

LAPORAN TUGAS KECIL 3
IF2211 STRATEGI ALGORITMA
SEMESTER II TAHUN 2020/2021

IMPLEMENTASI ALGORITMA A* UNTUK
MENENTUKAN LINTASAN TERPENDEK



Disusun oleh:

13519063 Melita
13519171 Fauzan Yubairi Indrayadi

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021

BAGIAN 1: KODE PROGRAM

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Tugas_Kecil_3_Stima
{
    public partial class mainForm : Form
    {
        public mainForm()
        {
            InitializeComponent();
        }

        private void eNToolStripMenuItem_Click(object sender, EventArgs e)
        {
            idToolStripMenuItem.Checked = false;
            eNToolStripMenuItem.Checked = true;
            fileToolStripMenuItem.Text = "File";
            exitToolStripMenuItem.Text = "Exit";
            languageToolStripMenuItem.Text = "Language";
            helpToolStripMenuItem.Text = "Help";
            guideToolStripMenuItem.Text = "Guide";
            aboutToolStripMenuItem.Text = "About";
            browseButton.Text = "Browse";
            fromText.Text = "From:";
            toText.Text = "To:";
            shortestText.Text = "Shortest Path:";
            distanceText.Text = "Distance:";
            searchButton.Text = "Search";
        }

        private void idToolStripMenuItem_Click(object sender, EventArgs e)
        {
            eNToolStripMenuItem.Checked = false;
            idToolStripMenuItem.Checked = true;
            fileToolStripMenuItem.Text = "Berkas";
            exitToolStripMenuItem.Text = "Keluar";
            languageToolStripMenuItem.Text = "Bahasa";
            helpToolStripMenuItem.Text = "Bantuan";
            guideToolStripMenuItem.Text = "Panduan";
            aboutToolStripMenuItem.Text = "Tentang";
            browseButton.Text = "Telusuri";
            fromText.Text = "Dari:";
            toText.Text = "Ke:";
            shortestText.Text = "Jalur Terpendek:";
            distanceText.Text = "Jarak:";
            searchButton.Text = "Cari";
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Environment.Exit(0);
        }

        bool[,] createMatriksJalan(string[] isifile2)
```

```

{
    int jumlahNode = Int32.Parse(isifile2[0]);
    bool[,] matriks = new bool[jumlahNode, jumlahNode];

    for (int i = 0; i < jumlahNode; i++)
    {
        string[] tempstring = isifile2[i+1].Split(' ');
        for(int j = 0; j<jumlahNode; j++)
        {
            if(tempstring[j] == "0")
            {
                matriks[i, j] = false;
            }
            else
            {
                matriks[i, j] = true;
            }
        }
    }
    return matriks;
}

float[,] createMatriksBobot(string[] isifile2)
{
    int jumlahNode = Int32.Parse(isifile2[0]);
    float[,] matriks = new float[jumlahNode, jumlahNode];
    for(int i = 0; i<jumlahNode; i++)
    {
        for(int j = 0; j<jumlahNode; j++)
        {
            matriks[i, j] = 0;
        }
    }

    for (int i = 0; i < jumlahNode; i++)
    {
        string[] pos1 = isifile2[i+1+jumlahNode].Split(' ');
        for (int j = i; j < jumlahNode; j++)
        {
            string[] pos2 = isifile2[j + 1 + jumlahNode].Split(' ');
            if (i!=j)
            {
                float lat1 = float.Parse(pos1[0],
System.Globalization.CultureInfo.InvariantCulture);
                float lon1 = float.Parse(pos1[1],
System.Globalization.CultureInfo.InvariantCulture);
                float lat2 = float.Parse(pos2[0],
System.Globalization.CultureInfo.InvariantCulture);
                float lon2 = float.Parse(pos2[1],
System.Globalization.CultureInfo.InvariantCulture);

                matriks[i, j] = haversine(lat1, lon1, lat2, lon2);
                matriks[j, i] = matriks[i, j];
            }
        }
    }
    return matriks;
}

float haversine(float lat1, float lon1, float lat2, float lon2)
{
    var R = 6371;
    var dLat = (lat2 - lat1) * (Math.PI/180);
    var dLon = (lon2 - lon1) * (Math.PI/180);
    var a = Math.Sin(dLat/2) * Math.Sin(dLat/2) + Math.Cos(lat1 *
(Math.PI/180)) * Math.Cos(lat2 * (Math.PI/180)) * Math.Sin(dLon/2) *
Math.Sin(dLon/2);

```

```

        var c = 2 * Math.Atan2(Math.Sqrt(a), Math.Sqrt(1 - a));
        var result = R * c * 1000;
        return (float) result;
    }

    private void browseButton_Click(object sender, EventArgs e)
    {
        // Terima input file, ubah jadi graf
        // Format file .txt:
        // Baris 1          : jumlah simpul (n)
        // Baris 2 sampai n+1 : matriks ketetanggan, 1 berarti ada jalan
        // Baris n+2 sampai 2n+1: matriks jarak, berisi semua jarak dari i ke
        // Baris 2n+2 sampai 3n+1: nama simpul jika diberi nama
        // Jika tidak diberi nama, gunakan angka 1, 2, 3, ..., n

