

**PERANCANGAN KUNCI PINTU OTOMATIS PADA SMART HOME  
MENGUNAKAN *FINGERPRINT* SCANNER GT-511C1 DENGAN  
VERIFIKASI PENGGUNA SECARA TERPUSAT MELALUI  
*WIRELESS SENSOR NETWORK***

**SKRIPSI**

**KEMINATAN TEKNIK KOMPUTER**

Diajukan untuk memenuhi sebagian persyaratan mencapai gelar  
Sarjana Komputer



Disusun oleh :

**ACHMAD PRAJA SASMITA**

**NIM. 115060901111008**

**UNIVERSITAS BRAWIJAYA**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**MALANG**

**2015**

## LEMBAR PERSETUJUAN

**Perancangan Kunci Pintu Otomatis Pada *Smart Home*  
Menggunakan *Fingerprint Scanner* GT-511C1 dengan Verifikasi  
Pengguna secara Terpusat melalui *Wireless Sensor Network***

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

**ACHMAD PRAJA SASMITA**

**NIM. 115060901111008**

Skripsi ini telah disetujui oleh dosen pembimbing

Dosen Pembimbing I

Dosen Pembimbing II

**Agung Setia Budi, S.T., M.T., M.Eng.**

**NIK. 2013048704231001**

**Hurriyatul Fitriyah, S.T., M.Sc.**

**NIK. 2013048510012001**

## LEMBAR PENGESAHAN

**PERANCANGAN KUNCI PINTU OTOMATIS PADA *SMART HOME*  
MENGUNAKAN *FINGERPRINT SCANNER* GT-511C1 DENGAN  
VERIFIKASI PENGGUNA SECARA TERPUSAT MELALUI  
*WIRELESS SENSOR NETWORK***

**SKRIPSI**

**KEMINATAN TEKNIK KOMPUTER**

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

**Achmad Praja Sasmita**

**NIM. 115060901111008**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 18 Juni 2015

Penguji I

Penguji II

**Fitri Utaminingrum, Dr. Eng., S.T, M.T**

**NIP. 19820710 200812 2 001**

**Rekyan Regasari Mardi P, S.T, M.T**

**NIK. 770414 06 1 2 0253**

Penguji III

**Barlian Henryranu Prasetyo, S.T., M.T.**

**NIK. 821024 06 1 1 0254**

Mengetahui

Ketua Program Studi Teknik Informatika

**Drs. Marji, M.T.**

**NIP. 19670801 199203 1 001**



## PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, Juni 2015

Mahasiswa,

**Achmad Praja Sasmita**  
**NIM. 115060901111008**

## KATA PENGANTAR

Puji syukur penulis panjatkan kahadirat Tuhan Yang Maha Esa, karena Rahmat-Nya lah penulis dapat menyelesaikan skripsi ini. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam menempuh ujian Sarjana Program Teknologi Informasi dan Ilmu Komputer. Judul skripsi yang disusun adalah: **“Perancangan Kunci Pintu Otomatis Pada Smart Home Menggunakan Fingerprint Scanner GT-511C1 dengan Verifikasi Pengguna secara Terpusat melalui Wireless Sensor Network”**

Banyak kesulitan dan hambatan yang dialami oleh penulis dalam menyusun skripsi ini terutama dalam mendapatkan data dan mengolahnnya, tetapi semua itu telah dapat diatasi dengan baik berkat dukungan dan bantuan dari berbagi pihak. Untuk itulah pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak dan Ibu yang saya cintai, yang telah memberikan dukungan, semangat serta do'a.
2. Bapak Agung Setia Budi, S.T., M.Eng. selaku dosen pembimbing I yang telah memberikan pengarahan dalam pembuatan proposal skripsi ini.
3. Ibu Hurriyatul Fitriyah, S.T., M.Sc. selaku dosen pembimbing II yang telah memberikan saran – saran yang sangat berguna dalam penyelesaian proposal skripsi ini.
4. Bapak Drs. Marji, M.T. selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer.
5. Bapak Adharul Muttaqin ST., MT selaku dosen penasihat akademik penulis.
6. Bapak Barlian Henryranu, S.T., M.T. selaku Kepala Laboratorium Sistem Komputer dan Robotika Program Teknologi Informasi dan Ilmu Komputer.
7. Aninditya Nugroho, S.T (Mas Didit) selaku laboran Kepala Laboratorium Sistem Komputer dan Robotika Fakultas Ilmu Komputer yang selalu membantu & menyediakan alat-alat skripsi untuk penulis.
8. Segenap dosen Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan Segenap staff dan pegawai Fakultas Ilmu Komputer Universitas Brawijaya atas segala bantuan yang

bersifat administratif.

9. Teman-teman satu bimbingan Andri, Ardi, Bagus, Putra, Dewi, Virgi dan Yuni yang telah memberikan bantuan baik ilmu, moral, dan material dalam proses pengerjaan skripsi dan berjuang bersama sepanjang hari.
10. Seluruh teman-teman angkatan 2011 dan pihak yang telah membantu yang tidak dapat saya sebutkan satu per satu.
11. Pak lek Khamim sekeluarga, Bu lek Umi sekeluarga dan Bu lek Anis yang telah memberikan bimbingan, semangat dan dukungan kepada penulis.

Penulis menyadari bahwa proposal skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga proposal skripsi ini dapat berguna bagi yang membutuhkannya.

Malang, Juni 2015

Penulis



## ABSTRAK

**Achmad Praja Sasmita. 2015. *Perancangan Kunci Pintu Otomatis Pada Smart Home Menggunakan Fingerprint Scanner GT-511C1 dengan Verifikasi Pengguna secara Terpusat melalui Wireless Sensor Network*.** Skripsi Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Pembimbing: Agung Setia Budi, S.T., M.Eng. dan Hurriyatul Fitriyah, S.T., M.Sc.

Keamanan pintu rumah yang menggunakan metode konvensional berupa kunci masih memiliki banyak kekurangan. Seperti contoh dengan sangat mudah pengguna lain menggandakan kunci keamanan tersebut. Oleh karena itu dalam penelitian ini, dibuat solusi kunci pintu dengan menggunakan verifikasi sidik jari secara terpusat melalui Wireless Sensor Network (WSN). Perancangan dari sistem ini adalah melakukan verifikasi pengguna sebanyak 2 kali yaitu pada modul *Fingerprint Scanner* dan server. Selain melakukan verifikasi pengguna, server juga sekaligus melakukan pengambilan data record berupa user, ID, dan jam akses setiap pengguna membuka pintu. Diharapkan dengan adanya verifikasi pada server dan pengambilan record akses tersebut akan dapat menjadi solusi terhadap masalah keamanan rumah khususnya pintu. Kesimpulan dari perancangan sistem ini adalah semua perancangan algoritma (enroll user, verifikasi user dan hapus user) dapat berjalan dengan baik. Waktu yang dibutuhkan untuk proses pada masing algoritma kurang dari 2 detik.

**Kata Kunci:** Smart Home, Kunci Pintu Otomatis, Sidik Jari, Server Terpusat, Wireless Sensor Network.

## ABSTRACT

**Achmad Praja Sasmita. 2015. *Design Of Automatic Door Lock On Smart Home Using Fingerprint Scanner GT-511C1 With Centrally User Verification Through Wireless Sensor Network*. Skripsi Faculty of Computer Science University of Brawijaya Malang.**

**Advisor:** Agung Setia Budi, S.T., M.Eng. and Hurriyatul Fitriyah, S.T., M.Sc.

*The home door's security that using conventional methods such as keys and locks have some of drawbacks. For example, the other persons can easily duplicate the keys. Therefore, on this research we make the lock doors with fingerprint verification centrally through Wireless Sensor Network (WSN). The design of this system is to verify the users twice, using the Fingerprint Scanner's module and Server's data. In addition to verifying the people, the server also simultaneously record the data collection from the user, user's ID and the time of access each user when opens the door. Hopefully, by the verification on the server and retrieval access record can be a solution to security problems, especially house doors security. The conclusion of this system design is all the design algorithm (enroll user, user verification, and delete user) can work well. The time required to process at each algorithm is less than 2 seconds.*

**Keywords :** Smart Home, automatic locks door , fingerprint, centralized server, Wireless Sensor Network.

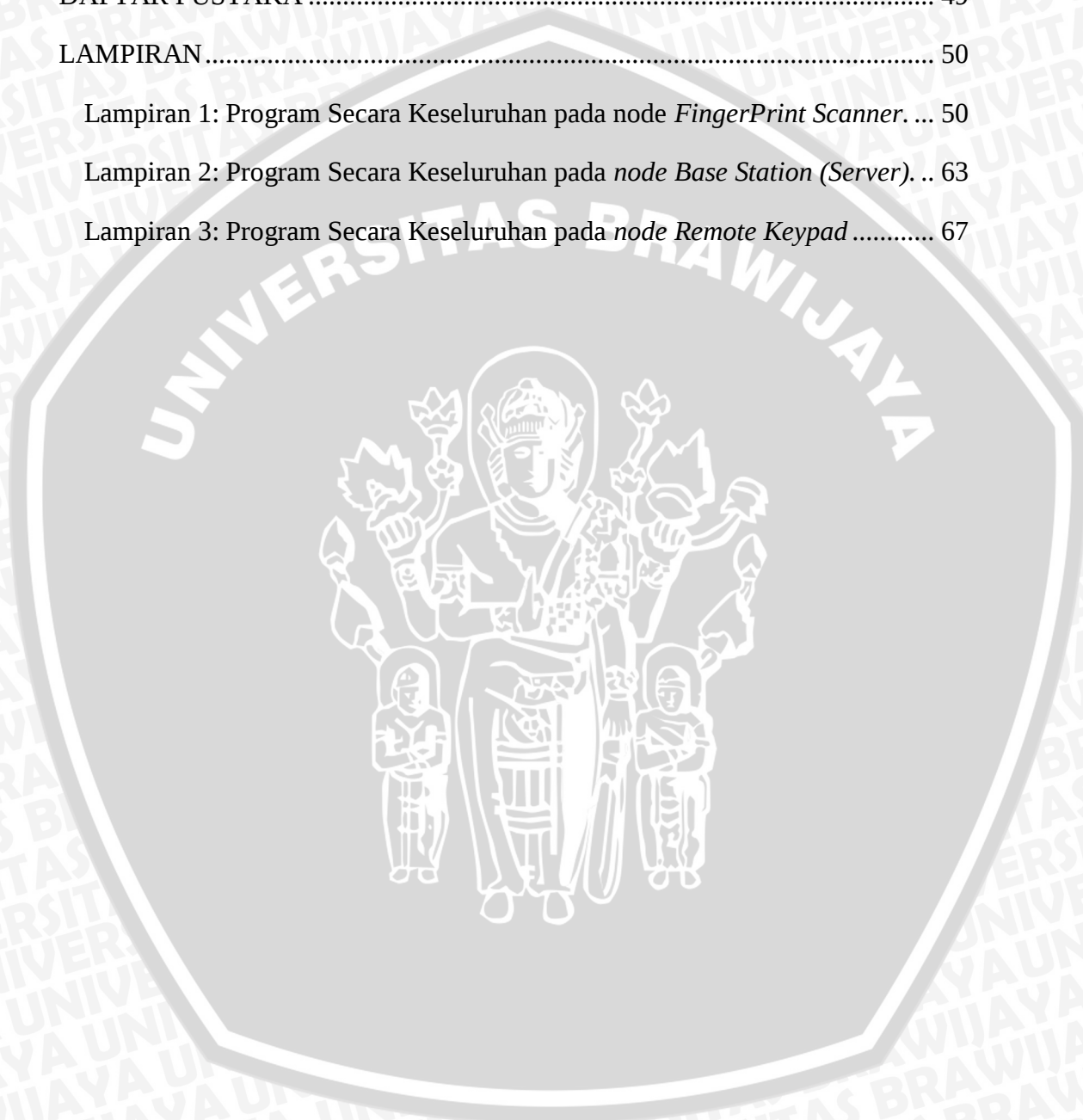
## DAFTAR ISI

|                                               |     |
|-----------------------------------------------|-----|
| KATA PENGANTAR .....                          | i   |
| ABSTRAK .....                                 | iii |
| ABSTRACT .....                                | iv  |
| DAFTAR ISI .....                              | v   |
| DAFTAR GAMBAR .....                           | ix  |
| DAFTAR TABEL .....                            | xi  |
| BAB I .....                                   | 1   |
| PENDAHULUAN .....                             | 1   |
| 1.1. Latar Belakang .....                     | 1   |
| 1.2. Rumusan masalah .....                    | 2   |
| 1.3. Batasan Masalah .....                    | 2   |
| 1.4. Tujuan .....                             | 2   |
| 1.5. Manfaat .....                            | 3   |
| 1.6. Sistematika Penulisan .....              | 3   |
| BAB II .....                                  | 5   |
| KAJIAN PUSTAKA DAN DASAR TEORI .....          | 5   |
| 2.1 Kajian Pustaka .....                      | 5   |
| 2.2 Dasar Teori .....                         | 6   |
| 2.2.1 Smart Home .....                        | 6   |
| 2.2.2 Arduino Pro Mini .....                  | 7   |
| 2.2.3 Liquid Crystal Display (LCD 2x16) ..... | 8   |
| 2.2.3.1 Konfigurasi LCD 2x16 .....            | 8   |
| 2.2.3.2 Fungsi Register LCD 2X16 .....        | 9   |

|                                    |                                                 |    |
|------------------------------------|-------------------------------------------------|----|
| 2.2.4                              | NRF24L01.....                                   | 10 |
| 2.2.5                              | <i>Fingerprint Scanner GT-511C1</i> .....       | 12 |
| 2.2.6                              | <i>4x4 Matrix Membrane Keypad</i> .....         | 12 |
| 2.2.7                              | <i>Wireless Sensor Network (WSN)</i> .....      | 14 |
| 2.2.8                              | MySQL.....                                      | 16 |
| BAB III.....                       |                                                 | 18 |
| METODOLOGI PENELITIAN.....         |                                                 | 18 |
| 3.1.                               | Studi Literatur.....                            | 18 |
| 3.2.                               | Analisis Kebutuhan .....                        | 19 |
| 3.2.1.                             | Kebutuhan Perangkat Keras .....                 | 19 |
| 3.2.2.                             | Kebutuhan Perangkat Lunak .....                 | 19 |
| 3.3.                               | Penyusunan Dasar Teori.....                     | 20 |
| 3.4.                               | Perancangan Sistem.....                         | 20 |
| 3.5.                               | Implementasi .....                              | 20 |
| 3.6.                               | Pengujian .....                                 | 21 |
| 3.7.                               | Analisis .....                                  | 21 |
| 3.8.                               | Kesimpulan.....                                 | 21 |
| BAB IV .....                       |                                                 | 22 |
| PERANCANGAN DAN IMPLEMENTASI ..... |                                                 | 22 |
| 4.1.                               | Perancangan.....                                | 22 |
| 4.1.1.                             | Analisis Kebutuhan Sistem .....                 | 22 |
| 4.1.2.                             | Diagram Blok Sistem .....                       | 22 |
| 4.1.3.                             | Perancangan Perangkat Keras .....               | 23 |
| 4.1.3.1.                           | Perancangan Antarmuka Arduino dan NRF24L01..... | 23 |
| 4.1.3.2.                           | Perancangan Board PCB.....                      | 24 |

|                             |                                                                              |    |
|-----------------------------|------------------------------------------------------------------------------|----|
| 4.1.4.                      | Perancangan Perangkat Lunak .....                                            | 24 |
| 4.1.4.1.                    | Perancangan Algoritma <i>Enroll</i> Data ke <i>Fingerprint Scanner</i> .     | 25 |
| 4.1.4.2.                    | Perancangan Algoritma Mendapatkan ID Dari <i>Fingerprint Scanner</i> .....   | 25 |
| 4.1.4.3.                    | Perancangan Algoritma Menghapus ID Dari <i>Fingerprint Scanner</i> .....     | 25 |
| 4.1.4.4.                    | Perancangan Paket Data.....                                                  | 29 |
| 4.1.4.5.                    | Perancangan Algoritma <i>Enroll</i> Data ke <i>Base Station (Server)</i>     | 31 |
| 4.1.4.6.                    | Perancangan Algoritma Verifikasi Data ke <i>Base Station (Server)</i> .....  | 33 |
| 4.1.4.7.                    | Perancangan Algoritma Menghapus Data di <i>Base Station (Server)</i> .....   | 34 |
| 4.2.                        | Implementasi .....                                                           | 35 |
| 4.2.1.                      | Implementasi Perangkat Keras.....                                            | 35 |
| 4.2.2.                      | Implementasi Perangkat Lunak.....                                            | 36 |
| BAB V.....                  |                                                                              | 38 |
| PENGUJIAN DAN ANALISIS..... |                                                                              | 38 |
| 5.1                         | Pengujian Algoritma <i>Enroll</i> Data ke <i>Fingerprint Scanner</i> .....   | 38 |
| 5.2                         | Pengujian Algoritma Verifikasi ID Pada <i>Fingerprint Scanner</i> .....      | 39 |
| 5.3                         | Pengujian Algoritma Menghapus ID Dari <i>Fingerprint Scanner</i> .....       | 40 |
| 5.4                         | Pengujian Komunikasi Data Dengan NRF24L01 .....                              | 41 |
| 5.5                         | Pengujian Algoritma <i>Enroll</i> Data ke <i>Base Station (Server)</i> ..... | 43 |
| 5.6                         | Pengujian Algoritma Verifikasi Data ke <i>Base Station (Server)</i> .....    | 44 |
| 5.7                         | Pengujian Algoritma Menghapus Data di <i>Base Station (Server)</i> .....     | 46 |
| BAB VI .....                |                                                                              | 48 |
| PENUTUP .....               |                                                                              | 48 |

|                                                                                     |                 |    |
|-------------------------------------------------------------------------------------|-----------------|----|
| 6.1                                                                                 | Kesimpulan..... | 48 |
| 6.2                                                                                 | Saran.....      | 48 |
| DAFTAR PUSTAKA .....                                                                |                 | 49 |
| LAMPIRAN.....                                                                       |                 | 50 |
| Lampiran 1: Program Secara Keseluruhan pada node <i>FingerPrint Scanner</i> . ...   |                 | 50 |
| Lampiran 2: Program Secara Keseluruhan pada node <i>Base Station (Server)</i> . ... |                 | 63 |
| Lampiran 3: Program Secara Keseluruhan pada node <i>Remote Keypad</i> .....         |                 | 67 |



