

Laporan Tugas Kecil 3 IF2211

Strategi Algoritma

Implementasi Algoritma A* untuk Menentukan Lintasan Terpendek



Disusun Oleh:
Reinaldo Antolis 13519015
Rehagana Kevin Christian Sembiring 13519117

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2021

A. Kode Program

Main.py

```
1  from flask import Flask
2  import folium
3  from Tools import * # Proses berawal dari Main Program pada Tools.py
4
5  app = Flask(__name__)
6
7  @app.route("/")
8  def index():
9      # Mencari rata-rata posisi simpul
10     avgLat = sum(node[0] for node in simpul)/len(simpul)
11     avgLon = sum(node[1] for node in simpul)/len(simpul)
12
13     # Membuka peta dengan titik tengah sesuai
14     # rata-rata posisi simpul
15     map = folium.Map(
16         location = [avgLat, avgLon], zoom_start = 100
17     )
18
19     # Menambahkan simpul pada peta
20     for node in simpul :
21         folium.Marker(
22             location = tuple([node[0],node[1]]),
23             popup = "<b>" + node[2] + "\n" + str(node[0]) + "," + str(node[1]) + "\n" + "</b>",
24             tooltip = "Click to show details"
25         ).add_to(map)
26
27     # Menambahkan sisi untuk setiap node yang bertetangga
28     for i in range (nodeCount) :
29         for j in range(i, nodeCount) :
30             tetangga = []
31             if (matrix[i][j] != 0) :
32                 tetangga.append(tuple([simpul[i][0], simpul[i][1]]))
33                 tetangga.append(tuple([simpul[j][0], simpul[j][1]]))
34                 folium.PolyLine(tetangga, color="grey", weight=4, opacity=1).add_to(map)
35
36     # Kasus bila terdapat lintasan terpendek antara simpul asal dan tujuan
37     if (len(shortestPath) != 0) :
38         # Menambahkan simpul yang terdapat dalam lintasan terpendek ke array shortestPathNodes
39         shortestPathNodes = []
40         for node in shortestPath :
41             shortestPathNodes.append(tuple([simpul[searchPos(node)][0], simpul[searchPos(node)][1]]))
42
43     # Menampilkan lintasan terpendek
44     folium.PolyLine(shortestPathNodes, color="blue", weight=4, opacity=1).add_to(map)
45
46     return map._repr_html_()
47
48
49     if __name__ == '__main__':
50         app.run()
```

Tools.py

```
 1  from math import radians, cos, sin, asin, sqrt
 2  import os
 3
 4  def readFile(nama_file) :
 5      simpul = [] # Menyimpan list tuple dengan tuple (latitude, longitude, nama simpul)
 6      matrix = [] # Menyimpan adjacency matrix
 7      # Membaca file
 8      filepath = os.path.join('..\test', nama_file)
 9      with open(filepath) as f:
10          lines = f.readlines()
11
12      lineNumber = 0
13      nodeCount = 0
14      for line in lines:
15          # lineNumber == 0 membaca jumlah node
16          if (lineNumber == 0) :
17              nodeCount = int(line)
18          else :
19              # Kasus membaca simpul
20              if (lineNumber < nodeCount + 1) :
21                  temp = line.split()
22                  temp[0] = float(temp[0])
23                  temp[1] = float(temp[1])
24                  simpul.append(temp)
25
26              # Kasus membaca adjacency matrix
27          else :
28              temp = line.split()
29              for i in range (len(temp)) :
30                  temp[i] = int(temp[i])
31              matrix.append(temp)
32      lineNumber += 1
33      return simpul,matrix,nodeCount
34
35  # Mencari indeks dari suatu nama tempat di array simpul
36  def searchPos(namaTempat) :
37      found = False
38      for i in range (len(simpul)) :
39          if (simpul[i][2] == namaTempat) :
40              return i
41      if(not found) :
42          return -1
43
44  # Mengubah adjacency matrix dari berisi boolean menjadi berisi weighted
45  def changeMatrix() :
46      for i in range(nodeCount) :
47          for j in range(i, nodeCount) :
48              if(matrix[i][j] != 0) :
49                  temp = haversine(searchPos(simpul[i][2]), searchPos(simpul[j][2]))
50                  matrix[i][j] = temp
51                  matrix[j][i] = temp
```

```

52     # Menghitung jarak antara dua titik di bumi dengan
53     # menggunakan garis lintang dan garis bujur
54     def haversine(pos1, pos2):
55         # Mengubah titik desimal menjadi radian
56         lat1, lon1 = simpul[pos1][0], simpul[pos1][1]
57         lat2, lon2 = simpul[pos2][0], simpul[pos2][1]
58         lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
59
60         # Rumus perhitungan haversine
61         # r merupakan radius bumi dalam satuan kilometer
62         r = 6371
63         dlon = lon2 - lon1
64         dlat = lat2 - lat1
65         h = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
66         x = asin(sqrt(h))
67         result = 2 * x * r
68

```

```

70     def aStar(start, goal) :
71         # Inisiasi
72         stack = [] # Stack yang menyimpan jalur yang dilalui (jalur terpendek disimpan disini)
73         visitedNodes = [] # Array boolean, menyimpan nodes yang sudah dilewati
74         currNode = start # Node saat ini
75         stack.append(start)
76         dFromStart = 0 # Jarak dari simpul asal
77         found = False # Boolean yang mengecek apakah sudah mencapai simpul tujuan
78
79         # Loop selama stack tidak kosong dan simpul tujuan belum ditemukan
80         while(len(stack) > 0 and found == False) :
81             # Inisialisasi nilai
82             # nilai fungsi evaluasi paling minimal
83             min = None
84             minPos = None
85             # boolean untuk mengecek apakah merupakan tetangga pertama yang ditemukan
86             isFirst = True
87             i = 0
88             # boolean untuk mengecek apakah suatu node masih memiliki tetangga
89             # yang belum dikunjungi
90             neighbour = False
91

