



LAMPIRAN

Surat Permohonan Izin 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
<hr/>	
Nomor	: 17 /TI/FST-ULB/XII/2025
Hal	: Permohonan Izin Penelitian
Kepada Yth. Bapak/Ibu Kepala Desa Lingga Tiga di - Tempat	
Sehubung dengan rencana Penelitian untuk Skripsi/Tugas Akhir Mahasiswa Program Studi S-1 Teknologi Informasi Fakultas Sains dan Teknologi tersebut dibawah ini :	
Nama	: FRIDA TRYSANTI
NPM	: 2208100026
Program Studi	: T-1 Teknologi Informasi
Judul Tugas Akhir	: RANCANG BANGUN SISTEM PEMANTAUAN WAKTU PELAYANAN DI KANTOR DESA LINGGA TIGA BERBASIS IOT
Lokasi Penelitian	: Lingga Tiga I, Lingga Tiga, Bilah Hulu, Labuhanbatu, Sumatera Utara 21451, N Empat Aek Nabara, Kec. Bilah Hulu, Kab. Labuhanbatu.
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, 30 Desember 2025 Fakultas Sains dan Teknologi Ka. Prodi Teknologi Informasi	
	
Rahmadani Pane, S.Kom, M.Kom NIDN. 0110058601	

Surat Balasan Izin Penelitian



PEMERINTAH KABUPATEN LABUHANBATU
KECAMATAN BILAH HULU
DESA LINGGA TIGA

Jalan Besar Lingga Tiga, Kode Pos. 21451

Lingga Tiga, 31 Desember 2025

Nomor : 145/ / Pem/ 2025
Sifat : Penting
Lamp : -
Hal : Izin Penelitian

Yth, Kepala Prodi Teknologi Informasi Universitas Labuhanbatu

di_ Tempat

Dengan hormat,

Sehubungan dengan Surat dari Fakultas Sains dan Teknologi Universitas Labuhanbatu Nomor : 17/TI/FST-ULB/XII/2025 tanggal 30 Desember 2025 Perihal Permohonan Izin Penelitian.

Maka dengan ini kami atas nama Pemerintah Desa Lingga Tiga memberikan Izin kepada :

Nama : FRIDA TRYANTI.
NPM : 2208100026.

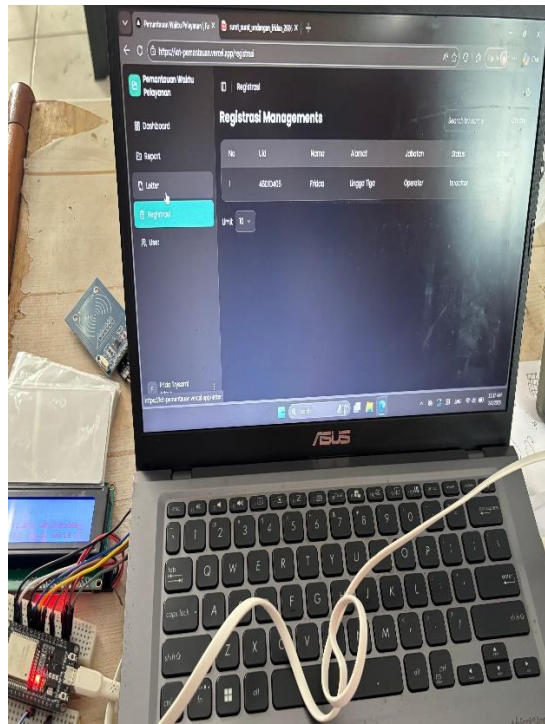
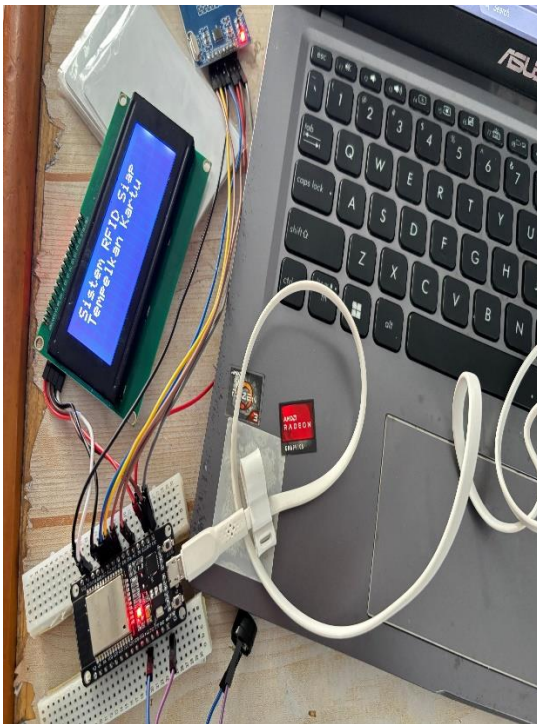
Untuk melaksanakan kegiatan Penelitian dengan Judul Tugas Akhir Rancang Bangun Sistem Pemantauan waktu pelayanan di Kantor Desa Lingga Tiga berbasis IOT.

