



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 095561

**SISTEM PENGAMAN PINTU RUMAH
MENGUNAKAN FINGERPRINT SCANNER
BERBASIS MIKROKONTROLER**

Mohammad Saiful Anwar
NRP 2213039005

Agam Abdillah
NRP 2213039011

Dosen Pembimbing
Ir. Rusdhianto Effendie A. K., M.T.
Subadi, S.Pd.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 095561

**HOME DOOR SECURITY SYSTEMS USING
FINGERPRINT SCANNER BASED ON
MICROCONTROLLER**

Mohammad Saiful Anwar
NRP 2213039005

Agam Abdillah
NRP 2213039011

Advisor
Ir. Rusdhianto Effendie A. K., M.T.
Subadi, S.Pd.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "Sistem Pengaman Pintu Rumah Menggunakan *Fingerprint Scanner* Berbasis Mikrokontroler" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 02 Juni 2016

Seey

Mohammad Saiful Anwar
NRP 2213039005



Agam Abdillah
NRP 2213039011

**SISTEM PENGAMAN PINTU RUMAH MENGGUNAKAN
FINGERPRINT SCANNER BERBASIS MIKROKONTROLER**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Elektro Industri
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing 1

Dosen Pembimbing 2


Ir. Rusdhianto Effendie A. K., M.T.
NIP. 19590218 198610 1 001


Subadi S.Pd.
NIP. 19630923 198603 1 018

**SURABAYA
JUNI, 2016**



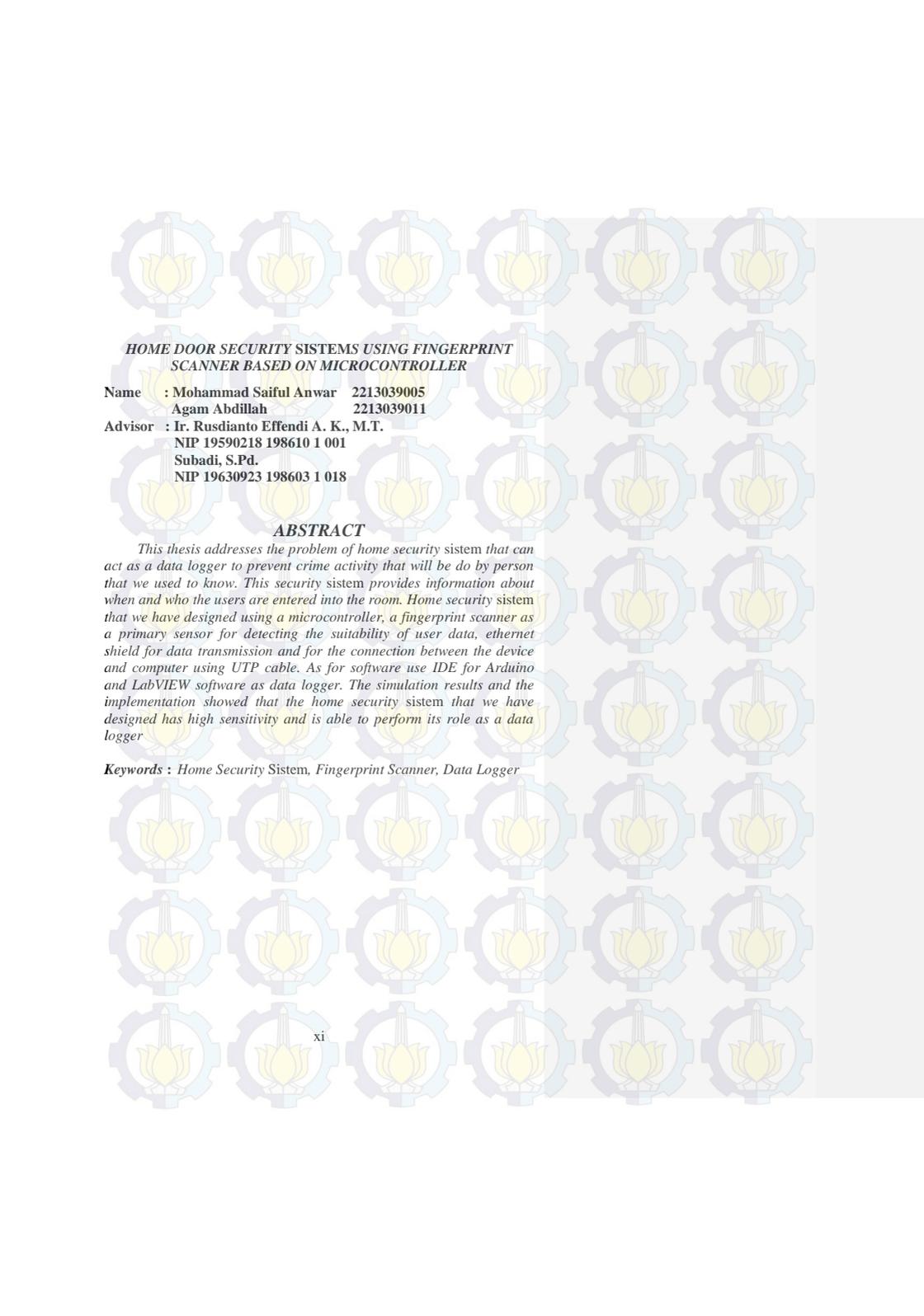
SISTEM PENGAMAN PINTU RUMAH MENGGUNAKAN *FINGERPRINT SCANNER* BERBASIS MIKROKONTROLER

Nama : Mohammad Saiful Anwar 2213039005
Agam Abdillah 2213039011
Pembimbing : Ir. Rusdianto Effendie A. K., M.T.
NIP 19590218 198610 1 001
Subadi, S.Pd.
NIP 19630923 198603 1 018

ABSTRAK

Pada Tugas akhir ini kami membahas sistem keamanan rumah yang praktis dan dapat berperan sebagai data *logger* untuk mencegah tindak kriminalitas yang dilakukan oleh orang dalam. Sistem keamanan ini memberikan informasi tentang kapan dan siapa pengguna yang masuk ke dalam ruangan. Sistem keamanan rumah yang telah kami rancang menggunakan mikrokontroler, *fingerprint scanner* sebagai sensor utama untuk mendeteksi kecocokan data pengguna, *ethernet shield* sebagai media pengiriman data dan koneksi data antara alat serta komputer menggunakan kabel UTP. Sedangkan untuk *software* menggunakan IDE untuk Arduino sebagai pemrograman inti dan *software* LabVIEW sebagai data *logger*. Hasil simulasi serta implementasi menunjukkan bahwa sistem keamanan rumah yang telah kami rancang memiliki sensitivitas yang tinggi dan mampu melakukan perannya sebagai data *logger*.

Kata Kunci : Sistem Keamanan Rumah, *Fingerprint Scanner*, Data *Logger*



**HOME DOOR SECURITY SYSTEMS USING FINGERPRINT
SCANNER BASED ON MICROCONTROLLER**

Name : Mohammad Saiful Anwar 2213039005
Agam Abdillah 2213039011
Advisor : Ir. Rusdianto Effendi A. K., M.T.
NIP 19590218 198610 1 001
Subadi, S.Pd.
NIP 19630923 198603 1 018

ABSTRACT

This thesis addresses the problem of home security system that can act as a data logger to prevent crime activity that will be do by person that we used to know. This security system provides information about when and who the users are entered into the room. Home security sistem that we have designed using a microcontroller, a fingerprint scanner as a primary sensor for detecting the suitability of user data, ethernet shield for data transmission and for the connection between the device and computer using UTP cable. As for software use IDE for Arduino and LabVIEW software as data logger. The simulation results and the implementation showed that the home security sistem that we have designed has high sensitivity and is able to perform its role as a data logger

Keywords : Home Security Sistem, Fingerprint Scanner, Data Logger



KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan D3 pada Bidang Studi Teknik Elektro Industri, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

SISTEM PENGAMAN PINTU RUMAH MENGGUNAKAN FINGERPRINT SCANNER BERBASIS MIKROKONTROLER

Dalam Tugas Akhir ini dirancang sebuah sistem keamanan pintu rumah menggunakan *fingerprint scanner* sebagai *user detector* agar pemilik rumah bisa masuk dan adanya data *logger* sebagai aplikasi pendukung sistem keamanan rumah.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Rusdhianto Effendie A. K., M.T., dan Bapak Subadi, S.Pd. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 02 Juni 2016

Penulis

DAFTAR ISI

HALAMAN

| | |
|--|-----------|
| HALAMAN JUDUL..... | i |
| HALAMAN JUDUL..... | i |
| PERNYATAAN KEASLIAN TUGAS AKHIR..... | v |
| HALAMAN PENGESAHAN..... | vii |
| ABSTRAK..... | ix |
| <i>ABSTRACT</i> | xi |
| KATA PENGANTAR..... | xiii |
| DAFTAR ISI..... | xv |
| DAFTAR GAMBAR..... | xvii |
| DAFTAR TABEL..... | xix |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Permasalahan..... | 1 |
| 1.3 Batasan Masalah..... | 1 |
| 1.4 Tujuan..... | 2 |
| 1.5 Metodologi Penelitian..... | 2 |
| 1.6 Sistematika Laporan..... | 2 |
| 1.7 Relevansi..... | 3 |
| BAB II TEORI DASAR..... | 5 |
| 2.1 Tinjauan Pustaka..... | 5 |
| 2.2 Sistem Keamanan Rumah..... | 6 |
| 2.3 <i>Fingerprint Scanner</i> | 6 |
| 2.4 Arduino UNO..... | 9 |
| 2.5 Arduino <i>Ethernet Shield</i> | 10 |
| 2.6 Solenoid <i>Doorlock</i> | 11 |
| 2.7 <i>Software LabVIEW</i> | 11 |
| BAB III PERANCANGAN SISTEM KONTROL..... | 13 |
| 3.1 Blok Fungsional Sistem..... | 13 |
| 3.2 Perancangan Perangkat Keras..... | 14 |
| 3.2.1 Perancangan Mekanik..... | 14 |
| 3.2.2 Perancangan Elektronik..... | 16 |

| | |
|---|-----------|
| 3.3 Perancangan Perangkat Lunak | 18 |
| BAB IV HASIL SIMULASI DAN IMPLEMENTASI | 27 |
| 4.1 Pengujian <i>Fingerprint Scanner</i> | 27 |
| 4.2 Pengujian <i>Driver Solenoid Doorlock</i> | 28 |
| 4.3 Pengujian Sistem secara Keseluruhan | 29 |
| BAB V PENUTUP | 33 |
| 5.1 Kesimpulan | 33 |
| 5.2 Saran | 33 |
| DAFTAR PUSTAKA | 35 |
| LAMPIRAN | 37 |
| 1. Program <i>Enrollment Sidik Jari</i> | 37 |
| 2. Program <i>Scanning Sidik Jari</i> | 42 |
| 3. Program <i>Setting IP Ethernet Shield</i> | 45 |
| 4. Program Sistem secara Keseluruhan | 46 |

