

# BAB I

## PENDAHULUAN

### A. Latar Belakang

Kendaraan bermotor saat ini merupakan salah satu kebutuhan yang harus dipenuhi. Tanpa sadar manusia pada masa sekarang ini sangat bergantung pada kendaraan bermotor baik itu roda dua maupun roda empat. Fakta tersebut dapat tercermin pada data yang dikeluarkan oleh Badan Pusat Statistik (2019) jumlah kendaraan bermotor di Indonesia terus bertambah dari 129.281.079 unit pada 2016 menjadi 138.556.669 unit pada tahun 2017.

Kasus pencurian khususnya sepeda motor di Indonesia dari tahun ke tahun mengalami penambahan yang sangat signifikan. Hal ini dikarenakan teknologi keamanan yang digunakan sepeda motor masih analog atau manual. Teknologi keamanan sepeda motor dapat di ganti menggunakan teknologi digital sehingga dapat meningkatkan keamanan dan mengurangi kasus pencurian. Penggunaan teknologi digital sudah banyak diterapkan dalam kehidupan sehari-hari contoh pembayaran, sistem kendali, sistem monitoring, dan lain-lain.

Alat pengaman kendaraan roda dua elektronik (sistem *immobilizer*, *keyless*, dan remot alarm). Sistem *immobilizer*, *keyless*, dan remot alarm mempunyai prinsip yang sama dengan sistem *star-stop engine* yang ada pada mobil. Semua sistem tersebut menggunakan gelombang *transmitter* sama seperti RF-IDE. Semua sistem tersebut

mempunyai kelemahan yang sama yaitu belum adanya sistem pengaman pada remot atau kunci kendaraan. Sehingga, pada saat kunci atau remot kendaraan hilang motorpun dapat dengan mudah dihidupkan orang lain (Dharmawan 2018).

Dari latar belakang di atas terdapat sebuah potensi yaitu alat pengaman kendaraan roda dua pada sistem pengapian berbasis sidik jari. Sistem pengapian merupakan sistem pada sepeda motor yang berfungsi mengatur proses pembakaran pada silinder (Yunianto, 2009:2). Sistem pengapian merupakan sistem utama sepeda motor. Tanpa adanya sistem pengapian, mesin sepeda motor tidak akan dapat dihidupkan. Pernyataan tersebut selaras dengan teori segitiga api dimana pembakaran mesin hanya dapat dilakukan jika ada tiga unsur sebagai berikut: bahan bakar, api, dan udara. Dengan mengontrol sistem pengapian guna mengamankan sepeda motor akan sangat efektif menanggulangi pencurian sepeda motor.

Sidik jari merupakan suatu pola garis yang ada pada jari manusia yang mempunyai bentuk berbeda-beda dan sering digunakan dalam teknologi biometrik (Gazali dan Gunawan, 2012:58). Dengan menggunakan sidik jari sebagai kunci sepeda motor dapat dipastikan bahwa sepeda motor hanya dapat digunakan pada orang-orang tertentu yang telah didaftarkan. Sehingga, kendaraan bermotor akan aman dari tindak pencurian. Namun, pada sistem ini terdapat sebuah kekurangan yaitu: sistem ini belum mampu mengidentifikasi umur manusia. Dari kekurangan dan kelebihan kedua sistem di atas, untuk menciptakan sebuah sistem yang mampu mengamankan kendaraan dan mampu mencegah pengendara kendaraan bermotor dengan usia kurang dari 17 tahun.

### **B. Rumusan Masalah**

1. Bagaimana desain alat pengaman kendaraan roda dua pada sistem kelistrikan berbasis sidik jari?
2. Bagaimana kelayakan alat pengaman kendaraan roda dua pada sistem kelistrikan berbasis sidik jari?

### **C. Tujuan Penelitian**

1. Mendesain alat pengaman kendaraan sistem pengapian berbasis sidik jari.
2. Menguji kinerja alat pengaman kendaraan roda dua pada sistem kelistrikan berbasis sidik jari.

### **D. Batasan Masalah**

1. Pengaman kendaraan yang ada sekarang masih kurang canggih. Pengamanan kendaraan dikatakan canggih jika kendaraan tidak dapat di curi.
2. Masih banyak pengendara sepeda motor di bawah usia 17 tahun sehingga menimbulkan angka kecelakaan yang tinggi. Pengendara sepeda motor di bawah usia 17 tahun .

### **E. Manfaat Penelitian**

Manfaat penelitian ini adalah dengan menggunakan alat pengaman roda dua pada sistem pengapian sidik jari kendaraan akan aman dari tindak pencurian. Dengan menggunakan alat pengaman kendaraan roda dua pada sistem kelistrikan berbasis sidik jari, pengendara di bawah umur tidak dapat mengendarai sepeda motor, sehingga angka kecelakaan lalu lintas akan turun.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Kajian Teori**

##### **1. Arduino Nano**

Arduino Nano adalah papan mikrokontroler berbasis ATmega328P yang berukuran kecil, dirancang untuk aplikasi yang memerlukan ukuran kompak dan fleksibilitas. Dengan dimensi yang lebih kecil dibandingkan dengan Arduino Uno, Arduino Nano ideal untuk proyek yang membutuhkan ruang terbatas. Meskipun ukurannya kecil, papan ini tetap menawarkan fungsi yang sama dengan papan Arduino lainnya, termasuk kemampuan untuk berinteraksi dengan berbagai sensor dan modul (Handoko, P, 2017).

Kelebihan:

- **Portabilitas:** Ukurannya yang kecil membuat Arduino Nano mudah digunakan dalam proyek-proyek portabel.
- **Fleksibilitas:** Arduino Nano bisa diterapkan dalam berbagai jenis aplikasi, seperti robotika, otomasi rumah, dan proyek IoT (Internet of Things).
- **Kompatibilitas:** Mendukung berbagai library dan modul, yang memudahkan integrasi dengan komponen lainnya.

Kekurangan:

- Keterbatasan Pin: Walaupun memiliki jumlah pin yang cukup untuk banyak aplikasi, Arduino Nano memiliki lebih sedikit pin dibandingkan papan yang lebih besar seperti Arduino Uno, yang mungkin menjadi kendala untuk proyek yang lebih kompleks.
- Kinerja: Meskipun cukup kuat untuk banyak aplikasi, kecepatan pemrosesan dan kapasitas memori Arduino Nano terbatas dibandingkan dengan mikrokontroler yang lebih canggih.



**Gambar 2. 1** Arduino Nano  
(Sumber: Handoko, P. 2017)

## 2. LCD

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan diberbagai bidang misalnya alat-alat elektronik seperti televisi, kalkulator, atau pun layar computer (Rahmat & Sidik, 2019). Pada postingan aplikasi LCD yang digunakan ialah LCD dot matrik dengan jumlah karakter 4 x 20. LCD sangat

berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat.



