

LAPORAN PRAKTIKUM

KOMUNIKASI SERIAL PC TO PC

Disusun untuk Memenuhi Tugas Mata Kuliah Praktikum Komunikasi Data
Semester 2

Dosen Pengampu:

Lis Diana Mustafa, S.T., M.T.



Penyusun:

Fadlillah Bashir Al Hakim

1C-D3TT

NIM 2231130059

JURUSAN TEKNIK ELEKTRO

PROGRAM STUDI TEKNIK TELEKOMUNIKASI

POLITEKNIK NEGERI MALANG

2023

DAFTAR ISI

COVER JUDUL	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iii
DAFTAR TABEL	iv
3.1 Tujuan Khusus.....	1
3.2 Tujuan Umum	1
3.3 Alat dan Bahan yang Digunakan.....	1
3.4 Landasan Teori	1
3.4.1 Komunikasi Serial (RS232)	1
3.4.2 Parameter Komunikasi Serial	2
3.4.3 Konfigurasi Pengkabelan Komunikasi Serial	4
3.5 Prosedur Praktikum	5
3.6 Flowchart Program.....	7
3.7 Listing Program	7
3.8 Tampilan Hasil Program.....	9
3.9 Analisa.....	10
3.9.1 Data Analisa.....	10
3.9.1 Analisa Masalah.....	15
3.10 Kesimpulan.....	15

DAFTAR GAMBAR

Gambar 3.1 Transmisi Serial	2
Gambar 3.2 Arah transmisi komunikasi serial.....	2
Gambar 3.3 Gelombang informasi komunikasi serial	3
Gambar 3.4 Konfigurasi pin out DB9.....	5
Gambar 3.5 Flowchart program	7
Gambar 3.6 Contoh tampilan hasil program.....	9
Gambar 3.7 Contoh tampilan setup dari Comport	9

DAFTAR TABEL

Tabel 3.1 Fungsi dari masing-masing pin keluaran	4
Tabel 3.2 Setting property komponen.....	6

PRAKTIKUM-III

KOMUNIKASI SERIAL PC TO PC

3.1 Tujuan Khusus

1. Mahasiswa mampu melakukan komunikasi antara dua laptop menggunakan interface serial RS 232.
2. Mahasiswa mampu memahami sistem pengkabelan untuk menghubungkan dua laptop melalui port serial.
3. Mahasiswa mampu memahami standarisasi komunikasi serial RS 232.
4. Mahasiswa mampu memahami prosedur untuk melakukan komunikasi serial RS 232.

3.2 Tujuan Umum

1. Mahasiswa mampu memahami komponen untuk melakukan komunikasi serial RS 232 menggunakan perangkat lunak *Delphi 7*.
2. Mahasiswa mampu melakukan instalasi komponen Comport komunikasi serial RS 232.
3. Mahasiswa mampu membuat program untuk komunikasi antara PC dengan PC.

3.3 Alat dan Bahan yang Digunakan

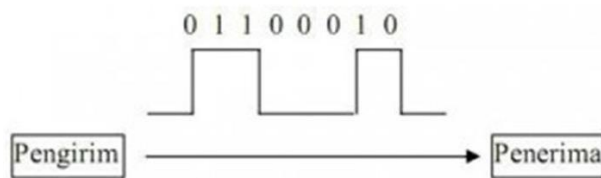
- | | |
|--|------------|
| 1. Perangkat Laptop/PC. | 2 buah |
| 2. Perangkat lunak <i>Delphi 7 Community</i> atau versi lainnya. | 1 buah |
| 3. Komponen <i>Delphi</i> komunikasi serial <i>Comport</i> . | 1 buah |
| 4. Konverter USB to RS232 tipe DB9. | 2 buah |
| 5. Kabel <i>jumper male to male</i> . | Secukupnya |

3.4 Landasan Teori

3.4.1 Komunikasi Serial (RS232)

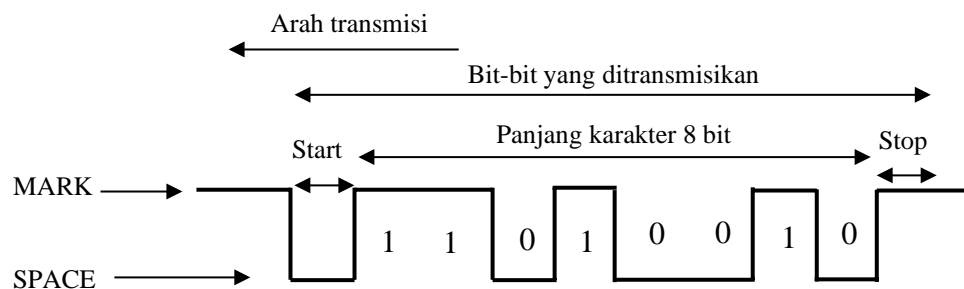
Komunikasi serial merupakan komunikasi yang pengiriman datanya dikirimkan per elemen. Dengan transmisi serial pengiriman data jarak jauh menjadi lebih efektif dibandingkan dengan transmisi paralel. Data paralel internal komputer dimasukkan ke pengubah paralel ke serial. Saluran serial mengirimkan setiap karakter per elemen sehingga hanya diperlukan satu atau dua penghantar, yaitu kirim data (TX) dan terima data (RX). Contoh transmisi serial ditunjukkan pada Gambar 3.1.

Transmisi Serial



Gambar 3.1 Transmisi Serial

Pada Gambar 3.2 ditampilkan bahwa transmisi serial membutuhkan waktu yang relatif lebih lama dibandingkan dengan transmisi paralel, hal ini dikarenakan untuk komunikasi serial hanya menggunakan satu saluran. Berikut ini gambar arah transmisi dari komunikasi serial.



