

## LAMPIRAN

- **Lampiran *main program* pada *delphi***

```
unit Unit1;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms,

    Dialogs, StdCtrls, CPort, Grids, DBGrids, ComCtrls, DB, CheckLst, IPL,
    OpenCV,

    ExtCtrls;

type

    TForm1 = class(TForm)

        Button1: TButton;

        OpenDialog1: TOpenDialog;

        ComPort1: TComPort;

        Button2: TButton;

        Button3: TButton;

        Button4: TButton;
```

Button5: TButton;

Button6: TButton;

Button7: TButton;

CheckListBox1: TCheckListBox;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Button8: TButton;

ComboBox1: TComboBox;

GroupBox1: TGroupBox;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Button9: TButton;

Panel6: TPanel;

Image6: TImage;

Panel5: TPanel;

Image5: TImage;

Panel4: TPanel;

Image4: TImage;

Panel3: TPanel;

Image3: TImage;

Panel2: TPanel;

Image2: TImage;

Panel1: TPanel;

Image1: TImage;

Timer1: TTimer;

tb1: TTrackBar;

Edit7: TEdit;

Edit8: TEdit;

tb2: TTrackBar;

tb3: TTrackBar;

Edit9: TEdit;

tb4: TTrackBar;

Edit10: TEdit;

Panel7: TPanel;

Image7: TImage;

Panel8: TPanel;

Image8: TImage;

Panel9: TPanel;

Image9: TImage;

tb5: TTrackBar;

Edit11: TEdit;

tb6: TTrackBar;

Edit12: TEdit;

Edit13: TEdit;

Edit14: TEdit;

Edit15: TEdit;

Edit16: TEdit;

Timer2: TTimer;

Edit17: TEdit;

Edit18: TEdit;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Button9Click(Sender: TObject);

procedure FormKeyPress(Sender: TObject; var Key: Char);

procedure FormDestroy(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure tb1Change(Sender: TObject);

```
procedure tb2Change(Sender: TObject);

procedure tb3Change(Sender: TObject);

procedure tb4Change(Sender: TObject);

procedure tb5Change(Sender: TObject);

procedure tb6Change(Sender: TObject);

procedure Timer2Timer(Sender: TObject);

private

  { Private declarations }

  procedure oneStep;

  procedure oneStep1;

  Procedure mySleep(Millisecs : longint);

public

  { Public declarations }

end;

var

  Form1: TForm1;

  baris : integer;

  autoRun : Boolean;
```

```
perintah : string;

counterMaju, sudutKepala : Integer;

jumlahTidakKetemu : Integer;

baruStart : boolean;

capture: PCvCapture;

frame: IplImage;

color: CvScalar;

bmp: TBitmap;

bmp1: TBitmap;

bmp2: TBitmap;

bmp3: TBitmap;

bmp4: TBitmap;

bmp5: TBitmap;

bmp6: TBitmap;

bmp7: TBitmap;

bmp8: TBitmap;

image: pIplImage = 0;

image1: pIplImage = 0;
```

image2: pIplImage = 0;

image3: pIplImage = 0;

image4: pIplImage = 0;

image5: pIplImage = 0;

image6: pIplImage = 0;

image7: pIplImage = 0;

image8: pIplImage = 0;

YCrCb: pIplImage = 0;

hsv: pIplImage = 0;

hue: pIplImage = 0;

sat: pIplImage = 0;

val: pIplImage = 0;

mask0: pIplImage = 0;

mask1: pIplImage = 0;

mask2: pIplImage = 0;

mask3: pIplImage = 0;

mask4: pIplImage = 0;