        OpenFileDialog dialog = new OpenFileDialog();
        dialog.Filter = "Text Files|*.txt";

        if (dialog.ShowDialog() == DialogResult.OK)
        {
            textBox2.Text = "";
            textBox3.Text = "";
            // Clear combobox
            comboBox1.Items.Clear();
            comboBox2.Items.Clear();

            // Ubah teks di sebelah browse button
            textBox1.Text = dialog.FileName;

            // Baca isi file
            string isifile = System.IO.File.ReadAllText(textBox1.Text);
            string[] isifile2 = isifile.Split(new[] { Environment.NewLine },
            StringSplitOptions.None);

            // Buat daftar huruf/nama akun
            List<string> daftarHuruf = new List<string>();
            int jumlahNode = Int32.Parse(isifile2[0]);
            if(isifile2.Length == jumlahNode*2+1)
            {
                // Tidak pakai nama
                for(int i = 0; i<jumlahNode; i++)
                {
                    daftarHuruf.Add((i+1).ToString());
                }
            }
            else
            {
                // Pakai nama
                for(int i = 0; i<jumlahNode; i++)
                {
                    daftarHuruf.Add(isifile2[jumlahNode*2+1+i]);
                }
            }

            // Ubah dropdown list
            foreach (var huruf in daftarHuruf)
            {
                comboBox1.Items.Add(huruf);
                comboBox2.Items.Add(huruf);
            }

            comboBox1.SelectedIndex = 0;
            comboBox2.SelectedIndex = 0;

            // Buat graf

```

```

        bool[,] matriksJalan = new bool[jumlahNode, jumlahNode];
        matriksJalan = createMatriksJalan(isifile2);
        float[,] matriksBobot = new float[jumlahNode, jumlahNode];
        matriksBobot = createMatriksBobot(isifile2);

        Microsoft.Msagl.Drawing.Graph graph = new
Microsoft.Msagl.Drawing.Graph("graph");

        // Tambah edge dan node graf
        for(int i = 0; i<jumlahNode; i++)
        {
            for(int j = i; j<jumlahNode; j++)
            {
                if(matriksJalan[i,j]==true)
                {
                    var edge =
graph.AddEdge(comboBox1.Items[i].ToString(), comboBox1.Items[j].ToString());
                    edge.Attr.ArrowheadAtTarget =
Microsoft.Msagl.Drawing.ArrowStyle.None;
                    string elabel = matriksBobot[i, j].ToString() + " m";
                    edge.LabelText = elabel;
                }
            }
        }
        gViewer1.Graph = graph;
    }

    private void searchButton_Click(object sender, EventArgs e)
    {
        int dari = comboBox1.SelectedIndex;
        int ke = comboBox2.SelectedIndex;
        if (dari == ke)
        {
            // Gagal mencari
            MessageBox.Show("Harap pilih 2 simpul yang berbeda.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        else
        {
            // Lakukan pencarian
            findPath(dari, ke);
        }
    }

    float getF(List<int> route, int goal, float[,] matriksBobot)
    {
        float g = 0;
        for(int i=0; i<route.Count-1; i++)
        {
            g += matriksBobot[route[i], route[i + 1]];
        }

        float h = matriksBobot[route[route.Count - 1], goal];

        return g + h;
    }

    void findPath(int dari, int ke)
    {
        // Reset edge graf
        foreach (var edge in gViewer1.Graph.Edges)
        {
            edge.Attr.Color = Microsoft.Msagl.Drawing.Color.Black;
            edge.Attr.LineWidth = 1.0;
        }
    }

```

```

gViewer1.Refresh();

string isifile = System.IO.File.ReadAllText(textBox1.Text);
string[] isifile2 = isifile.Split(new[] { Environment.NewLine },
StringSplitOptions.None);
int jumlahNode = Int32.Parse(isifile2[0]);
bool[,] matriksJalan = new bool[jumlahNode, jumlahNode];
matriksJalan = createMatriksJalan(isifile2);
float[,] matriksBobot = new float[jumlahNode, jumlahNode];
matriksBobot = createMatriksBobot(isifile2);

