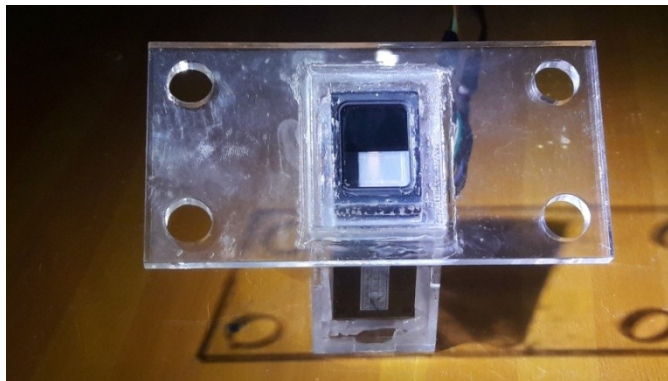


BAB IV

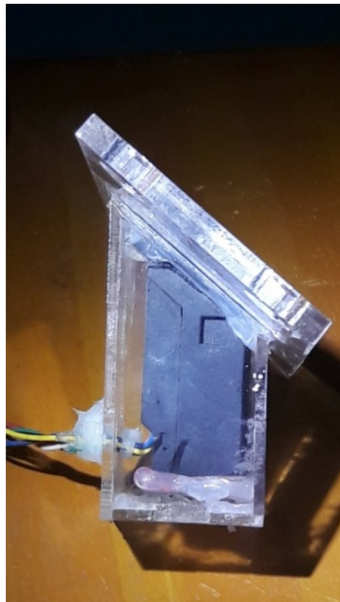
HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

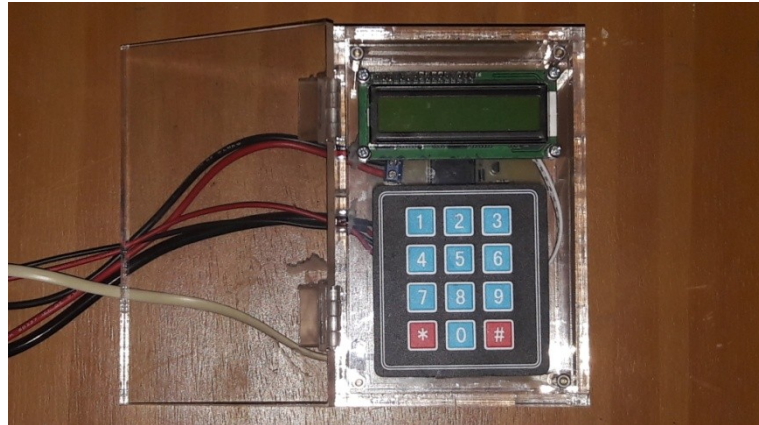
Berdasarkan penjelasan mengenai blok diagram, *flowchart* dan rangkaian yang telah dirancang pada penjelasan sebelumnya, maka hasil penelitian sistem pengaman sepeda motor dengan mengendalikan sistem kelistrikan dan pengereman menggunakan sidik jari di implementasikan oleh peneliti pada gambar 4.1 sampai 4.4.



Gambar 4.1. *Fingerprint sensor* tampak atas



Gambar 4.2. *Fingerprint sensor* tampak samping



Gambar 4.3. Kontroller dan komponen lainnya



Gambar 4.4. Aktuator dan sambungannya

Hasil dari sistem pengaman sepeda motor dengan mengendalikan sistem kelistrikan dan pengereman menggunakan sidik jari dilakukan pengujian, dimana pengujian tersebut dimaksudkan untuk mengetahui hasil kerja alat apakah bekerja dengan baik atau tidak. Pengujian dilakukan pada tiap-tiap blok sistem pengaman sepeda motor dengan mengendalikan sistem kelistrikan dan pengereman menggunakan sidik jari.

4.1.1. Pengujian *Keypad*

Pengujian *keypad* dilakukan untuk mengetahui data yang diterima oleh mikrokontroler apakah sesuai dengan karakter yang ditekan pada *keypad*. Pengujian *keypad* dilakukan dengan cara mengamati karakter yang tampil pada serial monitor arduino IDE ketika dilakukan penekanan pada *keypad*. Untuk

melakukan pengujian diperlukan peralatan dan melakukan prosedur pengujian sebagai berikut.

4.1.1.1. Peralatan

Peralatan pengujian yang digunakan adalah sebagai berikut:

- Arduino nano
- Keypad 3x4
- Komputer yang terinstal arduino IDE

4.1.1.2. Prosedur

Prosedur pengujian yang dilakukan adalah sebagai berikut:

- Menghubungkan *keypad* dengan *board* arduino nano dan menghubungkannya dengan komputer.
- Meng-*upload* program ke mikrokontroler dengan *sketch* seperti pada gambar 4.5.



```
PENGUJIAN_KEYPAD | Arduino 1.8.2
File Edit Sketch Tools Help
PENGUJIAN_KEYPAD
1 #include <Keypad.h>
2 char tombol[4][3] =
3 {
4   {'1', '2', '3'},
5   {'4', '5', '6'},
6   {'7', '8', '9'},
7   {'*', '0', '#'}
8 };
9 byte pinBaris[] = {4,5,6,7};
10 byte pinKolom[] = {8,9,10};
11 Keypad keypad = Keypad(makeKeymap(tombol),
12   pinBaris, pinKolom, 4, 3);
13 void setup()
14 { Serial.begin(9600);}
15 void loop()
16 {
17   char kode = keypad.getKey();
18   if (kode != NO_KEY)
19     Serial.println(kode);
20 }
21
Done compiling.
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 3124 bytes (10%) of program storage space. Maximum is 30720 bytes.
Global variables use 302 bytes (14%) of dynamic memory, leaving 1746 bytes for local variables. Max
21
Arduino Nano, ATmega328 on COM3
```

Gambar 4.5. Program pengujian keypad

- Membuka serial monitor arduino IDE

- Menekan tombol *keypad* dan mengamati data yang tampil pada serial monitor.

4.1.1.3. Hasil Pengujian

Hasil pengujian yang telah dilakukan ditunjukkan dalam tabel 4.1:

Tabel 4.1. Data pengujian *keypad*

| Tombol yang ditekan | Karakter di serial monitor arduino IDE |
|---------------------|--|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| * | * |
| 0 | 0 |
| # | # |

Data pada tabel diatas merupakan hasil dari pengujian *keypad* yang dilakukan, dari data diatas dapat diketahui bahwa data yang diterima mikrokontroler sesuai dengan karakter yang ditekan pada *keypad*.

4.1.2. Pengujian LCD

Pengujian LCD dilakukan untuk mengetahui apakah lcd dapat menampilkan karakter yang sesuai dengan karakter yang dikirimkan oleh mikrokontroler. Pengujian dilakukan dengan cara mengamati karakter yang tampil pada lcd setelah mengupload program pengujian. Untuk melakukan pengujian diperlukan peralatan dan melakukan prosedur pengujian sebagai berikut.

4.1.2.1. Peralatan

Peralatan pengujian yang dilakukan adalah sebagai berikut:

- Arduino nano
- LCD 16x2

- Komputer yang terinstal arduino IDE

4.1.2.2. Prosedur

Prosedur yang dilakukan adalah sebagai berikut:

- Menghubungkan LCD dengan *board* arduino nano melalui LCD I2c dan menghubungkannya dengan komputer.
- Meng-*upload* program ke mikrokontroler dengan *sketch* seperti pada gambar 4.6.

```

PENGUJIAN_LCD | Arduino 1.8.2
File Edit Sketch Tools Help
PENGUJIAN_LCD
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27,16,2);
4 void setup()
5 {
6   lcd.begin();
7   lcd.print("Tes..tes..");
8   lcd.setCursor(0, 1);
9   lcd.print("OK tampaknya!");
10 }
11 void loop()
12 {
13 }
14 }
15 }
Done compiling.
Using library Arduino-LiquidCrystal-I2C-library-master in folder: C:\Program Files (x86)\Arduino\lib
Sketch uses 2846 bytes (9%) of program storage space. Maximum is 30720 bytes.
Global variables use 239 bytes (11%) of dynamic memory, leaving 1809 bytes for local variables. Max
15 Arduino Nano, ATmega328 on COM3

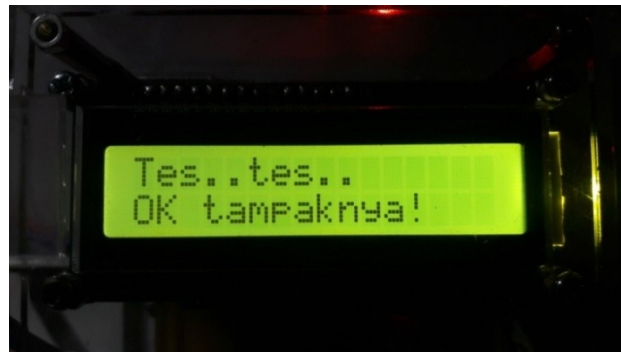
```

Gambar 4.6. Program pengujian lcd

- Mengamati karakter yang tampil pada layar LCD apakah sesuai dengan karkater yang ditulis pada sketch.

