

LAPORAN TUGAS KECIL 3 : ALGORITMA A*

Mata Kuliah IF2211 Strategi Algoritma

Dosen Pengampu: Rinaldi Munir, Nur Ulfa Maulidevi

Dwi Hendratmo Widyantoro, Rila Mandala



Disusun Oleh:

Girvin Junod	13519096
Hera Shafira	13519131

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

A. Kode Program

Algoritma A* yang diimplementasikan dalam program ini bekerja dengan menggunakan priority queue. Priority queue ini berisi array yang terdiri dari estimasi jarak ke tujuan, cost yang sudah ditempuh, path sejauh ini, dan node saat ini. Priority queuenya dibuat dengan prioritas melihat estimasi jarak saja dari yang terkecil. Jadi, program akan menerima sekumpulan koordinat sebagai node, node awal dan akhir pencarian, serta matrix adjacency untuk tiap node. Pertama dilakukan perhitungan haversine distance untuk tiap pasangan node. Digunakan haversine distance karena digunakan koordinat bumi yang aktual dan bumi bukanlah bidang euclidean sehingga lebih cocok memakai haversine formula untuk mencari jarak antara dua titik di permukaan bumi. Setelah itu node awal akan langsung dimasukkan ke priority queue dalam bentuk array tadi. Lalu akan dilakukan dequeue dari priority queue, dari hasil dequeue ini akan di-enqueue semua node tetangga dari node yang di-dequeue dalam bentuk array tadi yang sudah ada nilai estimasi jaraknya. Proses dequeue dan enqueue ini diloop sampai node yang di-dequeue adalah node target atau priority queue habis. Priority queue dapat habis ketika tidak ada jalur dari node awal ke tujuan. Tiap node yang di-dequeue juga dimasukkan ke list node visited dan node yang telah masuk list visited tidak bisa dimasukkan lagi ke priority queue. Ini dilakukan untuk meng-handle kasus tidak ada jalur. Jika node target ketemu sebelum priority queue habis maka ditemukan jalur sehingga outputnya berupa boolean true, jalur dari node awal ke akhir, dan jaraknya. Jika tidak maka outputnya boolean false sehingga akan ditampilkan bahwa tidak ada jalur antara kedua node.

File normal.py (main program non-bonus)

```
#Tucil Stima 3

#Girvin Junod 13519096 K02
#Hera Shafira 13519131 K03

import os
from pathlib import Path
from prioqueue import PrioQueue
from peta import MapV
from haversine import haversineDistance

def euclideanDistance(a, b): #gak dipake karena pakai koordinat bumi,
jadi ga bisa euclidean karena bumi ga datar
    return ((a[0] - b[0])**2 + (a[1] - b[1])**2)**0.5
def parsefile(namafile):
    #input file
```

```

filepath = os.path.dirname(Path(__file__).absolute().parent)
namafile1 = os.path.join(filepath, 'test', namafile)
f = open(namafile1) #baca soal
read = f.read().split('\n') #dipisahkan dari newline
listkoordinat = []
nama = []
matrixjalan = []
n = int(read[0])
#liat format di txt
for i in range(1, n+1):
    idxpostnama = read[i].index("|") #pisahin berdasarkan -
    nama.append(read[i][0:idxpostnama-1]) #isi list nama
    read[i] = read[i].replace(" ", "") #hapus spasi
    idxpostnama = read[i].index("|") + 1 #ambil indeks koordinat
    read[i] = read[i][idxpostnama:] #jadi koordinat aj
    templist = read[i].split(",") #pisahin berdasarkan koma
    listkoordinat.append((float(templist[0]), float(templist[1])))
#dibuat jadi tuple dan dimasukin ke list
nexti = i+1 #buat ngeliat matriks jalan di input
for i in range(nexti, len(read)):
    templist = read[i].split(" ") #pisahin dari spasi
    for j in range(len(templist)):
        templist[j] = int(templist[j]) #konversi input dari str ke
int, biar jadi boolean
    matrixjalan.append(templist)

#matrix buat jarak euclidean (jarak garis lurus antar simpul)
#Sebenarnya karena bumi ga datar, pakai haversine formula buat jarak
antar 2 simpul, ga euclidean
matrixgeo = [[0 for j in range(n)] for i in range(n)]
for i in range(n):
    for j in range(n):
        matrixgeo[i][j] = haversineDistance(listkoordinat[i],
listkoordinat[j])
return nama, matrixjalan, matrixgeo, listkoordinat

def inputnode(nama):
    #input node awal dan akhir
    print("Pilihan Node: ")

```

```

n = len(nama)
for i in range (n):
    print(str(i+1) + ": " + nama[i])
while(1):
    try:
        entry = int(input("Masukkan node awal (nomor): "))
        target = int(input("Masukkan node target (nomor): "))
        if (entry > n or entry < 1 or target > n or target < 1):
            print("input invalid, please try again")
            print()
        else:
            entry = nama[entry-1]
            target = nama[target-1]
            break
    except:
        print("input invalid, please try again")
return entry, target

def astar(nama, matrixjalan, matrixgeo,listkoordinat, entry, target):
    #Mulai algo A*
    listvisited = [] #Cek apakah sudah di visit atau blom, hanya untuk
handle kasus tidak ada jalur
    idxawal = nama.index(entry) #indeks awal
    idxtujuan = nama.index(target) #indeks tujuan
    #list awal
    #list[estimasi, sejauhini, path, nama]
    awal = [matrixgeo[idxawal][idxtujuan] ,0, [entry], entry]

    #prioqueue untuk A*, prioritas dari estimasi
    listjalan = PrioQueue()
    listjalan.enqueue(awal) #prioqueue dari list
    found = False
    while not listjalan.isEmpty():
        popped = listjalan.dequeue() #dequeue dari list
        listvisited.append(popped[3])
        if (popped[3] == target): #kalau sudah ketemu target di end
            found = True
            break
        path = popped[2] #ambil path sejauh ini

```

```

indeks = nama.index(popped[3]) #ambil indeks nama dari yang
didequeue
    for i in range(len(matrixjalan[indeks])):
        if (matrixjalan[indeks][i] and nama[i] not in listvisited):
#jika ada jalur
           sofar = popped[1] + matrixgeo[indeks][i]
            isipath = []
            for j in range(len(path)):
                isipath.append(path[j])
            isipath.append(nama[i])
            grup = [matrixgeo[i][idxtujuan] + sofar, sofar, isipath,
nama[i]]
            listjalan.enqueue(grup) #Masukkan ke prioqueue
    return popped, found

def output(popped, found, listkoordinat, matrixjalan, nama, entry,
target):
    #Isi latitude longitude buat map
    listlat = []
    listlon = []
    for i in range(len(matrixjalan)):
        listlat.append(listkoordinat[i][0]) #list latitude
        listlon.append(listkoordinat[i][1]) #list longitude
    #Buat map
    map = MapV(listlat, listlon, nama)

    #Tambah simpul awal akhir di map
    nwal = [entry]
    latawal = [listkoordinat[nama.index(entry)][0]]
    lonawal = [listkoordinat[nama.index(entry)][1]]
    nakhir = [target]
    latakhir = [listkoordinat[nama.index(target)][0]]
    lonakhir = [listkoordinat[nama.index(target)][1]]
    map.tambahAwal(latawal, lonawal, nwal)
    map.tambahAkhir(latakhir, lonakhir, nakhir)

    #Isi jalur di map
    for i in range(len(matrixjalan)):
        for j in range(len(matrixjalan)):
```

```

listlat = []
listlon = []
name = []
if (matrixjalan[i][j]):
    listlat.append(listkoordinat[i][0])
    listlat.append(listkoordinat[j][0])
    listlon.append(listkoordinat[i][1])
    listlon.append(listkoordinat[j][1])
    name.append(nama[i])
    name.append(nama[j])
    map.tambahjalur(listlat, listlon, name)
if found: #ada jalur
    hasil = popped[2] #Path dari awal ke akhir
    distance = popped[1] #total jarak akhir

    #memasukkan hasil ke map
    listlathasil = []
    listlonhasil = []
    namahasil = []
    for i in range(len(hasil)):
        idxhasil = nama.index(hasil[i])
        listlathasil.append(listkoordinat[idxhasil][0])
        listlonhasil.append(listkoordinat[idxhasil][1])
        namahasil.append(nama[idxhasil])

    map.tambahjalurhasil(listlathasil, listlonhasil, hasil)

    #output dari jalur dan distancenya
    print("Jalur terpendek:")
    for i in hasil:
        if i == target:
            print(i)
        else:
            print(i, end = " -> ")
    print("Panjang jalur:", distance, "km")
    print("Menampilkan map di browser Anda...")
    distanceformat = ('%.5f' % distance).rstrip('0').rstrip('.')
#buat jadi maks 5 angka dibelakang koma
    map.visualize(latawal[0], lonawal[0], ('Peta Jalur Terdekat (' +

