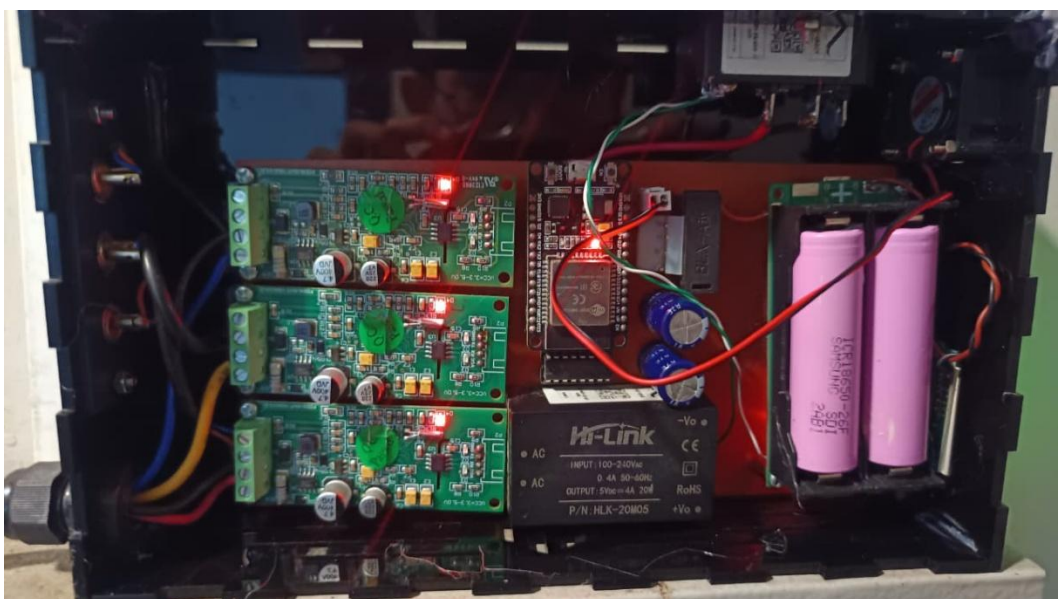
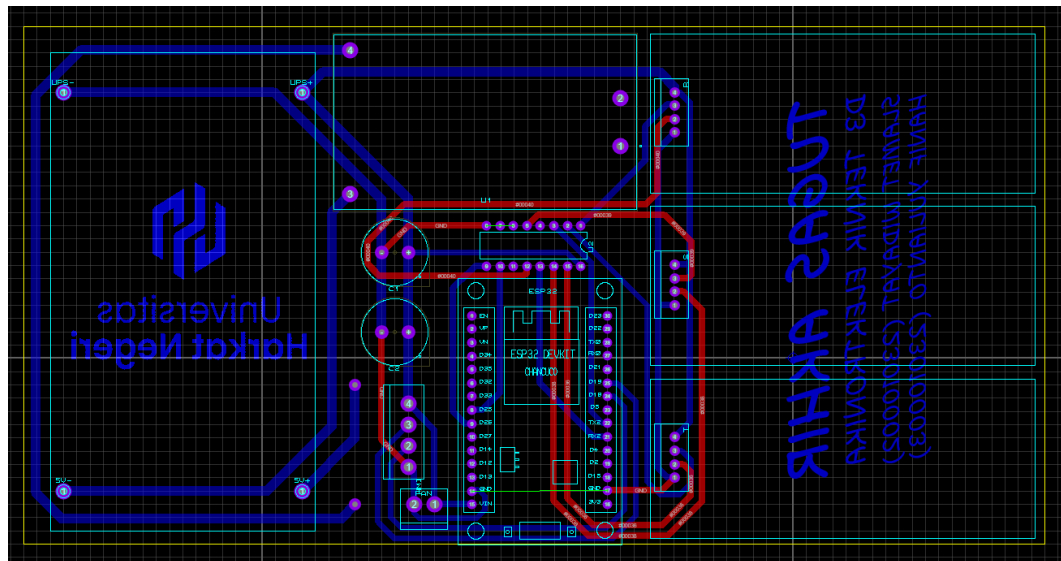


