

Workshop Aplikasi Mikroprosesor & Antarmuka

**PROGRAM STUDI
TEKNIK TELKOMUNIKASI**

Akuwan Saleh, MT

PENILAIAN

⇒ Laporan + Tugas + Presentasi = **60%**

eval-1(Lap.1-5) = 20%

eval-2(Lap.6-10) = 20%

eval-3(Lap.11-15 & (PPT+ presentasi)) = 20%

⇒ TPS = **40%**

eval-4 (TPS = Tugas Proyek Semester)

REFERENSI

- Rui Santos & Sara Santos, “ESP32 Web Server With Arduino IDE.pdf: Step By Step Project Guide”, <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>, juni 2020.
- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013
- Fabian Winkler, “Arduino/Processing Communication Workshop”, Fall, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springer, New York, 2011.
- Casey Reas and Ben Fry, “Getting Started with Processing”, O’Reilly Media, Inc., June 2010.
- Joshua Noble, “Programming Interactivity: A Designer’s Guide to Processing, Arduino, and openFrameworks”, O’Reilly Media, Inc., July 2009.
- Peter Hoddie, Lizzie Prader, “IoT Development for ESP32 and ESP8266 with JavaScript; A Practical Guide to XS and Moddable SDK”, Apress, Menlo Park, CA,USA, 2020.

MATERI

PENDAHULUAN

- 1. KOMUNIKASI MIKROKONTROLER DENGAN SOFTWARE PROCESSING**
- 2. ANALOG INPUT DAN AUDIO PROCESSING**
- 3. KONTROL MULTI LED MENGGUNAKAN ARDUINO DAN PROCESSING**
- 4. PENCAMPUR WARNA VIRTUAL MENGGUNAKAN ARDUINO DAN PROCESSING**
- 5. MONITORING SUHU DENGAN ARDUINO DAN PROCESSING**
- 6. MONITORING INTENSITAS CAHAYA DENGAN ARDUINO DAN PROCESSING**
- 7. KONTROL MOTOR DC MENGGUNAKAN ARDUINO DAN PROCESSING**
- 8. APLIKASI SENSOR ULTRASONIC MENGGUNAKAN ARDUINO DAN PROCESSING**

MATERI

9. KONTROL LAMPU AC 220 V BERBASIS ARDUINO DAN PROCESSING
10. MODUL WiFi ESP32 DENGAN ARDUINO IDE
11. KOMUNIKASI NIRKABEL MENGGUNAKAN MODUL RF 434 MHz DAN PROCESSING
12. ESP32 WEB SERVER UNTUK KONTROL LED DAN MENAMPILKAN GAMBAR
13. ANTARMUKA MODUL GPS DENGAN MIKROKONTROLER DAN PROCESSING
14. ESP32 WEB SERVER UNTUK PENGUKURAN SUHU DAN KELEMBABAN
15. KOMUNIKASI DATA BERBASIS BLUETOOTH DAN HP

11. KOMUNIKASI NIRKABEL MENGUNAKAN MODUL RF 434 MHz DAN PROCESSING

TUJUAN

- Membuat komunikasi nirkabel antar mikrokontroler menggunakan modul RF
- Membuat rangkaian Pemancar dan Penerima RF dengan frekuensi 434 MHz
- Menampilkan data transmisi menggunakan processing

DASAR TEORI

➤ Komunikasi Nirkabel

- ✓ Transmisi data secara nirkabel, tidak ada kerumitan kabel dan tidak ada kontak langsung dengan perangkat itu sendiri.
- ✓ Menggabungkan dua objek yaitu komunikasi nirkabel dengan mikrokontroler.
- ✓ Bisa membuat aplikasi : *Remote Controls, Automation System, Car Security System* dll.

➤ Modul RF

- ✓ Salah satu cara termudah dan termurah untuk mengimplementasikan komunikasi nirkabel adalah dengan menggunakan Modul RF (Modul Frekuensi Radio).
- ✓ Sebagai modul komunikasi nirkabel murah untuk aplikasi berbiaya rendah
- ✓ Terdiri dari pemancar dan penerima yang beroperasi pada rentang frekuensi radio.

