LAPORAN HASIL TUGAS KECIL 3

Implementasi Algoritma A* untuk Menentukan LIntasan Terpendek pada Graf Peta dalam Bahasa *Python*

DIAJUKAN UNTUK MEMENUHI TUGAS MATA KULIAH

IF2211 – Strategi Algoritma

SEMESTER II TAHUN 2020/2021

Disusun oleh :

Bryan Rinaldo	**13519103**
Maximillian Lukman	**13519153**

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2021

I. Kode Program

```
import math
import networkx as nx
import matplotlib.pyplot as plt

print("                          __                        _
___")
print("  /\/\   __ _ _   _    /\ \ \__ _  __ _ _ __   __ _(_)_ __
/___\_ __ ___")
print(" /    \ / _` | | | |  /  \/ / _` |/ _` | '_ \ / _` | | '_ \   //
// '_ ` _ \\")
print("/ /\/\ \ (_| | |_| | / /\  / (_| | (_| | |_) | (_| | | | | | /
\_//| | | | | |")
print("\/    \/\__,_|\__,_| \_\ \/ \__, |\__,_| .__/ \__,_|_|_| |_| \___/
|_| |_| |_|")
print("                            |___/      |_|")

#open dan read file nya
path = "C:/Users/Windows/Desktop/Tucil3_13519103/test/{}.txt"
print("test1 : Peta jalan di dalam ITB")
print("test2 : Peta jalan sekitar alun-alun Bandung")
print("test3 : Peta jalan sekitar Buahbatu")
print("test4 : Peta jalan sekitar Dago")
print("test5 : Peta kota-kota di Romania")
print("test6 : Peta tempat-tempat terkenal yang tidak saling berhubungan")
fileName = input("Masukkan file yang ingin diuji (test1-test6): ")
f = open(path.format(fileName), "r")
n = int(f.readline())    #banyak nya node tempat
print()

places = []      #initialize temporary array of places
adjMat = []      #initialize array for adj matrix
x = 0
for line in f:  #masukin adj matrix
    if (x >= n):
        adjMat.append([x.strip("\n") for x in line.split(" ")])
        x+=1
    else:
        places.append(line.strip("\n"))
```

```
        x+=1
# print(places)
# print(adjMat)

#masukin nama, posisi X, dan posisi Y
nama = []       #nama tempat (node)
posisiX = []    #koordinat X
posisiY = []    #koordinat Y
for i in places:
    splits = i.split()
    nama.append(splits[0])
    posisiX.append(splits[1])
    posisiY.append(splits[2])
# print(nama)

#MEMBUAT RUMUS EUCLIDEAN (KELUARAN HASIL FLOAT)
def Euclidean(tempatA,tempatB):
    indexA = nama.index(tempatA)
    indexB = nama.index(tempatB)
    xa = float(posisiX[indexA])
    ya = float(posisiY[indexA])
    xb = float(posisiX[indexB])
    yb = float(posisiY[indexB])
    dist = math.sqrt((xa-xb)**2 + (ya-yb)**2) * 100000
    hasil = float("{:.2f}".format(dist))
    return hasil

graph = []          #adj Matrix dengan bobot
graphTemp = []      #graph temporary
graphList =[]       #list graph terhubung

#MEMBUAT GRAPH DAN GRAPHLIST
for i in range(len(adjMat)):
    graph.append([])
    for j in range(len(adjMat[i])):
        if(adjMat[i][j] == '1'):
            graph[i].append(Euclidean(nama[i], nama[j]))
            graphTemp.append(nama[i])
            graphTemp.append(nama[j])
        else:
```

```
            graph[i].append(0)
sumEdges = 0
for i in range(len(graphTemp)//2):
    graphList.append([])
    for j in range(2):
        graphList[i].append(graphTemp[sumEdges])
        sumEdges+=1
# print(graph)
# print(graphList)
# print(nama)

#BUAT LIST YANG BERISI JARAK GARIS LURUS MENUJU KE TEMPAT YANG DITUJU
def getStraightDistance():
    dict = {}
    for key in nama:
        dict[key] = Euclidean(key, ke)
    return dict

#UNTUK MENCARI PATH (KELUARAN LIST INDEX YANG DILALUI)
def searchPath(startPoint, endPoint):
    # euclideanDict =  Euclideantempat(endPoint)
    path = []
    path.append(nama.index(startPoint))
    startPointIndex = nama.index(startPoint)
    while (startPointIndex != nama.index(endPoint)):
        temp = {}
        for i in range(n):
            if graph[startPointIndex][i] != 0:
                # print('tidaknol')
                if graph[startPointIndex][i] !=
straightDistDict[nama[startPointIndex]]:
                    # print("gak langsung bos")
                    sum = graph[startPointIndex][i] +
straightDistDict[nama[i]]
                    temp[i] = sum
                else:
                    sum = graph[startPointIndex][i]  #BAHAYA KALO MISALNYA
ANGKA EUCLIDIAN NYA BISA SAMA PERSIS
                    temp[i] = sum
                    # startPointIndex = nama.index(endPoint)
```

```
            # else:
            #     print('nol')
        # print(temp)
        #buat biar ga bolak balik dan ga masuk ke jalan buntu
        for i in path:
            for key in temp.copy():
                #cek kalo udh lewat
                if key == i:
                    del temp[key]
        #cek kalo ga buntu
        for key in temp.copy():
            banyakjalur = 0
            for x in range(n):
                if graph[key][x] != 0:
                    banyakjalur += 1
            # print(banyakjalur)
            if banyakjalur <= 1 and key != nama.index(endPoint):
                del temp[key]
                # print("kedelete")
        # print(temp)
        if not temp:
            # print("Jaka sembung bawa golok, ga nyambung goblok")
            path.clear()
            return path
        else:
            minDist = min(temp, key=temp.get)
        # print(minDist)
        path.append(minDist)
        startPointIndex = minDist
    # print(path)
    return path

#UNTUK BUAT LIST PATH YANG DILALUI OLEH SEARCH PATH
def jalur(path):
    templist = []
    for i in range(len(path)-1):
        templist.append([])
        for j in range(2):
            templist[i].append(nama[path[i+j]])
    return templist
```

