



**RANCANG BANGUN HARDWARE SISTEM KEAMANAN BERLAPIS
KUNCI PINTU RUMAH BERBASIS E-KTP DAN FINGERPRINT**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Studi
Jenjang Program Diploma Tiga

Oleh:

Nama : Via Aditia Stefanni

NIM : 18040016

**PROGRAM STUDI DIII TEKNIK KOMPUTER
POLITEKNIK HARAPAN BERSAMA TEGAL
2021**

HALAMAN PERNYATAAN KEASLIAN

Yang bertandatangan di bawah ini:

Nama : Via Aditia Stefanni
NIM : 18040016
Jurusan / Program Studi : Teknik Komputer
Jenis Karya : Tugas Akhir

Adalah mahasiswa Program Studi DIII Teknik Komputer Politeknik Harapan Bersama, dengan ini saya menyatakan bahwa laporan Tugas Akhir yang berjudul **“RANCANG BANGUN HARDWARE SISTEM KEAMANAN BERLAPIS KUNCI PINTU RUMAH BERBASIS E-KTP DAN FINGERPRINT”** Merupakan hasil pemikiran dan kerjassama sendiri secara orisinal dan saya susun secara mandiri dan tidak melanggar kode etika hak karya cipta. Pada pelaporan Tugas Akhir ini juga bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik tertentu di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau disebutkan dalam daftar pustaka. Apabila di kemudian hari ternyata Laporan Tugas Akhir ini terbukti melanggar kode etik karya cipta atau merupakan karya yang dikategorikan mengandung unsur plagiarisme, maka saya bersedia untuk melakukan penelitian baru dan menyusun laporannya sebagai Laporan Tugas Akhir, sesuai dengan ketentuan yang berlaku. Demikian pernyataan ini saya buat dengan sebenarnya dan sesungguhnya.

Tegal, 20 Mei 2021

Yang membuat pernyataan,



Via Aditia Stefanni
18040016

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPERLUAN AKADEMIS**

Sebagai civitas akademik PoliTeknik Harapan Bersama Tegal, Kami yang bertanda tangan dibawah ini :

Nama : Via Aditia Stefanni
NIM : 18040016
Jurusan / Program Studi : Teknik Komputer
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada PoliTeknik Harapan Bersama Tegal **Hak Bebas Royalti Non eksklusif** (*Non-exclusive Royalty Free Right*) atas Tugas Akhir kami yang berjudul :

**“RANCANG BANGUN HARDWARE SISTEM KEAMANAN KUNCI
PINTU RUMAH BERBASIS E-KTP DAN FINGERPRINT “**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti *Noneksklusif* ini Politeknik Harapan Bersama Tegal berhak menyimpan, mengalih media/formatkan, mengolah dalam bentuk pangkalan data (database), merawat dan mempublikasikan Tugas Akhir kami selama tetap mencantumkan nama kami sebagai penulis/pencipta dan pemilik Hak Cipta.

Demikian pernyataan ini kami buat dengan sebenarnya.

Dibuat di : Tegal

Pada Tanggal : 20 Mei 2021

Yang menyatakan



Via Aditia Stefanni
18040016

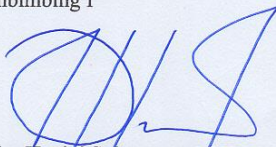
HALAMAN PERSETUJUAN

Tugas Akhir (TA) yang berjudul “RANCANG BANGUN HARDWARE SISTEM KEAMANAN KUNCI PINTU RUMAH BERLAPIS BERBASIS E-KTP DAN FINGERPRINT” yang disusun oleh Via Aditia Stefanni, NIM 18040016 telah mendapat persetujuan pembimbing dan siap dipertahankan di depan tim penguji Tugas Akhir (TA) Program Studi D-III Teknik Komputer PoliTeknik Harapan Bersama Tegal.

Tegal, 20 Mei 2021

Menyetujui,

Pembimbing I



Arfan Haqini Syasmoro, M.Kom
NIPY. 02.009.054

Pembimbing II



Achmad Sutanto, S.Kom
NIPY. 11.012.128

HALAMAN PENGESAHAN




Judul : RANCANG BANGUN HARDWARE SISTEM
KEAMANAN BERLAPIS KUNCI PINTU RUMAH
BERBASIS E-KTP DAN FINGERPRINT

Program Studi : Teknik Komputer
Jenjang : Diploma III

**Dinyatakan LULUS setelah dipertahankan di depan Tim Penguji Tugas Akhir
Program Studi DIII Teknik Komputer Politeknik Harapan Bersama Tegal**

Tegal, 20 Mei 2021

Tim Penguji :

Nama	Tanda Tangan
1. Ketua Penguji : Rais, S.pd., M.Kom	1. 
2. Anggota I : Eko Budihartono, ST, M.Kom	2. 
3. Anggota II : Achmad Sutanto, s.kom	3. 

Mengetahui,
Ketua Program Studi DIII Teknik Komputer,
PoliTeknik Harapan Bersama Tegal


Rais, S.Pd., M.Kom
NIPY. 07.011.083

MOTTO

- Selalu awali apapun dengan membaca Bismillah..
- Selalu niatkan apapun untuk beribadah dan selalu ikhlas.
- Jangan takut untuk memulai Kembali, atau engkau akan terperangkap dalam kesalahan yang sama.
- Bukan seberapa besar kesuksesan, Tapi menikmati prosesnya disetiap sudut kecilnya.
- Sesungguhnya sesudah kesulitan akan ada kemudahan (Qs. Al-Insyirah : 6).
- Allah tidak akan membebani seseorang melainkan sesuai dengan kesanggupannya (Qs. Al-baqarah : 286).

PERSEMBAHAN

Laporan Tugas Akhir ini kami Persembahkan kepada:

1. Allah SWT, karena hanya atas izin dan karunia Nya lah maka laporan ini dapat dibuat dan selesai pada waktunya.
2. Kepada kedua orang tua yang telah memberikan dukungan moril maupun materi serta do'a yang tiada hentinya.
3. Bapak Rais S.Pd M..Kom selaku Ka Prodi DIII Teknik Komputer Politeknik Harapan Bersama.
4. Bapak Arfan Haqiqi Sulasmoro, M.Kom selaku pembimbing I dan Bapak Achmad Sutanto, S.Kom selaku pembimbing II yang selama ini telah tulus dan ikhlas meluangkan waktu untuk membimbing dalam pembuatan tugas akhir ini.
5. Seluruh keluarga yang senantiasa memberikan dukungan semangat senyum dan doa untuk keberhasilan ini.
6. Sahabat dan teman perjuangan karena semangat dan tekad yang besar berasal dari kebersamaan yang besar juga.
7. Rekan-rekan seperjuangan Angkatan 2018 yang sangat luar biasa.

ABSTRAK

Sistem pengunci pintu saat ini masih menggunakan kunci konvensional, sehingga kurang efisien untuk rumah dengan banyak pintu karena terlalu banyak kunci yang harus dibawa ketika akan bepergian dari rumah dan seringkali pemilik rumah lupa bahkan kehilangan kunci, selain itu kunci konvensional mudah dibuka oleh pencuri. Sehingga diperlukan kunci yang lebih praktis dan efisien, dari masalah tersebut penulis mempunyai gagasan untuk menghasilkan alat pengaman pintu yang aman dan praktis berbasis *RFID* dengan memanfaatkan E-KTP sebagai *RFID tag* sebagai pengaman pintu rumah. Pada Tugas Akhir ini telah dirancang dan direalisasikan sebuah alat bantu untuk memonitor keamanan kunci rumah berbasis *internet of things*. Dengan menggunakan *NodeMCU* dan *Arduino UNO* sebagai *microcontroller* komponen utama, *fingerprint* sebagai sensor sidik jari, *solenoid door lock* sebagai kunci elektronik, *relay* sebagai penghambat arus listrik dan *Telegram* sebagai notifikasi. Hasil pengujian menunjukkan alat ini dapat bekerja sesuai dengan program yang telah diatur sebelumnya, *fingerprint* yang digunakan diharapkan dapat membatasi orang yang dapat memasuki rumah, serta mampu menampilkan pada lcd dan akan memberikan notifikasi yang dikirimkan ke aplikasi *telegram* diharapkan dapat mencatat identitas pembuka pintu dan waktu saat pintu terbuka. Sehingga orang-orang yang keluar masuk rumah tersebut dapat tercatat dengan baik.

Kata Kunci: E-KTP, *solenoid door lock*, *fingerprint*.

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT, Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya hingga terselesaikannya laporan Tugas Akhir dengan judul **“RANCANG BANGUN HARDWARE SISTEM KEAMANAN BERLAPIS KUNCI PINTU RUMAH BERBASIS E-KTP DAN FINGERPRINT”** ini selesai tepat pada waktunya.

Tugas Akhir merupakan suatu kewajiban yang harus dilaksanakan untuk memenuhi salah satu syarat kelulusan dalam mencapai derajat Ahli Madya Komputer pada program studi D III Teknik Komputer Politeknik Harapan Bersama Tegal. Selama melaksanakan penelitian dan kemudian tersusun dalam laporan Tugas Akhir ini, banyak pihak yang telah memberikan bantuan, dukungan dan bimbingan.

Pada kesempatan ini, tidak lupa diucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Nizar Suhendra , SE, MPP selaku Direktur Politeknik Harapan Bersama Tegal.
2. Bapak Rais, S. Pd, M. Kom selaku ketua Program Studi D III Teknik Komputer Politeknik Harapan Bersama Tegal.
3. Bapak Arfan Haqiqi Sulasmoro, M.Kom selaku pembimbing I.
4. Bapak Achmad Sutanto, S.Kom selaku pembimbing II.
5. Dea Rizki Y, Alfina Inayatuzzulfa yang telah membantu membuat tugas akhir ini.
6. Semua Pihak yang telah mendukung, membantu, serta mendoa kan penyelesaian tugas akhir ini.

Semoga laporan Tugas Akhir ini dapat memberikan sumbangan untuk pengembangan ilmu pengetahuan dan teknologi.

Tegal, 20 Mei 2021

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERNYATAAN KEASLIAN	Error! Bookmark not defined.
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI .	Error! Bookmark not defined.
HALAMAN PERSETUJUAN	Error! Bookmark not defined.
HALAMAN PENGESAHAN	Error! Bookmark not defined.
MOTTO.....	vi
PERSEMBAHAN.....	vii
ABSTRAK	viii
KATA PENGANTAR	ix
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan.....	3
1.5. Manfaat.....	4
1.5.1. Bagi Masyarakat	4
1.5.2. Bagi Politeknik Harapan Bersama	4
1.5.3. Bagi Mahasiswa	5
1.6. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	7
2.1. Penelitian Terkait	7
2.2. Landasan Teori.....	8
2.2.1. <i>Fingerprint</i>	8

2.2.2. <i>NodeMCU</i>	9
2.2.3. <i>Solenoid Door Lock</i>	10
2.2.4. <i>Adaptor</i>	10
2.2.5. <i>Relay</i>	11
2.2.6. <i>LCD OLED</i>	11
2.2.7. <i>Flowchart</i>	12
2.2.8. <i>RFID Module</i>	14
2.2.9. <i>Bread Board</i>	15
2.2.10. <i>Kabel Jumper</i>	16
2.2.11. <i>Arduino UNO</i>	17
BAB III METODOLOGI PENELITIAN	18
3.1. Prosedur Penelitian	18
3.1.1. <i>Rencana / Planning</i>	18
3.1.2. <i>Data Analisis</i>	18
3.1.3. <i>Desain</i>	19
3.1.4. <i>Coding</i>	19
3.1.5. <i>Testing</i>	20
3.1.6. <i>Implementation</i>	20
3.1.7. <i>Maintenanc</i>	20
3.2. Metode Pengumpulan Data	20
3.2.1. <i>Studi Literatur</i>	20
3.2.2. <i>Sumber Data</i>	20
3.3. Waktu dan Tempat Penelitian	21
3.3.1. <i>Waktu</i>	21
3.3.2. <i>Tempat Pelaksanaan</i>	22
BAB IV ANALISA DAN PERANCANGAN SISTEM	23
4.1. Analisa Permasalahan	23
4.2. Analisa Kebutuhan Sistem	24
4.2.1. <i>Analisa Perangkat Keras atau Hardware</i>	24
4.2.2. <i>Analisa Perangkat Lunak atau Software</i>	25
4.3. Perancangan Sistem	25
4.3.1. <i>Diagram Blok Sistem</i>	26
4.3.2. <i>Flowchart Sistem</i>	27

4.5. Desain Input/Output	29
BAB V HASIL DAN PEMBAHASAN.....	32
5.1. Implementasi Sistem	32
5.1.1. Implementasi perangkat keras	32
5.1.2. Implementasi perangkat lunak	33
5.2. Hasil dan Pengujian.....	34
BAB VI KESIMPULAN.....	37
6.1. Kesimpulan	37
6.2. Saran.....	38
DAFTAR PUSTAKA	39

DAFTAR TABEL

	Halaman
Tabel 2.1. Simbol Flowchart	12
Tabel 4.1. Rangkaian Keseluruhan	28
Tabel 5.1. Hasil Pengujian	34

DAFTAR GAMBAR

	Halaman
Gambar 2.1. <i>Fingerprint</i>	9
Gambar 2.2. <i>NodeMCU</i>	9
Gambar 2.3. <i>Solenoid Door Lock</i>	10
Gambar 2.4. <i>Adaptor</i>	11
Gambar 2.5. <i>Relay</i>	11
Gambar 2.6. <i>LCD OLED</i>	12
Gambar 2.7. <i>RFID Module</i>	15
Gambar 2.8. <i>Bread Board</i>	16
Gambar 2.9. <i>Kabel Jumper</i>	17
Gambar 2.10. <i>Arduino UNO</i>	17
Gambar 4.1. <i>Diagram Blok</i>	26
Gambar 4.2. <i>Flowchart Sistem</i>	27
Gambar 4.3. <i>Desain Input/Output</i>	29

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Kesediaan Membimbing	A-1
Lampiran 2 Coding Arduino IDE.....	B-1

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pada saat ini keamanan rumah masih menggunakan sistem penguncian manual yaitu dengan menggunakan kunci *konvensional*. Penggunaan kunci konvensional kurang praktis pada zaman sekarang, karena pemilik rumah harus membawa banyak kunci ketika akan bepergian dari rumah dan sering kali pemilik rumah lupa bahkan kehilangan kunci. Penggunaan kunci *konvensional* juga mudah dibuka oleh pencuri karena semakin berkembang cara pencuri untuk membuka pintu rumah [1]. Hal itu menjadikan kunci pintu *konvensional* menjadi kurang efektif dan kurang aman jika dibandingkan dengan pintu digital atau *smart door lock* berbasis *E-ktp* dan *fingerprint*.