```

```

92 |         # Loop selama jumlah iterasi tidak melebihi jumlah node
93 |         while(i < nodeCount) :
94 |             # Kasus ditemukan sisi yang nilainya tidak nol dan belum pernah dikunjungi
95 |             if ((matrix[searchPos(currNode)][i] != 0) and (simpul[i][2] not in visitedNodes)) :
96 |                 # Kasus sisi yang memenuhi syarat sebelumnya ditemukan pertama kalinya
97 |                 if (isFirst) :
98 |                     isFirst = False
99 |                     minPos = i
100 |                     min = evalFunc(dFromStart, currNode, simpul[i][2], goal)
101 |                     # Kasus sisi ditemukannya sisi setelah sisi pertama
102 |                     else :
103 |                         # Kasus ditemukannya fungsi evaluasi yang lebih minimal
104 |                         if (min > evalFunc(dFromStart, currNode, simpul[i][2], goal)) :
105 |                             minPos = i
106 |                             min = evalFunc(dFromStart, currNode, simpul[i][2], goal)
107 |                         # Menandakan mempunyai tetangga yang belum pernah dikunjungi
108 |                         neighbour = True
109 |                         i+=1
110 |
111 |             # Kasus sudah tidak mempunyai tetangga yang belum pernah dikunjungi
112 |             if (neighbour == False) :
113 |                 # Memasukkan currNode ke boolean visitedNodes (menandakan sudah dikunjungi)
114 |                 if (currNode not in visitedNodes) :
115 |                     visitedNodes.append(currNode)
116 |                     # Melakukan backtrack
117 |                     temp = stack.pop()
118 |                     if (len(stack) > 0) :
119 |                         # Mengurangi jarak dari simpul asal yang dicatat (karena backtrack, jadi mundur ke simpul sebelumnya)
120 |                         dFromStart -= haversine(searchPos(temp), searchPos(stack[len(stack) - 1]))
121 |                         # Backtrack untuk currNode
122 |                         currNode = stack[len(stack) - 1]
123 |                     else :
124 |                         # Push simpul dengan evaluation function paling minimal
125 |                         stack.append(simpul[minPos][2])
126 |                         # Memasukkan currNode ke boolean visitedNodes (menandakan sudah dikunjungi)
127 |                         if (currNode not in visitedNodes) :
128 |                             visitedNodes.append(currNode)
129 |                             # Menambah jarak dari simpul asal
130 |                             dFromStart = gn(dFromStart, currNode, simpul[minPos][2])
131 |                             # Mengubah currNode ke simpul dengan evaluation function paling minimal
132 |                             currNode = simpul[minPos][2]
133 |                             # Kasus simpul tujuan ditemukan
134 |                             if (currNode == goal) :
135 |                                 found = True

```

```

135     # Kasus stack kosong, artinya tidak ditemukan jalur
136     # antara simpul asal dan simpul tujuan
137     if (len(stack) == 0) :
138         # Set jarak dari simpul asal ke simpul tujuan menjadi 0
139         dFromStart = 0
140     return dFromStart, stack
141
142 # Menghitung fungsi evaluasi dengan rumus  $f(n) = g(n) + h(n)$ 
143 def evalFunc (dFromStart, currNode, exploreNode, goal) :
144     fn = gn (dFromStart, currNode, exploreNode) + hn(exploreNode, goal)
145     return fn
146
147 #  $g(n)$  merupakan fungsi yang menghitung total
148 # jarak dari simpul awal menuju exploreNode
149 def gn (dFromStart, currNode, exploreNode) :
150     result = dFromStart + matrix[searchPos(currNode)][searchPos(exploreNode)]
151     return result
152
153 #  $h(n)$  merupakan fungsi yang menghitung jarak
154 # garis lurus dari exploreNode menuju simpul tujuan
155 def hn (exploreNode, goal) :
156     return haversine (searchPos(exploreNode), searchPos(goal))
157
158 # Main Program
159 # Input file kasus uji
160 kasusUji = input("Masukkan nama file kasus uji: ")
161 # Mengolah input kasus uji
162 simpul, matrix, nodeCount = readFile(kasusUji)
163 # Mencetak semua simpul yang ada
164 print("List simpul: ")
165 no = 1
166 for node in simpul :
167     print(no, end="")
168     print(".", end="")
169     print(node[2])
170     no += 1
171 # Mengubah adjacency matrix yang berisi boolean menjadi
172 # berisi bobot sisi antar simpul
173 changeMatrix()
174 #Menerima input simpul asal dan simpul tujuan
175 startNode = int(input("No simpul asal: "))
176 goalNode = int(input("No simpul tujuan: "))
177
178 # Menjalankan algoritma A*
179 distance, shortestPath = aStar(simpul[startNode - 1][2], simpul[goalNode - 1][2])
180 # Kasus terdapat lintasan terpendek antara simpul asal dan tujuan
181 if (len(shortestPath) != 0) :
182     print("Jarak lintasan terpendek antara " + simpul[startNode - 1][2] + " dan " + simpul[goalNode - 1][2]
183          + " adalah " + str(distance) + " km")
184
185 # Kasus tidak terdapat lintasan antara simpul asal dan tujuan
186 else :
187     print("Tidak terdapat lintasan antara " + simpul[startNode - 1][2] + " dan " + simpul[goalNode - 1][2])

```

B. Peta/Graf Input

AlunAlun.txt

```
12
-6.922098759206205 107.60402119627975 Jl.Cibadak
-6.920801038813799 107.604147410824 Jl.Asia-Afrika
-6.923162138646535 107.60395758620203 Simpang_Karanganyar
-6.923848037422382 107.60384603986122 Kalipah_Apo
-6.92403099543224 107.60454049698961 Jl.Simpang
-6.923186403129165 107.60463923573735 Jl.Kapatihan
-6.922168612659663 107.60474039830164 Jl.Dalem-Kaum
-6.922416146338085 107.60642037614292 Menara_Mesjid_Raya_Bandung
-6.92341726980283 107.60625409980602 BRI_Dewi_Sartika
-6.924169895397717 107.60619834337999 Yoshihoya
-6.924375906493502 107.6072841350452 SDN_Balonggede
-6.922595677102735 107.60763242426096 Alun-Alun
0 1 1 0 0 0 1 0 0 0 0 0
1 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 1 0 1 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 0 1 0 0
0 0 1 0 1 0 1 0 1 0 0 0
1 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 1
0 0 0 0 0 1 0 1 0 1 0 0
0 0 0 0 1 0 0 0 1 0 1 0
0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
```

BuahBatu.txt

```
9
-6.937570 107.624031 Bandung_Eye_Center
-6.937938 107.625812 Masjid
-6.940041 107.625216 BCA
-6.939487 107.627190 SMKN3_Bandung
-6.937991 107.619385 The_Dreams_Cake
-6.939679 107.621595 Jl.Radio
-6.944876 107.630071 The_Harvest
-6.943758 107.630774 91COFFEE
-6.946176 107.630870 BankMualamat
```

```
0 1 0 0 1 0 0 0 1
1 0 1 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0
0 0 1 0 0 0 0 0 1
1 0 0 0 0 1 0 0 0
0 0 0 0 1 0 1 0 0
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1
1 0 0 1 0 0 0 1 0
```

