

PROYEK IEE2022
MIKROKONTROLER DAN SENSOR I



Disusun oleh:

Grace Theofie	(212100120)
Mastri Tasya Napitupulu	(212100370)
Martin Emannuel Chang	(212100199)
Priscilla Green Samosir	(212100332)

CALVIN INSTITUTE OF TECHNOLOGY

2023

DAFTAR ISI

DAFTAR ISI	2
PENDAHULUAN	3
1.1 Latar Belakang Masalah	3
1.2 Tujuan	4
1.3 Manfaat	4
PERANCANGAN	5
2.1 Identifikasi Kebutuhan & Cara Alat Kerja	5
a. Arduino	5
b. Sensor Ultrasonik HC - SRO4	5
c. Fingerprint Sensor	5
d. Light - Emitting Diode (LED)	5
e. Liquid Crystal Display (LCD)	5
f. Push Button	6
2.2 Cara Perakitan Rangkaian	6
2.3 Cara Pemograman	6
a. Arduino Sensor	6
b. Arduino Aktuator	17
2.4 Perangkat Lunak	19
a. TinkerCAD	19
b. Software Arduino Uno	19
KESIMPULAN DAN SARAN	20
3.1 Kesimpulan	20
3.2 Saran	20
LAMPIRAN	21
I. Bill of Material	21
II. Kontribusi Kelompok	22

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam kehidupan sehari-hari, kita sering menyimpan barang berharga di rumah atau di tempat kerja. Namun, tanpa perlindungan yang memadai, barang-barang berharga ini dapat menjadi sasaran pencurian atau kerusakan akibat bencana alam seperti banjir atau kebakaran. Oleh karena itu, kotak keamanan dirancang untuk memberikan tingkat keamanan tambahan dan meminimalkan risiko kehilangan barang berharga.

Lembaga tertentu juga contoh lain yang kadangkala memiliki barang yang sering dibutuhkan, namun ketersediaannya masih terbatas. Contohnya, di lingkup kecil seperti kampus kita di Calvin Institute of Technology, kebutuhan akan barang-barang di dalam lab seringkali terbatas dan dibutuhkan oleh banyak orang. Hal ini seringkali menjadi sumber masalah, seperti ketika barang rusak namun tidak diketahui siapa yang terakhir menggunakannya sehingga sulit untuk meminta pertanggungjawaban, atau ketika kita membutuhkan barang tersebut dengan cepat namun tidak dapat menemukannya secara langsung.

Namun, masih ada masalah terkait keberadaan barang tersebut, yaitu kurangnya sistem yang dapat mengatur dan mengendalikan siapa saja yang dapat mengaksesnya. Hal ini dapat menyebabkan masalah keamanan dan privasi yang cukup besar, karena orang yang tidak berwenang dapat dengan mudah mengakses barang tersebut. Pentingnya memiliki sistem pengaturan dan pengendalian yang baik tidak hanya berlaku untuk barang yang disimpan secara elektronik, tetapi juga untuk barang fisik. Misalnya, di sebuah perusahaan, ada banyak barang berharga seperti peralatan kantor, mesin, dan bahan baku yang perlu dijaga dan dilindungi dari akses yang tidak sah. Dengan adanya sistem pengaturan dan pengendalian yang baik, perusahaan dapat memastikan bahwa hanya orang yang memiliki hak akses yang dapat mengambil dan menggunakan barang tersebut, sehingga dapat mencegah kehilangan atau kerusakan yang tidak disengaja, dan meminimalkan risiko keamanan dan privasi.

1.2 Tujuan

Tujuan proyek kita adalah untuk memberikan keamanan dan perlindungan terhadap potensi ancaman seperti pencurian, kerusakan, atau kehilangan barang berharga. Selain itu, proyek kita bertujuan untuk memberi informasi keamanan barang berharga secara aman dan praktis.

1.3 Manfaat

Proyek kita memberikan banyak manfaat bagi pengguna, terutama dalam melindungi barang-barang berharga dari ancaman pencurian, kerusakan, atau kehilangan. Dengan menggunakan proyek kami, pengguna dapat merasa lebih tenang dan terjamin atas keamanan barang-barang berharga mereka. Proyek kami juga memberikan solusi penyimpanan yang aman dan praktis bagi barang berharga yang tidak sering digunakan. Bukan cuman itu, proyek kita dapat mengatur access yang diberikan dengan sangat gampang. Pengguna dapat mengurangi kerumitan dalam penyimpanan dan memberikan akses mudah ketika dibutuhkan.

