LAMPIRAN

```
#include "esp_camera.h"
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h"            // disable brownout problems
#include "soc/rtc_cntl_reg.h"   // disable brownout problems
#include "esp_http_server.h"
#include <WiFi.h>
// Telegram
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

// GPS
#include <SoftwareSerial.h>
#include <TinyGPS++.h>

// Replace with your network credentials
const char* ssid = "Sn";
const char* password = "0987654321";

// Initialize Telegram BOT
String BOTtoken =
"6700497583:AAGS8pRsvPhD9JNQ3BZ9PbkAYxRRz8Gs3vs";   // Replace
with your Bot Token
String CHAT_ID = "1133606550";     // Replace with your Chat ID

// GPS Module
```

```
SoftwareSerial gpsSerial(13, 15); // RX, TX (ESP32 GPIO13 and GPIO15)
TinyGPSPlus gps;

// Buzzer
#define BUZZER_PIN 12
//FLASH
#define FLASH_LED_PIN 4
bool flashState = LOW;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

bool sendPhoto = false;
bool buzzerState = LOW;
bool welcomeMessageSent = false; // Variable to track if the welcome message has been sent

int botRequestDelay = 1000;
unsigned long lastTimeBotRan;
// Function Declaration
String getGPSLocation();
void triggerBuzzer();

// Camera configuration
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"

#define PART_BOUNDARY "123456789000000000000987654321"

#define CAMERA_MODEL_AI_THINKER
```

```
// Konfigurasi pin kamera
#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

#define _STREAM_CONTENT_TYPE "multipart/x-mixed-replace; boundary="
PART_BOUNDARY
#define _STREAM_PART "Content-Type: image/jpeg\r\nContent-
Length: %u\r\n\r\n"
#define _STREAM_BOUNDARY "--" PART_BOUNDARY "\r\n"

httpd_handle_t camera_httpd = NULL;
httpd_handle_t stream_httpd = NULL;

static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<html>
  <head>
```

```
<title>CCTV SMARTHOME</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
 body { font-family: Arial; text-align: center; margin:0px auto; padding-top:
30px;}
 table { margin-left: auto; margin-right: auto; }
 td { padding: 8 px; }
 .button {
  background-color: #2f4468;
  border: none;
  color: white;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 18px;
  margin: 6px 3px;
  cursor: pointer;
  -webkit-touch-callout: none;
  -webkit-user-select: none;
  -khtml-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
  -webkit-tap-highlight-color: rgba(0,0,0,0);
 }
 img {  width: auto ;
  max-width: 100% ;
  height: auto ;
 }
</style>
```

```
  </head>
  <body>
    <h1>CCTV SMARTHOME</h1>
    <img src="https://your-streaming-url/81/stream" id="photo" >
  </body>
</html>
)rawliteral";

// Handler untuk halaman utama
static esp_err_t index_handler(httpd_req_t *req) {
  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, (const char *)INDEX_HTML,
strlen(INDEX_HTML));
}
// Handler untuk streaming kamera
static esp_err_t stream_handler(httpd_req_t *req) {
  camera_fb_t * fb = NULL;
  esp_err_t res = ESP_OK;
  size_t _jpg_buf_len = 0;
  uint8_t * _jpg_buf = NULL;
  char * part_buf[64];

  res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
  if (res != ESP_OK) {
    return res;
  }

  while (true) {
    fb = esp_camera_fb_get();
    if (!fb) {
      Serial.println("Camera capture failed");
```

```
      res = ESP_FAIL;
    } else {
      if (fb->width > 400) {
        if (fb->format != PIXFORMAT_JPEG) {
          bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
          esp_camera_fb_return(fb);
          fb = NULL;
          if (!jpeg_converted) {
            Serial.println("JPEG compression failed");
            res = ESP_FAIL;
          }
        } else {
          _jpg_buf_len = fb->len;
          _jpg_buf = fb->buf;
        }
      }
    }
    if (res == ESP_OK) {
      size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
      res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
    }
    if (res == ESP_OK) {
      res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
    }
    if (res == ESP_OK) {
      res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
    }
    if (fb) {
      esp_camera_fb_return(fb);
      fb = NULL;
```

```
      _jpg_buf = NULL;
    } else if (_jpg_buf) {
      free(_jpg_buf);
      _jpg_buf = NULL;
    }
    if (res != ESP_OK) {
      break;
    }
  }
  return res;
}
// Handler untuk perintah
esp_err_t cmd_handler(httpd_req_t *req) {
    char* buf;
    size_t buf_len;
    int ret;

    buf_len = httpd_req_get_url_query_len(req) + 1;
    if (buf_len > 1) {
        buf = (char*)malloc(buf_len);
        if (!buf) {
            httpd_resp_send_500(req);
            return ESP_FAIL;
        }
        if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
            if (httpd_query_key_value(buf, "cmd", buf, buf_len) == ESP_OK) {
                if (strcmp(buf, "location") == 0) {
                    // Logika untuk mendapatkan lokasi GPS
                     String location = getGPSLocation();
                    httpd_resp_send(req, location.c_str(),
HTTPD_RESP_USE_STRLEN);
```

```
            } else if (strcmp(buf, "alert") == 0) {
                // Logika untuk mengaktifkan buzzer
                triggerBuzzer();
                httpd_resp_send(req, "Buzzer activated",
HTTPD_RESP_USE_STRLEN);
