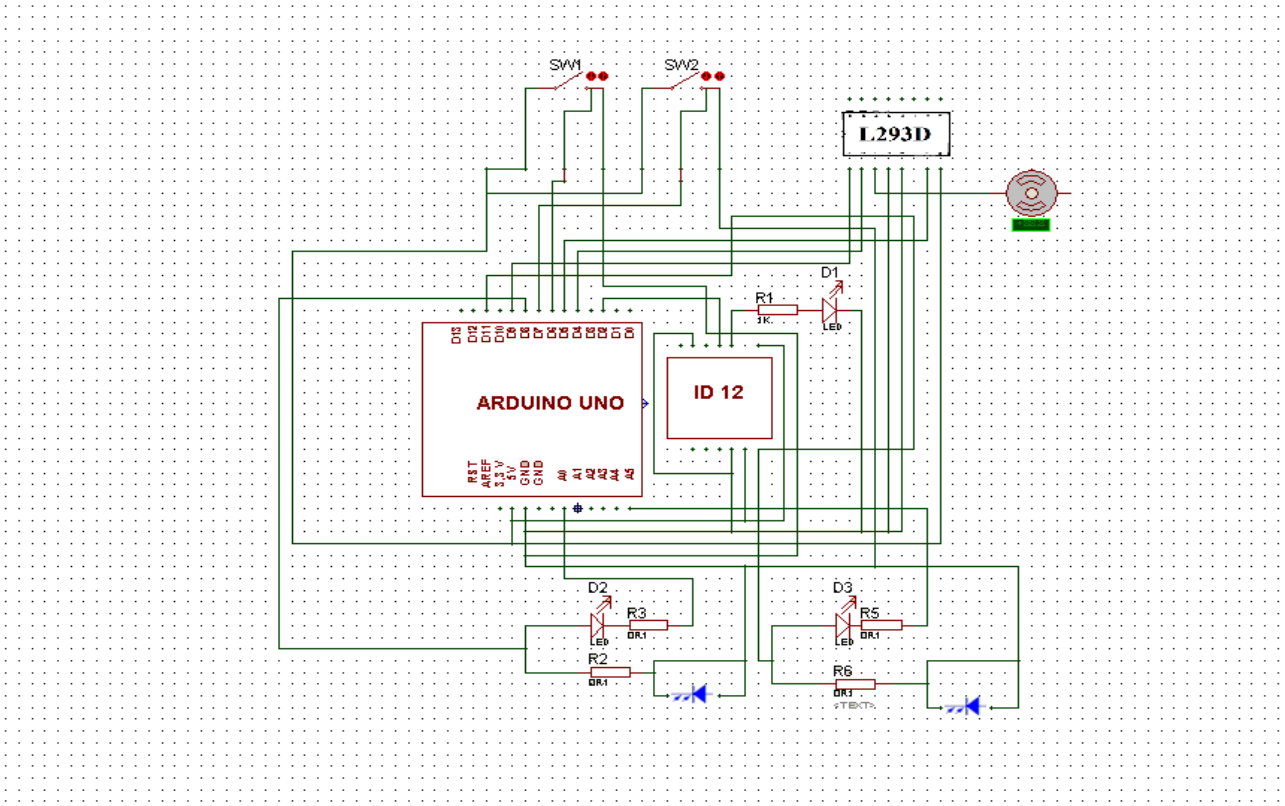
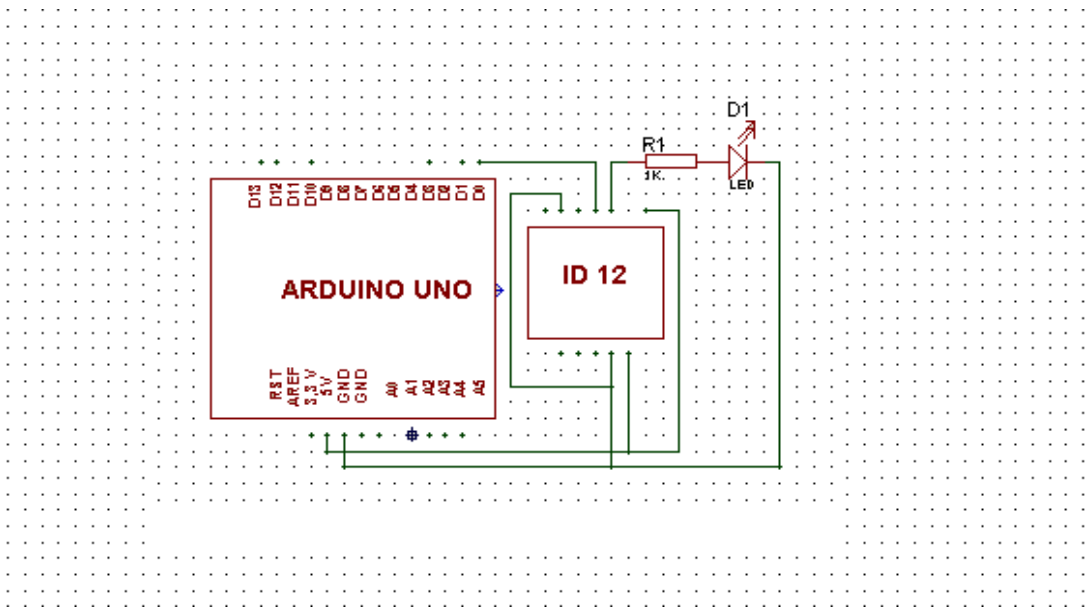
LAMPIRAN


