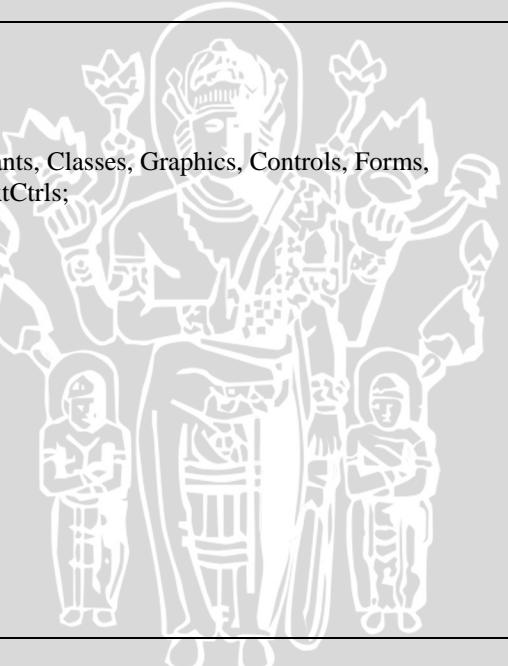


## LAMPIRAN 2 SOURCE CODE

### Lampiran 2.1 Source Code Nearest Neighbor

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Buttons, Grids, StdCtrls, ExtCtrls;
type
  range=1..25;
  himpunan=set of range;
  TForm1 = class(TForm)
    StringGrid1: TStringGrid;
    StringGrid2: TStringGrid;
    SpeedButton1: TSpeedButton;
    Edit1: TEdit;
    StringGrid3: TStringGrid;
    Panel1: TPanel;
    M: TLabel;
    Label1: TLabel;
    Label2: TLabel;
```



```
procedure SpeedButton1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  x,z:array[1..25,1..25] of real;
  y:array[1..25]of integer;
implementation

{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);
var i,j:integer;
begin
  for i:= 1 to 25 do
    begin
      stringgrid1.Cells[0,i]:=inttostr(i-1);
      stringgrid1.Cells[i,0]:=inttostr(i-1);
      stringgrid1.Cells[i,i]:='999';
    end;
end;
```



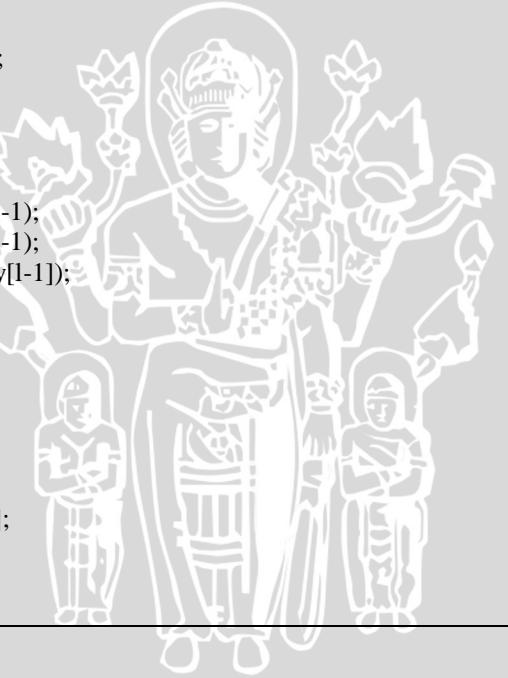
```
for j := 1 to 24 do
begin
stringgrid2.Cells[j,0]:=inttostr(j);
stringgrid2.Cells[0,1]:='delivery';
end;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
var tp, k:real;
i,j,kmax,sisa_demand,l,m,isi_truk:integer;
a:array[1..25,1..25]of real;
b:array[1..25]of real;
z1:himpunan;
begin
for i:=1 to 25 do
  for j:=i+1 to 25 do
x[i,j]:=strtofloat(stringgrid1.Cells[j,i]);
for i:=1 to 25 do
  for j:=i+1 to 25 do
StringGrid1.Cells[i,j]:=floattosrt(x[i,j]);
for i:=1 to 25 do
  for j:=1 to 25 do
    x[i,j]:=strtofloat(stringgrid1.Cells[j,i]);
sisa_demand:=0;
```