Range Frekuensi Radio (RF)

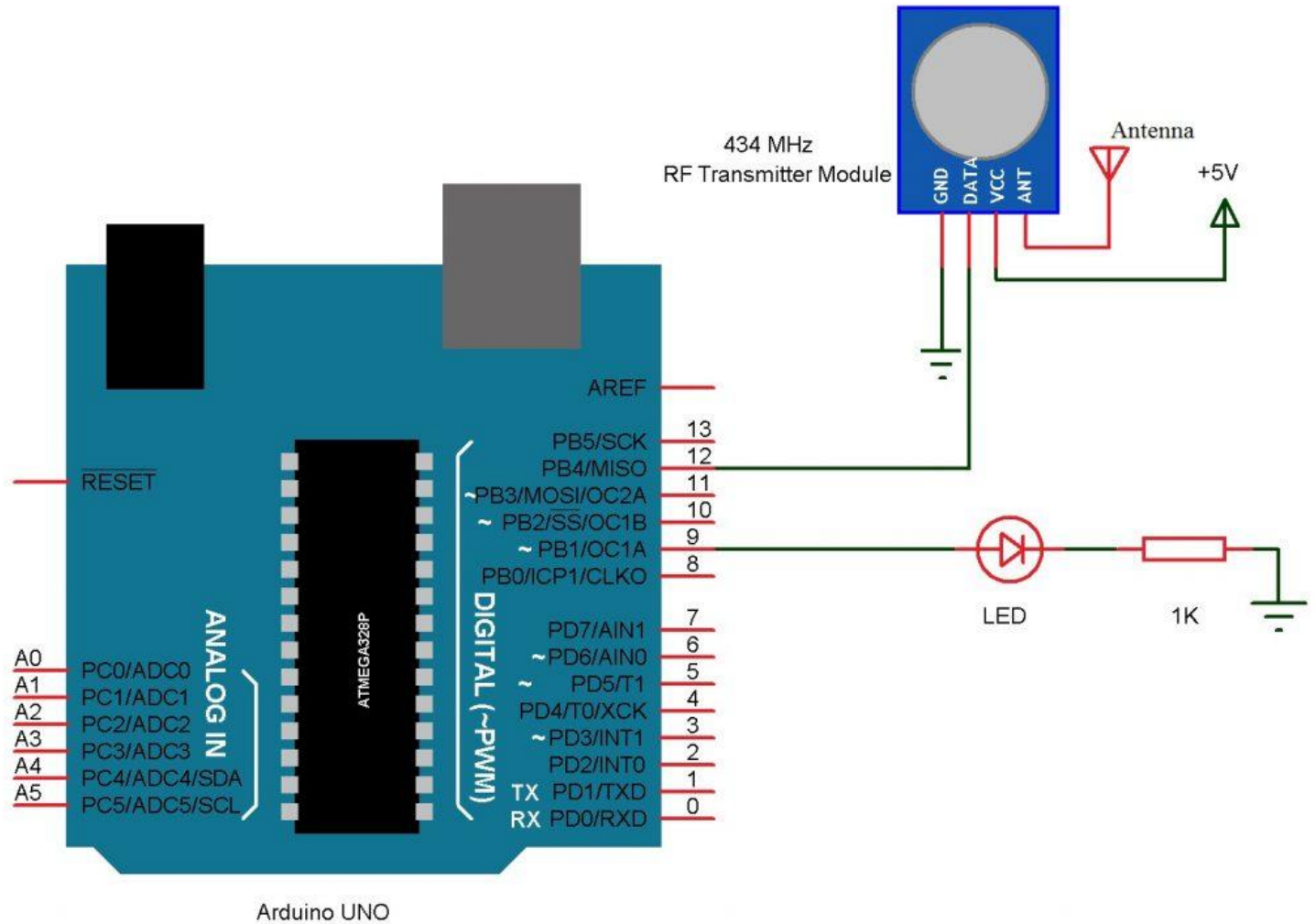
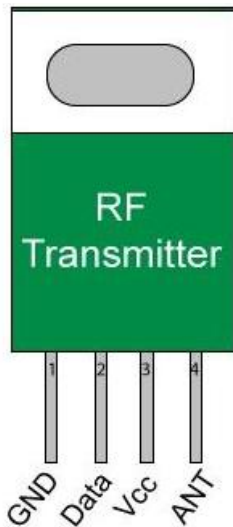
Frekuensi	Panjang gelombang	Nama band	Singkatan ^[1]
3 – 30 Hz	$10^4 - 10^5$ km	Extremely low frequency	ELF
30 – 300 Hz	$10^3 - 10^4$ km	Super low frequency	SLF
300 – 3000 Hz	$100 - 10^3$ km	Ultra low frequency	ULF
3 – 30 kHz	10 – 100 km	Very low frequency	VLF
30 – 300 kHz	1 – 10 km	Low frequency	LF
300 kHz – 3 MHz	100 m – 1 km	Medium frequency	MF
3 – 30 MHz	10 – 100 m	High frequency	HF
30 – 300 MHz	1 – 10 m	Very high frequency	VHF
300 MHz – 3 GHz	10 cm – 1 m	Ultra high frequency	UHF 
3 – 30 GHz	1 – 10 cm	Super high frequency	SHF
30 – 300 GHz	1 mm – 1 cm	Extremely high frequency	EHF
300 GHz - 3000 GHz	0.1 mm - 1 mm	Tremendously high frequency	THF