SW1
SW2
L293D
D1
R1
1K.
LED
ARDUINO UNO
ID 12
D2
R3
R2
D3
R5
R6


D1
R1
1K.
LED
ARDUINO UNO
ID 12

RFID 1

```
#include <SoftwareSerial.h>
#define rxPin 2
#define txPin 3

int motor1=4;
int motor2=5;
int switchPin1=6;
int switchPin2=7;
int enable=9;
int infrared1= 8;
int infrared2=11;
char tag=0;
int val=0;
int sensor1;
int sensor2;
int buka=0;

SoftwareSerial rfid = SoftwareSerial( rxPin, txPin );
void setup() {
  rfid.begin(9600);
  Serial.begin(9600);
  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  pinMode(switchPin1, INPUT);
  pinMode(switchPin2, INPUT);
  pinMode(infrared1, OUTPUT);
  pinMode(infrared2, OUTPUT);
  pinMode(enable, OUTPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
 digitalWrite(enable, LOW);
 digitalWrite(infrared1, HIGH);
 digitalWrite(infrared2, HIGH);

}

void loop() {

sensor1=analogRead(A0);
sensor2=analogRead(A5);

tag =rfid.read();
if(tag!=-1){
```

```
Serial.print(tag);

val= Serial.read();
if(val!=-1){

if (val=='a'){                          //buka pintu
digitalWrite(enable, HIGH);
digitalWrite(motor1, HIGH);
digitalWrite(motor2, LOW);

if (digitalRead(switchPin2) == HIGH) {
digitalWrite(enable, LOW);
buka=1;
}
}

else if (val=='b'){
digitalWrite(enable, LOW);
}

 if (val=='c'){                         //buka pintu
digitalWrite(enable, HIGH);
digitalWrite(motor1, HIGH);
digitalWrite(motor2, LOW);

if (digitalRead(switchPin2) == HIGH) {
digitalWrite(enable, LOW);
buka=1;
}
}

else if (val=='d'){
digitalWrite(enable, LOW);
}
}
}

void tutup() ;

}

void tutup() {
if((sensor1<=100)&&(buka==1){
delay (2000);
digitalWrite(enable, HIGH);
```

```
digitalWrite(motor1, LOW);
digitalWrite(motor2, HIGH);
buka=0;
}

if((sensor2<=100)&&(buka==1)){
delay (2000);
digitalWrite(enable, HIGH);
digitalWrite(motor1, LOW);
digitalWrite(motor2, HIGH);
buka=0;
}

if (digitalRead(switchPin1) == HIGH) {
digitalWrite(enable, LOW);

}
}
```