hasil1: pIplImage = 0;





```
if (sector and 1) <> 0 then
    p := p xor 255
else
    p := p xor 0;

rgb[sector_data[sector][0]] := 255;
rgb[sector_data[sector][1]] := 0;
rgb[sector_data[sector][2]] := p;

result := cvScalar_(rgb[2], rgb[1], rgb[0], 0);

end;

procedure main_cycle();

var

    //i, bin_w: integer;

    //_vmin, _vmax: integer;

    //max_val: float;

    //val: integer ;

    cs: CvSize;

    rec: TRect;

    baris,kolom,idx:integer;
```

```
offset,offsetmin1,offsetplus1 : longint;

dataByte,dataBytemin1,dataByteplus1 : PByteArray;

storage : PCvMemStorage;

contour : PCvSeq;

r : CVRect;

pt1, pt2,pt1x, pt2x : CvPoint; //pt3, pt4

ukuran : integer;

begin

begin

frame := cvQueryFrame( capture );

if not(assigned(frame) ) then

exit;

if not(assigned(image) ) then

begin

/* allocate all the buffers */

cs.width := frame.Width;

cs.height := frame.Height;

image := cvCreateImage( cs, 8, 3 );
```

```
image.Origin := frame.Origin;

image1 := cvCreateImage( cs, 8, 3 );
image1.Origin := frame.Origin;

image2 := cvCreateImage( cs, 8, 3 );
image2.Origin := frame.Origin;

image3 := cvCreateImage( cs, 8, 3 );
image3.Origin := frame.Origin;

image4 := cvCreateImage( cs, 8, 3 );
image4.Origin := frame.Origin;

image5 := cvCreateImage( cs, 8, 3 );
image5.Origin := frame.Origin;

image6 := cvCreateImage( cs, 8, 3 );
image6.Origin := frame.Origin;

image7 := cvCreateImage( cs, 8, 3 );
image7.Origin := frame.Origin;

image8 := cvCreateImage( cs, 8, 3 );
image8.Origin := frame.Origin;

hsv := cvCreateImage( cs, 8, 3 );
```

```
YCrCb := cvCreateImage( cs, 8, 3 );  
  
hue := cvCreateImage( cs, 8, 1 );  
  
sat := cvCreateImage( cs, 8, 1 );  
  
val := cvCreateImage( cs, 8, 1 );  
  
mask0 := cvCreateImage( cs, 8, 1 );  
  
mask1 := cvCreateImage( cs, 8, 1 );  
  
mask2 := cvCreateImage( cs, 8, 1 );  
  
mask3 := cvCreateImage( cs, 8, 1 );  
  
mask4 := cvCreateImage( cs, 8, 1 );  
  
hasil1 := cvCreateImage( cs, 8, 1 );  
  
hasil2 := cvCreateImage( cs, 8, 1 );  
  
hasil3 := cvCreateImage( cs, 8, 1 );  
  
end;  
  
begin  
  
assignfile( F , 'C:\Documents and Settings\anton\Desktop\test.txt');  
  
reset(F);  
  
Readln(F,S);  
  
CloseFile(F);
```

```

end;

form1.Edit18.Text := s;

cvCopy( frame, image, 0 );

cvCvtColor( image, YCrCb, CV_BGR2YCrCb );

cvCvtColor( image, hsv, CV_BGR2HSV );

cvCopy( hsv, image1, 0 );

cvSplit( image, hue, sat, val, 0 );

cvMerge(hue,hue,hue,0,image6);

cvMerge(sat,sat,sat,0,image7);

cvMerge(val,val,val,0,image8);

if ( S = 'red') then //merah

begin

cvInRangeS( hue, cvScalar_(0,0,0,0) , cvScalar_(70,0,0,0), mask0 );

cvInRangeS( sat, cvScalar_(0,0,0,0) , cvScalar_(70,0,0,0), mask1 );

cvInRangeS( val, cvScalar_(150,0,0,0) , cvScalar_(250,0,0,0), mask2
);

end

else if ( S = 'orange' ) then //orange

