PENGEMBANGAN APLIKASI BERBASIS *OCTAVE* DAN *VBA EXCEL* UNTUK MEDIA PEMBELAJARAN PENYELESAIAN *TSP* DENGAN *NEAREST NEIGHBOUR*

Ekra Sanggala 1), Rizki Rahma Putra 2)
[1]D4 Logistik Bisnis, Universitas Logistik & Bisnis Internasional
email: ekrasanggala@mail.ru
[2]D4 Logistik Bisnis, Universitas Logistik & Bisnis Internasional
email: rizkirahmaputra22@gmail.com

Abstrak

Travelling Salesman Problem (TSP) merupakan permasalahan pencarian rute terpendek, dimana rute ini diawali dari titik start kemudian mengunjungi sekumpulan titik tepat sekali dan diakhiri dengan kembali ke titik start. Nearest Neighbour (NN) mungkin merupakan heuristik paling sederhana dan paling mudah untuk menyelesaikan TSP. Walaupun NN merupakan heuristik yang sederhana dan mudah, seringkali untuk memahaminya tidak cukup dengan pemaparan melalui kalimat saja, oleh karena itu diperlukan media pembelajaran yang mampu menjelaskan NN dalam menyelesaikan TSP. Dengan memanfaatkan kemampuan GNU Octave yang cepat dalam menyelesaikan perhitungan dan kemampuan VBA Excel untuk menampilkan langkah-langkah perhitungan ke dalam spreadsheet, maka dapat dibuat sebuah media pembelajaran berbasis komputer untuk memahami NN dalam menyelesaikan TSP.

Kata Kunci: *TSP, Nearest Neighbour, Media Pembelajaran, Octave, VBA Excel*

1. PENDAHULUAN

Travelling Salesman Problem (*TSP*) merupakan permasalahan pencarian rute terpendek, dimana rute ini diawali dari titik *start* kemudian mengunjungi sekumpulan titik tepat sekali dan diakhiri dengan kembali ke titik *start*. *TSP* ini termasuk ke dalam *Non-Deterministic Polynomial-Time Hard Problems* (*NP-Hard Problems*), dimana jika titik yang harus dikunjungi berjumlah banyak, maka untuk mendapatkan rute terpendeknya diperlukan waktu yang sangat lama. Sebagai contoh jika terdapat 50 titik yang harus dikunjungi makan akan terdapat 50! (3,04140932017134 x 10^{64}) alternatif rute, dengan alternatif rute sebanyak ini tentunya akan memerlukan waktu yang sangat lama untuk mendapatkan rute terbaiknya, sekalipun perhitungannya menggunakan super komputer tercanggih [1]. Dikarenakan permasalahan perhitungan yang lama ini, maka algoritma yang perhitungannya cepat walaupun "hanya" menghasilkan solusi yang mendekati optimal, merupakan opsi yang sangat baik untuk menyelesaikan *TSP* [2]. Algoritma jenis ini dikenal dengan sebutan Heuristik. "Heuristik" merupakan sebuah kata dari bahasa Yunani yang berarti "menemukan" atau "mengeksplorasi". Heuristik disebut sebagai *approximate techniques*. Tujuan utama dari heuristik adalah untuk membangun sebuah model optimasi yang mudah dipahami dan mampu memberikan solusi yang baik dalam waktu perhitungan yang wajar [3]. Beberapa heuristik yang cukup populer untuk menyelesaikan *TSP* antara lain [2]:

1. *Nearest Neighbour*
2. *Nearest Insertion*
3. *Christofides Algorithm*
4. *Lin-Kernighan*

Nearest Neighbour (*NN*) mungkin merupakan heuristik paling sederhana dan paling mudah untuk menyelesaikan *TSP*. Inti dari *NN* adalah mengunjungi kota terdekat [2].

Tahapan-tahapan dari *NN* adalah sebagai berikut ini [2]:

1. Tentukan titik *start*.

2. Pilih titik terdekat yang belum terkunjungi sebagai titik yang akan dikunjungi.
3. Apa masih ada titik yang belum terkunjungi? Jika masih ada, ulangi langkah ke-2.
4. Kembali ke titik *start*.

Walaupun *NN* merupakan heuristik yang sederhana dan mudah, seringkali untuk memahaminya tidak cukup dengan pemaparan melalui kalimat saja, oleh karena itu diperlukan media pembelajaran yang mampu menjelaskan *NN* dalam menyelesaikan *TSP*.

