

DAFTAR LAMPIRAN

Lampiran 1. Kode Program Sistem Akses Masuk

```
/* Source code Access Masuk sebagai baris kode utama
   Mikrokontroler Masuk (ESP32 DEVKIT V1)
   Smart Access IoT - Akses Masuk
   Author : Bayu Afif Nuranto | 2020230903

*/

// Import Modules & Library
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <SPI.h>
#include <MFRC522.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <NTPClient.h>
#include <Discord_WebHook.h>
#include <ArduinoJson.h>
#include <DFRobotDFPlayerMini.h>

// ===== INISIALISASI VARIABLE
enum AudioClip {
    SELAMAT_DATANG = 1,
    PINTU_TERBUKA = 2,
    PINTU_TERTUTUP = 3,
    TRESSPASSING = 4,
    AKSES_DITOLAK = 5,
    AKSES_DITOLAK_SUDAHDIDALAM = 6
};

/* Inisialisasi User */
struct User {
    int id;
    String name;
};

// ===== DEFINISI PIN
#define magneticPin 25
#define SS_PIN 5
#define RST_PIN 4
#define BUZZER_PIN 12
#define RELAY_PIN 32
#define PIR_PIN 27
#define LED_MERAH 13
#define LED_HIJAU 14

// =====SETUP WIFI & API
=====
```

```

// Konfigurasi WiFi
const char* ssid = "NRTID";
const char* password = "Sabturu1234";

// URL API
const String apiURL = "https://smart-access-
1.bayunuranto.workers.dev/api";

// Setup Discord
String DISCORD_WEBHOOK =
"https://discordapp.com/api/webhooks/1364213154959200306/gDrpCoM
Lvjcg7KyH2Ris5112Z8H-t5P4bhcNUYKdXIyY9d3RFXl7seg8i5DeE4fXy-Xg";
String channel_id = "1364213154959200306";
String token = "gDrpCoMLvjcg7KyH2Ris5112Z8H-
t5P4bhcNUYKdXIyY9d3RFXl7seg8i5DeE4fXy-Xg";
Discord_Webhook discord;

// Layar OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// RFID
MFRC522 rfid(SS_PIN, RST_PIN);
String cardID;

// DF Player Mini
DFRobotDFPlayerMini myDFPlayer;
HardwareSerial FPSerial(1);

// NTP (Network Time Protocol)
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 25200, 60000); //
GMT+7 (WIB)

// ===== VARIABEL GLOBAL
unsigned long lastDisplayTime = 0;
unsigned long lastRFIDTime = 0;
unsigned long pirLastTrigger = 0;
int lastMagneticState = -1;
bool isDisplayOn = false;
bool isFirstBoot = true;
bool lastDoorStateClosed = false;
bool fromRelayAccess = false;

unsigned long lastMagneticChange = 0;
const unsigned long debounceDelay = 200;
const long unauthorizedTimeout = 5000;
const long pirCooldown = 1000;

// Fungsi JSON
String escapeJson(String input) {
  input.replace("\\", "\\");

```

```

    input.replace("\\", "\\");
    input.replace("\n", "\\n");
    return input;
}

String getCardID(byte *buffer, byte bufferSize) {
    String cardID = "";
    for (byte i = 0; i < bufferSize; i++) {
        cardID += String(buffer[i] < 0x10 ? "0" : "");
        cardID += String(buffer[i], HEX);
    }
    cardID.toUpperCase();
    return cardID;
}

void displayMessage(String message, uint8_t line) {
    if (!isDisplayOn) {
        display.clearDisplay();
        isDisplayOn = true;
    }

    display.fillRect(0, line * 10, SCREEN_WIDTH, 10,
SSD1306_BLACK);
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, line * 10);
    display.println(message);
    display.display();
}

void displayMessageCenter(String message, uint8_t line) {
    int16_t x, y;
    uint16_t w, h;
    display.setTextSize(1);
    display.getTextBounds(message, 0, line * 10, &x, &y, &w, &h);
    display.fillRect(0, line * 10, SCREEN_WIDTH, 10,
SSD1306_BLACK);
    int centerX = (SCREEN_WIDTH - w) / 2;
    display.setCursor(centerX, line * 10);
    display.print(message);
    display.display();
}

void activateBuzzer() {
    tone(BUZZER_PIN, 2000);
    delay(1000);
    noTone(BUZZER_PIN);
}

void putarSuara(AudioClip klip) {
    myDFPlayer.play((int)klip);
}

void showDefaultScreen() {
    if (!isDisplayOn) {
        display.clearDisplay();
    }
}