DAFTAR GAMBAR

HALAMAN

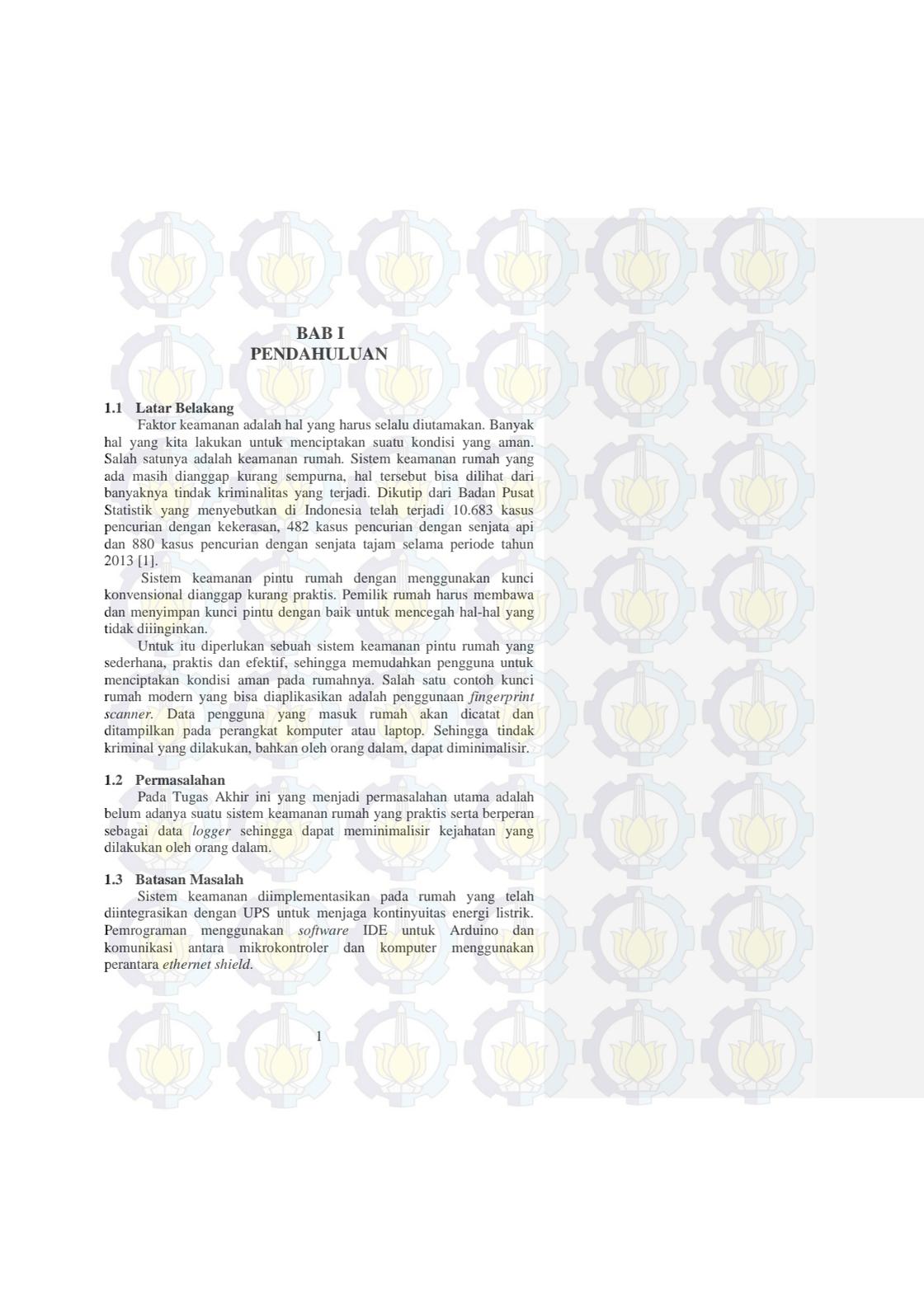
| | | |
|-------------|--|----|
| Gambar 2.1 | <i>Fingerprint Scanner dan Connector</i> | 8 |
| Gambar 2.2 | Arduino UNO | 9 |
| Gambar 2.3 | Arduino <i>Ethernet Shield</i> | 10 |
| Gambar 2.4 | <i>Solenoid Doorlock</i> | 11 |
| Gambar 2.5 | Contoh Blok Diagram <i>Software LabVIEW</i> | 12 |
| Gambar 3.1 | Blok Fungsional Sistem | 13 |
| Gambar 3.2 | Desain Sistem Keamanan Pintu Rumah..... | 14 |
| Gambar 3.3 | Desain <i>Solenoid Doorlock</i> dan <i>Handle Pintu</i> | 15 |
| Gambar 3.4 | Desain Pintu Tampak Depan..... | 15 |
| Gambar 3.5 | Koneksi <i>Fingerprint Scanner</i> ke Arduino..... | 16 |
| Gambar 3.6 | Rangkaian <i>Driver Solenoid Doorlock</i> | 17 |
| Gambar 3.7 | <i>Setting</i> Alamat IP dan Inisialisasi <i>Fingerprint Scanner</i> .. | 19 |
| Gambar 3.8 | <i>Flowchart</i> Pemrograman Arduino | 21 |
| Gambar 3.9 | Blok Diagram Program LabVIEW | 22 |
| Gambar 3.10 | Program <i>Enrollment</i> Sidik Jari..... | 23 |
| Gambar 3.11 | Program Penghapusan ID <i>Fingerprint Scanner</i> | 24 |
| Gambar 4.1 | Pengujian <i>Enrollment</i> Sidik Jari pada <i>Fingerprint Scanner</i> | 27 |
| Gambar 4.2 | Pengujian <i>Scanning</i> Kecocokan Sidik Jari..... | 27 |
| Gambar 4.3 | Rangkaian <i>Driver Solenoid Doorlock</i> | 28 |
| Gambar 4.4 | Gambar Pintu saat Kondisi <i>Standby</i> | 29 |
| Gambar 4.5 | Kondisi Lampu Indikator saat <i>Scanning</i> Sidik Jari yang Cocok | 30 |
| Gambar 4.6 | Kondisi Lampu Indikator saat <i>Scanning</i> Sidik Jari yang Salah | 30 |
| Gambar 4.7 | Pencatatan Data pada Notepad..... | 31 |



DAFTAR TABEL

HALAMAN

| | | |
|-----------|--|----|
| Tabel 2.1 | Deskripsi Arduino UNO | 10 |
| Tabel 4.1 | Tabel Pengukuran Arus <i>Solenoid Doorlock</i> | 29 |



BAB I PENDAHULUAN

1.1 Latar Belakang

Faktor keamanan adalah hal yang harus selalu diutamakan. Banyak hal yang kita lakukan untuk menciptakan suatu kondisi yang aman. Salah satunya adalah keamanan rumah. Sistem keamanan rumah yang ada masih dianggap kurang sempurna, hal tersebut bisa dilihat dari banyaknya tindak kriminalitas yang terjadi. Dikutip dari Badan Pusat Statistik yang menyebutkan di Indonesia telah terjadi 10.683 kasus pencurian dengan kekerasan, 482 kasus pencurian dengan senjata api dan 880 kasus pencurian dengan senjata tajam selama periode tahun 2013 [1].

Sistem keamanan pintu rumah dengan menggunakan kunci konvensional dianggap kurang praktis. Pemilik rumah harus membawa dan menyimpan kunci pintu dengan baik untuk mencegah hal-hal yang tidak diinginkan.

Untuk itu diperlukan sebuah sistem keamanan pintu rumah yang sederhana, praktis dan efektif, sehingga memudahkan pengguna untuk menciptakan kondisi aman pada rumahnya. Salah satu contoh kunci rumah modern yang bisa diaplikasikan adalah penggunaan *fingerprnt scanner*. Data pengguna yang masuk rumah akan dicatat dan ditampilkan pada perangkat komputer atau laptop. Sehingga tindak kriminal yang dilakukan, bahkan oleh orang dalam, dapat diminimalisir.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan utama adalah belum adanya suatu sistem keamanan rumah yang praktis serta berperan sebagai data *logger* sehingga dapat meminimalisir kejahatan yang dilakukan oleh orang dalam.

1.3 Batasan Masalah

Sistem keamanan diimplementasikan pada rumah yang telah diintegrasikan dengan UPS untuk menjaga kontinuitas energi listrik. Pemrograman menggunakan *software* IDE untuk Arduino dan komunikasi antara mikrokontroler dan komputer menggunakan perantara *ethernet shield*.

1.4 Tujuan

Pembuatan Sistem Pengaman Pintu Rumah Menggunakan *Fingerprint Scanner* Berbasis Mikrokontroler diharapkan berhasil menciptakan sebuah sistem kunci elektronik yang praktis dan mampu melakukan perannya sebagai data *logger*.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, perancangan sistem, pengujian alat, analisis data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur penulis mengumpulkan data dan informasi dengan membaca buku pustaka dan catatan kuliah, terutama mengenai materi yang berhubungan dengan *home security* sistem, *fingerprint scanner*, konsep data *logger*, serta materi lain yang berhubungan dengan masalah yang dibahas. Tahap perancangan sistem meliputi perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras meliputi pembuatan *prototype* pintu rumah dan pengkonstruksian *handle* pintu. Perancangan perangkat lunak meliputi pembuatan *flowchart*, pemrograman *software* menggunakan IDE untuk Arduino serta penggunaan LabVIEW sebagai *software* untuk data *logger*. Pengujian alat dilakukan setelah menggabungkan perangkat keras dan perangkat lunak menjadi satu sistem secara keseluruhan. Sistem diuji dengan memasukkan berbagai macam data sidik jari dan melihat respons dari alat terhadap variable yang dimasukkan. Data yang diperoleh dari pengujian alat digunakan sebagai dasar penyusunan laporan berupa buku Tugas Akhir.