```

```
begin

    cvInRangeS( hue, cvScalar_(0,0,0,0) , cvScalar_(100,0,0,0), mask0 );

    cvInRangeS( sat, cvScalar_(50,0,0,0) , cvScalar_(240,0,0,0), mask1 );

    cvInRangeS( val, cvScalar_(150,0,0,0) , cvScalar_(240,0,0,0), mask2 );

end

else if ( S = 'blue' ) then //biru

    begin

        cvInRangeS( hue, cvScalar_(130,0,0,0) , cvScalar_(250,0,0,0), mask0
    );

        cvInRangeS( sat, cvScalar_(0,0,0,0) , cvScalar_(110,0,0,0), mask1 );

        cvInRangeS( val, cvScalar_(0,0,0,0) , cvScalar_(250,0,0,0), mask2 );

    end

else if ( S = 'black' ) then //hitam

    begin

        cvInRangeS( hue, cvScalar_(10,0,0,0) , cvScalar_(20,0,0,0), mask0 );

        cvInRangeS( sat, cvScalar_(0,0,0,0) , cvScalar_(20,0,0,0), mask1 );

        cvInRangeS( val, cvScalar_(0,0,0,0) , cvScalar_(140,0,0,0), mask2 );

    end

end
```

```

else if( S = 'white' ) then //putih

begin

    cvInRangeS( hue, cvScalar_(140,0,0,0) , cvScalar_(240,0,0,0), mask0
);

    cvInRangeS( sat, cvScalar_(0,0,0,0) , cvScalar_(240,0,0,0), mask1 );

    cvInRangeS( val, cvScalar_(100,0,0,0) , cvScalar_(240,0,0,0), mask2
);

end;

end;

cvAND(mask0,mask1,mask3,0);

cvAND(mask3,mask2,hasil1,0);

cvMerge(hasil1,hasil1,hasil1,0,image3);

cvCopy(hasil1,mask4,0);

cvSmooth(mask4,mask3,CV_BLUR,15,0,0);

cvMerge(mask3,mask3,mask3,0,image4);

cvThreshold(mask3,mask4,20,255,CV_THRESH_BINARY);

cvMerge(mask4,mask4,mask4,0,image5);

cvCopy(mask4,mask3,0);