**Gambar 2. 2** LCD (*Liquid Crystal Display*) 4x20  
(Sumber : Syam, R, Dkk 2020 )

### 3. Keypad

*Keypad* adalah saklar-saklar push button yang disusun secara matriks yang berfungsi untuk menginput data. *Keypad* berfungsi sebagai interface antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah HMI (Human Machine Interface). Pada dasarnya *keypad* adalah sejumlah tombol yang disusun sedemikian rupa sehingga membentuk susunan tombol angka dan beberapa menu. Keypad matrix diatas memiliki ukuran fisik 77 x 69 x 1mm dengan panjang kabel connector  $\pm 80$ mm. Tegangan maksimum yang melintas pada setiap segment atau tombol adalah 24VDC dengan arus maksimum 30mA. Suhu pengoperasian antara 32 hingga 122 0F (0 hingga 500C). (Hidayat, R 2018)



**Gambar 2. 3** Keypad  
(*Sumber: Hidayat, R. 2018*)

#### 4. **Fingerprint Reader**

Sebuah sistem fingerprint scanner memiliki dua pekerjaan, yakni mengambil gambar sidik jari pengguna, dan memutuskan apakah pola alur sidik jari dari gambar yang diambil sama dengan pola alur sidik jari yang ada di database. Ada beberapa cara untuk mengambil gambar sidik jari seseorang, namun metode digunakan dalam penelitian ini adalah optical scanning. Inti dari scanner optical adalah Charge Coupled Device (CCD). Proses scan mulai berlangsung saat seseorang meletakkan jari pada lempengan kaca dan sebuah kamera CCD mengambil gambarnya. Scanner memiliki sumber cahaya sendiri, biasanya berupa larik light emitting diodes (LED), untuk menyinari alur sidik jarinya. Sistem CCD menghasilkan gambar jari yang terbalik, area yang lebih gelap merepresentasikan lebih banyak cahaya yang dipantulkan (bagian punggung dari alur sidik jari), dan area yang lebih terang merepresentasikan lebih sedikit cahaya yang dipantulkan (bagian lembah dari alur sidik jari).

Fingerprint Sensor akan bekerja membuat penambahan deteksi sidik jari dan verifikasi menjadi sangat sederhana. Di dalam modul ini terdapat chip DSP bertenaga tinggi yang dapat melakukan perenderan gambar dan kalkulasi secara cepat. Modul ini juga dapat mendaftarkan sidik jari baru secara langsung hingga 100 sidik jari dapat disimpan dalam memori. Pada dasarnya ada dua persyaratan untuk menggunakan Fingerprint sensor. Pertama, kita harus mendaftarkan sidik jari dan menetapkan ID #. Setelah kita mendaftarkan semua sidik jari yang dibutuhkan, kita dapat dengan mudah memverifikasi data sidik jari yang telah terdaftar. Keluaran pada sidik jari ini berupa ID #. Pada Fingerprint Sensor ini terdapat 6 pin, yaitu : pin VCC, TX, RX, ground, proximity, dan pin 3,3V menggunakan metode optical scanning. (Simatupang Dkk, 2021).

Implementasi sensor PIR dalam peralatan elektronik melibatkan penangkapan energi panas dari infra merah. Ketika seseorang melewati sensor, pancaran infra merah dari tubuhnya ditangkap oleh sensor piroelektrik, menghasilkan arus listrik. Arus ini diperkuat dan dibandingkan oleh komparator untuk menghasilkan output. (H. A. Adidharma, 2019).



**Gambar 2. 4** sensor Fingerprint  
(Sumber: H. A. Adidharma, 2019)

#### **5. Modul Stepdown LM2596**

Modul stepdown atau penurun tegangan DC LM2596 ini akan menyelesaikan masalah perbedaan tegangan yang dibutuhkan dengan yang tersedia. Dikarenakan dalam pembuatan rangkaian elektronika atau modul-modul mikrokontroler sering muncul perbedaan tegangan kerja antar modul sehingga memerlukan sebuah modul regulator untuk menyesuaikan tegangan.

Modul stepdown DC LM2596 termasuk DC Buck Converter. DC Buck Converter adalah rangkaian elektronika yang berfungsi sebagai penurun tegangan DC ke DC (konverter DC-to-DC atau Choppers) dengan metode switching. Secara garis besar rangkaian konverter dc to dc ini memakai komponen switching seperti MOSFET (Metal Oxide Semiconductor Field Effect Transistor), yang diatur melalui duty cycle



**Gambar 2. 5** Modul Stepdown LM2596  
(Sumber: Wilson, 2017)

#### 6. Modul Relay

Modul relay adalah salah satu piranti yang beroperasi berdasarkan prinsip elektromagnetik untuk menggerakkan kontaktor guna memindahkan posisi ON ke OFF atau sebaliknya dengan memanfaatkan tenaga listrik. Peristiwa tertutup dan terbukanya kontaktor ini terjadi akibat adanya efek induksi magnet yang timbul dari kumparan induksi listrik. Perbedaan yang paling mendasar antara relay dan sakelar adalah pada saat pemindahan dari posisi ON ke OFF. Relay melakukan pemindahannya secara otomatis dengan arus listrik, sedangkan sakelar dilakukan dengan cara manual. (Suherman, 2021). Di dalam relay terdapat beberapa komponen pendukung fungsi kerja relay, sebagai berikut :

- Penyangga (Armature)
- Kumparan (Coil)

- Pegas (Spring)
- Saklar (Switch Contact)
- Inti Besi (Iron Core)



**Gambar 2. 6** Modul Relay  
(Sumber: Isyanto, H., 2017)

## 7. Modul I2C

Modul I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati master. (Gazali, W., 2019).



**Gambar 2. 7** Modul I2C  
(Sumber: Gazali, W., 2019)

## 8. *Software Arduino IDE*

*Arduino IDE (Integrated Development Environment)* merupakan *software* untuk melakukan pembuatan program terintegrasi agar dalam melakukan pengembangan pada berbagai macam *hardware*. Pembuatan program, compile serta *upload* program ke *board* mikrokontroler yang digunakan yaitu *Arduino Uno*. Bahasa yang digunakan untuk pemrograman yaitu bahasa C pada *software* untuk membuat logika *input* dan *output* (Handoko, 2017).

