LAMPIRAN

Lampiran – 1 Listing Program

A. Listing Program *Node Server*

```
/*
OLED      ESP32
SDA       D21
SCL       D22
Switch    D32

LoraUART  ESP32
TX        RX2
RX        TX2
AUX       D15
VCC       5V
GND       GND
*/

#include "Arduino.h"
#include <LoRa.h>
HardwareSerial loraSerial(2);

// Import Wi-Fi library
#include <WiFi.h>
#include "ESPAsyncWebServer.h"
#include <SPIFFS.h>

//Libraries for OLED Display
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

// Libraries to get time from NTP Server
#include <NTPClient.h>
#include <WiFiUdp.h>

//Switch pin
#define SW 32

// Replace with your network credentials
const char* ssid     = "POCOF6FADHIL";
const char* password = "pokerface";
```

```
// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variables to save date and time
String formattedDate, day, hour;
String timestamp;
int counter=0;
unsigned long TAwal=0;
unsigned long TAkir = 10000; //millidetik

// Initialize variables to get and save LoRa data
String dataTX="", dataRX,CNum;
String tegangan, arus, energi, stLED;
String sukses= "LoRa Initializing OK!";
String gagal = "Starting LoRa failed!";
bool CSt=1, RSt=0;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

String processor(const String& var){
  if(var == "V1"){ return tegangan; }
  else if(var == "A1"){ return arus; }
  else if(var == "E1"){ return energi; }
  else if(var == "L1"){ return stLED; }
  else if(var == "TIMESTAMP"){ return timestamp; }
  return String();
}

//Initialize LoRa module
void startLoRA(){
  loraSerial.begin(9600,SERIAL_8N1,16,17);
  while (!loraSerial && counter < 10) {
    Serial.print(".");
    counter++;
    delay(100);
  }
}

void connectWiFi(){
  // Connect to Wi-Fi network with SSID and password
  Serial.print("Connecting to ");
  Serial.println(ssid);
```

```
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  displayIP();
}

// Function to get date and time from NTPClient
void getTimeStamp() {
  while(!timeClient.update()) {
    timeClient.forceUpdate();
  }
  // The formattedDate comes with the following format:
  // 2018-05-28T16:00:13Z
  // We need to extract date and time
  formattedDate = timeClient.getFormattedDate();
  //Serial.println(formattedDate);

  // Extract date
  int splitT = formattedDate.indexOf("T");
  day = formattedDate.substring(0, splitT);
  //Serial.println(day);
  // Extract time
  hour = formattedDate.substring(splitT+1, formattedDate.length()-1);
  //Serial.println(hour);
  timestamp = day+"  "+hour;
}

void displayIP() {
  lcd.setCursor(0,0);
  lcd.print("Alamat WebServer");
  lcd.setCursor(0,1);
  lcd.print(WiFi.localIP());
}

void displayData() {
  lcd.setCursor(0,0); lcd.print(F("Teg:     V L:   "));
  lcd.setCursor(5,0); lcd.print(tegangan);
  lcd.setCursor(13,0); lcd.print(stLED);
```

```
  lcd.setCursor(0,1); lcd.print(F("Arus:      A    "));
  lcd.setCursor(6,1); lcd.print(arus);
  //lcd.setCursor(0,1); lcd.print(F("E1:"));lcd.print(energi);
  //lcd.print(F(" kWh"));

}

//Initialize LCD display
void startLCD(){
  lcd.init(); lcd.backlight();
  lcd.setCursor(0,0); lcd.print("LORA SENDER");
  delay(2000);
}

void setup() {
  // Initialize Serial Monitor
  Serial.begin(115200);
  pinMode(SW, INPUT_PULLUP);

  startLCD();
  startLoRA();
  connectWiFi();

  if(!SPIFFS.begin()){
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
  }
  // Route for root / web page
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });
  // Route to load style.css file
     server.on("/style.css",    HTTP_GET,    [](AsyncWebServerRequest
*request){
    request->send(SPIFFS, "/style.css", "text/css");
  });
     server.on("/bgimage.jpg",   HTTP_GET,   [](AsyncWebServerRequest
*request){
    request->send(SPIFFS, "/bgimage.jpg", "image/jpg");
  });

//==========================                Customer                1
============================
     server.on("/tegangan",    HTTP_GET,    [](AsyncWebServerRequest
*request){
```

```
    request->send_P(200, "text/plain", tegangan.c_str());
  });
  server.on("/arus", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", arus.c_str());
  });
  server.on("/energi", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", energi.c_str());
  });
  server.on("/stLED", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", stLED.c_str());
  });
  // Route to set CSt to HIGH
  server.on("/LedON", HTTP_GET, [](AsyncWebServerRequest *request){
    CSt=1;
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });
  // Route to set CSt to LOW
  server.on("/LedOFF", HTTP_GET, [](AsyncWebServerRequest *request){
    CSt=0;
    request->send(SPIFFS, "/index.html", String(), false, processor);
  // Route to set RSt to HIGH
  });
  server.on("/RESET", HTTP_GET, [](AsyncWebServerRequest *request){
    RSt=1;
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

  // Start server
  server.begin();

  // Initialize a NTPClient to get time
  timeClient.begin();
   // Set offset time in seconds to adjust for your timezone, for
example:
  // GMT +1 = 3600
  // GMT +8 = 28800
  // GMT -1 = -3600
  // GMT 0 = 0
  timeClient.setTimeOffset(28800);
}

void kirimData() {
  dataTX = "C1/"+String(CSt)+"*"+String(RSt)+"~S";
  loraSerial.print(dataTX); //Kirimkan LoRa packet ke receiver
  Serial.println("Paket kirim : "+dataTX);
```

```
  delay(1500);
}

// Read LoRa packet and get the sensor readings
void terimaData() {
  if(loraSerial.available()>0) {
    dataRX = loraSerial.readStringUntil('S'); // Read packet
    // dataRX.trim(); // remove whitespace
    // dataRX format: CNum&tegangan#current$energy*CSt~S
    // String example: C1&224#0.07$0.013*ON~
    Serial.println("Paket terima: "+dataRX+"\n");
    int pos1 = dataRX.indexOf('&');
    int pos2 = dataRX.indexOf('#');
    int pos3 = dataRX.indexOf('$');
    int pos4 = dataRX.indexOf('*');
    int pos5 = dataRX.indexOf('~');

    CNum = dataRX.substring(0, pos1);
    if(CNum=="C1") {
      tegangan = dataRX.substring(pos1+1, pos2);
      arus = dataRX.substring(pos2+1, pos3);
      energi  = dataRX.substring(pos3+1, pos4);
      stLED = dataRX.substring(pos4+1, pos5);
    }
    TAwal=millis();
  }
  if((millis()-TAwal)>TAkir) {
      tegangan = "000";
      arus = "0.00";
      energi  = "000";
      stLED = " NC";
  }
}

void loop() {
    kirimData();
    terimaData();
    getTimeStamp();
    if(digitalRead(SW)) displayIP();
    else displayData();
}
```