Lampiran 1 Dokumentasi Pembuatan



Lampiran 2 Program Arduino IDE

```

#include <Firebase_ESP_Client.h>
#include <WiFi.h>
#include <PZEM004Tv30.h>
#include <HardwareSerial.h>
#include <time.h>
#include <math.h>

// WIFI
#define WIFI_SSID      "WIFI"
#define WIFI_PASSWORD  "087816092487"

// FIREBASE
#define API_KEY "AIzaSyA_fYDjte2ZTD1xTRoGe3VUGm0IHN48To0"
#define DATABASE_URL "https://monitoring-gedung-uhn-2ea1f-default-rtdb.asia-southeast1.firebaseio.com"

// CD4052
#define SEL_A 25
#define SEL_B 26

// UART PZEM
#define PZEM_RX 16
#define PZEM_TX 17
#define PZEM_BAUD 9600

// UART HMI DWIN
#define DWIN_RX 18
#define DWIN_TX 19
#define DWIN_BAUD 115200

// VP DATA VARIABLE DWIN
// Fase R
#define VP_R_VOLTAGE 0x5000
#define VP_R_CURRENT 0x5100
#define VP_R_POWER 0x5200
#define VP_R_ENERGY 0x5300
#define VP_R_FREQUENCY 0x5400
#define VP_R_PF 0x5500

// Fase S
#define VP_S_VOLTAGE 0x6000
#define VP_S_CURRENT 0x6100
#define VP_S_POWER 0x6200
#define VP_S_ENERGY 0x6300

```

```

#define VP_S_FREQUENCY 0x6400
#define VP_S_PF         0x6500

// Fase T
#define VP_T_VOLTAGE    0x7000
#define VP_T_CURRENT    0x7100
#define VP_T_POWER      0x7200
#define VP_T_ENERGY     0x7300
#define VP_T_FREQUENCY  0x7400
#define VP_T_PF         0x7500

// FIREBASE OBJECT
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
bool signupOK = false;

// NTP WIB (UTC+7)
const char* ntpServer1 = "pool.ntp.org";
const char* ntpServer2 = "time.google.com";
const char* ntpServer3 = "id.pool.ntp.org";
const long  gmtoffset_sec = 7L * 3600L;
const int   daylightOffset_sec = 0;

// SERIAL
HardwareSerial SerialPZEM(2);
HardwareSerial SerialDWIN(1);
PZEM004Tv30 pzem(&SerialPZEM, PZEM_RX, PZEM_TX);

// TIMER
unsigned long lastHMIUpdate = 0;
unsigned long lastFirebaseSend = 0;
unsigned long lastWiFiRetry = 0;
unsigned long lastFirebaseRetry = 0;

const unsigned long HMI_INTERVAL          = 5000UL;           //
Data angka HMI setiap 5 detik
const unsigned long FIREBASE_INTERVAL    = 900000UL;         // 15
menit
const unsigned long WIFI_RETRY_INTERVAL  = 10000UL;           // 10
detik
const unsigned long FIREBASE_RETRY_INTERVAL = 30000UL;        // 30
detik

bool sensorDataValid = false;

// BATAS STATUS KEAMANAN FASE BERDASARKAN ARUS