E-KTP dapat digunakan sebagai RFID *tag* karena didalamnya terdapat *chip* yang menyimpan nomor ID unik, alat pengaman pintu ini memanfaatkan E-KTP untuk membuka pintu. E-KTP berfungsi sebagai *transponder* dan tergolong dalam tag pasif karena tidak memiliki catu daya sendiri, catu dayanya berasal dari pancaran gelombang RFID *reader*. RFID *reader* berfungsi untuk membaca nomor ID pada E-KTP. Data yang tersimpan dalam *chip* akan terkirim atau terbaca melalui gelombang radio setelah tag-antena menerima pancaran gelombang radio dari *reader*-antena kemudian data akan dikirim ke *mikrokontroler*.

Smart door lock atau kunci pintu pintar, menggunakan perangkat digital untuk mendapatkan akses kedalam ruangan. Perangkat digital yang biasanya dipakai berupa RFID, *bluetooth*, *fingerprint*, atau aplikasi *smartphone*. Pada kunci pintar menggunakan *fingerprint* terdapat beberapa kelebihan jika dibandingkan dengan kunci pintu *konvensional*, diantaranya adalah lebih efisien karena tidak menggunakan anak kunci, sehingga tidak akan ada kasus kehilangan kunci atau ketinggalan, dan lebih aman, karena tidak ada sidik jari yang sama. *Fingerprint* yang digunakan diharapkan dapat membatasi orang yang dapat memasuki rumah, dan *notifikasi* yang dikirimkan ke aplikasi *telegram* diharapkan dapat mencatat identitas pembuka pintu dan waktu saat pintu terbuka. Sehingga orang – orang yang keluar masuk rumah tersebut dapat tercatat dengan baik.

Dari permasalahan yang ada muncul gagasan untuk membuat suatu terobosan baru yaitu alat “**RANCANG BANGUN HARDWARE SISTEM KEAMANAN BERLAPIS KUNCI PINTU RUMAH BERBASIS E-KTP DAN FINGERPRINT**” untuk mempermudah pemilik rumah dalam membuka dan mengunci pintu rumah serta dapat terhindar dari pencurian maupun kehilangan kunci tersebut. Alat ini dapat bekerja sesuai dengan program yang telah diatur sebelumnya, *fingerprint* yang digunakan diharapkan dapat membatasi orang yang dapat memasuki rumah, serta mampu menampilkan pada *lcd*. Diharapkan dapat mencatat identitas pembuka pintu dan waktu saat pintu terbuka. Sehingga orang-orang yang keluar masuk rumah tersebut dapat tercatat dengan baik.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah diatas, diperoleh rumusan masalah yaitu, Bagaimana merancang dan membangun suatu *hardware* sistem keamanan berlapis kunci pintu rumah berbasis E-KTP dan *fingerprint*.

1.3. Batasan Masalah

Agar tidak meluas dari maksud dan tujuan penelitian ini, maka permasalahannya dibatasi sebagai berikut:

1. *Mikrocontroller* menggunakan NodeMCU.
2. Alat ini di aplikasikan pada pintu rumah.
3. *Database* menggunakan MYSQL.
4. Sistem dibuat dengan bentuk *Prototype*.
5. Interface menggunakan *Telegram*.
6. Data menggunakan *RFID* E-KTP dan *Fingerprint*.
7. Menggunakan *Solenoid Door Lock* sebagai pengunci Pintu.

1.4. Tujuan

Berdasarkan dari rumusan masalah diatas, tujuan dari penelitian ini adalah merancang sebuah *Hardware* Sistem Keamanan Berlapis Kunci Pintu Rumah Berbasis E-KTP dan *Fingerprint* Agar dapat dikembangkan menjadi *system* nyata.

1.5. Manfaat

Sedangkan manfaat dari penelitian ini adalah:

1.5.1. Bagi Masyarakat

1. Memberikan keamanan pada rumah agar hanya orang-orang tertentu yang dapat memasukinya.
2. Dapat memberikan keamanan pada rumah yang sederhana, praktis, *efektif* dan lebih bersifat *kondusif*.
3. Memberikan kemudahan pada pemilik rumah dalam membuka dan mengunci pintu rumah.
4. Dapat memberikan manfaat bagi pemilik rumah ketika kehilangan maupun ketinggalan kunci.
5. Dapat terhindar dari adanya kejahatan kebobolan rumah.

1.5.2. Bagi Politeknik Harapan Bersama

1. Menerapkan pengalaman yang telah diperoleh selama perkuliahan.
2. Sebagai masukan untuk mengevaluasi sejauh mana mahasiswa memahami materi apa yang di dapat selama perkuliahan.
3. Mendapat masukan yang berguna untuk menyempurnakan kurikulum yang sesuai dengan kebutuhan tugas akhir.

1.5.3. Bagi Mahasiswa

1. Mahasiswa mampu mengasah kemampuan dan berkarya sesuai dengan pelajaran yang didapatkan selama perkuliahan.

2. Mahasiswa dapat membantu menyelesaikan permasalahan di masyarakat.

1.6. Sistematika Penulisan

Untuk memberikan gambaran yang jelas mengenai pokok pembahasan dalam penulisan tugas akhir ini, akan dilakukan secara sistematis dengan membagi kedalam enam bab yaitu:

BAB I PENDAHULUAN

Pada bab ini menerangkan tentang latar belakang, pokok permasalahan, batasan masalah, tujuan dan manfaat.

BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan tentang teori – teori terkait, dan Landasan teori yang mendukung dalam penulisan tugas akhir.

BAB III METODOLOGI PENELITIAN

Pada bab ini berisi tentang data-data yang diperlukan dalam penelitian, alat dan bahan yang digunakan, dan tahapan–tahapan dalam mengolah data tersebut. Dan juga waktu dan tempat penelitian.

BAB IV ANALIS DAN PERANCANGAN SISTEM

Pada bab ini berisi tentang analisa permasalahan, analisa kebutuhan sistem, perancangan sistem, dan Desain Input-Output dalam pembuatan tugas akhir.

BAB V HASIL DAN PEMBAHASAN

Pada bab ini berisi tentang implementasi sistem dan data hasil pengujian, untuk selanjutnya diolah dan disajikan dalam bentuk gambar dan tabel.

BAB VI KESIMPULAN DAN SARAN

Dalam bab ini akan disajikan secara garis besar hasil dari pengujian dan disertai dengan kesimpulan.

DAFTAR PUSTAKA

Bab ini berisi tentang semua referensi yang menjadi panduan penulisan dalam menyusun tugas akhir ini.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian yang dilakukan oleh Anggidhira Widyananda 2013 dalam penelitiannya yang berjudul Perancangan dan *Implementasi* Kunci Pintu Otomatis Dengan *Fingerprint* Berbasis *Mikrokontroler*, pada saat ini teknologi keamanan semakin berkembang, salah satunya adalah sistem keamanan pada pintu. Terdapat banyak sistem keamanan pada pintu yang telah dibuat seperti pintu dengan alarm otomatis, *password* angka dan lain sebagainya. Namun terkadang *password* dengan angka dirasa masih belum cukup dalam menjaga keamanan pada pintu. Maka harus digunakan *password* yang lebih spesifik dan mudah diingat sehingga hanya bisa dibuka oleh orang-orang yang terdaftar pada sistem keamanan tersebut [2].

Penelitian yang dilakukan oleh Tatik Juwariyah 2019 dalam jurnal penelitiannya yang berjudul Purwa Rupa Sistem Pengaman Sepeda Motor Berbasis *IoT*, Hasil penelitian tersebut adalah data rata-rata waktu respon sistem ketika memproses suatu data baik berupa sidik jari pada modul *fingerprint*. Penelitian terkait *IoT* (*Internet of Things*) adalah penelitian atau pengendalian pintu ruang server berbasis *mikrontroler* yang dirancang untuk mengendalikan perangkat elektronik pintu ruang server. Pengendalian tersebut tampil dalam bentuk waktu dan hari melalui web [3].

Penelitian yang telah dilakukan oleh Arifandi (2019) tentang *Prototype Sistem Keamanan Pintu Menggunakan Sidik Jari Berbasis Arduino Uno Atmega328 Dan Sms Gateway*. Penelitian tersebut menjelaskan Sistem pengaman pintu menggunakan sidik jari (*Fingerprint*) dapat mengidentifikasi penghuni rumah dan sebuah modul GSM yang dapat mengirimkan informasi kepada pemilik rumah berupa pesan text, dirancang untuk memberikan solusi pada rumah yang ditinggalkan oleh pemiliknya. Hasil pengujian yang telah dilakukan menunjukkan akses untuk membuka pintu menggunakan sidik jari (*Fingerpritrn*) dari luar dapat berjalan dengan baik. *Limit Switch* dan Modul GSM dapat bekerja dengan baik sesuai program yang telah diperintahkan, dan pemilik rumah dapat dengan mudah mendapatkan informasi ketika seseorang menjebol pintu secara paksa [4].

2.2. Landasan Teori

2.2.1. *Fingerprint*

Fingerprint adalah sebuah alat elektronik yang menerapkan sensor *scanning* untuk mengetahui sidik jari seseorang guna keperluan *verifikasi* identitas. Sensor *Fingerprint* seperti ini digunakan pada beberapa peralatan elektronik seperti *smartphone*, pintu masuk, alat absensi karyawan dan berbagai macam peralatan elektronik yang membutuhkan tingkat keamanan yang tinggi, dan hanya bisa diakses oleh orang-orang tertentu saja. Sebelum sensor *Fingerprint* ditemukan, dahulu sebuah data diamankan dengan menggunakan

password atau ID, ada juga yang menggunakan pola guna mengamankan suatu data [5].



Gambar 2.1. *Fingerprint*

2.2.2. *NodeMCU*

NodeMCU adalah sebuah *open source* platform *IoT* (*Internet of Thing*) dan pengembangan kit yang menggunakan bahasa pemrograman Lua untuk membantu makers dalam membuat *prototype* produk *IoT* (*Internet of Thing*) atau bisa dengan memakai sketch dengan *arduino IDE*. Dengan ukurannya yang kecil, *board* ini sudah dilengkapi dengan fitur wifi dan firmwarena yang bersifat *opensource*. *GPIO* (*General Purpose Input Output*) adalah pin generik pada sirkuit terpadu (*chip*) dapat dikontrol dan diprogram. *GPIO* bisa full kontrol lewat jaringan wifi [6].



Gambar 2.2. *NodeMCU*

2.2.3. *Solenoid Door Lock*

Solenoid merupakan perangkat elektromagnetik yang dapat mengubah energi listrik menjadi energi gerakan. Energi gerakan yang dihasilkan oleh *selonoid* biasanya hanya gerakan mendorong dan menarik. Pada dasarnya solenoid hanya terdiri dari sebuah kumparan listrik (*electrical coil*) yang dililitkan di sekitar tabung silinder dengan *aktuator ferro-magnetic* atau sebuah *Plunger* yang bebas bergerak masuk dan keluar dari bodi kumparan [7].



Gambar 2.3. *Solenoid Door Lock*

2.2.4. **Adaptor**

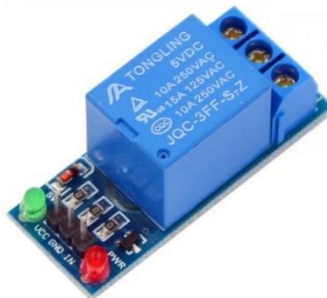
Adaptor yaitu piranti elektronik yang bisa mengubah tegangan listrik (AC) yang tinggi jadi tegangan listrik (DC) yang rendah, namun ada juga jenis adaptor yang bisa mengubah tegangan listrik yang rendah jadi tegangan listrik yang tinggi, dan ada banyak lagi macam-macam adaptor [8].



Gambar 2.4. Adaptor

2.2.5. *Relay*

Relay adalah suatu peranti yang bekerja berdasarkan *elektromagnetik* untuk menggerakkan sejumlah *kontaktor* yang tersusun atau sebuah saklar elektronis yang dapat dikendalikan dari rangkaian elektronik lainnya dengan memanfaatkan tenaga listrik sebagai sumber energinya [9].

Gambar 2.5. *Relay*

2.2.6. *LCD OLED*

Organic Light-Emitting Diode (OLED) adalah merupakan sebuah semikonduktor sebagai pemancar cahaya yang terbuat dari lapisan organik. OLED digunakan dalam teknologi elektroluminensi, seperti pada tampilan layar atau display [10].

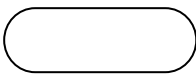

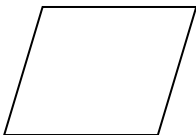




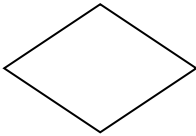
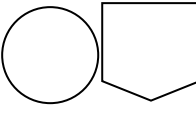
Gambar 2.6. LCD OLED

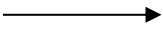
2.2.7. Flowchart

Flowchart adalah bagan alir yang menggambarkan tentang urutan langkah jalannya suatu program dalam sebuah bagan dengan simbol-simbol bagan yang sudah ditentukan [11].