B. Listing Program *Node* Pelanggan

```
/*
LGT8F  UART-LoRa
TX    -> RXD
RX    -> TXD
5V    -> VCC
GND   -> GND
       PZEM-004T
D2    -> RX
D3    -> TX
5V    -> VCC
GND   -> GND
D4    -> LED
*/
#include <PZEM004Tv30.h>
#include <Wire.h>
PZEM004Tv30 pzem(2,3); //D2->TX_PZEM, D3->RX_PZEM

#define LED 10
String stLED, CNum="C1", CRX, cnLED, cnRST;
String dataIn=""; //data dari Server yang diterima

unsigned int readingID = 0; //packet counter
int counter = 0;
String LoRaMessage = "";
float volt, arus, energi;

void startLoRA(){ //Initialize LoRa module
  Serial.begin(9600);
  delay(2000);
}

void bacaData(){
  volt = pzem.voltage();
  if(isnan(volt)) volt=0.0;
  arus = pzem.current();
  if((arus)<0.1) arus=0.00;
  energi = pzem.energy();
  stLED=(digitalRead(LED)==0? "OFF":"ON");
  delay(500);
}

void kirimData() {
  LoRaMessage = CNum+"&"+String(volt,0)+"#"+
                String(arus)+"$"+
```

```
                String(energi,3)+"*"+stLED+"~S";
  Serial.print(LoRaMessage); //Kirimkan data tanpa enter ke receiver
}

void setup() {
  pinMode(LED, OUTPUT);
  digitalWrite(LED,LOW);
  startLoRA();
  bacaData();
}
void loop() {
  if(Serial.available()>0) {
     dataIn = Serial.readStringUntil('S');
     int pos1 = dataIn.indexOf('/');
     int pos2 = dataIn.indexOf('*');
     int pos3 = dataIn.indexOf('~');
     CRX = dataIn.substring(0, pos1);
     //cnLED= dataIn.substring(pos1+1, dataIn.length());
     cnLED= dataIn.substring(pos1+1,pos2);
     cnRST= dataIn.substring(pos2+1,pos3);
     if(CNum==CRX) {
        bacaData();
        kirimData();
        if(cnLED=="0") {
           stLED="OFF";
           for(byte i=0; i<3; i++) {
             digitalWrite(LED,1); delay(100);
             digitalWrite(LED,0); delay(100);
           }
         }
        else {
           stLED="ON";
           for(byte i=0; i<3; i++) {
             digitalWrite(LED,0); delay(100);
             digitalWrite(LED,1); delay(100);
           }
        }
        if(cnRST=="1") {
           pzem.resetEnergy(); //reset energy
           delay(500);
           cnRST="0";
        }
     }
   }
}
```

