LAMPIRAN

Program NodeMCU ESP32

```
#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);

// Inisialisasi LCD I2C
LiquidCrystal_I2C lcd(0x27, 16, 2);

const int buzzer = 13;
const int espcam = 27;
const int selenoid = 26;

const unsigned long botRequestDelay = 3000;
unsigned long lastTimeBotRan;
int counter = 0;
int failedAttempts = 0;
bool alarmActive = false;
unsigned long alarmStartTime = 0;

// WiFi dan Telegram
const char* ssid = "Realme";
const char* password = "Realme08";
```

```
const char* botToken = "6618102043:AAF0eJj6kbeMq5KGZbyrsa_aMs8Du06-ooY";
const char* chatID = "5537824420";

WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);

void setup() {
  Serial.begin(115200);
  finger.begin(57600);

  delay(10);

  // Inisialisasi LCD
  lcd.init();
  lcd.backlight();

  // Inisialisasi fingerprint sensor
  if (finger.verifyPassword()) {
    Serial.println("Sensor sidik jari ditemukan.");
    lcd.setCursor(0, 0);
    lcd.print("Sensor sidikjari");
    lcd.setCursor(0, 1);
    lcd.print(" --CONNECTED--  ");
  } else {
    Serial.println("Sensor sidik jari tidak ditemukan, pastikan koneksi sudah benar.");
    lcd.setCursor(0, 0);
    lcd.print("Sensor sidikjari");
    lcd.setCursor(0, 1);
    lcd.print("--DISCONNECTED--");
```

```
  while (1) { delay(1); }
}
delay(1000);
lcd.clear();

Serial.println(F("Reading sensor parameters"));
finger.getParameters();
Serial.print(F("Status: 0x")); Serial.println(finger.status_reg, HEX);
Serial.print(F("Sys ID: 0x")); Serial.println(finger.system_id, HEX);
Serial.print(F("Capacity: ")); Serial.println(finger.capacity);
Serial.print(F("Security level: ")); Serial.println(finger.security_level);
Serial.print(F("Device address: ")); Serial.println(finger.device_addr, HEX);
Serial.print(F("Packet len: ")); Serial.println(finger.packet_len);
Serial.print(F("Baud rate: ")); Serial.println(finger.baud_rate);

  lcd.setCursor(0, 0);
  lcd.print("  BRANGKAS IOT  ");
  lcd.setCursor(0, 1);
  lcd.print("    BY SYADA    ");

// Inisialisasi pin
pinMode (buzzer, OUTPUT);
pinMode (selenoid, OUTPUT);
pinMode (espcam, OUTPUT);

digitalWrite (buzzer, LOW);
delay (1500);

digitalWrite (selenoid, HIGH);
digitalWrite (buzzer, HIGH);
digitalWrite (espcam, LOW);
```

```
// Tampilkan pesan awal
lcd.setCursor (0, 0);
lcd.print(F("INISIALISASI I/O"));
Serial.println("INISIALISASI I/O");

delay(1500);

Serial.print("Connecting");
lcd.setCursor (0, 0);
lcd.print(F("Connecting Wi-Fi"));
lcd.setCursor (0, 1);

// Inisialisasi koneksi WiFi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
  delay(200);
  if (++counter > 50)
    ESP.restart();
  Serial.print( "." );
}

Serial.println("Connected to WiFi");
lcd.setCursor (0, 1);
lcd.print(" --CONNECTED-- ");
delay (2000);
lcd.clear();

client.setInsecure(); // Disable SSL certificate verification

lcd.setCursor (0, 0);
```

```
 lcd.print(F("Connecting Tele"));
 lcd.setCursor (0, 1);
 lcd.print(" --CONNECTED-- ");
 delay (2000);
 lcd.clear();

 Serial.println();
 Serial.println("Wi-Fi & Telegram");
 Serial.println(" --Connected-- ");

 lcd.setCursor (0, 0);
 lcd.print(F("Wi-Fi & Telegram"));
 lcd.setCursor (0, 1);
 lcd.print(F(" --CONNECTED-- "));

 // Kirim notifikasi ke Telegram saat sistem mulai
 bot.sendMessage(chatID, "System Brangkas Ready!", "markdown");
 Serial.println("Pesan Terkirim");