- ✓ Menggunakan pasangan pemancar RF - Penerima 434 MHz.
- ✓ Dapat digunakan untuk komunikasi hingga jarak 40 meter.

Bagian Pemancar

- ✓ Terdiri dari Arduino UNO dan modul *Transmitter* 434 MHz.
- ✓ LED eksternal atau LED on board sebagai indikator.
- ✓ Desain bagian *Transmitter* adalah sebagai berikut:

Transmitter

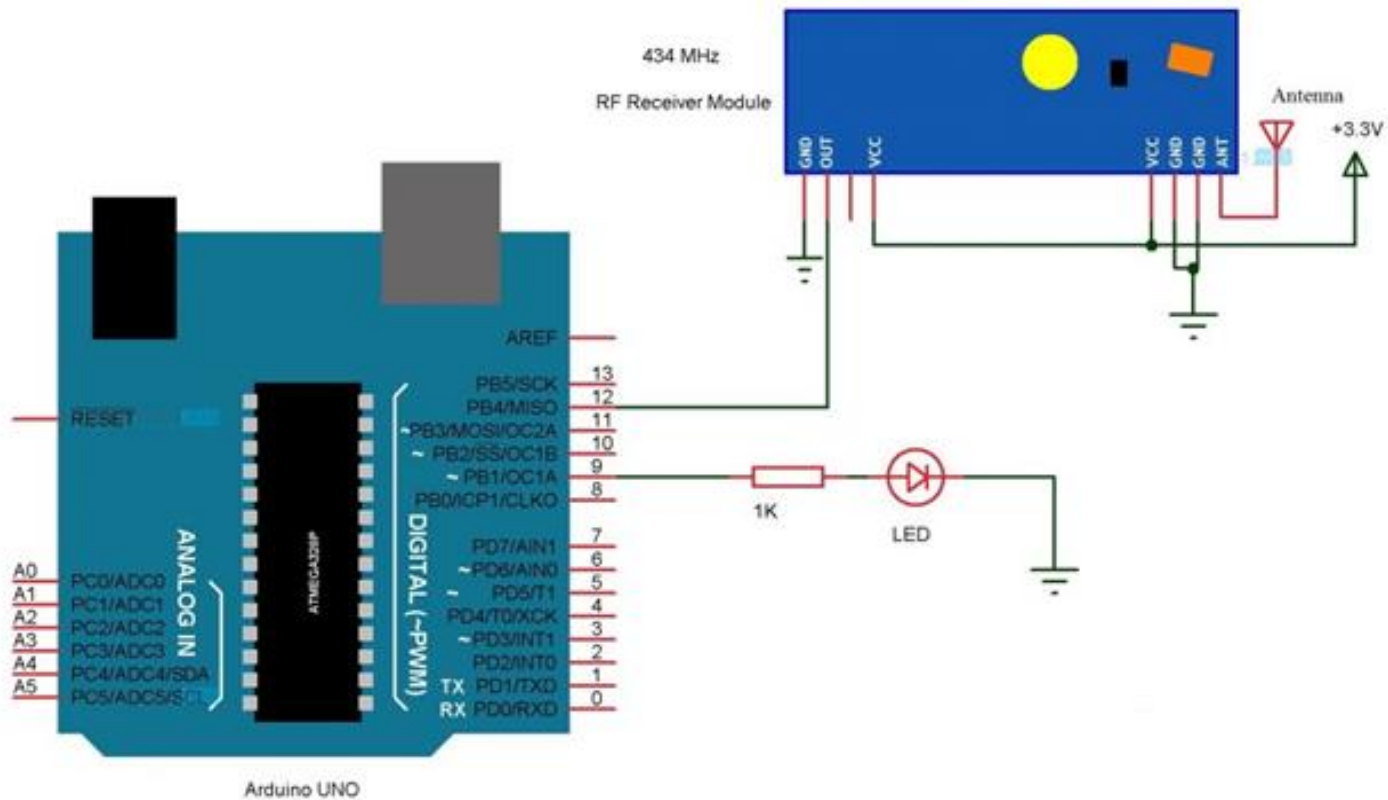
