

DAFTAR PUSTAKA

1. Susila, M. W. 2019. Sistem Keamanan Brankas Secara Otomatis dengan Menggunakan SMS serta PIN dan RFID. *Tesla*.
2. Rahmawati, F. 2021. *Rumah Dibobol maling, Terry Putri Kehilangan Brankas Berisi Harta yang Dikumpulkan Selama 20 Tahun*. Retrieved Juli 27, 2022, from Kompas TV: <https://www.kompas.tv/article/170481/rumah-dibobol-maling-terry-putri-kehilangan-brankas-berisi-harta-yang-dikumpulkan-selama-20-tahun>.
3. Annisya, L. H. 2018. Sistem Keamanan Buka Tutup Kunci Brankas Menggunakan Sidik Jari Berbasis Arduino Uno. *Jurnal Ilmiah Informatika Komputer*.
4. Purba, A. 2020. *Sistem Pengamanan Brankas Menggunakan Sensor sidik Jari*. Sumatra Utara: Universitas Sumatra Utara.
5. Andhika Ghifari Aditya, I. P. 2020. Sistem Pengunci Pintu RFID dan Pasword Berbasis Arduino Uno dengan One Time Pasword Melalui SMS. *SENAMIKA*.
6. Tobing, S. L. 2019. Rancang Bangun Pengaman Pintu Menggunakan Sensor Sidik Jari (Fingerprint dan Smarthphone Arduino Berbasis Mikrokontroler Atmega 8. 7.
7. Apri Siswanto, A. E. 2018. Alat Kontrol Akses Pintu Rumah dengan Teknologi Sidik Jari di Lingkungan Rumah Pintar dengan Dtata yang di Enskripsi. *JPPI*.
8. Eko Adji Saputro, H. A. 2020. Sistem Pembuka Pintu Rumah Menggunakan Sidik Jari dan Smarthphone Android.
9. Purba, A. 2020. *Sistem Pengamanan Brankas Menggunakan Sensor sidik Jari*. Sumatra Utara: Universitas Sumatra Utara.
10. Iwan Sanjaya, R. G. 2021. Penggunaan Modul Sensor Sidik Jari (Fingerprint) sebagai Verifikasi Ganda untuk Sistem Simulasi pemilu. *Jurnal Ilmiah Teknologi Energi, I(1)*, 35.42.
11. BS Rahayu Purwanti, F. A. 2018. Perancangan Sistem Presentasi Merespon Pola Sidik Jari dari Sensor Fingerprint. *Jurnal Poli-Teknologi, XVII(2)*.
12. Dedek Irwansyah, S. P. 2018. Implementasi Sensor Sidik Jari sebagai Kendali ON/OFF Mesin Sepeda Motor Honda. *Jurnal Elektrikal, V*.

13. Masnur, S. A. 2021. Rancang Bangun Sistem Keamanan Motor dengan Menggunakan Sidik Jari Berbasis Arduino Uno. *Sintaks Logika, I*.
14. Zikrul Khalid, K. A. 2020. Otomatisasi Sistem Keamanan Kunci Lemari Menggunakan Sensor Sidik Jari Berbasis Arduino Uno. *TEKSAGRO, 1*.
15. Wahyunita, S. 2021. *Implementasi IOT untuk Pengamanan Toolbox Peralatan Elektronika*. Cilacap: Politeknik Negeri Cilacap.
16. Rian Anjasmara, T. S. 2019. Implementasi Sistem Monitoring Kecepatan Angin, Suhu, dan Kelembaban Berbasis Web di Daerah Kepulauan. Politeknik Negeri Batam.

