LAMPIRAN I

Tabel Data Hasil Pengujian Serial Dengan Menggunakan Kabel 1 Meter

No.	Nama File	Kapasitas (bit)	Waktu Pengiriman (s)	Kecepatan(bps)
1	File A	800	0,100	8000,000
2	File B	1600	0,200	8000,000
3	File C	2400	0,300	8000,000
4	File D	3200	0,400	8000,000
5	File E	4000	0,500	8000,000
6	File F	4800	0,600	8000,000
7	File G	5600	0,700	8000,000
8	File H	6400	0,800	8000,000
9	File I	7200	0,900	8000,000
10	File J	8000	1,000	8000,000
11	File K	8800	1,100	8000,000
12	File L	9600	1,200	8000,000
13	File M	10400	1,300	8000,000
14	File N	11200	1,400	8000,000
15	File O	12000	1,500	8000,000

$\overline{v}$ = 8000,000 bps

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

Tabel Data Hasil Pengujian Paralel Dengan Menggunakan Kabel 1 Meter

No.	Nama File	Kapasitas (bit)	Waktu Pengiriman (s)	Kecepatan(bps)
1	File A	800	0,013	61538,462
2	File B	1600	0,025	64000,000
3	File C	2400	0,038	63157,895
4	File D	3200	0,050	64000,000
5	File E	4000	0,063	63492,063
6	File F	4800	0,075	64000,000
7	File G	5600	0,088	63636,363
8	File H	6400	0,100	64000,000
9	File I	7200	0,113	63716,814
10	File J	8000	0,125	64000,000
11	File K	8800	0,138	63768,116
12	File L	9600	0,150	64000,000
13	File M	10400	0,163	63803,681
14	File N	11200	0,175	64000,000
15	File O	12000	0,188	63829,787

