

DAFTAR PUSTAKA

- [1] Wirenro Sumargo, Soelthon Gussedyo Nanggara, Frionny Nainggolan, Isnenti Apriani. **“Potret Keadaan Hutan Indonesia”**. Forest Watch Indonesia, 2011.
- [2] Peraturan Pemerintah Republik Indonesia Nomor 60 Tahun 2012, **“Perubahan Atas Peraturan Pemerintah Nomor 10 Tahun 2010 Tentang Tata Cara Perubahan Peruntukan dan Fungsi Kawasan Hutan”**, 2012.
- [3] Dudy Subagdja, **“Nasib Hutan Kita dan Kebijakan Ekonomi Hijauh”**, Berita Kompasiana, 29 Maret 2013.
- [4] Christina Basaria S., **“Kajian Kelestarian Tegakan Dan Produksi Kayu Jati Jangka Panjang KPH Bojonegoro Perum Perhutani Unit II Jawa Timu”**, Institut Pertanian Bogor, 2009.
- [5] Ni Luh Putu Satyaning Pradnya Paramita, Irhamah **“Peramalan beban listrik menggunakan genetic algorithm-support vector machine (ga-svm) di PT. PLN (Persero) sub unit penyaluran dan pusat pengaturan beban (p3b) Jawa Timur-Bali”**, ITS, 2011.
- [6] Desi Yuniarti **“Peramalan jumlah penumpang yang berangkat melalui Bandar Udara Temindung Samarinda Tahun 2012 dengan metode ARIMA Box-Jenkins”**, Universitas Mulawarman, 2012.

- [7] Melly Lukman, Eko Susanto, “**Teknik exponensial smoothing untuk peramalan debit aliran sungai**”, Universitas Diponegoro, 2007.
- [8] I Made Mataram “**Peramalan beban hari libur menggunakan artificial neural network**” , Universitas Udayana, 2008.
- [9] T.Sutojo, Edy Mulyanto, Vincent Suhartono, 2011, “**Kecerdasan Buatan**”, penerbit ANDI Yogyakarta dengan UDINUS Semarang
- [10] Sam'ani, “**Rancangan bangun sistem penjadwalan perkuliahan dan ujian akhir semester dengan pendekatan algoritma genetika**”, Universitas Diponegoro, Semarang, 2012.
- [11] Son Kuswadi, 2007, “**Kendali Cerdas**”, Penerbit Andi Yogyakarta.
- [12] Sheng Lu, Zhong-jian Cai, “**Application of GA-SVM Time Series Prediction in Tax Forecasting**”, Chongqing Technology and Business University Chongqing, China, 2009
- [13] Xiaogang Chen, “**Prediction of Urban Water Demand Based on GA-SVM**”, Huaiyin Institute of Technology, Huaian, Chine, 2010.
- [14] Vahid Khatibi, Elham Khatibi dan Abdolreza Rasouli, “**A new support vector machine- genetic algorithm (SVM-GA) based method for stock market forecasting**”, Islamic Azad University, Bardsir, Iran, 2011.
- [15] Thi Nguyen, Lee Gordon-Brown, Peter Wheeler, “**GA-SVM Based Framework for Time Series Forecasting**”, Jim Peterson, Monash University, Australia, 2009.

- [16] Dyah Pratiwi, dkk., “**Penghitungan laju laus area hutan berbasis algoritma segmentasi warna local**”, Universitas Gunadarma, 2013.
- [17] F. Deppe, “**Forest Area Estimation Using Sample Survey and Landsat MSS and TM Data**”, Federal University of Rio Grande do Sul, American, 1998.
- [18] Dang Khoi dan Yuji Murayama, “**Forecasting Areas Vulnerable to Forest Conversion in the Tam Dao National Park Region**”, University of Tsukuba, 2010.
- [19] Andrew O. Finley, dkk., “**A Bayesian approach to multi-source forest area estimation**”, USA, 2007.
- [20] Loghman Ghahramany, Pariz Fatehi, Hedayat Ghazanfari, “**Estimation of Basal Area in West Oak Forests of Iran Using Remote Sensing Imagery**”, University of Kurdistan, Iran, 2010.
- [21] Oliver Diedershagen, Barbara Koch, “**AUTOMATIC ESTIMATION OF FOREST INVENTORY PARAMETERS BASED ON LIDAR, MULTI-SPECTRAL AND FOGIS DATA**”, Holgen Weinacker, University Freiburg, Germany, 2003.
- [22] Anuj Karpatne, dkk, “**Importance of Vegetation Type in Forest Cover Estimation**”, University of Minnesota, 2010.