```

```

        isDisplayOn = true;
    }
    display.clearDisplay();
    displayMessage("Sistem Smart Access", 0);
    displayMessageCenter("Silahkan Tap...", 2);
    putarSuara(SELAMAT_DATANG);
}

void manageDisplayTimeout() {
    if (isDisplayOn && (millis() - lastDisplayTime > 3000)) {
        display.clearDisplay();
        display.display();
        isDisplayOn = false;
        digitalWrite(LED_MERAH, LOW);
        digitalWrite(LED_HIJAU, LOW);
    }
}

void readMagneticSensor() {
    int state = digitalRead(magneticPin);
    unsigned long currentTime = millis();
    bool isClosed = (state == LOW);
    if (state != lastMagneticState && (currentTime -
lastMagneticChange) > debounceDelay) {
        lastMagneticChange = currentTime;
        lastMagneticState = state;

        if (!isDisplayOn) {
            display.clearDisplay();
            isDisplayOn = true;
        }

        display.clearDisplay();
        displayMessage("Sistem Smart Access", 0);
        displayMessageCenter("Silahkan Tap...", 2);

        if (state == LOW) {
            displayMessage("Pintu Tertutup", 4);
            if (!isFirstBoot && fromRelayAccess) {
                putarSuara(PINTU_TERTUTUP);
                fromRelayAccess = false;
            }
        } else {
            displayMessage("Pintu Terbuka!", 4);
        }
    }
    isFirstBoot = false;
}

void checkPIR() {
    int pirState = digitalRead(PIR_PIN);
    if (pirState == HIGH && millis() - pirLastTrigger >
pirCooldown) {
        Serial.println("✅ PIR DETEKSI GERAKAN!");
        pirLastTrigger = millis();
        lastDisplayTime = millis();
    }
}

```

```

    if (!isDisplayOn) {
        showDefaultScreen();
    } else {
        display.clearDisplay();
        display.setTextSize(1);
        displayMessage("Gerakan Terdeteksi", 1);
    }
}

void checkTrespassing() {
    int magneticState = digitalRead(magneticPin);
    if (magneticState == HIGH && !fromRelayAccess) {
        Serial.println("🚪 Pintu Terbuka - Cek Trespassing...");
        bool adaOrangKeluar = cekFlagKeluar();

        if (!adaOrangKeluar) {
            Serial.println("❌ TRESSPASSING TERDETEKSI!");
            displayMessage("TRESSPASSING!", 2);
            digitalWrite(LED_MERAH, HIGH);
            activateBuzzer();
            putarSuara(TRESSPASSING);

            String waktu = timeClient.getFormattedTime();
            String pesanTresspassing = "🚨 Tresspassing
Terdeteksi\\nWaktu: " + waktu;
            discord.sendEmbed("Peringatan: Ada Pendobran Pintu!",
pesanTresspassing, "#FF0000");
        } else {
            Serial.println("✅ Pintu dibuka karena akses keluar yang
sah");
        }
    }
}

bool cekStatusKeluar(String cardID) {
    if (WiFi.status() != WL_CONNECTED) {
        Serial.println("❌ WiFi tidak terhubung saat cek status
keluar");
        return false;
    }

    HTTPClient http;
    String url = apiURL + "/access/status/" + cardID;
    Serial.println("Mengakses: " + url);
    http.begin(url);
    int httpResponseCode = http.GET();

    if (httpResponseCode == 200) {
        String response = http.getString();
        Serial.println("Response API: " + response);

        StaticJsonDocument<256> doc;
        DeserializationError error = deserializeJson(doc, response);

```

```

    if (error) {
        Serial.println("✘ Gagal parsing JSON: " +
String(error.f_str()));
        http.end();
        return false;
    }

    if (doc.containsKey("already_inside")) {
        bool alreadyInside = doc["already_inside"];
        http.end();
        Serial.println("Status already_inside: " +
String(alreadyInside ? "true" : "false"));
        return !alreadyInside;
    } else {
        Serial.println("✘ Key 'already_inside' tidak ditemukan");
        http.end();
        return false;
    }
} else {
    Serial.println("✘ HTTP Error: " +
String(httpResponseCode));
    http.end();
    return false;
}
}

void loopWithTimeout() {
if (digitalRead(magneticPin) != LOW) {
    displayMessage("✘ Tutup pintu dahulu!", 3);
    activateBuzzer();
    delay(2000);
    return;
}

if (!rfid.PICC_IsNewCardPresent()) return;
if (!rfid.PICC_ReadCardSerial()) return;

String cardID = getCardID(rfid.uid.uidByte, rfid.uid.size);

display.clearDisplay();
display.setTextSize(1);
display.setCursor(0, 0);
display.println("RFID Detected!");
display.setCursor(0, 10);
display.print("UID: " + cardID);
display.display();

User userData;
bool userFound = getUserbyRFID(cardID, userData);

if (userFound) {
    Serial.println("Nama: " + userData.name);
} else {
    Serial.println("Pengguna tidak ditemukan");
    display.setCursor(0, 20);
}
}

