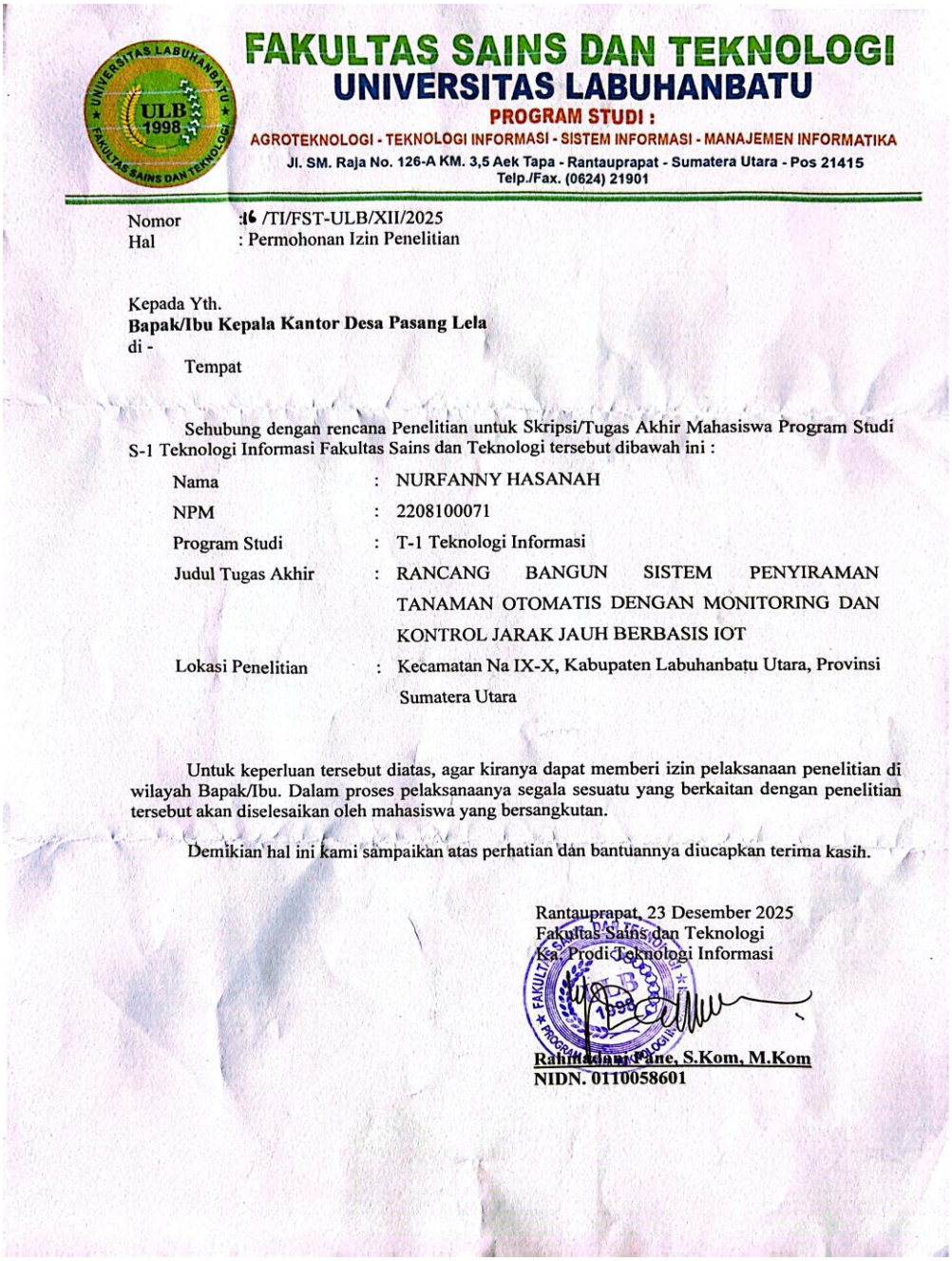
LAMPIRAN DOKUMENTASI

Lampiran 1 Tempat Perkebunan Tanaman

Lampiran 2 Dokumentasi Penulis

Lampiran 3 Dokumentasi Hasil Di Desa Pasang lela

LAMPIRAN SURAT



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 : /TI/FST-ULB/XII/2025
Hal : Permohonan Izin Penelitian

Kepada Yth.
Bapak/Ibu Kepala Kantor Desa Pasang Lela
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	:	NURFANNY HASANAH
NPM	:	2208100071
Program Studi	:	T-1 Teknologi Informasi
Judul Tugas Akhir	:	RANCANG BANGUN SISTEM PENYIRAMAN TANAMAN OTOMATIS DENGAN MONITORING DAN KONTROL JARAK JAUH BERBASIS IOT
Lokasi Penelitian	:	Kecamatan Na IX-X, Kabupaten Labuhanbatu Utara, Provinsi Sumatera Utara

