

LAMPIRAN

LAMPIRAN 1 Surat Kesediaan Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Yerry Febrian Sebanise, M.Kom.
NIDN : 0613028602
NIPY : 12.03.110
Jabatan Struktural : Sekertaris Program Studi
Jabatan Fungsional : Asisten Ahli

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Aninda Kusmeiga
NIM : 23041001
Progran Studi : D3 Teknik Komputer
Judul TA : RANCANG BANGUN SMART LOCKER
BERBASIS SIDIK JARI DAN ESP32 PADA PT.
TEGAL JAYAMAKMUR SEJAHTERA

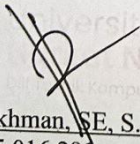
Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

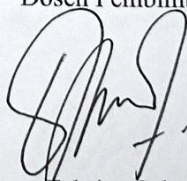
Tegal, 21 April 2026

Mengetahui

Ka. Prodi DIII Teknik Komputer,

Dosen Pembimbing II,


Arif Rakhman, SE, S.Pd, M.Kom
NIPY.05.016.291


Yerry Febrian Sebanise M.Kom.
NIPY. 12.03.110

LAMPIRAN 2 Surat Kesediaan Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : M. Teguh Prihandoyo, M.Kom
NIDN :
NIPY : 02.005.012
Jabatan Struktural :
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Aninda Kusmeiga
NIM : 23041001
Progran Studi : D3 Teknik Komputer
Judul TA : RANCANG BANGUN SMART LOCKER
BERBASIS SIDIK JARI DAN ESP32 PADA PT.
TEGAL JAYAMAKMUR SEJAHTERA

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 21 April 2026

Mengetahui

Ka. Prodi DIII Teknik Komputer,

Dosen Pembimbing I,


Arif Rahman, SE, S.Pd, M.Kom
NIPY.05.016.291


M. Teguh Prihandoyo M.Kom.
NIPY. 02.005.012

LAMPIRAN 3 Foto Dokumentasi Pembuatan Alat



LAMPIRAN 4 Foto Alat Bagian Depan



LAMPIRAN 5 Foto Alat Bagian Belakang



LAMPIRAN 6 Foto Dokumentasi Observasi



LAMPIRAN 7 Foto Dokumentasi Jadwal Bimbingan

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	22/05/2022	- Laporan TA BAB 1-3 - Revisi latar belakang - Revisi rumusan masalah - Perbaiki halaman	
2.	25/05/2022	- Revisi italic untuk bahasa asing	
3.	01/06/2022	- Revisi penomoran	
4.	02/06/2022	- BAB 1-3 Acc	

PEMBIMBING II:

BIMBINGAN LAPORAN TA

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	2/06/2026	Laporan TA Bab 4 - Perbaiki Flowchart - Menambahkan penahan pintu	
2.	4/06/2026	- Perbaiki Spasi - Merapikan Kabel	
3.	12/06/2026	- Revisi Rangkaian Sistem	
4	19/06/2026	- Menambahkan implementasi perangkat lunak	
5	17/06/2026	- Memperbaiki hasil pengujian	

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
6.	25/06/2026	BAB IV Aa	
7.	26/06/2026	BAB V + VI Aa	

LAMPIRAN 8 Foto Surat Observasi

 Universitas
Harkat Negeri

Sekolah Vokasi
Program Studi D-3 Teknik Komputer

Nomor : S37.03/KOM-HN/XII/2025
Lampiran :
Perihal : Permohonan Izin Observasi Tugas Akhir (TA)

Kepada Yth.
Pimpinan PT. Tegal Jaya Makmur Sejahtera
Di Jl. Raya No. KM. 5.5, kedondon, Munjung Agung, Kec. Kramat, Kabupaten Tegal, Jawa
Tengah 52181

Dengan Hormat,

Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester V Program Studi Diploma III Teknik Komputer Universitas Harkat Negeri. Maka dengan ini kami mengajukan izin observasi pengambilan data di PT. Tegal Jaya Makmur Sejahtera yang Bapak / Ibu Pimpin, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:

No	NIM	NAMA	NO HP
1	23041001	ANINDA KUSMEIGA	083115878047
2	23041002	SITI NUR FATIKHATUL AWALIYAH	082324052199

Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.