- [23] Farshad Keivan Behjou, Mahbobe Foshat, "**Using A Sampling Method For Estimation for Forest Canopy cover**", Ardabili, Iran, 2013.
- [24] Lauri Korhonen, Kari T. Khorhoneni, Miina Rautianen and Pauline Stenberg, "**Estimation of Forest Canopy Cover : a Comparison of field measurement Techniques**", University of Joensuu, Finland, 2006.

Listing Program Aplikasi Estimasi

```
unit Uinputdatapemicu;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, Buttons, ExtCtrls, DB, MemDS,
DBAccess, MyAccess;

type

Tinputdatapemicu = class(TForm)

Label2: TLabel;
Bevel1: TBevel;
GroupBox1: TGroupBox;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
DBGrid1: TDBGrid;
Label1: TLabel;
ComboBox1: TComboBox;
Label3: TLabel;
Label4: TLabel;
```

```
Label5: TLabel;  
  
MyQuery1: TMyQuery;  
  
MyQuery2: TMyQuery;  
  
MyQuery3: TMyQuery;  
  
ComboBox2: TComboBox;  
  
ComboBox3: TComboBox;  
  
Edit1: TEdit;  
  
DataSource1: TDataSource;  
  
MyQuery4: TMyQuery;  
  
MyQuery5: TMyQuery;  
  
procedure BitBtn1Click(Sender: TObject);  
  
procedure FormClose(Sender: TObject; var Action: TCloseAction);  
  
procedure BitBtn2Click(Sender: TObject);  
  
procedure FormCreate(Sender: TObject);  
  
procedure ComboBox1Change(Sender: TObject);  
  
procedure ComboBox2Change(Sender: TObject);  
  
procedure ComboBox3Change(Sender: TObject);  
  
procedure DBGrid1DblClick(Sender: TObject);  
  
procedure BitBtn3Click(Sender: TObject);  
  
procedure BitBtn4Click(Sender: TObject);  
  
private  
  { Private declarations }  
  
public  
  { Public declarations }  
  
end;
```

```
var  
  inputdatapemicu: Tinputdatapemicu;
```

```
implementation
```

```
uses UTNK;
```

```
{$R *.dfm}
```

```
procedure Tinputdatapemicu.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
  Combobox1.Text := ";
```

```
  Combobox2.Text := ";
```

```
  Combobox3.Text := ";
```

```
  Edit1.Text := ";
```

```
  bitbtn3.Enabled := false;
```

```
  bitbtn4.Enabled := false;
```

```
  bitbtn2.Enabled := true;
```

```
  Combobox1.SetFocus;
```

```
  FormCreate(Sender);
```

```
end;
```

```
procedure Tinputdatapemicu.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  Action := Cafree;
  TNK.InputDataPemicu1.Enabled := true;
end;

procedure Tinputdatapemicu.BitBtn2Click(Sender: TObject);
begin
  if Combobox1.Text = " then
    begin
      showmessage('Tahun Masih Kosong');
      Combobox1.SetFocus;
      exit;
    end;

  if Combobox2.Text = " then
    begin
      Showmessage('Wilayah Masih Kosong');
      Combobox2.SetFocus;
      exit;
    end;

  if Combobox3.Text = " then
    begin
```

```
Showmessage('Pemicu Masih Kosong');

Combobox3.SetFocus;

exit;

end;

if Edit1.Text = " then

begin

Showmessage('Luas Areal Masih Kosong');

Edit1.SetFocus;

exit;

end;

Myquery1.SQL.Clear;

Myquery1.SQL.Add('select count(*) from data_pemicu where tahun = :a1 and wilayah =
:a2 and pemicu = :a3');

Myquery1.Params[0].AsString := Combobox1.Text;

Myquery1.Params[1].AsString := Combobox2.Text;

Myquery1.Params[2].AsString := Combobox3.Text;

Myquery1.Open;

If Myquery1.Fields[0].AsInteger = 0 then

begin

Myquery2.SQL.Clear;

Myquery2.SQL.Add('insert into data_pemicu(tahun,wilayah,pemicu,luas_areal,k1,k2)
values(:a1,:a2,:a3,:a4,:a5,:a6)');

Myquery2.Params[0].AsString := Combobox1.Text;
```

```
Myquery2.Params[1].AsString := Combobox2.Text;  
Myquery2.Params[2].AsString := Combobox3.Text;  
Myquery2.Params[3].AsFloat := strtofloat(Edit1.Text);  
Myquery2.Params[4].AsString := ";"  
Myquery2.Params[5].AsString := ";"  
Myquery2.Execute;  
  
Showmessage('Data telah tersimpan');  
BitBtn1Click(Sender);  
FormCreate(Sender);  
end  
else  
begin  
Showmessage('Maaf data Sudah ada');  
end;  
  
end;  
  
procedure Tinputdatapemicu.FormCreate(Sender: TObject);  
begin  
myquery3.SQL.Clear;  
myquery3.SQL.Add('select * from data_pemicu order by tahun asc ');  
myquery3.Open;  
end;
```

```
procedure Tinputdatapemicu.ComboBox1Change(Sender: TObject);
begin
  myquery3.SQL.Clear;
  myquery3.SQL.Add('select * from data_pemicu where tahun = :a1 order by tahun asc ');
  myquery3.Params[0].AsString := combobox1.Text;
  myquery3.Open;

end;

procedure Tinputdatapemicu.ComboBox2Change(Sender: TObject);