```
for i:=1 to 24 do
begin
y[i]:=strtoint(stringgrid2.Cells[i,1]);
sisa_demand:=sisa_demand +y[i];
end;
edit1.Text:=inttostr(sisa_demand);
for i:=1 to 25 do
a[i,1]:=1;
z1:=[];
for i := 2 to 25 do
z1:=z1+[i];
for i := 0 to 2 do
for j := 0 to 100 do
stringgrid3.Cells[i,j]:="";
m:=1;
while (sisa_demand>0)and(z1<>[]) do
begin
kmax:=200; i:=1;
while (kmax>0)and(z1<>[]) do
begin
k:=1000;
for j:=1 to 25 do
if (x[i,j]<k)and(j in z1) then
begin
```



```
k:=x[i,j];
l:=j;
end;
if k=1000 then k:=0;
kmax:=kmax-y[l-1];
sisa_demand:=sisa_demand-y[l-1];
isi_truk:=200-kmax;
if (kmax>0)and(k<>0) then
begin
z1:=z1-[l];
stringgrid3.Cells[0,m]:=floattostr(i-1);
stringgrid3.Cells[1,m]:=floattostr(l-1);
stringgrid3.Cells[2,m]:=floattostr(y[l-1]);
i:=l;
inc(m);
end else z1:=z1+[l];
end;
if z1<>[] then
begin
kmax:=kmax+y[l-1];
sisa_demand:=sisa_demand+y[l-1];
isi_truk:=200-kmax;
end;
stringgrid3.Cells[0,0]:='Dari';
```



```
stringgrid3.Cells[1,0]:='Ke';
stringgrid3.Cells[2,0]:='Demand';
stringgrid3.Cells[0,m+1]:='Sisa demand';
stringgrid3.Cells[0,m+2]:='Sisa isi truk';
stringgrid3.Cells[0,m+3]:='Isi truk skg';
stringgrid3.Cells[2,m+1]:=floattostr(sisa_demand);
stringgrid3.Cells[2,m+2]:=floattostr(kmax);
stringgrid3.Cells[2,m+3]:=floattostr(isi_truk);
m:=m+6;
end;
end;
end.
```

## Lampiran 2.2 Source Code Tabu Search