Demikian Surat ini kami sampaikan, atas perhatiannya kami ucapkan terima kasih.

An. KEPALA DESA LINGGA TIGA,
Sekretaris Desa



Dokumentasi Penelitian



Code Program Arduino IDE

```
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <WiFiClientSecure.h>
#include <hd44780.h>
#include <hd44780ioClass/hd44780_I2Cexp.h>

#include <time.h>
#include <ctype.h>

#define SS_PIN 5
#define RST_PIN 4
#define BUZZER_PIN 12

const char* ssid = "Testing";
const char* password = "12345678";

String apiURL = "https://.../api/data";

const long  gmtOffset_sec      = 7 * 3600;
const int   daylightOffset_sec = 0;
const char* ntp1 = "pool.ntp.org";
const char* ntp2 = "time.nist.gov";

MFRC522 mfrc522(SS_PIN, RST_PIN);
hd44780_I2Cexp lcd;

bool buzzerAttached = false;

void buzzerInit() {
    buzzerAttached = ledcAttach(BUZZER_PIN, 2000, 8);
}

void buzzerOn(int freq) {
    if (!buzzerAttached) return;
    ledcWriteTone(BUZZER_PIN, freq);
    ledcWrite(BUZZER_PIN, 128);
}

void buzzerOff() {
    if (!buzzerAttached) return;
    ledcWrite(BUZZER_PIN, 0);
}

void beepOK() {
    buzzerOn(1200); delay(120); buzzerOff(); delay(80);
    buzzerOn(1200); delay(120); buzzerOff();
}

void beepERR() {
    buzzerOn(600); delay(200); buzzerOff(); delay(120);
}
```

```

    buzzerOn(600); delay(200); buzzerOff();
}

void showReady() {
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Sistem RFID Siap");
    lcd.setCursor(1, 1);
    lcd.print("Tempelkan Kartu");
}

String urlEncode(const String& s) {
    String out; out.reserve(s.length() * 3);
    const char *hex = "0123456789ABCDEF";

    for (size_t i = 0; i < s.length(); i++) {
        unsigned char c = (unsigned char)s[i];
        if (isalnum(c) || c=='-' || c=='_' || c=='.' || c=='~') {
            out += (char)c;
        } else {
            out += '%';
            out += hex[(c >> 4) & 0x0F];
            out += hex[c & 0x0F];
        }
    }
    return out;
}

void syncTimeNTP() {
    configTime(gmtOffset_sec, daylightOffset_sec, ntp1, ntp2);

    struct tm timeinfo;
    unsigned long start = millis();
    while (!getLocalTime(&timeinfo) && (millis() - start) < 10000) {
        delay(200);
    }
}

String getTimeISO() {
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) return "";

    char buf[25];
    strftime(buf, sizeof(buf), "%Y-%m-%dT%H:%M:%S", &timeinfo);
    return String(buf);
}

long getEpoch() {
    time_t now = time(nullptr);
    if (now < 1700000000) return -1;
    return (long)now;
}

String pad2(int v) { return (v < 10) ? "0" + String(v) : String(v); }
String toJakartaHHMMSS(const String& ts) {
    int tPos = ts.indexOf('T');
    if (tPos < 0) tPos = ts.indexOf(' ');
}