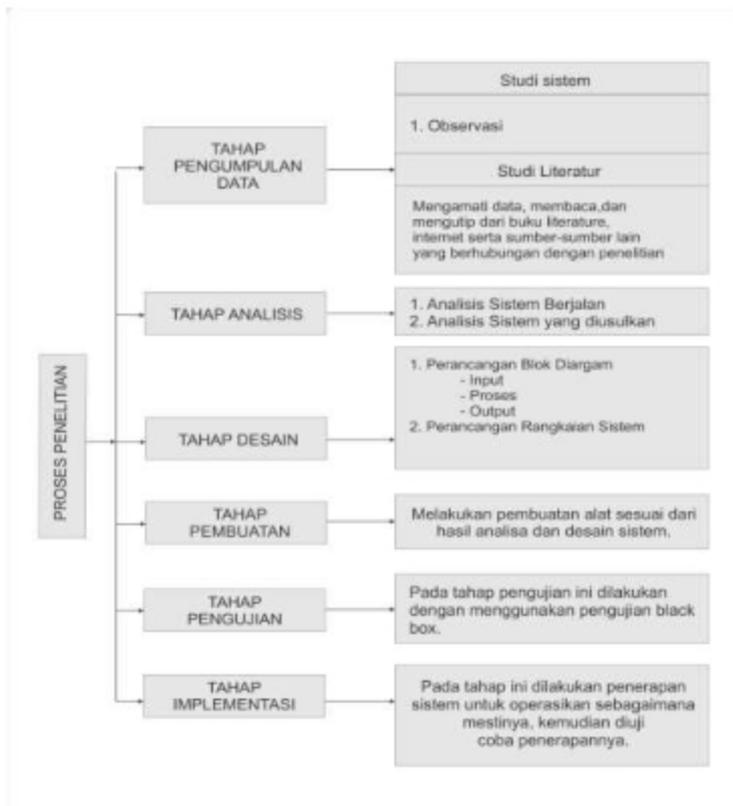


**Gambar 2. 8** *Arduino IDE (Integrated Development Environmet)*  
(*sumber: Dokumentasi pribadi*)

# BAB III

## METODE PENELITIAN

### A. Tahapan Penelitian



**Gambar 3. 1** Tahapan Penelitian  
(Sumber: dokumentasi pribadi)

## B. Waktu dan Tempat Penelitian

### 1. Lokasi Penelitian

Lokasi pelaksanaan penelitian dilakukan di laboratorium Teknik Elektro Universitas Muhammadiyah Parepare.

### 2. Waktu Penelitian

Waktu penelitian dapat dilihat pada tabel dibawah ini:

**Tabel 3. 1** Jadwal Pelaksanaan Penelitian

No.	Kegiatan	Waktu Kegiatan											
		Bulan Ke-1				Bulan Ke-2				Bulan Ke-3			
		1	2	3	4	1	2	3	4	1	2	3	4
1	<i>Studi Literatur</i>	■	■										
2	Perancangan Desain Alat		■	■	■								
3	Instalasi Komponen Utama					■							
4	Pembuatan Kerangka Alat						■	■					
5	Uji Lapangan										■		
6	Evaluasi Dan Analisis Hasil Uji											■	
7	Pembuatan Laporan Hasil Penelitian											■	■

### C. Alat dan Bahan

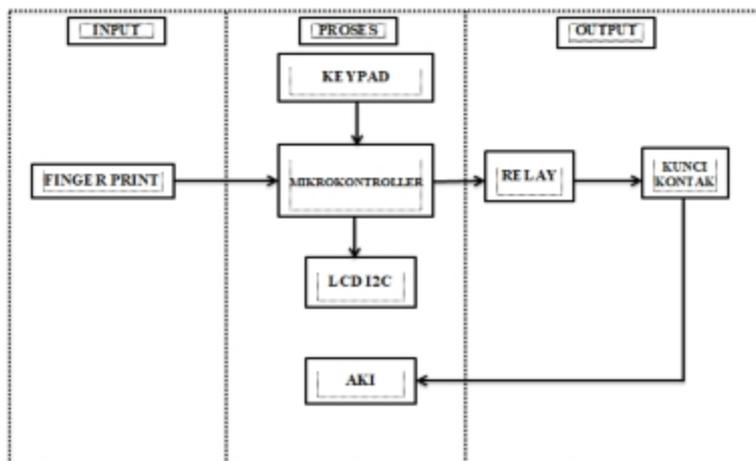
Berikut alat dan komponen yang digunakan pada perancangan ini adalah sebagai berikut :

**Tabel 3. 2** Alat dan Bahan yang Dibutuhkan

No.	Alat dan Bahan	Jumlah
1	<i>Finger reader</i>	1 buah
2	<i>Arduino Nano</i>	1 buah
3	<i>Modul lcd</i>	1 buah
4	<i>Modul I2C</i>	1 buah
5	<i>Modul Relay</i>	1 buah
6	<i>Keypad</i>	1 buah
7	<i>Down Converter</i>	1 buah

### D. Rancangan Penelitian

Rancangan alat yang akan diteliti dapat dilihat pada gambar blok diagram di bawah ini:



**Gambar 3. 2** Blok Diagram  
(sumber: Dokumentasi pribadi)

Blok diagram pada gambar 3.2 adalah Gambaran system *Prototype* Sistem Pengaman Kunci Sepeda Motor Menggunakan Sidik Jari Berbasis Mikrokontroler, untuk perancangan alat pada input yaitu ada Keypad untuk memasukkan pin, sensor Finger Print untuk mendeteksi sidik jari, kemudian data akan di cocokan apakah sama dengan sidik jari yang sudah pernah di simpan di database. Jika data sama atau cocok maka akses akan di buka, jika data tidak cocok maka akses akan tertutup. Setelah Arduino memproses semua data yang didapat dari sensor maka akan ada beberapa output yang menanggapi, seperti relay untuk mengalirkan arus listrik dari saklar utama, LCD 2x16 untuk menampilkan apa yang terjadi saat itu.

Pada alat tugas akhir ini mendapat sumber tegangan dari baterai atau aki motor yang memiliki tegangan sebesar 12V DC, lalu untuk menyuplai rangkaian arduino UNO tersebut, tegangan dari baterai atau aki akan diturunkan oleh DC-DC Konverter menjadi 9V DC.

Cara kerja starter motor dengan menggunakan sensor yang pertama yaitu pengguna harus memasukkan pin menggunakan keypad lalu menempelkan sidik jari yang sudah terdaftar untuk menghidupkan dan mematikan Relay. Pembacaan sidik jari berfungsi untuk mengaktifkan relay yang terhubung dengan CDI motor. Jika sensor Fingerprint membaca sidik jari yang salah maka kita akan diminta untuk memasukkan sidik jari kembali.

### **E. Teknik Pengumpulan Data**

Teknik pengumpulan data yang dilakukan pada penelitian ini dilaksanakan dalam beberapa tahapan.

#### **a. Data Primer**

Data primer yaitu data yang didapat langsung dilapangan. Data ini diperoleh melalui teknik observasi yang dilakukan untuk mengetahui secara langsung.