```

```

entry + ' -> ' + target + ': ' + distanceformat + ' km')) #panggil map
dengan center mengarah ke simpul awal
else: #tidak ada jalur
    print("Tidak ada jalur")
    print("Menampilkan map di browser Anda...")
    map.visualize(lataawal[0], lonawal[0],'Tidak ada jalur antara
kedua simpul (' + entry + ' -> ' + target + ')') #panggil map dengan
tidak ada solusi

#main
namafile = input("Nama file: ") #input file
nama, matrixjalan, matrixgeo, listkoordinat = parsefile(namafile) #parse
input
entry, target = inputnode(nama) #isi input node awal dan akhir
popped, found = astar(nama, matrixjalan, matrixgeo, listkoordinat, entry,
target) #algo astar
output(popped, found, listkoordinat, matrixjalan, nama, entry, target)
#output, diprint dan di map

```

File abintang.py (main program bonus)

```

from prioqueue import PrioQueue
from haversine import haversineDistance

def buatmatrixdistance(matrixjalan, listkoordinat):
    n = len(matrixjalan)
    matrixgeo = [[0 for j in range(n)] for i in range(n)]
    for i in range(n):
        for j in range(n):
            matrixgeo[i][j] = haversineDistance(listkoordinat[i],
listkoordinat[j])
    return matrixgeo

def astar2(entry,target, matrixjalan, matrixgeo, listkoordinat):
    #Mulai algo A*
    listvisited = [] #Cek apakah sudah di visit atau blom, hanya untuk
handle kasus tidak ada jalur
    idxawal = listkoordinat.index(entry) #indeks awal

```

```

idxtujuan = listkoordinat.index(target) #indeks tujuan
#list awal
#list[estimasi, sejauhini, path, nama]
awal = [matrixgeo[idxawal][idxtujuan] ,0, [entry], entry]
#prioqueue untuk A*, prioritas dari estimasi
listjalan = PrioQueue()
listjalan.enqueue(awal) #prioqueue dari list
found = False
while not listjalan.isEmpty():
    popped = listjalan.dequeue() #dequeue dari list
    listvisited.append(popped[3])
    if (popped[3] == target): #kalau sudah ketemu target di end
        found = True
        break
    path = popped[2] #ambil path sejauh ini
    indeks = listkoordinat.index(popped[3]) #ambil indeks nama dari
yang didequeue
    for i in range(len(matrixjalan[indeks])):
        if (matrixjalan[indeks][i] and listkoordinat[i] not in
listvisited): #jika ada jalur
            sofar = popped[1] + matrixgeo[indeks][i]
            isipath = []
            for j in range(len(path)):
                isipath.append(path[j])
            isipath.append(listkoordinat[i])
            grup = [matrixgeo[i][idxtujuan] + sofar, sofar, isipath,
listkoordinat[i]]
            listjalan.enqueue(grup) #Masukkan ke prioqueue
    listedge = []
if found:
    jalan = popped[2]
    distance = popped[1]
    for i in range(len(jalan) - 1):
        listedge.append([jalan[i], jalan[i+1]])
return jalan, listedge, found, distance
return popped, listedge, found, 0

```

File haversine.py

```
from math import pi,sin,cos,asin
def haversineDistance(koor1, koor2):
    #lengkapnya formulanya ada di
https://en.wikipedia.org/wiki/Haversine\_formula
    lat1 = koor1[0]
    lon1 = koor1[1]
    lat2 = koor2[0]
    lon2 = koor2[1]

    rad = pi / 180.0 #ubah dari derajat ke radian
    radiusbumi = 6371 #radiusbumi dalam km

    radlat = (lat2 - lat1) * rad
    radlon = (lon2 - lon1) * rad

    #fungsi haversine -> hav(x) = sin(x/2)**2
    #hav(θ) = hav(lat2-lat1) + cos(lat1)cos(lat2)hav(lon2-lon1)
    #θ = d/r, r = radius, d = distance
    #d = 2*radiusbumi * arcsin(hav(θ)**0.5)
    hav = (sin(radlat / 2))**2 + cos(lat1 * rad) * cos(lat2 * rad) *
(sin(radlon / 2))**2
    d = 2*radiusbumi * asin(hav**0.5)

    return d

#driver
if __name__ == '__main__':
    a = (44.457, 26.093)
    b = (46.181, 21.312)
    c = (-6.176716476545919, 106.98596393874463)
    d = (-6.176767387686766, 106.98672068536122)
    r = haversineDistance(c,d)
    print(r)
```

File prioqueue.py

```
class PrioQueue:
    #ctor
    def __init__(self):
        self.prioqueue = []

    #buat print prioqueue
    def __str__(self):
        return ','.join([str(i) for i in self.prioqueue])

    # cek kosong
    def isEmpty(self):
        return len(self.prioqueue) == 0

    # enqueue elemen, prionya ada di dequeue jadi ga usah diurutin di
enqueue
    def enqueue(self, data):
        self.prioqueue.append(data)

    # dequeue dari yg terkecil
    def dequeue(self):
        min = 0
        for i in range(len(self.prioqueue)):
            if self.prioqueue[i][0] < self.prioqueue[min][0]:
                min = i
        pilihmin = self.prioqueue[min]
        del self.prioqueue[min]
        return pilihmin

#driver
if __name__ == '__main__':
    #list[estimasi, sejauhini, path, nama]
    tes = PrioQueue()
    tes.enqueue([200, 100, ['a', 'b'], 'b'])
    tes.enqueue([150, 0, ['a'], 'a'])
    tes.enqueue([300, 100, ['a', 'b'], 'b'])
    print(tes)
    while not tes.isEmpty():
```

```
a = tes.dequeue()
print(a)
```

File peta.py (plotting graf output non-bonus)

```
import plotly.graph_objects as go
#Menggunakan plotly dan mapbox untuk visualisasi map

#px.set_mapbox_access_token('pk.eyJ1IjoiZ2lydmluanVub2QiLCJhIjoiY2tuMDB6
#zMfyMGpjOTJubW82ZWJ3em1heCJ9.k5m3PeaIZjpWBM1QUAKeNQ')
class MapV:
    def __init__(self, a, b, c): #ctor
        self.mapbox_access_token =
'pk.eyJ1IjoiZ2lydmluanVub2QiLCJhIjoiY2tuMDB6ZmfyMGpjOTJubW82ZWJ3em1heCJ9
.k5m3PeaIZjpWBM1QUAKeNQ'
        self.map= go.Figure(go.Scattermapbox(
            lat=a,
            lon=b,
            mode='markers',
            marker= dict(color = 'black', size = 14),
            text = c,
            name = "Simpul"
        ))
        self.count = 0

    def __str__(self): #untuk print(peta)
        return 'ini peta'

    def tambahAwal(self, latawal, lonawal, nawal): #tambah simpul awal
        self.map.add_trace(go.Scattermapbox(
            mode = "markers",
            lat = latawal,
            lon = lonawal,
            marker = dict(color = 'LightSkyBlue', size = 16),
            text = nawal,
            name = "Awal"
        ))

    def tambahAkhir(self, latahir, lonakhir, nakhir): #tambah simpul
```

```
akhir

    self.map.add_trace(go.Scattermapbox(
        mode = "markers",
        lat = latahir,
        lon = lonahir,
        marker = dict(color = 'purple', size = 16),
        text = nahir,
        name = "Akhir"
    ) )

def tambahjalur(self, jalurlat, jalurlon, name): #tambah jalur
    if (self.count > 0):
        self.map.add_trace(go.Scattermapbox(
            mode = "lines",
            lat = jalurlat,
            lon = jalurlon,
            line = dict(color = 'black', width = 2),
            text = name,
            legendgroup="a",
            name = "Jalur",
            showlegend = False
        ) )
    else:
        self.map.add_trace(go.Scattermapbox(
            mode = "lines",
            lat = jalurlat,
            lon = jalurlon,
            line = dict(color = 'black', width = 2),
            text = name,
            legendgroup="a",
            name = "jalur"
        ) )
    self.count +=1

def tambahjalurhasil(self, jalurlat, jalurlon, hasil): #tambah jalur
    hasil

        self.map.add_trace(go.Scattermapbox(
            mode = "lines",
            lat = jalurlat,
```

```

        lon = jalurlon,
        line = dict(color = 'red', width = 4),
        name = "Jalur terdekat",
        text = hasil
    ))
}

def visualize(self, latawal, lonawal, judul): #gambar map dengan
center simpul awal
    self.map.update_layout(
        hovermode='closest',
        title = judul,
        mapbox=dict(
            accesstoken=self.mapbox_access_token,
            bearing=0,
            center=go.layout.mapbox.Center(
                lat= latawal,
                lon= lonawal
            ),
            pitch=0,
            zoom= 15
        )
    )
    self.map.show()

#driver
if __name__ == '__main__':
    a = MapV([46.181, 44.457, 45.794], [21.312, 26.093, 24.128],
['Arad', 'Bucharest', 'Sibiu'])
    print(a)
    a.tambahjalur([46.181, 44.457],[21.312, 26.093], ['tes'] )
    a.visualize(46.181, 21.312, 'tes')