BAB 2

PERANCANGAN

2.1 Identifikasi Kebutuhan & Cara Alat Kerja

Alat yang kami hasilkan terdiri dari beberapa komponen seperti Arduino, sensor ultrasonik HC - SR04, fingerprint sensor, Light - Emitting Diode (LED), liquid crystal display(LCD), dan push button. Berikut adalah penjelasan cara kerja masing-masing komponen:

a. Arduino

Arduino digunakan sebagai master dalam komunikasi serial I2C. Tugas utama dari Arduino Uno dalam komunikasi serial I2C adalah menginisiasi transaksi I2C, mengirimkan sinyal start dan stop ke perangkat yang terhubung, dan mengirim dan menerima data melalui bus I2C.

b. Sensor Ultrasonik HC - SR04

Sensor ultrasonik digunakan untuk mengukur jarak antara sensor dan objek di depannya agar nantinya dapat mendeteksi gerakan objek di sekitarnya dan memberikan sinyal ketika ada pergerakan yang terdeteksi dan memantau sistem keamanan dan pengawasan.

c. Fingerprint Sensor

Fingerprint digunakan untuk mengambil dan memproses sidik jari seseorang menjadi data digital. Yang dimana data ini nanti digunakan untuk mengidentifikasi atau memverifikasi identitas seseorang. kebutuhan alat ini digunakan untuk membantu dalam memastikan bahwa orang yang diizinkan yang dapat memiliki akses untuk masuk.

d. Light - Emitting Diode (LED)

LED digunakan sebagai indikator lampu atau cahaya lampu penerangan rendah pada berbagai perangkat elektronik. Kebutuhan alat ini digunakan sebagai bagian dari sistem deteksi gerak yang memicu alarm atau notifikasi keamanan ketika gerakan terdeteksi

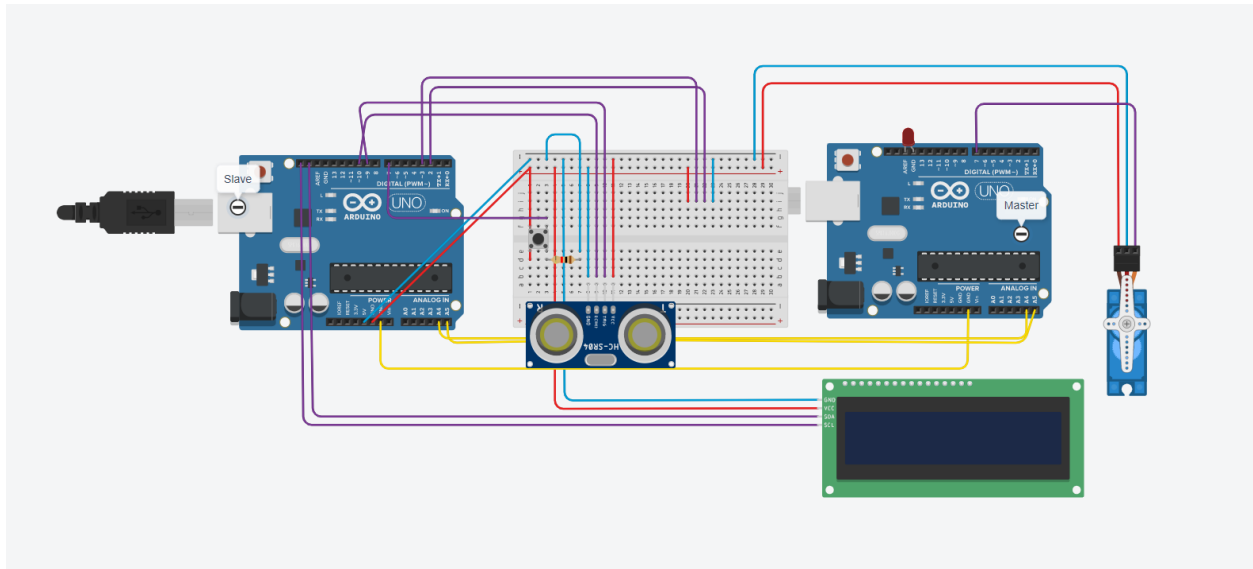
e. Liquid Crystal Display (LCD)