```

```

const float MAX_SAFE_CURRENT = 100.0f; // Ampere

// STRUCT
struct DateTimeText {
    String date;
    String time;
};

struct PZEMData {
    float voltage;
    float current;
    float power;
    float energy;
    float frequency;
    float pf;
};

PZEMData phaseR = {0, 0, 0, 0, 0, 0};
PZEMData phaseS = {0, 0, 0, 0, 0, 0};
PZEMData phaseT = {0, 0, 0, 0, 0, 0};

// HELPER
float safeValue(float value)
{
    if (isnan(value) || isinf(value))
        return 0.0f;

    return value;
}

String format2(float value)
{
    if (isnan(value) || isinf(value))
        return "0";

    value = roundf(value * 100.0f) / 100.0f;
    String result = String(value, 2);

    while (result.endsWith("0"))
        result.remove(result.length() - 1);

    if (result.endsWith("."))
        result.remove(result.length() - 1);

    return result;
}

```

```

uint16_t hmiWord(float value, float multiplier)
{
    value = safeValue(value);
    float scaled = roundf(value * multiplier);

    if (scaled < 0.0f)
        scaled = 0.0f;

    if (scaled > 65535.0f)
        scaled = 65535.0f;

    return (uint16_t)scaled;
}

// WIFI
void startWiFi()
{
    WiFi.mode(WIFI_STA);
    WiFi.setAutoReconnect(true);
    WiFi.persistent(false);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    Serial.print("Menghubungkan WiFi");

    unsigned long startAttempt = millis();
    while (WiFi.status() != WL_CONNECTED &&
           millis() - startAttempt < 20000UL)
    {
        Serial.print(".");
        delay(500);
    }

    Serial.println();

    if (WiFi.status() == WL_CONNECTED)
    {
        Serial.println("WiFi terhubung");
        Serial.print("IP ESP32: ");
        Serial.println(WiFi.localIP());
    }
    else
    {
        Serial.println("WiFi belum terhubung. HMI tetap
berjalan.");
    }
}

```

```

void maintainWiFi()
{
    if (WiFi.status() == WL_CONNECTED)
        return;

    if (millis() - lastWiFiRetry < WIFI_RETRY_INTERVAL)
        return;

    lastWiFiRetry = millis();
    Serial.println("Mencoba menyambungkan kembali WiFi...");
    WiFi.disconnect();
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
}

// FIREBASE
void setupFirebase()
{
    if (WiFi.status() != WL_CONNECTED)
        return;

    config.api_key = API_KEY;
    config.database_url = DATABASE_URL;
    Firebase.reconnectWiFi(true);

    if (Firebase.signUp(&config, &auth, "", ""))
    {
        signupOK = true;
        Serial.println("Autentikasi Firebase berhasil");
    }
    else
    {
        signupOK = false;
        Serial.print("Autentikasi Firebase gagal: ");
        Serial.println(config.signer.signupError.message.c_str());
    }
}

Firebase.begin(&config, &auth);
}

void maintainFirebase()
{
    if (signupOK || WiFi.status() != WL_CONNECTED)
        return;

    if (millis() - lastFirebaseRetry < FIREBASE_RETRY_INTERVAL)
        return;

```

```

    lastFirebaseRetry = millis();
    setupFirebase();
}

// NTP DAN WAKTU
bool getTimeInfoSafe(struct tm &timeinfo, uint32_t timeoutMs
= 5000)
{
    return getLocalTime(&timeinfo, timeoutMs);
}

bool waitForNTP(uint32_t timeoutMs = 20000)
{
    if (WiFi.status() != WL_CONNECTED)
        return false;

    Serial.print("Menunggu sinkronisasi NTP");
    unsigned long startWait = millis();
    struct tm timeinfo;

    while (millis() - startWait < timeoutMs)
    {
        if (getTimeInfoSafe(timeinfo, 500))
        {
            Serial.println();
            Serial.println("Waktu NTP tersedia");
            return true;
        }

        Serial.print(".");
        delay(500);
    }

    Serial.println();
    Serial.println("Waktu NTP belum tersedia");
    return false;
}

DateTimeText getDateText()
{
    struct tm timeinfo;

    if (!getTimeInfoSafe(timeinfo, 1000))
        return {"0000-00-00", "00-00-00"};

    char dateBuffer[16];