$\overline{v}$ = 63662,879 bps

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

LAMPIRAN II

LISTING PROGRAM KOMUNIKASI DATA

```
program iO;

uses
  Forms,
  uIO in 'uIO.pas' {fUtama},
  uCOM in 'uCOM.pas' {fCom},
  uLPT in 'uLPT.pas' {fLPT},
  pustaka in 'pustaka.pas';

{$R *.RES}

begin
  Application.Initialize;
  Application.Title := 'Komunikasi Data';
  Application.CreateForm(TfUtama, fUtama);
  Application.CreateForm(TfLPT, fLPT);
  Application.CreateForm(TfCom, fCom);
  Application.Run;
end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
unit uIO;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, ComCtrls, StdActns, ActnList, Menus, IEButton, StdCtrls,
  TFlatEditUnit;

type
  TfUtama = class(TForm)
    bar_top: TPanel;
    bar_status: TStatusBar;
    menuUtama: TMainMenu;
    Program1: TMenuItem;
    Konfigurasi1: TMenuItem;
    mp_Keluar: TMenuItem;
    DaftarPerintah: TActionList;
    EditCopy1: TEditCopy;
    EditCut1: TEditCut;
    EditDelete1: TEditDelete;
    EditPaste1: TEditPaste;
    EditSelectAll1: TEditSelectAll;
    EditUndo1: TEditUndo;
    WindowArrange1: TWindowArrange;
    WindowCascade1: TWindowCascade;
    WindowClose1: TWindowClose;
    WindowMinimizeAll1: TWindowMinimizeAll;
    WindowTileHorizontal1: TWindowTileHorizontal;
    WindowTileVertical1: TWindowTileVertical;
    Edit1: TMenuItem;
    Windows1: TMenuItem;
    Batal1: TMenuItem;
    Cut1: TMenuItem;
    Copy1: TMenuItem;
    Paste1: TMenuItem;
    PilihSemua1: TMenuItem;
    Arrange1: TMenuItem;
    Cascade1: TMenuItem;
    Tutup1: TMenuItem;
    MinimizeSemua1: TMenuItem;
    RataHorisontal1: TMenuItem;
    RataVertikal1: TMenuItem;
    mp_LPT: TMenuItem;
    mp_COM: TMenuItem;
    tb_LPT: TIEButton;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
    tb_COM: TIEButton;
    ed_pesan: TFlatEdit;
    IEButton1: TIEButton;
    procedure mp_KeluarClick(Sender: TObject);
    procedure mp_LPTClick(Sender: TObject);
    procedure mp_COMClick(Sender: TObject);
    procedure IEButton1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fUtama: TfUtama;

implementation

uses uLPT, uCOM, pustaka;

{$R *.DFM}

//****************** TfUtama.mp_KeluarClick
************************
procedure TfUtama.mp_KeluarClick(Sender: TObject);
begin
close;
end;

//****************** TfUtama.mp_LPTClick ************************
procedure TfUtama.mp_LPTClick(Sender: TObject);
begin
fLPT.show;
end;

//****************** TfUtama.mp_COMClick ************************
procedure TfUtama.mp_COMClick(Sender: TObject);
begin
fCom.show;
end;
procedure TfUtama.IEButton1Click(Sender: TObject);
begin
if second then {ER comm initialiseret?}
  fCom.comm.SendString (ed_pesan.text);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
end;

procedure TfUtama.FormShow(Sender: TObject);
begin
tilemode  := tbVertical;
tile;
end;

end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
unit uCOM;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  MyCom, ExtCtrls, StdCtrls, IEButton, Buttons;

type
  TfCom = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    CBBaud: TComboBox;
    CBPort: TComboBox;
    CBDataBit: TComboBox;
    CBParitet: TComboBox;
    CBstopbit: TComboBox;
    CBFlow: TComboBox;
    Xon: TEdit;
    Xoff: TEdit;
    GroupBox2: TGroupBox;
    ed_pesan: TEdit;
    BeskedBox: TMemo;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    Button7: TButton;
    Button8: TButton;
    GroupBox5: TGroupBox;
    EByte: TEdit;
    Bytebox: TListBox;
    tb_sendbyte: TButton;
    tb_getByte: TButton;
    Timer1: TTimer;
    SaveDialog1: TSaveDialog;
    Comm: TComm;
    OPret: TIEButton;
    tb_SetDTR: TIEButton;
    tb_SetRTS: TIEButton;
    tb_SetXON: TIEButton;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
    tb_ClrDTR: TIEButton;
    tb_ClrRTS: TIEButton;
    tb_setXOFF: TIEButton;
    BKSend: TIEButton;
    BKHent: TIEButton;
    cb_event: TComboBox;
    tb_info: TIEButton;
    tb_simpan: TIEButton;
    SpeedButton1: TSpeedButton;
    IEButton1: TIEButton;
    lbl_jam: TLabel;
    lbl_mulai: TLabel;
    lbl_selesai: TLabel;
    lbl_lama: TLabel;
    procedure FormCanResize(Sender: TObject; var NewWidth,
      NewHeight: Integer; var Resize: Boolean);
    procedure FormCreate(Sender: TObject);
    procedure OPretClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure BKSendClick(Sender: TObject);
    procedure BKHentClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button8Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure tb_getByteClick(Sender: TObject);
    procedure tb_sendbyteClick(Sender: TObject);
    procedure tb_SetDTRClick(Sender: TObject);
    procedure tb_SetRTSClick(Sender: TObject);
    procedure tb_SetXONClick(Sender: TObject);
    procedure tb_ClrDTRClick(Sender: TObject);
    procedure tb_ClrRTSClick(Sender: TObject);
    procedure tb_setXOFFClick(Sender: TObject);
    procedure tb_infoClick(Sender: TObject);
    procedure tb_simpanClick(Sender: TObject);
    procedure IEButton1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fCom: TfCom;

implementation
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
uses pustaka;

{$R *.DFM}

//******************* TfCom.FormCanResize *************************