```



```
storage := cvCreateMemStorage(0);

contour := 0;

cvFindContours( mask3, storage, @contour, sizeof(CvContour),
CV_RETR_EXTERNAL, CV_CHAIN_APPROX_NONE, cvPoint_(0,0));

idx := 0;

ukuran := 0;

while ( contour <> nil) do

begin

r := cvBoundingRect( contour, 0 );

if (r.y<140) then

begin

if (ukuran<(r.width*r.height)) then

begin

ukuran := r.width*r.height;

pt1.x := r.x + (r.width div 2);

pt1.y := r.y + (r.height div 2);

Form1.edit13.Text := inttostr(pt1.x) + ' ' + inttostr(pt1.y);
```

```
Form1.edit14.Text := inttostr(ukuran);

end;

idx := idx + 1;

end;

contour := contour.h_next;

end;

form1.edit15.text := inttostr(idx);

cvMerge(mask4,mask4,mask4,0,image2);

cvReleaseMemStorage(storage);

if idx>0 then

begin

if (csDSR in Form1.ComPort1.Signals) then

begin

if sudutKepala<=90 then

begin

sudutKepala := 95;

if pt1.x<120 then

begin
```

```
Form1.OpenDialog1.FileName := 'C:\Documents and  
Settings\anton\Desktop\Servo_prog\BALL2AN\gasing kiri.txt';
```

```
Form1.Button9.Click;
```

```
Form1.Button4.Click;
```

```
Form1.Edit16.Text := 'Gasing Kiri';
```

```
end
```

```
else if pt1.x>200 then
```

```
begin
```

```
Form1.OpenDialog1.FileName := 'C:\Documents and  
Settings\anton\Desktop\Servo_prog\BALL2AN\gasing kanan.txt';
```

```
Form1.Button9.Click;
```

```
Form1.Button4.Click;
```

```
Form1.Edit16.Text := 'Gasing Kanan';
```

```
end
```

```
else
```

```
begin
```

```
Form1.OpenDialog1.FileName := 'C:\Documents and  
Settings\anton\Desktop\Servo_prog\BALL2AN\JALAN.txt';
```

```
Form1.Button9.Click;

Form1.Button4.Click;

Form1.Edit16.Text := 'Jalan';

end;

Form1.Timer1.Interval := 100;

end

else

begin

Form1.Edit16.Text := 'dekat';

Form1.Timer1.Interval := 100;

end;

end;

end

else

begin

if sudutKepala<=50 then

begin
```

```
sudutKepala := 95;

jumlahTidakKetemu := jumlahTidakKetemu + 1;

if jumlahTidakKetemu >= 1 then

begin

    if (csDSR in Form1.ComPort1.Signals) then

        begin

            Form1.OpenDialog1.FileName := 'C:\Documents and
Settings\anton\Desktop\Servo_prog\BALL2AN\MUNDUR.txt';

            Form1.Button9.Click;

            Form1.Button4.Click;

            Form1.Button4.Click;

        end;

        Form1.Edit16.Text := 'Mundur';

        jumlahTidakKetemu := 0;

    end;

end;

perintah := 'SER 00 00 '+ inttostr(sudutKepala);
```

```
Form1.oneStep1;

sudutKepala := sudutKepala - 2;

Form1.Timer1.Interval := 100;

end;

{visualize the camera image in the window}

IplImage2Bitmap(image, bmp);

rec := form1.Image1.canvas.ClipRect;

form1.Image1.canvas.StretchDraw(rec , bmp);

IplImage2Bitmap(image1, bmp1);

rec := form1.Image2.canvas.ClipRect;

form1.Image2.canvas.StretchDraw(rec , bmp1);

IplImage2Bitmap(image2, bmp2);

rec := form1.Image3.canvas.ClipRect;

form1.Image3.canvas.StretchDraw(rec , bmp2);

IplImage2Bitmap(image3, bmp3);
```

```
rec := form1.Image4.canvas.ClipRect;
```

```
form1.Image4.canvas.StretchDraw(rec , bmp3);
```

```
IplImage2Bitmap(image4, bmp4);
```

```
rec := form1.Image5.canvas.ClipRect;
```

```
form1.Image5.canvas.StretchDraw(rec , bmp4);
```

```
IplImage2Bitmap(image5, bmp5);
```

```
rec := form1.Image6.canvas.ClipRect;
```

```
form1.Image6.canvas.StretchDraw(rec , bmp5);
```

```
IplImage2Bitmap(image6, bmp6);
```

```
rec := form1.Image7.canvas.ClipRect;
```

```
form1.Image7.canvas.StretchDraw(rec , bmp6);
```

```
IplImage2Bitmap(image7, bmp7);
```

```
rec := form1.Image8.canvas.ClipRect;
```

```
form1.Image8.canvas.StretchDraw(rec , bmp7);
```

```
    IpImage2Bitmap(image8, bmp8);

    rec := form1.Image9.canvas.ClipRect;

    form1.Image9.canvas.StretchDraw(rec , bmp8);

    form1.Repaint;

    {

    if (show_hist <> 0) then

    begin

        IpImage2Bitmap(histimg, bmp);