```
package javaapplication1;
public class combination {
    public static void main (String[]args){
        int bil[]={0,1,4,5,3,9,8,7,10,14,16,17,22,23,0};
        int bil[]={0,6,18,19,20,24,21,15,12,13,11,2,0};
        double jarak[][]={
```

|  |   |
|--|---|
|  | <p>{999,<br/>     11.52,18.24,14.16,12.48,13.08,18,13.92,13.44,15.12,15.12,20.76,22.68,24.24,16.32,22.08,19.01,20.76,23.<br/>     33,23.4,21.72,22.8,24.19,23.64,22.32},<br/>     {11.52,999,6.72,2.64,1.56,2.16,6,4.32,4.32,4.56,5.88,11.4,13.32,14.88,7.08,12.84,12.48,13.92,13.32,17.47<br/>     ,14.35,14.47,11.88,12.43,13.99},<br/>     {18.24,6.72,999,4.32,8.28,8.88,10.44,6.6,6.6,7.56,7.92,7.56,9.48,11.04,9.12,11.64,13.56,16.44,17.28,17.3<br/>     5,16.39,16.51,13.92,14.47,16.03},<br/>     {14.16,2.64,4.32,999,1.8,1.56,4.92,4.56,4.32,3.36,6,8.88,10.8,12.36,7.2,9.96,15.12,15.84,9.72,17.59,14.47<br/>     ,12.48,12,12.55,14.11},<br/>     {12.48,1.56,8.28,1.8,999,0.6,4.44,2.76,2.76,3,4.32,9.84,11.76,13.08,5.4,10.08,10.92,12,11.76,15.91,12.79,<br/>     10.8,10.32,10.87,12.43},<br/>     {13.08,2.16,8.88,1.56,0.6,999,3.84,2.16,2.16,2.4,3.72,9.24,11.16,12.48,4.92,9.48,10.32,11.64,11.16,15.31,<br/>     12.19,10.2,9.72,10.27,11.83},<br/>     {18,6,10.44,4.92,4.44,3.84,999,4.08,3.84,2.88,5.52,10.92,12.84,14.4,6.72,11.4,10.8,14.64,8.88,12.36,13.9<br/>     9,14.11,11.52,12.07,13.63},<br/>     {13.92,4.32,6.6,4.56,2.76,2.16,4.08,999,0.24,1.2,1.68,6.84,8.76,10.32,2.88,7.32,6.96,9.6,9.96,13.27,10.15,<br/>     10.27,7.68,8.23,9.79},<br/>     {13.44,4.32,6.6,4.32,2.76,2.16,3.84,0.24,999,0.96,1.68,7.08,9,10.56,2.88,7.56,6.96,9.6,9.72,13.27,10.15,1<br/>     0.27,7.68,8.23,9.79},<br/>     {15.12,4.56,7.56,3.36,3,2.4,2.88,1.2,0.96,999,2.64,8.04,9.96,11.52,3.84,8.52,7.92,10.56,8.76,12.24,11.11,<br/>     11.23,8.64,9.19,10.75},<br/>     {15.12,5.88,7.92,6.4,32,3.72,5.52,1.68,1.68,2.64,999,8.52,10.44,12,1.2,6.96,5.28,8.4,9.6,11.59,8.47,8.59,6</p> |
|--|---|

|   |
|---|
