

# **MODUL 4**

## **KOMUNIKASI SERIAL**

---

Komunikasi data merupakan bentuk komunikasi yang secara khusus berkaitan dengan transmisi atau pemindahan data antara komputer-komputer, komputer dengan piranti-piranti yang lain dalam bentuk data digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu masyarakat informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain.

Komunikasi Data saat ini menjadi bagian dari kehidupan masyarakat, karena telah diterapkan dalam berbagai bentuk aplikasi misal: komunikasi antar komputer yang populer dengan istilah internet, Handphone ke komputer, Handphone ke Handphone, komputer atau handphone ke perangkat lain misal: printer, fax, telpon, camera video, dan lain-lain.

Model komunikasi sederhana terdiri dari :

- Source
- Transmitter (Pengirim)
- Transmission System (Sistem Transmisi)
- Receiver (Penerima)
- Destination (Tujuan)

### **Source**

Alat ini membangkitkan data sehingga dapat ditransmisikan, contoh alat yang bisa disebut source adalah : telepon dan PC (Personal Computer)

### **Transmitter (Pengirim)**

Biasanya data yang dibangkitkan dari sistem sumber tidak ditransmisikan secara langsung dalam bentuknya aslinya. Sebuah trasmitter cukup memindah dan menandai informasi dengan cara yang sama seperti menghasilkan sinyal-sinyal elektromagnetik yang dapat ditransmisikan melewati beberapa sistem transmisi berurutan.

Contoh : Sebuah modem tugasnya menyalurkan suatu digital bit stream dari suatu alat yang sebelumnya sudah dipersiapkan misalnya PC (Personal Computer), dan mentransformasikan bit stream tersebut menjadi suatu sinyal analog yang dapat melintasi jaringan telepon.

### Transmission System (Sistem Transmisi)

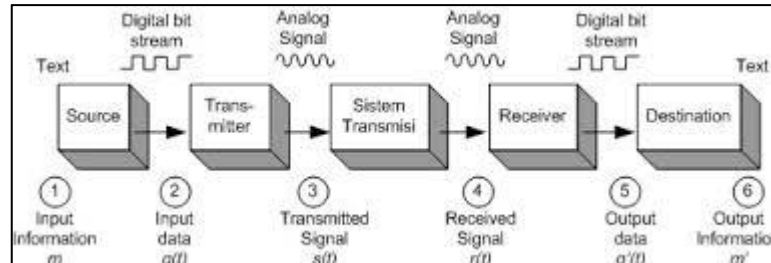
Sistem transmisi berupa jalur transmisi tunggal (single transmission line) atau jaringan kompleks (complex network) yang menghubungkan antara sumber dengan tujuan (destination).

### Receiver (Penerima)

Receiver menerima sinyal dari sistem transmisi dan menggabungkannya ke dalam bentuk tertentu yang dapat ditangkap oleh tujuan. Contoh : modem akan menerima sinyal analog yang datang dari jaringan atau jalur transmisi dan mengubahnya menjadi suatu digital bit stream.

### Destination (Tujuan)

Destination menangkap data yang dihasilkan oleh receiver



**Gambar 4.1** Alur data ke tujuan

Pada sistem komunikasi terdapat dua jenis komunikasi yaitu komunikasi paralel dan komunikasi serial.

1. Komunikasi paralel merupakan komunikasi yang menggunakan konsep pengiriman data secara paralel (bersamaan), semua bit data pada satu waktu sekaligus. Jalur yang dibutuhkan cukup banyak namun memiliki kecepatan yang cukup cepat dibandingkan komunikasi serial.
2. Komunikasi serial menggunakan konsep komunikasi pengiriman data secara serial (berurutan), satu demi satu bit secara bergantian. Komunikasi jenis ini walaupun lebih lambat dari komunikasi paralel namun jalur yang digunakan lebih sedikit.