```

File app.py (Flask penghubung frontend ke backend untuk bonus)

```

from flask import Flask, flash, request, redirect, url_for,
session,jsonify
from flask.wrappers import Response
from flask_cors import CORS, cross_origin
import json

```

```
from abintang import buatmatrixdistance
from abintang import astar2

# configuration
DEBUG = True

# instantiate the app
app = Flask(__name__)
app.config['CORS_HEADERS'] = 'Content-Type'
app.config.from_object(__name__)

# enable CORS
CORS(app, resources={r"/*": {"origins": "*"}})

# sanity check route
def matrixadj(listedge, listnode):
    matrixadj = [[0 for i in range(len(listnode))] for j in
range(len(listnode))]
    for i in listedge:
        awal, akhir = i
        idxawal = listnode.index(awal)
        idxakhir = listnode.index(akhir)
        matrixadj[idxawal][idxakhir] = 1
        matrixadj[idxakhir][idxawal] = 1
    return matrixadj

#POST method from client to server
@app.route('/', methods=['GET', 'POST', 'DELETE'])
@cross_origin(origin='*',headers=['Content-Type','Authorization'])
def getter():
    data = request.get_json()
    bismillah = matrixadj(data["edges"], data["nodes"])
    matrixdistance = buatmatrixdistance(bismillah, data["nodes"])
    astar = astar2(data["selected"][0], data["selected"][1], bismillah,
matrixdistance, data["nodes"])
```

```

        print(astar[1])
        return json.dumps([astar[1],astar[2],astar[3]])

if __name__ == '__main__':
    app.run()

```

File Openstreetmap.vue (main component vue dari web lokal untuk bonus)

```

<template>
    <div class="openstreetmap">
        <link rel="preconnect" href="https://fonts.gstatic.com">
        <link
            href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap"
            rel="stylesheet">
        <p class="title">Stimap</p>
        <button class="done" @click="isClickedB=true">Selesai membuat
        graf</button>
        <button class="send" @click="sendList()">OK!</button>
        <p v-if="counterSelected<=2 && isClickedB==true">Pilih 2
        titik!</p>
        <p v-if="counterSelected>2">Kamu sudah memilih 2 titik</p>
        <p v-if="cost!=0">Jarak : {{cost}} km</p>
        <p v-if="isFound==false">Kedua titik tidak berhubungan</p>
        <l-map style="height: 450px" :zoom="zoom" :center="center"
        @click="onMapClick">
            <l-tile-layer :url="url"
            :attribution="attribution"></l-tile-layer>
            <l-marker v-for="marker,index in markers" :lat-lng="marker"
            :icon="iconNormal" v-bind:key="index" @click="markerClick"></l-marker>
            <l-polyline v-for="elem,index in nodes" :lat-lngs="elem[0]"
            :color="elem[1]" v-bind:key="index"></l-polyline>
            <l-polyline v-for="e,i in astar" :lat-lngs="e[0]"
            :color="e[1]" v-bind:key="i"></l-polyline>
        </l-map>
    </div>
</template>
<style type="text/css">
@import

```

```
"https://unpkg.com/leaflet-geosearch@2.6.0/assets/css/leaflet.css";\n\n#app{\n    margin-top:0px !important;\n}\n.title{\n    font-family:'Pacifico', cursive;\n    font-size:60px;\n    margin-bottom:20px;\n    margin-top:0px;\n}\n.leaflet-popup-close-button {\n    display: none;\n}\n.done{\n    margin-right:7px;\n    margin-bottom:15px;\n    background-color:#43cacc;\n    color : white;\n    padding:10px;\n    border : none;\n    border-radius:5px;\n    font-size:18px;\n    cursor:pointer;\n}\n\n.send{\n    color : white;\n    background-color:#db932e;\n    padding:10px;\n    border : none;\n    border-radius:5px;\n    font-size:18px;\n    cursor:pointer;\n}\n\n</style>\n<script>\n    import { LMap, LTileLayer, LMarker, LPolyline} from 'vue2-leaflet';\n
```

```
import L from 'leaflet'
import 'leaflet/dist/leaflet.css'
import json from '../node.json'
import { OpenStreetMapProvider } from 'leaflet-geosearch';

delete L.Icon.Default.prototype._getIconUrl

export default {
  name: 'Openstreetmap',
  components: {
    LMap,
    LTileLayer,
    LMarker,
    LPolyline
  },
  data: () => ({
    isClickedB:false,
    zoom:20,
    center: L.latLng(-6.890364997716474, 107.61034998436544),
    url:'http://{s}.tile.osm.org/{z}/{x}/{y}.png',
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, +  

      '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, +  

      'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
    id: 'mapbox/streets-v11',
    myJson : json,
    newLoc: '',
    newLt : 0,
    newLng : 0,
    counter : 0,
    counterSelected : 0,
    iconNormal: L.icon({
      iconUrl:
"https://raw.githubusercontent.com/pointhi/leaflet-color-markers/master/img/marker-icon-blue.png",
      iconAnchor: [12, 41],
      popupAnchor: [0, -41]
```

```
        } ) ,
        iconAsal: L.icon({
            iconUrl:
"https://raw.githubusercontent.com/pointhi/leaflet-color-markers/master/
img/marker-icon-red.png",
            iconAnchor: [12, 41],
            popupAnchor: [0, -41]
        ) ,
        node:[],
        nodes:[],
        markers:[],
        selected:[],
        edges:[],
        astar:[],
        cost:0,
        isFound:null,
        geosearchOptions: { // Important part Here
            provider: new OpenStreetMapProvider(),
        },
    ) ,
    mounted ()  {

        this.$nextTick(() => {
            this.$refs.marker.mapObject.openPopup();
        });
        this.newLoc = this.myJson[0].name
        this.newLt = this.myJson[0].pos[0]
        this.newLng = this.myJson[0].pos[1]
        console.log(this.isClickedB)
    },
    methods: {
        onMapClick(e) {
            if (this.isClickedB==false) {
                this.markers.push([e.latlng.lat, e.latlng.lng]);
                console.log(this.markers);
                this.counter=this.counter+1
                if (this.counter%2!=0){
                    this.node.push([e.latlng.lat,e.latlng.lng]);
                }
            }
        }
    }
}
```

```

        else if (this.counter%2==0) {
            this.node.push([e.latlng.lat,e.latlng.lng]);
            this.nodes.push([this.node, "red"]);
            this.node=[];

        }
    }
} ,

markerClick(e) {
    if (this.isClickedB==false) {
        this.counter=this.counter+1
        if (this.counter%2!=0) {
            this.node.push([e.latlng.lat,e.latlng.lng]);
        }

        else if (this.counter%2==0) {
            this.node.push([e.latlng.lat,e.latlng.lng]);
            this.nodes.push([this.node, "red"]);
            console.log(this.nodes)
            this.node=[];
        }
    }
    else{
        if (this.counterSelected < 2) {
            this.selected.push([e.latlng.lat,e.latlng.lng]);
            e.target.setIcon(this.iconAsal);
        }
        this.counterSelected=this.counterSelected+1;
    }
} ,

sendList() {
    const stripNodes = this.nodes.map(e => e[0])
    var nodes_json = JSON.stringify({ "edges" : stripNodes,
"nodes":this.markers, "selected":this.selected})

    fetch('http://127.0.0.1:5000/', {

```

```

        method: 'POST',
        body: nodes_json,
        headers: {
          'Accept': 'application/json, text/plain, */*',
          'Content-Type': 'application/json'
        }
      }).then (response => {
        response.json().then(data => {
          this.astar=data[0].map(e => [e,"blue"]);
          this.cost = data[2];
          this.isFound = data[1];
          console.log(this.cost)
        }) ;
      }).catch(
        error => console.error(error)
      );
    }
  }
</script>

