

DAFTAR PUSTAKA

- Agustini, D. H. 2004. *Riset Operasional Konsep-Konsep Dasar*. Jakarta. Rineka Cipta.
- Anton, H. 2000. *Dasar-dasar Aljabar Linier*. Edisi Ketujuh. Jakarta. Erlangga
- Bronson, R. 1995. *Teori dan Soal-Soal Operations Research*. Jakarta Erlangga.
- Dimiyati, T dan Dimiyati, A. 2004. *Operation Research*. Edisi Bahasa Indonesia. Bandung. Sinar Baru Algensindo.
- Hillier, F. S dan Lieberman, G. J. 1990. *Introduction to Operations Research*. Fifth Edition. Singapura. Mc Graw Hill, Inc
- Jay, H dan Barry R. 2008. *Manajemen Operasi*. Edisi Ketujuh. Jakarta. Salemba Empat
- Jensen, P. A dan Bard, J. F. 2003. *Operation Research Models and Methods*. International Edition, John Wiley & Sons, Inc.
- Mathirajan. M dan Meenakshi. B. 2004. *Experimental Analysis of Some Variants of Vogel's Approximation Method*. Asia-Pacific Journal of Operation Research. Vol 21. No. 4. Hal. 447-462
- Nahmias, S. 2009. *Production and Operations Analysis*. Sixth Edition. Singapore. The McGraw-Hill Companies.
- Nasendi, B.D. 1985. *Program Linear Dan Variasinya*. Jakarta: PT. Gramedia.
- Prawirosentono, S. 2005. *Riset Operasi dan Ekonofisika*. Edisi Pertama. Jakarta. PT Bumi Aksara.
- Serdar, K dan Serkan, B. 2011. *An Improved Vogel's Approximation Method for the Transportation Problem*.

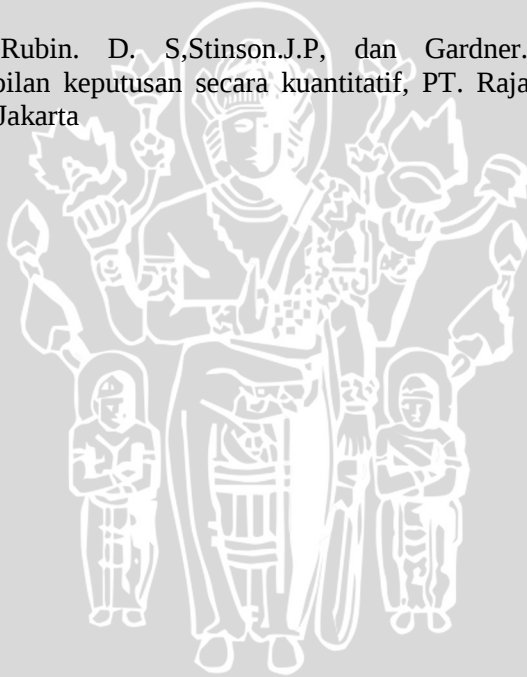
Association for Scientific Research. Vol 16. No. 2. Hal. 370-381.

Setiawan, E. 2011. Kamus Besar Bahasa Indonesia Offline Versi 1.3.
<http://pusatbahasa.kemdiknas.go.id/kbbi>. Tanggal akses: 21 juni 2011

Taha, H. A. 2003. *Operatin Research*. Seventh Edition. Pearson Education International.

Weiss, H. J dan Mark, E. G. 1989. *Production and Operations Management*. United States of America. Allyn and Bacon.

Levin. R.I, Rubin. D. S, Stinson.J.P, dan Gardner.E.S, 1993, Pengambilan keputusan secara kuantitatif, PT. Raja grafindo Persada, Jakarta



Lampiran 1

Perhitungan menggunakan *Vogel's Approximation Method* (VAM)

Dari	Ke	D ₁	D ₂	D ₃	D ₄	D ₅	Supply	Beda Kolom						
O ₁		46	74	9	28	99	922							
	2	120	136	-	664			19	19	19	37	37	37	53
O ₂		12	75	6	36	48	554							
	554	-	-	-	-	6		6	-	-	-	-	-	
O ₃		35	112	4	5	71	712							
	-	-	-	232	480	1		1	1	31	31	31	31	
O ₄		61	81	44	88	9	976							
	-	-	-	-	976	35		-	-	-	-	-	-	
O ₅		85	60	14	25	79	786							
	-	-	786	-	-	11		11	11	46	-	-	-	
Demand		556	120	922	232	2120	3950							
Beda Baris		23	14	2	20	39								
		23	14	2	20	23								
		11	14	5	20	8								
		11	14	5	-	8								
		11	38	5	-	28								
		11	-	5	-	28								
		11	-	-	-	28								

Hasil akhir yang diperoleh dengan menggunakan *Vogel's Approximation Method* (VAM) dalam 9 iterasi.

Total biaya distribusi (TC)

$$\begin{aligned} \text{TC} &= 2 \cdot 4600 + 120 \cdot 7400 + 136 \cdot 900 + 664 \cdot 9900 + 554 \cdot 1200 + \\ &\quad 232 \cdot 500 + 480 \cdot 7100 + 976 \cdot 900 + 786 \cdot 1400 \end{aligned}$$

$$= \text{Rp } 13.760.800,-$$

UNIVERSITAS BRAWIJAYA



Lampiran 2

Perhitungan menggunakan *Improved Vogel's Approximation Method* iterasi 1.

Ke Dari	D ₁	D ₂	D ₃	D ₄	D ₅	Supply	Beda kolom
O ₁	31	19	0	18	90	922	18
O ₂	0	23	0	29	42	554	0
O ₃	25	62	0	0	67	712	0
O ₄	46	26	35	78	0	976	26
O ₅	65	0	0	10	65	786	0
Demand	556	120	922	232	2120	3950	
Beda Baris	25	19	0	10	42		

Biaya pinalti tiap baris dan tiap kolom dicari terlebih dahulu seperti berikut

baris 1	selisih antara angka 0 dan 18 adalah 18
baris 2	selisih antara angka 0 dan 0 adalah 0
baris 3	selisih antara angka 0 dan 0 adalah 0
baris 4	selisih antara angka 0 dan 26 adalah 26
baris 5	selisih antara angka 0 dan 0 adalah 0
kolom 1	selisih antara angka 0 dan 25 adalah 25
kolom 2	selisih antara angka 0 dan 19 adalah 19
kolom 3	selisih antara angka 0 dan 0 adalah 0
kolom 4	selisih antara angka 0 dan 10 adalah 10
kolom 5	selisih antara angka 0 dan 42 adalah 42

Untuk iterasi pertama biaya pinalti terbesar terdapat pada kolom kelima yaitu 42. Dengan demikian dilakukan alokasi sebanyak mungkin pada sel dengan biaya terkecil pada kolom tersebut yaitu sel O_4D_5 . Pada sel ini akan dialokasikan sebanyak mungkin tanpa melanggar kapasitas pada baris dan kolom. Jadi, pada sel O_4D_5 dialokasikan sebanyak 976 boks dan sisa kapasitas pada kolom D_5 adalah 2120 boks dikurangi 976 boks yaitu sebesar 1144 boks. Karena kapasitas pada baris keempat telah dialokasikan sepenuhnya pada sel O_4D_5 maka sel lain pada baris yang sama tidak perlu diisi.

Lampiran 3

Perhitungan menggunakan *Improved Vogel's Approximation Method* iterasi 2.

Ke Dari	D ₁	D ₂	D ₃	D ₄	D ₅	Supply	Beda Kolom	
O ₁	31	19	0	18	90	922	18	18
O ₂	0 554	23	0	29	42	554	0	0
O ₃	25	62	0	0	67	712	0	0
O ₄	46 -	26	35	78	0 976	976	26	-
O ₅	65	0	0	10	65	786	0	0
Demand	556	120	922	232	2120	3950		
Beda Baris	25 25	19 19	0 0	10 10	42 23			

Biaya pinalti tiap baris dan tiap kolom dicari terlebih dahulu seperti berikut

baris 1	selisih antara angka 0 dan 18 adalah 18
baris 2	selisih antara angka 0 dan 0 adalah 0
baris 3	selisih antara angka 0 dan 0 adalah 0
baris 5	selisih antara angka 0 dan 0 adalah 0
kolom 1	selisih antara angka 0 dan 25 adalah 25
kolom 2	selisih antara angka 0 dan 19 adalah 19
kolom 3	selisih antara angka 0 dan 0 adalah 0
kolom 4	selisih antara angka 0 dan 10 adalah 10
kolom 5	selisih antara angka 0 dan 42 adalah 23

biaya pinalti pada baris keempat tidak perlu dicari. Seperti iterasi 1, biaya pinalti masing-masing kolom dan baris dicari. Didapatkan biaya pinalti terbesar terdapat pada kolom pertama yaitu 25 dan sel dengan biaya terkecil terletak pada O_2D_1 . Pada sel ini dialokasikan sebanyak 554 boks dan kapasitas pada kolom pertama bersisa 2 boks. Karena kapasitas pada baris kedua telah dialokasikan sepenuhnya pada sel O_2D_1 maka sel lain pada baris yang sama tidak dapat diisi.

Lampiran 4

Perhitungan menggunakan *Improved Vogel's Approximation Method* iterasi 3.