#### **b. Data Sekunder**

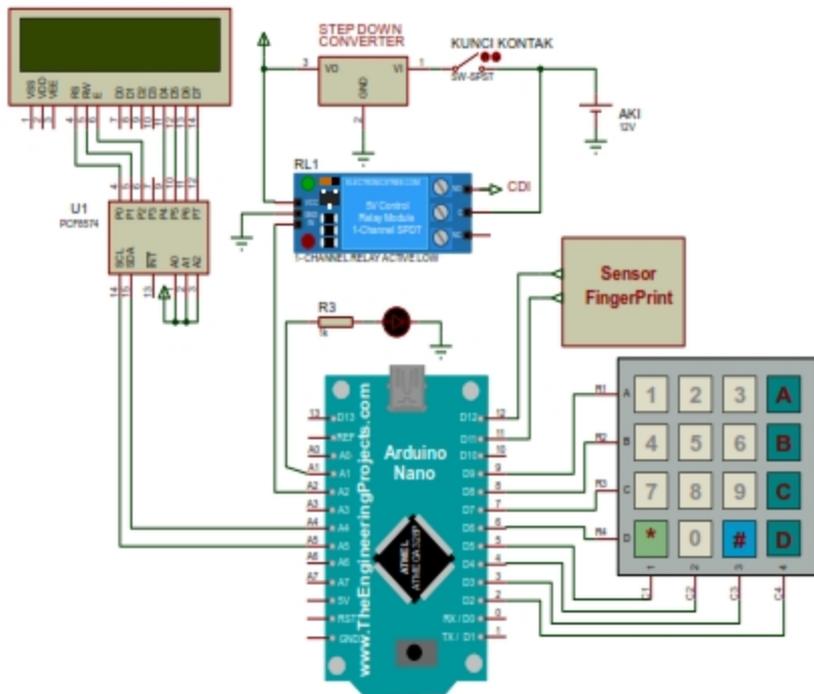
Data Sekunder yang di mana data di ambil dari buku jurnal atau penelitian yang berkaitan dengan penelitian ini. Dan mengambil data dari berbagai sumber yang berkaitan tentang penelitian yang diperlukan untuk mendapatkan referensi yang tepat dan akurat sesuai dengan penelitian ini.

## BAB IV

### HASIL DAN PEMBAHASAN

#### A. Rancangan *Hardware*

Perancangan *Hardware* atau perangkat keras pada *Prototype* Sistem Pengaman Kunci Sepeda Motor Menggunakan Sidik Jari Berbasis Mikrokontroler secara umum dapat dilihat pada gambar 4.1.



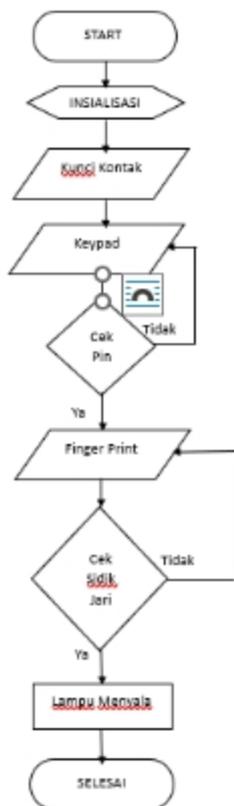
Gambar 4. 1 Perancangan *Hardware*

Adapun komponen yang digunakan pada Gambar 4.1 yaitu:

1. Modul LCD yang terhubung ke I2C berfungsi untuk menampilkan status, dimana pin SDA terhubung ke Pin A4 pada Arduino dan SCL ke Pin A5 pada Arduino.
2. Stepdown converter sebagai penurun tegangan, pin 1 terhubung ke kunci kontak, pin 2 ke ground dan pin 3 ke input 12V.
3. Kunci Kontak terhubung ke Aki dan stepdown converter.
4. Relay, pin VCC ke input 12V, GND ke Ground dan IN ke pin A2 pada Arduino.
5. Sensor finger berfungsi untuk mendeteksi sidik jari yang sudah didaftarkan, dimana pinnya terhubung ke pin 12 dan 11 pada Arduino.
6. Keypad berfungsi sebagai tombol untuk memasukkan PIN sebelum meletakkan sidik jari, Dimana pinnya terhubung ke pin 2 sampai pin 9 pada Arduino.

### **B. Perancangan *Software***

Perancangan perangkat lunak atau perancangan software ini bertujuan untuk mengatur kinerja pada input dan output dari perangkat keras dengan adanya instruksi-instruksi yang dimasukkan ke Arduino. Flowchart prinsip kerja sistem dapat dilihat pada gambar 4.2.



**Gambar 4. 2** Flowchart Sistem

Pada gambar 4.2 merupakan diagram alir dari sistem *Prototype* Sistem Pengaman Kunci Sepeda Motor Menggunakan Sidik Jari Berbasis Mikrokontroller, saat alat dinyalakan maka sistem akan melakukan inisialisasi terlebih dahulu, lalu nyalakan kunci kontak dan masukkan PIN menggunakan Keypad, jika PIN salah maka masukkan Ulang PIN, jika benar Sensor Finger Print dapat digunakan untuk

mendeteksi sidik jari, jika salah kita akan diminta untuk meletakkan kembali sidik jari untuk dideteksi, jika benar maka lampu akan menyala.

Adapun penjelasan *sketch* program pada flowchart penelitian ini adalah sebagai berikut:

```
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_Fingerprint.h>
#include <Keypad.h>
```

Potongan program diatas bertujuan untuk mengimpor beberapa pustaka penting yang diperlukan untuk mendukung fungsi utama dari program tersebut. Pustaka `EEPROM.h` memungkinkan program untuk membaca dan menulis data ke memori EEPROM pada mikrokontroler, yang berfungsi untuk menyimpan data secara permanen, seperti kode sandi atau pengaturan penting lainnya. Pustaka `SoftwareSerial.h` digunakan untuk menciptakan port serial tambahan, yang berguna untuk berkomunikasi dengan perangkat lain seperti sensor sidik jari melalui pin digital tambahan pada mikrokontroler, di luar port serial bawaan.

Selanjutnya, pustaka `LiquidCrystal\_I2C.h` digunakan untuk mengontrol tampilan LCD (Liquid Crystal Display) melalui antarmuka I2C, yang mempermudah menampilkan informasi seperti teks di layar. Pustaka `Adafruit\_Fingerprint.h` mengelola interaksi dengan sensor sidik jari dari Adafruit, memberikan fungsi-fungsi untuk mendaftarkan, menghapus, dan memverifikasi sidik jari. Terakhir, pustaka `Keypad.h` digunakan untuk mengontrol keypad matriks, memungkinkan program

mendeteksi dan memproses input dari tombol-tombol pada keypad. Secara keseluruhan, pustaka-pustaka ini mendukung operasi utama seperti penyimpanan data, komunikasi, kontrol tampilan, manajemen sidik jari, dan pembacaan input pengguna dalam sistem keamanan berbasis mikrokontroler.

```

SoftwareSerial mySerial(11,12); //11=kuning, 12=coklat

int u=0;
uint8_t id, ID_terbaca;

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);

#define resetPin A0
#define resetLed A3
#define ledPin A1
#define relayPin A2

char key, k1,k2,k3,k4;
String jawab, kunci, ganti;
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
int hitung = 0;

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

byte rowPins[ROWS] = {9,8,7,6}; //connect to the row pinouts
of the keypad
byte colPins[COLS] = {5,4,3,2}; //connect to the column
pinouts of the keypad
Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins,
ROWS, COLS );

```

Bagian Program ini menginisialisasi beberapa komponen dan variabel yang diperlukan untuk menjalankan sistem keamanan berbasis mikrokontroler. Berikut adalah penjelasan dari masing-masing bagian kode:

1. `SoftwareSerial mySerial(11,12); // 11=kuning, 12=coklat:`

Baris ini membuat port serial tambahan menggunakan pustaka 'SoftwareSerial'. Port serial ini digunakan untuk berkomunikasi dengan sensor sidik jari melalui pin digital 11 (kuning) dan 12 (coklat) pada mikrokontroler.