 delay (2000);
 lcd.clear();
}

void loop() {
 if (alarmActive) {
   // Jika alarm aktif, periksa apakah waktu alarm telah habis
   if (millis() - alarmStartTime >= 5000) {
     // Matikan buzzer dan reset alarm
     digitalWrite(buzzer, HIGH);
     digitalWrite(selenoid, HIGH);
     alarmActive = false;
     lcd.clear();
```

```
    bot.sendMessage(chatID, "Alarm telah dimatikan, System Kembali Normal",
"");
    failedAttempts = 0;
   }
   return;  // Alarm aktif, hentikan eksekusi loop
 }

 lcd.setCursor(0, 0);
 lcd.print("  BRANGKAS IOT  ");
 lcd.setCursor(0, 1);
 lcd.print("SCAN SIDIK JARI!");

 if (finger.getImage() == FINGERPRINT_OK){
  Serial.println("image taken");
   if (finger.image2Tz() == FINGERPRINT_OK) {
    Serial.println("Image Converted") ;
    if(finger.fingerFastSearch() == FINGERPRINT_OK) {
     digitalWrite(espcam, HIGH);
     Serial.println("Found a print match!");
     lcd.clear();
     lcd.setCursor(0, 0);
     lcd.print("Akses Diterima  ");
     lcd.setCursor(0, 1);
     lcd.print("ID: "); lcd.print(finger.fingerID);
     delay(1500);
     digitalWrite(espcam, LOW);
     lcd.clear();
     lcd.setCursor(0, 0);
     lcd.print("Send Notif Tele");
     bot.sendMessage(chatID, "Akses diterima, ID: " + String(finger.fingerID),
"");
```

```
    lcd.clear();
    digitalWrite(selenoid, LOW); // Buka kunci pintu
    lcd.setCursor(0, 0);
    lcd.print("  SYSTEM UNLOCK ");
    lcd.setCursor(0, 1);
    lcd.print(" BRANKAS TERBUKA");
    delay(6000);
    lcd.setCursor(0, 0);
    lcd.print("    BRANGKAS    ");
    lcd.setCursor(0, 1);
    lcd.print("    TERKUNCI    ");
    digitalWrite(selenoid, HIGH); // Kunci kembali pintu
    failedAttempts = 0; // Reset hitungan percobaan gagal
    delay(1000);
    lcd.clear();
  }
  else if (finger.fingerFastSearch() == FINGERPRINT_NOTFOUND) {
    digitalWrite(espcam, HIGH);
    Serial.println("Did not find a match");
    lcd.setCursor(0, 0);
    lcd.print("  data Finger   ");
    lcd.setCursor(0,1);
    lcd.print("Tidak Ditemukan ");
    delay(1000);
    digitalWrite(espcam, LOW);
    lcd.clear();

    failedAttempts++;
    if (failedAttempts >= 3) {
      lcd.clear();
      lcd.setCursor(0, 0);
```

```
          lcd.print("Akses Ditolak   ");
          lcd.setCursor(0, 1);
          lcd.print("Alarm Aktif     ");
          delay(1000);
          triggerAlarm();
          return;  // Alarm aktif, hentikan eksekusi loop
         }

         else {
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("Akses Ditolak   ");
          lcd.setCursor(0, 1);
          lcd.print("Percobaan: ");
          lcd.print(failedAttempts);
          bot.sendMessage(chatID, "Akses ditolak, Percobaan: " +
String(failedAttempts), "");
          delay(2000); // Tunda untuk menampilkan pesan
          lcd.clear();
         }
       }
       // else {
       //   Serial.println("unknown error");
       // }
     }
     // else {
     //   Serial.println("fail take fingerprint!!");
     // }
   }
   newmessage();
}
```

```
void triggerAlarm() {
  // Aktifkan buzzer dan setel waktu alarm
  digitalWrite(buzzer, LOW);
  alarmStartTime = millis();
  alarmActive = true;
  lcd.setCursor(0, 0);
  lcd.print("SYSTEM TERKUNCI!");
  lcd.setCursor(0, 1);
  lcd.print("                ");
  bot.sendMessage(chatID, "Alarm Aktif!", "");
  return;
}
```

