

LAMPIRAN

```
#define WIFI_SSID "robot"
#define WIFI_PASS ""

#include <WiFi.h>
#include <WiFiClient.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <HTTPClient.h>

// === Google Script Webhook URL ===
const char* googleScriptURL =
"https://script.google.com/macros/s/AKfycbyIgjsFghPeBJhgImPcEaWMKTRN
C9xvhbKj60TZhkTpg8Z4DKnoqT2EM9TjITUWdN7K/exec";

// === RFID ===
#define SS_PIN 5
#define RST_PIN 22
MFRC522 rfid(SS_PIN, RST_PIN);

// === LCD ===
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define SDA_PIN 25
#define SCL_PIN 26

// === Motor & Buzzer ===
#define BUZZER_PIN 14
#define IN1 16
#define IN2 4
#define ENA 17
#define PWM_CHANNEL 0
#define PWM_FREQ 1000
#define PWM_RES 8

// === Button ===
#define BUTTON_PIN 27

// === Sensor Infrared ===
#define IR_SENSOR_PIN 33 // Pin sensor IR

// === Data Pengguna ===
```

```

const int UID_COUNT = 4;
String authorizedUIDs[UID_COUNT] = {
    "44 63 0B 05",
    "B9 F6 0C 05",
    "CB 7D DA 05",
    "6A AF DA 05"
};
String userNames[UID_COUNT] = {
    "RONALDO",
    "MESSI",
    "HALLAND",
    "NEYMAR"
};
int accessCounts[UID_COUNT] = {0, 0, 0, 0};

int motorSpeed = 200;

// === Fungsi URL Encode ===
String urlEncode(String str) {
    String encodedString = "";
    char c;
    char code0;
    char code1;
    char code2;
    for (int i = 0; i < str.length(); i++) {
        c = str.charAt(i);
        if (isalnum(c)) {
            encodedString += c;
        } else {
            encodedString += '%';
            code1 = (c >> 4) & 0xF;
            code2 = c & 0xF;
            code0 = code1 + '0';
            if (code1 > 9) code0 = code1 - 10 + 'A';
            encodedString += code0;
            code0 = code2 + '0';
            if (code2 > 9) code0 = code2 - 10 + 'A';
            encodedString += code0;
        }
    }
    return encodedString;
}

// === Kirim ke Google Sheets (Opsional, kalau ada WiFi) ===
void sendToGoogle(String uid, String name, String status) {

```

```

if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;

    String uidEnc = urlencode(uid);
    String nameEnc = urlencode(name);
    String statusEnc = urlencode(status);

    String url = String(googleScriptURL) + "?uid=" + uidEnc +
"&name=" + nameEnc + "&status=" + statusEnc;

    http.begin(url);
    http.GET();
    http.end();
}
// Jika tidak ada WiFi → lewati saja (tidak ada Serial.println
error)
}

// === Fungsi Cek Sensor IR ===
bool isObjectDetected() {
    return digitalRead(IR_SENSOR_PIN) == LOW; // LOW = ada objek
}

void setup() {
    Serial.begin(115200);
    SPI.begin();
    rfid.PCD_Init();

    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    pinMode(IR_SENSOR_PIN, INPUT);

    ledcSetup(PWM_CHANNEL, PWM_FREQ, PWM_RES);
    ledcAttachPin(ENA, PWM_CHANNEL);
    ledcWrite(PWM_CHANNEL, 0);

    Wire.begin(SDA_PIN, SCL_PIN);
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("Sistem Siap");
    lcd.setCursor(0, 1);
    lcd.print("Tempelkan Kartu");
}

```

```

// ✔ Hubungkan WiFi tapi kalau gagal tetap lanjut
WiFi.begin(WIFI_SSID, WIFI_PASS);
unsigned long startAttempt = millis();
while (WiFi.status() != WL_CONNECTED && millis() - startAttempt <
5000) {
    delay(500);
    Serial.print(".");
}
if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWiFi Terhubung!");
} else {
    Serial.println("\nWiFi Gagal, lanjut offline.");
}
}

void loop() {
    // Tombol Manual
    if (digitalRead(BUTTON_PIN) == LOW) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Manual Buka");
        lcd.setCursor(0, 1);
        lcd.print("Pintu Terbuka");
        buzzOnce();
        runMotorSequence();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Tempelkan Kartu");

        sendToGoogle("MANUAL", "Petugas", "Manual Buka");

        delay(1000);
        return;
    }

    // RFID
    if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
        String scannedUID = "";
        for (byte i = 0; i < rfid.uid.size; i++) {
            scannedUID.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : "
"));
            scannedUID.concat(String(rfid.uid.uidByte[i], HEX));
        }
        scannedUID.toUpperCase();
    }
}

