LAMPIRAN

Lampiran 1. Gambar proses pengetesan

1. Pengetesan respon PWM terhadap tegangan



A **B**

Gambar 1. Pengetesan respon PWM (A, B)

Gambar diatas adalah gambar proses pengetesan respon PWM terhadap tegangan, gambar diatas proses pengetesan ialah pada gambar A pada posisi PWM 42 dengan respon tegangan 4,99V dan pada gambar B pada PWM 108 dengan respon tegangan 11,42V.

2. Pengetesan respon tegangan terhadap *setpoint* dan akurasi tegangan



A **B**

Gambar 2. Pengetesan tegangan terhadap *setpoint* (A, B)

Proses pengetesan respon tegangan terhadap *setpoint,* gambar A adalah gambar saat menentukan *setpoint* dan gambar B hasil dari pengetesan yang di ambil dari serial monitor.

3. Gambar pengetesan tegangan *output* terhadap alat ukur



A **B**

Gambar 3. Pengetesan tegangan terhadap alat ukur (A, B)

Proses pengetesan tegangan *output* terhadap alat ukur diatas menggunakan alat ukur Avometer, dari gambar diatas gambar A gambar pemantaua tegangan alat melalui LCD dan gambar B gambar pengukuran dengan menggunakan Avometer.

4. Gambar pengetesan tegangan *output* terhadap beban kipas



A



B

Gambar 4. Pengetesan tegangan terhadap kipas (A, B)

Proses pengetesan tegangan *output* terhadap beban kipas, gambar A menunjukan respon kipas saat diberi tegangan dan gambar B proses keseluruhan dalam pengetesan.

5. Gambar pengetesan tegangan *output* terhadap beban lampu



A



B

Gambar 5. Pengetesan tegangan terhadap lampu (A, B)

Proses pengetesan tegangan *output* terhadap beban lampu, gambar A menunjukan respon lampu saat diberi tegangan oleh alat dan pada gambar B proses keseluruhan pengetesan.

Lampiran 2. Tabel Respon Tegangan Terhadap Setpoint

Tabel dibawah ini adalah tabel yang digunakan pada sub bab 4.4.2 pengujian respon tegangan terhadap setpoint 1 sampai 24. Pada tabel dibawah menunjukan respond tegangan per setpoint yang dimasukan menjadi gambar grafik pada sub bab 4.4.2.