            } else {
                httpd_resp_send_404(req);
            }
        } else {
            httpd_resp_send_404(req);
        }
    } else {
        httpd_resp_send_500(req);
    }
    free(buf);
  } else {
    httpd_resp_send_404(req);
  }
  return ESP_OK;
}

// Memulai server kamera
void startCameraServer() {
 httpd_config_t config = HTTPD_DEFAULT_CONFIG();
 config.server_port = 80;

 httpd_uri_t index_uri = {
  .uri      = "/",
  .method   = HTTP_GET,
  .handler  = index_handler,
  .user_ctx  = NULL
```

```
};

httpd_uri_t stream_uri = {
  .uri       ="/stream",
  .method    = HTTP_GET,
  .handler   = stream_handler,
  .user_ctx  = NULL
};

httpd_uri_t cmd_uri = {
  .uri       = "/action",
  .method    = HTTP_GET,
  .handler   = cmd_handler,
  .user_ctx  = NULL
};


 if (httpd_start(&camera_httpd, &config) == ESP_OK) {
  Serial.println("Registering index URI");
  httpd_register_uri_handler(camera_httpd, &index_uri);
  Serial.println("Index URI registered");

  Serial.println("Registering cmd URI");
  httpd_register_uri_handler(camera_httpd, &cmd_uri);
  Serial.println("Cmd URI registered");

  Serial.println("Registering stream URI");
  httpd_register_uri_handler(camera_httpd, &stream_uri);
  Serial.println("Stream URI registered");
}
  config.server_port += 1;
```

```
    config.ctrl_port += 1;

    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
      httpd_register_uri_handler(stream_httpd, &stream_uri);
    }
  }

void triggerBuzzer() {
  tone(BUZZER_PIN, 2000); // Start buzzer with 1000 Hz frequency
  delay(5000);            // Keep the buzzer on for 1 second
  noTone(BUZZER_PIN);     // Stop the buzzer
}

String getGPSLocation() {
   String latitude, longitude;
  while (gpsSerial.available() > 0) {
    char c = gpsSerial.read();
    Serial.print(c);  // Debug: Cetak karakter yang dibaca
    gps.encode(c);
    if (gps.location.isUpdated()) {
      latitude = String(gps.location.lat(), 6);
      longitude = String(gps.location.lng(), 6);
      break;
    }
  }

  if (latitude.length() > 0 && longitude.length() > 0) {
    String locationURL = "https://www.google.com/maps?q=" + latitude + "," +
longitude;
    Serial.println("Location: " + locationURL); // Debug: Cetak URL lokasi
    return locationURL;
```

```
  } else {
    return "Unable to get GPS location.";
  }
}
// Fungsi untuk mengirim pesan Telegram dengan keyboard
void sendMenu(String chat_id) {
  String welcome = "Selamat datang di CHATBOT GPS TRACKER!\n";
  welcome += "Silakan pilih perintah dari menu di bawah:\n";

  // Membuat custom keyboard
  String keyboardJson = F("[[\"    Ambil Foto\"],[\"    Lokasi GPS\"],[\"
Aktifkan Buzzer\"],[\"    Toggle Flash\"]]");
  bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson, true);
}

void handleNewMessages(int numNewMessages) {
  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    String text = bot.messages[i].text;
    String from_name = bot.messages[i].from_name;
    if (from_name == "") from_name = "Guest";

    // Menangani perintah dari pengguna
    if (text == "/start") {
      sendMenu(chat_id);
    } else if (text == "    Ambil Foto") {
      sendPhoto = true;
      bot.sendMessage(chat_id, "Mengambil foto...", "");
    } else if (text == "    Lokasi GPS") {
      String location = getGPSLocation();
      bot.sendMessage(chat_id, location, "");
```

```
      Serial.println("Sent location: " + location);
    } else if (text == "    Aktifkan Buzzer") {
      triggerBuzzer();
      bot.sendMessage(chat_id, "Buzzer diaktifkan.", "");
    } else if (text == "    Toggle Flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
      bot.sendMessage(chat_id, flashState ? "Flash is ON" : "Flash is OFF", "");
    }
  }
}
String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

  //Dispose first picture because of bad quality
  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  esp_camera_fb_return(fb); // dispose the buffered image

  // Take a new photo
  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";

    size_t imageLen = fb->len;
    size_t extraLen = head.length() + tail.length();
    size_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); //disable brownout
detector

 Serial.begin(115200);
 Serial.setDebugOutput(true);
 Serial.println();
 // Setup Buzzer
 pinMode(BUZZER_PIN, OUTPUT);
 digitalWrite(BUZZER_PIN, LOW);
 // Setup Flash
 pinMode(FLASH_LED_PIN, OUTPUT);
 digitalWrite(FLASH_LED_PIN, LOW);
 // Setup GPS
 gpsSerial.begin(9600);

 // Initialize the camera
 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;

if (psramFound()) {
  config.frame_size = FRAMESIZE_UXGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
  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);
  return;
```

```
  }

  // Wi-Fi connection
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Telegram BOT
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root
certificate for api.telegram.org
  bot.updateToken(BOTtoken);

  startCameraServer();
}

void loop() {
   if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }
  if (millis() - lastTimeBotRan > botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
```

```
  }
  while (gpsSerial.available() > 0) {
    gps.encode(gpsSerial.read());
  }
}
```