Tegal, 05 Mei 2026
Ka. Prodi DIII Teknik Komputer
Universitas Harkat Negeri


Arif Rakhman SE, S.Pd, M.Kom
NIPY. 05.016.291

Program Studi D-3 Teknik Komputer, Sekolah Vokasi
Jedung Utama Rukung B-33 Kampus Madurejo, Jalan Madurejo No 1 Kota Tegal
WhatsApp: +62-85270213673 | Email: komputer@harkatnegeri.ac.id

LAMPIRAN 9 Script Program Pada Alat

```
#include <WiFi.h>
#include <Firebase_ESP_Client.h>

#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
#include <time.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// =====
// SMART LOCKER 3 LOKER
// 1 FINGERPRINT + 3 RELAY + LCD I2C + BUZZER
// ESP32-S3
// =====

// =====
// PIN RELAY
// =====

#define RELAY_LOKER1 4
#define RELAY_LOKER2 5
#define RELAY_LOKER3 6

// =====
// PIN FINGERPRINT AS608 (UART2 - tidak bentrok dengan
// Serial debug yang memakai UART0)
// =====

#define FP_RX 17
#define FP_TX 18
```

```

// =====

// PIN LCD I2C

// =====

#define LCD_SDA 15
#define LCD_SCL 16

// Alamat LCD biasanya 0x27, jika tidak tampil coba 0x3F
#define LCD_ADDR 0x27

#define LCD_COLS 16
#define LCD_ROWS 2

// =====

// PIN BUZZER

// =====

#define BUZZER_PIN 8

// =====

// WIFI

// =====

#define WIFI_SSID "HelmiNinda"
#define WIFI_PASSWORD "123nadira"

// =====

// FIREBASE

// =====

#define API_KEY "AlzaSyC_FYvEPCWUPWvqKTj5avwpQUcE7anAlic"
#define DATABASE_URL "smartlocker-47f5c-default-rtdb.asia-
southeast1.firebaseio.com"

// =====

```

```

// OBJECT

// =====

HardwareSerial FP(2);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&FP);

LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// =====

// DEKLARASI FUNCTION

// =====

bool enrollFingerprint(int id);
bool deleteFingerprint(int id);
bool gantiFingerprint(int id);

void cekEnrollRequest();
void cekDeleteRequest();
void bacaPerintahSerial();

int scanFinger();

void prosesAksesBerhasil(int fingerID);
void bukaLoker(int lokerId, String namaUser);

bool cariUserByFingerId(int fingerId, String &namaOut, int &lokerIdOut);
bool cekLCDTerdeteksi();

// =====

```

```

// VARIABEL LCD
// =====

String lcdBaris1Sekarang = "";
String lcdBaris2Sekarang = "";
bool lcdAktif = false;

// =====

// CEK FIREBASE
// =====

bool firebaseReady()
{
    return Firebase.ready();
}

// =====

// CEK LCD TERDETEKSI DI JALUR I2C
// -----

// Mengirim tes ACK langsung ke alamat LCD_ADDR sebelum
// lcd.init() dipanggil. Kalau LCD tidak merespons (misal
// karena VCC belum stabil / kurang tegangan), ini akan
// langsung terlihat di Serial Monitor tanpa perlu upload
// sketch scanner terpisah.
// =====

bool cekLCDTerdeteksi()
{
    Wire.beginTransmission(LCD_ADDR);
    byte error = Wire.endTransmission();
    return (error == 0);
}

// =====

```

```

// LCD HELPER

// =====

void tampilkanLCD(String baris1, String baris2)
{
    if (!lcdAktif) return;

    if (baris1.length() > LCD_COLS)
        baris1 = baris1.substring(0, LCD_COLS);

    if (baris2.length() > LCD_COLS)
        baris2 = baris2.substring(0, LCD_COLS);

    if (baris1 == lcdBaris1Sekarang && baris2 == lcdBaris2Sekarang)
        return;

    lcd.clear();

    lcd.setCursor(0, 0);
    lcd.print(baris1);

    lcd.setCursor(0, 1);
    lcd.print(baris2);

    lcdBaris1Sekarang = baris1;
    lcdBaris2Sekarang = baris2;
}

void tampilkanIdle()
{
    tampilkanLCD("Smart Locker", "Tempel Jari...");
}

