .....	25
4.13 Tegangan Setelah Diturunkan .....	26
4.14 Aki (Akumulator) .....	26
4.15 Perakitan Komponen Didalam Box .....	28
4.16 Peletakan Panel Kontrol .....	29
4.17 Setelah dirapikan .....	29
4.18 Pemasangan sensor .....	29

# **BAB I**

## **PENDAHULUAN**

### 1.1 Latar Belakang Masalah

Kendaraan motor beroda dua adalah alat transportasi utama masyarakat Indonesia. Berdasarkan laporan data dari Korps Lalu Lintas (Kakorlantas) Kepolisian Republik Indonesia, jumlah kendaraan bermotor di Indonesia pada tahun 2013 adalah 104.211 juta unit, dan 86.253 juta atau sekitar 82,27% dari jumlah tersebut adalah kendaraan sepeda motor [1]. Pada tahun 2015, jumlah mobil penumpang adalah 13.480.973 unit, mobil bis sejumlah 2.420.917 unit, mobil barang berjumlah 6.611.028 unit, dan jumlah sepeda motor mencapai 98.881.267 unit [2]. Selanjutnya, pada tahun 2016, jumlah sepeda motor di Indonesia mencapai 105.150.082 unit [3]. Dari data tersebut, dapat disimpulkan bahwa setiap tahun jumlah pengguna motor terus meningkat.

Di antara jenis kendaraan motor, jenis kendaraan sepeda motor banyak ditemukan di wilayah perkotaan maupun pedesaan di Indonesia. Selain harganya yang relatif terjangkau, kendaraan sepeda motor beroda dua memberikan banyak keuntungan bagi penggunanya, antara lain biaya operasional lebih murah, dan bisa menempuh medan jalan yang tidak bisa atau sulit dilalui kendaraan beroda empat. Kendaraan sepeda motor beroda dua juga memudahkan mobilisasi perjalanan para penggunanya, khususnya di kota-kota besar, padat dan sibuk di Indonesia dikarenakan jenis kendaraan ini membuat perjalanan para penggunanya lebih cepat sampai pada tempat tujuan.

Ironisnya, kendaraan sepeda motor beroda dua rentan mengalami pencurian. Biro Pusat Statistik (BPS) Indonesia mencatat jumlah sepeda motor yang hilang di Indonesia mencapai 25 unit dalam satu hari, dan jumlah kasus sepeda motor hilang pun meningkat [4]. Pada tahun 2009-2011, jumlah sepeda motor yang hilang meningkat dari 34.77 menjadi 39.27 atau sekitar 13% [5]. Selanjutnya, pada tahun 2015, berdasarkan laporan Polresta Barelang, Batam, terhitung dari Januari sampai Juni 2015, kasus pencurian motor mencapai 373 perkara [6]. Di Pulau Bangka, kasus pencurian sepeda motor beroda dua juga

sering terjadi. Sebagai contoh, di Kota Belinyu, sudah banyak sepeda motor di parkir digondol maling secara tiba-tiba [7].

Berbagai faktor yang menyebabkan kasus pencurian sepeda motor. Salah satunya adalah dikarenakan kelemahan sistem pengaman, bahkan ketiadaan sistem dan alat pengaman efektif yang terpasang pada sepeda motor. Pada umumnya, alat yang dipasang untuk menjaga keamanan sepeda motor terdiri dari kunci kontak variasi, kunci gembok dan kunci setang. Alat pengaman seperti yang disebutkan belum tentu menjamin keamanan sepeda motor, disebabkan para pencuri memiliki banyak cara untuk membobol sistem dan alat pengaman yang terpasang pada sepeda motor, misalnya dengan membuat kunci duplikat [4]. Dikarenakan hal tersebut, diperlukan sistem keamanan sepeda motor yang canggih untuk mengatasi permasalahan tersebut.

Sistem keamanan kendaraan bermotor roda dua terus berkembang. Sebagai contoh, sistem keamanan sepeda motor menggunakan Sriwijaya Motorcycle Security, dengan aplikasi password berbasis Mikrokontroler ATMEGA853 sudah rancang dengan baik. Bahwa Kelebihan dari Sriwijaya Motorcycle Security (SMS) ini selain cara penggunaannya *simple* dan *easy to use*, alat pengaman tambahan ini juga sangat ekonomis sehingga dapat menjangkau berbagai lapisan masyarakat, *password* dan jumlah digitnya bisa disesuaikan dengan keinginan pengguna serta efektif memperkecil kemungkinan terjadinya tindak pencurian sepeda motor sehingga memberikan rasa aman kepada pemilik kendaraan [8]. Namun, kelemahan dari sistem ini adalah jika pemilik keliru memasukkan pasword, lampu led dan klakson akan menyala. Tentunya ini akan menimbulkan kegaduhan apabila sang pemilik terus mencoba memasukkan password yang ia lupa.

Seiring dengan perkembangan aplikasi teknologi sekarang ini, sistem pengaman sepeda motor yang ada, tidak bisa mengatasi jumlah kehilangan sepeda motor. Salah satu solusi untuk mengatasi permasalahan tersebut adalah dengan penerapan teknologi sistem keamanan cerdas (*intelligent security system*) [9]. Sebagaimana diketahui, sistem keamanan cerdas, seperti menggunakan sidik jari, menjadi kebutuhan di segala aspek kehidupan manusia modern. Sensor sidik jari

yang berjudul rancang bangun sistem keamanan motor dengan pengenalan sidik jari adalah salah satu perkembangan teknologi yang memiliki keamanan yang cukup tinggi di mana hanya bisa diakses oleh orang yang sidik jarinya sudah di daftar kedalam sensor sidik jari [10]. Dilatarbelakangi hal tersebut, penulis berkeinginan untuk membuat sistem keamanan kendaraan beroda dua dengan menggunakan sidik jari sebagai pengaman kelistrikan motor dan selenoid sebagai pengunci setang sepeda motor, serta dilengkapi dengan menu yang dimiliki hanya 1 admin yaitu menu daftar, hapus admin dan user.

## **1.2 Perumusan Masalah**

### **1.2.1 Rumusan Masalah**

Adapun rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

- (1) Bagaimana membuat alat yang dapat menambah keamanan sepeda motor dengan menggunakan sidik jari.
- (2) Bagaimana cara menghubungkan elektrikal dari motor ke elektrikal arduino sebagai kontrol untuk pengamanan motor menggunakan sensor sidik jari.

### **1.2.2 Batasan Masalah**

Adapun batasan masalah dari tugas akhir kami adalah sebagai berikut :

- (1) Pendaftaran sidik jari tambahan hanya bisa menambahkan 5 sidik jari saja.
- (2) Pendaftaran admin dianjurkan pada keypad bernomor 0, karena keypad bernomor 1-8 digunakan untuk user.
- (3) Apabila aki dalam keadaan soak/habis daya-nya maka motor tidak bisa diakses.
- (4) Pendaftaran admin hanya bisa untuk 1 orang saja.

## **1.3 Tujuan**

Dengan mengacu latar belakang di atas, penelitian ini bertujuan untuk :

- (1) Membuat alat yang dapat menambah keamanan sepeda motor dengan menggunakan sidik jari.
- (2) Membuat modul sidik jari untuk menghubungkan elektrik dari motor ke elektrik arduino sebagai kontrol untuk pengamanan motor menggunakan sensor sidik jari.

## **BAB II**

### **DASAR TEORI**

#### **2.1 Pengaman Sepeda Motor**

Alat pengaman sepeda motor adalah peralatan yang berfungsi mengamankan sepeda motor dari pencurian ketika tidak ada pengawasan dari pemiliknya. Alat pengaman motor terbagi menjadi dua jenis, yang pertama yaitu pengaman sepeda motor elektrik dan yang kedua non elektrik. Contoh pengaman motor elektrik:

1. Sistem Keamanan Ganda Pada Sepeda Motor Untuk Pencegahan Pencurian dengan Smarty (*Smart Security*), alat ini dapat dikontrol dari jarak dekat maupun jauh dikarenakan menggunakan sistem SMS dan GPS satelit [11].
2. Sistem Keamanan Kendaraan Sepeda Motor Menggunakan Sriwijaya Motorcycle Security, dengan aplikasi password berbasis mikrokontroler ATMEGA853 sudah rancang baik [8].
3. Rancang Bangun Sistem Keamanan Motor Dengan Pengenalan Sidik Jari [10].
4. Perancangan Sistem Pengaman Sepeda Motor Menggunakan Mikrokontroler Raspberry Pi dan Smartphone Android [6].
5. Sistem Keamanan Kendaraan Suzuki Smash Menggunakan ATMEGA 8 Dengan Sensor Bluetooth hc-6 Berbasis Android [5].

contoh pengaman motor non elektrik:

1. Kunci pengaman motor menggunakan cakram gembok digunakan pada motor yang memiliki cakram.
2. Kunci pengaman motor menggunakan rantai yaitu dengan mengaitkan rantai ke tiang.
3. Kunci pengaman motor menggunakan alarm ini akan berbunyi jika motor di curi [12].



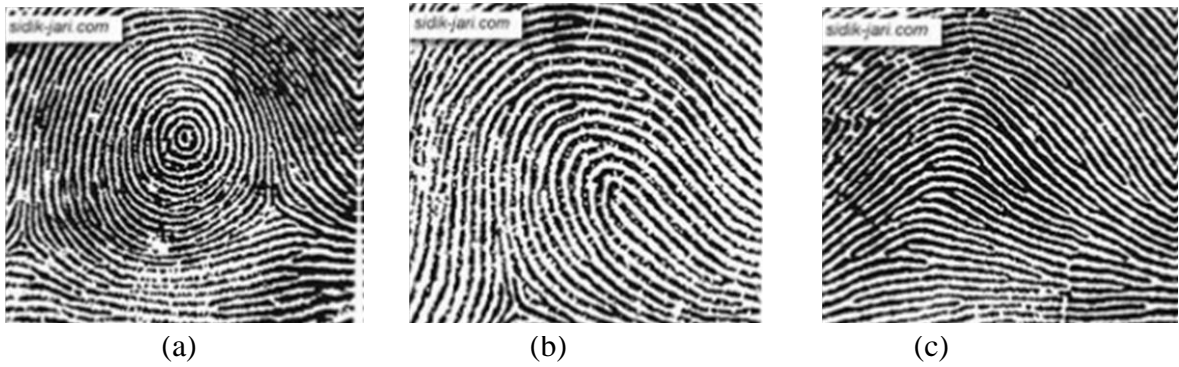
## 2.2 Sidik Jari

Sidik jari adalah hasil reproduksi tapak jari baik yang sengaja diambil, dicapkan dengan tinta, maupun bekas yang ditinggalkan pada benda karena tersentuh kulit telapak tangan [13]. Salah satu teknik yang sering digunakan dalam autentifikasi biometrik adalah dengan pengenalan sidik jari. Banyak negara yang telah menggunakan metode indentifikasi biometrik sidik jari, seperti Amerika Serikat, India dan Indonesia [14]. Penggunaan metode indentifikasi biometrik dilakukan untuk menghindari pemalsuan identitas seseorang. Teknologi biometrik mampu mengenali manusia melalui sidik jari, dan ini dapat diterapkan di pada berbagai kebutuhan. Salah satunya di sektor pengamanan rumah, kendaraan roda empat, brangkas dokumen suatu perusahaan dan daftar kehadiran para pegawai di perusahaan. Teknologi sidik jari mampu menggeser kata sandi (*password*) dikarenakan penerapan *password* terkadang membuat seseorang lupa dan bisa membuat kegaduhan ketika *password* salah.

Biometrik mempunyai keunggulan sifat, yaitu tidak bisa dihilangkan, dilupakan atau dipindahkan dari satu ke orang lain. Sifat-sifat yang dimiliki oleh sidik jari, antara lain :

1. *Perennial nature*, yaitu guratan-guratan pada sidik jari yang melekat pada kulit manusia seumur hidup.
2. *Immutability*, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapatkan kecelakaan yang serius.
3. *Individuality*, pola sidik jari adalah unik dan berbeda untuk setiap orang [15].

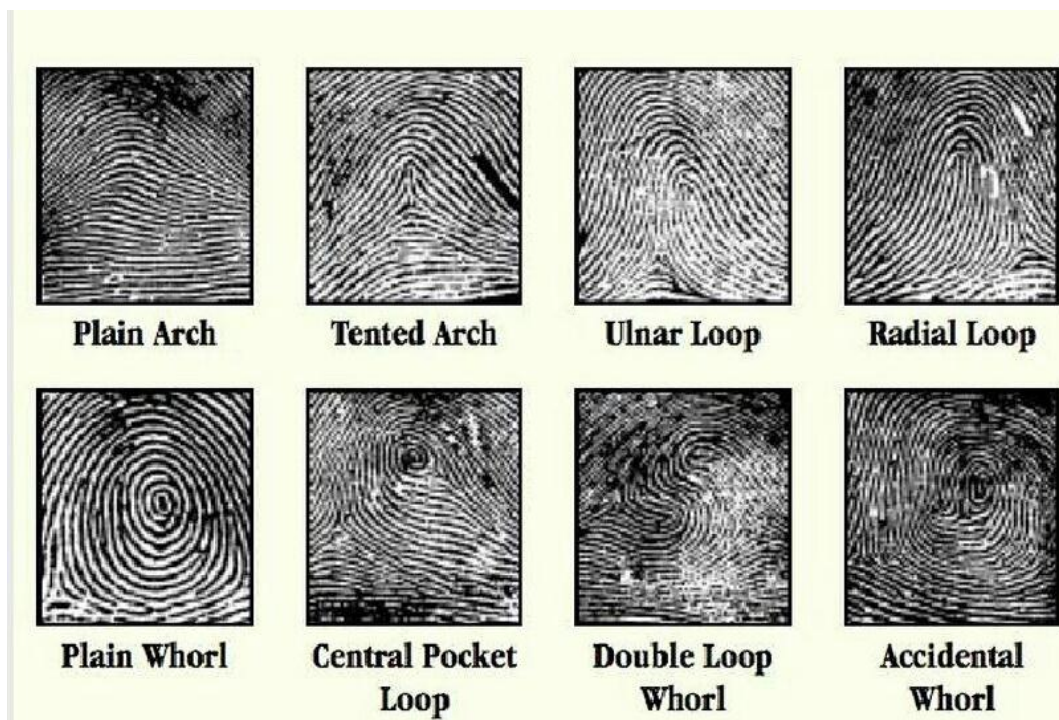
Selain itu, sidik jari juga mempunyai pola-pola sehingga bisa mencirikan seseorang. Ada 3 pola dasar dermatoglyphic tentang sidik jari yang perlu diketahui, yakni [10]:



Gambar 2.1 Pola sidik jari (a) *Whorl/Swirl*  
(b) *Loop*, (c) *Arch* [10]

### 2.2.1 Klarifikasi Pola Sidik Jari

Selain itu sidik jari dibagi menjadi beberapa kelas, ada 6 citra sidik jari dari beberapa kelas [15].



