

## LAMPIRAN



## FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS LABUHANBATU

PROGRAM STUDI :

AGROTEKNOLOGI - TEKNOLOGI INFORMASI - SISTEM INFORMASI - MANAJEMEN INFORMATIKA

Jl. SM. Raja No. 126-A KM. 3,5 Aek Tapa - Rantauprapat - Sumatera Utara - Pos 21415  
Telp./Fax. (0624) 21901

Nomor : 20/II/FST-ULB/I/2026  
Hal : Permohonan Izin Penelitian

Kepada Yth.  
**Bapak Kepala Desa Rintis**  
di -  
Tempat

Sehubungan dengan rencana Penelitian untuk Skripsi/Tugas Akhir Mahasiswa Program Studi S-1 Teknologi Informasi Fakultas Sains dan Teknologi tersebut dibawah ini :

Nama : NANDALIA DEVINA  
NPM : 2208100065  
Program Studi : T-1 Teknologi Informasi  
Judul Tugas Akhir : RANCANG BANGUN SMART LASER SECURITY SYSTEM BERBASIS IOT DENGAN NOTIFIKASI WHATSAPP  
Lokasi Penelitian : KANTOR DESA RINTIS

Untuk keperluan tersebut diatas, agar kiranya dapat memberi izin pelaksanaan penelitian di wilayah Bapak/Ibu. Dalam proses pelaksanaannya segala sesuatu yang berkaitan dengan penelitian tersebut akan diselesaikan oleh mahasiswa yang bersangkutan.

Demikian hal ini kami sampaikan atas perhatian dan bantuannya diucapkan terima kasih.

Rantauprapat, 9 Januari 2026  
Fakultas Sains dan Teknologi  
Ka. Prodi Teknologi Informasi

**Rahmadani Pane, S.Kom, M.Kom**  
NIDN. 0110058601



**PEMERINTAH KABUPATEN LABUHANBATU SELATAN**  
**KECAMATAN SILANGKITANG**  
**DESA RINTIS**

Alamat: Jalan Besar Rintis Desa Rintis No : Kode Pos 21461

SURAT KETERANGAN  
Nomor : 581/ 9 /Pem-Rts/2026

Yang bertanda tangan dibawah ini :

Nama : JURianto, S.H  
Jabatan : Sekretaris Desa Rintis

Dengan ini menerangkan bahwa :

Nama : NANDALIA DEVINA  
NPM : 2208100065  
Program Studi : T-I ( Teknologi Informasi )

Benar telah diberikan izin untuk pelaksanaan penelitian skripsi / tugas akhir di Desa Rintis Kecamatan Silangkitang Kabupaten Labuhanbatu Selatan dengan judul/tema " RANCANG BANGUN SMART LASER SECURITY SYSTEM BERBASIS IOT DENGAN NOTIFIKASI WHATSAP"

Demikian Surat Keterangan ini diperbuat dengan sebenarnya, untuk dapat digunakan sebagaimana mestinya.

Rintis, 13 Januari 2026  
A.n KEPALA DESA RINTIS  
Sekretaris Desa Rintis  
  
JURianto, S.H

```

#include <PubSubClient.h>
#include <ArduinoJson.h>

// ===== WIFI =====
const char* ssid = "SOLIN_PROJECT";
const char* password = "17AGUSTUS2025";

// ===== MQTT =====
const char* mqttServer = "broker.hivemq.com";
const int mqttPort = 1883;
const char* topicSensor = "lia/sensor";
const char* topicAlarm = "lia/alarm";
const char* topicCommand = "lia/command";
const char* topicStatus = "lia/status";

// ===== PIN =====
#define LDR1 34
#define LDR2 35
#define BUZZER 25

// ===== VARIABEL GLOBAL =====
int batasLaser = 2000;
bool sistemAktif = false;
volatile bool alarmBunyi = false;

unsigned long alarmStartTime = 0;
const unsigned long durasiAlarm = 20000; // 20 detik

WiFiClient espClient;
PubSubClient mqttClient(espClient);

TaskHandle_t TaskBuzzerHandle = NULL;
TaskHandle_t TaskSensorHandle = NULL;

// =====
// CORE 0 — TASK BUZZER (kedip non-blocking)
// =====
void TaskBuzzer(void* pvParameters) {
  for (;;) {
    if (!alarmBunyi || !sistemAktif) {
      digitalWrite(BUZZER, LOW);
      vTaskDelay(pdMS_TO_TICKS(50));
      continue;
    }

    // === WEE: 5x beep cepat (80ms ON, 60ms OFF) ===
    for (int i = 0; i < 5; i++) {
      if (!alarmBunyi || !sistemAktif) break;
      digitalWrite(BUZZER, HIGH);
      vTaskDelay(pdMS_TO_TICKS(80));
      digitalWrite(BUZZER, LOW);
      vTaskDelay(pdMS_TO_TICKS(60));
    }

    if (!alarmBunyi || !sistemAktif) continue;
    vTaskDelay(pdMS_TO_TICKS(120));

    // === WOO: 3x beep lambat (300ms ON, 200ms OFF) ===