        fHistogram.histimage.canvas.StretchDraw(fHistogram.histimage.canvas.ClipRect , bmp);

        end;

    }

    //end

end;

Procedure TForm1.mySleep(Millisecs : longint);
```



```
var Start : Longint;

begin

    Start := GetTickCount;

    repeat

        {$IFDEF WIN32}

        Application.ProcessMessages

        {$ENDIF}

    until GetTickCount - Start >= Millisecs ;

end ;

procedure TForm1.oneStep;

var idx,rotation,sudut:integer;

    buff,par1,par2,par3:string;

begin

    //send index sekarang

    buff := CheckListBox1.Items[CheckListBox1.ItemIndex];

    edit1.Text := buff;

    par1 := "";

    par2 := "";
```

```
par3 := "";

if sameText(copy(buff,1,4),'SER ') then

begin

    edit2.Text := 'ServoCommand';

    idx:=5;

    while buff[idx]<>' ' do

    begin

        par1:=par1 + buff[idx];

        inc(idx);

    end;

    idx:=idx+1;

    while buff[idx]<>' ' do

    begin

        par2:=par2 + buff[idx];

        inc(idx);

    end;

    idx:=idx+1;

    while (buff[idx]<>' ') and (idx<=length(buff)) do
```

```
begin

  par3:=par3 + buff[idx];

  inc(idx);

end;

edit3.Text := par1;

edit4.Text := par2;

edit5.Text := par3;

sudut:= strtoint(par3);

if sudut<0 then sudut:=0;

if sudut>180 then sudut:=180;

rotation := (sudut*1000) div 180 + 250;

edit6.Text := inttostr(rotation);

if rotation<300 then rotation:=300;

if rotation>1200 then rotation:=1200;

try

  Comport1.WriteStr(!);

  Comport1.WriteStr('S');

  Comport1.WriteStr('C');
```

```
Comport1.WriteStr(chr(strtoint(par1) mod 256));  
  
Comport1.WriteStr(chr(strtoint(par2) mod 256));  
  
Comport1.WriteStr(chr(rotation mod 256));  
  
Comport1.WriteStr(chr(rotation div 256));  
  
Comport1.WriteStr(#13);  
  
except  
  
  ShowMessage('COM Port error !');  
  
end;  
  
end  
  
else if sameText(copy(buff,1,4),'DEL ') then  
  
begin  
  
  edit2.Text := 'DelayCommand';  
  
  idx:=5;  
  
  while (buff[idx]<>' ') and (idx<=length(buff)) do  
  
  begin  
  
    par1:=par1 + buff[idx];  
  
    inc(idx);  
  
  end;
```

```
edit3.Text := par1;

edit4.Text := par2;

edit5.Text := par3;

mySleep(strtoint(par1));

end

else

begin

end;

baris := CheckListBox1.ItemIndex;

CheckListBox1.Checked[CheckListBox1.ItemIndex] := True;

CheckListBox1.Selected[CheckListBox1.ItemIndex] := False;

if baris < CheckListBox1.Count - 1 then

begin

    CheckListBox1.ItemIndex := baris + 1;

    CheckListBox1.Selected[CheckListBox1.ItemIndex] := True;

end

else
```

```
begin

  for idx:=0 to CheckListBox1.Count-1 do

    CheckListBox1.Checked[idx] := False;

    CheckListBox1.ItemIndex := 0;

    CheckListBox1.Selected[CheckListBox1.ItemIndex] := True;

  end;

end;

procedure TForm1.oneStep1;

var idx,rotation,sudut:integer;

    buff,par1,par2,par3:string;

begin

  //send index sekarang

  buff := perintah;

  edit1.Text := buff;

  par1 := ";

  par2 := ";

  par3 := ";

  if sameText(copy(buff,1,4),'SER ') then
```

```
begin
edit2.Text := 'ServoCommand';
idx:=5;
while buff[idx]<>' ' do
begin
par1:=par1 + buff[idx];
inc(idx);
end;
idx:=idx+1;
while buff[idx]<>' ' do
begin
par2:=par2 + buff[idx];
inc(idx);
end;
idx:=idx+1;
while (buff[idx]<>' ') and (idx<=length(buff)) do
begin
par3:=par3 + buff[idx];
```

```
inc(idx);

end;

edit3.Text := par1;

edit4.Text := par2;

edit5.Text := par3;

sudut:= strtoint(par3);

if sudut<0 then sudut:=0;

if sudut>180 then sudut:=180;

rotation := (sudut*1000) div 180 + 250;

edit6.Text := inttostr(rotation);

if rotation<300 then rotation:=300;

if rotation>1200 then rotation:=1200;

try

Comport1.WriteStr('!');

Comport1.WriteStr('S');

Comport1.WriteStr('C');

Comport1.WriteStr(chr(strtoint(par1) mod 256));

Comport1.WriteStr(chr(strtoint(par2) mod 256));
```



```