C. Listing Program *Index* HTML *Web Interface*

```
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="data:,">
  <title>ESP32 FADIL</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

    <h2>MONITORING<br/> METER LISTRIK TERSEGEL</h2>
    <strong>Kondisi Terahir Diterima:<br/>
    <span id="timestamp" class="warna">%TIMESTAMP% </span></strong>

   <div class="content">
    <div class="card-grid">
      <div class="card">
      <p>
       Tegangan: <span id="tegangan" class="readings">%V1%</span><sup>
Volt</sup> <br/>
       Arus: <span id="arus" class="readings">%A1%</span><sup> Amper</sup>
<br/>
       Pemakaian: <span id="energi" class="readings">%E1%</span> <sup>
kWh</sup><br/>
       LED: <span id="stLED" class="readings">%L1%</span> <br/>
         <a href="/LedON"><button class="button">ON</button></a>
         <a href="/LedOFF"><button class="button
button2">OFF</button></a><br/>

       <span id="RSt" </span> <a href="/RESET"><button class="button">RESET
kWH</button></a>


      </p>
     </div>
    </div>
  </div>


<script>
setInterval(updateValues, 500, "tegangan");
setInterval(updateValues, 500, "arus");
```

```
setInterval(updateValues, 500, "energi");
setInterval(updateValues, 500, "stLED");
setInterval(updateValues, 500, "timestamp");

function updateValues(value) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById(value).innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/" + value, true);
  xhttp.send();
}
</script>
</body>
</html>
```