Tabel 2.1. Simbol *Flowchart*

Simbol	Keterangan
	Terminator / Terminal Merupakan simbol yang digunakan untuk menentukan state awal dan state akhir suatu flowchart program.
	Preparation / Persiapan Merupakan simbol yang digunakan untuk mengidentifikasi variabel-variabel yang akan digunakan dalam program. Bisa berupa pemberian harga awal, yang ditandai dengan nama variabel sama dengan ('') untuk tipe string, (0) untuk tipe numeric, (.F./T.) untuk tipe Boolean dan ({//}) untuk tipe tanggal.
	Input output / Masukan keluaran Merupakan simbol yang digunakan untuk memasukkan nilai dan untuk menampilkan nilai dari suatu variabel. Ciri dari simbol ini adalah tidak ada operator baik operator aritmatika hingga operator perbandingan. Yang membedakan antara masukan dan keluaran adalah jika Masukan cirinya adalah variabel yang ada didalamnya belum mendapatkan operasi dari operator tertentu,

Simbol	Keterangan
	apakah pemberian nilai tertentu atau penambahan nilai tertentu. Adapun ciri untuk keluaran adalah biasanya variabelnya sudah pernah dilakukan pemberian nilai atau sudah dilakukan operasi dengan menggunakan operator tertentu.
	Process / Proses Merupakan simbol yang digunakan untuk memberikan nilai tertentu, apakah berupa rumus, perhitungna <i>counter</i> atau hanya pemrian nilai tertentu terhadap suatu variabel.
	Predefined Process / Proses Terdefinisi Merupakan simbol yang penggunaannya seperti link atau menu. Jadi proses yang ada didalam simbol ini harus dibuatkan penjelasan flowchart programnya secara tersendiri yang terdiri dari terminator dan diakhiri dengan terminator.
	Decision / simbol Keputusan Digunakan untuk menentukan pilihan suatu kondisi (Ya atau tidak). Ciri simbol ini dibandingkan dengan simbol-simbol flowchart program yang lain adalah simbol keputusan ini minimal keluaran arusnya 2 (dua), jadi Jika hanya satu keluaran maka penulisan simbol ini adalah salah, jadi diberikan pilihan jika kondisi bernilai benar (<i>true</i>) atau salah (<i>false</i>). Sehingga jika nanti keluaran dari simbol ini adalah lebih dari dua bisa dituliskan. Khusus untuk yang keluarannya dua, harus diberikan keterangan Ya dan Tidaknya pada arus yang keluar.
	Connector Konektor dalam satu halaman merupakan penghubung dari simbol yang satu ke simbol yang lain. Tanpa harus menuliskan arus yang panjang. Sehingga akan lebih menyederhanakan dalam penggambaran aliran programnya, simbol konektornya adalah lingkaran, sedangkan Konektor untuk menghubungkan antara simbol yang satu dengan simbol yang lainnya yang berbeda halaman, maka menggunakan simbol konektor yang segi lima, dengan diberikan identitasnya,

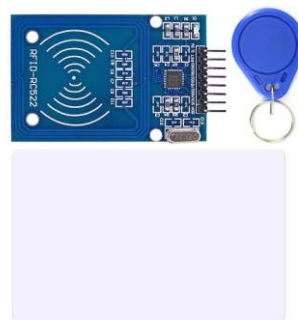
Simbol	Keterangan
	bisa berupa charater alpabet A – Z atau a – z atau angka 1 sampai dengan 9.
	Arrow / Arus Merupakan simbol yang digunakan untuk menentukan aliran dari sebuah flowchart program. Karena berupa arus, maka dalam menggambarkan arus data harus diberi simbol panah.

2.2.8. *RFID Module*

Radio Frequency Identification (RFID) adalah teknologi identifikasi yang mudah digunakan. *RFID* mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. *RFID* disediakan dalam *device* yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka *RFID* dapat menyediakan tingkat keamanan yang tinggi. Pada 14indak *RFID* umumnya, *tag* atau *transponder* ditempelkan pada suatu objek. Setiap *tag* dapat membawa informasi yang unik, di antaranya: *serial number*, model, warna, dan data lain dari objek tersebut. Ketika *tag* ini melalui medan yang dihasilkan oleh pembaca *RFID* yang *kompatibel*, *tag* akan mentransmisikan informasi yang ada pada *tag* kepada pembaca *RFID*, sehingga proses identifikasi objek dapat dilakukan [12].

Sistem *RFID* terdiri dari empat komponen, di antaranya sebagai berikut:

1. *Tag*: Ini adalah *device* yang menyimpan informasi untuk identifikasi objek. *Tag RFID* sering juga disebut sebagai transponder
2. Antena: untuk mentransmisikan sinyal *frekuensi* radio antara pembaca *RFID* dengan *tag RFID*. Pembaca *RFID*: adalah *device* yang kompatibel dengan *tag RFID* yang akan berkomunikasi secara *wireless* dengan *tag*.
3. *Software* Aplikasi: adalah aplikasi pada sebuah *workstation* atau PC yang dapat membaca data dari tag melalui pembaca *RFID*. Baik *tag* dan pembaca *RFID* dilengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

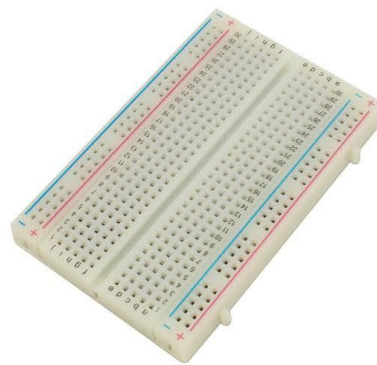


Gambar 2.7. *RFID Module*

2.2.9. *BreadBoard*

BreadBoard atau disebut juga dengan *project board* yang digunakan untuk membuat rangkaian elektronik sementara dengan

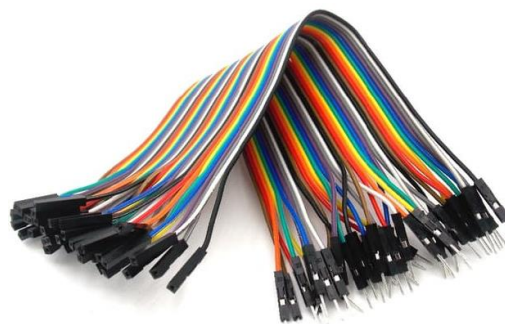
tujuan uji coba atau *prototype* tanpa harus menyolder. Dengan memanfaatkan *breadboard*, komponen-komponen elektronik yang dipakai tidak akan rusak dan dapat digunakan kembali untuk membuat rangkaian lain [13].



Gambar 2.8. *BreadBoard*

2.2.10. Kabel Jumper

Kabel *jumper* merupakan kabel *elektrik* yang berfungsi untuk menghubungkan antar komponen yang ada di *breadboard* tanpa harus memerlukan solder. Untuk menyambungkan rangkaian pada papan *breadboard*. Terdapat 3 macam kabel *jumper* yaitu *male to male*, *male to female* dan *female to female* [14].



Gambar 2.9. Kabel Jumper

2.2.11. *Arduino UNO*

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada *ATmega328*. *Arduino UNO* mempunyai 14 *pin* digital *input/output* (6 di antaranya dapat digunakan sebagai *output PWM*), 6 *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. *Arduino UNO* memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer¹⁵ dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya [15].



Gambar 2.10. *Arduino Uno*

BAB III

METODOLOGI PENELITIAN

3.1. Prosedur Penelitian

Prosedur penelitian disini merupakan langkah-langkah yang digunakan dalam proses pengumpul data yang meliputi perencanaan, analisa data, perancangan desain, penyusunan code, pengujian dan implementasi serta perawatan.

3.1.1. Rencana / *Planning*

Metode Rencana atau *planning* merupakan langkah awal dalam melakukan penelitian. Rencananya akan di buat sebuah Rancang Bangun *Hardware* Sistem Keamanan Berlapis Kunci Pintu Rumah Berbasis *E-Ktp* dan *Fingerprint* berikut langkah-langkah perancangannya:

1. Mencari permasalahan yang dapat digunakan untuk bahan perancangan sistem.
2. Mencari referensi yang sesuai dengan kebutuhan dalam perancangan sistem yang akan dibuat.
3. Pengumpulan alat dan bahan yang dibutuhkan dalam perancangan sistem.

3.1.2. Data Analisis

Melakukan Analisis permasalahan yaitu langkah-langkah awal pengumpulan data, penyusunan pembuatan rancang bangun *hardware*

sistem keamanan berlapis kunci pintu rumah berbasis E-KTP dan *fingerprint*, serta menganalisa data serta mendata *hardware* apa saja yang akan digunakan dalam pembuatan rancang bangun *hardware* sistem keamanan berlapis kunci pintu rumah berbasis *E-ktp* dan *fingerprint*.

3.1.3. Desain

Metode pencarian data dengan cara menghadapi langsung suatu permasalahan yang terjadi pada kasus yang dimaksud dengan melakukan perancangan terhadap sistem dan alat yang akan dibuat dalam bentuk *prototype* termasuk kebutuhan *software* dan *hardware* yang dibutuhkan. Setelah mengumpulkan data dan menganalisa data yang dibutuhkan serta berkaitan dengan perencanaan yaitu membuat suatu Rancang Bangun *Hardware* Sistem Keamanan Berlapis Kunci Pintu Rumah Berbasis *E-Ktp* dan *Fingerprint*, maka langkah berikutnya adalah membuat rancangan / desain dari perencanaan tersebut, seperti:

1. Membuat desain Alat Rancang Bangun *Hardware* Sistem Keamanan Berlapis Kunci Pintu Rumah Berbasis *E-Ktp* dan *Fingerprint*.
2. Membuat *database*.

3.1.4. Coding

Membuat suatu alat Rancang Bangun *Hardware* Sistem Keamanan Berlapis Kunci Pintu Rumah Berbasis *E-Ktp* dan

Fingerprint dengan menggunakan Bahasa pemrograman php dan bahasa pemrograman yang digunakan *Arduino IDE*.

3.1.5. Testing

Melakukan pengujian sistem dan alat yang dibuat dengan menggunakan *variable chip* E-ktp dan sidik jari.

3.1.6. Implementation

Setelah dilakukan pengujian maka sistem dan alat tersebut akan diimplementasikan di rumah dengan menggunakan chip E-ktp, sidik jari.

3.1.7. Maintenanc

Melakukan perawatan alat secara berkala.

3.2. Metode Pengumpulan Data

3.2.1. Studi Literatur

Metode penelitian yang di gunakan adalah Studi Literatur. Metode ini digunakan untuk mendapatkan teori untuk menyelesaikan permasalahan dengan mengumpulkan teori-teori yang mendukung dan membaca sumber seperti buku-buku yang berkaitan dengan penelitian, skripsi, jurnal, maupun karangan yang berkaitan.

3.2.2. Sumber Data

Untuk pencarian data keamanan rumah di dapatkan melalui referensi internet, berikut sumber data contoh kasus pencurian rumah di antaranya:

1. Kasus pencurian rumah Jeremy Thomas yang dilakukan oleh sepasang suami istri. Sejumlah harta benda di rumah raib digondol maling (Sumber data Liputan6.com).
2. Kasus pencurian rumah yang di tinggal Tarawih, rumah seorang warga di bekasi dibobol maling (Sumber data detiknews).
3. Ibu dan anak terlibat sindikat pencurian rumah kosong di Jakarta barat (Sumber Data detiknews).

Dari contoh kasus di atas dapat disimpulkan bahwa keamanan rumah berperan penting dalam kebutuhan sehari-hari. Dengan adanya sistem keamanan rumah berbasis telegram memudahkan pemilik rumah dalam *memonitor* kondisi keadaan rumah dan tidak perlu khawatir akan kehilangan kunci rumah. Tidak hanya *memonitor*, keamanan rumah juga dapat *meminimalisir* tindakan kejahatan seperti pembobolan rumah.

3.3. Waktu dan Tempat Penelitian

3.3.1. Waktu

Waktu yang digunakan peneliti untuk penelitian ini dilaksanakan sejak tanggal dikeluarkannya ijin penelitian dalam kurun waktu kurang lebih 2 (dua) bulan, 1 bulan pengumpulan data dan 1 bulan pengolahan data yang meliputi penyajian dalam bentuk tugas akhir dan proses bimbingan berlangsung.

3.3.2. Tempat Pelaksanaan

Tempat pelaksanaan penelitian ini adalah di Jl. Mejabung Kelurahan Panggung Kecamatan Tegal Timur di rumah kediaman Alfina Inayatuzzulfa.

BAB IV

ANALISA DAN PERANCANGAN SISTEM

4.1. Analisa Permasalahan

Pada saat ini keamanan rumah masih menggunakan sistem penguncian manual yaitu dengan menggunakan kunci konvensional. Penggunaan kunci konvensional kurang praktis pada zaman sekarang, karena pemilik rumah harus membawa banyak kunci ketika akan bepergian dari rumah dan sering kali pemilik rumah lupa bahkan kehilangan kunci. Penggunaan kunci konvensional juga mudah dibuka oleh pencuri karena semakin berkembang cara pencuri untuk membuka pintu rumah. Hal itu menjadikan kunci pintu konvensional menjadi kurang efektif dan kurang aman jika dibandingkan dengan pintu digital atau *smart door lock* berbasis E-KTP dan *fingerprint*.

E-KTP dapat digunakan sebagai RFID *tag* karena di dalamnya terdapat *chip* yang menyimpan nomor ID unik, alat pengaman pintu ini memanfaatkan E-KTP untuk membuka pintu. E-KTP berfungsi sebagai *transponder* dan tergolong dalam tag pasif karena tidak memiliki catu daya sendiri, catu dayanya berasal dari pancaran gelombang RFID *reader*. RFID (*Radio Frequency Identification*) *reader* berfungsi untuk membaca nomor ID pada E-KTP. Data yang tersimpan dalam *chip* akan terkirim atau terbaca melalui gelombang radio setelah *tag*-antena menerima pancaran gelombang radio dari *reader*-antena kemudian data akan dikirim ke *mikrokontroler*.

Smart door lock atau kunci pintu pintar, menggunakan perangkat digital untuk mendapatkan akses kedalam ruangan. Perangkat digital yang biasanya dipakai berupa RFID, *bluetooth*, *fingerprint*, atau aplikasi *smartphone*. Pada kunci pintar menggunakan *fingerprint* terdapat beberapa kelebihan jika dibandingkan dengan kunci pintu konvensional, diantaranya adalah lebih efisien karena tidak menggunakan anak kunci, sehingga tidak akan ada kasus kehilangan kunci atau ketinggalan, dan lebih aman, karena tidak ada sidik jari yang sama. *Fingerprint* yang digunakan diharapkan dapat membatasi orang yang dapat memasuki rumah, dan *notifikasi* yang dikirimkan ke aplikasi Telegram diharapkan dapat mencatat identitas pembuka pintu dan waktu saat pintu terbuka. Sehingga orang-orang yang keluar masuk rumah tersebut dapat tercatat dengan baik.