```

B. Peta/Graf Input

Nama file : alun2bandung.txt

```

12
Kebon Jati - Gardujati | -6.916289558795759, 107.59823109564569
Kebon Jati - Otista | -6.915799623402846, 107.60449846018152
Sudirman - Asia Afrika - Otista | -6.920773509104472,107.60406918863798
Asia Afrika - Alun-alun | -6.921231487380398 ,107.60771799675814
Alun-alun - Dalem Kaum | -6.9225308651860935, 107.60765360602662
Dalem Kaum - Dewi Sartika | -6.92241370831757, 107.6063559748914
Dalem Kaum - Otista - Cibadak | -6.9220622375374745, 107.60401552969505
Dewi Sartika - Simpang | -6.9241710582913045, 107.6062057298512
Simpang - Otista - Kalipah Apo | -6.9238089382260455,107.60386619993885
Kalipah Apo- Astana Anyar | -6.9232540419560005,107.59866591318783
Cibadak - Astana Anyar | -6.921423199084892, 107.59846719499465
Sudirman - Astana Anyar | -6.9201238182298335,107.59829548637724
0 1 0 0 0 0 0 0 0 0 1
1 0 1 0 0 0 0 0 0 0 0
0 1 0 1 0 0 1 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
0 0 0 0 1 0 1 1 0 0 0 0
0 0 1 0 0 1 0 0 1 0 1 0
0 0 0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 1 1 0 1 0 0
0 0 0 0 0 0 0 0 1 0 1 0
0 0 0 0 0 0 0 1 0 0 1 0
1 0 1 0 0 0 0 0 0 1 0

Nama file : buahbatu.txt

14

Batagor Riri -6.927562192722218,107.61959799634626 Karapitan - Emung -6.927025406721871,107.61690969806743 Burangrang - Emung -6.927472728417712,107.61945313196297 Emong - Emur -6.927312970717906,107.61796141334914 Karapitan - Emur -6.927504679951173,107.61675945302719 Karapitan - Cikawao -6.928793389996319,107.61639457221517 Cikawao - Sadakeling -6.92914485576096,107.61707067489624 Sadakeling - Emur -6.928537778366568,107.61768238684581 Emur - Wiradisastra -6.927824195166772,107.61782190009745 Wiradisastra - Sadakeling -6.928217198601289,107.61855260357831 Sadakeling - Talaga Bodas - Burangrang -6.928046790655269,107.61943701951742 Burangrang - Haruman -6.927077594291167,107.61945311720032 Haruman - Malabar -6.927871057396435,107.62254923820818 Malabar - Talaga Bodas -6.928414232712092,107.62231850475352

0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 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 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 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 0
--

Nama file : bucharest.txt

20
Bucharest | 44.457, 26.093
Arad | 46.181, 21.312
Zerind | 46.624, 21.518
Oradea | 47.089, 21.907
Sibiu | 45.794, 24.128
Fagaras | 45.842, 24.973
Timisoara | 45.756, 21.231
Lugoj | 45.691, 21.903
Mehadia | 44.904, 22.365
Drobeta | 44.639, 22.659
Cralova | 44.319, 23.794
Rimnicu Vilcea | 45.099, 24.369
Pitesti | 44.856, 24.869
Giurgiu | 43.905, 25.969
Urziceni | 44.718, 26.645
Neamt | 47.056, 26.506
Iasi | 47.158, 27.598
Vaslui | 46.641, 27.728
Hirsova | 44.690, 27.945
Eforie | 44.049, 28.653
0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 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 1 0 1 0 0 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 0 0 0 0
0 1 0 1 0 1 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 0 0 0 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 0
0 0 0 0 0 0 1 0 1 0 0 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 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 1 1 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 0 0 0 0 1 1 0
0 0 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 0 0 0 0 0 0 0 1 0 1 0 0
0 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 0 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 0 0 0 0 0 1 0 0 0 0 1
0 1 0

Nama file : dago.txt

29

Skywalk | -6.885112680505222,107.61019953300554
Siliwangi | -6.884804853041526,107.60914368104865
Siliwangi - Skywalk | -6.884783550096599,107.61119378599209
Taman Sari - Siliwangi | -6.884836807457129,107.6114190210842
Sumur Bandung - Siliwangi | -6.885369380733616,107.61299685054328
Taman Sari - Sumur Bandung | -6.886285405370851,107.61177322769747
Dayang Sumbi - Taman Sari | -6.887212079657689,107.61149415582035
Dayang Sumbi - Dago | -6.887393153732213,107.61359792843244
Dago - DU | -6.8852628661261415,107.6137052637698
Pintu Belakang ITB | -6.8877467805165,107.61033651729362
Tamfes | -6.887851164318019,107.60829556276724
Pintu Samping ITB | -6.890130559259928,107.60805942502509
Ganesha - Taman Sari | -6.893826564216093,107.60842402693942
Taman Sari - Gelap Nyawang | -6.894912992468297,107.60890754800417
Pintu Depan ITB | -6.893250330197495,107.61047343867449
Ganesha - Skanda | -6.8933366055619265,107.61000236844517
Skanda - Gelap Nyawang | -6.8947958287570135,107.61016337145116
Ciung Wanara - Gelap Nyawang | -6.8947681355281825,107.61168076489075
Ciung Wanara - Ganesha | -6.893655078760995,107.61195442549976
Ganesha - Dago | -6.893796740627806,107.6129552928273
Dago - Hasanudin | -6.894785177509091,107.6129755572898
Taman Sari - Pasupati | -6.898246820474195,107.60954082649454
Pasupati - Dago | -6.898960448340111,107.61283602135124
Hasanudin - DU | -6.894785177509091,107.61740850672243
DU - Teuku Umar | -6.892346035431296,107.6178807822068
Dago - Teuku Umar | -6.891515233080349,107.61326536270067
Teuku Umar - Raden Patah | -6.891621746283635,107.61460487317066
DU - Raden Patah | -6.8874996678613964,107.6152295993742
Gedung STEI | -6.890580892371696, 107.61100792249937
0 1 1 0
1 0 1 0
1 1 0 1 0
0 0 1 0 1 1 0
0 0 0 1 0 1 0
0 0 0 1 1 0 1 0
0 0 0 0 0 1 0 1 0
0 0 0 0 0 0 1 0 1 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 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 0 0 0 0 0 0 0 0 0 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 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 0 0 0 0 0

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

Nama file : dalamitb.txt