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, 23 Desember 2025
Fakultas Sains dan Teknologi
Ka. Prodi Teknologi Informasi

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

Lampiran 4 Surat Permohonan Izin Penelitian

PEMERINTAH KABUPATEN LABUHANBATU UTARA
KECAMATAN NA. IX - X
KEPALA DESA PASANG LELA
Jalan Mesjid Pasar Lori No. Kode Pos 21454

Nomor : 400 / 665 PL / 2025
Lamp : -
Hal : Balasan Surat Izin Penelitian

Dengan hormat,
Berdasarkan Surat dari Universitas Labuhanbatu Nomor: 16/TI/FST-ULB/XII/2025 tanggal 23 Desember 2025 tentang Permohonan Izin Penelitian di Desa Pasang Lela, dalam rangka penyusunan Skripsi Mahasiswa guna memperoleh gelar S1, maka dengan ini Desa Pasang Lela, Kecamatan NA.IX-X, Kabupaten Labuhanbatu Utara Provinsi Sumatera Utara mengizinkan bahwa:

Nama	:	**NURFANNY HASANAH**
NPM	:	2208100071
Program Studi	:	T-1 Teknologi Informasi
Judul Tugas Akhir	:	Rancang Bangun Sistem Penyiraman Tanaman Otomatis Dengan Monitoring dan Kontrol Jarak Jauh Berbasis IOT.

Akan melalukan izin penelitian di Desa Pasang Lela, Kecamatan NA.IX-X, Kabupaten Labuhanbatu Utara pada tanggal 23 Desember 2025 s/d selesai.

Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat dipergunakan sebagaimana mestinya.