Media pembelajaran berbasis komputer adalah media pembelajaran yang menggunakan *software* komputer. Melalui komputer diharapkan penyampaian materi pelajaran lebih mudah dan pembelajaran dapat berlangsung dengan lebih menyenangkan [4].

GNU Octave merupakan alat multifungsi untuk melakukan analisis numerik. Berbagai alat bantu yang tersedia di dalam *GNU Octave* adalah sebagai berikut ini [5]:

1. Sekumpulan *function* untuk menyelesaikan berbagai masalah.
2. Bahasa pemrograman yang dapat digunakan untuk mengembangkan kemampuan *GNU Octave*.
3. Berbagai fasilitas untuk melakukan *plotting*.

Penggunaan *Microsoft Office Excel* dapat memberi kemudahan dan kenyamanan dalam melakukan berbagai perhitungan [6]. *Excel* menampilkan hasil perhitungan dalam bentuk *spreadsheet* sehingga tahapan-tahapan perhitungan dapat lebih mudah dipahami [7]. Kombinasi antara *Visual Basic Application* (*VBA*) dengan *Excel* menjadi sebuah alat yang sangat baik. Dengan menggunakan *VBA*, tugas-tugas kita dalam *Excel* dapat diselesaikan lebih cepat. Jika menggunakan *Excel* untuk menyelesaikan tugas-tugas rutin, kita dapat menggunakan *VBA* untuk menyelesaikannya hanya dalam beberapa detik [8].

Dengan memanfaatkan kemampuan *GNU Octave* yang cepat dalam menyelesaikan perhitungan dan kemampuan *VBA Excel* untuk menampilkan langkah-langkah perhitungan ke dalam *spreadsheet*, maka dalam tulisan ini akan dibahas mengenai "Pengembangan Aplikasi Berbasis *Octave* dan *VBA Excel* Untuk Media Pembelajaran Penyelesaian *TSP* Dengan *Nearest Neighbour*".

2. METODE PENELITIAN

Secara garis besar langkah-langkah dalam penelitian ini adalah sebagai berikut ini:

1. Menyiapkan *Instance*
   Instance diperlukan sebagai contoh *TSP* yang akan diselesaikan. *Instance* yang akan digunakan adalah *Russia-10-Nodes-TSP*. Pada *instance* ini terdapat 10 kota, yaitu:
   1. Moscow (Latitude: 55,75222; Longitude: 37,61556).
   2. Saint Petersburg (59,93863; 30,31413).
   3. Novosibirsk (55,04150; 82,93460).
   4. Yekaterinburg (56,85190; 60,61220).
   5. Nizhniy Novgorod (56,32867; 44,00205).
   6. Samara (53,20007; 50,15000).
   7. Omsk (54,99244; 73,36859).
   8. Kazan (55,78874; 49,12214).
   9. Rostov-na-Donu (47,23135; 39,72328).
   10. Chelyabinsk (55,15402; 61,42915).

   Titik *start* adalah kota Moscow.

2. Menghitung jarak antara setiap kota.
   Untuk menghitung jarak antara setiap kota digunakan *Euclidean Distance* formula yang bentuk persamaanya seperti berikut ini [9]:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

   d_{ij} : Jarak kota i ke kota j
   x_i : Latitude kota i
   x_j : Latitude kota j
   y_i : Longitude kota i
   y_j : Longitude kota j

   Untuk menghitung jarak antara setiap kota ini digunakan *GNU Octave.*

3. Menentukan rute berdasarkan *Nearest Neighbour*.
   Perhitungan untuk menentukan rute berdasarkan *NN* akan dilakukan dengan bantuan *GNU Octave*, oleh karena itu akan dibuat *script* bahasa pemrograman dalam lingkungan *GNU Octave*. Hasil perhitungannya akan disimpan dalam file bertipe *text*.

4. Mengembangkan aplikasi berbasis *VBA Excel* untuk menampilkan file *text* yang telah dihasilkan ke dalam *spreadsheet*.

3. HASIL DAN PEMBAHASAN

3.1. Menyiapkan *Instance*

Instance Russia-10-Nodes-TSP disimpan dalam file *text* yang dapat dibaca dengan mudah oleh *script* yang akan dituliskan dalam lingkungan *GNU Octave*. Isi dari file *text* ini dapat dilihat pada lampiran 1.