```

```

    char timeBuffer[16];

    strftime(dateBuffer,    sizeof(dateBuffer),    "%Y-%m-%d",
&timeinfo);
    strftime(timeBuffer,    sizeof(timeBuffer),    "%H-%M-%S",
&timeinfo);

    return {String(dateBuffer), String(timeBuffer)};
}

// CD4052 DAN PZEM
void pilihPZEM(uint8_t sensor)
{
    switch (sensor)
    {
        case 1:
            digitalWrite(SEL_A, LOW);
            digitalWrite(SEL_B, LOW);
            break;

        case 2:
            digitalWrite(SEL_A, HIGH);
            digitalWrite(SEL_B, LOW);
            break;

        case 3:
            digitalWrite(SEL_A, LOW);
            digitalWrite(SEL_B, HIGH);
            break;

        default:
            digitalWrite(SEL_A, LOW);
            digitalWrite(SEL_B, LOW);
            break;
    }

    delay(20);
}

void clearPZEMSerialBuffer()
{
    while (SerialPZEM.available())
        SerialPZEM.read();
}

bool readPZEMCachedFrame(PZEMData &data)
{

```

```

float voltage = pzem.voltage();

if (isnan(voltage) || isinf(voltage))
    return false;

float current    = pzem.current();
float power      = pzem.power();
float energy     = pzem.energy();
float frequency  = pzem.frequency();
float pf         = pzem.pf();

if (isnan(current) || isinf(current) ||
    isnan(power)    || isinf(power)    ||
    isnan(energy)   || isinf(energy)   ||
    isnan(frequency) || isinf(frequency) ||
    isnan(pf)       || isinf(pf))
{
    return false;
}

data.voltage = voltage;
data.current = current;
data.power   = power;
data.energy  = energy;
data.frequency = frequency;
data.pf      = pf;
return true;
}

PZEMData readPZEM(uint8_t sensor)
{
    PZEMData data = {0, 0, 0, 0, 0, 0};

    pilihPZEM(sensor);

    delay(250);
    clearPZEMSerialBuffer();

    if (!readPZEMCachedFrame(data))
    {
        delay(250);
        clearPZEMSerialBuffer();

        if (!readPZEMCachedFrame(data))
        {

```

```

        Serial.printf(
            "PZEM sensor %u tidak merespons -> seluruh data fase
dibuat 0\n",
            sensor
        );
        return {0, 0, 0, 0, 0, 0};
    }
}

Serial.printf(
    "PZEM %u valid: V=%.1f, I=%.2f, P=%.1f, E=%.3f, F=%.1f,
PF=%.2f\n",
    sensor,
    data.voltage,
    data.current,
    data.power,
    data.energy,
    data.frequency,
    data.pf
);

return data;
}

void readAllPZEM()
{
    Serial.println();
    Serial.println("Membaca PZEM fase R, S, dan T...");

    phaseR = readPZEM(1);
    phaseS = readPZEM(2);
    phaseT = readPZEM(3);

    sensorDataValid = true;
}

// DWIN DATA VARIABLE
void dwinWriteWord(uint16_t vp, uint16_t value)
{
    uint8_t frame[8] = {
        0x5A,
        0xA5,
        0x05,
        0x82,
        (uint8_t)(vp >> 8),
        (uint8_t)(vp & 0xFF),
        (uint8_t)(value >> 8),
        (uint8_t)(value & 0xFF)
    };
}