begin
  myquery3.SQL.Clear;
  myquery3.SQL.Add('select * from data_pemicu where tahun = :a1 and wilayah = :a2
order by tahun asc ');
  myquery3.Params[0].AsString := combobox1.Text;
  myquery3.Params[1].AsString := combobox2.Text;
  myquery3.Open;

end;

procedure Tinputdatapemicu.ComboBox3Change(Sender: TObject);
begin
  myquery3.SQL.Clear;
  myquery3.SQL.Add('select * from data_pemicu where tahun = :a1 and wilayah = :a2
and pemicu = :a3 order by tahun asc ');
  myquery3.Params[0].AsString := combobox1.Text;
```

```
myquery3.Params[1].AsString := combobox2.Text;  
myquery3.Params[2].AsString := combobox3.Text;  
myquery3.Open;  
  
end;  
  
procedure Tinputdatapemicu.DBGrid1DblClick(Sender: TObject);  
begin  
bitbtn3.Enabled := true;  
bitbtn4.Enabled := true;  
bitbtn2.Enabled := false;  
combobox1.Text := dbgrid1.Fields[0].AsString;  
combobox2.Text := dbgrid1.Fields[1].AsString;  
combobox3.Text := dbgrid1.Fields[2].AsString;  
Edit1.Text := dbgrid1.Fields[3].AsString;  
  
myquery4.SQL.Clear;  
myquery4.SQL.Add('select nomor from data_pemicu where tahun = :a1 and wilayah =  
:a2 and pemicu = :a3 and luas_areal = :a4');  
myquery4.Params[0].AsString := combobox1.Text;  
myquery4.Params[1].AsString := combobox2.Text;  
myquery4.Params[2].AsString := combobox3.Text;  
myquery4.Params[3].AsString := edit1.Text;  
myquery4.open;  
  
end;
```

```
procedure Tinputdatapemicu.BitBtn3Click(Sender: TObject);
begin

myquery5.SQL.Clear;

myquery5.SQL.Add('update data_pemicu set tahun = :a1,wilayah = :a2,pemicu =
:a3,luas_areal = :a4 where nomor = :a5');

myquery5.Params[0].AsString := combobox1.Text;

myquery5.Params[1].AsString := combobox2.Text;

myquery5.Params[2].AsString := combobox3.Text;

myquery5.Params[3].AsString := edit1.Text;

myquery5.Params[4].AsString := myquery4.Fields[0].AsString;

myquery5.Execute;

Showmessage('Data telah diubah');

BitBtn1Click(Sender);

FormCreate(Sender);

end;

procedure Tinputdatapemicu.BitBtn4Click(Sender: TObject);
begin

myquery5.SQL.Clear;

myquery5.SQL.Add('delete from data_pemicu where nomor = :a1');

myquery5.Params[0].AsString := myquery4.Fields[0].AsString;

myquery5.Execute;
```

```
Showmessage('Data telah dihapus');

BitBtn1Click(Sender);

FormCreate(Sender);

end;

eunit UPerhitungan;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ComCtrls, DB, Grids, DBGrids, MemDS, DBAccess,
MyAccess, VirtualTable, ExtCtrls, TeEngine, Series, TeeProcs, Chart,
QRCtrls, QuickRpt;

type

TPerhitungan = class(TForm)

  MyQuery1: TMyQuery;

  MyQuery2: TMyQuery;

  DataSource1: TDataSource;

  VirtualTable1: TVirtualTable;

  VirtualTable2: TVirtualTable;

  Label5: TLabel;

  GroupBox2: TGroupBox;

  DBGrid2: TDBGrid;
```

```
MyQuery3: TMyQuery;  
MyQuery4: TMyQuery;  
MyQuery5: TMyQuery;  
Button2: TButton;  
Label2: TLabel;  
Edit1: TEdit;  
Button1: TButton;  
Button3: TButton;  
MyQuery6: TMyQuery;  
QuickRep1: TQuickRep;  
TitleBand1: TQRBand;  
QRLLabel1: TQRLLabel;  
QRShape1: TQRShape;  
DetailBand1: TQRBand;  
QRDBText1: TQRDBText;  
QRDBText2: TQRDBText;  
QRDBText3: TQRDBText;  
QRDBText4: TQRDBText;  
QRDBText5: TQRDBText;  
GroupBox1: TGroupBox;  
DBGrid1: TDBGrid;  
DataSource2: TDataSource;  
DataSource3: TDataSource;  
ComboBox1: TComboBox;  
Label1: TLabel;
```

```
GroupBox3: TGroupBox;
DBGrid3: TDBGrid;
MyQuery7: TMyQuery;
Button4: TButton;
Chart1: TChart;
Series1: TLineSeries;
Series2: TLineSeries;
GroupBox4: TGroupBox;
DBGrid4: TDBGrid;
MyQuery8: TMyQuery;
DataSource4: TDataSource;
Chart2: TChart;
Series3: TLineSeries;
Button5: TButton;
Memo9: TMemo;
Memo1: TMemo;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
```

```
private
{ Private declarations }

public
{ Public declarations }

end;

var
luar : array[1..10] of real;
kromosom : array[1..10,1..4] of real;
evkrom : array[1..10] of real;
fitnes : array[1..10] of real;
pi : array[1..10] of real;
c : array[1..10] of real;
rolet : array[1..10] of integer;
krombaru : array[1..10,1..4] of real;
r : array[1..10] of real;
crossover : array[1..10] of integer;
ov : array[1..10] of integer;
randmut : array[1..4] of integer;

k,w : integer;
tom,lu : real;
a1,a2,a3,totalfit:real;
Perhitungan: TPerhitungan;
```