1.6 Sistematika Laporan

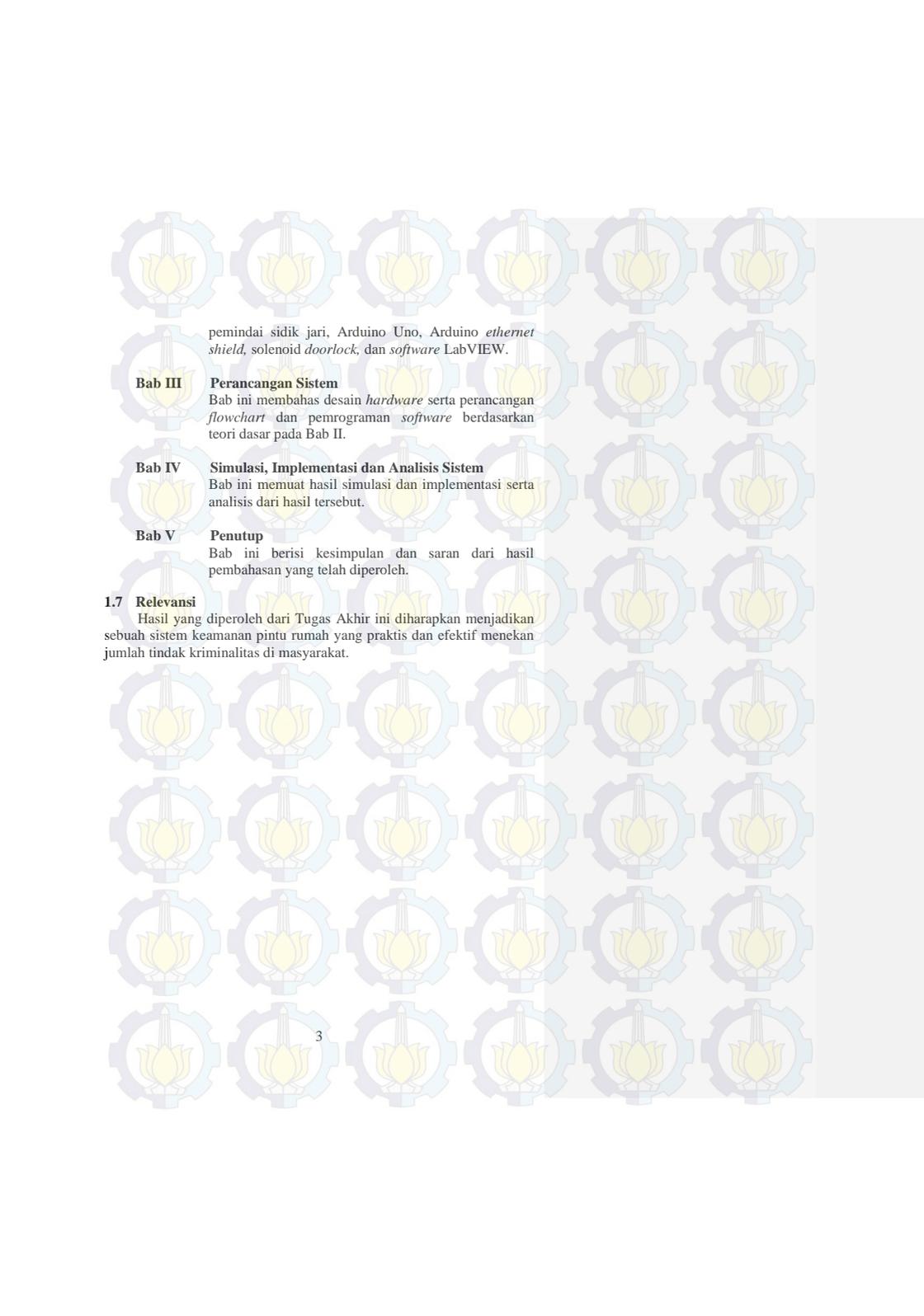
Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini memberikan penjelasan singkat tentang tinjauan pustaka, sistem keamanan rumah, sensor



pemindai sidik jari, Arduino Uno, Arduino *ethernet shield*, solenoid *doorlock*, dan *software* LabVIEW.

Bab III Perancangan Sistem

Bab ini membahas desain *hardware* serta perancangan *flowchart* dan pemrograman *software* berdasarkan teori dasar pada Bab II.

Bab IV Simulasi, Implementasi dan Analisis Sistem

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadikan sebuah sistem keamanan pintu rumah yang praktis dan efektif menekan jumlah tindak kriminalitas di masyarakat.



BAB II TEORI DASAR

2.1 Tinjauan Pustaka

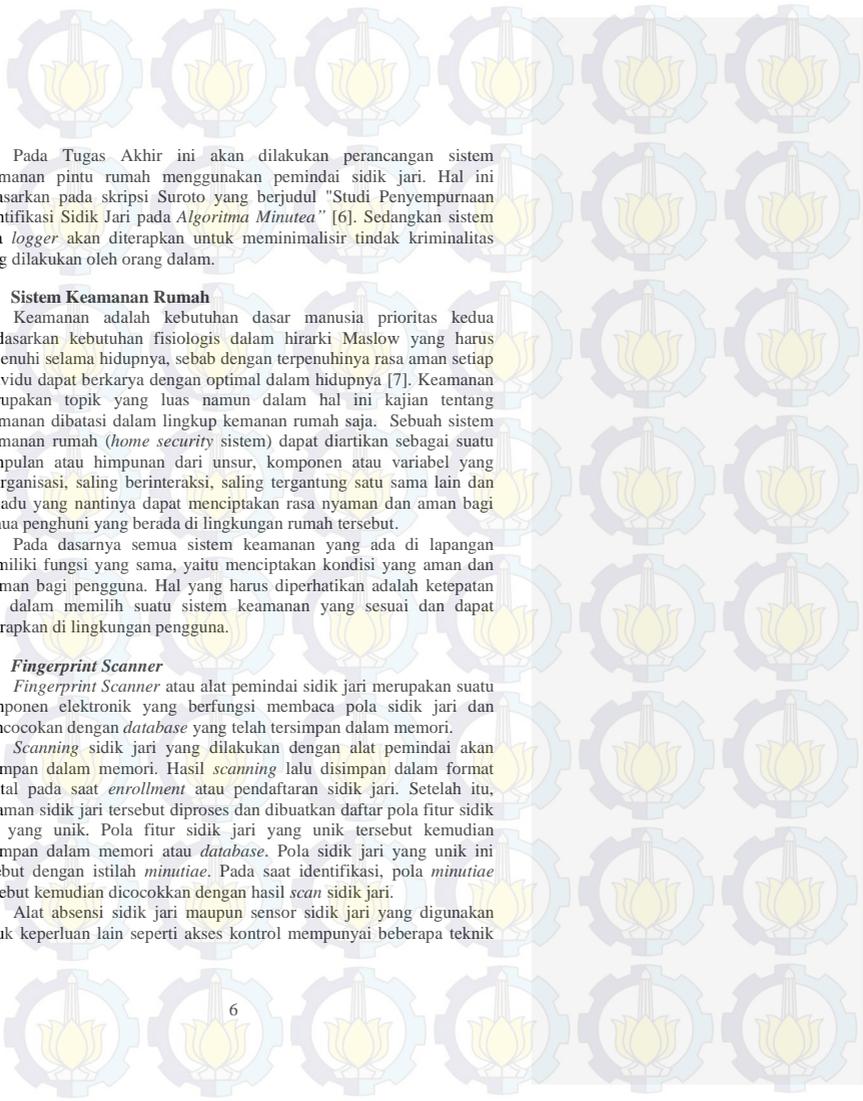
Ada banyak metode yang pernah diajukan untuk menyelesaikan permasalahan yang timbul pada sistem keamanan pintu rumah. Di antaranya menggunakan sensor gerak berbasis mikrokontroler AT89S51. Sistem ini menggunakan sensor *infrared* sebagai pendeteksi gerak untuk membuka pintu. Sistem ini dianggap kurang efektif karena tidak ada spesifikasi khusus yang harus dipenuhi pengguna agar bisa membuka pintu rumah [2].

Pada Tugas Akhir Gayung yang berjudul "Sistem Pengaman Rumah dengan *Security Password* Menggunakan Sensor Gerak Berbasis Mikrokontroler AT89S51", digunakan *security password* sebagai kode yang harus dimasukkan. Hasil ini dicapai terdapat kekurangan, yaitu sistem *password* dianggap kurang praktis karena pengguna harus menghafal digit *password* yang ada [3].

Pada penelitian yang dilakukan Darmawiguna dan Sunarya dalam Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMATI) Universitas Pendidikan Ganesha yang berjudul "Aplikasi *Motion Detection* untuk *Home Security* Sistem dengan Pelaporan Otomatis Berbasis *SMS Gateway*" dirancang sebuah sistem keamanan rumah dengan *motion* dan *SMS gateway*. Sistem ini mengirimkan SMS pada *user* dan pihak keamanan apabila ada perubahan posisi objek yang dimonitoring oleh sebuah *webcam*. Sistem seperti ini dianggap kurang efektif karena perubahan posisi objek masih kurang spesifik jika dijadikan indikator terjadinya sebuah tindak kriminalitas [4].

Pada skripsi Dwirani yang berjudul "Implementasi Sidik Jari untuk Sistem Absensi pada SDN II Cineam Tasikmalaya" digunakan *fingerprint scanner* sebagai sistem absensi yang ada pada perusahaan [5].

Sistem pemindai sidik jari dianggap sangat sensitif, sehingga setiap orang di dunia tidak akan memiliki kode sidik jari yang sama. Sidik jari dianggap sebagai bentuk identitas yang khusus dan spesifik, sehingga kemungkinan terjadinya *sabotase* pada suatu sistem adalah sangat kecil [6].



Pada Tugas Akhir ini akan dilakukan perancangan sistem keamanan pintu rumah menggunakan pemindai sidik jari. Hal ini didasarkan pada skripsi Suroto yang berjudul "Studi Penyempurnaan Identifikasi Sidik Jari pada *Algoritma Minutae*" [6]. Sedangkan sistem data *logger* akan diterapkan untuk meminimalisir tindak kriminalitas yang dilakukan oleh orang dalam.

2.2 Sistem Keamanan Rumah

Keamanan adalah kebutuhan dasar manusia prioritas kedua berdasarkan kebutuhan fisiologis dalam hirarki Maslow yang harus terpenuhi selama hidupnya, sebab dengan terpenuhinya rasa aman setiap individu dapat berkarya dengan optimal dalam hidupnya [7]. Keamanan merupakan topik yang luas namun dalam hal ini kajian tentang keamanan dibatasi dalam lingkup keamanan rumah saja. Sebuah sistem keamanan rumah (*home security* sistem) dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu yang nantinya dapat menciptakan rasa nyaman dan aman bagi semua penghuni yang berada di lingkungan rumah tersebut.

Pada dasarnya semua sistem keamanan yang ada di lapangan memiliki fungsi yang sama, yaitu menciptakan kondisi yang aman dan nyaman bagi pengguna. Hal yang harus diperhatikan adalah ketepatan kita dalam memilih suatu sistem keamanan yang sesuai dan dapat diterapkan di lingkungan pengguna.

2.3 Fingerprint Scanner

Fingerprint Scanner atau alat pemindai sidik jari merupakan suatu komponen elektronik yang berfungsi membaca pola sidik jari dan mencocokkan dengan *database* yang telah tersimpan dalam memori.