D. Listing Program *Style Web Interface*

```
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
  text-align: center;
  /*
  background-image: url('bgimage.jpg');
  background-repeat: no-repeat;
  background-size: cover;
  background-position: center;
  */
}

h2 {font-size: 1.5rem; }
p { font-size: 1.2rem; }

.units { font-size: 1.2rem; }
.readings { font-size: 1.5rem; color: red;}
.warna {color: blue; }
.button {
  display: inline-block;
  background-color: #b30000; //red color
  border: none;
  border-radius: 4px;
```

```
  color: white;
  padding: 10px 15px;
  text-decoration: none;
  font-size: 20px;
  margin: 2px;
  cursor: pointer;
}
.button2 { background-color: #364cf4; //blue color }
.button3 { background-color: green; //blue color }
.content { padding: 5px;}
.card-grid {
  max-width:350px;
  margin: 0 auto;
  display: grid;
  grid-gap: 2rem;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
}
.card {
  /*background-color: white;*/
  box-shadow: 2px 2px 12px 1px rgba(140,140,140,.5);
}
```

Lampiran – 2 Dokumentasi Alat

A. Dokumentasi Alat *Node Server*



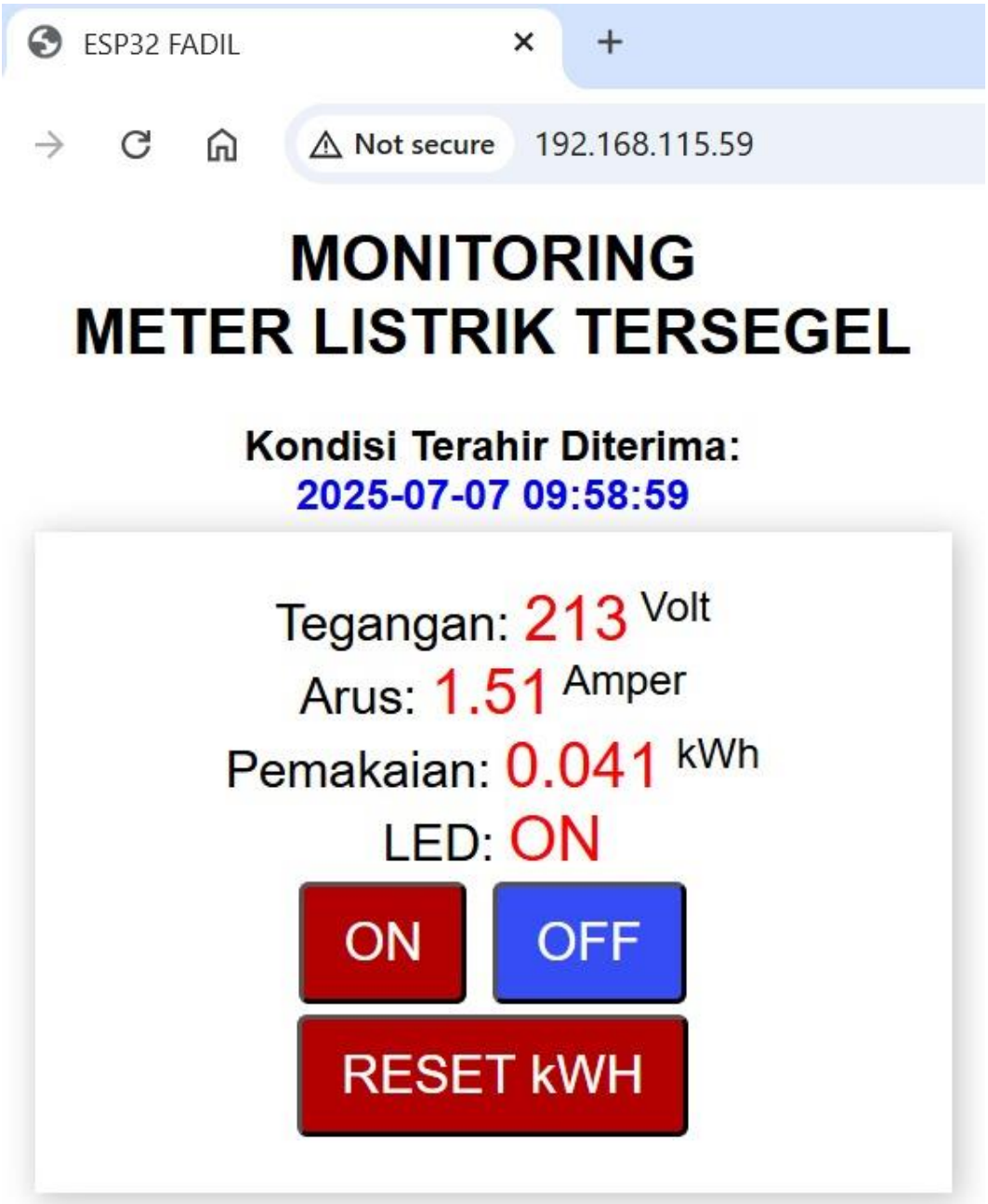


B. Dokumentasi Alat *Node* Pelanggan

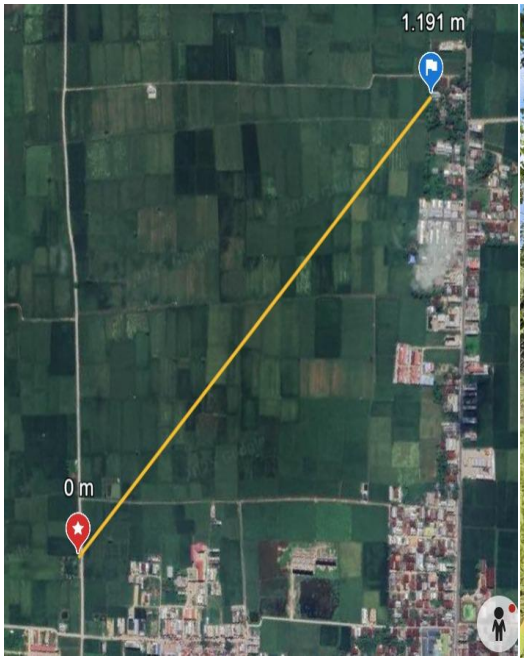




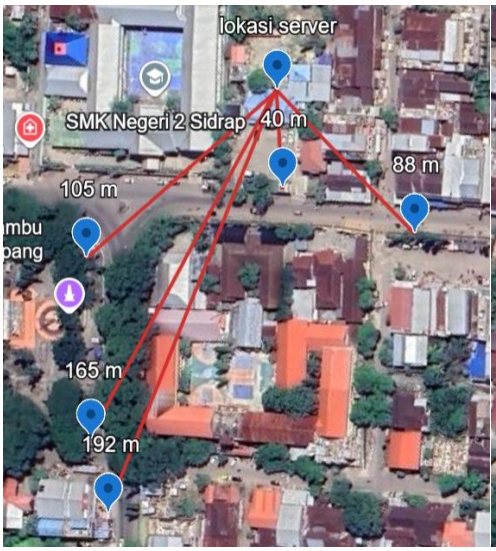
Lampiran – 3 Tampilan Aplikasi *Web Interface*

