

LAMPIRAN

Lampiran 1 Surat Kesiediaan Membimbing TA Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Ida Afriliana, S.T,M.Kom
NIPY : 12.013.168
Jabatan Struktural : KA. Prodi DIII Teknik Komputer
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Didik Khaeri Pambudi
NIM : 22040083
Program Studi : Diploma III Teknik Komputer

Judul TA : PENGEMBANGAN TEMPAT SAMPAH PINTAR BERBASIS IOT
UNTUK MONITORING VOLUME SAMPAH SECARA REAL TIME

Dengan ini menyatakan ini dibuat agar dilaksanakan sebagaimana mestinya

Tegal, 5 Februari 2025

Mengetahui

Ka Prodi DIII Teknik Komputer,



Ida Afriliana, ST, M.Kom
NIPY. 12.013.168

Dosen Pembimbing I,

Ida Afriliana, ST, M.Kom
NIPY. 12.013.168

Lampiran 2 Surat Kesiediaan Membimbing TA Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Abdul Basit, S.Kom, M.T
NIPY : 01.015.198
Jabatan Struktural : Sekretaris Prodi DIII Teknik Komputer
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Didik Khaeri Pambudi
NIM : 22040083
Program Studi : Diploma III Teknik Komputer

Judul TA : PENGEMBANGAN TEMPAT SAMPAH PINTAR BERBASIS IOT
UNTUK MONITORING VOLUME SAMPAH SECARA REAL TIME

Dengan ini menyatakan ini dibuat agar dilaksanakan sebagaimana mestinya

Tegal, 5 Februari 2025

Mengetahui

Ka Prodi DIII Teknik Komputer,



Miliana, ST, M.Kom
NIPY. 12.013.168

Dosen Pembimbing II,

Abdul Basit, S.Kom, M.T
NIPY. 01.015.198

Lampiran 3 Source Code *LoRa Node*

```
#include <SPI.h>
#include <LoRa.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>

// Pin Ultrasonik 1 (deteksi tinggi sampah)
#define TRIG1_PIN 3
#define ECHO1_PIN 4

// Pin Ultrasonik 2 (akses servo)
#define TRIG2_PIN 5
#define ECHO2_PIN 6

#define SERVO_PIN 7

// Pin LoRa
#define LORA_SS 10
#define LORA_RST 9
#define LORA_DIO0 8

#define FREQUENCY 433E6
#define MAX_TONG 50.0
const String NODE_ID_RAW = "Gedung B ";
String NODE_ID = "";

Servo myServo;
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Variabel waktu & status
unsigned long waktuTerakhirKirim = 0;
const unsigned long INTERVAL_KIRIM_DATA = 5000; // 5 detik

bool penuh = false;
bool servoSedangTerbuka = false;
unsigned long waktuServoBuka = 0;
const unsigned long DURASI_BUKA = 3000;

void setup() {
    Serial.begin(9600);

    // Bersihkan spasi dengan trim()
    NODE_ID = NODE_ID_RAW;
```

```

NODE_ID.trim();

pinMode(TRIG1_PIN, OUTPUT);
pinMode(ECHO1_PIN, INPUT);
pinMode(TRIG2_PIN, OUTPUT);
pinMode(ECHO2_PIN, INPUT);

myServo.attach(SERVO_PIN);
myServo.write(180); // posisi awal: tertutup

lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Node aktif...");

LoRa.setPins(LORA_SS, LORA_RST, LORA_DIO0);
if (!LoRa.begin(FREQUENCY)) {
    lcd.setCursor(0, 1);
    lcd.print("LoRa gagal...");
    while (true);
}

LoRa.setSpreadingFactor(7);
LoRa.setSignalBandwidth(125E3);
LoRa.setCodingRate4(5);

lcd.setCursor(0, 1);
lcd.print("LoRa Siap!");
delay(2000);
lcd.clear();
}

float bacaJarak(int trig, int echo) {
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);

    long durasi = pulseIn(echo, HIGH, 30000); // timeout 30ms
    if (durasi == 0 || durasi >= 30000) return -1;
    return durasi * 0.034 / 2.0;
}

void loop() {