// Cari jalur terpendek dengan A*
List<List<int>> allPath = new List<List<int>>();
List<int> first = new List<int>();
List<float> allPathF = new List<float>();
first.Add(dari);
allPath.Add(first);
allPathF.Add(getF(first, ke, matriksBobot));
bool found = false;
while(!found && allPath.Count>0)
{
    // Cek rute dengan f minimum
    int minidx = 0;
    for(int i=1;i<allPathF.Count;i++)
    {
        if (allPathF[minidx] > allPathF[i])
        {
            minidx = i;
        }
    }
    List<int> checkRoute = new List<int>(allPath[minidx]);
    allPath.RemoveAt(minidx);
    allPathF.RemoveAt(minidx);
    Console.WriteLine("Minimum found is ");
    for(int k=0;k<checkRoute.Count;k++)
    {
        Console.WriteLine(checkRoute[k]+1);
    }
    Console.WriteLine();

    if (checkRoute[checkRoute.Count-1]==ke)
    {
        found = true;
        allPath.Add(checkRoute);
        allPathF.Add(getF(checkRoute, ke, matriksBobot));
    }
    else
    {
        for (int i = 0; i < jumlahNode; i++)
        {
            if (!checkRoute.Contains(i) && matriksJalan[i,
checkRoute[checkRoute.Count-1]])
            {
                // Jika i belum dikunjungi di rute sekarang
                List<int> addedRoute = new List<int>(checkRoute);
                addedRoute.Add(i);
                allPath.Add(addedRoute);
                allPathF.Add(getF(addedRoute, ke, matriksBobot));
                Console.WriteLine("Added route ");
                Console.WriteLine(i + 1);
                Console.WriteLine("with f = ");
                Console.WriteLine(allPathF[allPathF.Count - 1]);
            }
        }
    }
}

```

```

    }
    if(allPath.Count>0)
    {
        // Jika ketemu jalan
        List<int> finalRoute = new List<int>(allPath[allPath.Count - 1]);
        textBox2.Text = "";
        for (int i = 0; i < finalRoute.Count - 1; i++)
        {
            textBox2.AppendText((finalRoute[i] + 1).ToString());
            textBox2.AppendText(" → ");
        }

        textBox2.AppendText((finalRoute[finalRoute.Count-1] +
1).ToString());
        textBox3.Text = (allPathF[allPathF.Count-1]).ToString();
        textBox3.AppendText(" m");

        // Highlight edge jalan pada graf
        foreach (var edge in gViewer1.Graph.Edges)
        {
            for(int i = 0;i<finalRoute.Count-1;i++)
            {
                if(edge.SourceNode.Id==comboBox1.Items[finalRoute[i]] &&
edge.TargetNode.Id==comboBox1.Items[finalRoute[i+1]] ||
                    edge.SourceNode.Id ==
comboBox1.Items[finalRoute[i+1]] && edge.TargetNode.Id ==
comboBox1.Items[finalRoute[i]])
                {
                    edge.Attr.Color = Microsoft.Msagl.Drawing.Color.Red;
                    edge.Attr.LineWidth = 2.0;
                }
            }
        }

        gViewer1.Refresh();
    }
    else
    {
        // Jika tidak ketemu jalan
        textBox2.Text = "Tidak ada jalan yang ditemukan";
        textBox3.Text = "N/A";
    }
}

private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{
    string msg;
    if (eToolStripMenuItem.Checked == true)
    {
        msg = "Tugas Kecil 3 Strategi Algoritma:\nShortest Path with
A*\n\n13519063 Melita\n13519171 Fauzan Yubairi Indrayadi";
        MessageBox.Show(msg, "About", MessageBoxButtons.OK);
    }
    else
    {
        msg = "Tugas Kecil 3 Strategi Algoritma:\nLintasan Terpendek
dengan A*\n\n13519063 Melita\n13519171 Fauzan Yubairi Indrayadi";
        MessageBox.Show(msg, "Tentang", MessageBoxButtons.OK);
    }
}

private void guideToolStripMenuItem_Click(object sender, EventArgs e)
{
    string msg;
    if(eToolStripMenuItem.Checked==true)
    {
        msg = "1. Choose input file with 'Browse'\n2. Pick a starting

```

```
node and a goal node\n3. Click 'Search';  
    MessageBox.Show(msg, "Guide", MessageBoxButtons.OK);  
}  
else  
{  
    msg = "1. Pilih berkas input dengan 'Telusuri'\n2. Pilih simpul  
awal dan simpul tujuan\n3. Tekan tombol 'Cari';  
    MessageBox.Show(msg, "Panduan", MessageBoxButtons.OK);  
}  
  
    }  
}  
}
```


BAGIAN 2: INPUT GRAF

01.txt (peta sekitar ITB):

```
10
0 1 0 1 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0
0 1 0 0 0 1 0 0 0 0
1 0 0 0 1 0 1 0 0 0
0 1 0 1 0 1 0 0 0 0
0 0 1 0 1 0 0 0 0 1
0 0 0 1 0 0 0 1 0 0
0 0 0 0 0 0 1 0 1 0
0 0 0 0 0 0 0 1 0 1
0 0 0 0 0 1 0 0 1 0
-6.894916 107.608823
-6.894789 107.610150
-6.894738 107.611758
-6.893814 107.608479
-6.893332 107.609959
-6.893624 107.611936
-6.887772 107.608402
-6.887656 107.609972
-6.887430 107.613633
-6.893776 107.612932
A
B
C
D
E
F
G
H
I
J
```

02.txt (peta sekitar Alun-Alun):

```
13
0 1 0 1 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0 1 0 0 0 0
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 1 0 0 0
0 0 0 0 0 0 1 0 1 0 1 0 1
0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0 1 0 1 0
```

