

# Sistem Komunikasi Antar Arduino Menggunakan Protokol RS485

Nurul Septianti<sup>1</sup>, Reni Rahmadewi<sup>2</sup>

<sup>1</sup> Teknik Elektro, Universitas Singaperbangsa Karawang, Karawang, Indonesia

<sup>2</sup> Teknik Elektro, Universitas Singaperbangsa Karawang, Karawang, Indonesia

## Article Info

### Article history:

Diterima 23 Maret 2024

Revisi 22 April 2024

Diterbitkan 23 April 2024

### Keywords:

Sistem Komunikasi

RS485

Master

Slave

Daya

## ABSTRAK

Kerja Praktik ini bertujuan untuk merancang Sistem Komunikasi Antar Arduino menggunakan Protokol RS485. Tujuannya adalah untuk mengidentifikasi komponen-komponen sistem tersebut, dan memahami cara kerja dari sistem komunikasi tersebut. Metode penelitian mencakup identifikasi masalah, perancangan alat, pembuatan alat, pengujian alat, dan analisis data. Pengumpulan data dalam Kerja Praktik ini menggunakan metode searching, observasi langsung, dan literatur. Sistem Komunikasi Antar Arduino menggunakan Protokol RS485 dirancang untuk mendukung komunikasi jarak jauh, walaupun pada percobaan digunakan kabel dengan panjang sekitar  $\pm 25$ cm. Sistem Komunikasi Antar Arduino menggunakan Protokol RS485 melibatkan mikrokontroler master sebagai pusat kendali yang mengirimkan perintah kepada mikrokontroler slave dan mengakses input dari push button, hasilnya ditampilkan pada layar LCD. Mikrokontroler slave berperan dalam menerima dan menjalankan perintah dari mikrokontroler master, dengan data terkait aliran daya listrik pada masing-masing slave (880 watt pada slave 1, 900 watt pada slave 2, dan 800 watt pada slave 3).

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Nurul Septianti

Universitas Singaperbangsa Karawang, Karawang 41361, Indonesia

Email: nurulseptianti01@gmail.com

## 1. PENDAHULUAN

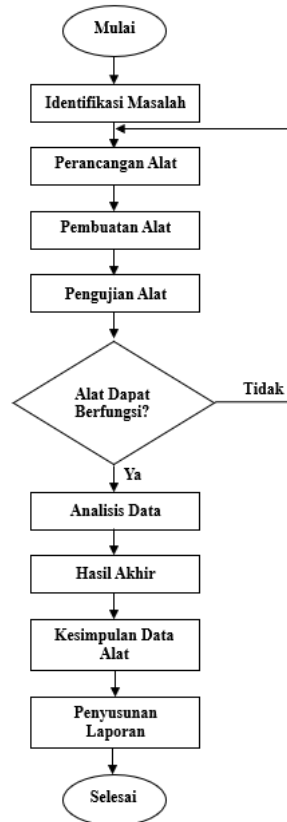
Kerja Praktik (KP) adalah suatu bentuk penerapan sistematis antara kurikulum akademis dan praktik di tempat kerja untuk mencapai penguasaan keterampilan khusus. Dengan adanya Kerja Praktik diharapkan mahasiswa mampu memahami dan juga mengembangkan teori-teori yang sudah dipelajari sehingga dapat di terapkan di dunia usaha maupun dunia industri [1].

Dalam konteks pengaplikasian teori ke dalam praktik, penting untuk memilih teknologi yang dapat diterapkan dengan efektif. Dalam hal ini, teknologi RS485 menjadi penting karena memungkinkan komunikasi data serial yang handal dan efisien dalam berbagai aplikasi praktis. RS485 adalah teknik komunikasi data serial yang dikembangkan di tahun 1983 dimana dengan teknik ini, komunikasi data dapat dilakukan pada jarak yang cukup jauh yaitu 1,2 Km. Berbeda dengan komunikasi serial RS232 yang mampu berhubungan secara one to one, maka komunikasi RS485 selain dapat digunakan untuk komunikasi multidrop yaitu berhubungan secara one to many dengan jarak yang jauh teknik ini juga dapat digunakan untuk menghubungkan 32 unit beban sekaligus hanya dengan menggunakan dua buah kabel saja tanpa memerlukan referensi ground yang sama antara unit yang satu dengan unit lainnya [2].

Pada penelitian ini, jenis komunikasi yang digunakan adalah Komunikasi Half Duplex. Komunikasi Half Duplex adalah media komunikasi dua arah di mana pengiriman dan penerimaan informasi terjadi secara bergantian. Jadi saat terjadi komunikasi antara A dan B. Saat A mengirim informasi (berbicara) maka B akan menerima informasi (mendengarkan). Demikian terjadi proses yang sebaliknya (vice versa). Contoh media yang menggunakan media ini adalah radio walkie talkie [3], di mana perangkat akan beralih antara mode pengiriman dan penerimaan pesan secara bergantian sesuai dengan kebutuhan komunikasi.