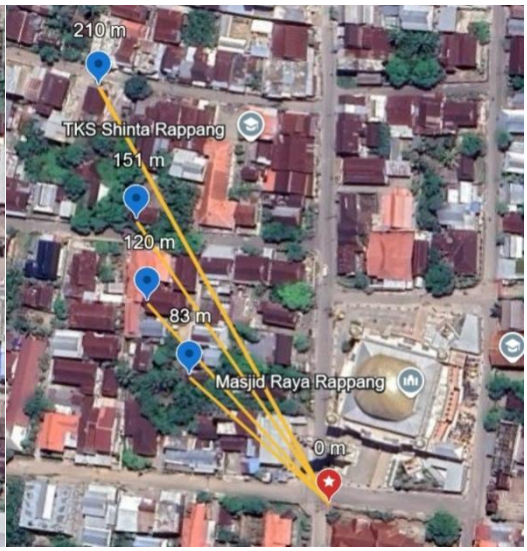


Lampiran – 4 Foto Lokasi Uji

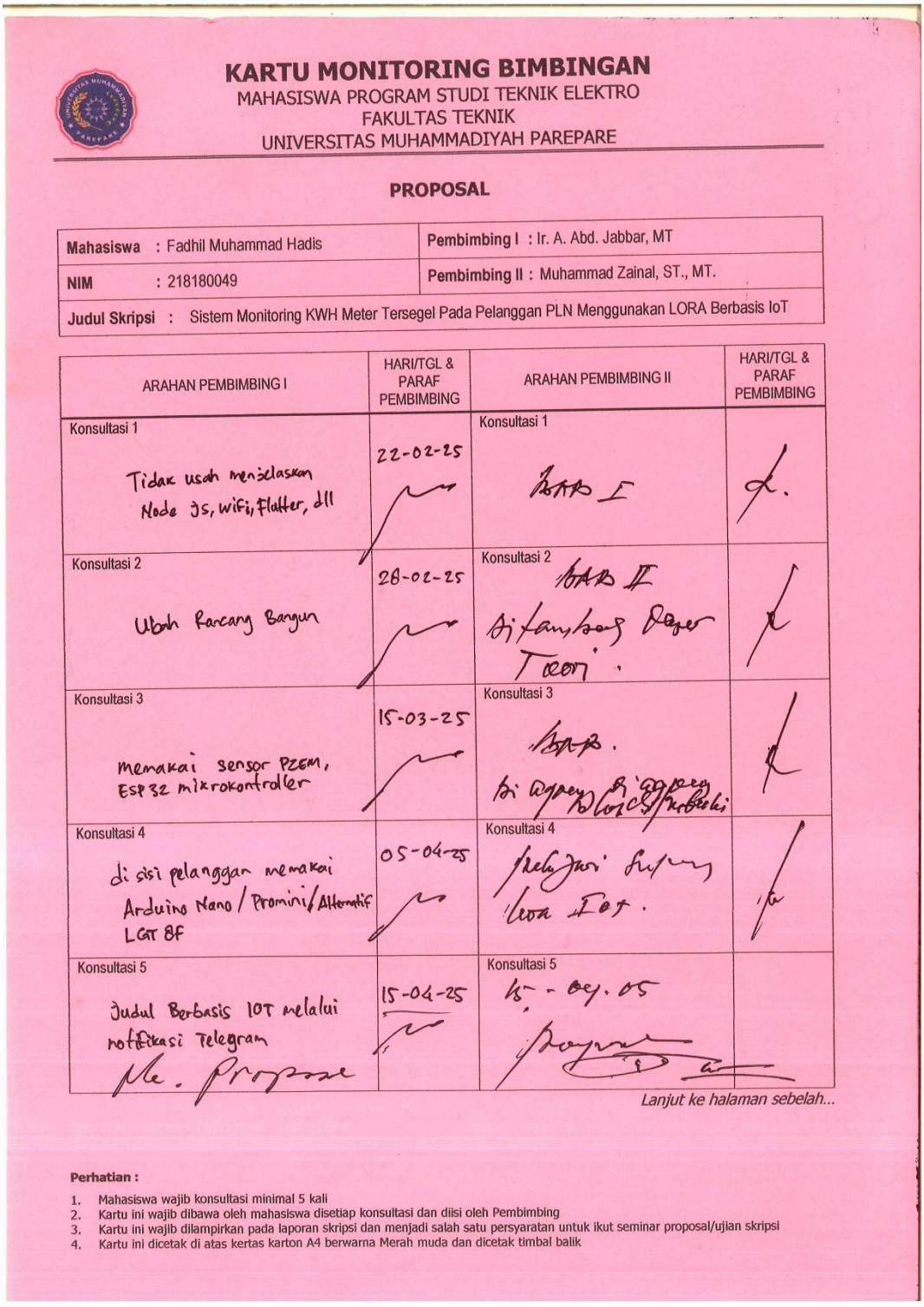








Lampiran-5 Kartu Kontrol Proposal



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : Fadhil Muhammad Hadis	**Pembimbing I** : Ir. A. Abd. Jabbar, MT
NIM : 218180049	**Pembimbing II** : Muhammad Zainal, ST., MT.
Judul Skripsi : Sistem Monitoring KWH Meter Tersegel Pada Pelanggan PLN Menggunakan LORA Berbasis IoT	