Gambar 2.2 Pola-Pola Sidik Jari [15]

1. *Plain Arch* adalah bentuk pokok sidik jari dimana garis-garis datang dari sisi lukisan yang satu mengalir ke arah sisi yang lain, dengan sedikit bergelombang naik ditengah.

2. *Tented arch* (Tiang Busur) adalah bentuk pokok sidik jari yang memiliki garis tegak atau sudut atau dua atau tiga ketentuan sangkutan.
3. *Ulnar loop* adalah garisnya memasuki pokok lukisan dari sisi yang searah dengan kelingking, melengkung ditengah pokok lukisan dan kembali atau cenderung kembali ke arah sisi semula.
4. *Radial loop* adalah garisnya memasuki pokok lukisan dari sisi yang searah dengan jempol, melengkung di tengah pokok lukisan dan kembali atau cenderung kembali ke arah sisi semula.
5. *Plain Whorl* (Lingkar) adalah bentuk pokok sidik jari, mempunyai dua delta dan sedikitnya satu garis melingkar di dalam pola area, berjalan didepan kedua delta.
6. *Central Pocket Loop* adalah bentuk yang terdiri dari satu atau lebih kurva ridge dan dua titik delta.
7. *Double loop* (Sangkutan Kembar) adalah mempunyai dua delta dan dua garis melingkar di dalam pola area, berjalan didepan kedua delta.
8. *Accidental* adalah pola yang mempunyai dua titik delta satu delta akan terhubung keatas, dan delta yang lain terhubung dengan yang lain.

### **2.3 Sepeda Motor**

Sepeda motor adalah kendaraan beroda dua yang digerakan oleh sebuah mesin [16]. Didalam sepeda motor ada yang dinamakan pengapian motor atau kelistrikan motor. Pada kelistrikan motor ada komponen yang berfungsi sebagai pengapian contohnya :

1. Aki/*Accumulator* berfungsi meyalakan sistem starter untuk menghidupkan mesin.
2. Busi berfungsi untuk memercikan bunga api diruang bakar pada akhir langkah kompresi busi pada saat menerima tegangan tinggi.
3. CDI (*Capacitor Discharge Ignition*) berfungsi mengatur waktu untuk meletikan api pada busi yang dibesarkan oleh coil untuk memicu pembakaran pada ruang bakar mesin.

4. *Coil* berfungsi menaikkan tegangan listrik dari aki 12 volt menjadi ribuan volt, arus listrik yang besar ini disalurkan ke busi sehingga busi mampu meletikan pijaran bunga api.
5. *Regulator Rectifier* (RR) berfungsi sebagai penstabil arus dan tegangan yang masuk pada aki agar aki bisa dicas.
6. Spul berfungsi sebagai pehasil listrik berupa tegangan AC (*Alternating Current*) [17].

Ditugas akhir yang berjudul sistem pengaman motor menggunakan sidik jari, sistem yang diamankan adalah kelistrikan motor letaknya di kabel *coil* motor. Karena komponen *coil* sangat berpengaruh besar dalam pengapian busi. Adapun bentuk fisik dari *coil* bisa dilihat pada Gambar 2.3.



Gambar 2.3 *Coil* Motor

#### **2.4 *Fingerprint* Sensor Adafuit**

*Fingerprint* sensor adalah sebuah perangkat elektronik yang digunakan untuk menangkap gambar digital dari pola sidik jari. Dimana pedeteksi dilakukan dengan pembacaan kontur (tinggi rendahnya permukaan) pola sidik jari. *Fingerprint* sensor ini sendiri akan membuat penambahan deteksi sidik jari dan vertifikasinya lebih sederhana.



Gambar 2.4 *fingerprint*

#### 2.4.1 Spesifikasi *Fingerprint* Adafruit

Spesifikasi dari *Fingerprint* adafruit dapat dilihat pada Tabel 2.1 berikut.

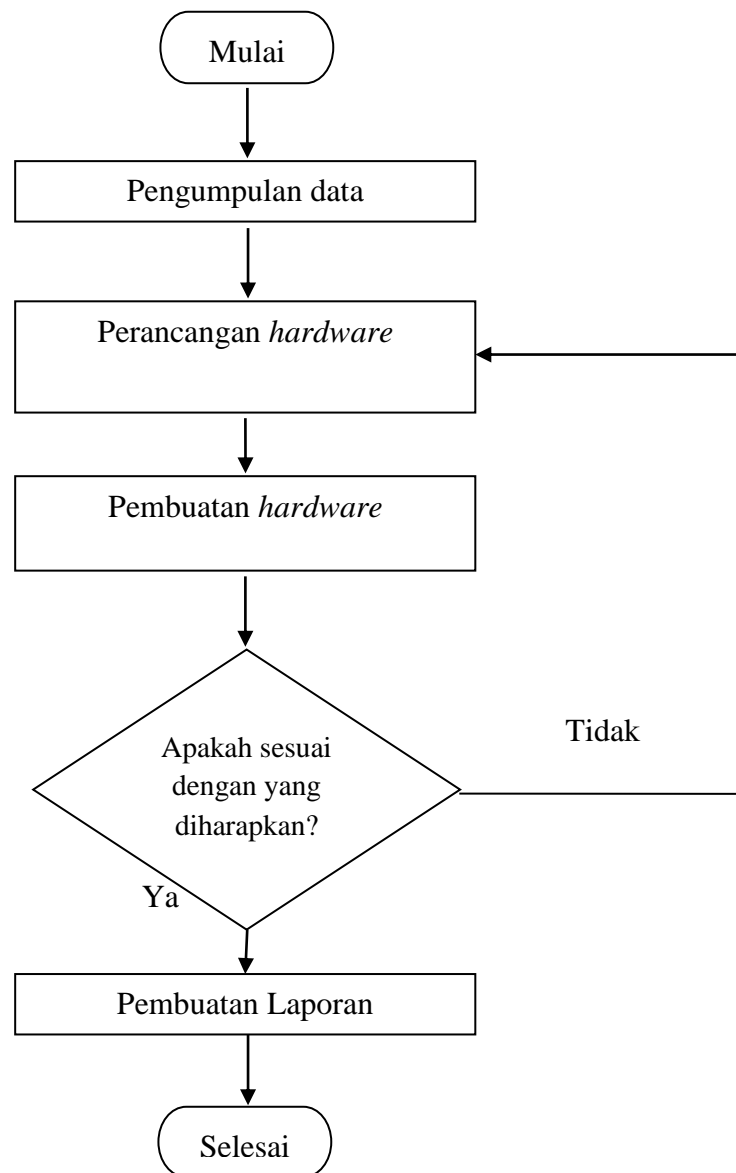
Tabel 2.1 Spesifikasi *Fingerprint* Adafruit [18]

No	Spesifikasi	Keterangan
1.	Fingerprint Adafruit	AS606
2.	Supply voltage	3.6 - 6.0VDC
3.	Operating current: 120mA max	120mA max
4.	Peak current: 150mA max	150mA max
5.	Fingerprint imaging time	<1.0 seconds
6.	Window area	14mm x 18mm
7.	Signature file	256 bytes
8.	Template file	512 bytes
9.	Storage capacity	162 templates
10.	Safety ratings	(1-5 low to high safety)
11.	False Acceptance Rate	<0.001% (Security level 3)
12.	False Reject Rate	<1.0% (Security level 3)
13.	Interface	TTL Serial
14.	Baud rate	9600, 19200, 28800, 57600
15.	Working temperature	-20C to +50C
16.	ratingWorking humidyFull	40%-85% RH
17.	Dimensions	56 x 20 x 21.5mm
18.	Exposed Dimensions (when placed in box)	21mm x 21mm x 21mm triangular
19.	Weight	20 grams

### BAB III

## METODE PELAKSANAAN

Agar pelaksanaan proyek akhir dapat berjalan dengan lancar maka harus mengikuti *flowchart* pada Gambar 3.1 sebagai berikut.



Gambar 3.1 *Flowchart* Metode Pelaksanaan

Dari gambar di atas dapat dijelaskan metode penelitian proyek akhir ini meliputi beberapa data:

### **3.1 Pengumpulan Data**

Pengumpulan data berfungsi untuk mendapatkan daftar semua peralatan dan komponen yang akan digunakan dalam pembuatan proyek akhir. Selain itu juga berfungsi untuk mengetahui prinsip kerja penggunaan sensor *Fingerprint*. Pada studi ini didapatkan data dari internet, referensi dari buku-buku yang menunjang materi tugas akhir serta bimbingan dosen dan instruktur Polman Negeri Bangka Belitung. Berikut ini adalah metode pengumpulan data secara primer dan sekunder tersebut yaitu :

1. Pengumpulan data secara primer

Pengumpulan data secara primer yaitu pengumpulan data yang sumber datanya diperoleh secara langsung dari narasumber seperti bimbingan. Bimbingan dari dosen pembimbing diperlukan karena dosen pembimbing yang akan mengarahkan, membantu serta memberi materi kepada para mahasiswa. Pengambilan data dari bimbingan berupa pembuatan program dan pembuatan tampilan aplikasi serta perancangan dan pembuatan program.

2. Pengumpulan data secara sekunder

Pengumpulan data secara sekunder adalah pengumpulan data yang sumber datanya diperoleh secara tidak langsung dari narasumber seperti studi literatur. Studi literatur dapat digunakan untuk pemecahan masalah dengan cara menelusuri sumber – sumber tulisan yang pernah dibuat sebelumnya, data dari berbagai sumber yang terkait dengan program ini didapat melalui internet dan diambil dari berbagai macam buku yang berhubungan dengan dasar teori materi yang dibuat yaitu pengambilan data tentang tata cara penggunaan sensor *fingerprint* adafruit.

### **3.2 Perancangan *Hardware* dan *Software***

Perancangan *hardware* bertujuan agar alat yang dibuat dapat mencapai target yang diinginkan dan semua sistem dapat bekerja dengan baik. Setelah perancangan *hardware* selesai, maka langkah selanjutnya adalah menguji

hardware ke elektrika motor. Pada tahapan ini ada beberapa hal yang dilakukan yaitu:

- a) Merakit penempatan sensor, lcd dan keypad.
- b) Pembuatan rangkaian dan pemasangan sensor yang akan digunakan.
- c) Pembuatan program pada arduino dan sensor fingerprint.
- d) Pengujian pada modul sensor.
- e) Pemasangan elektrikal sensor pada motor.

Setelah *hardware* dibuat maka langkah selanjutnya yaitu pemasangan pada motor untuk mengambil data.

### **3.4 Uji Coba**

Proses uji coba dilakukan sebagai tolak ukur berhasil atau tidaknya alat “*Sistem Pengaman Motor Menggunakan Sidik Jari*” yang dibuat. Dengan begitu, kita dapat mengevaluasi sistem kerja dari alat tersebut. Apabila sistem kerja alat tidak sesuai dengan yang diinginkan, maka kembali ke proses perancangan *hardware* untuk perbaikan sistem kerja pada alat.

### **3.5 Pembuatan Laporan**

Tahap ini adalah tahap akhir jika alat yang dibuat telah memenuhi tuntutan yang ingin dicapai, maka proses pembuatan alat dapat dinyatakan selesai. Dengan selesainya alat maka penulis akan membuat laporan detail tentang proses pembuatan alat dari tahap awal sampai tahap akhir.

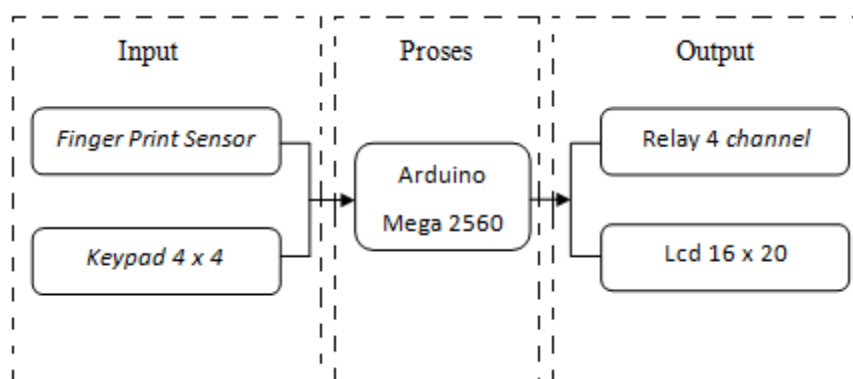


## BAB IV PEMBAHASAN

Pada bab ini akan dibahas mengenai proses desain dan pengembangan produk serta uji coba peralatan yang dikembangkan yaitu sebagai berikut.

### 4.1 Diagram Blok

Berikut merupakan diagram blok dari Pengaman Motor Menggunakan Sidik Jari .