```

```

        (uint8_t)(value & 0xFF)
    };

    SerialDWIN.write(frame, sizeof(frame));
    SerialDWIN.flush();
    delay(3);
}

void dwinWritePhase(
    const PZEMData &data,
    uint16_t vpVoltage,
    uint16_t vpCurrent,
    uint16_t vpPower,
    uint16_t vpEnergy,
    uint16_t vpFrequency,
    uint16_t vpPF)
{
    // Pengaturan desimal pada DGUS:
    dwinWriteWord(vpVoltage,    hmiWord(data.voltage,    10.0f));
    dwinWriteWord(vpCurrent,    hmiWord(data.current,    100.0f));
    dwinWriteWord(vpPower,      hmiWord(data.power,      1.0f));
    dwinWriteWord(vpEnergy,     hmiWord(data.energy,     10.0f));
    dwinWriteWord(vpFrequency,   hmiWord(data.frequency, 10.0f));
    dwinWriteWord(vpPF,         hmiWord(data.pf,         100.0f));
}

void sendCachedDataToHMI()
{
    dwinWritePhase(
        phaseR,
        VP_R_VOLTAGE,
        VP_R_CURRENT,
        VP_R_POWER,
        VP_R_ENERGY,
        VP_R_FREQUENCY,
        VP_R_PF
    );

    dwinWritePhase(
        phaseS,
        VP_S_VOLTAGE,
        VP_S_CURRENT,
        VP_S_POWER,
        VP_S_ENERGY,
        VP_S_FREQUENCY,
        VP_S_PF
    );
}

```

```

);

dwinWritePhase(
    phaseT,
    VP_T_VOLTAGE,
    VP_T_CURRENT,
    VP_T_POWER,
    VP_T_ENERGY,
    VP_T_FREQUENCY,
    VP_T_PF
);

Serial.println("Data angka PZEM dikirim ke HMI DWIN");
}

void updateHMI()
{
    readAllPZEM();
    sendCachedDataToHMI();
}

String getPhaseStatus(const PZEMData &data)
{
    // Status hanya ditentukan dari arus.
    // Arus <= 100 A = Aman, arus > 100 A = Tidak aman.
    return (data.current <= MAX_SAFE_CURRENT)
        ? "Aman"
        : "Tidak aman";
}

// FIREBASE JSON
FirebaseJson createPhaseJson(const PZEMData &data)
{
    FirebaseJson phase;

    // Firebase tetap menggunakan string dan satuan seperti
    program awal.
    phase.set("Voltage",    format2(data.voltage) + " V");
    phase.set("Current",    format2(data.current) + " A");
    phase.set("Power",      format2(data.power) + " W");
    phase.set("Energy",     format2(data.energy) + " kWh");
    phase.set("Frequency",  format2(data.frequency) + " Hz");
    phase.set("PF",         format2(data.pf));
    phase.set("Status",     getPhaseStatus(data));

    return phase;
}

```

```

void printPhase(const char* phaseName, const PZEMData &data)
{
    Serial.println(
        String(phaseName) + " : " +
        format2(data.voltage) + " V | " +
        format2(data.current) + " A | " +
        format2(data.power) + " W | " +
        format2(data.energy) + " kWh | " +
        format2(data.frequency) + " Hz | PF=" +
        format2(data.pf) + " | Status=" +
        getPhaseStatus(data)
    );
}

void sendDataToFirebase()
{
    if (WiFi.status() != WL_CONNECTED)
    {
        Serial.println("Firebase dilewati: WiFi tidak
terhubung");
        return;
    }

    if (!signupOK || !Firebase.ready())
    {
        Serial.println("Firebase belum siap");
        return;
    }

    if (!sensorDataValid)
        readAllPZEM();

    DateTimeText dt = getDateText();

    Serial.println();
    Serial.println("===== DATA FIREBASE =====");
    Serial.println(dt.date + " " + dt.time);
    printPhase("R", phaseR);
    printPhase("S", phaseS);
    printPhase("T", phaseT);

    FirebaseJson realtime;
    realtime.set("Timestamp", dt.date + " " + dt.time);
    realtime.set("R", createPhaseJson(phaseR));
    realtime.set("S", createPhaseJson(phaseS));
    realtime.set("T", createPhaseJson(phaseT));
}