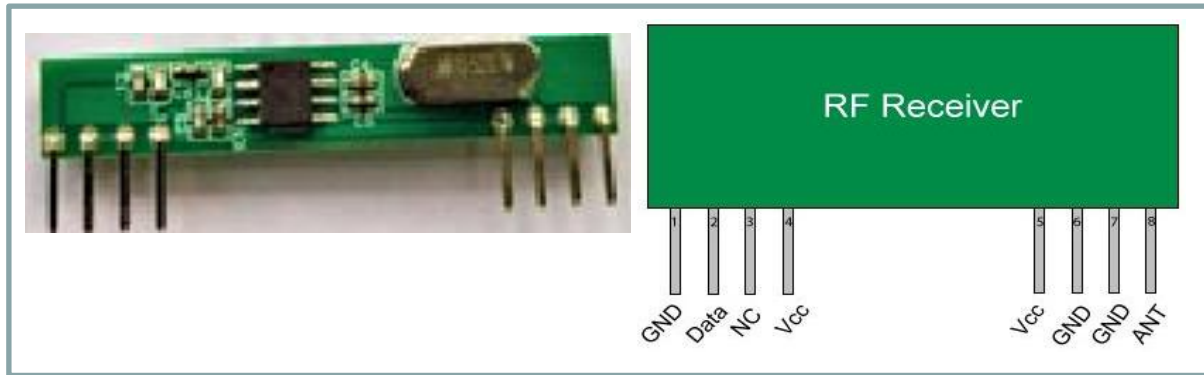


Modul Pemancar RF terdiri dari 4 - pin: VCC, GND, Data dan Antena.

Bagian Penerima

- ✓ Terdiri dari Arduino UNO dan modul *Receiver* 434 MHz.
- ✓ LED eksternal atau LED on board sebagai indikator.
- ✓ Desain bagian *Receiver* adalah sebagai berikut:

Receiver



Modul Penerima RF terdiri dari 8 - pin: 2xVCC, 3xGND, Data, Antena dan NC.