No	SV	PV
1	1	1,38
2	1	1,09
3	1	1,09
4	1	1,38
5	1	1,66
6	1	1,95
7	1	1,68
8	1	1,41
9	1	1,14
10	1	1,42

No	SV	PV
1	2	2,74
2	2	2,79
3	2	2,83
4	2	2,79
5	2	2,71
6	2	2,62
7	2	2,34
8	2	2,05
9	2	1,76
10	2	1,48

No	SV	PV
1	3	2,88
2	3	3,21
3	3	3,10
4	3	3,10
5	3	3,20
6	3	2,98
7	3	3,15
8	3	3,12
9	3	3,13
10	3	3,19

No	SV	PV
1	4	3,65
2	4	3,71
3	4	3,78
4	4	3,86
5	4	3,94
6	4	3,96
7	4	3,97
8	4	3,98
9	4	3,98
10	4	3,98

No	SV	PV
1	5	4,76
2	5	4,83
3	5	4,89
4	5	4,95
5	5	5,01
6	5	5,06
7	5	5,06
8	5	5,05
9	5	5,04
10	5	5,00

No	SV	PV
1	6	5,84
2	6	5,90
3	6	5,96
4	6	6,01
5	6	6,04
6	6	6,03
7	6	6,02
8	6	6,02
9	6	6,00
10	6	5,98

No	SV	PV
1	7	6,83
2	7	6,91
3	7	6,95
4	7	7,00
5	7	7,05
6	7	7,06
7	7	7,07
8	7	7,09
9	7	7,07
10	7	7,04

No	SV	PV
1	8	7,82
2	8	7,90
3	8	7,93
4	8	7,96
5	8	8,01
6	8	8,02
7	8	8,02
8	8	8,03
9	8	8,01
10	8	7,98

No	SV	PV
1	9	8,66
2	9	8,77
3	9	8,88
4	9	8,97
5	9	9,01
6	9	9,04
7	9	9,07
8	9	9,06
9	9	9,04
10	9	9,05

No	SV	PV
1	10	9,79
2	10	9,90
3	10	9,96
4	10	10,01
5	10	10,07
6	10	10,10
7	10	10,09
8	10	10,09
9	10	10,10
10	10	10,08

No	SV	PV
1	11	10,84
2	11	10,88
3	11	10,93
4	11	10,97
5	11	10,98
6	11	10,99
7	11	11,00
8	11	11,01
9	11	11,02
10	11	11,03

No	SV	PV
1	12	11,86
2	12	11,92
3	12	11,95
4	12	11,98
5	12	12,01
6	12	12,01
7	12	12,02
8	12	12,02
9	12	12,01
10	12	12,00

No	SV	PV
1	13	12,81
2	13	12,87
3	13	12,93
4	13	12,98
5	13	12,99
6	13	13,00
7	13	13,03
8	13	13,04
9	13	13,04
10	13	13,04

No	SV	PV
1	14	13,82
2	14	13,89
3	14	13,92
4	14	13,95
5	14	13,97
6	14	13,99
7	14	13,97
8	14	13,96
9	14	13,97
10	14	13,96

No	SV	PV
1	15	14,76
2	15	14,86
3	15	14,96
4	15	14,99
5	15	15,03
6	15	15,07
7	15	15,07
8	15	15,07
9	15	15,07
10	15	15,05

No	SV	PV
1	16	15,83
2	16	15,89
3	16	15,91
4	16	15,93
5	16	15,96
6	16	15,96
7	16	15,97
8	16	15,97
9	16	15,97
10	16	15,97

No	SV	PV
1	17	16,77
2	17	16,86
3	17	16,94
4	17	16,99
5	17	17,04
6	17	17,06
7	17	17,05
8	17	17,04
9	17	17,05
10	17	17,03

No	SV	PV
1	18	17,77
2	18	17,86
3	18	17,91
4	18	17,97
5	18	18,02
6	18	18,04
7	18	18,04
8	18	18,04
9	18	18,05
10	18	18,03

No	SV	PV
1	19	18,85
2	19	18,92
3	19	18,94
4	19	18,96
5	19	19,00
6	19	19,01
7	19	19,01
8	19	19,01
9	19	19,01
10	19	19,01

No	SV	PV
1	20	19,75
2	20	19,86
3	20	19,95
4	20	19,98
5	20	20,01
6	20	20,06
7	20	20,06
8	20	20,07
9	20	20,06
10	20	20,03

No	SV	PV
1	21	20,75
2	21	20,86
3	21	20,91
4	21	20,97
5	21	21,02
6	21	21,04
7	21	21,03
8	21	21,02
9	21	21,03
10	21	21,02

No	SV	PV
1	22	21,70
2	22	21,79
3	22	21,88
4	22	21,92
5	22	21,95
6	22	21,99
7	22	21,99
8	22	22,00
9	22	22,00
10	22	22,01

No	SV	PV
1	23	22,93
2	23	23,07
3	23	23,20
4	23	23,12
5	23	23,11
6	23	23,09
7	23	23,15
8	23	23,11
9	23	23,17
10	23	23,14

No	SV	PV
1	24	24,15
2	24	24,11
3	24	24,08
4	24	24,04
5	24	24,01
6	24	23,99
7	24	23,96
8	24	23,97
9	24	23,99
10	24	24,00

Data pada tebel diatas ialah data yang digunakan pada gambar pengujian stabilitas keluaran tegangan terhadap *setpoint,* data diatas diambil dari rentang tegangan 1 sampai 24V dengan masing masing 10 sampel.