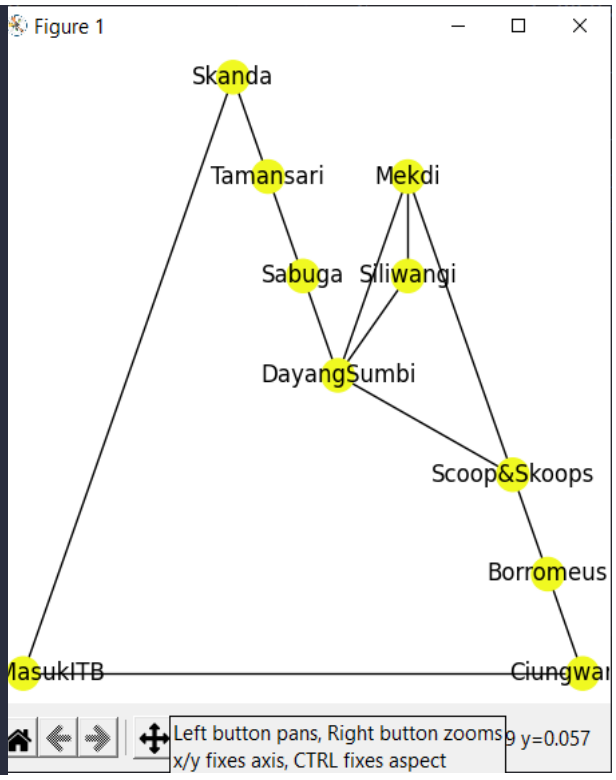
```
#UNTUK MENAMPILKAN GRAF
def drawFinalGraph(graph):
    # nodes = set([n1 for n1, n2 in graph] + [n2 for n1, n2 in graph])
    G=nx.Graph()
    # for node in nodes:
    #     G.add_node(node)
    G.add_nodes_from(nama)
    for edge in graph:
        G.add_edge(edge[0], edge[1], color='black')

    #KASIH WARNA KE NODE
    val_map = {}
    for key in path:
        val_map[nama[key]] = 0.0

    #KASIH WARNA KE EDGE
    for x in range(len(pathdone)):
        G.add_edge(pathdone[x][0],pathdone[x][1],color='r')

    colors = nx.get_edge_attributes(G,'color').values()

    values = [val_map.get(node, 0.25) for node in G.nodes()]
    cek = 0
    for i in values:
        if i == 0.0 :
            cek +=1
    # GAMBAR GRAF NYA
    pos = nx.planar_layout(G)
    if cek == n :
        nx.draw(G, pos, with_labels=True, node_color='y',
edge_color=colors)
    else:
        nx.draw(G, pos, with_labels=True,cmap=plt.get_cmap('plasma_r'),
node_color=values, edge_color=colors)
    plt.show()

def drawInitGraph(graph):
    # extract nodes from graph
    # nodes = set([n1 for n1, n2 in graph] + [n2 for n1, n2 in graph])
```

```
    # create networkx graph
    G=nx.Graph()
    # add nodes
    # for node in nodes:
    #     G.add_node(node)
    G.add_nodes_from(nama)
    # add edges
    for edge in graph:
        G.add_edge(edge[0], edge[1])
    val_map = {}
    values = [val_map.get(node, 0.25) for node in G.nodes()]
    # draw graph
    pos = nx.planar_layout(G)
    nx.draw(G, pos, with_labels=True, cmap=plt.get_cmap('plasma_r'),
node_color=values)
    # show graph
    plt.show()

def measureSumDistance(path):
    sum = 0
    for i in range(len(path)-1):
        sum += graph[path[i]][path[i+1]]
    return sum

print("Berikut simpul yang dapat dipilih: " + str(nama))
print()
choice = input("Apakah ingin menampilkan graf peta? (Y/N) ").lower()
if choice == 'y':
    print()
    print("Tutup window yang terbuka untuk melanjutkan...")
    drawInitGraph(graphList)
dari = input("Masukkan simpul asal: ")
ke = input("Masukkan simpul tujuan: ")

straightDistDict = getStraightDistance()
path = searchPath(dari,ke)
if len(path) != 0:
    print()
    print("Jalan yang dilalui adalah: ")
    for i in path:
```