4.1.2.3. Hasil Pengujian

Gambar 4.7 merupakan gambar hasil pengujian LCD.



Gambar 4.7. Pengujian LCD

Gambar 4.7 merupakan hasil dari pengujian lcd, dapat diketahui dari gambar tersebut bahwa lcd menampilkan karakter yang sesuai dengan karakter yang dikirimkan oleh mikrokontroler.

4.1.3. Pengujian Sensor Sidik Jari

Pengujian Sensor Sidik Jari dilakukan untuk mengetahui apakah sensor sidik jari dapat digunakan untuk pendaftaran dan pembacaan sidik jari. Pengujian sensor sidik jari dilakukan dengan cara mendaftarkan sidik jari kemudian dilakukan pembacaan untuk sidik jari yang telah didaftarkan. Untuk melakukan pengujian diperlukan peralatan dan melakukan prosedur pengujian sebagai berikut.

4.1.3.1. Peralatan

Peralatan pengujian yang digunakan adalah sebagai berikut:

- Arduino nano
- Sensor Sidik Jari
- Komputer yang terinstal arduino IDE

4.1.3.2. Prosedur

Prosedur yang dilakukan adalah sebagai berikut:

- Menghubungkan sensor sidik jari dengan *board* arduino nano dan menghubungkannya dengan komputer.
- Meng-*upload* program untuk mendaftarkan sidik jari ke mikrokontroler dengan *sketch* pada gambar 4.8.

```

1 #include <Adafruit_Fingerprint.h>
2 #include <SoftwareSerial.h>
3 SoftwareSerial mySerial(2, 3);
4 Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
5 uint8_t id;
6 void setup()
7 {
8   Serial.begin(9600);
9   finger.begin(57600);
10  if (finger.verifyPassword()) {
11    Serial.println("sensor fingerprint ditemukan");
12  }
13 }
14 uint8_t readnumber(void) {
15   uint8_t num = 0;
16   while (num == 0) {
17     while (! Serial.available());
18     num = Serial.parseInt();
19   }
20   return num;
21 }
22 void loop()
23 {
24   Serial.println("Siap untuk mendaftarkan");
25   Serial.println("Masukan nomor ID # (1 sampai 9)");
26   id = readnumber();
27   if (id == 0) {return;}
28   Serial.print("Mendaftarkan ID #");
29   Serial.println(id);
30   while (! getFingerprintEnroll() );
31 }
32 uint8_t getFingerprintEnroll() {
33   int p = -1;

```

Done compiling

Using library SoftwareSerial at version 1.0 in folder: C:\Program Files (x86)\Arduino\hardware

Sketch uses 6220 bytes (20%) of program storage space. Maximum is 30720 bytes.
Global variables use 782 bytes (38%) of dynamic memory, leaving 1266 bytes for local variables.

Arduino Nano, ATmega328 on COM3

(a) Bagian 1

```

34   Serial.print("Tempelkan sidik jari untuk pendaftaran ID #"); Serial.println(id);
35   while (p != FINGERPRINT_OK) {
36     p = finger.getImage();
37     switch (p) {
38       case FINGERPRINT_OK:
39         Serial.println("Mendapatkan gambar");
40         break;
41       case FINGERPRINT_NOFINGER:
42         //Serial.println(".");
43         Serial.print(".");
44         break;
45       case FINGERPRINT_PACKETRECEIVEERR:
46         Serial.println("Komunikasi error");
47         break;
48       case FINGERPRINT_IMAGEFAIL:
49         Serial.println("Gambar error");
50         break;
51       default:
52         Serial.println("Tidak diketahui");
53         break;
54     }
55   }
56   // OK success!
57   p = finger.image2Tz(1);
58   switch (p) {
59     case FINGERPRINT_OK:
60       Serial.println("Mengkonversi gambar");
61       break;
62     case FINGERPRINT_IMAGEERR:
63       Serial.println("Gambar kacau");
64       return p;
65     case FINGERPRINT_PACKETRECEIVEERR:
66       Serial.println("Komunikasi error");

```

Done compiling

Using library SoftwareSerial at version 1.0 in folder: C:\Program Files (x86)\Arduino\hardware

Sketch uses 6220 bytes (20%) of program storage space. Maximum is 30720 bytes.
Global variables use 782 bytes (38%) of dynamic memory, leaving 1266 bytes for local variables.

Arduino Nano, ATmega328 on COM3

(b) Bagian 2

```
67     return p;
68   case FINGERPRINT_FEATUREFAIL:
69     Serial.println("Tidak dapat menemukan fitur");
70     return p;
71   case FINGERPRINT_INVALIDIMAGE:
72     Serial.println("Tidak dapat menemukan fitur");
73     return p;
74   default:
75     Serial.println("Tidak diketahui");
76     return p;
77 }
78 Serial.println("Menghapus sidik jari");
79 delay(2000);
80 p = 0;
81 while (p != FINGERPRINT_NOFINGER) {
82   p = finger.getImage();
83 }
84 Serial.print("ID "); Serial.println(id);
85 p = -1;
86 Serial.println("Tempelkan sidik jari lagi");
87 while (p != FINGERPRINT_OK) {
88   p = finger.getImage();
89   switch (p) {
90     case FINGERPRINT_OK:
91       Serial.println("Mendapatkan gambar");
92       break;
93     case FINGERPRINT_NOFINGER:
94       Serial.print(".");
95       break;
96     case FINGERPRINT_PACKETRECEIVEERR:
97       Serial.println("Komunikasi error");
98       break;
99     case FINGERPRINT_IMAGEFAIL:
```

Done compiling

Using library SoftwareSerial at version 1.0 in folder: C:\Program Files (x86)\Arduino\hardware\ Sketch uses 6220 bytes (20%) of program storage space. Maximum is 30720 bytes. Global variables use 782 bytes (38%) of dynamic memory, leaving 1266 bytes for local variables.

Arduino Nano, ATmega328 on COM3

(c) Bagian 2

```
67     return p;
68   case FINGERPRINT_FEATUREFAIL:
69     Serial.println("Tidak dapat menemukan fitur");
70     return p;
71   case FINGERPRINT_INVALIDIMAGE:
72     Serial.println("Tidak dapat menemukan fitur");
73     return p;
74   default:
75     Serial.println("Tidak diketahui");
76     return p;
77 }
78 Serial.println("Menghapus sidik jari");
79 delay(2000);
80 p = 0;
81 while (p != FINGERPRINT_NOFINGER) {
82   p = finger.getImage();
83 }
84 Serial.print("ID "); Serial.println(id);
85 p = -1;
86 Serial.println("Tempelkan sidik jari lagi");
87 while (p != FINGERPRINT_OK) {
88   p = finger.getImage();
89   switch (p) {
90     case FINGERPRINT_OK:
91       Serial.println("Mendapatkan gambar");
92       break;
93     case FINGERPRINT_NOFINGER:
94       Serial.print(".");
95       break;
96     case FINGERPRINT_PACKETRECEIVEERR:
97       Serial.println("Komunikasi error");
98       break;
99     case FINGERPRINT_IMAGEFAIL:
```

Done compiling

Using library SoftwareSerial at version 1.0 in folder: C:\Program Files (x86)\Arduino\hardware\ Sketch uses 6220 bytes (20%) of program storage space. Maximum is 30720 bytes. Global variables use 782 bytes (38%) of dynamic memory, leaving 1266 bytes for local variables.