```

```

if (tPos < 0 || ts.length() < tPos + 9) return ts;

int hour = ts.substring(tPos + 1, tPos + 3).toInt();
String mm = ts.substring(tPos + 4, tPos + 6);
String ss = ts.substring(tPos + 7, tPos + 9);

int offsetH = 0;
if (ts.endsWith("Z")) {
    offsetH = 0;
} else {
    int plus = ts.lastIndexOf('+');
    int minus = ts.lastIndexOf('-');
    int signPos = (plus > minus) ? plus : minus;

    if (signPos > tPos) {
        int sign = (signPos == minus) ? -1 : 1;
        String off = ts.substring(signPos + 1);
        int oh = off.substring(0, 2).toInt();
        offsetH = sign * oh;
    }
}

int add = 7 - offsetH;
int h = (hour + add) % 24;
if (h < 0) h += 24;

return pad2(h) + ":" + mm + ":" + ss;
}

void connectWiFi() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Menghubungkan WiFi");

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

    int dots = 0;
    unsigned long lastUpdate = 0;
    unsigned long startAttempt = millis();

    while (WiFi.status() != WL_CONNECTED) {
        if (millis() - lastUpdate > 400) {
            lastUpdate = millis();
            lcd.setCursor(0, 1);
            lcd.print("Menghubungkan");
            lcd.setCursor(13, 1);
            lcd.print(" ");
            lcd.setCursor(13, 1);
            for (int i = 0; i < dots; i++) lcd.print(".");
            dots = (dots + 1) % 4;
        }

        if (millis() - startAttempt > 20000) {
            WiFi.disconnect(true);
            delay(300);
            WiFi.begin(ssid, password);
        }
    }
}

```

```

        startAttempt = millis();
    }
    delay(10);
}

Serial.println("WiFi Terhubung!");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("WiFi Terhubung");
lcd.setCursor(0, 1);
lcd.print(WiFi.localIP());
delay(1200);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sinkron Waktu...");
syncTimeNTP();
delay(500);
}

String getUIDString() {
    String uid = "";
    uid.reserve(mfrc522.uid.size * 2);

    for (byte i = 0; i < mfrc522.uid.size; i++) {
        if (mfrc522.uid.uidByte[i] < 0x10) uid += "0";
        uid += String(mfrc522.uid.uidByte[i], HEX);
    }

    uid.toUpperCase();
    return uid;
}

String hhmmsFromISO(const String& iso) {
    if (iso.length() >= 19) return iso.substring(11, 19);
    if (iso.length() >= 16) return iso.substring(11, 16);
    return iso;
}

void sendToAPI(String uid);

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

    SPI.begin();
    mfrc522.PCD_Init();

    pinMode(BUZZER_PIN, OUTPUT);
    buzzerInit();
    buzzerOff();

    lcd.begin(20, 4);
    lcd.clear();
    lcd.print("Booting...");

    connectWiFi();
    showReady();
}