Comport1.WriteStr(chr(rotation mod 256));

Comport1.WriteStr(chr(rotation div 256));

Comport1.WriteStr(#13);

except

  ShowMessage('COM Port error !');

end;

end

else if sameText(copy(buff,1,4),'DEL ') then

begin

  edit2.Text := 'DelayCommand';

  idx:=5;

  while (buff[idx]<>' ') and (idx<=length(buff)) do

  begin

    par1:=par1 + buff[idx];

    inc(idx);

  end;

  edit3.Text := par1;
```

```
edit4.Text := par2;

edit5.Text := par3;

mySleep(strtoint(par1));

end

else

begin

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

var SomeTxtFile : TextFile;

    buff : string;

begin

if OpenFileDialog1.Execute then

begin

AssignFile(SomeTxtFile, OpenFileDialog1.FileName) ;

try

Reset(SomeTxtFile) ;

CheckListBox1.Clear;
```

```
baris := 0;

while not EOF(SomeTxtFile) do

begin

    ReadLn(SomeTxtFile, buff) ;

    CheckListBox1.Items.Add(buff);

    baris := baris + 1;

end;

finally

    CloseFile(SomeTxtFile) ;

    CheckListBox1.Selected[0] := True;

    CheckListBox1.ItemIndex := 0;

end;

edit16.Text := OpenFileDialog1.FileName;

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

    Comport1.ShowSetupDialog;
```

```
end;

procedure TForm1.Button3Click(Sender: TObject);

begin

    if Comport1.Connected then

        begin

            Comport1.Close;

            Button3.Caption := 'Open COM';

            GroupBox1.Enabled := False;

        end

    else

        begin

            Comport1.Open;

            Button3.Caption := 'Close COM';

            GroupBox1.Enabled := True;

        end;

    end;

end;

procedure TForm1.Button7Click(Sender: TObject);
```

```
begin

    Button7.Enabled := False;

    oneStep;

    Button7.Enabled := True;

end;

procedure TForm1.Button4Click(Sender: TObject);

var idx:integer;

begin

    Button4.Enabled := False;

    autoRun := True;

    For idx:=CheckListBox1.ItemIndex to CheckListBox1.Count-1 do

        begin

            oneStep;

            mySleep(strtoint(ComboBox1.Items[ComboBox1.ItemIndex]));

            Application.ProcessMessages;

            if not autoRun then break;

        end;

    Button4.Enabled := True;
```

```
end;

procedure TForm1.Button5Click(Sender: TObject);

begin

    autoRun := false;

end;

procedure TForm1.Button8Click(Sender: TObject);

var idx:integer;

begin

    for idx:=0 to CheckListBox1.Count-1 do

        CheckListBox1.Checked[idx] := False;

        CheckListBox1.ItemIndex := 0;

        CheckListBox1.Selected[CheckListBox1.ItemIndex] := True;

    end;

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

    Button6.Enabled := False;

    autoRun := True;

    while autoRun do
```

```
begin

    oneStep;

    mySleep(strtoint(ComboBox1.Items[ComboBox1.ItemIndex]));

    Application.ProcessMessages;

end;

Button6.Enabled := True;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

    sudutKepala := 95;

    jumlahTidakKetemu := 0;

    Timer1.Enabled := False;

    baruStart := True;

    counterMaju := 0;

    Edit17.Text := 'baruStart = True';

    if Comport1.Connected then

begin

    Comport1.Close;
```

```
Button3.Caption := 'Open COM';

GroupBox1.Enabled := False;

end

else

begin

Comport1.Open;

Button3.Caption := 'Close COM';

GroupBox1.Enabled := True;

end;

capture := cvCaptureFromCAM( 0);

if not(assigned(capture )) then

begin

MessageDlg('Could not initialize capturing from camera!!', mtError,
[mbOK], 0);

halt;

end;

bmp := TBitmap.Create;

bmp.PixelFormat := pf24bit;
```



```
    bmp1 := TBitmap.Create;

    bmp1.PixelFormat := pf24bit;

    bmp2 := TBitmap.Create;

    bmp2.PixelFormat := pf24bit;

    bmp3 := TBitmap.Create;

    bmp3.PixelFormat := pf24bit;

    bmp4 := TBitmap.Create;

    bmp4.PixelFormat := pf24bit;

    bmp5 := TBitmap.Create;