3.2. Menghitung Jarak Antara Setiap Kota

Dengan bantuan *GNU Octave* akan dihitung jarak antara setiap kota berdasarkan *Euclidean Distance* formula. *Script* perhitungan jarak antara setiap kota ini dapat dilihat pada lampiran 2, sedangkan jarak antara setiap kotanya dapat dilihat pada lampiran 3.

3.3. Menentukan Rute Berdasarkan *Nearest Neighbour*

Dengan bantuan *GNU Octave* akan dilakukan perhitungan untuk menentukan rute berdasarkan *NN*. *Script* perhitungan penentuan rute ini dapat dilihat pada lampiran 4, sedangkan file *text* yang dihasilkan dapat dilihat pada lampiran 5.

3.4. Mengembangkan Aplikasi Berbasis *VBA Excel*

Aplikasi Berbasis *VBA Excel* ini berguna untuk menampilkan file *text* yang telah dihasilkan oleh *GNU Octave*, ke dalam *spreadsheet* sehingga lebih mudah dibaca. Tampilan dari aplikasi yang dihasilkan dapat dilihat pada lampiran 6, sedangkan *script*-nya dapat dilihat pada lampiran 7 dan untuk tampilan *spreadsheet*-nya dapat dilihat pada lampiran 8.

4. KESIMPULAN

Pembelajaran mengenai penyelesaian *TSP* dengan *Nearest Neighbour* seringkali sulit dipahami jika hanya dijelaskan dengan kalimat saja. Dengan adanya media pembelajaran berbasis komputer diharapkan dapat membantu memahami penyelesaian *TSP* dengan *Nearest Neighbour*. Media pembelajaran berbasis komputer ini dapat memberi tampilan *Visual* yang tentunya akan memberi kemudahan dalam memahami suatu hal. Dengan adanya media pembelajaran berbasis komputer untuk menjelaskan penyelesaian *TSP* dengan *Nearest Neighour*, diharapkan langkah-langkah penyelesaian *TSP* dengan *NN* dapat lebih mudah dipahami.

5. REFERENSI

Lampiran 1
Russia-10-Nodes-TSP.txt

1. D.L. Applegate, R.E. Bixby, V. Chvatal, and W.J. Cook, *The Traveling Salesman Problem A Computational Study*, Princeton University Press, New Jersey, 2006.
2. C. Nilsson, *Heuristics for The Traveling Salesman Problem*, Linkoping University.
3. K. Kumar, D. Zindani, and J. Paulo Davim, *Metaheuristics Optimizing Engineering Problems Through Heuristic Techniques*, CRC Press, USA, 2020.
4. I. Mulyani Sihotang, *Media Pembelajaran Berbasis Komputer Dalam Menyusun Laporan Keuangan*, Universitas Negeri Padang, 2015.
5. J. Schmidt Hansen, *GNU Octave Beginner's Guide*, PACKT, Birmingham, 2011.
6. M. Catana, C. Neagu, and S. Tonoiu, *Scheduling of Job-Shop Production Projects with Microsoft Office Excel*, Academic Journal of Manufacturing Engineering, Vol. 7, pp.31-36, April 2009.
7. A. Katz, *Beginning Microsoft Excel 2010*, Apress, New York, 2006.
8. J. Walkenbach, *Excel 2010 Power Programming with VBA*, Wiley Publishing, Indiana 2010.
9. https://www.omnicalculator.com/math/euclidean-distance

1	0	55.75222	37.61556
2	1	59.93863	30.31413
3	2	55.04150	82.9346
4	3	56.85190	60.6122
5	4	56.32867	44.00205
6	5	53.20007	50.15
7	6	54.99244	73.36859
8	7	55.78874	49.12214
9	8	47.23135	39.72328
10	9	55.15402	61.42915

Lampiran 2
EUCLIDEAN_DISTANCE.m

EUCLIDEAN_DISTANCE.m

```
clear;

%INSTANCES DIRECTORY
cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP\Instances');

INSTANCES=readdir(".");

%BACK TO TSP DIRECTORY
cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP');

for i=3:size(INSTANCES,1)
  INSTANCE=char(INSTANCES(i))(1,1:size(char(INSTANCES(i)),2)-4)

  %INSTANCES DIRECTORY
  cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP\Instances');

  NODE=load(strcat(INSTANCE,".txt"));

  %CALCULATING EUCLIDEAN DISTANCE
  DISTANCE=round(sqrt(((NODE'(2,:).-NODE(:,2)).^2)+((NODE'(3,:).-NODE(:,3)).^2)));

  %DISTANCES DIRECTORY
  cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP\Distances');

  dlmwrite(strcat(INSTANCE,"_Distance.txt"),DISTANCE,"delimiter","\t");

  clear("NODE");
  clear("DISTANCE");
endfor

%BACK TO TSP DIRECTORY
cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP');
```