```
10  
Pintu Utama ITB | -6.892659329143158, 107.61042200983438  
Simpang Aula Barat | -6.892622915818573, 107.60874173131332  
Simpang Aula Timur | -6.89245443845861, 107.61187380880716  
Simpang Tengah ITB | -6.891943084070294, 107.61039475853205  
Simpang Lapangan Basket | -6.891936986979818, 107.60976832885514  
Simpang Lapangan Cinta | -6.891868829079793, 107.6109784965552  
Simpang Departemen Fisika | -6.891024383581631, 107.60868317628682  
Simpang CC Barat | -6.8910149553990285, 107.60972212467959  
Simpang CC Timur | -6.89101337605272, 107.61105032829197  
Simpang Teknik Lingkungan | -6.890974503626445, 107.61210326886459  
0 1 1 1 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
```

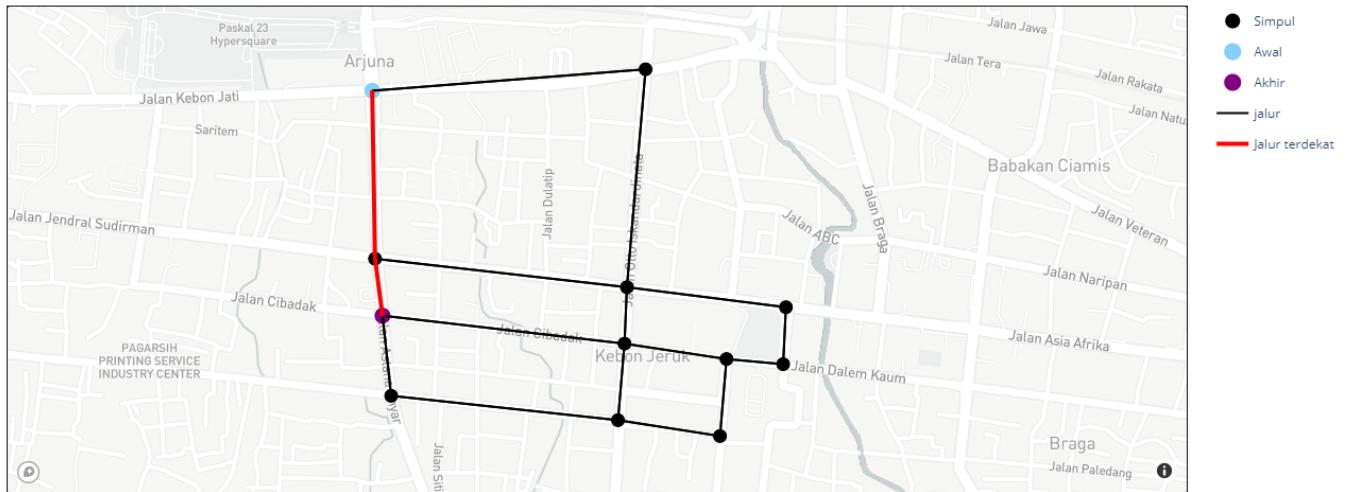
Nama file : harapanindah.txt

36

C. Screenshot lintasan terpendek

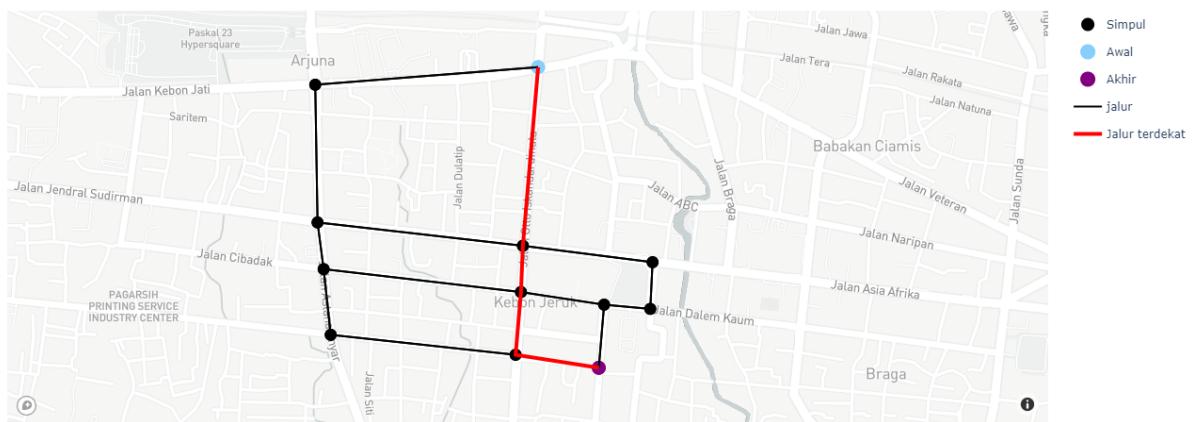
1. Non-Bonus:

Peta Jalur Terdekat (Kebon Jati - Gardujati -> Cibadak - Astana Anyar: 0.57213 km)



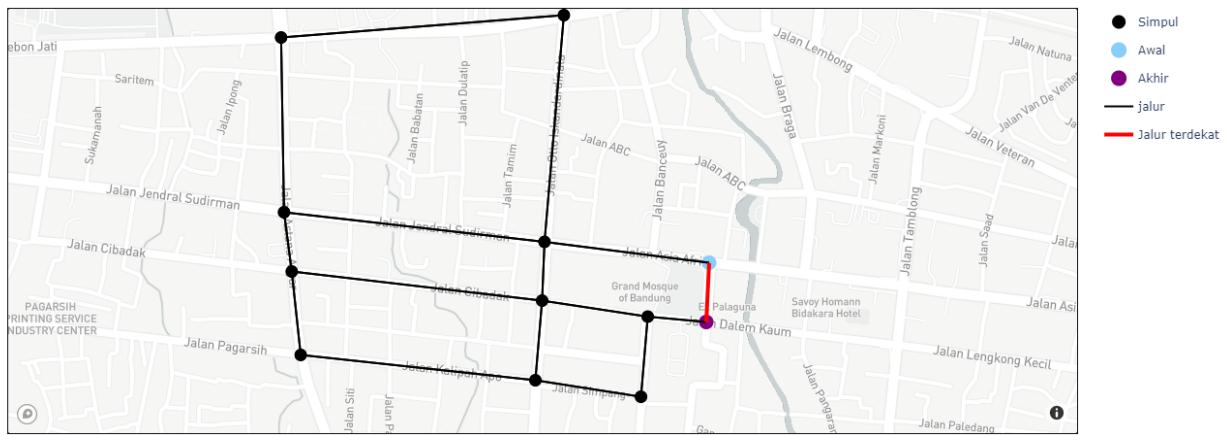
Gambar 1 : Output non-bonus dengan file input “alun2bandung.txt” dari “Kebon Jati - Gardujati” ke “Cibadak - Astana Anyar”

Peta Jalur Terdekat (Kebon Jati - Otista -> Dewi Sartika - Simpang: 1.15481 km)



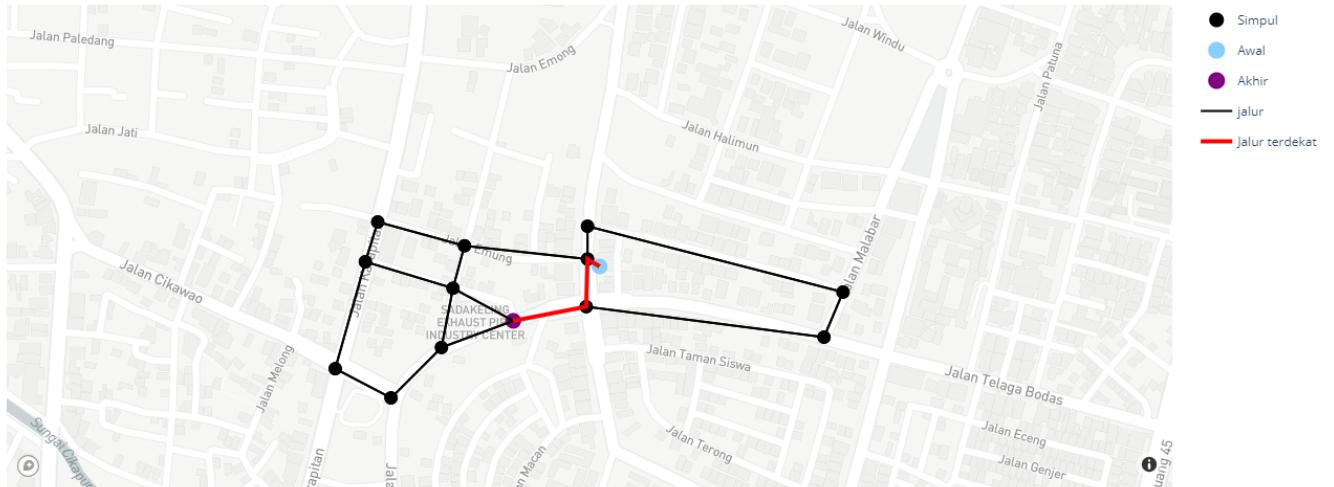