Ke Dari	D ₁	D ₂	D ₃	D ₄	D ₅	Supply	Beda Kolom		
O ₁	31	19	0	18	90	922	18	18	18
O ₂	0 554	23 -	0 -	29 -	42 -	554	0	0	-
O ₃	25	62	0	0	67	712	0	0	0
O ₄	46 -	26 -	35 -	78 -	0 976	976	26	-	-
O ₅	65 120	0 120	0	10	65	786	0	0	10
Demand	556	120	922	232	2120	3950			
Beda Baris	25	19	0	10	42				
	25	19	0	10	23				
	6	19	0	10	2				

Biaya pinalti tiap baris dan tiap kolom dicari terlebih dahulu seperti berikut

baris 1	selisih antara angka 0 dan 18 adalah 18
baris 3	selisih antara angka 0 dan 0 adalah 0
baris 5	selisih antara angka 0 dan 0 adalah 0
kolom 1	selisih antara angka 25 dan 31 adalah 6
kolom 2	selisih antara angka 0 dan 19 adalah 19
kolom 3	selisih antara angka 0 dan 0 adalah 0
kolom 4	selisih antara angka 0 dan 10 adalah 10
kolom 5	selisih antara angka 65 dan 67 adalah 2

biaya pinalti pada baris kedua tidak perlu dicari. Biaya pinalti terbesar terdapat pada kolom kedua yaitu sebesar 19. Sel dengan biaya terkecil terletak pada O_5D_2 . Pada sel ini dialokasikan sebanyak 120 boks dan sisa kapasitas pada baris kelima adalah 786 boks dikurangi 120 boks yaitu 666 boks. Karena kapasitas pada kolom kedua telah dialokasikan sepenuhnya pada sel O_5D_2 maka sel lain pada kolom yang sama tidak dapat diisi.



Lampiran 5

Perhitungan menggunakan *Improved Vogel's Approximation Method* iterasi 4.

Dari Ke	D ₁	D ₂	D ₃	D ₄	D ₅	Supply	Beda Kolom			
O ₁	31	19	0	18	90	922	18	18	18	18
O ₂	0	23	0	29	42	554	0	0	-	-
O ₃	25	62	0	0	67	712	0	0	0	0
O ₄	46	26	35	78	0	976	26	-	-	-
O ₅	65	0	0	10	65	786	0	0	10	10
Demand	556	120	922	232	2120	3950				
Beda Baris	25	19	0	10	42					
	25	19	0	10	23					
	6	19	0	10	2					
	6	-	0	10	2					

Biaya pinalti tiap baris dan tiap kolom dicari terlebih dahulu seperti berikut

baris 1	selisih antara angka 0 dan 18 adalah 18
baris 3	selisih antara angka 0 dan 0 adalah 0
baris 5	selisih antara angka 0 dan 10 adalah 10
kolom 1	selisih antara angka 25 dan 31 adalah 6
kolom 3	selisih antara angka 0 dan 0 adalah 0
kolom 4	selisih antara angka 0 dan 10 adalah 10
kolom 5	selisih antara angka 65 dan 67 adalah 2

biaya pinalti pada kolom keempat tidak perlu dicari. Baris pertama memiliki biaya pinalti terbesar yaitu 18 dan sel dengan biaya terkecil yaitu O_1D_3 . Pada sel ini dialokasikan sebanyak 922 boks. Karena kapasitas pada baris pertama dan kolom ketiga telah dialokasikan sepenuhnya pada sel O_1D_3 maka sel lain pada baris dan kolom yang sama tidak dapat diisi.



Lampiran 6

Perhitungan menggunakan *Improved Vogel's Approximation Method* iterasi 5.

Ke Dari	D ₁	D ₂	D ₃	D ₄	D ₅	Supply	Beda Kolom				
O ₁	31 -	19 -	0 922	18 -	90 -	922	18	18	18	18	1
O ₂	0 554	23 -	0 -	29 -	42 -	554	0	0	-	-	-
O ₃	25 2	62 -	0 -	0 232	67 478	712	0	0	0	0	25
O ₄	46 -	26 -	35 -	78 -	0 976	976	26	-	-	-	-
O ₅	65 -	0 120	0 -	10 -	65 666	786	0	0	10	10	10
Demand	556	120	922	232	2120	3950					
Beda Baris	25	19	0	10	42						
	25	19	0	10	23						
	6	19	0	10	2						
	6	-	0	10	2						
	6	-	0	10	2						

Biaya pinalti tiap baris dan tiap kolom dicari terlebih dahulu seperti berikut

baris 1	selisih antara angka 18 dan 19 adalah 1
baris 3	selisih antara angka 0 dan 25 adalah 25
baris 5	selisih antara angka 0 dan 10 adalah 10
kolom 1	selisih antara angka 25 dan 31 adalah 6
kolom 3	selisih antara angka 0 dan 0 adalah 0
kolom 4	selisih antara angka 0 dan 10 adalah 10
kolom 5	selisih antara angka 65 dan 67 adalah 2

biaya pinalti baris pertama dan kolom kolom ketiga tidak perlu dicari. Biaya pinalti terbesar terdapat pada baris ketiga sebesar 25 dan sel O_3D_4 adalah sel dengan biaya terkecil. Pada sel ini dialokasikan sebanyak 232 boks dan sisa kapasitas pada baris ketiga 712 boks dikurangi 232 boks yaitu 480 boks. Kapasitas pada kolom keempat dialokasikan sepenuhnya pada sel O_3D_4 sehingga sel lain pada kolom yang sama tidak dapat diisi.

Sel yang belum terisi yaitu sel O_3D_1 , O_3D_5 , O_5D_1 , dan O_5D_5 . Pada sel-sel tersebut dialokasikan sejumlah sisa kapasitas pada masing-masing kolom maupun baris. Untuk sel O_3D_1 dialokasikan sejumlah 2 boks sesuai dengan sisa kapasitas pada kolom pertama, untuk sel O_3D_5 dialokasikan sejumlah 478 boks yang didapatkan dari sisa kapasitas pada baris ketiga yaitu 480 boks dikurangi 2 boks, dan untuk sel O_5D_5 dialokasikan sebesar sisa kapasitas pada kolom kelima yaitu 666 boks.



Lampiran 7

Hasil alokasi optimal dengan menggunakan metode Simpleks

Dari Ke	D ₁	D ₂	D ₃	D ₄	D ₅	Supply
O ₁	46 -	74 -	9 992	28 -	99 -	922
O ₂	12 554	75 -	6 -	36 -	48 -	554
O ₃	35 2	100 -	4 -	5 -	71 710	712
O ₄	61 -	81 -	44 -	88 -	9 976	976
O ₅	85 -	60 120	14 -	25 232	79 434	786
Demand	556	120	922	232	2120	3950

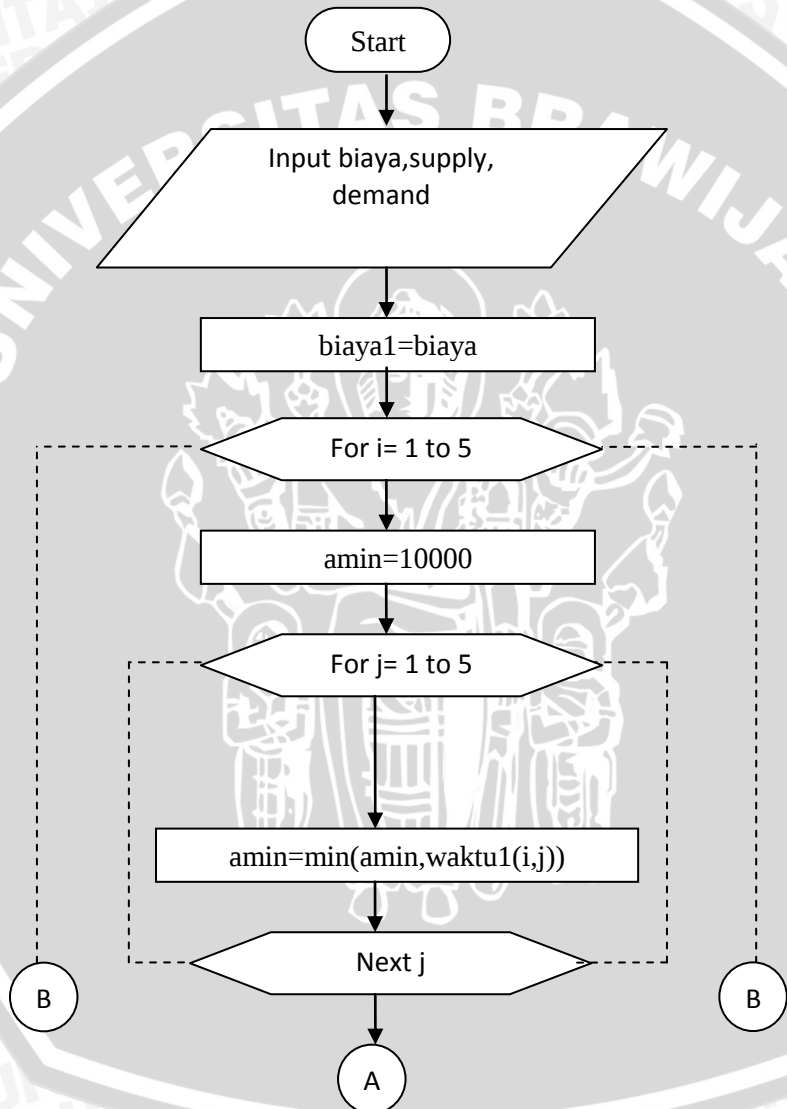
Hasil alokasi optimal dengan menggunakan metode Simpleks memerlukan lima iterasi dan total biaya distribusi sebesar Rp 11.871.200,-