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

REKOMENDASI UJIAN TUGAS AKHIR

Pembimbing Laporan Tugas Akhir, memberikan rekomendasi ujian laporan tugas akhir kepada,

Nama	:	Ajeng Putri Lestari
NIM	:	062130701710
Jurusan / Program Studi	:	Teknik Komputer / DIII Teknik Komputer
Judul Tugas Akhir	:	Rancang Bangun Alat Pelacak Hewan Berbasis *Internet of Things*

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Tugas Akhir, pada Tahun Akademik 2023/2024.

Palembang, 2024

Disetujui oleh,
Pembimbing I

Slamet Widodo, M.Kom
NIP. 197305162002121001

Pembimbing II

Mustaziri, S.T., M.Kom
NIP. 196909282005011002

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

LEMBAR BIMBINGAN LAPORAN AKHIR

Nama Mahasiswa	:	Ajeng Putri Lestari
NIM	:	062130701710
Jurusan/Program Studi	:	Teknik Komputer / DIII Teknik Komputer
Dosen Pembimbing 2	:	Slamet Widodo,M.Kom
Judul	:	Rancang Bangun Alat Pelacak Hewan Berbasis Internet of Things

NO.	TANGGAL	URAIAN	PARAF PEMBIMBING
1	24 Juni 2024	Revisi Bab 1, 2, 3	
2	01 Juli 2024	Revisi bab 1, 3	
3	09 Juli 2024	Revisi alat	
4.	10 Juli 2024	ACC BAB I	
5.	11 Juli 2024	ACC BAB II	
6.	15 Juli 2024	ACC BAB III	
7.	18 Juli 2024	Revisi Kapasitas Camera	
8.	22 Juli 2024	Bab IV	
		ACC BAB IV	
9.	25/07 2024.	ACC BAB V	

Palembang, 2024
Mengetahui,
Ketua Jurusan Teknik Komputer

Azwardi, S.T.,M.T
NIP. 197005232005011004

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

LEMBAR BIMBINGAN LAPORAN AKHIR

Nama Mahasiswa	:	Ajeng Putri Lestari
NIM	:	062130701710
Jurusan/Program Studi	:	Teknik Komputer / DIII Teknik Komputer
Dosen Pembimbing 2	:	Mustaziri, S.T.,M.Kom
Judul	:	Rancang Bangun Alat Pelacak Hewan Berbasis Internet of Things

