LAMPIRAN A

SCRIPT DALAM PEMBUATAN PROGRAM UTAMA

```
unit UProgramCCTV;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Menus, jpeg, ExtCtrls, ImgList;

type
  TfrmProgramCCTV = class(TForm)
    GroupBox1: TGroupBox;
    MainMenu1: TMainMenu;
    Programs1: TMenuItem;
    ProgramCapture1: TMenuItem;
    N1: TMenuItem;
    ProgramVideoRecord1: TMenuItem;
    N2: TMenuItem;
    Exit1: TMenuItem;
    Image1: TImage;
    procedure ProgramCapture1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure ProgramVideoRecord1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  frmProgramCCTV: TfrmProgramCCTV;

implementation

uses UCapture, UVideo;

{$R *.dfm}

procedure TfrmProgramCCTV.ProgramCapture1Click(Sender: TObject);
begin
frmCapture.edJam.Text := FormatDateTime('hh mm ss',Now);
frmCapture.edTgl.Text := FormatDateTime('dd mmmm yyyy',Now);
frmCapture.show;
end;

procedure TfrmProgramCCTV.Exit1Click(Sender: TObject);
begin
Application.terminate;
end;

procedure TfrmProgramCCTV.ProgramVideoRecord1Click(Sender: TObject);
begin
frmVideo.show;
end;
end.
```

LAMPIRAN B

SCRIPT DALAM PEMBUATAN PROGRAM CAPTURE

```
unit UCapture;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DSPack, DirectShow9, DSUtil, StdCtrls, DBCtrls, ExtCtrls, Menus,
  Buttons, DB, ADODB, Mask, Grids, DBGrids;

type
  TfrmCapture = class(TForm)
    FilterGraph1: TFilterGraph;
    SampleGrabber1: TSampleGrabber;
    Filter1: TFilter;
    MainMenu1: TMainMenu;
    Device1: TMenuItem;
    Panel1: TPanel;
    Panel2: TPanel;
    VideoWindow1: TVideoWindow;
    DBImage1: TDBImage;
    btnManualCap: TButton;
    btnClose: TBitBtn;
    Label1: TLabel;
    Label2: TLabel;
    Timer1: TTimer;
```

```
btnAutoCap: TButton;
btnDisableAutoCap: TButton;
edJam: TEdit;
Label3: TLabel;
Label4: TLabel;
edTgl: TEdit;
SpeedButton1: TSpeedButton;
edInterval: TEdit;
Label5: TLabel;
Timer2: TTimer;
btnTimer: TBitBtn;
edTimer: TEdit;
Label6: TLabel;
SpeedButton2: TSpeedButton;
Button1: TButton;
Button2: TButton;
Label7: TLabel;
TimerCW: TTimer;
TimerCCW: TTimer;
procedure btnManualCapClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure btnCloseClick(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure btnAutoCapClick(Sender: TObject);
procedure btnDisableAutoCapClick(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure btnTimerClick(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure TimerCWTimer(Sender: TObject);
```

```
    procedure TimerCCWTimer(Sender: TObject);
    procedure Button1MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button1MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button2MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button2MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);


  private
    { Private declarations }
  public
    { Public declarations }
    Procedure OnSelectDevice(sender: TObject);
  end;

var
  frmCapture: TfrmCapture;
  SysDev: TSysDevEnum;
  simpan : tBitmap;
  x: integer;
  y: integer;

implementation
 function Out32(wAddr:word;bOut:byte):byte; stdcall; external 'inpout32.dll';
 function Inp32(wAddr:word):byte; stdcall; external 'inpout32.dll';


{$R *.dfm}
```

```
procedure TfrmCapture.btnManualCapClick(Sender: TObject);
begin
 edJam.Text := FormatDateTime('hh mm ss',Now);
 edTgl.Text := FormatDateTime('dd mmmm yyyy  ',Now);
 simpan:=TBitmap.Create;
 SampleGrabber1.GetBitmap(DBImage1.Picture.Bitmap);
 simpan:= DBImage1.Picture.Bitmap;
 simpan.SaveToFile( edTgl.Text   +  edJam.Text + '.bmp');


end;

procedure TfrmCapture.FormCreate(Sender: TObject);
var i: integer ;
    Device: TMenuItem;
Begin
SysDev:= TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
 if SysDev.CountFilters > 0 then
  for i := 0 to SysDev.CountFilters - 1 do
  begin
  Device := TMenuItem.Create(Device1);
  Device.Caption := SysDev.Filters[i].FriendlyName;
  Device.Tag := i ;
  Device.OnClick := OnSelectDevice ;
  Device1.Add(Device);
  end;
  Timer1.Enabled := false;
  Timer2.Enabled := false;
  btnTimer.Enabled := false;
```

```
end;

procedure TfrmCapture.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
begin
SysDev.Free;
FilterGraph1.ClearGraph;
FilterGraph1.Active := false;
end;

procedure TfrmCapture.OnSelectDevice(sender: TObject);
begin
FilterGraph1.ClearGraph;
FilterGraph1.Active := False;
Filter1.BaseFilter.Moniker := SysDev.GetMoniker(TMenuItem(Sender).Tag);
FilterGraph1.Active := True;
with FilterGraph1 as ICaptureGraphBuilder2 do
 RenderStream(@PIN_CATEGORY_PREVIEW, nil, Filter1 as IBaseFilter,
SampleGrabber1
 as IBaseFilter, VideoWindow1 as IBaseFilter);
 FilterGraph1.Play;
end;

procedure TfrmCapture.btnCloseClick(Sender: TObject);
begin
Timer1.Enabled :=  false;
frmCapture.Close;
end;

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