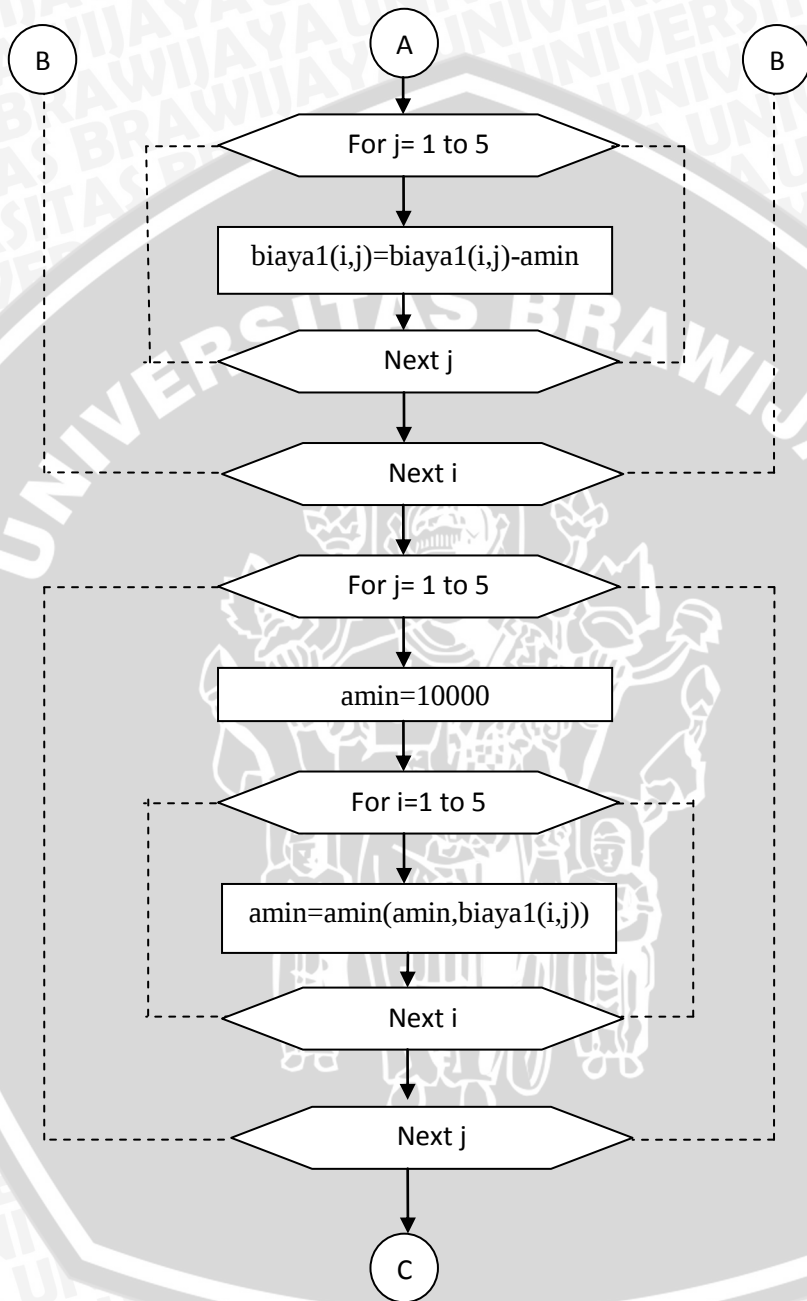
UNIVERSITAS BRAWIJAYA

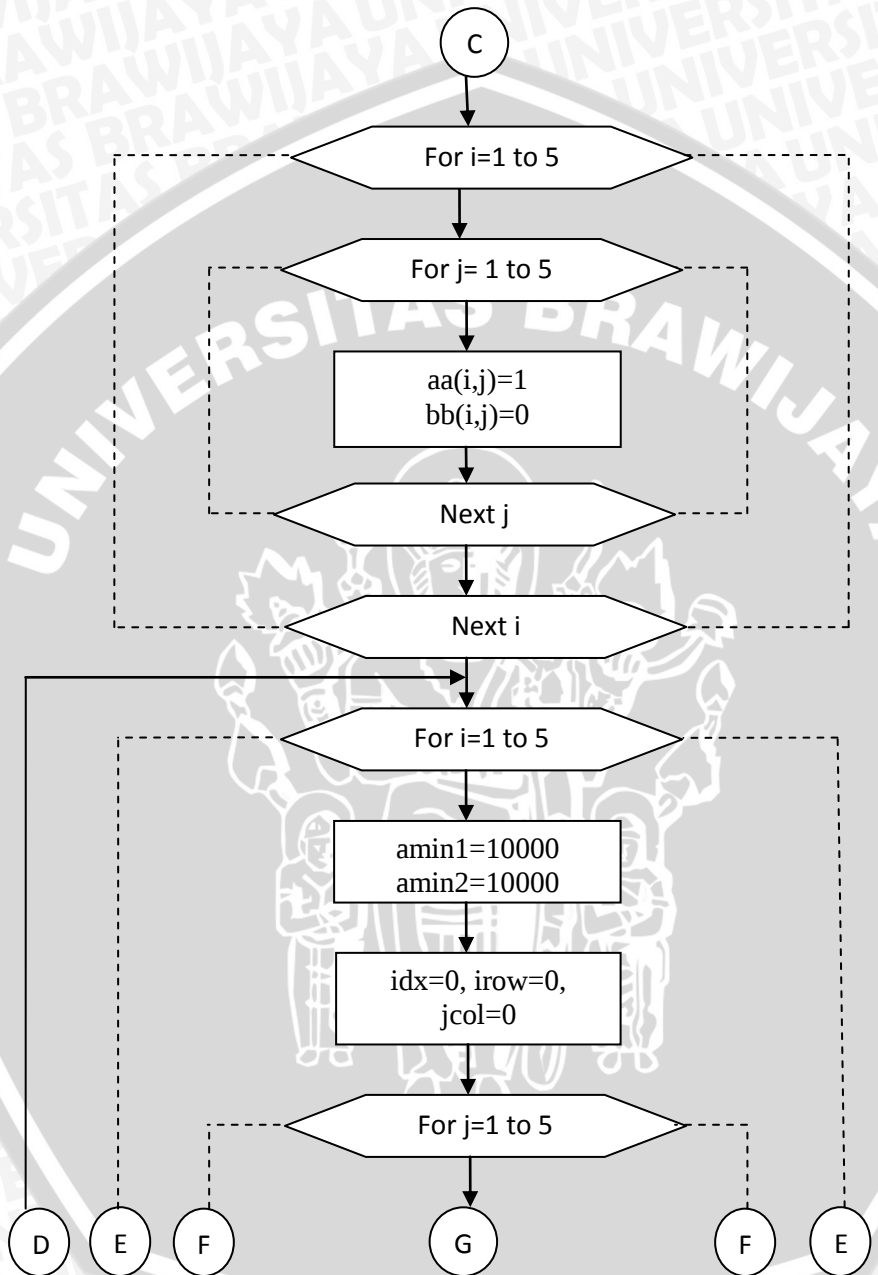


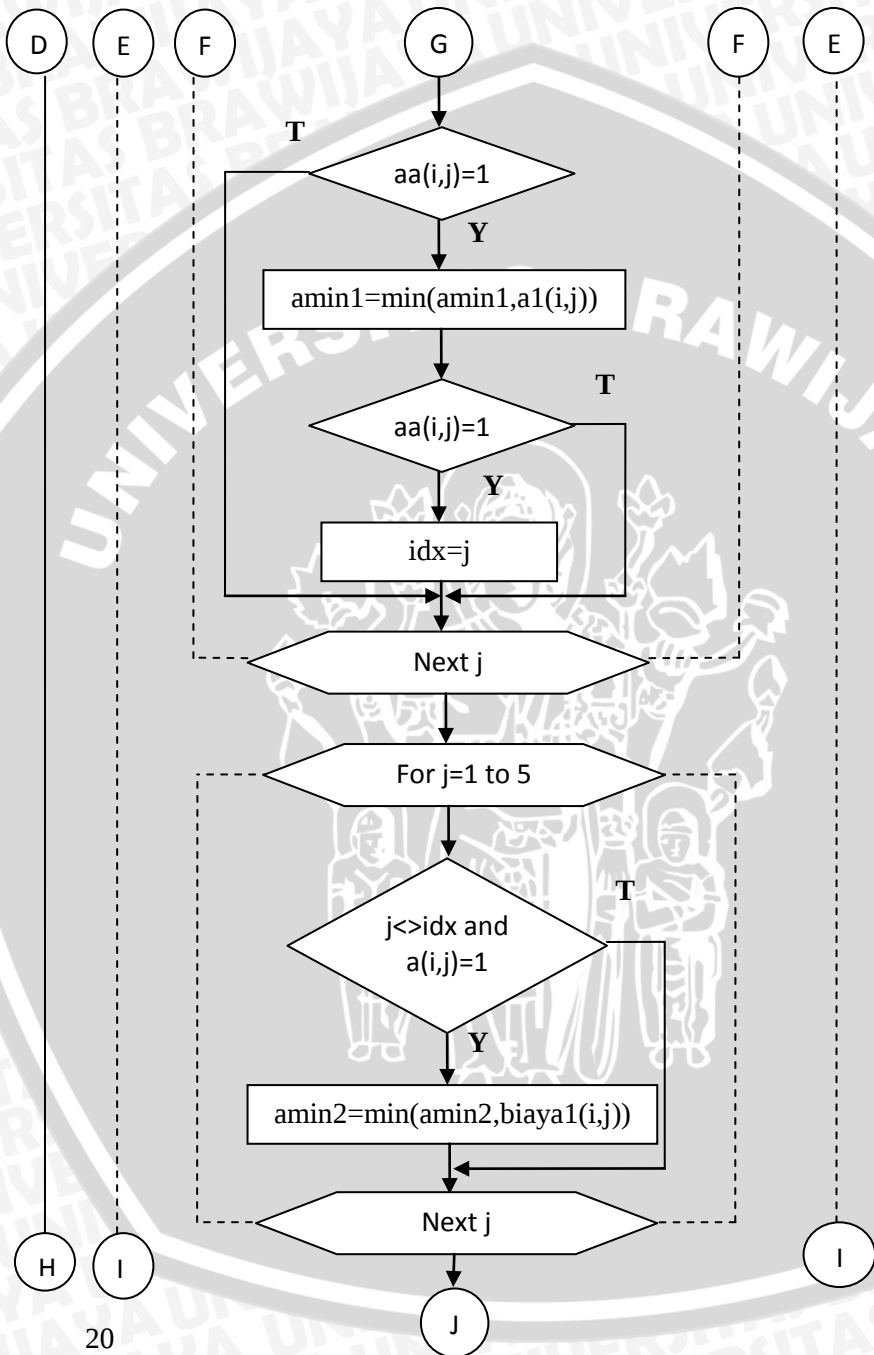