Gambar 4.1 Blok Diagram Pengaman Motor Menggunakan Sidik Jari

### 4.2 Hardware Pengaman Motor

Perancangan *hardware* Pengaman Motor terdiri dari box sebagai tempat peletakan komponen-komponen dari beberapa alat yang digabungkan dan perancangan sidik jari dan keypad yang diletakan pada body motor. Proses pembuatan *hardware* dibuat secara bertahap dan dimulai dari pemasangan masing-masing *part* atau bagian dari box pengaman motor. Adapun ukuran atau dimensi dari box *hardware* pengaman motor yang akan dibuat yaitu 18,5 cm x11,5 cm x6 cm. Setelah masing-masing *part* atau bagian dari box pemasangan motor selesai dibuat, selanjutnya adalah proses pemasangan sidik jari dan keypad

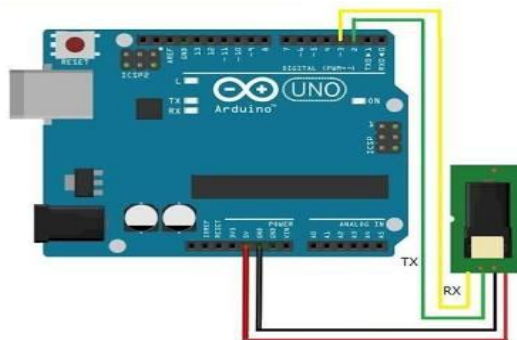
yang diletakkan pada body depan motor. Kemudian akan diproses *assembling* masing-masing *part* menjadi satu seperti pada gambar berikut.



Gambar 4.2 Desain Peletakan Alat Pada Motor

#### 4.2.1 *Finger Print Sensor*

*Finger Print Sensor* digunakan untuk mendeteksi sidik jari yang berupa *input* dari beberapa *user* yang digunakan untuk diproses menjadi sebuah data didalam arduino. Berikut adalah gambar skematik pemasangan pin rangkaian *Finger Print sensor* pada Arduino Mega 2560 yang dibuat menggunakan *software* ISIS Proteus.



Gambar 4.3 Pemasangan Pin *Finger Print Sensor*

#### 4.2.1.1 Pengujian *Finger Print Sensor*

Pengujian *fingerprint sensor* dilakukan dengan membuat program untuk mendaftarkan sidik jari dari *user* yang ingin menggunakannya dan juga menguji *fingerprint sensor* dengan membuat program penghapusan *user* itu sendiri. Sehingga memudahkan pengguna sepeda motor jika ingin meminjamkan motornya kepada keluarga atau pun teman terdekat.

➤ Berikut adalah *list* program arduino untuk pendaftaran *finger print sensor*.

```
//Program Daftar
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() );
}

```

➤ Berikut adalah program penghapusan sidik jari yang tersimpan di memori *internal finger print*.

**// Penghapusan Sidik Jari**

```

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
{
    for(int i=49;i<128;i++)
    {
        Serial.println("Please type in the ID # (from 1 to 127) you want
to delete...");
        //uint8_t id = readnumber();
        if (i == 0) { // ID #0 not allowed, try again!
            return;
        }

        Serial.print("Deleting ID #");
        Serial.println(i);

        deleteFingerprint(i);
        delay(500);
    }
}

```

Berikut adalah tabel hasil percobaan pendaftaran sidik jari pada *finger print sensor* pada pin Arduino Mega 2560, berdasarkan *list* program yang dibuat. Berikut hasil tabel pengujian dari *finger print sensor* :

Tabel 4.1 Hasil Pengujian Pendaftaran Finger Print Sensor

No	User	Keterangan	Status
1	Muhammad Reza N	Terdaftar	ID 1
2	Fatur Rivando	Terdaftar	ID 2
3	KK Anggrainy	Terdaftar	ID 3
4	Steven Novaldy	Terdaftar	ID 4
5	Rizki adrian	Terdaftar	ID 5
6	Bagas Donikara	Terdaftar	ID 6

Berikut adalah gambar data *serial monitor* setelah melalui proses pendaftaran sidik jari.

```

Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127) you want to save this finger as...
Enrolling ID #1
Waiting for valid finger to enroll as #1
.
.
.
.
.
.
.
Image taken
Image converted
Remove finger
ID 1
Place same finger again
.....
Image converted
Creating model for #1
Prints matched!
ID 1
Stored!
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127) you want to save this finger as...

```

Gambar 4.4 Hasil Pendaftaran Sidik Jari

Tabel 4.2 Hasil Penghapusan *Finger Print Sensor*

No	User	Keterangan	Status
1	Muhammad Reza N	Terhapus	-
2	Fatur Rivando	Terhapus	-
3	KK Anggrainy	Terhapus	-
4	Steven Novaldy	Terhapus	-
5	Rizki adrian	Terhapus	-
6	Bagas Donikara	Terhapus	-

Berikut adalah gambar proses penghapusan yang ada di *serial monitor*

```

Please type in the ID # (from 1 to 127) you want to delete...
Deleting ID #1
Deleted!
Please type in the ID # (from 1 to 127) you want to delete...
Deleting ID #2
Deleted!
Please type in the ID # (from 1 to 127) you want to delete...
Deleting ID #3
Deleted!
Please type in the ID # (from 1 to 127) you want to delete...
Deleting ID #4
Deleted!
Please type in the ID # (from 1 to 127) you want to delete...
Deleting ID #5
Deleted!
Please type in the ID # (from 1 to 127) you want to delete...
Deleting ID #6
Deleted!

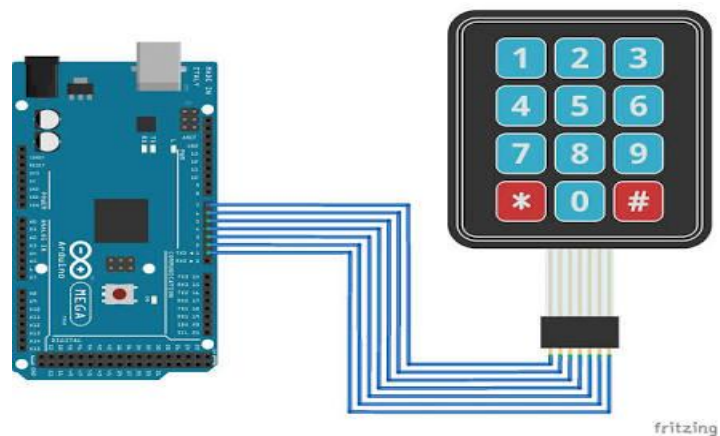
```

Gambar 4.5 Hasil Penghapusan *Finger print Sensor*

Dari hasil pengujian yang dilakukan dapat menunjukkan bahwa program yang dibuat dapat digunakan untuk mengetahui proses sidik jari yang terbaca dan tersimpan dari *finger print sensor*. Dari hasil pengujian *finger print sensor* tersebut didapatkan bahwa pada pengujian pendaftaran dapat terbaca dengan baik dengan keterangan ID *user* yang telah terdaftar. Pada pengujian penghapusan data *finger print sensor* akan menghapus data *user ID* yang akan digunakan. Dari hasil pengujian dan analisa dapat disimpulkan bahwa *finger print sensor* yang digunakan sesuai dengan perancangan yang diinginkan.

### 4.2.2 Keypad 4 x 4 Matrix Arduino

Keypad 4 x 4 digunakan sebagai *input* dalam menu untuk mendaftar dan menghapus sidik jari berupa *input* dari beban yang digunakan untuk mendaftarkan dan menghapus sidik jari yang telah di program didalam arduino. Berikut adalah Gambar 4.4 skematik pemasangan pin keypad pada Arduino Mega 2560 yang dibuat menggunakan *software* ISIS Proteus.



Gambar 4.6 Pemasangan Pin keypad 4 x 4 ke Arduino Mega 2560

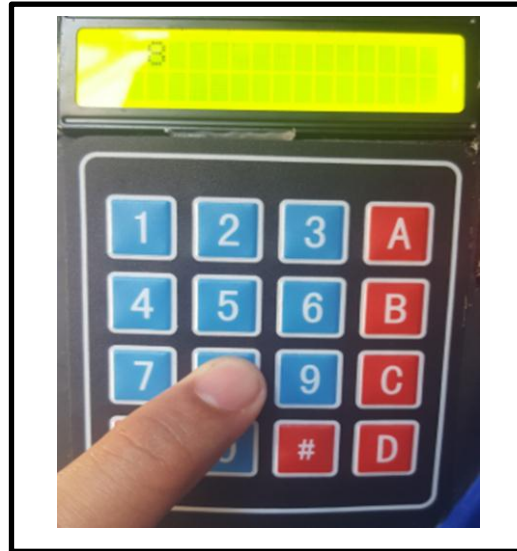
#### 4.2.2.1 Pengujian Keypad 4 x 4 Matrix Arduino

Pengujian keypad dilakukan dengan membuat program *input char* yang akan ditampilkan pada Lcd 16 x 2 yang akan di proses pada Arduino Mega 2560. Berikut adalah *list* program arduino untuk menampilkan angka pada keypad yang akan ditampilkan pada lcd 16 x 2.

```
//Program Keypad
void setup(){
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
}
void loop(){
  char keypressed= keypad.getKey()
```



Berikut adalah gambar percobaan pada keypad yang tampil di LCD.

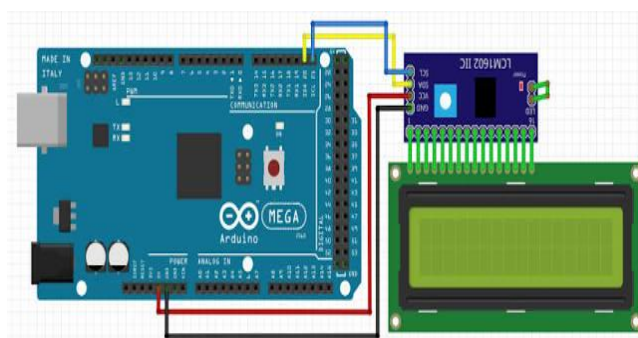


Gambar 4.7 Tampilan *input char* keypad 4 x 4 ke LCD 16 x 2

Dari hasil pengujian yang dilakukan menunjukkan bahwa program yang dibuat dapat digunakan untuk menginput *char* yang kita tekan pada keypad dan akan ditampilkan pada Lcd 16x2. Dari hasil pengujian dan analisa dapat disimpulkan bahwa keypad 4 x 4 yang digunakan sesuai dengan perencanaan yang diinginkan.

#### 4.2.3 LCD 16 x 2

Lcd 16 x 2 digunakan untuk menampilkan tampilan menu pendaftaran dan penghapusan sidik jari *user* maupun admin. Berikut adalah gambar skematik pemasangan pin rangkaian Lcd 16 x 2 pada Arduino Mega 2560 yang dibuat menggunakan *software* ISIS Proteus.



Gambar 4.8 Pemasangan Pin pada Lcd

#### 4.2.3.1 Pengujian LCD 16 x 2

Langkah awal yang dilakukan dalam pengujian Lcd 16 x 2 adalah menampilkan menu daftar dan hapus pada Lcd 16 x 2 yang akan di proses oleh Arduino Mega 2560. Berikut adalah *list* program pengujian Lcd 16 x 2.

```
//Program Lcd
void setup(){
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
}
void loop(){
  if (proses==1) //Menu admin
  {
    Serial.println("Proses 1");
    Serial.println("Menu Admin");
    lcd.print("Daftar (A)          ");
    lcd.setCursor(0, 1);
    lcd.print("Hapus (B)          ");
    lcd.setCursor(0, 0);
  }
}
```

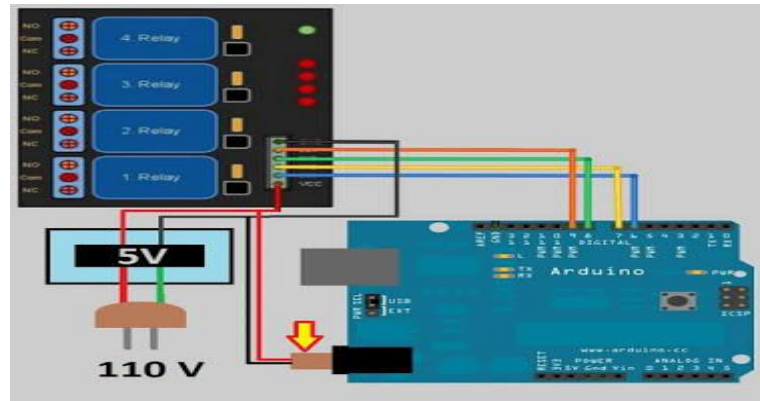


Gambar 4.9 Tampilan Menu Pada LCD 16 x 2

Adapun hasil pengujian yang dilakukan pada Lcd 16 x 2 adalah untuk menampilkan menu daftar dan hapus. Dari pengujian yang dilakukan dapat dianalisa bahwa Lcd 16 x 2 berjalan sesuai dengan perencanaan yang diinginkan.

#### 4.2.4 Modul *Relay 4 Channel*

Modul relay 4 channel ini adalah salah satu bagian utama untuk mengontrol serta pada kelistrikan motor.



Gambar 4.10 Pemasangan Pin Pada Modul Relay 4 Channel

##### 4.2.4.1 Pengujian Modul Relay 4 Channel

Berikut adalah *list* program arduino untuk pengecekan modul relay 4 *channel* menggunakan komunikasi serial.

```
//Program Relay
void setup()
{
  Serial.begin(9600);
  pinMode(R1, OUTPUT);
  pinMode(R2, OUTPUT);
  pinMode(R3, OUTPUT);
  pinMode(R4, OUTPUT);

  digitalWrite(R1, HIGH);
  digitalWrite(R2, HIGH);
  digitalWrite(R3, HIGH);
  digitalWrite(R4, HIGH);
}

void loop()
{
  digitalWrite(R1, LOW);
  delay(1000);
}
```

```
digitalWrite(R2, LOW);  
delay(1000);  
digitalWrite(R3, LOW);  
delay(1000);  
digitalWrite(R4, LOW);  
delay(1000);  
}
```

#### 4.2.5 Modul *Stepdown* LM 2596

Modul *Stepdown* LM 2596 ini adalah salah satu komponen yang sangat penting untuk *supply* tegangan pada arduino Mega 2560. Ketika tegangan tidak sesuai dengan *data sheet* atau melebihi tegangan yang telah dianjurkan maka arduino Mega 2560 akan mudah rusak.



