

LAMPIRAN

Surat Penelitian



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 : 42/TI/FST-ULB/III/2026
Hal : Permohonan Izin Penelitian

Kepada Yth.
Kepala Desa Bandar Tinggi
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 : PUTRI RITONGA
NPM : 2208100074
Program Studi : T-1 Teknologi Informasi
Judul Tugas Akhir : PENERAPAN AIRTAG DENGAN MENGGUNAKAN
METODE BLUETOOTH
Lokasi Penelitian : Kantor Desa Bandar Tinggi

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, 31 Maret 2026
Fakultas Sains dan Teknologi
Kampus Teknologi Informasi



[Signature]
Rahmawati Pane, S.Kom, M.Kom
NIDN. 0110058601



PEMERINTAH KABUPATEN LABUHAN BATU
KECAMATAN BILAH HULU
DESA BANDAR TINGGI

Nomor	: 423.4/ 165 /III/PEM/2026	Bandar Tinggi, 30 Maret 2026
Sifat	: Penting.	Kepada :
Lampiran	: -	Yth, Dekan Fakultas Sains dan Teknologi
Perihal	: Persetujuan Penelitian	Universitas Labuhanbatu
		Di-
		Rantauprapat

Menindak lanjuti Surat dari Fakultas Sains dan Teknologi Universitas Labuhanbatu dengan Nomor 42/TI/FST-ULB/III/2026 pada Tanggal 30 Maret 2026 yang lalu, demi untuk meningkatkan peran dan Fungsi Mahasiswa dengan ini kami memberikan persetujuan kepada Mahasiswa dari Fakultas Sains dan Teknologi Universitas Labuhanbatu untuk melaksanakan Penelitian di Desa Bandar Tinggi Kecamatan Bilah Hulu Kabupaten Labuhanbatu sejak Bulan November 2025 s/d Selesai.

Demikian surat ini kami sampaikan atas kerjasamanya kami ucapkan terima kasih.



Dokumentasi Penelitian



Sketch Yang Digunakan

```

#defineBLYNK_TEMPLATE_ID"TMPL660oQUuOa"
#defineBLYNK_TEMPLATE_NAME"ESP32 Tracker"
#defineBLYNK_AUTH_TOKEN"v-dTgy72bx3nz4WIAHpnMEb-uy0PpzHe"

#include<WiFi.h>
#include<BlynkSimpleEsp32.h>
#include<BLEDevice.h>
#include<BLEServer.h>
#include<BLEUtils.h>
#include<BLE2902.h>

// ===== WIFI =====
char ssid[] = "Airtag";
char pass[] = "12345678";

// ===== PIN =====
#defineLED_PIN2
#defineBUZZER_PIN4

// ===== BLE UUID =====
#defineSERVICE_UUID "12345678-1234-1234-1234-1234567890ab"
#defineCHARACTERISTIC_UUID"abcd1234-5678-1234-5678-abcdef123456"

// ===== GLOBAL =====
bool deviceConnected = false;
bool lastState = false;

bool alarmTriggered = false;

```

```

bool notifSent = false;

unsignedlong disconnectTime = 0;
unsignedlong alarmDelay = 5000;

int disconnectCount = 0;
int modeAman = 1;

BLEServer* pServer;
BLEAdvertising* pAdvertising;

BlynkTimer timer;

// ===== FIND DEVICE =====
bool findActive = false;
unsignedlong findStart = 0;

// ===== BUZZER PATTERN =====
unsignedlong lastBeep = 0;
bool beepState = false;

// ===== BLE CALLBACK =====
class MyServerCallbacks: public BLEServerCallbacks {
void onConnect(BLEServer* pServer){
    deviceConnected = true;
    Serial.println("HP CONNECTED");
}

void onDisconnect(BLEServer* pServer){
    deviceConnected = false;
    disconnectTime = millis();
    alarmTriggered = false;
    Serial.println("HP DISCONNECTED");

    delay(100);
    pAdvertising->start();
}
};

// ===== BLYNK =====
BLYNK_WRITE(V1){ // Mode Aman
    modeAman = param.asInt();

    // Reset baseline supaya tidak langsung bunyi
    disconnectTime = millis();
    alarmTriggered = false;
    notifSent = false;
}

BLYNK_WRITE(V2){ // Find Device
    int find = param.asInt();

    if(find == 1){
        findActive = true;
        findStart = millis();
    }
}

