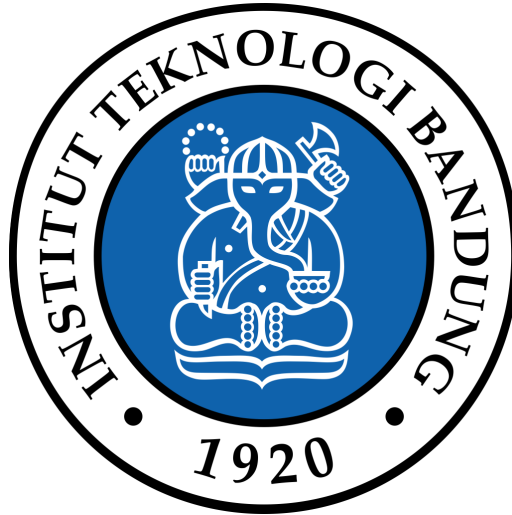


# **LAPORAN TUGAS KECIL 2 STRATEGI ALGORITMA**

Implementasi Algoritma A\* untuk Menentukan Lintasan Terpendek



Oleh :

Rhapsodya Piedro Asmorobangun (13519084)

Made Kharisma Jagaddhita (13519176)

**Institut Teknologi Bandung  
2021**

## A. Algoritma A\*

Algoritma A\* juga merupakan algoritma traversal graf terinformasi sehingga algoritma ini memakai graf dengan bobot (weighted graph). Pencarian dimulai dari node awal tertentu dari grafik. Tujuan pencarian adalah untuk menemukan jalur ke node tujuan tertentu yang memiliki *cost* terkecil (jarak tempuh terkecil, waktu terpendek, dll.). Berikut merupakan langkah-langkah dari algoritma A\*.

1. Inisialisasi open list dan closed list. Open list nantinya akan berisi simpul-simpul yang masih hidup. Closed list akan berisi simpul-simpul yang telah dikunjungi.
2. Masukkan simpul awal ke dalam open list.
3. Iterasi semua simpul pada open list dan pilih simpul yang memiliki estimasi *cost* paling kecil. Perhitungan simpul yang ingin diekspansi menggunakan rumus  $f(n) = g(n) + h(n)$  dimana  $f(n)$  adalah estimasi *cost* dengan melalui jalur  $n$  ke tujuan,  $g(n)$  adalah *cost* sejauh ini untuk mencapai  $n$ , dan  $h(n)$  adalah perkiraan biaya  $n$  ke tujuan. Perhitungan  $h(n)$  dapat dilakukan secara heuristik. Dalam tugas ini akan menggunakan jarak euclidean.
4. Masukkan simpul yang dipilih tadi ke dalam closed list.
5. Ekspansi simpul yang dipilih. jika hasil ekspansi dari simpul tidak berada di closed list, maka masukkan simpul tersebut ke open list.
6. Ulangi langkah 3 sampai 5. Jika simpul yang diekspansi ternyata merupakan simpul tujuan, maka path telah ditemukan. Jika tidak ada lagi simpul di open list dan path belum ditemukan, maka simpul tujuan tidak dapat tercapai dari simpul awal.

Algoritma A\* memiliki kompleksitas eksponensial ( $O(b^m)$ ), menggunakan ruang sebesar ( $O(b^m)$ ), dan optimal. Algoritma A\* akan admissible jika untuk setiap simpul  $n$ ,  $h(n) \leq h^*(n)$  dimana  $h^*(n)$  adalah *cost* sebenarnya (bukan estimasi) untuk mencapai simpul tujuan dari simpul  $n$

## B. Source code

Source code dapat diakses pada link ini: [https://github.com/kharismajd/Tucil3\\_Stima](https://github.com/kharismajd/Tucil3_Stima)

file graph.py

```
1 from math import radians, cos, sin, asin, sqrt
2 import copy
3
4 class Coordinate:
5     # Private
6     __lat = None # Float
7     __long = None # Float
8
9     # Public
10    def __init__(self, lat, long):
11        self.__lat = lat
12        self.__long = long
13
14    def getLat(self):
15        return self.__lat
16
17    def getLong(self):
18        return self.__long
19
20    def setLat(self, lat):
21        self.__lat = lat
22
23    def setLong(self, long):
24        self.__long = long
25
26 class Vertice:
27     # Private
28     __name = None # String
29     __coordinate = None # Objek Coordinate
30
31     # Public
32    def __init__(self, name, lat, long):
33        self.__name = name
34        self.__coordinate = Coordinate(lat, long)
35
36    def getName(self):
37        return self.__name
38
39    def getCoordinate(self):
40        return self.__coordinate
41
42    def setName(self, name):
43        self.__name = name
44
45    def setCoordinate(self, lat, long):
46        self.__coordinate = Coordinate(lat, long)
47
48 class Graph:
49     # Private
50     __vertices = None # Array of objek Vertice
51     __adj_matrix = None # Array of float
52
53     # Public
54    def __init__(self, file_name):
55        self.__vertices = []
56        self.__adj_matrix = []
57
58        file = open(file_name, 'r')
59
60        vertices_count = int(file.readline())
61        for i in range(vertices_count):
62            line = file.readline()
63            vertice_name = ""
64            vertice_lat = ""
65            vertice_long = ""
66
67            for j in range(len(line)):
68                if line[j] != '(':
69                    vertice_name += line[j]
70                else:
71                    break
72
73            for j in range(j + 1, len(line)):
74                if line[j] != ',':
75                    vertice_lat += line[j]
76                else:
77                    break
78
79            for j in range(j + 1, len(line)):
80                if line[j] != ')':
81                    vertice_long += line[j]
82                else:
83                    break
84
85            while (vertice_name[len(vertice_name) - 1] == ' '):
86                vertice_name = vertice_name[:-1]
87            vertice_lat = float(vertice_lat)
88            vertice_long = float(vertice_long)
89
90            self.__vertices.append(Vertice(vertice_name, vertice_lat, vertice_long))
91
92        for i in range(vertices_count):
93            line = file.readline()
94            row = [float(x) for x in line.split(" ")]
95            self.__adj_matrix.append(row)
96
```

```

97     file.close()
98
99     def getVertices(self):
100         return self.__vertices
101
102     def getAdjMatrix(self):
103         return self.__adj_matrix
104
105     def getVertexByName(self, name):
106         for vertex in self.__vertices:
107             if vertex.getName() == name:
108                 return vertex
109         return None
110
111     def getVertexByIndex(self, idx):
112         return self.__vertices[idx]
113
114     def getVertexIndex(self, name):
115         for i in range(len(self.__vertices)):
116             if self.__vertices[i].getName() == name:
117                 return i
118         return None
119
120     def haversine(self, from_vertex_lat, from_vertex_long, to_vertex_lat, to_vertex_long):
121         from_vertex_lat, from_vertex_long, to_vertex_lat, to_vertex_long = map(radians, [from_vertex_lat, from_vertex_long, to_vertex_lat, to_vertex_long])
122         dlon = to_vertex_long - from_vertex_long
123         dlat = to_vertex_lat - from_vertex_lat
124         a = sin(dlat/2)**2 + cos(from_vertex_lat) * cos(to_vertex_lat) * sin(dlon/2)**2
125         c = 2 * asin(sqrt(a))
126         r = 6371 * 1000
127         return c * r
128
129     def astarPath(self, from_vertex, to_vertex):
130         vertex_count = len(self.__vertices)
131         # closedList menyimpan apakah vertex telah dikunjungi atau belum
132         closedList = [False for i in range(vertex_count)]
133         # openList adalah list kemungkinan vertex yang dikunjungi selanjutnya
134         openList = [[0.0, from_vertex]]
135         # vertexParam menyimpan parameter untuk setiap vertex
136         # setiap indeks masing-masing menyimpan parent, parameter f, parameter g, dan parameter h
137         vertexParam = [[-1, -1.0, 0.0, 0.0] for i in range(vertex_count)]
138
139         minIdx = 0
140         while openList:
141             current = openList.pop(minIdx)
142             closedList[current[1]] = True
143             latOver = self.getVertexByIndex(to_vertex).getCoordinate().getLat()
144             longOver = self.getVertexByIndex(to_vertex).getCoordinate().getLong()