Peralatan:

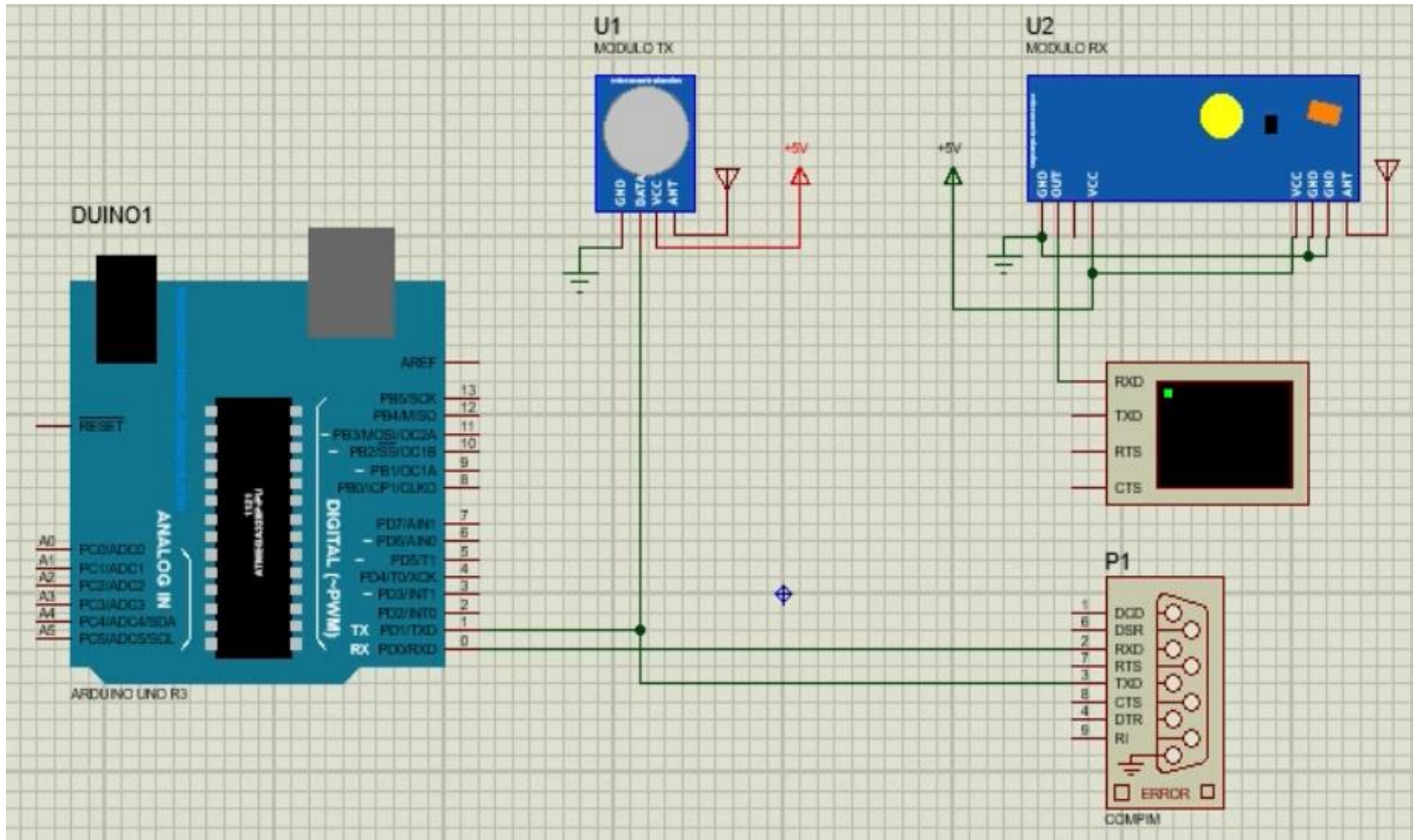
Bagian Transmitter

- Arduino UNO
- 434 MHz Transmitter Modul
- LED
- 1 K Ω Resistor
- Bread board
- Kabel Jumper
- Software Processing