procedure TfCom.FormCanResize(Sender: TObject; var NewWidth,
  NewHeight: Integer; var Resize: Boolean);
begin
resize    := tidak;
end;

//******************* TfCom.FormCreate *************************
procedure TfCom.FormCreate(Sender: TObject);
begin
CBPort.Itemindex:=0;
CBFlow.ItemIndex:=0;
CBBaud.ItemIndex:=3;
CBStopbit.Itemindex:=0;
CBDatabit.Itemindex:=4;
CBparitet.Itemindex:=0;
end;

//******************* TfCom.OPretClick *************************
procedure TfCom.OPretClick(Sender: TObject);
var
  Fpar : integer;
  FStop : Integer;
  i : integer;

begin
if second then
   begin
   second := false;
   comm.done;
   exit;
   end;
  Second :=true;
  with comm do
  begin
    Parity:=tparitet(CBParitet.itemindex);
    Stopbits:=tStopbits(CBstopbit.itemindex);
    DataBits:=tDatabit(CBDatabit.itemindex+4);
    FlowControl:=tFlowcontrol (CBFlow.itemindex);
    Baud:=TBaud(cbBaud.itemindex+2);
    Port:=TPort(CbPort.ItemIndex);
    XON:=Char(StrToInt(self.XON.Text));
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
    XOFF:=Char(StrToInt(Self.XOFF.Text));
    try
      connect;
    finally
    end;
  end;

end;

//****************** TfCom.Timer1Timer ************************
procedure TfCom.Timer1Timer(Sender: TObject);
begin
lbl_jam.caption     := timetostr( time);
lbl_mulai.caption   := timetostr( comm.mulai);
lbl_selesai.caption := timetostr( comm.selesai);
lbl_lama.caption    := timetostr( comm.lama);
{If second then
  with comm do
    begin
    Caption:='Ind: '+IntToStr(InQue)+
             ' Ud: '+IntToStr (OutQue)+
             ' Fejl:'+IntToStr (Error)+
             ' Flag:'+GetFlags;;
    end;}
end;

//****************** TfCom.BKSendClick ************************
procedure TfCom.BKSendClick(Sender: TObject);
begin
if second then {ER comm initialiseret?}
  with comm do
    begin
    SendString (ed_pesan.text);
    end;
end;

//****************** TfCom.BKHentClick ************************
procedure TfCom.BKHentClick(Sender: TObject);
begin
if second then {ER comm initialiseret?}
  with comm do
    begin
    if InQue>0 then
      BeskedBox.lines.add (GetString(InQue));
    end;
end;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
//******************* TfCom.FormClose ************************
procedure TfCom.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action := caNone;
end;

//******************* TfCom.Button8Click *************************
procedure TfCom.Button8Click(Sender: TObject);
var
 ev : integer;
begin
cb_event.Text:='';
if second then
  with comm do
    begin
    GetEvent (ev);
    If (ev and ev_Cts)>0 then
      cb_event.Text:=cb_event.Text+'[EV_CTS]';
    If (ev and ev_DSR)>0 then
      cb_event.Text:=cb_event.Text+'[EV_DSR]';
    If (ev and ev_ERR)>0 then
      cb_event.Text:=cb_event.Text+'[EV_ERR]';
    If (ev and ev_RING)>0 then
      cb_event.Text:=cb_event.Text+'[EV_RING]';
    If (ev and ev_RLSD)>0 then
      cb_event.Text:=cb_event.Text+'[EV_RLSD]';
 end;

end;

//******************* TfCom.Button7Click *************************
procedure TfCom.Button7Click(Sender: TObject);
var
 myc : integer;
 i : integer;
begin
if second then
  with comm do
    begin
    For i:=0 to Cbparitet.items.count-1 do
      if cb_event.items[i]=cb_event.text then MyC:=i;
    case myc of
      0: SetEvent (EV_CTS);
      1: SetEvent (EV_DSR);
      2: SetEvent (EV_ERR);
      3: SetEvent (EV_RING);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
      4: SetEvent (EV_RLSD);
    end;
    end;
end;

procedure TfCom.tb_getByteClick(Sender: TObject);
var
  a : byte;
begin
if second then
  with comm do
    begin
    a:=GetByte;
    Bytebox.items.add (intToStr (a));
    end;
end;

procedure TfCom.tb_sendbyteClick(Sender: TObject);
begin
if second then
  with comm do
    Sendbyte(StrToInt(Ebyte.text));

end;

procedure TfCom.tb_SetDTRClick(Sender: TObject);
begin
if second then
  with comm do
    SetEscCode (SETDTR);

end;

procedure TfCom.tb_SetRTSClick(Sender: TObject);
begin
if second then
  with comm do
    SetEscCode (SETRTS);

end;

procedure TfCom.tb_SetXONClick(Sender: TObject);
begin
if second then
  with comm do
    SetEscCode (SETXON);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
end;

procedure TfCom.tb_ClrDTRClick(Sender: TObject);
begin
  if second then
  with comm do
    SetEscCode (CLRDTR);

end;

procedure TfCom.tb_ClrRTSClick(Sender: TObject);
begin
  if second then
  with comm do
    SetEscCode (CLRRTS);

end;

procedure TfCom.tb_setXOFFClick(Sender: TObject);
begin
  if second then
  with comm do
    SetEscCode (SETXOFF);
end;

//******************* TfCom.tb_infoClick *************************
procedure TfCom.tb_infoClick(Sender: TObject);
begin
If second then
   Caption:='Ind: '+IntToStr(Comm.InQue)+
              ' Ud: '+IntToStr (Comm.OutQue);
end;

//******************* TfCom.tb_simpanClick *************************
procedure TfCom.tb_simpanClick(Sender: TObject);
var
  fil : textfile;
  str : string;

begin
if second then
   with comm do
     begin
     if (inque>0) and (Savedialog1.execute) then
       begin
       Str:=GetString(InQue);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
        System.assign (fil,savedialog1.filename);
        Rewrite (fil);