```

```

// =====
// BUZZER
// =====

void bunyiBuzzer(int jumlah, int durasiMs = 150, int jedaMs = 150)
{
    for (int i = 0; i < jumlah; i++)
    {
        digitalWrite(BUZZER_PIN, HIGH);
        delay(durasiMs);
        digitalWrite(BUZZER_PIN, LOW);

        if (i < jumlah - 1)
            delay(jedaMs);
    }
}

void bunyiSukses() { bunyiBuzzer(1, 200); }
void bunyiGagal() { bunyiBuzzer(2, 120, 120); }
void bunyiProsesSelesai() { bunyiBuzzer(3, 80, 80); }

// =====
// KIRIM LOG FIREBASE
// =====

void kirimLog(String loker, String aktivitas, String user)
{
    if (!firebaseReady()) return;

    time_t now = time(nullptr);

    String path = "/logs/" + String((long long)now * 1000);

```

```

FirebaseJson json;
json.set("nama_loker", loker);
json.set("aktivitas", aktivitas);
json.set("user", user);
json.set("waktu", (long long)now * 1000);

if (!Firebase.RTDB.setJSON(&fbdo, path.c_str(), &json))
{
    Serial.print("Gagal kirim log: ");
    Serial.println(fbdo.errorReason());
}
}

// =====
// CARI USER BERDASARKAN FINGERPRINT ID
// =====

bool cariUserByFingerId(int fingerId, String &namaOut, int &lokerIdOut)
{
    if (!firebaseReady()) return false;

    for (int i = 1; i <= 20; i++)
    {
        String basePath = "/users/user" + String(i);

        if (!Firebase.RTDB.getJSON(&fbdo, basePath))
            continue;

        FirebaseJson *json = fbdo.jsonObjectPtr();
        FirebaseJsonData result;

```

```

int fidTerdaftar = -1;

if (json->get(result, "finger_id"))
{
    fidTerdaftar = result.to<int>();
}

if (fidTerdaftar == fingerId)
{
    if (json->get(result, "nama"))
        namaOut = result.to<String>();
    else
        namaOut = "-";

    if (json->get(result, "loker_id"))
        lokerIdOut = result.to<int>();
    else
        lokerIdOut = 0;

    return true;
}

return false;
}

// =====
// UPDATE STATUS PINTU
// =====

void updatePintu(int nomorLoker, String status)
{

```

```

if (!firebaseReady()) return;

String path = "/lockers/" + String(nomorLoker) + "/pintu";

if (!Firebase.RTDB.setString(&fbdo, path.c_str(), status))
{
    Serial.print("Gagal update pintu: ");
    Serial.println(fbdo.errorReason());
}
}

// =====
// TIMER
// =====

unsigned long lastFailedLog = 0;
unsigned long lastPollRequest = 0;
const unsigned long POLL_INTERVAL_MS = 5000;

// =====
// SETUP
// =====

void setup()
{
    Serial.begin(115200);
    delay(1000);

    Serial.println();
    Serial.println("=====");
    Serial.println("SMART LOCKER");
    Serial.println("1 FINGERPRINT SHARED");
    Serial.println("LCD I2C + BUZZER");

```

```

Serial.println("=====");

// BUZZER
pinMode(BUZZER_PIN, OUTPUT);
digitalWrite(BUZZER_PIN, LOW);

// =====
// LCD I2C
// =====

Serial.println("Memulai LCD I2C...");

Wire.begin(LCD_SDA, LCD_SCL);

// Beri waktu lebih untuk LCD stabil setelah power-on
// sebelum dikirim perintah init pertama kali.
delay(300);

if (cekLCDTerdeteksi())
{
    Serial.println("LCD terdeteksi di alamat 0x27 (I2C ACK OK)");
    lcdAktif = true;

    // Beberapa modul LiquidCrystal_I2C perlu init() dipanggil
    // dua kali agar controller HD44780 benar-benar ter-reset
    // setelah power-on. Memanggil sekali saja kadang membuat
    // backlight menyala tapi karakter tidak pernah tampil.
    lcd.init();
    delay(50);
    lcd.init();

    lcd.backlight();