Gambar 2 : Output non-bonus dengan file input “alun2bandung.txt” dari “Kebon Jati - Otista” ke “Dewi Sartika - Simpang”

Peta Jalur Terdekat (Asia Afrika - Alun-alun -> Alun-alun - Dalem Kaum: 0.14466 km)



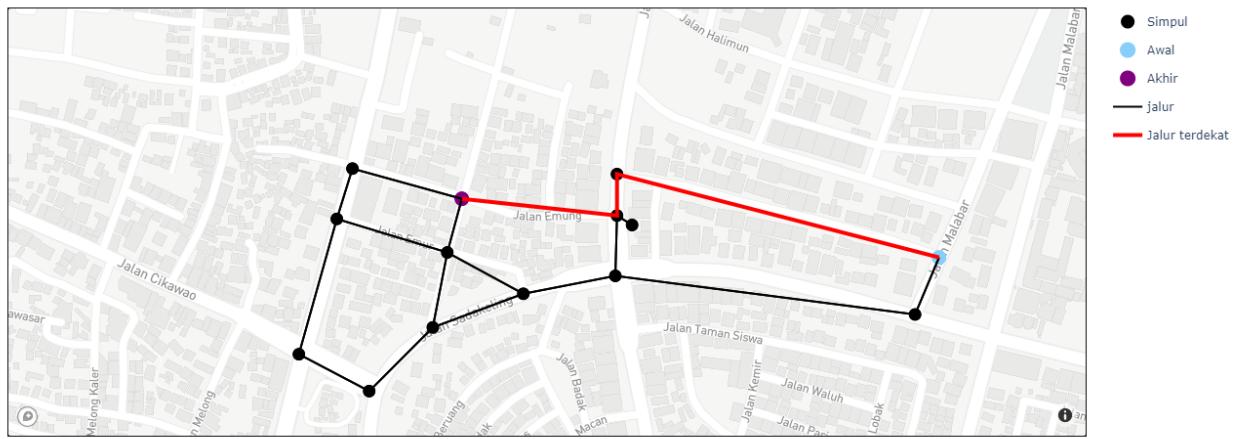
Gambar 3 : Output non-bonus dengan file input “alun2bandung.txt” dari “Asia Afrika - Alun-alun” ke “Alun-alun - Dalem Kaum”

Peta Jalur Terdekat (Batagor Riri -> Wiradisastra - Sadakeling: 0.18214 km)



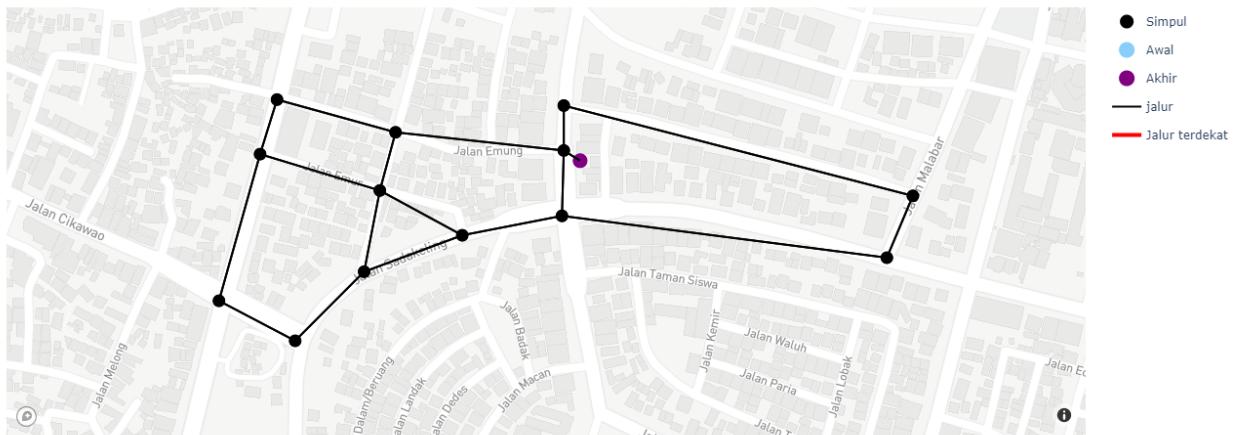
Gambar 4 : Output non-bonus dengan file input “buahbatu.txt” dari “Batagor Riri” ke”Wiradisastra - Sadakeling”

Peta Jalur Terdekat (Haruman - Malabar -> Emong - Emur: 0.56252 km)



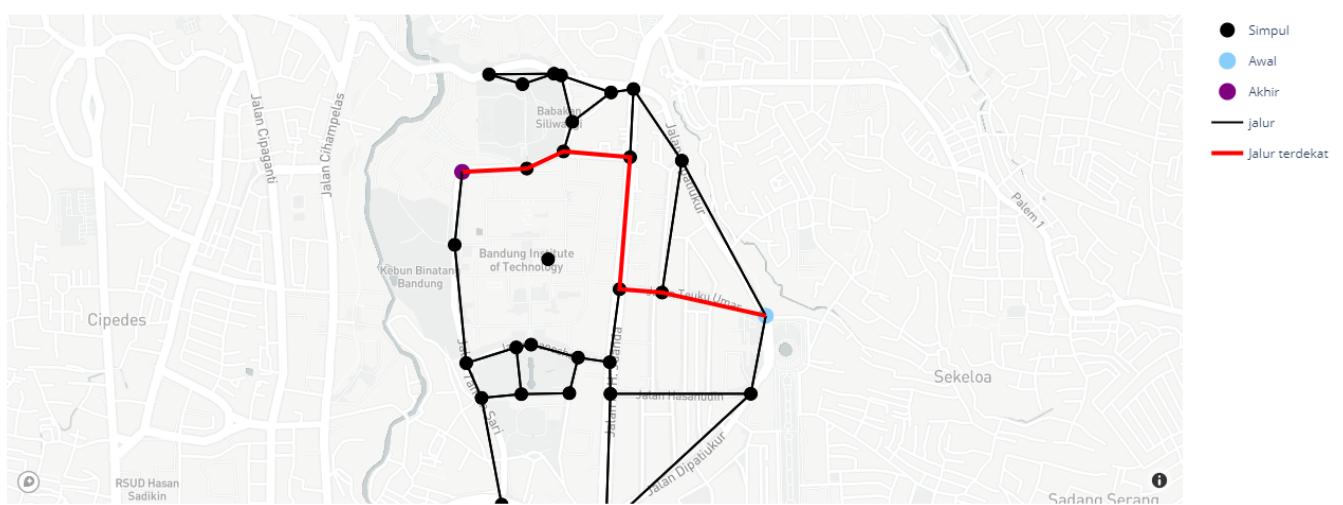
Gambar 5 : Output non-bonus dengan file input “buahbatu.txt” dari “Haruman - Malabar” ke”Emong - Emur”

Peta Jalur Terdekat (Batagor Riri -> Batagor Riri: 0 km)



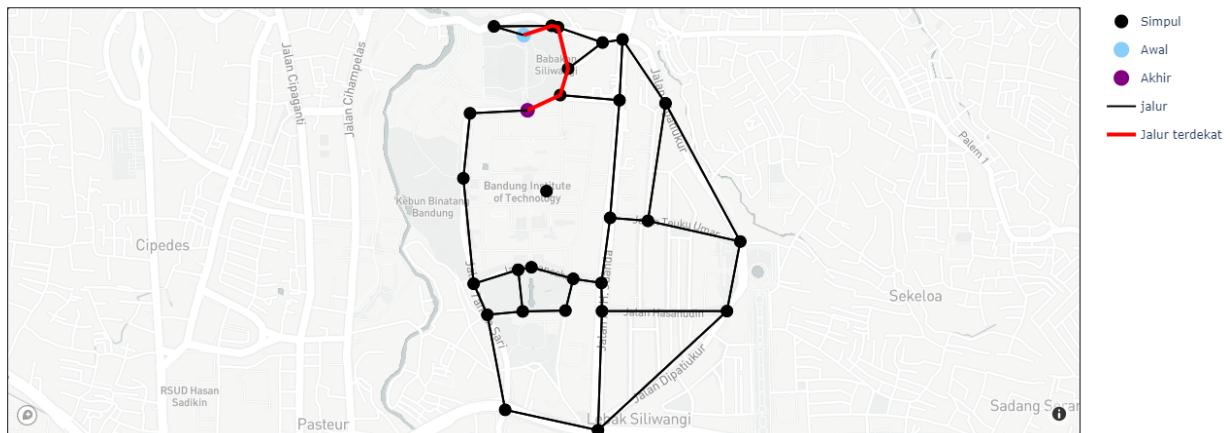
Gambar 6 : Output non-bonus dengan file input “buahbatu.txt” dari “Batagor Riri” ke”Batagor Riri” (Kasus node awal dan tujuan sama)

Peta Jalur Terdekat (DU - Teuku Umar -> Tamfes: 1.57832 km)



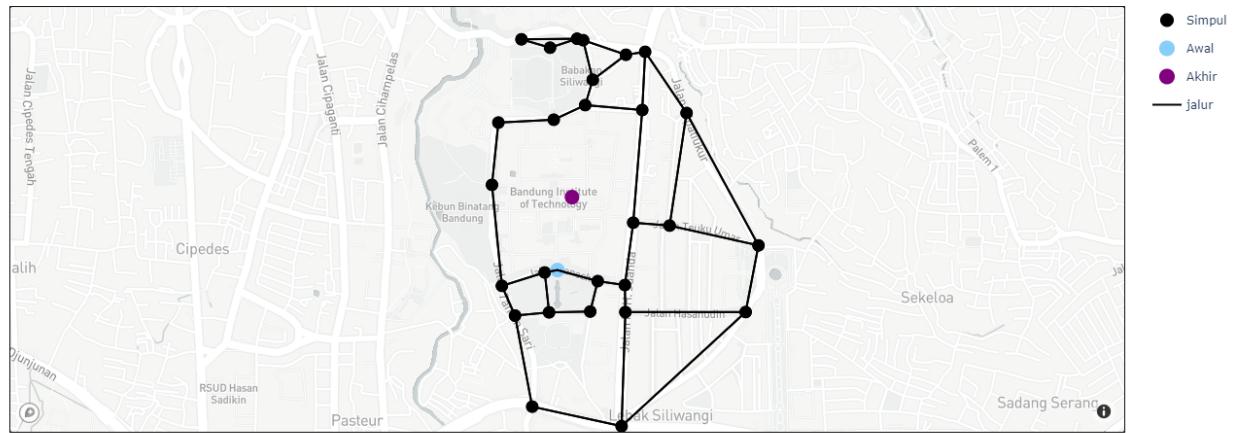
Gambar 7 : Output non-bonus dengan file input “dago.txt” dari “DU - Teuku Umar” ke “Tamfes”

Peta Jalur Terdekat (Skywalk -> Pintu Belakang ITB: 0.55551 km)