        write (fil,str);
        closefile (fil);
        end;
      end;

end;

procedure TfCom.IEButton1Click(Sender: TObject);
begin
second := false;
comm.done;
end;

end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
unit uLPT;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  SmallPort, Spin, StdCtrls, ExtCtrls, IEButton, TB97Ctls;

type
  TfLPT = class(TForm)
    GroupBox4: TGroupBox;
    GroupBox5: TGroupBox;
    Bit1: TCheckBox;
    Bit0: TCheckBox;
    Bit2: TCheckBox;
    Bit3: TCheckBox;
    Bit4: TCheckBox;
    Bit5: TCheckBox;
    Bit6: TCheckBox;
    Bit7: TCheckBox;
    GroupBox6: TGroupBox;
    Error: TCheckBox;
    Select: TCheckBox;
    PE: TCheckBox;
    Ask: TCheckBox;
    Busy: TCheckBox;
    RadioGroup1: TRadioGroup;
    LPT1: TRadioButton;
    LPT2: TRadioButton;
    GroupBox8: TGroupBox;
    Label2: TLabel;
    SpinEdit2: TSpinEdit;
    SmallPort1: TSmallPort;
    tb_BacaPort: TIEButton;
    tb_tulis: TIEButton;
    GroupBox2: TGroupBox;
    BKSend: TIEButton;
    BKHent: TIEButton;
    BeskedBox: TMemo;
    Edit971: TEdit97;
    GroupBox1: TGroupBox;
    lbl_jam: TLabel;
    lbl_jammulai: TLabel;
    lbl_jamselesai: TLabel;
    lbl_lama: TLabel;
    tmr: TTimer;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
    procedure FormCanResize(Sender: TObject; var NewWidth,
      NewHeight: Integer; var Resize: Boolean);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure tb_BacaPortClick(Sender: TObject);
    procedure tb_tulisClick(Sender: TObject);
    procedure tmrTimer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fLPT: TfLPT;

implementation

uses pustaka;

{$R *.DFM}

procedure TfLPT.FormCanResize(Sender: TObject; var NewWidth,
  NewHeight: Integer; var Resize: Boolean);
begin
resize    := tidak;
end;

procedure TfLPT.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action := caNone;
end;

//******************* TfLPT.tb_BacaPortClick ************************
procedure TfLPT.tb_BacaPortClick(Sender: TObject);
var b,b1: BYTE;
begin
if SmallPort1.Opened then
  begin
  if LPT1.Checked then BASE:=$378 else BASE:=278;
  b:=SmallPort1.Port[BASE];
  b1:=SmallPort1.Port[BASE+1];
  if(b1 and $08)<>0 then Error.Checked:=true  else Error.Checked:=false;
  if(b1 and $10)<>0 then Select.Checked:=true else Select.Checked:=false;
  if(b1 and $20)<>0 then PE.Checked:=true     else PE.Checked:=false;
  if(b1 and $40)<>0 then ASK.Checked:=true    else ASK.Checked:=false;
  if(b1 and $80)<>0 then BUSY.Checked:=true   else BUSY.Checked:=false;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
  if(b and $01)<>0  then Bit0.Checked:=true   else Bit0.Checked:=false;
  if(b and $02)<>0  then Bit1.Checked:=true   else Bit1.Checked:=false;
  if(b and $04)<>0  then Bit2.Checked:=true   else Bit2.Checked:=false;
  if(b and $08)<>0  then Bit3.Checked:=true   else Bit3.Checked:=false;
  if(b and $10)<>0  then Bit4.Checked:=true   else Bit4.Checked:=false;
  if(b and $20)<>0  then Bit5.Checked:=true   else Bit5.Checked:=false;
  if(b and $40)<>0  then Bit6.Checked:=true   else Bit6.Checked:=false;
  if(b and $80)<>0  then Bit7.Checked:=true   else Bit7.Checked:=false;
 end;
end;

procedure TfLPT.tb_tulisClick(Sender: TObject);
begin
if SmallPort1.Opened then
  begin
  if LPT1.Checked then BASE:=$378 else BASE:=278;
  SmallPort1.Port[$378]:=SpinEdit2.Value;
  end;

end;

procedure TfLPT.tmrTimer(Sender: TObject);
begin
lbl_jam.caption := timetostr( time);
end;

end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
unit pustaka;

interface


const
        Ya                      = true;
        Tidak                   = false;
        benar                   = true;
        Salah                   = false;
        Sukses                  = true;
        Gagal                   = false;
        second : boolean=false;

var
   Dir                          : string;
   BASE                         : WORD;

implementation


end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
unit MyCom;

interface

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

type
TBaud=(b110,b300,b1200,b2400,b4800,b9600,b14400,b38400,b56000,
     b57600,b115200,b128000,b256000);
tStopBits=(s10,s15,s20);
tParitet=(Zero,Odd,Even,Binary0,Binary1);
tDataBit=4..8;
tPort=(COM1,COM2,COM3,COM4,COM5,COM6,COM7,COM8);
tFlowControl= (Hardware,XON_XOFF);
TComError=procedure (Sender : Tobject;ECode : integer) of object;
pComm=^TComm;
TComm= class (TComponent)
  constructor Create (AOwner: TComponent); override;
  procedure Connect; virtual; // Set up connection
  function GetOut : integer;
  function GetIn : integer;
  function GetError : integer;
  procedure SetError  (ECode : integer);
  function GetFlags :string;
  procedure SetEvent (Event : integer);
  function  GetEvent (var Event : integer) : boolean;
  procedure SetEscCode (EscCode : integer);
  procedure SendByte (byt : byte); virtual;
  function GetByte : byte; virtual;
  procedure SendString (str : string); virtual;
  function GetString (Len : integer) : string; virtual;
  destructor done;
  private
   FOnError : TComError;
   FStat : pComStat;
   Fport : integer;
   fComPort : tport;
   FBaud : tbaud;
   FDataBit : tdatabit;
   FParitet : tParitet;