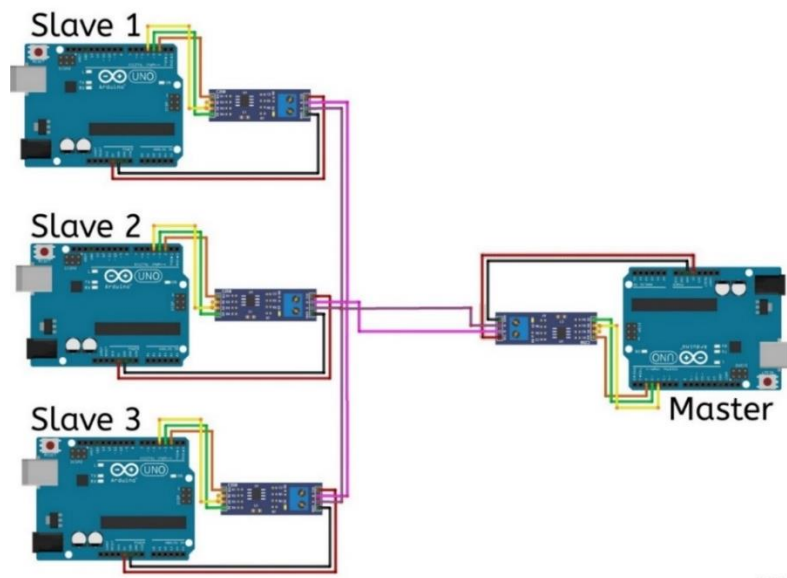
Dengan fokus pada penggunaan RS485, penulis memilih topik Komunikasi antar Arduino dengan RS485 untuk dijadikan Laporan Kerja Praktik dengan judul “Sistem Komunikasi Antar Arduino Menggunakan Protokol RS485”. Dengan harapan dapat memperbanyak pengalaman, pengetahuan praktikal, dan pemahaman teknis mahasiswa dalam dunia industri.

## 2. METODE



Gambar 1. Alur Penelitian

### 2.1. Perancangan Alat



Gambar 2. Rancangan Sistem Komunikasi Antar Arduino Menggunakan Protokol RS485

Rancangan dan Rangkaian Sistem Komunikasi Antar Arduino Menggunakan Protokol RS485 ditunjukkan pada gambar 2 meliputi beberapa bagian yaitu:

1. Mikrokontroler master bertindak sebagai pusat pengendalian data dengan mengirimkan perintah melalui mikrokontroler slave dan menjalankan input dari push button serta menampilkan informasinya pada LCD.
2. Mikrokontroler slave bertugas menerima perintah dari mikrokontroler master untuk mengontrol data secara langsung.
3. Push Button berperan sebagai sakelar yang dapat membuka dan menutup aliran listrik. Push button juga berfungsi sebagai penggerak untuk mentransmisikan dan menerima instruksi dari mikrokontroler master.
4. LCD Display berfungsi sebagai tampilan informasi untuk data yang sedang dikirim atau diterima.

## 2.2. Analisis Kebutuhan

### 2.2.1. Alat dan bahan

- a. Laptop (2 Unit)
- b. Software Arduino IDE
- c. Arduino Uno (4 buah)
- d. RS485 (4 buah)
- e. Push Button (4 buah)
- f. Project Board/ Breadboard (2 buah)
- g. Liquid Crystal Display (LCD) 20x4 (1 buah)
- h. Kabel Jumper (Secukupnya)
- i. Kabel USB (4 buah)
- j. Kabel AWG 22 (2 buah : 25cm)

### 2.2.1. Interkoneksi Komponen

Untuk membuat sistem rangkaian berjalan dengan baik, dibutuhkan beberapa interkoneksi komponen yang penting. Berikut ini adalah tabel interkoneksi komponen arduino master, arduino slave, RS485, push button, dan LCD:

Tabel 1. Interkoneksi Komponen Arduino Master, Arduino Slave, Rs485, Push Button, Lcd