Program ESP32-CAM

```
#include <WiFi.h>
#include <Wire.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

const char* ssid = "Realme";
const char* password = "Realme08";
// Initialize Telegram BOT
String BOTtoken = "6618102043:AAF0eJj6kbeMq5KGZbyrsa_aMs8Du06-ooY";
// your Bot Token (Get from Botfather)
// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
```

```
// message you
String CHAT_ID = "5537824420";
bool sendPhoto = false;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_LED_PIN 4
bool flashState = LOW;

// Motion Sensor
bool movementalert = false;

//Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

//CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

```
static void IRAM_ATTR detectsMovement(void * arg){
  Serial.println("gerakan terdeteksi");
  movementalert = true;
}

void configInitCamera(){
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
```

```
  config.jpeg_quality = 10;  //0-63 lower number means higher quality
  config.fb_count = 2;
 } else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;  //0-63 lower number means higher quality
  config.fb_count = 1;
 }

 // camera init
 esp_err_t err = esp_camera_init(&config);
 if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  delay(1000);
  ESP.restart();
 }

 // Drop down frame size for higher initial frame rate
 sensor_t * s = esp_camera_sensor_get();
 s->set_framesize(s,                FRAMESIZE_CIF);                                //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA


 // PIR Motion Sensor mode INPUT_PULLUP
 //err = gpio_install_isr_service(0);
 err = gpio_isr_handler_add(GPIO_NUM_13, &detectsMovement, (void *) 13);
 if (err != ESP_OK){
  Serial.printf("handler add failed with error 0x%x \r\n", err);
 }
 err = gpio_set_intr_type(GPIO_NUM_13, GPIO_INTR_POSEDGE);
 if (err != ESP_OK){
  Serial.printf("set intr type failed with error 0x%x \r\n", err);
 }
}
```

```
void handleNewMessages(int numNewMessages) {
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;
    if (text == "/start") {
      String welcome = "Welcome , " + from_name + "\n";
      welcome += "Gunakan Perintah dibawah ini untuk berkomunikasi ke ESP32-
CAM \n";
      welcome += "/photo : Ambil foto keadaan sekarang\n";
      bot.sendMessage(CHAT_ID, welcome, "");
    }
    if (text == "/flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
      Serial.println("Change flash LED state");
    }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("New photo request");
```

```
  }
 }
}

String sendPhotoTelegram() {
 const char* myDomain = "api.telegram.org";
 String getAll = "";
 String getBody = "";
 camera_fb_t * fb = NULL;
 fb = esp_camera_fb_get();
 if(!fb) {
  Serial.println("Camera capture failed");
  delay(1000);
  ESP.restart();
  return "Camera capture failed";
 }
 Serial.println("Connect to " + String(myDomain));
 if (clientTCP.connect(myDomain, 443)) {
  Serial.println("Connection successful");

  String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat_id\"; \r\n\r\n" + CHAT_ID + "\r\n--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
  String tail = "\r\n--RandomNerdTutorials--\r\n";

  uint16_t imageLen = fb->len;
  uint16_t extraLen = head.length() + tail.length();
  uint16_t totalLen = imageLen + extraLen;

  clientTCP.println("POST /bot"+BOTtoken+"/sendPhoto HTTP/1.1");
```

```
  clientTCP.println("Host: " + String(myDomain));
  clientTCP.println("Content-Length: " + String(totalLen));
  clientTCP.println("Content-Type: multipart/form-data;
boundary=RandomNerdTutorials");
  clientTCP.println();
  clientTCP.print(head);

  uint8_t *fbBuf = fb->buf;
  size_t fbLen = fb->len;
  for (size_t n=0;n<fbLen;n=n+1024) {
   if (n+1024<fbLen) {
    clientTCP.write(fbBuf, 1024);
    fbBuf += 1024;
   }
   else if (fbLen%1024>0) {
    size_t remainder = fbLen%1024;
    clientTCP.write(fbBuf, remainder);
   }
  }

  clientTCP.print(tail);

  esp_camera_fb_return(fb);

  int waitTime = 10000;   // timeout 10 seconds
  long startTimer = millis();
  boolean state = false;

  while ((startTimer + waitTime) > millis()){
   Serial.print(".");
   delay(100);
```

