

# **Teknik Antarmuka Komputer**

## **#4 Antarmuka Komunikasi Serial**

Eka Maulana, ST, MEng.

# Tujuan Kuliah

**Setelah mengikuti perkuliahan ini, mahasiswa dapat:**

1. Menjelaskan mekanisme komunikasi serial
2. Mengetahui jenis-jenis protokol komunikasi serial
3. Memahami antarmuka komunikasi serial
4. menganalisis teknik antarmuka serial
5. Mengetahui teknik pemrograman Komunikasi Serial

# Outline Materi

- Parameter Komunikasi Serial
- Jenis Komunikasi Serial
- Komunikasi Asinkron/ Sinkron
- Teknik antarmuka serial
- Jenis pengembangan komunikasi serial
- Aplikasi teknik pemrograman antarmuka serial

# Jenis Komunikasi Serial

- Sinkron

Data dikirim bersama dengan sinyal clock

- I<sup>2</sup>C (Inter-Integrated Circuit)
- SPI (Serial Peripheral Interface)
  - Keyboard, mouse

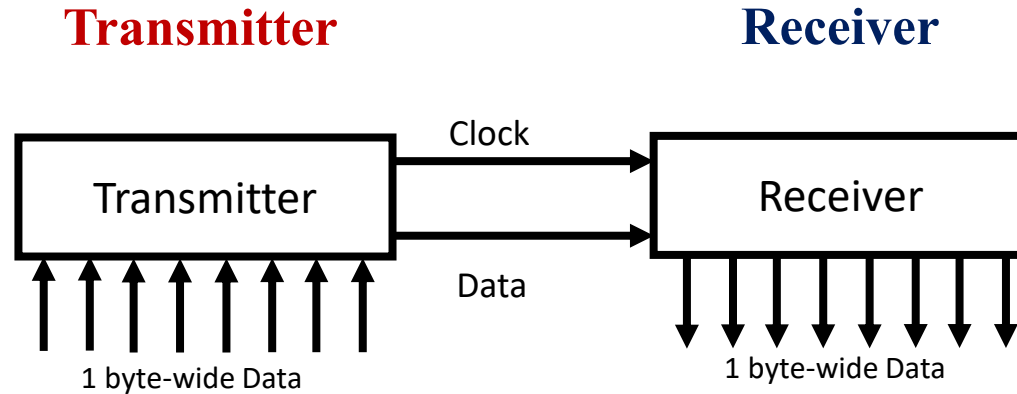
- Asinkron

Data dikirim tanpa sinyal clock

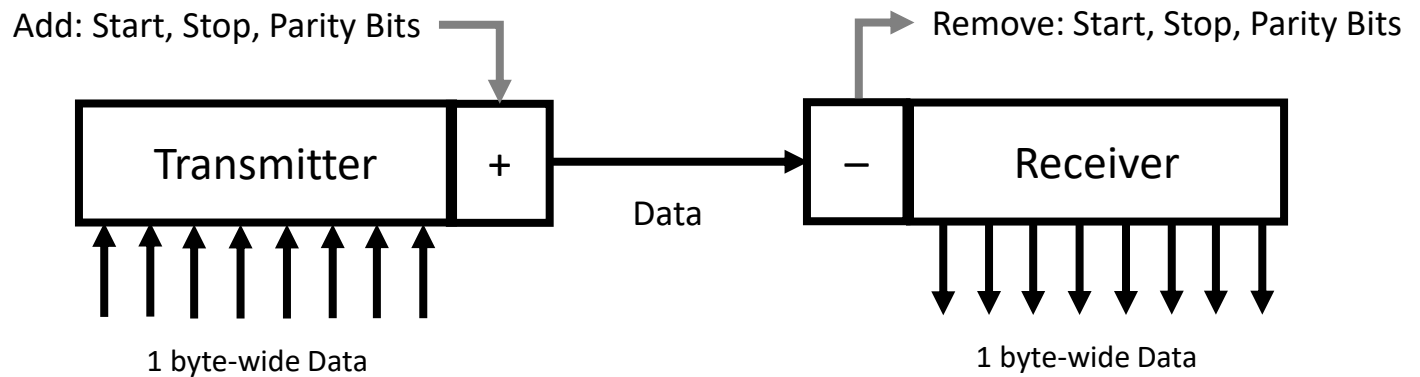
- UART (RS-232)
  - Modem

# Komunikasi Sinkron vs Asinkron

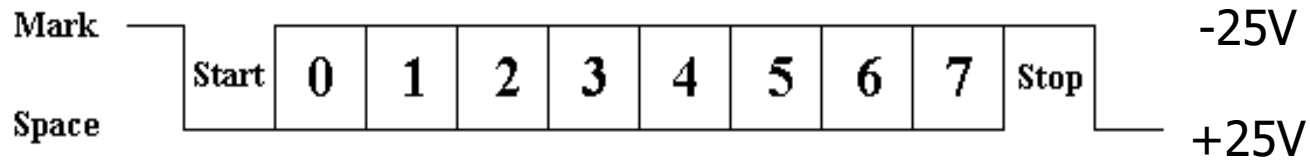
## SINKRON



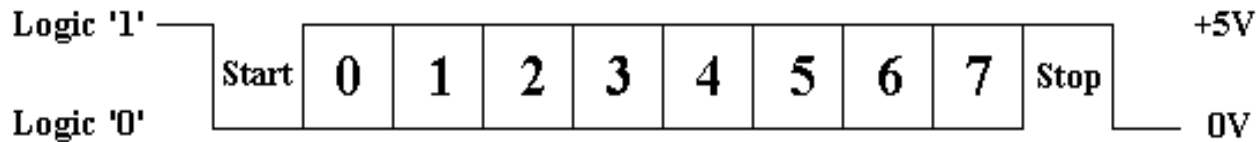