Arduino Nano, ATmega328 on COM3

(d) Bagian 4


```

PENGUIAN_SIDI_JARI_PENDAFTARAN_
...
129 // OK converted!
130 Serial.print("Membuat model untuk ID #"); Serial.println(id);
131 p = finger.createModel();
132 if (p == FINGERPRINT_OK) {
133   Serial.println("Cetakan Cocok");
134 } else if (p == FINGERPRINT_FACRETRCIEVEERR) {
135   Serial.println("Komunikasi error");
136   return p;
137 } else if (p == FINGERPRINT_ENROLLMISMATCH) {
138   Serial.println("Sidik jari tidak cocok");
139   return p;
140 } else {
141   Serial.println("Tidak diketahui");
142   return p;
143 }
144 Serial.print("ID "); Serial.println(id);
145 p = finger.storeModel(id);
146 if (p == FINGERPRINT_OK) {
147   Serial.println("Tersimpan");
148 } else if (p == FINGERPRINT_FACRETRCIEVEERR) {
149   Serial.println("Komunikasi error");
150   return p;
151 } else if (p == FINGERPRINT_BADLOCATION) {
152   Serial.println("Tidak dapat menyimpan");
153   return p;
154 } else if (p == FINGERPRINT_FLASHERR) {
155   Serial.println("Kesalahan penulisan");
156   return p;
157 } else {
158   Serial.println("Tidak diketahui");
159   return p;
160 }
161 }
Done compiling
Using library SoftwareSerial at version 1.0 in folder: C:\Program Files (x86)\Arduino\hardware
Sketch uses 6220 bytes (20%) of program storage space. Maximum is 30720 bytes.
Global variables use 782 bytes (38%) of dynamic memory, leaving 1266 bytes for local variables.
Arduino Nano, ATmega328 on COM3

```

(e) Bagian 5

Gambar 4.8. program pengujian pendaftaran sidik jari

- Membuka serial monitor arduino IDE
- Memberikan nomor id untuk sidik jari yang akan didaftarkan dengan memasukan angka pada serial monitor.
- Menempelkan jari pada sensor sidik jari 2 kali
- Mengamati pada serial monitor untuk mengetahui apakah pola sidik jari berhasil tersimpan. Pola sidik jari berhasil tersimpan ditandai dengan kata “tersimpan” pada serial monitor.
- Setelah berhasil melakukan pendaftaran sidik jari selanjutnya adalah membaca sidik jari yang telah didaftarkan.
- Meng-*upload* program untuk membaca sidik jari ke mikrokontroler dengan *sketch* seperti pada gambar 4.9.

```

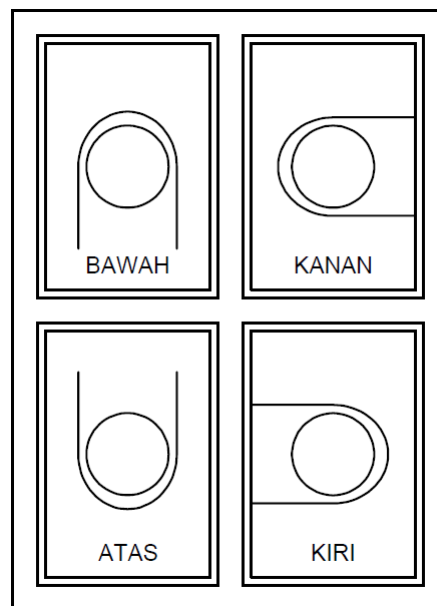
PENGUJIAN_SIDIK_JARI_PEMBACAAN_
1 #include <Adafruit_Fingerprint.h>
2 #include <SoftwareSerial.h>
3 SoftwareSerial mySerial(2, 3);
4 Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
5 void setup()
6 {
7   Serial.begin(9600);
8   finger.begin(57600);
9   if (finger.verifyPassword())
10  {
11    Serial.println("sensor fingerprint ditemukan");
12  }
13 }
14 void loop()
15 {
16   getFingerprintIDez();
17   delay(50);
18 }
19 int getFingerprintIDez()
20 {
21   uint8_t p = finger.getImage();
22   if (p != FINGERPRINT_OK) return -1;
23   p = finger.image2Tz();
24   if (p != FINGERPRINT_OK) return -1;
25   p = finger.fingerFastSearch();
26   if (p != FINGERPRINT_OK) return -1;
27   Serial.print("Ditemukan ID #"); Serial.print(finger.fingerID);
28   Serial.print("Keakuratan = "); Serial.println(finger.confidence);
29   return finger.fingerID;
30 }
Done compiling.
Using library SoftwareSerial at version 1.0 in folder: C:\Program Files (x86)\Arduino\hardware\arduino\avr\li
Sketch uses 4882 bytes (15%) of program storage space. Maximum is 30720 bytes.
Global variables use 405 bytes (19%) of dynamic memory, leaving 1643 bytes for local variables. Maximum is 20

```

Gambar 4.9. program pengujian pembacaan sidik jari

- Menempelkan sidik jari yang telah terdaftar
- Menempelkan sidik jari yang tidak terdaftar
- Menempelkan sidik jari yang telah terdaftar dari 4 arah penempelan sidik jari.

Untuk ilustrasinya dapat dilihat pada gambar 4.10.



Gambar 4.10. Ilustrasi penempelan sidik jari pada sensor

- Proses pembacaan berhasil ditandai dengan tampilnya nomor id dan akurasi pembacaan pada serial monitor.

4.1.3.3. Hasil Pengujian

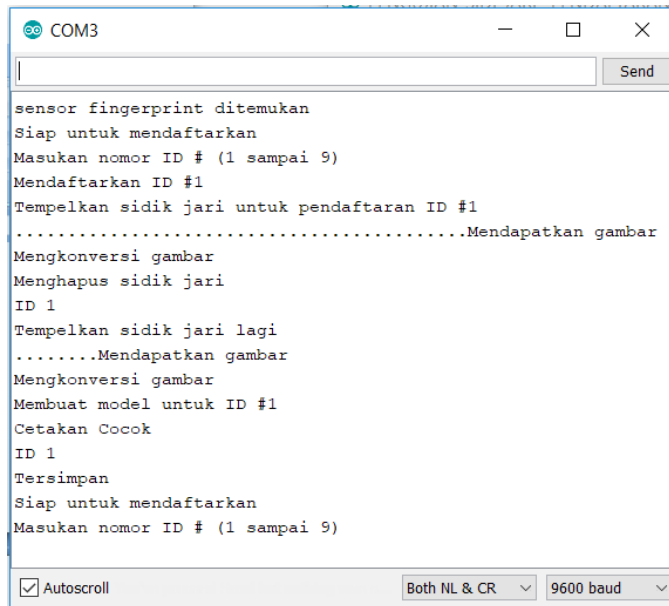
Hasil yang didapatkan pada pengujian sidik jari adalah hasil pengujian pendaftaran sidik jari dan pembacaan sidik jari. Pada pengujian pembacaan sidik jari dilakukan dengan dua cara yaitu pembacaan sidik jari yang telah terdaftar dan sidik jari yang belum terdaftar. Pada pengujian pembacaan sidik jari diasumsikan jari tangan kanan adalah sidik jari yang telah terdaftar dan untuk jari tangan kiri merupakan sidik jari yang belum terdaftar.

Hasil pengujian pendaftaran sidik jari ditunjukkan dalam tabel 4.2:

Tabel 4.2. Data pengujian pendaftaran sidik jari

| Sampel | Kriteria | Nomor id | Hasil pengujian |
|--------|-------------------|----------|-----------------|
| 1 | Jempol kanan | 1 | Tersimpan |
| 2 | Telunjuk kanan | 2 | Tersimpan |
| 3 | Jari tengah kanan | 3 | Tersimpan |
| 4 | Jari manis kanan | 4 | Tersimpan |
| 5 | Kelingking kanan | 5 | Tersimpan |

Data pada tabel 4.2 merupakan hasil dari pendaftaran sidik jari, dimana semua percobaan pendaftaran menggunakan tangan kanan, dari data tersebut dapat diketahui bahwa sensor sidik jari dapat melakukan pendaftaran sidik jari. Proses pendaftaran sidik jari berhasil ditandai dengan notifikasi tersimpan yang tampil pada serial monitor, untuk lebih jelasnya dapat dilihat pada gambar 4.11.



Gambar 4.11 Notifikasi pada serial monitor pada saat pendaftaran sidik jari

Hasil pengujian pembacaan sidik jari dapat dilihat pada tabel 4.3 dan 4.4.