4.2. Analisa Kebutuhan Sistem

Analisa kebutuhan dilakukan untuk mengetahui kebutuhan apa saja yang diperlukan dalam penelitian, menentukan keluaran yang akan dihasilkan sistem, masukan yang dihasilkan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran serta kontrol terhadap sistem.

4.2.1. Analisa Perangkat Keras atau Hardware

Adapun perangkat keras yang digunakan dalam pembuatan “Prototype Sistem Keamanan Berlapis Kunci Pintu Rumah Berbasis E-KTP dan Fingerprint” adalah sebagai berikut:

1. *NodeMCU*
2. *Arduino Uno*

3. *Relay 12V*
4. *Fingerprint*
5. *RFID Module (Radio Frequency Identification)*
6. *LCD OLED Display Module (Organic Light Emitting Diode)*
7. *Kabel Jumper*
8. *Adaptor 12V*
9. *Solenoid Door lock*
10. *Breadboard*
11. *Kayu*
12. *ACP (Aluminium Composite Panel)*

4.2.2. Analisa Perangkat Lunak atau Software

Adapun spesifikasi perangkat lunak yang dapat digunakan selama penelitian pengembangan adalah sebagai berikut:

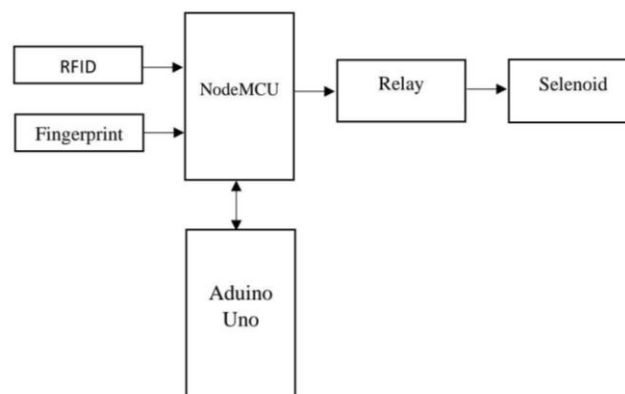
1. *Arduino IDE (Integrated Development Environment)* berfungsi sebagai text editor untuk membuat, mengedit dan juga memvalidasi kode program.
2. *Firebase* berfungsi untuk mengolah database.
3. *Google Chrome* berfungsi untuk menampilkan website dari *firebase* tersebut.
4. *Telegram* berfungsi untuk mengirim dan menampilkan notifikasi.

4.3. Perancangan Sistem

Dalam penelitian ada beberapa hal yang harus diperhatikan, seperti pemilihan komponen, rangkaian yang akan dimuat dan bahan atau material dari alat sampai harga komponen dan ketersediaan dipasaran.

4.3.1. Diagram Blok Sistem

Perancangan dilakukan berdasarkan blok perblok dari setiap rangkaian, dimana tiap-tiap blok mempunyai fungsi masing-masing dan blok rangkaian yang satu dengan blok rangkaian yang lain merupakan satu kesatuan yang saling terkait dan berhubungan serta membentuk satu kesatuan yang saling menunjang kerja dari sistem. Blok rangkaian dari rangkaian ini dapat dilihat selengkapnya pada gambar 4.1.

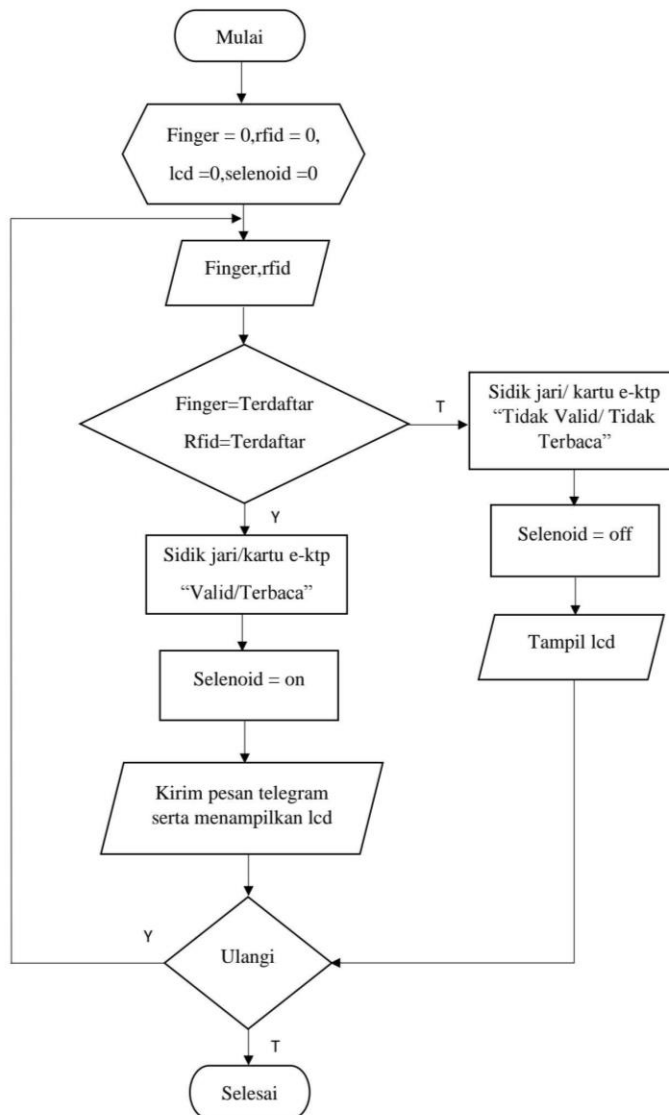


Gambar 4.1 Diagram Blok

Diagram blok diatas dijelaskan bahwa *input* yang masuk pada mikrokontroler, dan *output* pada mikrokontroler menggerakkan *solenoid doorlock* melalui *relay*. Disini dapat dilihat pentingnya peranan mikrokontroler, yang mana mikrokontroler akan mengolah *input* dan mengatur *output*. Jadi *mikrokontroler* merupakan pengendali utama dari sistem.

4.3.2. Flowchart Sistem

Flowchart sistem keamanan berlapis kunci pintu rumah berbasis e-ktp dan *fingerprint* yang dirancang adalah sebagai berikut:



Gambar 4.2 Flowchart sistem keamanan berlapis kunci pintu rumah berbasis E-KTP dan *fingerprint* berbasis telegram.

Pada *flowchart* di atas, sistem dimulai dengan membaca *chip* E-KTP dan sidik jari dikenali maka sistem akan menentukan siapa

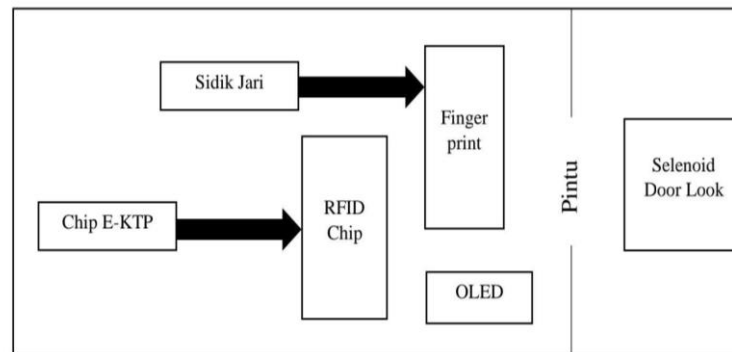
pemilik *chip* E-KTP dan sidik jari tersebut, setelah itu data pemilik *chip* E-KTP dan sidik jari akan dikirimkan ke telegram melalui koneksi internet, dan nama si pemilik *chip* E-KTP dan sidik jari akan ditampilkan di *LCD OLED* kemudian sistem akan membuka kunci pintu dengan menarik tuas pengait dari *solenoid door lock*.

Keterangan perangkat keras (*Hardware*) dapat dilihat pada tabel 4.1:

Tabel 4.1 Rangkaian keseluruhan Perangkat Keras

NodeMCU	Fingerprint
3V3	VA
GND	GND
D5	RX
D6	TX
NodeMCU	Relay
D0	S
GND	GND
3V3	POWER
Arduino	RFID Module RC522
5V	3.3V
GND	GND
9	RST
10	SDA
11	MOSI
12	MISO
13	SCK
Arduino	NodeMCU
TX	RX

4.4. Desain Input/Output



Gambar 4.3 Desain Input/Output

Rangkaian komponen rancang bangun hardware sistem keamanan berlapis kunci pintu rumah berbasis E-KTP dan *fingerprint* yang dibuat ini adalah sebagai berikut:

1. Rangkaian Mikrokontroler NodeMCU

Rangkaian ini merupakan jantung dan pusat rangkaian sebagai pengendali utama seluruh rangkaian yang ada, Mikrokontroler NodeMCU. Mikrokontroler ini terdapat 9 GPIO yang di dalamnya ada 3 *pin PWM*, 1 x *ADC Channel*, dan *pin RX TX*, 3 *pin ground*, tombol *reset*, *port usb*, dan tombol *flash*.

2. Rangkaian Mikrokontroler Arduino Uno

Rangkaian ini juga merupakan jantung dan pusat rangkaian sebagai pengendali utama seluruh rangkaian yang ada, Mikrokontroler Arduino Uno. Mikrokontroler ini memiliki 14 *pin digital input/output*, 6 analog *input*, sebuah resonator keramik 16MHz, koneksi *USB*, colokan *power input*, *ICSP header*, dan sebuah tombol *riset*.

3. Rangkaian *Solenoid Doorlock*

Rangkaian ini digunakan untuk mengendalikan *solenoid doorlock* yang akan digunakan pada sistem ini, *solenoid doorlock* yang akan diperlakukan ada satu yang bertegangan 12v, yang mana untuk satu *solenoid doorlock* diperlakukan empat *relay*. Penggunaan *relay* ini bertujuan untuk menghasilkan keluaran yang memiliki daya yang cukup besar agar kerja *solenoid* dapat bekerja dengan maksimal.

4. Rangkaian Sensor Fingerprint

Rangkaian sensor *fingerprint* ini digunakan sebagai inputan dari *solenoid doorlock*.

5. Rangkaian Sensor RFID

Rangkaian sensor *rfid* ini digunakan sebagai inputan dari *solenoid doorlock*.

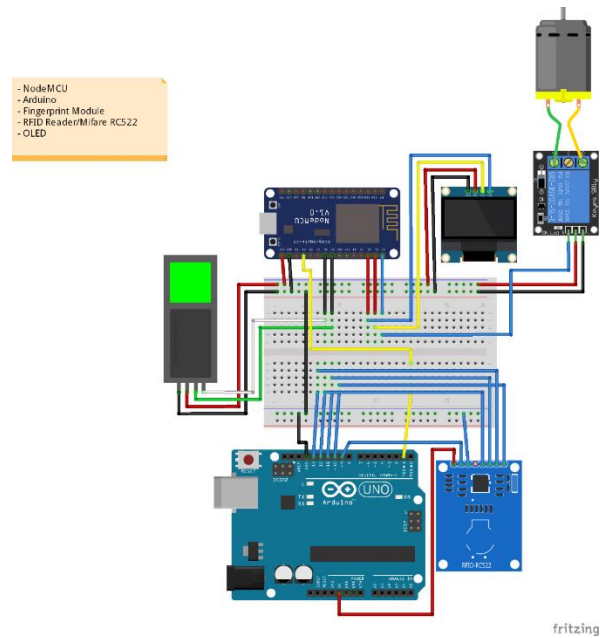
6. Rangkaian OLED LCD

Rangkaian OLED LCD ini biasanya digunakan sebagai tampilan layar.

7. Rangkaian Keseluruhan Sistem

Rangkaian keseluruhan sistem ini akan memperlihatkan keterkaitan seluruh sistem yang ada, mulai dari mikrokontroler NodeMCU dan Arduino Uno sebagai pusat dari pengendali sampai sensor-sensor sebagai *input* dan *solenoid doorlock* dan pesan ke aplikasi Telegram berupa *chatbot*, sehingga tercatat *user* yang membuka pintu

beserta waktu kejadian, yang secara tidak langsung menjadi sistem *history* dari alat tersebut.



Gambar 4.4 Rangkaian Keseluruhan

BAB V

HASIL DAN PEMBAHASAN

5.1. Implementasi Sistem

Implementasi sistem keamanan berlais kunci pintu rumah menggunakan E-KTP dan *fingerprint* berbasis telegram pada pembahasan bab ini terdiri dari implementasi perangkat keras (*hardware*) dan implementasi *software*. Implementasi *software* ditulis di Arduino IDE (*Integrated Development Environment*) kemudian *source code* hasilnya dijadikan *file* berekstensi .ino dan diunggah pada Mikrokontroler NodeMCU dan Arduino UNO.

5.1.1. Implementasi perangkat keras

Implementasi perangkat keras dimulai dari proses instalasi alat atau perakitan alat yang digunakan dalam membangun sistem *rfid* dan *fingerprint* yang akan mengirim data ke telegram. Adapun perangkat keras yang digunakan untuk memenuhi kriteria dalam pengoprasian objek pada sistem ini adalah sebagai berikut:

1. *NodeMCU*
2. *Arduino UNO*
3. *Relay 12V*
4. *Fingerprint*
5. *RFID Module (Radio Frequency Identification)*
6. *LCD OLED (Organic Light Emitting Diode)*
7. *Kabel Jumper*

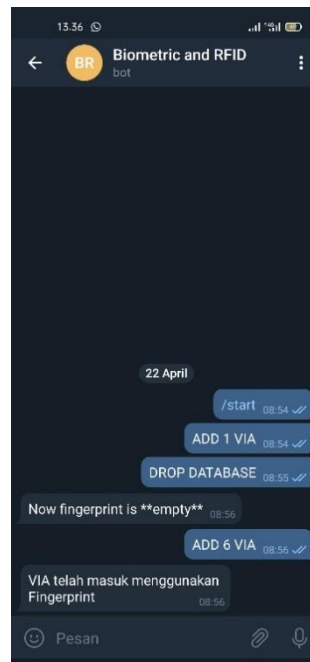
8. Adaptor 12V
9. *Solenoid Door lock*
10. *Breadboard*

Semua perangkat keras dirangkai dan disusun membentuk prototype sistem keamanan berlapis kunci pintu rumah menggunakan E-KTP dan *fingerprint* berbasis telegram.