```

```

145         for branch in range(vertex_count):
146             # cek apakah ada hubungan antara branch dengan vertex yang sedang dikunjungi
147             if self.__adj_matrix[branch][current[1]] != 0:
148                 # cek apakah branch telah dikunjungi atau belum
149                 if not closedList[branch]:
150                     gNew = vertexParam[current[1]][2] + self.__adj_matrix[branch][current[1]]
151
152                     latBranch = self.getVertexByIndex(branch).getCoordinate().getLat()
153                     longBranch = self.getVertexByIndex(branch).getCoordinate().getLong()
154                     hNew = self.haversine(latBranch, longBranch, latOver, longOver)
155
156                     fNew = gNew + hNew
157                     # Jika vertex belum memiliki parent atau jarak ke parent sebelumnya lebih jauh,
158                     # set parent dan parameternya
159                     if (vertexParam[branch][1] == -1 or vertexParam[branch][1] > fNew):
160                         openList.append([fNew, branch])
161                         vertexParam[branch][0] = current[1]
162                         vertexParam[branch][1] = fNew
163                         vertexParam[branch][2] = gNew
164                         vertexParam[branch][3] = hNew
165
166         # Algoritma hanya akan lanjut jika terdapat vertex yang dapat dikunjungi
167         if openList:
168             # Ambil vertex dari openList dengan nilai f paling kecil
169             minIdx = openList.index(min(openList))
170
171             branch = openList[minIdx][1]
172             # Jika vertex akhir ditemukan, return path ke vertex tersebut
173             if branch == to_vertex:
174                 vertexParam[branch][0] = current[1]
175                 return self.getPath(vertexParam, to_vertex)
176
177         # Jika tidak ada path dari vertex awal ke vertex akhir, return None
178         return None
179
180     def getPath(self, vertexParam, to_vertex):
181         path = [self.getVertexByIndex(to_vertex).getName()]
182         pred = vertexParam[to_vertex][0]
183         while (pred != -1):
184             path.append(self.getVertexByIndex(pred).getName())
185             pred = vertexParam[pred][0]
186         path.reverse()
187         return path
188
189

```

```

190     def getPathMatrix(self, path):
191         vertex_count = len(self.__vertices)
192         path_matrix = [[0 for i in range(vertex_count)] for j in range(vertex_count)]
193         for i in range(len(path) - 1):
194             from_idx = self.getVertexIndex(path[i])
195             dest_idx = self.getVertexIndex(path[i + 1])
196             path_matrix[from_idx][dest_idx] = 1
197             path_matrix[dest_idx][from_idx] = 1
198         return path_matrix
199
200     def getPathDistance(self, path):
201         distance = 0
202         for i in range(len(path) - 1):
203             from_idx = self.getVertexIndex(path[i])
204             dest_idx = self.getVertexIndex(path[i + 1])
205             distance += self.__adj_matrix[from_idx][dest_idx]
206         return distance
207
208     def showVerticesName(self):
209         for i in range(len(self.__vertices)):
210             print(str(i + 1) + ". " + self.__vertices[i].getName())

```

## file Tucil.ipynb

```
from graph import *
from ipywidgets import HTML
from ipyleaflet import Map, Polyline, Marker
import time #Kalau ngga ada time.sleep(), inputnya bakal ada di atas print

'''-----'''
'''-----Setup-----'''
'''-----'''

print("Masukkan nama file:")
time.sleep(1)
file_name = input()
graph = Graph("../test/" + file_name)

vertices = graph.getVertices()
vertice_count = len(graph.getVertices())
adj_matrix = graph.getAdjMatrix()

center_lat = vertices[0].getCoordinate().getLat()
center_long = vertices[0].getCoordinate().getLong()
center = (center_lat, center_long)

'''-----'''
'''-----Bagian visualisasi graph-----'''
'''-----'''

map = Map(center = center, zoom = 16)
```

```
for vertice in vertices:
    marker_lat = vertice.getCoordinate().getLat()
    marker_long = vertice.getCoordinate().getLong()
    marker_name = vertice.getName()

    marker_message = HTML()
    marker_message.value = marker_name

    marker = Marker(location = (marker_lat, marker_long), draggable=False)
    marker.popup = marker_message

    map.add_layer(marker);

for i in range(vertice_count):
    for j in range(i, vertice_count):
        if adj_matrix[i][j] != 0:
            from_lat = vertices[i].getCoordinate().getLat()
            from_long = vertices[i].getCoordinate().getLong()
            dest_lat = vertices[j].getCoordinate().getLat()
            dest_long = vertices[j].getCoordinate().getLong()
            line = Polyline(
                locations=
                [
                    [from_lat, from_long],
                    [dest_lat, dest_long]
                ],
                color = "blue",
                fill = False
            )
            map.add_layer(line)
```

```
display(map)
time.sleep(1)

'''-----'''
'''-----Bagian visualisasi path-----'''
'''-----'''
```

```

graph.showVerticesName()
validVertex = False
while (not validVertex):
    print("Masukkan nomor simpul asal:")
    from_vertex_idx = int(input())
    if (from_vertex_idx > 0 and from_vertex_idx <= vertex_count):
        validVertex = True
    else:
        print("Nomor simpul tidak valid")
validVertex = False
while (not validVertex):
    print("Masukkan nomor simpul tujuan:")
    dest_vertex_idx = int(input())
    if (dest_vertex_idx > 0 and dest_vertex_idx <= vertex_count and dest_vertex_idx != from_vertex_idx):
        validVertex = True
    else:
        print("Nomor simpul tidak valid")

# from_vertex_name = graph.getVertexByIndex(from_vertex_idx - 1).getName()
# dest_vertex_name = graph.getVertexByIndex(dest_vertex_idx - 1).getName()

path = graph.aStarPath(from_vertex_idx - 1, dest_vertex_idx - 1)