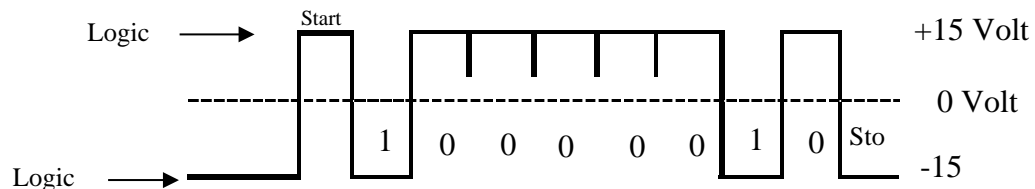
Gambar 3.2 Arah transmisi komunikasi serial

Sebagai contoh, jika akan dikirim data serial 10011010, maka agar data tersebut dapat dikirim dan diterima dengan baik, selang waktu yang digunakan oleh pengirim dari penerima satu dengan yang lain harus sama. Jika penerima telah menerima penyesuaian bit, maka seharusnya juga harus segera menerima penyesuaian karakter, dan penerima juga harus mengetahui awal dan akhir blok data yang dikirim. Penyesuaian yang diperlukan dapat diperoleh secara sinkron maupun asinkron, data yang dikirim oleh terminal komputer lewat jalur RX dimasukan ke pengubah seri ke paralel sebelum diteruskan ke komputer.

3.4.2 Parameter Komunikasi Serial

Pada komunikasi serial data yang dikirimkan berupa bit '1' (*high*) atau bit '0' (*low*), untuk bit '1' diwakili sebagai tegangan +3 s/d +15 *volt* dan bit '0' diwakili sebagai tegangan -3 s/d -15 *volt* dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimal sebesar 30 *volt*.

Suatu karakter yang akan ditransfer harus didahului oleh kondisi (*high*) ke rendah (*low*) yang dinamakan *start bit*, yang digunakan untuk mensinkronkan antara pengirim dan penerima. Setelah *start bit*, selanjutnya berisi karakter yang diikuti dengan *parity bit* dan terakhir adalah *stop bit*. Berikut ini contoh sinyal informasi komunikasi serial RS232 pada pengiriman huruf "A" dalam format *ASCII* tanpa bit paritas.



Gambar 3.3 Gelombang informasi komunikasi serial

Gambar 3.3 menunjukkan bentuk gelombang komunikasi serial dengan format 8N1, yaitu 8 bit-data, tanpa *parity* dan 1 stop bit. Pada keadaan *idle* atau menganggur, jalur RS232 ditandai dengan *mark state* atau logika high. Pengiriman data diawali dengan *start bit* yang berlogika nol '0' atau *low*, berikutnya data dikirimkan bit demi bit mulai dari *LSB (Least Significant Bit)* atau bit ke-0 (nol). Pengiriman setiap *byte* diakhiri dengan *stop bit* yang berlogika *high*.

Gambar 3.3 memperlihatkan kondisi *low* setiap stop bit, ini adalah start bit yang menandakan data berikutnya akan dikirimkan. Jika tidak ada lagi data yang ingin dikirim, maka jalur transmisi ini akan dibiarkan dalam keadaan high. Ada yang disebut *break* sinyal, yaitu keadaan low yang lamanya cukup untuk mengirimkan 8-bit data. Jika pengirim menyebabkan jalur komunikasi dalam keadaan seperti ini, penerima akan menganggap ini adalah *break* sinyal atau sinyal rusak.

Data yang dikirimkan dengan cara seperti gambar 2.11 ini disebut data yang terbingkai (*to be framed*) oleh *start* dan *stop bit*. Jika *stop bit* dalam keadaan *low*, berarti telah terjadi *framing error*. Biasanya hal ini terjadi karena perbedaan kecepatan komunikasi (*baudrate*) antara pengirim dan penerima. Berikut ini beberapa parameter dari komunikasi serial, antara lain :

- *Data Bits*

Jumlah bit yang akan ditransmisikan mempunyai nilai adalah 5 sampai 8 bit

- *Parity Bit*

Untuk mendeteksi kesalahan (*error*) yang berbentuk *odd* (ganjil), *even* (genap) atau tanpa *parity*.

- *Start Bit* dan *Stop Bit*

Start Bit digunakan untuk sinkronisasi penerima, start bit berukuran 1 bit.

Stop Bit menandakan akhir dari data yang dikirimkan, *stop bit* berukuran 1 bit, 1.5 bit atau 2 bit.

- *Baud Rate* atau kecepatan data (bps)

Merupakan kecepatan transmisi data dari *transmitter* ke *receiver*, mempunyai nilai antara (50 s/d 19,2 kbps).