RFID 2

```
char val = 0; // variable to store the data from the serial port

void setup() {
Serial.begin(9600); // connect to the serial port
}

void loop () {
// read the serial port

if(Serial.available() > 0) {
val = Serial.read();
Serial.print(val, BYTE);

}
}
```

Listing Program Pada Delphi

Program Menu Utama

```
unit U_MenuUtama;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, XPMan, ExtCtrls, ToolWin, ComCtrls, Buttons, Menus;

type
  TfrmMenu = class(TForm)
    Bevel1: TBevel;
    StatusBar1: TStatusBar;
    Timer1: TTimer;
    MainMenu1: TMainMenu;
    in1: TMenuItem;
    DataID1: TMenuItem;
    DataLog1: TMenuItem;
    Keluar1: TMenuItem;
    procedure Timer1Timer(Sender: TObject);
    procedure in1Click(Sender: TObject);
    procedure DataID1Click(Sender: TObject);
    procedure DataLog1Click(Sender: TObject);
    procedure Keluar1Click(Sender: TObject);


  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmMenu: TfrmMenu;

implementation

uses U_DataId, U_Input, U_Mod, U_Hapus,U_Status, U_DataLog,
  U_login;

{$R *.dfm}
```

```
procedure TfrmMenu.Timer1Timer(Sender: TObject);
begin
StatusBar1.Panels[1].Text := FormatDateTime('dd mmmm yyyy',now);
StatusBar1.Panels[2].Text := FormatDateTime('hh:mm:ss', now);
end;



procedure TfrmMenu.in1Click(Sender: TObject);
begin
frmStatus.Show;
end;

procedure TfrmMenu.DataID1Click(Sender: TObject);
begin
frmLogin.show;
end;

procedure TfrmMenu.DataLog1Click(Sender: TObject);
begin
frmDataLog.Show;
end;

procedure TfrmMenu.Keluar1Click(Sender: TObject);
begin
application.Terminate;
end;

end.
```

Program Data ID

```
unit U_DataId;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Grids, DBGrids, Buttons, ToolWin, ComCtrls,
  DB, IBCustomDataSet, IBQuery, RpCon, RpConDS, RpDefine, RpRave, Menus;

type
  TfrmDataId = class(TForm)
    dbgMobil: TDBGrid;
```

```
    StatusBar1: TStatusBar;
    MainMenu1: TMainMenu;
    InputData1: TMenuItem;
    HapusData1: TMenuItem;
    Keluar1: TMenuItem;
    procedure FormActivate(Sender: TObject);
    procedure InputData1Click(Sender: TObject);
    procedure HapusData1Click(Sender: TObject);
    procedure Keluar1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmDataId: TfrmDataId;

implementation

uses U_Input, U_MenuUtama, U_Mod, U_Hapus,U_Status;

{$R *.dfm}


procedure TfrmDataId.FormActivate(Sender: TObject);
begin
with dm.qProses Do
    begin
    close;
    sql.clear;
    sql.add('Select * From RFID order by nama');
    open;
    end;

end;

procedure TfrmDataId.InputData1Click(Sender: TObject);
begin
frmInput.Show;

end;

procedure TfrmDataId.HapusData1Click(Sender: TObject);
```

```
begin
frmHapus.show;
end;

procedure TfrmDataId.Keluar1Click(Sender: TObject);
begin
close;
end;

end.
```

Program Login

```
unit U_login;

interface

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

type
  TfrmLogin = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    btnOk: TBitBtn;
    btnCancel: TBitBtn;
    edNama: TEdit;
    edPass: TEdit;
    procedure btnOkClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmLogin: TfrmLogin;

implementation

uses U_MenuUtama, U_DataId;

{$R *.dfm}
```

```
procedure TfrmLogin.btnOkClick(Sender: TObject);
begin
if (EdNama.Text= 'admin') and (EdPass.Text<>'123') then
begin
MessageDlg('ID atau password salah', MTWarning, [MBOK], 0);
Exit;
end;
frmDataId.Show;
frmLogin.hide;
end;

end.
```