implementation

uses UTNK;

{\$R *.dfm}

procedure TPerhitungan.FormClose(Sender: TObject;

var Action: TCloseAction);

begin

action := cafree;

tnk.Perhitungan1.Enabled := true;

end;

procedure TPerhitungan.Button2Click(Sender: TObject);

var rand,rand1 : single;

at,bt,thn,i,j,y,z,iterasi,cak,aw,ak : integer;

ar,rab : real;

begin

thn := 2013;

// randomize;

w := strtoint(edit1.Text);

virtualtable2.First;

myquery6.SQL.Clear;

```
myquery6.SQL.Add('TRUNCATE TABLE krombaik');

myquery6.Execute;

myquery6.SQL.Clear;

myquery6.SQL.Add('TRUNCATE TABLE generasi');

myquery6.Execute;

memo1.Clear;

i := 1;

while not virtualtable2.Eof do

begin

for j := 1 to 4 do

memo1.Lines.Add(floattostr(virtualtable2.Fields[j].AsFloat));

inc(i);

virtualtable2.Next;

end;

z := 0;

for i := 1 to 10 do

begin

for j := 1 to 4 do

begin

kromosom[i,j] := strtofloat(memo1.Lines[z]);

z:=z+1;
```

```
end;  
end;  
  
aw := 1;  
ak := 100;  
  
for iterasi := 1 to w do  
begin  
//Evaluasi kromosom  
for i := 1 to 10 do  
begin  
tom := 0;  
for j := 1 to 4 do  
begin  
if j = 4 then  
tom := tom - kromosom[i,j]  
else  
tom := tom + kromosom[i,j];  
end;  
evkrom[i] := (198629/luar[i])-1;  
end;  
  
totalfit := 0;  
//seleksi kromosom  
for i := 1 to 10 do
```

```

begin

fitnes[i] := 1/(evkrom[i]+1);

totalfit := totalfit + fitnes[i];

end;

//menghitung P[i]

for i := 1 to 10 do

begin

pi[i] := fitnes[i]/totalfit;

end;

//menghitung komulatif probabilitas

c[1] := pi[1];

c[2] := pi[1]+pi[2];

c[3] := pi[1]+pi[2]+pi[3];

c[4] := pi[1]+pi[2]+pi[3]+pi[4];

c[5] := pi[1]+pi[2]+pi[3]+pi[4]+pi[5];

c[6] := pi[1]+pi[2]+pi[3]+pi[4]+pi[5]+pi[6];

c[7] := pi[1]+pi[2]+pi[3]+pi[4]+pi[5]+pi[6]+pi[7];

c[8] := pi[1]+pi[2]+pi[3]+pi[4]+pi[5]+pi[6]+pi[7]+pi[8];

c[9] := pi[1]+pi[2]+pi[3]+pi[4]+pi[5]+pi[6]+pi[7]+pi[8]+pi[9];

c[10] := pi[1]+pi[2]+pi[3]+pi[4]+pi[5]+pi[6]+pi[7]+pi[8]+pi[9]+pi[10];

//pemilihan induk dengan roulette