Lampiran 3
Russia-10-Nodes-TSP_Distance.txt

0	4	2	3	4	6	6	7	12	9
4	0	5	4	5	8	7	7	15	9
2	5	0	2	2	4	4	5	10	7
3	4	2	0	1	4	4	4	11	6
4	5	2	1	0	3	2	3	10	5
6	8	4	4	3	0	2	3	7	4
6	7	4	4	2	2	0	1	8	3
7	7	5	4	3	3	1	0	9	2

Lampiran 4
NEAREST_NEIGHBOUR_TSP.m

NEAREST_NEIGHBOUR_TSP.m

```
clc;

clear;

%INSTANCES DIRECTORY
cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP\Instances');

INSTANCES=readdir(".");

global DISTANCE;

for i=3:size(INSTANCES,1)
  INSTANCE=char(INSTANCES(i))(1,1:size(char(INSTANCES(i)),2)-4)

  %DISTANCES DIRECTORY
  cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP\Distances');

  DISTANCE=load(strcat(INSTANCE,"_Distance.txt"));

  %INDIVIDUAL NEAREST NEIGHBOUR
  DISTANCE_DUMMY=DISTANCE;
  j=1;

  START_FINISH_NODE=1;

  NEXT_NODE=START_FINISH_NODE;
  INDIVIDUAL_NEAREST_NEIGHBOUR(1,j)=NEXT_NODE;

  %SOLUTIONS NEAREST NEIGHBOUR DIRECTORY
  cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP\Solutions\Nearest Neighbour');

  INSTANCE_SIZE=size(DISTANCE,2);
  dlmwrite(strcat(INSTANCE,"_Size.txt"),INSTANCE_SIZE,"delimiter","\t");
```

NEAREST_NEIGHBOUR_TSP.m

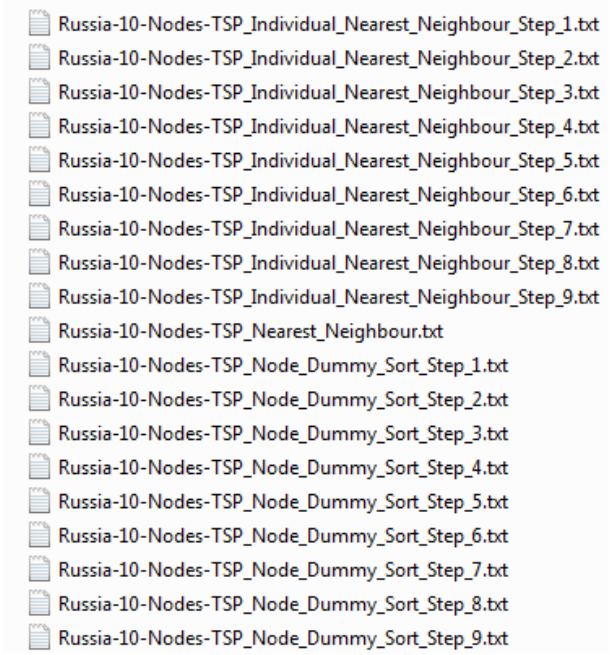
```
34
35  STEP=0;
36  for j=2:size(DISTANCE,2)
37    STEP=STEP+1;
38    [DISTANCE_DUMMY_SORT, NODE_DUMMY_SORT]=sort(DISTANCE_DUMMY(NEXT_NODE,:));
39    dlmwrite(strcat(INSTANCE,"_Distance_Dummy_Sort_Step_",num2str(STEP),".txt"),DISTANCE_DUMMY_SORT(1,2:size(DISTANCE_DUMMY_SORT,2)),"delimiter","\t");
40    dlmwrite(strcat(INSTANCE,"_Node_Dummy_Sort_Step_",num2str(STEP),".txt"),NODE_DUMMY_SORT(1,2:size(NODE_DUMMY_SORT,2)),"delimiter","\t");
41
42    NEXT_NODE=NODE_DUMMY_SORT(1,2);
43    INDIVIDUAL_NEAREST_NEIGHBOUR(1,j)=NEXT_NODE;
44    dlmwrite(strcat(INSTANCE,"_Individual_Nearest_Neighbour_Step_",num2str(STEP),".txt"),INDIVIDUAL_NEAREST_NEIGHBOUR,"delimiter","\t");
45
46    DISTANCE_DUMMY(INDIVIDUAL_NEAREST_NEIGHBOUR(1,j-1),:)=999999999;
47    DISTANCE_DUMMY(:,INDIVIDUAL_NEAREST_NEIGHBOUR(1,j-1))=999999999;
48    dlmwrite(strcat(INSTANCE,"_Distance_Dummy_Step_",num2str(STEP),".txt"),DISTANCE_DUMMY,"delimiter","\t");
49
50  endfor
51  j=j+1;
52  NEXT_NODE=START_FINISH_NODE;
53  INDIVIDUAL_NEAREST_NEIGHBOUR(1,j)=NEXT_NODE;
54
55  %BACK TO TSP DIRECTORY
56  cd ('D:\Files\Politeknik Pos\Laboratorium SCM\TSP');
57
58  INDIVIDUAL_NEAREST_NEIGHBOUR(1,size(INDIVIDUAL_NEAREST_NEIGHBOUR,2)+1)=...
59  FUNCTION_LENGTH(INDIVIDUAL_NEAREST_NEIGHBOUR(1,1:size(INDIVIDUAL_NEAREST_NEIGHBOUR,2)));
60
61  %SOLUTIONS NEAREST NEIGHBOUR DIRECTORY
62  cd ('D:\Files\Politeknik POS\Laboratorium SCM\TSP\Solutions\Nearest Neighbour');
63  dlmwrite(strcat(INSTANCE,"_Nearest_Neighbour.txt"),INDIVIDUAL_NEAREST_NEIGHBOUR,"delimiter","\t");
64
65  clear('DISTANCE');
66  clear('INDIVIDUAL_NEAREST_NEIGHBOUR');
```