Gambar 4.11 Modul *Stepdown* LM 2596

##### 4.2.5.1 Pengujian Modul *Stepdown* LM 2596

Berikut adalah tegangan awal ketika pengambilan sumber yang belum diturunkan pada tegangan aki ke arduino mega 2560.



Gambar 4.12 Tegangan sumber

Setelah mengukur tegangan awal didapatkan tegangan bernilai 12,65 Vdc, berikut adalah gambar tegangan yang telah diturunkan menggunakan modul *stepdown*.



Gambar 4.13 Tegangan Setelah Diturunkan

Dari data diatas dapat diketahui bahwa tegangan yang diukur sebesar 6,22 Vdc. Sesuai dengan tegangan kerja arduino dan dari data ini dapat disimpulkan bahwa modul *stepdown* lm 2596 sesuai dengan perencanaan yang diinginkan.

#### 4.2.6 Aki (Akumulator)

Aki menjadi sumber utama pada pemakaian komponen elektronika, pada alat pengaman ini. Aki yang digunakan adalah aki dengan tipe GTZ5S yang memiliki tegangan 12 volt dengan arus 4 Ampere. Seperti *supply* tegangan dc yang lain aki pun mempunyai polaritas yaitu kutub (+) dan kutub (-). Tegangan dan arus sangat penting bagi rangkaian kontrolnya. Jika tegangan dan arus dari aki dibawah tegangan kerja arduino kurang, maka arduino tidak akan berjalan dengan stabil. Aki yang masih bagus adalah aki yang mengeluarkan tegangan dan arus yang stabil sehingga pada saat motor distarter, maka tidak akan mempengaruhi sistem kerja dari rangkaian kontrol lain.



Gambar 4.14 Aki (Akumulator)

### **4.3 Software Pengaman Motor**

#### **4.3.1 Perancangann Pembuatan Program Arduino Mega 2560**

Perencanaan program pada Arduino Mega 2560 merupakan langkah awal dalam mendesain atau merancang bagaimana sistem pengontrolan yang akan dibuat. Perancangan program Arduino Mega 2560 ini dibuat untuk melihat alur pemograman pada pengaman motor menggunakan sidik jari. Perencanaan program yang dilakukan meliputi pembacaan data pada sensor sidik jari yang digunakan, perancangan pemograman pendaftaran sidik jari *user*, perancangan program penghapusan sidik jari *user*, perancangan program penghapusan admin, dan perancangan program keypad sebagai menu serta pilihan *id user* yang akan ditampilkan di Lcd yang di program melalui Arduino Mega 2560.

##### **4.3.1.1 Pembuatan Program Arduino Mega 2560**

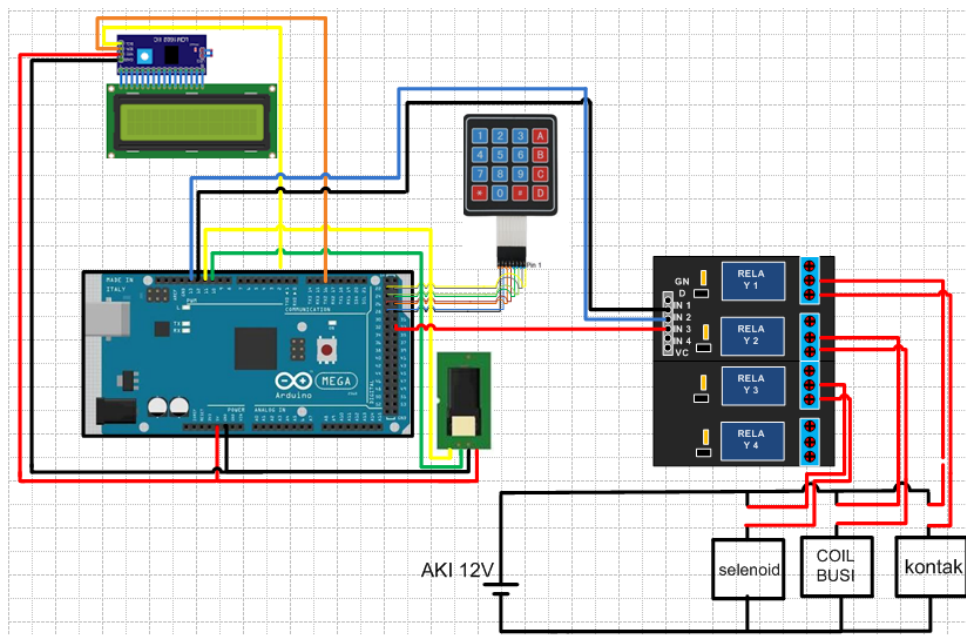
Pembuatan program dilakukan menggunakan *software* Arduino. Pemograman Pengaman Motor Menggunakan Sidik Jari dibagi menjadi beberapa tahapan sesuai dengan perancangan yang telah dibuat yaitu sebagai berikut :

- a. Pemograman *Fingerprint Sensor* untuk membaca sidik jari admin dan *user*.
- b. Pemograman Modul Relay 4 Channel untuk mengontrol selenoid, elektrik motor, serta *coil* motor.
- c. Pemograman Keypad 4 x 4 untuk membuat menu serta id untuk *user* dan admin pengguna sepeda motor.
- d. Pemograman sensor Lcd 16 x 2 untuk menampilkan menu pendaftaran id *user* dan admin, serta tampilan bahwa motor siap di gunakan.
- e. Pemograman Eeprom Arduino sebagai penyimpanan data admin.
- f. Pemograman gabungan pengaman motor menggunakan sidik jari secara keseluruhan.

### **4.4 Peletakan Tempat *Box* Kontrol Dan Tampilan LCD Serta Keypad**

Pada peletakan panel box kontrol akan diletakan dalam jok motor sedangkan sensor sidik jari, lcd dan keypad akan diletakan di bodi depan motor.

Proses pendaftaran, penghapusan dan proses akses memasukan sidik jari pengaman motor dilakukan di bodi depan motor. Peletakan box bisa dilihat pada Gambar 4.16. Sebelum peletakan box panel sidik jari, maka hal yang dilakukan adalah mengamankan kelistrikan motor letaknya di *coil*. Hal ini dilakukan dengan cara memutuskan kabel coil dan kabel kontak. Berikut gambar rangkaian bisa dilihat pada Gambar 4.15.



Gambar 4.15 Rangkaian Kontrol Keseluruhan

1. Proses perakitan komponen didalam box kontrol.



Gambar 4.16 Perakitan Komponen Didalam Box

2. Proses peletakan modul box panel kontrol.



Gambar 4.17 Peletakan Box Panel

3. Proses perapian box.



Gambar 4.18 Setelah dirapikan

4. Proses jadi setelah pemasangan sensor, keypad dan lcd.



Gambar 4.19 Pemasangan sensor



#### 4.5 Pengujian Pengaman Motor Menggunakan Sidik Jari

Setelah bagian *hardware* dan *software* dari Pengaman Motor Menggunakan Sidik Jari. Telah dibuat, proses selanjutnya adalah pengujian terhadap system yang telah dibuat pada Pengaman Motor Menggunakan Sidik Jari. Pengujian sistem ini dilakukan untuk menguji serta melihat bagaimana hasil yang akan didapatkan setelah melakukan perancangan, pembuatan, dan pengujian pada masing-masing *hardware* maupun *software*. Berikut adalah hasil data pengujian pengaman motor menggunakan sidik jari yang telah dibuat yaitu sebagai berikut berikut Tabel 4.3 pengujian keseluruhan alat dan aplikasi.

Tabel 4.3 Pendaftaran Sidik Jari.

No	ID Number Fingerprint	Nama Pengguna	Sidik Jari Yang Didaftarkan				
			Ibu Jari	Jari Telunjuk	Jari Tengah	Jari Manis	Jari Kelingking
1.	2	(A) Adam	√	-	-	-	-
2.	3	(B) Reza	-	√	-	-	-
3.	4	(C) Habibi	-	-	√	-	-
4.	5	(D) Aji	-	-	-	√	-
5.	6	(E) Ranggi	-	-	-	-	√
6.	7	(F) Yosi	√	-	-	-	-
7.	8	(G) Ferdi	-	-	√	-	-
8.	-	(H)	-	-	-	-	-
9.	-	(I)	-	-	-	-	-
10.	-	(J)	-	-	-	-	-

Tabel 4.4 Pengujian Untuk Menghidupkan Kelistrikan Motor

Nama Pengguna	Respon Fingerprint	Keterangan
Pengguna A	Aktif	Motor Bisa Di Starter
Pengguna B	Aktif	Motor Bisa Di Starter
Pengguna C	Aktif	Motor Bisa Di Starter
Pengguna D	Aktif	Motor Bisa Di Starter
Pengguna E	Aktif	Motor Bisa Di Starter
Pengguna F	Aktif	Motor Bisa Di Starter
Pengguna G	Aktif	Motor Bisa Di Starter
Pengguna H	Tidak Aktif	Tidak Bisa Di Starter
Pengguna I	Tidak Aktif	Tidak Bisa Di Starter
Pengguna J	Tidak Aktif	Tidak Bisa Di Starter

Tabel 4.5 Pengujian Sidik Jari Pengguna A Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	√	Kelistrikan Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.6 Pengujian Sidik Jari Pengguna A Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.7 Pengujian Sidik Jari Pengguna B Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	√	Kelistrikan Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.8 Pengujian Sidik Jari Pengguna B Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.9 Pengujian Sidik Jari Pengguna C Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.10 Pengujian Sidik Jari Pengguna C Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	√	Kelistrikan Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.11 Pengujian Sidik Jari Pengguna D Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.12 Pengujian Sidik Jari Pengguna D Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	√	Kelistrikan Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.13 Pengujian Sidik Jari Pengguna E Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.14 Pengujian Sidik Jari Pengguna E Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	√	Kelistrikan Aktif

Tabel 4.15 Pengujian Sidik Jari Pengguna F Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	√	Kelistrikan Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.16 Pengujian Sidik Jari Pengguna F Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.17 Pengujian Sidik Jari Pengguna G Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.18 Pengujian Sidik Jari Pengguna G Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	√	Kelistrikan Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.19 Pengujian Sidik Jari Pengguna H Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.20 Pengujian Sidik Jari Pengguna H Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.21 Pengujian Sidik Jari Pengguna I Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.22 Pengujian Sidik Jari Pengguna I Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.23 Pengujian Sidik Jari Pengguna J Untuk Tangan Kanan

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Tabel 4.24 Pengujian Sidik Jari Pengguna J Untuk Tangan Kiri

Nama Jari	Respon Sidik Jari	Keterangan
Ibu Jari	-	Kelistrikan Tidak Aktif
Jari Telunjuk	-	Kelistrikan Tidak Aktif
Jari Tengah	-	Kelistrikan Tidak Aktif
Jari Manis	-	Kelistrikan Tidak Aktif
Jari Kelingking	-	Kelistrikan Tidak Aktif

Dari hasil pengujian yang dilakukan dapat di simpulkan bahwa hanya sidik jari yang didaftarkan saja yang bisa mengakses motor dan dapat kita lihat dari tabel pengujian bahwa setiap jari masing-masing orang mempunyai sidik jari yang berbeda.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian dan analisa terhadap fungsi alat pada Proyek Akhir dengan judul “pengaman motor menggunakan sidik jari” ini maka dapat ditarik kesimpulan sebagai berikut:

1. Pengaman motor menggunakan sidik jari ini, dapat mengamankan motor dengan cara mengakses sidik jari yang telah di daftarkan.
2. Pengaman motor menggunakan sidik jari ini, mengontrol kelistrikan motor.
3. Pengaman motor menggunakan sidik jari ini juga di lengkapi menu untuk pendaftaran dan penghapusan *user*.
4. Pengaman motor menggunakan sidik jari tidak akan merespon sidik jari yang tidak terdaftar oleh admin.

#### **5.2 Saran**

Apabila alat ini akan dikembangkan lebih lanjut, fungsi yang perlu diperbaiki dan ditambahkan antara lain :

1. Pendaftaran hanya bisa dilakukan maksimal 5 *user* saja, jadi jika ingin mengembangkan maka harus bisa mendaftarkan lebih dari 5 *user*.
2. Kalo sidik jari terkena air maka harus dikeringkan.