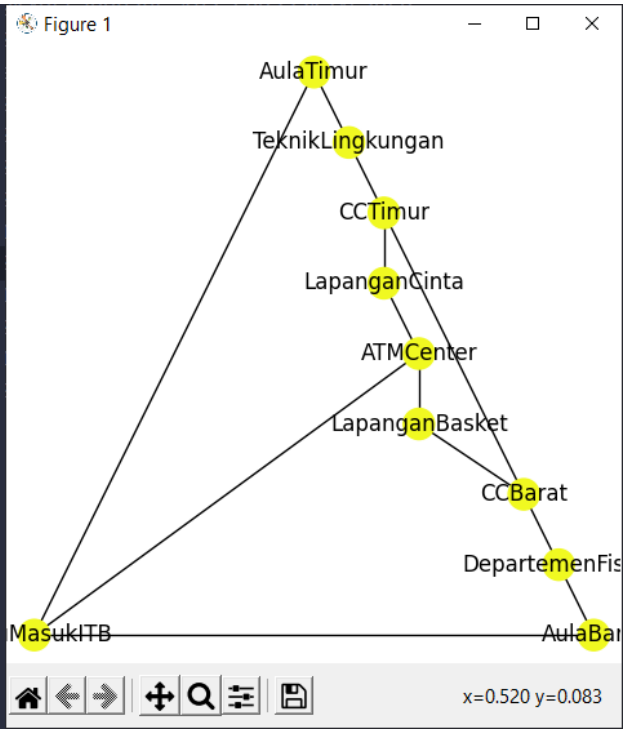
```
        print(nama[i], end=' ')
    print()
    sumDistance = measureSumDistance(path)
    print("dengan jarak yang ditempuh: " +
str(float("{:.2f}".format(sumDistance))) + " meter")
else:
    print()
    print("Tidak terdapat jalur yang menyambungkan simpul asal dan
tujuan")

pathdone = jalur(path)
drawFinalGraph(graphList)
```

II. Peta atau Graf Input

```
10
PintuMasukITB -6.893225 107.610447
Ciungwanara -6.893587 107.611944
Borromeus -6.893748 107.612985
Skanda -6.893346 107.610043
Tamansari -6.893873 107.608446
Sabuga -6.887704 107.609943
DayangSumbi -6.887227 107.611477
Scoop&Skoops -6.887376 107.613553
Mekdi -6.885208 107.613682
Siliwangi -6.884979 107.611504
0 1 0 1 0 0 0 0 0 0
1 0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 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 0 0 0 1 0 1 1 1
0 0 1 0 0 0 1 0 1 0
0 0 0 0 0 0 1 1 0 1
0 0 0 0 0 0 1 0 1 0
```