## DAFTAR GAMBAR

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| Gambar 2.1 Board Arduino Pro Mini .....                                                    | 7  |
| Gambar 2.2 Modul LCD yang digunakan.....                                                   | 8  |
| Gambar 2.3 Display Data RAM (DDRAM) .....                                                  | 10 |
| Gambar 2.4 NRF24L01.....                                                                   | 11 |
| Gambar 2.5 NRF24L01.....                                                                   | 12 |
| Gambar 2.6 4x4 matrix membrane <i>Keypad</i> .....                                         | 13 |
| Gambar 2.7 koneksi 4x4 matrix membrane <i>Keypad</i> .....                                 | 14 |
| Gambar 2.8 WSN Kategori 1: multipoint-to-point,sistem multihop dengan routing dinamis..... | 16 |
| Gambar 2.9 WSN Kategori2: point-to-point,sistem singlehop dengan routing statis.....       | 16 |
| Gambar 3.2 Diagram Blok Analisis Kebutuhan Sistem.....                                     | 19 |
| Gambar 4.1 Diagram Blok Sistem .....                                                       | 23 |
| Gambar 4.2 Rangkaian Arduino ke NRF24L01 .....                                             | 23 |
| Gambar 4.3 Schematic Board PCB.....                                                        | 24 |
| Gambar 4.4 Diagram Alir Enroll Data ke <i>Fingerprint Scanner</i> .....                    | 26 |
| Gambar 4.5 Diagram Alir Mendapatkan ID dari <i>Fingerprint Scanner</i> .....               | 27 |
| Gambar 4.5 Diagram Alir Menghapus ID dari <i>Fingerprint Scanner</i> .....                 | 28 |
| Gambar 4.6 Diagram Alir Enroll Data ke <i>Base Station (Server)</i> .....                  | 32 |
| Gambar 4.7 Diagram Alir Verifikasi Data ke <i>Base Station (Server)</i> .....              | 33 |
| Gambar 4.8 Diagram Alir Hapus Data di <i>Base Station (Server)</i> .....                   | 34 |
| Gambar 4.8 Board Modul .....                                                               | 35 |
| Gambar 4.9 Modul Wireless .....                                                            | 36 |
| Gambar 4.10 Diagram Alir pada <i>Base Station (Server)</i> .....                           | 37 |

|                                         |    |
|-----------------------------------------|----|
| Gambar 5.2 log door Database.....       | 45 |
| Gambar 5.3 User Database Hapus .....    | 47 |
| Gambar 5.4 Diagram alir komunikasi..... | 42 |



## DAFTAR TABEL

|                                                                                                    |    |
|----------------------------------------------------------------------------------------------------|----|
| Table 2.1 Spesifikasi Arduino Pro Mini .....                                                       | 7  |
| Table 2.2 Spesifikasi Arduino Pro Mini .....                                                       | 9  |
| Table 2.3 RAM (DDRAM) dan character generator (CGRAM) .....                                        | 10 |
| Tabel 4.1 Rancangan Paket Data dari Arduino ke Komputer .....                                      | 29 |
| Tabel 4.2 Keterangan Paket Data dari Arduino ke Komputer .....                                     | 29 |
| Tabel 4.3 Rancangan Paket Data dari Komputer ke Arduino .....                                      | 30 |
| Tabel 4.4 Keterangan Paket Data .....                                                              | 30 |
| Tabel 4.5 Rancangan Paket Data .....                                                               | 30 |
| Tabel 4.6 Keterangan Paket Data .....                                                              | 31 |
| Tabel 4.7 Komponen Pada Board .....                                                                | 35 |
| Tabel 5.1 Pengujian Algoritma Enroll (Penambahan User) Pada Modul <i>Fingerprint Scanner</i> ..... | 38 |
| Tabel 5.2 Pengujian Algoritma Verifikasi Pada Modul <i>Fingerprint Scanner</i> .....               | 39 |
| Tabel 5.3 Pengujian Algoritma Menghapus ID dari Modul <i>Fingerprint Scanner</i> .....             | 40 |
| Tabel 5.4 Pengujian Algoritma Enroll Pada <i>Base Station (Server)</i> .....                       | 43 |
| Tabel 5.5 Pengujian Algoritma Verifikasi Data ke <i>Base Station (Server)</i> .....                | 44 |
| Tabel 5.6 Pengujian Algoritma Menghapus Data di <i>Base Station (Server)</i> .....                 | 46 |

## BAB I

### PENDAHULUAN

Bab ini berisi pembahasan tentang hal yang menjadi latar belakang topik, permasalahan yang akan dibahas, batasan dari analisis, tujuan dan manfaat topik yang akan dibahas, serta sistematika dalam penulisan skripsi.

#### 1.1. Latar Belakang

Manusia didorong untuk berfikir kreatif guna menciptakan suatu terobosan baru dalam kehidupan sehari-hari. Salah satu masalah yang disorot adalah tentang keamanan. Keamanan dapat mencakup tentang keamanan lingkungan rumah, keamanan berkendara dan lain sebagainya. Untuk sistem keamanan lingkungan rumah saat ini (konvensional) dirasa masih kurang dalam menjaga sekaligus memberi rasa nyaman kepada pemilik rumah. Hal ini karena terkadang pemilik rumah lupa untuk mengunci pintu pada saat akan meninggalkan rumah. Beberapa kasus pembobolan sistem keamanan juga terjadi dimana pelaku dapat masuk ke dalam rumah tanpa merusak bagian yang dapat menimbulkan kecurigaan lingkungan sekitar.

Teknologi untuk mengatasi masalah diatas adalah sistem keamanan pada kunci pintu otomatis. Teknologi keamanan kunci pintu otomatis pun beragam, mulai dari penambahan alarm, password angka dan lain sebagainya. Namun teknologi dari sistem diatas dianggap masih kurang dalam menjaga keamanan dan kenyamanan pemilik rumah, karena ditemukan kasus dimana pelaku pencurian masih mempunyai hubungan keluarga dengan pemilik rumah sehingga pelaku mengetahui kunci akses (password angka) untuk masuk ke dalam rumah. Solusinya, harus diterapkan suatu teknologi yang lebih spesifik dan tidak mudah diduplikasi guna memberikan solusi keamanan yang lebih akurat.

Satu aplikasi yang dapat diterapkan adalah sistem keamanan kunci pintu otomatis (*door lock*) dengan biometrika sidik jari. Pada sistem ini hanya pengguna terdaftar yang dapat membuka kunci pintu dengan identifikasi dan verifikasi biometrika sidik jari sehingga lebih aman. Data dari pengguna terdaftar disimpan dalam server terpusat, dimana server dapat digunakan untuk mengakomodir semua

sistem pada *smart home*. Komunikasi yang digunakan adalah dengan *wireless sensor network*.

### 1.2. Rumusan masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahan pada tugas akhir ini yaitu sebagai berikut:

1. Bagaimana membuat desain dan implementasi kunci pintu otomatis dengan identifikasi pola sidik jari ?
2. Berapa waktu yang dibutuhkan untuk proses verifikasi pada server *smart home* melalui *wireless sensor network* ?

### 1.3. Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian tugas akhir ini dibatasi dalam hal:

1. Menggunakan Arduino sebagai Mikrokontroller
2. Menggunakan NRF24L01 sebagai media transfer data
3. Untuk verifikasi pada modul sidik jari menggunakan algoritma yang sudah tertanam pada modul *Fingerprint Scanner GT-511C1*
4. Kondisi jari dari user yang akan melakukan scanning harus dalam keadaan bersih dan kering, karena jika tidak dapat menyebabkan kegagalan dalam proses scanning pola sidik jari.
5. Diasumsikan setelah user masuk ke dalam ruangan, pintu akan ditutup kembali.

### 1.4. Tujuan

Berdasarkan rumusan masalah di atas, maka dapat dirumuskan tujuan pada tugas akhir ini yaitu sebagai berikut:

1. Membuat kunci pintu otomatis dengan identifikasi pola sidik jari.
2. Membuat proses verifikasi pengguna pada server *smart home* melalui *wireless sensor network*.

### 1.5. Manfaat

Penelitian ini diharapkan dapat bermanfaat untuk berbagai pihak. Manfaat dari penelitian ini adalah sebagai berikut:

1) Manfaat Bagi Penulis

Sistem ini diharapkan dapat memaksimalkan keamanan dan kenyamanan terhadap pemilik rumah yang mempunyai teknologi *smart home*.

2) Manfaat Bagi Pembaca

Mengetahui cara membuat sistem kunci pintu otomatis dengan identifikasi pola sidik jari

### 1.6. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

#### **BAB I Pendahuluan**

Memuat latar belakang permasalahan, identifikasi dan pembatasan masalah, rumusan masalah, tujuan dan manfaat, dan sistematika penulisan.

#### **BAB II Kajian Pustaka dan Dasar teori**

Menjelaskan teori dan referensi penelitian yang terkait dengan “Perancangan Kunci Pintu Otomatis Pada *Smart Home* Menggunakan *Fingerprint Scanner* GT-511C1 dengan Verifikasi Pengguna secara Terpusat melalui *Wireless Sensor Network*”

#### **BAB III Metodologi Penelitian**

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, analisis kebutuhan, penyusunan dasar teori, perancangan sistem, implementasi, pengujian, analisis serta pengambilan kesimpulan.

#### **BAB IV Perancangan**

Membahas perancangan dan implementasi untuk pengujian “Perancangan Kunci Pintu Otomatis Pada *Smart Home* Menggunakan

*Fingerprint Scanner GT-511C1 dengan Verifikasi Pengguna secara Terpusat melalui Wireless Sensor Network”*

## **BAB V Pengujian dan Analisis**

Memuat proses dan hasil pengujian terhadap metode perancangan yang telah direalisasikan.

## **BAB VI Penutup**

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian untuk pengembangan lebih lanjut.



## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan teori dan referensi penelitian yang terkait dan memiliki tujuan dan perancangan yang hampir sama dengan penelitian ini.

#### 2.1 Kajian Pustaka

Penelitian lain yang menjadi landasan dan referensi dari sistem yang akan dibuat adalah “Perancangan dan Implementasi Kunci Pintu Otomatis dengan Sidik Jari Berbasis Mikrokontroller” dilakukan oleh Unang Sunarya ST., MT., Sugondo Hadiyoso, ST., MT, Anggidhira Widyananda . Pada penelitian tersebut pengguna harus melakukan verifikasi sidik jari untuk dapat membuka kunci pintu, verifikasi hanya dilakukan pada modul *Fingerprint Scanner*.

Penelitian lain yang juga menjadi landasan dan referensi adalah “Pembuatan Sistem Monitoring Pengguna Laboratorium Komputer Berbasis Sidik Jari” dilakukan oleh I Kadek Dwi Sucipta. Pada penelitian tersebut individu yang akan menggunakan komputer pada laboratorium harus melakukan verifikasi (log in) terlebih dahulu menggunakan sidik jari. Setelah verifikasi (log in) berhasil baru user tersebut dapat menggunakan komputer. Penggunaan komputer tersebut dapat dimonitoring dari komputer pusat (*Server*). Data penggunaan komputer yang meliputi jam pemakaian, jam selesai, tanggal dan identitas dari pengguna tersebut akan disimpan pada server dan dapat di cetak.

Adapun penelitian lain yang juga menjadi landasan dan referensi dari sistem yang akan dibuat adalah “Aplikasi Pengenalan Individu Berbasis Sidik Jari pada Pengembangan Portal Otomatis” dilakukan oleh Catharina Hilda Asritirtany dan Nurul Hidayat . Penelitian diatas membuat sistem portal pada area parkir yang menggunakan verifikasi sidik jari, terdapat dua parameter lain yang digunakan yaitu foto dari pengguna dan nomor kendaraan. Pada saat masuk area parkir pengguna harus melakukan pendaftaran dengan menggunakan sidik jari. Data sidik jari pengguna tersebut akan disimpan pada databases berserta dengan dua parameter lain yaitu foto pengguna dan nomor kendaraan. Pada saat akan keluar pengguna harus melakukan verifikasi dengan sidik jari, data tersebut akan dicocokkan dengan

sidik jari yang diperoleh pada saat masuk. Apabila pada waktu keluar tidak terdeteksi kesamaan sidik jari pada saat masuk maka sistem akan mendeteksi adanya kejanggalan.

Dari beberapa penelitian diatas maka penulis akan melakukan penelitian yang berjudul “Perancangan Kunci Pintu Otomatis Pada *Smart Home* Menggunakan *Fingerprint Scanner* GT-511C1 dengan Verifikasi Pengguna secara Terpusat melalui *Wireless Sensor Network*”. Pada sistem ini verifikasi dilakukan dua kali, yaitu pada modul *Fingerprint Scanner* dan *Base Station (Server)*. Data akses dari kunci pintu tersebut akan disimpan pada database, yang meliputi tanggal dan waktu akses pintu serta ID pengguna. Diharapkan dengan adanya data tersebut dapat membantu monitoring pada ruangan yang menggunakan aplikasi kunci pintu otomatis.

## 2.2 Dasar Teori

Berdasarkan beberapa informasi dari beberapa kajian pustaka, maka dalam penulisan penelitian yang berjudul “Perancangan Kunci Pintu Otomatis Pada *Smart Home* Menggunakan *Fingerprint Scanner* GT-511C1 dengan Verifikasi Pengguna secara Terpusat melalui *Wireless Sensor Network*” terdapat beberapa dasar teori, antara lain:

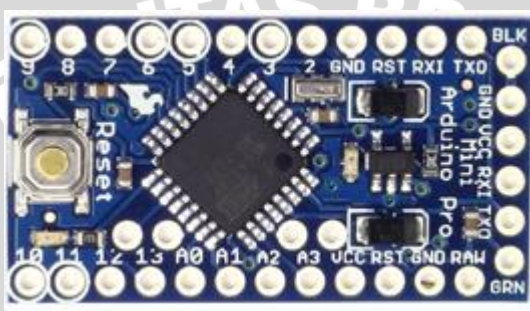
### 2.2.1 *Smart Home*

*Smart Home* (Rumah Pintar) adalah sebuah teknologi terbaru yang mengimplementasikan sebuah sistem canggih disebuah rumah atau bangunan yang dapat berjalan otomatis dalam mengendalikan perangkat dan perlengkapan yang ada dirumah tersebut. Teknologi *Smart Home* diharapkan mampu terintegrasi dengan *smart home* lain serta menciptakan kemudahan, kenyamanan, keamanan dan efisiensi bagi penggunanya.

Teknologi *Smart Home* merupakan gabungan dari beberapa teknologi yang sudah ada dan saling terintegrasi satu dengan yang lainnya, antara lain teknologi *Wireless sensor network* (WSN), sensor canggih serta algoritma yang semakin cerdas. Sehingga teknologi *smart home* ini mampu dapat memonitor, mengontrol, memprediksi, mengantisipasi serta membuat keputusan dalam alur kerja beberapa sistem tersebut.

### 2.2.2 Arduino Pro Mini

Arduino Pro Mini adalah board berbasis mikrokontroler ATmega328. Board ini memiliki 14 digital input / output pin (6 pin dapat digunakan sebagai output PWM), 8 pin analog, 16 Mhz osilator Kristal serta tombol reset. Untuk menulis program dari komputer ke arduino promini digunakan FTDI. Bentuk fisik arduino pro mini ditunjukkan dalam Gambar 2.1. Spesifikasi arduino promini ditunjukkan dalam Tabel 2.1.



Gambar 2.1 Board Arduino Pro Mini

Table 2.1 Spesifikasi Arduino Pro Mini

|                        |                                                   |
|------------------------|---------------------------------------------------|
| Microcontroller        | Atmega328                                         |
| Operating Voltage      | 5V ( <i>depending on model</i> )                  |
| Input Voltage          | 5 - 12 V (5V model)                               |
| Digital I/O Pins       | 14 ( <i>of which 6 provide PWM output</i> )       |
| Analog Input Pins      | 8                                                 |
| DC Current per I/O Pin | 40 mA                                             |
| Flash Memory           | 16 KB ( <i>of which 2 KB used by bootloader</i> ) |
| SRAM                   | 1 KB                                              |
| EEPROM                 | 512 bytes                                         |
| Clock Speed            | 16 MHz (5V model)                                 |

Arduino pro mini memiliki fitur-fitur sebagai berikut:

1. Serial: Pin yang digunakan untuk komunikasi Serial adalah Pin 0 sebagai TX dan pin 1 sebagai RX

2. I<sup>2</sup>C: pin yang digunakan untuk komunikasi I<sup>2</sup>C adalah pin SDA(A4) dan SDL (A5).
3. Pin Analog: Arduino Pro mini memiliki 6 pin analog yaitu A0, A1, A2, A3, A4, dan A5.
4. Pin Digital: ArduinoPro mini memiliki 14 pin digital yaitu pin 0 sampai pin 13.
5. SPI: Pin yang digunakan adalah 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).
6. PWM: Arduino pro mini sudah memiliki fitur PWM yaitu menggunakan pin 3, 5, 6, 9, 10, dan 11 dengan format 8 bit.
7. Interrupts external: pin yang digunakan adalah pin 2 dan pin 3.

### 2.2.3 Liquid Crystal Display (LCD 2x16)

LCD merupakan singkatan dari *Liquid Crystal Display* (Indonesia: Penampil Kristal Cair) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. Ada banyak jenis LCD yang beredar di pasaran. Namun ada standarisasi yang cukup populer digunakan merupakan modul LCD dengan tampilan 2x16 (2 baris x 16 kolom) dengan konsumsi daya rendah. Bentuk LCD 2x16 ditunjukkan dalam Gambar 2.2.



Gambar 2.2 Modul LCD yang digunakan

#### 2.2.3.1 Konfigurasi LCD 2x16

Modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk Mengendalikan LCD. LCD dengan jenis seperti ini memungkinkan pemrogram untuk mengoperasikan komunikasi data secara 8 bit atau 4 bit. Susunan pin LCD bertipe 2x16 secara umum ditunjukkan dalam Tabel 2.2.