```

```

if path != None:
    map_path = Map(center = center, zoom = 16)

    path_matrix = graph.getPathMatrix(path)
    distance = graph.getPathDistance(path)

    for vertex in vertices:
        marker_lat = vertex.getCoordinate().getLat()
        marker_long = vertex.getCoordinate().getLong()
        marker_name = vertex.getName()

        marker_message = HTML()
        marker_message.value = marker_name

        marker = Marker(location = (marker_lat, marker_long), draggable=False)
        marker.popup = marker_message

        map_path.add_layer(marker);

```

```

# Dua kali looping agar polyline merah ada di atas
for i in range(vertex_count):
    for j in range(i, vertex_count):
        if adj_matrix[i][j] != 0 and path_matrix[i][j] != 1:
            from_lat = vertices[i].getCoordinate().getLat()
            from_long = vertices[i].getCoordinate().getLong()
            dest_lat = vertices[j].getCoordinate().getLat()
            dest_long = vertices[j].getCoordinate().getLong()
            line = Polyline(
                locations=[
                    [from_lat, from_long],
                    [dest_lat, dest_long]
                ],
                color = "blue",
                fill = False
            )
            map_path.add_layer(line)

```

```

for i in range(vertice_count):
    for j in range(i, vertice_count):
        if path_matrix[i][j] == 1:
            from_lat = vertices[i].getCoordinate().getLat()
            from_long = vertices[i].getCoordinate().getLong()
            dest_lat = vertices[j].getCoordinate().getLat()
            dest_long = vertices[j].getCoordinate().getLong()
            line = Polyline(
                locations=[
                    [from_lat, from_long],
                    [dest_lat, dest_long]
                ],
                color = "red",
                fill = False
            )
            map_path.add_layer(line)

display(map_path)
time.sleep(1)

print("Jalur yang ditemukan: " + path[0], end = "")
for i in range(1, len(path)):
    print(" --> " + path[i], end = "")
print("")
print("Jarak: " + str(distance) + " meter")
else:
    print("Simpul tujuan tersebut tidak dapat dicapai dari simpul awal")

```

### C. Pengujian Program

Checklist pengujian:

Poin	Ya	Tidak
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		✓

Screenshot pengujian:

#### 1. Sekitar\_ITB.txt :

```
17
Ciungwanara A (-6.8947568207787935, 107.61169727452938)
Ciungwanara B (-6.8964623488955095, 107.61133249414877)
Gelap Nyawang (-6.894767472027137, 107.61014695778496)
Tamansari A (-6.893863446462258, 107.60843570852492)
Tamansari B (-6.894875315915152, 107.60884876871873)
Tamansari C (-6.895727414803047, 107.60930742638318)
Tamansari D (-6.896835141093129, 107.60965611354067)
Tamansari E (-6.898018754428389, 107.60957430617096)
Skanda (-6.893421418611346, 107.61004503386513)
Ganesa A (-6.893282951970821, 107.60990019458451)
Ganesa B (-6.8932296955598655, 107.61021133081695)
Ganesa C (-6.893549233935907, 107.611933308931)
Badak Singa (-6.897575399045794, 107.6114558757582)
Taman Kota A (-6.8967286290737855, 107.61085506096451)
Taman Kota B (-6.8967286290737855, 107.61027570384203)
Taman Kota C (-6.896941653108604, 107.61053587810535)
Mundinglya (-6.898230446493914, 107.61034812347313)
0 193.87 171.14 0 0 0 0 0 0 0 136.78 0 0 0 0 0
193.87 0 0 0 0 0 0 0 0 0 124.51 0 0 0 0
171.14 0 0 0 143.80 0 0 0 150.09 0 0 0 0 0 0 0
0 0 0 0 121.40 0 0 0 0 174.07 0 0 0 0 0 0
0 0 143.80 121.40 0 107.42 0 0 0 0 0 0 0 0 0 0
0 0 0 0 107.42 0 129.04 0 0 0 0 0 0 0 0 0
0 0 0 0 0 129.04 0 131.92 0 0 0 0 0 69.41 0 0
0 0 0 0 0 131.92 0 0 0 0 0 0 0 0 88.60
0 0 150.09 0 0 0 0 0 22.19 28.13 0 0 0 0 0 0
0 0 0 174.07 0 0 0 0 22.19 0 34.85 0 0 0 0 0
0 0 0 0 0 0 0 28.13 34.85 0 193.38 0 0 0 0 0
136.78 0 0 0 0 0 0 0 0 193.38 0 0 0 0 0 0
0 124.51 0 0 0 0 0 0 0 0 0 115.17 0 0 0
```



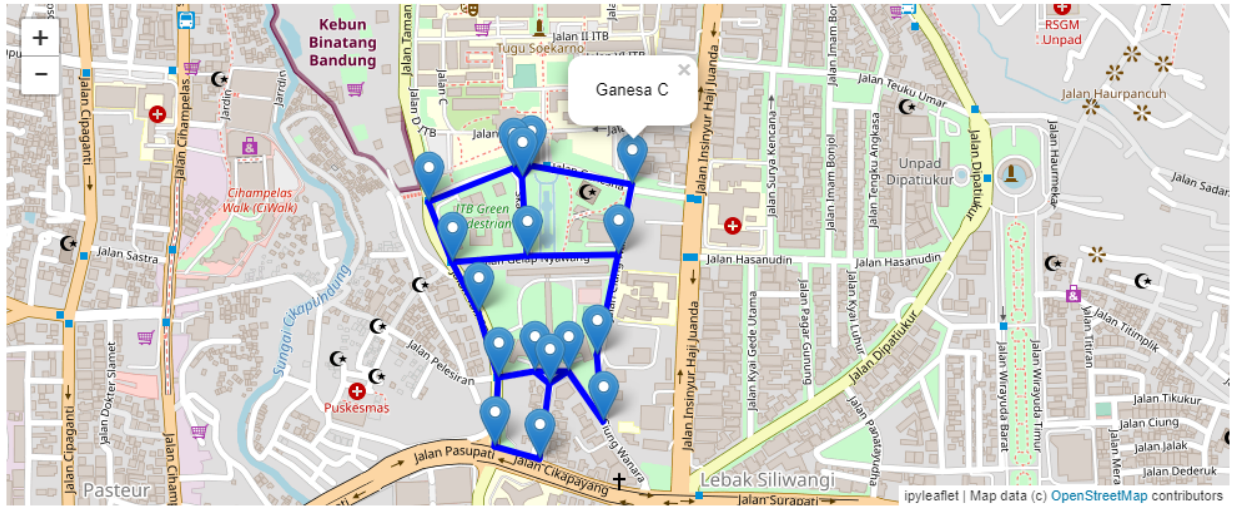
```

0 0 0 0 0 0 0 0 0 0 0 0 115.17 0 63.95 42.45 0
0 0 0 0 0 0 69.41 0 0 0 0 0 63.95 0 37.22 0
0 0 0 0 0 0 0 0 0 0 0 0 42.45 37.22 0 144.79
0 0 0 0 0 0 0 88.60 0 0 0 0 0 0 0 144.79 0

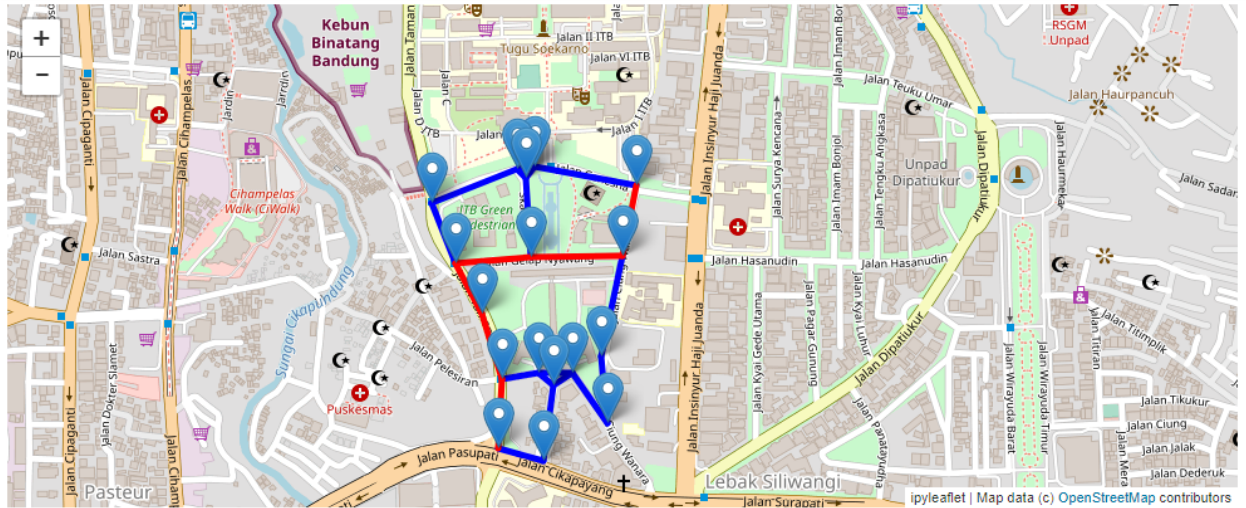
```