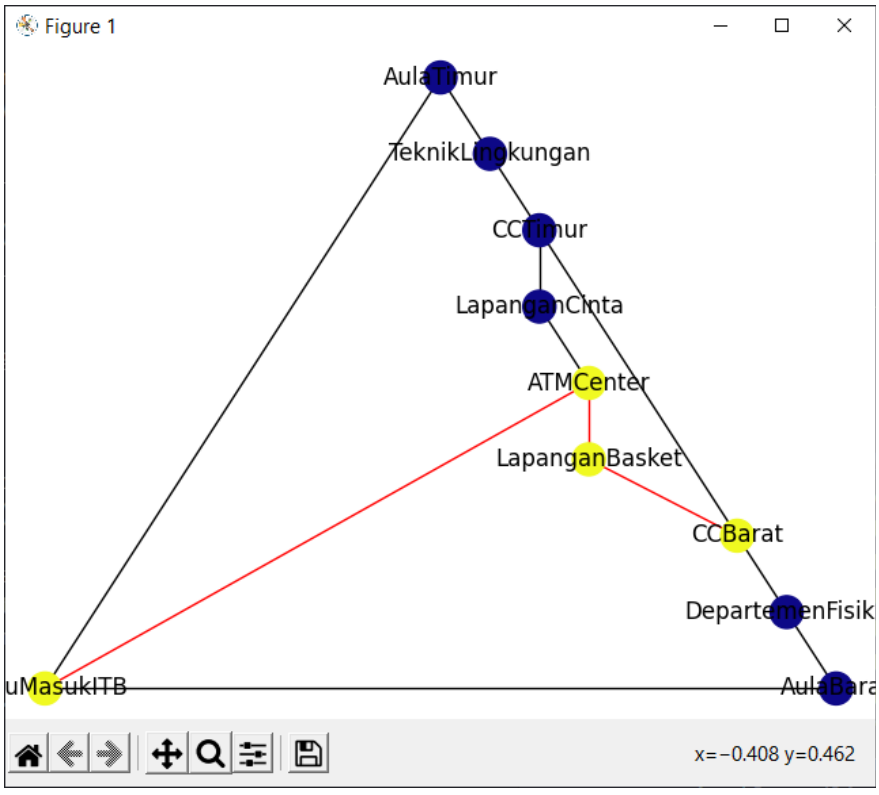

```
10
PintuMasukITB -6.892659329143158 107.61042200983438
AulaBarat -6.89262291581857 107.60874173131332
AulaTimur -6.89245443845861 107.61187380880716
ATMCenter -6.89194308407029 107.61039475853205
LapanganBasket -6.891936986979818 107.60976832885514
LapanganCinta -6.891868829079793 107.6109784965552
DepartemenFisika -6.891024383581631 107.60868317628682
CCBarat -6.8910149553990285 107.60972212467959
CCTimur -6.89101337605272 107.61105032829197
TeknikLingkungan -6.890974503626445 107.61210326886459
0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0 0 1
1 0 0 0 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0
0 0 0 1 0 0 0 0 1 0
0 1 0 0 0 0 0 1 0 0
0 0 0 0 1 0 1 0 1 0
0 0 0 0 0 1 0 1 0 1
0 0 1 0 0 0 0 0 1 0
```



III. *Screenshot* Hasil *Test Case*

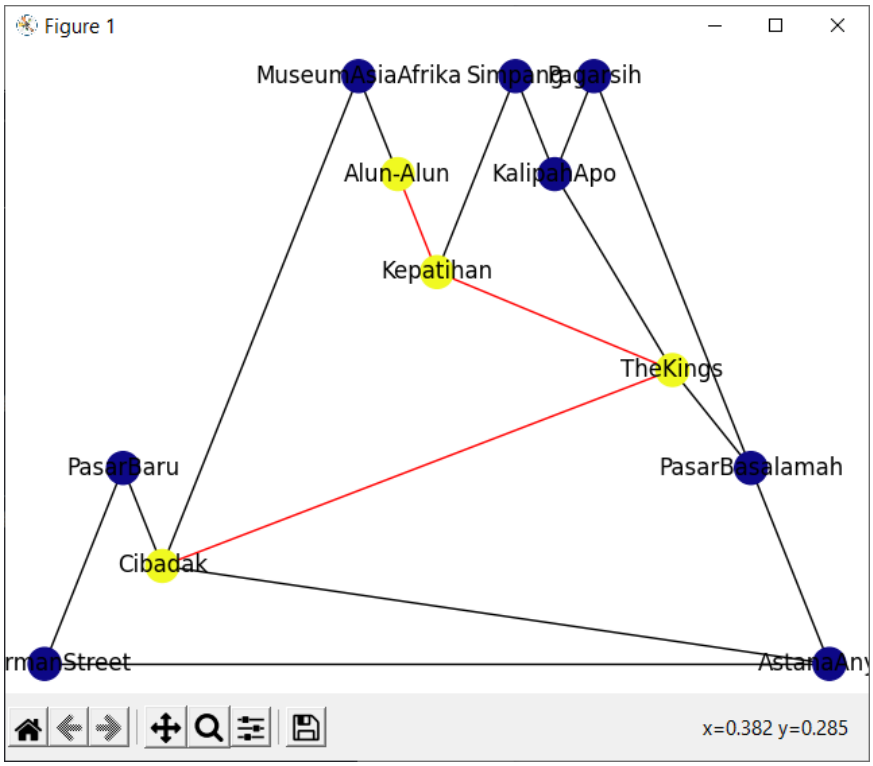
1. Peta jalan di dalam ITB

```
test1 : Peta jalan di dalam ITB
test2 : Peta jalan sekitar alun-alun Bandung
test3 : Peta jalan sekitar Buahbatu
test4 : Peta jalan sekitar Dago
test5 : Peta kota-kota di Romania
test6 : Peta tempat-tempat terkenal yang tidak saling berhubungan
Masukkan file yang ingin diuji (test1-test6): test1

Berikut simpul yang dapat dipilih: ['PintuMasukITB', 'AulaBarat', 'AulaTimur',
'ATMCenter', 'LapanganBasket', 'LapanganCinta', 'DepartemenFisika', 'CCBarat',
'CCTimur', 'TeknikLingkungan']Apakah ingin menampilkan graf peta? (Y/N) y

Tutup window yang terbuka untuk melanjutkan...
Masukkan simpul asal: PintuMasukITB
Masukkan simpul tujuan: CCBarat

Jalan yang dilalui adalah:
PintuMasukITB ATMCenter LapanganBasket CCBarat
dengan jarak yang ditempuh: 226.65 meter
```