```

```

        display.println("User: Tidak Dikenali");
        display.display();
    }

    if (userFound) {
        bool sudahKeluar = cekStatusKeluar(cardID);
        if (sudahKeluar) {
            aksesDisetujui(userData, cardID);
            sendRFIDToAPI(cardID);
        } else {
            displayMessage("Anda sudah di dalam!", 4);
            String msg = "⚠️ Gagal Masuk\\nRFID: " + cardID +
                "\\nStatus: Anda Belum Keluar";
            discord.sendEmbed("Akses Ditolak", msg, "#FFA500");
            putarSuara(AKSES_DITOLAK_SUDAHDIDALAM);
            activateBuzzer();
        }
    } else {
        aksesDitolak(cardID);
    }
    isDisplayOn = true;
    lastDisplayTime = millis();

    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
}

bool sendRFIDToAPI(String cardID) {
    if (WiFi.status() != WL_CONNECTED) return false;
    unsigned long mulaiWaktu = millis();
    HTTPClient http;
    http.begin(apiURL + "/access/in");
    http.addHeader("Content-Type", "application/json");
    String jsonData = "{\"rfid_tag\":\"" + cardID + "\"}";
    int httpResponseCode = http.POST(jsonData);
    float latensi = ukurLatensi(mulaiWaktu);
    if (httpResponseCode > 0) {
        String response = http.getString();
        Serial.println("Respons API: " + response);
        http.end();
        return response.indexOf("{\"success\":true") != -1;
    } else {
        Serial.println("Error koneksi API");
        http.end();
        return false;
    }
}

bool getUserbyRFID(String cardID, User &user) {
    if (WiFi.status() != WL_CONNECTED) return false;
    unsigned long mulaiWaktu = millis();
    HTTPClient http;
    http.begin(apiURL + "/users/rfid/" + cardID);
    int httpResponseCode = http.GET();
    float latensi = ukurLatensi(mulaiWaktu);
    if (httpResponseCode > 0) {

```

```

String response = http.getString();
const size_t bufferSize = JSON_OBJECT_SIZE(10) + 200;
StaticJsonDocument<bufferSize> doc;
DeserializationError error = deserializeJson(doc, response);
if (error) {
    Serial.print("Gagal parsing JSON: ");
    Serial.println(error.c_str());
    return false;
}
if (!doc.containsKey("name") || doc["name"].as<String>() ==
"" ) {
    return false;
}
if (doc.containsKey("id")) {
    user.id = doc["id"];
} else {
    user.id = 0;
}
user.name = doc["name"].as<String>();
return true;
}

http.end();
return false;
}

void aksesDisetujui(User userData, String cardID) {
    timeClient.update();
    String currentTime = timeClient.getFormattedTime();
    display.clearDisplay();
    displayMessageCenter("AKSES", 1);
    displayMessageCenter("DITERIMA", 2);
    displayMessageCenter("Nama: " + userData.name, 3);
    displayMessageCenter("Waktu: " + currentTime, 4);
    displayMessageCenter("Silahkan Masuk", 5);
    display.display();
    putarSuara(PINTU_TERBUKA);

    String messageSuccess = "Akses Diterima\\nAkses Masuk\\nNama:
" + escapeJson(userData.name) +
        "\\nRFID Tag: " + cardID + "\\nWaktu:
" + currentTime;
    unsigned long start = millis();
    bool status = discord.sendEmbed("Akses Diterima",
messageSuccess, "#008000");
    unsigned long end = millis();
    float latency = (end - start) / 1000.0;
    Serial.print("🕒 Latensi pengiriman Discord (Akses Diterima):
");
    Serial.print(latency, 3);
    Serial.println(" detik");
    if (!status) {
        Serial.println("❌ Gagal mengirim notifikasi Discord!");
    }
    digitalWrite(LED_HIJAU, HIGH);
    digitalWrite(RELAY_PIN, HIGH);
}