Table 2.2 Spesifikasi Liquid Crystal Display

| Pin No. | Symbol | Level       | Description                    |
|---------|--------|-------------|--------------------------------|
| 1       | VSS    | OV          | Ground                         |
| 2       | VDD    | 5V          | Supply voltage for logic       |
| 3       | VO     | (Variable)  | Operating voltage for LCD      |
| 4       | RS     | H/L         | H : Data, L : Instruction code |
| 5       | R/W    | H/L         | H : Read, W : Write            |
| 6       | E      | H, H -> L   | Chip Enable Signal             |
| 7       | DB0    | H/L         | Data Bit 0                     |
| 8       | DB1    | H/L         | Data Bit 1                     |
| 9       | DB2    | H/L         | Data Bit 2                     |
| 10      | DB3    | H/L         | Data Bit 3                     |
| 11      | DB4    | H/L         | Data Bit 4                     |
| 12      | DB5    | H/L         | Data Bit 5                     |
| 13      | DB6    | H/L         | Data Bit 6                     |
| 14      | DB7    | H/L         | Data Bit 7                     |
| 15      | A      | 4,2 – 4,6 V | LED +                          |
| 16      | K      | 0V          | LED -                          |

### 2.2.3.2 Fungsi Register LCD 2X16

Modul display LCD sudah dilengkapi dengan sebuah kontroler yang memiliki dua register 8 bit yaitu instruction register (IR) dan data register (DR). IR menyimpan kode instruksi, seperti display clear, cursor shift dan informasi address untuk display data RAM (DDRAM) dan character generator (CGRAM). RAM (DDRAM) dan character generator (CGRAM) LCD 2x16 ditunjukkan dalam Tabel 2.3.

#### 1. Busy Flag (BF)

Busy flag = 1 saat kontroler sedang mengerjakan instruksi, selama instruksi tersebut belum selesai dikerjakan, kontroler tidak akan menerima instruksi

apapun. Ketika RS=0 dan R/W=1, busy flag mengeluarkan logika 1 pada DB7. Instruksi berikutnya akan siap diterima ketika busy flag = 0.

## 2. Address Counter (AC)

Address counter berisi address DDRAM dan CGRAM.

## 3. Display Data RAM (DDRAM)

DDRAM menyimpan data display dalam bentuk kode karakter 8 bit. Kapasitasnya adalah 80 karakter. Berikut adalah posisi address DDRAM pada LCD 16x2:

| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F |

Gambar 2.3 Display Data RAM (DDRAM)

## 4. Character Generator ROM (CGROM)

CGROM akan menghasilkan karakter dengan format 5x8 dot atau 5x10 dot. Lihat tabel Format Karakter CGROM

## 5. Character Generator RAM (CGRAM)

Dengan CGRAM, user dapat membuat sendiri format karakter yang diinginkan. Untuk format 5x8 dot bisa dibuat 8 karakter, untuk format 5x10 dapat dibuat 4 karakter. Lihat table

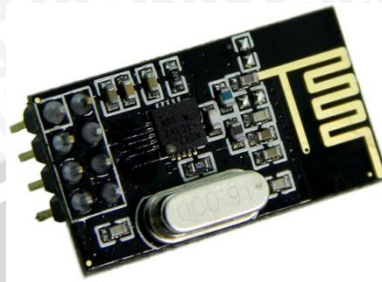
Table 2.3 RAM (DDRAM) dan character generator (CGRAM)

| RS | R/W | Operation                                                    |
|----|-----|--------------------------------------------------------------|
| 0  | 0   | Menulis instruksi ke IR, seperti display clear, cursor shift |
| 0  | 1   | Membaca busy flag (DB7) dan address counter (DB0 s/d DB7)    |
| 1  | 0   | Menulis data ke DDRAM atau CGRAM                             |
| 1  | 1   | Membaca data dari DDRAM atau CGRAM                           |

### 2.2.4 NRF24L01

NRF24L01 adalah sebuah modul komunikasi jarak jauh dengan single-chip RF-transceiver yang ditujukan untuk aplikasi pada gelombang 2.4 GHz ISM

band. Modul NRF24L01 menggunakan antar muka SPI (Serial Peripheral Interface) untuk komunikasi dengan tegangan kerja 5V DC.



Gambar 2.4 NRF24L01

Spesifikasi dari modul NRF24L01 adalah sebagai berikut:

- Worldwide 2.4GHz ISM band operation
- Jangkauan data di udara yaitu 250kbps, 1Mbps and 2Mbps
- Ultra low power operation
- 11.3mA TX at 0dBm output power
- 12.3mA RX at 2Mbps air data rate
- Power turun pada 900nA
- 22 $\mu$ A in standby-I
- On chip voltage regulator
- 1.9 to 3.6V supply range
- Enhanced ShockBurst™
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™
- Drop-in compatibility with Nrf24L01
- On-air compatible in 250kbps dan 1Mbps dengan Nrf2401A, Nrf2402, Nrf24E1 dan Nrf24E2
- Low cost BOM
- $\pm 60$ ppm 16MHz crystal
- 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

### 2.2.5 Fingerprint Scanner GT-511C1

*Fingerprint Scanner GT-511C1* merupakan modul sidik jari yang menggunakan 32-bit CPU dan berkomunikasi secara *TTL Serial*. Pada pembacaan sidik jari, modul ini dapat menangkap hingga 30° gambar dan dapat menyimpan sebanyak 20 *template fingerprint* yang berbeda. Modul ini juga dilengkapi dengan *software demo* dimana dapat mempermudah untuk mencoba dari masing-masing fungsi dari *device* tersebut. Pengguna juga dapat langsung mengambil gambar sidik jari dari *fingerprint* maupun gambar *RAW* dari *Optical Sensor*. Pada modul ini terdapat 4 buah pin untuk digunakan mengakses, yaitu: VCC, GND, TX, RX sehingga akan lebih mudah digunakan oleh pengguna. Untuk ukuran gambar sidik jari yang disimpan adalah 216 x 240 Pixels dan mempunyai resolusi sebesar 450 dpi.



Gambar 2.5 NRF24L01

### 2.2.6 4x4 Matrix Membrane Keypad

4x4 Matrix Membrane Keypad mempunyai beberapa fitur dan spesifikasi seperti berikut:

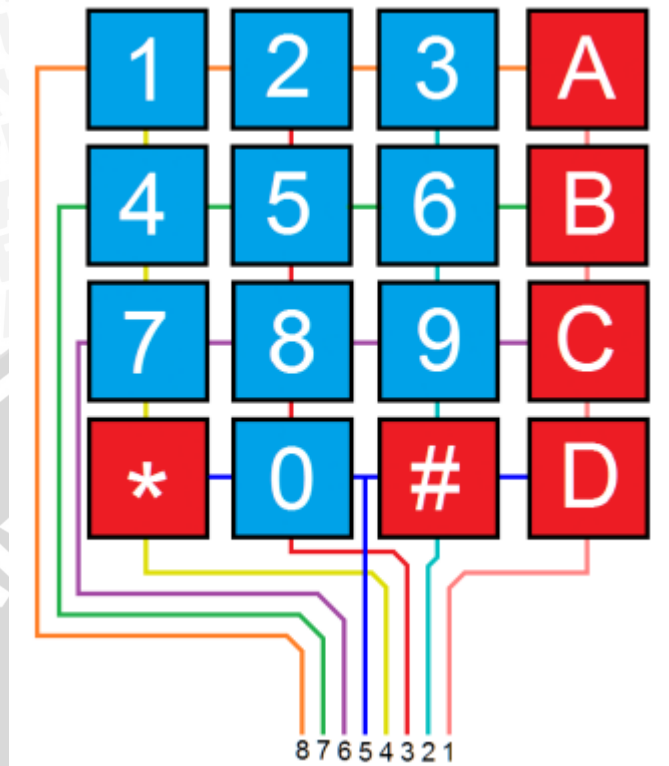
- ❖ Fitur
  - Mempunyai desain yang sangat tipis
  - Mempunyai harga yang murah dan kinerja yang baik
  - Menggunakan antarmuka yang mudah untuk setiap mikrokontroller
- ❖ Spesifikasi

- Maksimum input 24 VDC, 30mA
  - Suhu maksimal penggunaan 0 - 50°C
  - Menggunakan 8 pin untuk mengakses
  - Dimensi *Keypad* (6.9 x 7.6 cm), kabel (2,0 x 8,8 cm)
- ❖ Implementasi *Keypad*
- Sistem keamanan
  - Entry data untuk embeded sistem



Gambar 2.6 4x4 matrix membrane *Keypad*

Cara kerja *Keypad* diatas adalah 4 kolom diberikan nilai rendah/ tinggi tergantung dari program yang digunakan, setelah itu ditekan tekan salah satu dari keempat baris. Dari kondisi itu mikrokontroller akan mengetahui kondisi yang ditekan. Konfigurasi pin dari 4x4 matrix membran *Keypad* adalah sebagai berikut :



Gambar 2.7 koneksi 4x4 matrix membrane Keypad

### 2.2.7 Wireless Sensor Network (WSN)

*Wireless Sensor Network* atau disingkat dengan WSN adalah suatu peralatan sistem embedded yang didalamnya terdapat satu atau lebih sensor dan dilengkapi dengan peralatan sistem komunikasi. Sensor disini digunakan untuk menangkap informasi sesuai dengan karakteristik informasi yang diinginkan.

Sebuah jaringan sensor merupakan sebuah insfratuktur yang terdiri dari sensing (masuring), komputasi dan elemen komunikasi yang memberikan administrator untuk melakukan instrumentsasi, mengamati, dan bereaksi terhadap fenomena dalam lingkungan tertentu. Administrator yang dimaksud adalah entitas sipil, komersial, ataupun industrial. Efisiensi daya di WSN umumnya dicapai dalam tiga cara yaitu:

1. Operasi *Low-duty-cycle*
2. Proses local atau di sebuah jaringan untuk operasi mengurangi volume data (karenanya transmisi waktu)
3. Jaringan multihop untuk mengurangi kebutuhan transmisi jarak jauh.

Pada saat ini ada berbagai jenis jaringan sensor nirkabel. *Wireless Sensor Network* saat ini dilengkapi dengan transceiver radio atau perangkat komunikasi nirkabel dan sumber energi yang biasanya berupa baterai. Ada empat komponen dasar dalam jaringan sensor:

- a) Perakitan sensor secara distribusi atau lokal
- b) Jaringan interkoneksi
- c) Titik pusat dari informasi clustering
- d) Satu set sumber daya komputasi pada titik pusat (atau di luar) untuk menangani korelasi data, acara tren, status query, dan data mining.

Terdapat 2 kategori dalam WSN, yaitu:

1) WSN Kategori 1 (C1WSNs)

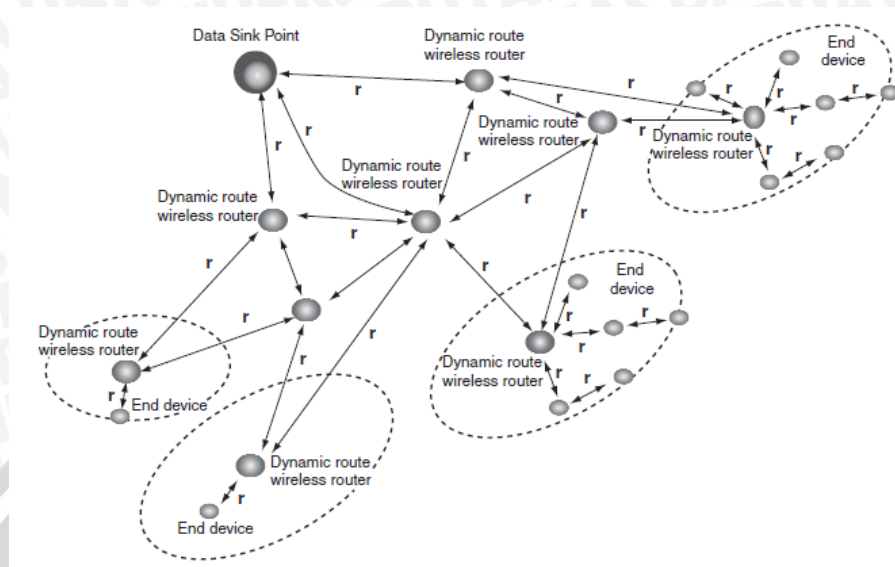
Pada kategori 1, jaringan menggunakan topologi mesh dengan jaringan radio multihop di kalangan atau diantara node pada WSN, dan memanfaatkan routing dinamis pada wireless dan kabel jaringan.

C1WSNs merupakan jaringan dimana perangkat akhir (*end device*) yaitu sensor, diijinkan untuk lebih dari satu radio hop yang jauh dari *forwarding node*.

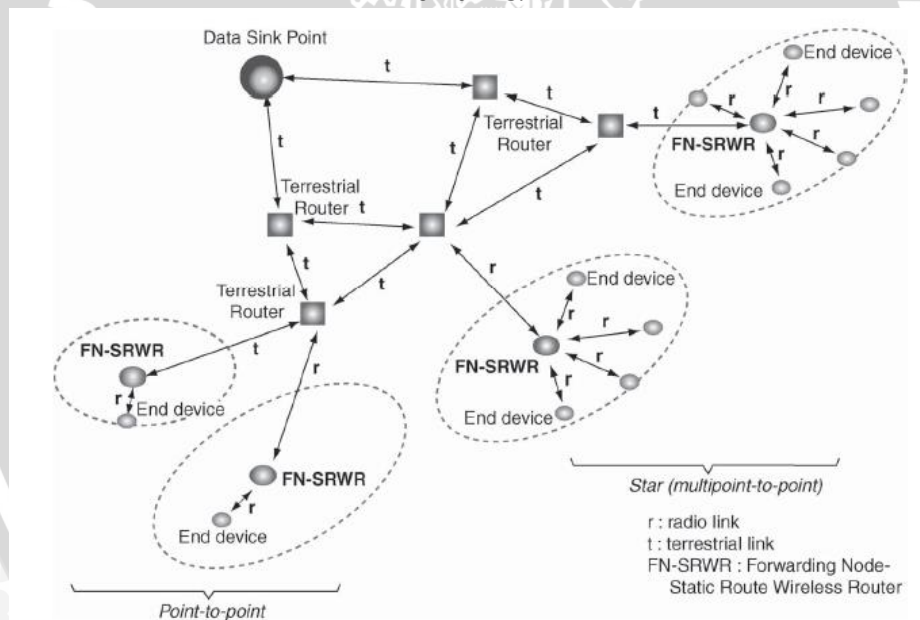
2) WSN Kategori 2 (C2WSNs)

Pada kategori 2, jaringan menggunakan topologi star (point-to-point atau multipoint-to-point) dengan jaringan radio single-hop, memanfaatkan routing statis melalui jaringan nirkabel.

C2WSNs merupakan jaringan dimana perangkat akhir (*end device*) yaitu sensor, hanya terdapat satu radio hop yang jauh dari *forwarding node*.



Gambar 2.8 WSN Kategori 1: multipoint-to-point, sistem multihop dengan routing dinamis.



Gambar 2.9 WSN Kategori 2: point-to-point, sistem singlehop dengan routing statis

## 2.2.8 MySQL

MySQL adalah sebuah sistem manajemen database yang paling populer yang merupakan Open Source SQL database yang dikembangkan, didistribusikan, dan didukung oleh Oracle Corporation. MySQL merupakan salah satu RDBMS (*Relational Data Base Managemnet System*) terbaik yang digunakan untuk mengembangkan web berbasis aplikasi perangkat lunak.

Data Base merupakan kumpulan data yang terstruktur. Misalnya untuk MySQL server, diperlukan sebagai sebuah sistem manajemen Data Base agar dapat dengan mudah menambah, mengakses, serta memproses data yang disimpan dalam computer data base. SQL (*Structured Query Language*) merupakan bahasa standar yang paling umum digunakan dalam mengakses database. MySQL bersifat Open Source, yang artinya siapa saja dapat menggunakan dan memodifikasinya dengan mengunduh perangkat MySQL secara gratis.

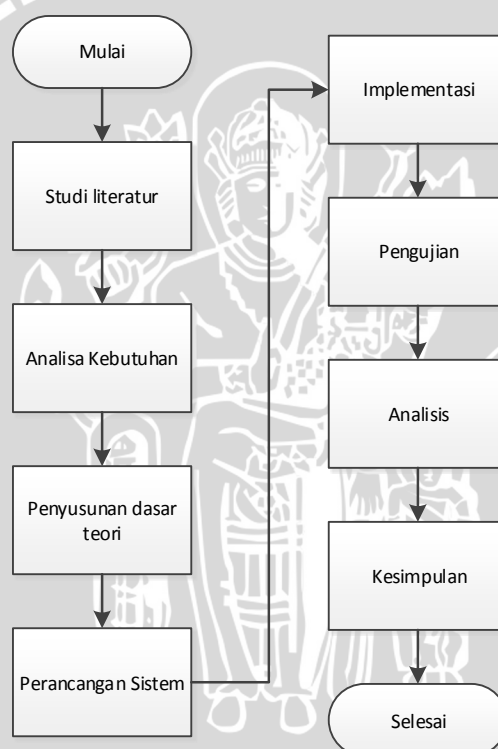
MySQL Server bekerja di client/server atau pada sistem embedded. Data base MySQL merupakan sistem client/server yang terdiri dari server SQL multi-threaded yang mendukung backends yang berbeda, beberapa program client dan *libraries* yang berbeda, alat administrasi dan beberapa antarmuka aplikasi pemrograman (*Application Programming Interfaces* atau APIs).



## BAB III

### METODOLOGI PENELITIAN

Bab ini membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, perancangan, implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran dan Membahas analisis kebutuhan dan perancangan untuk pengujian perangkat keras (*hardware*). Kesimpulan dan saran disertakan sebagai catatan atas performa dari penelitian ini dan kemungkinan arah pengembangan penelitian selanjutnya. Berikut ini merupakan diagram alir runtutan pengerjaan penelitian ini:



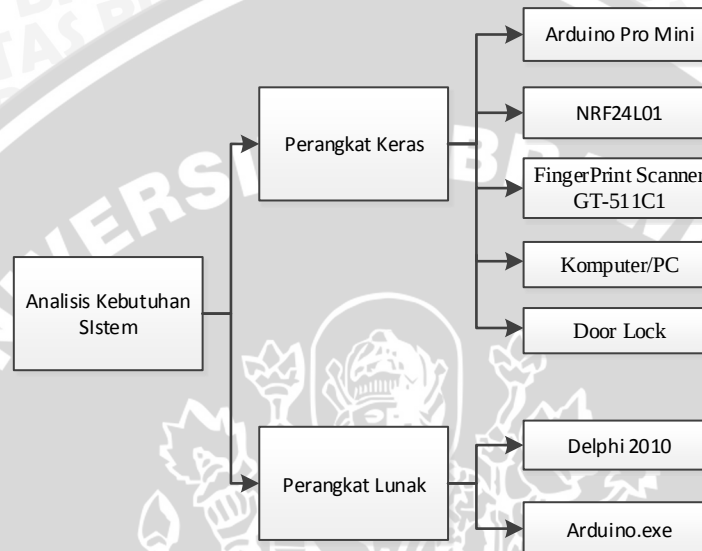
Gambar 3.1 Flowchart Pengerjaan Penelitian

#### 3.1. Studi Literatur

Studi literatur mempelajari mengenai penjelasan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut diperoleh dari buku, jurnal, e-book, dan dokumentasi project.