NO.	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	26 Juni 2024	Revisi Bab 1	
2.	1 Juli 2024	Revisi Bab II	
3.	4 Juli 2024	Revisi Bab III	
4.	10 Juli 2024	Revisi Bab IV	
5.	15 Juli 2024	Revisi Bab IV	
6.	17 Juli 2024	Revisi alat	
7.	18 Juli 2024	Acc Alat	
8.	22 Juli 2024	Revisi tabel pengujian	
9.	22 Juli 2024	Acc Bab IV	
10.	25 Juli 2024	Acc Bab V	

Palembang, 2024
Mengetahui,
Ketua Jurusan Teknik Komputer

Azwardi, S.T.,M.T
NIP. 197005232005011004

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

PELAKSANAAN REVISI UJIAN TUGAS AKHIR (TA)

Nama Mahasiswa : Ajeng Putri Lestari
NIM : 062130701710
Jurusan/Program Studi : Teknik Komputer/D3-Teknik Komputer
Judul Tugas Akhir : Rancang Bangun Alat Pelacak Hewan Berbasis *Internet of Things.*

Telah melaksanakan revisi terhadap Laporan Tugas Akhir yang diujikan pada hari Selasa tanggal 29 bulan Juli tahun 2024.
Pelaksanaan revisi terhadap Laporan tugas Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi.

No	Komentar	Nama Dosen Penguji	Tanggal/ Bulan	Tanda Tangan
1	Acc	Yulian Mirza, S.T., M.Kom	12/8 2024	
2	Acc	Ir. Alan Novi Tompunu, S.T., M.T., IPM., ASEAN Eng	16/8 24	
3	Acc	Hartati Deviana, S.T., M.Kom	16/8 2024	
4	Acc	Rian Rahmanda Putra, S.Kom., M.Kom	12/8 2024	

Palembang, Juli 2024
Ketua Penguji

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

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR (TA)

Dosen Penguji	: Yulian Mirza, S.T., M.Kom
Nama	: Ajeng Putri Lestari
NIM	062130701710
Jurusan/Program Studi	: Teknik Komputer/D3-Teknik Komputer
Judul Tugas Akhir	: Rancang Bangun Alat Pelacak Hewan Berbasis *Internet of Things.*

No	Uraian	Paraf
	tata tulis Pembahasan Kesimpulan	

Palembang, Juli 2024
Dosen Penguji

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

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id **E-mail :** info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR (TA)

Dosen Penguji : Ir. Alan Novi Tompunu, S.T., M.T., IPM., ASEAN Eng
Nama : Ajeng Putri Lestari
NIM : 062130701710
Jurusan/Program Studi : Teknik Komputer/D3-Teknik Komputer
Judul Tugas Akhir : Rancang Bangun Alat Pelacak Hewan Berbasis *Internet of Things.*

No	Uraian	Paraf
	Saran datata.kp@gmail.com	[signature]

Palembang, Juli 2024
Dosen Penguji

Ir. Alan Novi Tompunu, S.T., M.T., IPM., ASEAN Eng
NIP. 197611082000031002

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR (TA)

Dosen Penguji	: Hartati Deviana, S.T., M.Kom
Nama	: Ajeng Putri Lestari
NIM	062130701710
Jurusan/Program Studi	: Teknik Komputer/D3-Teknik Komputer
Judul Tugas Akhir	: Rancang Bangun Alat Pelacak Hewan Berbasis *Internet of Things*.

No	Uraian	Paraf
	Perbaiki bab IV Tambahkan narasi dari pengujian.	

Palembang, Juli 2024
Dosen Penguji

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

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR (TA)

Dosen Penguji	: Rian Rahmanda Putra, S.Kom., M.Kom
Nama	: Ajeng Putri Lestari
NIM	062130701710
Jurusan/Program Studi	: Teknik Komputer/D3-Teknik Komputer
Judul Tugas Akhir	: Rancang Bangun Alat Pelacak Hewan Berbasis *Internet of Things.*

No	Uraian	Paraf
1		

Palembang, Juli 2024
Dosen Penguji

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