Program Input Data

```
unit U_Input;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, StdCtrls, Buttons, ComCtrls, ExtCtrls;

type
  TfrmInput = class(TForm)
    Label2: TLabel;
    Label4: TLabel;
    edNama: TEdit;
    edNoId: TEdit;
    BitBtn1: TBitBtn;
    BitBtn4: TBitBtn;
    StatusBar1: TStatusBar;
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmInput: TfrmInput;

implementation
```

```
uses U_MenuUtama, U_Mod, U_DataId,U_Status, U_Tanya;

{$R *.dfm}

procedure TfrmInput.BitBtn4Click(Sender: TObject);
begin
frmDataId.show;
frmInput.Hide;
end;

procedure TfrmInput.BitBtn1Click(Sender: TObject);
begin
With dm.QData Do
  Begin
    Close;
    Sql.Clear;
    Sql.Add('Insert Into RFID(Noid, nama) '+
    'Values(:no, :na)');
    Prepare;
    ParamByName('no').AsString := EdNoId.Text;
    ParamByName('na').AsString := EdNama.Text;


    ExecSql;
  End;
  MessageDlg('Apakah data telah diisi dengan lengkap?',mtWarning,[mbOK],0);
  dm.trGerbang.Commit;

  EdNoId.Clear;
  EdNama.Clear;

  EdNoId.SetFocus;


  With dm.QProses Do
  Begin
    Close;
    Sql.Clear;
    Sql.Add('Select * From RFID');
    Open;
    frmTanya.show;
    frmInput.Hide;
  End;
end;
```

```
end.
```

Program Hapus Data

```
unit U_Hapus;

interface

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

type
  TfrmHapus = class(TForm)
    Label1: TLabel;
    edNama: TEdit;
    btnOk: TBitBtn;
    btnCancel: TBitBtn;
    procedure btnOkClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmHapus: TfrmHapus;

implementation

uses U_Mod, U_Input, U_DataId, U_Tanya2;

{$R *.dfm}

procedure TfrmHapus.btnOkClick(Sender: TObject);
begin
With dm.QProses Do
  Begin
    Close;
    Sql.Clear;
    Sql.Add('Delete From RFID');
    Sql.Add('Where Noid =:no');
    Prepare;
```

```
   ParamByName('no').AsString := EdNama.Text;
   ExecSql;
  End;

  dm.trGerbang.Commit;
  MessageDlg('Hapus data yang telah dipilih?',mtWarning,[mbOK],0);

  With dm.QProses Do
  Begin
    Close;
    Sql.Clear;
    Sql.Add('Select * From RFID');
    Open;
    frmTanya2.show;
  frmHapuspelanggan.Hide;
end;
end;
procedure TfrmHapus.btnCancelClick(Sender: TObject);
begin
frmDataId.show;
frmHapus.Hide;
end;

end.
```