Bagian Penerima

- Arduino UNO
- 434 MHz Receiver Modul
- LED
- 1 K Ω Resistor
- Bread board
- Kabel Jumper

Rangkaian: Percobaan-1



PROGRAM-1:

Arduino IDE

```
void setup() {  
  Serial.begin(9600);  
  
}  
  
void loop() {  
  Serial.println("Hello World");  
  delay(1000);  
}
```

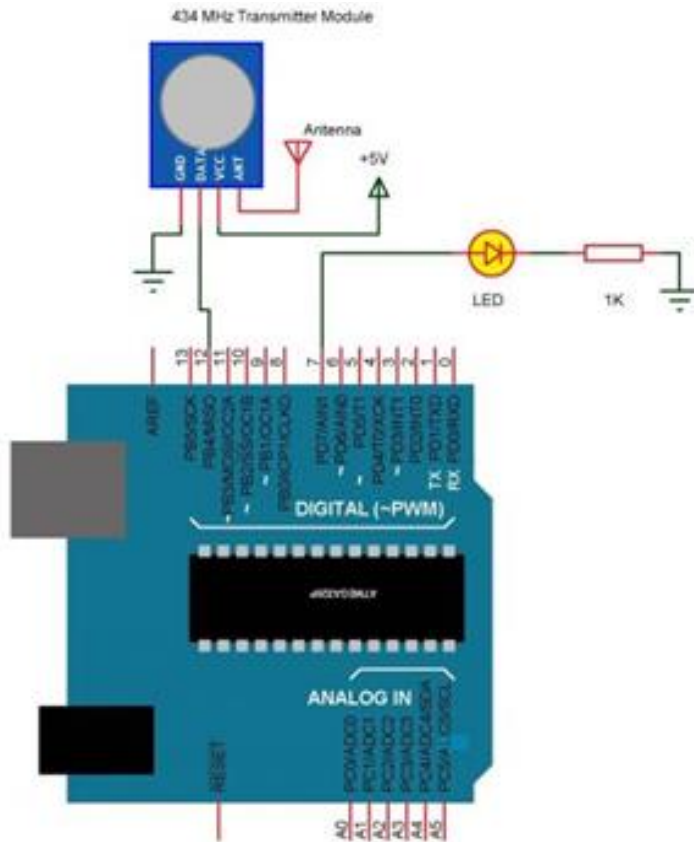
PROGRAM-1:

Processing

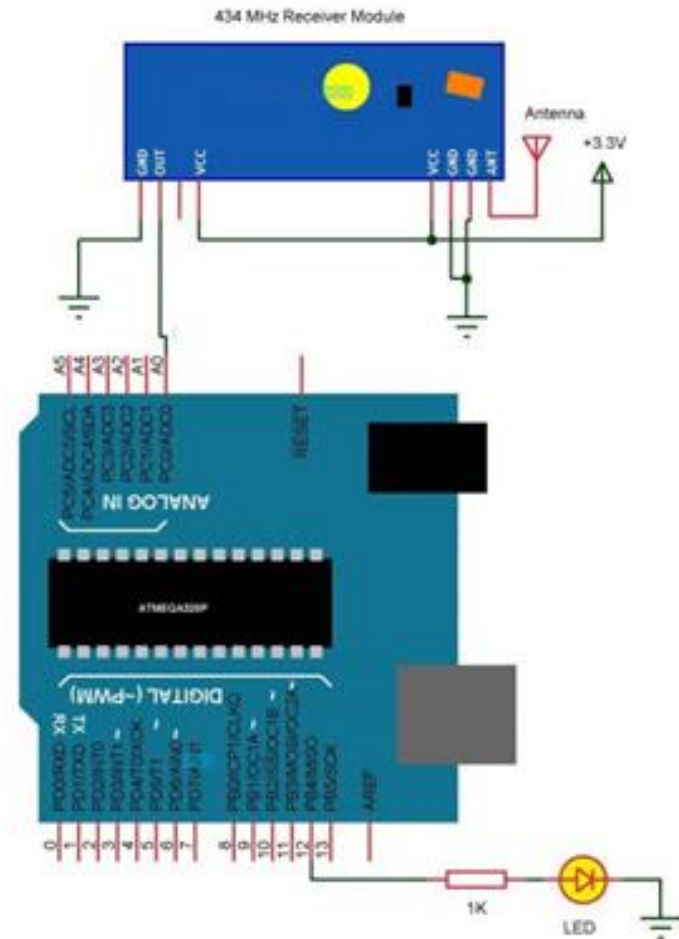
```
import processing.serial.*;
Serial myPort;
String val;
void setup()
{
String portName = Serial.list()[0];
myPort = new Serial(this, portName, 9600);
}
void draw()
{
if ( myPort.available() > 0)
{
val = myPort.readStringUntil('\n');
}
println(val);
}
```