Tabel 4.3. Data pengujian pembacaan sidik jari yang telah terdaftar

| Sampel | Kriteria | Pengujian sensor sidik jari | | | Hasil pengujian |
|--------|-------------------|-----------------------------|--------------|--------------|-----------------|
| | | Akurasi ke-1 | Akurasi ke-2 | Akurasi ke-3 | |
| 1 | Jempol kanan | 73 | 108 | 108 | Terbaca |
| 2 | Telunjuk kanan | 108 | 108 | 108 | Terbaca |
| 3 | Jari tengah kanan | 108 | 108 | 92 | Terbaca |
| 4 | Jari manis kanan | 106 | 108 | 108 | Terbaca |
| 5 | Kelingking kanan | 108 | 104 | 108 | Terbaca |

Tabel 4.4. Data pengujian pembacaan sidik jari yang tidak terdaftar

| Sampel | Kriteria | Pengujian sensor sidik jari | | | Hasil pengujian |
|--------|----------|-----------------------------|---------|---------|-----------------|
| | | Akurasi | Akurasi | Akurasi | |

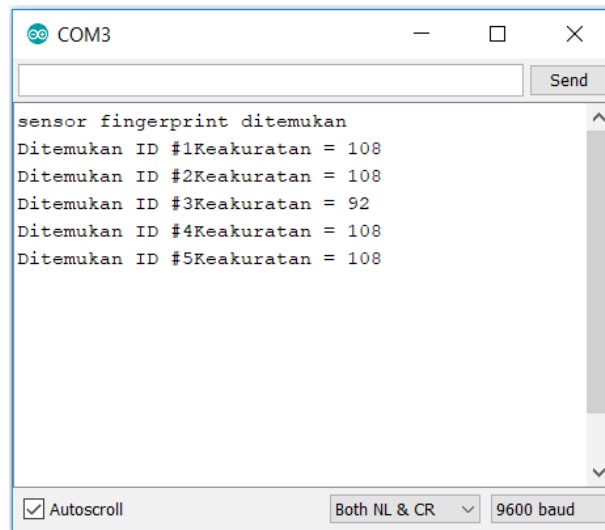
| | | ke-1 | ke-2 | ke-3 | |
|---|---------------------|------|------|------|------------------|
| 1 | Jempol kiri | - | - | - | Tidak terbaca |
| 2 | Telunjuk kiri | - | - | - | Tidak terbaca |
| 3 | Jari tengah kiri | - | - | - | Tidak terbaca |
| 4 | Jari manis kiri | - | - | - | Tidak terbaca |
| 5 | Kelingkin g kiri | - | - | - | Tidak terbaca |

Data pada tabel 4.3 merupakan data hasil pembacaan sensor sidik jari , dari data tersebut dapat diketahui bahwa sensor sidik jari dapat mengidentifikasi sidik jari yang telah terdaftar. Dari tabel 4.3 tersebut juga menunjukkan nilai akurasi sidik jari yang terbaca. Dari nilai akurasi yang terbaca, nilai yang paling tinggi adalah sebesar 108 dan nilai akurasi yang terkecil adalah 73.

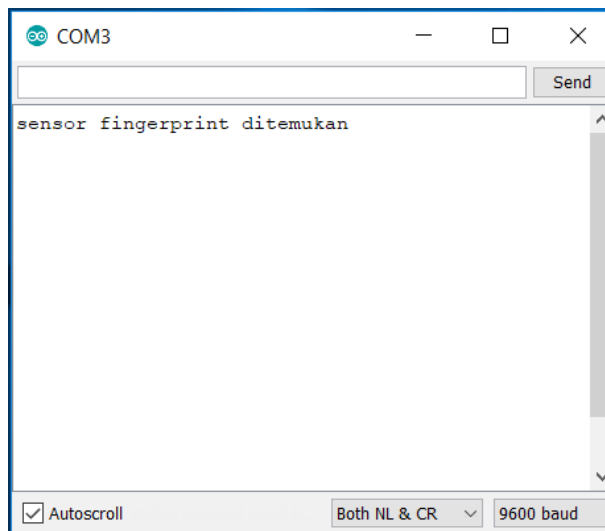
Nilai akurasi adalah nilai yang menunjukkan berapa banyak pola sidik jari yang cocok ketika dilakukan pembacaan sidik jari.

Data pada tabel 4.4 merupakan data hasil pembacaan sidik jari, dari data tersebut dapat diketahui bahwa sensor sidik jari tidak dapat mengidentifikasi sidik jari yang tidak terdaftar.

Untuk mengetahui sensor sidik jari dapat mengidentifikasi sidik jari atau tidak dapat diamati pada serial monitor ketika dilakukan pembacaan sidik jari. Untuk lebih jelasnya dapat dilihat pada gambar 4.12 dan 4.13.







Gambar 4.12 Notifikasi pada serial monitor pada saat pembacaan sidik jari yang terdaftar

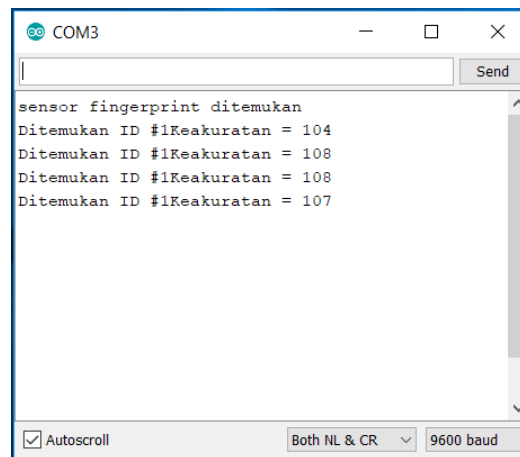


Gambar 4.13 Notifikasi pada serial monitor pada saat pembacaan sidik jari yang tidak terdaftar

Hasil pengujian pembacaan sidik jari berdasarkan arah penempelan sidik jari dapat dilihat pada tabel 4.5.

Tabel 4.5. Data pengujian pembacaan sidik jari berdasarkan arah penempelan sidik jari

| Sampel | Arah penempelan | Akurasi | Hasil pengujian | Gambar |
|--------------|-----------------|---------|-----------------|---|
| Jempol kanan | Bawah | 104 | Terbaca |  |
| Jempol kanan | Kanan | 108 | Terbaca |  |
| Jempol kanan | Atas | 108 | Terbaca |  |
| Jempol kanan | Kiri | 107 | Terbaca |  |



Gambar 4.14 Notifikasi pada serial monitor pada saat pembacaan sidik jari berdasarkan arah penempelan sidik jari

4.1.4. Pengujian *Relay* Kontak

Pengujian *relay* kontak dilakukan untuk mengetahui apakah *relay* bekerja sesuai dengan perintah mikrokontroler. Pengujian dilakukan dengan cara mengamati pin *output relay* menggunakan multimeter. Untuk melakukan pengujian diperlukan peralatan dan melakukan prosedur pengujian sebagai berikut.

4.1.4.1. Peralatan

Peralatan pengujian yang digunakan adalah sebagai berikut:

- Arduino nano
- Modul *relay*
- Komputer yang terinstal arduino IDE
- Multimeter

4.1.4.2. Prosedur

Prosedur yang dilakukan adalah sebagai berikut:

- Menghubungkan modul *relay* dengan board arduino nano dan menghubungkannya dengan komputer.
- Meng-*upload* program ke mikrokontroler dengan *sketch* seperti pada gambar 4.15 dan 4.16.


```
PENGUJIAN_RELAY_KONTAK_HIGH
1 int relayKontak = 12;
2 void setup()
3 {
4   pinMode(relayKontak, OUTPUT);
5 }
6
7 void loop()
8 {
9   digitalWrite(relayKontak, HIGH);
10 }
11
```

Done compiling.

"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -R .eeprom "C:\Users\M...
Sketch uses 724 bytes (2%) of program storage space. Maximum is 30720 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum

11 Arduino Nano, ATmega328 on COM3

Gambar 4.15. Program pengujian *relay* kondisi *HIGH*

```
PENGUJIAN_RELAY_KONTAK_LOW_
1 int relayKontak = 12;
2 void setup()
3 {
4   pinMode(relayKontak, OUTPUT);
5 }
6
7 void loop()
8 {
9   digitalWrite(relayKontak, LOW);
10 }
11
```

Done compiling.

"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -R .eeprom "C:\Users\M...
Sketch uses 726 bytes (2%) of program storage space. Maximum is 30720 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum

11 Arduino Nano, ATmega328 on COM3

Gambar 4.16. Program pengujian *relay* kondisi *LOW*

- Mengamati apakah terjadi perubahan kondisi pada pin NO dan NC pada *relay* dengan menggunakan multimeter.