# **LAMPIRAN**



# **LAMPIRAN 1**

## **DAFTAR RIWAYAT HIDUP**

## **DAFTAR RIWAYAT HIDUP**

### **1. Data Pribadi**

Nama lengkap : Muhammad Reza Nafianto  
Tempat dan tanggal lahir : Batam, 25 Sempember 1997  
Alamat rumah :  
Telp : -  
Hp : 0822 8050 5625  
Email : nafiantoreza@gmail.com  
Jenis kelamin : Laki-laki  
Agama : Islam

### **2. Riwayat Pendidikan**

SD Islam Al-Barkah	Batam	2003-2009
SMP Setia Budi	Batam	2009-2012
SMK Setia Budi	Pangkal Pinang	2012-2015

### **3. Pendidikan Non Formal**

-

Sungailiat, 3 Agustus 2018

Muhammad Reza Nafianto

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama lengkap : Fatur Rivando  
Tempat dan tanggal lahir : Pemali, 10 November 1996  
Alamat rumah : Jl. Lembawai No 12, Bangka  
Telp : -  
Hp : 0853 8117 1269  
Email : faturpolman@gmail.com  
Jenis kelamin : Laki-laki  
Agama : Islam

### 2. Riwayat Pendidikan

SD N 2	Pemali	2003-2009
SMP N 2	Pemali	2009-2012
SMA N 1	Pemali	2012-2015

### 3. Pendidikan Non Formal

-

Sungailiat, 3 Agustus 2018

Fatur Rivando

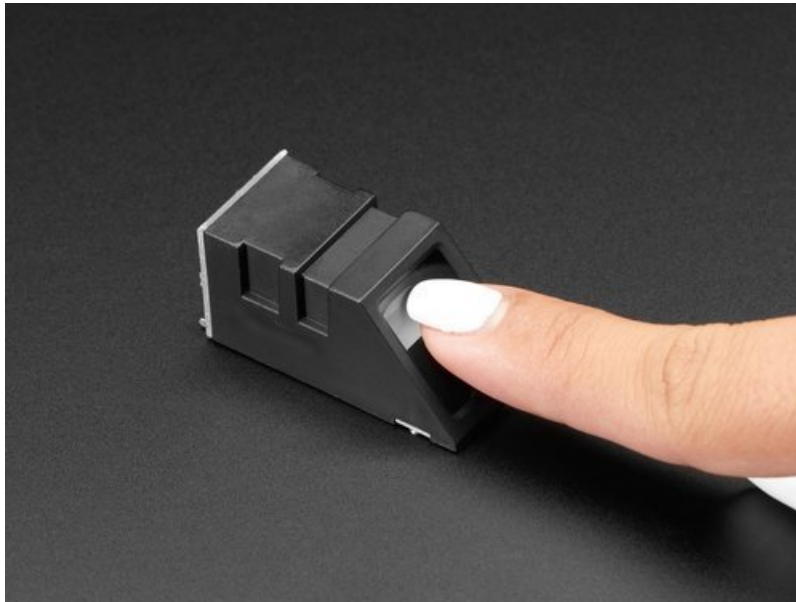
# **LAMPIRAN 2**

*Datasheet Finger Print Sensor AS606*



## Adafruit Optical Fingerprint Sensor

Created by lady ada



Last updated on 2017-12-31 09:39:09 PM UTC

## Guide Contents

Guide Contents	2
Overview	3
Enrolling vs. Searching	5
Enrolling New Users with Windows	6
Searching with the Software	12
Wiring for use with Arduino	13
Arduino UNO & Compatible Wiring	13
Hardware Serial Wiring	14
Soft & Hard Serial	15
Upload	15
Enrolling with Arduino	17
CircuitPython	18
Installing Library	18
Usage	19
Enrolling Prints	22
Finding Prints	23
Deleting Fingerprints	24
Downloads	25

## Overview



Secure your project with biometrics - this all-in-one optical fingerprint sensor will make adding fingerprint detection and verification super simple. These modules are typically used in safes - there's a high powered DSP chip that does the image rendering, calculation, feature-finding and searching. Connect to any microcontroller or system with TTL serial, and send packets of data to take photos, detect prints, hash and search. You can also enroll new fingers directly - up to 162 finger prints can be stored in the onboard FLASH memory. There's a red LED in the lens that lights up during a photo so you know its working.

We like this particular sensor because not only is it easy to use, it also comes with fairly straight-forward Windows software that makes testing the module simple - you can even enroll using the software and see an image of the fingerprint on your computer screen. But, of course, we wouldn't leave you a datasheet and a "good luck!" - [we wrote a full Arduino library so that you can get running in under 10 minutes. The library can enroll and search so its perfect for any project.](#) We've also [written a detailed tutorial on wiring and use.](#) This is by far the best fingerprint sensor you can get.

- **Supply voltage:** 3.6 - 6.0VDC
- **Operating current:** 120mA max
- **Peak current:** 150mA max
- **Fingerprint imaging time:** <1.0 seconds
- **Window area:** 14mm x 18mm
- **Signature file:** 256 bytes
- **Template file:** 512 bytes
- **Storage capacity:** 162 templates
- **Safety ratings** (1-5 low to high safety)
- **False Acceptance Rate:** <0.001% (Security level 3)
- **False Reject Rate:** <1.0% (Security level 3)
- **Interface:** TTL Serial

- **Baud rate:** 9600, 19200, 28800, 38400, 57600 (default is 57600)
- **Working temperature rating:** -20C to +50C
- **Working humidity:** 40%-85% RH
- **Full Dimensions:** 56 x 20 x 21.5mm
- **Exposed Dimensions** (when placed in box): 21mm x 21mm x 21mm triangular
- **Weight:** 20 grams



## Enrolling vs. Searching

---

There are basically two requirements for using the optical fingerprint sensor. First is you'll need to **enroll** fingerprints - that means assigning ID #'s to each print so you can query them later. Once you've enrolled all your prints, you can easily 'search' the sensor, asking it to identify which ID (if any) is currently being photographed.

You can enroll using the Windows software (easiest and neat because it shows you the photograph of the print) or with the Arduino sketch (good for when you don't have a Windows machine handy or for on-the-road enrolling).

## Enrolling New Users with Windows

The easiest way to enroll a new fingerprint is to use the Windows software. The interface/test software is unfortunately windows-only *but* you only need to use it once to enroll, to get the fingerprint you want stored in the module.

First up, you'll want to connect the sensor to the computer via a USB-serial converter. The easiest way to do this is to connect it directly to the USB/Serial converter in the Arduino. To do this, you'll need to upload a 'blank sketch' this one works well for "traditional" Arduinos, like the Uno and the Mega:

```
// this sketch will allow you to bypass the Atmega chip
// and connect the fingerprint sensor directly to the USB/Serial
// chip converter.

// Red connects to +5V
// Black connects to Ground
// White goes to Digital 0
// Green goes to Digital 1

void setup() {}
void loop() {}
```

The "blank" sketch won't work for "native USB" based Arduinos like the Leonardo, Micro, Zero, etc! Use the `Leo_passthu` sketch instead!

If you're using a Leonardo, Micro, Yun, Zero, or other native-USB device like ATSAM21 or ATmega32U4-based controller, use the `Leo_passthu` sketch instead of the "blank" sketch.

```
//Leo_passthu
// Allows Leonardo to pass serial data between fingerprint reader and Windows.
//
// Red connects to +5V
// Black connects to Ground
// Green goes to Digital 0
// White goes to Digital 1

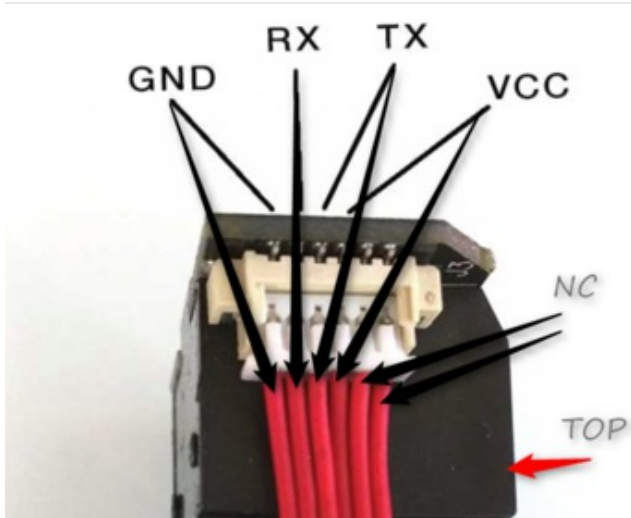
void setup() {
  // put your setup code here, to run once:
  Serial1.begin(57600);
  Serial.begin(57600);
}

void loop() {

  while (Serial.available())
    Serial1.write(Serial.read());
  while (Serial1.available())
    Serial.write(Serial1.read());
}
```

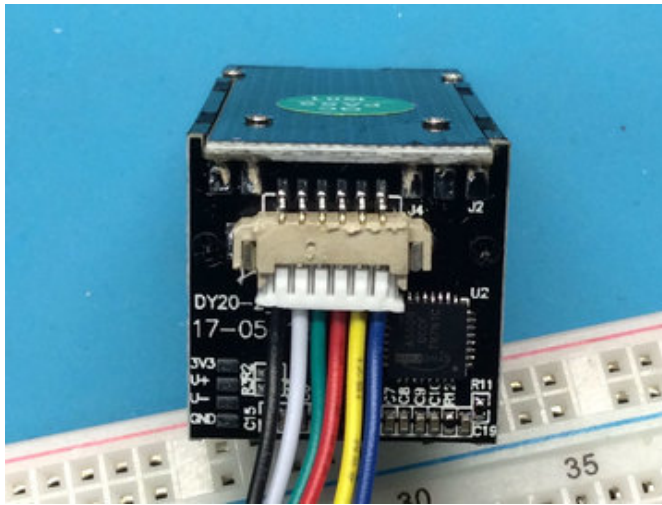
Wire up the sensor as described in the sketch comments **after** uploading the sketch. Since the sensor wires are so thin and short, we stripped the wire a bit and melted some solder on so it made better contact but you may want to solder the wires to header or similar if you're not getting good contact. When you plug in the power, you should see the red

LED blink to indicate the sensor is working.

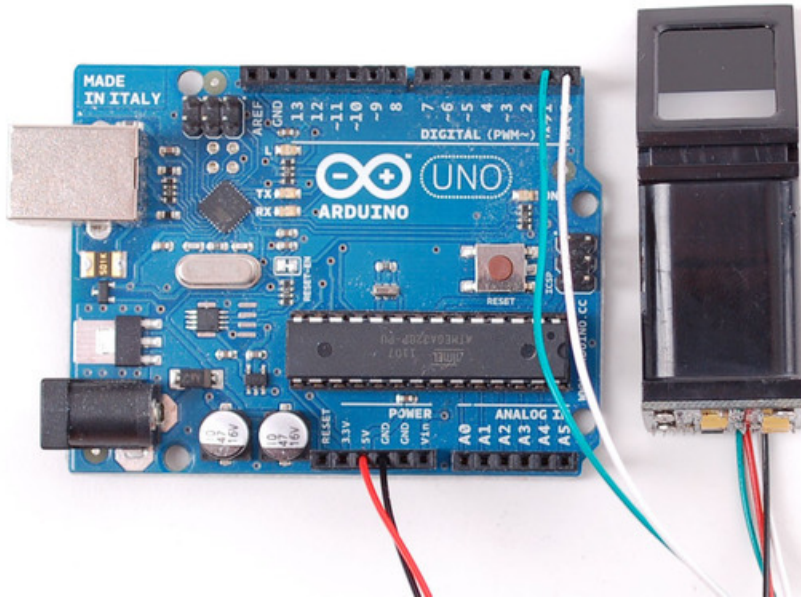


If your sensor has all the same-color wires, The first wire from the left is ground, then the two data pins, then power. You'll have to cut, strip and solder the wires.

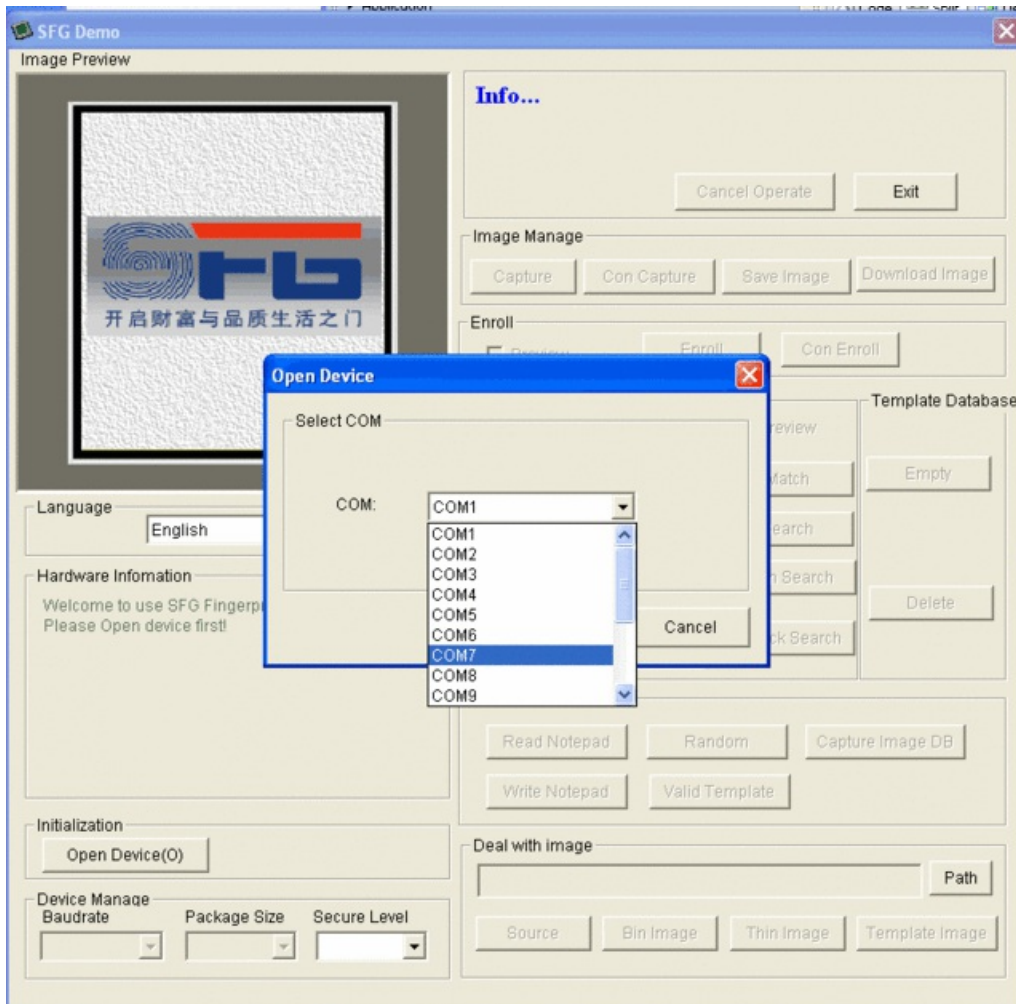
RX is the same as the White wire  
TX is the same as the Green wire



If your sensor has different wires, The first wire from the left should be the black wire ground, then the two data pins, RX is the white wire, TX is the green wire then the red power wire. You'll have to cut, strip and solder the wires.