```

```

lcd.clear();

lcd.setCursor(0, 0);
lcd.print("Smart Locker");
lcd.setCursor(0, 1);
lcd.print("Booting...");
}
else
{
    // LCD tidak merespons sama sekali di jalur I2C. Ini BUKAN
    // masalah kode - kemungkinan besar VCC LCD belum mendapat
    // tegangan yang cukup (idealnya 5V, cek jalur suplai LCD),
    // atau wiring SDA/SCL/GND belum benar. Sistem tetap
    // dilanjutkan tanpa LCD (mode tanpa layar) supaya loker
    // tetap berfungsi walau LCD bermasalah.
    Serial.println("LCD TIDAK TERDETEKSI! Cek VCC/wiring LCD.");
    Serial.println("Sistem tetap lanjut tanpa LCD.");
    lcdAktif = false;
}

delay(1500);

// WIFI
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Menghubungkan WiFi");

tampilkanLCD("Connecting WiFi", "Mohon Tunggu...");

while (WiFi.status() != WL_CONNECTED)
{

```

```

    delay(500);
    Serial.print(".");
}

Serial.println();
Serial.println("WiFi Connected");

tampilkanLCD("WiFi Connected", "");
delay(1000);

// WAKTU NTP
configTime(25200, 0, "pool.ntp.org");

Serial.println("Menunggu waktu internet...");
tampilkanLCD("Sinkron Waktu", "Mohon Tunggu...");

time_t now = time(nullptr);

while (now < 100000)
{
    delay(500);
    Serial.print(".");
    now = time(nullptr);
}

Serial.println();
Serial.println("Waktu berhasil sinkron");

// FIREBASE
config.api_key = API_KEY;
config.database_url = DATABASE_URL;

```

```

if (Firebase.signUp(&config, &auth, "", ""))
{
    Serial.println("Anonymous Sign Up OK");
}
else
{
    Serial.printf("Sign Up Error : %s\n",
        config.signer.signupError.message.c_str());
}

Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);

fbdo.setBSSLBufferSize(1024, 512);

tampilkanLCD("Firebase", "Menghubungkan...");

Serial.print("Menunggu token Firebase");

unsigned long tStart = millis();

while (!Firebase.ready() && millis() - tStart < 15000)
{
    delay(300);
    Serial.print(".");
}

Serial.println();

if (Firebase.ready())

```

```

{
  Serial.println("Firebase READY");
  Serial.print("Free heap: ");
  Serial.println(ESP.getFreeHeap());

  tampilkanLCD("Firebase READY", "");

  Firebase.RTDB.setString(&fbdo, "/test", "ESP32 Connected");

  delay(1000);
}
else
{
  Serial.println("Firebase belum READY");

  tampilkanLCD("Firebase GAGAL", "Mode Offline");

  delay(1500);
}

// RELAY
Serial.println("Inisialisasi Relay...");

pinMode(RELAY_LOKER1, OUTPUT);
pinMode(RELAY_LOKER2, OUTPUT);
pinMode(RELAY_LOKER3, OUTPUT);

digitalWrite(RELAY_LOKER1, HIGH);
digitalWrite(RELAY_LOKER2, HIGH);
digitalWrite(RELAY_LOKER3, HIGH);