2. `int u=0;:`

Variabel ini digunakan sebagai penanda atau indikator dalam program, yang mungkin digunakan untuk melacak status tertentu, seperti status LED atau relay.

3. `uint8_t id, ID_terbaca;:`

Variabel 'id' digunakan untuk menyimpan ID sidik jari yang akan direkam atau dihapus, sedangkan 'ID\_terbaca' menyimpan ID sidik jari yang terdeteksi saat proses verifikasi.

4. `Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);:`

Baris ini menginisialisasi objek 'finger' dari kelas 'Adafruit\_Fingerprint', yang digunakan untuk mengelola komunikasi dengan sensor sidik jari. Objek ini menggunakan port serial tambahan 'mySerial' untuk berkomunikasi dengan sensor.

5. `#define resetPin A0, #define resetLed A3, #define ledPin A1, #define relayPin A2:`

Baris-baris ini mendefinisikan pin-pin analog (A0, A3, A1, dan A2) untuk berbagai fungsi dalam sistem, seperti pin untuk reset, LED, dan relay.

6. `char key, k1, k2, k3, k4;:`

Variabel-variabel ini digunakan untuk menyimpan input dari keypad, seperti tombol yang ditekan ('key') dan digit sandi (k1, k2, k3, k4).

7. `String jawab, kunci, ganti;:`

Variabel-variabel string ini digunakan untuk menyimpan sandi yang dimasukkan ('jawab'), sandi yang tersimpan ('kunci'), dan sandi baru yang akan diganti ('ganti').

8. `const byte ROWS = 4; // four rows, const byte COLS = 4; // four columns:`

Konstanta ini mendefinisikan jumlah baris dan kolom pada keypad, yaitu 4x4.

9. `int hitung = 0;:`

Variabel ini digunakan untuk menghitung sesuatu dalam program, mungkin terkait dengan proses tertentu seperti iterasi atau pencatatan jumlah.

10. `char keys[ROWS][COLS];`

Matriks ini mendefinisikan tata letak tombol pada keypad 4x4, di mana setiap elemen merepresentasikan karakter yang akan dihasilkan ketika tombol ditekan.

11. `byte rowPins[ROWS] = {9,8,7,6}; // connect to the row pinouts of the keypad:`

Array ini mendefinisikan pin-pin digital yang terhubung ke baris-baris pada keypad.

12. `byte colPins[COLS] = {5,4,3,2}; // connect to the column pinouts of the`

keypad:

Array ini mendefinisikan pin-pin digital yang terhubung ke kolom-kolom pada keypad.

13. `Keypad myKeypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);:`

Baris ini menginisialisasi objek `myKeypad` dari kelas `Keypad`, yang mengatur tata letak dan interaksi dengan keypad. `makeKeymap(keys)` digunakan untuk mengaitkan matriks tombol dengan pin-pin pada mikrokontroler.

```
void setup() {
  Serial.begin(9600);
  finger.begin(57600);
  pinMode(ledPin, OUTPUT);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);
  pinMode(13,OUTPUT);
  byte k0 = EEPROM.read(1);
  if(k0==255) {
    EEPROM.write(1,'1');
    EEPROM.write(2,'2');
    EEPROM.write(3,'3');
    EEPROM.write(4,'4');
  }
  baca_kunci();
  lcd.init();
}
```

```

lcd.setBacklight(50);

if(finger.verifyPassword()) {
  bacaParameter();
  lcd.clear(); lcd.print(F("Module Terdeksi "));
  lcd.setCursor(0,1);
  lcd.print(F("Sidik Ada : "));
  lcd.print(finger.templateCount);
  delay(2000);
}
else {
  lcd.setCursor(0,0); lcd.print(F("Koneksi Error  "));
  lcd.setCursor(0,1); lcd.print(F("Periksa Koneksi "));
  while (1);
}
intro();
}

```

Bagian program diatas adalah fungsi `setup()`, yang merupakan bagian dari kode Arduino dan dijalankan sekali ketika mikrokontroler dinyalakan atau di-reset. Berikut penjelasan dari setiap bagian:

1. `Serial.begin(9600);`:

Fungsi ini memulai komunikasi serial pada kecepatan 9600 baud. Ini memungkinkan mikrokontroler untuk berkomunikasi dengan komputer atau perangkat lain melalui port serial.

2. `finger.begin(57600);`:

Menginisialisasi komunikasi dengan sensor sidik jari (`finger`) dengan baud rate 57600. Ini adalah kecepatan komunikasi data antara mikrokontroler dan sensor sidik jari.

3. ``pinMode(ledPin, OUTPUT);`` dan ``pinMode(relayPin, OUTPUT);``:  
Mengatur pin yang didefinisikan sebelumnya (``ledPin`` dan ``relayPin``) sebagai pin keluaran (OUTPUT). Ini berarti pin tersebut akan digunakan untuk mengendalikan LED dan relay.
4. ``digitalWrite(relayPin, HIGH);``:  
Mengatur pin relay (``relayPin``) ke status HIGH. Biasanya ini akan mengaktifkan relay atau memastikan perangkat yang dikendalikan relay dalam keadaan aktif.
5. ``pinMode(13, OUTPUT);``:  
Mengatur pin 13 sebagai pin keluaran. Pin ini sering digunakan untuk LED onboard pada papan Arduino, jadi ini mungkin untuk indikasi visual atau debugging.
6. ``byte k0 = EEPROM.read(1);``:  
Membaca nilai dari alamat 1 di memori EEPROM dan menyimpannya dalam variabel ``k0``. EEPROM adalah memori non-volatile yang digunakan untuk menyimpan data yang harus dipertahankan setelah mikrokontroler dimatikan.
7. ``if (k0 == 255) {``:  
Mengecek apakah nilai yang dibaca dari EEPROM sama dengan 255. Nilai 255 biasanya digunakan sebagai nilai default untuk menandai bahwa EEPROM belum diatur.
8. ``EEPROM.write(1, '1');`` dan seterusnya:

Jika nilai dari EEPROM adalah 255, maka EEPROM ditulis dengan nilai-nilai default (1, 2, 3, 4) pada alamat-alamat 1 hingga 4. Ini sering digunakan untuk menginisialisasi pengaturan awal jika EEPROM kosong atau belum diatur.

9. ``baca_kunci();``:

Memanggil fungsi ``baca_kunci()`` untuk membaca kunci atau sandi dari EEPROM dan menyimpannya dalam variabel yang diperlukan.