```

```

unsigned long sekarang = millis();

if (sekarang - waktuTerakhirKirim >= INTERVAL_KIRIM_DATA) {
    waktuTerakhirKirim = sekarang;
    kirimData();
}

kontrolServo(sekarang);
}

void kirimData() {
    float jarak = bacaJarak(TRIG1_PIN, ECHO1_PIN);
    float tinggi = (jarak > 0 && jarak < MAX_TONG) ? (MAX_TONG -
jarak) : 0;
    tinggi = constrain(tinggi, 0, MAX_TONG);

    String status;
    if (tinggi >= 46) {
        status = "PENUH";
        penuh = true;
    } else {
        status = (tinggi > 35) ? "SEDANG" : "AMAN";
        penuh = false;
    }

    // Format parsing: Gedung A#20#AMAN#Kirim
    String data = NODE_ID + "#" + String((float)tinggi) + "#" +
status + "#Kirim";

    if (LoRa.beginPacket()) {
        LoRa.print(data);
        LoRa.endPacket();
        Serial.println("Data dikirim: " + data);
    } else {
        Serial.println("Gagal mulai kirim LoRa.");
    }

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Tinggi: " + String((float)tinggi) + "cm");
    lcd.setCursor(0, 1);
    lcd.print("Status: " + status);
}

void kontrolServo(unsigned long waktuSekarang) {

```

```
float jarak = bacaJarak(TRIG2_PIN, ECHO2_PIN);

if (!penuh && jarak > 0 && jarak < 10 && !servoSedangTerbuka) {
    myServo.write(0); // buka
    servoSedangTerbuka = true;
    waktuServoBuka = waktuSekarang;
}

if (servoSedangTerbuka && (waktuSekarang - waktuServoBuka >=
DURASI_BUKA)) {
    myServo.write(180); // tutup
    servoSedangTerbuka = false;
}
}
```

Lampiran 4 Source Code LoRa Gateway

```
#include <SPI.h>
#include <LoRa.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

// WiFi
const char* ssid = "PoltekHarber_Lab3";
const char* password = "poltekharber";

// Alamat server lokal (bisa juga IP lokal)
//const String serverName =
"http://192.168.18.13/monitoring/simpan_data.php";
const String serverName =
"http://bujunk.dheimuts.web.id/monitoring/simpan_data.php";

// Pin LoRa ESP8266
#define SS    15 // D8
#define RST    16 // D0
#define DI00    2 // D4

void setup() {
    Serial.begin(115200);
    delay(100);

    // Koneksi WiFi
    WiFi.begin(ssid, password);
    Serial.print("🔌 Menghubungkan ke WiFi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\n✅ WiFi Terhubung!");

    // Inisialisasi LoRa
    LoRa.setPins(SS, RST, DI00);
    if (!LoRa.begin(433E6)) {
        Serial.println("❌ Gagal memulai LoRa!");
        while (1);
    }

    LoRa.setSpreadingFactor(7);
    LoRa.setSignalBandwidth(125E3);
```

```

LoRa.setCodingRate4(5);

Serial.println("✅ Gateway Siap Terima Data LoRa!");
Serial.println("-----");
}

void loop() {
    int packetSize = LoRa.parsePacket();
    if (packetSize) {
        String incoming = "";
        while (LoRa.available()) {
            incoming += (char)LoRa.read();
        }
        incoming.trim();

        Serial.println("📧 Data Masuk: " + incoming);

        // Format: Gedung A#20#AMAN#Kirim
        int p1 = incoming.indexOf('#');
        int p2 = incoming.indexOf('#', p1 + 1);
        int p3 = incoming.indexOf('#', p2 + 1);

        if (p1 > 0 && p2 > p1 && p3 > p2) {
            String nama_tempat = incoming.substring(0, p1);
            String tinggi = incoming.substring(p1 + 1, p2);
            String status = incoming.substring(p2 + 1, p3);
            String perintah = incoming.substring(p3 + 1);

            Serial.println("📍 Tempat : " + nama_tempat);
            Serial.println("📏 Tinggi : " + tinggi + " cm");
            Serial.println("🔒 Status : " + status);
            Serial.println("📜 Perintah: " + perintah);

            if (perintah == "Kirim") {
                kirimKeServer(nama_tempat, tinggi, status);
            }

            Serial.println("-----");
        } else {
            Serial.println("⚠️ Format data salah!");
        }
    }
}