3.4.3 Konfigurasi Pengkabelan Komunikasi Serial

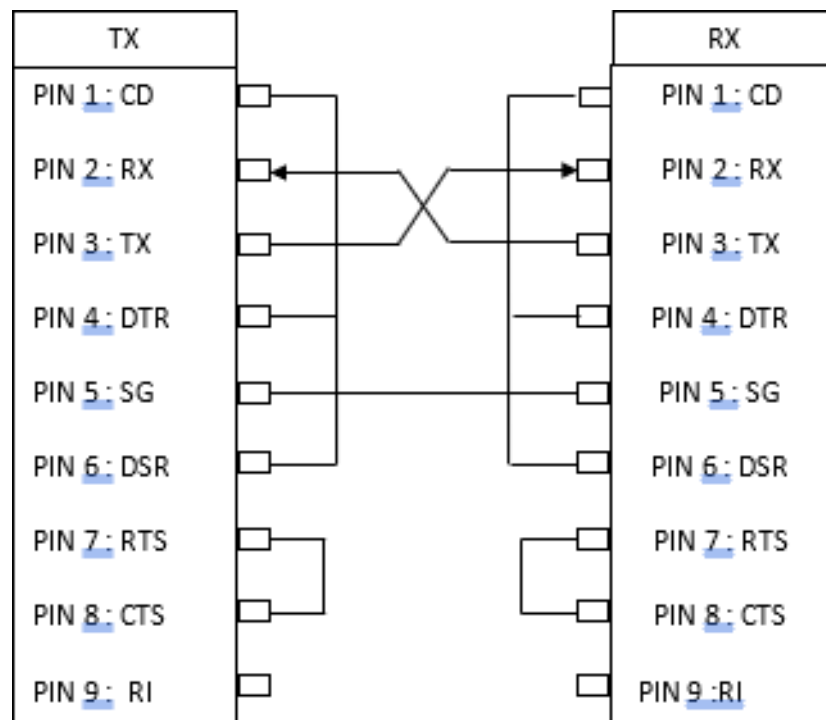
Untuk dapat melakukan komunikasi serial antara dua komputer maka di butuhkan media kabel untuk mengirimkan informasi. Dalam komunikasi serial antara dua komputer dibutuhkan media kabel serta konektor yang berupa DB9 (*female*). DB9 (*female*) memiliki 9 pin out yang tiap-tiap pin out memiliki fungsi masing-masing. Pada Tabel 3.1 dibawah ini, menunjukkan konfigurasi pin out beserta fungsi dari masing-masing pin.

Tabel 3.1 Fungsi dari masing-masing pin keluaran

Fungsi Pin Out	Keterangan	Fungsi
TX	Transmit Data	Untuk pengiriman data serial
RX	Receive Data	Untuk penerimaan data serial
CTS	Clear To Send	Digunakan untuk memberitahukan bahwa modem siap untuk melakukan pertukaran data
CD	Carrier Detect	Saat modem mendeteksi suatu ' <i>carrier</i> ' dari modem lain (dari tempat lain) maka sinyal ini akan diaktifkan
DSR	Data Set Ready	Memberitahukan UART bahwa modem siap untuk melakukan komunikasi (<i>Link</i>)

Fungsi Pin Out	Keterangan	Fungsi
DTR	Data Terminal Ready	Kebalikan dari DSR, untuk memberitahukan bahwa UAT siap untuk melakukan hubungan komunikasi
RTS	Request To Send	Sinyal untuk menginformasikan modem bahwa UART siap untuk melakukan pertukaran data
RI	Ring Induktor	Akan aktif jika modem mendeteksi adanya sinyal dering dari saluran telepon

Untuk konfigurasi pin out antara DB9 (*female*) dengan DB9 (*female*) yang lain ditunjukkan pada gambar berikut:



Gambar 3.4 Konfigurasi pin out DB9

Pada gambar 3.4 di atas merupakan konfigurasi untuk komunikasi antara dua komputer yang menggunakan koneksi *point-to-point*, artinya hanya komunikasi dalam mode *half duplex*.

3.5 Prosedur Praktikum

1. Buka dan pelajari terlebih dahulu modul praktikum komunikasi serial (PC to PC) yang telah diberikan oleh Dosen.

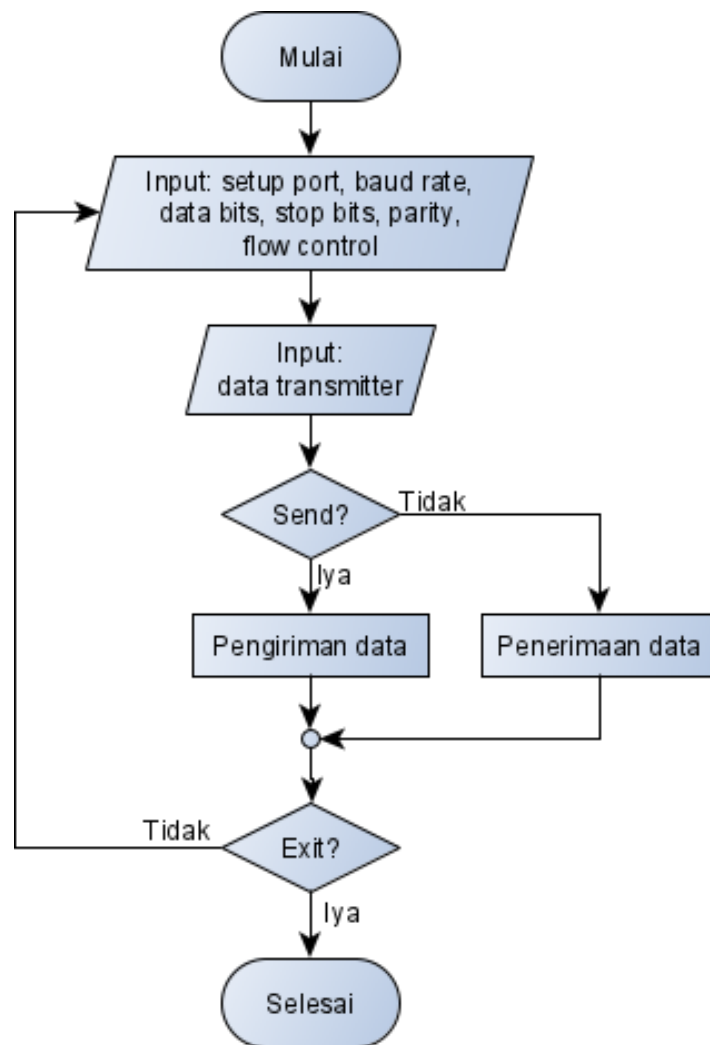
2. Buka *software* Delphi, pilih menu *New* lalu *Application* untuk membuat *form* baru
3. Selalu *save all* terlebih dahulu sebelum mengerjakan sebuah program.
4. Lalu siapkan komponen pada *form* tersebut seperti, 2 buah *label*, 2 buah *memo*, 1 buah *edit*, 3 buah *button*, dan 1 buah *comport1* . Selanjutnya atur tata letak komponen – komponen tersebut sesuai dengan modul yang telah diberikan.
5. Ubah *property* komponen-komponen tersebut menjadi sesuai Tabel 3.2.