   FStopBit : tStopBits;
   fFlow : tFlowcontrol;
   FErrorCode : Integer;
   fconnected : boolean;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
    FXON : Char;
    FXOFF : CHAR;
    fLama    : ttime;
    fMulai   : ttime;
    fSelesai : ttime;
  public
  published
   // Events
   property OnComError : TComError read FOnError write FonError;
   // properties
   property Baud: tBaud read fBaud write fbaud default b9600;
   property StopBits: tStopbits read fStopbit write fstopbit default s10;
   property Parity: tParitet read fparitet write fparitet default zero;
   property DataBits: tDataBit read fDatabit write fDatabit default 8;
  property FlowControl: tFlowControl read fFlow write fFlow default hardware;
   property Port: tport read fComport write fComport default com1;
   property XON : char read FXON WRITE FXON default #17;
   property XOFF : char read FXOFF WRITE FXOFF default #19;
   // Runtime only
   property InQue : integer read GetIn;
   property OutQue : integer read GetOut;
   property mulai : ttime read fMulai write fMulai;
   property selesai : ttime read fSelesai write fSelesai;
   property lama   : ttime read fLama write fLama;
end;

procedure Register;

implementation

constructor tComm.Create (AOwner: TComponent);

begin
  Inherited create (AOwner);
  fbaud:=b9600;
  fstopbit:=s10;
  fparitet:=zero;
  fDatabit:=8;
  fFlow:=hardware;
  fComport:=com1;
  fXON:=#17;
  fXOFF:=#19;
  New (Fstat);
  fConnected:=false;
end;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
procedure tComm.Connect;

var
  aDCB : TDCB;
  temp : byte;

begin
  fstat^.cbinque:=0;
  if fconnected then
    closehandle (Fport);
  Temp:=byte (fComport)+1;
  FPort:=CreateFile (pchar('COM'+IntToStr(temp)),
                GENERIC_READ or GENERIC_WRITE,0,NIl,
                OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL,0);
  if FPort = -1 then
    begin
//    SetError (GetLastError);
    exit;
    end;
  If not SetupComm (fPort,1600,1600) then
      begin
      SetError (GetLastError);
      exit;
      end;
  If Not GetCommState (fport,aDCB) then
    SetError (GetLastError);;
  case fbaud of
   b110:   ADCB.baudrate:=CBR_110;
   b300:   ADCB.baudrate:=CBR_300;
   b1200:  ADCB.baudrate:=CBR_1200;
   b2400:  ADCB.baudrate:=CBR_2400;
   b4800:  ADCB.baudrate:=CBR_4800;
   b9600:  ADCB.BaudRate:=CBR_9600;
   b14400: ADCB.baudrate:=CBR_14400;
   b38400: ADCB.baudrate:=CBR_38400;
   b56000: ADCB.baudrate:=CBR_56000;
   b57600: ADCB.baudrate:=CBR_57600;
   b115200: ADCB.baudrate:=CBR_115200;
   b128000: ADCB.baudrate:=CBR_128000;
   b256000: ADCB.baudrate:=CBR_256000;
  end;
  ADCB.bytesize:=FDatabit;
  ADCB.Parity:=Integer(FParitet);
  ADCB.StopBits:=Integer(FStopbit);
  FErrorCode:=0;
  If FFlow=XON_XOFF then
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
  begin
    aDcb.flags:=ADcb.flags OR (3 shl 9);
    ADCB.XONChar:=FXON;
    ADCB.XOFFChar:=FXOFF;
  end else
  begin
    aDcb.flags:=ADcb.flags OR (3 shl 9);
    aDcb.flags:=ADcb.flags XOR (3 shl 9);
  end;;
  if Not SetCommState (fport,aDcb) then
    SetError (GetLastError);
  fConnected:=true;
end;

procedure TComm.SetError  (ECode : integer);

begin
  if ECode<>0 then
  begin
    FErrorCode:=ECode;
      OnComError (Self,ECode);
  end;
end;

function TComm.GetFlags :string;

var
  aDCb : tdcb;
  i,test : integer;

begin
  test :=1;
  if not GetCommState (fPort,aDcb) then
    SetError (GetLastError);
  Result:='';
  for i:=0 to 15 do
  begin
    if (test and aDcb.flags)>0 then
      Result:='1'+Result else
      Result:='0'+Result;
    Test:=test shl 1;
  end;
end;

function Tcomm.GetError : integer;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
begin
  result:=fErrorCode;
  fErrorCode:=0;
end;

procedure tComm.SendString (Str : string);

var
  a: array [0..1600] of byte;
  Nof : dword;
  i : byte;

begin
  fmulai :=time;
  for i:=1 to length (str) do
    a[i-1]:=Byte(Str[i]);
   fselesai := time;
  flama    := fselesai - fmulai;
  if not WriteFile (fPort,a,Length(str),NOF,NIL) then
    SetError (GetLastError)
  else
    flama := fselesai - fmulai;
end;

function tComm.GetString (Len:Integer) :string;

var
  a: array [0..1600] of byte;
  Nof : dword;
  i : byte;
  TStr : string;

begin
  TStr:='';
  if not ReadFile(fPort,a,Len,NOF,NIL) then
    SetError (GetLastError);
  for i:=1 to Nof do
    Tstr:=TStr+Char(a[i-1]);
  Result:=Tstr;
end;

procedure Tcomm.SendByte;

var
  Nof : dword;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
begin
  if not writeFile (Fport,byt,1,NOF,NIL) then
    SetError (GetLastError);
end;

function Tcomm.GetByte : byte;

var
  Nof : dword;
  Byt : byte;

begin
  if not ReadFile (Fport,byt,1,NOF,NIL) then
    SetError (GetLastError);
  result:=byt;
end;

function Tcomm.GetEvent (var Event : integer) : boolean;

var
  ev : dword;

begin
  if not GetCommMask (fport,Ev) then
    SetError (GetLastError);
  Result:=ev=event;
  Event:=ev;

end;

procedure TComm.SetEvent (Event : integer);

begin
  if not SetCommMask (FPort,Event) then
    SetError (GetLastError);
end;

procedure TComm.SetEscCode;

begin
  if not EscapeCommFunction (FPort,EscCode) then
    SetError (GetLastError);

end;

function tComm.GetOut : integer;
```