```

```

Serial.println("RELAY OK");

// FINGERPRINT
FP.begin(57600, SERIAL_8N1, FP_RX, FP_TX);
finger.begin(57600);

Serial.println();
Serial.println("=====");
Serial.println("CEK SENSOR FINGERPRINT");
Serial.println("=====");

tampilkanLCD("Cek Sensor", "Fingerprint...");

bool fpOk = finger.verifyPassword();

if (fpOk)
{
    Serial.println("Fingerprint OK");
    tampilkanLCD("Sensor FP", "OK");
    bunyiSukses();

    Serial.print("Jumlah template tersimpan: ");
    Serial.println(finger.getTemplateCount());
}
else
{
    Serial.println("Fingerprint ERROR");
    tampilkanLCD("Sensor FP", "ERROR!");
    bunyiGagal();
}

```

```

delay(1500);

// SMART LOCKER READY
Serial.println();
Serial.println("SMART LOCKER READY");

updatePintu(1, "tertutup");
updatePintu(2, "tertutup");
updatePintu(3, "tertutup");

tampilkanIdle();

Serial.println();
Serial.println("Perintah serial yang tersedia:");
Serial.println(" enroll<id> -> daftar sidik jari baru di ID tsb");
Serial.println(" delete<id> -> hapus sidik jari di ID tsb");
Serial.println(" ganti<id> -> hapus lalu langsung daftar baru di ID tsb");
Serial.println(" contoh: ganti1 ganti3");
}

// =====
// CEK ENROLL REQUEST
// =====

void cekEnrollRequest()
{
    if (!firebaseReady()) return;

    int fingerId = 0;
    String status = "";

    if (Firebase.RTDB.getInt(&fbdo, "/enroll_request/finger_id"))

```

```

fingerId = fbdo.intData();

if (Firebase.RTDB.getString(&fbdo, "/enroll_request/status"))
    status = fbdo.stringData();

if (status == "waiting")
{
    Serial.println("REQUEST ENROLL DITERIMA");

    tampilkanLCD("Enroll Sidik Jari", "ID: " + String(fingerId));

    Firebase.RTDB.setString(&fbdo, "/enroll_request/status", "processing");
    Firebase.RTDB.setString(&fbdo, "/enroll_request/message", "Tempelkan jari");

    bool berhasil = enrollFingerprint(fingerId);

    if (berhasil)
    {
        tampilkanLCD("Enroll Berhasil", "ID: " + String(fingerId));
        bunyiProsesSelesai();

        Firebase.RTDB.setString(&fbdo, "/enroll_request/status", "success");
        Firebase.RTDB.setString(&fbdo, "/enroll_request/message", "Fingerprint
berhasil");
    }
    else
    {
        tampilkanLCD("Enroll Gagal", "ID: " + String(fingerId));
        bunyiGagal();

        Firebase.RTDB.setString(&fbdo, "/enroll_request/status", "failed");
    }
}

```

```

        Firebase.RTDB.setString(&fbdo, "/enroll_request/message", "Fingerprint
gagal");
    }

    delay(1500);
    tampilkanIdle();
}
}

// =====
// CEK DELETE REQUEST
// =====

void cekDeleteRequest()
{
    if (!firebaseReady()) return;

    int fingerId = 0;
    String status = "";

    if (Firebase.RTDB.getInt(&fbdo, "/delete_request/finger_id"))
        fingerId = fbdo.intData();

    if (Firebase.RTDB.getString(&fbdo, "/delete_request/status"))
        status = fbdo.stringData();

    if (status == "pending")
    {
        Serial.println("REQUEST DELETE DITERIMA");

        tampilkanLCD("Hapus Sidik Jari", "ID: " + String(fingerId));
    }
}

```

```

Firebase.RTDB.setString(&fbdo, "/delete_request/status", "processing");

bool berhasil = deleteFingerprint(fingerId);

if (berhasil)
{
    tampilkanLCD("Hapus Berhasil", "ID: " + String(fingerId));
    bunyiProsesSelesai();

    Firebase.RTDB.deleteNode(&fbdo, "/delete_request");
}
else
{
    tampilkanLCD("Hapus Gagal", "ID: " + String(fingerId));
    bunyiGagal();

    Firebase.RTDB.setString(&fbdo, "/delete_request/status", "failed");
    Firebase.RTDB.setString(&fbdo, "/delete_request/message", "Hapus
fingerprint gagal");
}

delay(1500);
tampilkanIdle();
}
}

// =====
// LOOP
// =====

void loop()
{