```

```

}
}

// ===== STATUS UPDATE =====
void updateStatus() {

  if (deviceConnected != lastState) {

    if (deviceConnected) {

      Blynk.virtualWrite(V0, "CONNECTED");
      Blynk.virtualWrite(V4, 1);

      digitalWrite(LED_PIN, HIGH);
      digitalWrite(BUZZER_PIN, LOW);

      alarmTriggered = false;
      notifSent = false;
    }
    else {

      Blynk.virtualWrite(V0, "DISCONNECTED");
      Blynk.virtualWrite(V4, 0);

      digitalWrite(LED_PIN, LOW);

      disconnectCount++;
      Blynk.virtualWrite(V3, disconnectCount);
    }

    lastState = deviceConnected;
  }

  // ===== ALARM KEAMANAN =====
  if (!deviceConnected && modeAman == 1 && !findActive) {

    if (millis() - disconnectTime > alarmDelay) {

      if (!alarmTriggered) {
        alarmTriggered = true;

        if (!notifSent) {
          Blynk.logEvent("tracker_alert", "Perangkat Terlalu Jauh!");
          notifSent = true;
        }
      }
    }
  }
}

// ===== BUZZER CONTROL =====
void handleBuzzer() {

  // ===== FIND DEVICE MODE =====
  if (findActive) {

```

```

if(millis() - findStart >3000){
  findActive = false;
  digitalWrite(BUZZER_PIN, LOW);
  return;
}

if(millis() - lastBeep >200){
  beepState = !beepState;
  digitalWrite(BUZZER_PIN, beepState);
  lastBeep = millis();
}

return;
}

// ===== SECURITY ALARM MODE =====
if(alarmTriggered){

  if(millis() - lastBeep >300){
    beepState = !beepState;
    digitalWrite(BUZZER_PIN, beepState);
    lastBeep = millis();
  }

} else {
  digitalWrite(BUZZER_PIN, LOW);
}
}

// ===== SETUP =====
voidsetup(){

  Serial.begin(115200);
  delay(1000);

  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  digitalWrite(LED_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);

  Serial.println("=== ESP32 Tracker Start ===");

  // WIFI
  WiFi.begin(ssid, pass);
  Serial.print("Connecting WiFi");

  while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi Connected");
}

```

```

Blynk.config(BLYNK_AUTH_TOKEN);
Blynk.connect();

Serial.println("Blynk Connected");

// BLE
BLEDevice::init("ESP32-TRACKER");

pServer = BLEDevice::createServer();
pServer->setCallbacks(new MyServerCallbacks());

BLEService *pService = pServer->createService(SERVICE_UUID);

BLECharacteristic *pCharacteristic = pService->createCharacteristic(
    CHARACTERISTIC_UUID,
    BLECharacteristic::PROPERTY_READ |
    BLECharacteristic::PROPERTY_NOTIFY
);

pCharacteristic->addDescriptor(new BLE2902());
pCharacteristic->setValue("READY");

pService->start();

pAdvertising = BLEDevice::getAdvertising();
pAdvertising->addServiceUUID(SERVICE_UUID);
pAdvertising->setScanResponse(true);
pAdvertising->start();

Serial.println("BLE Advertising Started");
Serial.println("=== SYSTEM READY ===");
}

// ===== LOOP =====
void loop(){

    if(WiFi.status() == WL_CONNECTED){
        Blynk.run();
    }

    timer.run();

    updateStatus();
    handleBuzzer();
}

```