No.	Komponen	Pin	Terhubung ke
1.	Arduino Uno(Warna Hitam : Master)	2 (RX)	Pin RO RS485
		3 (TX)	Pin DI RS485
		4	Pin DE dan RE RS485
		5V	Pin VCC (5V) RS485
		GND	Pin GND (0V) RS485
2.	Arduino Uno(Warna Biru : Slave 1&3)	2 (RX)	Pin RO RS485
		3 (TX)	Pin DI RS485
		4	Pin DE dan RE RS485
		5V	Pin VCC (5V) RS485
		GND	Pin GND (0V) RS485
3.	RS485	DI (Driver Input : TX Pin)	Pin 3(TX) Arduino Uno Master & Slave 1-3
		DE (Driver Enable : High-Enable)	Pin 4 Arduino Uno Master & Slave 1-3
		RE (Receiver Enable : Low-Enable)	Pin 4 Arduino Uno Master & Slave 1-3
		RO (Receiver Output (RX Pin)	Pin 2(RX) Arduino Uno Master & Slave 1-3
		GND (0V)	Pin GND Arduino Uno Master & Slave 1-3
		A (Non-inverting receiver input) (Non-inverting Driver Output)	Pin A RS485 1-4
		B (Inverting Receiver Input) (Inverting Driver Output)	Pin B RS485 1-4
VCC (5V)	Pin VCC Arduino Uno Master & Slave 1-3		
4.	LCD 20x4 dengan Modul I2c	GND (0V)	Pin GND (0V) Arduino Uno Master
		VCC	Pin VCC (5V) Arduino Uno Master
		SDA	Pin SDA Arduino Uno Master
		SCL	Pin SCL Arduino Uno Master
5.	Push Button 1 (Slave 1 Data 1)	Pin 1	Pin 5 Arduino Uno Master
		Pin 2	GND

6.	Push Button 2 (Slave 2 Data 2)	Pin 1 Pin 2	Pin 6 Arduino Uno Master GND
7.	Push Button 3 (Slave 3 Data 3)	Pin 1 Pin 2	Pin 8 Arduino Uno Master GND
8.	Push Button 4 (Reset)	Pin 1 Pin 2	Pin 7 Arduino Uno Master GND

### 2.3. Langkah Percobaan

- Siapkan alat dan bahan yang akan digunakan, lalu susun sesuai dengan gambar 2 dan gambar 3 dengan mengikuti panduan penyambungan pin pada tabel 1.
- Buatlah program master pada software arduino. Berikut programnya:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>           // menambahkan library LCD I2C
#include <SoftwareSerial.h>
#define enTxPin 4                       // HIGH: TX and LOW: RX
SoftwareSerial RS485Serial(2, 3);       // RX, TX
LiquidCrystal_I2C lcd(0x3f, 20, 4);     // set alamat I2C dan ukuran LCD
void setup() {
  lcd.init();                          // Inisiasi LCD
  lcd.backlight();                     // menyalakan backlight
  Serial.begin(9600);
  Serial.println("Starting System");
  RS485Serial.begin(9600);
  pinMode(enTxPin, OUTPUT);
  digitalWrite(enTxPin, HIGH);         // default TX
  pinMode(5, INPUT_PULLUP);
  pinMode(6, INPUT_PULLUP);
  pinMode(8, INPUT_PULLUP);
  pinMode(7, INPUT_PULLUP);
}
void loop() {
  if (digitalRead(5) == 0) {
    lcd.clear();                       // menghapus tampilan LCD
    RS485Serial.print("AR1%");
    digitalWrite(enTxPin, LOW);        // change pin to RX
    Serial.println("waiting data");
    if (RS485Serial.available()) {
      data();
    }
  }
  else if (digitalRead(6) == 0) {
    lcd.clear();                       // menghapus tampilan LCD
    RS485Serial.print("AR2%");
    digitalWrite(enTxPin, LOW);        // change pin to RX
    Serial.println("waiting data");
    if (RS485Serial.available()) {
      data();
    }
  }
  else if (digitalRead(8) == 0) {
    lcd.clear();                       // menghapus tampilan LCD
    RS485Serial.print("AR3%");
    digitalWrite(enTxPin, LOW);        // change pin to RX
    Serial.println("waiting data");
    if (RS485Serial.available()) {
      data();
    }
  }
  else if (digitalRead(7) == 0) {
    lcd.clear();                       // menghapus tampilan LCD
    Serial.println("lcd clear");
  }
  else {
    digitalWrite(enTxPin, HIGH);
    RS485Serial.print("NO DATA%");
  }
  /*Serial.println("waiting loop");
  Serial.println();*/
```