LCD digunakan sebagai bagian dari sistem keamanan untuk menampilkan informasi tentang status keamanan kapan memasuki atau mengakses sistem tersebut. Kebutuhan alat ini juga digunakan untuk memberikan notifikasi ketika melakukan penginputan data sidik jari seseorang oleh fingerprint sensor yang nanti nya akan berisi informasi Open dan Lock dan untuk keadaan ketersediaan status barang Yes or NO

f. Push Button

Push button digunakan untuk mendaftarkan diri dengan menekan tombol push button dalam menggunakan sistem keamanan. Kebutuhan alat ini digunakan sebagai suatu proses sebelum melakukan penginputan atau pendaftaran sidik jari oleh fingerprint sensor.

2.2 Cara Perakitan Rangkaian



2.3 Cara Pemrograman

a. Arduino Sensor

```
#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal_I2C.h>
#define SLAVE1_ADDRESS 0x9A
#include <Wire.h>

#if (defined(__AVR__) || defined(ESP8266)) &&
!defined(__AVR_ATmega2560__)
  SoftwareSerial mySerial(2, 3);
#else
  #define mySerial Serial1
#endif
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;
```

```

String state = "";
int led_pin = 13;
int trig_pin = 9;
int echo_pin = 10;
int button_pin = 7;
int lock_state = 0;

long duration;
int distance;
LiquidCrystal_I2C lcd(0x27,16,2); // addresss i2c = 0x27

void setup() {
  Wire.begin(SLAVE1_ADDRESS);
  Wire.onRequest(requestEvent);
  Serial.begin(9600);
  pinMode(button_pin, INPUT_PULLUP);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }

  Serial.println(F("Reading sensor parameters"));
  finger.getParameters();
  Serial.print(F("Status: 0x")); Serial.println(finger.status_reg, HEX);
  Serial.print(F("Sys ID: 0x")); Serial.println(finger.system_id, HEX);
  Serial.print(F("Capacity: ")); Serial.println(finger.capacity);
  Serial.print(F("Security level: "));
  Serial.println(finger.security_level);
  Serial.print(F("Device address: ")); Serial.println(finger.device_addr,
  HEX);
  Serial.print(F("Packet len: ")); Serial.println(finger.packet_len);
  Serial.print(F("Baud rate: ")); Serial.println(finger.baud_rate);

  pinMode(led_pin, OUTPUT);
  pinMode(trig_pin, OUTPUT); // Sets the trigPin as an Output

```

```

    pinMode(echo_pin, INPUT); // Sets the echoPin as an Input echoPin as an
    Input
    lcd.init();
    lcd.clear();
    lcd.backlight();          // Make sure backlight is on
    lcd.setCursor(0,1);
    lcd.print("Status: ");
    lcd.setCursor(0,0);      //Set cursor to character 2 on line 0
    lcd.print("Available: ");
}
uint8_t readnumber(void) {
    uint8_t num = 0;

    while (num == 0) {
        while (! Serial.available());
        num = Serial.parseInt();
    }
    return num;
}

```

Kode di atas adalah program C++ untuk mengambil dan memproses data dari sensor untuk open access system kami. Program ini menggunakan 2 sensor yaitu sensor ultrasonik HC - SRO4 dan fingerprint sensor, sebuah push button, dan dua aktuator yaitu servo dan LED.

Kami pertama mendefinisikan beberapa library dan variabel. Library yang digunakan antara lain adalah "Adafruit_Fingerprint" yang digunakan untuk mengakses fingerprint sensor, "LiquidCrystal_I2C" yang digunakan untuk menampilkan informasi pada LCD, dan "Wire" yang digunakan untuk komunikasi I2C. Kemudian beberapa variabel juga didefinisikan, seperti "led_pin" yang digunakan untuk menentukan pin untuk LED, "trig_pin" dan "echo_pin" yang digunakan untuk sensor ultrasonik, "button_pin" yang digunakan untuk tombol, dan "lock_state" yang merupakan status kunci.