```

```

for (int i = 0; i < 3; i++) {
  if (!alarmBunyi || !sistemAktif) break;
  digitalWrite(BUZZER, HIGH);
  vTaskDelay(pdMS_TO_TICKS(300));
  digitalWrite(BUZZER, LOW);
  vTaskDelay(pdMS_TO_TICKS(200));
}

if (!alarmBunyi || !sistemAktif) continue;
vTaskDelay(pdMS_TO_TICKS(200));
}
}

// ===== CALLBACK MQTT =====
void callback(char* topic, byte* payload, unsigned int length) {
  String msg = "";
  for (unsigned int i = 0; i < length; i++) msg += (char)payload[i];
  msg.trim();

  if (String(topic) == topicCommand) {
    if (msg == "ON") {
      sistemAktif = true;
      mqttClient.publish(topicStatus, "ON");
    } else if (msg == "OFF") {
      sistemAktif = false;
      alarmBunyi = false;
      mqttClient.publish(topicStatus, "OFF");
    }
  }
}

// ===== MQTT RECONNECT =====
void connectMQTT() {
  while (!mqttClient.connected()) {
    String clientId = "ESP32-LIA-" + String(random(0xffff), HEX);
    if (mqttClient.connect(clientId.c_str())) {
      mqttClient.subscribe(topicCommand);
      mqttClient.publish(topicStatus, sistemAktif ? "ON" : "OFF");
    } else {
      delay(3000);
    }
  }
}

// =====
// CORE 1 — TASK SENSOR + MQTT
// =====
void TaskSensor(void* pvParameters) {
  for (;;) {
    if (!mqttClient.connected()) connectMQTT();
    mqttClient.loop();

    int ldr1 = analogRead(LDR1);
    int ldr2 = analogRead(LDR2);

    // Publish data sensor
    StaticJsonDocument<128> docSensor;
    docSensor["ldr1"] = ldr1;

```

```

docSensor["ldr2"] = ldr2;
docSensor["alarm"] = alarmBunyi;
char bufSensor[128];
serializeJson(docSensor, bufSensor);
mqttClient.publish(topicSensor, bufSensor);

if (sistemAktif) {
  bool adaHalangan = (ldr1 > batasLaser || ldr2 > batasLaser);

  if (adaHalangan && !alarmBunyi) {
    alarmBunyi = true;
    alarmStartTime = millis();
    Serial.println("INTRUDER DETECTED!");

    StaticJsonDocument<128> docAlarm;
    docAlarm["type"] = "INTRUDER";
    docAlarm["ldr1"] = ldr1;
    docAlarm["ldr2"] = ldr2;
    char bufAlarm[128];
    serializeJson(docAlarm, bufAlarm);
    mqttClient.publish(topicAlarm, bufAlarm);
  }

  // Timer 20 detik — prioritas utama, halangan masih ada pun alarm tetap mati
  if (alarmBunyi && (millis() - alarmStartTime >= durasiAlarm)) {
    alarmBunyi = false;
    Serial.println("Alarm otomatis mati (20 detik)");
  }

  } else {
    alarmBunyi = false;
  }
  vTaskDelay(pdMS_TO_TICKS(200));
}
}

// ===== SETUP =====
void setup() {
  Serial.begin(115200);
  pinMode(BUZZER, OUTPUT);
  digitalWrite(BUZZER, LOW);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) delay(500);
  Serial.println("WiFi Connected!");

  mqttClient.setServer(mqttServer, mqttPort);
  mqttClient.setCallback(callback);
  connectMQTT();

  xTaskCreatePinnedToCore(TaskBuzzer, "TaskBuzzer", 2048, NULL, 1, &TaskBuzzerHandle, 0);
  xTaskCreatePinnedToCore(TaskSensor, "TaskSensor", 8192, NULL, 1, &TaskSensorHandle, 1);
}

void loop() {
  vTaskDelay(pdMS_TO_TICKS(1000));
}

```