Lampiran 8

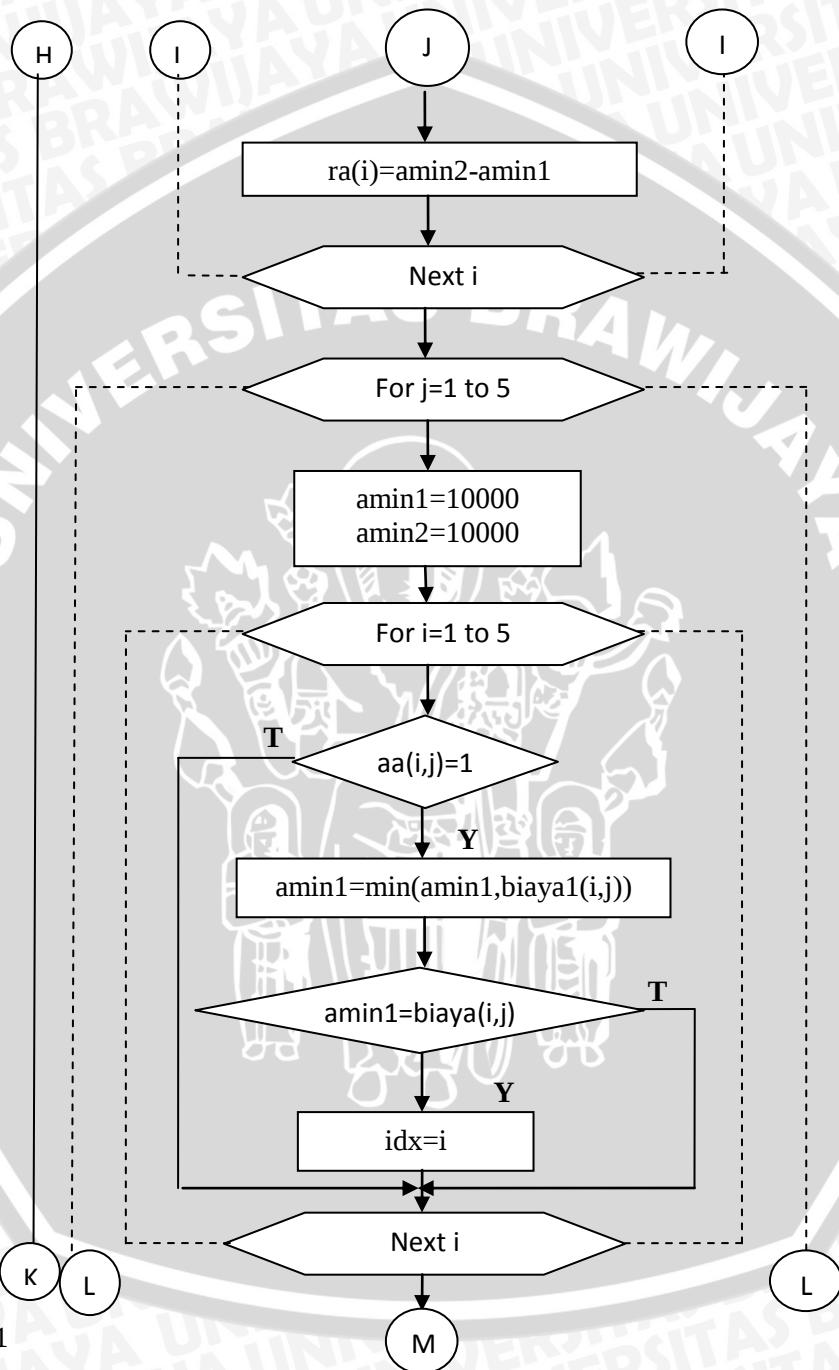
Flowchart dari program *Improved Vogel's Approximation Method* dengan menggunakan software Delphi 7.0.