```
0 0 0 0 0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 1 0 0 0 1 0
-6.923201 107.603952
-6.923201 107.604590
-6.923467 107.606249
-6.922112 107.604042
-6.922200 107.604731
-6.922416 107.606389
-6.922479 107.607537
-6.921213 107.607844
-6.920845 107.604080
-6.922770 107.609757
-6.921555 107.609949
-6.919680 107.609923
-6.919604 107.608316
```

03.txt (peta sekitar Buahbatu):

```
14
0 1 0 1 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 0 0 0 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 1 0 1 0 0 0 0
0 0 0 0 1 0 1 0 1 0 1 0 0 0
0 0 0 0 0 1 0 1 0 0 0 1 0 0
0 0 0 0 0 0 1 0 0 0 1 0 1 0
0 0 0 0 0 0 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 0 0 0 0 0 0 1 0 1 0
-6.947286 107.659360
-6.947328 107.659599
-6.947427 107.660433
-6.947655 107.659144
-6.947620 107.659601
-6.947717 107.660400
-6.947881 107.658984
-6.947890 107.659590
-6.947946 107.660383
-6.948126 107.658842
-6.948151 107.659565
-6.948189 107.660368
-6.948335 107.658694
-6.948404 107.659544
```

04.txt (peta sekitar GBK):

```
11
0 1 1 0 0 0 0 0 0 0 0
1 0 0 1 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 1 0 1 0 0 0 0
0 0 1 1 0 1 0 0 0 0 0
0 0 0 0 1 0 0 1 1 0 0
0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 0 1 1 0 0 1 0
0 0 1 0 0 1 0 0 0 0 1
0 0 0 0 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 1 1 0
-6.215579 106.798849
-6.218533 106.798870
-6.215557 106.800447
-6.218565 106.801305
-6.217264 106.801799
-6.217312 106.803483
-6.220138 106.802604
-6.218624 106.803913
-6.215624 106.804636
-6.218600 106.806814
-6.215688 106.806782
```

05.txt (peta sekitar SCBD):