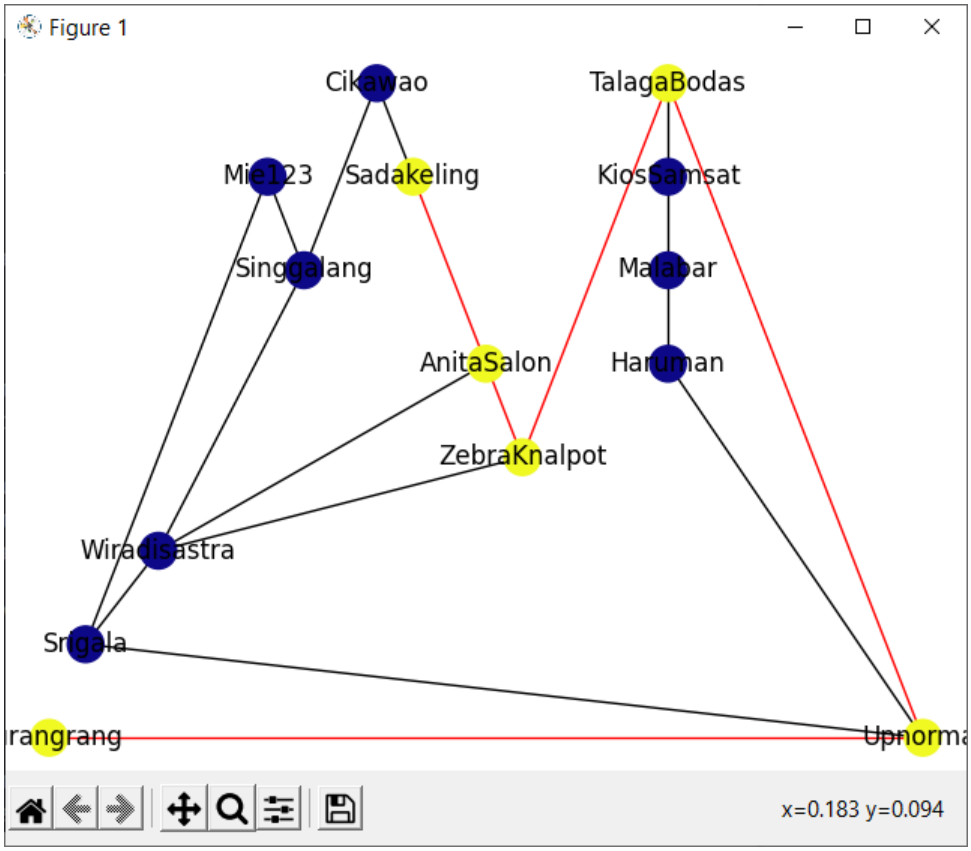
2. Peta jalan sekitar alun-alun Bandung

```
test1 : Peta jalan di dalam ITB
test2 : Peta jalan sekitar alun-alun Bandung
test3 : Peta jalan sekitar Buahbatu
test4 : Peta jalan sekitar Dago
test5 : Peta kota-kota di Romania
test6 : Peta tempat-tempat terkenal yang tidak saling berhubungan
Masukkan file yang ingin diuji (test1-test6): test2

Berikut simpul yang dapat dipilih: ['SudirmanStreet', 'PasarBaru', 'Cibadak', '
MuseumAsiaAfrika', 'Alun-Alun', 'Kepatihan', 'TheKings', 'Simpang', 'KalipahApo
', 'Pagarsih', 'PasarBasalamah', 'AstanaAnyar']
Apakah ingin menampilkan graf peta? (Y/N) y

Tutup window yang terbuka untuk melanjutkan...
Masukkan simpul asal: Alun-Alun
Masukkan simpul tujuan: Cibadak

Jalan yang dilalui adalah:
Alun-Alun Kepatihan TheKings Cibadak
dengan jarak yang ditempuh: 495.94 meter
```