Pada bagian setup, dilakukan inisialisasi beberapa variabel dan sensor yang digunakan, seperti fingerprint sensor dan LCD. Selain itu, dilakukan juga pengecekan koneksi dengan fingerprint sensor. Kemudian, juga dilakukan setting pada pin LED, sensor ultrasonik, dan tombol. Fungsi "readnumber" digunakan untuk membaca input dari user yang dikirim melalui serial monitor.


```

uint8_t getFingerprintEnroll() {

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

    Serial.println("Remove finger");
    delay(2000);
    p = 0;
    while (p != FINGERPRINT_NOFINGER) {
        p = finger.getImage();
    }
    Serial.print("ID "); Serial.println(id);
    p = -1;
    Serial.println("Place same finger again");
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.print(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                break;
            default:
                Serial.println("Unknown error");
                break;
        }
    }

    // OK success!

    p = finger.image2Tz(2);
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image converted");
            break;
        case FINGERPRINT_IMAGEMESS:

```

```

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

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

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

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

```

```

        return p;
    } else {
        Serial.println("Unknown error");
        return p;
    }

    return true;
}

```

Fungsi bernama "getFingerprintEnroll()" kami bertujuan untuk melakukan proses enrolment atau pendaftaran sidik jari ke dalam sebuah database sidik jari menggunakan sensor sidik jari. Fungsi ini memiliki tipe data kembalian uint8_t, yang merupakan tipe data unsigned integer 8 bit.

Proses enrolment dimulai dengan menunggu hingga sensor sidik jari mendapatkan gambar sidik jari yang valid. Hal ini dilakukan dengan menggunakan perintah "getImage()" dari objek "finger". Jika sidik jari sudah valid, maka akan muncul pesan "Image taken" di dalam serial monitor. Selanjutnya, gambar sidik jari yang sudah diambil akan dikonversi ke dalam bentuk template sidik jari menggunakan perintah "image2Tz(1)". Proses konversi ini harus berhasil dilakukan agar proses enrolment dapat dilanjutkan. Jika proses konversi gagal, maka fungsi akan mengembalikan nilai p berupa kode kesalahan yang terkait dengan penyebab gagalnya konversi. Setelah konversi berhasil dilakukan, user akan diminta untuk menghapus sidik jari dari sensor sidik jari. Proses ini dilakukan dengan meminta user untuk melepas jari dari sensor sidik jari dan menunggu selama 2 detik dengan perintah "delay(2000)".

Setelah sidik jari dihapus dari sensor, user diminta untuk menempatkan kembali sidik jari yang sama ke sensor sidik jari. Proses yang sama akan dilakukan seperti pada tahap sebelumnya, yaitu menunggu hingga sensor sidik jari mendapatkan gambar sidik jari yang valid dan melakukan konversi gambar tersebut ke dalam bentuk template sidik jari. Jika proses konversi berhasil dilakukan, maka fungsi akan menciptakan model sidik jari menggunakan perintah "createModel()". Jika pembuatan model berhasil dilakukan, maka user akan diberitahu bahwa sidik jari telah terdaftar dengan pesan "Prints matched!".

Terakhir, model sidik jari yang telah dibuat akan disimpan ke dalam database menggunakan perintah "storeModel(id)". Jika penyimpanan berhasil dilakukan, user akan diberitahu bahwa sidik jari telah berhasil disimpan dengan pesan "Stored!". Namun, jika penyimpanan gagal, maka fungsi akan mengembalikan nilai p berupa kode kesalahan yang terkait dengan penyebab gagalnya penyimpanan.

```
uint8_t getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println("No finger detected");
            delay(500);
            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!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of ");
Serial.println(finger.confidence);
if (finger.confidence > 100){
    lcd.setCursor(10,1);
    lcd.print("OPENED");
    lock_state = 1;
    delay(15000);
    Serial.write("CLOSE");
    lcd.setCursor(10,1);
    lock_state = 0;
}else{

}

delay(15);
return finger.fingerID;
}

```

Fungsi bernama “getFingerprintID()” kami berfungsi untuk mendapatkan ID fingerprint dari modul fingerprint sensor. Fungsi ini akan mengembalikan nilai ID fingerprint jika berhasil ditemukan dan cocok dengan yang terdaftar di database, serta memberikan pesan kesalahan jika terjadi kesalahan dalam prosesnya.