Tabel 3.2 *Setting property* komponen

Komponen	<i>Property</i>	<i>Setting</i>
Form	<i>Caption</i>	Komunikasi PC to PC
	<i>Name</i>	Frmpctopc
Label1	<i>Caption</i>	Kirim
Label2	<i>Caption</i>	Terima
Memo1	<i>Lines</i>	(kosongkan string list editor)
	<i>Name</i>	Mekirim
Memo2	<i>Lines</i>	(kosongkan string list editor)
	<i>Name</i>	Meterima
Button1	<i>Caption</i>	&Send
	<i>Name</i>	Btsend
Button2	<i>Caption</i>	Se&tup
	<i>Name</i>	Btsetup
Button3	<i>Caption</i>	E&xit
	<i>Name</i>	Btexit
Edit1	<i>Caption</i>	(kosongkan)
	<i>Name</i>	Edinputteks
Comport1	<i>Name</i>	Comserial

6. Kemudian masukkan kode script dengan men-*double click* pada setiap komponen yang telah ditentukan pada modul komunikasi serial.
7. Jika sudah selesai, selanjutnya dapat di-*run* untuk menjalankan program dengan menekan tombol F9 atau klik run pada *software* Delphi.

3.6 Flowchart Program



Gambar 3.5 Flowchart program

3.7 Listing Program

```
unit Latihan_3_1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, CPort;

type
  TFrmpctopc = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Mengirim: TMemo;
    Menerima: TMemo;
    Comserial: TComPort;
```

```

    GroupBox1: TGroupBox;
    Btsend: TButton;
    Btsetup: TButton;
    Btexit: TButton;
    Edinputteks: TEdit;
    procedure BtsendClick(Sender: TObject);
    procedure BtsetupClick(Sender: TObject);
    procedure BtexitClick(Sender: TObject);
    procedure ComserialRxChar(Sender: TObject; Count:
Integer);
    procedure FormCreate(Sender: TObject);
    private
    { Private declarations }
    public
    { Public declarations }
    end;

var
    Frmpctopc: TFrmpctopc;

implementation

{$R *.dfm}

procedure TFrmpctopc.ComserialRxChar(Sender: TObject; Count:
Integer);
var
    str:string;
begin
    Comserial.ReadStr(Str, Count);
    Menerima.Text := Menerima.Text + str;
    Menerima.Lines.Add('');
end;

procedure TFrmpctopc.FormCreate(Sender: TObject);
begin
    Comserial.Open;
end;

procedure TFrmpctopc.BtsendClick(Sender: TObject);
var
    i,n:integer;
    str:string;
begin
    Str := Edinputteks.Text;
    Comserial.WriteStr(Str);
    Mengirim.Lines.Add('Saya :' + Edinputteks.Text);
end;

procedure TFrmpctopc.BtsetupClick(Sender: TObject);
begin
    Comserial.ShowSetupDialog;

```

```
end;
```

```
procedure TFrmpctopc.BtexitClick(Sender: TObject);  
begin
```

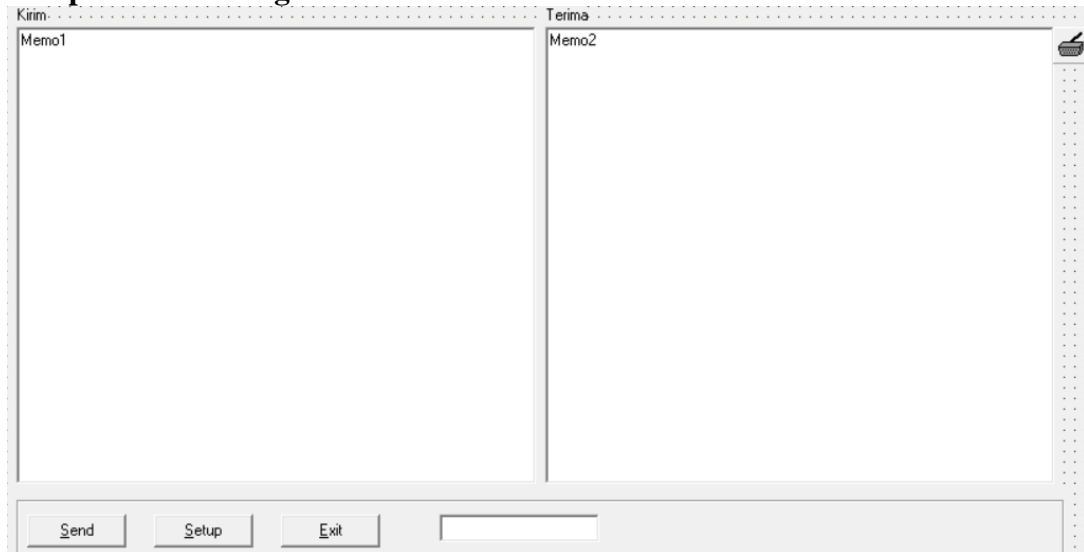
```
    Comserial.close; //menutup port com
```

```
    application.terminate; //keluar dari program
```