```
12
0 1 1 0 0 0 0 0 0 0 0
1 0 1 1 0 0 0 1 0 0 0
1 1 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 1 0 0 0 0
0 0 1 0 0 0 0 0 1 0 0
0 0 0 1 0 0 1 0 0 0 0
0 0 0 1 0 1 0 1 0 1 0
0 1 0 0 0 0 1 0 1 0 1
0 0 1 0 1 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 0 0 0 1 0 1 0
0 0 0 0 0 0 0 1 0 1 0
-6.223142 106.809732
-6.225616 106.809002
-6.225648 106.810612
-6.225819 106.808048
-6.226246 106.812221
-6.226534 106.806492
-6.227111 106.808052
-6.227143 106.808964
-6.227164 106.810563
-6.228199 106.807966
```

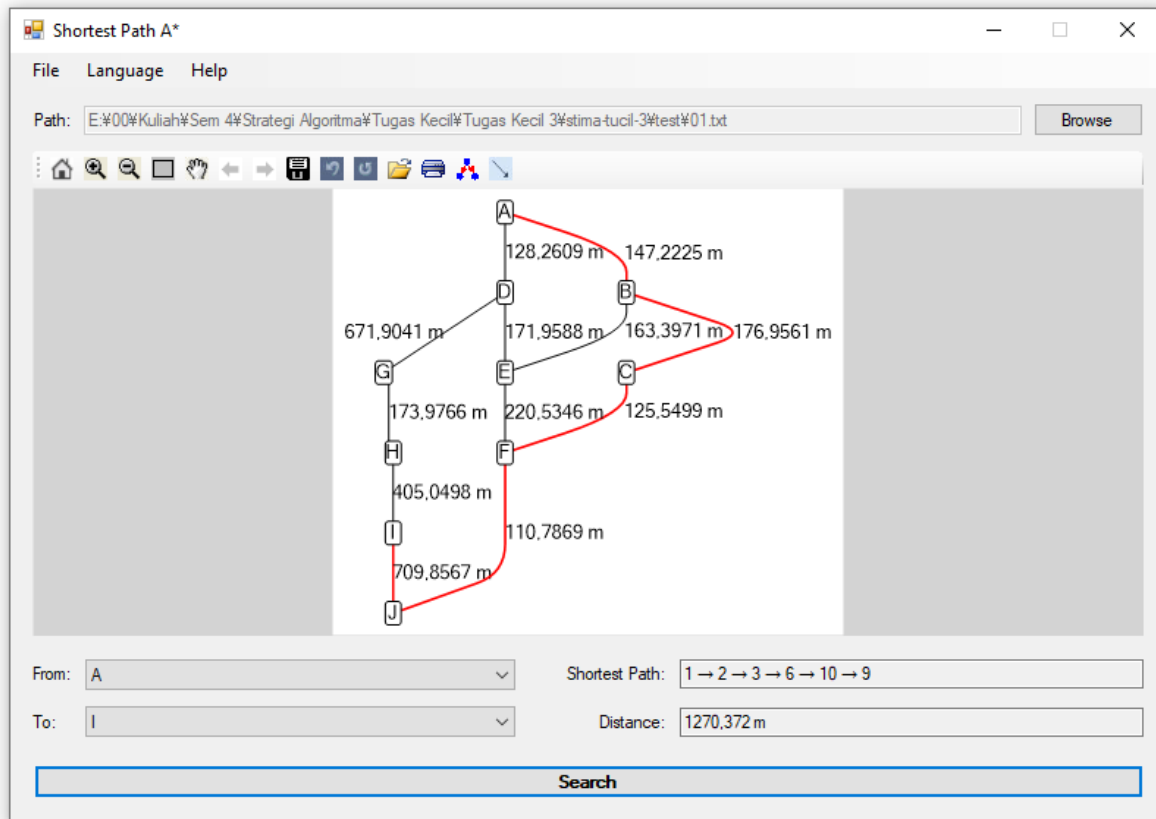
-6.228306 106.808911
-6.229586 106.809705

06.txt (peta sekitar Menteng):

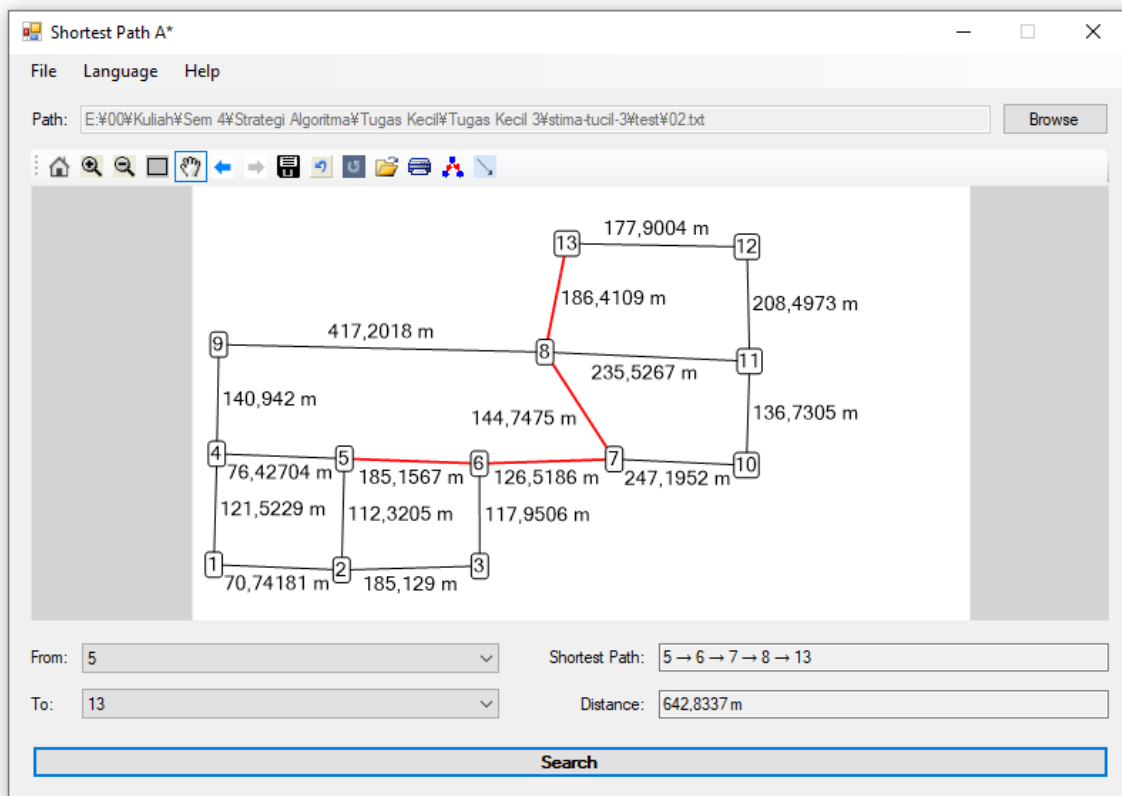
20
0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
-6.195374 106.828848
-6.195620 106.830382
-6.195865 106.832013
-6.196036 106.833064
-6.196420 106.830286
-6.196622 106.831906
-6.196964 106.828633
-6.197145 106.829599
-6.197145 106.830168
-6.197348 106.831820
-6.197668 106.829530