## Hasil kompilasi:

Masukkan nama file:  
Sekitar\_ITB.txt



1. Ciungwanara A
  2. Ciungwanara B
  3. Gelap Nyawang
  4. Tamansari A
  5. Tamansari B
  6. Tamansari C
  7. Tamansari D
  8. Tamansari E
  9. Skanda
  10. Ganesa A
  11. Ganesa B
  12. Ganesa C
  13. Badak Singa
  14. Taman Kota A
  15. Taman Kota B
  16. Taman Kota C
  17. Mundinglaya
- Masukkan nomor simpul asal:
- 12
- Masukkan nomor simpul tujuan:
- 8



Jalur yang ditemukan: Ganesa C --> Ciungwanara A --> Gelap Nyawang --> Tamansari B --> Tamansari C --> Tamansari D --> Tamansari E  
 Jarak: 820.0999999999999 meter

## 2. Alun\_alun.txt:

22

Dalem Kaum (-6.92236836378364, 107.60645087404654)  
 Alun-Alun Timur (-6.922573388306235, 107.60764177478431)  
 Balonggede A (-6.923051334731237, 107.60757874293246)  
 Balonggede B (-6.923215087849953, 107.6079918031041)  
 Balonggede C (-6.923876460423505, 107.60792892957292)  
 Balonggede D (-6.923943026620883, 107.60733482029158)  
 Dewi Sartika A (-6.924362393457703, 107.60725301288808)  
 Dewi Sartika B (-6.924314465841124, 107.60620561031065)  
 Kepatihan A (-6.923403840127779, 107.60630485201412)  
 Kepatihan B (-6.923068346002527, 107.60393243818677)  
 Otto Iskandar Dinata (-6.92208249779924, 107.60401022223334)  
 Asia Afrika A (-6.920805751415412, 107.60410409955993)  
 Asia Afrika B (-6.92104272857961, 107.6064537146372)  
 Asia Afrika C (-6.921213139051219, 107.60778945467743)  
 Asia Afrika D (-6.921308994913492, 107.60801476024297)  
 Asia Afrika E (-6.921311657575069, 107.60877382537062)  
 Naripan A (-6.919706069280748, 107.60899913090797)  
 Naripan B (-6.919543646281805, 107.60834467195197)  
 Belakang Factory A (-6.9202519166546015, 107.60817301058644)  
 Belakang Factory B (-6.919990975061936, 107.60658514295544)  
 Pasundan A (-6.9238837433208325, 107.60833398815875)  
 Pasundan B (-6.922648272980293, 107.60839836117476)  
 0 133.41 0 0 0 0 0 0 116.26 0 271.27 0 0 0 0 0 0 0 0 0 0  
 133.41 0 53.59 0 0 0 0 0 0 0 0 0 0 152.12 0 0 0 0 0 0 83.92  
 0 53.59 0 49.09 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 49.09 0 73.86 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 73.86 0 65.99 0 0 0 0 0 0 0 0 0 0 0 0 44.71 0  
 0 0 0 0 65.99 0 47.49 0 0 0 0 0 0 0 0 0 0 0 0 0

```

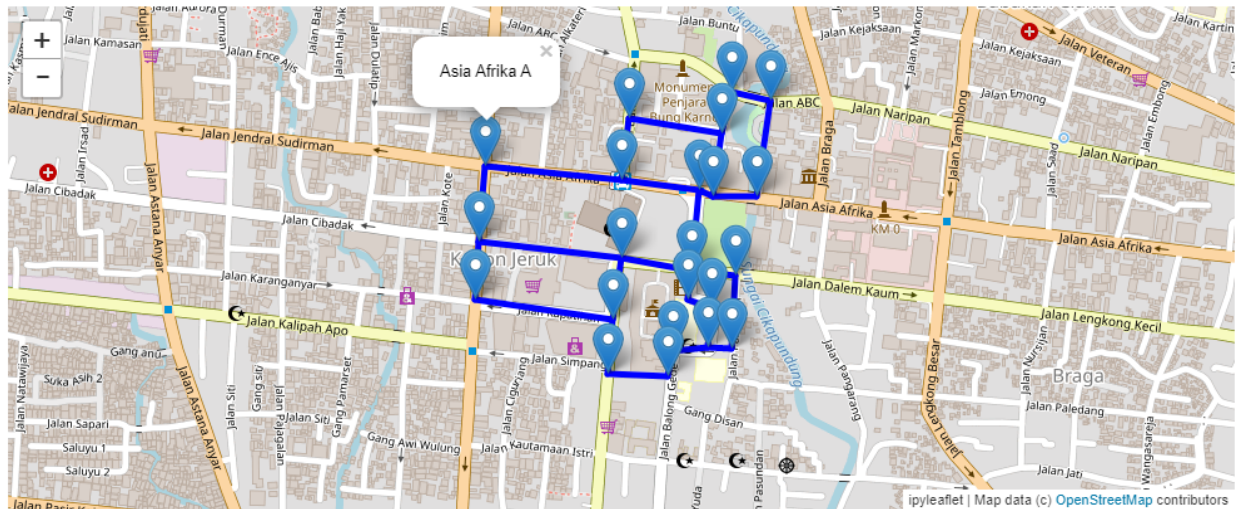
0 0 0 0 0 47.49 0 115.73 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 115.73 0 101.84 0 0 0 0 0 0 0 0 0 0 0 0
116.26 0 0 0 0 0 0 101.84 0 264.52 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 264.52 0 109.95 0 0 0 0 0 0 0 0 0 0 0
271.27 0 0 0 0 0 0 0 109.95 0 142.34 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 142.34 0 260.69 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 260.69 0 148.65 0 0 0 0 0 117.84 0 0
0 152.12 0 0 0 0 0 0 0 148.65 0 27.05 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 27.05 0 83.78 0 0 118.83 0 0 0
0 0 0 0 0 0 0 0 0 83.78 0 180.25 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 180.25 0 74.46 0 0 0 0 0
0 0 0 0 0 0 0 0 0 74.46 0 81.00 0 0 0 0
0 0 0 0 0 0 0 0 0 118.83 0 81.00 0 177.66 0 0
0 0 0 0 0 0 0 0 0 117.84 0 0 0 0 177.66 0 0 0
0 0 0 0 44.71 0 0 0 0 0 0 0 0 0 0 0 0 137.56
0 83.92 0 0 0 0 0 0 0 0 0 0 0 0 0 0 137.56 0