LAMPIRAN A

DAFTAR PROGRAM ARDUINO

A. Program Arduino untuk Sensor Sidik Jari

```
#include <Adafruit_Fingerprint.h>
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);
```

```
void setupFingerprint() {
  Serial.begin(9600);
  Serial2.begin(57600);
}
```

```
int fingerPrintGetId() {
  int result = 0;
  while (1) {
    result = getFingerprint();
    if (result == -3) {
      break;
    }
    else if (result >= 0) {
      break;
    }
    delay(10);
    listenCommand();
  }
  return result;
}
```

```
int fingerprintCount() {
  finger.getTemplateCount();
  return finger.templateCount;
}
```

```
int getFingerprint() {
  uint8_t p = finger.getImage();
  p = finger.getImage();
  if (p != FINGERPRINT_OK) return -1;
  p = finger.image2Tz();
```

```

if (p != FINGERPRINT_OK) return -2;
p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -3;
return finger.fingerID;
}

uint8_t fingerprintAdd(int id) {
int p = -1;
Serial.print("id="); Serial.println(id);
Serial.println("tempelkan jari");
lcdPrint(0, 1, "Tempelkan Jari ");
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.print(".");
break;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
}
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_PACKETRECI EVEERR:
    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("angkat jari");
lcdPrint(0, 1, "Angkat Jari ");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
p = -1;
Serial.println("tempelkan jari yang sama");
lcdPrint(0, 1, "T'pel'n Jari Sma");
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.print(".");
            break;
        case FINGERPRINT_PACKETRECI EVEERR:
            Serial.println("Communication error");
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");
            break;
        default:

```

```

        Serial.println("Unknown error");
        break;
    }
}
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_PACKETRECIIEVEERR:
        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("membuat model");
p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
}
else if (p == FINGERPRINT_PACKETRECIIEVEERR) {
    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;
}
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;
}
}

void fingerprintDeleteAllFinger() {
    finger.emptyDatabase();
}

```

B. Program Arduino untuk Keypad

```

#include <Keypad.h>

const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},

```

```

    {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {23, 25, 27, 29};
byte colPins[COLS] = {31, 33, 35, 37};

```

```

Keypad customKeypad = Keypad( makeKeymap(hexaKeys),
rowPins, colPins, ROWS, COLS);

```

```

char keypadGetChar() {
    char c = customKeypad.getKey();
    if (c) {
        return c;
    }
    return false;
}

```

C. Program Arduino untuk LCD I2C

```

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setupLcd() {
    lcd.begin();
    lcd.backlight();
    lcd.clear();
}

void lcdPrint(int x, int y, String msg) {
    lcd.setCursor(x, y);
    lcd.print(msg);
}

void lcdClear() {
    lcd.clear();
}

```

D. Program Arduino untuk Modul SD Card

```

#include <SPI.h>

```



```

#include <SD.h>
File myFile;

void setupSd() {
  if (!SD.begin(53)) {
    Serial.println("SD initialization failed!");
  }
  delay(1000);
}

void sdAddContact(String id, String number) {
  myFile = SD.open("contact.txt", FILE_WRITE);
  if (myFile) {
    myFile.println(id + ',' + number);
    myFile.close();
  }
}

String sdGetContact(String id) {
  myFile = SD.open("contact.txt");
  if (myFile) {
    String content = "";
    while (myFile.available()) {
      content += (char) myFile.read();
    }
    char lastChar = '\n';
    bool isFound = false;
    int firstIndex = 0;
    int lastIndex = 0;
    for (int i = 0; i < content.length(); i++) {
      char c = content[i];
      if (lastChar == '\n') {
        if (isFound) {
          lastIndex = i;
          break;
        }
      }
      if ((String) c == id) {
        isFound = true;
        firstIndex = i;
      }
    }
  }
}

```

```

        lastChar = c;
    }
    myFile.close();
    if (lastIndex == 0) lastIndex = content.length();
    return content.substring(firstIndex + 2, lastIndex - 2);
}
}

void sdClearContact() {
    SD.remove("contact.txt");
}

void sdWriteLog(String id, String number, String pwGen, String
pwInt, String access) {
    String ts = simGetTime();
    myFile = SD.open("log.csv", FILE_WRITE);
    if (myFile) {
        myFile.println(ts + ',' + id + ',' + number + ',' + pwGen + ',' +
pwInt + ',' + access);
        myFile.close();
    }
}

void sdReadLog() {
    myFile = SD.open("log.csv");
    if (myFile) {
        String content = "";
        while (myFile.available()) {
            content += (char) myFile.read();
        }
        Serial.println(content);
        myFile.close();
    }
    else {
        Serial.println("error opening log.csv");
    }
}

void sdClearLog() {
    SD.remove("log.csv");
}
}