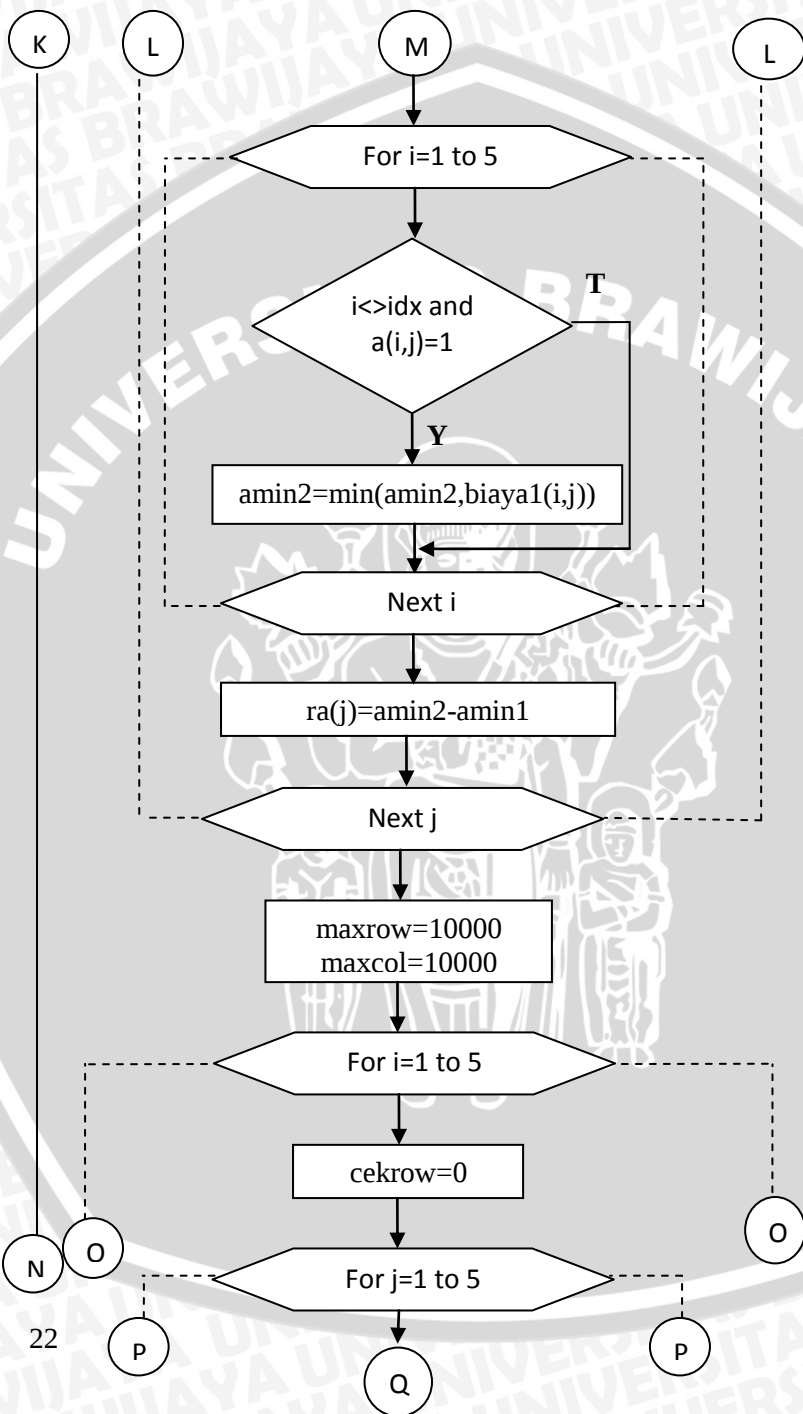


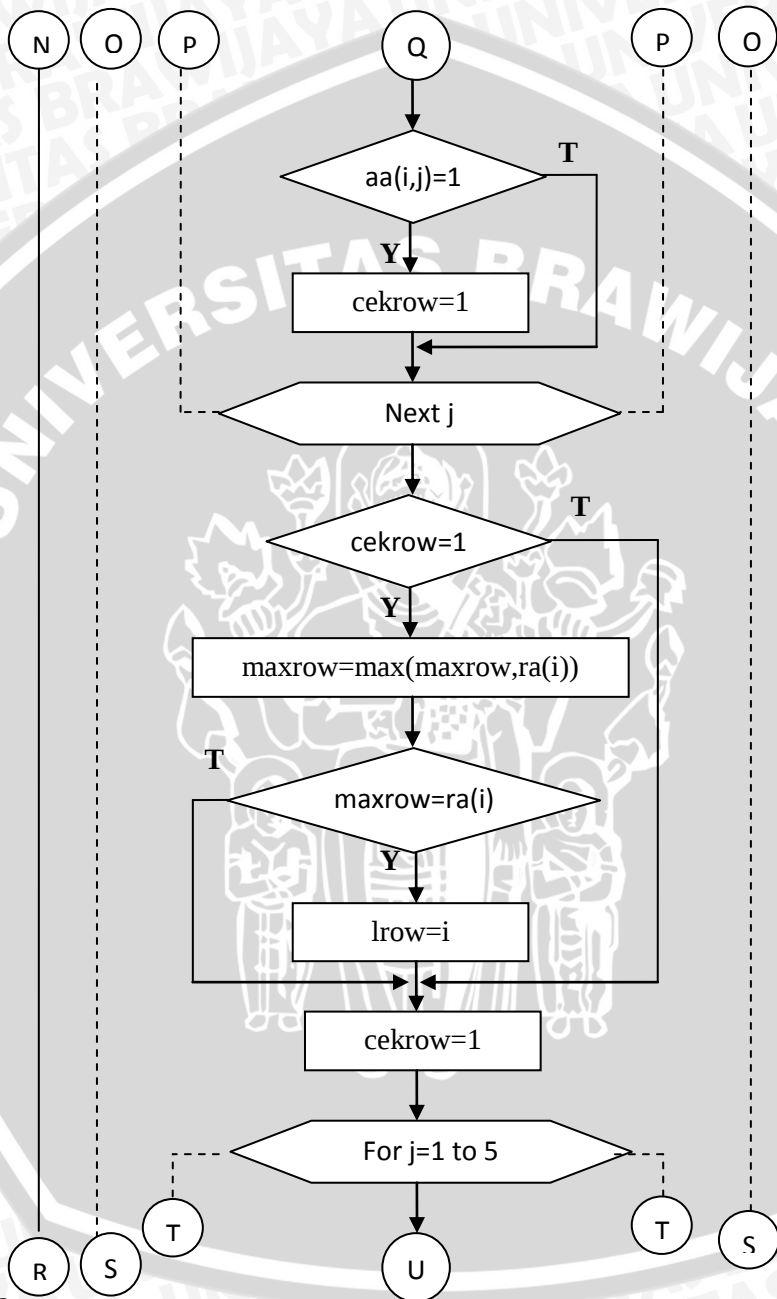


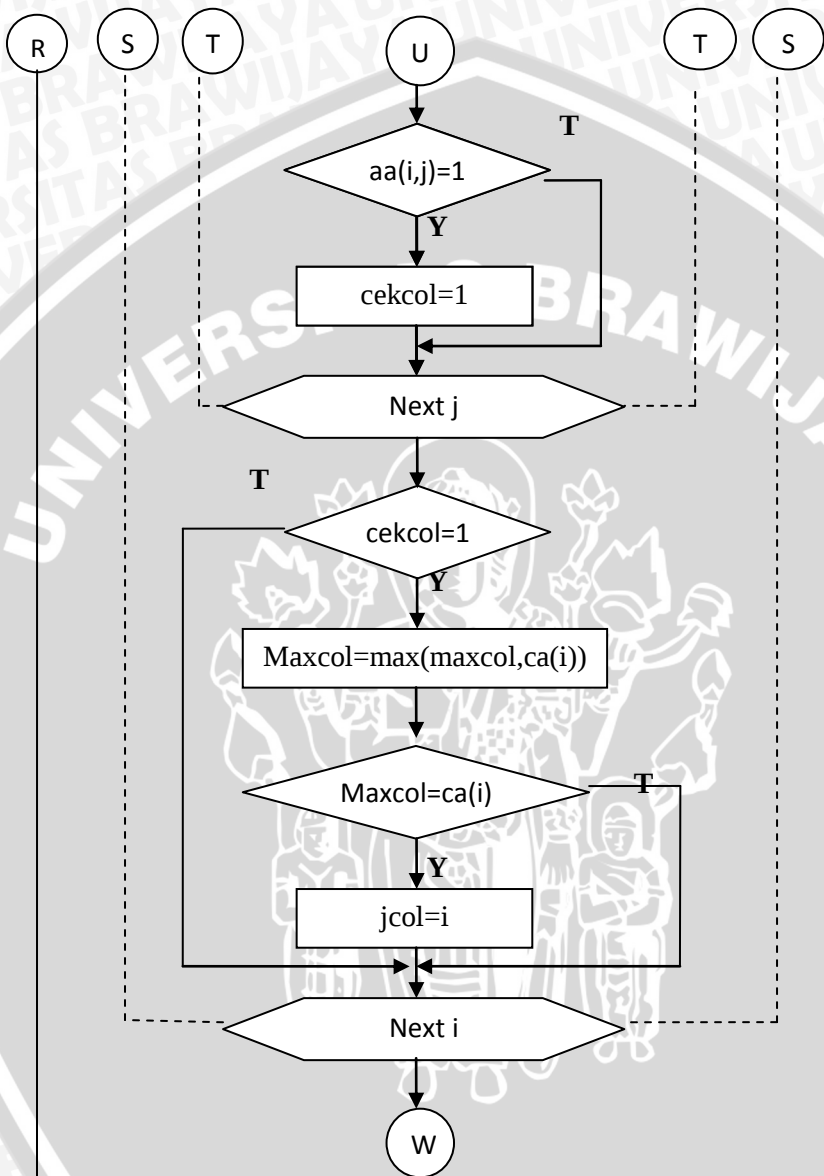


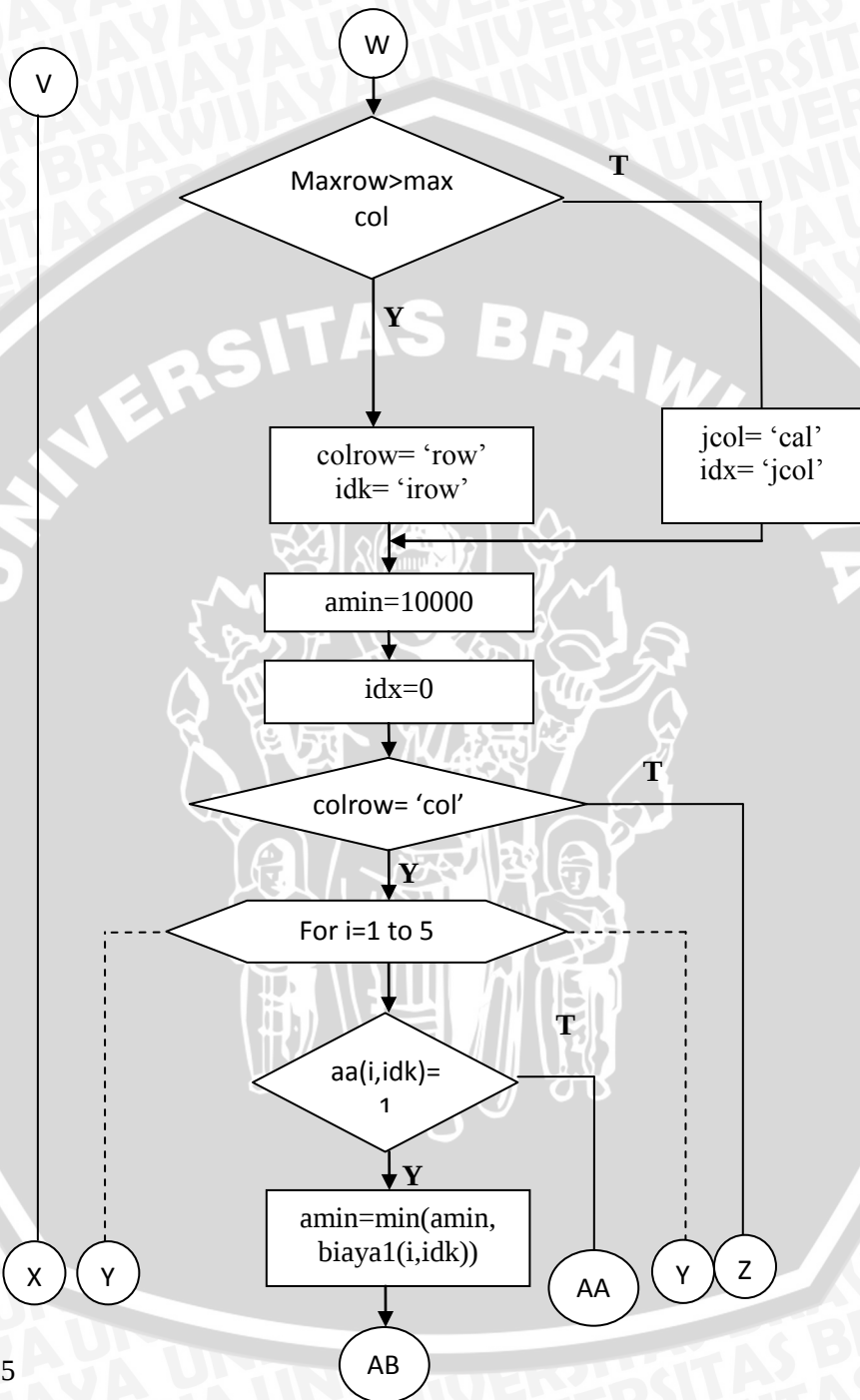


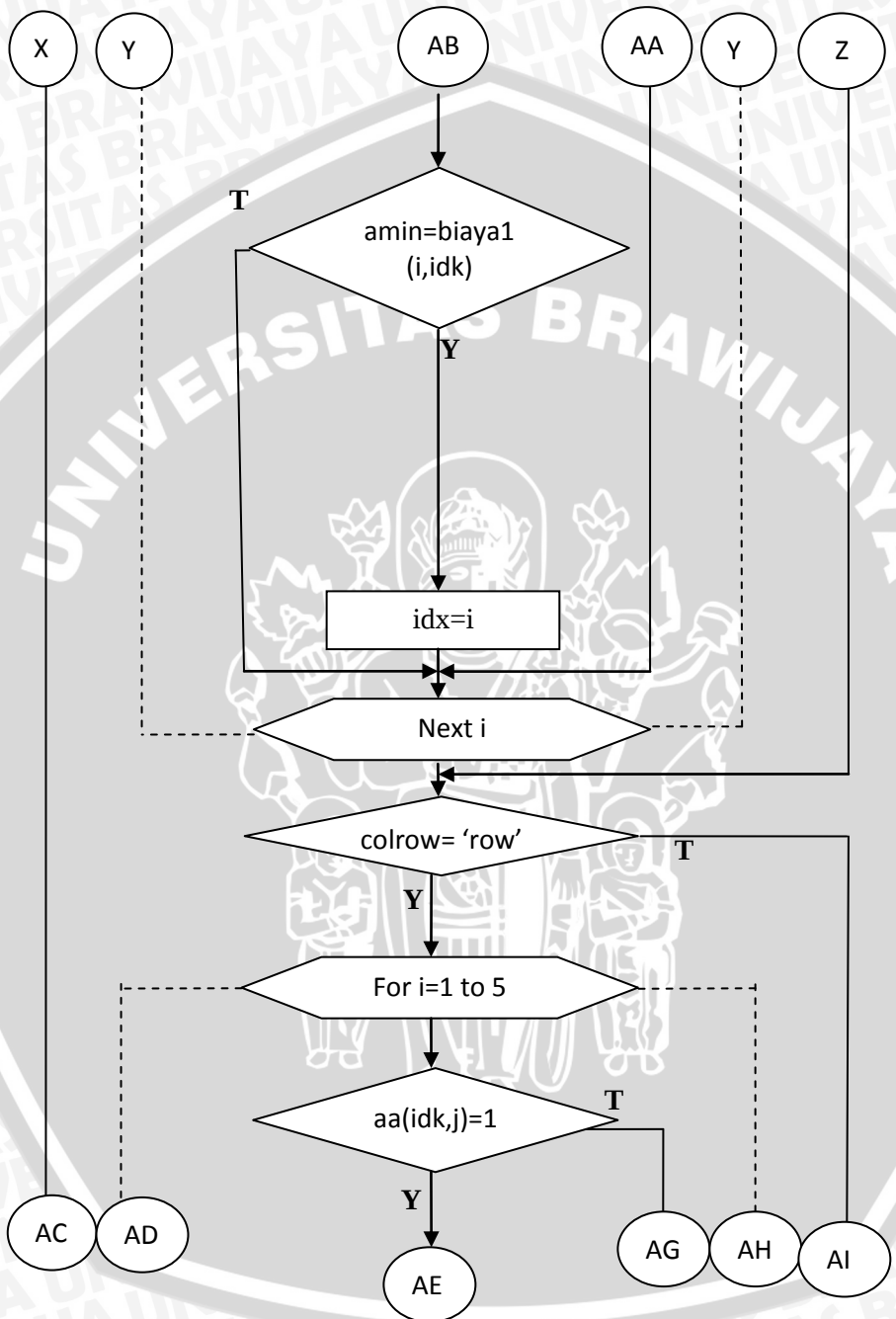


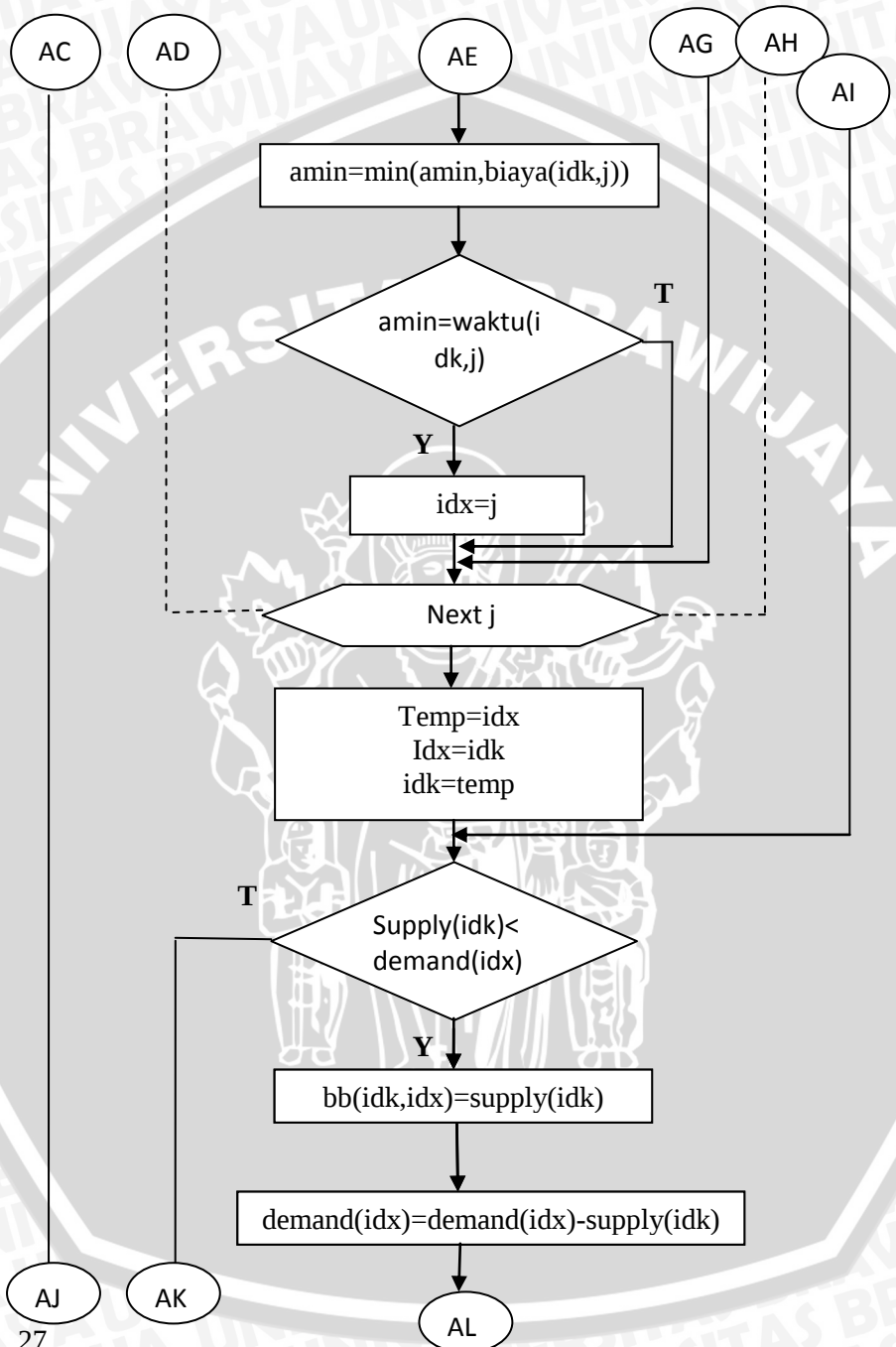


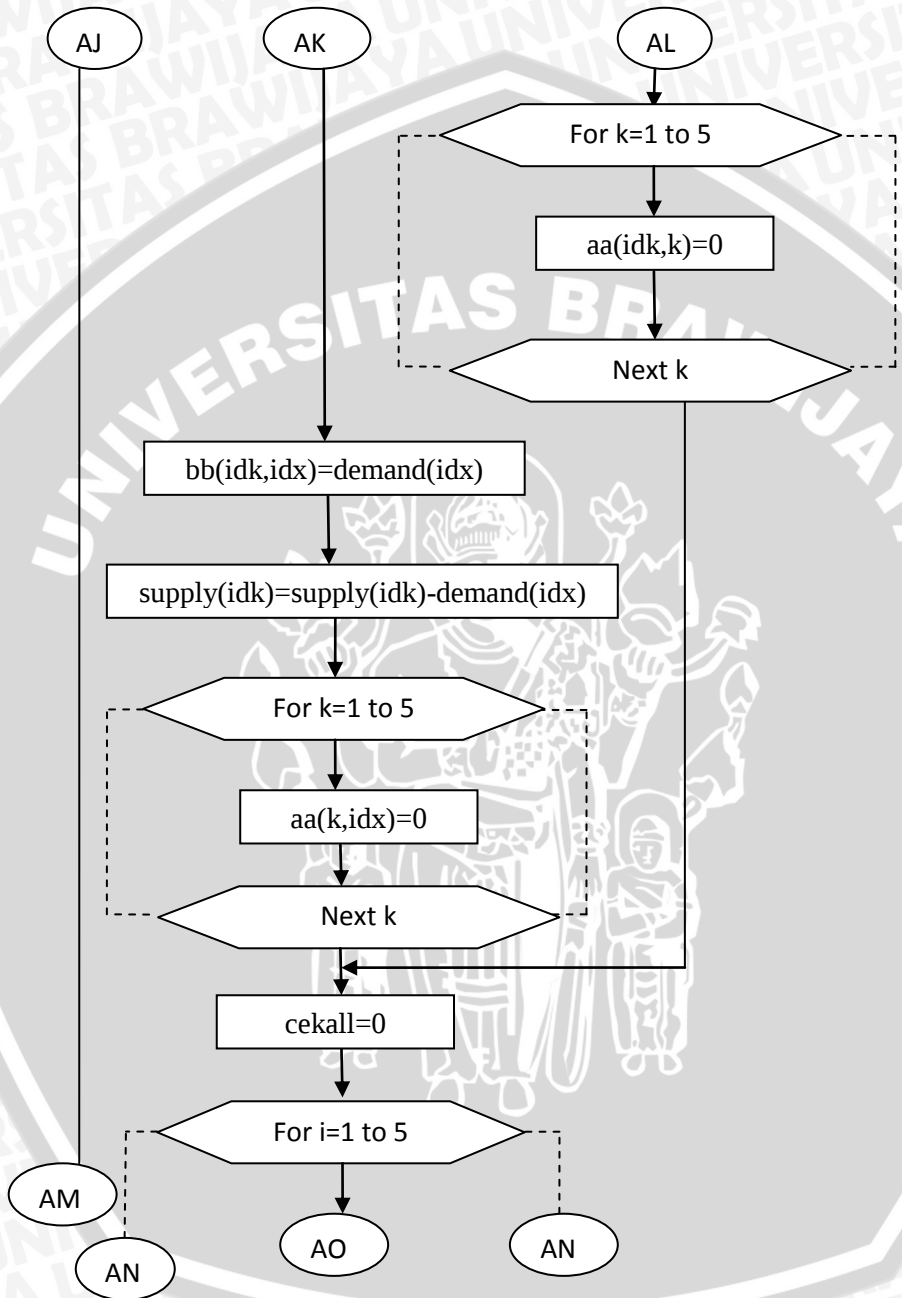


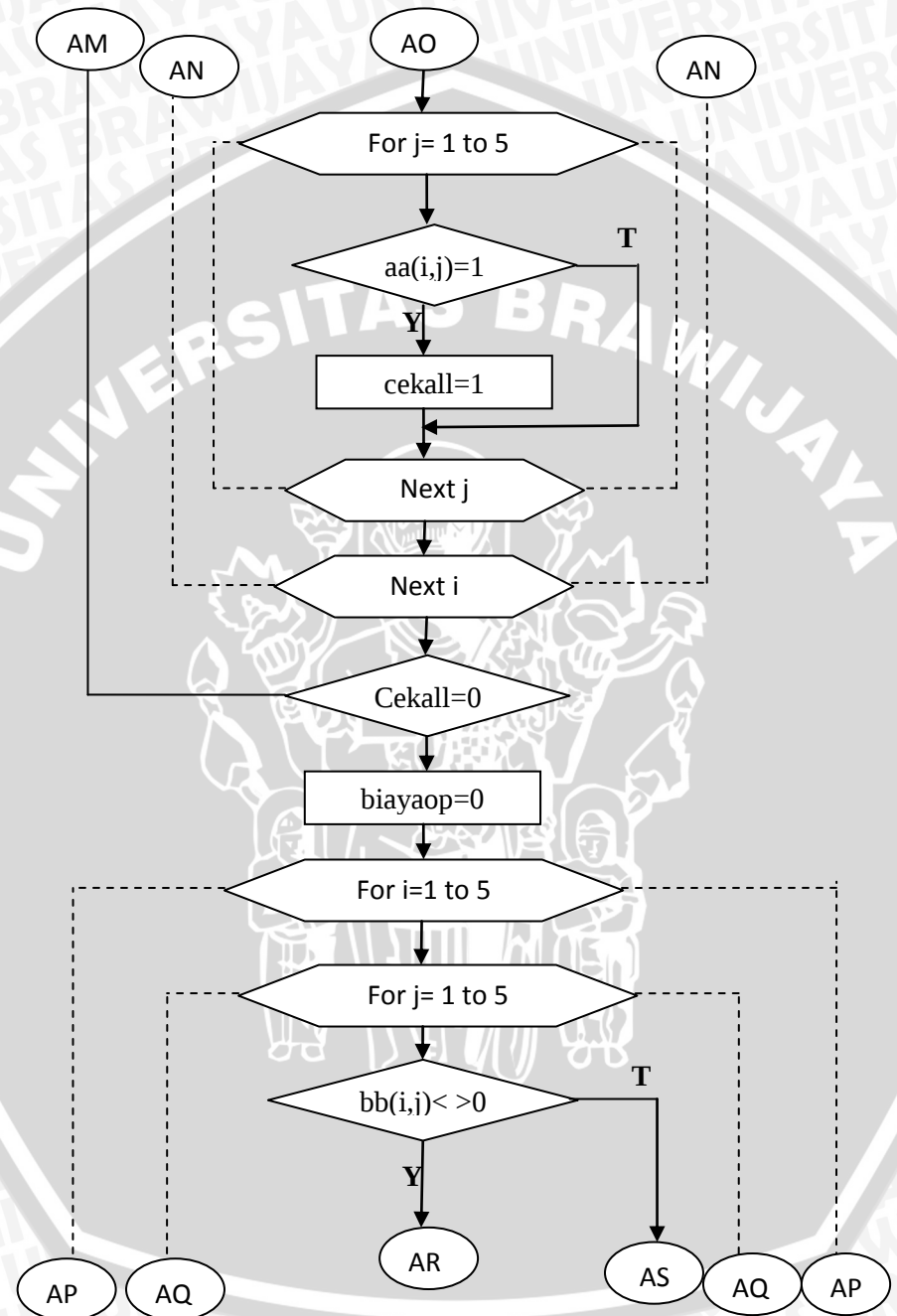