4.1.4.3. Hasil Pengujian

Hasil pengujian *relay* kontak yang telah dilakukan ditunjukkan dalam tabel 4.6:

Tabel 4.6. Data pengujian *relay* kontak

| Kondisi output mikrokontroler | Pin Com dan NC | Pin Com dan NO | Hasil pengujian |
|-------------------------------|-----------------|-----------------|-----------------|
| <i>HIGH</i> | Tidak terhubung | Terhubung | Berhasil |
| <i>LOW</i> | Terhubung | Tidak terhubung | Berhasil |

Data pada 4.6 di atas merupakan data hasil pengujian *relay*, dari data tersebut dapat diketahui bahwa kerja *relay* sesuai dengan perintah mikrokontroler.

4.1.5. Pengujian Aktuator

Pengujian Aktuator dilakukan untuk mengetahui apakah aktuator dapat melakukan kerjanya sesuai dengan perintah mikrokontroler. Pengujian dilakukan dengan mengamati pergerakan aktuator setelah program di *upload* ke mikrokontroler. Untuk melakukan pengujian diperlukan peralatan dan melakukan prosedur pengujian sebagai berikut.

4.1.5.1. Peralatan

Peralatan pengujian yang digunakan adalah sebagai berikut:

- Arduino nano
- Aktuator
- Modul *relay*
- Power supply
- Komputer yang terinstal arduino IDE

4.1.5.2. Prosedur

Prosedur pengujian yang dilakukan adalah sebagai berikut:

- Menghubungkan aktuator dengan modul *relay* kemudian hubungkan modul *relay* dengan board arduino nano.
- Menghubungkan rangkaian aktuator dengan computer dan power supply
- Meng-*upload* program ke mikrokontroler dengan *sketch* seperti pada gambar berikut berikut 4.17 dan 4.18.

```

PENGUJIAN_AKTUATOR
1 int relayAktuator = 11;
2 void setup()
3 {
4   pinMode(relayAktuator, OUTPUT);
5 }
6
7 void loop()
8 {
9   digitalWrite(relayAktuator, HIGH);
10 }

```

Done compiling.

"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -R .eeprom "C:\Users\M...
Sketch uses 724 bytes (2%) of program storage space. Maximum is 30720 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum

10

Arduino Nano, ATmega328 on COM3

Gambar 4.17. Program pengujian aktuator kondisi *HIGH*



Gambar4.18. Program pengujian aktuator kondisi *LOW*

- Mengamati apakah aktuator bergerak maju dan mundur

4.1.5.3. Hasil Pengujian

Hasil pengujian yang telah dilakukan ditunjukkan dalam tabel 4.7:

Tabel 4.7. Data pengujian aktuator

| Kondisi output mikrokontroler | Pergerakan aktuator |
|-------------------------------|---------------------|
| <i>HIGH</i> | Memendek |
| <i>LOW</i> | Memanjang |

Data pada tabel 4.7 merupakan hasil dari pengujian aktuator, dari data tersebut dapat diketahui bahwa ketika output mikrokontroler bernilai *HIGH* maka aktuator akan bergerak memendek, sedangkan pada kondisi output *LOW* aktuator bergerak memanjang. Dengan rangkaian *relay* yang telah dibuat dan hasil pengujian ini dapat diketahui bahwa aktuator dapat bekerja sesuai dengan yang diperintahkan mikrokontroler.

4.1.6. Pengujian Arus

Pengujian arus dilakukan untuk mengetahui besarnya arus yang masuk ke rangkaian dan aktuator. Pengujian arus dilakukan dengan cara mengukur arus yang masuk ke rangkaian dan aktuator. Untuk melakukan pengujian diperlukan peralatan dan melakukan prosedur pengujian sebagai berikut.

4.1.6.1. Peralatan

Peralatan pengujian yang digunakan adalah sebagai berikut:

- Rangkaian yang telah dibuat
- Multimeter
- Sepeda motor

4.1.6.2. Prosedur

Prosedur pengujian yang dilakukan adalah sebagai berikut:

- Merangkai secara seri multimeter dengan rangkaian dan aktuator pada bagian input tegangan
- Mengamati nilai arus yang mengalir ketika rangkaian menyala

4.1.6.3. Hasil Pengujian

Hasil pengujian yang telah dilakukan ditunjukkan dalam tabel 4.8.

Tabel 4.8. Data pengukuran arus

| Jenis pengukuran | Kondisi | Arus terukur |
|------------------|--------------------------|--------------|
| Rangkaian | <i>Relay</i> belum aktif | 75,7 mA |
| | <i>Relay</i> aktif | 213,7 mA |
| Aktuator | Tidak bergerak | 0 A |
| | bergerak | 0,651 A |

Dari data pada tabel 4.8 merupakan data pengukuran arus yang telah dilakukan. Pada pengujian ini pengukuran arus dilakukan pada dua komponen yaitu rangkaian dan aktuator. Pada pengukuran arus rangkaian, pengukuran dilakukan pada kondisi relay belum aktif dengan arus yang terukur adalah 75,7 mA dan pada kondisi relay aktif dengan arus terukur adalah 213,7 mA. Kemudian untuk pengukuran arus aktuator dilakukan pengukuran pada dua kondisi yaitu pada

kondisi aktuator tidak bergerak dengan arus terukur adalah 0 A dan pada saat aktuator bergerak dengan arus terukur adalah 0,651 A.

Dari hasil pengukuran arus yang telah dilakukan dapat dihitung berapa lama baterai sepeda motor dapat menyuplai arus pada rangkaian dengan asumsi hanya baterai yang digunakan untuk sumber energinya.

Diketahui:

- 1) Baterai sepeda motor:
 - Tegangan : 12 VDC
 - Kapasitas : 3 Ah
- 2) Aktuator:
 - Tegangan : 12 VDC
 - Arus : 0,651 A
 - Kecepatan : 90 MM/S
 - Perpanjangan : 50 MM
- 3) Rangkaian (Arduino Nano, *relay*, LCD, dll)
 - Tegangan : 12 V
 - Arus : 213,7 mA

Energi baterai:

$$E = 12 \times 3 \text{ Ah}$$

$$E = 36 \text{ Wh}$$

Waktu yang dibutuhkan aktuator untuk melakukan perpanjangan:

$$(90/1) = (50/t)$$

$$t = (50/90)$$

$$t = 0,55 \text{ detik}$$

$$t = 0,00015 \text{ jam}$$

jadi waktu yang dibutuhkan aktuator adalah 0,00015 jam

Energi aktuator:

$$E = (12 \times 0,651 \text{ A}) \times 0,00015$$

$$E = 7,812 \times 0,00015$$

$$E = 0,00117 \text{ Wh}$$

Energi rangkaian:

$$E = (12 \times 0,2137) \times 1 \text{ jam}$$

$$E = 2,56 \text{ Wh}$$

Perkiraan masa penggunaan batrai:

$$\begin{aligned} \text{Lama penggunaan} &= (36 \text{ Wh} - 0,00117 \text{ Wh}) / 2,56 \text{ Wh} \\ &= 14,06 \text{ jam} \end{aligned}$$

Dari perhitungan di atas dapat diketahui masa penggunaan batrai adalah 14,06 jam.

Sumber energi listrik yang ada pada sepeda motor tidak hanya berasal dari batrai sepeda motor melainkan dari generator sepeda motor juga. Batrai sepeda motor hanya digunakan untuk proses awal sebelum mesin pada sepeda motor menyala, setelah mesin sepeda motor menyala energi listrik disuplai oleh generator sepeda motor. generator pada sepeda motor selain digunakan untuk menyuplai energi listrik ke beban-beban pada sepeda motor juga digunakan untuk pengisian aki. Jika kondisi mesin sepeda motor dalam keadaan menyala sehingga sumber energi listrik yang digunakan adalah generator pada sepeda motor, maka masa penggunaan batrai akan lebih lama, karena selain generator yang menyuplai energi listrik pada rangkaian dan aktuator, juga generator melakukan pengisian energi listrik pada batrai.

4.1.7. Pengujian Sistem Secara Keseluruhan

Pengujian ini dilakukan untuk mengetahui apakah sistem dapat melakukan pengamanan terhadap sepeda motor. pengujian dilakukan dengan cara mengamati keadaan kelistrikan dan rem sepeda motor pada saat sebelum dan sesudah sidik jari diverifikasi oleh sistem. Untuk melakukan pengujian diperlukan peralatan dan melakukan prosedur pengujian sebagai berikut.

