

LAMPIRAN DOKUMENTASI







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 : 140/TI/FST-ULB/VIII/2026
Hal : Permohonan Izin Penelitian

Kepada Yth.
IRWAN TAN
Kecamatan Rantau Utara Kabupaten Labuhanbatu
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 : HARDIANUR SAKINA
NPM : 2208100031
Program Studi : T-1 Teknologi Informasi
Judul Tugas Akhir : RANCANG BANGUN MESIN PENGAYAKAN PASIR
BERBASIS IOT
Lokasi Penelitian : TOKO PASIR IRWAN TAN

Untuk keperluan tersebut diatas, agar kiranya dapat memberi izin pelaksanaan penelitian di wilayah Bapak/Ibu. Dalam proses pelaksanaanya 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, 11 Agustus 2026
Fakultas Sains dan Teknologi
Ka. Prodi Teknologi Informasi

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

LAMPIRAN CODINGAN

```
#define BLYNK_TEMPLATE_ID "TMPL603nr61r7"
#define BLYNK_TEMPLATE_NAME "Pengayakan Pasir"
#define BLYNK_AUTH_TOKEN "rwITSWZ5FwUBfdMEjvWmkgk7IFNaq3Lp"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h> #include <HX711.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

char ssid[] = "QWER";
char pass[] = "qwer12345";

// Pin ESP32
#define PIN_HX711_DT      32
    #define PIN_HX711_SCK  33
    #define PIN_SW420      15
    #define PIN_RELAY      13
    #define PIN_BUZZER     2
    #define PIN_BUTTON     14

// LCD 16x2 I2C, ubah alamat 0x27 ke 0x3F jika LCD tidak tampil.
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Frekuensi Buzzer pasif.
const uint16_t BUZZER_FREQ = 4000;

HX711 scale;
BlynkTimer timer;

float calibration_factor = -7050;          // Sesuaikan saat kalibrasi
load cell.

// Relay dikendalikan HANYA oleh tombol manual atau tombol Blynk.
// Sensor berat dan sensor getaran hanya digunakan untuk monitoring.
bool motorStatus = false;
bool gangguan = false;

unsigned long lastButtonTime = 0; bool lastButtonPressed = false;
unsigned long motorStartTime = 0; unsigned long noVibrationStart = 0;
unsigned long BuzzerUntil = 0;
```

```

// Sesuaikan dengan jenis Relay dan sensor yang digunakan.
bool RelayAktifLow = true;
bool sensorGetarAktifHigh = true;
bool buttonAktifHigh = true; // Modul tombol 3 pin: ubah ke false
jika OUT = LOW saat ditekan.

String statusSistem = "Sistem siap";

void setMotor(bool state) {
    motorStatus = state;

    if (RelayAktifLow) {
        digitalWrite(PIN_RELAY, state ? LOW : HIGH);
    } else {
        digitalWrite(PIN_RELAY, state ? HIGH : LOW);
    }

    if (state) {
        motorStartTime = millis();
        noVibrationStart = 0;
        gangguan = false;
        statusSistem = "Motor ON - Mengayak";
    } else {
        noVibrationStart = 0;
        gangguan = false;
        statusSistem = "Motor OFF";
    }

    // Tampilkan perubahan Relay ke Serial Monitor.
    Serial.print("[RELAY] Motor: ");
    Serial.print(motorStatus ? "ON" : "OFF");
    Serial.print(" | Status: ");
    Serial.println(statusSistem);

    // Sinkronkan indikator dan tombol Blynk dengan kondisi Relay
    sebenarnya.
    Blynk.virtualWrite(V1, motorStatus ? 1 : 0);
    Blynk.virtualWrite(V3, motorStatus ? 1 : 0);
}

void bunyiBuzzer(int durasiMs) {
    tone(PIN_BUZZER, BUZZER_FREQ);
    BuzzerUntil = millis() + durasiMs;
}

void updateBuzzer() {

```

```

    if (BuzzerUntil > 0 && millis() >= BuzzerUntil) {
        noTone(PIN_BUZZER);
        BuzzerUntil = 0;
    }
}

float bacaBerat() {
    if (scale.is_ready()) {
        float berat = scale.get_units(5);

        if (berat < 0.03 && berat > -0.03) {
            berat = 0;
        }

        return berat;
    }

    return 0;
}

bool bacaGetaran() {
    int nilai = digitalRead(PIN_SW420);

    if (sensorGetarAktifHigh) {
        return nilai == HIGH;
    }

    return nilai == LOW;
}

// START mesin sekarang langsung mengaktifkan Relay.
// Tidak ada pengecekan berat atau sensor lain sebelum Relay ON.
void mulaiMesin() {
    setMotor(true);
    statusSistem = "Motor ON - Mengayak";
    bunyiBuzzer(300);

    Blynk.virtualWrite(V4, statusSistem);
}

void stopMesin(String alasan) {
    setMotor(false);
    statusSistem = alasan;
    bunyiBuzzer(700);

    Blynk.virtualWrite(V4, statusSistem);
}