```

```

        scannedUID.trim();

        int userIndex = getUserIndex(scannedUID);
        lcd.clear();

        if (userIndex != -1) {
            String name = userNames[userIndex];
            accessCounts[userIndex]++;

            lcd.setCursor(0, 0);
            lcd.print("Selamat Datang");
            lcd.setCursor(0, 1);
            lcd.print(name);

            buzzOnce();
            runMotorSequence();
            sendToGoogle(scannedUID, name, "Hadir");
        } else {
            lcd.setCursor(0, 0);
            lcd.print("Akses Ditolak");
            lcd.setCursor(0, 1);
            lcd.print("Kartu Tidak Sah");
            buzzLong();
            sendToGoogle(scannedUID, "Tidak Dikenal", "Ditolak");
        }

        rfid.PICC_HaltA();
        rfid.PCD_StopCrypto1();
        delay(2000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Tempelkan Kartu");
    }
}

int getUserIndex(String uid) {
    for (int i = 0; i < UID_COUNT; i++) {
        if (uid == authorizedUIDs[i]) return i;
    }
    return -1;
}

void runMotorSequence() {
    lcd.setCursor(0, 1);
    lcd.print("Pintu Terbuka ");
}

```

```

moveForward(motorSpeed);
delay(4000);
stopMotor();
lcd.setCursor(0, 1);
lcd.print("Pintu Diam   ");
delay(5000);

// Tunggu sampai objek hilang sebelum menutup
while (isObjectDetected()) {
    lcd.setCursor(0, 1);
    lcd.print("Objek Terdeteksi");
    delay(200);
}

// Tambah delay 2 detik sebelum pintu menutup
delay(2000);

moveBackward(motorSpeed);
lcd.setCursor(0, 1);
lcd.print("Pintu Tertutup");
delay(4000);
stopMotor();
lcd.setCursor(0, 1);
lcd.print("Selesai   ");
}

void moveForward(int speed) {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    ledcWrite(PWM_CHANNEL, speed);
}

void moveBackward(int speed) {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    ledcWrite(PWM_CHANNEL, speed);
}

void stopMotor() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    ledcWrite(PWM_CHANNEL, 0);
}

void buzzOnce() {

```

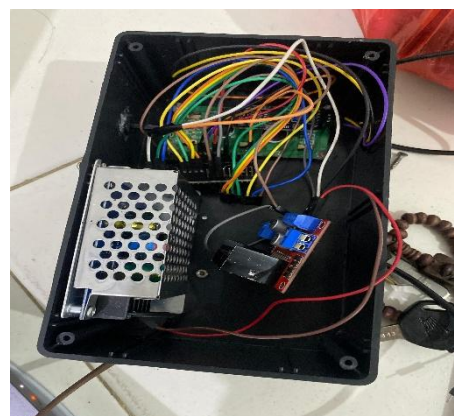
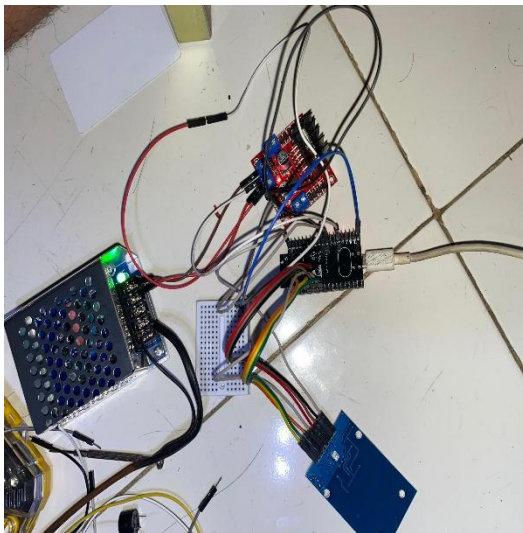
```

digitalWrite(BUZZER_PIN, HIGH);
delay(300);
digitalWrite(BUZZER_PIN, LOW);
}

void buzzLong() {
  digitalWrite(BUZZER_PIN, HIGH);
  delay(1000);
  digitalWrite(BUZZER_PIN, LOW);
}

```

Code Program



Proses Perangkaian Prangkat Keras



Proses Pembuatan Pinion/Gear



Proses Pembuatan Breket Pintu



Proses Pengeboran Pintu dan Dudukan Pintu