ITB.txt

```
10
-6.894788167404735 107.61017376801503 Jl.Gelap-Nyawang
-6.894885626313626 107.6088552107555 Jl.Taman-Sari
-6.894757942430549 107.61172971738506 BumiMedika
-6.893597937264786 107.61194598979635 Parkir_Timur
-6.893233130948845 107.61038442446555 Monumen_Kubus
-6.893324732021865 107.6098496087129 Jl.Ganesa
-6.893882418057512 107.60845458443058 Kebun_Binatang
-6.892649285227788 107.61043948779529 FSRD
-6.896881681403216 107.60966307582477 Pelesiran
-6.897638254596191 107.61148376439564 SMA_Alfa_Centuri

0 1 1 0 1 1 0 0 0 0
1 0 0 0 0 0 1 0 1 0
1 0 0 1 0 0 0 0 0 1
0 0 1 0 1 0 0 0 0 0
1 0 0 1 0 1 0 1 0 0
1 0 0 0 1 0 1 0 0 0
0 1 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
```

Medan.txt

```
12
3.591287 98.684146 Monumen_Medan_Area
3.593001 98.688888 Simpang_Jl.Veteran
3.593407 98.690112 Jl.Malaka
3.593793 98.691506 Simpang_Jl.Malaka
3.590859 98.689639 Yayasan_Perguruan_Sutomo
3.591352 98.690863 Jl.HOS_Cokroaminoto
3.591887 98.692193 Jl.Tapanuli
3.587127 98.685675 Medan_Mall
3.589001 98.690374 Methodist_2
3.589269 98.691061 Thamrin_Plaza
3.589547 98.691673 BCA_KCP_Sumatera
3.590029 98.692917 Simpang_Jl.Tapanuli

0 1 0 0 0 0 0 1 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 0 0
0 1 0 0 0 1 1 0 1 0 0 0
0 0 1 0 1 0 1 0 0 0 1 0
0 0 0 1 1 1 0 0 0 0 0 1
1 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 1 0 1 0 0
0 0 0 0 0 0 0 0 1 0 1 0
0 0 0 0 0 1 0 0 0 1 0 1
0 0 0 0 0 0 1 0 0 0 1 0
```

Palembang.txt

```
14
-2.9803734917493045 104.75672319353404 Sudirman_City_Center
-2.9788462497983574 104.75892672567176 Lapangan_Hatta_Kiri_Depan
-2.9775991001469677 104.7584451263536 Lapangan_Hatta_Kanan_Depan
-2.977271240949234 104.7593979138422 Lapangan_Hatta_Kanan_Belakang
-2.9784605343720503 104.75983571719992 Lapangan_Hatta_Kiri_Belakang|
-2.9782869619381502 104.76033360210506 Dempo_Car_Area
-2.9819721937568 104.75863361219454 Martabak_Pulau_Mas
-2.9813357637663827 104.75959498294843 Gereja
-2.9804912676867543 104.76119364649877 Lingkaran_Park
-2.9779926846319946 104.760099415706004 Jl.Air_Mancur
-2.9771676796351474 104.75977971966459 Jl.Gunung_Seminung
-2.9762826735178964 104.7591339105803 Jl.Menumbung
-2.97642410347436 104.75803534481075 Universitas_Taman_Siswa
-2.9756569548379765 104.75791310312262 Jl.Veteran
0 1 0 0 0 1 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 0 0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0 0 0 0 1 1 0 0
0 1 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 1 1 0 0 0
1 0 0 0 0 0 0 1 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 0 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 0 0 0 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 0 0 0 0 0 1 0
```

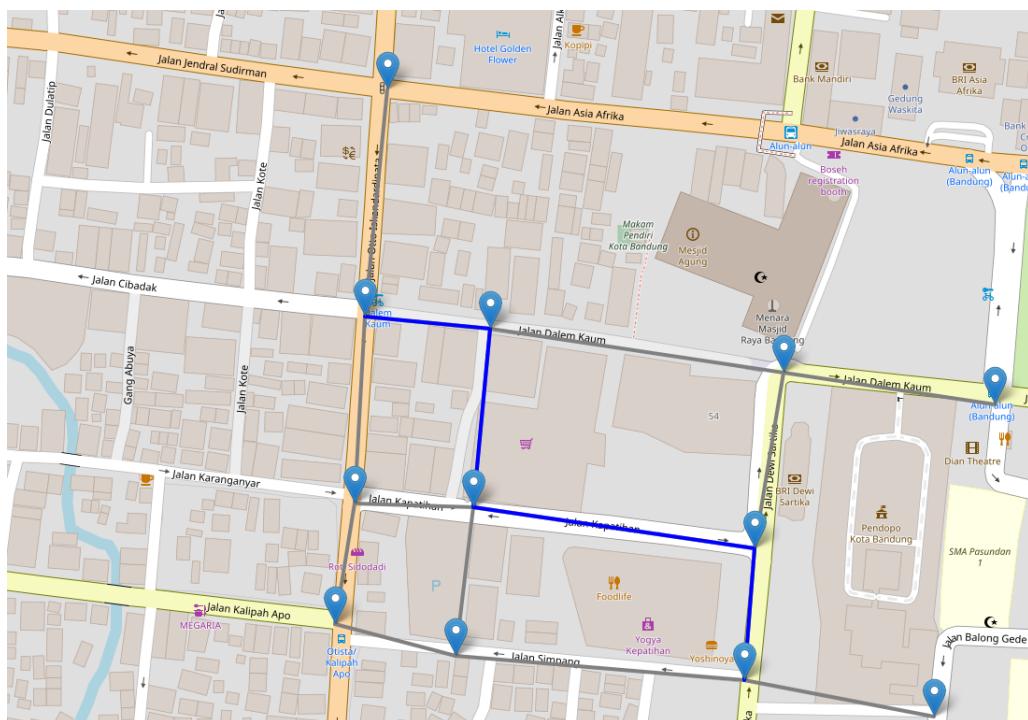
Pluit.txt

```
16
-6.125727 106.799537 RS_Pluit
-6.125551 106.797177 SPBU_Shell
-6.126965 106.797258 Food_Court_35
-6.126218 106.792961 RS_Atma_Jaya
-6.126960 106.793009 Jl.Pluit_Raya
-6.126330 106.789399 Bank_Mandiri
-6.125375 106.790944 Gereja_Bethel
-6.122223 106.791663 Jakarta_Fruit_Market
-6.122095 106.789157 Bank_Dinar
-6.122175 106.794237 Jl.Pluit_Timur_Raya
-6.124138 106.794248 Aston_Pluit
-6.120282 106.788197 MNC_Bank
-6.122276 106.788197 BRI
-6.123743 106.788492 Apotik_Kencana_Raya
-6.120186 106.790418 Jl.Taman_Pluit_Kencana_Utara
-6.122239 106.790450 Jl.Pluit_Sakti
0 1 0 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 0 0 0
1 1 0 0 1 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 1 1 0 1 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 1 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 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
0 1 0 0 0 0 0 0 0 1 0 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 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 0 0 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0
```