Program Status

```
unit U_Status;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, CPort, CPortCtl, ExtCtrls, Buttons, ComCtrls, Menus;

type
  TfrmStatus = class(TForm)
    ComPort1: TComPort;
    Edit1: TEdit;
    StatusBar1: TStatusBar;
    Timer2: TTimer;
    Edit2: TEdit;
    Edit3: TEdit;
    BitBtn1: TBitBtn;
    Edit4: TEdit;
```

```
    Timer3: TTimer;
    Label7: TLabel;
    Label8: TLabel;
    Label16: TLabel;
    Label18: TLabel;
    ComPort2: TComPort;
    CheckBox1: TCheckBox;
    Label1: TLabel;
    edNoId: TEdit;
    MainMenu1: TMainMenu;
    Keluar1: TMenuItem;
    CheckBox2: TCheckBox;

    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure btnKeluarClick(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure Timer3Timer(Sender: TObject);
    procedure ComPort2RxChar(Sender: TObject; Count: Integer);
    procedure CheckBox1Click(Sender: TObject);
    procedure Keluar1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);


  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmStatus: TfrmStatus;
  awal,akhir : tDateTime;
implementation
 uses U_Mod, U_MenuUtama;
{$R *.dfm}


procedure TfrmStatus.ComPort1RxChar(Sender: TObject; Count: Integer);
var
Str:String;

begin
```

```
comport1.ReadStr(Str,count);


edit1.Text:=edit1.Text+Str;
if (copy(edit1.text,1,1)='   ') and (copy(edit1.text,16,1)='   ') then
begin
  str:=Copy(edit1.Text,2,length(edit1.text)-4);
 edNoid.Text:=Str;


  With dm.QData Do
 Begin
  Close;
  Sql.Clear;
  Sql.Add('Select * From RFID');
  sql.Add('Where noid =:no');
  prepare;
  parambyname('no').AsString := trim(edNoid.Text);
  Open;
    edit1.Text := fields[0].AsString;

  if(edit1.Text='') then
  begin
  label18.Caption:= '  ';
  end
  else
  label18.Caption:='Mobil Keluar';
  checkBox1.Checked:=True;
    timer3.Enabled := false;
    awal := time;
    edit2.Text := timetostr(time);
    With dm.qData do
      begin
      close;
      sql.clear;
      dm.qData.Sql.Add('update log set jamkeluar =:jk');
      dm.qData.Sql.Add('where noid =:no and jammasuk =:jm');
      Prepare;
      ParamByName('no').AsString := Trim(edNoId.Text);
      ParamByName('jk').AsString := Trim(edit2.Text);
      ParamByName('jm').AsString := Trim(edit3.Text);
      execsql;
      end;
      dm.trGerbang.Commit;
      edit1.Clear;
```

```
      end;
      end;


end;

procedure TfrmStatus.btnKeluarClick(Sender: TObject);
begin
close;
end;

procedure TfrmStatus.Timer2Timer(Sender: TObject);
begin
akhir := time;
edit3.Text := timetostr(akhir);
edit4.Text := datetostr(date);
end;


procedure TfrmStatus.FormCreate(Sender: TObject);
begin
comport1.Open;
comport2.Open;
timer2.Interval := 1000;

end;

procedure TfrmStatus.BitBtn1Click(Sender: TObject);
begin
timer2.Enabled := false;
timer2.Enabled := true;
edit3.Text := timetostr(time);
timer3.Enabled := false;
timer3.Enabled := true;
edit2.Text := timetostr(time);
checkbox1.Checked:=false;
checkbox2.Checked:=false;

end;

procedure TfrmStatus.Timer3Timer(Sender: TObject);
begin
akhir := time;
edit2.Text := timetostr(akhir);
end;
```

```
procedure TfrmStatus.ComPort2RxChar(Sender: TObject; Count: Integer);
var
Str:String;

begin
comport2.ReadStr(Str,count);


edit1.Text:=edit1.Text+Str;
if (copy(edit1.text,1,1)='   ') and (copy(edit1.text,16,1)='   ') then
begin
  str:=Copy(edit1.Text,2,length(edit1.text)-4);
 edNoId.Text:=Str;

  With dm.QData Do
 Begin
  Close;
  Sql.Clear;
  Sql.Add('Select * From RFID');
  sql.Add('Where noid =:no');
  prepare;
  parambyname('no').AsString := trim(edNoid.Text);
  Open;
  edit1.Text := fields[0].AsString;

  if(edit1.Text='') then
  begin
  label18.Caption:= '  ';
  end
  else

  label18.Caption:='Mobil masuk';
  checkBox2.Checked:=True;
  timer2.Enabled := false;
  awal := time;
  edit3.Text := timetostr(time);
   With dm.QData Do
 Begin
  Close;
  Sql.Clear;
 Sql.Add('Insert Into log(Noid, jammasuk, tanggal) '+
 'Values(:no, :jm, :ta)');
```

```
   Prepare;
  ParamByName('no').AsString := EdNoId.Text;
  ParamByName('jm').AsString := Edit3.Text;
  ParamByName('ta').AsString := Edit4.Text;

   ExecSql;
 End;

 dm.trGerbang.Commit;
 edit1.Clear;

     end;

end;
end;

procedure TfrmStatus.CheckBox1Click(Sender: TObject);
begin
  if CheckBox1.Checked=true then
   comport1.WriteStr('a')
   else
   comport1.WriteStr('b')

   end;

procedure TfrmStatus.Keluar1Click(Sender: TObject);
begin
 close;
end;

procedure TfrmStatus.CheckBox2Click(Sender: TObject);
begin
   if CheckBox2.Checked=true then
   comport1.WriteStr('c')
   else
   comport1.WriteStr('d')
end;

end.
```