## ASINKRON



# Logika Serial



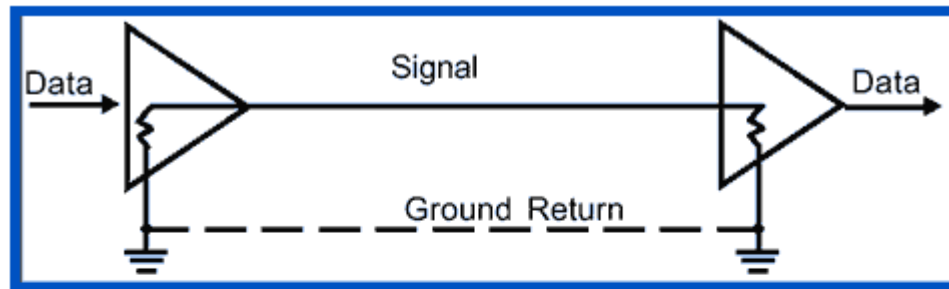
RS-232 Logic Waveform



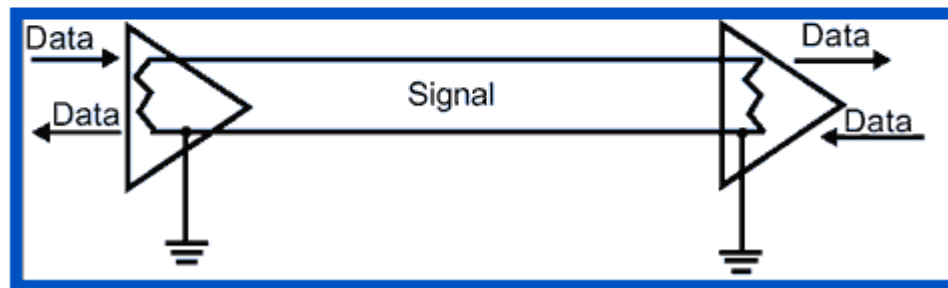
TTL/CMOS Serial Logic Waveform

# RS-232

- Kecepatan max. 115,200 bps
- Panjang media/kabel max 60 meter
- Single-ended (terhadap ground)
- Voltage swing  $\pm 25V$
- Short circuit max 500mA



Unbalanced Single-Wire Transmission



Unbalanced Two-Wire Transmission

# Port Address

- Com1                      IRQ4                      3F8~3FF
- Com2                      IRQ3                      2F8~2FF
- **Address Specification**
  - PortAddr                      3F8
  - PortAddr+1                      3F9
  - ⋮
  - PortAddr+7                      3FF



# Parameter Komunikasi RS232

- Baud Rate
  - 150, 300, 1200, ..., 9600, 19200, 38400, 115200
- Parity
  - None , Even , Odd
- Data Bit
  - 7 , 8
- Stop Bit
  - 1 , 2
- Start Bit
  - 1 , 1.5

# a Pemrograman yang mendukung Kom Serial

- C, C++, C#, F#
- Basic, VB, VB.NET
- Pascal, Delphi
- Python, Java
- PHP, ASP, Javascript
- Ruby, Perl
- Erlang, Scala
- Haskel, Groovy
- dll