NEAREST_NEIGHBOUR_TSP.m

```
66    clear('INDIVIDUAL_NEAREST_NEIGHBOUR');
67  endfor
68
69  %BACK TO TSP DIRECTORY
70  cd ('D:\Files\Politeknik Pos\Laboratorium SCM\TSP');
```

Lampiran 5

File *text* yang Dihasilkan



Lampiran 6

Visualization of Nearest Neighbour.xlsm

Visualization of Nearest Neighbour

Directory File Master Visualization of NN	D:\Files\Politeknik Pos\Laboratorium SCM\TSP
File Master Visualization of NN	Visualization of Nearest Neighbour.xlsm
Directory Instances	D:\Files\Politeknik Pos\Laboratorium SCM\TSP\Instances
Directory Distances	D:\Files\Politeknik Pos\Laboratorium SCM\TSP\Distances
Directory Solutions	D:\Files\Politeknik Pos\Laboratorium SCM\TSP\Solutions\Nearest Neighbour
Directory Visualization	D:\Files\Politeknik Pos\Laboratorium SCM\TSP\Visualization

Visualize

Lampiran 7

Visualization of Nearest Neighbour.xlsm

(General) (Declarations)

```
Dim Directory_File_Master_Visualization_of_NN As String
Dim File_Master_Visualization_of_NN As String
Dim Directory_Instances As String
Dim Directory_Distance As String
Dim Directory_Solutions As String
Dim Directory_Visualization As String

Private Sub CommandButton1_Click()
    Application.DisplayAlerts = False

    Directory_File_Master_Visualization_of_NN = Sheets("Options").Cells(2, 2).Value
    File_Master_Visualization_of_NN = Sheets("Options").Cells(3, 2).Value
    Directory_Instances = Sheets("Options").Cells(4, 2).Value
    Directory_Distance = Sheets("Options").Cells(5, 2).Value
    Directory_Solutions = Sheets("Options").Cells(6, 2).Value
    Directory_Visualization = Sheets("Options").Cells(7, 2).Value

    File_Instances = Dir(Directory_Instances & "\")

    No_Instances = 0

    While (File_Instances <> "")
        Instance = Left(File_Instances, Len(File_Instances) - 4)

        Workbooks.Add
        Sheets("Sheet1").Cells(1, 1) = Instance
        Sheets("Sheet1").Cells(3, 1) = "Distance"

        File_Name_Visualization = Instance & "_Visualization.xlsx"

        ActiveWorkbook.SaveAs Filename:=Directory_Visualization & "\" & File_Name_Visualization _
            , FileFormat:=xlOpenXMLWorkbook, CreateBackup:=False

        File_Name_Instance_Size = Instance & "_Size.txt"

        Workbooks.OpenText Filename:= _
            Directory_Solutions & "\" & File_Name_Instance_Size _
```