```

    delay(500);
}
void data() {
  int dataIN1;
  int dataIN2;
  int dataIN3;
  dataIN1 = RS485Serial.parseInt();
  Serial.println(dataIN1);
  lcd.setCursor(0, 0); // set posisi cursor di kolom 0 baris 0
  lcd.print("f= "); // menampilkan teks pada LCD
  lcd.setCursor(3, 0); // set posisi cursor di kolom 3 baris 0
  lcd.print(dataIN1);
  lcd.setCursor(7, 0); // set posisi cursor di kolom 7 baris 0
  lcd.print("Hz"); // menampilkan teks pada LCD
  dataIN2 = RS485Serial.parseInt();
  Serial.println(dataIN2);
  lcd.setCursor(0, 1); // set posisi cursor di kolom 0 baris 1
  lcd.print("E= "); // menampilkan teks pada LCD
  lcd.setCursor(3, 1); // set posisi cursor di kolom 3 baris 1
  lcd.print(dataIN2);
  lcd.setCursor(7, 1); // set posisi cursor di kolom 7 baris 1
  lcd.print("V"); // menampilkan teks pada LCD
  dataIN3 = RS485Serial.parseInt();
  Serial.println(dataIN3);
  lcd.setCursor(0, 2); // set posisi cursor di kolom 0 baris 2
  lcd.print("I= "); // menampilkan teks pada LCD
  lcd.setCursor(3, 2); // set posisi cursor di kolom 3 baris 2
  lcd.print(dataIN3);
  lcd.setCursor(7, 2); // set posisi cursor di kolom 7 baris 2
  lcd.print("A"); // menampilkan teks pada LCD
  int P;
  P = dataIN2 * dataIN3;
  Serial.println(P);
  lcd.setCursor(0, 3); // set posisi cursor di kolom 0 baris 3
  lcd.print("P=E*I= "); // menampilkan teks pada LCD
  lcd.setCursor(7, 3); // set posisi cursor di kolom 7 baris 3
  lcd.print(P); // menampilkan teks pada LCD
  lcd.setCursor(13, 3); // set posisi cursor di kolom 13 baris 3
  lcd.print("V"); // menampilkan teks pada LCD
  digitalWrite(enTxPin, HIGH);
}

```

- c. Selanjutnya buatlah program slave 1 pada software arduino. Berikut programnya:

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h> // menambahkan library LCD I2C
#include <SoftwareSerial.h>
#define enTxPin 4 // HIGH: TX and LOW: RX
SoftwareSerial RS485Serial(2, 3); // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2); // set alamat I2C dan ukuran LCD
void setup() {
  lcd.init(); // Inisiasi LCD
  lcd.backlight(); // menyalakan backlight
  Serial.begin(9600);
  Serial.println("Starting System");
  RS485Serial.begin(9600);
  pinMode(enTxPin, OUTPUT);
  digitalWrite(enTxPin, LOW); // default RX
}
void loop() {
  if (RS485Serial.available()) {
    String dataIN = "";
    dataIN = RS485Serial.readStringUntil('%');
    Serial.println(dataIN);
    lcd.clear(); // menghapus tampilan LCD
    lcd.setCursor(0, 0); // set posisi cursor di kolom 0 baris 0
    lcd.print(dataIN);
    if (dataIN.substring(0, 3) == "AR1") {
      Serial.println("kirim data ke master");
      lcd.setCursor(0, 1); // set posisi cursor di kolom 0 baris 1
      lcd.print("kirim data"); // menampilkan teks pada LCD
      digitalWrite(enTxPin, HIGH); // change pin to HIGH for TX
      RS485Serial.println(50); //frekuensi(Hz)
      RS485Serial.println(220); //tegangan(V)
      RS485Serial.println(4); //Arus(A)
    }
  }
}

```

```

    digitalWrite(enTxPin, LOW);          // default RX
  }
}
delay(10);
}

```

- d. Selanjutnya buatlah program slave 2 pada software arduino. Berikut programnya:

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>          // menambahkan library LCD I2C
#include <SoftwareSerial.h>
#define enTxPin 4                       // HIGH: TX and LOW: RX
SoftwareSerial RS485Serial(2, 3);       // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2);    // set alamat I2C dan ukuran LCD
void setup() {
  lcd.init();                           // Inisiasi LCD
  lcd.backlight();                       // menyalakan backlight
  Serial.begin(9600);
  Serial.println("Starting System");
  RS485Serial.begin(9600);
  pinMode(enTxPin, OUTPUT);
  digitalWrite(enTxPin, LOW);           // default RX
}
void loop() {
  if (RS485Serial.available()) {
    String dataIN = "";
    dataIN = RS485Serial.readStringUntil('%');
    Serial.println(dataIN);
    lcd.clear();                         // menghapus tampilan LCD
    lcd.setCursor(0, 0);                 // set posisi cursor di kolom 0 baris 0
    lcd.print(dataIN);
    if (dataIN.substring(0, 3) == "AR2") {
      Serial.println("kirim data ke master");
      lcd.setCursor(0, 1);              // set posisi cursor di kolom 0 baris 1
      lcd.print("kirim data");          // menampilkan teks pada LCD
      digitalWrite(enTxPin, HIGH);      // change pin to HIGH for TX
      RS485Serial.println(50);          //frekuensi(Hz)
      RS485Serial.println(150);         //tegangan(V)
      RS485Serial.println(6);          //Arus(A)
      digitalWrite(enTxPin, LOW);       // default RX
    }
  }
  delay(10);
}

```

- e. Selanjutnya buatlah program slave 3 pada software arduino. Berikut programnya:

```

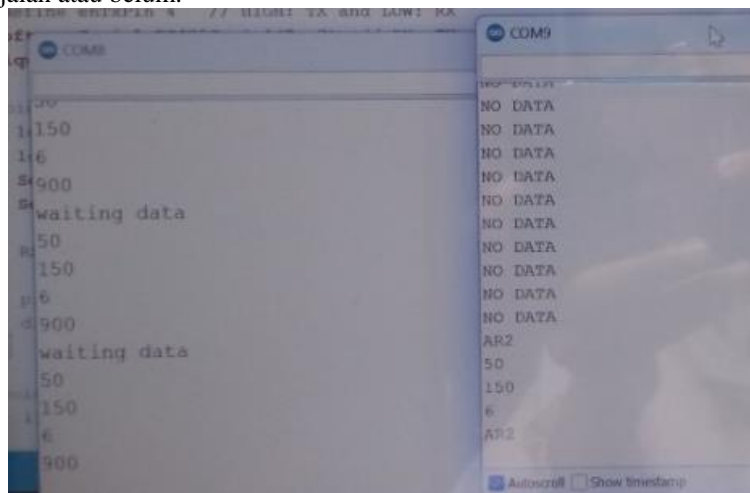
#include <Wire.h>
#include <LiquidCrystal_I2C.h>          // menambahkan library LCD I2C
#include <SoftwareSerial.h>
#define enTxPin 4                       // HIGH: TX and LOW: RX
SoftwareSerial RS485Serial(2, 3);       // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2);    // set alamat I2C dan ukuran LCD
void setup() {
  lcd.init();                           // Inisiasi LCD
  lcd.backlight();                       // menyalakan backlight
  Serial.println("Starting System");
  RS485Serial.begin(9600);
  pinMode(enTxPin, OUTPUT);
  digitalWrite(enTxPin, LOW);           // default RX
}
void loop() {
  if (RS485Serial.available()) {
    String dataIN = "";
    dataIN = RS485Serial.readStringUntil('%');
    Serial.println(dataIN);
    lcd.clear();                         // menghapus tampilan LCD
    lcd.setCursor(0, 0);                 // set posisi cursor di kolom 0 baris 0
    lcd.print(dataIN);
    if (dataIN.substring(0, 3) == "AR3") {
      Serial.println("kirim data ke master");
      lcd.setCursor(0, 1);              // set posisi cursor di kolom 0 baris 1
      lcd.print("kirim data");          // menampilkan teks pada LCD
      digitalWrite(enTxPin, HIGH);      // change pin to HIGH for TX
    }
  }
}

```

```

RS485Serial.println(50);           //frekuensi(Hz)
RS485Serial.println(50);           //tegangan(V)
RS485Serial.println(16);           //Arus(A)
digitalWrite(enTxPin, LOW);        // default RX
}
}
delay(10);
}
    