-6.197748 106.830082
-6.197902 106.831702
-6.198366 106.828438
-6.198460 106.829417
-6.198836 106.832137
-6.198484 106.832764
-6.199524 106.828344
-6.200036 106.831954
-6.200158 106.833108

BAGIAN 3: SCREENSHOT LINTASAN TERPENDEK

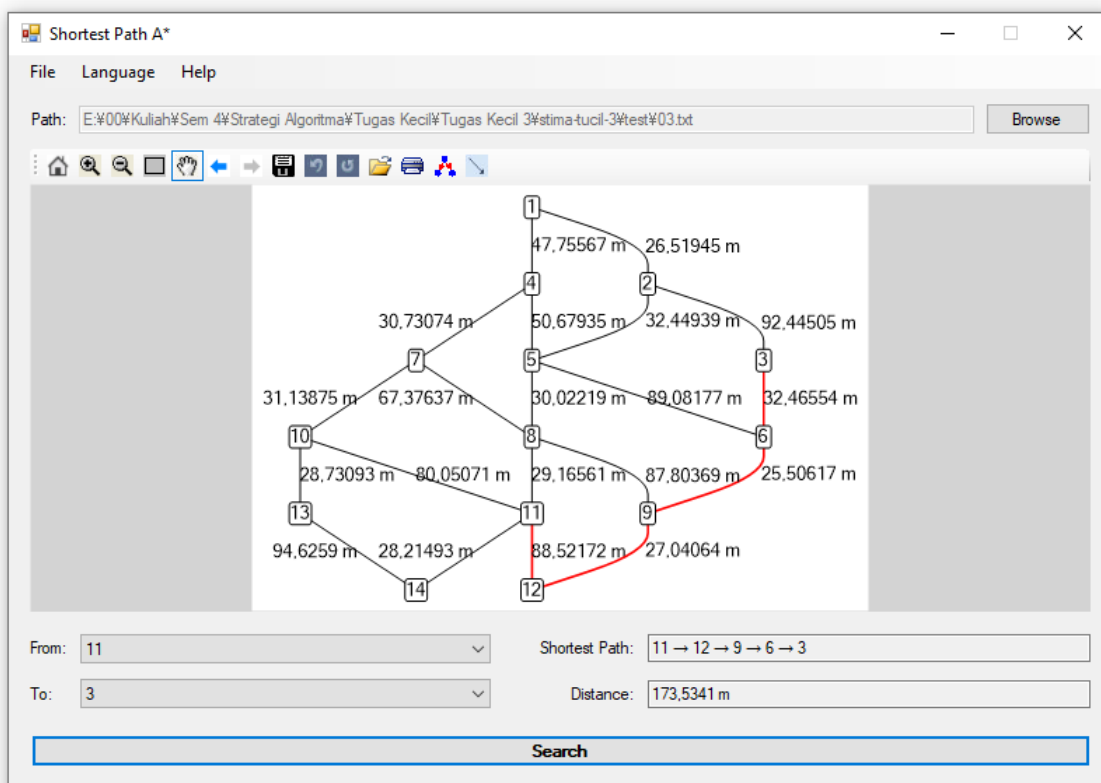
01.txt, simpul A ke I:



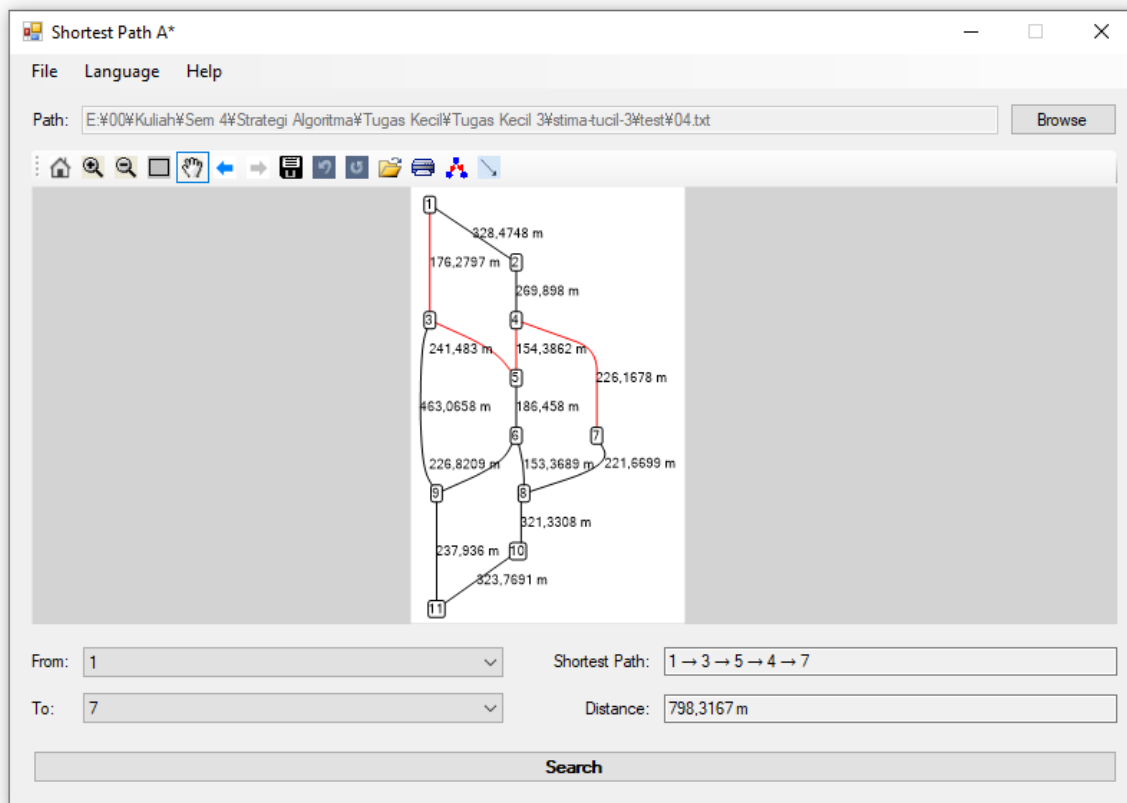
02.txt, simpul 5 ke 13 setelah penyesuaian posisi simpul:



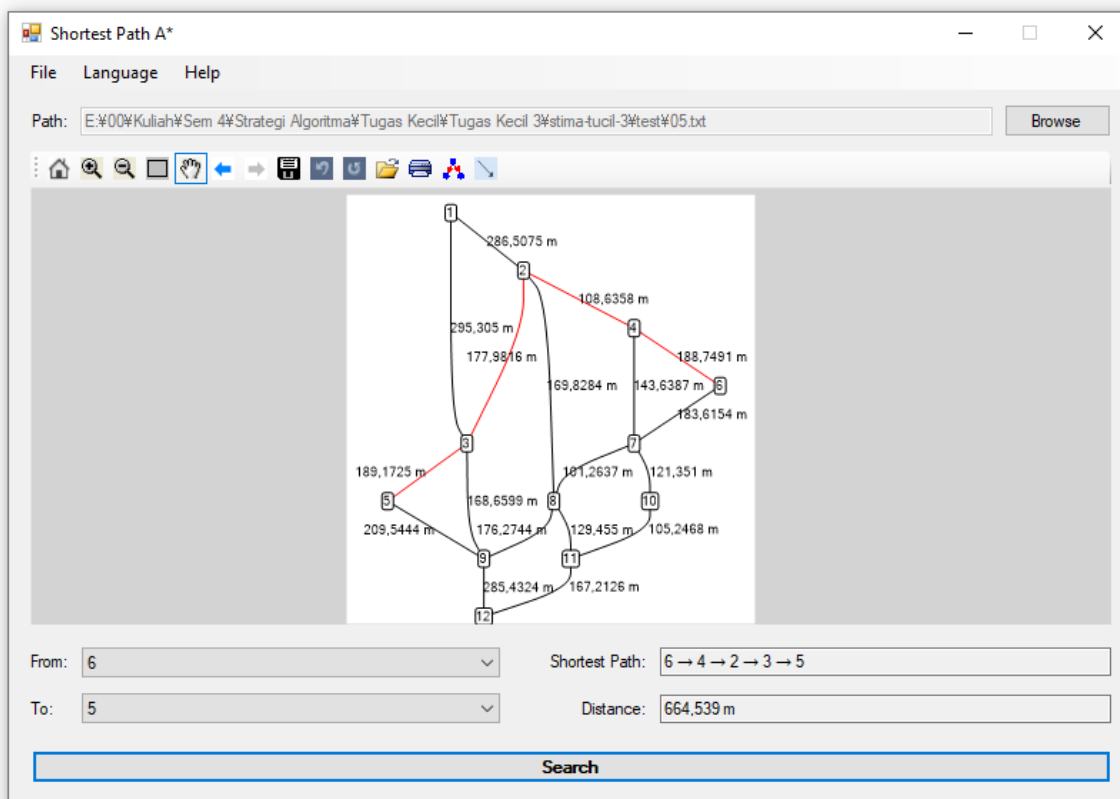
03.txt, simpul 11 ke 3:



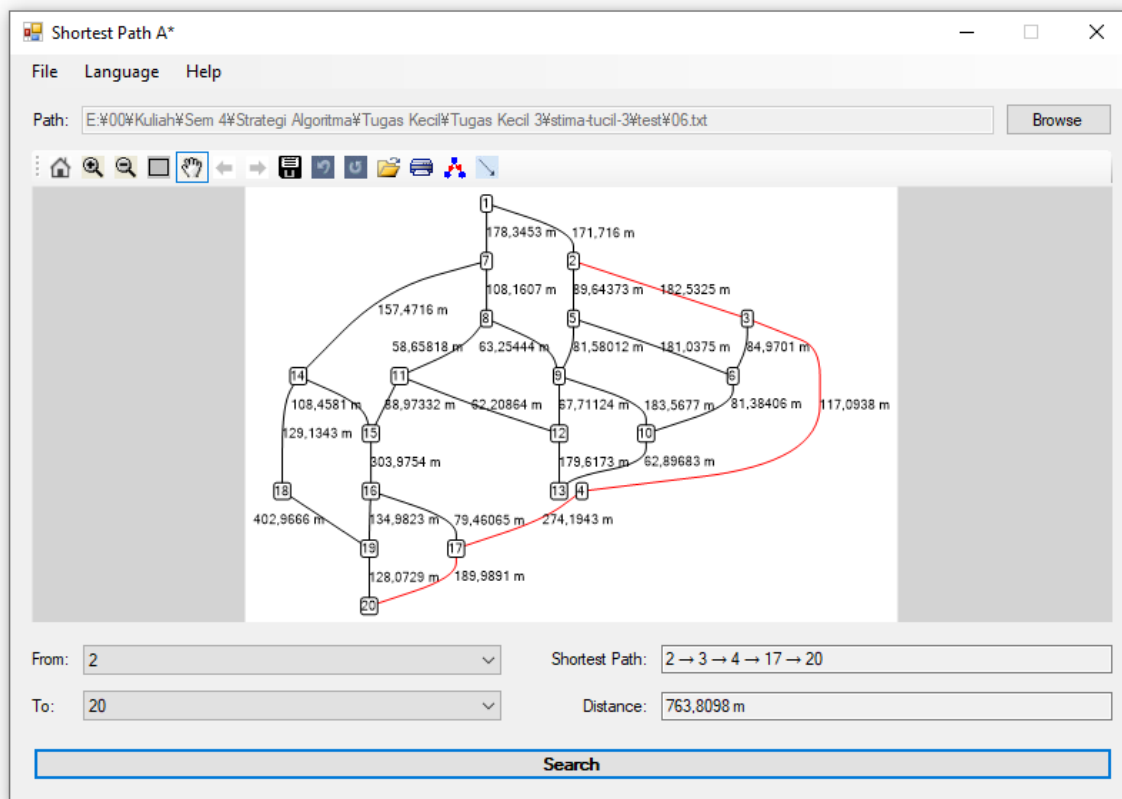
04.txt, simpul 1 ke 7:



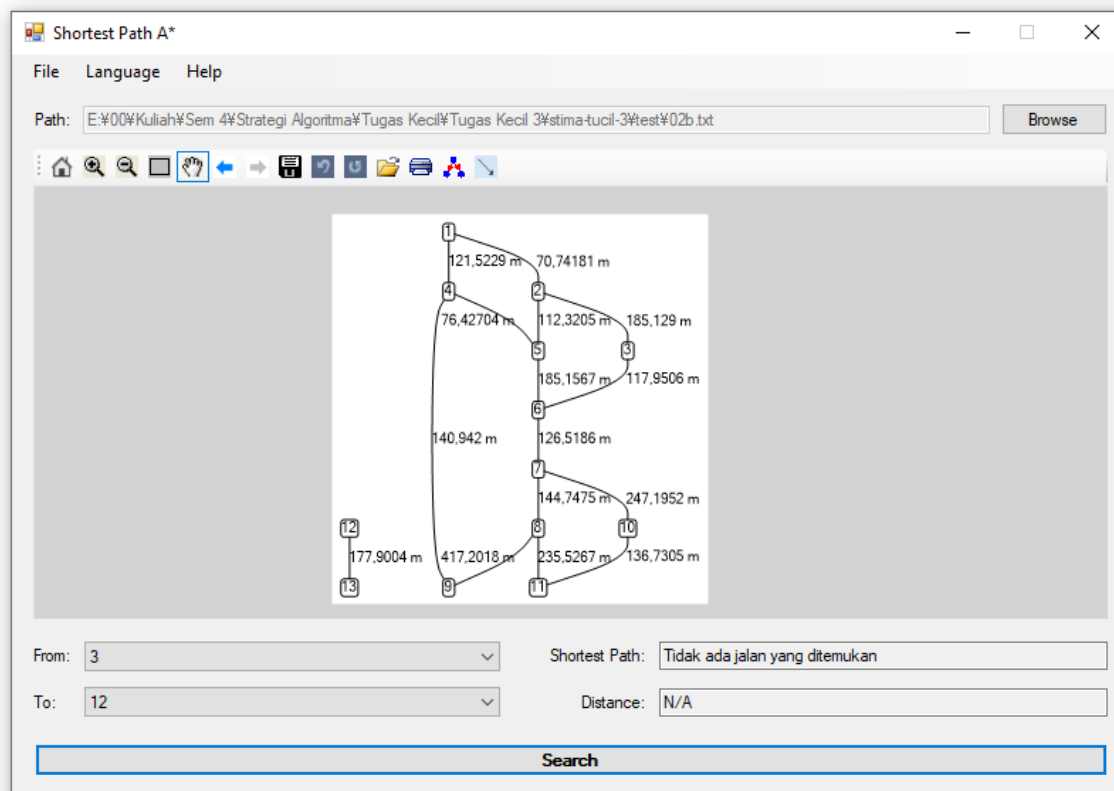
05.txt, simpul 6 ke 5:



06.txt, simpul 2 ke 20:



02.txt, dengan jalan 8-13 dan 11-12 dihilangkan sehingga tak ada jalur:



BAGIAN 4: TABEL PENILAIAN

1	Program dapat menerima input graf	✓
2	Program dapat menghitung lintasan terpendek	✓
3	Program dapat menampilkan lintasan terpendek serta jaraknya	✓
4	Bonus: Program dapat menerima input peta dengan Google Map API dan menampilkan peta	