(General) (Declarations)

```
Workbooks.OpenText Filename:= _
    Directory_Solutions & "\" & File_Name_Instance_Size _
    , Origin:=xlMSDOS, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
    xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, _
    Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(1, 1), _
    TrailingMinusNumbers:=True

Instance_Size = Sheets(1).Cells(1, 1).Value
ActiveWorkbook.Close

Windows(File_Name_Visualization).Activate

For i = 1 To Instance_Size
    Sheets("Sheet1").Cells(3 + i, 1).Value = i

    Sheets("Sheet1").Cells(3, 1 + i).Value = i
Next

File_Name_Distance = Instance & "_Distance.txt"

Workbooks.OpenText Filename:= _
    Directory_Distance & "\" & File_Name_Distance _
    , Origin:=xlMSDOS, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
    xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, _
    Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), _
    Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _
    Array(9, 1), Array(10, 1)), TrailingMinusNumbers:=True

Sheets(1).Range(Selection, Selection.End(xlToRight)).Select
Sheets(1).Range(Selection, Selection.End(xlDown)).Select
Selection.Copy
Windows(File_Name_Visualization).Activate
Sheets("Sheet1").Cells(4, 2).Select
ActiveSheet.Paste
Sheets("Sheet1").Cells(1, 1).Select

Windows(File_Name_Distance).Close
```

(General) (Declarations)

```
Acuan_Row_Step = 3 + Instance_Size + 2

For j = 1 To Instance_Size - 1
    Sheets("Sheet1").Cells(Acuan_Row_Step, 1) = "Step " & j
    Sheets("Sheet1").Cells(Acuan_Row_Step + 1, 1) = "Distance Sorted"
    Sheets("Sheet1").Cells(Acuan_Row_Step + 1, 1).Columns.AutoFit

    File_Distance_Sorted = Instance & "_Distance_Dummy_Sort_Step_" & j & ".txt"

    Workbooks.OpenText Filename:= _
        Directory_Solutions & "\" & File_Distance_Sorted _
        , Origin:=xlMSDOS, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
        xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, _
        Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), _
        Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _
        Array(9, 1)), TrailingMinusNumbers:=True

    Sheets(1).Range(Selection, Selection.End(xlToRight)).Select
    Selection.Copy
    Windows(File_Name_Visualization).Activate
    Sheets("Sheet1").Cells(Acuan_Row_Step + 1, 2).Select
    ActiveSheet.Paste
    Sheets("Sheet1").Cells(1, 1).Select

    Windows(File_Distance_Sorted).Close

    Sheets("Sheet1").Cells(Acuan_Row_Step + 2, 1) = "Node Sorted"

    File_Node_Sorted = Instance & "_Node_Dummy_Sort_Step_" & j & ".txt"

    Workbooks.OpenText Filename:= _
        Directory_Solutions & "\" & File_Node_Sorted _
        , Origin:=xlMSDOS, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
        xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, _
        Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), _
        Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _
        Array(9, 1)), TrailingMinusNumbers:=True
```

(General) (Declarations)

```
            Sheets(1).Range(Selection, Selection.End(xlToRight)).Select
            Selection.Copy
            Windows(File_Name_Visualization).Activate
            Sheets("Sheet1").Cells(Acuan_Row_Step + 2, 2).Select
            ActiveSheet.Paste
            Sheets("Sheet1").Cells(1, 1).Select

            Windows(File_Node_Sorted).Close

            Sheets("Sheet1").Cells(Acuan_Row_Step + 3, 1) = "Route"

            File_Route = Instance & "_Individual_Nearest_Neighbour_Step_" & j & ".txt"

            Workbooks.OpenText Filename:= _
                Directory_Solutions & "\" & File_Route _
                , Origin:=xlMSDOS, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
                xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, _
                Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), _
                Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _
                Array(9, 1)), TrailingMinusNumbers:=True

            Sheets(1).Range(Selection, Selection.End(xlToRight)).Select
            Selection.Copy
            Windows(File_Name_Visualization).Activate
            Sheets("Sheet1").Cells(Acuan_Row_Step + 3, 2).Select
            ActiveSheet.Paste
            Sheets("Sheet1").Cells(1, 1).Select

            Windows(File_Route).Close

            Acuan_Row_Step = Acuan_Row_Step + 5
        Next



        Sheets("Sheet1").Cells(Acuan_Row_Step, 1) = "NN Route"
```