```
begin
if Timer1.Enabled = true then
 begin
 btnManualCap.Click;
 end
end;

procedure TfrmCapture.btnAutoCapClick(Sender: TObject);
begin
btnAutoCap.Enabled := false ;
btnManualCap.Enabled := false;
btnTimer.Enabled := true;
Timer1.Enabled := true ;

end;

procedure TfrmCapture.btnDisableAutoCapClick(Sender: TObject);
begin
btnAutoCap.Enabled := true;
btnManualCap.Enabled := true ;
Timer1.Enabled := false ;

end;



procedure TfrmCapture.SpeedButton1Click(Sender: TObject);
begin
Timer1.Interval := StrToInt(edInterval.Text);
edInterval.Color := clCream ;
SpeedButton1.Enabled := false ;
```

```
end;

procedure TfrmCapture.Timer2Timer(Sender: TObject);
begin
if Timer2.Enabled = true then
 begin
 btnDisableAutoCap.Click;
 btnTimer.Enabled := false;
 end
end;

procedure TfrmCapture.btnTimerClick(Sender: TObject);
begin
Timer2.Enabled := true ;
end;

procedure TfrmCapture.SpeedButton2Click(Sender: TObject);
begin
Timer2.Interval := StrToInt(edTimer.Text);
edTimer.Color := clCream ;
SpeedButton2.Enabled := false ;
end;

procedure TfrmCapture.TimerCWTimer(Sender: TObject);
begin
case x of
1: Out32 ($378,$01);
2: Out32 ($378,$02);
3: Out32 ($378,$04);
4: Out32 ($378,$08);
end;
```

```
inc(x);
if x=5 then
begin
x:=0;
end;
end;

procedure TfrmCapture.TimerCCWTimer(Sender: TObject);
begin
case y of
1: Out32 ($378,$08);
2: Out32 ($378,$04);
3: Out32 ($378,$02);
4: Out32 ($378,$01);
end;
inc(y);
if y=5 then
begin
y:=0;
end;
end;

procedure TfrmCapture.Button1MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
TimerCW.Enabled := true ;
end;

procedure TfrmCapture.Button1MouseUp(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
```

```
begin
TimerCW.Enabled := false ;
end;

procedure TfrmCapture.Button2MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
TimerCCW.Enabled := true ;
end;

procedure TfrmCapture.Button2MouseUp(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
TimerCCW.Enabled := false ;
end;

end.
```

LAMPIRAN C

SCRIPT DALAM PEMBUATAN PROGRAM VIDEO RECORD

```
unit UVideo;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DSPack, DirectShow9, DSUtil, StdCtrls, DBCtrls, ExtCtrls, Menus,
  Buttons, DB, ADODB, Mask, Grids, DBGrids, ComCtrls;

type
  TfrmVideo = class(TForm)
    ListBox1: TListBox;
    ListBox2: TListBox;
    ListBox3: TListBox;
    ListBox4: TListBox;
    Label1: TLabel;
    FilterGraph1: TFilterGraph;
    Filter1: TFilter;
    Filter2: TFilter;
    SaveDialog1: TSaveDialog;
    Timer1: TTimer;
    btnSave: TButton;
    btnStart: TButton;
    btnStop: TButton;
    VideoWindow1: TVideoWindow;
```