```

## Hasil kompilasi:

Masukkan nama file:

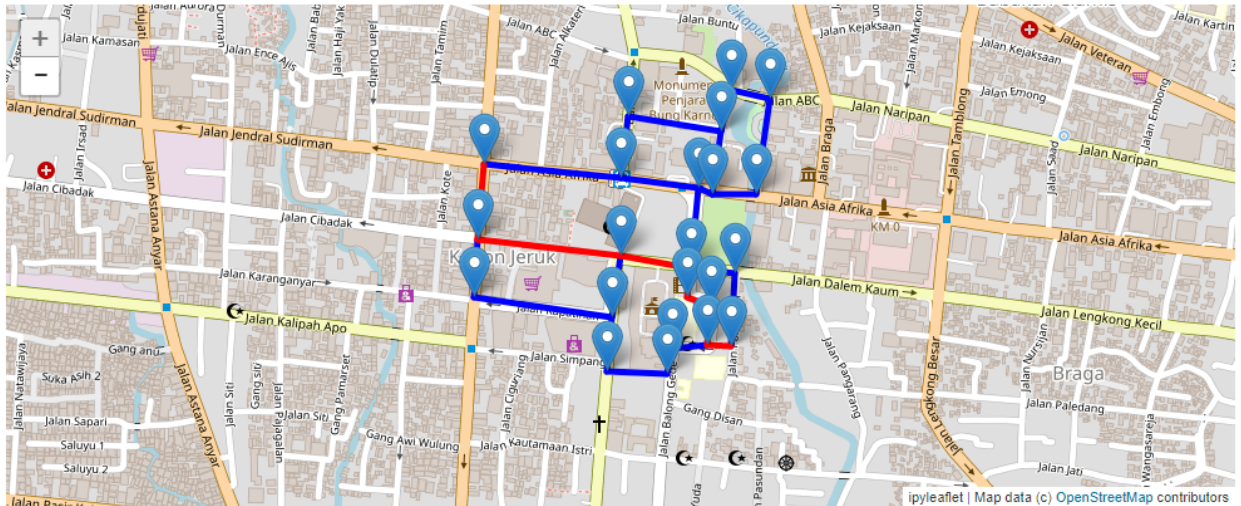
Alun-alun.txt



1. Dalem Kaum
2. Alun-Alun Timur
3. Balonggede A
4. Balonggede B
5. Balonggede C
6. Balonggede D
7. Dewi Sartika A
8. Dewi Sartika B
9. Kepatihan A
10. Kepatihan B
11. Otto Iskandar Dinata
12. Asia Afrika A
13. Asia Afrika B
14. Asia Afrika C
15. Asia Afrika D
16. Asia Afrika E
17. Naripan A
18. Naripan B
19. Belakang Factory A
20. Belakang Factory B
21. Pasundan A
22. Pasundan B

Masukkan nomor simpul asal:  
21

Masukkan nomor simpul tujuan:  
12



Jalur yang ditemukan: Pasundan A --> Balonggede C --> Balonggede B --> Balonggede A --> Alun-Alun Timur --> Dalem Kaum --> Otto Iskandar Dinata --> Asia Afrika A  
Jarak: 768.27 meter

### 3. Buah\_batu.txt:

25

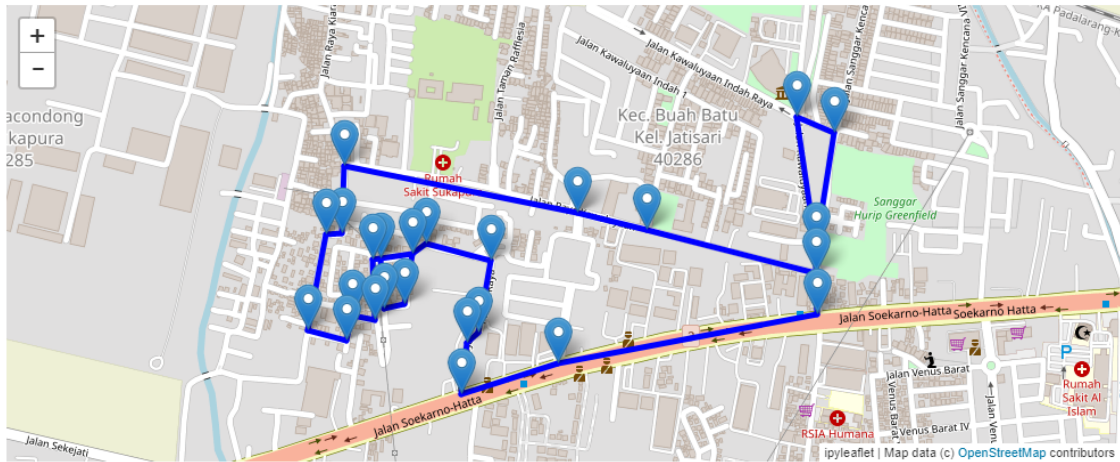
Palem Segitiga (-6.93876084357985, 107.65606486632208)  
Palem Kipas A (-6.938840720491008, 107.65567594602942)  
Palem Kipas B (-6.939047928438553, 107.6555106699152)  
Palem Raya A (-6.937938969872457, 107.6554503202124)  
Palem Raya B (-6.937914147450664, 107.65560352639085)  
Palem Raya C (-6.937831607809763, 107.65619897675248)  
Palem Raya D (-6.937626589920711, 107.6564645154464)  
Palem Raya E (-6.937948776751901, 107.65769296911277)  
Palem Raya F (-6.939288046889146, 107.65745425251653)  
Palem Raya G (-6.939487738926127, 107.65723431139196)  
Soekarno-Hatta A (-6.940435609319805, 107.65712434083929)