### 3.2. Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh sistem yang akan dibangun dan diuji. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat didalamnya. Berikut analisis kebutuhan dalam penelitian yang akan dilakukan.



Gambar 3.2 Diagram Blok Analisis Kebutuhan Sistem

#### 3.2.1. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

- 1) Arduino Pro mini dan Board PCB digunakan sebagai kalkulasi algoritma.
- 2) *Fingerprint Scanner* GT-511C1 sebagai sensor sidik jari.
- 3) 4x4 Membrane Keypad.
- 4) *Door Lock* sebagai aktuator pengunci pintu.

#### 3.2.2. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam penelitian ini adalah Arduino IDE sebagai *compiler* program yang akan dituliskan ke dalam arduino promini.

### 3.3. Penyusunan Dasar Teori

Penyusunan dasar teori dilakukan setelah mendapatkan referensi yang tepat untuk mendukung penulisan penelitian ini. Teori-teori pendukung tersebut meliputi:

1. *Smart Home*
2. *Wireless Sensor Network (WSN)*
3. *Arduino Pro mini*
4. *NRF24L01*
5. *Liquid Crystal Display (LCD 2x16)*
6. *FingePrint Scanner GT-511C1*
7. *4x4 Membrane Keypad*

### 3.4. Perancangan Sistem

Pada perancangan sistem ini dibentuk dalam diagram blok yang diagram blok tersebut memiliki bagian-bagian dari beberapa sistem yang dibuat. Bagian-bagian dari diagram blok tersebut mempunyai tugas yang berbeda sehingga dari beberapa blok tersebut menjadi sebuah sistem yang utuh sesuai dengan tujuan penelitian ini. Diantara diagram tersebut terdapat sistem menambahkan user baru pada *Fingerprint Scanner GT-511C1*, menghapus user pada *Fingerprint Scanner GT-511C1*, menambahkan user baru pada *Base Station (Server)*, menghapus user pada *Base Station (Server)*, verifikasi pada modul sidik jari dan pada *Base Station (Server)*.

### 3.5. Implementasi

Implementasi sistem ini dilakukan dengan mengacu pada perancangan sistem. Implementasi perangkat keras dan perangkat lunak meliputi:

- 1) Perancangan *hardware* (perangkat keras)
- 2) Perancangan *algoritma enroll* data sidik jari ke *Fingerprint Scanner*
- 3) Perancangan *algoritma* mendapatkan ID data sidik jari dari *Fingerprint Scanner*
- 4) Perancangan *algoritma* hapus ID data sidik jari dari *Fingerprint Scanner*

- 5) Perancangan *algoritma enroll* data ke *base station*
- 6) Perancangan *algoritma* verifikasi data dari *base station*
- 7) Perancangan *algoritma* hapus user dari *base station*

### 3.6. Pengujian

Pengujian pada sistem dilakukan dengan mengacu pada perancangan sistem, yang meliputi:

- 1) Pengujian *algoritma enroll* data sidik jari ke *Fingerprint Scanner*
- 2) Pengujian *algoritma* mendapatkan ID data sidik jari dari *Fingerprint Scanner*
- 3) Pengujian *algoritma* hapus ID data sidik jari dari *Fingerprint Scanner*
- 4) Pengujian *algoritma enroll* data ke *base station*
- 5) Pengujian *algoritma* verifikasi data dari *base station*
- 6) Pengujian *algoritma* hapus user dari *base station*
- 7) Pengujian *DoorLock*
- 8) Pengujian komunikasi data pada WSN (node)

### 3.7. Analisis

Dari pengujian yang dilakukan, data dari masing-masing tahap pengujian kemudian dianalisis untuk menentukan apakah sistem sudah berjalan baik sesuai dengan tujuan penulisan.

### 3.8. Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

## BAB IV

### PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi perancangan dan implementasi untuk penelitian. Perancangan membahas tentang rancangan yang ada pada penelitian. Dan implementasi membahas tentang pengimplementasian penelitian pada sistem yang ada di lingkungan.

#### 4.1. Perancangan

Tahap perancangan berisi tentang analisis perangkat keras dan perangkat lunak yang digunakan untuk membuat sistem tersebut. Ada beberapa tahap di antaranya sebagai berikut:

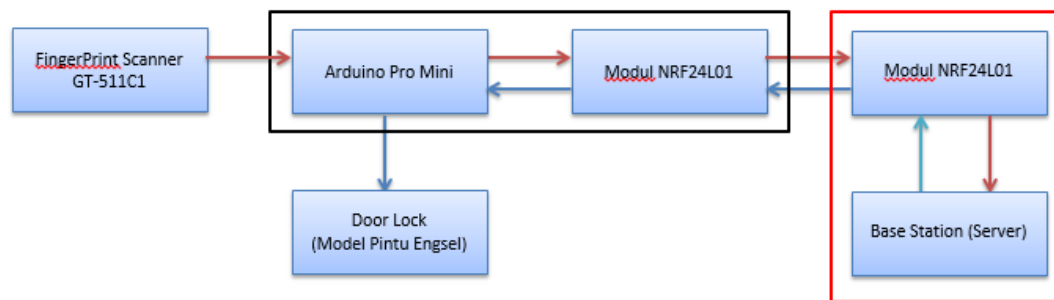
##### 4.1.1. Analisis Kebutuhan Sistem

Analisis kebutuhan yang dibutuhkan seperti perangkat keras dan perangkat lunak yang akan digunakan untuk proses implementasi dari sistem yang ada. Di antaranya kebutuhan yang dibutuhkan yaitu:

- a. Arduino dan NRF24L01
- b. *Fingerprint Scanner GT-511C1* sebagai modul sidik jari
- c. DoorLock (Solenoid)
- d. LCD 16x2
- e. Rangkain
- f. Arduino IDE
- g. PC/ Komputer sebagai server (Base Station)

##### 4.1.2. Diagram Blok Sistem

Dari analisis kebutuhan yang ada dapat kita buat diagram blok dari sistem tersebut yang terdiri dari bagian-bagiannya. Diagram blok sistem ditunjukkan dalam Gambar 4.1. Pada diagram blok yang ditunjukkan dalam Gambar 4.1 sistem menggunakan input yang berasal dari modul *Fingerprint Scanner* yang selanjutnya akan dilakukan pencocokan pada *Base Station (Server)*. Setelah data cocok kemudian dikirim perintah pada aktuator.



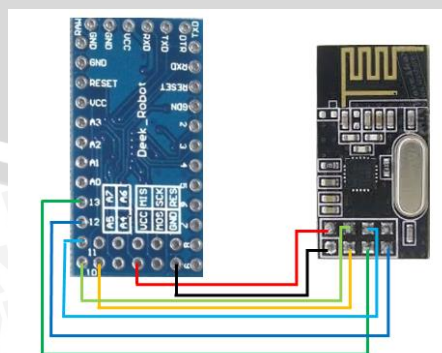
Gambar 4.1 Diagram Blok Sistem

#### 4.1.3. Perancangan Perangkat Keras

Pada perancangan perangkat keras terdapat beberapa perancangan, yaitu antara lain: perancangan *hardware*. Perancangan *hardware* yang akan diimplementasikan berdasarkan analisis kebutuhan sistem. Diantaranya Arduino Pro Mini, NRF24L01, *Fingerprint Scanner*, *Keypad*, dan *Door Lock*.

##### 4.1.3.1. Perancangan Antarmuka Arduino dan NRF24L01

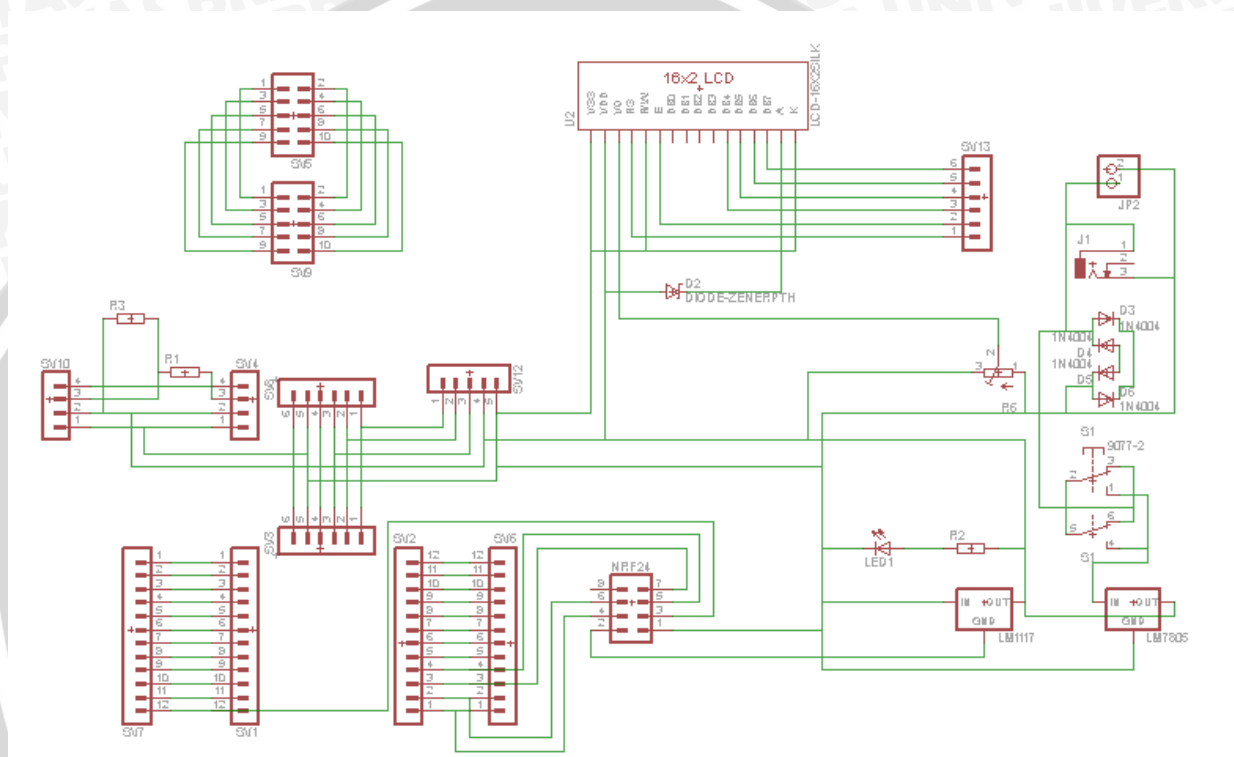
Rangkaian media komunikasi radio frekuensi (RF) NRF24L01 dan arduino digunakan sebagai media pengiriman data atau perintah dari pengguna untuk menyalakan atau mematikan lampu dari jarak jauh. Protokol yang digunakan arduino dan NRF24L01 untuk komunikasi adalah Serial Peripheral interface (SPI), maka pin yang dipakai adalah pin yang digunakan adalah 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Serta dua pin tambahan yaitu pin A5 untuk *Clock Enable* dan pin 10 untuk CSN. Skematik rangkaian arduino dan NRF24L01 ditunjukkan pada Gambar dibawah 4.2.



Gambar 4.2 Rangkaian Antarmuka Arduino dan NRF24L01

#### 4.1.3.2. Perancangan Board PCB

Perancangan Board PCB adalah merupakan bagian pokok dari perangkat keras. Pada perancangan ini berisi tentang schema antar pin pada masing-masing device (arduino, nrf24l01, *Fingerprint Scanner* dll). Schematic Board PCB ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Schematic Board PCB

#### 4.1.4. Perancangan Perangkat Lunak

Pada perancangan perangkat lunak terdapat beberapa perancangan, yaitu antara lain: perancangan algoritma *enroll* data ke *Fingerprint Scanner*, perancangan algoritma mendapatkan id dari *Fingerprint Scanner*, perancangan algoritma menghapus id dari *Fingerprint Scanner*, perancangan paket data, perancangan algoritma *enroll* data ke *Base Station (Server)*, perancangan algoritma verifikasi data ke *Base Station (Server)*, perancangan algoritma menghapus data di *Base Station (Server)* dan aktuator.

#### 4.1.4.1. Perancangan Algoritma *Enroll Data ke Fingerprint Scanner*

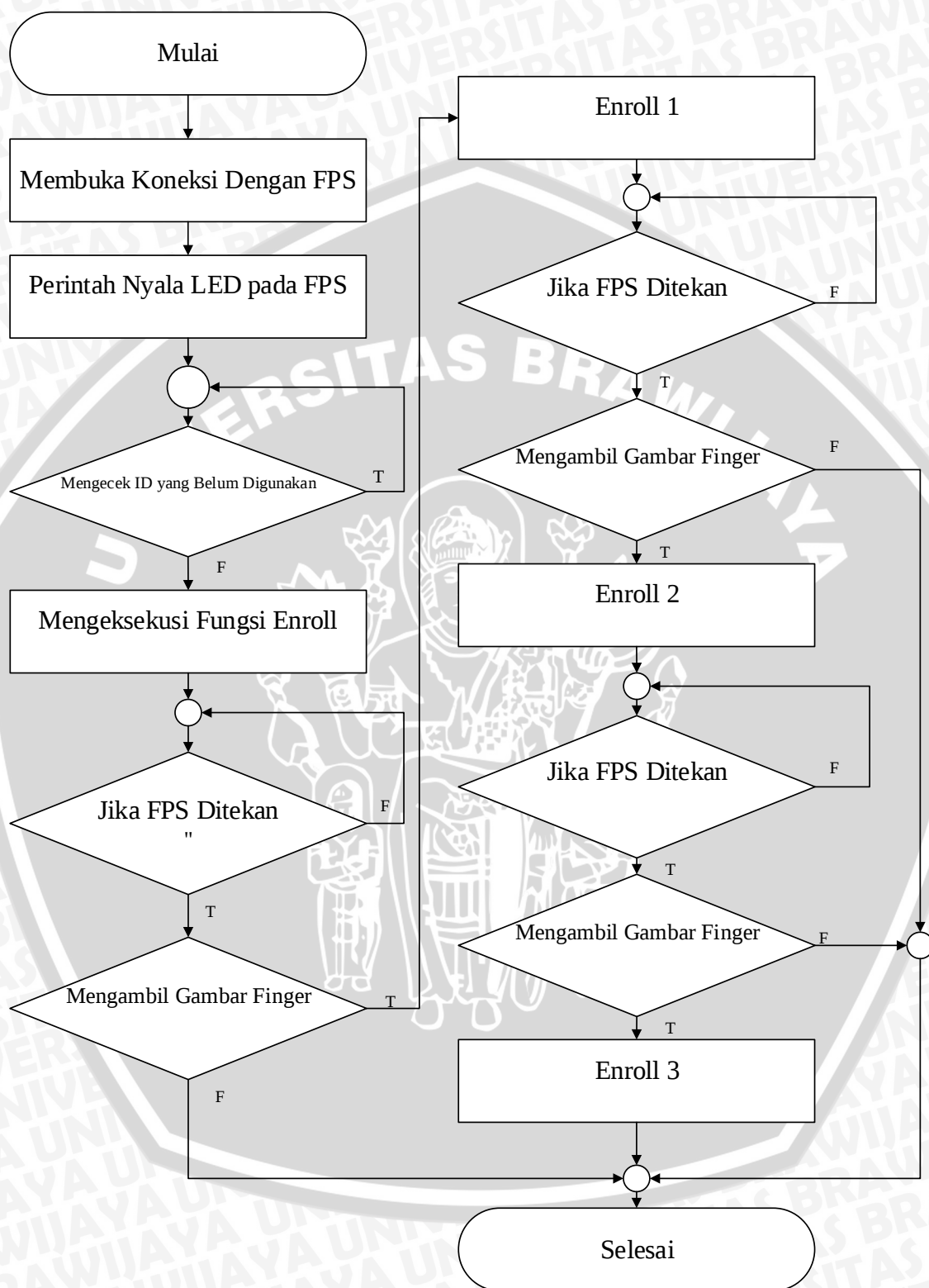
Perancangan algoritma *enroll data ke Fingerprint Scanner* digunakan untuk memasukan data pengguna/ user yang berupa *template* sidik jari. Untuk masing-masing pengguna melakukan scan *template* sidik jari sebanyak 3 kali. *Template* sidik jari pengguna/ user tersebut masing-masing akan mendapatkan identitas yaitu berupa nomor ID. Diagram alir dari proses ditunjukkan pada Gambar 4.4.