10. ``lcd.init();``:

Menginisialisasi layar LCD I2C. Ini adalah langkah awal untuk memastikan bahwa LCD siap digunakan untuk menampilkan informasi.

11. ``lcd.setBacklight(50);``:

Mengatur tingkat kecerahan lampu latar LCD. Nilai 50 adalah level kecerahan yang diatur, meskipun detail tentang pengaturan ini bisa bervariasi tergantung pada pustaka LCD yang digunakan.

12. ``if (finger.verifyPassword()) {``:

Mengecek apakah password untuk sensor sidik jari benar. Jika benar, maka program akan melanjutkan ke bagian berikutnya.

13. ``bacaParameter();``:

Memanggil fungsi ``bacaParameter()`` untuk membaca parameter sensor sidik jari dan menampilkan informasi terkait.

14. ``lcd.clear();`, `lcd.print(F("Module Terdeksi "));`, `lcd.setCursor(0, 1);`,  
`lcd.print(F("Sidik Ada : ")); lcd.print(finger.templateCount);``:

Membersihkan layar LCD, kemudian menampilkan pesan "Module Terdeksi" dan jumlah template sidik jari yang ada di sensor. `finger.templateCount` menunjukkan jumlah sidik jari yang terdaftar.

15. `delay(2000);`:

Menunggu selama 2 detik sebelum melanjutkan ke bagian berikutnya.

16. `else { lcd.setCursor(0, 0); lcd.print(F("Koneksi Error ")); lcd.setCursor(0, 1); lcd.print(F("Periksa Koneksi ")); while (1); }`:

Jika password sensor sidik jari tidak benar, maka akan menampilkan pesan kesalahan di layar LCD dan mengunci program dalam loop tak berujung dengan `while (1);`. Ini mengindikasikan bahwa ada masalah dengan koneksi sensor dan program akan berhenti berfungsi sampai masalah tersebut diperbaiki.

17. `intro();`:

Memanggil fungsi `intro()` untuk menampilkan pesan pengantar atau informasi awal pada LCD.

Secara keseluruhan, fungsi `setup()` menyiapkan semua komponen yang diperlukan, termasuk komunikasi serial, sensor sidik jari, dan tampilan LCD, serta melakukan pengecekan awal untuk memastikan semua komponen berfungsi dengan benar.

```
void loop() {
  //check key pressed
  key = myKeypad.getKey();

  if(key=='#'){
    key = NO_KEY; while (key == NO_KEY) {key =
```

```

myKeypad.getKey());};
    if(key=='#') {
        lcd.clear(); lcd.setCursor(0, 0); lcd.print(F("Input
Password !"));
        cek_password(); if(jawab==kunci) checkKeys();
        else { lcd.clear(); lcd.setCursor(0, 0);
            lcd.print(F("Password Salah !"));
            delay(3000); intro();
        }
    }
}

if(key=='*'){
    key = NO_KEY; while (key == NO_KEY) {key =
myKeypad.getKey());};
    if(key=='*') {
        lcd.clear(); lcd.setCursor(0, 0); lcd.print(F("Enter
Old Pswd !"));
        cek_password();
        if(jawab==kunci){
            lcd.clear();
            lcd.print(F("Enter New Pswd !"));
ganti_password();
        }
        else { lcd.clear(); lcd.setCursor(0, 0);
            lcd.print(F("Password Salah !"));
            delay(3000); intro();
        }
    }
}}

if(key=='A' || key=='B'){
    if(digitalRead(relayPin)==HIGH){
        for(int i=0;i<3;i++){ //cek sidik jari 3 kali
            lcd.clear(); lcd.print(F("Letakkan Jari !!"));
delay(1000);
            getFingerprintID(); delay(500);
        }
    }
}
if(digitalRead(relayPin)==LOW){
    lcd.clear(); lcd.print(F(" Kunci Enable  "));
    delay(5000);
}

```

```
    }  
    intro();  
  }  
}
```

Fungsi `loop()` dalam program ini berfungsi untuk menangani interaksi dengan pengguna melalui keypad dan layar LCD, serta mengelola proses otentikasi sidik jari dan pengaturan password. Pada awal fungsi, program memeriksa apakah ada tombol yang ditekan pada keypad. Jika tombol `#` ditekan, program meminta pengguna untuk memasukkan password, dan jika password yang dimasukkan sesuai dengan yang disimpan, fungsi `checkKeys()` dipanggil untuk verifikasi lebih lanjut. Jika password salah, tampilan LCD akan menampilkan pesan kesalahan dan program akan kembali ke tampilan awal. Sebaliknya, jika tombol `\*` ditekan, program akan meminta pengguna untuk memasukkan password lama, dan jika benar, pengguna akan diminta untuk mengatur password baru melalui fungsi `ganti\_password()`. Jika password lama salah, pesan kesalahan akan ditampilkan. Selain itu, jika tombol `A` atau `B` ditekan, program memeriksa status relay. Jika relay aktif, program meminta pengguna untuk meletakkan jari pada sensor sidik jari dan melakukan pemindaian hingga tiga kali. Jika relay non-aktif, program menampilkan pesan bahwa kunci telah diaktifkan. Setelah proses selesai, fungsi `intro()` dipanggil untuk mengembalikan tampilan awal pada LCD. Seluruh proses ini memungkinkan pengguna untuk mengelola password dan akses melalui keypad dan sensor sidik jari dengan umpan balik visual dari layar LCD.

```

void Enroll() {
    int count=0;
    lcd.clear(); lcd.print(F("Merekam Sidik "));
    lcd.setCursor(0,1); lcd.print(F("Nomor ID :"));

    while(1){
        lcd.setCursor(10,1);
        if(count<10) lcd.print("0");
        lcd.print(count);

        key = NO_KEY;
        while (key == NO_KEY) {key = myKeypad.getKey();};

        if(key=='A'){count++; if(count>25) count=0; }
        else if(key=='B') {count--; if(count<0) count=25; }
        else if(key == 'C') { //Enroll
            id=count; getFingerprintEnroll(); return;
        }
        else { key=NO_KEY; return;} //exit
    }
}

```

Ketika fungsi `Enroll()` dipanggil, tampilan LCD akan dihapus dan menampilkan pesan "Merekam Sidik" dan "Nomor ID :" di baris kedua. Fungsi ini kemudian memasuki loop yang berfungsi untuk memilih ID sidik jari yang akan direkam.

Di dalam loop, LCD menampilkan ID yang sedang dipilih pada posisi kursor yang ditentukan. ID ditampilkan dengan dua digit, dan jika nilai ID kurang dari 10, angka nol ditambahkan di depannya untuk format yang konsisten.

Program kemudian menunggu input dari keypad. Jika tombol 'A' ditekan, nilai `count` (yang mewakili ID sidik jari) akan bertambah. Jika `count` melebihi 25, ia akan direset menjadi 0. Jika tombol 'B' ditekan, `count` akan dikurangi, dan jika nilai `count` menjadi kurang dari 0, ia akan direset menjadi 25.

