

LAMPIRAN

Coding arduino Absensi

```
#include <Arduino.h> // library ke database
#include <WiFi.h> // library esp32
#include <HTTPClient.h> // library ke database
#include <SPI.h>
#include <MFRC522.h>

const char* ssid = "ARB";
const char* password = "03102002";

#define SS_PIN 5
#define RST_PIN 15

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

int buzzer = 27;
int tambah = 26;
int masuk = 25;
int keluar = 33;
bool data_tambah = 0;
bool data_masuk = 0;
bool data_keluar = 0;
#define ditekan 0
int val = 0;

MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class

String id;
void setup() {
  Serial.begin(115200);
  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522
  lcd.begin();
  lcd.backlight();
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  lcd.setCursor(1, 0);
  lcd.print("Sistem Absensi");
  while (WiFi.status() != WL_CONNECTED) {
```

```

    delay(100);
    Serial.println("Connecting to WiFi..");
    lcd.setCursor(0, 1);
    lcd.print("Inisialisasi..");
}
Serial.println(WiFi.localIP());
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("Sistem Absensi");
lcd.setCursor(0, 1);
lcd.print("WiFi Connected");
pinMode(buzzer, OUTPUT);
pinMode(tambah, INPUT_PULLUP);
pinMode(masuk, INPUT_PULLUP);
pinMode(keluar, INPUT_PULLUP);
}

void loop() {
    data_tambah = digitalRead(tambah);
    data_masuk = digitalRead(masuk);
    data_keluar = digitalRead(keluar);
    if (data_tambah == ditekan) {
        Serial.println("add");
        val = 1;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Mode Tambah Data");
        lcd.setCursor(0, 1);
        lcd.print("Tempelkan Kartu");
    }

    if (data_masuk == ditekan){
        Serial.println("in");
        val = 2;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Mode Masuk");
        lcd.setCursor(0, 1);
        lcd.print("Tempelkan Kartu");
    }

    if (data_keluar == ditekan) {
        Serial.println("out");
    }
}

```

```

    val = 3;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Mode Keluar");
    lcd.setCursor(0, 1);
    lcd.print("Tempelkan Kartu");
}

switch (val) {
  case 1:
    if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
      digitalWrite(buzzer, HIGH);
      delay(200);
      digitalWrite(buzzer, LOW);
      Serial.println();
      Serial.print(F("In dec: "));
      for (byte i = 0; i < rfid.uid.size; i++) {
        id=id+(rfid.uid.uidByte[i]<0x10 ? "0" : "") +
String(rfid.uid.uidByte[i],HEX) + (i==3 ? "" : ":");
      }
      id.toUpperCase();
      rfid.PICC_HaltA();
      rfid.PCD_StopCrypto1();
      Serial.print(id);
      send_data();
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Mode Tambah Data");
      lcd.setCursor(0, 1);
      lcd.print(id);
      id = "";
      delay(1000);
      lcd.setCursor(0, 0);
      lcd.print("Mode Tambah Data");
      lcd.setCursor(0, 1);
      lcd.print("Tempelkan Kartu ");
    }
    break;
  case 2:
    if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
      digitalWrite(buzzer, HIGH);
      delay(200);
      digitalWrite(buzzer, LOW);

```

```

    Serial.println();
    Serial.print(F("In dec: "));
    for (byte i = 0; i < rfid.uid.size; i++) {
        id=id+(rfid.uid.uidByte[i]<0x10 ? "0" : "") +
String(rfid.uid.uidByte[i],HEX) + (i==3 ? "" : ":");
    }
    id.toUpperCase();
    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
    Serial.print(id);
    send_datain();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Mode Masuk");
    lcd.setCursor(0, 1);
    lcd.print("Absen Berhasil");
    Serial.print(id);
    id = "";
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("Mode Masuk");
    lcd.setCursor(0, 1);
    lcd.print("Tempelkan Kartu ");
}
break;
case 3:
    if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
        digitalWrite(buzzer, HIGH);
        delay(200);
        digitalWrite(buzzer, LOW);
        Serial.println();
        Serial.print(F("In dec: "));
        for (byte i = 0; i < rfid.uid.size; i++) {
            id=id+(rfid.uid.uidByte[i]<0x10 ? "0" : "") +
String(rfid.uid.uidByte[i],HEX) + (i==3 ? "" : ":");
        }
        id.toUpperCase();
        rfid.PICC_HaltA();
        rfid.PCD_StopCrypto1();
        Serial.print(id);
        send_dataout();
        lcd.clear();
        lcd.setCursor(0, 0);

```

```

        lcd.print("Mode Keluar");
        lcd.setCursor(0, 1);
        lcd.print("Absen Berhasil");
        Serial.print(id);
        id = "";
        delay(1000);
        lcd.setCursor(0, 0);
        lcd.print("Mode Keluar");
        lcd.setCursor(0, 1);
        lcd.print("Tempelkan Kartu ");
    }
    break;
}
}

void send_data() {
    String datareq = "http://ctech-store.my.id/data_absensi/tambah.php?";
    datareq += "tag=";
    datareq += id;
    Serial.println(datareq);
    HTTPClient http;
    http.begin(datareq);
    int httpCode = http.GET();
    if (httpCode > 0) {
        Serial.printf("[HTTP] GET... code: %d\n", httpCode);
        if (httpCode == HTTP_CODE_OK) {
            String payload = http.getString();
            Serial.println(payload);

        } else {
            Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
        }
        http.end();
    }
}

void send_datain() {
    String datareq = "http://ctech-store.my.id/data_absensi/absen.php?";
    datareq += "tag=";
    datareq += id;
    Serial.println(datareq);
    HTTPClient http;

```

```

http.begin(datareq);
int httpCode = http.GET();
if (httpCode > 0) {
    Serial.printf("[HTTP] GET... code: %d\n", httpCode);
    if (httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        Serial.println(payload);
    } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
    }
    http.end();
}
}

void send_dataout() {
String datareq = "http://ctech-store.my.id/data_absensi/keluar.php?";
datareq += "tag=";
datareq += id;
Serial.println(datareq);
HTTPClient http;
http.begin(datareq);
int httpCode = http.GET();
if (httpCode > 0) {
    Serial.printf("[HTTP] GET... code: %d\n", httpCode);
    if (httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        Serial.println(payload);
    } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
    }
    http.end();
}
}
}

```