#### 4.1.4.2. Perancangan Algoritma Mendapatkan ID Dari *Fingerprint Scanner*

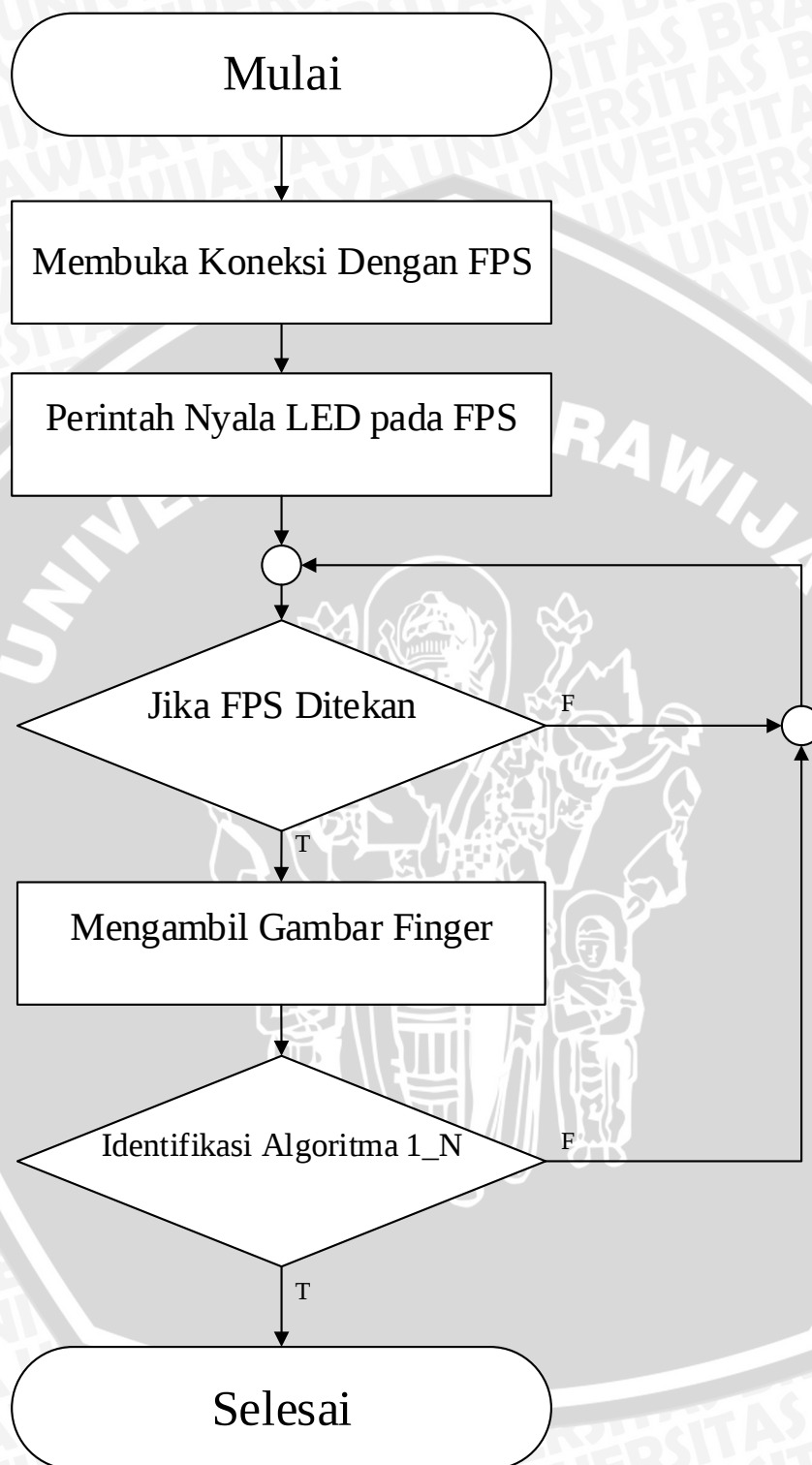
Perancangan algoritma mendapatkan ID dari *Fingerprint Scanner* digunakan untuk proses verifikasi pada *Fingerprint Scanner*. Pada proses verifikasi tersebut hasil akhir yang didapat adalah berupa nomor ID dari *template* sidik jari yang telah tersimpan pada database *Fingerprint Scanner*. Diagram alir dari proses ditunjukkan pada Gambar 4.5.

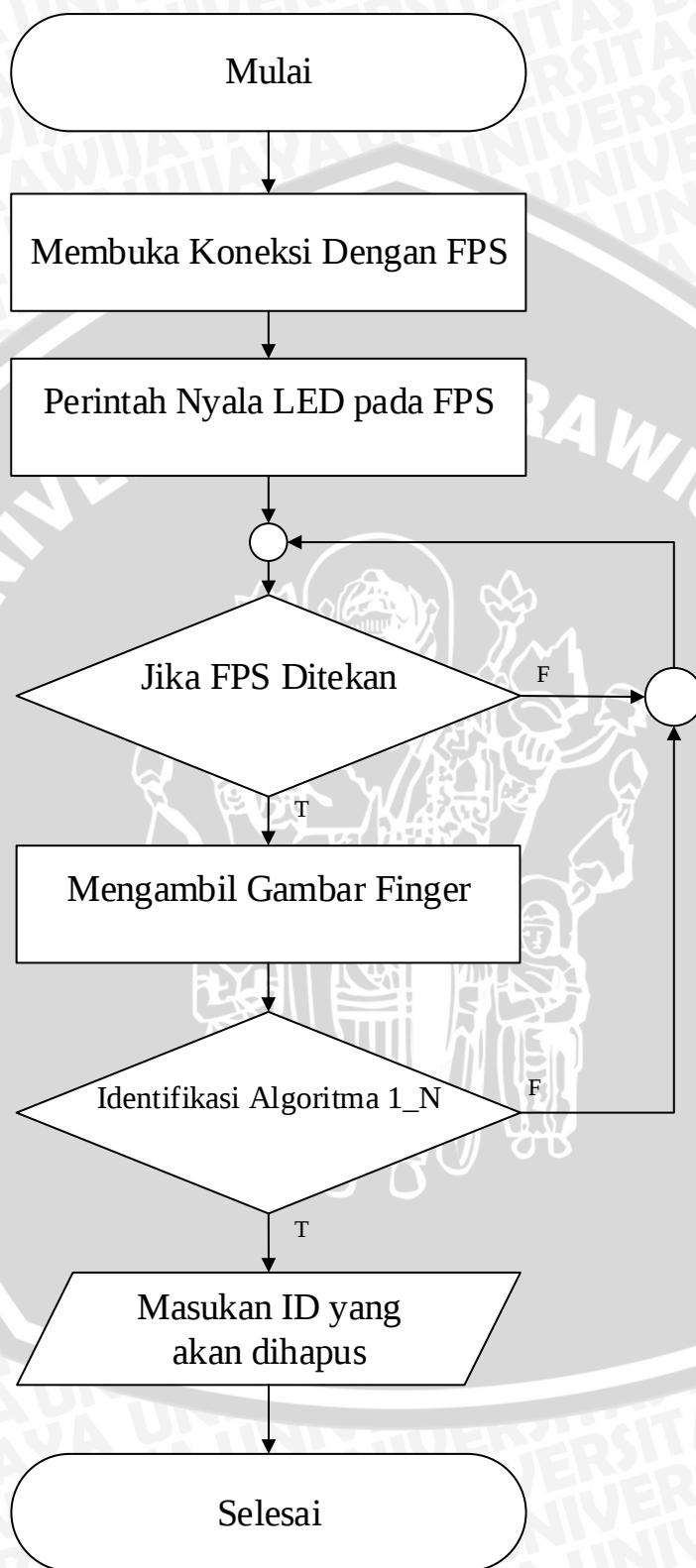
#### 4.1.4.3. Perancangan Algoritma Menghapus ID Dari *Fingerprint Scanner*

Perancangan algoritma menghapus ID dari *Fingerprint Scanner* digunakan untuk menghapus nomor ID beserta dengan tempalte sidik jari pengguna/ user yang terdapat pada databases *Fingerprint Scanner*. Diagram alir dari proses ditunjukkan pada Gambar 4.6.



Gambar 4.4 Diagram Alir Enroll Data ke Fingerprint Scanner

Gambar 4.5 Diagram Alir Mendapatkan ID dari *Fingerprint Scanner*

Gambar 4.6 Diagram Alir Menghapus ID dari *Fingerprint Scanner*

#### 4.1.4.4. Perancangan Paket Data

Perancangan paket data pada penelitian ini terdiri atas dua bagian yaitu paket data antara Arduino ke Komputer dan paket data dari Arduino ke *node* yang lain.

1. Perancangan paket data dari arduino ke komputer dan sebaliknya.

Perancangan paket data dari Arduino ke komputer diperlukan dalam proses pengiriman data untuk mengaktifkan atau mematikan objek. Data dari objek yang diterima oleh Arduino akan dikirim ke komputer sesuai dengan paket data yang dirancang. Komputer akan menerima data berupa nilai dan perintah dari objek pengirim sesuai dengan *source\_ID* yang dikirim dari Arduino.

Paket data dari komputer ke Arduino digunakan sebagai pengembalian perintah ke objek sesuai dengan ID objek tersebut. Dengan adanya perancangan paket data ini diharapkan perintah yang harus dieksekusi tepat sasaran pada objek yang dituju.

- a. Paket Data dari Arduino ke Komputer

Rancangan paket data dari Arduino ke komputer ditunjukkan pada Tabel 4.1. dan keterangan dari paket data tersebut ditunjukkan pada Tabel 4.2.

Tabel 4.1 Rancangan Paket Data dari Arduino ke Komputer

| Panjang   | 1    | 1        | 1     | 1         | 1    |
|-----------|------|----------|-------|-----------|------|
| Data      | Byte | Byte     | Byte  | Byte      | Byte |
| Nama Slot | {    | Perintah | Nilai | Source_Id | }    |

Tabel 4.2 Keterangan Paket Data dari Arduino ke Komputer

| No | Nama Slot |   | Deskripsi                                                                                                                                            |
|----|-----------|---|------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | {         |   | <i>Header</i>                                                                                                                                        |
| 2. | Perintah  | - | - untuk lampu, nilai 0-100 menunjukkan intensitas cahaya<br>- untuk beban yang bersifat <i>ON/OFF</i> , nilai 0 = <i>false</i> dan 100 = <i>true</i> |
| 3. | To_Id     |   | Untuk <i>finger print</i> : mengambil ID<br>Untuk PIR : mengambil kondisi                                                                            |
| 4. | }         |   | <i>Header</i>                                                                                                                                        |

b. Paket Data dari Komputer ke Arduino

Rancangan paket data dari komputer ke Arduino dapat dilihat pada Tabel

4.3. dan keterangan dari paket data tersebut ditunjukkan pada Tabel 4.4.

Tabel 4.3 Rancangan Paket Data dari Komputer ke Arduino

| Panjang Data | 1 Byte | 1 Byte   | 1 Byte | 1 Byte |
|--------------|--------|----------|--------|--------|
| Nama Slot    | {      | perintah | To_ID  | }      |

Tabel 4.4 Keterangan Paket Data

| No | Nama Slot | Deskripsi                                                                                                                                   |
|----|-----------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | {         | Header                                                                                                                                      |
| 2. | Perintah  | 1 : insert data <i>finger print</i><br>1 : verifikasi <i>finger print</i><br>2 : hapus data <i>finger print</i><br>3 : terima data dari PIR |
| 3. | Nilai     | Untuk <i>finger print</i> : mengambil ID<br>Untuk PIR : mengambil kondisi                                                                   |
| 4. | Source_ID | Alamat node pengirim                                                                                                                        |
| 5. | }         | Header                                                                                                                                      |

2. Perancangan Paket Data dari Arduino ke Node Lain

Tabel 4.5 Rancangan Paket Data

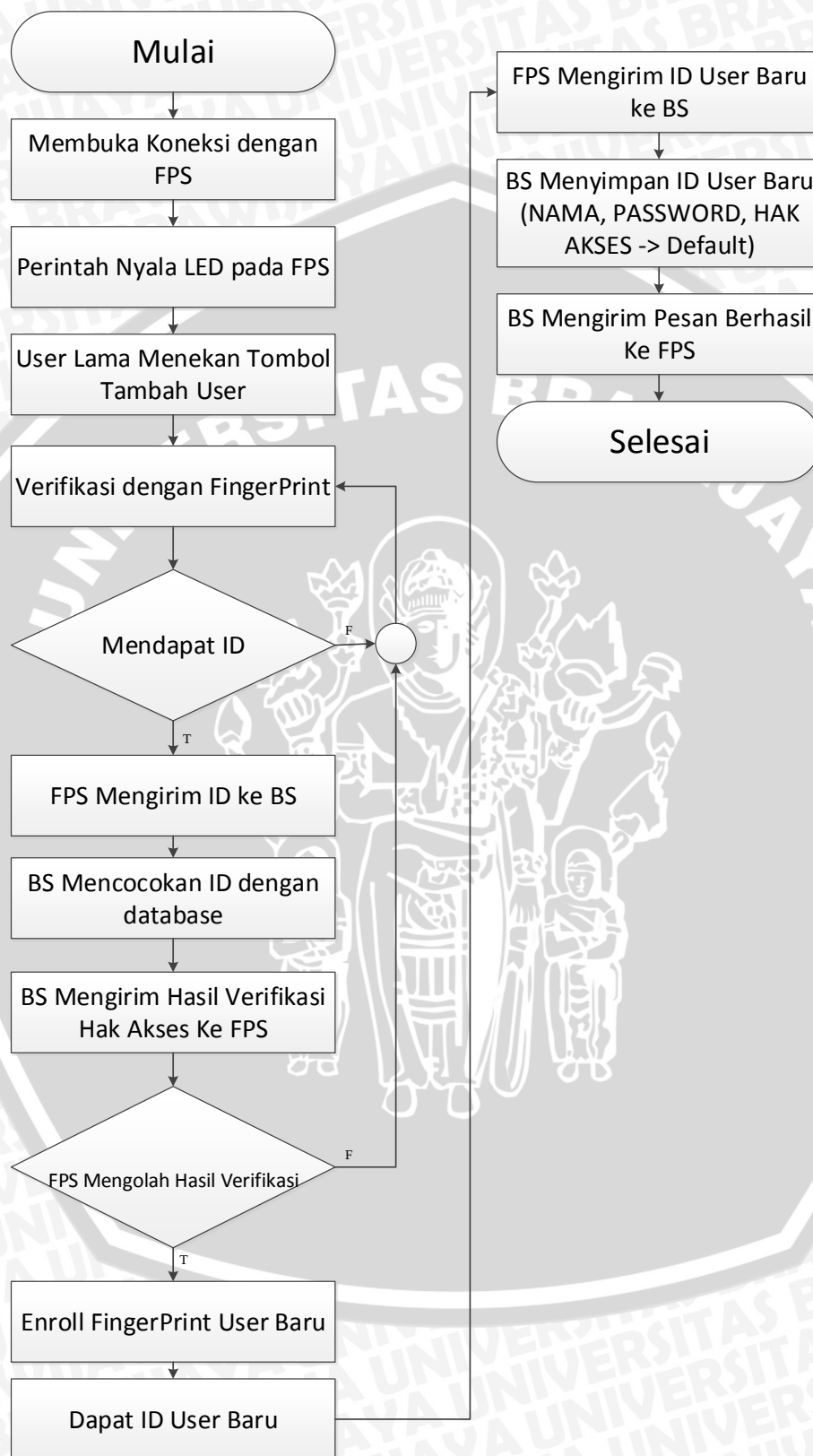
| Panjang Data | 1 Byte | 1 Byte     | 1 Byte     | 1 Byte  | 1 Byte | 1 Byte | 1 Byte | 1 Byte | 1 Byte | 1 Byte          | 1 Byte   |
|--------------|--------|------------|------------|---------|--------|--------|--------|--------|--------|-----------------|----------|
| Nama Slot    | Header | Ready/Busy | Read/Write | Form_ID | To_ID  | Jam    | Menit  | Detik  | Nilai  | General Purpose | Checksum |

Tabel 4.6 Keterangan Paket Data

| No  | Nama Slot       | Deskripsi                                                                                                            |
|-----|-----------------|----------------------------------------------------------------------------------------------------------------------|
| 1.  | Header          | Paket data ini digunakan untuk mendeteksi awal data                                                                  |
| 2.  | Ready/Busy      | Nilai pada paket ini mengganti status node apakah ready atau busy. Nilainya diisi 1 jika busy, diisi 0 jika ready    |
| 3.  | Read/Write      | Nilai pada paket ini diisi 1 untuk operasi write, dan diisi 0 untuk operasi read                                     |
| 4.  | From_ID         | Slot ini berisi informasi ID pengirim                                                                                |
| 5.  | To_ID           | Slot ini berisi informasi ID penerima                                                                                |
| 6.  | Jam             | Variabel ini khusus untuk node sensor yang menggunakan RTC untuk menyimpan waktu                                     |
| 7.  | Menit           | Variabel ini khusus untuk node sensor yang menggunakan RTC untuk menyimpan waktu                                     |
| 8.  | Detik           | Variabel ini khusus untuk node sensor yang menggunakan RTC untuk menyimpan waktu                                     |
| 9.  | Nilai           | Variabel ini berisi nilai yang digunakan untuk apa saja sesuai kebutuhan. Contohnya untuk menyimpan nilai intensitas |
| 10. | General Purpose | Variabel ini adalah variabel untuk kebutuhan khusus jika dibutuhkan                                                  |
| 11. | Checksum        | Variabel ini digunakan untuk pengecekan kerusakan paket data.                                                        |

#### 4.1.4.5. Perancangan Algoritma *Enroll Data ke Base Station (Server)*

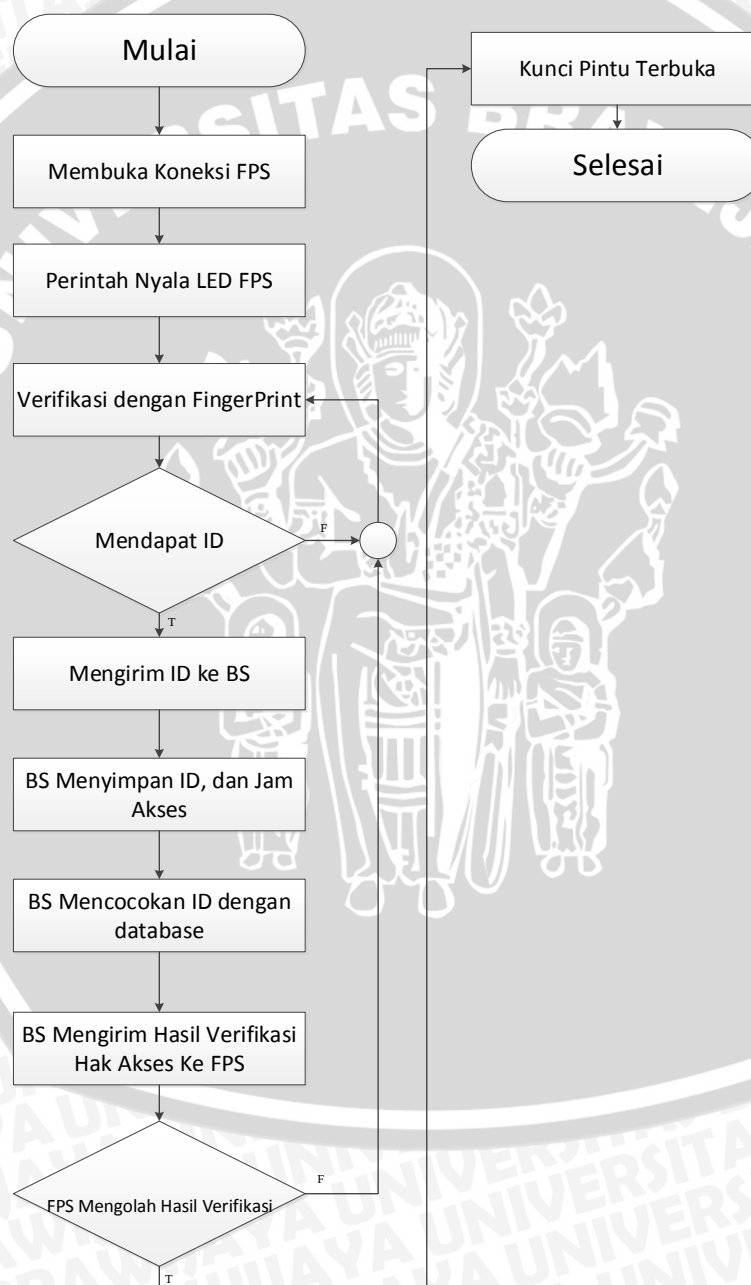
Perancangan algoritma *enroll data ke Base Station (Server)* digunakan untuk memasukan data pengguna yang telah disimpan pada *Fingerprint Scanner*. Data yang dikirim dari *Fingerprint Scanner* adalah nomor ID dari masing-masing pengguna/ user. Nomor ID tersebut akan dilengkapi dengan data pengguna/ user yang lain seperti nama, password dll, selanjutnya disimpan pada databases server. Diagram alir ditunjukkan pada Gambar 4.6.