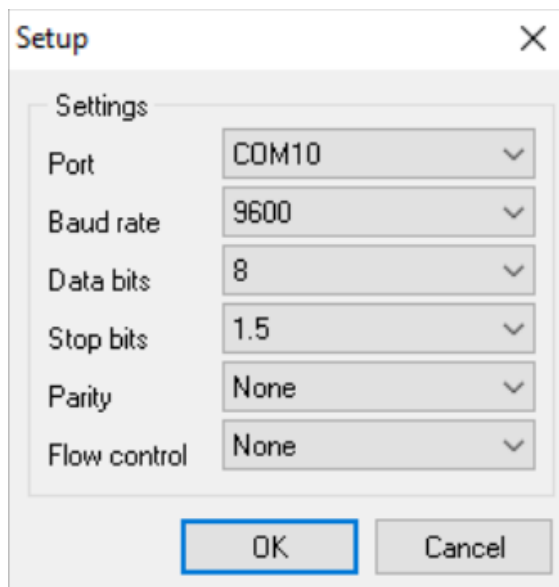
```
end;
```

```
end.
```

3.8 Tampilan Hasil Program



Gambar 3.6 Contoh tampilan hasil program



Gambar 3.7 Contoh tampilan setup dari Comport

3.9 Analisa

3.9.1 Data Analisa

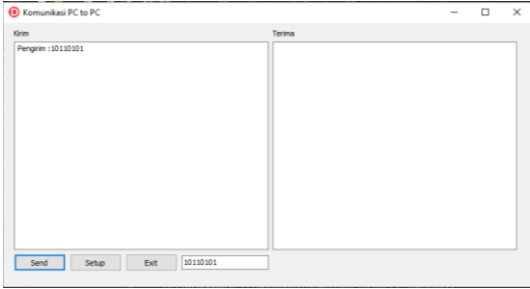
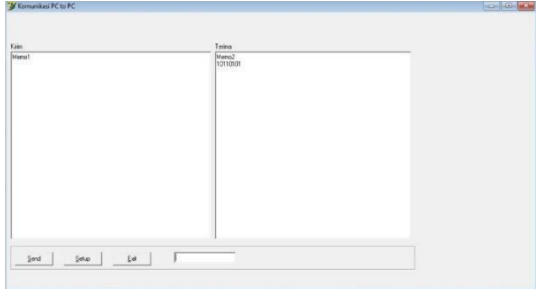
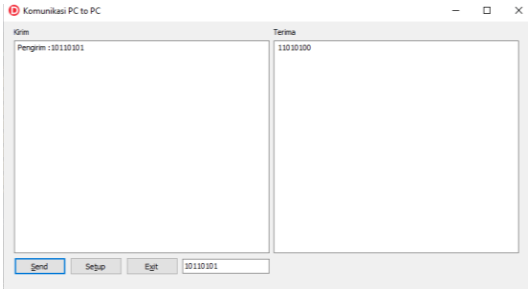
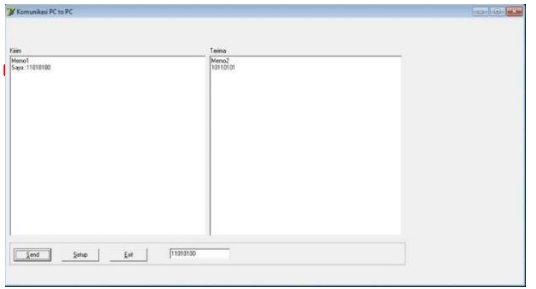
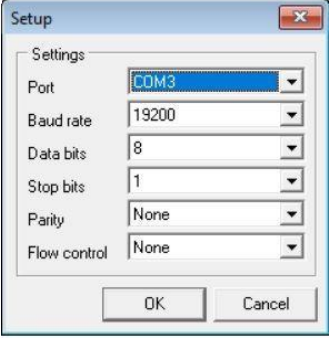
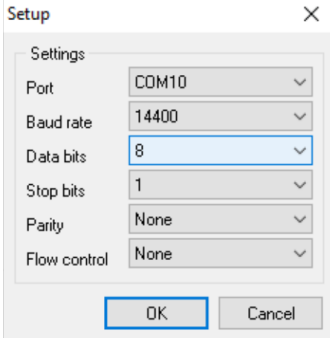
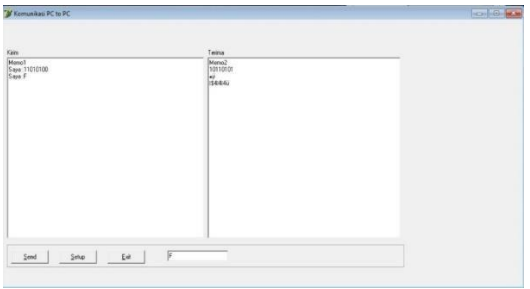
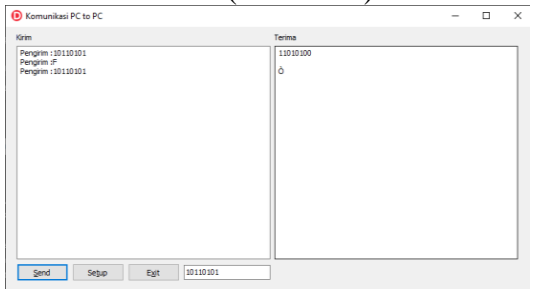
1. Data berupa bit