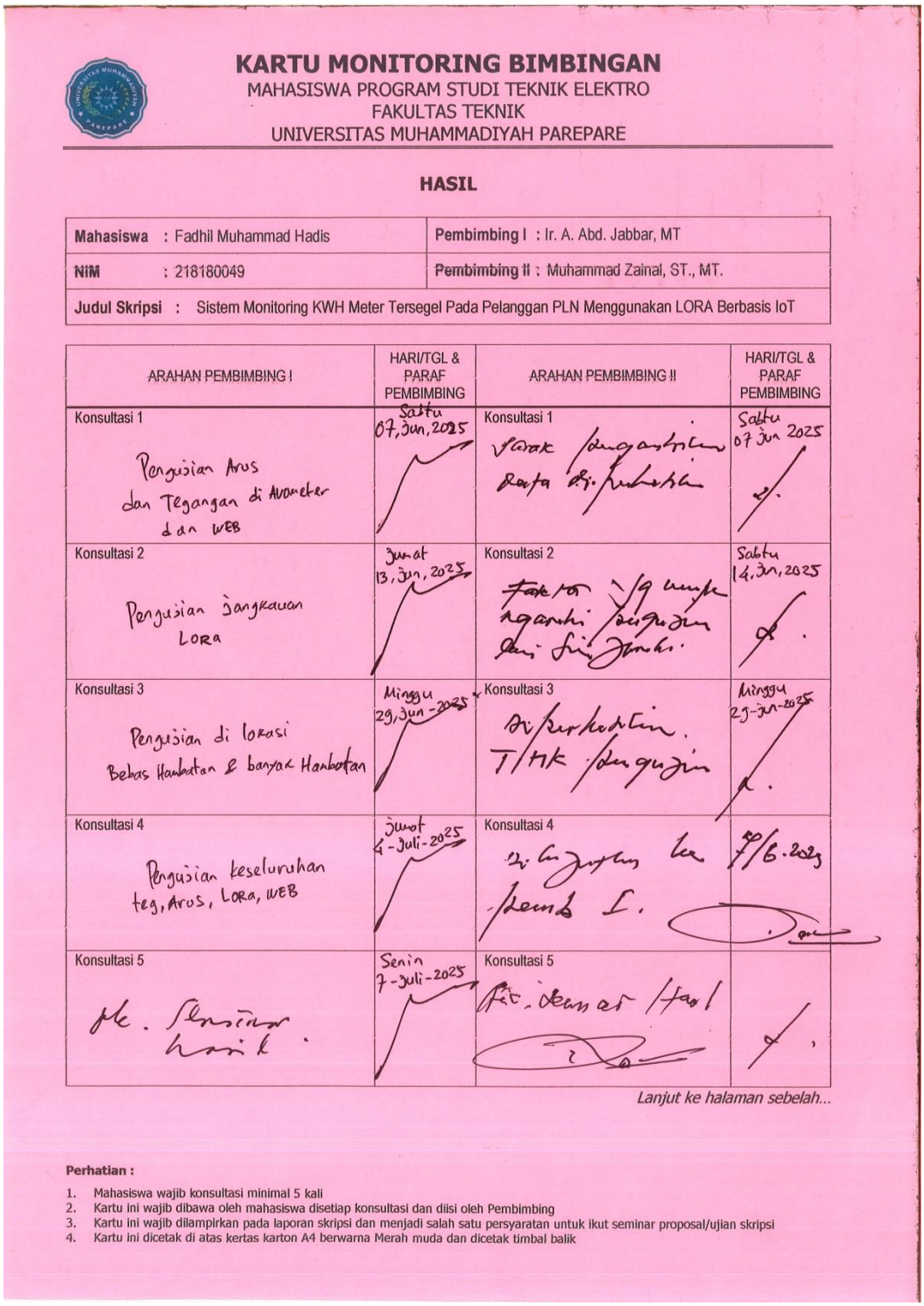
ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Tidak usah menjelaskan Node JS, wifi, flatter, dll	22-02-25	Konsultasi 1 BAB I	
Konsultasi 2 Ubah Rancang Bangun	28-02-25	Konsultasi 2 BAB II ditambahkan Dasar Teori.	
Konsultasi 3 memakai sensor PZEM, ESP32 mikrokontroller	15-03-25	Konsultasi 3 BAB.	
Konsultasi 4 di sisi pelanggan memakai Arduino Nano / Promini / Alternatif LGT 8F	05-04-25	Konsultasi 4 Pelajari Sistem Lora IoT.	
Konsultasi 5 Judul Berbasis IOT melalui notifikasi Telegram Ke. Proposal	15-04-25	Konsultasi 5 ACC. 05	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Merah muda dan dicetak timbal balik