```

```

}

void loop() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi Putus, reconnect...");
    connectWiFi();
    showReady();
  }

  if (!mfrc522.PICC_IsNewCardPresent()) return;
  if (!mfrc522.PICC_ReadCardSerial()) return;

  String uid = getUIDString();

  Serial.print("UID: ");
  Serial.println(uid);

  lcd.clear();
  lcd.setCursor(1, 0);
  lcd.print("Terdeteksi UID");
  lcd.setCursor(1, 1);
  lcd.print(uid);

  sendToAPI(uid);

  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();

  delay(800);
}

void sendToAPI(String uid) {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi Hilang!");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WiFi Hilang!");
    lcd.setCursor(0, 1);
    lcd.print("Coba Reconnect");
    beepERR();
    return;
  }

  String waktuISO = getTimeISO();
  long epoch = getEpoch();

  if (waktuISO == "" || epoch < 0) {
    syncTimeNTP();
    waktuISO = getTimeISO();
    epoch = getEpoch();
  }

  WiFiClientSecure client;
  client.setInsecure();

  HTTPClient http;
  http.setTimeout(8000);
}

```

```

String endpoint = apiURL + "?uid=" + uid;
if (waktuISO != "") endpoint += "&waktu=" + urlEncode(waktuISO);
if (epoch > 0) endpoint += "&epoch=" + String(epoch);

Serial.println("GET: " + endpoint);
http.begin(client, endpoint);

int code = http.GET();
Serial.println("HTTP Response : " + String(code));

if (code > 0) {
    String res = http.getString();
    Serial.println("Response:");
    Serial.println(res);

    StaticJsonDocument<1024> doc;
    DeserializationError error = deserializeJson(doc, res);

    if (error) {
        Serial.println("Kesalahan JSON!");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("JSON Error");
        lcd.setCursor(0, 1);
        lcd.print("Cek Response");
        beepERR();
        http.end();
        delay(1500);
        showReady();
        return;
    }

    String nama = doc["nama"] | "";
    String alamat = doc["alamat"] | "";
    String jabatan = doc["jabatan"] | "";
    String status = doc["status"] | "";
    String message = doc["message"] | "";
    String lastStart = doc["last_start_service"] | "";
    String startServer = doc["start_service"] | "";

    bool isBusy = (code == 409);
    if (!isBusy && message.length() > 0) {
        String msgLower = message;
        msgLower.toLowerCase();
        if (msgLower.indexOf("proses") >= 0 ||
            msgLower.indexOf("berjalan") >= 0 ||
            msgLower.indexOf("tunggu") >= 0) {
            isBusy = true;
        }
    }
}

if (isBusy) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SCAN DITOLAK!");
    lcd.setCursor(0, 1);
}

```

```

        lcd.print("Proses masih jalan");

        String t = lastStart.length() ? lastStart : startServer;
        lcd.setCursor(0, 2);
        if (t.length()) lcd.print("Mulai: " + toJakartaHHMMSS(t));
        else             lcd.print("Tunggu selesai dulu");

        lcd.setCursor(0, 3);
        lcd.print("Coba scan lagi");

        beepERR();
        http.end();
        delay(2500);
        showReady();
        return;
    }

    lcd.clear();

    if (status == "terdaftar") {
        lcd.setCursor(0, 0);
        lcd.print("Nama   : " + nama);
        lcd.setCursor(0, 1);
        lcd.print("Alamat : " + alamat);
        lcd.setCursor(0, 2);
        lcd.print("Jabatan: " + jabatan);

        lcd.setCursor(0, 3);
        String mulai = startServer.length() ? startServer : waktuISO;
        if (mulai.length()) lcd.print("Mulai: " + toJakartaHHMMSS(mulai));
        else lcd.print("Pelayanan Dimulai");

        beepOK();
    } else {
        lcd.setCursor(1, 0);
        lcd.print("Kartu Tidak");
        lcd.setCursor(1, 1);
        lcd.print("Terdaftar");
        lcd.setCursor(1, 2);
        lcd.print("UID:");
        lcd.setCursor(5, 2);
        lcd.print(uid);

        beepERR();
    }

    } else {
        Serial.println("Permintaan Gagal!");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("HTTP Gagal");
        lcd.setCursor(0, 1);
        lcd.print("Code: " + String(code));
        beepERR();
    }
}

http.end();

```

```
    delay(2000);  
    showReady();  
}
```