```

```

if (millis() - lastPollRequest > POLL_INTERVAL_MS)
{
    lastPollRequest = millis();

    cekEnrollRequest();
    cekDeleteRequest();
}

int fingerID = scanFinger();

if (fingerID >= 0)
{
    prosesAksesBerhasil(fingerID);
}

    bacaPerintahSerial();
}

// =====
// SERIAL COMMAND
// =====

void bacaPerintahSerial()
{
    if (!Serial.available())
        return;

    String cmd = Serial.readStringUntil('\n');
    cmd.trim();

    if (cmd.startsWith("ganti"))
    {

```

```

int id = cmd.substring(5).toInt();

Serial.println(gantiFingerprint(id) ? "Ganti sidik jari berhasil" : "Ganti sidik jari
gagal");
}
else if (cmd.startsWith("enroll"))
{
int id = cmd.substring(7).toInt();

Serial.println(enrollFingerprint(id) ? "Enroll berhasil" : "Enroll gagal");
}
else if (cmd.startsWith("delete"))
{
int id = cmd.substring(7).toInt();

Serial.println(deleteFingerprint(id) ? "Delete berhasil" : "Delete gagal");
}
}

// =====
// SCAN FINGERPRINT
// =====

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

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

p = finger.fingerSearch();

```

```

if (p == FINGERPRINT_OK)
{
    Serial.print("ID TERDETEKSI : ");
    Serial.println(finger.fingerID);

    return finger.fingerID;
}

if (p == FINGERPRINT_NOTFOUND)
{
    if (millis() - lastFailedLog > 5000)
    {
        lastFailedLog = millis();

        Serial.println("AKSES GAGAL");

        tampilkanLCD("Akses Ditolak", "Sidik Jari Salah");
        bunyiGagal();

        kirimLog("-", "akses gagal", "-");

        delay(1200);
        tampilkanIdle();
    }
}

return -1;
}

// =====

```

```

// PROSES AKSES BERHASIL
// =====
void prosesAksesBerhasil(int fingerID)
{
    if (fingerID < 1 || fingerID > 3)
    {
        Serial.println("ID TIDAK TERDAFTAR");

        tampilkanLCD("Akses Ditolak", "ID Tidak Terdaftar");
        bunyiGagal();

        kirimLog("-", "akses gagal", "-");

        delay(1500);
        tampilkanIdle();

        return;
    }

    int lokerId = fingerID;
    String namaUser = "-";
    int lokerUser = 0;

    if (cariUserByFingerId(fingerID, namaUser, lokerUser))
    {
        Serial.print("Nama User: ");
        Serial.println(namaUser);

        if (lokerUser >= 1 && lokerUser <= 3)
        {
            lokerId = lokerUser;

```

```

    }
}
else
{
    Serial.println("Nama user tidak ditemukan");
}

    bukaLoker(lokId, namaUser);
}

// =====
// BUKA LOKER
// =====

void bukaLoker(int lokId, String namaUser)
{
    int pinRelay;

    String namaLoker = "Loker " + String(lokId);

    if (lokId == 1) pinRelay = RELAY_LOKER1;
    else if (lokId == 2) pinRelay = RELAY_LOKER2;
    else if (lokId == 3) pinRelay = RELAY_LOKER3;
    else return;

    Serial.println(namaLoker + " TERBUKA");

    tampilkanLCD(namaLoker + " Terbuka", namaUser);
    bunyiSukses();

    updatePintu(lokId, "terbuka");
    kirimLog(namaLoker, "dibuka", namaUser);
}

```

```

digitalWrite(pinRelay, LOW);
delay(5000);
digitalWrite(pinRelay, HIGH);

Serial.println(namaLoker + " TERKUNCI");

updatePintu(lokerId, "tertutup");
 kirimLog(namaLoker, "tertutup", namaUser);

tampilkanLCD(namaLoker, "Terkunci");
delay(1000);

tampilkanIdle();
}

// =====
// ENROLL FINGERPRINT
// =====

bool enrollFingerprint(int id)
{
    int p = -1;
    unsigned long timeout;

    Serial.print("DAFTAR ID : ");
    Serial.println(id);

    Serial.println("Tempelkan Jari");
    tampilkanLCD("Tempelkan Jari", "ID: " + String(id));

    timeout = millis();