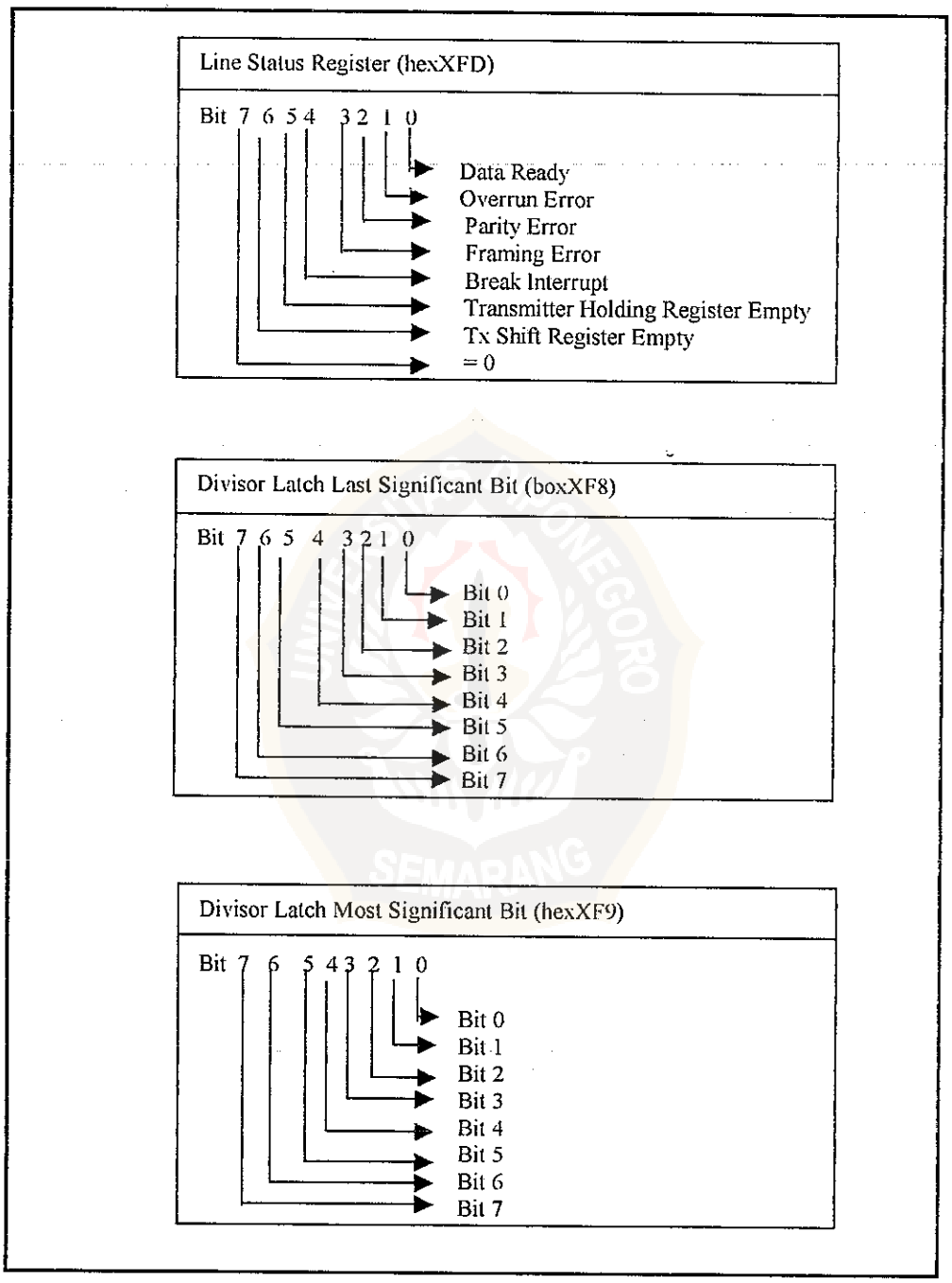
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

```
var
  mErrorCode : dword;

begin
  ClearCommError (FPort,mErrorCode,fStat);
  If FerrorCode = 0 then
    SetError(mErrorCode);
  Result:=fstat^.cboutQue;
end;

function tComm.GetIn : integer;

var
  mErrorCode : dword;

begin
  ClearCommError (FPort,mErrorCode,fStat);
  If FerrorCode = 0 then
    SetError(mErrorCode);
  Result:=fstat^.cbinQue;
end;

destructor tComm.done;

begin
  dispose (Fstat);
  closeHandle (fPort);
  FConnected:=False;
end;

procedure Register;

begin
  RegisterComponents ('io',[tComm]);
end;

end.
```