Scanning sidik jari yang dilakukan dengan alat pemindai akan disimpan dalam memori. Hasil *scanning* lalu disimpan dalam format digital pada saat *enrollment* atau pendaftaran sidik jari. Setelah itu, rekaman sidik jari tersebut diproses dan dibuatkan daftar pola fitur sidik jari yang unik. Pola fitur sidik jari yang unik tersebut kemudian disimpan dalam memori atau *database*. Pola sidik jari yang unik ini disebut dengan istilah *minutiae*. Pada saat identifikasi, pola *minutiae* tersebut kemudian dicocokkan dengan hasil *scan* sidik jari.

Alat absensi sidik jari maupun sensor sidik jari yang digunakan untuk keperluan lain seperti akses kontrol mempunyai beberapa teknik



pembacaan sidik jari. Teknik pembacaan sidik jari oleh mesin absensi sidik jari dapat dikelompokkan menjadi 4 tipe, yaitu teknik optik, ultrasonik, kapasitansi dan *thermal*.

Dengan teknik optik, pola sidik jari direkam dengan menggunakan cahaya. Alat perekam (*fingerprint scanner*) yang digunakan adalah berupa kamera digital. Tempat untuk meletakkan ujung jari disebut permukaan sentuh (*scan area*). Di bawah *scan area*, terdapat lampu atau pemancar cahaya yang menerangi permukaan ujung jari. Hasil pantulan cahaya dari ujung jari ditangkap oleh alat penerima yang selanjutnya menyimpan gambar sidik jari tersebut ke dalam memori.

Kelemahan metode ini adalah hasil *scanning* sangat tergantung dari kualitas sidik jari. Jika kualitas sidik jari miskin (*poor*) atau luka, maka kualitas hasil pembacaan akan tidak bagus. Kelemahan lain adalah teknik ini bisa diakali dengan jari palsu. Tapi teknik ini mempunyai keuntungan mudah dilakukan dan tidak membutuhkan biaya yang mahal.

Teknik *scanning* dengan metode ultrasonik ini hampir sama dengan teknik yang digunakan dalam dunia kedokteran. Dalam teknik ini, digunakan suara berfrekuensi sangat tinggi untuk menembus lapisan *epidermis* kulit. Suara frekuensi tinggi tersebut dibuat dengan menggunakan *transducer piezoelectric*. Setelah itu, pantulan energi tersebut ditangkap menggunakan alat yang sejenis. Pola pantulan ini dipergunakan untuk menyusun citra sidik jari yang dibaca. Dengan cara ini, tangan yang kotor tidak menjadi masalah. Demikian juga dengan permukaan scanner yang kotor tidak akan menghambat proses pembacaan.

Teknik kapasitansi ini menggunakan cara pengukuran kapasitansi untuk membentuk citra sidik jari. *Scan area* berfungsi sebagai lempeng kapasitor, dan kulit ujung jari berfungsi sebagai lempeng kapasitor lainnya. Karena adanya *ridge* (gundukan) dan *valley* (lembah) pada sidik jari, maka kapasitas dari kapasitor masing-masing orang akan berbeda. Kelemahan ini adalah adanya listrik statis pada tangan. Untuk menghilangkan listrik statis ini, tangan harus *digrounding*.

Teknik ini menggunakan perbedaan suhu antara *ridge* (gundukan) dengan *valley* (lembah) sidik jari untuk mengetahui pola sidik jari. Cara yang dilakukan adalah dengan menggosokkan ujung jari (*swap*) ke *scan area*. Bila ujung jari hanya diletakkan saja, dalam waktu singkat, suhunya akan sama karena adanya proses keseimbangan.



Gambar 2.1 *Fingerprint Scanner dan Connector*

Pada gambar 2.1 di atas adalah contoh modul pemindai sidik jari yang dapat dikoneksikan dengan Arduino. Dengan DSP prosesor yang berkecepatan tinggi, modul *fingerprint scanner* ini juga dapat diaplikasikan dengan *serial device* yang lain, seperti MSP430, AVR, PIC, STM32, ARM dan FPGA *device*. Modul ini memiliki memori yang dapat menyimpan hingga 1000 data sidik jari.

Fingerprint scanner memiliki kemampuan pembacaan sidik jari dengan tingkat sensitivitas yang tinggi baik dalam keadaan basah maupun kering. Selain itu alat ini memiliki kecepatan tinggi saat melakukan sistem pemindaian, pencarian dan perbandingan pola sidik jari.

Dengan fitur-fitur dan segala keunggulan tersebut, *fingerprint scanner* ini dapat difungsikan dalam berbagai bidang, terutama yang bersangkutan dengan masalah keamanan. Dapat difungsikan sebagai piranti absensi modern, saklar elektronik maupun pengganti *password* dengan sensitivitas yang tinggi.

2.4 Arduino UNO



Gambar 2.2 Arduino UNO

Pada Gambar 2.2 di atas merupakan contoh board Arduino Uno. Arduino Uno adalah *board* berbasis mikrokontroler pada ATmega 328. *Board* ini memiliki 14 digital *input/output* pin (dimana 6 pin dapat digunakan sebagai output PWM), 6 *input* analog, 16 MHz osilator kristal, koneksi USB, *jack* listrik dan tombol *reset*. Pin – pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tekanan bisa didapat dari adaptor AC – DC atau baterai untuk menggunakannya (Arduino, Inc., 2009).

Arduino Uno berbeda dengan semua *board* sebelumnya karena Arduino Uno ini tidak menggunakan *chip driver* FTDI *USB-to-serial*. Melainkan menggunakan fitur dari ATmega 16U2 yang diprogram sebagai konverter *USB-to-serial*.

Pada *pin out* Arduino UNO ditambahkan SDA dan SCL pin yang dekat ke pin *aref* dan dua pin baru lainnya ditempatkan dekat ke pin *RESET*, dengan I/O REF yang memungkinkan sebagai *buffer* untuk beradaptasi dengan tegangan yang disediakan dari *board* sistem. Pengembangannya, sistem lebih kompatibel dengan prosesor yang menggunakan AVR, yang beroperasi pada tegangan 5V. Sirkuit *reset* pada Arduino unth menyatel kembali ke program awal. ATmega 16U2 diganti dengan 8U yang digunakan sebagai konverter *USB-to-serial*.

Tabel 2.1 Deskripsi Arduino UNO

| | |
|-----------------------------|------------------------------|
| Mikrokontroler | ATMega 328 |
| Operating Voltage | 5 V |
| Input Voltage (recommended) | 7 – 12 V |
| Input Voltage (limit) | 6 – 20 V |
| Digital I/O Pins | 14 (6 diantaranya input PWM) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3,3V Pin | 50 mA |
| Flash Memory | 32 KB |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Clock Speed | 16 MHz |

2.5 Arduino Ethernet Shield



Gambar 2.3 Arduino Ethernet Shield

Pada Gambar 2.3 di atas merupakan contoh *ethernet shield* yang umum di pasaran. *Ethernet Shield* menambah kemampuan arduino board agar terhubung ke jaringan komputer. *Ethernet shield* berbasis *chip ethernet* Wiznet W5100. *Ethernet library* digunakan dalam menulis program agar arduino board dapat terhubung ke jaringan dengan menggunakan arduino *ethernet shield*.



Gambar 2.4 *Solenoid Doorlock*

Pada *ethernet shield* terdapat sebuah *slot micro SD*, yang dapat digunakan untuk menyimpan *file* yang dapat diakses melalui jaringan. *Onboard micro SD card reader* diakses dengan menggunakan *SD card library*. *Arduino board* berkomunikasi dengan *W5100* dan *SD card* menggunakan bus *SPI (Serial Peripheral Interface)*. Komunikasi ini diatur oleh *library SPI.h* dan *Ethernet.h*. Bus *SPI* menggunakan pin digital 11, 12 dan 13 pada *Arduino Uno*.

2.6 Solenoid Doorlock

Pada Gambar 2.4 di atas merupakan contoh *solenoid doorlock* yang ada di pasaran. *Solenoid doorlock* adalah alat elektronik yang dibuat khusus untuk pengunci pintu. Alat ini sering digunakan pada kunci pintu otomatis. *Solenoid* ini akan bekerja apabila diberi tegangan. Tegangan *solenoid doorlock* ini rata-rata yang dijual dipasaran adalah 12 volt tapi ada juga yang 6 volt dan 24 volt.

Jika kita akan merangkai kunci pintu elektronik tentunya kita membutuhkan *solenoid doorlock* sebagai penguncinya. Pada kondisi normal solenoid dalam posisi tuas terkunci. Jika diberi tegangan tuas akan membuka kunci pintu. *Solenoid* ini bisa digabungkan dengan sistem pengunci elektrik berbasis *RFID* dan *password*. Cocok dipakai untuk pengunci pintu, lemari maupun brankas.

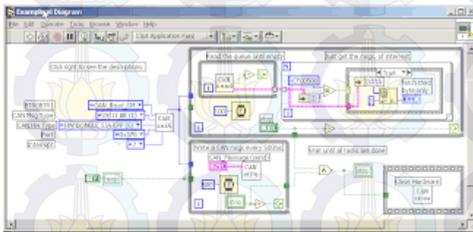
2.7 Software LabVIEW

LabVIEW (singkatan dari *Laboratory Virtual Instrumentation Engineering Workbench*) adalah perangkat lunak komputer untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali instrumentasi serta automasi industri yang pertama kali dikembangkan oleh perusahaan *National Instruments* pada tahun 1986. Perangkat lunak

ini dapat dijalankan pada sistem operasi Linux, Unix, Mac OS X dan Windows.

Pemrograman pada *software* LabVIEW dianggap lebih mudah dan menarik karena menggunakan blok diagram sebagai bahasa pemrogramannya. Ketika error terjadi pun dapat dimonitoring di mana letak terjadinya kesalahan.

Pada Gambar 2.5 di bawah ini merupakan contoh blok diagram yang ada di LabVIEW. Pada *software* LabVIEW memiliki beberapa keistimewaan dibanding *software* pemrograman yang lain. Dengan menggunakan konsep blok diagram akan lebih memudahkan kita dalam melakukan *programming* daripada menggunakan *software* yang menggunakan kata-kata sebagai dasar pemrogramannya



Gambar 2.5 Contoh Blok Diagram *Software* LabVIEW

BAB III PERANCANGAN ALAT

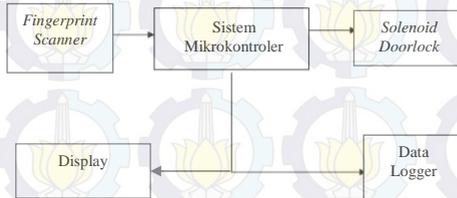
Commented [a1]: Disini juga dijelaskan kelogisan pembelian komponen berkali2 karena rusak

3.1 Blok Fungsional Sistem

Perancangan sistem dalam pembuatan alat ini secara garis besar disertai urutan dan cara kerja alat ini di ilustrasikan pada Gambar 3.1.