Pertama, fungsi ini akan memanggil fungsi getImage() dari objek finger untuk mengambil gambar sidik jari, kemudian nilai kembalian dari fungsi ini akan diproses melalui sebuah switch case untuk menentukan apakah gambar berhasil diambil atau tidak. Jika gambar berhasil diambil, maka akan ditampilkan pesan "Image taken". Namun jika tidak ada jari yang terdeteksi, maka akan ditampilkan pesan "No finger detected" dan nilai kembalian dari fungsi ini adalah FINGERPRINT_NOFINGER.

Selanjutnya, gambar sidik jari yang sudah diambil akan dikonversi ke dalam template dengan memanggil fungsi image2Tz() dari objek finger. Seperti sebelumnya, nilai kembalian dari fungsi ini akan diproses melalui switch case untuk menentukan apakah konversi berhasil atau tidak. Jika berhasil, maka akan ditampilkan pesan "Image converted". Namun jika gambar terlalu buram atau tidak ditemukan fitur sidik jari, maka akan ditampilkan pesan kesalahan dan nilai kembalian dari fungsi ini adalah FINGERPRINT_IMAGEMESS atau FINGERPRINT_FEATUREFAIL atau FINGERPRINT_INVALIDIMAGE. Selanjutnya, template sidik jari yang sudah dihasilkan akan dicocokkan dengan database sidik jari menggunakan fungsi fingerSearch(). Jika terdapat sidik jari yang cocok, maka akan ditampilkan pesan "Found a print match!". Namun jika terjadi kesalahan dalam komunikasi atau sidik jari tidak cocok, maka akan ditampilkan pesan kesalahan dan nilai kembalian dari fungsi ini adalah FINGERPRINT_PACKETRECEIVEERR atau FINGERPRINT_NOTFOUND.

Jika sidik jari berhasil ditemukan dan cocok dengan database, maka fungsi ini akan mengembalikan nilai ID sidik jari dan menampilkan pesan "Found ID #" disertai dengan nomor ID dan tingkat kepercayaan. Jika tingkat kepercayaan lebih dari 100, maka fungsi ini akan mengubah status kunci menjadi terbuka, menampilkan pesan "OPENED" pada LCD, dan menunggu selama 15 detik sebelum mengirimkan pesan "CLOSE" ke perangkat lain untuk mengunci kembali kunci. Jika tingkat kepercayaan kurang dari atau sama dengan 100, maka fungsi ini tidak melakukan apa-apa di dalam blok else. Terakhir, fungsi ini akan menunggu selama 15 milidetik dan mengembalikan nilai ID sidik jari.

```

void requestEvent(){
    Wire.write(lock_state);
}

void loop() {
    digitalWrite(trig_pin, LOW);
    delayMicroseconds(5);
    digitalWrite(trig_pin, HIGH);
    delayMicroseconds(15);
    digitalWrite(trig_pin, LOW);
    duration = pulseIn(echo_pin, HIGH);
    // Calculating the distance
    int x = duration ;
    distance = (3*pow(10,-6)*pow(x,2) + 0.0139*x + 0.2854);
    // Prints the distance on the Serial Monitor
    // Print a message on both lines of the LCD.

    lcd.setCursor(10,1);
    lcd.print("LOCKED");
    if (distance >30){
        state = "NAH";
        digitalWrite(led_pin, LOW);
        lcd.setCursor(10,0);
        lcd.print(state);
    }else{
        state = "YES";
        digitalWrite(led_pin, HIGH);
        lcd.setCursor(10,0);
        lcd.print(state);
    }

    if (digitalRead(button_pin) == HIGH){
        getFingerprintID();
        delay(50);
    }else
    {
        Serial.println("Ready to enroll a fingerprint!");
        Serial.println("Please type in the ID # (from 1 to 127) you want to
save this finger as...");
        id = readnumber();
        if (id == 0) // ID #0 not allowed, try again!
            return;
    }
}

```



```

Serial.print("Enrolling ID #");
Serial.println(id);

while (! getFingerprintEnroll() );
}

}

```

Kode di atas adalah sebuah sistem yang menggunakan sensor jarak, LCD, LED, tombol, dan pemindai sidik jari untuk mengontrol sebuah kunci pintu elektronik. Code ini berisi beberapa fungsi yang mengambil data dari sensor jarak, menampilkan data pada layar LCD, mengontrol LED, dan memproses input dari tombol dan pemindai sidik jari.