Jika tombol 'C' ditekan, ID yang dipilih akan disimpan dalam variabel 'id', dan fungsi 'getFingerprintEnroll()' akan dipanggil untuk memulai proses perekaman sidik jari dengan ID tersebut. Setelah proses perekaman dimulai, fungsi 'Enroll()' akan berhenti dan keluar dengan 'return'.

Jika tombol lain ditekan atau jika input tidak sesuai, loop akan diakhiri dengan 'key=NO\_KEY; return', yang membuat fungsi keluar tanpa melakukan tindakan lebih lanjut.

```
void cek_password() {
  jawab="";
  // jawaban 4 digit pertama yang masuk dipindah ke string
  key = NO_KEY;
  while (key == NO_KEY) {key = myKeypad.getKey();};
  jawab += key; lcd.setCursor(0, 1); lcd.print("*");
  key = NO_KEY;
  while (key == NO_KEY) {key = myKeypad.getKey();};
  jawab += key; lcd.setCursor(1, 1); lcd.print("*");
  key = NO_KEY;
  while (key == NO_KEY) {key = myKeypad.getKey();};
  jawab += key; lcd.setCursor(2, 1); lcd.print("*");
  key = NO_KEY;
  while (key == NO_KEY) {key = myKeypad.getKey();};
  jawab += key; lcd.setCursor(3, 1); lcd.print("*");
  //lcd.setCursor(10, 1); lcd.print(jawab);
  delay(1000);
}
```

Fungsi 'cek\_password()' dirancang untuk mengumpulkan dan memverifikasi password yang dimasukkan pengguna melalui keypad, serta menampilkannya di layar LCD dengan karakter bintang (\*). Ketika fungsi ini dipanggil, ia pertama-tama menginisialisasi variabel 'jawab' sebagai string kosong. Fungsi kemudian memasuki loop yang menunggu hingga tombol pada keypad ditekan. Setelah tombol ditekan,

nilai tombol tersebut ditambahkan ke variabel `jawab`, dan karakter bintang (\*) ditampilkan di LCD pada posisi yang sesuai. Proses ini diulang empat kali untuk menangkap empat digit password. Setiap digit yang dimasukkan ditampilkan sebagai bintang (\*) pada posisi kursor yang berbeda di baris kedua LCD. Setelah semua empat digit password dimasukkan, fungsi menunggu selama satu detik sebelum selesai, memberikan waktu bagi pengguna untuk melihat bahwa password telah dimasukkan dengan benar.

```
void ganti_password() {
  cek_password();
  ganti=jawab;

  lcd.clear();
  lcd.setCursor(0, 0); lcd.print(F("ReEnter New Pswd"));

  cek_password();

  if(jawab==ganti) {
    lcd.setCursor(0, 0); lcd.print(F(" New Password  "));
    lcd.setCursor(0, 1); lcd.print(F("      OK      "));
    delay(3000);

    EEPROM.write(1,jawab[0]); //jawab adalah variabel array
6 elemen (byte)
    EEPROM.write(2,jawab[1]);
    EEPROM.write(3,jawab[2]);
    EEPROM.write(4,jawab[3]);

    baca_kunci();
  }
  else {lcd.clear();
    lcd.setCursor(0, 0); lcd.print(F(" New Password
"));
    lcd.setCursor(0, 1); lcd.print(F("  ERROR !!!
"));
    delay(3000);
```

```

    }
}

uint8_t getFingerprintEnroll() {
    int p = -1;
    lcd.clear(); lcd.print(F("Nomor ID:")); lcd.print(id);
    lcd.setCursor(0,1); lcd.print(F("Letakkan Jari "));
    delay(2000);

    Serial.print("Waiting for valid finger to enroll as #");
    Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                lcd.clear(); lcd.print(F("Membaca Sidik "));
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                lcd.clear(); lcd.print(F("Tak Ada Jari"));
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                break;
            default:
                Serial.println("Unknown error");
                break;
        }
    }
}

// OK success!
p = finger.image2Tz(1);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");

```

```

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

Serial.println("Remove finger");
lcd.clear(); lcd.print(F("Angkat Jari Anda"));
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
lcd.clear();          lcd.print(F("Letakkan Jari"));
lcd.setCursor(0,1); lcd.print(F("  Kembali  "));

while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image taken");
        lcd.clear(); lcd.print(F("Membaca Sidik "));
        break;
    case FINGERPRINT_NOFINGER:
        Serial.print(".");
        break;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        break;
    case FINGERPRINT_IMAGEFAIL:

```

```

        Serial.println("Imaging error");
        break;
    default:
        Serial.println("Unknown error");
        break;
    }
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

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

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {

```

```

    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

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

return true;
}

int getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken"); break;
        case FINGERPRINT_NOFINGER:
            Serial.println("No finger detected");
            ID_terbaca = 0;
            finger.LEDcontrol(FINGERPRINT_LED_OFF, 0,
                FINGERPRINT_LED_BLUE);
            finger.LEDcontrol(FINGERPRINT_LED_OFF, 0,
                FINGERPRINT_LED_RED);
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:

```

```

        Serial.println("Communication error");
        return p;
    case FINGERPRINT_IMAGEFAIL:
        Serial.println("Imaging error");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted"); break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy"); return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error"); return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
    finger.LEDcontrol(FINGERPRINT_LED_FLASHING, 25,
FINGERPRINT_LED_PURPLE, 10);
    delay(1000);
    if(u==0) { digitalWrite(ledPin,HIGH); u=1; }
    else if(u==1) { digitalWrite(ledPin,LOW); u=0; }
    digitalWrite(relayPin,LOW);
}
}

```

```

else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
}
else if (p == FINGERPRINT_NOTFOUND) {
    finger.LEDcontrol(FINGERPRINT_LED_FLASHING, 25,
FINGERPRINT_LED_RED, 10);
    delay(1000);
    Serial.println("Did not find a match"); return p;
}
else {
    Serial.println("Unknown error"); return p;
}

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

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

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

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) {
        lcd.clear(); lcd.print(F("Anda Gagal "));
        lcd.setCursor(0,1); lcd.print(F("Coba lagi !"));
        delay(2000);
        return -1;
    }

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

```

```
Serial.println(finger.confidence);  
    return finger.fingerID;  
}
```

Fungsi `ganti_password()` digunakan untuk mengubah password yang tersimpan dalam EEPROM dengan password baru yang diinputkan oleh pengguna. Fungsi ini dimulai dengan memanggil `cek_password()` untuk membaca dan menyimpan password baru pertama yang dimasukkan pengguna ke dalam variabel `ganti`. Kemudian, layar LCD akan menampilkan pesan meminta pengguna untuk memasukkan password baru sekali lagi. Jika password kedua yang dimasukkan cocok dengan password pertama yang disimpan dalam `ganti`, maka password baru tersebut akan disimpan ke EEPROM di lokasi yang sesuai. Jika tidak cocok, fungsi akan menampilkan pesan kesalahan di layar LCD.