```

```

fromRelayAccess = true;
unsigned long startWait = millis();
while (digitalRead(magneticPin) != LOW && millis() - startWait
< 10000) {
    delay(100);
}

if (digitalRead(magneticPin) == LOW) {
    putarSuara(PINTU_TERTUTUP);
    Serial.println("✅ Pintu tertutup, relay dimatikan.");
} else {
    Serial.println("⚠️ Timeout: Pintu belum tertutup setelah 10
detik.");
}
putarSuara(PINTU_TERTUTUP);
digitalWrite(RELAY_PIN, LOW);
digitalWrite(LED_HIJAU, LOW);
}

void aksesDitolak(String cardID) {
    timeClient.update();
    String currentTime = timeClient.getFormattedTime();
    display.clearDisplay();
    displayMessage("AKSES", 1);
    displayMessage("DITOLAK!", 2);
    displayMessage("RFID Invalid", 3);
    displayMessage("Waktu: " + currentTime, 5);
    display.display();
    activateBuzzer();
    digitalWrite(LED_MERAH, HIGH);
    putarSuara(AKSES_DITOLAK);
    String errorMessage = "Akses Ditolak\\nRFID Tag: " + cardID +
"\\nWaktu: " + currentTime + "\\nStatus: Kartu tidak
terdaftar";
    unsigned long start = millis();
    bool status = discord.sendEmbed("Akses Ditolak", errorMessage,
"#FF0000");
    unsigned long end = millis();
    float latency = (end - start) / 1000.0;
    Serial.print("🕒 Latensi pengiriman Discord (Akses Ditolak):
");
    Serial.print(latency, 3);
    Serial.println(" detik");
    if (!status) {
        Serial.println("❌ Gagal mengirim notifikasi Discord!");
    }
    delay(2000);
    digitalWrite(LED_MERAH, LOW);
}

float ukurLatensi(unsigned long waktuMulai) {
    unsigned long waktuSelesai = millis();
    float latensiDetik = (waktuSelesai - waktuMulai) / 1000.0;
    Serial.print("🕒 Latensi: ");
    Serial.print(latensiDetik, 3);
}

```

```

    Serial.println(" detik");
    return latensiDetik;
}

bool cekFlagKeluar() {
    if (WiFi.status() != WL_CONNECTED) return false;
    HTTPClient http;
    http.begin(apiURL + "/flag/keluar");
    int httpCode = http.GET();
    if (httpCode == 200) {
        String response = http.getString();
        StaticJsonDocument<128> doc;
        DeserializationError err = deserializeJson(doc, response);
        if (!err && doc["status"] == true) {
            Serial.println("✅ Dikonfirmasi: Ada yang keluar (flag aktif)");
            http.end();
            return true;
        }
    }
    Serial.println("⚠️ Tidak ada flag keluar aktif");
    http.end();
    return false;
}

void setup() {
    Serial.begin(9600);
    WiFi.begin(ssid, password);
    Serial.print("Menghubungkan ke WiFi...");
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.print(".");
    }
    Serial.println("\nWiFi Terhubung!");
    discord.begin(channel_id, token);
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println("OLED allocation failed");
        for (;;)
    }

    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.println("Sistem Smart Access");
    display.setCursor(0, 20);
    display.println("Loading...");
    display.display();
    delay(1500);
    SPI.begin();
    rfid.PCD_Init();
    byte version = rfid.PCD_ReadRegister(MFRC522::VersionReg);
    if (version == 0x91 || version == 0x92) {
        Serial.println("MFRC522 detected and initialized.");
    }
}

```

```

pinMode(BUZZER_PIN, OUTPUT);
noTone(BUZZER_PIN);
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, LOW);
pinMode(PIR_PIN, INPUT);
pinMode(LED_MERAH, OUTPUT);
pinMode(LED_HIJAU, OUTPUT);
digitalWrite(LED_MERAH, LOW);
digitalWrite(LED_HIJAU, LOW);
pinMode(magneticPin, INPUT);

FPSerial.begin(9600, SERIAL_8N1, 16, 17);
Serial.println("Inisialisasi DFPlayer Mini...");
if (!myDFPlayer.begin(FPSerial)) {
    Serial.println("DFPlayer Mini Gagal di inisialisasi");
    while (true);
}
myDFPlayer.volume(20);
putarSuara(SELAMAT_DATANG);
delay(1000);
timeClient.begin();
timeClient.update();
digitalWrite(LED_HIJAU, HIGH);
tone(BUZZER_PIN, 1000);
delay(500);
noTone(BUZZER_PIN);
digitalWrite(LED_HIJAU, LOW);
Serial.println("Setup selesai!");
}

void loop() {
    checkPIR();
    readMagneticSensor();
    checkTrespassing();
    loopWithTimeout();
    manageDisplayTimeout();
}