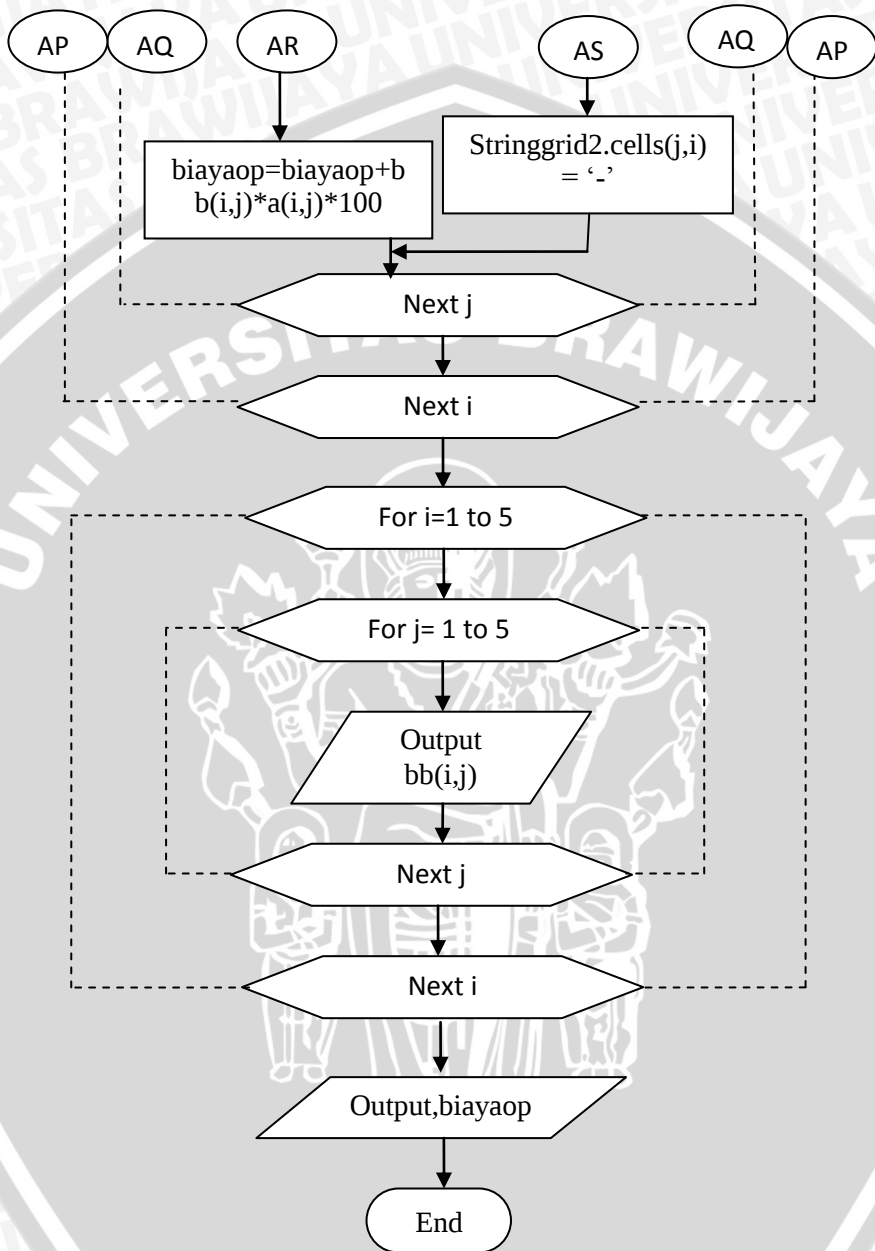












Lampiran 9

Listing program *Improved Vogel's Approximation Method* menggunakan software Delphi.

unit Unit1;
Interface
Uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, StdCtrls, ComCtrls, Buttons, math, ExtCtrls;
Type
TForm1 = class(TForm)
PageControl1: TPageControl;
TabSheet1: TTabSheet;
TabSheet2: TTabSheet;
Panel1: TPanel;
StringGrid2: TStringGrid;
SpeedButton2: TSpeedButton;
StringGrid1: TStringGrid;
Panel3: TPanel;
SpeedButton1: TSpeedButton;
GroupBox1: TGroupBox;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Edaventis: TEdit;
Ednovatris: TEdit;
Edmerck: TEdit;
Label10: TLabel;
Label11: TLabel;
Edbayer: TEdit;