C. Screenshot Peta dan Lintasan Terpendek

AlunAlun.txt

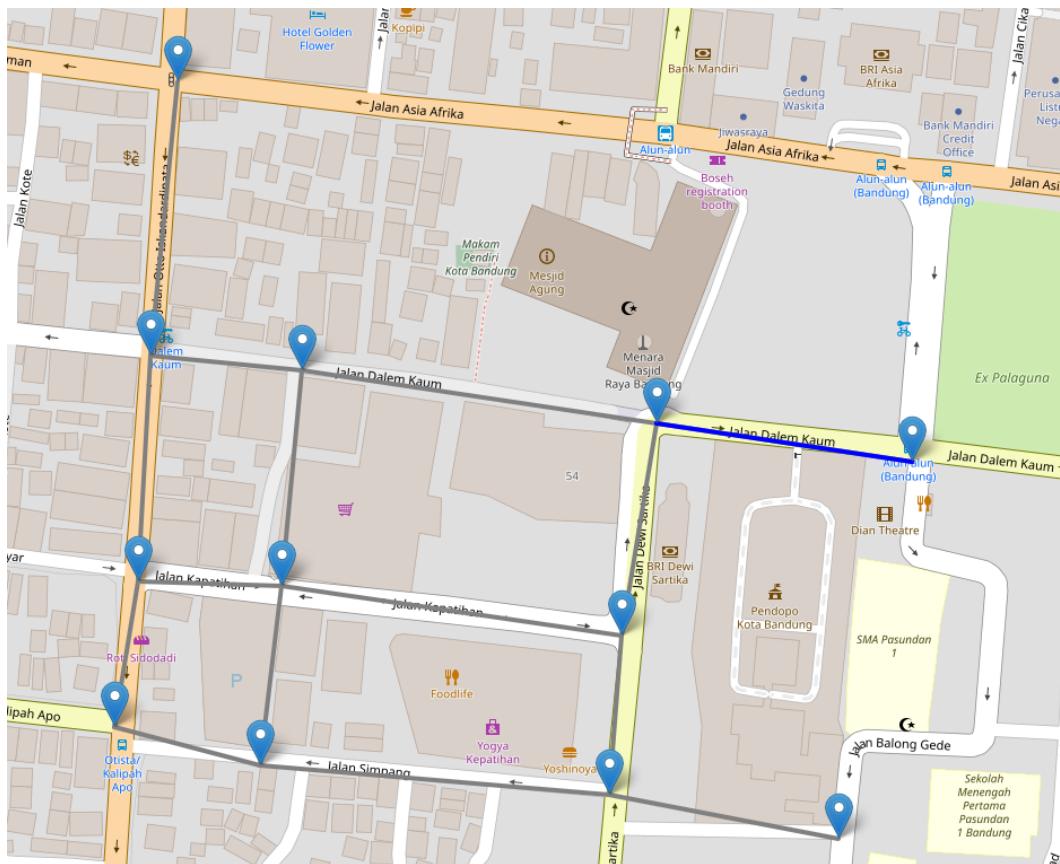
```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: AlunAlun.txt
List simpul:
1.Jl.Cibadak
2.Jl.Asia-Afrika
3.Simpang_Karanganyar
4.Kalipah_Apo
5.Jl.Simpang
6.Jl.Kapatihan
7.Jl.Dalem_Kaum
8.Menara_Mesjid_Raya_Bandung
9.BRI_Dewi_Sartika
10.Yoshinoya
11.SDN_Balonggede
12.Alun-Alun
No simpul asal: 1
No simpul tujuan: 10
Jarak lintasan terpendek antara Jl.Cibadak dan Yoshinoya adalah 0.45749902388802083 km
```



```

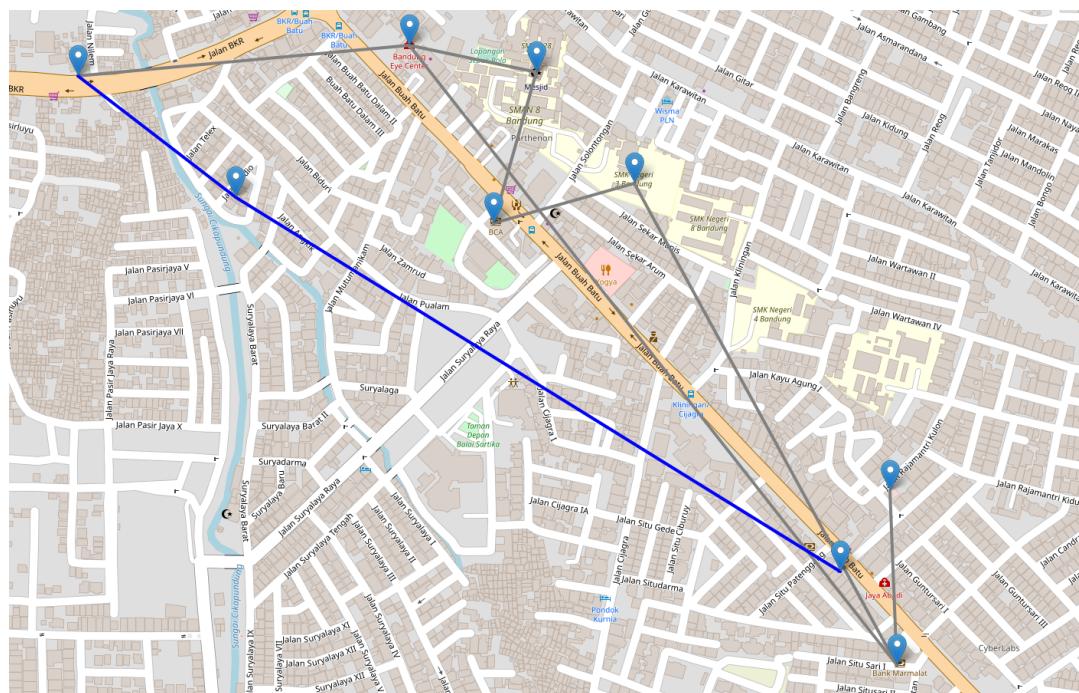
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: AlunAlun.txt
List simpul:
1.Jl.Cibadak
2.Jl.Aisia-Afrika
3.Simpang_Karanganyar
4.Kalipah_Apo
5.Jl.Simpang
6.Jl.Kapatihan
7.Jl.Dalem-Kaum
8.Menara_Mesjid_Raya_Bandung
9.BRI_Dewi_Sartika
10.Yoshinoya
11.SDN Balonggede
12.Alun-Alun
No simpul asal: 8
No simpul tujuan: 12
Jarak lintasan terpendek antara Menara Mesjid Raya Bandung dan Alun-Alun adalah 0.1352722420884711 km