```
    while (clientTCP.available()) {
      char c = clientTCP.read();
      if (state==true) getBody += String(c);
      if (c == '\n') {
        if (getAll.length()==0) state=true;
        getAll = "";
      }
      else if (c != '\r')
        getAll += String(c);
      startTimer = millis();
    }
    if (getBody.length()>0) break;
  }
  clientTCP.stop();
  Serial.println(getBody);
 }
 else {
  getBody="Connected to api.telegram.org failed.";
  Serial.println("Connected to api.telegram.org failed.");
 }
 return getBody;
}

void setup(){
 WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
 // Init Serial Monitor
 Serial.begin(115200);

 // Set LED Flash as output
 pinMode(FLASH_LED_PIN, OUTPUT);
 digitalWrite(FLASH_LED_PIN, flashState);
```

```
  // Config and init the camera
  configInitCamera();

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);    //    Add    root
certificate for api.telegram.org
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
  Serial.println(WiFi.localIP());

  bot.sendMessage(CHAT_ID, "System Kamera Siap!", "");
  Serial.println("Pesan Terkirim");
}

void loop() {
  if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }

  if(movementalert){
```

```
   // bot.sendMessage(CHAT_ID, "Gerakan terdeteksi di depan pintu", "");
   Serial.println("Gerakan terdeteksi");
   // bot.sendMessage(CHAT_ID, "Foto Kamera 1", "");
   sendPhotoTelegram();
   movementalert = false;
   sendPhoto = false;
  }

  if (millis() > lastTimeBotRan + botRequestDelay)  {
   int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
   while (numNewMessages) {
    Serial.println("got response");
    handleNewMessages(numNewMessages);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
   }
   lastTimeBotRan = millis();
  }
}
```

Program Enroll untuk mendaftrakan Sidik Jari

```
#include <Adafruit_Fingerprint.h>
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);
uint8_t id;

void setup()
{
  Serial.begin(9600);
  Serial2.begin(57600);
  // while (!Serial);  // For Yun/Leo/Micro/Zero/...
  // delay(100);
```

```
  Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

  // set the data rate for the sensor serial port
  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }

  Serial.println(F("Reading sensor parameters"));
  finger.getParameters();
  Serial.print(F("Status: 0x")); Serial.println(finger.status_reg, HEX);
  Serial.print(F("Sys ID: 0x")); Serial.println(finger.system_id, HEX);
  Serial.print(F("Capacity: ")); Serial.println(finger.capacity);
  Serial.print(F("Security level: ")); Serial.println(finger.security_level);
  Serial.print(F("Device address: ")); Serial.println(finger.device_addr, HEX);
  Serial.print(F("Packet len: ")); Serial.println(finger.packet_len);
  Serial.print(F("Baud rate: ")); Serial.println(finger.baud_rate);
}

uint8_t readnumber(void) {
  uint8_t num = 0;
  while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
  }
  return num;
}
void loop()                     // run over and over again
{
```

```
 Serial.println("Ready to enroll a fingerprint!");
 Serial.println("Please type in the ID # (from 1 to 127) you want to save this finger
as...");
 id = readnumber();
 if (id == 0) {// ID #0 not allowed, try again!
    return;
 }
 Serial.print("Enrolling ID #");
 Serial.println(id);

 while (! getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

 int p = -1;
 Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
 while (p != FINGERPRINT_OK) {
   p = finger.getImage();
   switch (p) {
   case FINGERPRINT_OK:
     Serial.println("Image taken");
     break;
   case FINGERPRINT_NOFINGER:
     Serial.print(".");
     break;
   case FINGERPRINT_PACKETRECIEVEERR:
     Serial.println("Communication error");
     break;
   case FINGERPRINT_IMAGEFAIL:
     Serial.println("Imaging error");
     break;
   default:
```

```
    Serial.println("Unknown error");
    break;
  }
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
```