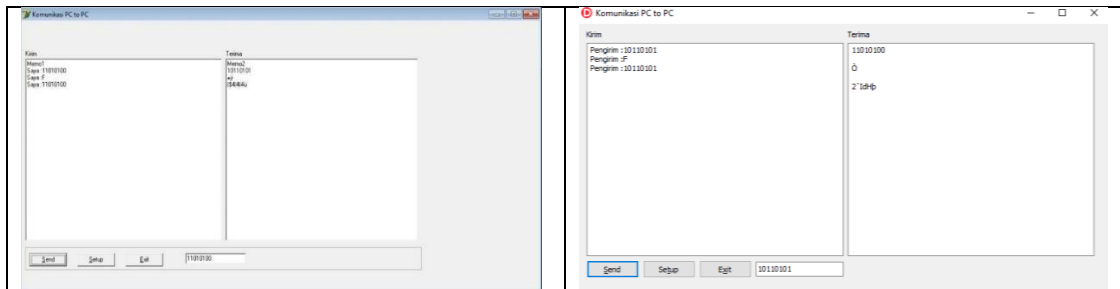
Setup Sama antar Client	
Client 1	Client 1
TX: "10110101"	RX: "10110101"
RX: "11010100"	TX: "11010100"
Setup Berbeda: Baud Rate	
Client 1, Baud rate: 14400	Client 2, Baud rate: 19200
TX: "10110101"	RX: (data salah)
RX: (data salah)	TX: "11010100"
Setup Berbeda: Stop Bits	
Client 1, Stop bits: 1,5	Client 2, Stop bits: 1
TX: "10110101"	RX: "10110101"
RX: "11010100"	TX: "11010100"
Setup Berbeda: Parity	
Client 1: <u>Error run program</u>	Client 2: <u>Error run program</u>
-	-
-	-

2. Data berupa karakter (*char*)

Setup Berbeda: Baud Rate	
Client 1, Baud rate:	Client 2, Baud rate:
TX: "F"	RX: (data salah)
RX: (data salah)	TX: "F"
Setup Berbeda: Stop Bits	
Client 1, Stop bits:	Client 2, Stop bits:
TX: "F"	RX: "F"
RX: "F"	TX: "F"
Setup Berbeda: Data bits	
Client 1, Data bits:	Client 2, Data bits:
TX: "F"	RX: (data salah)
RX: (data salah)	TX: "F"

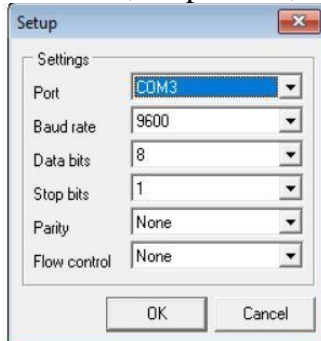
3. Lampiran tangkapan layar saat sesudah dilakukan rangkaian percobaan.

Setup Sama antar Client	
Client 1	Client 1
<p>TX</p> 	<p>RX</p> 
<p>RX</p> 	<p>TX</p> 
Setup Berbeda: Baud Rate	
<p>Client 1, Baud rate: 14400</p> 	<p>Client 2, Baud rate: 19200</p> 
<p>TX: "10110101"</p> 	<p>RX: (data salah)</p> 
<p>RX: (data salah)</p>	<p>TX: "11010100"</p>

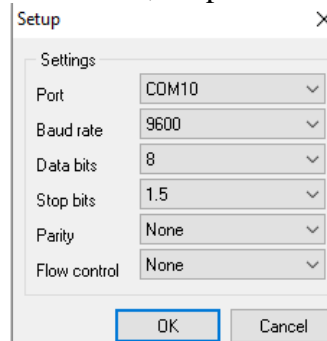


Setup Berbeda: Stop Bits

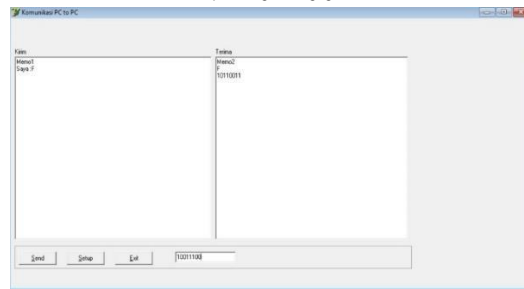
Client 1, Stop bits: 1,5



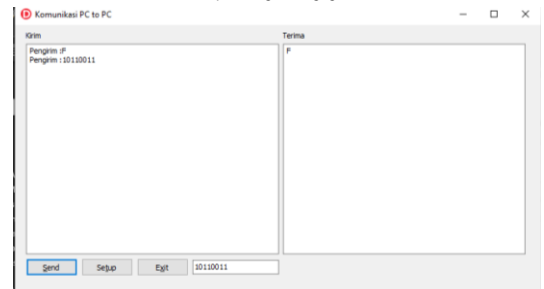
Client 2, Stop bits: 1



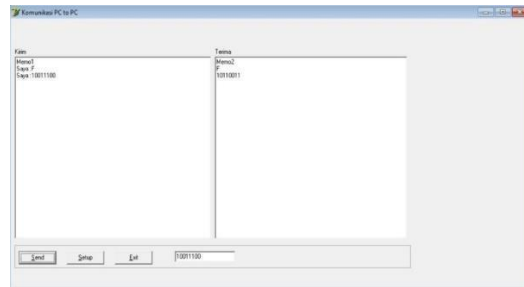
TX: "10110011"



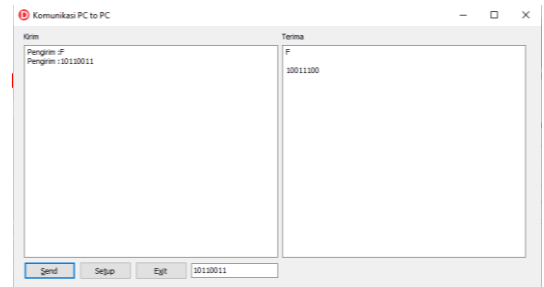
RX: "10110011"