Gambar 4.6 Diagram Alir *Enroll* Data ke *Base Station* (Server)

#### 4.1.4.6. Perancangan Algoritma Verifikasi Data ke Base Station (Server)

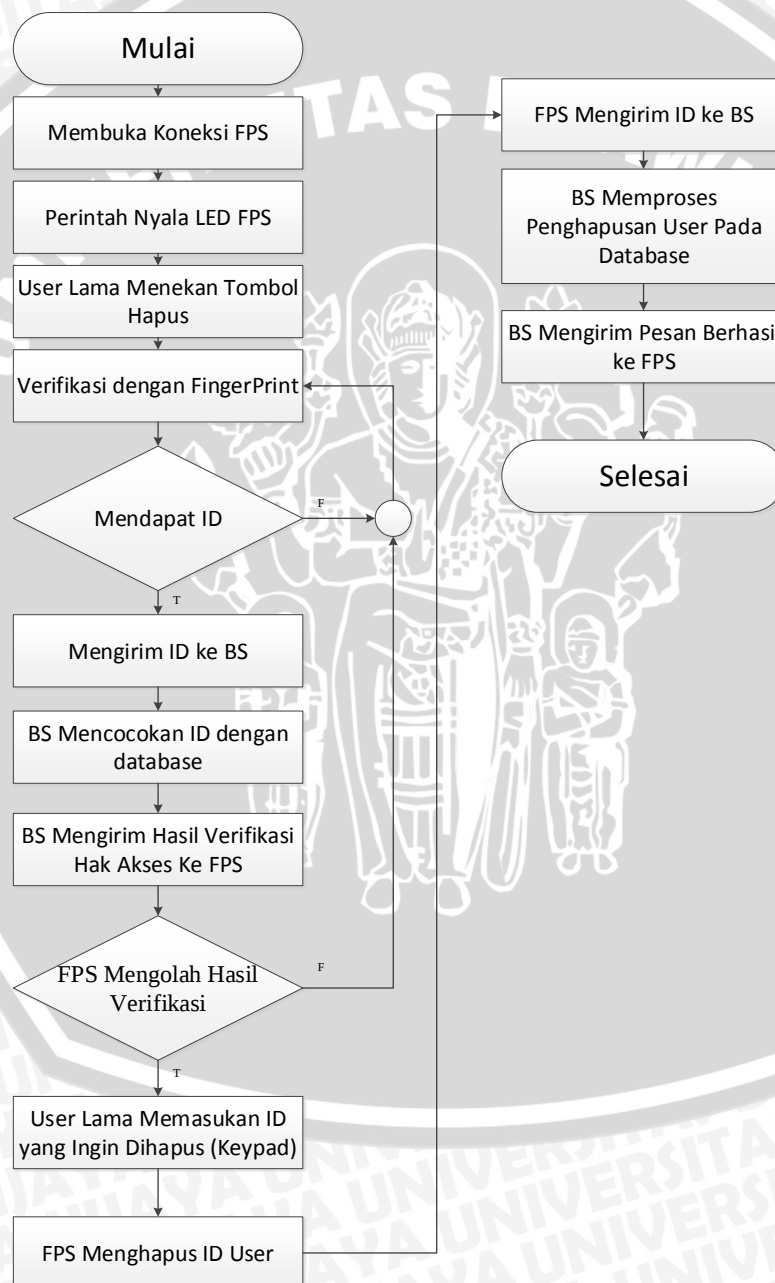
Perancangan algoritma verifikasi data ke base statiton (*Server*) digunakan untuk proses verifikasi pada base station. Nomor ID yang diterima akan dicocokkan dengan data ID yang ada pada database server. Pada saat proses verifikasi, tanggal dan jam akses dari pengguna/ user juga akan disimpan pada databases server untuk proses monitoring. Diagram alir ditunjukkan pada Gambar 4.7.



Gambar 4.7 Diagram Alir Verifikasi Data ke Base Station (Server)

#### 4.1.4.7. Perancangan Algoritma Menghapus Data di *Base Station (Server)*

Perancangan algoritma menghapus data di *Base Station* digunakan untuk menghapus data pengguna/ user pada databases server. Proses ini bertujuan untuk menyamakan antara data pengguna/ user yang tersimpan di *Fingerprint Scanner* dan data pengguna/ user yang tersimpan di databases server. Diagram alir ditunjukkan pada Gambar 4.8.



Gambar 4.8 Diagram Alir Hapus Data di *Base Station (Server)*

#### 4.2.1. Implementasi Perangkat Keras

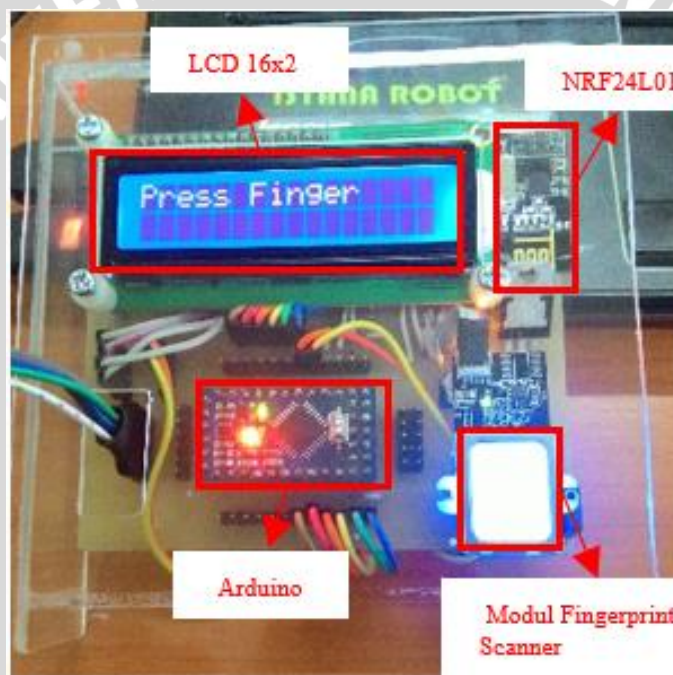
### Gambar 4.9 Board Modul

### Tabel 4.7 Komponen Pada Board

| Nama            | Keterangan | Jumlah (buah) |
|-----------------|------------|---------------|
| Arduino Promini | -          | 1             |
| LCD 16x2        | -          | 1             |
| Modul nRF24L01  | -          | 1             |
| Regulator       | 7805 CV    | 1             |

|               |                 |        |
|---------------|-----------------|--------|
| Jack power DC | Socket          | 1      |
| Regulator     | LM1117T         | 1      |
| LED           | 5mm             | 1      |
| Kapasitor     | 100 $\mu$ Farad | 1      |
| Resistor      | 330 $\Omega$    | 6      |
| Pin header    | Male<br>Female  | 3<br>3 |
| Dioda zener   | 4.3V            | 1      |

Pada Gambar 4.9 diatas menunjukan desain yang dirancang pada *software* Eagle 6.5. Selanjutnya desain board tersebut akan dicetak dan akan menjadi sebuah perangkat keras yang akan menjadi modul wireless seperti pada Gambar 4.10

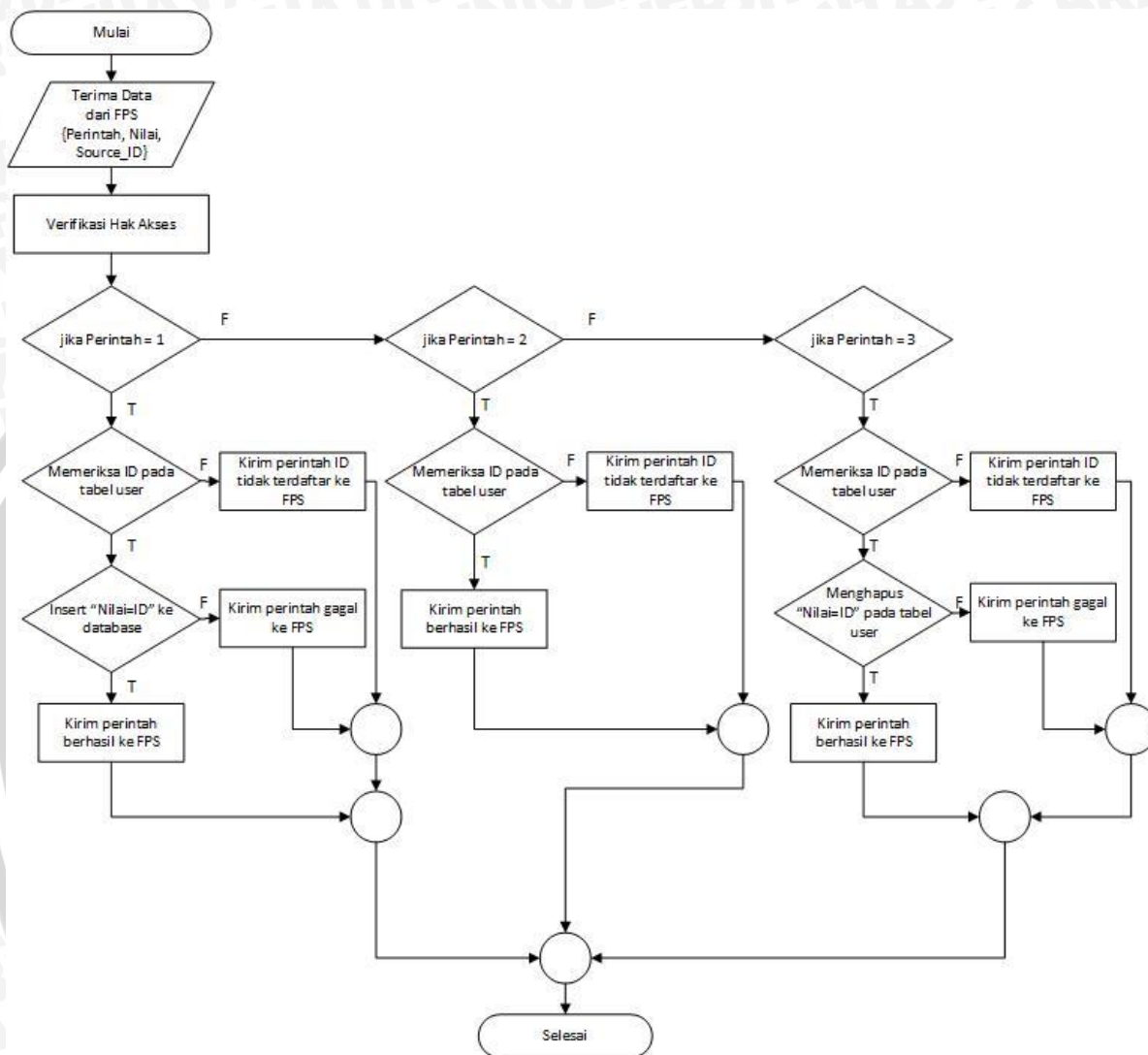


Gambar 4.10 Modul Wireless

#### 4.2.2. Implementasi Perangkat Lunak

Perancangan perangkat lunak pada Arduino IDE menggunakan Bahasa C. Implementasi dari perangkat lunak pada skripsi ini adalah bagaimana sistem tersebut dapat berjalan sesuai dengan harapan yaitu pada masing-masing algoritma dapat berjalan dengan benar. Arduino promini merupakan komponen utama yang digunakan untuk mengatur seluruh kerja sistem. Untuk modul pengiriman data menggunakan nrf24l01 dimana data dari mikrokontroller akan dikirim menuju

server ataupun sebaliknya. Diagram alir program pada *Base Station (Server)* dapat dilihat pada gambar 4.11



Gambar 4.11 Diagram Alir pada *Base Station (Server)*

## BAB V

### PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan pembahasan sistem hasil implementasi dan pengujian dari penelitian ini. Proses pengujian dilakukan melalui dua macam pengujian yaitu pengujian perangkat keras dan pengujian perangkat lunak. Pengujian perangkat keras digunakan untuk menguji system dapat berjalan sesuai harapan serta konsumsi daya listrik dan Pengujian perangkat lunak digunakan untuk menguji algoritma dari *Fingerprint Scanner* yang diterapkan pada perangkat keras.

Proses pembahasan bertujuan untuk mendapatkan kesimpulan dari hasil pengujian implementasi dan pengujian algoritma dari *Fingerprint Scanner* pada perangkat keras Arduino pro mini dalam aplikasi akses kunci pintu otomatis menggunakan sidik jari. Proses pembahasan mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Pembahasan dilakukan terhadap hasil pengujian di setiap tahap pengujian.

#### 5.1 Pengujian Algoritma Enroll Data ke Fingerprint Scanner

Pada bagian ini dilakukan pengujian penambahan *user* baru (*Enroll*) pada modul *Fingerprint Scanner* untuk mengetahui kinerja modul pada beberapa *user* yang berbeda. Pengujian dilakukan pada 21 *user* (jumlah maksimal *user*) dengan kondisi jari kering. Hasil pengujian dapat dilihat pada tabel 5.1 berikut:

Tabel 5.1 Pengujian Algoritma *Enroll* (Penambahan User) Pada Modul  
*Fingerprint Scanner*

| User ke | Scan Sidik Jari | FPS Enroll (Penambahan User) | Waktu (Detik) |
|---------|-----------------|------------------------------|---------------|
| 0       | √               | √                            | 0,919         |
| 1       | √               | √                            | 0,809         |
| 2       | √               | √                            | 0,776         |
| 3       | √               | √                            | 0,729         |
| 4       | √               | √                            | 0,927         |
| 5       | √               | √                            | 0,937         |
| 6       | √               | √                            | 0,806         |

|    |   |   |       |
|----|---|---|-------|
| 7  | √ | √ | 0,914 |
| 8  | √ | √ | 0,816 |
| 9  | √ | √ | 0,930 |
| 10 | √ | √ | 0,935 |
| 11 | √ | √ | 0,957 |
| 12 | √ | √ | 0,988 |
| 13 | √ | √ | 0,972 |
| 14 | √ | √ | 0,993 |
| 15 | √ | √ | 0,964 |
| 16 | √ | √ | 0,982 |
| 17 | √ | √ | 0,934 |
| 18 | √ | √ | 0,986 |
| 19 | √ | √ | 0,923 |
| 20 | √ | √ | 0,978 |

Dari Tabel 5.1 dapat disimpulkan bahwa modul fingerprint dapat mendeteksi sidik jari dengan baik pada 21 *user* yang berbeda dengan waktu rata-rata 0,913 detik

## 5.2 Pengujian Algoritma Verifikasi ID Pada *Fingerprint Scanner*

Pada bagian ini dilakukan pengujian tentang algoritma verifikasi ID *user* pada *Fingerprint Scanner*. Sidik jari akan dilakukan scan dan dicocokkan sesuai dengan data yang ada pada modul. Hasil pengujian dapat dilihat pada tabel 5.2 berikut:

Tabel 5.2 Pengujian Algoritma Verifikasi Pada Modul *Fingerprint Scanner*

| User ke | Scan Sidik Jari | FPS Verifikasi ID User | Waktu (Detik) |
|---------|-----------------|------------------------|---------------|
| 0       | √               | √                      | 0,615         |
| 1       | √               | √                      | 0,630         |
| 2       | √               | √                      | 0,659         |
| 3       | √               | √                      | 0,729         |
| 4       | √               | √                      | 0,676         |
| 5       | √               | √                      | 0,714         |
| 6       | √               | √                      | 0,794         |
| 7       | √               | √                      | 0,792         |
| 8       | √               | √                      | 0,864         |
| 9       | √               | √                      | 0,747         |
| 10      | √               | √                      | 0,784         |
| 11      | √               | √                      | 0,721         |
| 12      | √               | √                      | 0,711         |
| 13      | √               | √                      | 0,856         |

|    |   |   |       |
|----|---|---|-------|
| 14 | √ | √ | 0,834 |
| 15 | √ | √ | 0,945 |
| 16 | √ | √ | 0,857 |
| 17 | √ | √ | 0,756 |
| 18 | √ | √ | 0,845 |
| 19 | √ | √ | 0,823 |
| 20 | √ | √ | 0,812 |

Dari Tabel 5.2 dapat disimpulkan bahwa modul fingerprint dapat melakukan verifikasi sidik jari dengan baik pada 21 *user* yang berbeda. Rata-rata waktu yang dibutuhkan untuk proses verifikasi pada modul *Fingerprint Scanner* adalah 0,72 detik.

### 5.3 Pengujian Algoritma Menghapus ID Dari *Fingerprint Scanner*

Pada bagian ini dilakukan pengujian tentang algoritma menghapus ID *user* dari modul fingerprint. Seperti dijelaskan pada perancangan, yang berhak menghapus harus sudah terdaftar pada modul *Fingerprint Scanner*. Alur dari algoritma adalah:

- 1) Pengguna yang sudah terdaftar menekan tombol hapus pada *Keypad device remote*.
- 2) Pengguna melakukan verifikasi sidik jari.
- 3) Jika sudah terverifikasi, kemudian memasukan ID yang akan dihapus dengan menggunakan *Keypad* pada device remote.
- 4) Proses hapus ID selesai.