Lampiran 3. Program Arduino

```
// ===== LCD =====
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

// ===== CONTROLLER =====
int runMode = 0, stateButton = 0;
int pvVolt, svVolt = 000, valuePwm;
String inputNominal;
#define pinPwm 10
unsigned long lastAdd;

// ====== TEGANGAN =====
#define maxN 10
unsigned long totalVolt;
int dataVolt[maxN], n;
int meanVolt;

// ===== KEYPAD =====
#include <Keypad.h>
const byte row = 4;
const byte col = 3;
const byte pinRow[row] = {3, 4, 5, 6};
const byte pinCol[col] = {7, 8, 9};
char keys[row][col] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};
Keypad keypad = Keypad(makeKeymap(keys), pinRow, pinCol, row, col);
char keyPushed;

// ===== MODBUS =====
#include <SoftwareSerial.h>
SoftwareSerial pzem(2, 11); // RX, TX
byte perintah[] = {0x01, 0x04, 0x00, 0x00, 0x00, 0x08};//readInputRegisters
byte bufferDataModbus[100];
byte *ptr;
bool getVoltages = false;

// ===== DISPLAY =====
#define durDisplay 100
#define durBlink 500
#define durPush 250
unsigned long lastDisplay, lastBlink, lastPush;
```

```
bool stateBlink;
char buff[16];
int cnt = 0;

void setup() {
  Serial.begin(9600); delay(150);
  Serial.println("Inisialisasi...");
  // ===== INIT LCD =====

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0); lcd.print("  Power Supply  ");
  lcd.setCursor(0, 1); lcd.print("    Variable    ");

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0); lcd.print("    Andyfha     ");
  lcd.setCursor(0, 1); lcd.print("Syahroni Irwana ");


  // ===== INIT KEYPAD =====
  keypad.setDebounceTime(20);

  // ===== INIT OUTPUT =====
  pinMode(pinPwm, OUTPUT);

  // ===== INIT PZEM =====
  pzem.begin(9600);
  ptr = bufferDataModbus;
  delay(1500);
}

void loop() {
  if (runMode == 0) {
    getVoltage();
    if(getVoltages==true)setOutput();
    updateLCD();

    keyPushed = keypad.getKey();
    if (keyPushed != NO_KEY) {
      runMode = 1;
      blinkLCD(1, 250);
      lcd.clear();
    }
  }
  else {
    modeInput();
```

```
  }
  delay(1);
}

void modeInput() {
  if (millis() > lastBlink + durBlink) {
    lastBlink = millis();
    lcd.setCursor(inputNominal.length() + 7, 1);
    if (stateBlink)lcd.print(char(255)); else lcd.print("        ");
    stateBlink = !stateBlink;
    lcd.setCursor(0, 0); lcd.print("Input Voltage : ");
    lcd.setCursor(12, 1); lcd.print("  V ");
  }

  if (millis() > lastPush + durPush) stateButton = 0;
  if (stateButton == 0) {
    keyPushed = keypad.getKey();
    if (keyPushed != NO_KEY) stateButton = 1;
  }
  if (stateButton == 1) {
    lastPush = millis();
    stateButton = 2;

    if (keyPushed == '*') {
      inputNominal = inputNominal.substring(0, inputNominal.length() - 1);
    }
else if (keyPushed == '#') {
      if (inputNominal.length() == 0, 9 && inputNominal.substring(0).toInt()>0)
      if (inputNominal.length() == 0, 9 && inputNominal.substring(0).toInt()<2)
      {
        lcd.setCursor(0, 0); lcd.print("  tidak setabil  ");
        lcd.setCursor(0, 1); lcd.print("   tegangan 1V   ");
        inputNominal = "1";
        blinkLCD(5, 100);
        delay(1000);("  tidak setabil  ");
      }
      lcd.setCursor(0,0);lcd.print("MENYESUAIKAN...    ");
      svVolt = inputNominal.toInt()*1000;

      if (inputNominal.length() == 0, 9 && inputNominal.substring(0).toInt()>1)
      if (inputNominal.length() == 0, 9 && inputNominal.substring(0).toInt()<3)
      {
        lcd.setCursor(0, 0); lcd.print("  tidak setabil  ");