```

```

while (p != FINGERPRINT_OK)
{
    p = finger.getImage();

    if (millis() - timeout > 15000)
    {
        Serial.println("TIMEOUT Jari 1");
        return false;
    }
}

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

Serial.println("Angkat Jari");
tampilkanLCD("Angkat Jari", "Tunggu...");
delay(2000);

p = 0;
timeout = millis();

while (p != FINGERPRINT_NOFINGER)
{
    p = finger.getImage();

    if (millis() - timeout > 15000)
    {
        Serial.println("TIMEOUT Angkat Jari");
        return false;
    }
}

```

```

}

Serial.println("Tempelkan Lagi");
tampilkanLCD("Tempelkan Lagi", "ID: " + String(id));

p = -1;
timeout = millis();

while (p != FINGERPRINT_OK)
{
    p = finger.getImage();

    if (millis() - timeout > 15000)
    {
        Serial.println("TIMEOUT Jari 2");
        return false;
    }
}

p = finger.image2Tz(2);
if (p != FINGERPRINT_OK) return false;

if (finger.createModel() == FINGERPRINT_OK)
{
    if (finger.storeModel(id) == FINGERPRINT_OK)
    {
        Serial.println("SIDIK JARI BERHASIL DISIMPAN");
        return true;
    }
    else
    {

```

```

        Serial.println("GAGAL MENYIMPAN ID");
        return false;
    }
}
else
{
    Serial.println("SIDIK JARI TIDAK COCOK");
    return false;
}
}

// =====
// DELETE FINGERPRINT
// =====

bool deleteFingerprint(int id)
{
    uint8_t result = finger.deleteModel(id);

    if (result == FINGERPRINT_OK)
    {
        Serial.print("ID ");
        Serial.print(id);
        Serial.println(" BERHASIL DIHAPUS");
        return true;
    }
    else
    {
        Serial.print("GAGAL MENGHAPUS ID ");
        Serial.print(id);
        Serial.print(" - kode error: 0x");
        Serial.println(result, HEX);
    }
}

```

```

        return false;
    }
}

// =====
// GANTI FINGERPRINT (HAPUS LALU DAFTAR BARU)
// -----
// Menghapus template lama pada ID tertentu, lalu langsung
// memulai proses enroll baru pada ID yang sama. Cocok
// dipakai lewat Serial Monitor: ketik "ganti1" atau
// "ganti3" untuk mengganti sidik jari pada ID tsb.
//
// Kalau ID belum pernah didaftarkan sebelumnya, kegagalan
// deleteModel() tidak menghentikan proses - sistem tetap
// lanjut ke tahap enroll, karena tujuannya memang membuat
// ID tersebut berisi sidik jari yang baru.
// =====

bool gantiFingerprint(int id)
{
    Serial.print("MENGGANTI SIDIK JARI ID : ");
    Serial.println(id);

    tampilkanLCD("Hapus ID Lama", "ID: " + String(id));

    bool hasilHapus = deleteFingerprint(id);

    if (hasilHapus)
    {
        Serial.println("Template lama berhasil dihapus, lanjut daftar baru...");
    }
    else

```

```

    {
        Serial.println("Template lama tidak ditemukan/gagal dihapus, tetap lanjut daftar
baru...");
    }

    delay(500);

    bool hasilEnroll = enrollFingerprint(id);

    if (hasilEnroll)
    {
        tampilkanLCD("Ganti Berhasil", "ID: " + String(id));
        bunyiProsesSelesai();
    }
    else
    {
        tampilkanLCD("Ganti Gagal", "ID: " + String(id));
        bunyiGagal();
    }

    delay(1500);
    tampilkanIdle();

    return hasilEnroll;
}

```