# Pemrograman Serial pada Visual Basic

- Dengan komponen MSComm

## *Setup:*

```
MSComm1.CommPort = 1
```

```
MSComm1.Settings = "9600,N,8,1"
```

```
MSComm1.InputLen = 0
```

```
MSComm1.PortOpen = True
```

## *Baca/tulis:*

```
MSComm1.Output = "data"
```

```
Baca = MSComm1.Input
```

```
MSComm1.PortOpen = False
```

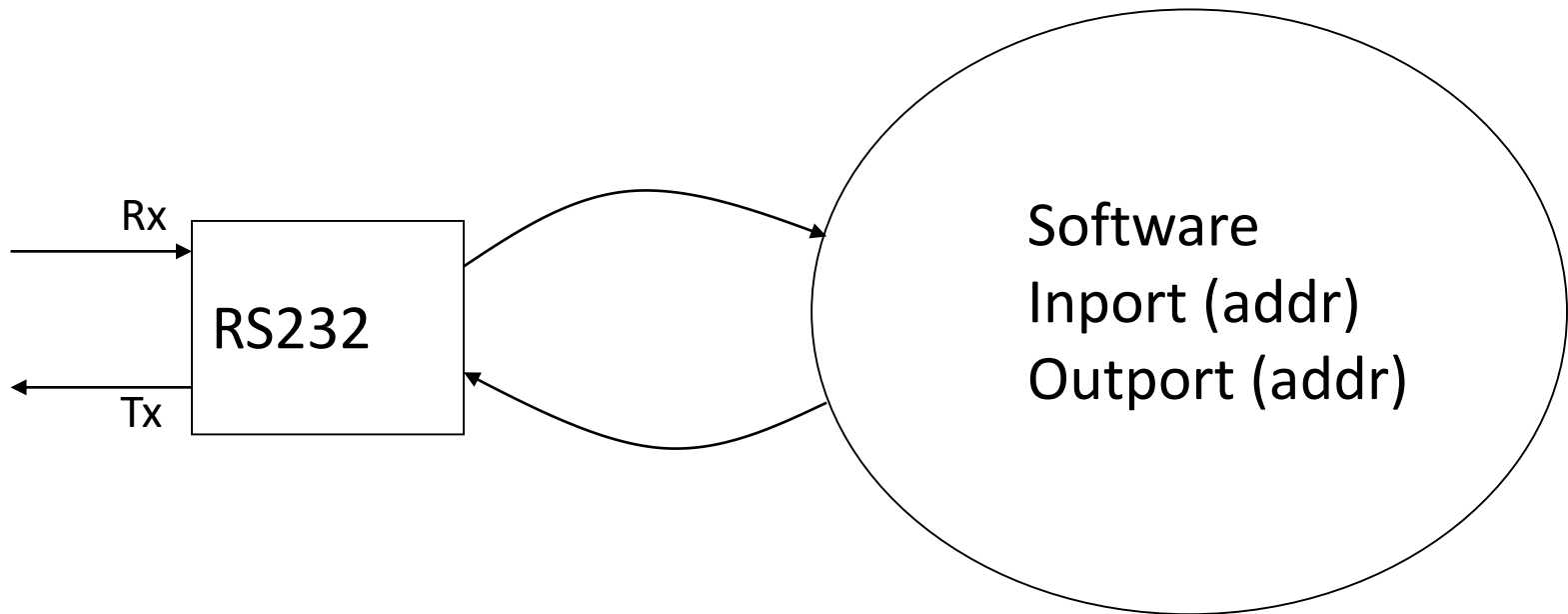
# Pemrograman Serial pada C#

```
SerialPort(string portName,  
int baudRate)
```

```
SerialPort mySerialPort = new SerialPort( "COM3", 9600);
```

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.IO;  
using System.IO.Ports;  
  