| ,6.55,8.11},<br>{20.76,11.4,7.56,8.88,9.84,9.24,10.92,6.84,7.08,8.04,8.52,999,1.92,3.48,7.32,8.16,10.08,13.56,14.4,17.83,<br>14.71,14.95,12.36,12.91,14.47},<br>{22.68,13.32,9.48,10.8,11.76,11.16,12.84,8.76,9,9.96,10.44,1.92,999,1.56,9.24,5.16,11.04,12,20.78,16.03,<br>12.91,13.03,10.44,10.99,12.55},<br>{24.24,14.88,11.04,12.36,13.08,12.48,14.4,10.32,10.56,11.52,12,3.48,1.56,999,10.8,6.48,12.6,11.52,21.67<br>,17.11,13.99,14.04,11.52,12.07,13.56},<br>{16.32,7.08,9.12,7.2,5.4,4.92,6.72,2.88,2.88,3.84,1.2,7.32,9.24,10.8,999,5.76,4.08,6.72,8.4,10.39,7.27,7.3<br>9.4,8.5,35.6,91},<br>{22.08,12.84,11.64,9.96,10.08,9.48,11.4,7.32,7.56,8.52,6.96,8.16,5.15,6.48,5.76,999,5.88,6.36,10.2,11.71,<br>7.75,9.43,5.28,5.83,7.39},<br>{19.01,12.48,13.56,15.12,10.92,10.32,10.8,6.96,6.96,7.92,5.28,10.08,11.04,12.6,4.08,5.88,999,3.84,4.32,5<br>.83,6.55,6.67,4.08,4.63,6.19},<br>{20.76,13.92,16.44,15.84,12,11.64,14.64,9.6,9.6,10.56,8.4,13.56,12,11.52,6.72,6.36,3.84,999,8.16,6.36,5.<br>28,6.36,1.92,2.47,5.88},<br>{22.33,13.32,17.28,9.72,11.76,11.16,8.88,9.96,9.72,8.76,9.6,14.4,20.78,21.67,8.4,10.2,4.32,8.16,999,3.48,<br>6.6,7.68,9.07,8.52,7.2},<br>{23.4,17.47,17.35,17.59,15.91,15.31,12.35,13.27,13.27,12.24,11.59,17.83,16.03,17.11,10.39,11.71,5.83,6.<br>36,3.48,999,3.12,4.2,5.59,5.04,3.72},<br>{21.72,14.35,16.39,14.47,12.79,12.19,13.99,10.15,10.15,11.11,8.47,14.71,12.91,13.99,7.27,7.75,6.55,5.28<br>.6.6,3.12,999,1.08,2.47,1.92,0.6},<br>{22.8,14.47,16.51,12.48,10.8,10.2,14.11,10.27,10.27,11.23,8.59,14.95,13.03,14.04,7.39,9.43,6.67,6.36,7.6} |
|---|

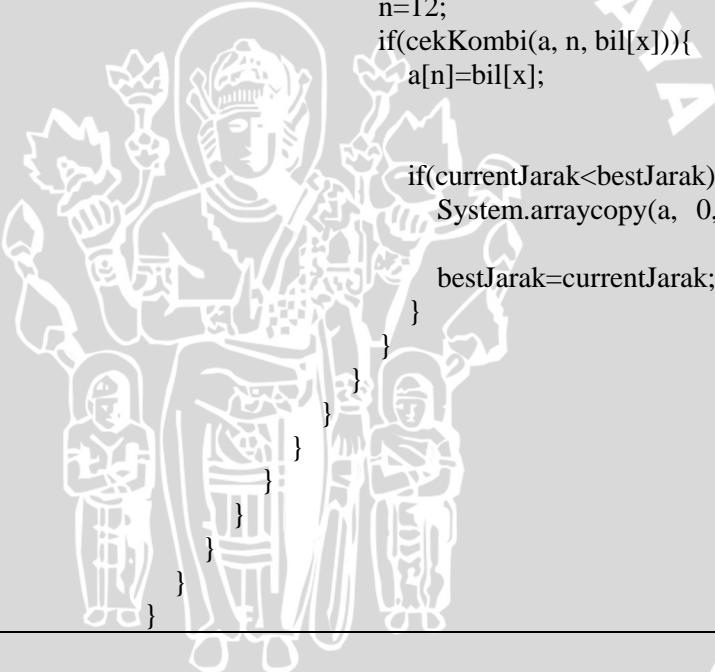
```
8.4.2,1.08,999,2.59,2.04,0.48},  
{24.19,11.88,13.92,12,10.32,9.72,11.52,7.68,7.68,8.64,6,12.36,10.44,11.52,4.8,5.28,4.08,1.92,9.07,5.59,2.  
47,2.59,999,0.55,2.11},  
{23.64,12.43,14.47,12.55,10.87,10.27,12.07,8.23,8.23,9.19,6.55,12.91,10.99,12.07,5.35,5.83,4.63,2.47,8.5  
2.5,04,1.92,2.04,0.55,999,1.56},  
{22.32,13.99,16.03,14.11,12.43,11.83,13.63,9.79,9.79,10.75,8.11,14.47,12.55,13.56,6.91,7.39,6.19,5.88,7.  
2,3.72,0.6,0.48,2.11,1.56,999}  
};  
kombinasi_13f(bil, jarak);  
kombinasi_11f(bil, jarak);  
}  
  