```

```

void kirimKeServer(String nama_tempat, String tinggi, String
status) {
    if (WiFi.status() == WL_CONNECTED) {
        WiFiClient client;
        HTTPClient http;

        http.begin(client, serverName); // serverName hanya berisi
        URL tanpa parameter

        http.addHeader("Content-Type", "application/x-www-form-
        urlencoded");

        // Format data POST
        String postData = "nama_tempat=" + nama_tempat +
            "&ketinggian=" + tinggi +
            "&status=" + status;

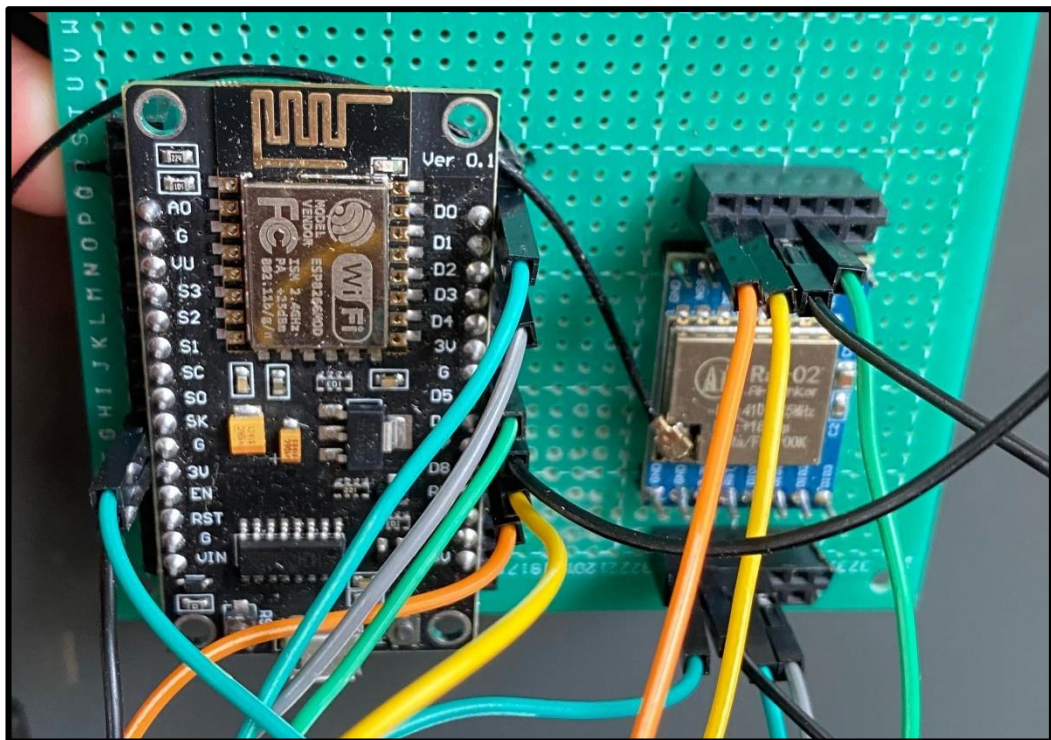
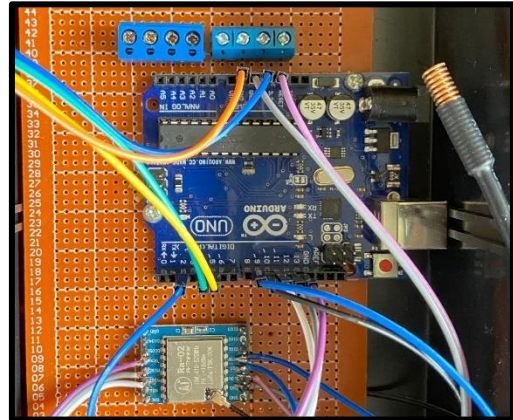
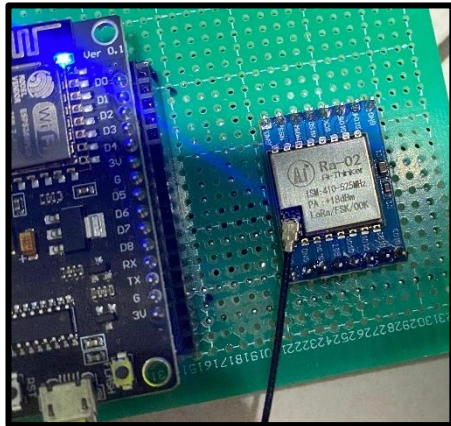
        int httpCode = http.POST(postData);

        if (httpCode > 0) {
            Serial.println("📡 HTTP Response: " + String(httpCode));
            String payload = http.getString();
            Serial.println("📄 Balasan Server: " + payload);
        } else {
            Serial.println("❌ Gagal kirim data: " +
            http.errorToString(httpCode));
        }

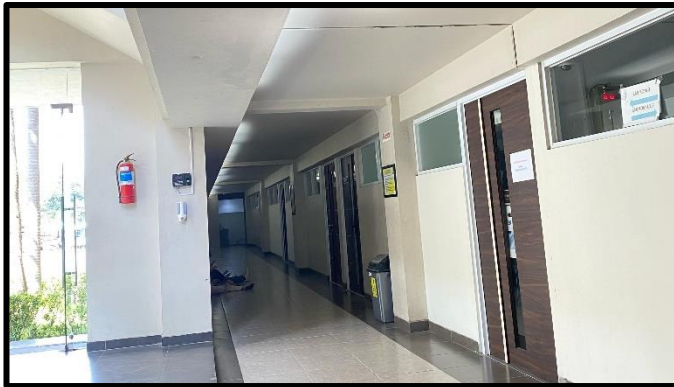
        http.end();
    } else {
        Serial.println("❌ WiFi belum tersambung!");
    }
}