## Komunikasi Serial Arduino

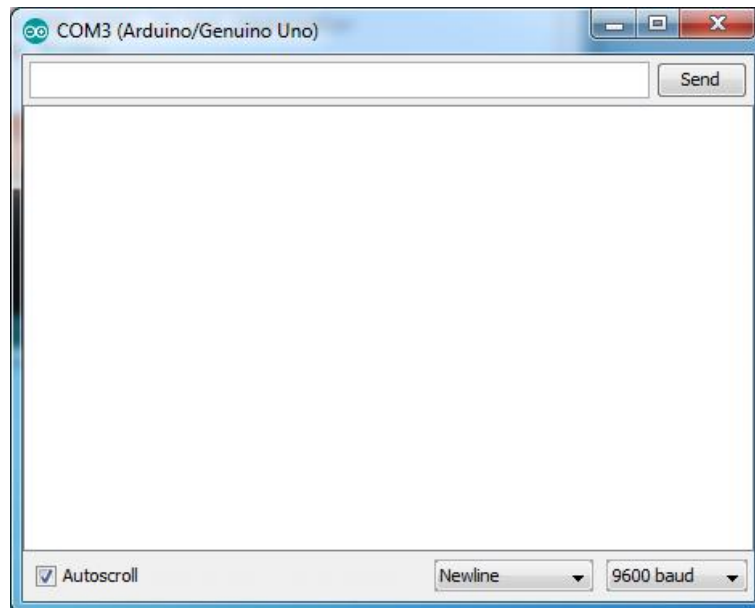
Pada Arduino terdapat PIN yang berfungsi untuk komunikasi data serial, untuk pengiriman data yaitu menggunakan Transmitter (Tx), sedangkan untuk menerima data yaitu menggunakan Receiver (Rx). Dalam komunikasi data serial Arduino dipermudah dengan adanya fungsi-fungsi yang sudah disediakan dari Arduino, diantaranya yaitu:

**Tabel 4.1.** Fungsi Komunikasi Serial Arduino

<b>if(Serial)</b>	:	Untuk mengecek kesiapan Port
<b>Serial.available()</b>	:	Untuk mengecek data pada Buffer Penerima
<b>Serial.begin()</b>	:	Untuk mengeset kecepatan Transmisi Data
<b>Serial.end()</b>	:	Untuk menon-aktifkan PIN Rx dan Tx sebagai fungsi serial dan kembali sebagai PIN Input atau Output
<b>Serial.find()</b>	:	Untuk mencari string dalam Buffer Data
<b>Serial.findUntil()</b>	:	Untuk mencari Buffer data sampai panjang data sesuai dengan sama yang ditemukan
<b>Serial.flush()</b>	:	Untuk menunggu data terkirim semua
<b>Serial.parseFloat()</b>	:	Untuk mengambil data float pertama dari data di Buffer Serial
<b>Serial.parseInt()</b>	:	Untuk mengambil data integer pertama dari data di Buffer Serial
<b>Serial.peek()</b>	:	Untuk mengambil data berikutnya di Buffer Penerima
<b>Serial.print()</b>	:	Untuk mengirim data ASCII
<b>Serial.println()</b>	:	Untuk mengirim data ASCII + CR,LF (kode Enter)
<b>Serial.read()</b>	:	Untuk membaca data yang diterima
<b>Serial.readBytes()</b>	:	Untuk membaca data Byte yang diterima
<b>Serial.readBytesUntil()</b>	:	Untuk membaca data Byte yang diterima sampai ditemukan data yang sesuai
<b>Serial.setTimeout()</b>	:	Untuk menatur batas maksimum waktu tunggu (timeout) transmisi data
<b>Serial.write()</b>	:	Untuk mengirim data byte (numerik)
<b>Serial.serialEvent()</b>	:	Untuk interupsi pengiriman data serial

Selain fungsi-fungsi untuk komunikasi data serial, pada Arduino juga menyediakan Serial Monitor untuk melihat pengiriman data yang ada pada Arduino disertai jenis transmisi yang digunakan atau kecepatan pengiriman data serial. Untuk menggunakan Serial Monitor tersebut dapat dibuka dengan cara Pilih **Menu Bar Tools >> Pilih Serial Monitor** atau dengan cara cepat pada keyboard dengan menekan tombol **Ctrl + Shift + M**. Sebelum menggunakan Serial Monitor, yang perlu diperhatikan terlebih dahulu yaitu kondisi Arduino harus hidup yang mana terhubung

ke PC atau laptop dengan Aplikasi Arduino. Setelah dibuka dengan cara tersebut maka akan muncul tampilan jendela dari Serial Monitor seperti berikut.



**Gambar 4.1.** Tampilan Serial Monitor Arduino

Untuk komunikasi data serial dasar dapat dicoba dengan membuat program dibawah ini.