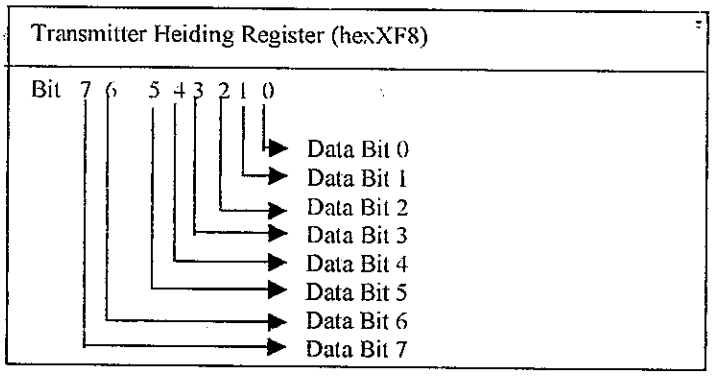
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

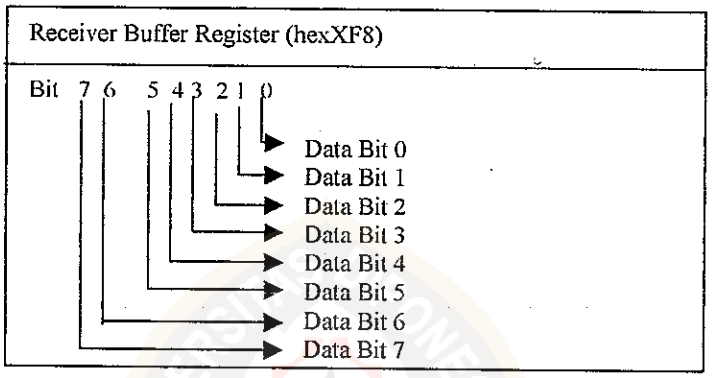
LAMPIRAN III

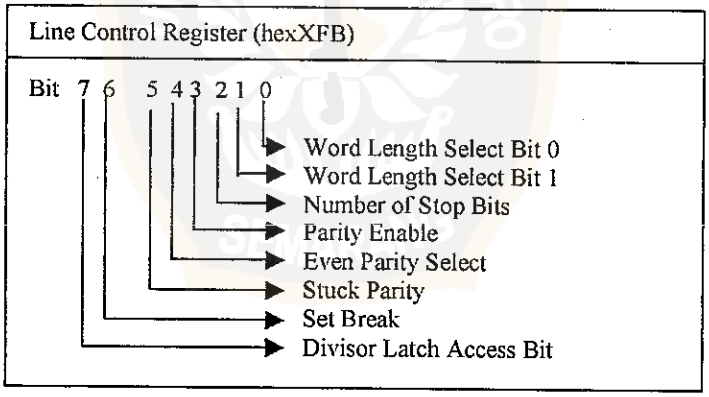
PENGESETAN PADA REGISTER



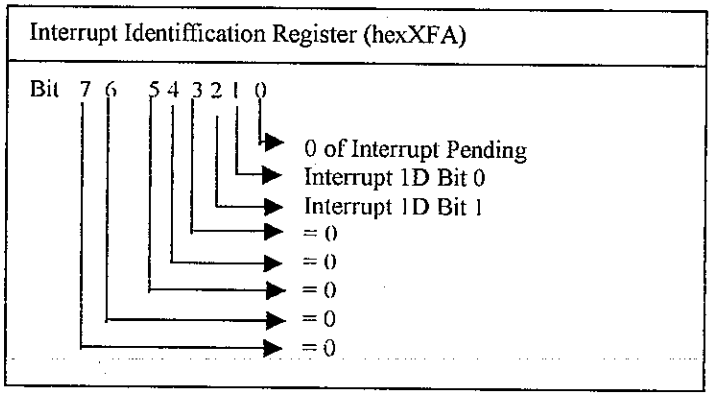
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