Soekarno-Hatta B (-6.939872724907053, 107.6589454802588)  
Soekarno-Hatta C (-6.938946153893205, 107.6638405116362)  
Kawalayaan A (-6.938176672548111, 107.66382575951154)  
Kawalayaan B (-6.937381895491837, 107.66062051979084)  
Kawalayaan C (-6.9370570616120695, 107.6593142840138)  
Kawalayaan D (-6.936177649727831, 107.65491249825531)  
Cidurian Utara A (-6.9374290600254644, 107.65488299396574)  
Cidurian Utara B (-6.937467667319758, 107.65457185772678)  
Cidurian Utara C (-6.939254809799223, 107.65422876929155)  
Palem Lilin A (-6.939447312639151, 107.65496886370066)  
Palem Lilin B (-6.93897562325154, 107.65506765343225)  
Kawalayaan Raya A (-6.937722704582644, 107.6638257595108)  
Kawalayaan Raya B (-6.935270471277724, 107.66346634354166)  
Sanggar Kencana (-6.935574005730196, 107.66416908225672)  
0 43.83 0 0 0 104.38 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
43.83 0 0 0 103.33 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 123.49 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 49.55 0 0 0  
0 0 123.49 0 17.13 0  
0 103.33 0 17.13 0 66.36 0  
104.38 0 0 0 66.36 0 37.13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 37.13 0 140.25 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 140.25 0 151.23 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 151.23 0 32.90 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 32.90 0 106.09 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 106.09 0 210.53 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 210.53 0 550.05 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 550.05 0 85.57 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 85.57 0 364.66 0 0 0 0 0 0 0 0 50.47 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 364.66 0 148.63 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 148.63 0 495.61 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 495.61 0 139.18 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 139.18 0 34.61 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 34.61 0 202.29 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202.29 0 84.44 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 84.44 0 53.57 0 0 0 0  
0 0 49.55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 53.57 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 50.47 0 0 0 0 0 0 0 0 275.54 241.91  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 275.54 0 84.59  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 241.91 84.59 0

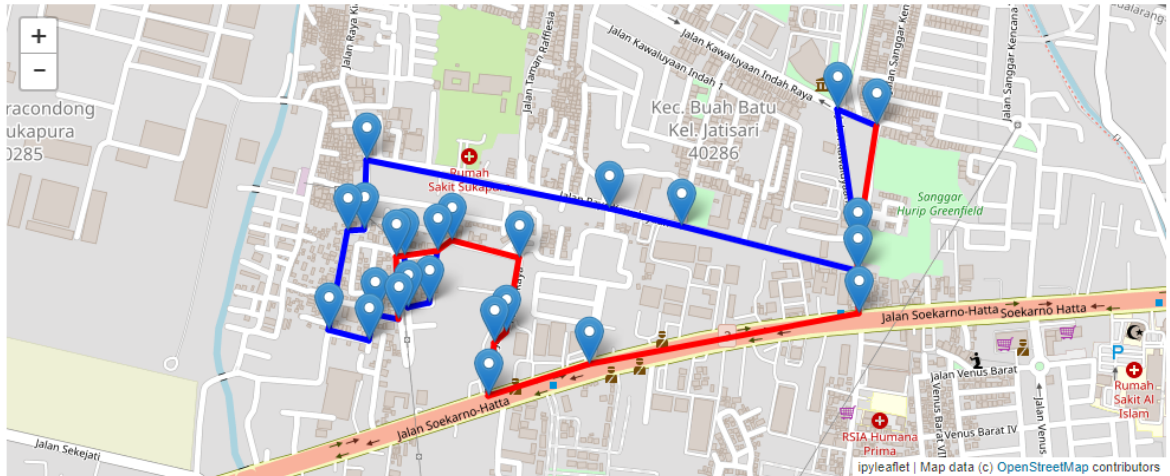


## Hasil kompilasi:

Masukkan nama file:  
Buah\_batu.txt



1. Palem Segitiga
  2. Palem Kipas A
  3. Palem Kipas B
  4. Palem Raya A
  5. Palem Raya B
  6. Palem Raya C
  7. Palem Raya D
  8. Palem Raya E
  9. Palem Raya F
  10. Palem Raya G
  11. Soekarno-Hatta A
  12. Soekarno-Hatta B
  13. Soekarno-Hatta C
  14. Kawalayaan A
  15. Kawalayaan B
  16. Kawalayaan C
  17. Kawalayaan D
  18. Cidurian Utara A
  19. Cidurian Utara B
  20. Cidurian Utara C
  21. Palem Lilin A
  22. Palem Lilin B
  23. Kawalayaan Raya A
  24. Kawalayaan Raya B
  25. Sanggar Kencana
- Masukkan nomor simpul asal:  
25  
Masukkan nomor simpul tujuan:  
3



Jalur yang ditemukan: Sanggar Kencana --> Kawalayaan Raya A --> Kawalayaan A --> Soekarno-Hatta C --> Soekarno-Hatta B --> Soekarno-Hatta A --> Palem Raya G --> Palem Raya F --> Palem Raya E --> Palem Raya D --> Palem Raya C --> Palem Raya B --> Palem Raya A --> Palem Kipas B  
 Jarak: 1813.110000000001 meter