        lcd.setCursor(0, 1); lcd.print("   tegangan 2V   ");
        inputNominal = "2";
        blinkLCD(5, 100);
        delay(1000);("  tidak setabil  ");
```

```
      }
      lcd.setCursor(0,0);lcd.print("MENYESUAIKAN...    ");
      svVolt = inputNominal.toInt()*1000;

      if (inputNominal.length() == 1, 9 && inputNominal.substring(0).toInt() > 24)
       {
        lcd.setCursor(0, 0); lcd.print("     Max 24V     ");
        lcd.setCursor(0, 1); lcd.print("  kembali ke 0V  ");
        inputNominal = "0";
        blinkLCD(5, 100);
        delay(1000);
      }
            lcd.setCursor(0,0);lcd.print("MENYESUAIKAN...    ");
      svVolt = inputNominal.toInt()*1000;
      runMode = 0;
      inputNominal = "";
      unsigned long tempSv = 2*pow(10,-05)*pow(svVolt,2) + 0.4717*svVolt -
536.69;//PERSAMAAN SETELAH KALIBRASI
      valuePwm = map(tempSv,0,24000,0,255);
      analogWrite(pinPwm, valuePwm);
      for(int i=0;i<10;i++){
        getVoltage();
      }
      blinkLCD(2, 250);
    }

    else {
      if (inputNominal.length() == 5)blinkLCD(1, 300);
      else  inputNominal += keyPushed;
      //      Serial.println(inputNominal);
    }
    lcd.setCursor(7, 1); lcd.print(inputNominal);
  }
}
void updateLCD() {
  if (millis() > lastDisplay + durDisplay) {
    Serial.println(String(svVolt) + "," + String(meanVolt));
    sprintf(buff, " PV :  %5d mV%s", meanVolt,getVoltages?" ":"*");
    lcd.setCursor(0, 0); lcd.print(buff);

    sprintf(buff, " SV :  %5d V ", svVolt/1000);
    lcd.setCursor(0, 1); lcd.print(buff);
  }
}
void setOutput() {
  int error = svVolt - pvVolt;
  float p = 1 + abs(error / 800); // Faktor proporsional
```

```
  // Penstabil pada setpoint 1 dan 2 volt
  if (svVolt == 1000 || svVolt == 2000) {
    if (abs(error) < 50)
    if (abs(error) <-50) {
      // Kurangi efek proporsional agar tegangan stabil di sekitar setpoint
      p = 0.5;
    }
  }

  // Logika kontrol PWM
  if (error > 50)     valuePwm += p;
  if (error < -50)    valuePwm -= p;
  if (valuePwm > 255) valuePwm = 255;
  if (valuePwm < 0)   valuePwm = 0;

  analogWrite(pinPwm, valuePwm);
}
void getVoltage() {
  uint16_t crc = calcCRC(perintah, sizeof(perintah));
  delay(1);
  pzem.write(perintah, sizeof(perintah));
  pzem.write(lowByte(crc));
  pzem.write(highByte(crc));
  delay(10);

  long millisResponModbus = millis() + 100;
  while (!pzem.available())
  {
    if (millisResponModbus < millis())
    {
      break;//timeout
    }
  }

  while (pzem.available())
  {
    byte b = pzem.read();
    *ptr++ = b;
    delay(2);
  }
  //  lcd.setCursor(0,0);
  if (memcmp(bufferDataModbus, perintah, 2) == 0) {
    ptr = bufferDataModbus;
    pvVolt = ((ptr[0 + 3] << 8) + ptr[1 + 3]) * 10;

    memset(bufferDataModbus, 0x00, sizeof(bufferDataModbus));
```

```
    dataVolt[n] = pvVolt;
    totalVolt = 0;
    for (int i = 0; i < maxN; i++) {
      totalVolt += dataVolt[i];
    }
    meanVolt = totalVolt / maxN;
    //    pvVolt = meanVolt;
    n++; if (n >= maxN)n = 0;

    getVoltages = true;
  }
  else getVoltages = false;
}
uint16_t calcCRC(byte *data, byte panjang) {
  int i;
  uint16_t crc = 0xFFFF;
  for (byte p = 0; p < panjang; p++)
  {
    crc ^= data[p];
    for (i = 0; i < 8; ++i)
    {
      if (crc & 1)
        crc = (crc >> 1) ^ 0xA001;
      else
        crc = (crc >> 1);
    }
  }
  return crc;
}
void blinkLCD(int looping, int dur) {
  for (; looping > 0; looping--) {
    lcd.noBacklight(); delay(dur);
    lcd.backlight(); delay(dur);
  }
}
```