CommandButton1 Click

```
        File_NN_Route = Instance & "_Nearest_Neighbour.txt"
        Workbooks.OpenText Filename:= _
            Directory_Solutions & "\" & File_NN_Route _
            , Origin:=xlMSDOS, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
            xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, _
            Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), _
            Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _
            Array(9, 1)), TrailingMinusNumbers:=True

        Sheets(1).Range(Selection, Selection.End(xlToRight)).Select
        Selection.Copy
        Windows(File_Name_Visualization).Activate
        Sheets("Sheet1").Cells(Acuan_Row_Step, 2).Select
        ActiveSheet.Paste
        Sheets("Sheet1").Cells(1, 1).Select

        Windows(File_NN_Route).Close

        ActiveWorkbook.Save

        No_Instances = No_Instances + 1

        File_Instances = Dir
    Wend
End Sub
```

Lampiran 8

Russia-10-Nodes-TSP_Visualization.xlsx

	A	B	C	D	E	F	G	H	I	J	K
1	Russia-10-Nodes-TSP										
2											
3	Distance	1	2	3	4	5	6	7	8	9	10
4	1	0	4	2	3	4	6	6	7	12	9
5	2	4	0	5	4	5	8	7	7	15	9
6	3	2	5	0	2	2	4	4	5	10	7
7	4	3	4	2	0	1	4	4	4	11	6
8	5	4	5	2	1	0	3	2	3	10	5
9	6	6	8	4	4	3	0	2	3	7	4
10	7	6	7	4	4	2	2	0	1	8	3
11	8	7	7	5	4	3	3	1	0	9	2
12	9	12	15	10	11	10	7	8	9	0	8
13	10	9	9	7	6	5	4	3	2	8	0
14											
15	Step 1										
16	Distance Sorted	2	3	4	4	6	6	7	9	12	
17	Node Sorted	3	4	2	5	6	7	8	10	9	
18	Route	1	3								
19											
20	Step 2										
21	Distance Sorted	2	2	4	4	5	5	7	10	1E+09	
22	Node Sorted	4	5	6	7	2	8	10	9	1	
23	Route	1	3	4							
24											
25	Step 3										
26	Distance Sorted	1	4	4	4	4	6	11	1E+09	1E+09	
27	Node Sorted	5	2	6	7	8	10	9	1	3	
28	Route	1	3	4	5						
29											
30	Step 4										
31	Distance Sorted	2	3	3	5	5	10	1E+09	1E+09	1E+09	
32	Node Sorted	7	6	8	2	10	9	1	3	4	
33	Route	1	3	4	5	7					

	A	B	C	D	E	F	G	H	I	J	K	L	M
35	Step 5												
36	Distance Sorted	1	2	3	7	8	1E+09	1E+09	1E+09	1E+09			
37	Node Sorted	8	6	10	2	9	1	3	4	5			
38	Route	1	3	4	5	7	8						
39													
40	Step 6												
41	Distance Sorted	2	3	7	9	1E+09	1E+09	1E+09	1E+09	1E+09			
42	Node Sorted	10	6	2	9	1	3	4	5	7			
43	Route	1	3	4	5	7	8	10					
44													
45	Step 7												
46	Distance Sorted	4	8	9	1E+09	1E+09	1E+09	1E+09	1E+09	1E+09			
47	Node Sorted	6	9	2	1	3	4	5	7	8			
48	Route	1	3	4	5	7	8	10	6				
49													
50	Step 8												
51	Distance Sorted	7	8	1E+09	1E+09	1E+09	1E+09	1E+09	1E+09	1E+09			
52	Node Sorted	9	2	1	3	4	5	7	8	10			
53	Route	1	3	4	5	7	8	10	6	9			
54													
55	Step 9												
56	Distance Sorted	15	1E+09	1E+09	1E+09	1E+09	1E+09	1E+09	1E+09	1E+09			
57	Node Sorted	2	1	3	4	5	6	7	8	10			
58	Route	1	3	4	5	7	8	10	6	9	2		
59													
60	NN Route	1	3	4	5	7	8	10	6	9	2	1	40