public static void kombinasi_13f(int []b,double[][]jarak){  
int bil[]=new int[b.length-2];  
double bestJarak=1000000;  
int bestSol[]=new int[bil.length];  
for(int i=0;i<bil.length;i++){  
    bil[i]=b[i+1];  
}  
int a[]=new int[bil.length];  
int n=0;long m=0;  
for(int i=0;i<bil.length;i++){
```

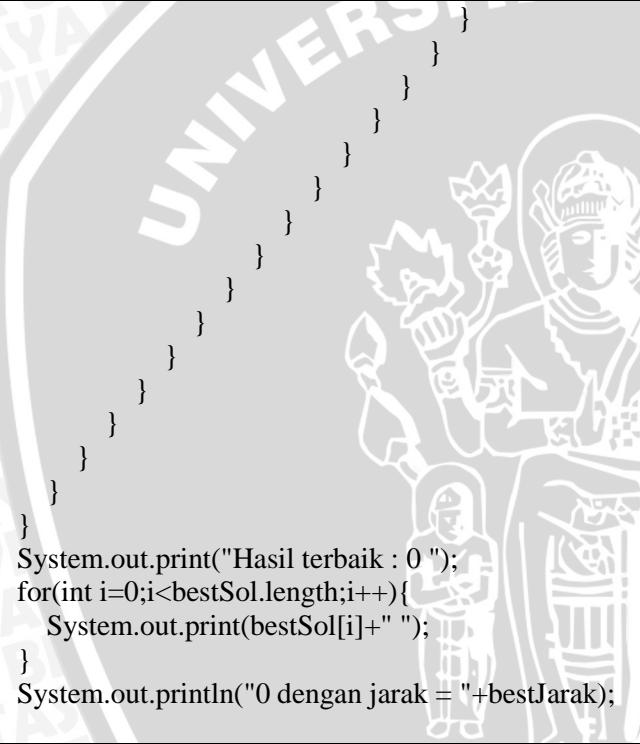
```
n=0;
a[n]=bil[i];
for(int j=0;j<bil.length;j++){
    n=1;
    if(i!=j){
        a[n]=bil[j];
        for(int k=0;k<bil.length;k++){
            n=2;
            if(cekKombi(a, n, bil[k])){
                a[n]=bil[k];
                for(int l=0;l<bil.length;l++){
                    n=3;
                    if(cekKombi(a, n, bil[l])){
                        a[n]=bil[l];
                        for(int p=0;p<bil.length;p++){
                            n=4;
                            if(cekKombi(a, n, bil[p])){
                                a[n]=bil[p];
                                for(int q=0;q<bil.length;q++){
                                    n=5;
                                    if(cekKombi(a, n, bil[q])){
                                        a[n]=bil[q];
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
for(int r=0;r<bil.length;r++){
    n=6;
    if(cekKombi(a, n, bil[r])){
        a[n]=bil[r];
        for(int s=0;s<bil.length;s++){
            n=7;
            if(cekKombi(a, n, bil[s])){
                a[n]=bil[s];
                for(int t=0;t<bil.length;t++){
                    n=8;
                    if(cekKombi(a, n, bil[t])){
                        a[n]=bil[t];
                        for(int u=0;u<bil.length;u++){
                            n=9;
                            if(cekKombi(a, n, bil[u])){
                                a[n]=bil[u];
                                for(int v=0;v<bil.length;v++){
                                    n=10;
                                    if(cekKombi(a, n, bil[v])){
                                        a[n]=bil[v];
                                        for(int w=0;w<bil.length;w++){
                                            n=11;
```