```

E. Program Arduino untuk Modul GSM

```
void setupSim() {
  Serial3.begin(9600);
}

void simSendCode(String number, String code) {
  Serial3.println("AT+CMGF=1");
  delay(1000);
  Serial3.println("AT+CNMI=2,2,0,0,0");
  Serial3.println("AT+CMGF=1");
  delay(1000);
  Serial3.println("AT+CMGS=\"" + number + "\"\r");
  delay(1000);
  Serial3.println(code);
  delay(100);
  Serial3.println((char)26);
  delay(1000);
}

  Serial3.println(msg);
  delay(100);
  Serial3.println((char)26);
  delay(1000);
}

void simSendWarning(String number, String code) {
  Serial3.println("AT+CMGF=1");
  delay(1000);
  Serial3.println("AT+CNMI=2,2,0,0,0");
  Serial3.println("AT+CMGF=1");
  delay(1000);
  Serial3.println("AT+CMGS=\"" + number + "\"\r");
  delay(1000);
  Serial3.println(code);
  delay(100);
  Serial3.println((char)26);
  delay(1000);
}

String simGetTime() {
  String result = "";
```

```

Serial3.end();
Serial3.begin(9600);
Serial3.println("AT+CCLK?");
while (!Serial3.available());
if (Serial3.available()) {
    String data = Serial3.readStringUntil("\n");
    int tsBegin = data.indexOf("") + 1;
    int tsEnd = data.indexOf('+', tsBegin);
    result = data.substring(tsBegin, tsEnd);
}
return result;
}

```

F. Program Arduino untuk Solenoid

```

#define pinSolenoid 7

void setupSolenoid() {
    pinMode(pinSolenoid, OUTPUT);
}
void solenoidSet(bool s) {
    digitalWrite(pinSolenoid, s);
}

```

G. Program Arduino untuk Keseluruhan Sistem

```

const String MASTERNUMBER = "089647040989";

void setup() {
    Serial.begin(9600);
    setupLcd();
    setupFingerprint();
    setupSolenoid();
    setupSim();
    setupSd();
}

int failedTry = 0;
void loop() {

```

```

lcdPrint(0, 0, "Tempelkan Jari");
int fingerprintDetected = fingerPrintGetId();
if (fingerprintDetected >= 0) {
    lcdPrint(0, 1, "id=" + String(fingerprintDetected));
    delay(1000);
    lcdClear();
    String passwordGenerated = generatePassword();
    String number = "";
    if (fingerprintDetected == 0) {
        number = MASTERNUMBER;
    }
    else {
        number = (String) sdGetContact(String(fingerprintDetected));
    }
    Serial.print(number);
    Serial.print('\t');
    Serial.println(passwordGenerated);
    lcdPrint(0, 0, "Mengirim SMS");
    simSendCode(number, passwordGenerated);
    lcdClear();
    lcdPrint(0, 0, "Masukkan Password");
    lcdPrint(12, 1, "_____");
    String passwordReceived = "";
    int secondLeft = 180;
    long lastDecreaseSecond = 0;
    while (passwordReceived.length() < 4 && secondLeft > 0) {
        if (millis() > lastDecreaseSecond + 1000) {
            lastDecreaseSecond = millis();
            secondLeft--;
            lcdPrint(0, 1, "  ");
            lcdPrint(0, 1, String(secondLeft / 60) + ':' +
String(secondLeft % 60));
        }
        char c = keypadGetChar();
        if (c == '*') {
            passwordReceived = passwordReceived.substring(0,
passwordReceived.length() - 1);
            lcdPrint(12, 1, "_____");
            lcdPrint(12, 1, passwordReceived);