```

```
for i := 1 to 10 do
begin
rand := random;
for j := 1 to 10 do
begin
if rand < c[j] then
begin
rolet[i] := j;
break;
end;
end;
```

```
//menyusun kromosom baru hasil seleksi
for i := 1 to 10 do
begin
for j := 1 to 4 do
begin
k := rolet[i];
krombaru[i,j] := kromosom[k,j];
end;
end;
```

```

//crossover

for i := 1 to 10 do
begin
rand := random;
r[i] := rand;
end;

j:=1;

for i := 1 to 10 do
begin
if r[i] < 0.2 then
begin
crossov[j] := i;
j:=j+1;
end;
end;

//mengetahui banyaknya kromosom yang akan di crossover

y:=0;

for i := 1 to 10 do
begin
if crossov[i] <> 0 then
begin
y:= y+1;

```

```

end;

end;

//cut-point

z:=1;

for i := y downto 1 do

begin

rand:= 1 + random(4);

for j := 1 to 4 do

begin

if j<rand then

krombaru[z,j] := krombaru[i,j];

end;

z:=z+1;

end;

//mutasi

for z := 1 to 4 do

begin

y:=0;

cak := 1 + Random(40);

randmut[z] := cak;

for i := 1 to 10 do

```

```

begin

rand1 := aw+Random(ak);

for j := 1 to 4 do

begin

y:= y+1;

if cak = y then

krombaru[i,j] := rand1;

end;

end;

//update kromosom;

for i := 1 to 10 do

begin

for j := 1 to 4 do

begin

kromosom[i,j] := krombaru[i,j];

end;

end;

tom := 0;

for i := 1 to 10 do

begin

tom := tom + evkrom[i];

```

```

end;

myquery5.SQL.Clear;

myquery5.SQL.Add('insert
krombaik(tahun,kebakaran,perambahan,ilegal,reboisasi,total,k1)
values(:a1,:a2,:a3,:a4,:a5,:a6,:a7)');

myquery5.Params[0].AsString := floattosrt(tom/10);

for j := 1 to 4 do

myquery5.Params[j].AsString := "";

myquery5.Params[5].AsString := "";

myquery5.Params[6].AsString := "";

myquery5.Execute;

rab := 0;

for i := 1 to 10 do

begin

ar := 0;

myquery5.SQL.Clear;

myquery5.SQL.Add('insert
krombaik(tahun,kebakaran,perambahan,ilegal,reboisasi,total,k1)
values(:a1,:a2,:a3,:a4,:a5,:a6,:a7)');

myquery5.Params[0].AsString := 'G-' + inttostr(iterasi);

for j := 1 to 4 do

begin

myquery5.Params[j].AsString := floattosrt(krombaru[i,j]);

if j = 4 then

ar := ar - krombaru[i,j]

```

```

else
ar := ar + krombaru[i,j];
end;

rab := rab + ar;

myquery5.Params[5].AsString := floattosrt(rab);

myquery5.Params[6].AsString := "";

myquery5.Execute;

if i = 10 then
begin
myquery5.SQL.Clear;

myquery5.SQL.Add('insert
generasi(tahun,kebakaran,perambahan,ilegal,reboisasi,total,k1)
values(:a1,:a2,:a3,:a4,:a5,:a6,:a7)');

myquery5.Params[0].AsString := inttostr(thn);

for j := 1 to 4 do
begin
myquery5.Params[j].AsString := floattosrt(krombaru[i,j]);

if j = 4 then
ar := ar - krombaru[10,j]

else
ar := ar + krombaru[10,j];

end;

myquery5.Params[5].AsString := floattosrt(rab);

myquery5.Params[6].AsString := "";

myquery5.Execute;

end;