Edabbot: TEdit;
GroupBox2: TGroupBox;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Edkediri: TEdit;
Edjember: TEdit;
Edblitar: TEdit;
Edmadiun: TEdit;
Edpasuruan: TEdit;
TabSheet3: TTabSheet;
Panel2: TPanel;
StringGrid5: TStringGrid;
StringGrid3: TStringGrid;
Label9: TLabel;
Label13: TLabel;
Label24: TLabel;
LOP2: TLabel;
Label12: TLabel;
Label14: TLabel;
procedure SpeedButton1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
Private
{ Private declarations }
Public
{ Public declarations }
end;
Var
Form1: TForm1;
a, a1, aa, bb: array [1..5,1..5] of real;

supplyawal,supply,demandawal,demand,jarak,biaya,kapasit as:array[1..5] of real;
sum:real;
Implementation
{SR *.dfm}
procedure TForm1.SpeedButton1Click(Sender: TObject);
var i,j:integer; amin,jum:real;
Begin
a[1,1]:=46; a[1,2]:=74; a[1,3]:=9; a[1,4]:=28; a[1,5]:=99;
a[2,1]:=12; a[2,2]:=75; a[2,3]:=6; a[2,4]:=36; a[2,5]:=48;
a[3,1]:=35; a[3,2]:=112; a[3,3]:=4; a[3,4]:=5; a[3,5]:=71;
a[4,1]:=61; a[4,2]:=81; a[4,3]:=44; a[4,4]:=88; a[4,5]:=9;
a[5,1]:=85; a[5,2]:=60; a[5,3]:=14; a[5,4]:=25; a[5,5]:=79;
StringGrid1.Cells[0,0]:='i,j';
StringGrid1.Cells[6,0]:='supply';
StringGrid1.Cells[0,6]:='demand';
StringGrid5.Cells[0,0]:='i,j';
StringGrid5.Cells[6,0]:='supply';
StringGrid5.Cells[0,6]:='demand';
StringGrid3.Cells[0,0]:='i,j';
StringGrid3.Cells[6,0]:='supply';
StringGrid3.Cells[0,6]:='demand';
supply[1]:=StrToFloat(Edaventis.Text);
supply[2]:=StrToFloat(Ednovatris.Text);
supply[3]:=StrToFloat(Edmerck.Text);
supply[4]:=StrToFloat(Edbayer.Text);
supply[5]:=StrToFloat(Edabbot.Text);
demand[1]:=StrToFloat(Edkediri.Text);

demand[2]:=StrToFloat(Edjembar.Text);
demand[3]:=StrToFloat(Edblitar.Text);
demand[4]:=StrToFloat(Edmadiun.Text);
demand[5]:=StrToFloat(Edpasuruan.Text);
supplyawal:=supply;
demandawal:=demand;
for j:=1 to 5 do
Begin
StringGrid1.Cells[j,0]:='b'+IntToStr(j);
StringGrid1.Cells[j,6]:=FloatToStr(demand[j]);
StringGrid5.Cells[j,0]:='b'+IntToStr(j);
StringGrid5.Cells[j,6]:=FloatToStr(demand[j]);
StringGrid3.Cells[j,0]:='b'+IntToStr(j);
StringGrid3.Cells[j,6]:=FloatToStr(demand[j]);
end;
jum:=0;
for i:=1 to 5 do
Begin
StringGrid1.Cells[0,i]:='a'+IntToStr(i);
StringGrid1.Cells[6,i]:=FloatToStr(supply[i]);
StringGrid5.Cells[0,i]:='a'+IntToStr(i);
StringGrid5.Cells[6,i]:=FloatToStr(supply[i]);
StringGrid3.Cells[0,i]:='a'+IntToStr(i);
StringGrid3.Cells[6,i]:=FloatToStr(supply[i]);
jum:=jum+supply[i];
end;
sum:=jum;
StringGrid1.Cells[6,6]:=FloatToStr(jum);
StringGrid5.Cells[6,6]:=FloatToStr(jum);
StringGrid3.Cells[6,6]:=FloatToStr(jum);
for i:=1 to 5 do
for j:=1 to 5 do
StringGrid1.Cells[j,i]:=FloatToStr(a[i,j]);
end;
a1:=a;

for i:=1 to 5 do
Begin
amin:=100000;
for j:=1 to 5 do
amin:=min(amin,a1[i,j]);
for j:=1 to 5 do
a1[i,j]:=a1[i,j]-amin;
end;
for i:=1 to 5 do
for j:=1 to 5 do
StringGrid5.Cells[j,i]:=FloatToStr(a1[i,j]);
for j:=1 to 5 do
Begin
amin:=100000;
for i:=1 to 5 do
amin:=min(amin,a1[i,j]);
for i:=1 to 5 do
a1[i,j]:=a1[i,j]-amin;
end;
for i:=1 to 5 do
for j:=1 to 5 do
StringGrid3.Cells[j,i]:=FloatToStr(a1[i,j]);
PageControl1.ActivePage:=TabSheet2;
TabSheet2.Enabled:=True;
end;
procedure TForm1.FormCreate(Sender: TObject);
Begin
PageControl1.ActivePage:=TabSheet1;
end;

procedure alokasikan(idk,idx:integer);
var k:integer;
Begin
if supply[idk]<demand[idx] then
Begin
bb[idk,idx]:=supply[idk];
demand[idx]:=demand[idx]-supply[idk];
for k:=1 to 5 do
aa[idk,k]:=0;
End
Else
Begin
bb[idk,idx]:=demand[idx];
supply[idk]:=supply[idk]-demand[idx];
for k:=1 to 5 do
aa[k,idx]:=0;
end;
end;
procedure alokasi(colrow:string; idk:integer);
var amin:real;
i,j,idx:integer;
Begin
amin:=1000000;
idx:=0;
if colrow='col' then
Begin
for i:=1 to 5 do
if aa[i,idx]=1 then
Begin
amin:=min(amin,a1[i,idx]);
if amin=a1[i,idx] then
idx:=i;
end;
alokasikan(idx,idx);

end;
if colrow='row' then
Begin
for j:=1 to 5 do
if aa[idk,j]=1 then
Begin
amin:=min(amin,a1[idk,j]);
if amin=a1[idk,j] then
idx:=j;
end;
alokasikan(idk,idx);
end;
end;
end;
procedure TForm1.SpeedButton2Click(Sender: TObject);
var ca,ra:array[1..5] of real;
it,i,j,irow,jcol,cekrow,cekcol,cekall,idx:integer;
amin1,amin2,maxrow,maxcol,biayaop:real;
Begin
supply:=supplyawal;
demand:=demandawal;
for i:=1 to 5 do
for j:=1 to 5 do
Begin
aa[i,j]:=1;
bb[i,j]:=0;
end;
it:=0;
Repeat
it:=it+1;
for i:=1 to 5 do
Begin
amin1:=10000;

amin2:=10000;
idx:=0;
irow:=0;
jcol:=0;
for j:=1 to 5 do
Begin
if aa[i,j]=1 then
Begin
amin1:=min(amin1,a1[i,j]);
if amin1=a1[i,j] then
idx:=j;
end;
end;
if (j<>idx) and (aa[i,j]=1) then
amin2:=min(amin2,a1[i,j]);
ra[i]:=amin2-amin1;
end;
for j:=1 to 5 do
Begin
amin1:=10000;
amin2:=10000;
for i:=1 to 5 do
Begin
if aa[i,j]=1 then
Begin
amin1:=min(amin1,a1[i,j]);
if amin1=a1[i,j] then
idx:=i;
end;
end;
if (i<>idx) and (aa[i,j]=1) then
amin2:=min(amin2,a1[i,j]);

ca[j]:=amin2-amin1;
end;
maxrow:=-10000;
maxcol:=-10000;
for i:=1 to 5 do
Begin
cekrow:=0;
for j:=1 to 5 do
if aa[i,j]=1 then
cekrow:=1;
if cekrow=1 then
Begin
maxrow:=max(maxrow,ra[i]);
if maxrow=ra[i] then
irow:=i;
end;
cekcol:=0;
for j:=1 to 5 do
if aa[j,i]=1 then
cekcol:=1;
if cekcol=1 then
Begin
maxcol:=max(maxcol,ca[i]);
if maxcol=ca[i] then
jcol:=i;
end;
end;
end;
if maxrow>maxcol then
alokasi('col',jcol)
Else
alokasi('row',irow);
cekall:=0;

for i:=1 to 5 do
for j:=1 to 5 do
if aa[i,j]=1 then
cekall:=1;
until (cekall=0);
StringGrid2.Cells[0,0]:='i,j';
StringGrid2.Cells[6,0]:='supply';
StringGrid2.Cells[0,6]:='demand';
for j:=1 to 5 do
Begin
StringGrid2.Cells[j,0]:='b'+IntToStr(j);
StringGrid2.Cells[j,6]:=FloatToStr(demandawal[j]);
end;
for i:=1 to 5 do
Begin
StringGrid2.Cells[0,i]:='a'+IntToStr(i);
StringGrid2.Cells[6,i]:=FloatToStr(supplyawal[i]);
end;
StringGrid2.Cells[6,6]:=FloatToStr(sum);
biayaop:=0;
for i:=1 to 5 do
for j:=1 to 5 do
if bb[i,j]<>0 then
Begin
StringGrid2.Cells[j,i]:=FloatToStr(bb[i,j]);
biayaop:=biayaop+bb[i,j]*a[i,j]*100;
End
Else

Lampiran 10



PT. NUSA SARI PHARMA

Jl. Anusapati 59 Singosari
Malang Jawa Timur
Fax. 0341 - 441911
Telp. 0341 - 453719

SURAT KETERANGAN

Yang bertanda tangan di bawah ini:

Nama : WIJANG PRAMUDI

Jabatan : Manajer Operasional PT. Nusa Sari Pharma

Dengan ini menerangkan bahwa:

Nama : Ester Nababan

NIM : 0810940036

Program Studi : Matematika

Judul TA : Penerapan *Improved version of Vogel's Approximation Method*

MENYATAKAN

Bahwa yang bersangkutan tersebut diatas telah melakukan konsultasi dan permintaan data untuk keperluan menyusun Tugas Akhir.

Demikian atas perhatian dan kerjasamanya kami sampaikan terima kasih

Singosari, 4 Agustus 2012

Manajer Operasional