```

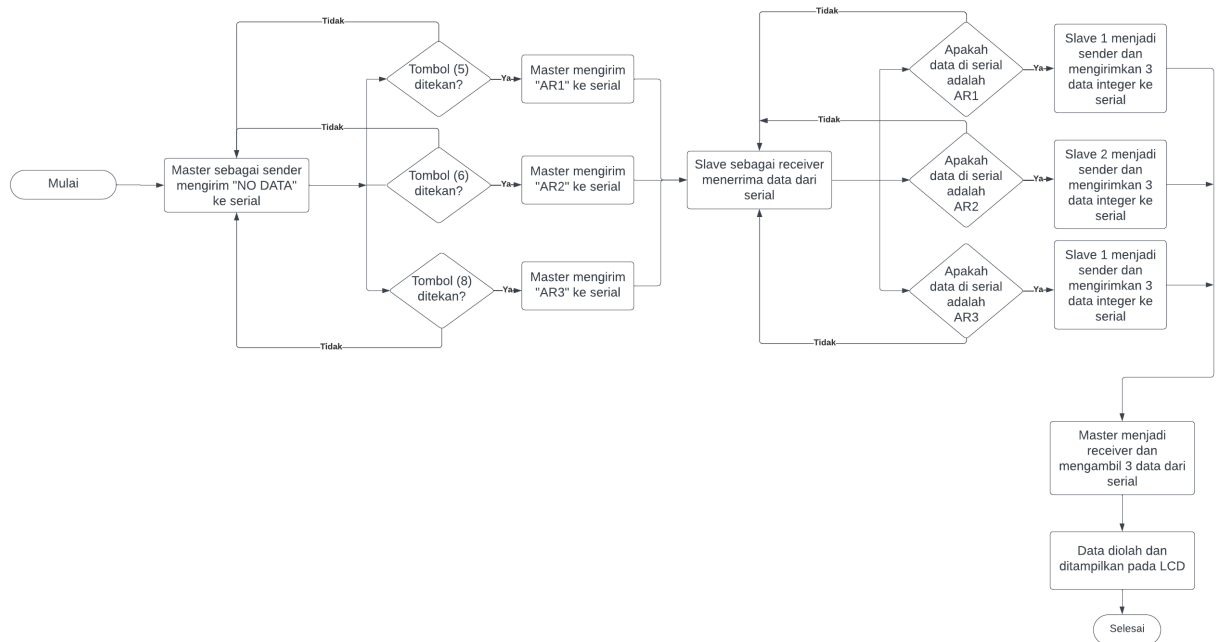
- f. Setelah menyelesaikan pembuatan program untuk master, slave 1, slave 2, dan slave 3, selanjutnya gunakan kabel USB untuk mengunggah program-program tersebut ke Arduino Uno yang masing-masing dijadikan master, slave 1, slave 2, dan slave 3 secara berurutan.
- g. Setelah berhasil mengunggah program master, slave 1, slave 2, dan slave 3, langkah selanjutnya adalah membuka serial monitor untuk memeriksa apakah input, output, dan perintah program yang telah dibuat sudah berjalan atau belum.



Gambar 3. Serial Monitor Mengirim & Menerima Data

- h. Pada gambar 3, com8 berfungsi sebagai master sementara com9 berperan sebagai slave 1. Setiap kali master mengirim paket data ke slave (AR1%), master akan menunggu respons data dari slave tersebut (waiting data). Setelah data diterima (50, 220, 4), master akan kembali mengirimkan paket data ke slave berikutnya. Keadaan "No Data" terjadi ketika master tidak melakukan tindakan apapun.

2.4. Flowchart Sistem



Gambar 4. Flowchart Cara Kerja Sistem

Berdasarkan gambar flowchart di atas, dapat diuraikan sebagai berikut:

1. Ketika sistem dimulai, master diposisikan sebagai pengirim karena pada void setup enTxPin dalam keadaan HIGH. Sedangkan pada slave, posisi awalnya adalah sebagai penerima karena pada void setup enTxPin dalam keadaan LOW.
2. Kemudian, jika salah satu push button ditekan, maka akan mengirimkan data ke masing-masing slave. Jika tombol DigitalRead 5 (push button 1) ditekan, master akan mengirim data ke RS485Serial berupa string yang berisi "AR1%" dan mengubah posisi master menjadi penerima. Karena ada data yang masuk, maka slave akan membaca data dari RS485Serial, dan hanya membaca sampai tanda "%".
3. Pada slave 1, jika data serial RS485 yang masuk adalah "AR1%", maka posisi slave akan berubah menjadi pengirim karena ada perintah enTxPin yang HIGH. Kemudian slave 1 mengirimkan ketiga data yang terdapat pada program ke RS485Serial. Master menerima data pada slave 1 dan memanggil perintah void data.
4. Pada slave 2 yang mendapat data "AR1%", karena data tidak sesuai dengan program pada slave 2, maka slave 2 tetap menjadi penerima data dan tidak melakukan apapun. Jika tombol DigitalRead 6 (push button 2) yang ditekan, maka akan terjadi proses seperti sebelumnya namun tidak melibatkan slave 1 melainkan slave 2 dan data yang dikirim master adalah "AR2%". Jika master tidak melakukan apapun maka master akan mengirimkan data yang berisi "NO DATA". Namun jika tombol DigitalRead 7 (push button 4) yang ditekan maka akan me-reset LCD.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Data Slave 1



Gambar 5. Hasil Slave1 Data 1

Pada Gambar 5 merupakan hasil jika push button 1 ditekan, maka master akan mengirim data berupa string yang berisi "AR1%" dan mengubah posisi master menjadi penerima. Karena ada data yang masuk, maka slave 1 akan membaca data dari RS485Serial. Pada slave 1, jika data yang masuk adalah "AR1%", maka posisi slave akan berubah menjadi pengirim karena ada perintah enTxPin yang HIGH. Kemudian slave 1 mengirimkan data berupa daya aliran listrik yang seharusnya rumusnya adalah  $P = V * I$ , tetapi pada percobaan terjadi kesalahan penulisan tegangan dan satuan daya. Master akan menerima data pada slave 1 dan akan ditampilkan pada LCD. Datanya sebagai berikut:

$$\begin{aligned}
 F &= 50\text{Hz} \\
 V &= 220\text{v} \\
 I &= 4 \text{ A} \\
 P &= V * I \\
 &= 220\text{V} * 4\text{A} \\
 &= 880 \text{ watt}
 \end{aligned}$$

Jika ingin me-reset tampilan LCD maka tekan push button 4.