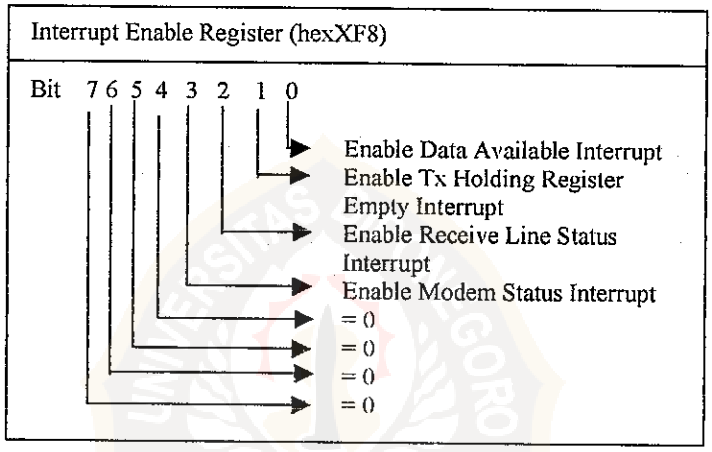






This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)





This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

LAMPIRAN IV

Tabel Karakter ASCII

Kar	Biner	Hex	Desimal	Kar	Biner	Hex	Desimal
Null	00000000	00	000	@	01000000	40	064
SOH	00000001	01	001	A	01000001	41	065
STX	00000010	02	002	B	01000010	42	066
ETX	00000011	03	003	C	01000011	43	067
EOT	00000100	04	004	D	01000100	44	068
ENQ	00000101	05	005	E	01000101	45	069
ACK	00000110	06	006	F	01000110	46	070
BEL	00000111	07	007	G	01000111	47	071
BS	00001000	08	008	H	01001000	48	072
HT	00001001	09	009	I	01001001	49	073
LF	00001010	0A	010	J	01001010	4A	074
VT	00001011	0B	011	K	01001011	4B	075
FF	00001100	0C	012	L	01001100	4C	076
CR	00001101	0D	013	M	01001101	4D	077
SO	00001110	0E	014	N	01001110	4E	078
SI	00001111	0F	015	O	01001111	4F	079
DLE	00010000	10	016	P	01010000	50	080
DC1	00010001	11	017	Q	01010001	51	081
DC2	00010010	12	018	R	01010010	52	082
DC3	00010011	13	019	S	01010011	53	083
DC4	00010100	14	020	T	01010100	54	084
NAK	00010101	15	021	U	01010101	55	085
SYN	00010110	16	022	V	01010110	56	086
ETB	00010111	17	023	W	01010111	57	087
CAN	00011000	18	024	X	01011000	58	088
EM	00011001	19	025	Y	01011001	59	089
SUB	00011010	1A	026	Z	01011010	5A	090
ESCAPE	00011011	1B	027	[	01011011	5B	091
FS	00011100	1C	028	bkslh	01011100	5C	092
GS	00011101	1D	029	]	01011101	5D	093
RS	00011110	1E	030	|	01011110	5E	094
US	00011111	1F	031	←	01011111	5F	095
SPACE	00100000	20	032	-	01100000	60	096

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)

Kar	Biner	Hex	Desimal	Kar	Biner	Hex	Desimal
!	00100001	21	033	a	01100001	61	097
"	00100010	22	034	b	01100010	62	098
#	00100011	23	035	c	01100011	63	099
$	00100100	24	036	d	01100100	64	100
%	00100101	25	037	e	01100101	65	101
&	00100110	26	038	f	01100110	66	102
'	00100111	27	039	g	01100111	67	103
(	00101000	28	040	h	01101000	68	104
)	00101001	29	041	i	01101001	69	105
*	00101010	2A	042	j	01101010	6A	106
+	00101011	2B	043	k	01101011	6B	107
,	00101100	2C	044	l	01101100	6C	108
_	00101101	2D	045	m	01101101	6D	109
.	00101110	2E	046	n	01101110	6E	110
/	00101111	2F	047	o	01101111	6F	111
0	00110000	30	048	p	01110000	70	112
1	00110001	31	049	q	01110001	71	113
2	00110010	32	050	r	01110010	72	114
3	00110011	33	051	s	01110011	73	115
4	00110100	34	052	t	01110100	74	116
5	00110101	35	053	u	01110101	75	117
6	00110110	36	054	v	01110110	76	118
7	00110111	37	055	w	01110111	77	119
8	00111000	38	056	x	01111000	78	120
9	00111001	39	057	y	01111001	79	121
:	00111010	3A	058	z	01111010	7A	122
;	00111011	3B	059	{	01111011	7B	123
<	00111100	3C	060	--	01111100	7C	124
=	00111101	3D	061	}	01111101	7D	125
>	00111110	3E	062	~	01111110	7E	126
?	00111111	3F	063	DEL	01111111	7F	127

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purposes of security, back-up and preservation. (http://eprints.undip.ac.id)