```

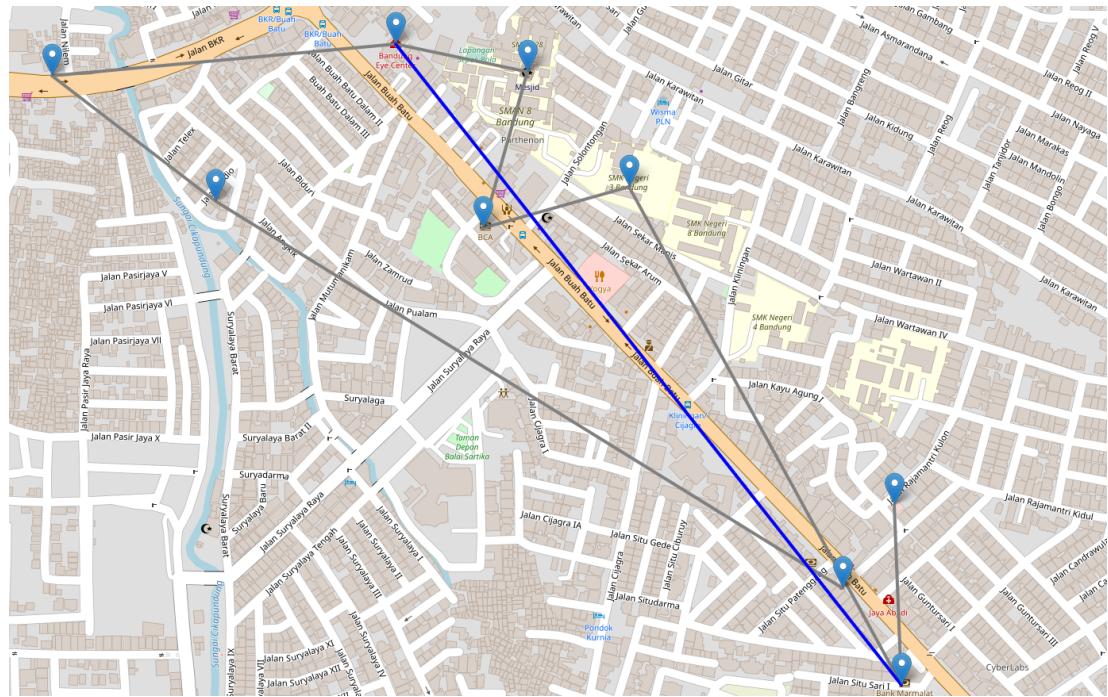


BuahBatu.txt

```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: BuahBatu.txt
List simpul:
1.Bandung_Eye_Center
2.Masjid
3.BCA
4.SMKN3_Bandung
5.The_Dreams_Cake
6.Jl.Radio
7.The_Harvest
8.91COffee
9.BankMualamat
No simpul asal: 7
No simpul tujuan: 5
Jarak lintasan terpendek antara The_Harvest dan The_Dreams_Cake adalah 1.4074542850313814 km
```

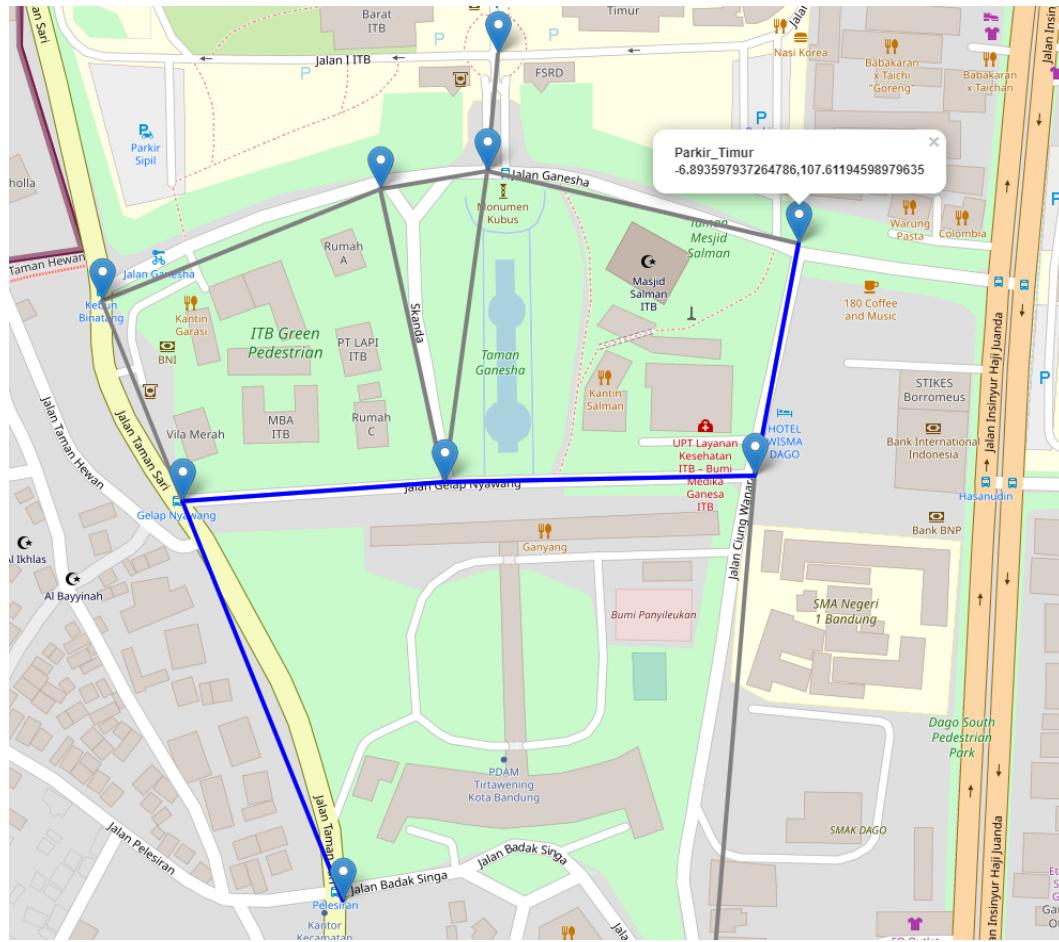


```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: BuahBatu.txt
List simpul:
1.Bandung_Eye_Center
2.Masjid
3.BCA
4.SMKN3_Bandung
5.The_Dreams_Cake
6.Jl.Radio
7.The_Harvest
8.91COffee
9.BankMualamat
No simpul asal: 1
No simpul tujuan: 9
Jarak lintasan terpendek antara Bandung_Eye_Center dan BankMualamat adalah 1.218850225544789 km
```