```

```
end;

myquery6.SQL.Clear;

myquery6.SQL.Add('select * from krombaik order by nomor asc');

myquery6.Open;

if rab > 198000 then

break;

inc(thn);

aw := aw + 25;

ak := ak + 75;

// akhir iterasi

end;

end;

procedure TPerhitungan.FormShow(Sender: TObject);

var ar,rab : real;

treng : integer;

begin

myquery1.SQL.Clear;
```

```
myquery1.SQL.Add('select distinct(tahun) from data_pemicu order by tahun desc');

myquery1.Open;

rab:=0;

virtualtable2.Clear;

myquery1.First;

while not myquery1.Eof do

begin

myquery2.SQL.Clear;

myquery2.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2');

myquery2.Params[0].AsString := myquery1.Fields[0].AsString;

myquery2.Params[1].AsString := 'Kebakaran Hutan';

myquery2.Open;

myquery3.SQL.Clear;

myquery3.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2');

myquery3.Params[0].AsString := myquery1.Fields[0].AsString;

myquery3.Params[1].AsString := 'Perambahan Hutan';

myquery3.Open;

myquery4.SQL.Clear;

myquery4.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2');

myquery4.Params[0].AsString := myquery1.Fields[0].AsString;

myquery4.Params[1].AsString := 'Illegal Logging';
```

```

myquery4.Open;

myquery5.SQL.Clear;

myquery5.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2');

myquery5.Params[0].AsString := myquery1.Fields[0].AsString;

myquery5.Params[1].AsString := 'Penanaman Kembali';

myquery5.Open;

virtualtable2.Insert;

virtualtable2.Fields[0].AsString := myquery1.Fields[0].AsString;

virtualtable2.Fields[1].AsString := format('%. *f',[1,strtofloat(myquery2.Fields[0].AsString))];

virtualtable2.Fields[2].AsString := format('%. *f',[1,strtofloat(myquery3.Fields[0].AsString))];

virtualtable2.Fields[3].AsString := format('%. *f',[1,strtofloat(myquery4.Fields[0].AsString))];

virtualtable2.Fields[4].AsString := format('%. *f',[1,strtofloat(myquery5.Fields[0].AsString))];

virtualtable2.Post;

myquery1.Next;

end;

treng := 1;

virtualtable2.First;

while not virtualtable2.Eof do

begin

ar := 0;

```

```
ar := (strtofloat(virtualtable2.Fields[1].AsString)+strtofloat(virtualtable2.Fields[2].AsString)+strtofloat(virtualtable2.Fields[3].AsString))-strtofloat(virtualtable2.Fields[4].AsString);

rab := rab + ar;

virtualtable2.Edit;

virtualtable2.Fields[5].AsString := floattosstr(rab);

virtualtable2.Post;

luar[treng] := rab;

inc(treng);

virtualtable2.Next;

end;

myquery1.SQL.Clear;

myquery1.SQL.Add('select distinct(wilayah) from data_pemicu order by wilayah asc');

myquery1.Open;

while not myquery1.Eof do

begin

combobox1.Items.Add(myquery1.Fields[0].AsString);

myquery1.Next;

end;

end;
```

```
procedure TPerhitungan.Button1Click(Sender: TObject);
begin
  groupbox4.Visible := true;
  myquery8.SQL.Clear;
  myquery8.SQL.Add('select * from generasi order by nomor asc');
  myquery8.Open;

  while not myquery8.Eof do
    begin
      Chart2.Series[0].Add(strtofloat(myquery8.Fields[6].AsString),
myquery8.Fields[1].AsString);
      myquery8.Next;
    end;
  end;

procedure TPerhitungan.ComboBox1Change(Sender: TObject);
var ar,rab : real;
begin
  if combobox1.Text = 'Wilayah I Suka Rahmat' then
    begin
      rab := 0;
      myquery1.SQL.Clear;
      myquery1.SQL.Add('select distinct(tahun) from data_pemicu where wilayah = :a1 order
by tahun desc');
      myquery1.Params[0].AsString := combobox1.Text;
```

```
myquery1.Open;

virtualtable2.Clear;

while not myquery1.Eof do

begin

myquery2.SQL.Clear;

myquery2.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery2.Params[0].AsString := myquery1.Fields[0].AsString;

myquery2.Params[1].AsString := 'Kebakaran Hutan';

myquery2.Params[2].AsString := combobox1.Text;

myquery2.Open;

myquery3.SQL.Clear;

myquery3.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery3.Params[0].AsString := myquery1.Fields[0].AsString;

myquery3.Params[1].AsString := 'Perambahan Hutan';

myquery3.Params[2].AsString := combobox1.Text;

myquery3.Open;

myquery4.SQL.Clear;

myquery4.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery4.Params[0].AsString := myquery1.Fields[0].AsString;

myquery4.Params[1].AsString := 'Illegal Logging';

myquery4.Params[2].AsString := combobox1.Text;
```

```

myquery4.Open;

myquery5.SQL.Clear;

myquery5.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery5.Params[0].AsString := myquery1.Fields[0].AsString;

myquery5.Params[1].AsString := 'Penanaman Kembali';

myquery5.Params[2].AsString := combobox1.Text;

myquery5.Open;

virtualtable2.Insert;

virtualtable2.Fields[0].AsString := myquery1.Fields[0].AsString;

virtualtable2.Fields[1].AsString := format('%.*f',[1,strtofloat(myquery2.Fields[0].AsString)]);

virtualtable2.Fields[2].AsString := format('%.*f',[1,strtofloat(myquery3.Fields[0].AsString)]);

virtualtable2.Fields[3].AsString := format('%.*f',[1,strtofloat(myquery4.Fields[0].AsString)]);

virtualtable2.Fields[4].AsString := format('%.*f',[1,strtofloat(myquery5.Fields[0].AsString)]);

virtualtable2.Post;

myquery1.Next;

end;

virtualtable2.First;

while not virtualtable2.Eof do