```

```

    }
    else if (c) {
        passwordReceived += c;
        lcdPrint(12, 1, passwordReceived);
    }
}
delay(1000);
lcdClear();
if (passwordGenerated == passwordReceived) {
    failedTry = 0;
    sdWriteLog(String(fingerprintDetected), number,
passwordGenerated, passwordReceived, "Akses Diterima");
    lcdPrint(0, 0, "Akses Diterima");
    solenoidSet(true);
    delay(1000);
    lcdClear();
    lcdPrint(0, 0, ".*.Selesai");
    if (fingerprintDetected == 0) {
        lcdPrint(0, 1, "#.Pengaturan");
    }
    while (1) {
        char c = keypadGetChar();
        if (c == '#') {
            setting();
            lcdClear();
            lcdPrint(0, 0, ".*.Selesai");
            if (fingerprintDetected == 0) {
                lcdPrint(0, 1, "#.Pengaturan");
            }
        }
        else if (c != false) {
            break;
        }
    }
    solenoidSet(false);
}
else {
    failedTry++;
}

```

```

        sdWriteLog(String(fingerprintDetected), number,
passwordGenerated, passwordReceived, "Akses Ditolak");
        lcdPrint(0, 0, "Akses Ditolak");
        delay(1000);
    }
}
else {
    failedTry++;
    sdWriteLog("-", "-", "-", "-", "Percobaan Fingerprint");
    lcdPrint(0, 1, "Tidak Terdeteksi");
    delay(1000);
}
if (failedTry >= 3) {
    lcdClear();
    lcdPrint(0, 0, "Memproses");
    simSendWarning("Peringatan!!!\nPercobaan sidik jari tidak
terdeteksi sebanyak " + String(failedTry) + " kali");
}
lcdClear();
}

```

```

String generatePassword() {
    long rnd = millis();
    String result = (String) rnd;
    return result.substring(result.length() - 4, result.length());
}

```

```

void setting() {
    lcdClear();
    lcdPrint(0, 0, "A.Tambah ID");
    lcdPrint(0, 1, "B.Hapus Semua ID");
    while (1) {
        char c = keypadGetChar();
        if (c == 'A') {
            settingAddId();
            break;
        }
        else if (c == 'B') {
            settingRemoveAllId();
        }
    }
}

```

```

        break;
    }
    else if (c != false) {
        break;
    }
}
}

void settingAddId() {
    lcdPrint(0, 0, "Tambah Fingerprint");
    int targetId = fingerprintCount();
    uint8_t result = fingerprintAdd(targetId);
    if (result == 0) {
        lcdPrint(0, 1, "Selesai ");
        delay(1000);
        lcdPrint(0, 0, "Tambah Nomor Hp ");
        lcdPrint(0, 1, " ");
        String number = "";
        while (1) {
            char c = keypadGetChar();
            if (c == '0' || c == '1' || c == '2' || c == '3' || c == '4' || c == '5' || c
            == '6' || c == '7' || c == '8' || c == '9') {
                number += c;
                lcdPrint(0, 1, number);
            }
            else if (c == '*') {
                number = number.substring(0, number.length() - 1);
                lcdPrint(0, 1, " ");
                lcdPrint(0, 1, number);
            }
            else if (c == '#') {
                break;
            }
        }
        Serial.println(targetId);
        Serial.println(number);
        sdAddContact(String(targetId), number);
        lcdPrint(0, 1, "Selesai ");
        delay(1000);
    }
}

```



```

    }
    else {
        lcdPrint(0, 1, "Error      ");
        delay(1000);
    }
}

void settingRemoveAllId() {
    sdClearContact();
    fingerprintDeleteAllFinger();
    lcdPrint(0, 0, "Fingerprnt Utama");
    while (1) {
        uint8_t result = fingerprintAdd(0);
        if (result == 0) {
            lcdPrint(0, 1, "Selesai      ");
            delay(1000);
            break;
        }
        else {
            lcdPrint(0, 1, "Error      ");
            delay(1000);
        }
    }
}

void listenCommand() {
    if (Serial.available()) {
        String command = Serial.readStringUntil("\n");
        Serial.print(command);
        if (command == "log?\n") {
            sdReadLog();
        }
        if (command == "log-clear\n") {
            sdClearLog();
        }
    }
}
}

```


LAMPIRAN B

MANUAL BOOK BRANKAS

Langkah pertama untuk menggunakan brankas adalah colokan adaptor ke dalam sumber listrik.