Kode di atas menjalankan pemindai sidik jari dan sensor proximity untuk mengontrol status kunci. Fungsi `requestEvent()` digunakan untuk mengirimkan sinyal ke perangkat terhubung menggunakan protokol I2C. Fungsi `loop()` membaca data dari sensor proximity untuk menentukan jarak sebuah objek dan menampilkan hasilnya di layar LCD. Program ini juga memeriksa apakah tombol ditekan dan memanggil fungsi `getFingerprintID()` untuk mengidentifikasi sidik jari seseorang. Jika sidik jari tidak teridentifikasi, program akan meminta pengguna untuk mendaftarkan sidik jarinya. Jika jarak objek kurang dari atau sama dengan 30, program akan menyalakan LED dan mengatur status kunci menjadi "LOCKED". Jika tombol ditekan dan sidik jari yang valid teridentifikasi, program akan membuka kunci dengan mematikan LED dan mengirimkan sinyal "CLOSE" ke perangkat terhubung control LED, dan memproses input dari tombol dan pemindai sidik jari.

b. Arduino Aktuator

```

#include <Servo.h>
#include <Wire.h>
#define SLAVE1_ADDRESS 0x9A

Servo myservo;
int pos = 0;
int led_pin = 13;
int lock_state = 0;

```

```

void setup()
{
    Wire.begin();
    Serial.begin(9600);
    myservo.attach(9);
    pinMode(led_pin, OUTPUT);
}

void loop()
{
    Wire.requestFrom(SLAVE1_ADDRESS, 1);
    if (Wire.available()){
        lock_state = Wire.read();
        Serial.println(lock_state);
    }
    delay(500);

    if (lock_state == 1){
        digitalWrite(13, HIGH);
        myservo.write(90);
        delay(15000);
        myservo.write(-90);
    }
    if (lock_state == 0){
        digitalWrite(13, LOW);
        delay(500);
    }
}

```

Kode di atas akan menjalankan sebuah servo motor dan lampu LED berdasarkan nilai dari variabel "lock_state" yang diterima melalui komunikasi I2C dari perangkat lain. Program ini menggunakan library "Servo.h" dan "Wire.h". Pada bagian setup, program memulai komunikasi I2C dan serial, dan menghubungkan pin 9 ke servo motor dan pin 13 ke lampu LED. Pada loop, program membaca nilai lock_state yang diterima melalui I2C dan memeriksa apakah nilainya adalah 1 atau 0. Jika nilainya adalah 1, program akan menyalakan LED, menggerakkan servo motor ke posisi 90 derajat selama 15 detik, dan

kemudian menggerakkan kembali servo motor ke posisi awal. Jika nilainya adalah 0, program akan mematikan LED selama 500 milidetik.

2.4 Perangkat Lunak

a. TinkerCAD

TinkerCAD merupakan perangkat lunak yang kami gunakan untuk memvisualisasikan perancangan model 3D berbasis web. Kami tidak melakukan dan pengujian kami di tinkerCAD dikarenakan ada model yang tidak ada di dalamnya seperti fingerprint. Jadi kami hanya menggunakan tinkerCAD untuk memvisualisasikan rangkaian kami secara 3D untuk mempermudah jika ada orang lain yang ingin membuat proyek sama seperti kami.

b. *Software* Arduino Uno

Software Arduino Uno merupakan perangkat lunak yang sangat membantu kami dalam mengembangkan proyek ini. Di sini kami menggunakan papan board Arduino Uno untuk mengontrol berbagai macam komponen elektronik kami yaitu ada sensor, LED, Push Button dan LCD. Software Arduino memungkinkan kami untuk membuat program yang dapat dijalankan pada papan board Arduino Uno, sementara papan board Arduino Uno menyediakan lingkungan fisik yang dibutuhkan untuk menjalankan program tersebut pada perangkat elektronik.