Fungsi `getFingerprintEnroll()` bertanggung jawab untuk mendaftarkan sidik jari baru ke dalam sistem. Pertama, fungsi meminta pengguna untuk meletakkan jarinya pada sensor dan kemudian mengubah gambar sidik jari menjadi template yang bisa disimpan. Proses ini melibatkan beberapa tahap, termasuk memeriksa apakah sidik jari telah diambil dan apakah gambar telah dikonversi dengan sukses. Setelah template sidik jari kedua berhasil diambil dan dikonversi, model sidik jari akan dibuat dan disimpan ke lokasi yang ditentukan.

Fungsi `getFingerprintID()` digunakan untuk mencocokkan sidik jari yang ditempatkan pada sensor dengan sidik jari yang tersimpan dalam database. Jika sidik jari terdeteksi dan berhasil dicocokkan, fungsi akan mengaktifkan atau mematikan LED dan mengendalikan relay sesuai dengan hasil pencocokan. Jika tidak ada

kecocokan atau terjadi kesalahan komunikasi, fungsi akan menampilkan pesan yang sesuai di Serial Monitor dan LCD.

Fungsi `getFingerprintIDez()` adalah versi yang lebih sederhana dari `getFingerprintID()`, dan dirancang untuk pencocokan sidik jari cepat. Fungsi ini akan mengembalikan ID sidik jari yang cocok jika ditemukan, atau -1 jika pencocokan gagal. Fungsi ini juga memberikan umpan balik kepada pengguna melalui LCD jika pencocokan gagal, dan meminta pengguna untuk mencoba lagi.

### **C. Pengujian**

Pengujian bertujuan untuk memastikan bahwa alat yang dibangun telah bekerja sesuai seperti yang diinginkan. Pengujian yang dilakukan yaitu pengujian sensor finger print dengan melakukan pengujian sidik jari yang sudah terdaftar dan menambahkan sidik jari baru.

#### **1. Pengujian Sidik Jari Terdaftar.**

Pengujian ini bertujuan untuk mengetahui apakah sensor dapat membaca sidik jari dengan baik, dengan cara meletakkan sidik jari yang sudah terdaftar beberapa kali.



**Gambar 4. 3** Pengujian Sidik Jari Terdaftar

**Tabel 4. 1** Pengujian Sidik Jari Terdaftar

<b>Percobaan</b>	<b>Kondisi</b>
Sidik jari Jempol	On
Sidik Jari Kelingking	On
Sidik Jari Telunjuk	On
Sidik Jari Tengah	On
Sidik Jari Manis	On

Dari Tabel 4.1 dapat dilihat bahwa hasil dari beberapa pengujian sidik jari semuanya terbaca dengan baik tanpa kendala, dengan tanda menyala lampu indikator pada alat.

## 2. Pengujian Penambahan Sidik Jari.

Pengujian Penambahan Sidik Jari ini bertujuan untuk mengetahui apakah alat yang sudah dibuat dapat menyimpan sidik jari yang baru dan mengeksekusinya dengan baik.



**Gambar 4. 4** Pengujian Penambahan Sidik Jari

**Tabel 4. 2** Pengujian Penambahan Sidik Jari

<b>Percobaan</b>	<b>Status</b>	<b>Kondisi</b>
Sidik Jari 1	Berhasil	On
Sidik Jari 2	Berhasil	On
Sidik Jari 3	Berhasil	On
Sidik Jari 4	Berhasil	On
Sidik Jari 5	Berhasil	On

Pada Tabel 4.2 dapat dilihat Pengujian Penambahan Sidik Jari dengan menggunakan 5 (lima) sidik jari baru semuanya mendapatkan status berhasil dan sidik jari baru berhasil untuk mengaktifkan lampu indicator.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. Kesimpulan**

Setelah melakukan pengujian pada alat *Prototype* Sistem Pengaman Kunci Sepeda Motor Menggunakan Sidik Jari Berbasis Mikrokontroler maka didapatkan kesimpulan sebagai berikut:

1. *Prototype* Sistem Pengaman Kunci Sepeda Motor Menggunakan Sidik Jari Berbasis Mikrokontroler dirancang menggunakan Arduino Nano, Kunci Kontak, Keypad dan Fingerprint Sensor.
2. Dari Pengujian Sidik Jari Terdaftar sensor sidik jari berhasil membaca semua sidik jari yang terdaftar tanpa kendala. Dari Pengujian Penambahan Sidik Jari Baru dapat disimpulkan bahwa alat yang dirancang dapat menambahkan sidik jari baru dan dapat digunakan untuk menyalakan lampu indicator.

Dari semua pengujian dapat disimpulkan bahwa alat bekerja dengan baik.

#### **D. Saran**

Saran yang dapat dilakukan untuk pengembangan *Prototype* Sistem Pengaman Kunci Sepeda Motor Menggunakan Sidik Jari Berbasis Mikrokontroler adalah dengan penambahan sensor-sensor dan membuatnya menjadi otomatis tanpa menggunakan kunci kontak lagi

## DAFTAR PUSTAKA

- Gazali, W., dan A. A. S. Gunawan. 2019. Analisis dan Pembuatan Sistem Pengenalan Sidik Jari Berbasis Komputer Di Polda Metro Jaya. *Jurnal Mat Stat* 12(1): 55-65.
- H. A. Adidharma, "PERANCANGAN KUNCI KONTAK DAN *ELECTRIC STARTER* SEPEDA MOTOR DENGAN FINGERPRINT SENSOR BERBASIS ARDUINO" J.T. Elektro Universitas Kristen Satya Wacana pp. 2019.
- Handoko, P. (2017). Sistem Kendali Perangkat Elektronika Monolitik Berbasis Arduino Nano. *Prosiding Semnastek*.
- Hidayat, R. 2018. Teknologi Wireless RFID Untuk Perpustakaan Polnes: Suatu Peluang. *Jurnal Informatika Mulawarman* 5(1): 42-49.
- Isyanto, H., H. Muchtar, dan J. Burhan. 2017. Perancangan Security System Kendaraan Menggunakan Finger Print. *Jurnal eLEKTUM* 12(1): 1-4.
- Susanto, R., A. Ananta, A. Santoso, dan M. Trianto. 2018. Sistem Absensi Berbasis RFID. *Jurnal Teknik Komputer* 17(1): 67-74.
- Syam, R., M. Hariadi, dan M.H. Purnama. 2020. Penentuan Nilai Standar Distorsi Berminyak Pada Akuisisi Citra Sidik Jari. *Makara, Teknologi* 15(1): 55-62
- Wilson. 2017. *PN532 NFC RFID Module User Guide*. Edisi 3. [www.elechouse.com](http://www.elechouse.com). 5 Februari 2019 (21.30).
- Wulan Veronikaa Samosir, Penganalisis Sistem Sensor Pintu Menggunakan Sidik Jari Berbasis Ardiuno Jakarta : UNIVERSITAS PERSADA INDONESIA Y.A.I 2020. (FINGERPRINT)