Dari Gambar diatas dapat dilihat bahwa sistem tersebut terdiri dari beberapa blok fungsional yaitu;

- Pemindai sidik jari (*fingerprint scanner*), digunakan untuk mengetahui kecocokan sidik jari pengguna dengan *database* yang ada di memori
- *Arduino UNO*, merupakan mikrokontroler yang berfungsi sebagai pengontrol perangkat elektronik dan dapat menyimpan program didalamnya.
- *Solenoid Doorlock* digunakan sebagai aktuator untuk menjaga pintu tetap terkunci atau terbuka.
- Data *Logger* ditampilkan pada PC atau laptop untuk mengetahui siapa dan kapan seseorang masuk ruangan.
- Tampilan (*Display*), untuk tampilan digunakan LED sebagai indikator bahwa sidik jari telah dikonfirmasi kebenarannya.



Gambar 3.1 Blok Fungsional Sistem

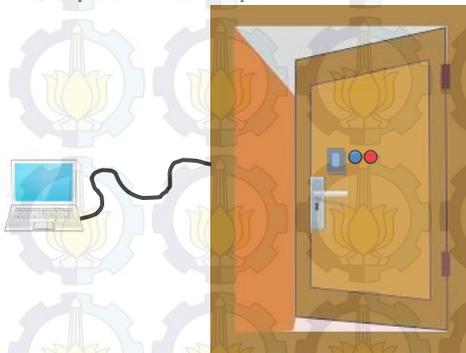
3.2 Perancangan Perangkat Keras

Pada perancangan perangkat keras ini, prosesnya dibagi menjadi 2 bagian, yaitu perancangan mekanik dan elektrik. Masing-masing perancangan tersebut selanjutnya akan dibahas lebih mendalam pada sub bab berikutnya.

3.2.1 Perancangan Mekanik

Perangkat untuk membuka pintu dengan *fingerprint scanner* tersebut merupakan alat untuk membuka kunci pintu solenoid *doorlock* dengan kemudahan dan tingkat kepraktisan yang tinggi. Secara umum perangkat keras ini terdiri dari *fingerprint scanner* sebagai masukan, penampil data berupa laptop atau komputer, sistem mikrokontroler, pintu yang sudah didesain sedemikian rupa seperti Gambar 3.2 di bawah ini.

Untuk menyesuaikan desain alat agar dapat bekerja maksimal, maka diperlukan desain *handle* pintu yang tepat. Komponen solenoid *doorlock* harus berada di dalam *handle* pintu agar alat dapat bekerja maksimal. Selain itu, desain solenoid *doorlock* yang dimasukkan ke dalam *handle* pintu akan menambah praktis alat tersebut.



Gambar 3.2 Desain Sistem Keamanan Pintu Rumah



Gambar 3.3 Desain Solenoid Doorlock dan Handle Pintu



Gambar 3.4 Desain Pintu Tampak Depan

Desain seperti pada Gambar 3.3 akan memudahkan kita untuk mengaplikasikannya pada pintu-pintu rumah biasa. Dengan desain seperti di atas, pemasangan *handle* pintu akan lebih praktis dan efisien. Solenoid *doorlock* yang biasanya dipasang di luar *body* pintu, dimodifikasi terpasang di dalam *handle* pintu.

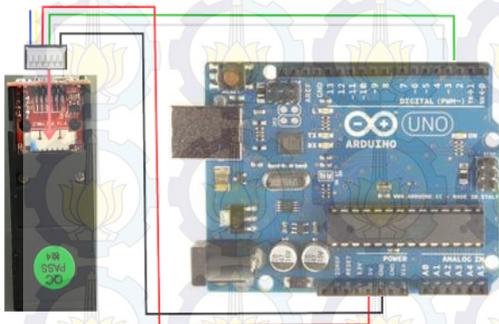
Pada Gambar 3.4 pemasangan *fingerprint scanner* akan diletakkan di sisi depan pintu. Dilengkapi dengan LED yang akan menjadi indikator kecocokan sidik jari pengguna dan *database* yang tersimpan di memori. Dalam keadaan semula LED merah akan menyala, jika *fingerprint scanner* mendeteksi kecocokan sidik jari pengguna maka LED biru akan menyala dan pintu dapat dibuka.

Pemasangan *fingerprint scanner* akan diletakkan di sisi depan pintu. Dilengkapi dengan LED yang akan menjadi indikator kecocokan sidik jari pengguna dan *database* yang tersimpan di memori. Dalam keadaan semula LED merah akan menyala, jika *fingerprint scanner* mendeteksi kecocokan sidik jari pengguna maka LED biru akan menyala dan pintu dapat dibuka.

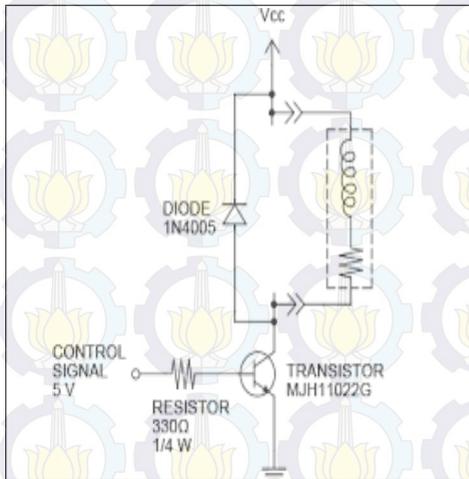
3.2.2 Perancangan Elektronik

Perancangan elektronik ini meliputi desain layout PCB serta pengkabelan. Rangkaian elektrik pada *plant* ini meliputi rangkaian *fingerprint scanner connections*, *driver solenoid doorlock*, Arduino UNO dan *Ethernet Shield*.

Pada Gambar 3.5 digambarkan koneksi dari *fingerprint scanner* ke Arduino. Pin 1 dengan kabel warna hitam adalah *ground*. Pin 2 dan 3 dengan warna putih dan hijau adalah *data input* dan *data output*. Dikoneksikan dengan pin 2 dan 3 dari Arduino. Pin 4 dengan kabel warna merah adalah untuk *power input* 5 Volt. Sedangkan pin 5 dan 6 adalah untuk *induction signal output* dan *touch induction power input* (biasanya diabaikan).



Gambar 3.5 Koneksi Fingerprint Scanner ke Arduino



Gambar 3.6 Rangkaian Driver Solenoid Doorlock

Pada Gambar 3.6 rangkaian *driver solenoid doorlock* menggunakan transistor Darlington sebagai penguat sinyal dari mikrokontroler. Hal tersebut dikarenakan *output* dari mikrokontroler kurang kuat untuk mengaktifkan *solenoid doorlock* tersebut. Sinyal *input* pada rangkaian transistor Darlington akan dikuatkan dua kali ukuran semula.

Sedangkan koneksi ke Arduino secara lengkap adalah sebagai berikut. Arduino memiliki 14 *digital input output* dan 6 *analog input output*. Pada arduino UNO juga terdapat *pin* yang menggunakan sinyal PWM sebagai *input* dan *output* yaitu *pin* 3, 5, 6, 9, 10, dan 11. Pada Tugas Akhir ini digunakan beberapa buah *input output pin* dengan rincian sebagai berikut:

- *Pin 2* : *input* data dari *fingerprint scanner*
- *Pin 3* : *output* data ke *fingerprint scanner*
- *Pin 5* : *output* ke lampu indikator
- *Pin 6* : *output* ke lampu indikator
- *Pin 7* : *output* ke lampu indikator
- *Pin 8* : *output* ke *driver solenoid doorlock*

3.3 Perancangan Perangkat Lunak

Pada Tugas Akhir ini *software* yang digunakan adalah program arduino UNO untuk membuat dan merencanakan program dalam bahasa C. Pemrograman *software* arduino dirancang dengan menggunakan *software* yang bernama IDE Arduino dengan menggunakan bahasa pemrograman C. Arduino sangatlah berbeda sekali dengan mikrokontroler karena bahasa ini lebih mudah digunakan dari pada mikrokontroler. Karena dalam arduino sudah terdapat beberapa *library* yang sudah digunakan untuk merancang pemrograman yang diinginkan.

Dalam perancangan program pada *software* arduino dengan fungsi terkait diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu. Tahapan tersebut adalah membuat algoritma dari alat yang sudah kita jalankan setelah membuat *flowchart* dari algoritma kita tersebut agar alat tersebut lebih sederhana. Setelah itu barulah kita memprogram fungsi terkait yang dikodingkan dalam bahasa C.

Untuk memprogram arduino juga harus dilakukan beberapa tahapan sebagai berikut:

Dalam pemrograman *software* arduino harus di *setting* terlebih dahulu *board* arduino agar penggunaan arduino cocok. Dalam purwarupa kali ini arduino menggunakan arduino UNO. Untuk *setting board* arduino bisa masuk ke *tools – board* – setelah itu pilihlah *board* arduino yang sesuai.

Serial ini merupakan kabel arduino yang dihubungkan kepada komputer atau laptop. *Serial* ini mempunyai dua fungsi yang bisa digunakan. Pertama *serial port* digunakan untuk mendownload program dari arduino yang kedua *serial* digunakan sebagai komunikasi *serial* pada arduino dengan komputer. *Setting serial* bisa masuk *tools – serial* - lali pilih COM yang sesuai dengan arduino yang terpasang.

Apabila program tidak dapat di *download* karena *serial port*, maka cek terlebih dahulu *serial* yang benar pada *device manager*. Lalu dalam *software* arduino untuk memilih *serial port*nya samakan dengan *serial port* untuk arduino dalam *device manager* tersebut. Untuk masuk ke

device manager dapat masuk start windows, lalu ketikkan *device manager* klik dua kali dan masuk ke dalam port COM.