BAB 3

KESIMPULAN DAN SARAN

3.1 Kesimpulan

Dalam kesimpulan proyek ini, dapat disimpulkan bahwa sistem teknologi keamanan dengan menggunakan fingerprint dapat memastikan keamanan suatu barang. Dengan menggunakan sistem ini, hanya orang-orang yang terdaftar dalam database yang dapat mengakses kotak keamanan tersebut. Selain itu, sistem juga memberikan informasi mengenai keberadaan barang di dalam kotak, sehingga pengguna dapat dengan mudah mengetahui apakah barang tersebut ada di dalam kotak atau tidak. Keuntungan lain dari sistem teknologi keamanan ini adalah penggunaannya yang mudah dan praktis. Dalam waktu singkat, pengguna dapat membuka dan menutup kotak hanya dengan menggunakan fingerprint. Dalam hal ini, sistem teknologi keamanan dengan menggunakan fingerprint dapat membantu meningkatkan tingkat keamanan barang, sehingga pengguna dapat merasa lebih aman dan nyaman dalam menyimpan barang berharga mereka.

3.2 Saran

Saran kami adalah untuk meningkatkan fitur database pada sistem dan menambahkan fitur identifikasi pengguna menggunakan teknologi fingerprint setelah berhasil melakukan verifikasi ID identitas. Fitur ini akan memungkinkan pengguna untuk memasukkan data pribadi mereka dengan aman dan menyimpannya di database yang lebih besar dan terjamin keamanannya. Dengan demikian, sistem akan lebih aman dan pengguna dapat memastikan bahwa barang yang mereka pinjam akan terjaga dengan baik oleh pihak yang bertanggung jawab. Hal ini sangat penting untuk memastikan keamanan dan kepercayaan pengguna terhadap sistem serta menjamin integritas data yang ada dalam sistem.

LAMPIRAN

I. Bill of Material

No.	Nama Komponen	Tipe Komponen	Jumlah Unit	Harga/pc	Link Pembelian
1.	Arduino + kabel	UNO	2	Rp129.900	https://tokopedia.link/F3jVnND7lzb
2.	Breadboard (30 rows)	-	1	Rp. 8.900	https://www.tokopedia.com/supreme/mb-102-400-hole-breadboard-arduino-prototyping
3.	Fingerprint Sensor	-	1	Rp. 119.000	https://www.tokopedia.com/cncstorebandung/fingerprint-sensor-recognition-module-as608-sidik-jari-for-arduino
4.	Set kabel jumper	-	1	Rp. 12.500	https://tokopedia.link/EB1eB7K7lzb
5.	Button	-	1	Rp. 350	https://tokopedia.link/tw4VICX7lzb
6.	LCD	16x4	1	Rp.65.000	https://tokopedia.link/ynuVezK7lzb
7.	LED	YL-69	1	Rp.56.000	https://www.tokopedia.com/tokocemarasewu/super-murah-lampu-led-kecil-untuk-praktik

					um-rangkain-listrik-generator
8.	Micro Servo	DHT22	1	Rp. 15.300	https://www.tokopedia.com/xyzhobby/tower-pro-micro-servo-sg-90-9g-motor-servo-sg90-towerpro
9.	Sensor UltraSonik	HC-Sr04	1	Rp. 10.800	https://www.tokopedia.com/dc-motor-shop/hc-sr04-ultrasonic-module-sensor-ultra-sonic-pengukur-jarak-ultrasonik
10.	Resistor	8KOhm	1	Rp. 10.000	https://www.tokopedia.com/nanoelectronic/100pcs-resistor-8kohm-8k-ohm-1-4w-1-metal-film

II. Kontribusi Kelompok

Penilaian mandiri oleh masing-masing individu di dalam sebuah kelompok:

- Anggota kelompok 1: 212100120, Grace Theofie
- Anggota kelompok 2: 212100370, Mastri tasya Napitupulu
- Anggota kelompok 3: 212100199, Martin Emmanuel Chang
- Anggota kelompok 4: 212100332, Priscilla Green Samosir

	Kontribusi menurut	Kontribusi menurut	Kontribusi menurut	Kontribusi menurut	Rata-rata kontribusi anggota
--	--------------------	--------------------	--------------------	--------------------	------------------------------

	anggota kelompok 1	anggota kelompok 2	anggota kelompok 3	anggota kelompok 4	kelompok
Anggota kelompok 1	25%	25%	25%	25%	25%
Anggota kelompok 2	25%	25%	25%	25%	25%
Anggota kelompok 3	25%	25%	25%	25%	25%
Anggota kelompok 4	25%	25%	25%	25%	25%
	100%	100%	100%	100%	100%