#### 3.2. Data Slave 2



Gambar 6. Hasil Slave 2 Data 2



Pada gambar 6 merupakan hasil jika push button 2 ditekan, maka master akan mengirim data berupa string yang berisi "AR2%" dan mengubah posisi master menjadi penerima. Karena ada data yang masuk, maka slave 2 akan membaca data dari RS485Serial. Pada slave 2, jika data yang masuk adalah "AR2%", maka posisi slave akan berubah menjadi pengirim karena ada perintah enTxPin yang HIGH. Kemudian slave 2 mengirimkan data berupa daya aliran listrik yang seharusnya rumusnya adalah  $P = V \cdot I$ , tetapi pada percobaan terjadi kesalahan penulisan tegangan dan satuan daya. Master akan menerima data pada slave 2 dan akan ditampilkan pada LCD. Datanya sebagai berikut:

$$\begin{aligned} F &= 50\text{Hz} \\ V &= 150\text{v} \\ I &= 6 \text{ A} \\ P &= V \cdot I \\ &= 150\text{V} \cdot 6\text{A} \\ &= 900 \text{ watt} \end{aligned}$$

Jika ingin me-reset tampilan LCD maka tekan push button 4.

### 3.3. Data Slave 3



Gambar 7. Hasil Slave 3 Data 3

Pada gambar 7 merupakan hasil jika push button 3 ditekan, maka master akan mengirim data berupa string yang berisi "AR3%" dan mengubah posisi master menjadi penerima. Karena ada data yang masuk, maka slave 3 akan membaca data dari RS485Serial. Pada slave 3, jika data yang masuk adalah "AR3%", maka posisi slave akan berubah menjadi pengirim karena ada perintah enTxPin yang HIGH. Kemudian slave 3 mengirimkan data berupa daya aliran listrik yang seharusnya rumusnya adalah  $P = V \cdot I$ , tetapi pada percobaan terjadi kesalahan penulisan tegangan dan satuan daya. Master akan menerima data pada slave 3 dan akan ditampilkan pada LCD. Datanya sebagai berikut:

$$\begin{aligned} F &= 50\text{Hz} \\ V &= 50\text{v} \\ I &= 16 \text{ A} \\ P &= V \cdot I \\ &= 50\text{V} \cdot 16\text{A} \\ &= 800 \text{ watt} \end{aligned}$$

Jika ingin me-reset tampilan LCD maka tekan push button 4.

Tabel 2. Hasil Percobaan

Push button 1	Master	Slave 1
On	Mengirim string berupa AR1% dan mengubah posisi master menjadi penerima. Master akan menerima data daya aliran listrik pada slave 1 dan akan ditampilkan pada LCD	Akan membaca data AR1% slave akan berubah menjadi pengirim. Slave 1 mengirimkan data berupa daya aliran listrik yaitu : F= 50Hz V= 220v I = 4 A P= V * I = 220V * 4A = 880 watt
Push button 2	Master	Slave 2
On	Mengirim string berupa AR2% dan mengubah posisi master menjadi penerima. Master akan menerima data daya aliran listrik pada slave 2 dan akan ditampilkan pada LCD	Akan membaca data AR2% slave akan berubah menjadi pengirim. Slave 2 mengirimkan data berupa daya aliran listrik yaitu : F= 50Hz V= 150v I = 6 A

		$P = V * I$ $= 150V * 6A = 900 \text{ watt}$
Push button 3	Master	Slave 3
On	Mengirim string berupa AR3% dan mengubah posisi master menjadi penerima. Master akan menerima data daya aliran listrik pada slave 3 dan akan ditampilkan pada LCD	Akan membaca data AR3% slave akan berubah menjadi pengirim. Slave 3 mengirimkan data berupa daya aliran listrik yaitu : F= 50Hz V= 50v I = 16 A P= V * I $= 50V * 16A = 800 \text{ watt}$
Push button 4		
Reset LCD		