```

```

if (Firebase.RTDB.setJSON(&fbdo, "/Realtime", &realtime))
{
    Serial.println("Realtime berhasil dikirim");
}
else
{
    Serial.print("Realtime gagal: ");
    Serial.println(fbdo.errorReason());
}

FirebaseJson history;
history.set("R", createPhaseJson(phaseR));
history.set("S", createPhaseJson(phaseS));
history.set("T", createPhaseJson(phaseT));

FirebaseJson root;
root.set(dt.time, history);

String path = "/Riwayat/" + dt.date;

if (Firebase.RTDB.updateNode(&fbdo, path.c_str(), &root))
{
    Serial.println("Riwayat berhasil dikirim");
}
else
{
    Serial.print("Riwayat gagal: ");
    Serial.println(fbdo.errorReason());
}

Serial.println("=====");
}

// SETUP
void setup()
{
    Serial.begin(115200);
    delay(500);

    pinMode(SEL_A, OUTPUT);
    pinMode(SEL_B, OUTPUT);
    digitalWrite(SEL_A, LOW);
    digitalWrite(SEL_B, LOW);

    SerialPZEM.begin(
        PZEM_BAUD,

```

```

    SERIAL_8N1,
    PZEM_RX,
    PZEM_TX
);

SerialDWIN.begin(
    DWIN_BAUD,
    SERIAL_8N1,
    DWIN_RX,
    DWIN_TX
);

delay(1000);

startWiFi();

configTime(
    gmtOffset_sec,
    daylightOffset_sec,
    ntpServer1,
    ntpServer2,
    ntpServer3
);

waitForNTP();

setupFirebase();

updateHMI();
lastHMIUpdate = millis();

sendDataToFirebase();
lastFirebaseSend = millis();

    Serial.println("Sistem siap: navigasi halaman dikendalikan
langsung oleh HMI");
}

// LOOP
void loop()
{
    maintainWiFi();
    maintainFirebase();

    unsigned long now = millis();

    if (now - lastHMIUpdate >= HMI_INTERVAL)

```

```
{
  lastHMIUpdate = now;
  updateHMI();
}

if (now - lastFirebaseSend >= FIREBASE_INTERVAL)
{
  lastFirebaseSend = now;
  sendDataToFirebase();
}

delay(5);
}
```

Lampiran 3 Kesiadaan Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TUGAS AKHIR

Yang bertanda tangan

Nama : Bahrn Niam, M.T

NIPY : NIPY. 09.015.277

Jabatan: Dosen Tetap Prodi DIII Teknik Elektronika

Dengan ini dinyatakan bersedia menjadi Pembimbing I Tugas Akhir Mahasiswa berikut :

Nama	: HANIF YULIANTO
NIM	: 23010003
Program Studi	: DIII Teknik Elektronika
Judul Laporan Tugas Akhir	: RANCANG BANGUN SISTEM MONITORING PARAMETER LISTRIK 3 FASA BERBASIS ESP32 DAN SENSOR PZEM-004T PADA GEDUNG D LANTAI 4, UNIVERSITAS HARKAT NEGERI DENGAN ANTARMUKA HMI

Demikian Pernyataan ini di buat agar dapat di laksanakan sebagai mestinya

Tegal, 16 Februari 2026

Calon Dosen Pembimbing I

Mengetahui,

Ka. Prodi DIII Teknik Elektronika

Universitas Harkat Negeri



Rony Darpono, M.T

NIPY. 09.015.282



Bahrn Niam, M.T

NIPY. 09.015.277

Lampiran 4 Kesiediaan Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TUGAS AKHIR

Yang bertanda tangan

Nama : Much Sobri Sungkar, M.Kom

NIPY : NIPY. 09.012.144

Jabatan: Dosen Tetap Prodi DIII Teknik Elektronika

Dengan ini dinyatakan bersedia menjadi Pembimbing II Tugas Akhir Mahasiswa berikut :