```

```

begin

ar := 0;

ar
:=
(strtofloat(virtualtable2.Fields[1].AsString)+strtofloat(virtualtable2.Fields[2].AsString)+strto
float(virtualtable2.Fields[3].AsString))-strtofloat(virtualtable2.Fields[4].AsString);

rab := rab + ar;

virtualtable2.Edit;

virtualtable2.Fields[5].AsString := floattostr(rab);

virtualtable2.Post;

virtualtable2.Next;

end;

end;

if combobox1.Text = 'Wilayah II Sangatta' then

begin

rab := 0;

myquery1.SQL.Clear;

myquery1.SQL.Add('select distinct(tahun) from data_pemicu where wilayah = :a1 order
by tahun desc');

myquery1.Params[0].AsString := combobox1.Text;

myquery1.Open;

virtualtable2.Clear;

while not myquery1.Eof do

begin

myquery2.SQL.Clear;

```

```
myquery2.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery2.Params[0].AsString := myquery1.Fields[0].AsString;

myquery2.Params[1].AsString := 'Kebakaran Hutan';

myquery2.Params[2].AsString := combobox1.Text;

myquery2.Open;

myquery3.SQL.Clear;

myquery3.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery3.Params[0].AsString := myquery1.Fields[0].AsString;

myquery3.Params[1].AsString := 'Perambahan Hutan';

myquery3.Params[2].AsString := combobox1.Text;

myquery3.Open;

myquery4.SQL.Clear;

myquery4.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery4.Params[0].AsString := myquery1.Fields[0].AsString;

myquery4.Params[1].AsString := 'Illegal Logging';

myquery4.Params[2].AsString := combobox1.Text;

myquery4.Open;

myquery5.SQL.Clear;

myquery5.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery5.Params[0].AsString := myquery1.Fields[0].AsString;
```

```

myquery5.Params[1].AsString := 'Penanaman Kembali';

myquery5.Params[2].AsString := combobox1.Text;

myquery5.Open;

virtualtable2.Insert;

virtualtable2.Fields[0].AsString := myquery1.Fields[0].AsString;

virtualtable2.Fields[1].AsString := format('%.*f',[1,strtofloat(myquery2.Fields[0].AsString)]);

virtualtable2.Fields[2].AsString := format('%.*f',[1,strtofloat(myquery3.Fields[0].AsString)]);

virtualtable2.Fields[3].AsString := format('%.*f',[1,strtofloat(myquery4.Fields[0].AsString)]);

virtualtable2.Fields[4].AsString := format('%.*f',[1,strtofloat(myquery5.Fields[0].AsString)]);

virtualtable2.Post;

myquery1.Next;

end;

virtualtable2.First;

while not virtualtable2.Eof do

begin

ar := 0;

ar := (strtofloat(virtualtable2.Fields[1].AsString)+strtofloat(virtualtable2.Fields[2].AsString)+strtofloat(virtualtable2.Fields[3].AsString))-strtofloat(virtualtable2.Fields[4].AsString);

rab := rab + ar;

virtualtable2.Edit;

virtualtable2.Fields[5].AsString := floattos(rab);