```
  p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
```

```
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
Serial.print("Creating model for #");  Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
  Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
  Serial.println("Fingerprints did not match");
  return p;
```

```
 } else {
  Serial.println("Unknown error");
  return p;
 }

 Serial.print("ID "); Serial.println(id);
 p = finger.storeModel(id);
 if (p == FINGERPRINT_OK) {
  Serial.println("Stored!");
 } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  Serial.println("Communication error");
  return p;
 } else if (p == FINGERPRINT_BADLOCATION) {
  Serial.println("Could not store in that location");
  return p;
 } else if (p == FINGERPRINT_FLASHERR) {
  Serial.println("Error writing to flash");
  return p;
 } else {
  Serial.println("Unknown error");
  return p;
 }

 return true;
}
```

KEMENTRIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

SURAT KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama	: Amrina Rosyada
NIM	: 062130700202
Jurusan	: Teknik Komputer
Program Studi	: D3 Teknik Komputer

Pihak Kedua

Nama	: Azwardi, S.T., MT.
NIP	: 197005232005011004
Jurusan	: Teknik Komputer
Program Studi	: D3 Teknik Komputer

Pada hari ini Jum'at tanggal 22 bulan Maret tahun 2024 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.
Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Jum'at pukul 02.00 PM - Selesai tempat di Politeknik Negeri Sriwijaya.
Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Palembang, Juli 2024

Pihak Pertama,

Amrina Rosyada
NIM. 062130700202

Pihak Kedua,

Azwardi, S.T., MT.
NIP 197005232005011004

Mengetahui,
Ketua Jurusan

Azwardi, S.T., MT.
NIP 197005232005011004

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara Bukit Besar – Palembang 30139 Telp. 0711-353414
Laman : http://polsri.ac.id, Pos El : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN TUGAS AKHIR

Nama	: Amrina Rosyada
NIM	: 062130700202
Jurusan/Program Studi	: Teknik Komputer/DIII Teknik Komputer
Dosen Pembimbing	: Azwardi, ST, MT.
Judul	: Rancang Bangun Sistem Pengamanan Brankas Menggunakan *Fingerprint* Dan Esp32-Cam Berbasis *Internet Of Things* (IoT).

No	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
	24/06 2024	Bab 1 dan Bab 2	
	27/06 2024	Revisi bab 1 dan Bab 2	
	2/07 2024	Acc bab 1 dan Bab 2	
		Bab 3, revisi	
	12/07 2024	Acc Bab 3, revisi bab 4	
	19/07 2024	Acc Bab 4	
	25/07 2024	Acc Bab 5	
	29/07 2024	Acc Bab 1, Bab 2, Bab 3, Bab 4, Bab 5	

Palembang, Juli 2024

Ketua Jurusan,

Azwardi, ST, MT.

NIP. 197005232005011004

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara Bukit Besar – Palembang 30139 Telp. 0711-353414
Laman : http://polsri.ac.id, Pos El : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN TUGAS AKHIR

Nama	: Amrina Rosyada
NIM	: 062130700202
Jurusan/Program Studi	: Teknik Komputer/DIII Teknik Komputer
Dosen Pembimbing	: Ali Firdaus, M. Kom.
Judul	: Rancang Bangun Sistem Pengamanan Brankas Menggunakan *Fingerprint* Dan Esp32-Cam Berbasis *Internet Of Things* (IoT).

No	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
	24/06 2024	Bab 1 dan Bab 2	
		Revisi Bab 1 dan Bab 2	
	8/07 2024	Acc Bab 1 dan Bab 2	
		Bab 3 dan Bab 4	