RX: "11010100"

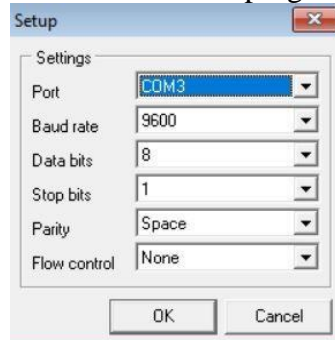


TX: "11010100"

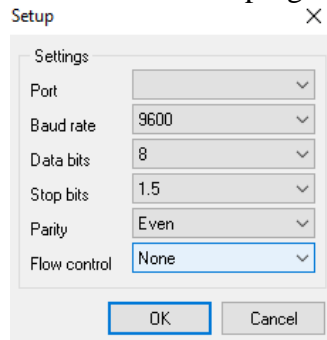


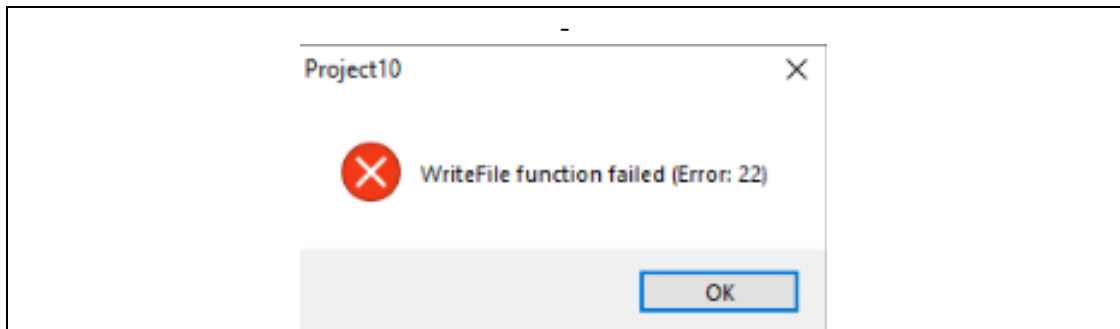
Setup Berbeda: Parity

Client 1: Error run program



Client 2: Error run program



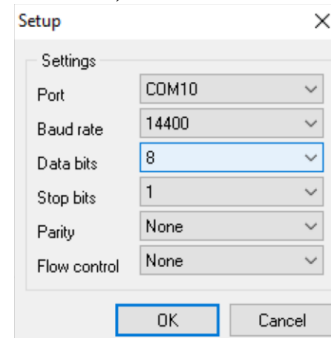


Setup Berbeda: Baud Rate

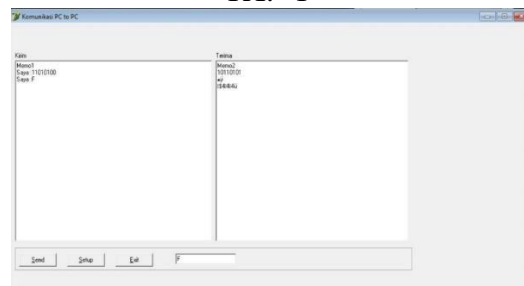
Client 1, Baud rate: 14400



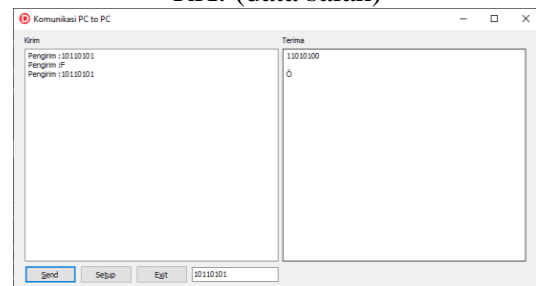
Client 2, Baud rate: 19200



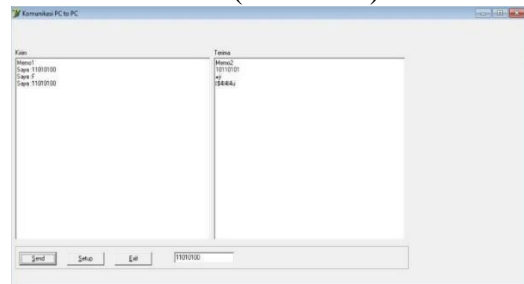
TX: "F"



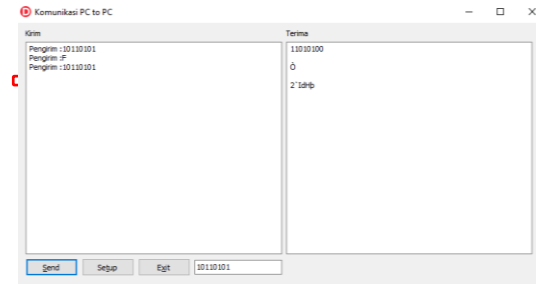
RX: (data salah)



RX: (data salah)



TX: "F"