```

Lampiran 2. Kode Program Sistem Akses Keluar

```
/* Source code Access Keluar
   Mikrokontroler Keluar (NodeMCU ESP8266)
   Smart Access IoT - Akses Keluar
   Author : Bayu Afif Nuranto | 2020230903
*/

// Import Library
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <Discord_WebHook.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecureBearSSL.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

// Inisialisasi PIN
#define SS_PIN 15 // D8 = GPIO15
#define RST_PIN 16 // D0 = GPIO16
#define BUZZER_PIN 5 // D1 = GPIO05
#define RELAY_PIN 4 // D2 = GPIO04 (Relay)
#define LED_MERAH 0 // D3 = GPIO0
#define LED_HIJAU 2 // D4 = GPIO2

// Konfigurasi WiFi, API URL & Discord Setup
const char* ssid = "NRTID";
const char* password = "Sabturu1234";
const String apiURL = "https://smart-access-1.bayunuranto.workers.dev/api";
String DISCORD_WEBHOOK =
"https://discordapp.com/api/webhooks/1375156882255122583/ZFZDha8bXUs-qJO9y6hDKgvoNmGbIo7ZIS-ISYOWsixlzXfVfHCgLxiufzghcZmzHo_I";
String channel_id = "1375156882255122583";
String token = "ZFZDha8bXUs-qJO9y6hDKgvoNmGbIo7ZIS-ISYOWsixlzXfVfHCgLxiufzghcZmzHo_I";
Discord_Webhook discord;
MFRC522 rfid(SS_PIN, RST_PIN);

// Akses Kontrol Solenoid
unsigned long solenoidUnlockTime = 0;
bool solenoidActive = false;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 25200, 60000);

// Fungsi JSON
String escapeJson(String input) {
  input.replace("\\", "\\");
  input.replace("\"", "\\");
  input.replace("\n", "\\n");
  return input;
}
```

```

String getCardID(byte *buffer, byte bufferSize) {
    String cardID = "";
    for (byte i = 0; i < bufferSize; i++) {
        cardID += String(buffer[i] < 0x10 ? "0" : "");
        cardID += String(buffer[i], HEX);
    }
    cardID.toUpperCase();
    return cardID;
}

// Feedback Buzzer
void buzz(int times, int duration) {
    for (int i = 0; i < times; i++) {
        digitalWrite(BUZZER_PIN, HIGH);
        delay(duration);
        digitalWrite(BUZZER_PIN, LOW);
        delay(duration);
    }
}

void buzzSuccess() {
    digitalWrite(LED_HIJAU, HIGH);
    digitalWrite(LED_MERAH, LOW);
    buzz(2, 200);
    delay(2000);
    digitalWrite(LED_HIJAU, LOW);
}

void buzzError() {
    digitalWrite(LED_MERAH, HIGH);
    digitalWrite(LED_HIJAU, LOW);
    buzz(1, 800);
    delay(200);
    buzz(2, 200);
    delay(2000);
    digitalWrite(LED_MERAH, LOW);
}

// ===== Relay Akses =====
void fromRelayAccess() {
    digitalWrite(RELAY_PIN, HIGH); // Aktifkan solenoid
    digitalWrite(LED_HIJAU, HIGH);
    solenoidUnlockTime = millis();
    solenoidActive = true;
}

void sendPostRequest(String cardID) {
    timeClient.update();
    String currentTime = timeClient.getFormattedTime();
    if (WiFi.status() == WL_CONNECTED) {
        std::unique_ptr<BearSSL::WiFiClientSecure> client(new
BearSSL::WiFiClientSecure);
        client->setInsecure();
        HTTPClient https;
        String payload = "{\"rfid_tag\":\"" + cardID + "\"}";
        Serial.println("Sending POST request...");
    }
}