Karena kita menggunakan *ethernet shield* sebagai media komunikasi, maka kita harus melakukan *setting* terlebih dahulu untuk mengatur alamat IP dan *port* yang digunakan. Selain itu kita juga harus melakukan inisialisasi pada program *fingerprint scanner*.



```
File Edit Sketch Tools Help
coba_ethernet2
Based on ChatServer example by David A. Mellis

*/
#include <SPI.h>
#include <Ethernet.h>
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 100, 11);

//server menggunakan port 5 (client juga harus menggunakan port yg
EthernetServer ArduinoTCP = EthernetServer(5);
boolean alreadyConnected = false; // apakah sudah terkoneksi sebel
int getFingerprintIDez();
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

Done compiling.
```

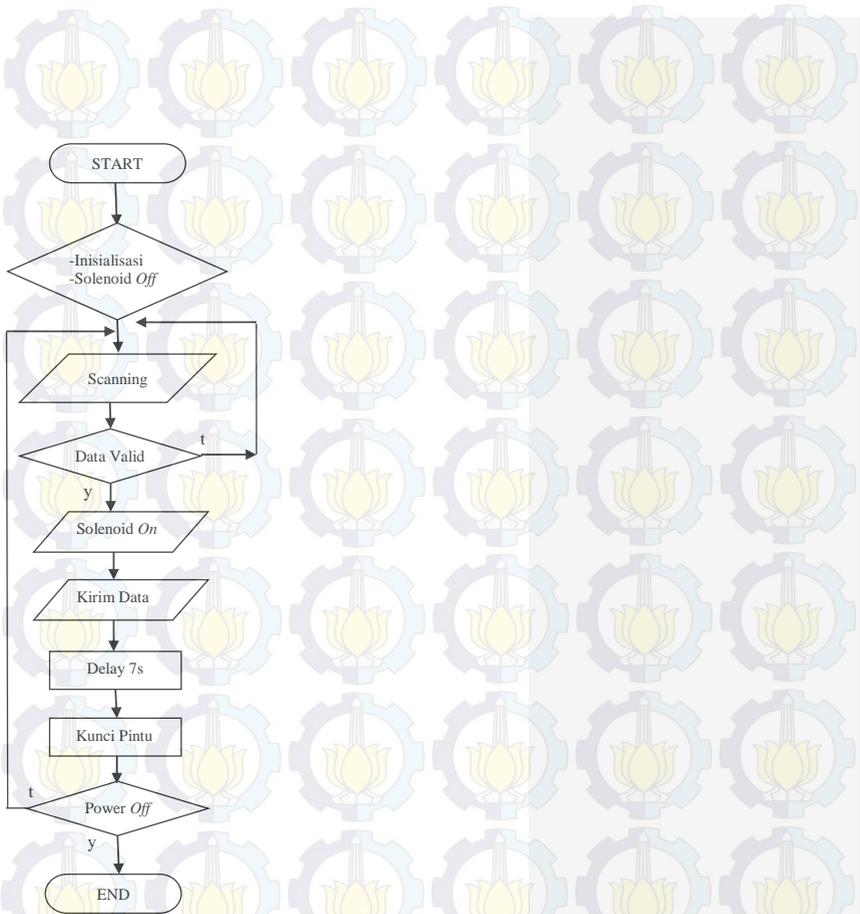
Gambar 3.7 *Setting* Alamat IP dan Inisialisasi *Fingerprint Scanner*



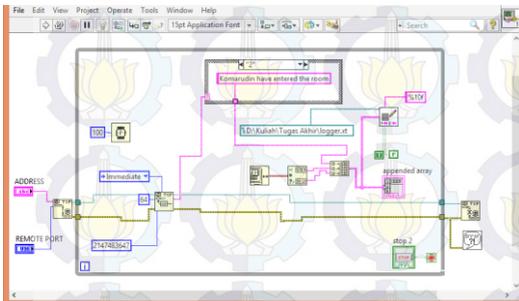
Pada Gambar 3.7 merupakan daftar program untuk melakukan inisialisasi pada *ethernet shield* dan *fingerprint scanner*. Dalam program di atas IP address diatur pada alamat 192.168.100.11 dan port diatur pada port nomor 5.

Dalam perancangan program pada *software arduino* dengan fungsi terkait diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu. Tahapan tersebut adalah membuat algoritma dari alat yang sudah kita jalankan setelah membuat *flowchart* dari algoritma kita tersebut agar alat tersebut lebih sederhana. Setelah itu barulah kita memprogram fungsi terkait yang dikodingkan dalam bahasa C.

Sebelum menyusun sebuah program yang akan kita upload ke Arduino, terlebih dulu kita harus menyusun terlebih dahulu algoritma berpikir kita dan mendesain sebuah *flowchart* seperti pada Gambar 3.8. Hal ini akan mempermudah kita dalam menuangkan pola pikir kita ke dalam bahasa pemrograman yang kita gunakan. Berikut ini algoritma program utama dari sistem keamanan pintu rumah menggunakan *fingerprint scanner*:



Gambar 3.8 Flowchart Pemrograman Arduino



Gambar 3.9 Blok Diagram Program LabVIEW

Pada Gambar 3.9 ditampilkan program *software* LabVIEW secara keseluruhan mulai dari proses pengambilan alamat IP sampai dengan proses penyimpanan data dalam memori laptop.

Setelah melakukan *programming* dan pengiriman data dari *fingerprint scanner* menggunakan *software* Arduino, maka perlu dilakukan pengaturan program menggunakan *software* LabVIEW. Program pada LabVIEW digunakan untuk membaca data yang dikirimkan oleh *ethernet shield* melalui perantara kabel UTP (*Unshielded Twisted Pair*).

Penggunaan kabel UTP ini dimaksudkan agar jarak antara *fingerprint scanner* dan komputer sebagai memori penyimpan data dapat lebih jauh. Hal tersebut lebih menguntungkan daripada menggunakan komunikasi serial yang jarak transfer datanya lebih pendek, sehingga penempatan komputer tidak harus berada di dekat *fingerprint scanner*.

Proses pengambilan data yang dikirim oleh *ethernet shield* akan berjalan apabila terdapat kesesuaian antara *IP address* dan *remote port* yang telah diupload ke *ethernet shield* dengan *IP address* dan *remote port* yang dimasukkan lewat *software* LabVIEW.

Setelah data ID yang dikirimkan oleh *fingerprint scanner* lewat *ethernet shield* didapatkan oleh komputer, maka *software* LabVIEW akan mengambil data waktu dan tanggal pada komputer. Ketiga data yang telah didapatkan akan diolah menjadi sekumpulan data yang siap untuk ditampilkan.

Keseluruhan data yang telah diambil oleh *software* LabVIEW akan diubah ke dalam file dengan format *.xt* dan akan disimpan pada memori laptop atau komputer. Dengan proses tersebut maka sistem akan dapat melakukan tugasnya sebagai data *logger*.

Sistem pengaman pintu rumah ini didesain agar hanya bisa diakses oleh orang yang telah didaftarkan oleh *admin*. Untuk melakukan pendaftaran *user baru* admin akan menjalankan program *enrollment* sidik jari.



```
File Edit Sketch Tools Help
enroll
{
  Serial.println("Ready to enroll a fingerprint! Please Type in the ID #");
  id = readnumber();
  Serial.print("Enrolling ID #");
  Serial.println(id);

  while (! getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {
  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
      case FINGERPRINT_OK:
        Serial.println("Image taken");
        break;
    }
  }
}
```

Gambar 3.10 Program *Enrollment* Sidik Jari

Pada Gambar 3.10 merupakan program untuk pendaftaran sidik jari untuk *user* yang baru. Pada saat pendaftaran, *user* baru akan diminta memasukkan data *image* ke dalam memori *fingerprint scanner* beserta nomor ID untuk *user* baru. Ketika nomor ID dan data *image* sidik jari telah tersimpan di memori maka *user* telah bisa mengakses sistem keamanan tersebut.

Selain melakukan pendaftaran *user* baru, sistem keamanan pintu menggunakan *fingerprint scanner* ini juga dapat menghapus memori data *user* yang sudah tidak terpakai. *Admin* akan menjalankan program *delete fingerprint memori* dan akan memasukkan nomor ID yang akan dihapus dari memori.



```
File Edit Sketch Tools Help
delete
id += c - '0';
}
Serial.print("deleting ID #");
Serial.println(id);

deleteFingerprint(id);
}

uint8_t deleteFingerprint(uint8_t id) {
  uint8_t p = -1;

  p = finger.deleteModel(id);

  if (p == FINGERPRINT_OK) {
    Serial.println("Deleted!");
  } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
  }
}
```

Gambar 3.11 Program Penghapusan ID *Fingerprint Scanner*



Pada Gambar 3.11 merupakan program untuk penghapusan sidik jari untuk *user* yang telah tidak terpakai. Pada saat penghapusan ID, *admin* akan memasukkan nomer ID yang akan dihapus. Setelah *user* ID dihapus dari memori, maka *user* tersebut sudah tidak bisa lagi mengakses sistem keamanan menggunakan *fingerprint scanner* ini.

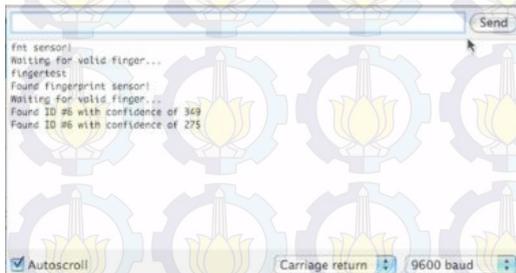
BAB IV PENGUJIAN DAN ANALISA DATA

4.1 Pengujian *Fingerprint Scanner*

Fingerprint Scanner diuji dengan menggunakan program yang diupload ke dalam Arduino. Ada beberapa macam program yang digunakan untuk menguji *fingerprint scanner*, diantaranya pendaftaran sidik jari (*enroll*) dan pengujian kecocokan sidik jari.



Gambar 4.1 Pengujian *Enrollment* Sidik Jari pada *Fingerprint Scanner*



Gambar 4.2 Pengujian *Scanning* Kecocokan Sidik Jari

Commented [a2]: Menyajikan data, semakin banyak data semakin bagus
Berisi hasil dan .engapa data seperti itu. pengujian secara detail itu bagus

Pengujian secara global secara konteks masing2

Pada Gambar 4.1 di atas, merupakan tampilan pada fasilitas *serial monitor* yang ada pada *software IDE Arduino*. Setelah melakukan pengujian *enrollment* sidik jari, dilanjutkan pada pengujian kecocokan kecocokan sidik jari menggunakan *fingerprint scanner* seperti pada Gambar 4.2.

Pada pengujian kecocokan sidik jari dengan *fingerprint scanner* semakin tinggi nilai *confidence* pada *serial monitor* semakin tinggi pula tingkat kecocokan sidik jari tersebut. Tingkat kecocokan dapat ditentukan oleh posisi sidik jari, kondisi sidik jari dan lain sebagainya, tergantung tipe *fingerprint scanner* yang digunakan.

4.2 Pengujian Driver Solenoid Doorlock

Pada Gambar 4.3 merupakan rangkaian *driver solenoid doorlock* pada sistem yang kami rancang. *Solenoid doorlock* berperan sebagai pengunci pintu pada sistem keamanan yang kami rancang. Rangkaian *driver solenoid doorlock* menggunakan transistor Darlington sebagai penguat sinyal dari mikrokontroler. Hal tersebut dikarenakan *output* dari mikrokontroler kurang kuat untuk mengaktifkan *solenoid doorlock* tersebut. Sinyal *input* pada rangkaian transistor Darlington akan dikuatkan dua kali ukuran semula. Pengujian dilakukan dengan menggunakan tang amper untuk mengetahui besar arus yang mengalir pada *solenoid doorlock* dan hasilnya ditampilkan pada Tabel 4.1.