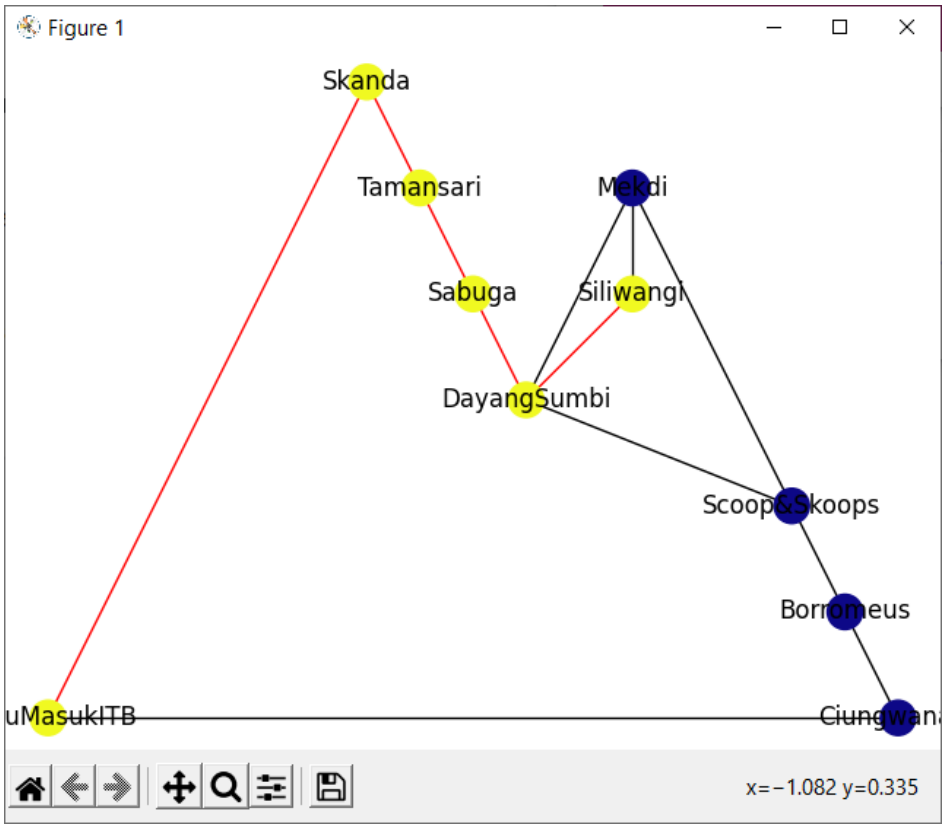
3. Peta jalan sekitar Buahbatu

```
test1 : Peta jalan di dalam ITB
test2 : Peta jalan sekitar alun-alun Bandung
test3 : Peta jalan sekitar Buahbatu
test4 : Peta jalan sekitar Dago
test5 : Peta kota-kota di Romania
test6 : Peta tempat-tempat terkenal yang tidak saling berhubungan
Masukkan file yang ingin diuji (test1-test6): test3

Berikut simpul yang dapat dipilih: ['Burangrang', 'Mie123', 'Upnormal', 'Srigal
a', 'Singgalang', 'Cikawao', 'Sadakeling', 'AnitaSalon', 'Wiradisastra', 'Zebra
Knalpot', 'TalagaBodas', 'Haruman', 'Malabar', 'KiosSamsat']

Apakah ingin menampilkan graf peta? (Y/N) n
Masukkan simpul asal: Burangrang
Masukkan simpul tujuan: Sadakeling

Jalan yang dilalui adalah:
Burangrang Upnormal TalagaBodas ZebraKnalpot AnitaSalon Sadakeling
dengan jarak yang ditempuh: 343.45 meter
```