ITB.txt

```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: ITB.txt
List simpul:
1.Jl.Gelap-Nyawang
2.Jl.Taman-Sari
3.BumiMedika
4.Parkir_Timur
5.Monumen_Kubus
6.Jl.Ganesa
7.Kebun_Binatang
8.FSRD
9.Pelesiran
10.SMA_Alfa_Centuri
No simpul asal: 4
No simpul tujuan: 9
Jarak lintasan terpendek antara Parkir_Timur dan Pelesiran adalah 0.6881300487365367 km
```



D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py

Masukkan nama file kasus uji: ITB.txt

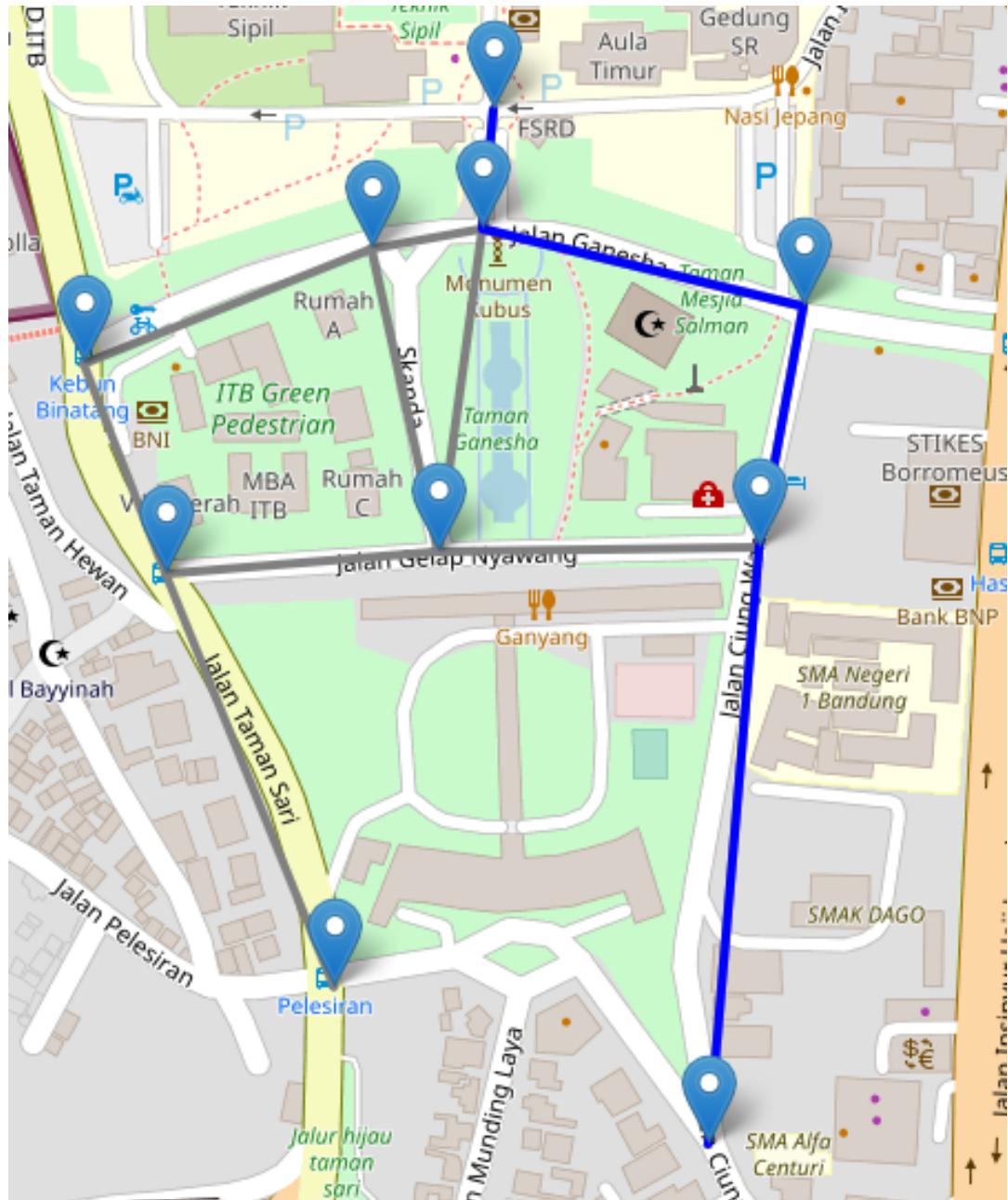
List simpul:

- 1.Jl.Gelap-Nyawang
- 2.Jl.Taman-Sari
- 3.BumiMedika
- 4.Parkir_Timur
- 5.Monumen_Kubus
- 6.Jl.Ganesa
- 7.Kebun_Binatang
- 8.FSRD
- 9.Pelesiran
- 10.SMA_Alfa_Centuri