LAMPIRAN 6- KARTU KONTROL HASIL



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : Fadhil Muhammad Hadis	**Pembimbing I** : Ir. A. Abd. Jabbar, MT
NiM : 218180049	**Pembimbing II** : Muhammad Zainal, ST., MT.
Judul Skripsi : Sistem Monitoring KWH Meter Tersegel Pada Pelanggan PLN Menggunakan LORA Berbasis IoT	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Pengisian Arus dan Tegangan di Avometer dan WEB	Sabtu 07, Jun, 2025	Konsultasi 1 Revisi dengan data yg. sebenarnya	Sabtu 07 Jun 2025
Konsultasi 2 Pengisian jangkauan LORA	Jumat 13, Jun, 2025	Konsultasi 2 Tolong diganti dengan data yang sebenarnya	Sabtu 14, Jun, 2025
Konsultasi 3 Pengisian di lokasi Bebas Hambatan & banyak Hambatan	Minggu 29, Jun - 2025	Konsultasi 3 Di perbaiki. ACC untuk diujikan	Minggu 29-Jun-2025
Konsultasi 4 Pengisian keseluruhan teg, Arus, LORA, WEB	Jumat 4 - Juli - 2025	Konsultasi 4 Di lanjutkan ke Seminar Hasil.	7/6.2025
Konsultasi 5 Ke. Seminar hasil.	Senin 7 - Juli - 2025	Konsultasi 5 Tit. dengan Hasil	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Merah muda dan dicetak timbal balik

LAMPIRAN 7- KARTU KONTROL SKRIPSI

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Fadhil Muhammad Hadis	**Pembimbing I : Ir. A. Abd. Jabbar, MT.**
NIM : 218180049	**Pembimbing II : Muhammad Zainal, ST., MT.**
Judul Skripsi : Sistem Monitoring KWH Meter Tersegel Pada Pelanggan PLN Menggunakan LoRa Terintegrasi Melalui Website Berbasis IOT	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Perbaiki latar belakang	13-Juli-2025	Konsultasi 1 Perbaiki latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, BAB II, III, IV.	13-Juli-2025
Konsultasi 2 ubah rumusan masalah	13-Juli 2025	Konsultasi 2	
Konsultasi 3 ubah batasan masalah	13-Juli-2025	Konsultasi 3	
Konsultasi 4 berikan penjelasan terkait pencurian Listrik	14-Juli-2025	Konsultasi 4	
Konsultasi 5 Ok. Tutup	14-Juli-2025	Konsultasi 5 Ke ujian Skripsi 14/8-2025	14-Juli-2025

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswadisetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Merah muda dan dicetak timbal balik