```
  StatusBar1: TStatusBar;
  ComboBox1: TComboBox;
  Label2: TLabel;
  Button1: TButton;
  Button2: TButton;
  TimerCw: TTimer;
  TimerCCw: TTimer;
  procedure FormCreate(Sender: TObject);
  procedure ListBox1Click(Sender: TObject);
  procedure ListBox3Click(Sender: TObject);
  procedure btnSaveClick(Sender: TObject);
  procedure btnStartClick(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure btnStopClick(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure TimerCwTimer(Sender: TObject);
  procedure TimerCCwTimer(Sender: TObject);
  procedure Button1MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure Button1MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure Button2MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure Button2MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  frmVideo: TfrmVideo;
  CapEnum: TSysDevEnum;
  VideoMediaTypes, AudioMediaTypes : TEnumMediaType;
  CapFile : WideString ='c:\Record.avi' ;
  x: integer;
  y: integer;
  SysDev: TSysDevEnum;



implementation

function Out32(wAddr:word;bOut:byte):byte; stdcall; external 'inpout32.dll';
function Inp32(wAddr:word):byte; stdcall; external 'inpout32.dll';



{$R *.dfm}

procedure TfrmVideo.FormCreate(Sender: TObject);
var i : integer ;
begin
CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
for i := 0 to CapEnum.CountFilters - 1 do
 ListBox1.Items.Add(CapEnum.Filters[i].FriendlyName);

 CapEnum.SelectGUIDCategory(CLSID_AudioInputDeviceCategory);
 for i := 0 to CapEnum.CountFilters - 1 do
 ListBox3.Items.Add(CapEnum.Filters[i].FriendlyName);

 VideoMediaTypes := TEnumMediaType.Create;
```

```
 AudioMediaTypes := TEnumMediaType.Create;
end;

procedure TfrmVideo.ListBox1Click(Sender: TObject);
var
PinList : TPinList;
i : Integer;
begin
 CapEnum.SelectGUIDCategory(CLSID_VideoInputDeviceCategory);
 if ListBox1.ItemIndex <> -1 then
 begin
 Filter1.BaseFilter.Moniker := CapEnum.GetMoniker(ListBox1.ItemIndex);
 Filter1.FilterGraph := FilterGraph1;
 FilterGraph1.Active := True;
 PinList := TPinList.Create(Filter1 as IBaseFilter);
 ListBox2.Clear;
 VideoMediaTypes.Assign(PinList.First);
 for i:= 0 to VideoMediaTypes.Count - 1 do
 ListBox2.Items.Add(VideoMediaTypes.MediaDescription[i]);
 FilterGraph1.Active := false;
 PinList.Free;
 btnStart.Enabled := true;
 end;

end;


procedure TfrmVideo.ListBox3Click(Sender: TObject);
var
PinList : TPinList;
i, LineIndex : Integer;
```

```
ABool : LongBool;
begin
CapEnum.SelectGUIDCategory(CLSID_AudioInputDeviceCategory);
if ListBox3.ItemIndex <> -1 then
 begin
 Filter2.BaseFilter.Moniker := CapEnum.GetMoniker(ListBox3.ItemIndex);
 Filter2.FilterGraph := FilterGraph1;
 FilterGraph1.Active := True;
 PinList := TPinList.Create(Filter2 as IBaseFilter);
 ListBox4.Clear;
 i:= 0;
 while i< PinList.Count do
  if PinList.PinInfo[i].dir = PINDIR_OUTPUT then
   begin
   AudioMediaTypes.Assign(PinList.Items[i]);
   PinList.Delete(i);
   end else inc(i);

 for i:= 0 to AudioMediaTypes.Count - 1 do
 begin
 ListBox4.Items.Add(AudioMediaTypes.MediaDescription[i]);
 end;

 FilterGraph1.Active := False;
 ComboBox1.Clear;
 LineIndex := -1 ;
 for i := 0 to PinList.Count - 1 do
 begin
 ComboBox1.Items.Add(PinList.PinInfo[i].achName);
 with (PinList.Items[i] as IAMAudioInputMixer) do get_Enable(ABool);
 if ABool then LineIndex := i ;
```

```
  end;

  ComboBox1.ItemIndex := LineIndex;
  PinList.Free;
  btnStart.Enabled := True;
  end;
end;


procedure TfrmVideo.btnSaveClick(Sender: TObject);
begin
if SaveDialog1.Execute then
begin
CapFile := SaveDialog1.FileName;
end;
end;

procedure TfrmVideo.btnStartClick(Sender: TObject);
var
multiplexer : IBaseFilter;
Writer : IFileSinkFilter;
PinList : TPinList;
i : Integer;
begin
FilterGraph1.Active := True;

if Filter2.FilterGraph <> nil then
 begin
 PinList := TPinList.Create(Filter2 as IBaseFilter);
 i := 0 ;
 while i < PinList.Count do
```