namespace Serial_Test  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            SerialPort mySerialPort = new SerialPort("COM3", 9600);  
            try  
            {  
                mySerialPort.Open();  
                mySerialPort.Close();  
            }  
            catch (IOException ex)  
            {  
                Console.WriteLine(ex);  
            }  
        }  
    }  
}
```

# (Polling)



# C (1)

- `#include <stdio.h>`
- `#include <dos.h>`
- `unsigned char i, In_data, IMR, MCR, IIR;`
- `int addr;`
  
- `void main(void)`
- `{`
- `addr=0x2f8;`
  
- `/* set baud rate */`
- `outportb(addr+3,inportb(addr+3) | 0x80);`
- `outportb(addr,0x0c);`
- `outportb(addr+1,0x00);`

# C (2)

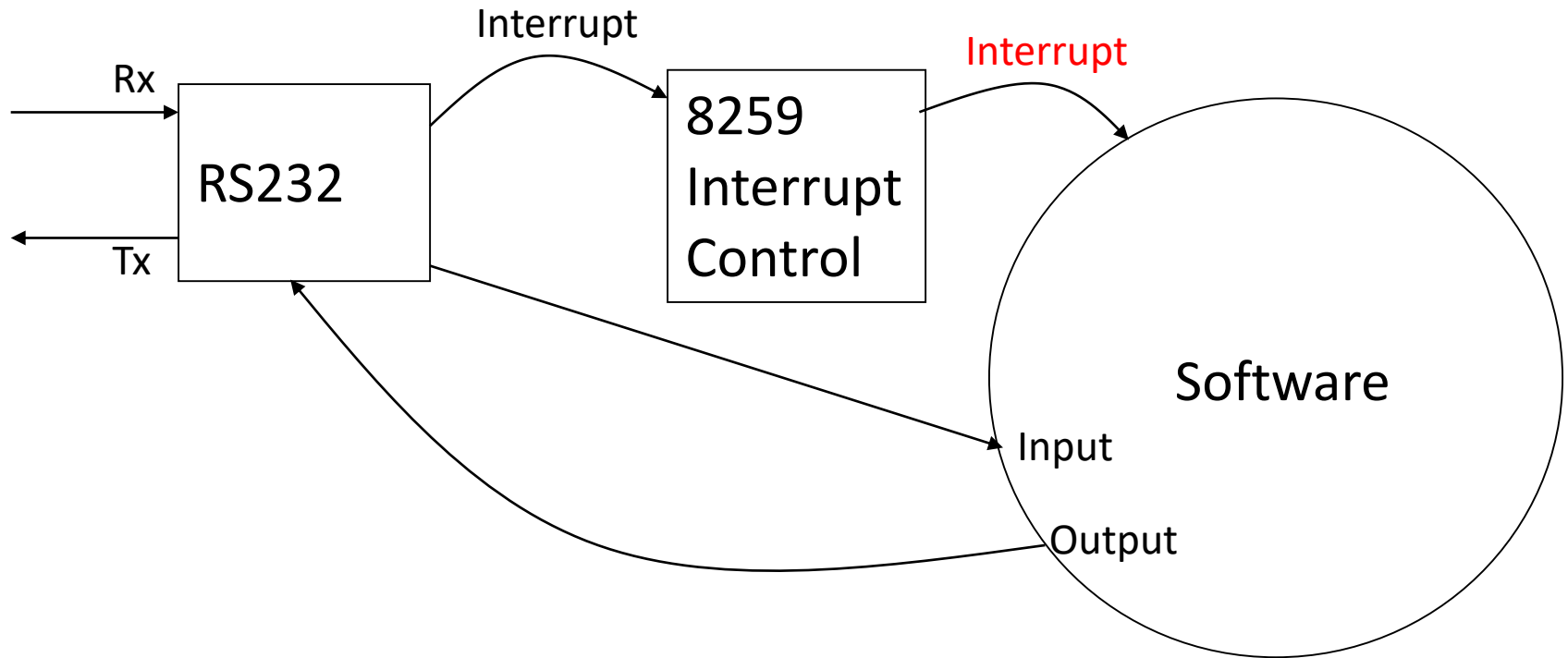
- `/* set communication parameter */`
- `outportb(addr+3,0x1b);`
  
- `/* set interrupt enable register */`
- `outportb(addr+1,0x01);`
  
- `/* set out1, out2, and internal loopback */`
- `MCR=0x1b;`
- `outportb(addr+4,MCR);`
- `delay(100);`
  
- `for (i='a';i<='d';i++)`
- `{`
- `In_data='@';`
- `outportb(addr,i);`
- `delay(1000);`

# C (3)

- `/* get Interrupt Identification register */`
- `IIR=inportb(addr+2)&0x07;`
- `In_data=inportb(addr);`
- `printf("received %c %d\n",In_data,IIR);`
- `}`
- `exit(0);`
- `}`



# Interrupt



# C (1)