5.1.2. Implementasi perangkat lunak

Implementasi perangkat lunak merupakan proses penerapan sistem dari sisi perangkat lunak (*software*). *Software ini* memanfaatkan aplikasi telegram sebagai media monitor untuk memberikan informasi dalam bentuk notifikasi sehingga dapat diketahui apabila ada seseorang yang masuk ke dalam rumah.

Pesan yang dikirimkan melalui aplikasi telegram telah dibuat dengan memanfaatkan fitur *bot* yang dapat memberikan respon secara otomatis berdasarkan pengaturan yang telah dilakukan sebelumnya. Tampilan *bot* telegram tampak seperti pada gambar 5.2 di bawah ini:



Gambar 5.2 Tampilan *Notifikasi* Telegram

5.2. Hasil Pengujian

Tahap pengujian merupakan hal yang dilakukan untuk menentukan apakah perangkat lunak dan perangkat keras sudah berjalan dengan lancar dan sudah sesuai yang diharapkan. Pengujian yang dilakukan pada sistem keamanan berlapis kunci pintu rumah dalam penelitian ini mencakup pengujian sensor *fingerprint*, sensor *rfid*, pengujian *solenoid doorlock* dan pengujian *LCD OLED* serta pengujian pengiriman notifikasi oleh *bot* telegram. Hasil pengujian dapat dilihat pada tabel berikut:

Tabel 5.1 Tabel Hasil Pengujian

No	Nama Pengujian	Kondisi Pengujian	Hasil Pengujian
1	Sensor <i>fingerprint</i>	Jika sensor <i>fingerprint</i> diinputkan sidik jari yang benar	<i>Solenoid doorlock ON</i>
		Jika sensor <i>fingerprint</i> diinputkan sidik jari yang salah	<i>Solenoid doorlock OFF</i>
2	Sensor <i>rfid</i>	Jika sensor <i>rfid</i> diinputkan <i>chip</i> E-KTP yang benar	<i>Solenoid doorlock ON</i>
		Jika sensor <i>rfid</i> diinputkan <i>chip</i> E-KTP yang salah	<i>Solenoid doorlock OFF</i>
3	<i>Solenoid Door Lock</i>	Jika <i>solenoid doorlock on</i>	Sidik jari dan <i>chip</i> E-KTP benar
		Jika <i>solenoid doorlock off</i>	Sidik jari dan <i>chip</i> E-KTP salah
4	<i>LCD OLED</i>	Jika sensor <i>fingerprint</i> dan <i>rfid</i> diinputkan sidik jari dan <i>chip</i> E-KTP yang benar	<i>LCD OLED</i> akan menampilkan tanda centang (valid) pintu terbuka dan akan mengirim <i>notifikasi</i> ke Telegram
		Jika sensor <i>fingerprint</i> dan <i>rfid</i> diinputkan sidik jari dan <i>chip</i> E-KTP yang salah	<i>LCD OLED</i> akan menampilkan tanda silang (X) pintu tertutup.

Hasil pengujian pada sistem diatas menunjukkan beberapa keadaan diantaranya yaitu:

- a. Sistem hanya membaca sidik jari dan *chip* E-KTP yang telah didaftarkan serta dapat di-*setting* untuk mengenali lebih dari satu sidik jari dan *chip* E-KTP.
- b. Sistem akan terbuka jika sidik jari dan *chip* E-KTP yang di-*input* telah terdaftar.
- c. Jika sensor *fingerprint* mendeteksi sidik jari dan rfid medeteksi *chip* E-KTP yang salah maka akan menampilkan tanda silang (X) pada LCD OLED.
- d. Jika sensor *fingerprint* mendeteksi sidik jari dan *chip* E-KTP yang benar maka akan mendapatkan peringatan menampilkan tanda centang (valid) pada *LCD OLED* dan *notifikasi* ke Telegram.

BAB VI

KESIMPULAN

6.1. Kesimpulan

Dari uraian pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Sensor *fingerprint* dan *rfid* dapat digunakan sebagai *input* dalam sistem keamanan berlapis kunci pintu rumah, dimana ketika sidik jari dan *chip* E-KTP yang dikenali maka pintu akan secara otomatis terbuka.
2. Notifikasi pesan ke aplikasi Telegram yang berisi identitas dari pemilik sidik jari dan *chip* E-KTP si pembuka pintu hanya dikirim ke nomor *smartphone* yang terdaftar.
3. Sistem ini dapat menghemat biaya pengadaan alat pengamanan pintu, karena jika menggunakan kunci konvensional apabila kunci hilang maka harus membeli yang baru. Tetapi dengan sistem ini tidak perlu membeli kunci yang baru.
4. *Output* yang dihasilkan oleh sistem ini berupa pergerakan *solenoid door lock* dan *relay* ketika sidik jari dan *chip* E-KTP yang ditempelkan pada sensor *fingerprint* dan *rfid valid* kemudian akan dikirimkan melalui *notifikasi* Telegram.

6.2. Saran

Adapun saran untuk pengembangan penelitian selanjutnya yaitu:

1. Pada pembuatan sistem berikutnya sebaiknya perlu diperhatikan tata letak komponen dan kabel agar lebih rapi.
2. Lebih baik ada penambahan *notifikasi*, jika alat *fingerprint* dan *rfid* tersebut di bobol secara paksa.
3. Untuk menyempurnakan sistem keamanan kunci pintu rumah tersebut, sebaiknya ada penambahan *notifikasi* jika seseorang keluar dari rumah tersebut.

DAFTAR PUSTAKA

- [1] Suyoko, D."ALAT PENGAMAN PINTU RUMAH MENGGUNAKAN RFID (RADIO FREQUENCY IDENTIFICATION) 125KHz BERBASIS MIKROKONTROLER ATMEGA328". Skripsi.Program Studi Teknik Elektronika Universitas Negeri Yogyakarta.Yogyakarta, 2012
- [2] Widyananda, A.,et all., "PERANCANGAN DAN IMPLEMENTASI KUNCI PINTU OTOMATIS DENGAN", 2013
- [3] Juwariyah, T., et all., "Purwa Rupa Sistem Pengaman Sepeda Motor Berbasis IoT (Internet of Things)", vol. 11, 2019.
- [4] Arifandi, M., *"PROTOTYPE SISTEM KEAMANAN PINTU MENGGUNAKAN SIDIK JARI BERBASIS ARDUINO UNO ATMega328 DAN SMS GATEWAY"*, 2019.
- [5] Ilman, A K N.,, *"ABSENSI FINGERPRINT BERBASIS SMS GATEWAY"*, 2019.
- [6] Saputra, I N T J E.,, *"SISTEM KEMANAN PINTU RUMAH BERBASIS WEB MENGGUNAKAN NODEMCU ESP8266 V.3"*, 2017.
- [7] Yudhana, A.,et all.,, *"PERANCANGAN PENGAMAN PINTU RUMAH BERBASIS SIDIK JARI MENGGUNAKAN METODE UML"*, vol. 10, 2018.
- [8] Shintia., et all, in *Sistem Keamanan Pintu Ruang Server Menggunakan Fingerprint Berbasis IoT*, Tegal, Politeknik Harapan Bersama, 2020.

- [9] Shintia., et all, in *Sistem Keamanan Pintu Ruang Server Menggunakan Fingerprint Berbasis IoT*, Tegal, Politeknik Harapan Bersama, 2020.
- [10] Firmansyah, I , "*JAKET ANTISIPASI DINI AKIBAT KANTUK BERBASIS IOT*", 2019.
- [11] Sulasmoro, A H., in *MODUL ALGORITMA DAN PEMROGRAMAN*, Tegal, Politeknik Harapan Bersama, 2010.
- [12] Hadinata, A F., in *RANCANG BANGUN PENGAMAN PINTU RUMAH OTOMATIS MENGGUNAKAN E-KTP BERBASIS MIKROKONTROLLER*, Lampung, INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA, p. 2019.
- [13] Abidin, Z, Tijaniyah, dan Bachrudin M., in *RANCANG BANGUNPENGOPERASIAN LAMPU MENGGUNAKAN SINYAL ANALOG SMARTPHONE BERBSIS MIKROKONTROLER*, vol. 1, 2019.
- [14] Shintia., et all, in *Sistem Keamanan Pintu Ruang Server Menggunakan Fingerprint Berbasis IoT*, Tegal, Politeknik Harapan Bersama, 2020.
- [15] Firmansyah, I , "*JAKET ANTISIPASI DINI AKIBAT KANTUK BERBASIS IOT*", 2019.

LAMPIRAN

Lampiran 1 Surat Ketersediaan Membimbing

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini :

Nama : Arfan Haqiqi Sulasmoro, M.Kom
NIDN : 0623037704
NIPY : 02.009.054
Jabatan Struktural : -
Jabatan Fungsional : Asisten Ahli

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut :

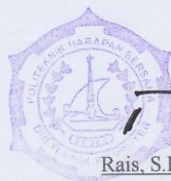
NO	Nama	NIM	Program Studi
1.	Via Aditia Stefanni	18040016	DIII Teknik Komputer

Judul TA : “ Rancang Bangun Hardware Sistem Keamanan Berlapis Kunci Pintu Rumah Menggunakan E-KTP dan Fingerprint ”

Demikian Pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 9 Februari 2021

Mengetahui
Ka Prodi DIII Teknik Komputer



Rais, S.Pd., M.Kom
NIPY.07.011.083

Calon Dosen Pembimbing I

Arfan Haqiqi Sulasmoro, M.Kom
NIPY.02.009.054

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini :

Nama : Achmad Sutanto, S.Kom

NIPY : 11.012.128

Jabatan Struktural : Staff UPT. Sistem Informasi

Jabatan Fungsional : Dosen Tetap DIII Teknik Komputer

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut :

No	Nama	NIM	Program Studi
1	Via Aditia Stefanni	18040016	DIII Teknik Komputer

Judul TA : "Rancang Bangun Hardware Sistem Keamanan Berlapis Kunci Pintu Rumah Berbasis E-KTP dan Fingerprint".

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 14 April 2020

Calon Dosen Pembimbing II



Achmad Sutanto, S.Kom
NIPY. 11.012.128

Mengetahui,
Ka. Prodi DIII Teknik Komputer



Lampiran 2 Kode Program Arduino IDE

NODEMCU

```
//*****LIBRARIES*****
*****//
#include <CTBot.h>;

#include <FirebaseESP8266.h>

#include <SPI.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>

#include <Adafruit_GFX.h>
//https://github.com/adafruit/Adafruit-GFX-Library
#include <Adafruit_SSD1306.h>
//https://github.com/adafruit/Adafruit_SSD1306
#include <Adafruit_Fingerprint.h>
//https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library

CTBotInlineKeyboard myKbd;

//***** FINGERPRINT
*****//
#define FINGERPRINT_RX D5
#define FINGERPRINT_TX D6
int FingerID = 0;
uint8_t id;

//***** DECLARATION FOR SSD1306 DISPLAY CONNECTED
USING SOFTWARE I2C *****//
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET 0

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

//***** FINGERPRINT PIN
*****//
SoftwareSerial mySerial(FINGERPRINT_RX, FINGERPRINT_TX);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

//***** CONFIG WIFI
*****//
#define WIFI_SSID "Wakil President Jancukers"
#define WIFI_PASS "jancukers"

//***** TELEGRAM *****//
#define TOKEN "1796027528:AAHh066GQ8nUcmnE8JL6LEs5Xgx4CI6YxNw"
#define CALLBACK_USERS "UsersData"
#define CALLBACK_ADD "AddData"
```



```

#define WIFI_CONNECTED_WIDTH 63
#define WIFI_CONNECTED_HEIGHT 49
const uint8_t PROGMEM WIFI_CONNECTED_BITS[] = {0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff,
0xf8, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x3f, 0xff,
0xc0, 0x07, 0xff, 0xf8, 0x00, 0x00, 0xff, 0xf8, 0x00, 0x00, 0x3f,
0xfe, 0x00, 0x03, 0xff, 0x80, 0x00, 0x00, 0x03, 0xff, 0x80, 0x07,
0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xc0, 0x1f, 0xf8, 0x00, 0x00,
0x00, 0x00, 0x3f, 0xf0, 0x3f, 0xe0, 0x01, 0xff, 0xff, 0x00, 0x0f,
0xf8, 0x7f, 0x80, 0x0f, 0xff, 0xff, 0xe0, 0x03, 0xfc, 0xff, 0x00,
0x7f, 0xff, 0xff, 0xfc, 0x01, 0xfe, 0xfc, 0x01, 0xff, 0xff, 0xff,
0xff, 0x00, 0x7e, 0x78, 0x07, 0xff, 0xc0, 0x07, 0xff, 0xc0, 0x3c,
0x00, 0x0f, 0xfc, 0x00, 0x00, 0x00, 0x7f, 0xe0, 0x00, 0x00, 0x1f, 0xf0,
0x00, 0x00, 0x1f, 0xf0, 0x00, 0x00, 0x00, 0x3f, 0xc0, 0x00, 0x00, 0x07,
0xf8, 0x00, 0x00, 0x7f, 0x00, 0x01, 0x00, 0x01, 0xfc, 0x00, 0x00,
0x7e, 0x00, 0x7f, 0xfc, 0x00, 0xfc, 0x00, 0x00, 0x3c, 0x03, 0xff,
0xff, 0x80, 0x78, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xc0, 0x00,
0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00,
0x3f, 0xf0, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x3f, 0x80, 0x03,
0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0x00, 0x01, 0xf8, 0x00, 0x00,
0x00, 0x00, 0x1c, 0x00, 0x00, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xf0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x3f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f,
0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xf8, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x1f, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00};

//*****FINGERPRINT START
ICONS*****
#define FINGERPRINT_START_WIDTH 64
#define FINGERPRINT_START_HEIGHT 64
const uint8_t PROGMEM FINGERPRINT_START_BITS[] = {0x00, 0x00, 0x00,
0x1f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xfe, 0x00,
0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00,
0x00, 0x0f, 0xc0, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x00,
0x01, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x7c, 0x00,
0x00, 0x00, 0x00, 0x78, 0x00, 0x00, 0x3e, 0x00, 0x00, 0x00, 0x00,
0xf0, 0x3f, 0xf8, 0x0f, 0x00, 0x00, 0x00, 0x01, 0xe0, 0xff, 0xfe,
0x07, 0x80, 0x00, 0x00, 0x03, 0xc3, 0xff, 0xff, 0x03, 0x80, 0x00,
0x00, 0x03, 0x87, 0xc0, 0x07, 0xc3, 0xc0, 0xc0, 0x00, 0x00, 0x0f, 0x0f,
0x00, 0x03, 0xe1, 0xc0, 0x00, 0x00, 0x0f, 0xe0, 0x00, 0x00, 0xe0,
0xe0, 0x00, 0x00, 0x0e, 0x1c, 0x00, 0x00, 0xf0, 0xe0, 0x00, 0x00,
0x0c, 0x3c, 0x1f, 0xe0, 0x70, 0xe0,