```

Lampiran 5 Dokumentasi Rangkaian Alat Sistem



Lampiran 6 Dokumentasi Observasi



Manual Book Penggunaan Alat **Tempat Sampah Pintar Berbasis *IoT* untuk Monitoring Volume Sampah Secara *Real-Time***

1. Informasi Umum

- **Nama Produk:** BU-JUNK – Tempat Sampah Pintar *IoT*
- **Versi:** v1.0
- **Pengembang:** Didik Khaeri Pambudi
- **Tujuan:** Untuk memantau volume sampah secara *real-time* dan otomatis membuka-tutup tempat sampah saat objek terdeteksi.

2. Komponen Utama

1. Sensor Ultrasonik HC-SR04
2. Arduino UNO (LoRa Node)
3. Modul LoRa SX1278
4. NodeMCU ESP8266 (LoRa Gateway)
5. LCD I2C 20x4
6. *Motor Servo* (untuk membuka/tutup otomatis)
7. *Toolbox* (tempat rangkaian sistem)
8. Sumber daya: Adaptor 12V & Baterai *Li-Ion*

3. Cara Pemasangan Alat

a. Tempat Sampah:

- Letakkan sensor ultrasonik pada bagian atas dalam tempat sampah (menghadap permukaan sampah).
- Pasang *Motor Servo* pada bagian penutup atas.

b. Rangkaian Node:

- Hubungkan sensor dan servo ke Arduino UNO.
- Sambungkan modul *LoRa* ke *Arduino* sesuai pin.

c. Gateway:

- Pasang NodeMCU + LoRa Receiver + baterai *Li-ion* di lokasi yang memiliki sinyal *WiFi* stabil.
- Pastikan koneksi ke internet aktif.

d. LCD:

- LCD I2C dipasang di sisi depan tempat sampah (eksternal), untuk menampilkan status volume.

4. Cara Penggunaan Alat

1. Nyalakan alat:

- Sambungkan adaptor ke *Arduino*, dan nyalakan power pada *Gateway*.

2. Deteksi otomatis:

- Saat seseorang mendekat, sensor mendeteksi objek → *servo* membuka penutup → siap menerima sampah.

3. Pemantauan volume:

- Sensor membaca jarak permukaan sampah ke sensor:
 - < 15 cm → **PENUH**
 - <50 & > 15 cm → **SEDANG**
 - > 50 cm → **AMAN**

4. Pengiriman data:

- Data jarak dikirim via *LoRa* ke *Gateway* → diteruskan ke Website Monitoring secara *real-time*.

5. LCD Output:

- Menampilkan status sampah: “Volume: 12 cm – AMAN”

5. Indikasi dan Notifikasi

Status	Jarak ke Sampah	Indikator di LCD
AMAN	> 50 cm	“Volume AMAN”
SEDANG	< 50 & > 15 cm	“Volume SEDANG”
PENUH	< 15 cm	“Sampah PENUH”

6. Perawatan dan Pemeliharaan

- Bersihkan bagian sensor secara berkala agar tidak terhalang kotoran.
- Pastikan kabel dan koneksi tidak longgar.
- Lakukan pengecekan baterai secara rutin (pada bagian Gateway).
- Pastikan koneksi internet Gateway aktif untuk pengiriman ke website.

7. Catatan & Peringatan

- Jangan menyiram air ke arah sensor atau motor servo.
- Hindari benturan keras pada bagian atas tempat sampah.
- Pastikan tidak ada penghalang di jalur buka tutup servo.

8. Kontak & Dukungan

- Untuk perbaikan, update firmware, atau troubleshooting, hubungi:

Didik Khaeri Pambudi

Email: didiksolen@gmail.com

Instagram: [@didiksolen](https://www.instagram.com/didiksolen)



PERHATIAN

PENGGUNA TEMPAT SAMPAH PINTAR

**JANGAN BUKA TUTUP TEMPAT
SAMPAH SECARA MANUAL!**



Tutup tempat sampah ini dilengkapi dengan motor servo otomatis. Membuka secara paksa dapat merusak sistem mekanik.

CARA PENGGUNAAN YANG BENAR:



Dekatkan tangan atau sampah Anda ke arah sensor ultrasonik di bagian depan.

Tutup akan terbuka secara otomatis saat sensor mendeteksi keberadaan Anda. Setelah terbuka, silakan buang sampah dengan aman.

JIKA ALAT TIDAK BERFUNGSI:

Hubungi teknisi atau petugas kebersihan — jangan membuka paksa!