4.1.7.1. Peralatan

Peralatan pengujian yang digunakan adalah sebagai berikut:

- Sistem yang telah di rancang
- Sepeda motor
- Komputer yang terinstal arduino IDE

- Multimeter

4.1.7.2. Prosedur

Prosedur penelitian yang dilakukan adalah sebagai berikut:

- Meng-*upload* program ke mikrokontroler dengan *sketch* seperti pada gambar 4.10.



```
SISTEM_KESELURAHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURAHAN
1 #include <Adafruit_Fingerprint.h>
2 #include <SoftwareSerial.h>
3 SoftwareSerial mySerial(2, 3);
4 Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
5 char data;
6 char id;
7 #include <Wire.h>
8 #include <LiquidCrystal_I2C.h>
9 LiquidCrystal_I2C lcd(0x27, 16, 2);
10 #include <Keypad.h>
11 char tombol [4][3]=
12 {
13   {'1', '2', '3'},
14   {'4', '5', '6'},
15   {'7', '8', '9'},
16   {'*', '0', '#'}
17 };
18 byte pinBaris[]={4,5,6,7};
19 byte pinKolom[]={8,9,10};
20 Keypad keypad = Keypad(makeKeypad(tombol),pinBaris, pinKolom, 4,3);
21 int relayAktuator = 11;
22 int relayKontak = 12;
23 void setup()
24 {
25   lcd.begin();
26   Serial.begin(9600);
27
28   finger.begin(57600);
29   if (finger.verifyPassword()) {
30     Serial.println("sensor fingerprint ditemukan");
31     lcd.print("sensor fingerprint ditemukan");
32   }
33   pinMode(relayAktuator, OUTPUT);
Done compiling
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
Arduino Nano, ATmega328 on COM3
```

(a) Bagian 1


```
SISTEM_KESELURUHAN
34 pinMode(relayKontak, OUTPUT);
35 digitalWrite(relayAktuator, LOW);
36 digitalWrite(relayKontak, LOW);
37 }
38
39
40 void loop()
41 {
42   menu:
43   Serial.println("FINGERPRINT");
44   Serial.println("(+) MENU");
45   while(1)
46   {
47     lcd.clear();
48     lcd.setCursor(2,0);
49     lcd.print("FINGERPRINT");
50     lcd.setCursor(9,1);
51     lcd.println("(+)MENU");
52     data=keypad.getKey();
53     if (data != NO_KEY)
54       Serial.println(data);
55     if (data == '*'){delay(300);goto menu1;}
56     getFingerprintIdex();
57     delay(50);
58   }
59   menu1:
60   Serial.println("HOME");
61   Serial.println("(+) OK  (#) NEXT");
62   while(1)
63   {
64     lcd.clear();
65     lcd.setCursor(2,0);
66     lcd.print("HOME");
```

Done compiling

Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.

Arduino Nano, ATmega328 on COM3

(b) Bagian 2

```
SISTEM_KESELURUHAN
67   lcd.setCursor(0,1);
68   lcd.print("(+) OK  (#) NEXT");
69   data=keypad.getKey();
70   if (data != NO_KEY)
71     Serial.println(data);
72   if (data == '*'){delay(300);goto menu;}
73   if (data == '#'){delay(300);goto menu2;}
74   delay(200);
75 }
76 menu2:
77 Serial.println("DAFTAR");
78 Serial.println("(+) OK  (#) NEXT");
79 while(1)
80 {
81   lcd.clear();
82   lcd.setCursor(2,0);
83   lcd.print("DAFTAR");
84   lcd.setCursor(0,1);
85   lcd.print("(+) OK  (#) NEXT");
86   data=keypad.getKey();
87   if (data != NO_KEY)
88     Serial.println(data);
89   if (data == '*'){delay(300);goto menu3;}
90   if (data == '#'){delay(300);goto menu1;}
91   delay(200);
92 }
93 menu3:
94 while(1)
95 {
96   lcd.clear();
97   lcd.setCursor(2,0);
98   lcd.print("MENDAFTAR");
99   char data=keypad.getKey();
```

Done compiling

Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.

Arduino Nano, ATmega328 on COM3

(c) Bagian 3

```

SISTEM_KESELURUHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURUHAN
100 Serial.println(data);
101 if (data == ''){delay(300);goto menu;}
102 delay(200);
103 ///////////////////////////////////////////////////
104 Serial.println("Siap untuk mendaftarkan");
105 Serial.println("Masukan nomor ID # (1 sampai 9)");
106 lcd.clear();
107 lcd.setCursor(0,0);
108 lcd.print("Siap Untuk");
109 lcd.setCursor(0,1);
110 lcd.print("Mendaftarkan");
111 delay(1000);
112 lcd.clear();
113 lcd.setCursor(0,0);
114 lcd.print("Tekan Angka 1-9");
115 lcd.setCursor(0,1);
116 lcd.print("Untuk Selesai");
117 data=0;
118 while (data == 0)
119 {
120   data= keypad.getKey();
121   if (data != NO_KEY)
122     Serial.println(data);
123 }
124 if (data == ''){delay(300);goto menu;}
125 if (data == '0' || data == '*' || data == '#') {return;}
126 id = data;
127 Serial.print("Mendaftarkan ID #");
128 Serial.println(id);
129
130 lcd.clear();
131 lcd.print("Mendaftarkan ID#");
132 lcd.print(id);
Done compiling.
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
Arduino Nano, ATmega328 on COM3

```

(d) Bagian 4

```

SISTEM_KESELURUHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURUHAN
133 delay(1000);
134 getFingerprintEnroll();
135 }
136 }
137 //Baca sidik jari
138 int getFingerprintIdx() {
139   uint8_t p = finger.getImage();
140   if (p != FINGERPRINT_OK) return -1;
141   p = finger.image2Tz();
142   if (p != FINGERPRINT_OK) return -1;
143   p = finger.fingerFastSearch();
144   if (p != FINGERPRINT_OK) return -1;
145   // found a match!
146   uint8_t nomorID=finger.fingerID - 48;
147   Serial.print("Ditemukan ID #"); Serial.print(nomorID);
148   Serial.print("Keakuratan = "); Serial.println(finger.confidence);
149   digitalWrite(relayAktuator, HIGH);
150   digitalWrite(relayKontak, HIGH);
151   lcd.clear();
152   lcd.setCursor(0,0);
153   lcd.print("Ditemukan ID #");lcd.print(nomorID);
154   lcd.setCursor(0,1);
155   lcd.print("Keakuratan = ");lcd.print(finger.confidence);
156   delay(1000);
157 }
158 //daftar sidik jari
159 uint8_t getFingerprintEnroll() {
160   int p = -1;
161   Serial.print("Tempelkan sidik jari untuk pendaftaran ID #"); Serial.println(id);
162   lcd.clear();
163   lcd.setCursor(0,0);
164   lcd.print("Tempelkan Jari!");
165   while (p != FINGERPRINT_OK) {
Done compiling.
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
Arduino Nano, ATmega328 on COM3

```

(e) Bagian 5

```

SISTEM_KESELURUHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURUHAN
166 p = finger.getImage();
167 switch (p) {
168 case FINGERPRINT_OK:
169   Serial.println("Mendapatkan gambar");
170   break;
171 case FINGERPRINT_NOFINGER:
172   Serial.print(".");
173   break;
174 case FINGERPRINT_PACKETRECEIVEERR:
175   Serial.println("Komunikasi error");
176   break;
177 case FINGERPRINT_IMAGEFAIL:
178   Serial.println("Gambar error");
179   break;
180 default:
181   Serial.println("Tidak diketahui");
182   break;
183 }
184 }
185 // OK success!
186 p = finger.image2Tz(1);
187 switch (p) {
188 case FINGERPRINT_OK:
189   Serial.println("Mengkonversi gambar");
190   break;
191 case FINGERPRINT_IMAGEESS:
192   Serial.println("Gambar kacau");
193   return p;
194 case FINGERPRINT_PACKETRECEIVEERR:
195   Serial.println("Komunikasi error");
196   return p;
197 case FINGERPRINT_FEATUREFAIL:
198   Serial.println("Tidak dapat menemukan fitur");
199 }
Done compiling
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
Arduino Nano, ATmega328 on COM3