```

```

0x00, 0x00, 0x00, 0x38, 0x3f, 0xf0, 0x38, 0x70, 0x00, 0x00, 0x00,
0x78, 0x78, 0xf8, 0x38, 0x70, 0x00, 0x00, 0x00, 0x70, 0x70, 0x3c,
0x18, 0x70, 0x00, 0x00, 0x00, 0xe0, 0xe0, 0x1e, 0x1c, 0x70, 0x00,
0x00, 0x03, 0xe1, 0xe0, 0x0e, 0x1c, 0x70, 0x00, 0x00, 0x0f, 0xc1,
0xc3, 0x0e, 0x1c, 0x70, 0x00, 0x00, 0x3f, 0x03, 0xc3, 0x8e, 0x1c,
0x70, 0x00, 0x00, 0x3e, 0x03, 0x87, 0x0e, 0x1c, 0x70, 0x00, 0x00,
0x30, 0x07, 0x07, 0x0e, 0x18, 0xe0, 0x00, 0x00, 0x00, 0x0e, 0x0e,
0x0e, 0x38, 0xe0, 0x00, 0x00, 0x00, 0x3e, 0x1e, 0x1e, 0x38, 0xe0,
0x00, 0x00, 0x00, 0xf8, 0x1c, 0x1c, 0x38, 0xe0, 0x00, 0x00, 0x03,
0xf0, 0x38, 0x3c, 0x38, 0xe0, 0x00, 0x00, 0x3f, 0xc0, 0xf8, 0x78,
0x38, 0xe0, 0x00, 0x00, 0x7f, 0x01, 0xf0, 0x70, 0x38, 0xf0, 0x00,
0x00, 0x78, 0x03, 0xe0, 0xe0, 0x38, 0x70, 0x00, 0x00, 0x00, 0x0f,
0x81, 0xe0, 0x38, 0x7c, 0x00, 0x00, 0x00, 0x3f, 0x03, 0xc0, 0x38,
0x3e, 0x00, 0x00, 0x00, 0xfc, 0x0f, 0x80, 0x38, 0x1e, 0x00, 0x00,
0x07, 0xf0, 0x1f, 0x1c, 0x1c, 0x04, 0x00, 0x00, 0x3f, 0xc0, 0x3e,
0x3f, 0x1e, 0x00, 0x00, 0x00, 0x7f, 0x00, 0xf8, 0x7f, 0x0f, 0x00,
0x00, 0x00, 0x38, 0x01, 0xf0, 0xf7, 0x07, 0xc0, 0x00, 0x00, 0x00,
0x07, 0xe1, 0xe3, 0x83, 0xf8, 0x00, 0x00, 0x00, 0x3f, 0x87, 0xc3,
0xc0, 0xfc, 0x00, 0x00, 0x01, 0xfe, 0x0f, 0x81, 0xe0, 0x3c, 0x00,
0x00, 0x0f, 0xf8, 0x1f, 0x00, 0xf0, 0x00, 0x00, 0x00, 0x1f, 0xc0,
0x7c, 0x00, 0x7c, 0x00, 0x00, 0x00, 0x1e, 0x01, 0xf8, 0x00, 0x3f,
0x00, 0x00, 0x00, 0x00, 0x07, 0xe0, 0x78, 0x0f, 0xc0, 0x00, 0x00,
0x00, 0x3f, 0x81, 0xfe, 0x07, 0xf0, 0x00, 0x00, 0x01, 0xfe, 0x07,
0xff, 0x01, 0xf0, 0x00, 0x00, 0x07, 0xf8, 0x0f, 0x87, 0x80, 0x30,
0x00, 0x00, 0x07, 0xc0, 0x3f, 0x03, 0xe0, 0x00, 0x00, 0x00, 0x06,
0x00, 0xfc, 0x01, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x03, 0xf0, 0x00,
0x7e, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xc0, 0x00, 0x3f, 0x80, 0x00,
0x00, 0x00, 0x7f, 0x00, 0xf8, 0x0f, 0x80, 0x00, 0x00, 0x00, 0xfc,
0x03, 0xfe, 0x01, 0x80, 0x00, 0x00, 0x00, 0xf0, 0x1f, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x07, 0xe0, 0x00, 0x00, 0x00,
0x00, 0x00, 0xfc, 0x03, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x03, 0xf0,
0x00, 0x78, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xc0, 0x00, 0x18, 0x00,
0x00, 0x00, 0x00, 0x0f, 0x01, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x07, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xfe,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1e, 0x0e, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x18, 0x00, 0x00, 0x00, 0x00};

```

```

//*****FINGERPR
INT VALID
ICONS*****
#define FINGERPRINT_VALID_WIDTH 64
#define FINGERPRINT_VALID_HEIGHT 64
const uint8_t PROGMEM FINGERPRINT_VALID_BITS[] = {0x00, 0x00, 0x03,
0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xe0, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00,
0x00, 0xfc, 0x00, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x03, 0xe0, 0x00,
0x1f, 0x00, 0x00, 0x00, 0x00, 0x07, 0xc0, 0x00, 0x07, 0x80, 0x00,
0x00, 0x00, 0x0f, 0x80, 0x00, 0x03, 0xe0, 0x00, 0x00, 0x00, 0x0e,
0x03, 0xff, 0x01, 0xe0, 0x00, 0x00, 0x00, 0x1c, 0x1f, 0xff, 0xe0,
0xf0, 0x00, 0x00, 0x00, 0x3c, 0x3f, 0xff, 0xf0, 0x78, 0x00, 0x00,
0x00, 0x78, 0x7c, 0x00, 0xf8, 0x3c, 0x00, 0x00, 0x00, 0x70, 0xf0,
0x00, 0x3c, 0x1c, 0x00, 0x00, 0x00, 0xe1, 0xe0, 0x00, 0x1e, 0x1c,
0x00, 0x00, 0x00, 0xe1, 0xc0, 0x00, 0x00, 0x0f, 0x0e, 0x00, 0x00, 0x00,
0xc3, 0x81, 0xfc, 0x07, 0x0e, 0x00, 0x00, 0x00, 0x03, 0x83, 0xff,
0x07, 0x8e, 0x00, 0x00, 0x00, 0x07, 0x07, 0x8f, 0x83, 0x87, 0x00,
0x00, 0x00, 0x0f, 0x0f, 0x03, 0xc3, 0x87, 0x00, 0x00, 0x00, 0x1e,