```

```
virtualtable2.Post;

virtualtable2.Next;

end;

end;

if combobox1.Text = 'Wilayah III Manamang' then

begin

rab := 0;

myquery1.SQL.Clear;

myquery1.SQL.Add('select distinct(tahun) from data_pemicu where wilayah = :a1 order
by tahun desc');

myquery1.Params[0].AsString := combobox1.Text;

myquery1.Open;

virtualtable2.Clear;

while not myquery1.Eof do

begin

myquery2.SQL.Clear;

myquery2.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery2.Params[0].AsString := myquery1.Fields[0].AsString;

myquery2.Params[1].AsString := 'Kebakaran Hutan';

myquery2.Params[2].AsString := combobox1.Text;

myquery2.Open;
```

```
myquery3.SQL.Clear;

myquery3.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery3.Params[0].AsString := myquery1.Fields[0].AsString;

myquery3.Params[1].AsString := 'Perambahan Hutan';

myquery3.Params[2].AsString := combobox1.Text;

myquery3.Open;

myquery4.SQL.Clear;

myquery4.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery4.Params[0].AsString := myquery1.Fields[0].AsString;

myquery4.Params[1].AsString := 'Illegal Logging';

myquery4.Params[2].AsString := combobox1.Text;

myquery4.Open;

myquery5.SQL.Clear;

myquery5.SQL.Add('select sum(luas_areal) from data_pemicu where tahun = :a1 and
pemicu = :a2 and wilayah = :a3');

myquery5.Params[0].AsString := myquery1.Fields[0].AsString;

myquery5.Params[1].AsString := 'Penanaman Kembali';

myquery5.Params[2].AsString := combobox1.Text;

myquery5.Open;

virtualtable2.Insert;

virtualtable2.Fields[0].AsString := myquery1.Fields[0].AsString;
```

```

    virtualtable2.Fields[1].AsString          := 
format('%.*f',[1,strtofloat(myquery2.Fields[0].AsString)]);

    virtualtable2.Fields[2].AsString          := 
format('%.*f',[1,strtofloat(myquery3.Fields[0].AsString)]);

    virtualtable2.Fields[3].AsString          := 
format('%.*f',[1,strtofloat(myquery4.Fields[0].AsString)]);

    virtualtable2.Fields[4].AsString          := 
format('%.*f',[1,strtofloat(myquery5.Fields[0].AsString)]);

    virtualtable2.Post;

myquery1.Next;

end;

virtualtable2.First;

while not virtualtable2.Eof do

begin

ar := 0;

ar
:=
(strtofloat(virtualtable2.Fields[1].AsString)+strtofloat(virtualtable2.Fields[2].AsString)+strto
float(virtualtable2.Fields[3].AsString))-strtofloat(virtualtable2.Fields[4].AsString);

rab := rab + ar;

virtualtable2.Edit;

virtualtable2.Fields[5].AsString := floattosr(rab);

virtualtable2.Post;

virtualtable2.Next;

end;

end;

```

```
procedure TPerhitungan.Button3Click(Sender: TObject);
begin
  // groupbox3.Visible := true;
  virtualtable2.First;
  while not virtualtable2.Eof do
  begin
    myquery2.SQL.Clear;
    myquery2.SQL.Add('select count(*) from generasi where tahun = :a1');
    myquery2.Params[0].AsString := virtualtable2.Fields[0].AsString;
    myquery2.Open;

    if myquery2.Fields[0].AsInteger > 0 then
    begin
      myquery3.SQL.Clear;
      myquery3.SQL.Add('update generasi set aktual = :a1 where tahun = :a2');
      myquery3.Params[0].AsString := virtualtable2.Fields[5].AsString;
      myquery3.Params[1].AsString := virtualtable2.Fields[0].AsString;
      myquery3.Execute;
    end;
    virtualtable2.Next;
  end;
  myquery7.SQL.Clear;
  myquery7.SQL.Add('select * from generasi order by nomor asc');
  myquery7.Open;
```

```
series1.Clear;  
series2.Clear;  
  
while not myquery7.Eof do  
  
begin  
  
    Chart1.Series[0].Add(strtofloat(myquery7.Fields[7].AsString),  
myquery7.Fields[1].AsString);  
  
    Chart1.Series[1].Add(strtofloat(myquery7.Fields[6].AsString),  
myquery7.Fields[1].AsString);  
  
    myquery7.Next;  
  
end;  
  
end;  
  
procedure TPerhitungan.Button4Click(Sender: TObject);  
  
begin  
  
    groupbox3.Visible := false;  
  
end;  
  
procedure TPerhitungan.Button5Click(Sender: TObject);  
  
begin  
  
    groupbox4.Visible := False;  
  
end;  
  
procedure TPerhitungan.FormCreate(Sender: TObject);  
  
begin  
  
end;  
  
end.
```