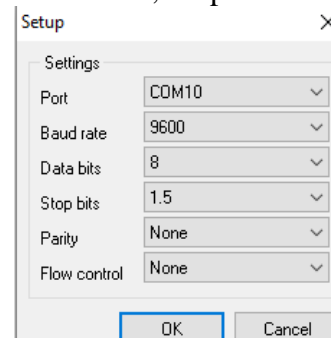


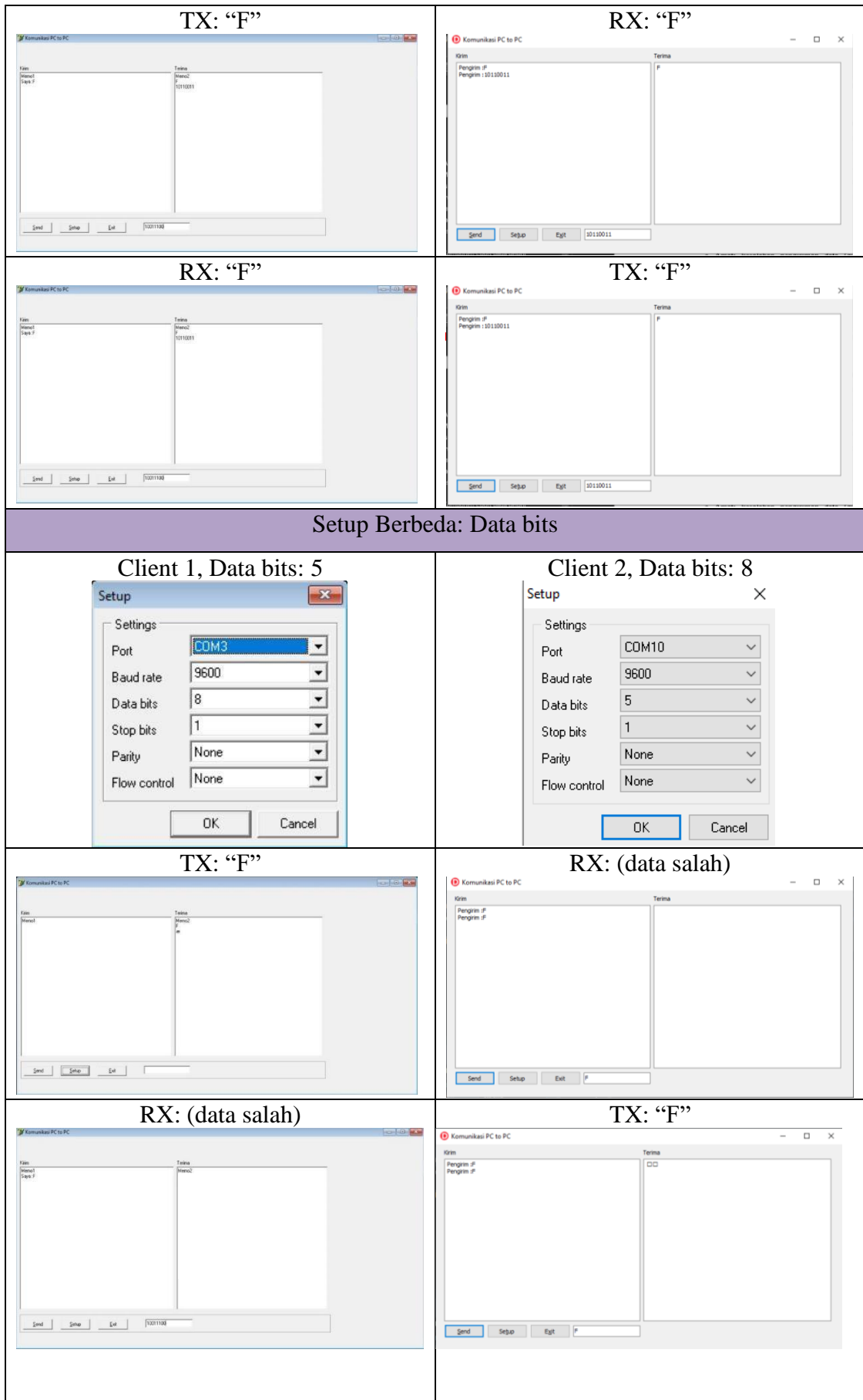
Setup Berbeda: Stop Bits

Client 1, Stop bits: 1,5



Client 2, Stop bits: 1





3.9.1 Analisa Masalah

1. Pada *setup baudrate* yang disetel tidak sama antar *client*, data yang ditransmisikan tidak sesuai dengan data yang dikirim.
2. Batas untuk pengiriman data dalam baudrate ialah senilai 115200 *baud*. Lebih dari nilai tersebut maka proses transmisi tidak dapat dilakukan bahkan terjadi data hilang.
3. Pada penyetelan *parity* yang berbeda antar *client*, terjadi *error* dalam menjalankan programnya. Hal ini dikarenakan adanya kesalahan pada *user client* 1 di mana pada *setup port* tidak dicantumkan *port* berapa yang akan digunakan untuk *port USB to RS232 converter*-nya.
4. Jika konfigurasi *databits* antar client berbeda, maka data yang dikirim rusak atau tidak sesuai dengan yang aslinya.
5. Penyetelan *stop bits* yang berbeda tidak dapat mempengaruhi kinerja dalam transmisi data serial.

3.10 Kesimpulan

Berdasarkan pada hasil percobaan dan analisa yang dilakukan, dapat disimpulkan bahwa dalam proses transmisi data secara serial dengan menggunakan alat *interface RS 232* terdapat beberapa konfigurasi yang sangat berpengaruh pada data yang ditransmisikan tidak rusak atau *lost data* (Gagal). Hal pertama yang harus diperhatikan ialah *port* yang digunakan sudah terpakai apa belum, jika hal ini dilupakan, demikian proses kedepannya akan terjadi *error*. Hal selanjutnya juga dalam mengatur *setup* pada komponen *Comport* harus diperhatikan terutama pada setelan *baudrate* dan *data bits*. Oleh karena itu, konfigurasi dalam pengaturan *baudrate*, *parity*, dan *data bits* sangat penting untuk menjaga integritas transmisi data serial.