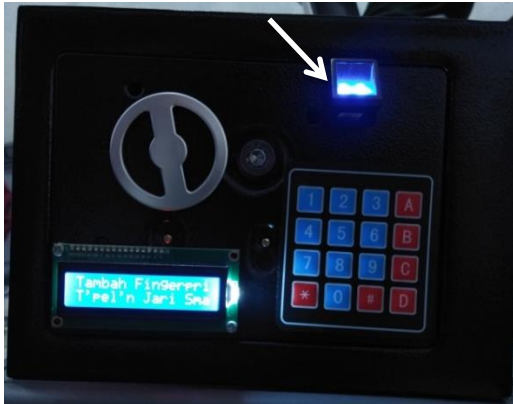
1. Tampilan awal pada LCD yaitu menunjukkan perintah tempelkan sidik jari pada sensor sidik jari dengan menggunakan tangan yang bersih dan kering. Jika sidik jari utama belum terdaftar maka sistem akan otomatis masuk ke afar pendaftaran.



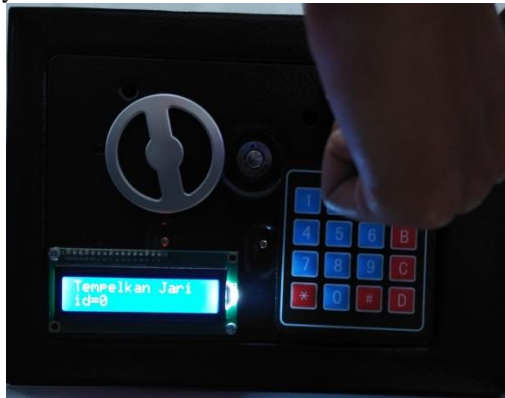
2. Kemudian LCD akan memberikan perintah angkat jari pada sensor sidik jari untuk pendaftaran sidik jari



3. Kemudian tempelkan sidik jari yang sama ketika muncul perintah pada LCD



4. Sistem akan memproses sidik jari dan menambahkan menjadi sidik jari utama yaitu Id=0.



5. Setelah itu sistem akan otomatis mengirim SMS ke nomor ponsel yang telah disediakan dari sistem.



6. Jika sudah mengirim SMS, maka masukan OTP (4 digit) yang dikirim melalui SMS pada keypad. Dalam menginput OTP diberi waktu 3 menit. Dan masukan OTP yang dikirim dengan benar.

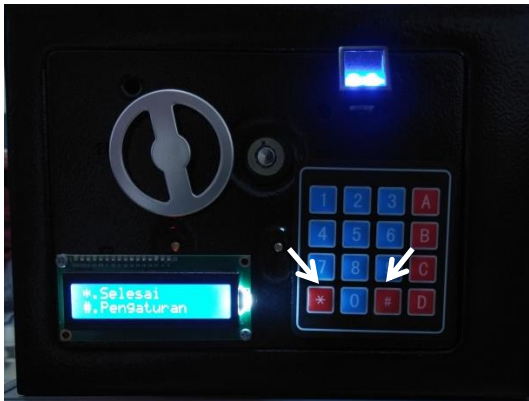


4

7. Jika sudah bunyi “cetek” maka kunci telah terbuka. Kemudian putar engsel ke kanan untuk membuka pintu brankas.



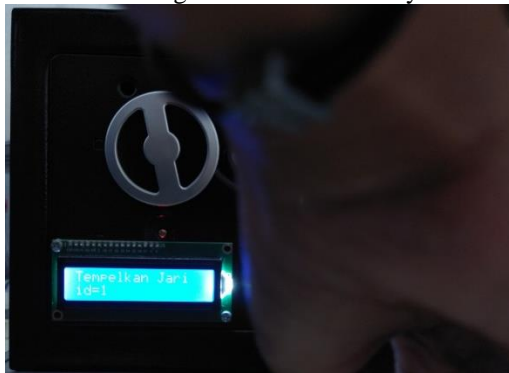
8. Untuk menambahkan fitur sidik jari, tekan tombol # untuk ke pengaturan. Jika tidak tekan tombol * untuk selesai.