Nama	: HANIF YULIANTO
NIM	: 23010003
Program Studi	: DIII Teknik Elektronika
Judul Laporan Tugas Akhir	: RANCANG BANGUN SISTEM MONITORING PARAMETER LISTRIK 3 FASA BERBASIS ESP32 DAN SENSOR PZEM-004T PADA GEDUNG D LANTAI 4, UNIVERSITAS HARKAT NEGERI DENGAN ANTARMUKA HMI

Demikian Pernyataan ini di buat agar dapat di laksanakan sebagai mestinya

Tegal, 16 Februari 2026

Calon Dosen Pembimbing 2

Mengetahui,

Ka. Prodi DIII Teknik Elektronika

Universitas Harkat Negeri



Rony Darpono, M.T

NIPY. 09.015.282


Much Sobri Sungkar, M.Kom

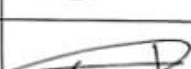
NIPY. 09.012.144

Lampiran 5 Form Pembimbing 1

FORM BIMBINGAN TUGAS AKHIR

NAMA HANIF YULIANTO
 NIM 23010003
 JUDUL TA RANCANG BANGUN SISTEM MONITORING
 PARAMETER LISTRIK 3 FASA BERBASIS ESP32
 DAN SENSOR PZEM-004T PADA GEDUNG D
 LANTAI 4, UNIVERSITAS HARKAT NEGERI
 DENGAN ANTARMUKA HMI

Pembimbing 1

No	Hari/Tanggal	Uraian	Tanda Tangan
1	Kamis/7 Mei 2026	Bimbingan Judul	
2	Selasa/12 Mei 2026	Observasi Panel Listrik	
3	Jumat/22 Mei 2026	Pembagian Pembahasan	
4	Rabu/10 Juni 2026	Konsultasi Terkait UART ESP32 dan Sensor	
5	Senin/29 Juni 2026	Revisi bab I	
6	Rabu/15 Juli 2026	Revisi Gambar bab II	
7	Kamis/23 Juli 2026	Revisi Bab III	
8	Senin/27 Juli 2026	Revisi bab IV	

Lampiran 6 Form Pembimbing 2

FORM BIMBINGAN TUGAS AKHIR

NAMA HANIF YULIANTO
 NIM 23010003
 JUDUL TA RANCANG BANGUN SISTEM MONITORING
 PARAMETER LISTRIK 3 FASA BERBASIS ESP32
 DAN SENSOR PZEM-004T PADA GEDUNG D
 LANTAI 4, UNIVERSITAS HARKAT NEGERI
 DENGAN ANTARMUKA HMI

Pembimbing 2

No	Hari/Tanggal	Uraian	Tanda Tangan
1	Selasa/9 Juni 2026	Bimbingan desain box	
2	Selasa/16 Juni 2026	Bimbingan komponen	
3	Kamis/25 Juni 2026	Bimbingan Bab I - III	
4	Senin/20 Juni 2026	Revisi Bab IV	
5	Rabu/29 Juli 2026	Bimbingan Bab IV - V	

Lampiran 7 Penilaian Pembimbing

PENILAIAN BIMBINGAN TUGAS AKHIR INDIVIDU

Judul : "RANCANG BANGUN SISTEM MONITORING
PARAMETER LISTRIK 3 FASA BERBASIS ESP32 DAN
SENSOR PZEM-004T PADA GEDUNG D LANTAI 4,
UNIVERSITAS HARKAT NEGERI DENGAN
ANTARMUKA HMI"

Nama : HANIF YULIANTO

NIM : 23010003

Program Studi : Teknik Elektronika

I. Nilai Bimbingan Tugas Akhir (Pembimbing I)

NO	Unsur yang di nilai	Nilai
1	Kedisiplinan dalam bimbingan	95
2	Kreativitas Pemecahan dalam bimbingan	95
3	Penguasaan Materi Tugas Akhir	90
4	Kelengkapan dan Refensi Tugas Akhir	85
	Total Nilai = (Jumlah Nilai) : 4	91,25