Rangkaian: Percobaan-2

Pemancar



Penerima



PROGRAM-2:

Arduino IDE

```
# PEMANCAR (Tx)  
#define dataout 12  
#define ledPin 7  
void setup()  
{  
  pinMode(dataout, OUTPUT);  
  pinMode(ledPin, OUTPUT);  
}  
void loop()  
{  
  digitalWrite(dataout, HIGH);  
  digitalWrite(ledPin, HIGH);  
  delay(2000);  
  digitalWrite(dataout, LOW);  
  digitalWrite(ledPin, LOW);  
  delay(2000);  
}
```

PROGRAM-2:

Arduino IDE

```
# PENERIMA (Rx)
```

```
#define datain A0
```

```
#define ledPin 12
```

```
unsigned int temp = 0;
```

```
const unsigned int upperThreshold = 600;
```

```
const unsigned int lowerThreshold = 50;
```

```
void setup()
```

```
{
```

```
  pinMode(ledPin, OUTPUT);
```

```
}
```

```
void loop(){
```

```
  temp=analogRead(datain);
```

```
  if(temp<lowerThreshold)
```

```
  {
```

```
    digitalWrite(ledPin, HIGH);
```

```
  }
```

```
else if(temp>upperThreshold)
```

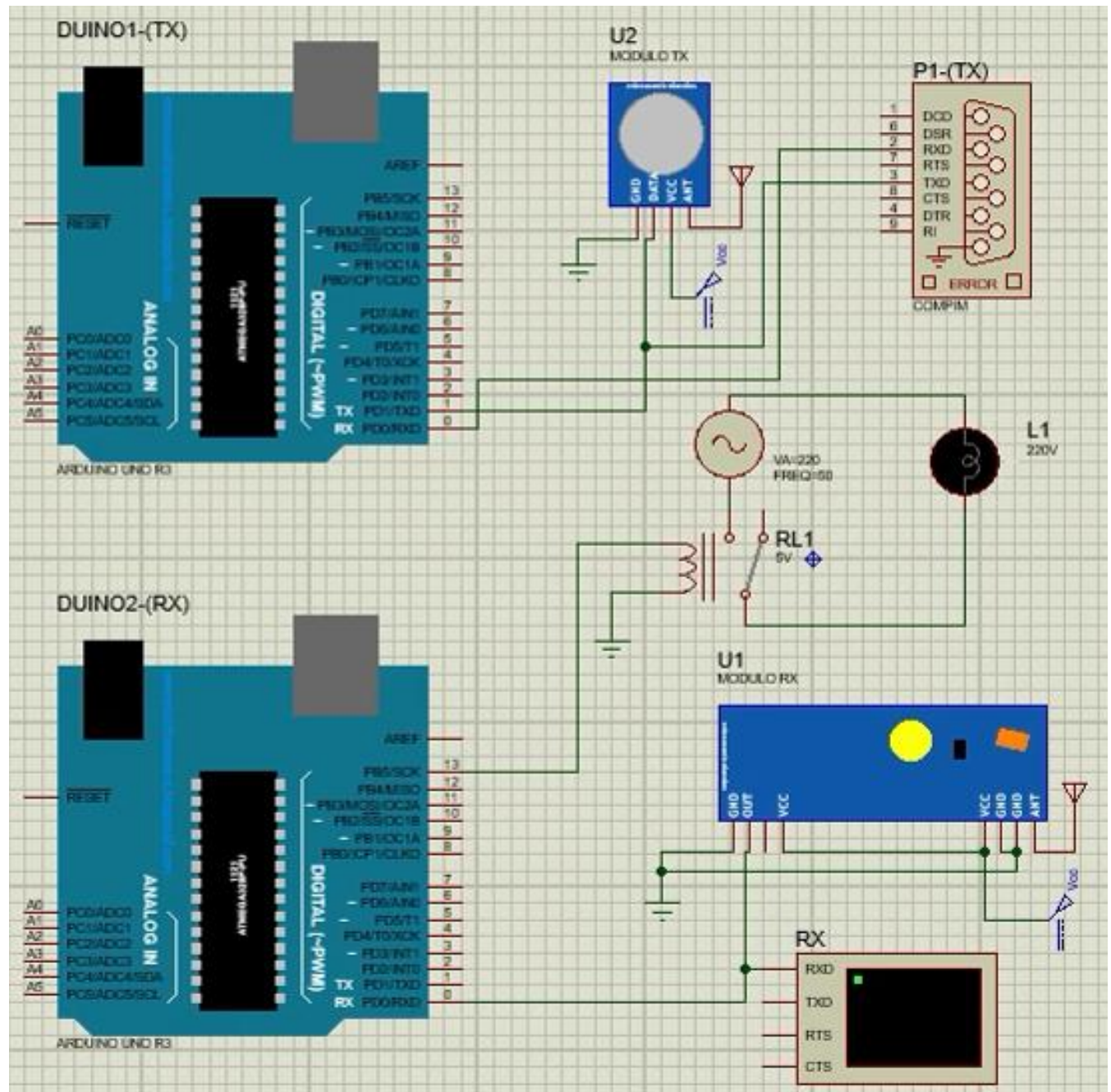
```
{
```

```
  digitalWrite(ledPin, LOW);
```

```
}
```

```
}
```

Rangkaian: Percobaan-3



PROGRAM-3:

Arduino IDE

PEMANCAR (Tx)

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("ON");  
  delay(1000);  
}
```

PEMANCAR (Tx)

Processing

```
import processing.serial.*;  