```

```

Serial.println("Payload: " + payload);
https.begin(*client, apiURL + "/access/out");
https.addHeader("Content-Type", "application/json");
int httpResponseCode = https.POST(payload);

if (httpResponseCode == 200) {
    String response = https.getString();
    Serial.println("Response: " + response);

    if (response.indexOf("\"success\":true") >= 0) {
        // Kirim notifikasi Discord
        String messageSuccess = "***Akses Diterima**\n**Akses
Keluar RFID Tag:** **" + cardID + "**\n**Waktu:** **" +
currentTime + "***";
        discord.sendEmbed("Akses Diterima", messageSuccess,
"#008000");
        buzzSuccess();
        fromRelayAccess();
        HTTPClient flagRequest;
        flagRequest.begin(*client, apiURL + "/access/out/flag");
        flagRequest.addHeader("Content-Type",
"application/json");
        int flagResponse = flagRequest.POST(payload);
        Serial.println("Flag keluar response: " +
String(flagResponse));
        flagRequest.end();
    } else {
        String messageError = "***Akses RFID Tag: ** **" + cardID
+ "*** **Ditolak**\n**Waktu:** **" + currentTime + "***";
        discord.sendEmbed("Akses Ditolak", messageError,
"#FF0000");
        buzzError();
    }
    } else {
        Serial.println("HTTP Request Failed, code: " +
String(httpResponseCode));
        buzzError();
    }
    https.end();
} else {
    Serial.println("WiFi not connected!");
    buzzError();
}
}

void setup() {
    Serial.begin(9600);
    SPI.begin();
    rfid.PCD_Init();
    rfid.PCD_SetAntennaGain(rfid.RxGain_max);
    discord.begin(channel_id, token);
    discord.addWiFi(ssid, password);
    discord.connectWiFi();
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(LED_MERAH, OUTPUT);
}

```

```

pinMode(LED_HIJAU, OUTPUT);
digitalWrite(RELAY_PIN, LOW);
digitalWrite(BUZZER_PIN, LOW);
digitalWrite(LED_MERAH, LOW);
digitalWrite(LED_HIJAU, LOW);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
}
Serial.println("Connected to WiFi!");
Serial.println("RFID Access Control System (Keluar)");
Serial.println("Silahkan Scan Kartu RFID Anda ...");
timeClient.begin();
timeClient.update();
}

void loop() {
    if (rfid.PICC_IsNewCardPresent() &&
rfid.PICC_ReadCardSerial()) {
        String cardID = getCardID(rfid.uid.uidByte, rfid.uid.size);
        Serial.println("ID Card Anda : " + cardID);
        sendPostRequest(cardID);
        rfid.PICC_HaltA();
        rfid.PCD_StopCryptol();
    }

    if (solenoidActive && millis() - solenoidUnlockTime >= 5000) {
        digitalWrite(RELAY_PIN, LOW);
        digitalWrite(LED_HIJAU, LOW);
        solenoidActive = false;
    }
}

```

Lampiran 3. Kode Program Sistem Aplikasi

```

// === Import & Inisialisasi ===
import { Client } from '@neondatabase/serverless'
import { Hono } from 'hono'
import { dashboardPage, managementUser, loginPageHTML } from
'./pages'

const app = new Hono()

// === Konfigurasi Cookie & Database ===
function getCookie(c, n) { return
c?.match(`(^|;)\s*${n}\s*=\s*([^\s;]+)`)?.[2] }
function setAuthCookie(t) { return `auth_token=${t}; path=/;
HttpOnly; Secure; SameSite=Strict` }
function clearAuthCookie() { return `auth_token=; path=/;
expires=Thu, 01 Jan 1970; HttpOnly; Secure; SameSite=Strict` }

app.use(async (c, next) => {

```

```

    c.env.db = new Client({ connectionString:
'postgresql://...' })
    await c.env.db.connect()
    c.authToken = getCookie(c.req.header('Cookie'), 'auth_token')
    await next()
  })

// === Route Login & Logout ===
app.post('/api/login', async (c) => {
  const { email, password } = await c.req.json()
  const result = await c.env.db.query(
    'SELECT * FROM users WHERE email=$1 AND password=$2',
    [email, password]
  )
  const user = result.rows[0]
  if (user?.role === 'Admin')
    return c.json({ success: true }, 200, { 'Set-Cookie':
setAuthCookie('admin-token') })
  return c.json({ success: false }, 403)
})

app.post('/api/logout', (c) =>
  c.json({ success: true }, 200, { 'Set-Cookie':
clearAuthCookie() })
)

// === Route Data Utama ===
app.get('/api/users', async (c) => {
  if (c.authToken !== 'admin-token') return c.json({ error:
'Unauthorized' }, 403)
  const res = await c.env.db.query('SELECT * FROM users')
  return c.json(res.rows)
})

app.post('/api/users', async (c) => {
  const { name, email, password, rfid_tag, role } = await
c.req.json()
  const t = new Date().toISOString()
  await c.env.db.query(
    'INSERT INTO users
(name,email,password,rfid_tag,role,created_at) VALUES
($1,$2,$3,$4,$5,$6)',
    [name, email, password, rfid_tag, role, t]
  )
  return c.json({ success: true })
})

// === Route Akses RFID ===
app.post('/api/access/in', async (c) => {
  const { rfid_tag } = await c.req.json()
  const db = c.env.db, time = new Date().toISOString()
  const user = (await db.query('SELECT * FROM users WHERE
rfid_tag=$1', [rfid_tag])).rows[0]
  if (!user) return c.json({ success: false, error: 'RFID tidak
terdaftar' })
}