Gambar 4.3 Rangkaian Driver Solenoid Doorlock

Tabel 4.1 Tabel Pengukuran Arus *Solenoid Doorlock*

| NO | Tegangan Input | Trigger Basis Transistor | Arus <i>Solenoid Doorlock</i> |
|----|----------------|--------------------------|-------------------------------|
| 1 | 9 Volt | On | 0,4 A |
| 2 | 12 Volt | On | 0,45 A |
| 3 | 15 Volt | Off | 0 A |
| 4 | 15 Volt | On | 0,5 A |

4.3 Pengujian Sistem secara Keseluruhan



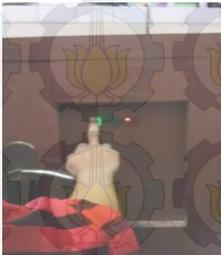
Gambar 4.4 Gambar Pintu saat Kondisi *Standby*

Pengujian perangkat keras sistem keamanan untuk membuka pintu dengan *fingerprint scanner*, pastikan alat mendapatkan catu daya melalui listrik. Sebelum melakukan pengujian pembukaan pintu, pastikan semua rangkaian dan koneksi telah terpasang dengan benar. Pengujian yang utama adalah membuka kunci pintu dan proses data *logger*.

Pada Gambar 4.4 lampu indikator merah menyala, artinya pintu masih terkunci. Diperlukan pola sidik jari yang cocok untuk membuka kunci pintu.



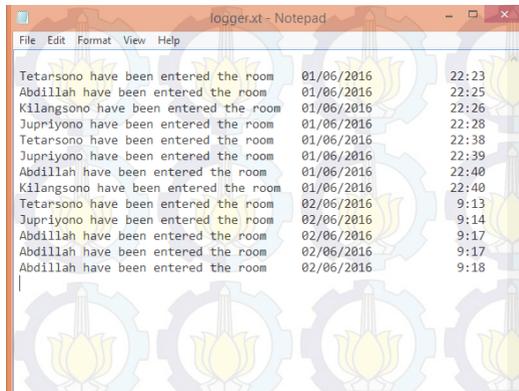
Gambar 4.5 Kondisi Lampu Indikator saat *Scanning* Sidik Jari yang Cocok



Gambar 4.6 Kondisi Lampu Indikator saat *Scanning* Sidik Jari yang Salah

Pengujian dilakukan pertama dengan menggunakan sidik jari yang telah terdaftar pada memori *fingerprint scanner*. Hasilnya dapat dilihat pada Gambar 4.5. Lampu indikator berubah warna biru yang berarti pintu sudah dapat dibuka.

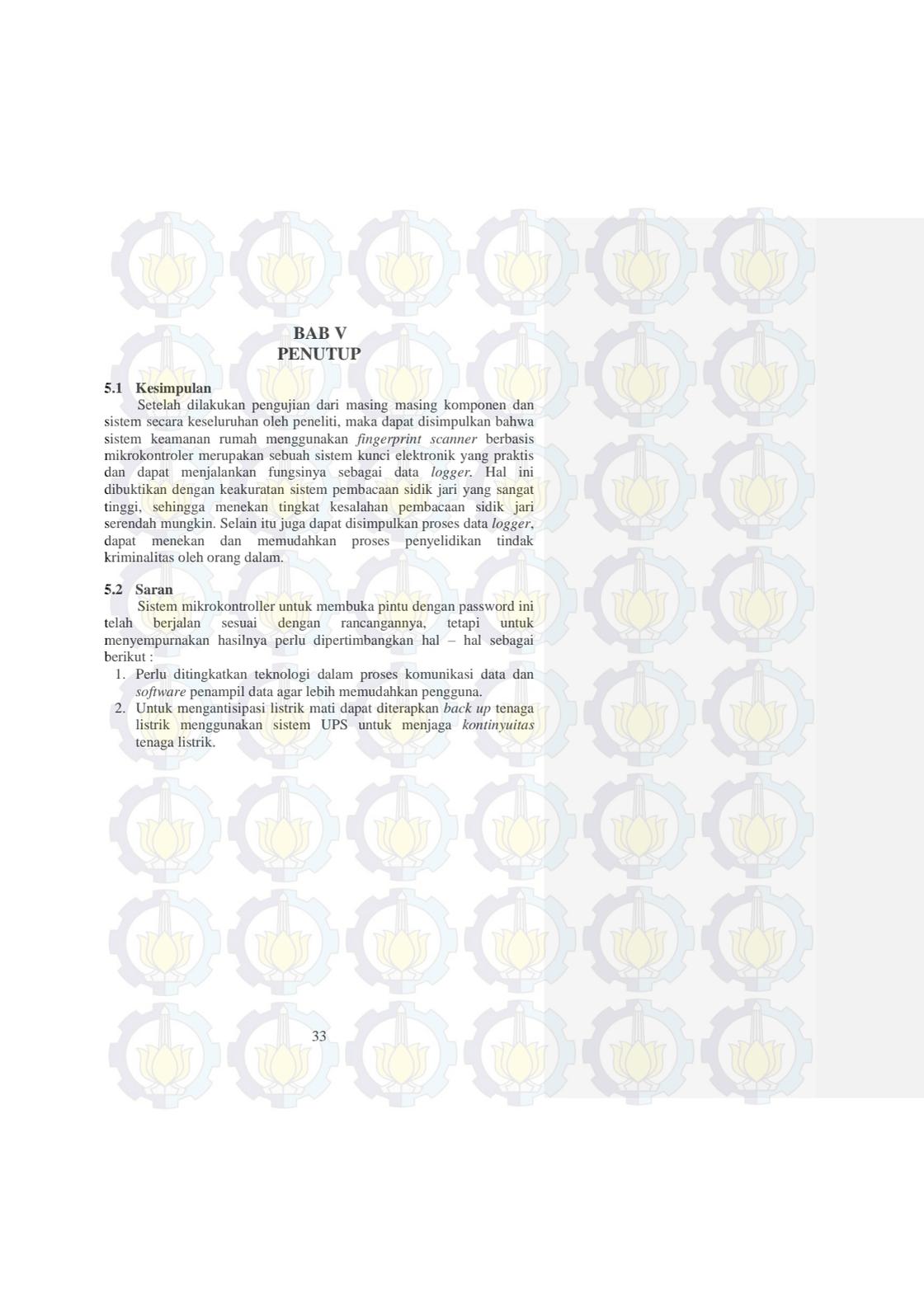
Pada Gambar 4.6 pola sidik jari yang dimasukkan tidak sesuai dengan sidik jari yang telah didaftarkan pada memori *fingerprint scanner*. Hasil yang didapatkan pada gambar tersebut, lampu indikator merah tetap menyala yang artinya kunci pintu belum *dinon-aktifkan*. Apabila kita memaksa membuka pintu maka akan mengaktifkan rangkaian *buzzer*.



Gambar 4.7 Pencatatan Data pada Notepad

Selain melakukan pengujian data terhadap tampilan pada lampu indikator, dilakukan pula pengujian data yang dicatat pada memori laptop. Hal ini dimaksudkan untuk mengetahui apakah sistem telah dapat melakukan tugasnya sebagai data *logger*.

Pada Gambar 4.7 di atas ditampilkan siapa saja yang telah memasuki ruangan dan jam berapa orang tersebut memasuki ruangan. Dengan pencatatan data seperti di atas diharapkan dapat mencegah terjadinya kejahatan yang dilakukan bahkan oleh orang dalam.



BAB V PENUTUP

5.1 Kesimpulan

Setelah dilakukan pengujian dari masing masing komponen dan sistem secara keseluruhan oleh peneliti, maka dapat disimpulkan bahwa sistem keamanan rumah menggunakan *fingerprint scanner* berbasis mikrokontroler merupakan sebuah sistem kunci elektronik yang praktis dan dapat menjalankan fungsinya sebagai data *logger*. Hal ini dibuktikan dengan keakuratan sistem pembacaan sidik jari yang sangat tinggi, sehingga menekan tingkat kesalahan pembacaan sidik jari serendah mungkin. Selain itu juga dapat disimpulkan proses data *logger*, dapat menekan dan memudahkan proses penyelidikan tindak kriminalitas oleh orang dalam.

5.2 Saran

Sistem mikrokontroler untuk membuka pintu dengan password ini telah berjalan sesuai dengan rancangannya, tetapi untuk menyempurnakan hasilnya perlu dipertimbangkan hal – hal sebagai berikut :

1. Perlu ditingkatkan teknologi dalam proses komunikasi data dan *software* penampil data agar lebih memudahkan pengguna.
2. Untuk mengantisipasi listrik mati dapat diterapkan *back up* tenaga listrik menggunakan sistem UPS untuk menjaga *kontinuitas* tenaga listrik.



DAFTAR PUSTAKA

- [1] BPS, "Statistik Kriminal 2014", *Badan Pusat Statistik*, Jakarta, Desember 2014.
- [2] Pudiatmoko, A., Fadlilah, U., dan Basith, A., "Sistem Keamanan Kamar Kos dengan Peringatan Alarm dan SMS Berbasis Mikrokontroler ATmega 32", *Jurnal Emitor*, Universitas Muhammadiyah Surakarta, Surakarta, 2007.
- [3] Gayung A., "Sistem Pengaman Rumah dengan *Security Password* Menggunakan Sensor Gerak Berbasis Mikrokontroler AT89S51", *Tugas Akhir*, Universitas Sumatera Utara, Medan, 2009.
- [4] Darmawiguna, I. G. M., Sunarya, I. M. G., "Aplikasi *Motion Detection* untuk *Home Security* Sistem dengan Pelaporan Otomatis Berbasis *SMS Gateway*", *Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMATI)*, Universitas Pendidikan Ganesha, Singaraja, 2013.
- [5] Dwirani, E. I., "Implementasi Sidik Jari untuk Sistem Absensi pada SDN II Cineam Tasikmalaya", *Skripsi*, Universitas Komputer Indonesia, Bandung, 2004.
- [6] Suroto, "Studi Penyempurnaan Identifikasi Sidik Jari pada *Algoritma Minutea*", *Skripsi*, Universitas Indonesia, Depok, 2009.
- [7] Prayoga R. D., "Perancangan *Home Security* Sistem Menggunakan Solenoid *Doorlock* Berbasis Arduino", *Tugas Akhir*, Universitas Mercu Buana, Bekasi, 1997.