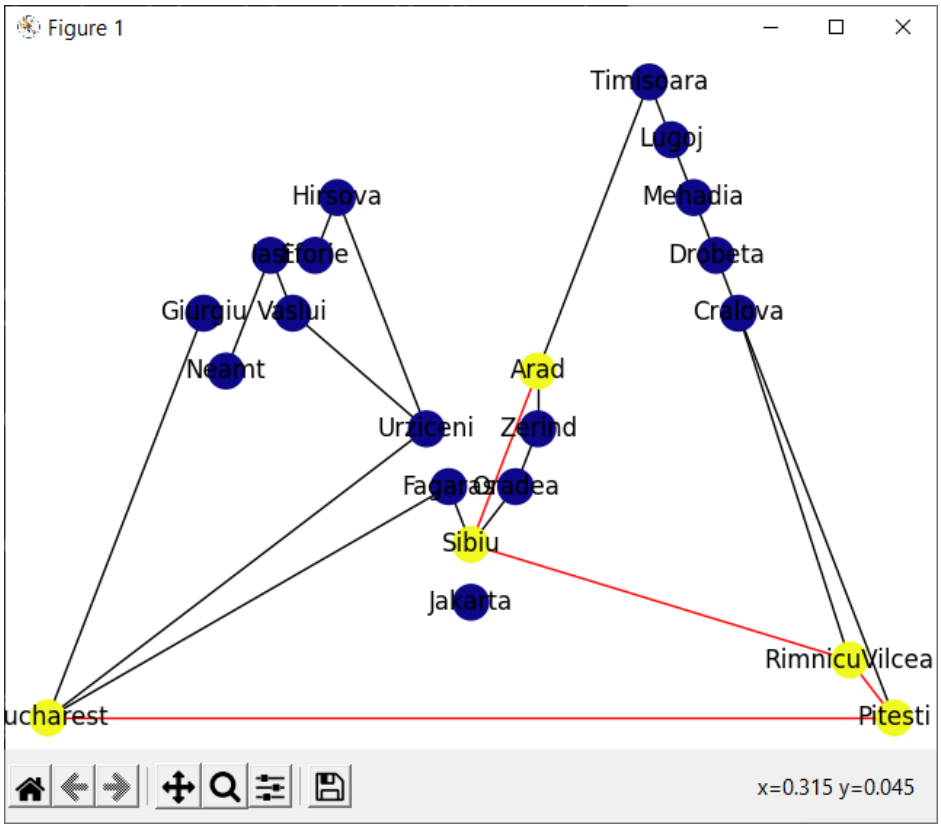
4. Peta jalan sekitar Dago

```
  /\/\   __ _ _   _    /\ \ \__ _  __ _ _ __   __ _(_)_ __     /___\_ __ ___
 /    \ / _` | | | |  /  \/ / _` |/ _` | '_ \ / _` | | '_ \   //  // '_ ` _ \
/ /\/\ \ (_| | |_| | / /\  / (_| | (_| | |_) | (_| | | | | | / \_//| | | | | |
\/    \/\__,_|\__,_| \_\ \/ \__, |\__,_| .__/ \__,_|_|_| |_| \___/ |_| |_| |_|
                             |___/      |_|
test1 : Peta jalan di dalam ITB
test2 : Peta jalan sekitar alun-alun Bandung
test3 : Peta jalan sekitar Buahbatu
test4 : Peta jalan sekitar Dago
test5 : Peta kota-kota di Romania
test6 : Peta tempat-tempat terkenal yang tidak saling berhubungan
Masukkan file yang ingin diuji (test1-test6): test4

Berikut simpul yang dapat dipilih: ['PintuMasukITB', 'Ciungwanara', 'Borromeus'
, 'Skanda', 'Tamansari', 'Sabuga', 'DayangSumbi', 'Scoop&Skoops', 'Mekdi', 'Sil
iwangi']

Apakah ingin menampilkan graf peta? (Y/N) n
Masukkan simpul asal: PintuMasukITB
Masukkan simpul tujuan: Siliwangi

Jalan yang dilalui adalah:
PintuMasukITB Skanda Tamansari Sabuga DayangSumbi Siliwangi
dengan jarak yang ditempuh: 1230.61 meter
```