    bmp5.PixelFormat := pf24bit;

    bmp6 := TBitmap.Create;

    bmp6.PixelFormat := pf24bit;

    bmp7 := TBitmap.Create;

    bmp7.PixelFormat := pf24bit;

    bmp8 := TBitmap.Create;

    bmp8.PixelFormat := pf24bit;

    timer2.enabled := true;

end;
```

```
procedure TForm1.Button9Click(Sender: TObject);

var SomeTxtFile : TextFile;

    buff : string;

begin

if length(OpenDialog1.FileName)>0 then

begin

AssignFile(SomeTxtFile, OpenDialog1.FileName) ;

try

Reset(SomeTxtFile) ;

CheckListBox1.Clear;

baris := 0;

while not EOF(SomeTxtFile) do

begin

ReadLn(SomeTxtFile, buff) ;

CheckListBox1.Items.Add(buff);

baris := baris + 1;

end;

finally
```

```
CloseFile(SomeTxtFile) ;

CheckListBox1.Selected[0] := True;

CheckListBox1.ItemIndex := 0;

end;

end;

end;

procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);

begin

if (Key='S') OR (Key='s') then

begin

Button5.Click;

end

else if (Key='R') OR (Key='r') then

begin

Button4.Click;

end

else if (Key='L') OR (Key='l') then

begin
```

```
    Button6.Click;

end

else

begin

end;

end;

procedure TForm1.FormDestroy(Sender: TObject);

begin

    cvReleaseCapture( @capture );

end;

procedure TForm1.Timer1Timer(Sender: TObject);

begin

    Timer1.Enabled := False;

    main_cycle;

    application.HandleMessage;

    Timer1.Enabled := True;
```

```
if (csCTS in Form1.ComPort1.Signals) then
```

```
begin
```

```
    Timer1.Enabled := False;
```

```
    baruStart := True;
```

```
    counterMaju :=0;
```

```
end;
```

```
end;
```

```
procedure TForm1.tb1Change(Sender: TObject);
```

```
begin
```

```
    Edit7.Text := inttostr(tb1.Position);
```

```
end;
```

```
procedure TForm1.tb2Change(Sender: TObject);
```

```
begin
```

```
    Edit8.Text := inttostr(tb2.Position);
```

```
end;
```

```
procedure TForm1.tb3Change(Sender: TObject);
```

```
begin
```

```
    Edit9.Text := inttostr(tb3.Position);
```

```
end;
```

```
procedure TForm1.tb4Change(Sender: TObject);
```

```
begin
```

```
    Edit10.Text := inttostr(tb4.Position);
```

```
end;
```

```
procedure TForm1.tb5Change(Sender: TObject);
```

```
begin
```

```
    Edit11.Text := inttostr(tb5.Position);
```

```
end;
```

```
procedure TForm1.tb6Change(Sender: TObject);
```

```
begin
```

```

Edit12.Text := intostr(tb6.Position);

end;

procedure TForm1.Timer2Timer(Sender: TObject);

begin
    Timer2.Enabled := False;

    if baruStart then

        begin

            if (csDSR in Form1.ComPort1.Signals) then

                begin

                    counterMaju := counterMaju + 1;

                    Form1.OpenDialog1.FileName := 'C:\Documents and
Settings\anton\Desktop\Servo_prog\BALL2AN\JALAN.txt';

                    Form1.Button9.Click;

                    Form1.Button4.Click;

                    Form1.Edit16.Text := 'standBy';

                    if counterMaju>1 then

```

```
begin

    Timer1.Enabled := True;

    baruStart := False;

    Edit17.Text := 'baruStart = False';

end;

end;

end;

Timer2.Enabled := True;

end;

end.
```

- **Lampiran penulisan teks pada notepad**

```
unit Unit1;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms,

    Dialogs, ExtCtrls, MPlayer;
```



type

TForm1 = class(TForm)

Timer1: TTimer;

MediaPlayer1: TMediaPlayer;

procedure FormCreate(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

F: TextFile;

S: string;

implementation

{SR \*.dfm}

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
assignfile( F , 'C:\Documents and Settings\anton\Desktop\test.txt');
```

```
rewrite(F);
```

```
writeln(F,'red');
```

```
CloseFile(F);
```

```
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);
```

```
begin
```

```
Application.Terminate;
```

```
end;
```

```
end.
```