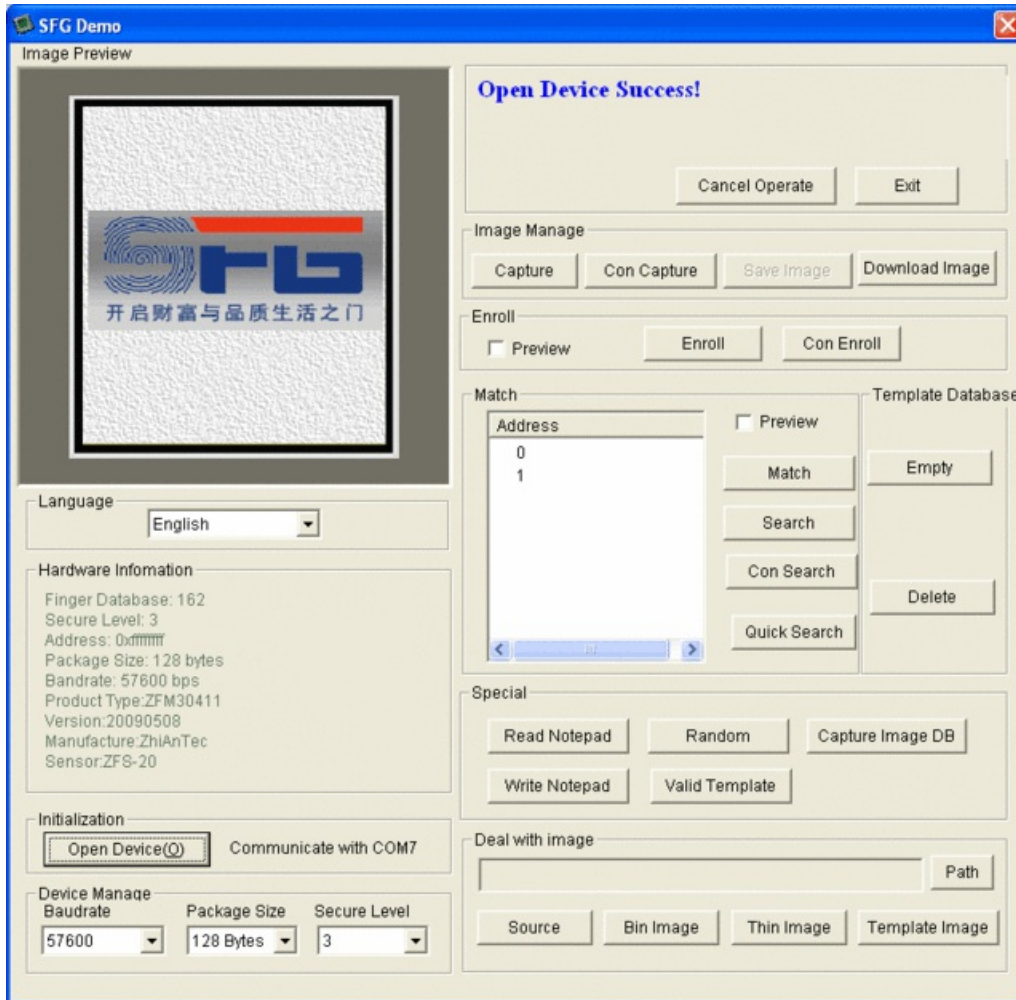


Start up the SFGDemo software and click **Open Device** from the bottom left corner. Select the **COM port** used by the Arduino.

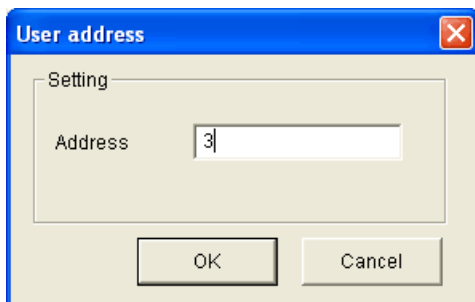


And press OK when done. You should see the following, with a blue success message and some device statistics in the bottom corner. You can change the baud rate in the bottom left hand corner, as well as the "security level" (how sensitive it is) but we suggest leaving those alone until you have everything running and you want to experiment. They should default to 57600 baud and security level 3 so set them if they're wrong

If you get an error when you Open Device, check your wiring, try swapping the RX and TX wires on the sensor, that's a common mixup!



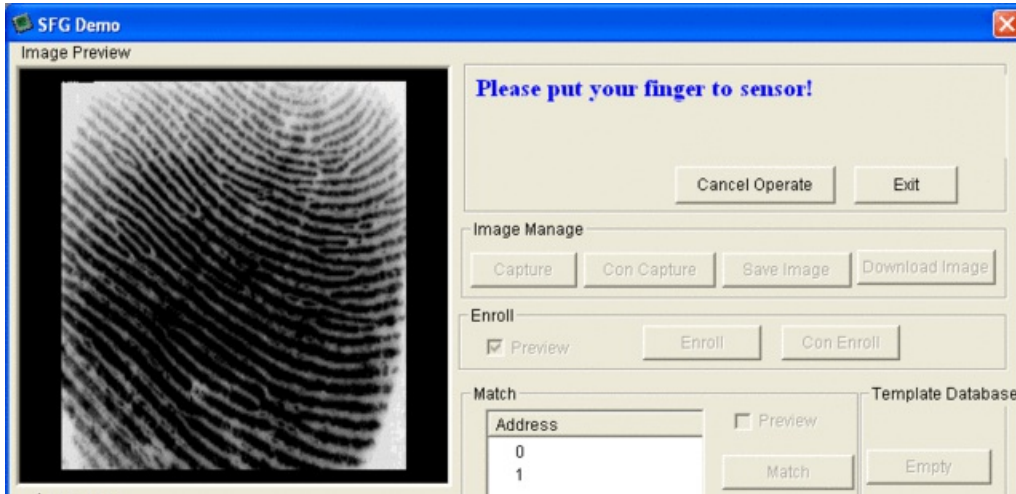
Lets enroll a new finger! Click the **Preview** checkbox and press the **Enroll** button next to it (**Con Enroll** means 'Continuous' enroll, which you may want to do if you have many fingers to enroll). When the box comes up, enter in the ID # you want to use. You can use up to 162 ID numbers.



The software will ask you to press the finger to the sensor

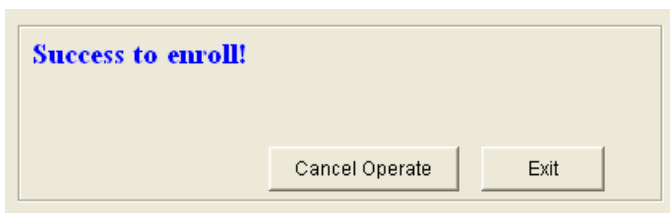


You can then see a preview (if you clicked the preview checkbox) of the fingerprint.



You will then have to repeat the process, to get a second clean print. Use the same finger!

On success you will get a notice.



If there's a problem such as a bad print or image, you'll have to do it again.

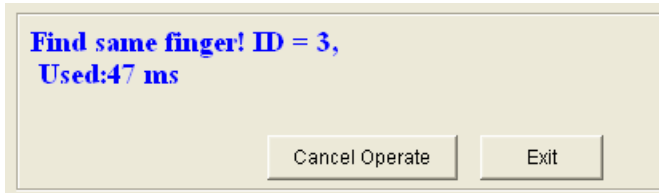
## Searching with the Software

---

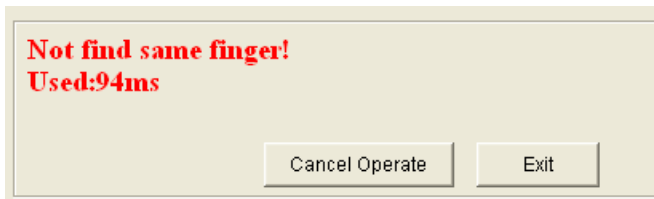
Once you have the finger enrolled, it's a good idea to do a quick test to make sure it can be found in the database. Click on the **Search** button on the right hand side.

When prompted, press a different/same finger to the sensor.

If it is the same finger, you should get a match with the ID #



If it is not a finger in the database, you will get a failure notice.

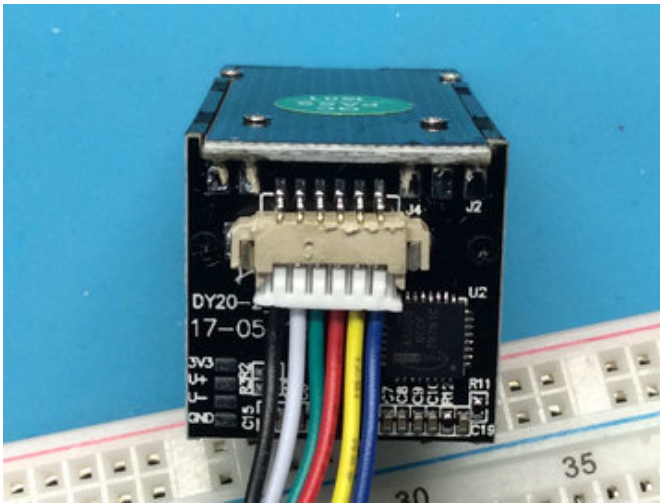




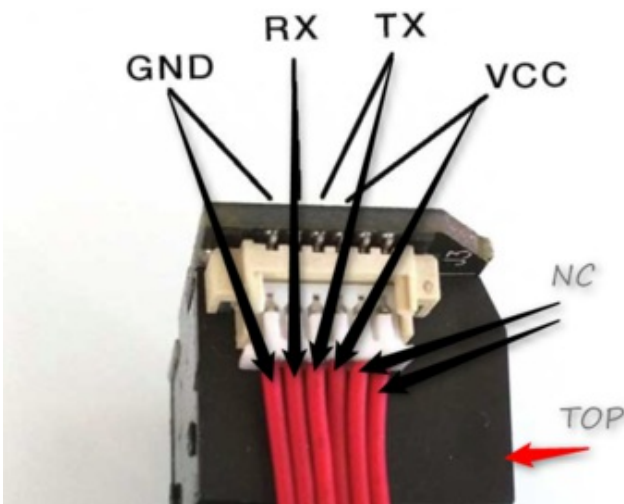
## Wiring for use with Arduino

Once you've tested the sensor, you can now use it within a sketch to verify a fingerprint. We'll need to rewire the sensor. Disconnect the green and white wires and plug the green wire into digital **2** and the white wire to digital **3**. (For ESP8266 use **4 & 5**, for devices with Hardware UART use **0 & 1**)

It is normal for the sensor to blink the LED quickly once powered, after that the LED will be off until you've started to request data from it



If your sensor has different wires, The first wire from the left should be the black wire ground, then the two data pins, RX is the white wire, TX is the green wire then the red power wire. You'll have to cut, strip and solder the wires.



If your sensor has all the same-color wires, The first wire from the left is ground, then the two data pins, then power. You'll have to cut, strip and solder the wires.

RX is the same as the White wire  
TX is the same as the Green wire

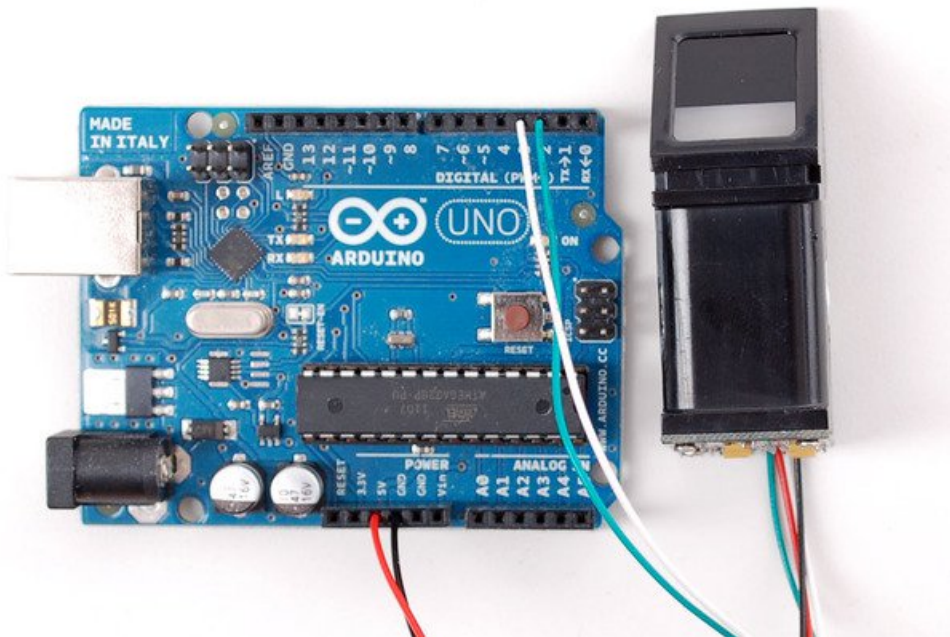
## Arduino UNO & Compatible Wiring

This example sketch uses pins **2** and **3** for software serial (on ATmega328P type boards by default) - Not all boards support Software Serial on all pins so check board documentation! For example on ESP8266 we used **4 & 5**