#### 4. KESIMPULAN

Sistem komunikasi antar Arduino menggunakan protokol RS485 dirancang untuk mendukung komunikasi jarak jauh dengan maksimum jarak/panjang kabel sekitar 1200 meter. Namun, pada gambar percobaan, menggunakan kabel dengan panjang  $\pm 25\text{cm}$ . Komponen yang digunakan pada sistem komunikasi antar arduino menggunakan protokol RS485 adalah Laptop, Software Arduino IDE, Arduino Uno, RS485, Push Button, Project Board/ Breadboard, Liquid Crystal Display (Lcd) 20x4, Kabel Jumper, Kabel Usb, Kabel AWG 22. Cara kerja sistem komunikasi antar Arduino menggunakan protokol RS485 yaitu mikrokontroler master berperan sebagai pusat kendali data, sedangkan mikrokontroler slave bertugas untuk mengontrol data secara langsung. Data yang didapatkan merupakan data aliran daya listrik dari setiap slavenya. Pada slave 1, daya yang terdeteksi adalah sebesar 880 watt. Pada slave 2, daya yang terdeteksi adalah sebesar 900 watt. Sementara pada slave 3, daya yang terdeteksi adalah sebesar 800 watt.

#### UCAPAN TERIMA KASIH

Terima kasih kepada semua pihak yang telah mendukung penulis selama Kerja Praktik ini. Tanpa bantuan dan bimbingan kalian, pencapaian penulis tidak akan tercapai. Penulis juga sangat berterima kasih atas kesempatan dan pengalaman berharga yang telah diberikan.

#### REFERENSI

- [1] Si.uinsu.ac.id (2023). Kerja Praktik. Diakses pada 16 Maret 2023 pukul 20.00 WIB, dari <https://si.uinsu.ac.id/kerja-praktik>
- [2] Cahyono dkk. (2017). "PROTOTIPE PANEL MONITORING LAMPU LISTRIK TERPUSAT MENGGUNAKAN KOMUNIKASI RS485". Seminar Nasional Ilmu Terapan (SNITER), Universitas Widya Kartika.
- [3] Dosen Pendidikan 3. (16 Maret, 2023). Pengertian Simplex Half Duplex Full Duplex. Diakses pada 17 Mei 2023 pukul 22.00 WIB, dari <https://www.dosenpendidikan.co.id/pengertian-simplex/>
- [4] Repository.usm.ac.id. (2014). BAB II DASAR TEORI 2.1. ARDUINO UNO. Diakses pada 16 Maret 2023 pukul 21.00 WIB, dari <https://repository.usm.ac.id/files/skripsi/C41A/2014/C.431.14.0014/C.431.14.0014-05-BAB-II-20190905115100.pdf>
- [5] Erintafifah. (Oct 8, 2021). Mengenal Perangkat Lunak Arduino IDE. Diakses pada 30 Maret 2023 pukul 21.00 WIB, dari <https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide>
- [6] Eprints.umm.ac.id. BAB II TINJAUAN PUSTAKA 2.3 Liquid Crystal Display (LCD) 20x4. Diakses pada 22 Maret 2023 pukul 19.30 WIB, dari <https://eprints.umm.ac.id/73866/3/BAB%20II.pdf>
- [7] Ester ambarita. "Project Board". Diakses pada 30 Maret 2023 pukul 21.00 WIB, dari [https://www.academia.edu/35849146/Project\\_Board](https://www.academia.edu/35849146/Project_Board)
- [8] Razor. (2020). Push Button Arduino: Pengertian, Fungsi, dan Prinsip Kerja. Diakses pada 30 Maret 2023 pukul 23.00 WIB, dari <https://www.aldyrazor.com/2020/05/push-button-arduino.html>
- [9] Perpustakaan.poltektegal.ac.id. BAB III LANDASAN TEORI 3.2 Kabel Jumper. Diakses pada 1 April 2023 pukul 21.30 WIB, dari <https://perpustakaan.poltektegal.ac.id/index.php?p=fstream-pdf&fid=20236&bid=13243>
- [10] Wajiansyah, dkk. (2020). "Implementasi Master-slave Pada Embedded System Menggunakan Komunikasi RS-485". ELKHA, 12(1), 26-31.
- [11] Mulyana, A. & Tosin.(2021). "Perancangan dan Implementasi Komunikasi RS-485 Menggunakan Protokol Modbus RTU dan Modbus TCP pada Sistem Pick-by-Light". Jurnal Sistem Komputer, 10(1), 85-85.
- [12] Mulyadi dkk. (2021). "Modul Komunikasi Modbus RTU over RS485 Berbasis Arduino". JOURNAL OF APPLIED ELECTRICAL ENGINEERING, 5(1).
- [13] Prayudha dkk. (2020). "Implementasi Teknik Komunikasi Serial Half Duplex Pada Kendali Jarak Jauh Lampu Ruangan Rumah Berbasis Internet Of Things (IOT)". Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD, 3(1), 32-40.
- [14] Krisnandi dkk. (2017). " Implementasi Mikrokontroler Single Master - Multi Slave untuk Pengiriman dan Penerimaan Data pada Prototipe Stasiun Cuaca". Widyariset, 3(1), 19-34.
- [15] Najib dkk. (2021). "Pengambilan Data Praktikum Modul Photovoltaic dalam Penerapan Protokol Kesehatan Memanfaatkan Komunikasi Rs485". AINET, 3(1), 38-45.