```

```
0x0e, 0x01, 0xc3, 0x87, 0x00, 0x00, 0x00, 0x3c, 0x1c, 0x00, 0xe1,
0x87, 0x00, 0x00, 0x00, 0xf8, 0x1c, 0x30, 0xe1, 0x87, 0x00, 0x00,
0x07, 0xf0, 0x38, 0x70, 0xe1, 0x86, 0x00, 0x00, 0x07, 0xc0, 0x78,
0x70, 0xe3, 0x8e, 0x00, 0x00, 0x02, 0x00, 0xf0, 0xf0, 0xe3, 0x8e,
0x00, 0x00, 0x00, 0x01, 0xe0, 0xe0, 0xe3, 0x8e, 0x00, 0x00, 0x00,
0x03, 0xc1, 0xe1, 0xc3, 0x8e, 0x00, 0x00, 0x00, 0x0f, 0x83, 0xc3,
0xc3, 0x8e, 0x00, 0x00, 0x00, 0x7f, 0x07, 0x83, 0x83, 0x0e, 0x00,
0x00, 0x07, 0xfc, 0x0f, 0x07, 0x83, 0x0e, 0x00, 0x00, 0x07, 0xf0,
0x1e, 0x0f, 0x03, 0x0e, 0x00, 0x00, 0x07, 0x80, 0x7c, 0x1e, 0x03,
0x07, 0x00, 0x00, 0x00, 0x00, 0xf8, 0x3c, 0x03, 0x87, 0x80, 0x00,
0x00, 0x03, 0xf0, 0x78, 0x03, 0x83, 0xc0, 0x00, 0x00, 0x1f, 0xc0,
0xf0, 0x02, 0x00, 0x00, 0x00, 0x00, 0xff, 0x01, 0xe1, 0xc0, 0x0c,
0x00, 0x00, 0x07, 0xfc, 0x03, 0xc3, 0xe1, 0xff, 0xc0, 0x00, 0x07,
0xe0, 0x0f, 0x87, 0xc7, 0xff, 0xf0, 0x00, 0x07, 0x00, 0x3f, 0x0f,
0x0f, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x7c, 0x3e, 0x3f, 0xff, 0xfe,
0x00, 0x00, 0x03, 0xf8, 0x7c, 0x3f, 0xff, 0xff, 0x00, 0x00, 0x1f,
0xe0, 0xf0, 0x7f, 0xff, 0xff, 0x80, 0x00, 0xff, 0x83, 0xe0, 0xff,
0xff, 0xff, 0x80, 0x01, 0xfc, 0x07, 0xc1, 0xff, 0xff, 0xe3, 0xc0,
0x01, 0xe0, 0x1f, 0x01, 0xff, 0xff, 0xc3, 0xc0, 0x00, 0x00, 0xfe,
0x01, 0xff, 0xff, 0x87, 0xe0, 0x00, 0x03, 0xf8, 0x13, 0xff, 0xff,
0x0f, 0xe0, 0x00, 0x1f, 0xe0, 0x73, 0xff, 0xfe, 0x1f, 0xe0, 0x00,
0x7f, 0x81, 0xf3, 0xff, 0xfc, 0x1f, 0xe0, 0x00, 0xfc, 0x03, 0xe3,
0xef, 0xf8, 0x3f, 0xe0, 0x00, 0x60, 0x0f, 0xc3, 0xc7, 0xf0, 0x7f,
0xe0, 0x00, 0x00, 0x3f, 0x03, 0xc3, 0xe0, 0xff, 0xe0, 0x00, 0x00,
0xfc, 0x03, 0xc1, 0xc1, 0xff, 0xe0, 0x00, 0x07, 0xf0, 0x13, 0xe0,
0x83, 0xff, 0xe0, 0x00, 0x0f, 0xc0, 0x7b, 0xf8, 0x07, 0xff, 0xe0,
0x00, 0x0f, 0x01, 0xf9, 0xfc, 0x0f, 0xff, 0xc0, 0x00, 0x00, 0x07,
0xf1, 0xfe, 0x1f, 0xff, 0xc0, 0x00, 0x00, 0x1f, 0xc0, 0xff, 0x3f,
0xff, 0x80, 0x00, 0x00, 0x7e, 0x00, 0xff, 0xff, 0xff, 0x80, 0x00,
0x00, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0x00, 0x00, 0x00, 0xf0, 0x1f,
0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x1f, 0xff, 0xfc,
0x00, 0x00, 0x00, 0x01, 0xff, 0x8f, 0xff, 0xf8, 0x00, 0x00, 0x00,
0x03, 0xe0, 0xe3, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00,
0x7f, 0x00, 0x00};
```

```
/******
 *FINGERPRINT INVALID
 ICONS*****
 #define FINGERPRINT_INVALID_WIDTH 64
 #define FINGERPRINT_INVALID_HEIGHT 64
 const uint8_t PROGMEM FINGERPRINT_INVALID_BITS[] = {0x00, 0x00,
0x03, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xe0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xf8, 0x00, 0x00, 0x00,
0x00, 0x00, 0xfc, 0x00, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x03, 0xe0,
0x00, 0x1f, 0x00, 0x00, 0x00, 0x00, 0x07, 0xc0, 0x00, 0x07, 0x80,
0x00, 0x00, 0x00, 0x0f, 0x80, 0x00, 0x03, 0xe0, 0x00, 0x00, 0x00,
0x0e, 0x03, 0xff, 0x01, 0xe0, 0x00, 0x00, 0x00, 0x1c, 0x1f, 0xff,
0xe0, 0xf0, 0x00, 0x00, 0x00, 0x3c, 0x3f, 0xff, 0xf0, 0x78, 0x00,
0x00, 0x00, 0x78, 0x7c, 0x00, 0xf8, 0x3c, 0x00, 0x00, 0x00, 0x70,
0xf0, 0x00, 0x3c, 0x1c, 0x00, 0x00, 0x00, 0xe1, 0xe0, 0x00, 0x1e,
0x1c, 0x00, 0x00, 0x00, 0xe1, 0xc0, 0x00, 0x0f, 0x0e, 0x00, 0x00,
0x00, 0xc3, 0x81, 0xfc, 0x07, 0x0e, 0x00, 0x00, 0x00, 0x03, 0x83,
0xff, 0x07, 0x8e, 0x00, 0x00, 0x00, 0x07, 0x07, 0x8f, 0x83, 0x87,
0x00, 0x00, 0x00, 0x0f, 0x0f, 0x03, 0xc3, 0x87, 0x00, 0x00, 0x00,
0x1e, 0x0e, 0x01, 0xc3, 0x87, 0x00, 0x00, 0x00, 0x3c, 0x1c, 0x00,
0xe1, 0x87, 0x00, 0x00, 0x00, 0xf8, 0x1c, 0x30, 0xe1, 0x87, 0x00,
```

```
0x00, 0x07, 0xf0, 0x38, 0x70, 0xe1, 0x86, 0x00, 0x00, 0x07, 0xc0,
0x78, 0x70, 0xe3, 0x8e, 0x00, 0x00, 0x02, 0x00, 0xf0, 0xf0, 0xe3,
0x8e, 0x00, 0x00, 0x00, 0x01, 0xe0, 0xe0, 0xe3, 0x8e, 0x00, 0x00,
0x00, 0x03, 0xc1, 0xe1, 0xc3, 0x8e, 0x00, 0x00, 0x00, 0x0f, 0x83,
0xc3, 0xc3, 0x8e, 0x00, 0x00, 0x00, 0x7f, 0x07, 0x83, 0x83, 0x0e,
0x00, 0x00, 0x07, 0xfc, 0x0f, 0x07, 0x83, 0x0e, 0x00, 0x00, 0x07,
0xf0, 0x1e, 0x0f, 0x03, 0x0e, 0x00, 0x00, 0x07, 0x80, 0x7c, 0x1e,
0x03, 0x07, 0x00, 0x00, 0x00, 0x00, 0xf8, 0x3c, 0x03, 0x87, 0x80,
0x00, 0x00, 0x03, 0xf0, 0x78, 0x03, 0x83, 0xc0, 0x00, 0x00, 0x1f,
0xc0, 0xf0, 0x02, 0x00, 0x00, 0x00, 0x00, 0xff, 0x01, 0xe1, 0xc0,
0x00, 0x00, 0x00, 0x07, 0xfc, 0x03, 0xc3, 0xe1, 0xff, 0xc0, 0x00,
0x07, 0xe0, 0x0f, 0x87, 0xc7, 0xff, 0xf0, 0x00, 0x07, 0x00, 0x3f,
0x0f, 0x0f, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x7c, 0x3e, 0x1f, 0xff,
0xfe, 0x00, 0x00, 0x03, 0xf8, 0x7c, 0x3f, 0xff, 0xff, 0x00, 0x00,
0x1f, 0xe0, 0xf0, 0x7f, 0xff, 0xff, 0x00, 0x00, 0xff, 0x83, 0xe0,
0xfe, 0xff, 0xbf, 0x80, 0x01, 0xfc, 0x07, 0xc0, 0xfc, 0x7f, 0x1f,
0xc0, 0x01, 0xe0, 0x1f, 0x01, 0xf8, 0x3e, 0x0f, 0xc0, 0x00, 0x00,
0xfe, 0x01, 0xf8, 0x1c, 0x07, 0xe0, 0x00, 0x03, 0xf8, 0x13, 0xf8,
0x00, 0x0f, 0xe0, 0x00, 0x1f, 0xe0, 0x73, 0xfc, 0x00, 0x1f, 0xe0,
0x00, 0x7f, 0x81, 0xf3, 0xfe, 0x00, 0x3f, 0xe0, 0x00, 0xfc, 0x03,
0xe3, 0xff, 0x00, 0x7f, 0xe0, 0x00, 0x60, 0x0f, 0xc3, 0xff, 0x80,
0xff, 0xe0, 0x00, 0x00, 0x3f, 0x03, 0xff, 0x00, 0x7f, 0xe0, 0x00,
0x00, 0xfc, 0x03, 0xfe, 0x00, 0x3f, 0xe0, 0x00, 0x07, 0xf0, 0x13,
0xfc, 0x00, 0x1f, 0xe0, 0x00, 0x0f, 0xc0, 0x79, 0xf8, 0x08, 0x0f,
0xe0, 0x00, 0x0f, 0x01, 0xf9, 0xf8, 0x1c, 0x0f, 0xc0, 0x00, 0x00,
0x07, 0xf1, 0xfc, 0x3e, 0x1f, 0xc0, 0x00, 0x00, 0x1f, 0xc0, 0xfe,
0x7f, 0x3f, 0x80, 0x00, 0x00, 0x7e, 0x00, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0x00, 0x00, 0x00, 0xf0,
0x1f, 0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x1f, 0xff,
0xfc, 0x00, 0x00, 0x00, 0x01, 0xff, 0x8f, 0xff, 0xf8, 0x00, 0x00,
0x00, 0x03, 0xe0, 0xe3, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x01, 0x80,
0x00, 0x7f, 0x00, 0x00};
```

```
/******FINGERPRINT FAILED
ICONS*****
#define FINGERPRINT_FAILED_WIDTH 64
#define FINGERPRINT_FAILED_HEIGHT 64
const uint8_t PROGMEM FINGERPRINT_FAILED_BITS[] = {0x00, 0x00, 0x3f,
0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xfe, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0f, 0xc0, 0x1f, 0x80, 0x00, 0x00, 0x00, 0x00,
0x1e, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x78, 0x00, 0x00,
0xf0, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x38, 0x00, 0x00,
0x00, 0x01, 0xc0, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x03, 0x80,
0x00, 0x00, 0x0e, 0x00, 0x00, 0x00, 0x07, 0x00, 0x7f, 0xe0, 0x07,
0x00, 0x00, 0x00, 0x06, 0x01, 0xff, 0xf8, 0x03, 0x00, 0x00, 0x00,
0x0c, 0x03, 0xc0, 0x3c, 0x03, 0x80, 0x00, 0x00, 0x1c, 0x0f, 0x00,
0x0e, 0x01, 0x80, 0x00, 0x00, 0x18, 0x0c, 0x00, 0x03, 0x00, 0xc0,
0x00, 0x00, 0x18, 0x18, 0x00, 0x01, 0x80, 0xc0, 0x00, 0x00, 0x30,
0x38, 0x00, 0x01, 0xc0, 0xe0, 0x00, 0x00, 0x30, 0x30, 0x0f, 0x00,
0xc0, 0x60, 0x00, 0x00, 0x30, 0x30, 0x3f, 0xc0, 0xe0, 0x60, 0x00,
0x00, 0x70, 0x60, 0x78, 0xe0, 0x60, 0x60, 0x00, 0x00, 0x60, 0x60,
0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
0x30, 0x00, 0x00, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
0x60, 0x60, 0x60, 0x30, 0x30, 0x20, 0x00, 0x00, 0x60, 0x60, 0x60,
0x30, 0x30, 0x01, 0xe0, 0x00, 0x60, 0x60, 0x60, 0x30, 0x30, 0x0f,
0xfc, 0x00, 0x60, 0x60, 0x60, 0x30, 0x30, 0x3f, 0xff, 0x00, 0x60,
```

```
0x60, 0x60, 0x30, 0x18, 0x78, 0x03, 0x80, 0x60, 0x60, 0x60, 0x30,
0x1c, 0x60, 0x01, 0x80, 0x60, 0x60, 0x30, 0x38, 0xc0, 0xc0, 0x00,
0xc0, 0x00, 0x60, 0x30, 0x18, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0x60,
0x30, 0x18, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xe0, 0x30, 0x0c, 0x01,
0xc0, 0x00, 0xe0, 0x00, 0xc0, 0x18, 0x0e, 0x01, 0xc0, 0x00, 0xe0,
0x01, 0xc0, 0x00, 0xe0, 0x01, 0xc0, 0x1c,
0x03, 0x81, 0xc0, 0x00, 0xe0, 0x01, 0x80, 0x0c, 0x01, 0xc1, 0xc0,
0x00, 0xe0, 0x03, 0x80, 0x0e, 0x00, 0xf1, 0xc0, 0x00, 0xe0, 0x0f,
0x00, 0x06, 0x00, 0x01, 0xc0, 0x00, 0xe0, 0x3e, 0x01, 0x03, 0x00,
0x01, 0xc0, 0x00, 0xe0, 0x30, 0x03, 0x83, 0x80, 0x1f, 0xff, 0xff,
0xfe, 0x00, 0x03, 0x81, 0xc0, 0x3f, 0xff, 0xff, 0xff, 0x00, 0x07,
0xc0, 0xe0, 0x30, 0x00, 0x00, 0x03, 0x00, 0x0e, 0xc0, 0x78, 0x30,
0x00, 0x00, 0x03, 0x00, 0x3c, 0x60, 0x1e, 0x30, 0x00, 0x00, 0x03,
0x00, 0x78, 0x70, 0x0f, 0x30, 0x00, 0x00, 0x03, 0x03, 0xe0, 0x38,
0x03, 0x30, 0x00, 0x00, 0x03, 0x07, 0x80, 0x1c, 0x00, 0x30, 0x00,
0x00, 0x03, 0xc0, 0x00, 0x0f, 0x00, 0x30, 0x00, 0x00, 0x03, 0xc0,
0x00, 0x03, 0x80, 0x30, 0x01, 0xe0, 0x03, 0x00, 0x18, 0x01, 0xe0,
0x30, 0x03, 0xf0, 0x03, 0x00, 0x00, 0x18, 0x00, 0x7c, 0x30, 0x07, 0x38,
0x03, 0x00, 0x0c, 0x00, 0x00, 0x1f, 0x30, 0x06, 0x18, 0x03, 0x18, 0x0e,
0x00, 0x07, 0x30, 0x06, 0x18, 0x03, 0x0c, 0x07, 0x80, 0x00, 0x30,
0x07, 0x38, 0x03, 0x0e, 0x03, 0xc0, 0x00, 0x30, 0x03, 0x30, 0x03,
0x07, 0x00, 0xf0, 0x00, 0x30, 0x03, 0x30, 0x03, 0x03, 0x00, 0x7e,
0x00, 0x30, 0x03, 0x30, 0x03, 0x01, 0x80, 0x1f, 0xc0, 0x30, 0x03,
0x30, 0x03, 0x01, 0xc0, 0x03, 0xe1, 0x30, 0x07, 0xf8, 0x03, 0x00,
0xf0, 0x00, 0x01, 0x30, 0x03, 0xf0, 0x03, 0x00, 0x38, 0x00, 0x00,
0x30, 0x00, 0x00, 0x03, 0x00, 0x1e, 0x00, 0x00, 0x30, 0x00, 0x00,
0x03, 0x00, 0x07, 0xc0, 0x00, 0x30, 0x00, 0x00, 0x03, 0x00, 0x01,
0xff, 0x80, 0x3f, 0xff, 0xff, 0xff, 0x00, 0x00, 0x3f, 0x80, 0x1f,
0xff, 0xff, 0xfe};
```

```
//*****FINGERPRINT SCAN
ICONS*****
#define FINGERPRINT_SCAN_WIDTH 64
#define FINGERPRINT_SCAN_HEIGHT 64
const uint8_t PROGMEM FINGERPRINT_SCAN_BITS[] = {0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xfc, 0x7f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x03,
0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x80, 0x00, 0xf0, 0x00, 0x00,
0x00, 0x00, 0x0e, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x1c,
0x1f, 0xfc, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x7f, 0xfe, 0x0e,
0x00, 0x00, 0x00, 0x00, 0x78, 0xf8, 0x0f, 0x87, 0x00, 0x00, 0x00,
0x00, 0x71, 0xe0, 0x03, 0xc7, 0x00, 0x00, 0x00, 0x00, 0xe3, 0x80,
0x01, 0xc3, 0x80, 0x00, 0x00, 0x00, 0xc3, 0x83, 0xc0, 0xe3, 0x80,
0x00, 0x00, 0x00, 0xc7, 0x0f, 0xf0, 0x71, 0x80, 0x00, 0x00, 0x00,
0x06, 0x1f, 0xf8, 0x71, 0xc0, 0x00, 0x00, 0x00, 0x0e, 0x1c, 0x3c,
0x31, 0xc0, 0x00, 0x00, 0x00, 0x1c, 0x38, 0x1c, 0x31, 0xc0, 0x00,
0x00, 0x00, 0x38, 0x70, 0x0e, 0x39, 0xc0, 0x00, 0x00, 0x01, 0xf0,
0x71, 0x8e, 0x39, 0xc0, 0x00, 0x00, 0x03, 0xe0, 0xe1, 0x86, 0x31,
0xc0, 0x00, 0x00, 0x03, 0x81, 0xe3, 0x8e, 0x31, 0x80, 0x00, 0x00,
0x00, 0x03, 0xc3, 0x8e, 0x33, 0x80, 0x00, 0x00, 0x07, 0x87,
0x0c, 0x73, 0x80, 0x00, 0x00, 0x00, 0x1f, 0x0e, 0x1c, 0x73, 0x80,
0x00, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0xff, 0xff,
```

```

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x03, 0xf0, 0x1e, 0x3e, 0x1c, 0x00, 0x00, 0x00,
0x03, 0x80, 0x7c, 0x77, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x01, 0xf0,
0xe3, 0x07, 0xc0, 0x00, 0x00, 0x00, 0x07, 0xe3, 0xc3, 0x81, 0xf0,
0x00, 0x00, 0x00, 0x3f, 0x87, 0x81, 0xc0, 0x60, 0x00, 0x00, 0x01,
0xfc, 0x1f, 0x00, 0xf0, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x3c, 0x00,
0x7c, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf8, 0x78, 0x1f, 0x00, 0x00,
0x00, 0x00, 0x07, 0xe0, 0xfc, 0x0f, 0xc0, 0x00, 0x00, 0x00, 0x3f,
0x83, 0xef, 0x03, 0xc0, 0x00, 0x00, 0x00, 0xfc, 0x0f, 0x87, 0x80,
0x00, 0x00, 0x00, 0x00, 0x70, 0x1f, 0x03, 0xe0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x7c, 0x00, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x01, 0xf0,
0x00, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xc0, 0xf8, 0x0f, 0x00,
0x00, 0x00, 0x00, 0x1f, 0x03, 0xfe, 0x02, 0x00, 0x00, 0x00, 0x00,
0x0c, 0x0f, 0x8f, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0x03,
0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf8, 0x00, 0xf0, 0x00, 0x00,
0x00, 0x00, 0x01, 0xe0, 0x00, 0x00, 0x30, 0x00, 0x00, 0x00, 0x01,
0xc0, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xfe, 0x00,
0x00, 0x00, 0x00, 0x00, 0x0f, 0x8e, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00};

```

```

int doorLockDuration = 4000;
bool daftarKartu = false;
void setup()
{
  //===== SETUP relay =====//
  pinMode(relay, OUTPUT);
  digitalWrite(relay, LOW);

  //===== INITIALIZE THE SERIAL =====//
  Serial.begin(115200);
  Serial.println("Starting the System");

  //===== INITIALIZE OLED =====//
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3D))
  { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;)
      ; // Don't proceed, loop forever
  }
  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  // you can delet these three lines if you don't want to get the
  Adfruit logo appear
  display.display();
  delay(2000); // Pause for 2 seconds
  display.clearDisplay();

  //===== SETUP FIREBASE =====//
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