```
 if PinList.PinInfo[i].dir = PINDIR_OUTPUT then
  begin
  if ListBox4.ItemIndex <> -1 then
   with (PinList.Items[i] as IAMStreamConfig) do
  SetFormat(AudioMediaTypes.Items [ListBox4.ItemIndex].AMMediaType^);
   PinList.Delete(i);
   end else inc(i);
  if ComboBox1.ItemIndex <> -1 then
   with (PinList.Items[ComboBox1.ItemIndex] as IAMAudioInputMixer) do
   put_Enable(true);
   PinList.Free;
   end;

  if Filter1.FilterGraph <> nil then
  begin
  PinList := TPinList.Create(Filter1 as IBaseFilter);
  if ListBox2.ItemIndex <> - 1 then
   with (PinList.First as IAMStreamConfig) do
   SetFormat(VideoMediaTypes.Items [ListBox2.ItemIndex].AMMediaType^);
   PinList.Free;
   end;

   with FilterGraph1 as IcaptureGraphBuilder2 do
   begin
   SetOutputFileName(MEDIASUBTYPE_Avi,          PWideChar(CapFile),
multiplexer, Writer );

  if Filter1. BaseFilter.DataLength > 0 then
  RenderStream(@PIN_CATEGORY_PREVIEW , nil, Filter1 as IBaseFilter, nil,
VideoWindow1 as IBaseFilter);
```

```
  if Filter1.FilterGraph <> nil then
  RenderStream(@PIN_CATEGORY_CAPTURE, nil, Filter1 as IBaseFilter, nil,
multiplexer as IBaseFilter);

  if Filter2.FilterGraph <> nil then
   begin
   RenderStream(nil, nil, Filter2 As IBaseFilter, nil, multiplexer as IBaseFilter);
   end;
   end;

 FilterGraph1.Play;
 btnStop.enabled := true;
 btnStart.enabled := false;
 ListBox4.enabled := false;
 ListBox3.enabled := false;
 ListBox2.enabled := false;
 ListBox1.enabled := false;
 Timer1.enabled := true;

end;

procedure TfrmVideo.Timer1Timer(Sender: TObject);
var
position : Int64;
Hour, Min, Sec, Msec : Word;
const MiliSecInOneDay = 86400000;
begin
if FilterGraph1.Active then ;
begin
with FilterGraph1 as IMediaSeeking do
GetCurrentPosition(position);
```

```
DecodeTime(position div 1000 / MiliSecInOneDay, Hour, Min, Sec, MSec);
StatusBar1.SimpleText := Format('%d:%d:%d:%d', [Hour, Min, Sec, MSec]);
end;
end;

procedure TfrmVideo.btnStopClick(Sender: TObject);
begin
Timer1.Enabled := False;
btnStop.Enabled := False;
btnStart.Enabled := True;
FilterGraph1.Stop;
FilterGraph1.Active := False;
ListBox4.Enabled := true ;
ListBox3.Enabled := true ;
ListBox2.Enabled := true ;
ListBox1.Enabled := true ;
end;

procedure TfrmVideo.FormDestroy(Sender: TObject);
begin
CapEnum.Free;
VideoMediaTypes.Free;
AudioMediaTypes.Free;
end;

procedure TfrmVideo.TimerCwTimer(Sender: TObject);
begin
case x of
1: Out32 ($378,$01);
2: Out32 ($378,$02);
3: Out32 ($378,$04);
```

```
4: Out32 ($378,$08);
end;
inc(x);
if x=5 then
begin
x:=0;
end;
end;

procedure TfrmVideo.TimerCCwTimer(Sender: TObject);
begin
case y of
1: Out32 ($378,$08);
2: Out32 ($378,$04);
3: Out32 ($378,$02);
4: Out32 ($378,$01);
end;
inc(y);
if y=5 then
begin
y:=0;
end;
end;

procedure TfrmVideo.Button1MouseDown(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
TimerCw.Enabled := true ;
end;
```

```
procedure TfrmVideo.Button1MouseUp(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
TimerCw.Enabled := false ;
end;

procedure TfrmVideo.Button2MouseDown(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
TimerCCw.Enabled := true ;
end;

procedure TfrmVideo.Button2MouseUp(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
TimerCCw.Enabled := false ;
end;

end.
```