You can power the sensor from **3.3V** or **5V**

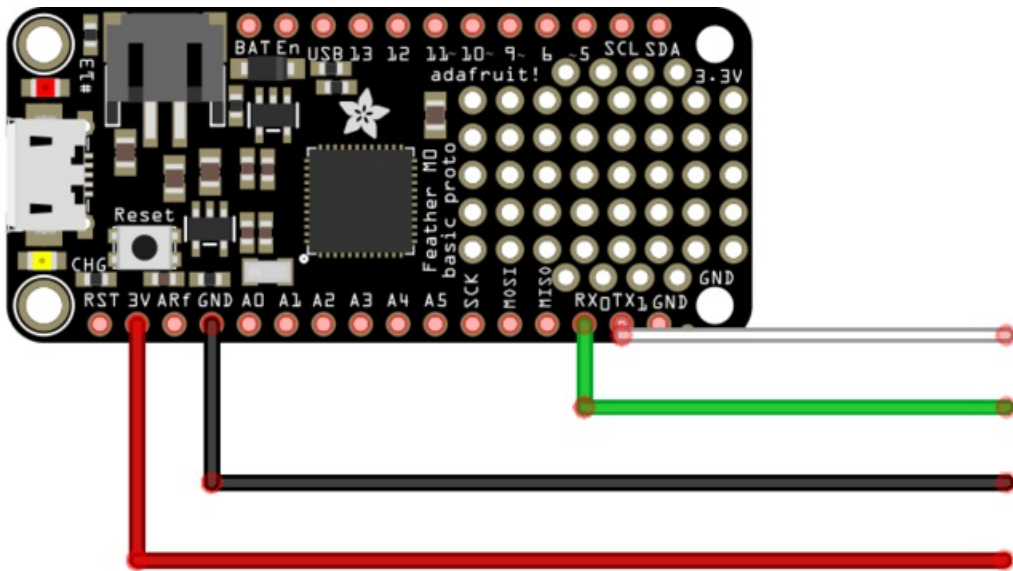
In the diagrams below we show the wires plugged directly into the Arduino. However, this does not work well because the wires are so thin and they don't make contact. You should solder thicker solid core wires to each

wire, to make good contact



## Hardware Serial Wiring

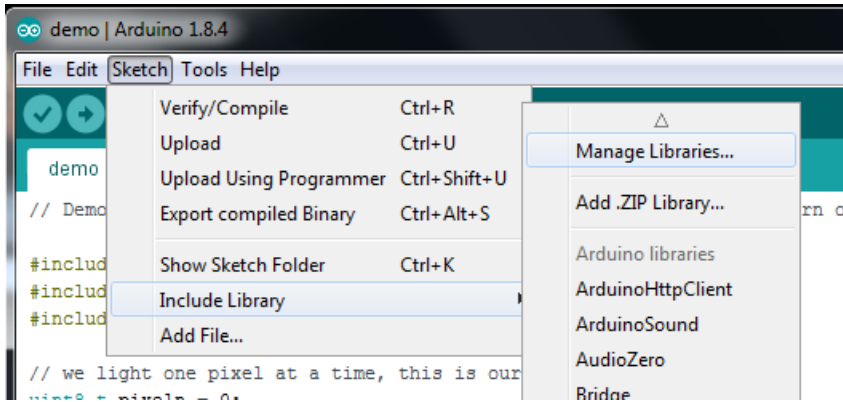
If you have a device with hardware serial, you should use that instead. Often this is pins #0 and #1



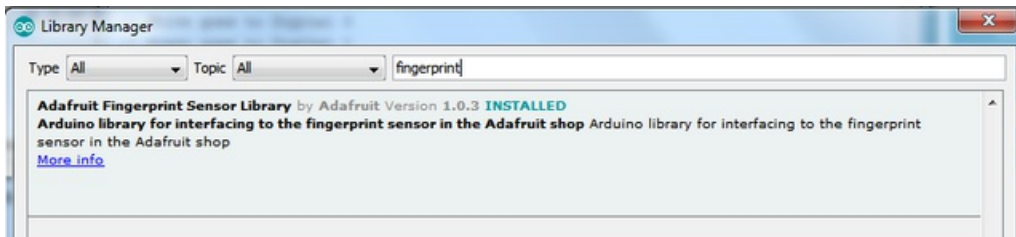
fritzing

Next, you'll need to install [the Adafruit Fingerprint sensor library](#) (also available from github).

Open up the Arduino Library Manager:



Type in **Fingerprint** until you see the **Adafruit Fingerprint** library show up!



Click Install! That's it. Now you should be able to select the **File**→**Examples**→**Adafruit\_Fingerprint**→**fingerprint** example sketch.

## Soft & Hard Serial

By default the sketch uses software serial (Arduino UNO & compatibles). If you are using a device with Hardware Serial, e.g does not have a USB-Serial converter chip, use that instead! Usually those are on pins 0 & 1

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

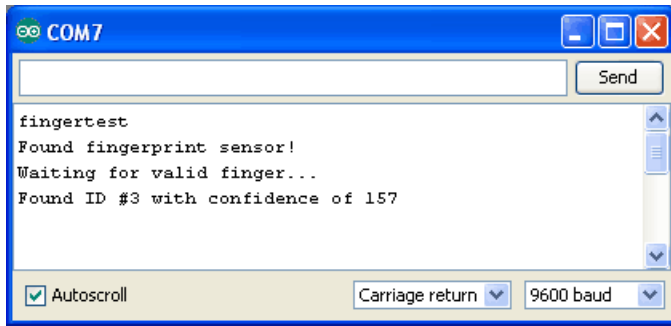
// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
// comment these two lines if using hardware serial
//#include <SoftwareSerial.h>
//SoftwareSerial mySerial(2, 3);
```

If necessary, uncomment/comment lines for hardware serial support

## Upload

Upload it to your Arduino as usual. Open up the serial monitor at 9600 baud and when prompted place your finger against the sensor that was already enrolled.

You should see the following:



The 'confidence' is a score number (from 0 to 255) that indicates how good of a match the print is, higher is better. Note that if it matches at all, that means the sensor is pretty confident so you don't have to pay attention to the confidence number unless it makes sense for high security applications.

Of course you have to have **enrolled** a fingerprint first! If you did this using the Windows program, that's good to go. If you have not yet, you should at least have gotten a **Found fingerprint sensor!** printout. You can go ahead to the next step to enroll fingerprints.

If you get **Did not find fingerprint sensor :(** check your wiring, maybe swap the RX and TX wire as that's a common issue

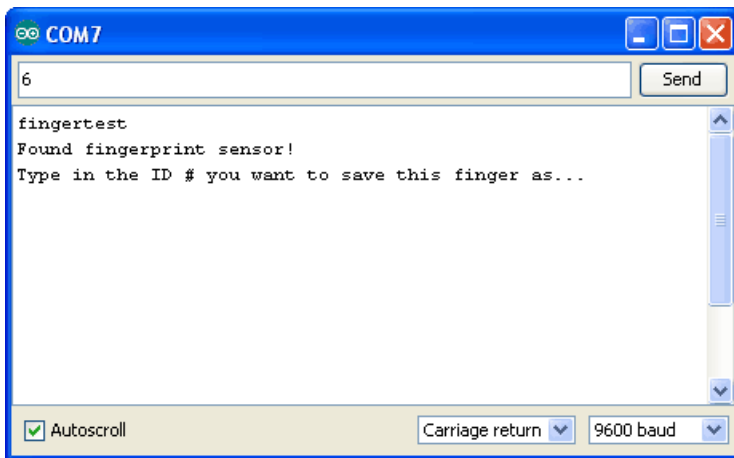
If you want to have a more detailed report, change the `loop()` to run `getFingerprintID()` instead of `getFingerprintIDez()` - that will give you a detailed report of exactly what the sensor is detecting at each point of the search process.

## Enrolling with Arduino

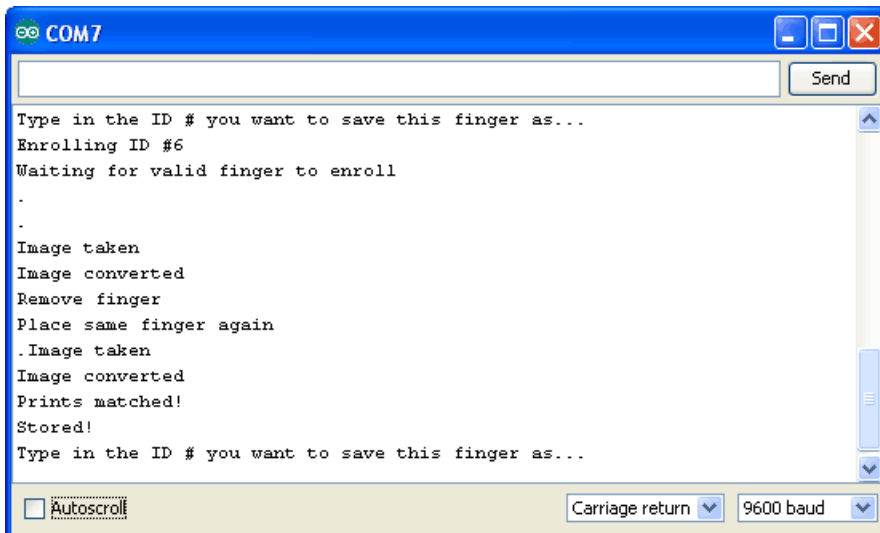
We did put together a simple sketch for enrolling a new finger via Arduino - its not as easy to use as the Windows program but it does work!

Run the **File→Examples→Adafruit\_Fingerprint→enroll** sketch and upload it to the Arduino, use the same wiring as above.

When you open up the serial monitor, it will ask for you to type in the ID to enroll - use the box up top to type in a number and click Send.



Then go through the enrollment process as indicated. When it has successfully enrolled a finger, it will print **Stored!**

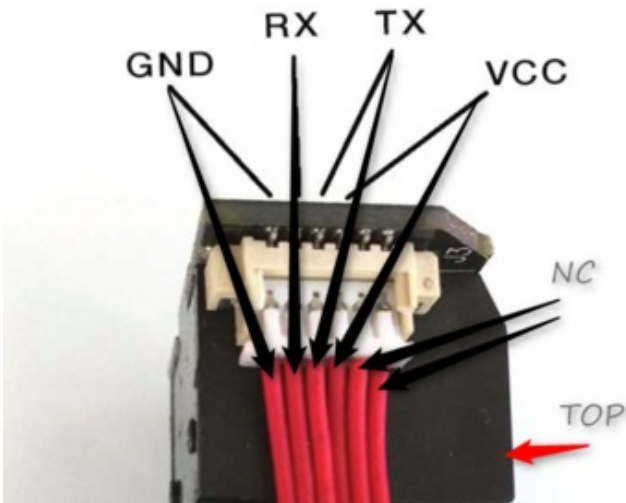


Don't forget to do a search test when you're done enrolling to make sure its all good!

## CircuitPython

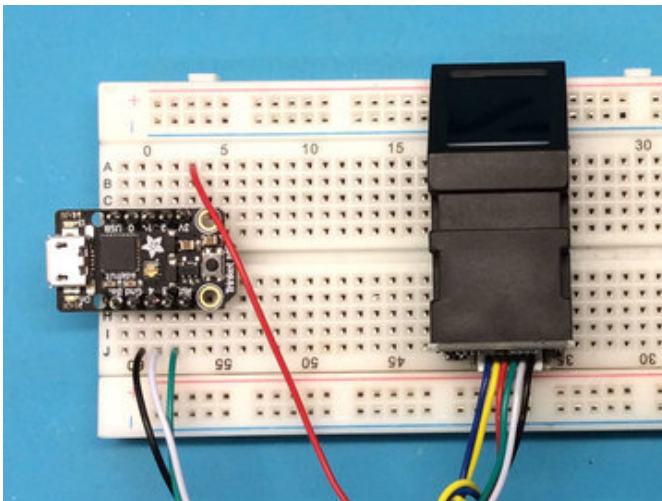


If your sensor has different wires, The first wire from the left should be the black wire ground, then the two data pins, RX is the white wire, TX is the green wire then the red power wire. You'll have to cut, strip and solder the wires.



If your sensor has all the same-color wires, The first wire from the left is ground, then the two data pins, then power. You'll have to cut, strip and solder the wires.

RX is the same as the White wire  
TX is the same as the Green wire



Every CircuitPython board has a hardware UART. Check the product page or look for **RX** and **TX** written on the board. Remember that the RX from the sensor goes to the TX on the board! If you have problems try swapping them, its a common mistake

## Installing Library

To use the Fingerprint sensor you'll need to install the [Adafruit CircuitPython Fingerprint](https://learn.adafruit.com/adafruit-optical-fingerprint-sensor) library on your CircuitPython

board.

First make sure you are running the [latest version of Adafruit CircuitPython](#) for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from [Adafruit's CircuitPython library bundle](#). Our introduction guide has [a great page on how to install the library bundle](#) for both express and non-express boards.

Remember for non-express boards like the, you'll need to manually install the necessary libraries from the bundle:

- `adafruit_fingerprint.mpy`

You can also download the `adafruit_fingerprint.mpy` from [its releases page on Github](#).

Before continuing make sure your board's lib folder or root filesystem has the `adafruit_fingerprint.mpy` file copied over.

Next [connect to the board's serial REPL](#) so you are at the CircuitPython `>>>` prompt.

## Usage

---

To demonstrate the usage of the sensor, we'll use the example python script included with the library. This sensor is fairly complex so its hard to run it just from the REPL.

Once you've installed the library, run this `main.py` example on your CircuitPython board.