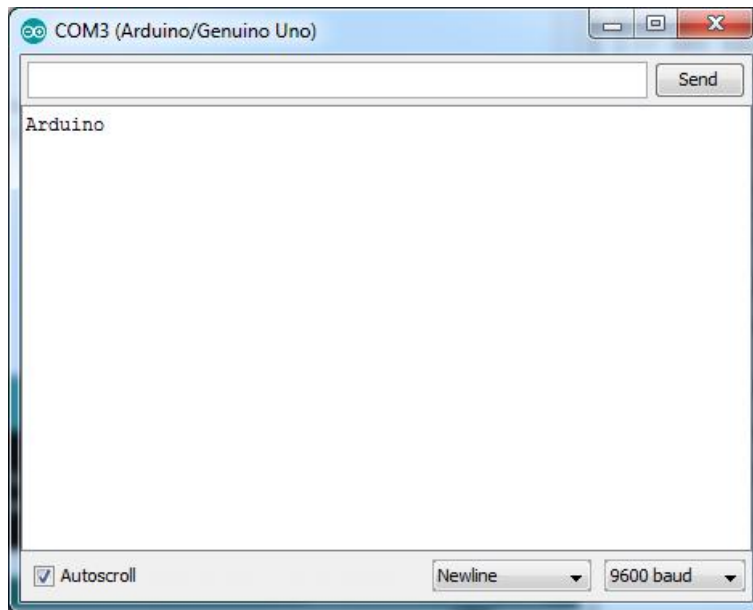
```
String nama = "Arduino";

void setup()
{
  // set baudrate 9600
  Serial.begin(9600);
}

void loop()
{
  // data yg diterima dikirim kembali.
  Serial.print(nama);
  delay(500);
}
```

**Modul 4.1.** Pengiriman Data Serial Sederhana

Dari program diatas pengiriman data menggunakan baudrate 9600. Program diatas yaitu untuk mengirim data berupa string dengan nama variabel nama yang memiliki nilai string yaitu Arduino. Data akan dikirim setiap 500 milidetik, untuk hasil dari program bisa dilihat pada gambar berikut.



**Gambar 4.2.** Tampilan Pengiriman data dilihat dari Serial Monitor Arduino

Pada komunikasi data serial perlu dideklarasikan Baudrate dari pengiriman data tersebut, terdapat banyak jenis baudrate untuk pengiriman data yang tersedia antara lain 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200. Untuk jenis Arduino memiliki batasan baudrate yang berbeda, sehingga ketika ingin menggunakan komunikasi data serial diharapkan tau spesifikasi dari Arduino yang digunakan. Semakin besar baudrate yang digunakan maka akan semakin cepat pengiriman datanya.

```
int angkaint = 123;
float angkafloat 123.5;
char datachar = 123;

void setup()
{
    // set baudrate = 19200
    Serial.begin(19200);
}

void loop()
{
    // kirim string/text
    Serial.print(""apa kabar"");

    // kirim tipe angka integer 123
    Serial.println(angkaint);

    // kirim angka type float 123.5
    Serial.println(angkafloat);

    // kirim code ascii 123 yaitu tanda kurung buka {
    Serial.println(datachar);
}
```

```

// kirim angka dlm bentuk heksa , akan dikirim 7BH
Serial.println(angkaInt,HEX);

// kirim angka dlm bentuk biner , 0111 1011
Serial.println(angkaInt,BIN);
}

```

#### **Modul 4.2. Contoh Pengiriman Data Serial dengan Tipe Data**

Pada saat pengiriman data terdapat interupsi yang dapat mengeksekusi data secara langsung ketika terdapat data masuk. Ketika terdapat data maka yang akan dieksekusi yaitu fungsi `serialEvent()`. Berikut contoh interupsi data pada pengiriman data serial.

```

String inputString = "";
boolean stringComplete = false;

void setup()
{
  Serial.begin(9600);
  inputString.reserve(200);
}

void loop()
{
  if (stringComplete)
  {
    Serial.println(inputString);
    inputString = "";
    stringComplete = false;
  }
}

void serialEvent()
{
  while (Serial.available())
  {
    char inChar = (char)Serial.read();
    inputString += inChar;
    if (inChar == '\n')
    {
      stringComplete = true;
    }
  }
}

```

#### **Modul 4.3. Contoh Pengiriman Data Serial dengan Interupsi**

## TUGAS PRAKTIKUM

1. Buatlah program yang bisa mencetak karakter set ASCII dan bentuk desimalnya dimulai dari **whitespace (spasi) sampai tilde (~)**. Susunlah output serial program seperti contoh berikut.

```
...  
A = 65  
B = 66  
...
```

Pastikan program hanya menampilkan whitespace hingga tilde saja (artinya apabila sampai tilde maka kembali lagi menampilkan whitespace dan berulang seterusnya)