Program Data Log

```
unit U_DataLog;

interface
```

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Grids, DBGrids, Menus;

type
  TfrmDataLog = class(TForm)
    DBGrid1: TDBGrid;
    MainMenu1: TMainMenu;
    k1: TMenuItem;
    procedure FormActivate(Sender: TObject);

    procedure k1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmDataLog: TfrmDataLog;

implementation
  uses U_Mod, U_MenuUtama;
{$R *.dfm}

procedure TfrmDataLog.FormActivate(Sender: TObject);
begin
with dm.qProses Do
    begin
    close;
    sql.clear;
    sql.add('Select * From log order by noid');
    open;
    end;
end;



procedure TfrmDataLog.k1Click(Sender: TObject);
begin
close;
end;

end.
```

Program Tanya

```
unit U_Tanya;

interface

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

type
  TfrmTanya = class(TForm)
    Label1: TLabel;
    btnCancel: TBitBtn;
    btnOk: TBitBtn;
    procedure btnOkClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmTanya: TfrmTanya;

implementation

uses U_Input, U_DataId;

{$R *.dfm}

procedure TfrmTanya.btnOkClick(Sender: TObject);
begin
frmInput.Show;
frmTanya.Hide;
end;

procedure TfrmTanya.btnCancelClick(Sender: TObject);
begin
frmDataId.Show;
frmTanya.Hide;
end;
```

```
end.
```

Program Tanya2

```
unit U_Tanya2;

interface

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

type
  TfrmTanya2 = class(TForm)
    Label1: TLabel;
    btnCancel: TBitBtn;
    btnOk: TBitBtn;
    procedure btnOkClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmTanya2: TfrmTanya2;

implementation

uses U_Hapus, U_DataId;

{$R *.dfm}

procedure TfrmTanya2.btnOkClick(Sender: TObject);
begin
frmHapus.show;
frmTanya2.Hide;
end;

procedure TfrmTanya2.btnCancelClick(Sender: TObject);
begin
frmDataId.Show;
frmTanya2.Hide;
```

end;

end.

Pintu geser (*Sliding Door*) merupakan sebuah cara yang baik untuk menciptakan ruang yang bisa digunakan untuk parkir dari daerah yang memerlukan banyak ruang.

Pintu geser (*Sliding door*) juga merupakan solusi yang baik jika jarak gedung lebih jauh dari gerbang dan untuk membuka pintu gerbang dapat dilakukan dari jarak jauh.

. Pintu geser (*Sliding Door*) yang ideal dapat dioperasikan dengan manual atau otomatis. Tips yang bagus untuk memasang Pintu Geser(Sliding Door), adalah diperlukan gerbang tunggal, terdiri dari 2 roda mudah-*roll* (dipasang ke pintu gerbang. Salah satu contoh gerbang otomatis:

MERK : DITEC Made in Italy

TYPE : CROSS 8E (untuk berat s/ d 800 Kg)

Pagar Otomatis *Sliding/ Swing* merupakan alat/ mesin untuk membuka dan menutup pintu gerbang secara otomatis. Tipe CROSS 8E dapat menggerakan beban pintu gerbang sampai dengan berat 800 Kg. Untuk membuka gerbang dapat menggunakan

Remote Control atau diintegrasikan dengan Access Control atau Home Automation System.