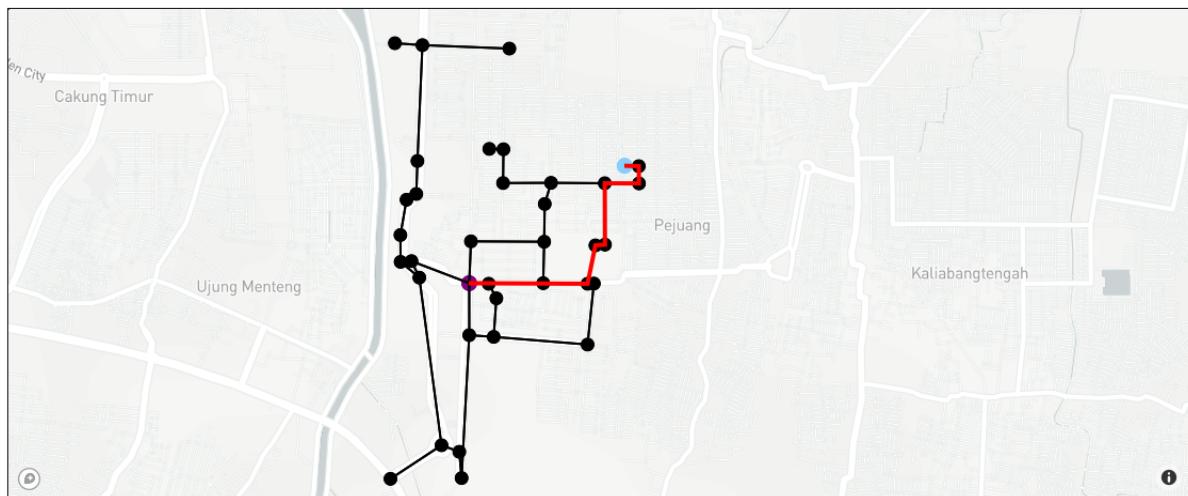
Gambar 8 : Output non-bonus dengan file input “dago.txt” dari “Skywalk” ke “Pintu Belakang ITB”

Tidak ada jalur antara kedua simpul (Pintu Depan ITB -> Gedung STEI)



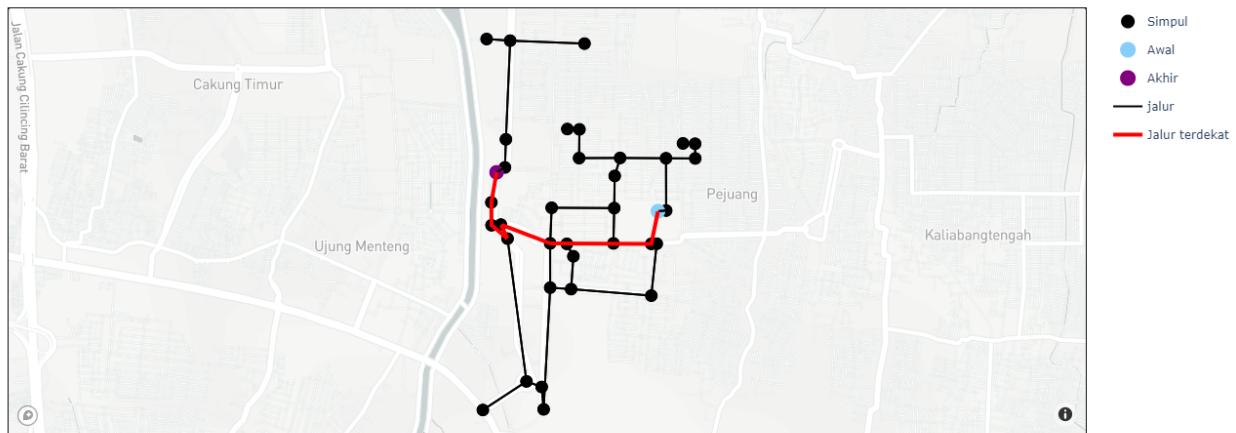
Gambar 9 : Output non-bonus dengan file input “dago.txt” dari “Pintu Depan ITB” ke “Gedung STEI” (Kasus tidak ada jalur)

Peta Jalur Terdekat (Rumah Girvin -> Bunderan SC: 1.8223 km)



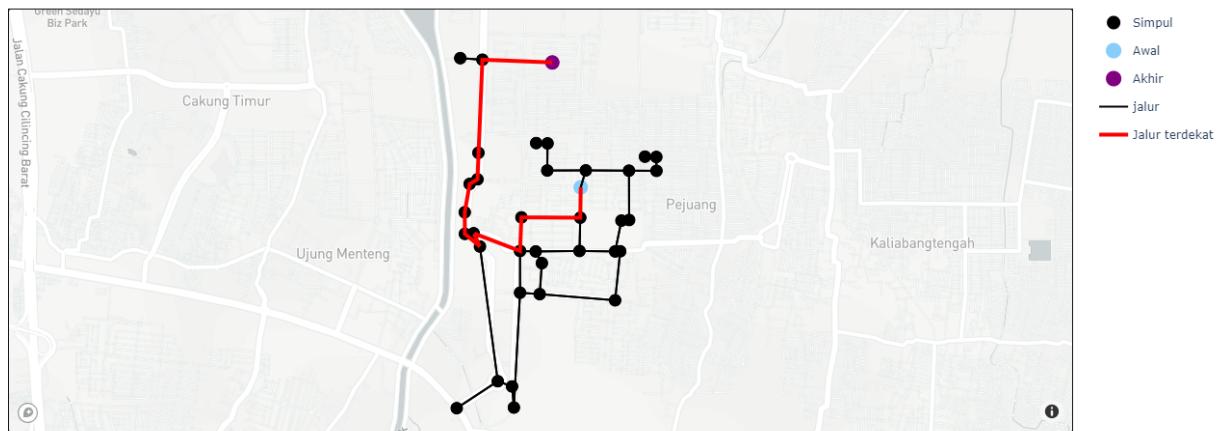
Gambar 10 : Output non-bonus dengan file input “harapanindah.txt” dari “Rumah Girvin” ke “Bunderan SC”

Peta Jalur Terdekat (Pasar Family -> BPK Penabur Harapan Indah: 2.0118 km)



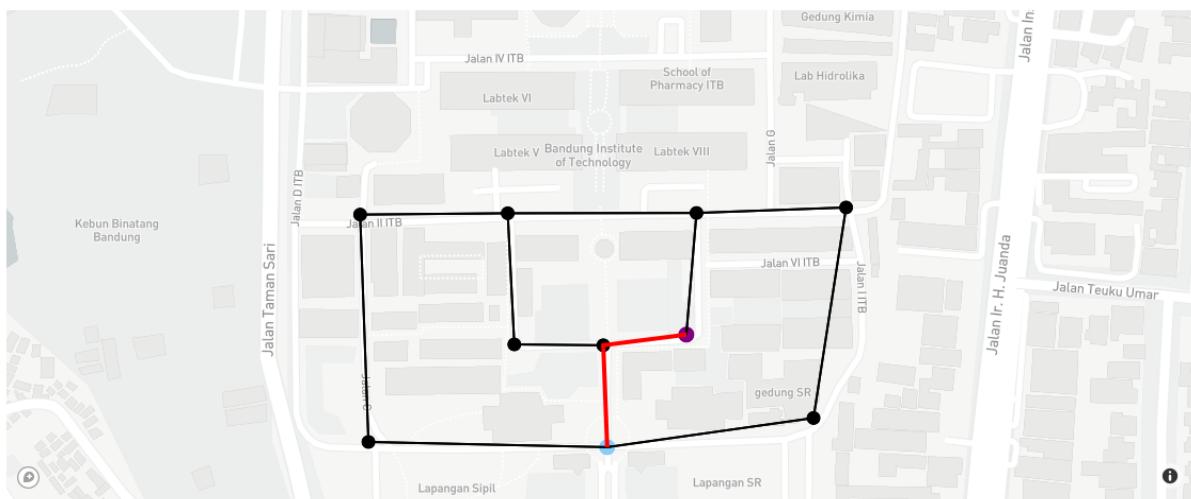
Gambar 11 : Output non-bonus dengan file input “harapanindah.txt” dari “Pasar Family” ke “BPK Penabur Harapan Indah”

Peta Jalur Terdekat (Bunderan Stainless -> Gerbang Depan Taman Sari: 3.51357 km)



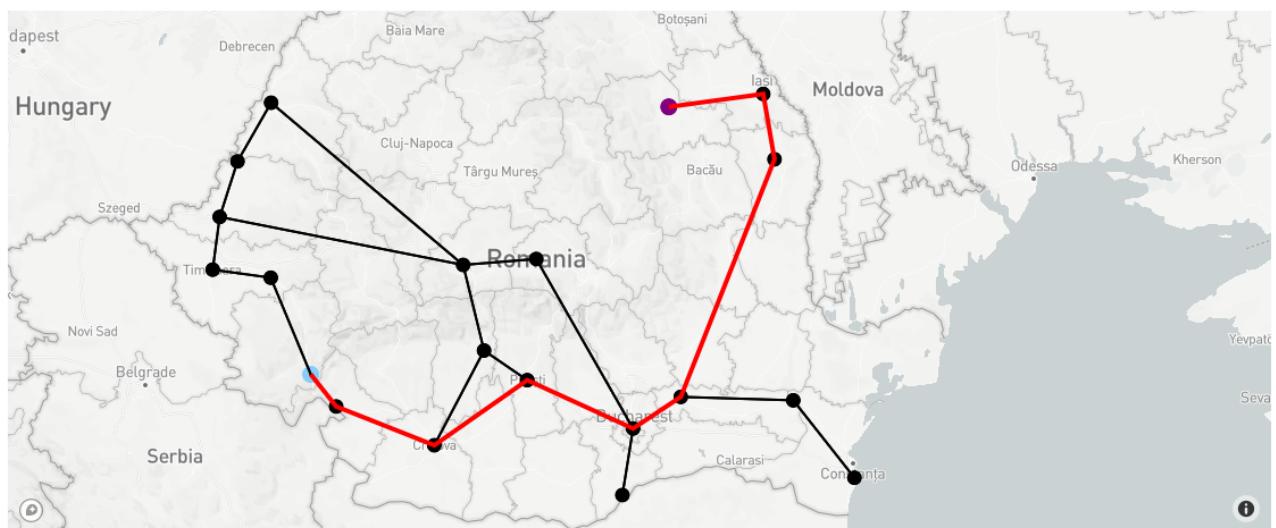
Gambar 12 : Output non-bonus dengan file input “harapanindah.txt” dari “Bunderan Stainless” ke “Gerbang Depan Taman Sari”

Peta Jalur Terdekat (Pintu Utama ITB > Simpang Lapangan Cinta: 0.14467 km)



Gambar 13 : Output non-bonus dengan file input “dalamitb.txt” dari “Pintu Utama ITB” ke “Simpang Lapangan Cinta”

Peta Jalur Terdekat (Mehadia -> Neamt: 768.8074 km)