No simpul asal: 10

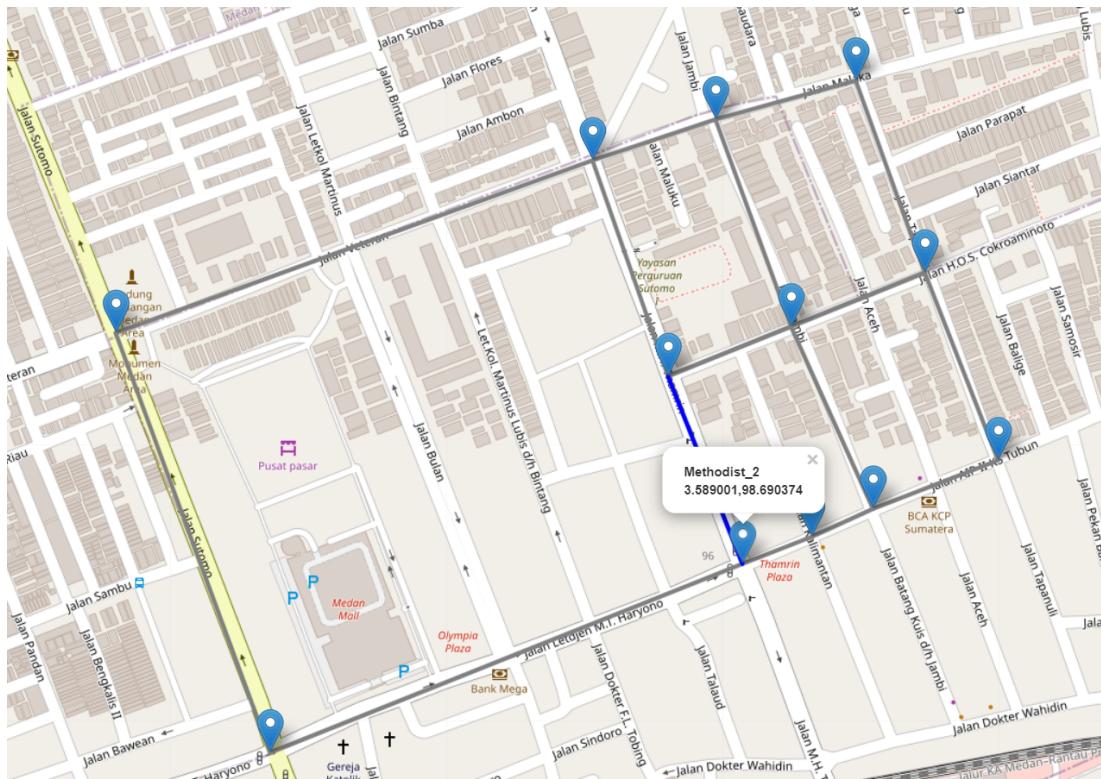
No simpul tujuan: 8

Jarak lintasan terpendek antara SMA_Alfa_Centuri dan FSRD adalah 0.6948984854647153 km

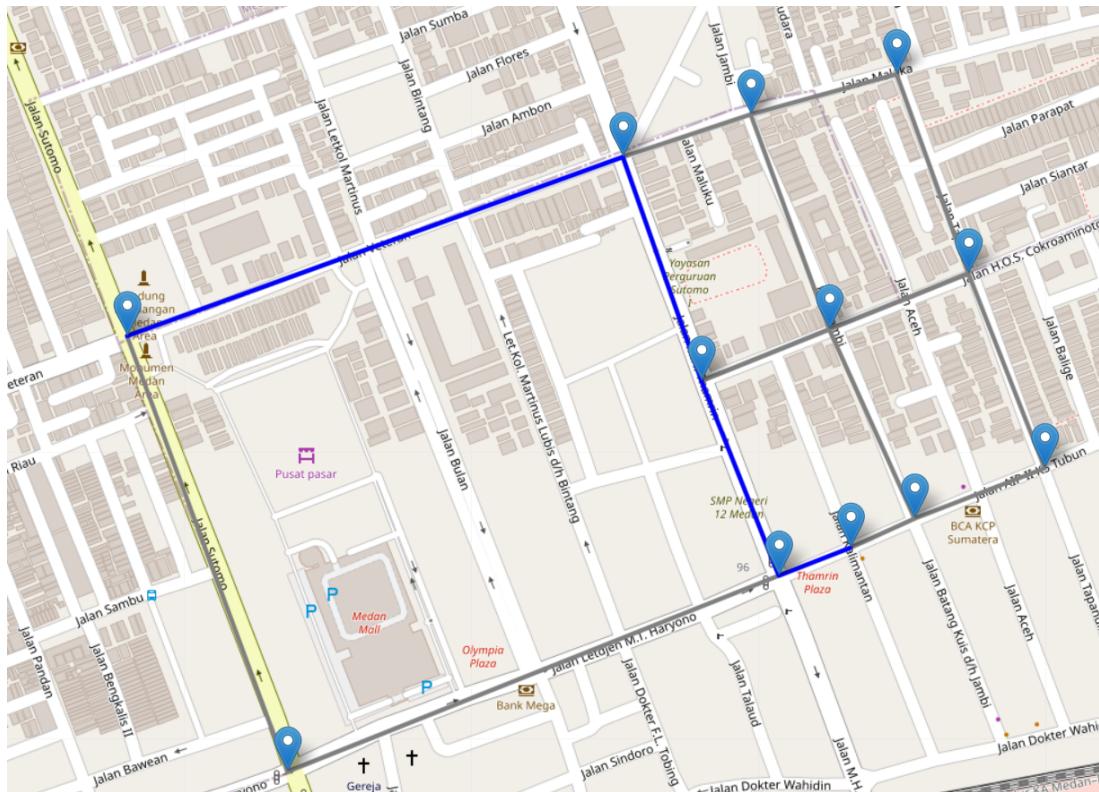


Medan.txt

```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: Medan.txt
List simpul:
1.Monumen_Medan_Area
2.Simpang_Jl.Veteran
3.Jl.Malaka
4.Simpang_Jl.Malaka
5.Yayasan_Perguruan_Sutomo
6.Jl.HOS_Cokroaminoto
7.Jl.Tapanuli
8.Medan_Mall
9.Methodist_2
10.Thamrin_Plaza
11.BCA_KCP_Sumatera
12.Simpang_Jl.Tapanuli
No simpul asal: 5
No simpul tujuan: 9
Jarak lintasan terpendek antara Yayasan_Perguruan_Sutomo dan Methodist_2 adalah 0.22211923382050464 km
```



```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: Medan.txt
List simpul:
1.Monumen_Medan_Area
2.Simpang_Jl.Veteran
3.Jl.Malaka
4.Simpang_Jl.Malaka
5.Yayasan_Perguruan_Sutomo
6.Jl.HOS_Cokroaminoto
7.Jl.Tapanuli
8.Medan_Mall
9.Methodist_2
10.Thamrin_Plaza
11.BCA_KCP_Sumatera
12.Simpang_Jl.Tapanuli
No simpul asal: 1
No simpul tujuan: 10
Jarak lintasan terpendek antara Monumen_Medan_Area dan Thamrin_Plaza adalah 1.1160169382536944 km
```