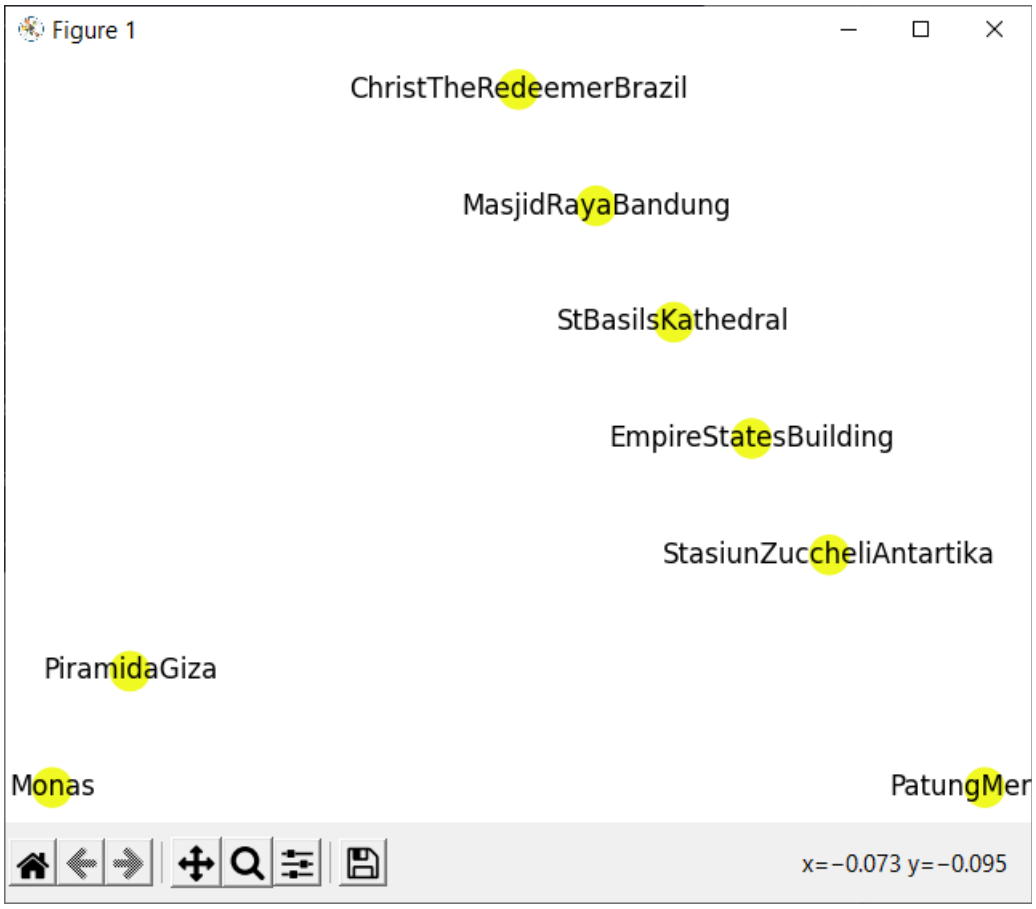
5. Peta jalan kota-kota di Romania

```
test1 : Peta jalan di dalam ITB
test2 : Peta jalan sekitar alun-alun Bandung
test3 : Peta jalan sekitar Buahbatu
test4 : Peta jalan sekitar Dago
test5 : Peta kota-kota di Romania
test6 : Peta tempat-tempat terkenal yang tidak saling berhubungan
Masukkan file yang ingin diuji (test1-test6): test5

Berikut simpul yang dapat dipilih: ['Bucharest', 'Arad', 'Zerind', 'Oradea', 'S
ibiu', 'Fagaras', 'Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Cralova', 'Rimni
cuVilcea', 'Pitesti', 'Giurgiu', 'Urziceni', 'Neamt', 'Iasi', 'Vaslui', 'Hirsov
a', 'Eforie', 'Jakarta']

Apakah ingin menampilkan graf peta? (Y/N) n
Masukkan simpul asal: Arad
Masukkan simpul tujuan: Bucharest

Jalan yang dilalui adalah:
Arad Sibiu RimnicuVilcea Pitesti Bucharest
dengan jarak yang ditempuh: 542138.05 meter
```



6. Peta tempat-tempat terkenal yang tidak saling berhubungan

```
test1 : Peta jalan di dalam ITB
test2 : Peta jalan sekitar alun-alun Bandung
test3 : Peta jalan sekitar Buahbatu
test4 : Peta jalan sekitar Dago
test5 : Peta kota-kota di Romania
test6 : Peta tempat-tempat terkenal yang tidak saling berhubungan
Masukkan file yang ingin diuji (test1-test6): test6

Berikut simpul yang dapat dipilih: ['Monas', 'PatungMerlion', 'StasiunZuccheliA
ntartika', 'EmpireStatesBuilding', 'StBasilsKathedral', 'MasjidRayaBandung', 'C
hristTheRedeemerBrazil', 'PiramidaGiza']

Apakah ingin menampilkan graf peta? (Y/N) n
Masukkan simpul asal: Monas
Masukkan simpul tujuan: PiramidaGiza

Tidak terdapat jalur yang menyambungkan simpul asal dan tujuan
```



IV. Tabel Kelengkapan

Centang (✓) jika ya

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	X

V. Alamat Tempat Kode Program

https://github.com/maximillian03/Mau-Ngapain-Om