#### 4. Kedonganan\_Bali.txt:

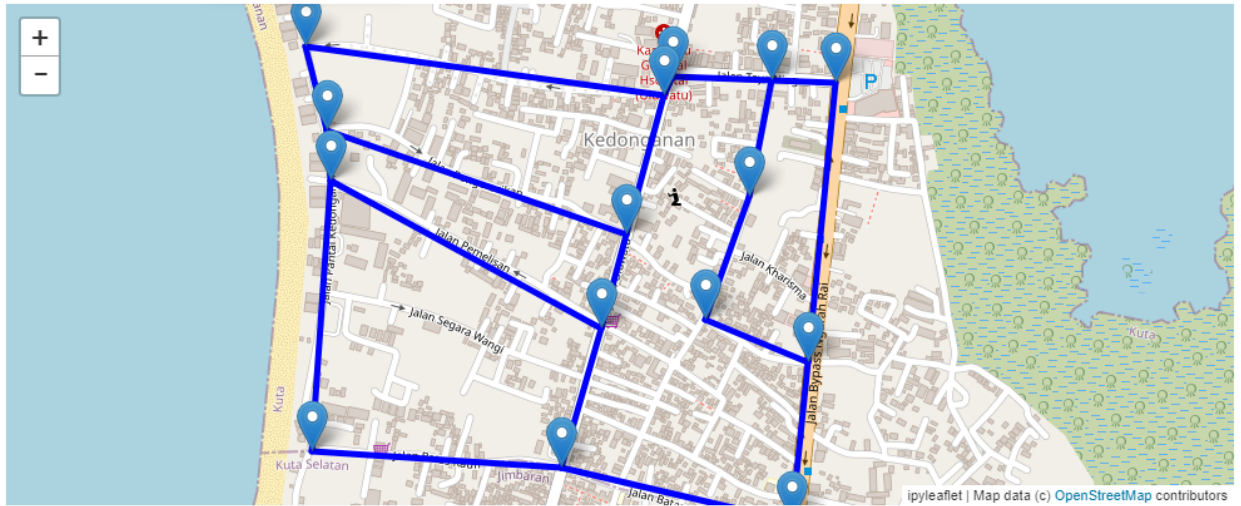
15

```
Pantai Kedonganan A (-8.76731304501074, 115.16979838617918)
Pantai Kedonganan B (-8.762721712011016, 115.170128297914)
Pantai Kedonganan C (-8.76187077165552, 115.17006124269548)
Pantai Kedonganan D (-8.760463137066527, 115.16969646228641)
Uluwatu A (-8.761276967892428, 115.17583335643705)
Uluwatu B (-8.763638922782356, 115.17520035510972)
Uluwatu C (-8.765218852820853, 115.17475510840882)
Uluwatu D (-8.767586084411525, 115.17408455621258)
By Pass Ngurah Rai A (-8.76849533060252, 115.17802203902885)
By Pass Ngurah Rai B (-8.765808846410296, 115.17831329000647)
By Pass Ngurah Rai C (-8.761069043695661, 115.17877462994605)
Toyaning A (-8.76102928000882, 115.1776722420928)
Toyaning B (-8.760965658067935, 115.1759797682345)
Pesraman A (-8.765077202961894, 115.17653498549957)
Pesraman B (-8.763004205331901, 115.1773020972517)
0 511.81 0 0 0 0 0 472.00 0 0 0 0 0 0 0
511.81 0 94.90 0 0 0 0 579.34 0 0 0 0 0 0 0
0 94.90 0 161.57 0 598.01 0 0 0 0 0 0 0 0 0
0 0 161.57 0 680.47 0 0 0 0 0 0 0 0 0 0
0 0 0 680.47 0 271.69 0 0 0 0 0 0 38.17 0 0
0 0 598.01 0 271.69 0 182.36 0 0 0 0 0 0 0 0
0 579.34 0 0 0 182.36 0 273.34 0 0 0 0 0 0 0
472.00 0 0 0 0 0 273.34 0 444.36 0 0 0 0 0 0
0 0 0 0 0 0 0 444.36 0 300.43 0 0 0 0 0
0 0 0 0 0 0 0 0 300.43 0 529.47 0 0 211.68 0
0 0 0 0 0 0 0 0 0 529.47 0 121.23 0 0 0
0 0 0 0 0 0 0 0 0 0 121.23 0 186.13 0 223.33
0 0 0 0 38.17 0 0 0 0 0 0 186.13 0 0 0
0 0 0 0 0 0 0 0 0 211.68 0 0 0 0 245.43
```

0 0 0 0 0 0 0 0 0 0 0 223.33 0 245.43 0

## Hasil kompilasi:

Masukkan nama file:  
Kedonganan\_Bali.txt



1. Pantai Kedonganan A
2. Pantai Kedonganan B
3. Pantai Kedonganan C
4. Pantai Kedonganan D
5. Uluwatu A
6. Uluwatu B
7. Uluwatu C
8. Uluwatu D
9. By Pass Ngurah Rai A
10. By Pass Ngurah Rai B
11. By Pass Ngurah Rai C
12. Toyaning A
13. Toyaning B
14. Pesnaman A
15. Pesnaman B

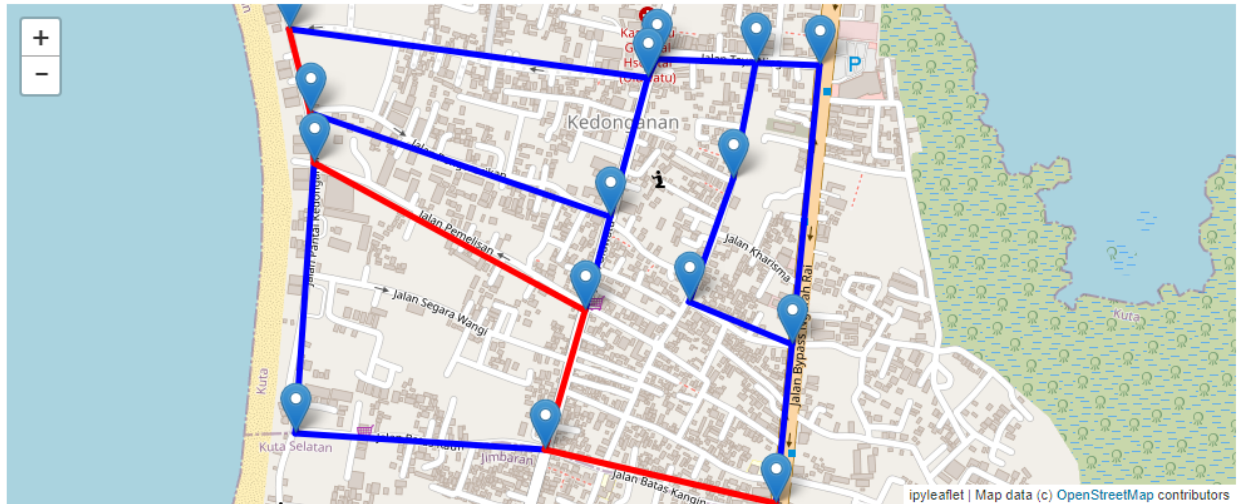
Masukkan nomor simpul asal:

9

Masukkan nomor simpul tujuan:

4





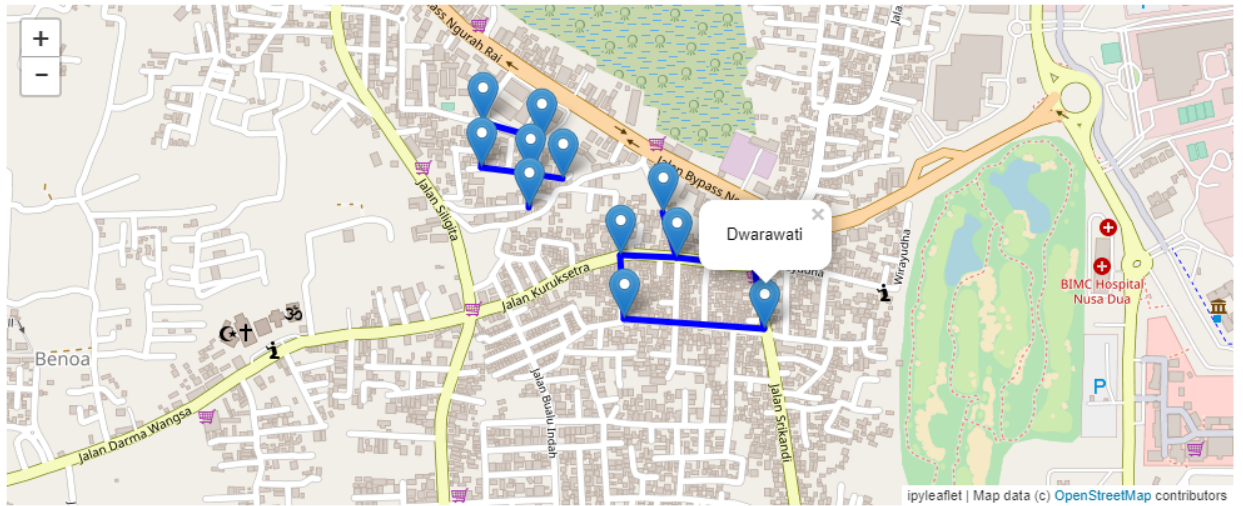
Jalur yang ditemukan: By Pass Ngurah Rai A --> Uluwatu D --> Uluwatu C --> Pantai Kedonganan B --> Pantai Kedonganan C --> Pantai Kedonganan D  
 Jarak: 1553.51 meter