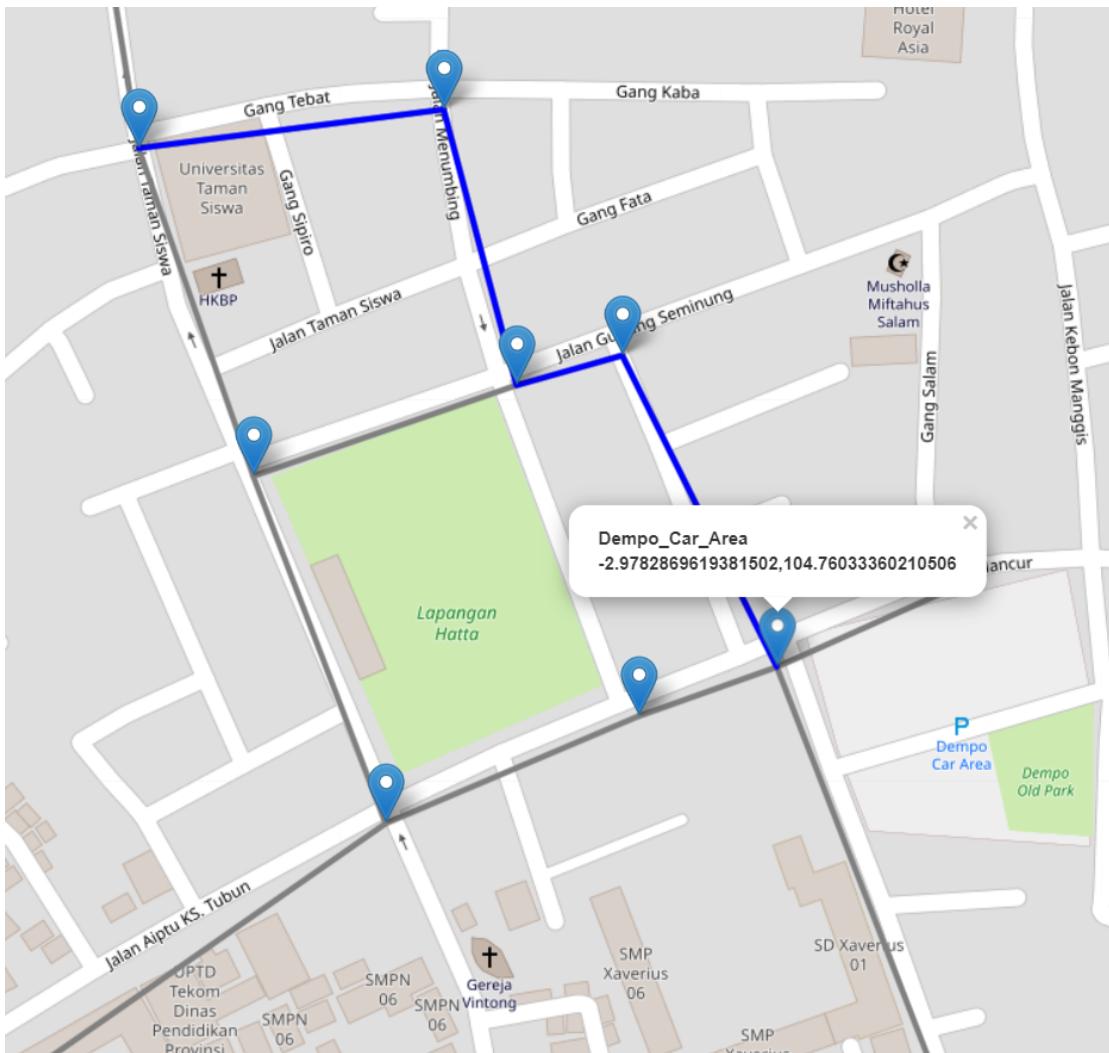


Palembang.txt

```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: Palembang.txt
List simpul:
1.Sudirman_City_Center
2.Lapangan_Hatta_Kiri_Depan
3.Lapangan_Hatta_Kanan_Depan
4.Lapangan_Hatta_Kanan_Belakang
5.Lapangan_Hatta_Kiri_Belakang
6.Dempo_Car_Area
7.Martabak_Pulau_Mas
8.Gereja
9.Lingkaran_Park
10.Jl.Air_Mancur
11.Jl.Gunung_Seminung
12.Jl.Menumbung
13.Universitas_Taman_Siswa
14.Jl.Veteran
No simpul asal: 1
No simpul tujuan: 9
Jarak lintasan terpendek antara Sudirman_City_Center dan Lingkaran_Park adalah 0.7291354658341226 km
```



```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: Palembang.txt
List simpul:
1.Sudirman_City_Center
2.Lapangan_Hatta_Kiri_Depan
3.Lapangan_Hatta_Kanan_Depan
4.Lapangan_Hatta_Kanan_Belakang
5.Lapangan_Hatta_Kiri_Belakang
6.Dempo_Car_Area
7.Martabak_Pulau_Mas
8.Gereja
9.Lingkaran_Park
10.Jl.Air Mancur
11.Jl.Gunung_Seminung
12.Jl.Menumbing
13.Universitas_Taman_Siswa
14.Jl.Veteran
No simpul asal: 6
No simpul tujuan: 13
Jarak lintasan terpendek antara Dempo_Car_Area dan Universitas Taman Siswa adalah 0.41952584057401743 km
```



Pluit.txt

```
D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: Pluit.txt
List simpul:
1.RS_Pluit
2.SPBU_Shell
3.Food_Court_35
4.RS_Atma_Jaya
5.Jl.Pluit_Raya
6.Bank_Mandiri
7.Gereja_Bethel
8.Jakarta_Fruit_Market
9.Bank_Dinar
10.Jl.Pluit_Timur_Raya
11.Aston_Pluit
12.MNC_Bank
13.BRI
14.Apotik_Kencana_Raya
15.Jl.Taman_Pluit_Kencana_Utara
16.Jl.Pluit_Sakti
No simpul asal: 1
No simpul tujuan: 4
Jarak lintasan terpendek antara RS_Pluit dan RS_Atma_Jaya adalah 0.8395642392751396 km
```



D:\Semester 4\Stigma\Tucil-III-Stima\src>python Main.py
Masukkan nama file kasus uji: Pluit.txt

List simpul:

- 1.RS_Pluit
- 2.SPBU_Shell
- 3.Food_Court_35
- 4.RS_Atma_Jaya
- 5.Jl.Pluit_Raya
- 6.Bank_Mandiri
- 7.Gereja_Bethel
- 8.Jakarta_Fruit_Market
- 9.Bank_Dinar
- 10.Jl.Pluit_Timur_Raya
- 11.Aston_Pluit
- 12.MNC_Bank
- 13.BRI
- 14.Apotik_Kencana_Raya
- 15.Jl.Taman_Pluit_Kencana_Utara
- 16.Jl.Pluit_Sakti

No simpul asal: 9

No simpul tujuan: 7

Jarak lintasan terpendek antara Bank_Dinar dan Gereja_Bethel adalah 0.63681875243891 km



D. Link Menuju Kode Program

<https://github.com/ReinaldoAntolis/Tucil-III-Stima>

E. Checklist:

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	