Hasil pengujian dapat dilihat pada tabel 5.3 berikut:

Tabel 5.3 Pengujian Algoritma Menghapus ID dari Modul *Fingerprint Scanner*

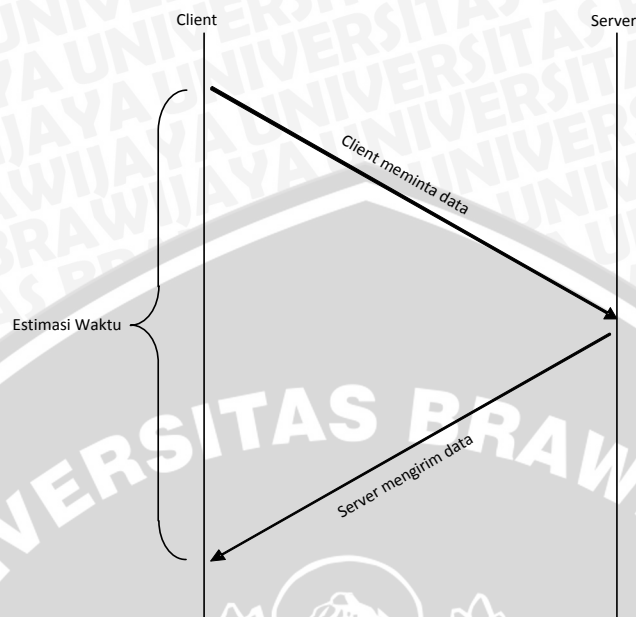
| User Ke | Scan Sidik Jari | FPS Menghapus ID User | Waktu (Detik) |
|---------|-----------------|-----------------------|---------------|
| 0       | √               | √                     | 0,144         |
| 1       | √               | √                     | 0,145         |
| 2       | √               | √                     | 0,144         |
| 3       | √               | √                     | 0,144         |
| 4       | √               | √                     | 0,145         |
| 5       | √               | √                     | 0,145         |
| 6       | √               | √                     | 0,145         |
| 7       | √               | √                     | 0,144         |
| 8       | √               | √                     | 0,144         |

|    |   |   |       |
|----|---|---|-------|
| 9  | √ | √ | 0,146 |
| 10 | √ | √ | 0,144 |
| 11 | √ | √ | 0,144 |
| 12 | √ | √ | 0,144 |
| 13 | √ | √ | 0,146 |
| 14 | √ | √ | 0,146 |
| 15 | √ | √ | 0,147 |
| 16 | √ | √ | 0,144 |
| 17 | √ | √ | 0,145 |
| 18 | √ | √ | 0,145 |
| 19 | √ | √ | 0,144 |
| 20 | √ | √ | 0,145 |

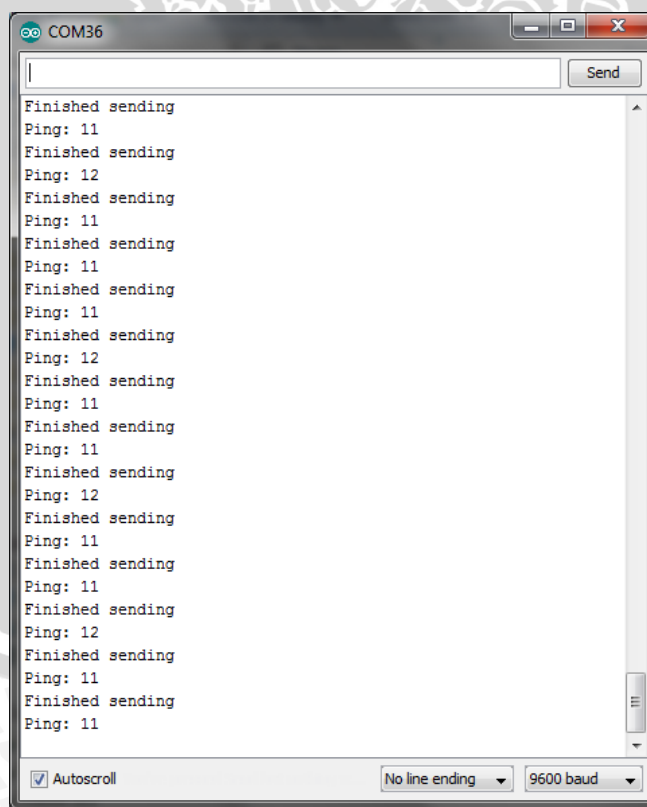
Dari Tabel 5.3 dapat disimpulkan bahwa modul fingerprint dapat melakukan proses menghapus ID dengan benar dengan waktu rata-rata 0,144 detik.

#### 5.4 Pengujian Komunikasi Data Dengan NRF24L01

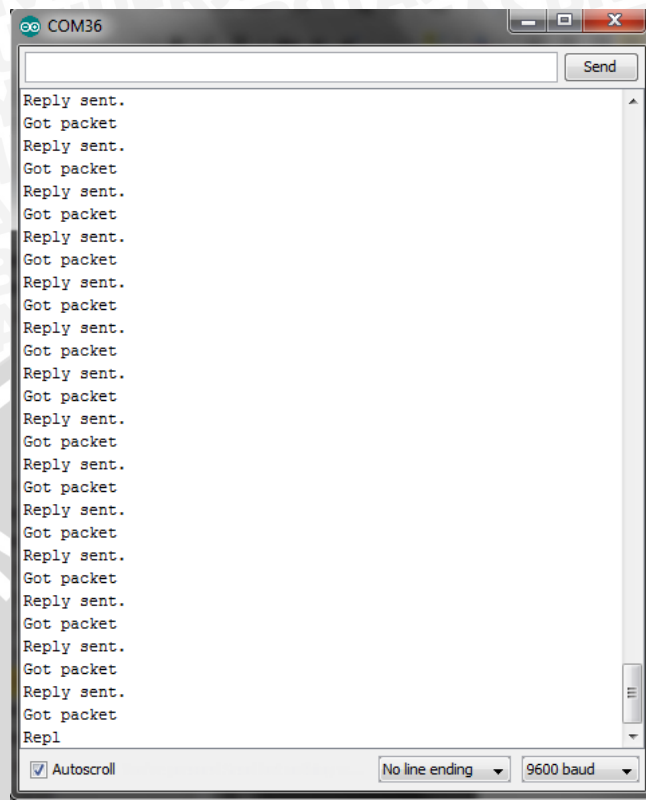
Pengujian Komunikasi pada Modul NRF24L01 menggunakan 2 buah arduino yang masing – masing terdapat modul NRF24L01. Pengujian ini bertujuan untuk mengetahui waktu pengiriman data antar modul. Pengujian ini dilakukan dengan cara modul 1 ( Gambar 5.5 client ) mengirimkan permintaan data ke modul 2 ( Gambar 5.4 server ), selanjutnya modul 2 mengirimkan data ke modul 1, dari pengiriman data dari kedua modul tersebut bisa diambil selisih waktunya, waktu inilah yang menjadi acuan seberapa lama modul ini berkomunikasi, untuk lebih jelasnya dapat dilihat diagram alir dari proses tersebut pada Gambar 5.4 dan hasil dari pengujian ini dapat dilihat pada Gambar 5.5 pada sisi client.



Gambar 5.4 Diagram alir komunikasi



Gambar 5.5 Modul pada client



**Gambar 5.6** Modul pada server

## 5.5 Pengujian Algoritma *Enroll Data ke Base Station (Server)*

Pada bagian ini dilakukan pengujian penambahan *user* baru (*Enroll*) pada *Base Station (Server)*. Pengujian dilakukan pada 10 *user* dengan kondisi jari kering. Hasil pengujian dapat dilihat pada tabel 5.7 berikut:

**Tabel 5.7** Pengujian Algoritma *Enroll* Pada *Base Station (Server)*

| User ke | Scan Sidik Jari | Enroll ID User Server | Waktu (Detik) |
|---------|-----------------|-----------------------|---------------|
| 0       | √               | √                     | 1,268         |
| 1       | √               | √                     | 1,172         |
| 2       | √               | √                     | 1,146         |
| 3       | √               | √                     | 1,058         |
| 4       | √               | √                     | 1,074         |
| 5       | √               | √                     | 1,068         |
| 6       | √               | √                     | 1,072         |
| 7       | √               | √                     | 1,008         |
| 8       | √               | √                     | 1,103         |
| 9       | √               | √                     | 1,085         |
| 10      | √               | √                     | 1,155         |

|    |   |   |       |
|----|---|---|-------|
| 11 | √ | √ | 1,177 |
| 12 | √ | √ | 1,090 |
| 13 | √ | √ | 1,082 |
| 14 | √ | √ | 1,095 |
| 15 | √ | √ | 1,174 |
| 16 | √ | √ | 1,088 |
| 17 | √ | √ | 1,139 |
| 18 | √ | √ | 1,089 |
| 19 | √ | √ | 1,043 |
| 20 | √ | √ | 1,088 |

Hasil *enroll* pada *Base Station (Server)* dapat dilihat di tabel user database seperti gambar 5.1 berikut:

| USR_ID | USR_NAME | USR_PASSWORD | USR_LEVEL |
|--------|----------|--------------|-----------|
| 1      | guest    | guest        | D01       |
| 2      | guest    | guest        | D01       |
| 0      | guest    | guest        | D01       |
| 3      | guest    | guest        | D01       |
| 4      | guest    | guest        | D01       |
| 5      | guest    | guest        | D01       |
| 6      | guest    | guest        | D02       |
| 7      | guest    | guest        | D02       |
| 8      | guest    | guest        | D02       |
| 9      | guest    | guest        | D02       |
| 10     | guest    | guest        | D02       |

Gambar 5.1 User Database *Enroll*

Dari Tabel 5.7 dan Gambar 5.1 dapat disimpulkan bahwa algoritma *enroll* pada *base station (Server)* berjalan dengan baik dengan waktu rata-rata 1,057 detik

## 5.6 Pengujian Algoritma Verifikasi Data ke *Base Station (Server)*

Pada bagian ini dilakukan pengujian algoritma verifikasi data ID dan *record* akses pada *Base Station (Server)*. Pengujian dilakukan pada 21 *user* dengan kondisi jari kering. Hasil pengujian dapat dilihat pada tabel 5.8 berikut:

Tabel 5.8 Pengujian Algoritma Verifikasi Data ke *Base Station (Server)*

| User ke | Scan Sidik Jari | Verifikasi ID User Server | Waktu (Detik) |
|---------|-----------------|---------------------------|---------------|
| 0       | √               | √                         | 0,702         |
| 1       | √               | √                         | 0,657         |

|    |   |   |       |
|----|---|---|-------|
| 2  | √ | √ | 0,648 |
| 3  | √ | √ | 0,730 |
| 4  | √ | √ | 0,709 |
| 5  | √ | √ | 0,720 |
| 6  | √ | √ | 0,803 |
| 7  | √ | √ | 0,798 |
| 8  | √ | √ | 0,870 |
| 9  | √ | √ | 0,770 |
| 10 | √ | √ | 0,789 |
| 11 | √ | √ | 0,726 |
| 12 | √ | √ | 0,730 |
| 13 | √ | √ | 0,840 |
| 14 | √ | √ | 0,846 |
| 15 | √ | √ | 0,955 |
| 16 | √ | √ | 0,860 |
| 17 | √ | √ | 0,867 |
| 18 | √ | √ | 0,870 |
| 19 | √ | √ | 0,856 |
| 20 | √ | √ | 0,848 |

Hasil verifikasi pada *Base Station (Server)* dapat dilihat di tabel *log door database* seperti gambar 5.2 berikut:

| LOG_ID | DOBJ_OBJECT_ID | ID_USR | LOG_TIME              | LOG_STATUS |
|--------|----------------|--------|-----------------------|------------|
| 1      | D01            |        | 1 2014-10-19 19:59:50 | 1          |
| 2      | D02            |        | 2 2015-01-02 15:19:12 | 1          |
| 5      | D03            |        | 3 2015-04-10 17:07:22 | 1          |
| 4      | D04            |        | 4 2015-02-24 17:20:20 | 1          |
| 6      | D03            |        | 3 2015-04-10 17:09:00 | 1          |
| 7      | admin          |        | 2 2015-04-10 17:12:26 | 1          |
| 8      | admin          |        | 1 2015-04-18 20:30:09 | 1          |
| 9      | admin          |        | 1 2015-04-18 20:32:19 | 1          |
| 10     | admin          |        | 1 2015-04-18 20:42:23 | 1          |
| 227    | admin          |        | 1 2015-05-05 07:54:04 | 1          |
| 228    | admin          |        | 2 2015-05-05 07:54:08 | 1          |
| 229    | D01            |        | 3 2015-05-05 07:54:12 | 1          |
| 230    | D01            |        | 4 2015-05-05 07:54:15 | 1          |
| 231    | D01            |        | 5 2015-05-05 07:54:23 | 1          |

Gambar 5.2 log door Database

Dari Tabel 5.8 dan Gambar 5.2 diatas dapat disimpulkan bahwa algoritma verifikasi pada *base station (Server)* berjalan dengan baik dengan waktu rata-rata 0,79 detik

### 5.7 Pengujian Algoritma Menghapus Data di *Base Station (Server)*

Pada bagian ini dilakukan pengujian tentang algoritma menghapus ID *user* dari modul fingerprint. Seperti dijelaskan pada perancangan, yang berhak menghapus harus sudah terdaftar pada modul *Fingerprint Scanner*. Alur dari algoritma adalah:

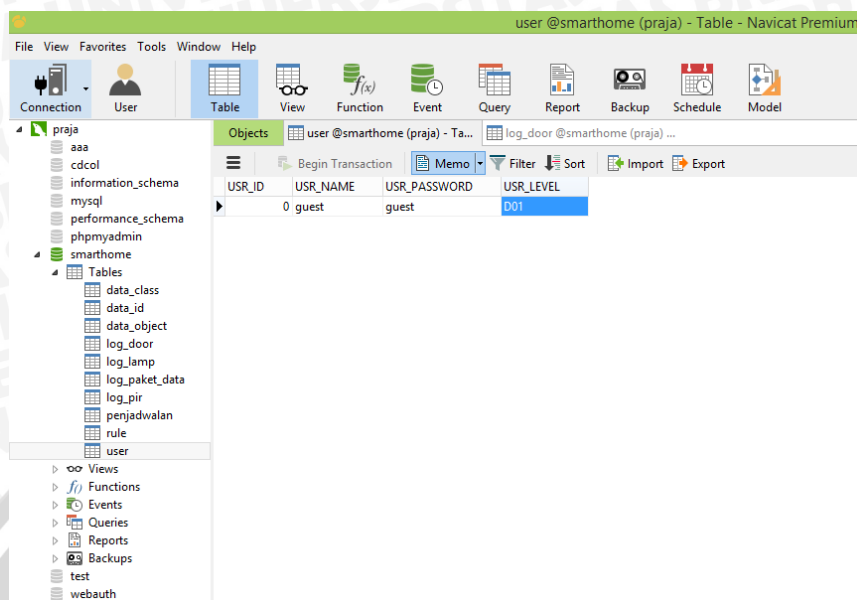
- 1) Pengguna yang sudah terdaftar menekan tombol hapus pada *Keypad* device remote.
- 2) Pengguna melakukan verifikasi sidik jari.
- 3) Jika sudah terverifikasi pada server, kemudian memasukan ID yang akan dihapus dengan menggunakan *Keypad* pada device remote.
- 4) Proses hapus ID selesai.

Hasil pengujian dapat dilihat pada tabel 5.9 berikut:

Tabel 5.9 Pengujian Algoritma Menghapus Data di *Base Station (Server)*

| User Ke | Scan Sidik Jari | Menghapus ID User Server | Waktu (Detik) |
|---------|-----------------|--------------------------|---------------|
| 0       | √               | √                        | 1,115         |
| 1       | √               | √                        | 1,095         |
| 2       | √               | √                        | 1,094         |
| 3       | √               | √                        | 1,044         |
| 4       | √               | √                        | 1,103         |
| 5       | √               | √                        | 1,087         |
| 6       | √               | √                        | 1,102         |
| 7       | √               | √                        | 1,092         |
| 8       | √               | √                        | 1,078         |
| 9       | √               | √                        | 1,102         |
| 10      | √               | √                        | 1,079         |
| 11      | √               | √                        | 1,105         |
| 12      | √               | √                        | 1,112         |
| 13      | √               | √                        | 1,056         |
| 14      | √               | √                        | 1,125         |
| 15      | √               | √                        | 1,114         |
| 16      | √               | √                        | 1,067         |
| 17      | √               | √                        | 1,087         |
| 18      | √               | √                        | 1,056         |
| 19      | √               | √                        | 1,256         |
| 20      | √               | √                        | 1,067         |

Hasil *enroll* pada *Base Station (Server)* dapat dilihat di tabel *user database* seperti gambar 5.3 berikut:



Gambar 5.3 User Database Hapus

Dari Tabel 5.9 dan Gambar 5.3 diatas dapat disimpulkan bahwa algoritma menghapus data pada *base sation (Server)* berjalan dengan baik dan waktu rata-rata 1,096 detik.

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil dari pengujian, fungsionalitas dari perancangan software berjalan sesuai dengan perancangan. Pada perancangan software terdapat 3 fungsi yang digunakan, yaitu *enroll*, verifikasi dan *delete*. Masing-masing dari ketiga fungsi tersebut dilakukan sebanyak 2 kali, yaitu pada modul *fingerprint* dan *base station (server)*.
2. Berdasarkan hasil dari pengujian, waktu rata-rata yang dibutuhkan untuk proses *enroll* pada *base station (server)* adalah 1,057 detik, verifikasi pada *base station (server)* adalah 0,79 detik dan *delete* pada *base station (server)* adalah 1,096 detik. Pada proses verifikasi di server akan dilakukan *record* berupa *ID user*, tanggal dan jam akses untuk dapat mengetahui *history* penggunaan pintu.

#### 6.2 Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain:

1. Untuk pengembangan lebih lanjut, pada sistem ini bisa ditambahkan *limit switch* yang berfungsi sebagai sensor untuk mendeteksi keadaan dari pintu. Jika sensor mendeteksi pintu tertutup maka sistem akan mengunci kembali pintu tersebut.
2. Untuk pengembangan lebih lanjut, pada sistem ini dapat ditambahkan proses *record* untuk *user* yang tidak terdaftar. Jika user tersebut melakukan verifikasi akan direcord pada server sebagai *unknow user*.