LAMPIRAN

1. Program Enrollment Sidik Jari

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
uint8_t id;
uint8_t getFingerprintEnroll();
// Software serial for when you dont have a hardware serial port
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
// On Leonardo/Micro/Yun, use pins 8 & 9. On Mega, just grab a
hardware serialport
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
// On Leonardo/Micro or others with hardware serial, use those!
#0 is green wire, #1 is white
//Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);

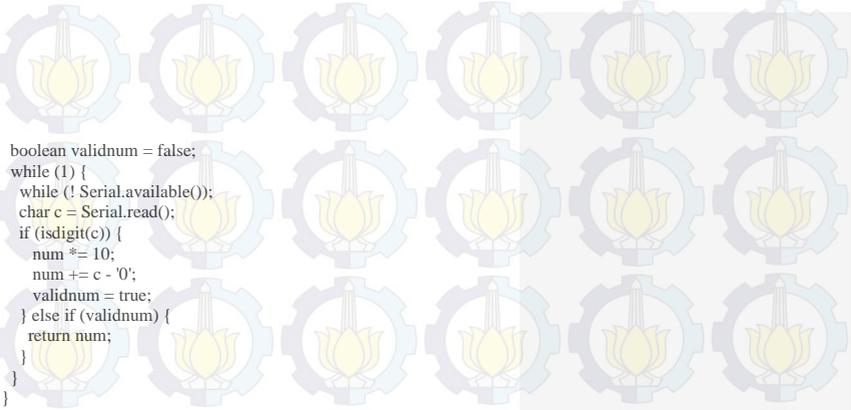
void setup()
{
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(500);

  Serial.begin(9600);
  Serial.println("Adafruit 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);
  }
}

uint8_t readnumber(void) {
  uint8_t num = 0;
```

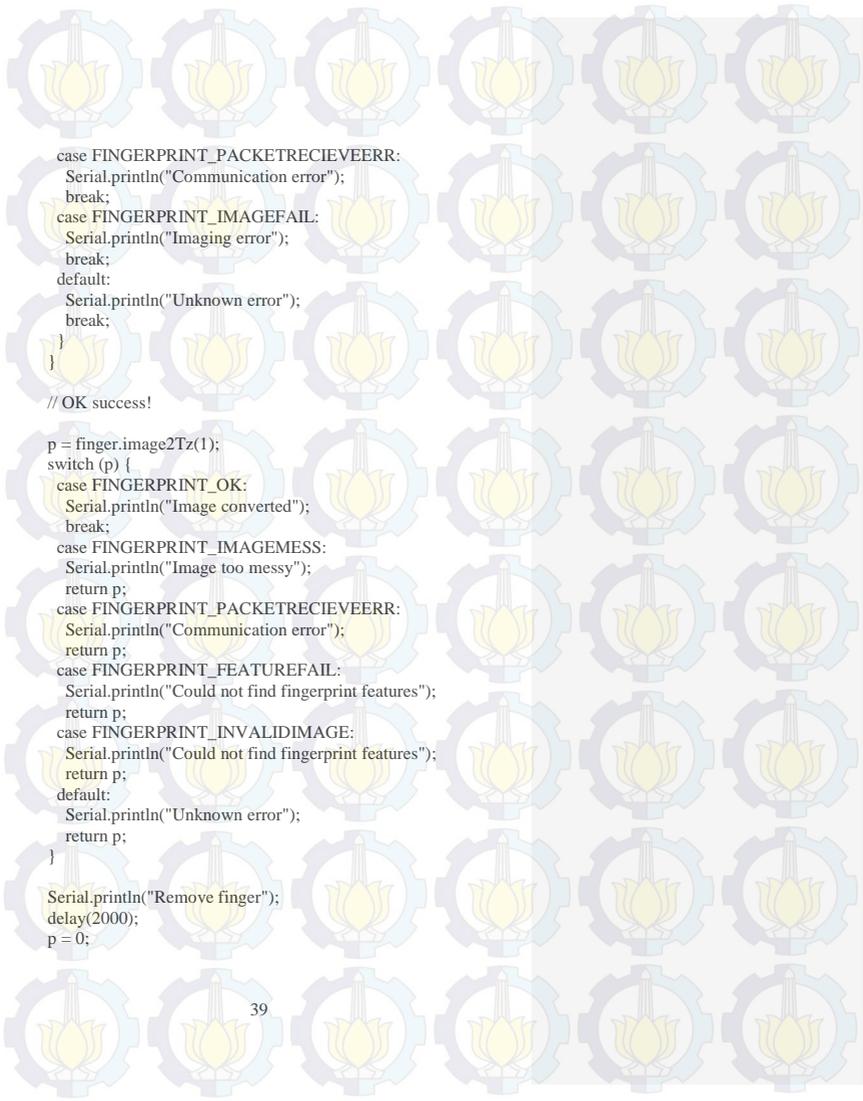


```
boolean validnum = false;
while (1) {
    while (! Serial.available());
    char c = Serial.read();
    if (isdigit(c)) {
        num *= 10;
        num += c - '0';
        validnum = true;
    } else if (validnum) {
        return num;
    }
}
```

```
void loop() // run over and over again
{
    Serial.println("Ready to enroll a fingerprint! Please Type in the
ID # you want to save this finger as...");
    id = readnumber();
    Serial.print("Enrolling ID #");
    Serial.println(id);

    while (! getFingerprintEnroll());
}
uint8_t getFingerprintEnroll() {
```

```
int p = -1;
Serial.print("Waiting for valid finger to enroll as #");
Serial.println(id);
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println(".");
            break;
```



```
case FINGERPRINT_PACKETRECIEVEERR:
  Serial.println("Communication error");
  break;
case FINGERPRINT_IMAGEFAIL:
  Serial.println("Imaging error");
  break;
default:
  Serial.println("Unknown error");
  break;
}
}
// OK success!

p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
  Serial.println("Image converted");
  break;
case FINGERPRINT_IMAGEMESS:
  Serial.println("Image too messy");
  return p;
case FINGERPRINT_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;
}
Serial.println("Remove finger");
delay(2000);
p = 0;
```



```
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
}

// OK success!
p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
```

```

return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
}

```

```

} else if (p == FINGERPRINT_FLASHERR) {
  Serial.println("Error writing to flash");
  return p;
} else {
  Serial.println("Unknown error");
  return p;
}
}

```

2. Program Scanning Sidik Jari

```

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

int getFingerprintIDez();

// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

// On Leonardo/Micro or others with hardware serial, use those! #0
// is green wire, #1 is white
//Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);

void setup()
{
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  Serial.begin(9600);
  Serial.println("Adafruit finger detect test");

  // 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);
}
Serial.println("Waiting for valid finger...");
}
void loop() // run over and over again
{
  getFingerprintIDez();
  delay(50); //don't ned to run this at full speed.
}
uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unkown error");
      return p;
  }
  // OK success!

  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:

```



```
Serial.println("Image too messy");
return p;
case FINGERPRINT_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!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_NOTFOUND) {
Serial.println("Did not find a match");
return p;
} else {
Serial.println("Unknown error");
return p;
}

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

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
```

```

uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;

p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;

p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;
// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of ");
Serial.println(finger.confidence);
return finger.fingerID;
}

```

3. Program Setting IP Ethernet Shield

```

#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192, 168, 100, 11);

//server menggunakan port 5 (client juga harus menggunakan port
yg sama)
EthernetServer ArduinoTCP = EthernetServer(5);
boolean alreadyConnected = false; // apakah sudah terkoneksi
sebelumnya
void setup()
{
  // inisialisasi ethernet device
  //Serial.begin(9600);
  Ethernet.begin(mac, ip);
  // listen untuk client yg datang
}

void loop()
{
}

```

4. Program Sistem secara Keseluruhan

```
#include <SPI.h>
#include <Ethernet.h>
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 100, 11);

//server menggunakan port 5 (client juga harus menggunakan port
yg sama)
EthernetServer ArduinoTCP = EthernetServer(5);
boolean alreadyConnected = false; // apakah sudah terkoneksi
sebelumnya
int getFingerprintIDez();
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  // inisialisasi ethernet device
  //Serial.begin(9600);
  Ethernet.begin(mac, ip);
  // listen untuk client yg datang
  while (!Serial); // For Yun/Leo/Micro/Zero/...

  Serial.begin(9600);
  Serial.println("Adafruit finger detect test");

  // 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);
  }
}
```

```
Serial.println("Waiting for valid finger...");  
}
```

```
void loop()
```

```
{  
  getFingerprintIDez();  
  digitalWrite(7,HIGH);  
  delay (500);  
  //don't ned to run this at full speed.  
}
```

```
uint8_t getFingerprintID() {  
  uint8_t p = finger.getImage();  
  switch (p) {  
    case FINGERPRINT_OK:  
      Serial.println("Image taken");  
      break;  
    case FINGERPRINT_NOFINGER:  
      Serial.println("No finger detected");  
      return p;  
    case FINGERPRINT_PACKETRECIEVEERR:  
      Serial.println("Communication error");  
      return p;  
    case FINGERPRINT_IMAGEFAIL:  
      Serial.println("Imaging error");  
      return p;  
    default:  
      Serial.println("Unkown error");  
      return p;  
  }
```

```
// OK success!
```

```
p = finger.image2Tz();  
switch (p) {  
  case FINGERPRINT_OK:  
    Serial.println("Image converted");  
    break;  
  case FINGERPRINT_IMAGEMESS:
```

```

Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_NOTFOUND) {
Serial.println("Did not find a match");
return p;
} else {
Serial.println("Unknown error");
return p;
}

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

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {

```



```
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;

p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;

p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;

// found a match!
digitalWrite(7,LOW);
digitalWrite(6,HIGH);
digitalWrite(5,HIGH);
digitalWrite(8,HIGH);
delay(7000);
digitalWrite(6,LOW);
digitalWrite(5,LOW);
digitalWrite(8,LOW);

delay(500);
digitalWrite(7,HIGH);

delay(1000);
}
```

RIWAYAT PENULIS



NAMA : MOHAMMAD SAIFUL A

TTL : BLITAR, 10 NOVEMBER 1994

ALAMAT : JL. GEBANG LOR NO 12 SBY

TELEPON : 081231453120

HOBI : BERMAIN SEPAK BOLA

MOTTO : SELALU BERSYUKUR

RIWAYAT PENDIDIKAN :

MI MWB BAJANG

SMPN 1 WLINGI

SMAN 1 TALUN

D3 TEKNIK ELEKTRO INDUSTRI FTI-ITS



RIWAYAT PENULIS

NAMA : AGAM ABDILLAH
TTL : SURABAYA, 30 JULI 1995
ALAMAT : JL. JENDRAL KATAMSO
TELEPON : 085230253522
HOBI : BOLA BASKET
MOTTO : BERSEDEKAH

RIWAYAT PENDIDIKAN :

SD MUHAMMADIYAH 1 WARU
SMP MUHAMMADIYAH 3 WARU
SMAN 1 GEDANGAN
D3 TEKNIK ELEKTRO INDUSTRI FTI-ITS