II. Nilai Bimbingan Tugas Akhir (Pembimbing II)

NO	Unsur yang di nilai	Nilai
1	Kedisiplinan dalam bimbingan	80
2	Kreativitas Pemecahan dalam bimbingan	80
3	Penguasaan Materi Tugas Akhir	80
4	Kelengkapan dan Refensi Tugas Akhir	80
	Total Nilai = (Jumlah Nilai) : 4	80

Nilai bimbingan = (Total nilai pembimbing I + Total nilai pembimbing II) :
2

=

Tegal, 10 Juli 2026

Mengetahui,

Pembimbing I

Pembimbing II



Bahrin Niam, M.T
NIPY. 09.015.277



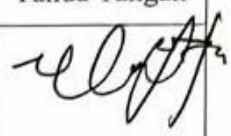
Much Sobri Sungkar, M.Kom
NIPY. 09.012.144

Lampiran 8 Form Revisi Ketua Penguji

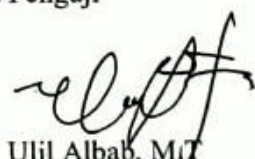
FORM REVISI UJIAN TUGAS AKHIR

NAMA HANIF YULIANTO
 NIM 23010003
 JUDUL TA RANCANG BANGUN SISTEM MONITORING
 PARAMETER LISTRIK 3 FASA BERBASIS ESP32
 DAN SENSOR PZEM-004T PADA GEDUNG D
 LANTAI 4, UNIVERSITAS HARKAT NEGERI
 DENGAN ANTARMUKA HMI

Ketua Penguji

No	Hari/Tanggal	Uraian	Tanda Tangan
1	4-8-2026	Revisi proyek	
2	16-8-2026	Acc laporan & proyek	

Ketua Penguji

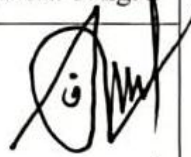

Ulil Albab, M.T.
 NIPY. 04.015.271

Lampiran 9 Form Revisi Penguji 1

FORM REVISI UJIAN TUGAS AKHIR

NAMA HANIF YULIANTO
 NIM 23010003
 JUDUL TA RANCANG BANGUN SISTEM MONITORING
 PARAMETER LISTRIK 3 FASA BERBASIS ESP32
 DAN SENSOR PZEM-004T PADA GEDUNG D
 LANTAI 4, UNIVERSITAS HARKAT NEGERI
 DENGAN ANTARMUKA HMI

Penguji 1

No	Hari/Tanggal	Uraian	Tanda Tangan
1.	Selasa/ 4 Agustus 2026	Revisi Tampilan HMI	
2.	Senin / 10 Agustus 2026	Revisi BAB 4	
3.	Jumat / 14 Agustus 2026	Acc Laporan	

Penguji 1



Qirom, S.Pd, M.T
 NIPY. 09.015.281

Lampiran 10 Form Revisi Penguji 2

FORM REVISI UJIAN TUGAS AKHIR

NAMA HANIF YULIANTO
 NIM 23010003
 JUDUL TA RANCANG BANGUN SISTEM MONITORING
 PARAMETER LISTRIK 3 FASA BERBASIS ESP32
 DAN SENSOR PZEM-004T PADA GEDUNG D
 LANTAI 4, UNIVERSITAS HARKAT NEGERI
 DENGAN ANTARMUKA HMI

Penguji 2

No	Hari/Tanggal	Uraian	Tanda Tangan
1.	Selasa / 4-8-2026	Revisi Alat	
2.	Senin / 10-8-2026	Revisi Laporan BAB 4	
3.	Jumat / 14-8-2026	<u>ACC Laporan</u>	

Penguji 2


 Much Sobri Sungkar, M.Kom
 NIPY. 09.012.144