```
m++;  
currentJarak=hit_jarak(jarak, a);  
  
a.length);
```

```
if(cekKombi(a, n, bil[w])){  
    a[n]=bil[w];  
    for(int x=0;x<bil.length;x++){  
        n=12;  
        if(cekKombi(a, n, bil[x])){  
            a[n]=bil[x];  
        }  
    }  
    double  
    if(currentJarak<bestJarak){  
        System.arraycopy(a, 0, bestSol, 0,  
        bestJarak=currentJarak;  
    }  
}
```





```
        }  
    }  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
System.out.print("Hasil terbaik : 0 ");  
for(int i=0;i<bestSol.length;i++){  
    System.out.print(bestSol[i]+ " ");  
}  
System.out.println("0 dengan jarak = "+bestJarak);  
}
```

```
public static void kombinasi_11f(int []b,double[][]jarak){  
    int bil[]=new int[b.length-2];  
    double bestJarak=1000000;  
    int bestSol[]=new int[bil.length];  
    for(int i=0;i<bil.length;i++){  
        bil[i]=b[i+1];  
    }  
    int a[]=new int[bil.length];  
    int n=0;long m=0;  
    for(int i=0;i<bil.length;i++){  
        n=0;  
        a[n]=bil[i];  
        for(int j=0;j<bil.length;j++){  
            n=1;  
            if(i!=j){  
                a[n]=bil[j];  
                for(int k=0;k<bil.length;k++){  
                    n=2;  
                    if(cekKombi(a, n, bil[k])){  
                        a[n]=bil[k];  
                        for(int l=0;l<bil.length;l++){  
                            n=3;  
                            if(cekKombi(a, n, bil[l])){  
                                a[n]=bil[l];  
                                m=a[0]+a[1]+a[2]+a[3];  
                                if(m<bestJarak){  
                                    bestJarak=m;  
                                    bestSol[0]=a[0];bestSol[1]=a[1];bestSol[2]=a[2];bestSol[3]=a[3];  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```
n=3;
if(cekKombi(a, n, bil[1])){
    a[n]=bil[1];
    for(int p=0;p<bil.length;p++){
        n=4;
        if(cekKombi(a, n, bil[p])){
            a[n]=bil[p];
            for(int q=0;q<bil.length;q++){
                n=5;
                if(cekKombi(a, n, bil[q])){
                    a[n]=bil[q];
                    for(int r=0;r<bil.length;r++){
                        n=6;
                        if(cekKombi(a, n, bil[r])){
                            a[n]=bil[r];
                            for(int s=0;s<bil.length;s++){
                                n=7;
                                if(cekKombi(a, n, bil[s])){
                                    a[n]=bil[s];
                                    for(int t=0;t<bil.length;t++){
                                        n=8;
                                        if(cekKombi(a, n, bil[t])){
```

```
a);
```

```
a.length);
```

```
a[n]=bil[t];  
for(int u=0;u<bil.length;u++){  
    n=9;  
    if(cekKombi(a, n, bil[u])){  
        a[n]=bil[u];  
        for(int v=0;v<bil.length;v++){  
            n=10;  
            if(cekKombi(a, n, bil[v])){  
                a[n]=bil[v];  
                m++;  
                double currentJarak=hit_jarak(jarak,  
                    if(currentJarak<bestJarak){  
                        System.arraycopy(a, 0, bestSol, 0,  
                            bestJarak=currentJarak;
```

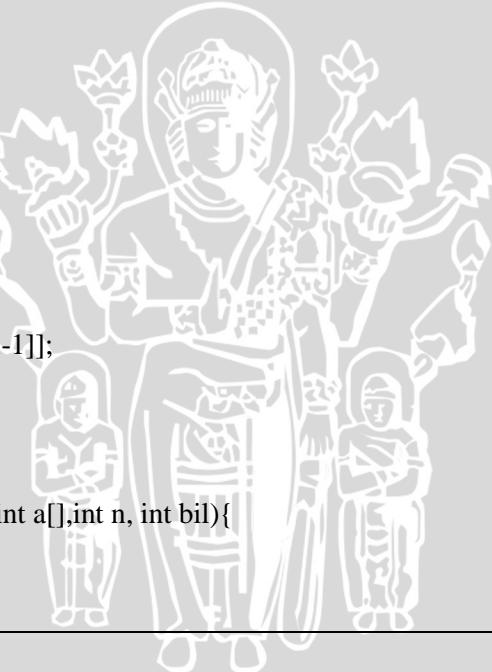


```
        }
    }
}
}

System.out.print("Hasil terbaik : 0 ");
for(int i=0;i<bestSol.length;i++){
    System.out.print(bestSol[i]+" ");
}
System.out.println("0 dengan jarak = "+bestJarak);
}
```

```
public static double hit_jarak(double jarak[][],int a[]){
    double jmlJarak=0;
    int bil[]=new int[a.length+2];
    for(int i=0;i<bil.length;i++){
        if(i==0||i==bil.length-1){
            bil[i]=0;
        }
        else
            bil[i]=a[i-1];
    }
    for(int i=0;i<bil.length;i++){
        if(i>=1)
            jmlJarak+=jarak[bil[i]][bil[i-1]];
    }
    return jmlJarak;
}

public static boolean cekKombi(int a[],int n, int bil){
    boolean status=true;
    for(int i=0;i<n;i++){
        if(a[i]==bil){
```



```
        status=false;
        break;
    }
    return status;
}

public static void cetakKombi(int a[]){
    for(int i=0;i<a.length;i++){
        System.out.print(a[i]+" ");
    }
}
```

UNIVERSITAS BRAWIJAYA