```

```

    await db.query('INSERT INTO logs
(rfid_tag,name,status,timestamp_in) VALUES ($1,$2,$3,$4)',
    [rfid_tag, user.name, 'Disetujui', time])
    return c.json({ success: true, message: `Akses masuk
${user.name}` })
})

app.post('/api/access/out', async (c) => {
    const { rfid_tag } = await c.req.json()
    const db = c.env.db, time = new Date().toISOString()
    const log = (await db.query(
    'SELECT log_id FROM logs WHERE rfid_tag=$1 AND timestamp_out
IS NULL ORDER BY timestamp_in DESC LIMIT 1',
    [rfid_tag])).rows[0]
    if (!log) return c.json({ success: false, error: 'Belum ada
log aktif' })
    await db.query('UPDATE logs SET timestamp_out=$1 WHERE
log_id=$2', [time, log.log_id])
    return c.json({ success: true, message: 'Akses keluar
berhasil' })
})

// === Route Halaman ===
app.get('/', (c) => c.authToken === 'admin-token'
? c.html(dashboardPage())
: c.html(loginPageHTML())
)

export default app

```

Lampiran 4. Surat Keterangan Bebas Plagiat



UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN
Gedung Rektorat Lantai 3,
Jl. Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

SURAT KETERANGAN HASIL PENGECEKAN TURNITIN

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : RANCANG BANGUN SISTEM SMART ACCESS BERBASIS IOT
UNTUK KEAMANAN DAN MONITORING RUANGAN (Studi
Kasus: Ruang Lab SMK Dinamika Pembangunan 1 Jakarta)

Penulis : Bayu Afif Nuranto

NIM : 2020230903

Tgl pemeriksaan : 25 Juli 2025

Dengan hasil Tingkat Kesamaan (*similarity index*) **9%**

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 25 Juli 2025

Ka.UPT Perpustakaan Unsada

Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan
dan Pasca Sarjana

Lampiran 5. Hasil Turnitin

2020230903_Bayu Afif Nuranto			
ORIGINALITY REPORT			
9%	9%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	docplayer.info Internet Source		1%
2	elib.pnc.ac.id Internet Source		1%
3	journal.literasisains.id Internet Source		1%
4	repository.dinamika.ac.id Internet Source		<1%
5	dspace.uui.ac.id Internet Source		<1%
6	journal.binadarma.ac.id Internet Source		<1%
7	text-id.123dok.com Internet Source		<1%
8	123dok.com Internet Source		<1%
9	jie.pnp.ac.id Internet Source		<1%
10	repo.darmajaya.ac.id Internet Source		<1%
11	ejurnal.unisan.ac.id Internet Source		<1%
12	eprints.pktj.ac.id Internet Source		<1%

13 repository.binadarma.ac.id <1%
Internet Source

14 repository.widyamandala.ac.id <1%
Internet Source

15 widuri.raharja.info <1%
Internet Source

16 Agus Prayitno, Chusnul Chotimah, Syaiful Nugraha, Teddy Istanto. "Sistem Keamanan Pintu Laboratorium Perikanan Berbasis Mikrokontroler", Jurnal Riset Rekayasa Elektro, 2024 <1%
Publication

17 journal.iaincurup.ac.id <1%
Internet Source

18 repository.its.ac.id <1%
Internet Source

19 journal-computing.org <1%
Internet Source

20 journal.unj.ac.id <1%
Internet Source

21 repository.uinjkt.ac.id <1%
Internet Source

22 sefhiabudiarti.blogspot.com <1%
Internet Source

23 core.ac.uk <1%
Internet Source

24 etd.umy.ac.id <1%
Internet Source

25 journal.aptii.or.id <1%
Internet Source

26 Dinda Yassa Descania. "PENERAPAN METODE PROTOTYPE PADA PENGEMBANGAN SISTEM ANTRIAN ONLINE DI KEMENTERIAN ATR/BPN KAB. SUKABUMI", INDEXIA, 2023

Publication

<1%

27 amirudin254.blogspot.com

Internet Source

<1%

28 ejournal3.undip.ac.id

Internet Source

<1%

29 repository.upi.edu

Internet Source

<1%

30 Ahlis Noor Kholili, Ahmad Mubais. "Mengeksplorasi Cerita Rakyat Indonesia melalui Aplikasi Mobile sebagai Paradigma baru Pendidikan Budaya", Jurnal Informatika Komputer, Bisnis dan Manajemen, 2025