```

```

void cekTombolManual() {
    int nilaiOut = digitalRead(PIN_BUTTON);

    // Modul tombol 3 pin memiliki VCC, GND, dan OUT.
    // OUT dibaca sebagai sinyal digital, jadi tidak memakai
    INPUT_PULLUP.
    bool tombolDitekan = buttonAktifHigh ? (nilaiOut == HIGH) :
(nilaiOut == LOW);

    // Jalankan aksi hanya saat tombol baru ditekan (edge detection).
    // Saat motor OFF -> tombol ditekan -> Relay langsung ON.
    // Saat motor ON -> tombol ditekan -> Relay langsung OFF.
    if (tombolDitekan && !lastButtonPressed && millis() -
lastButtonTime > 200) {
        lastButtonTime = millis();

        if (motorStatus) {
            stopMesin("Dimatikan manual");
        } else {
            mulaiMesin();
        }
    }

    lastButtonPressed = tombolDitekan;
}

// Sensor getaran HANYA untuk monitoring.
// Fungsi ini tidak pernah mematikan Relay/motor.
void cekMonitoringGetaran() {
    bool getaran = bacaGetaran();

    if (!motorStatus) {
        noVibrationStart = 0;
        gangguan = false;
        return;
    }

    // Beri waktu motor untuk mulai berputar sebelum status getaran
    dinilai.
    if (millis() - motorStartTime <= 2000) {
        return;
    }

    if (!getaran) {
        if (noVibrationStart == 0) {
            noVibrationStart = millis();

```

```

    }

    // Jika 3 detik tidak ada getaran, hanya tampilkan alarm
    monitoring.
    if (millis() - noVibrationStart > 3000) {
        if (!gangguan) {
            gangguan = true;
            statusSistem = "Motor ON - Cek getar";
            Blynk.logEvent("getaran_tidak_normal", "Motor ON tetapi
getaran tidak terdeteksi");
        }
    }
    else {
        noVibrationStart = 0;

        if (gangguan) {
            gangguan = false;
            statusSistem = "Motor ON - Mengayak";
        }
    }
}

void kirimDataKeBlynk() {
    float berat = bacaBerat();
    bool getaran = bacaGetaran();

    // PENTING:
    // Berat dan getaran hanya dikirim sebagai data monitoring.
    // Tidak ada kondisi sensor yang mematikan atau menghidupkan Relay
    di sini.
    Blynk.virtualWrite(V0, berat);
    Blynk.virtualWrite(V1, motorStatus ? 1 : 0);
    Blynk.virtualWrite(V2, getaran ? 1 : 0);
    Blynk.virtualWrite(V4, statusSistem);
    Blynk.virtualWrite(V5, gangguan ? 1 : 0);

    // Tampilkan data monitoring ke Serial Monitor setiap 1 detik.
    Serial.print("Berat: ");
    Serial.print(berat, 2);
    Serial.print(" kg | Motor: ");
    Serial.print(motorStatus ? "ON" : "OFF");
    Serial.print(" | Getaran: ");
    Serial.print(getaran ? "TERDETEKSI" : "TIDAK");
    Serial.print(" | Gangguan: ");
    Serial.print(gangguan ? "YA" : "TIDAK");
    Serial.print(" | Status: ");
    Serial.println(statusSistem);
}

```



```

// Tampilan LCD 16x2. Setiap baris dibatasi maksimal 16 karakter.
char baris1[17];
char baris2[17];

snprintf(baris1, sizeof(baris1), "Berat:%6.2fkg", berat);
snprintf(baris2, sizeof(baris2), "M:%s G:%s",
    motorStatus ? "ON " : "OFF",
    getaran ? "OK" : "NO");

lcd.setCursor(0, 0);
lcd.print("                ");
lcd.setCursor(0, 0);
lcd.print(baris1);

lcd.setCursor(0, 1);
lcd.print("                ");
lcd.setCursor(0, 1);
lcd.print(baris2);
}

// Tombol Blynk V3 juga langsung mengendalikan Relay tanpa syarat sensor.
BLYNK_WRITE(V3) {
    int nilai = param.asInt();

    if (nilai == 1) {
        mulaiMesin();
    } else {
        stopMesin("Dimatikan Blynk");
    }
}

BLYNK_CONNECTED() {
    Blynk.syncVirtual(V3);
}

void setup() {
    Serial.begin(115200);
    delay(500);
    Serial.println();
    Serial.println("=====");
    Serial.println("MESIN PENGAYAK PASIR - START");
    Serial.println("Serial Monitor baud rate: 115200");
    Serial.println("=====");

    pinMode(PIN_SW420, INPUT);
}

```

```

pinMode(PIN_RELAY, OUTPUT);
pinMode(PIN_BUZZER, OUTPUT);
pinMode(PIN_BUTTON, INPUT);

noTone(PIN_BUZZER);
setMotor(false);

lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Pengayak Pasir");
lcd.setCursor(0, 1);
lcd.print("Inisialisasi...");

Serial.println("[HX711] Inisialisasi load cell...");
scale.begin(PIN_HX711_DT, PIN_HX711_SCK);
scale.set_scale(calibration_factor);
scale.tare();
Serial.println("[HX711] Tare selesai");

Serial.println("[BLYNK] Mencoba terhubung ke WiFi/Blynk...");
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
Serial.println("[BLYNK] Terhubung");

timer.setInterval(1000L, kirimDataKeBlynk);
timer.setInterval(200L, cekTombolManual);
timer.setInterval(500L, cekMonitoringGetaran);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sistem Siap");
lcd.setCursor(0, 1);
lcd.print("Tekan START");

Serial.println("[SYSTEM] Sistem siap. Data sensor tampil setiap 1
detik.");
}

void loop() {
  Blynk.run();
  timer.run();
  updateBuzzer();
}

```