```
#include <stdio.h>
#include <dos.h>
unsigned char Int_id,i, In_data,IMR,MCR;
int addr;

void interrupt comm5()
{
    disable();
    In_data=inportb(addr);
    outportb(0x20,0x20);
    enable();
}
void main(void)
{
    addr=0x2f8;
    /* assign Interrupt Service Routine address to IRQ3 */
```

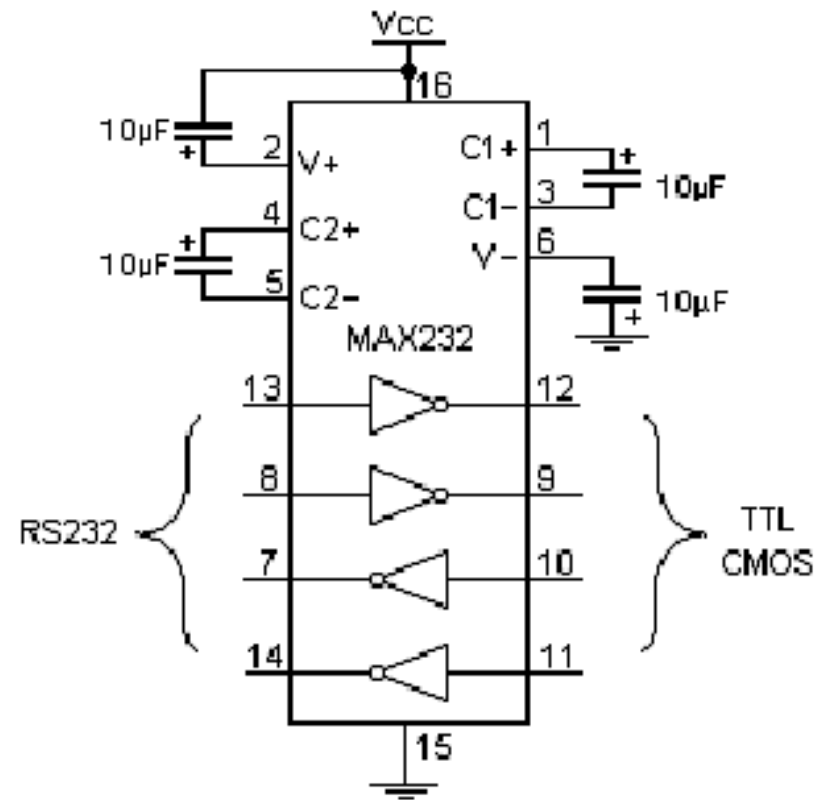
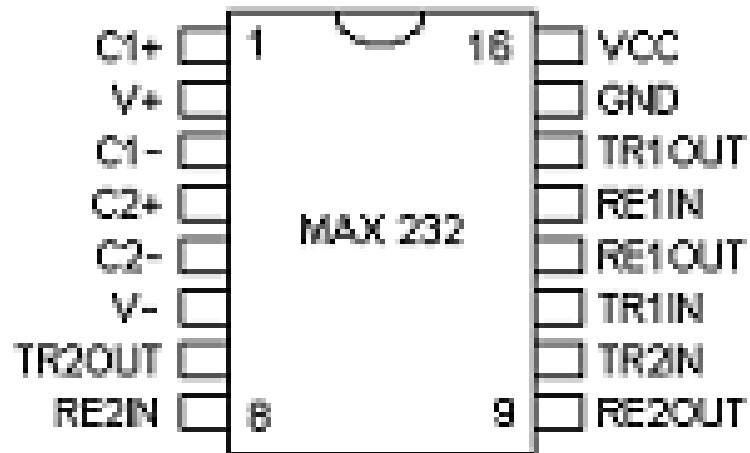
# C (2)

```
disable();
setvect(0x0b,comm5);
enable();
/* enable Interrupt mask register */
IMR=inportb(0x21);
outportb(0x21,IMR&0xf7);
/* set baud rate */
outportb(addr+3,inportb(addr+3) | 0x80);
outportb(addr,0x0c);
outportb(addr+1,0x00);
/* set communication parameter*/
outportb(addr+3,0x1b);
/* set interrupt enable register */
outportb(addr+1,0x01);
```

# C (3)

```
/* set out1, out2, and use external loopback */
MCR=0x0b;
outportb(addr+4,MCR);
delay(100);
for (i='a';i<='d';i++)
{
In_data='@';
outportb(addr,i);
delay(1000);
printf("received %c %d\n",In_data,inportb(addr+2)&0x07);
}
outportb(0x21,IMR|0x01);
exit(0);
}
```

# Profil MAX232 (Converter)



# RS232 Converter Schematic