```

(f) Bagian 6

```

SISTEM_KESELURUHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURUHAN
199 return p;
200 case FINGERPRINT_INVALIDIMAGE:
201   Serial.println("Tidak dapat menemukan fitur");
202   return p;
203 default:
204   Serial.println("Tidak diketahui");
205   return p;
206 }
207 Serial.println("Menghapus sidik jari");
208 delay(2000);
209 p = 0;
210 while (p != FINGERPRINT_NOFINGER) {
211   p = finger.getImage();
212 }
213 Serial.print("ID "); Serial.println(id);
214 p = -1;
215 Serial.println("Tempelkan sidik jari lagi");
216 lcd.clear();
217 lcd.setCursor(0,0);
218 lcd.print("Tempelkan Jari");
219 lcd.setCursor(0,1);
220 lcd.print("Sekali Lagi !");
221 while (p != FINGERPRINT_OK) {
222   p = finger.getImage();
223   switch (p) {
224 case FINGERPRINT_OK:
225     Serial.println("Mendapatkan gambar");
226     break;
227 case FINGERPRINT_NOFINGER:
228     Serial.print(".");
229     break;
230 case FINGERPRINT_PACKETRECEIVEERR:
231     Serial.println("Komunikasi error");
232 }
}
Done compiling
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
Arduino Nano, ATmega328 on COM3

```

(g) Bagian 7

```

SISTEM_KESELURUHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURUHAN
232     break;
233 case FINGERPRINT_IMAGEFAIL:
234     Serial.println("Gambar error");
235     break;
236 default:
237     Serial.println("Tidak diketahui");
238     break;
239 }
240 }
241 // OK success!
242 p = finger.image2zx(2);
243 switch (p) {
244 case FINGERPRINT_OK:
245     Serial.println("Mengkonversi gambar");
246     break;
247 case FINGERPRINT_IMAGEMESS:
248     Serial.println("Gambar kacau");
249     return p;
250 case FINGERPRINT_PACKETRECEIVEERR:
251     Serial.println("Komunikasi error");
252     return p;
253 case FINGERPRINT_FEATUREFAIL:
254     Serial.println("Tidak dapat menemukan fitur");
255     return p;
256 case FINGERPRINT_INVALIDIMAGE:
257     Serial.println("Tidak dapat menemukan fitur");
258     return p;
259 default:
260     Serial.println("Tidak diketahui");
261     return p;
262 }
263 // OK converted!
264 Serial.print("Membuat model untuk ID #"); Serial.println(id);
Done compiling.
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
Arduino Nano, ATmega328 on COM3

```

(h) Bagian 8

```

SISTEM_KESELURUHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURUHAN
265
266 p = finger.createModel();
267 if (p == FINGERPRINT_OK) {
268     Serial.println("Cetakan Cocok");
269 } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
270     Serial.println("Komunikasi error");
271     return p;
272 } else if (p == FINGERPRINT_ENROLLMISMATCH) {
273     Serial.println("sidik jari tidak cocok");
274     return p;
275 } else {
276     Serial.println("Tidak diketahui");
277     return p;
278 }
279 Serial.print("ID "); Serial.println(id);
280 lcd.clear();
281 lcd.setCursor(0,0);
282 lcd.print("ID "); lcd.print(id);
283 delay(1000);
284 p = finger.storeModel(id);
285 if (p == FINGERPRINT_OK) {
286     Serial.println("Tersimpan");
287     lcd.clear();
288     lcd.print("Tersimpan");
289     delay(2000);
290 } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
291     Serial.println("Komunikasi error");
292     return p;
293 } else if (p == FINGERPRINT_BADLOCATION) {
294     Serial.println("Tidak dapat menyimpan");
295     return p;
296 } else if (p == FINGERPRINT_FLASHERR) {
297     Serial.println("Kesalahan penulisan");
Done compiling.
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1335 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
Arduino Nano, ATmega328 on COM3

```

(i) Bagian 9

```

SISTEM_KESELURUHAN | Arduino 1.8.2
File Edit Sketch Tools Help
SISTEM_KESELURUHAN$
279 Serial.print("ID "); Serial.println(id);
280 lcd.clear();
281 lcd.setCursor(0,0);
282 lcd.print("ID ");lcd.print(id);
283 delay(1000);
284 p = finger.storeModel(id);
285 if (p == FINGERPRINT_OK) {
286   Serial.println("Tersimpan");
287   lcd.clear();
288   lcd.print("Tersimpan");
289   delay(2000);
290 } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
291   Serial.println("Komunikasi error");
292   return p;
293 } else if (p == FINGERPRINT_BADLOCATION) {
294   Serial.println("Tidak dapat menyimpan");
295   return p;
296 } else if (p == FINGERPRINT_FLASHERR) {
297   Serial.println("Kesalahan penulisan");
298   return p;
299 } else {
300   Serial.println("Tidak diketahui");
301   return p;
302 }
303 }
304
305
306
307
308
309
310
311
Done compiling.
Using library Keypad in folder: C:\Program Files (x86)\Arduino\libraries\Keypad (legacy)
Sketch uses 10564 bytes (34%) of program storage space. Maximum is 30720 bytes.
Global variables use 1395 bytes (65%) of dynamic memory, leaving 713 bytes for local variables.
318 Arduino Nano, ATmega328 on COM3

```

(j) Bagian 10

Gambar 4.19. Program pengujian sistem keseluruhan

- Melakukan instalasi antara sistem yang telah dibuat dengan sepeda motor.
- Mendaftarkan sidik jari dengan mengikuti menu yang tampil di LCD.
- Mengaktifkan kelistrikan dan menonaktifkan sistem pengereman dengan menempelkan sidik jari yang telah terdaftar.

4.1.7.3. Hasil Pengujian

Hasil pengujian pendaftaran sidik jari dapat dilihat pada tabel 4.9.

Tabel 4.9. Data pengujian pendaftaran sidik jari

| Nomor id | Kriteria | Hasil pendaftaran |
|----------|-------------------|-------------------|
| 1 | Jempol kanan | Tersimpan |
| 2 | Telunjuk kanan | Tersimpan |
| 3 | Jari tengah kanan | Tersimpan |
| 4 | Jari manis kanan | Tersimpan |
| 5 | Kelingking kanan | Tersimpan |

Data pada tabel diatas merupakan data hasil pendaftaran sidik jari oleh sistem, dari data tersebut dapat diketahui bahwa sistem dapat melakukan pendaftaran sidik jari.

Hasil pengujian pembacaan sidik jari dapat dilihat pada tabel 4.10 dan 4.11.

Tabel 4.10. Data pengujian pembacaan sidik jari yang telah terdaftar

| Kriteria | Nomor ID | Kondisi Kelistrikan | Kondisi Rem | Hasil Pengujian | Error (%) |
|-------------------|----------|---------------------|----------------|-----------------|-----------|
| Jempol kanan | ID # 1 | Menyala | Tidak terkunci | Berhasil | 0 % |
| Telunjuk kanan | ID # 2 | Menyala | Tidak terkunci | Berhasil | 0 % |
| Jari tengah kanan | ID # 3 | Menyala | Tidak terkunci | Berhasil | 0 % |
| Jari manis kanan | ID # 4 | Menyala | Tidak terkunci | Berhasil | 0 % |
| Kelingking kanan | ID # 5 | Menyala | Tidak terkunci | Berhasil | 0 % |

Tabel 4.11. Data pengujian pembacaan sidik jari yang tidak terdaftar

| Kriteria | Nomor ID | Kondisi Kelistrikan | Kondisi Rem | Hasil Pengujian | Error (%) |
|------------------|----------|---------------------|-------------|-----------------|-----------|
| Jempol kiri | - | Mati | Terkunci | Berhasil | 0 % |
| Telunjuk kiri | - | Mati | Terkunci | Berhasil | 0 % |
| Jari tengah kiri | - | Mati | Terkunci | Berhasil | 0 % |
| Jari manis kiri | - | Mati | Terkunci | Berhasil | 0 % |
| Kelingking kiri | - | Mati | Terkunci | Berhasil | 0 % |

Data pada tabel 4.10 dan 4.11 adalah data pembacaan sidik jari , kondisi kelistrikan dan kondisi rem sepeda motor. Data pada tabel 4.10 merupakan pembacaan sidik jari yang telah terdaftar sebelumnya, pada pengujian ini sidik jari dapat mengakses sistem ditandai dengan kondisi kelistrikan yang tadinya mati menjadi aktif setelah sidik jari ditempelkan pada sensor. Data pada tabel 4.11

merupakan pembacaan sidik jari yang tidak terdaftar, pada pengujian ini sidik jari tidak dapat mengakses sistem yang ditandai dengan kondisi kelistrikan yang tadinya mati tetap mati setelah sidik jari ditempelkan pada sensor. Data pada tabel 4.10 dan 4.11 juga menampilkan kondisi rem pada sepeda motor setelah dilakukan penempelan sidik jari yang terdaftar dan tidak terdaftar. Pada saat sidik jari yang telah terdaftar di tempelkan pada sensor, kondisi rem pada sepeda motor yang tadinya terkunci menjadi tidak terkunci. Pada saat sidik jari yang tidak terdaftar ditempelkan pada sensor, kondisi rem pada sepeda motor yang tadinya terkunci tetap terkunci. Data pada tabel 4.8 dan 4.9 serta penjelasan diatas menunjukkan bahwa sistem ini dapat digunakan sebagai pengaman sepeda motor.