```

```

//===== SETUP TELEGRAM =====//
myBot.setTelegramToken(TOKEN);
// myBot.wifiConnect(WIFI_SSID,WIFI_PASS);
// SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V
internally
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
{ // Address 0x3D for 128x64
  Serial.println(F("SSD1306 allocation failed"));
  for (;;)
    ; // Don't proceed, loop forever
}
// Show initial display buffer contents on the screen --
// the library initializes this with an Adafruit splash screen.
// you can delete these three lines if you don't want to get the
Adafruit logo appear
display.display();
delay(2000); // Pause for 2 seconds
display.clearDisplay();

if (myBot.testConnection())
{
  Serial.println("Test Connection OK");
}
else
{
  Serial.println("Test Connection Error");
}

// inline keyboard - only a button called "My button"
myKbd.addButton("Users", CALLBACK_USERS,
CTBotKeyboardButtonQuery);
myKbd.addButton("Add", CALLBACK_ADD, CTBotKeyboardButtonQuery);
myKbd.addButton("Delete", CALLBACK_DELETE,
CTBotKeyboardButtonQuery);
myKbd.addButton("Drop", CALLBACK_DROP,
CTBotKeyboardButtonQuery);

//===== SETUP FINGERPRINT =====//
finger.begin(57600);
Serial.println("\n\nAdafruit finger detect test");

if (finger.verifyPassword())
{
  Serial.println("Found fingerprint sensor!");
  display.clearDisplay();
  display.drawBitmap(34, 0, FINGERPRINT_VALID_BITS,
FINGERPRINT_VALID_WIDTH, FINGERPRINT_VALID_HEIGHT, WHITE);
  display.display();
}
else
{
  Serial.println("Did not find fingerprint sensor :(");
  display.clearDisplay();
}

```

```

    display.drawBitmap(32, 0, FINGERPRINT_FAILED_BITS,
FINGERPRINT_FAILED_WIDTH, FINGERPRINT_FAILED_HEIGHT, WHITE);
    display.display();
    while (1)
    {
        delay(1);
    }
}

//===== SETUP WIFI =====//
WiFi.begin(WIFI_SSID, WIFI_PASS);
Serial.println("Connectiong to WiFi : " + String(WIFI_SSID));
while (WiFi.status() != WL_CONNECTED)
{
    displayWiFiStart();
    delay(300);
}

displayWiFiConnected();
}

void loop()
{

    TBMessage msg;
    String strMsg;

    FingerID = getFingerprintId(); // Get the Fingerprint ID from
the Scanner
    delay(50);

    if (FingerID > 0)
    {
        String nameUser = getNameFromDB(String(FingerID));
        nameUser.toUpperCase();
        myBot.sendMessage(msg.sender.id, nameUser + " telah masuk
menggunakan Fingerprint");
        digitalWrite(relay, HIGH); // turn relay on
        delay(doorLockDuration);
    }

    digitalWrite(relay, LOW); // turn relay on

    // put your main code here, to run repeatedly:

    String data = "";

    while (Serial.available() > 0)
    {
        data += char(Serial.read());
    }

    if (data != "")
    {
        digitalWrite(relay, HIGH); // turn relay on
        String nameUser = getNameFromDB(data);

```

```

if (nameUser != "")
{
    displayText("Silahkan masuk!");
    nameUser.toUpperCase();
    myBot.sendMessage(msg.sender.id, nameUser + " telah masuk
menggunakan Fingerprint");
    delay(doorLockDuration);
}
else
{
    displayText("Silahkan masuk!");
    delay(2000);
}
}

delay(500);

//-----

displayFingerprintId();

//-----

if (strMsg == "OK")
{
    //Perintah dari
telegram ke perangkat
    myBot.sendMessage(msg.sender.id, "*OK*"); //Balasan dari
perangkat ke Bot Telegram
}
else if (strMsg.substring(0, 3) == "ADD")
{
    int indexId = strMsg.indexOf(" ") + 1;
    int indexName = strMsg.indexOf(" ", indexId + 1);
    String strId = strMsg.substring(indexId, indexName);
    String nama = strMsg.substring(indexName + 1);
    Serial.println(strId);
    Serial.println(nama);
    id = strId.toInt();
    getFingerprintEnroll();
    Firebase.setString(fbData, "/USERS/" + strId, nama);
}
else if (strMsg == "DROP DATABASE")
{
    //Lampu dihidupkan
    myBot.sendMessage(msg.sender.id, "Now fingerprint is
&#42;empty&#42;"); //Balasan dari perangkat ke Bot Telegram
    deleteAllFingerprint();
}
else if (strMsg == "COMMAND")
{
    // the received message is a text message --> reply with the
inline keyboard
    myBot.sendMessage(msg.sender.id, "Welcome *" +
msg.sender.firstName + " " + msg.sender.lastName + "*\n Please
select button to running system Biometric and RFID :", myKbd);
}
}

```

```

}
else if (strMsg.substring(0, 8) == "ADD CARD")
{
    int indexId = strMsg.indexOf(" ",8) + 1;
    int indexName = strMsg.indexOf(" ", indexId + 1);
    String strId = strMsg.substring(indexId, indexName);
    String nama = strMsg.substring(indexName + 1);
    displayText("Dekatkan kartu!");
    String idCard = getIdCard();

    while (idCard != "")
    {
        Firebase.setString(fbData, "/USERS/" + getIdCard(), nama);
        myBot.sendMessage(msg.sender.id, "ID Card Sudah terdaftar");
//Balasan dari perangkat ke Bot Telegram
        displayText("ID Card Sudah terdaftar");
    }
}

// Function Register Card //
String getIdCard()
{
    String data = "";

    while (Serial.available() > 0)
    {
        data += char(Serial.read());
    }

    return data;
}
//=====FUNCTION
FINGERPRINT=====//

void displayFingerprintId()
{
    //Fingerprint has been detected
    if (FingerID > 0)
    {
        displayValidBits();
    }
//-----
//No finger detected
else if (FingerID == 0)
{
    displayStartBits();
}
//-----
//Didn't find a match
else if (FingerID == -1)
{
    displayInvalidBits();
}
//-----

```

```

    //Didn't find the scanner or there an error
    else if (FingerID == -2)
    {
        displayFailedBits();
    }
}

String getNameFromDB(String idUser)
{
    String url = "/USERS/" + idUser;
    Serial.println(url);
    Firebase.getString(fbData, url);
    String nama = fbData.stringData();
    Serial.println("SEND MESSAGE TO " + nama);
    return nama;
}

int getFingerprintId()
{
    uint8_t p = finger.getImage();
    switch (p)
    {
        case FINGERPRINT_OK:
            //Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            //Serial.println("No finger detected");
            return 0;
        case FINGERPRINT_PACKETRECEIVEERR:
            //Serial.println("Communication error");
            return -2;
        case FINGERPRINT_IMAGEFAIL:
            //Serial.println("Imaging error");
            return -2;
        default:
            //Serial.println("Unknown error");
            return -2;
    }
    // OK success!
    p = finger.image2Tz();
    switch (p)
    {
        case FINGERPRINT_OK:
            //Serial.println("Image converted");
            break;
        case FINGERPRINT_IMAGEMESS:
            //Serial.println("Image too messy");
            return -1;
        case FINGERPRINT_PACKETRECEIVEERR:
            //Serial.println("Communication error");
            return -2;
        case FINGERPRINT_FEATUREFAIL:
            //Serial.println("Could not find fingerprint features");
            return -2;
        case FINGERPRINT_INVALIDIMAGE:
            //Serial.println("Could not find fingerprint features");

```

```

    return -2;
default:
    //Serial.println("Unknown error");
    return -2;
}
// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK)
{
    //Serial.println("Found a print match!");
}
else if (p == FINGERPRINT_PACKETRECEIVEERR)
{
    //Serial.println("Communication error");
    return -2;
}
else if (p == FINGERPRINT_NOTFOUND)
{
    //Serial.println("Did not find a match");
    return -1;
}
else
{
    //Serial.println("Unknown error");
    return -2;
}
// 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 deleteFingerprint(int id)
{
    uint8_t p = -1;

    p = finger.deleteModel(id);

    if (p == FINGERPRINT_OK)
    {
        displayText("Deleted");
    }
    else if (p == FINGERPRINT_PACKETRECEIVEERR)
    {
        displayText("Communication error");
        return p;
    }
    else if (p == FINGERPRINT_BADLOCATION)
    {
        displayText("Could not delete in that location!");
        return p;
    }
    else if (p == FINGERPRINT_FLASHERR)

```

```

{
    displayText("Error writing to flash!");
    return p;
}
else
{
    displayText("Uknown error: 0x");
    return p;
}
}

uint8_t getFingerprintEnroll()
{

    int p = -1;
    displayScanBits();
    while (p != FINGERPRINT_OK)
    {
        p = finger.getImage();
        switch (p)
        {
            case FINGERPRINT_OK:
                displayValidBits();
                break;
            case FINGERPRINT_NOFINGER:
                displayText("Scanning..");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                displayInvalidBits();
                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:
            displayValidBits();
            break;
        case FINGERPRINT_IMAGEMESS:
            displayInvalidBits();
            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;
}

displayText("remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER)
{
    p = finger.getImage();
}
Serial.print("ID ");
Serial.println(id);
p = -1;
display.clearDisplay();
display.drawBitmap(34, 0, FINGERPRINT_SCAN_BITS,
FINGERPRINT_SCAN_WIDTH, FINGERPRINT_SCAN_HEIGHT, WHITE);
display.display();
while (p != FINGERPRINT_OK)
{
    p = finger.getImage();
    switch (p)
    {
        {
        case FINGERPRINT_OK:
            displayValidBits();
            break;
        case FINGERPRINT_NOFINGER:
            displayText(F("scanning"));
            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:
    displayValidBits();
    break;
case FINGERPRINT_IMAGEMESS:

```

```

Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECIEVEERR:
    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)
{
    displayValidBits();
}
else if (p == FINGERPRINT_PACKETRECIEVEERR)
{
    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)
{
    displayValidBits();
}
else if (p == FINGERPRINT_PACKETRECIEVEERR)
{
    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;
}
}

bool deleteAllFingerprint()
{
    if (finger.emptyDatabase() == FINGERPRINT_OK)
    {
        Serial.println("Now database is empty :)");
        return true;
    }
    else
    {
        Serial.println("Database can't emptied");
        return false;
    }
}

//=====FUNCTION
TELEGRAM=====//
String getMessage()
{
    TBMessage tbMsg;
    String msg = "No Message";
    if (myBot.getNewMessage(tbMsg))
    {
        msg = tbMsg.text;
        return msg;
    }
    else
    {
        return msg;
    }
}

bool sendMessage(String msg)
{
    TBMessage tbMsg;
    if (myBot.sendMessage(tbMsg.sender.id, msg))
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

```

    }
}

//===== FUNCTION DISPLAY
=====//
void displayWiFiStart()
{
    display.clearDisplay();
    display.setTextSize(1);      // Normal 1:1 pixel scale
    display.setTextColor(WHITE); // Draw white text
    display.setCursor(0, 0);     // Start at top-left corner
    display.print(F("Connecting to \n"));
    display.setCursor(0, 50);
    display.setTextSize(2);
    display.print(WIFI_SSID);
    display.drawBitmap(73, 10, WIFI_START_BITS, WIFI_START_WIDTH,
WIFI_START_HEIGHT, WHITE);
    display.display();
}

void displayWiFiConnected()
{
    display.clearDisplay();
    display.setTextSize(2);      // Normal 1:1 pixel scale
    display.setTextColor(WHITE); // Draw white text
    display.setCursor(8, 0);     // Start at top-left corner
    display.print(F("Connected \n"));
    display.drawBitmap(33, 15, WIFI_CONNECTED_BITS,
WIFI_CONNECTED_WIDTH, WIFI_CONNECTED_HEIGHT, WHITE);
    display.display();
}

void displayStartBits()
{
    display.clearDisplay();
    display.drawBitmap(32, 0, FINGERPRINT_START_BITS,
FINGERPRINT_START_WIDTH, FINGERPRINT_START_HEIGHT, WHITE);
    display.display();
}

void displayScanBits()
{
    display.clearDisplay();
    display.drawBitmap(34, 0, FINGERPRINT_SCAN_BITS,
FINGERPRINT_SCAN_WIDTH, FINGERPRINT_SCAN_HEIGHT, WHITE);
    display.display();
}

void displayValidBits()
{
    Serial.println("Image taken");
    display.clearDisplay();
    display.drawBitmap(34, 0, FINGERPRINT_VALID_BITS,
FINGERPRINT_VALID_WIDTH, FINGERPRINT_VALID_HEIGHT, WHITE);
    display.display();
}

```

```

void displayInvalidBits()
{
    Serial.println("Imaging error");
    display.clearDisplay();
    display.drawBitmap(34, 0, FINGERPRINT_INVALID_BITS,
FINGERPRINT_INVALID_WIDTH, FINGERPRINT_INVALID_HEIGHT, WHITE);
    display.display();
}

void displayFailedBits()
{
    Serial.println("Did not find fingerprint sensor :(");
    display.clearDisplay();
    display.drawBitmap(32, 0, FINGERPRINT_FAILED_BITS,
FINGERPRINT_FAILED_WIDTH, FINGERPRINT_FAILED_HEIGHT, WHITE);
    display.display();
}

void displayText(String msg)
{
    Serial.println(msg);
    display.clearDisplay();
    display.setTextSize(2);           // Normal 2:2 pixel scale
    display.setTextColor(WHITE);     // Draw white text
    display.setCursor(0, 0);         // Start at top-left corner
    display.print(msg);
    display.display();
}

```

ARDUINO

```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.

void setup()
{
  Serial.begin(115200); // Initiate a serial communication
  SPI.begin(); // Initiate SPI bus
  mfrc522.PCD_Init(); // Initiate MFRC522
}

void loop()
{
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent() )
  {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial() )
  {
    return;
  }
  //Show UID on serial monitor
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" :
" "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  content.toUpperCase();
  content.trim();
  if (content.substring(1) == "E7 C8 37 B4") //change here the UID
of the card/cards that you want to give access
  {
    digitalWrite(2, HIGH);
    delay(3000);
  }else {
    delay(3000);
  }
  Serial.println(content);
}
```