## DAFTAR PUSTAKA

- [1] Faundez-Zanuy, Marcos. 2004, "A Door-Opening System Using A Low-Cost *Fingerprint Scanner* and a PC". Escola Universitaria Politcnica de Matarb. IEEE A&E SYSTEMS MAGAZINE, AUGUST 2004
- [2] Asritirtany, Catharina Hilda,. dan Nurul Hidayat. 2012, "Aplikasi Sistem Pengenalan Individu Berbasis Sidik Jari pada Pengembangan Portal Otomatis". Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember Surabaya.
- [3] Widyananda, Anggidhira. 2013 "Perancangan Dan Implementasi Kunci Pintu Otomatis Dengan Sidik Jari Berbasis Mikrokontroler". D3 Teknik Telekomunikasi – Institut Teknologi Telkom
- [4] Sucipta, I Kadek Dwi. 2011, "Pembuatan Sistem Monitoring Pengguna Laboratorium Komputer Berbasis Sidik Jari". Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember Surabaya

## LAMPIRAN

### Lampiran 1: Program Secara Keseluruhan pada node *FingerPrint Scanner*.

```
1. #include <Keypad.h>
2. #include <FPS_GT511C3.h>
3. #include <SoftwareSerial.h>
4. #include <LiquidCrystal.h>
5. #include <SPI.h> //library spi
6. #include <Mirf.h> //library komunikasi untuk nrf24l01
7. #include <nRF24L01.h> //library komunikasi untuk nrf24l01
8. #include <MirfHardwareSpiDriver.h> //library komunikasi spi
9.
10. //Deklarasi Variabel
11. byte Header=0;
12. byte Status=0;
13. byte RW=0;
14. byte From_ID=0;
15. byte To_ID=0;
16. byte Jam=0;
17. byte Menit=0;
18. byte Detik=0;
19. byte Nilai=0;
20. byte Checksum=0;
21. byte General_Purpose=0;
22.
23. byte alamat=5; //alamat node ini
24.
25. byte out[11];
26. byte in[11];
27.
28. int enrollid, id;
29. unsigned long start, start_enroll, start_deleted, finished,
30.     finished_enroll, finished_deleted, elapsed,
31.     elapsed_enroll, elapsed_deleted;
32. String Variabel_Perintah;
33. byte Perintah;
34.
35.
```

```
36. FPS_GT511C3 fps(A0, A1);
37. LiquidCrystal lcd(4, 5, 6, 7, 8, 9);
38. int Pintu=3;
39.
40. boolean bDeleting = false;
41.
42. void setup() {
43.     Serial.begin(9600);
44.
45.     pinMode(Pintu, OUTPUT);
46.     digitalWrite(Pintu, LOW);
47.     lcd.begin(16, 2);
48.     lcd.setCursor(0, 1);
49.
50.     Mirf.cePin = A5;
51.     Mirf.csnPin = 10;
52.     Mirf.spi = &MirfHardwareSpi;
53.     Mirf.init();
54.
55.     Mirf.setTADDR((byte *)"serv1");
56.     Mirf.setRADDR((byte *)"serv1");
57.     Mirf.payload = 11;
58.     Mirf.channel = 100;
59.
60.     Mirf.config();
61.
62.     Header='A';
63.     Status='B';
64.     RW='W';
65.     From_ID=0;
66.     To_ID=0;
67.     Jam=0;
68.     Menit=0;
69.     Detik=0;
70.     Nilai=0;
71.     General_Purpose=0;
72. }
```

```
73. void loop() {
74.     if(!bDeleting)
75.     {
76.         idfinger();
77.     }
78.     delay(1000);
79. }
80.
81. void serialEvent() {
82.     while (Serial.available()) {
83.         char terima = (char)Serial.read();
84.         Variabel_Perintah += terima;
85.
86.         lcd.clear();
87.         lcd.setCursor(0, 1);
88.         lcd.print(Variabel_Perintah);
89.
90.         if (terima == '*'){
91.             Perintah = Variabel_Perintah[0];
92.             if (Perintah == 'A')
93.             {
94.                 lcd.clear();
95.                 lcd.setCursor(0, 1);
96.                 lcd.print("Enroll");
97.                 delay(1000);
98.                 idfinger();
99.             }
100.            else if (Perintah == 'B')
101.            {
102.                lcd.clear();
103.                lcd.setCursor(0, 1);
104.                lcd.print("Deleted");
105.                delay(1000);
106.                idfinger();
107.            }
108.            Variabel_Perintah="";
109.        }
```

```
110.         if(terima == '#'){
111.             uint8_t ID = getID(Variabel_Perintah);
112.             deleteID(ID);
113.             Variabel_Perintah="";
114.         }
115.     }
116. }
117.
118. uint8_t getID(String sData)
119. {
120.     uint8_t hasil;
121.     if (sData.length() == 3)
122.     {
123.         uint8_t puluhan = (sData[0] - 48) * 10;
124.         uint8_t satuan = sData[1] - 48;
125.         hasil = puluhan + satuan;
126.     }
127.     else
128.     {
129.         hasil = sData[0] - 48;
130.     }
131.     return hasil;
132. }
133.
134. void idfinger()
135. {
136.     fps.Open();
137.     fps.SetLED(true);
138.     lcd.clear();
139.     lcd.setCursor(0, 0);
140.     lcd.print("Press Finger");
141.     if (fps.IsPressFinger())
142.     {
143.         lcd.clear();
144.         lcd.print("Please Wait");
145.         fps.CaptureFinger(false);
146.     }
```

```
147.         start=millis();
148.         id = fps.Identify1_N();
149.         if (id <200)
150.         {
151.             lcd.clear();
152.             lcd.print(id);
153.             To_ID=255;
154.             From_ID=alamat;
155.             RW='W';
156.             Nilai=id;
157.             General_Purpose=2;
158.             assamble_frame();
159.             kirim();
160.
161.             if(TungguData())
162.             {
163.                 terima();
164.             }
165.             else
166.             {
167.                 lcd.clear();
168.                 lcd.setCursor(0, 0);
169.                 lcd.print("GAGAL!");
170.                 lcd.setCursor(0, 1);
171.                 lcd.print(elapsed);
172.                 lcd.print("milisecond");
173.             }
174.         }
175.         else {
176.             lcd.clear();
177.             lcd.setCursor(0, 0);
178.             lcd.print("Finger Not");
179.             lcd.setCursor(0, 1);
180.             lcd.print("Found");
181.         }
182.     }
183. }
```

```
184. void Enroll()  
185. {  
186.     fps.Open();  
187.     fps.SetLED(true);  
188.     lcd.clear();  
189.     lcd.print("Enroll");  
190.     enrollid = 0;  
191.     bool usedid = true;  
192.     while (usedid == true)  
193.     {  
194.         usedid = fps.CheckEnrolled(enrollid);  
195.         if (usedid==true) enrollid++;  
196.     }  
197.     fps.EnrollStart(enrollid);  
198.     lcd.clear();  
199.     lcd.print("Press Finger ");  
200.     lcd.print(enrollid);  
201.     delay(2000);  
202.     while(fps.IsPressFinger() == false) delay(1000);  
203.     bool bret = fps.CaptureFinger(true);  
204.     int iret = 0;  
205.     if (bret != false)  
206.     {  
207.         lcd.clear();  
208.         lcd.print("Remove Finger");  
209.         fps.Enroll11();  
210.         while(fps.IsPressFinger() == true) delay(1000);  
211.         lcd.clear();  
212.         lcd.setCursor(0, 0);  
213.         lcd.print("Press Finger");  
214.         lcd.setCursor(0, 1);  
215.         lcd.print("Again");  
216.         while(fps.IsPressFinger() == false) delay(1000);  
217.         bret = fps.CaptureFinger(true);  
218.         if (bret != false)  
219.         {  
220.             lcd.clear();
```

```
221. lcd.print("Remove Finger");
222. fps.Enroll2();
223. while(fps.IsPressFinger() == true);
224. delay(1000);
225. lcd.clear();
226. lcd.setCursor(0, 0);
227. lcd.print("Press Finger");
228. lcd.setCursor(0, 1);
229. lcd.print("Again");
230. while(fps.IsPressFinger() == false);
231. delay(1000);
232. bret = fps.CaptureFinger(true);
233. start_enroll=millis();
234. if (bret != false)
235. {
236.     lcd.clear();
237.     lcd.print("Press Finger");
238.
239.     iret = fps.Enroll3();
240.     if (iret == 0)
241.     {
242.         lcd.clear();
243.         lcd.setCursor(0, 0);
244.         lcd.print("Enroll Success");
245.         To_ID=255;
246.         From_ID=alamat;
247.         RW='W';
248.         Nilai=enrollid;
249.         General_Purpose=1;
250.         assamble_frame();
251.         kirim();
252.
253.         if(TungguData())
254.         {
255.             terima();
256.             finished_enroll=millis();
257.
```

```
258.         elapsed_enroll=finished_enroll-
259.             start_enroll;
260.         lcd.clear();
261.         lcd.setCursor(0, 0);
262.         lcd.print("Enroll Done");
263.         lcd.setCursor(0, 1);
264.         lcd.print(elapsed_enroll);
265.         lcd.print("milisecond");
266.         delay(1000);
267.     }
268.     else
269.     {
270.         lcd.clear();
271.         lcd.setCursor(0, 1);
272.         lcd.print("GAGAL!");
273.     }
274.     delay(1000);
275. }
276. else
277. {
278.     lcd.clear();
279.     lcd.print("Enroll Failed");
280. }
281. }
282. else
283. {
284.     lcd.clear();
285.     lcd.print("Enroll Failed");
286. }
287. }
288. else
289. {
290.     lcd.clear();
291.     lcd.print("Enroll Failed");
292. }
293. }
294.
```

```
295.     else
296.     {
297.         lcd.clear();
298.         lcd.print("Failed Captured");
299.     }
300. }
301.
302. void deleteID(uint8_t ID)
303. {
304.     fps.Open();
305.     fps.SetLED(true);
306.     start_deleted=millis();
307.     fps.DeleteID(ID);
308.
309.     To_ID=255;
310.     From_ID=alamat;
311.     RW='W';
312.     Nilai=ID;
313.     General_Purpose=3;
314.     assamble_frame();
315.     kirim();
316.
317.     if(TungguData())
318.     {
319.         terima();
320.         finished_deleted=millis();
321.         elapsed_deleted=finished_deleted-start_deleted;
322.         lcd.clear();
323.         lcd.setCursor(0, 0);
324.         lcd.print("delete done");
325.         lcd.setCursor(0, 1);
326.         lcd.print(elapsed_deleted);
327.         lcd.print("milisecond");
328.         delay(1000);
329.     }
330.     else
331.     {
```

```
331.         lcd.clear();
332.         lcd.print("GAGAL!");
333.         lcd.clear();
334.         lcd.setCursor(0, 1);
335.         lcd.print("milisecond");
336.     }
337. }
338.
339. void assamble_frame()
340. {
341.     out[0]=Header;
342.     out[1]=Status;
343.     out[2]=RW;
344.     out[3]=From_ID;
345.     out[4]=To_ID;
346.     out[5]=Jam;
347.     out[6]=Menit;
348.     out[7]=Detik;
349.     out[8]=Nilai;
350.     out[9]=General_Purpose;
351.     out[10]=out[0]^out[1]^out[2]^out[3]^out[4]^out[5]^
352.             out[6]^out[7]^out[8]^out[9];
353. }
354.
355. void kirim()
356. {
357.     Mirf.send((byte *)&out);
358.     while(Mirf.isSending())
359.     {}
360. }
361.
362. boolean TungguData()
363. {
364.     boolean bStatus;
365.     unsigned long time1 = millis();
366.     lcd.clear();
367.     lcd.print("Berhasil Kirim");
```

```
368.     bStatus = true;
369.     while(!Mirf.dataReady())
370.     {
371.         if((millis()-time1) > 2000)
372.         {
373.             lcd.clear();
374.             lcd.print("RTO");
375.             bStatus = false;
376.             break;
377.         }
378.     }
379.     return bStatus;
380. }
381.
382. void terima()
383. {
384.     Mirf.getData((byte *)&in);
385.     if(in[4]==alamat&&in[2]=='R'&&in[8]==100&&in[9]==1)
386.     {
387.         lcd.clear();
388.         lcd.print("Berhasil Enroll");
389.     }
390.     else if(in[4]==alamat&&in[2]=='R'&&in[8]==0&&in[9]==1)
391.     {
392.         lcd.clear();
393.         lcd.print("Gagal Enroll");
394.     }
395.     else if(in[4]==alamat&&in[2]=='R'&&in[8]==100&&in[9]==2)
396.     {
397.         finished=millis();
398.         elapsed=finished-start;
399.         lcd.clear();
400.         lcd.setCursor(0, 0);
401.         lcd.print("Verifikasi");
402.         lcd.setCursor(0, 1);
403.         lcd.print(elapsed);
404.         lcd.print("milisecond");
```

```
405.    digitalWrite(Pintu, HIGH);
406.    delay(2000);
407.    digitalWrite(Pintu, LOW);
408.    if (Perintah == 'A')
409.    {
410.        lcd.clear();
411.        lcd.setCursor(0, 0);
412.        lcd.print("Masuk Enroll");
413.        delay(1000);
414.        Enroll();
415.        Variabel_Perintah="";
416.    }
417.    else if (Perintah == 'B')
418.    {
419.        lcd.clear();
420.        lcd.setCursor(0, 0);
421.        lcd.print("Masuk Deleted");
422.        delay(1000);
423.        bDeleting = true;
424.        lcd.clear();
425.        lcd.setCursor(0, 0);
426.        lcd.print("Masukkan ID");
427.        while(!Serial.available())
428.        {}
429.        Variabel_Perintah="";
430.    }
431.    }
432.    else if(in[4]==alamat&&in[2]=='R'&&in[8]==0&&in[9]==2)
433.    {
434.        lcd.clear();
435.        lcd.print("Gagal");
436.    }
437.    else if(in[4]==alamat&&in[2]=='R'&&in[8]==100&&in[9]==3)
438.    {
439.        lcd.clear();
440.        lcd.setCursor(0, 0);
441.        lcd.print("Berhasil dihapus");
```

```
442.     delay(1000);  
443.     bDeleting = false;  
444. }  
445. else if(in[4]==alamat&&in[2]=='R'&&in[8]==0&&in[9]==3)  
446. {  
447.     lcd.clear();  
448.     lcd.print("Gagal Hapus");  
449. }  
450. }
```



**Lampiran 2: Program Secara Keseluruhan pada node Base Station (Server).**

```
1. #include <SPI.h> //library spi
2. #include <Mirf.h> //library komunikasi untuk nrf24l01
3. #include <nRF24L01.h> //library komunikasi untuk nrf24l01
4. #include <MirfHardwareSpiDriver.h> //library komunikasi spi
5. #include "SoftwareSerial.h"
6.
7.
8.
9. //Deklarasi Variabel
10. byte Header=0;
11. byte Status=0;
12. byte RW=0;
13. byte From_ID=0;
14. byte To_ID=0;
15. byte Jam=0;
16. byte Menit=0;
17. byte Detik=0;
18. byte Nilai=0;
19. byte Checksum=0;
20. byte General_Purpose=0;
21.
22. byte alamat=255; //alamat node ini
23.
24.
25. byte out[11];
26. byte in[11];
27. char cetak[500];
28. String data, firstValue, secondValue;
29. byte bAlamatTujuan, bNilai, bPerintah;
30.
31.
32. void setup()
33. {
34.     Serial.begin(9600);
35.
36.     Mirf.spi = &MirfHardwareSpi;
37.     Mirf.init();
38.
39.     Mirf.setTADDR((byte *)"serv1");
40.     Mirf.setRADDR((byte *)"serv1");
```

```
48.   Mirf.payload = 11;
49.   Mirf.channel = 100;
50.
51.
52.   Mirf.config();
53.
54.   pinMode(9,OUTPUT);
55.   digitalWrite(9,HIGH);
56.
57.
58.   Header='A';
59.   Status='B';
60.
61.   RW='W';
62.   From_ID=1;
63.   To_ID=2;
64.   Jam=3;
65.   Menit=4;
66.   Detik=5;
67.   Nilai=0;
68.   General_Purpose=7;
69.
70.   }
71.
72.
73.
74. void loop()
75. {
76.
77.   while (Serial.available())
78.   {
79.
80.     char terima = (char)Serial.read();
81.     data += terima;
82.
83.     if (terima == '}')
84.     {
85.
86.       bPerintah = data[1];
87.       bNilai = data[2];
88.       bAlamatTujuan = data[3];
89.
90.
91.       General_Purpose = bPerintah;
92.       To_ID=bAlamatTujuan;
93.       From_ID=alamat;
94.       RW='R';
95.
96.
```

```

97.         Nilai=bNilai;
98.         assamble_frame();
99.         kirim();
100.        data = "";
101.
102.    }
103.
104. }
105.     terima();
106. }
107.
108. void assamble_frame()
109. {
110.     out[0]=Header;
111.     out[1]=Status;
112.     out[2]=RW;
113.     out[3]=From_ID;
114.     out[4]=To_ID;
115.     out[5]=Jam;
116.     out[6]=Menit;
117.     out[7]=Detik;
118.     out[8]=Nilai;
119.     out[9]=General_Purpose;
120.     out[10]=out[0]^out[1]^out[2]^out[3]^out[4]^out[5]^
121.             out[6]^out[7]^out[8]^out[9];
122.
123. }
124.
125. void kirim()
126. {
127.     Mirf.send((byte *)&out);
128.     while(Mirf.isSending())
129.     {}
130. }
131.
132. void terima()
133. {
134.     if(!Mirf.isSending() && Mirf.dataReady())
135.     {
136.         Mirf.getData((byte *)&in);
137.

```

```
146.         if(in[4]==alamat&&in[2]=='W')
147.         {
148.             Header=in[0];
149.             Status=in[1];
150.             RW=in[2];
151.             From_ID=in[3];
152.             To_ID=in[4];
153.             Jam=in[5];
154.             Menit=in[6];
155.             Detik=in[7];
156.             Nilai=in[8];
157.             if(From_ID != 254)
158.             {
159.                 General_Purpose=in[9];
160.                 Serial.print('{');
161.                 Serial.write(General_Purpose);
162.                 Serial.write(Nilai);
163.                 Serial.write(From_ID);
164.                 Serial.print('}');
165.             }
166.         }
167.     }
168. }
```

**Lampiran 3: Program Secara Keseluruhan pada node Remote Keypad**

```
1. #include <Keypad.h>
2. #include <SPI.h> //library spi
3. #include <Mirf.h> //library komunikasi untuk nrf24l01
4. #include <nRF24L01.h> //library komunikasi untuk nrf24l01
5. #include <MirfHardwareSpiDriver.h> //library komunikasi spi
6. #include <SoftwareSerial.h>
7.
8.
9.
10. const byte ROWS = 4; //four rows
11. const byte COLS = 4; //three columns
12. char keys[ROWS][COLS] = {
13.     {'1','2','3','A'},
14.     {'4','5','6','B'},
15.     {'7','8','9','C'},
16.     {'*','0','#','D'}
17. };
18.
19. };
20.
21. byte rowPins[ROWS] = {2, 3, 4, 5}; //row pinouts keypad
22. byte colPins[COLS] = {6, 7, 8, 9}; //column pinouts keypad
23. Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins,
24. ROWS, COLS );
25.
26.
27. void setup()
28. {
29.     Serial.begin(9600);
30. }
31.
32.
33. void loop()
34. {
35.     char key = keypad.getKey();
36.     if(key)
37.     {
38.         Serial.write(key);
39.     }
40. }
```