Gambar 14 : Output non-bonus dengan file input “bucharest.txt” dari “Mehadia” ke “Neamt”

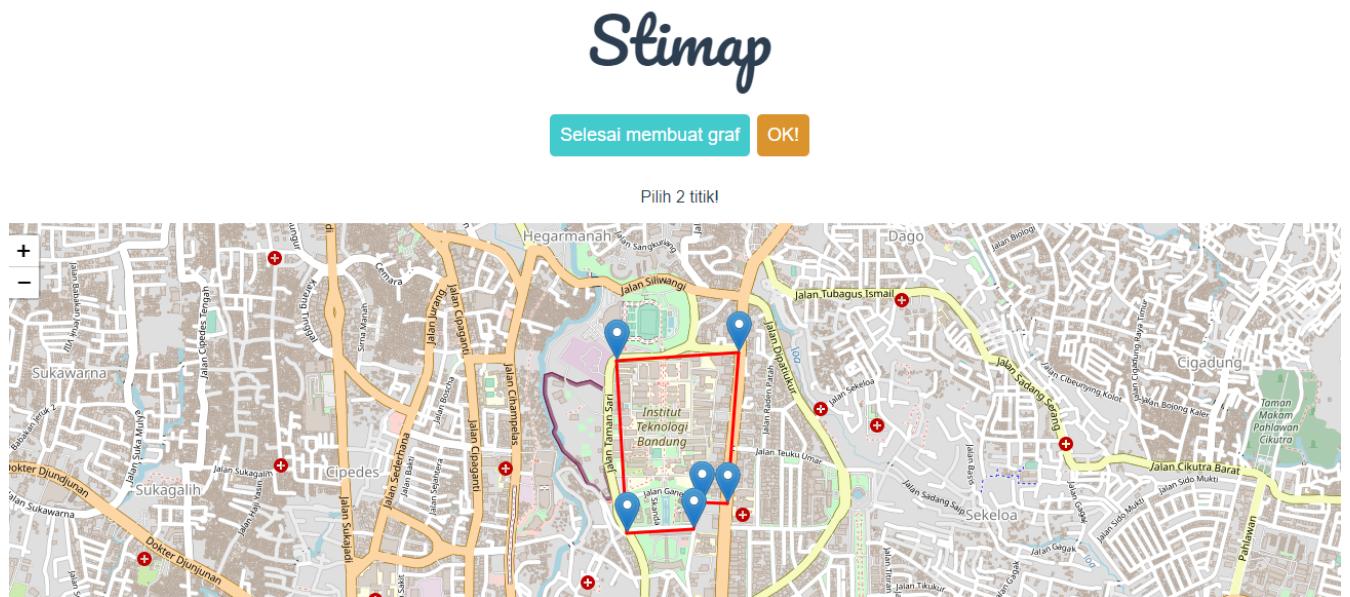
2. Bonus :

Keterangan :

- Polylines merah adalah graf yang terbentuk
- Polylines biru adalah hasil pathfinding
- Marker map biru adalah node dari graf
- Marker map merah adalah node awal dan tujuan pathfinding



Gambar 15 : Output bonus sebelum membuat graf



Gambar 16 : Output bonus setelah membuat graf

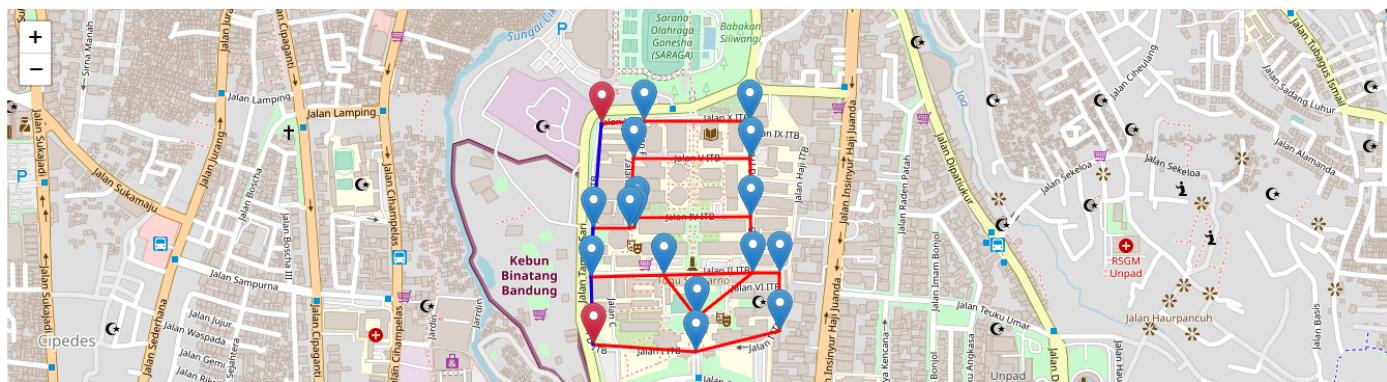
Stimap

Selesai membuat graf

OK!

Pilih 2 titik!

Jarak : 0.5048896637178801 km



Gambar 17 : Output bonus setelah user membuat graf, memilih titik awal dan tujuan, dan mengklik tombol “OK!”

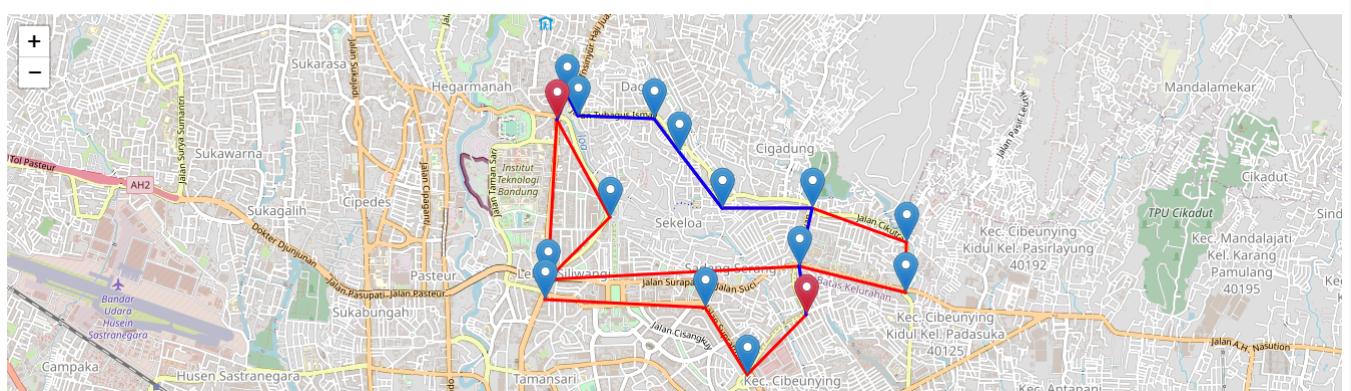
Stimap

Selesai membuat graf

OK!

Pilih 2 titik!

Jarak : 4.136406414467957 km



Gambar 18 : Output bonus setelah user membuat graf, memilih titik awal dan tujuan, dan mengklik tombol “OK!”

Stimap

Selesai membuat graf

OK!

Pilih 2 titik!

Kedua titik tidak berhubungan



Gambar 18 : Output bonus setelah user membuat graf, memilih titik awal dan tujuan, mengklik tombol “OK!, dan tidak ada path di antara 2 titik”

D. Lampiran

1. Alamat source code

[Repository Github](#)

2. Tabel:

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	✓