## 5. Nusa Dua.txt:

```
12
Kuruksetra A (-8.799026551115055, 115.22010415044159)
Kuruksetra B (-8.799067215382012, 115.22106434956154)
Kuruksetra C (-8.799227063523956, 115.2223909398667)
Metila (-8.80010613881368, 115.22016484190235)
Dwarawati (-8.800286069790184, 115.2225621545735)
Gang Sunyi (-8.79830242423142, 115.22082849479206)
Tebesari A (-8.798217108346547, 115.2185431464968)
Tebesari B (-8.797645975415653, 115.21855378253058)
Praja Sentral A (-8.797066590587137, 115.21874511695447)
Praja Sentral B (-8.796776560573589, 115.21774375542277)
Gang Sale A (-8.797545250625571, 115.21772497996022)
Gang Sale B (-8.79773637205245, 115.21910811886978)
0 105.60 0 120.22 0 0 0 0 0 0 0 0
105.60 0 146.85 0 0 88.90 0 0 0 0 0 0
0 146.85 0 0 119.24 0 0 0 0 0 0 0
120.22 0 0 0 264.1 0 0 0 0 0 0 0
0 0 119.24 264.1 0 0 0 0 0 0 0 0
0 88.90 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 63.51 0 0 0 0 0
0 0 0 0 0 0 63.51 0 67.76 0 91.76 61.73
0 0 0 0 0 0 67.76 0 114.66 0 0 0
0 0 0 0 0 0 0 114.66 0 85.49 0
0 0 0 0 0 0 0 91.76 0 85.49 0 0
0 0 0 0 0 0 0 61.73 0 0 0 0
```

Hasil kompilasi:

Masukkan nama file:  
Nusa Dua.txt

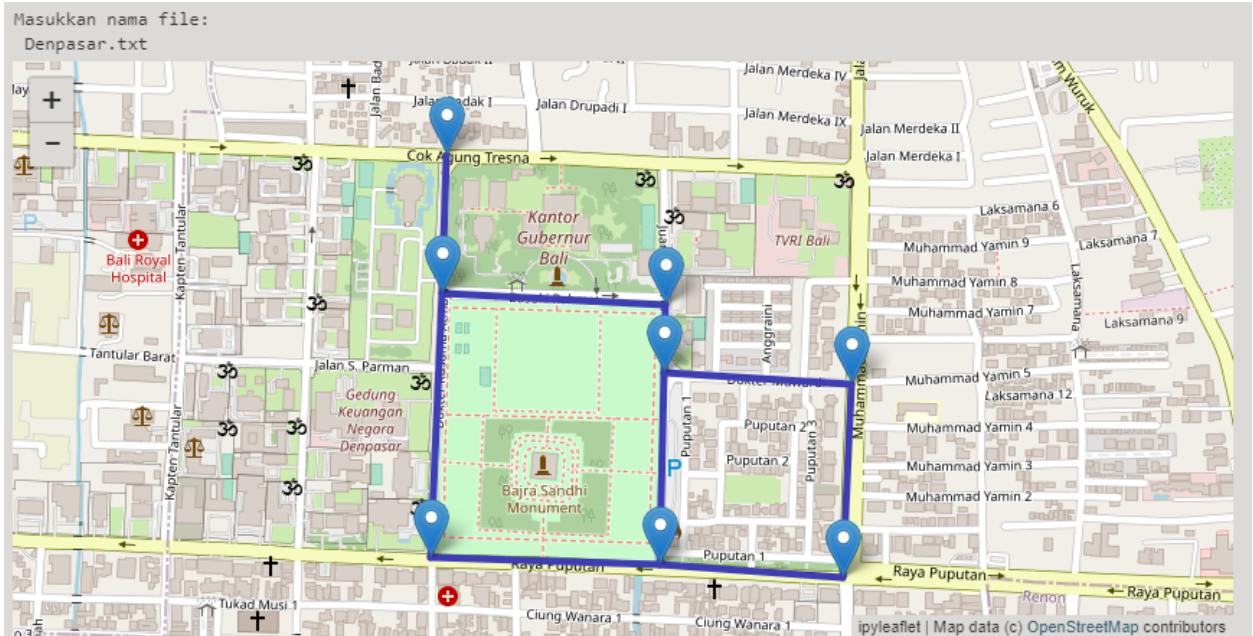


1. Kuruksetra A  
2. Kuruksetra B  
3. Kuruksetra C  
4. Metila  
5. Dwarawati  
6. Gang Sunyi  
7. Tebesari A  
8. Tebesari B  
9. Praja Sentral A  
10. Praja Sentral B  
11. Gang Sale A  
12. Gang Sale B  
Masukkan nomor simpul asal:  
7  
Masukkan nomor simpul tujuan:  
5  
Simpul tujuan tersebut tidak dapat dicapai dari simpul awal

## 6. Denpasar.txt:

8  
Kusuma Atmaja A (-8.667144689175501, 115.23242133757562)  
Kusuma Atmaja B (-8.669186584968058, 115.23235837522914)  
Basuki Rahmat (-8.669361214850515, 115.23568517262082)  
Jalan Raya Puputan A (-8.673080622147397, 115.23217851312165)  
Jalan Raya Puputan B (-8.673186655219274, 115.23559026686571)  
Ir. H. Juanda (-8.670349062121565, 115.23566333165823)  
Prof. Moh. Yamin A (-8.670528374442895, 115.23845077333321)  
Prof. Moh. Yamin B (-8.673373192711207, 115.23832999496446)  
0 227.15 0 0 0 0 0 0  
227.15 0 366.21 433.44 0 0 0 0  
0 366.21 0 0 0 109.86 0 0  
0 433.44 0 0 375.21 0 0 0  
0 0 0 375.21 0 315.62 0 301.87  
0 0 109.86 0 315.62 0 307.05 0  
0 0 0 0 0 307.05 0 316.60  
0 0 0 0 301.87 0 316.60 0

## Hasil kompilasi:



1. Kusuma Atmaja A
  2. Kusuma Atmaja B
  3. Basuki Rahmat
  4. Jalan Raya Puputan A
  5. Jalan Raya Puputan B
  6. Ir. H. Juanda
  7. Prof. Moh. Yamin A
  8. Prof. Moh. Yamin B
- Masukkan nomor simpul asal:  
1
- Masukkan nomor simpul tujuan:  
8