Publication

<1%

31 Oktovianus MOU, Helena Saway, Elvis Pawan, Budi Soepriyanto. "PERANCANGAN SISTEM INFORMASI PENJUALAN BAHAN BANGUNAN BERBASIS WEBSITE PADA TOKO ALFI JAYA", Jurnal Ilmiah Sistem Informasi (JISI), 2023

Publication

<1%

32 Risna Risna, Harrizki Arie Pradana. "Rancang Bangun Aplikasi Monitoring Penggunaan Air PDAM Berbasis Mikrokontroler Arduino Uno", Jurnal Sisfokom (Sistem Informasi dan Komputer), 2014

Publication

<1%

33 hendrigemuk.blogspot.com

Internet Source

<1%

34 library.binus.ac.id

Internet Source

<1%

35	repository.polman-babel.ac.id Internet Source	<1%
36	e-journal.uajy.ac.id Internet Source	<1%
37	eprints.ums.ac.id Internet Source	<1%
38	es.scribd.com Internet Source	<1%
39	zh.scribd.com Internet Source	<1%
40	e-jurnal.lppmunsera.org Internet Source	<1%
41	eprints.umk.ac.id Internet Source	<1%
42	journal.ipm2kpe.or.id Internet Source	<1%
43	repository.unair.ac.id Internet Source	<1%
44	cliquersnet.blogspot.com Internet Source	<1%
45	ejournal.itn.ac.id Internet Source	<1%
46	elektro.unsyiah.ac.id Internet Source	<1%
47	eprints.unipdu.ac.id Internet Source	<1%
48	www.coursehero.com Internet Source	<1%
49	www.unisbank.ac.id Internet Source	<1%

50	amaziabernikeatmadja.blogspot.com Internet Source	<1%
51	digilib.unila.ac.id Internet Source	<1%
52	docobook.com Internet Source	<1%
53	eprints.unram.ac.id Internet Source	<1%
54	files1.simpkb.id Internet Source	<1%
55	ind.small-business-tracker.com Internet Source	<1%
56	jurnal.api.ac.id Internet Source	<1%
57	jurnal.poliupg.ac.id Internet Source	<1%
58	kumpulantesis1.blogspot.com Internet Source	<1%
59	projects.co.id Internet Source	<1%
60	repository.radenfatah.ac.id Internet Source	<1%
61	repository.unimor.ac.id Internet Source	<1%
62	telkomuniversity.ac.id Internet Source	<1%
63	Klancar, G.. "Tracking-error model-based predictive control for mobile robots in real time", <i>Robotics and Autonomous Systems</i> , 20070630 Publication	<1%

64	Rizki Wicaksono, Ryan Rakryan, Novianti Madhona Faizah. "Aplikasi Pengaduan Sarana dan Prasarana: Studi Kasus di SMK Bhayangkari Delog Berbasis Web dengan Metode Rapid Application Development Menggunakan Sublime Text dan MySQL", Jurnal Ilmu Komputer dan Teknologi Informasi, 2025 Publication	<1%
65	artikelpendidikan.id Internet Source	<1%
66	ejournal.unis.ac.id Internet Source	<1%
67	eprints.upnyk.ac.id Internet Source	<1%
68	garuda.kemdikbud.go.id Internet Source	<1%
69	idoc.pub Internet Source	<1%
70	kulon.undip.ac.id Internet Source	<1%
71	pt.scribd.com Internet Source	<1%
72	repository.ub.ac.id Internet Source	<1%
73	repository.uinsu.ac.id Internet Source	<1%
74	repository.unj.ac.id Internet Source	<1%
75	repository.upnjatim.ac.id Internet Source	<1%

76 scholar.unand.ac.id <1%
Internet Source

77 www.dicoding.com <1%
Internet Source

78 Mohammad Hanif Yuhdi, Anggi Indah Yuliana, <1%
Sujono Sujono. "Rancang Bangun Alat Deteksi
dan Pengendali Hama Burung Pipit Pada
Pertanaman Padi Sawah Berbasis Arduino
WeMos ESP8266", Exact Papers in
Compilation (EPiC), 2023
Publication

79 Sawitri Nurhayati, Teuku Fadjar Shadek, Darpi <1%
Darpi, Dodi Lasandi. "PERANCANGAN SISTEM
PEREKRUTAN SECURITY BADAN USAHA JASA
PENGAMANAN BERBASIS WEB PADA PT.
KRAKATAU JASA INDUSTRI", Journal of
Innovation And Future Technology (IFTECH),
2025
Publication

Exclude quotes Off Exclude matches Off
Exclude bibliography Off