Serial myPort;  
String val;  
void setup()  
{  
  String portName = Serial.list()[0];  
  myPort = new Serial(this, portName, 9600);  
}  
void draw()  
{  
  if ( myPort.available() > 0)  
  {  
    val = myPort.readStringUntil('\n');  
  }  
  println(val);  
}
```

PROGRAM-3:

Arduino IDE

```
# PENERIMA (Rx)
```

```
int lampu=13 ;
```

```
char mystr[10]; //Initialized variable to store recieved data
```

```
void setup(){
```

```
pinMode(lampu, OUTPUT);
```

```
Serial.begin(9600);
```

```
}
```

```
void loop(){
```

```
Serial.readBytes(mystr,5);
```

```
if (char mystr='ON'){
```

```
digitalWrite(lampu, HIGH); // Turns ON Relays
```

```
delay(1000); }
```

```
digitalWrite(lampu,LOW); // Turns Relay Off
```

```
delay(1000);
```

```
}
```


Hasil :

- Amati pada IDE Processing dan virtual terminal, catat data yang dikirim dan yg diterima.
- Amati Relay dan lampu AC 220V.

Latihan :

1. Pada rangkaian percobaan-3, pada bagian penerima(Rx) gantilah virtual terminal dengan LCD 20x4, buatlah rangkaian dan programnya.
2. Pada rangkaian percobaan-2 jika data yang dikirim berasal dari sensor intensitas cahaya (LDR), buatlah rangkaian beserta programnya dan tampilkan data dibagian penerima(Rx) pada virtual terminal.