Pasang Lela, 23 Desember 2025
Kepala Desa Pasang Lela

PONIDI

Lampiran 5 Surat Balasan penelitian

Lampiran Codingan

```
#define BLYNK_TEMPLATE_ID   "TMPL61OO8CTpB"
#define BLYNK_TEMPLATE_NAME "Penyiraman Tanaman"
#define BLYNK_AUTH_TOKEN    "05yFPCXuRq1z-TipzgmuBGmOPPVjRsOw"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char ssid[] = "SOLIN_PROJECT";
char pass[] = "17AGUSTUS2025";

// ========== PIN ==========
#define RELAY_PIN       14
#define BUZZER_PIN      27
#define WIFI_LED_PIN     2
#define BLYNK_LED_PIN   25
#define SOIL_SENSOR_PIN 34

// ========== VIRTUAL PIN BLYNK ==========
#define VPIN_MANUAL_RELAY  V0
#define VPIN_TIMER_SLIDER  V1
#define VPIN_START_TIMER   V2
#define VPIN_STATUS        V3
#define VPIN_KELEMBAPAN    V4
#define VPIN_AUTO_MODE     V5

// ========== KALIBRASI SENSOR (hasil pengukuran nyata) ==========
#define SOIL_DRY_VALUE  4095   // ADC saat sensor di udara
#define SOIL_WET_VALUE  2048   // ADC saat sensor dicelup air

// Threshold + hysteresis untuk cegah relay nyala-mati terus
#define KELEMBAPAN_MIN  60     // < 60%  → relay ON
#define KELEMBAPAN_MAX  65     // >= 65% → relay OFF

// ========== VARIABEL STATUS ==========
bool relayStatus   = false;
bool timerActive   = false;
bool manualActive  = false;
bool autoMode      = true;
bool autoWatering  = false;

int  timerMinutes     = 5;
int  kelembapanPersen = 0;
int  rawADC           = 0;

unsigned long timerStartTime      = 0;
unsigned long timerDuration       = 0;
unsigned long manualStartTime     = 0;
unsigned long lastStatusUpdate    = 0;
unsigned long lastSensorRead      = 0;
unsigned long lastReconnectAttempt = 0;

const unsigned long manualMaxDuration   = 5  * 60 * 1000UL;
const unsigned long SENSOR_READ_INTERVAL = 2000UL;
const unsigned long reconnectInterval   = 30 * 1000UL;
```

```
// ========== BUZZER ==========
bool buzzerActive  = false;
int  buzzerCount   = 0;
unsigned long lastBuzzerTime = 0;

// ================================================================
//  BACA SENSOR — rata-rata 10 sample untuk kurangi noise
// ================================================================
int bacaKelembapan() {
  long sum = 0;
  for (int i = 0; i < 10; i++) {
    sum += analogRead(SOIL_SENSOR_PIN);
    delay(10);
  }
  rawADC = sum / 10;

  // DRY (4095) → 0%,  WET (2048) → 100%
  int persen = map(rawADC, SOIL_DRY_VALUE, SOIL_WET_VALUE, 0, 100);
  persen = constrain(persen, 0, 100);

  Serial.print("[SENSOR] RAW=");
  Serial.print(rawADC);
  Serial.print("  Kelembapan=");
  Serial.print(persen);
  Serial.println("%");

  return persen;
}

// ================================================================
//  RELAY & BUZZER
// ================================================================
void setRelayState(bool state) {
  relayStatus = state;
  digitalWrite(RELAY_PIN, state ? LOW : HIGH);
  startBuzzer();
  Blynk.virtualWrite(VPIN_MANUAL_RELAY, state ? 1 : 0);
  updateStatusDisplay();
}

void startBuzzer() {
  buzzerActive   = true;
  buzzerCount    = 0;
  lastBuzzerTime = millis();
}

void handleBuzzer() {
  if (!buzzerActive) return;
  if (millis() - lastBuzzerTime >= 300) {
    if (buzzerCount < 2) {
      digitalWrite(BUZZER_PIN, LOW);
      delay(150);
      digitalWrite(BUZZER_PIN, HIGH);
      buzzerCount++;
    } else {
      buzzerActive = false;
    }
    lastBuzzerTime = millis();
```

```
  }
}

// ============================================================
//  STATUS DISPLAY
// ============================================================
void updateStatusDisplay() {
  String status = "";

  if (autoMode) {
    status += "🤖 Mode: OTOMATIS\n";
    status += autoWatering ? "🚿 Sedang menyiram...\n" : "✅ Kelembapan OK\n";
  } else {
    status += "✋ Mode: MANUAL\n";
  }

  status += "💧 Kelembapan: " + String(kelembapanPersen) + "%\n";
  status += "🎯 Target: >= " + String(KELEMBAPAN_MIN) + "%\n";

  if (timerActive) {
    unsigned long elapsed   = millis() - timerStartTime;
    unsigned long remaining = (timerDuration > elapsed) ? (timerDuration - elapsed) : 0;
    status += "⏱ Timer: " + String(remaining / 60000) + "m " +
              String((remaining % 60000) / 1000) + "s\n";
  } else if (manualActive) {
    unsigned long elapsed   = millis() - manualStartTime;
    unsigned long remaining = (manualMaxDuration > elapsed) ? (manualMaxDuration - elapsed) :
0;
    status += "✋ Manual: " + String(remaining / 60000) + "m " +
              String((remaining % 60000) / 1000) + "s\n";
  }

  status += "WiFi: "  + String(WiFi.status() == WL_CONNECTED ? "ON" : "OFF");
  status += " | Blynk: " + String(Blynk.connected() ? "ON" : "OFF");

  Blynk.virtualWrite(VPIN_STATUS, status);
}

// ============================================================
//  LOGIKA OTOMATIS (hysteresis 60–65%)
// ============================================================
void handleAutoMode() {
  if (!autoMode)    return;
  if (manualActive) return;
  if (timerActive)  return;

  if (kelembapanPersen < KELEMBAPAN_MIN && !relayStatus) {
    autoWatering = true;
    setRelayState(true);
    Serial.println("[AUTO] Tanah kering! Relay ON.");