4.1.8. Pengujian Alat Saat Mesin Motor Menyala

Pengujian ini dilakukan untuk mengetahui apakah alat yang dibuat aman digunakan ketika mesin motor dalam keadaan menyala atau sedang di kendarai. pengujian ini dilakukan dengan cara mengendarai sepeda motor yang telah terpasang sistem yang dibuat. Pengujian dilakukan secara bertahap, mulai dari 15 menit sampai dengan 1 jam. Tabel 4.12 merupakan hasil pengujian pengujian alat saat mesin motor menyala.

Tabel 4.12 Data pengujian alat saat mesin motor menyala

| Lama Pengujian | kendala | Hasil pengujian |
|----------------|-----------|-----------------|
| 15 menit | Tidak ada | Berhasil |
| 30 menit | Tidak ada | Berhasil |
| 45 menit | Tidak ada | Berhasil |
| 1 jam | Tidak ada | Berhasil |

Data pada tabel 4.12 adalah data pengujian alat saat mesin motor menyala. Hasil yang didapat pada pengujian ini adalah alat yang dibuat aman digunakan pada saat mesin motor dalam keadaan menyala.

4.2. Pembahasan

Berdasarkan hasil pengujian sistem pengaman sepeda motor dengan mengendalikan sistem kelistrikan dan pengereman menggunakan sidik jari secara

keseluruhan yang telah dilakukan dapat diketahui informasi tentang kinerja alat. Hasil pengujian yang diketahui adalah *keypad*, LCD, sensor sidik jari, *relay* kontak, aktuator, arus dan sistem secara keseluruhan.

Pengujian perangkat *input* yaitu *keypad* dapat mengirimkan karakter ke mikrokontroler ketika karakter pada *keypad* di tekan. *Output keypad* berkondisi *HIGH* ketika karakter pada *keypad* ditekan, kemudian dijadikan input mikrokontroler. Kemudian input tersebut diolah oleh mikrokontroler sebagai karakter yang sesuai dengan karakter yang ditekan pada *keypad*. Perangkat *input* lainnya yaitu sensor sidik jari, sensor sidik jari mendeteksi sidik jari saat melakukan pembacaan dan memverifikasi apakah sidik jari tersebut telah terdaftar sebelumnya. Jika sidik jari tersebut telah terdaftar sebelumnya maka sensor sidik jari akan memberi tahu mikrokontroler bahwa sidik jari tersebut benar, kemudian mikrokontroler akan mengolah kondisi tersebut untuk mengaktifkan kelistrikan dan rem sepeda motor. selain membaca sidik jari, sensor sidik jari juga mendaftarkan sidik jari baru dimana sensor sidik jari akan memindai sidik jari baru sebanyak dua kali yang sebelumnya telah diberi nomor id. Proses pendaftaran sidik jari yaitu, sensor sidik jari memindai sidik jari, kemudian pola sidik jari tersebut dikonversikan ke bentuk biner untuk disimpan di *database* sensor sidik jari. Pada saat pembacaan sensor sidik jari akan mengubah pola sidik jari yang terbaca ke bentuk biner dan mencocokkan data tersebut dengan data yang ada pada *database* sensor sidik jari.

Pengujian perangkat output yaitu LCD, LCD ini bekerja sesuai dengan perintah yang dikirimkan oleh mikrokontroler. Saat sistem bekerja lcd akan menampilkan menu untuk proses pendaftaran sidik jari, informasi tentang proses pendaftaran sidik jari, informasi tentang sidik jari yang tersimpan dan informasi tentang keakuratan sidik jari yang terbaca. Kemudian perangkat output selanjutnya adalah *relay* kontak, *relay* ini berfungsi untuk memutus dan menghubungkan arus listrik pada sistem kelistrikan sepeda motor sesuai dengan perintah mikrokontroler. Ketika *relay* ini memutuskan arus listrik sistem kelistrikan sepeda motor maka sepeda motor tidak dapat digunakan, kemudian *relay* akan menghubungkan sistem kelistrikan sepeda motor ketika mendapat

perintah dari mikrokontroler. Mikrokontroler akan mengirimkan perintah ke *relay* untuk mengaktifkan sistem kelistrikan sepeda motor pada saat ada sidik jari yang terbaca, dimana sidik jari tersebut telah terdaftar sebelumnya. Kemudian perangkat output yang lainnya adalah aktuator, aktuator ini bekerja untuk mendorong pedal rem pada sepeda motor. gerakan pada aktuator adalah maju dan mundur. Pada saat gerakan maju aktuator akan mendorong pedal rem, sehingga roda pada sepeda motor tidak dapat berputar. Gerakan mundur aktuator akan melepaskan pedal rem sehingga roda pada sepeda motor dapat digerakan atau berputar. Kerja aktuator dikontrol oleh mikrokontroler melalui sebuah rangkaian *driver* motor. kondisi maju pada aktuator yaitu ketika sistem kelistrikan sepeda dan arus listrik yang menyuplai kontroler pada sistem ini terputus oleh kunci kontak sepeda motor. kemudian untuk kondisi mundur pada aktuator yaitu pada saat ada sidik jari yang terbaca, dimana sidik jari tersebut telah terdaftar sebelumnya.

Arduino nano sebagai sistem kendali dalam mengolah sinyal input menjadi sinyal output dalam sistem pengaman sepeda motor dengan mengendalikan sistem kelistrikan dan pengereman menggunakan sidik jari dilakukan berdasarkan pada program yang ditanamkan pada mikrokontroler. Dalam sistem ini arduino nano dapat bekerja sesuai dengan prinsip kerja alat. Perangkat *input* dan *output* dapat melakukan tugas-tugasnya sesuai dengan perintah dari mikrokontroler.

Untuk mengetahui apakah sistem ini dapat bekerja pada sepeda motor, maka peneliti melakukan instalasi pada sepeda motor jenis Yamaha vixion, dan hasilnya adalah sistem bekerja dengan baik. Prinsip kerjanya adalah sistem kelistrikan sepeda motor terputus dan roda belakang sepeda motor terkunci ketika tidak ada sidik jari yang terbaca, dimana sidik jari tersebut telah terdaftar sebelumnya. Sistem kelistrikan sepeda motor akan aktif dan roda belakang sepeda motor tidak terkunci ketika ada sidik jari yang terbaca, dimana sidik jari tersebut telah terdaftar sebelumnya. Pada saat kontak dimatikan maka aktuator akan mendorong pedal rem untuk mengunci roda belakang, agar ketika sepeda motor diparkir sepeda motor dalam keadaan aman karena roda belakang terkunci sehingga tidak dapat di gerakan atau dipindahkan. Dari hasil kerja sistem ini dapat diketahui bahwa sistem ini dapat menjadi pengaman tambahan pada sepeda motor,

dimana ketika kelistrikan terputus maka sepeda motor tidak dapat digunakan dan ketika roda belakang tidak dapat berputar maka sepeda motor tidak dapat digerakan atau dipindahkan.

Pada alat ini juga terdapat kelemah- kelemahan diantaranya:

- 1) Tidak adanya *casing* untuk aktuator, sehingga memungkinkan pencuri untuk melepas aktuator.
- 2) Ketika energi listrik pada batrai sepeda motor habis maka sepeda motor tidak dapat dihidupkan sekalipun dengan *kick starter* karena jalur kelistrikan sepeda motor yang terputus oleh *relay*.
- 3) Box rangkaian terlalu besar sehingga memerlukan tempat yang cukup besar.