```
import time
import board
import busio
from digitalio import DigitalInOut, Direction
import adafruit_fingerprint

led = DigitalInOut(board.D13)
led.direction = Direction.OUTPUT

uart = busio.UART(board.TX, board.RX, baudrate=57600)

finger = adafruit_fingerprint.Adafruit_Fingerprint(uart)

#####

def get_fingerprint():
    """Get a finger print image, template it, and see if it matches!"""
    print("Waiting for image...")
    while finger.get_image() != adafruit_fingerprint.OK:
        pass
    print("Templating...")
    if finger.image_2_tz(1) != adafruit_fingerprint.OK:
        return False
    print("Searching...")
    if finger.finger_fast_search() != adafruit_fingerprint.OK:
        return False
    return True

# pylint: disable=too-many-branches
```

```

def get_fingerprint_detail():
    """Get a finger print image, template it, and see if it matches!
    This time, print out each error instead of just returning on failure"""
    print("Getting image...", end="")
    i = finger.get_image()
    if i == adafruit_fingerprint.OK:
        print("Image taken")
    else:
        if i == adafruit_fingerprint.NOFINGER:
            print("No finger detected")
        elif i == adafruit_fingerprint.IMAGEFAIL:
            print("Imaging error")
        else:
            print("Other error")
        return False

    print("Templating...", end="")
    i = finger.image_2_tz(1)
    if i == adafruit_fingerprint.OK:
        print("Templated")
    else:
        if i == adafruit_fingerprint.IMAGEMESS:
            print("Image too messy")
        elif i == adafruit_fingerprint.FEATUREFAIL:
            print("Could not identify features")
        elif i == adafruit_fingerprint.INVALIDIMAGE:
            print("Image invalid")
        else:
            print("Other error")
        return False

    print("Searching...", end="")
    i = finger.finger_fast_search()
    if i == adafruit_fingerprint.OK:
        print("Found fingerprint!")
        return True
    else:
        if i == adafruit_fingerprint.NOTFOUND:
            print("No match found")
        else:
            print("Other error")
        return False

# pylint: disable=too-many-statements
def enroll_finger(location):
    """Take a 2 finger images and template it, then store in 'location'"""
    for fingerimg in range(1, 3):
        if fingerimg == 1:
            print("Place finger on sensor...", end="")
        else:
            print("Place same finger again...", end="")

        while True:
            i = finger.get_image()
            if i == adafruit_fingerprint.OK:
                print("Image taken")
                break
            elif i == adafruit_fingerprint.NOFINGER:
                print(".", end="")
            elif i == adafruit_fingerprint.IMAGEFAIL:

```



```

        print("Imaging error")
        return False
    else:
        print("Other error")
        return False

print("Templating...", end="")
i = finger.image_2_tz(fingerimg)
if i == adafruit_fingerprint.OK:
    print("Templated")
else:
    if i == adafruit_fingerprint.IMAGEMESS:
        print("Image too messy")
    elif i == adafruit_fingerprint.FEATUREFAIL:
        print("Could not identify features")
    elif i == adafruit_fingerprint.INVALIDIMAGE:
        print("Image invalid")
    else:
        print("Other error")
    return False

if fingerimg == 1:
    print("Remove finger")
    time.sleep(1)
    while i != adafruit_fingerprint.NOFINGER:
        i = finger.get_image()

print("Creating model...", end="")
i = finger.create_model()
if i == adafruit_fingerprint.OK:
    print("Created")
else:
    if i == adafruit_fingerprint.ENROLLMISMATCH:
        print("Prints did not match")
    else:
        print("Other error")
    return False

print("Storing model #%d..." % location, end="")
i = finger.store_model(location)
if i == adafruit_fingerprint.OK:
    print("Stored")
else:
    if i == adafruit_fingerprint.BADLOCATION:
        print("Bad storage location")
    elif i == adafruit_fingerprint.FLASHERR:
        print("Flash storage error")
    else:
        print("Other error")
    return False

return True

#####

def get_num():
    """Use input() to get a valid number from 1 to 127. Retry till success!"""
    i = 0

```

```

while (1 > 127) or (1 < 1):
    try:
        i = int(input("Enter ID # from 1-127: "))
    except ValueError:
        pass
    return i

while True:
    print("-----")
    if finger.read_templates() != adafruit_fingerprint.OK:
        raise RuntimeError('Failed to read templates')
    print("Fingerprint templates:", finger.templates)
    print("e) enroll print")
    print("f) find print")
    print("d) delete print")
    print("-----")
    c = input("> ")

    if c == 'e':
        enroll_finger(get_num())
    if c == 'f':
        if get_fingerprint():
            print("Detected #", finger.finger_id, "with confidence", finger.confidence)
        else:
            print("Finger not found")
    if c == 'd':
        if finger.delete_model(get_num()) == adafruit_fingerprint.OK:
            print("Deleted!")
        else:
            print("Failed to delete")

```

It's fairly long but it will help you set-up and test your sensor!

When you first start up, you should get something like this:

```

Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
main.py output:
-----
Fingerprint templates: [2]
e) enroll print
f) find print
d) delete print
-----
> |

```

If you get an error like `RuntimeError: Failed to read data from sensor` it means something went wrong - check your wiring and baud rate!

This menu system is fairly simple, you have three things you can do

- Enroll print - you will use your finger to take images and 'store' the model in the sensor
- Find print - determine whether a fingerprint is known and stored
- Delete print - clear out a model

## Enrolling Prints

Enrolling a finger print is easy. Type `e` to start the process. You'll need to select a location. The sensor can store up to

127 print locations. Pick a valid number, then place your finger *twice* to enroll.

```
Fingerprint templates: [2]
e) enroll print
f) find print
d) delete print
-----
> e
Enter ID # from 1-127: 0
Enter ID # from 1-127: 199
Enter ID # from 1-127: 5
Place finger on sensor.....Image taken
Templating...Templated
Remove finger
Place same finger again....Image taken
Templating...Templated
Creating model...Created
Storing model #5...Stored
-----
Fingerprint templates: [2, 5]
```

Note that after success, the **Fingerprint templates: [...]** printout will include the new template id.

If an error occurs, the sensor will give you an error, such as if the two prints don't match, or if it failed to store or generate a model:

```
> e
Enter ID # from 1-127: 4
Place finger on sensor.....Image taken
Templating...Templated
Remove finger
Place same finger again.....Image taken
Templating...Templated
Creating model...Prints did not match
```

## Finding Prints

---

Once you've enrolled fingerprints you can then test them. Run the **find** command, and try various fingers! Once the fingerprint id identified it will tell you the location number, in this case **#5**

```
-----
> f
Waiting for image...
Templating...
Searching...
Finger not found
-----
Fingerprint templates: [2, 5]
e) enroll print
f) find print
d) delete print
-----
> f
Waiting for image...
Templating...
Searching...
Detected # 5 with confidence 102
```

## Deleting Fingerprints

---

If you made a mistake you can remove fingerprint models from the database. For example, here's how to delete #5. Note the **Fingerprint templates: [...]** printout changes!

```
-----
Fingerprint templates: [2, 5] ←
e) enroll print
f) find print
d) delete print
-----
> d
Enter ID # from 1-127: 5
Deleted!
-----
Fingerprint templates: [2] ←
e) enroll print
f) find print
d) delete print
-----
```

## Downloads

---

- [Arduino interface library on github](#)
- [User Manual](#)
- [Datasheet \(its not really a great datasheet and its Chinese but its better than nothing\)](#)
- [English version of the User Manual](#)
- ["SFGDemo" Windows-only test software](#)

# **LAMPIRAN 3**

**Program Arduino Mega 2560: Pengaman Motor Menggunakan  
Sidik Jari**

```

#include <Adafruit_Fingerprint.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

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

SoftwareSerial mySerial(10, 11);

#define Relay_1 12
#define Relay_2 13

const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
int proses=0;
int pindah=0;
uint8_t num = 0;
char keypressed;
char id;
byte rowPins[ROWS] = {25, 24, 23, 22}; //connect to the row
pinouts of the keypad
byte colPins[COLS] = {29, 28, 27, 26}; //connect to the
column pinouts of the keypad

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

```

```

};

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins,
ROWS, COLS );

void setup()
{
  lcd.init();
  lcd.backlight();
  delay(1000);
  lcd.setCursor(5,0);
  lcd.print(" POLMAN ");
  lcd.setCursor(5,1);
  lcd.print(" BABEL ");
  lcd.begin(16,2);

  pinMode(12,OUTPUT);
  pinMode(13, OUTPUT);
  digitalWrite(12,HIGH);
  digitalWrite(13,HIGH);
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);

  lcd.setCursor(2,0);
  lcd.print("Sensor Oke");
  finger.begin(57600);
  finger.getTemplateCount();

  if (finger.verifyPassword())
  {
    lcd.clear();
  }
}

```



```

        lcd.setCursor(2,0);
        lcd.println("Sidik Ditemukan");
    }
    else
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.println("Sidik no temukan");
        while (1) { delay(1); }
    }
}

uint8_t readnumber(void)
{
    keypressed= keypad.getKey();
    num = keypressed;

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

void loop()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Welcome To Project ");
    lcd.setCursor(1,1);
    lcd.print("POLMAN BABEL ");
    delay(500);
}

```

```

    if (finger.templateCount < 2) //Kondisi Belum ada user
(Belum ada admin)
    {
        lcd.clear();
        lcd.setCursor(1, 0);
        lcd.print("Daftar Admin");
        lcd.setCursor(2, 1);
        lcd.print("Pilih nomor 0 ");
        delay(500);
        proses=1;
    }
    if (finger.templateCount > 1) //kalau sudah ada admin /
user
    {
        proses=2;
    }
while (proses == 1) //Mendaftarkan Admin
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Daftar Admin");
        lcd.setCursor(2,2);
        lcd.print("Pilih ID Admin          ");
        delay(2000);
        id = readnumber();
        EEPROM.write(0, id);
        if (id == 0)
        {
            return;
        }
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Enrolling ID #");
        while (! getFingerprintEnroll() );
    }

```

```

        proses=2;
    }

while (proses==2) //Masukan sidik jari
{
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("  Masukan Sidik");
    lcd.setCursor(1,1);
    lcd.print("      Jari");
    getFingerprintIDez();
}

while (proses==3) //Menu admin
{
    lcd.print("Daftar (A)          ");
    lcd.setCursor(0, 1);
    lcd.print("Hapus (B)          ");
    lcd.setCursor(0, 0);
    keypressed= keypad.getKey();

    if (keypressed=='A') proses=4;
    if (keypressed=='B') proses=9;
}

while (proses==4)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Daftar User");
    lcd.setCursor(0,1);
    lcd.print("Pilih ID User          ");

```

```

        delay(2000);
        id = readnumber()+1;
        if (id==0)
        {
            proses=4;
            return;
        }

        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Enrolling ID #");
        while (! getFingerprintEnroll() );
        proses=6;
    }

while (proses==9)
    {
        lcd.print("Hapus admin (A)          ");
        lcd.setCursor(0, 1);
        lcd.print("Hapus (B)          ");
        lcd.setCursor(0, 0);
        keypressed= keypad.getKey();

        if (keypressed=='A') proses=8;
        if (keypressed=='B') proses=5;
    }

while (proses==5)
    {
        lcd.clear();
        lcd.print("masukan id          ");
        lcd.setCursor(0,0);
        proses=7;
    }

```

```

while (proses==7)
{
    {
        proses=2;
    }

    id = readnumber()+1;
    if (id == 0) { // ID #0 not allowed, try again!
        return;
    }

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Deleting ID #");
    lcd.println(id);

    deleteFingerprint(id);
    delay(2000);
    proses=2;
}

//Delete Admin
while(proses==8)
{
    int i;
    for(i=48;i<58;i++)
    {
        //Serial.println("Please type in the ID # (from 1
to 127) you want to delete...");
        //uint8_t id = readnumber();
        if (i == 0)
        {
            // ID #0 not allowed, try again!
            lcd.setCursor(0,0);

```

```

        lcd.print("i=");
        return;
    }
    deleteFingerprint(i);
    //delay(500);
}
delay(500);
proses=1;
}

while(proses==6) //Menghidupkan Sistem
{
    lcd.setCursor(0,0);
    lcd.print("  Motor ON  ");
    lcd.setCursor(1,1);
    lcd.print("Please Savety ");
    digitalWrite(Relay_1,LOW);
    digitalWrite(Relay_2,LOW);
    delay(500);
}
}

uint8_t getFingerprintEnroll()
{
    int p = -1;
    Serial.print("Waiting for valid finger to enroll as #");
Serial.println(id);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Menunggu sidik  ");
    lcd.setCursor(0,0);
    lcd.print("Masukan Sidik  ");
    lcd.setCursor(0,1);
    lcd.print("  jari  ");
}

```

```

lcd.print(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;
  }
  if (keypressed=='C'){Serial.println("C
ditekan!");proses=4;}
}
} // 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.setCursor(1,0);
lcd.println(" Angkat Jari      ");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("ID ");
lcd.println(id);

```

```

p = -1;
Serial.println(" Place same finger again");
lcd.clear();
lcd.setCursor(1,0);
lcd.println(" TEMPATKAN LAGI      ");
lcd.setCursor(2,1);
lcd.println("SIDIK JARI      ");

```



```

while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            lcd.clear();
        lcd.setCursor(1,0);
        lcd.println(" Tersimpan ");
            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;
    }
}

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);
lcd.clear();
lcd.setCursor(0,0);
lcd.println("Creating model for #");
lcd.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);

```

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("ID");
lcd.println(id);
delay(1000);

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 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;
    }
}

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);
    Serial.println("IDEZ");

    if(finger.fingerID==EEPROM.read(0)) //Data Admin
    {
        for(int i=0;i<100;i++)
        {
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Pengaturan (B)");
            delay(100);
        }
    }
}

```

```
keypressed= keypad.getKey();
if (keypressed=='B')
{
    proses=3;
    break;
}
    delay(1);

}
    if (proses!=3)
    proses=6;
}
if(finger.fingerID > 49) // Data User
{
    proses=6;
}
    else
{
    Serial.println("Dak De");
}
return finger.fingerID;
}
```