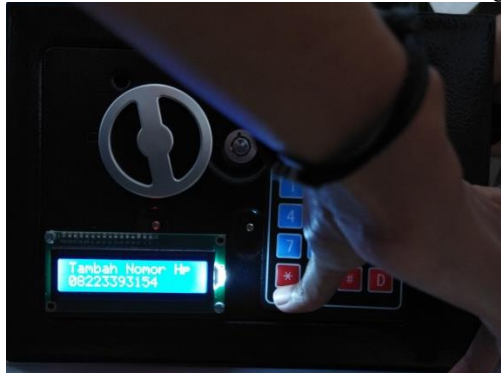
9. Jika menekan pengaturan maka akan muncul pilihan A. tambah ID dan B. Hapus Semua ID.



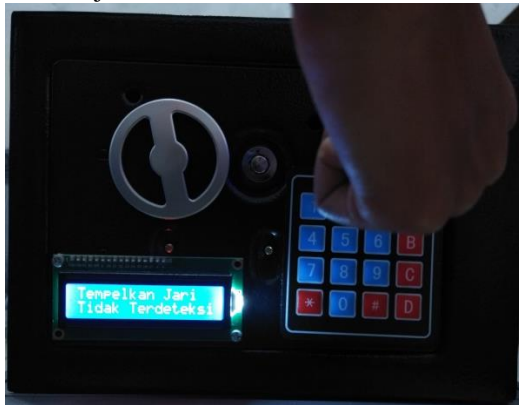
10. Jika menekan A. Tambah ID maka sistem akan memberi perintah yang sama pada langkah 1-3. Jika prosesnya sesuai langkah maka sidik jari akan terdapat dengan ID 1 dan seterusnya.



11. Kemudian tambahkan nomor ponsel yang aktif, jika dalam penginput nomor ada kesalahan maka tekan * untuk menghapus.



12. Jika menekan B. Hapus Semua ID, maka sistem akan otomatis menghapus sidik jari dan mulai dari langkah awal.
13. Jika sidik jari tidak terdaftar maka sistem akan menampilkan pada LCD, bahwa sidik jari tidak terdeteksi.



14. Jika alat mengalami eror dalam tampilan LCD maka tekan tombol reset yang ada di belakang brankas.



LAMPIRAN C

DOKUMENTASI MENGGUNAKAN ALAT

1. Proses melakukan pendaftaran sidik jari



2. Proses memasukkan nomor posel pada keypad



2

3. Proses membuka pintu brankas



LAMPIRAN D
DOKUMENTASI ALAT



Gambar (a) Brankas tampak bagian luar



Gambar (b) Brankas dalam keadaan menyala

2



Gambar (c) Brankas tampak belakang

BIODATA PENULIS



Nama : Viel Vina
Tempat/Tanggal Lahir : Cilacap, 7 Mei 2001
Email : Viel.vina@gmail.com
Alamat : Jalan Lapang RT 02 RW 03 Desa
Glempang, Kec. Maos, Kab. Cilacap
Telepon/HP : 089647040989
Hobi : Menulis
Moto : Tuk petualangan ini, mari kita ketuk
pintu yang sama
Riwayat Pendidikan :

- TK Pertiwi Glempang Tahun 2006-2007
- SD Negeri Glempang 01 Tahun 2007-2013
- SMP Negeri 1 Sampang Tahun 2013-2016
- SMA Negeri 1 Maos Tahun 2016-2019

Penulis telah mengikuti seminar Tugas Akhir pada tanggal 26 Juli 2022 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).