	12/07 2024	Revisi Bab 3 dan Bab 4	
	15/07 2024	Acc Bab 3 , Revisi Bab 4	
	19/07 2024	Acc Bab 4	
	26/07 2024	Acc Bab 1, Bab 2, Bab 3, Bab 4 Bab 5	

Palembang, Juli 2024
Ketua Jurusan,

Azwardi, ST, MT.
NIP. 197005232005011004

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI TUGAS AKHIR (TA)	

Dosen Penguji : Yulian Mirza, S.T., M.Kom
Nama Mahasiswa : Amrina Rosyada
NIM : 062130700202
Jurusan/Program Studi : Teknik Komputer/DIII Teknik Komputer
Judul LA/Skripsi : Rancang Bangun Sistem Pengamanan Brankas Menggunakan *Fingerprint* Dan Esp32-Cam Berbasis *Internet Of Things* (IoT).

No	Uraian Revisi	Paraf
	data uji pembahasan kesimpulan	

Palembang, Juli 2024
Dosen Penguji,

Yulian Mirza, S.T., M.Kom
NIP. 196607121990031003

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI LAPORAN TUGAS AKHIR (TA)	

Dosen Penguji : Ir. Alan Novi Tompunu, S.T., M.T.
Nama Mahasiswa : Amrina Rosyada
NIM : 062130700202
Jurusan/Program Studi : Teknik Komputer/DIII Teknik Komputer
Judul LA/Skripsi : Rancang Bangun Sistem Pengamanan Brankas Menggunakan *Fingerprint* Dan Esp32-Cam Berbasis *Internet Of Things* (IoT).

No	Uraian Revisi	Paraf
	Jatala.kp @ gmail.com.	

Palembang, Juli 2024
Dosen Penguji,

Ir. Alan Novi Tompunu, S.T., M.T.
NIP. 197611082000031002

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI LAPORAN TUGAS AKHIR (TA)	

Dosen Penguji : Hartati Deviana, S.T.,M.Kom.
Nama Mahasiswa : Amrina Rosyada
NIM : 062130700202
Jurusan/Program Studi : Teknik Komputer/DIII Teknik Komputer
Judul LA/Skripsi : Rancang Bangun Sistem Pengamanan Brankas Menggunakan *Fingerprint* Dan Esp32-Cam Berbasis *Internet Of Things* (IoT).

No	Uraian Revisi	Paraf
	Perbaiki tata tulis Perbaiki bab 2, bab 3, bab 4.	

Palembang, Juli 2024
Dosen Penguji,

Hartati Deviana, S.T.,M.Kom.
NIP. 197405262008122001

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI LAPORAN TUGAS AKHIR (TA)	

Dosen Penguji : Rian Rahmanda Putra, S.Kom., M.Kom
Nama Mahasiswa : Amrina Rosyada
NIM : 062130700202
Jurusan/Program Studi : Teknik Komputer/DIII Teknik Komputer
Judul LA/Skripsi : Rancang Bangun Sistem Pengamanan Brankas Menggunakan *Fingerprint* Dan Esp32-Cam Berbasis *Internet Of Things* (IoT).

No	Uraian Revisi	Paraf
	Tambahkan arsitektur sistem IOT	

Palembang, Juli 2024
Dosen Penguji,

Rian Rahmanda Putra, S.Kom., M.Kom
NIP. 198901252019031013

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	PELAKSANAAN REVISI LAPORAN TUGAS AKHIR	

Mahasiswa berikut,

Nama : Amrina Rosyada

NIM : 062130700202

Jurusan/Program Studi : Teknik Komputer/DIII Teknik Komputer

Judul Laporan Tugas Akhir : Rancang Bangun Sistem Pengamanan Brankas Menggunakan *Fingerprint* Dan Esp32-Cam Berbasis Internet Of Things (IoT).

Telah melaksanakan revisi terhadap Laporan Akhir (LA) yang diseminarkan pada Hari Rabu Tanggal 31 Bulan Juli Tahun 2024. Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penilai yang memberikan revisi:

No.	Komentar	Nama Dosen Penilai	Tanggal	Tanda Tangan
1.	Acc	Yulian Mirza, S.T., M.Kom.	23/9-2024	
2.	Acc	Ir. Alan Novi Tompunu, S.T.,M.T., IPM., ASEAN Eng	30/9/2024	
3.	Acc	Hartati Deviana, S.T.,M.Kom.	25/9/2024	
4.	Acc	Rian Rahmanda Putra, S.Kom.,M.Kom.	30/09-2024	

Palembang, Juli 2024

Ketua Penguji,

Yulian Mirza, S.T., M.Kom

NIP. 196607121990031003