LAMPIRAN

Scrip ESP32

```
// ESP32 code using CTBot for Telegram Bot integration with Arduino UNO via UART
#include <CTBot.h>
#include <WiFi.h>

CTBot myBot;

// WiFi credentials
const char* ssid = "Linda";
const char* password = "lindahsb";

// Telegram Bot Token
String token = "7877644001:AAGNTZ2gROuKXnjkKK4Nc1o2r6iub21U6e4";

// Telegram Chat IDs yang diizinkan
int64_t allowedChatIds[] = {5383352245, 1723508423};

// UART connection ke Arduino UNO
#define RXD2 16
#define TXD2 17

void setup() {
  Serial.begin(115200);
  Serial2.begin(115200, SERIAL_8N1, RXD2, TXD2);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected!");

  myBot.wifiConnect(ssid, password);
```

```
  myBot.setTelegramToken(token);

  if (myBot.testConnection()) {
    Serial.println("Bot terhubung ke Telegram");
    for (int i = 0; i < sizeof(allowedChatIds) / sizeof(allowedChatIds[0]); i++) {
      myBot.sendMessage(allowedChatIds[i], "🤖 Bot siap! Ketik /start atau /buka untuk membuka pintu.");
    }
  } else {
    Serial.println("Gagal terhubung ke Telegram");
  }
}

void loop() {
  // Terima data dari Arduino UNO
  if (Serial2.available()) {
    String pesan = Serial2.readStringUntil('\n');
    pesan.trim();
    Serial.println("Dari Arduino: " + pesan);

    for (int i = 0; i < sizeof(allowedChatIds) / sizeof(allowedChatIds[0]); i++) {
      myBot.sendMessage(allowedChatIds[i], pesan);
    }
  }

  // Cek pesan dari Telegram
  TBMessage msg;
  if (myBot.getNewMessage(msg)) {
    int64_t chatId = msg.sender.id;
    String text = msg.text;
    bool allowed = false;
    for (int i = 0; i < sizeof(allowedChatIds) / sizeof(allowedChatIds[0]); i++) {
      if (chatId == allowedChatIds[i]) {
        allowed = true;
        break;
```

```
    }
   }

   if (allowed) {
    if (text.equalsIgnoreCase("/buka")) {
     Serial2.println("BUKA");
     myBot.sendMessage(chatId, "🔓 Pintu dibuka!");
    } else if (text.equalsIgnoreCase("/start")) {
     myBot.sendMessage(chatId,
      "👋 Selamat datang!\n\nKetik:\n`/buka` untuk membuka pintu.\n\nSiap membantu kamu!",
      "Markdown");
    }
   } else {
    myBot.sendMessage(chatId, "❌ Maaf, kamu tidak memiliki izin mengakses bot ini.");
   }
  }

  delay(500);
}
```

Scrip Arduino Uno

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Keypad.h>

// Definisi pin
#define RST_PIN 9
#define SS_PIN 10
#define RELAY_PIN 7
#define BUZZER_PIN 8  // Buzzer
MFRC522 rfid(SS_PIN, RST_PIN);
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Keypad
const byte ROWS = 4, COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {5, 4, 3, 2};
byte colPins[COLS] = {A3, A2, A1, A0};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// PIN
String correctPIN = "1234";
String enteredPIN = "";

void setup() {
  Serial.begin(115200);
  SPI.begin();
  rfid.PCD_Init();  // Init RFID
```

```
  lcd.init();
  lcd.backlight();

  pinMode(RELAY_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);

  lcd.setCursor(0, 0);
  lcd.print("Scan kartu atau");
  lcd.setCursor(0, 1);
  lcd.print("Masukkan PIN");
}

void loop() {
  // Cek kartu RFID
  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    String uid = "";
    for (byte i = 0; i < rfid.uid.size; i++) {
      uid += String(rfid.uid.uidByte[i], HEX);
    }
    rfid.PICC_HaltA();          // Stop komunikasi
    rfid.PCD_StopCrypto1();     // Stop enkripsi

    Serial.println("UID Kartu: " + uid);

    if (uid == "d3af2e2") {  // Ganti sesuai UID kartu
      unlockDoor("Akses diterima");
    } else {
      rejectAccess("Akses ditolak!");
    }
  }

  // Cek input dari keypad
  char key = keypad.getKey();
  if (key) {
```

```
    if (key == '#') {
      if (enteredPIN == correctPIN) {
        unlockDoor("PIN: Akses diterima");
      } else {
        rejectAccess("PIN: Akses ditolak!");
      }
      enteredPIN = "";
    } else {
      enteredPIN += key;
      lcd.setCursor(0, 1);
      lcd.print(enteredPIN);
    }
  }

  // Cek perintah dari ESP32 via UART
  if (Serial.available()) {
    String command = Serial.readStringUntil('\n');
    command.trim();
    Serial.println("Perintah dari ESP32: " + command);

    if (command == "BUKA") {
      unlockDoor("Dibuka via Telegram");
    }
  }
}

// Fungsi untuk membuka kunci
void unlockDoor(String message) {
  digitalWrite(RELAY_PIN, HIGH);
  tone(BUZZER_PIN, 1000, 200);  // Bunyi pendek
  sendToESP32(message);

  delay(5000);  // Tunggu 5 detik
  digitalWrite(RELAY_PIN, LOW);
```

```
  rfid.PCD_Init();  // Reset ulang RFID biar bisa baca lagi
}

// Fungsi jika akses ditolak
void rejectAccess(String message) {
  sendToESP32(message);
  tone(BUZZER_PIN, 1000, 1000);  // Bunyi panjang
  delay(1000);
  noTone(BUZZER_PIN);

  rfid.PCD_Init();  // Reset RFID juga kalau gagal
}

// Fungsi kirim ke ESP32 dan update LCD
void sendToESP32(String msg) {
  Serial.println(msg);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(msg);
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scan kartu atau");
  lcd.setCursor(0, 1);
  lcd.print("Masukkan PIN");
}
```

LEMBAR REVISI

No	Uraian	Revisi
1	Perbaiki metode penelitia	✓
2	Perbaiki tabel	✓
3	Perbaiki perbaiki tulisan	✓

Rantauprapat, 09 Juli 2025

Penguji

Dr.Iwan Purnama, S.Kom.,M.Kom