  } else if (kelembapanPersen >= KELEMBAPAN_MAX && relayStatus && autoWatering) {
    autoWatering = false;
    setRelayState(false);
    Serial.println("[AUTO] Kelembapan cukup. Relay OFF.");
  }
}

// ============================================================
```

```
//  TIMER
// ============================================================
void startWateringTimer() {
  if (timerMinutes <= 0 || timerMinutes > 15) {
    Blynk.virtualWrite(VPIN_START_TIMER, 0);
    return;
  }
  autoWatering   = false;
  manualActive   = false;
  timerActive    = true;
  timerStartTime = millis();
  timerDuration  = (unsigned long)timerMinutes * 60 * 1000UL;

  setRelayState(true);
  Blynk.virtualWrite(VPIN_START_TIMER, 0);
  Serial.println("[TIMER] Dimulai: " + String(timerMinutes) + " menit.");
}

void stopWateringTimer() {
  timerActive  = false;
  autoWatering = false;
  setRelayState(false);
  Serial.println("[TIMER] Selesai. Relay OFF.");
}

void stopManualTimeout() {
  manualActive = false;
  autoWatering = false;
  setRelayState(false);
  Serial.println("[MANUAL] Timeout 5 menit. Relay OFF.");
}

// ============================================================
//  BLYNK CALLBACKS
// ============================================================
BLYNK_WRITE(VPIN_MANUAL_RELAY) {
  int val = param.asInt();
  if (timerActive) timerActive = false;
  autoWatering = false;

  if (val == 1) {
    manualActive    = true;
    manualStartTime = millis();
    setRelayState(true);
    Serial.println("[MANUAL] Relay ON.");
  } else {
    manualActive = false;
    setRelayState(false);
    Serial.println("[MANUAL] Relay OFF.");
  }
}

BLYNK_WRITE(VPIN_TIMER_SLIDER) {
  timerMinutes = param.asInt();
  if (timerMinutes > 15) timerMinutes = 15;
  updateStatusDisplay();
}

BLYNK_WRITE(VPIN_START_TIMER) {
```

```
  if (param.asInt() == 1) {
    if (!timerActive && !manualActive) startWateringTimer();
    else Blynk.virtualWrite(VPIN_START_TIMER, 0);
  }
}

BLYNK_WRITE(VPIN_AUTO_MODE) {
  autoMode = param.asInt();
  if (!autoMode && autoWatering) {
    autoWatering = false;
    setRelayState(false);
  }
  Serial.println("[MODE] Auto: " + String(autoMode ? "ON" : "OFF"));
  updateStatusDisplay();
}

BLYNK_CONNECTED() {
  digitalWrite(BLYNK_LED_PIN, HIGH);
  Blynk.syncVirtual(VPIN_MANUAL_RELAY);
  Blynk.syncVirtual(VPIN_TIMER_SLIDER);
  Blynk.syncVirtual(VPIN_AUTO_MODE);
  Blynk.virtualWrite(VPIN_TIMER_SLIDER, timerMinutes);
  Blynk.virtualWrite(VPIN_AUTO_MODE, autoMode ? 1 : 0);
  updateStatusDisplay();
  Serial.println("[BLYNK] Terhubung!");
}

BLYNK_DISCONNECTED() {
  digitalWrite(BLYNK_LED_PIN, LOW);
  Serial.println("[BLYNK] Terputus.");
}

// ================================================================
//  SETUP
// ================================================================
void setup() {
  Serial.begin(115200);
  Serial.println("\n=== Sistem Penyiraman Cabai ===");
  Serial.println("DRY=" + String(SOIL_DRY_VALUE) + "  WET=" +
String(SOIL_WET_VALUE));

  pinMode(RELAY_PIN,     OUTPUT); digitalWrite(RELAY_PIN,     HIGH);
  pinMode(BUZZER_PIN,    OUTPUT); digitalWrite(BUZZER_PIN,    HIGH);
  pinMode(WIFI_LED_PIN,  OUTPUT); digitalWrite(WIFI_LED_PIN,  LOW);
  pinMode(BLYNK_LED_PIN, OUTPUT); digitalWrite(BLYNK_LED_PIN, LOW);

  WiFi.begin(ssid, pass);
  Serial.print("Menghubungkan WiFi");
  int timeout = 0;
  while (WiFi.status() != WL_CONNECTED && timeout < 20) {
    delay(1000);
    Serial.print(".");
    timeout++;
  }

  if (WiFi.status() == WL_CONNECTED) {
    digitalWrite(WIFI_LED_PIN, HIGH);
    Serial.println("\nWiFi terhubung! IP: " + WiFi.localIP().toString());
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
```

```
  } else {
    Serial.println("\nGagal konek WiFi. Akan coba lagi otomatis...");
  }
}

// ============================================================
//  LOOP
// ============================================================
void loop() {
  digitalWrite(WIFI_LED_PIN, WiFi.status() == WL_CONNECTED ? HIGH : LOW);

  if (WiFi.status() == WL_CONNECTED) {
    Blynk.run();
    digitalWrite(BLYNK_LED_PIN, Blynk.connected() ? HIGH : LOW);

    handleBuzzer();

    // Baca sensor tiap 2 detik
    if (millis() - lastSensorRead >= SENSOR_READ_INTERVAL) {
      kelembapanPersen = bacaKelembapan();
      Blynk.virtualWrite(VPIN_KELEMBAPAN, kelembapanPersen);
      handleAutoMode();
      lastSensorRead = millis();
    }

    // Cek timer selesai
    if (timerActive && millis() - timerStartTime >= timerDuration) {
      stopWateringTimer();
    }

    // Cek timeout manual
    if (manualActive && millis() - manualStartTime >= manualMaxDuration) {
      stopManualTimeout();
    }

    // Update status tiap 1 detik (hanya saat aktif)
    if (millis() - lastStatusUpdate >= 1000) {
      if (timerActive || manualActive || autoWatering) {
        updateStatusDisplay();
      }
      lastStatusUpdate = millis();
    }

  } else {
    digitalWrite(BLYNK_LED_PIN, LOW);
    if (millis() - lastReconnectAttempt > reconnectInterval) {
      Serial.println("[WIFI] Mencoba reconnect...");
      WiFi.reconnect();
      lastReconnectAttempt = millis();
    }
  }

  delay(100);
}
```