

LAPORAN TUGAS KECIL III
IF2211 STRATEGI ALGORITMA
Semester II Tahun 2020/2021



Disusun oleh:

13519021	Arjuna Marcelino
13519092	Sharon Bernadetha Marbun

Himpunan Mahasiswa Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

Daftar Isi

I. Implementasi Program	2
II. Peta Input	13
III. Hasil Eksekusi Program	18
IV. Komponen Penilaian	22

I. Implementasi Program

Program ini diimplementasikan dalam bahasa pemrograman Python dan disimpan dalam file yang bernama InputFile.py, Node.py, Main.py dan gui.py. Kode program untuk masing-masing file ditampilkan sebagai berikut.

1. InputFile.py

```
def inputFile(fileName, numNode, tabNama, tabKoor, tabAdj):  
    f = open("../test/" + fileName, "r")  
    lines = f.readlines()  
  
    tabAdj.clear()  
    tabNama.clear()  
    tabKoor.clear()  
  
    #menyimpan banyaknya simpul ke dalam numNode  
    numNodeNew = int(lines[0])  
  
    #menyimpan data nama simpul ke dalam tabNama  
    tabNamaNew = ['*'] * numNodeNew  
    for i in range(1,numNodeNew+1):  
        tabNamaNew[i-1] = lines[i]  
        tabNamaNew[i-1] = tabNamaNew[i-1].replace('\n','')  
  
    #menyimpan data koordinat ke dalam tabKoor  
    tabKoorNew = [[0.0 for j in range(2)] for i in range(numNodeNew)]  
    for i in range(numNodeNew+1,numNodeNew*2+1):  
        words = lines[i].split()  
        tabKoorNew[i-numNodeNew-1][0] = float(words[0])  
        tabKoorNew[i-numNodeNew-1][1] = float(words[1])  
  
    #menyimpan data matriks ketetanggaan ke dalam tabAdj
```

```
        tabAdjNew = [[False for j in range(numNodeNew)] for i in
range(numNodeNew)]

    for i in range(numNodeNew*2+1, numNodeNew*3+1):
        numbers = lines[i].split()

        for j in range(numNodeNew):
            if numbers[j] == "1":
                tabAdjNew[i-numNodeNew*2-1][j] = True

    tabNama += tabNamaNew
    tabKoor += tabKoorNew
    tabAdj += tabAdjNew
```

2. Node.py

```
from copy import deepcopy

#kelas yang merepresentasikan simpul dari graf
class Node:

    def __init__(self,idx,tabHn,tabNama):
        self.idx = idx                    #indeks dari simpul
        self.name = findName(idx,tabNama) #nama simpul
        self.track = "null"               #nama-nama simpul yang
dikunjungi sebelum simpul ini dikunjungi
        self.f = 0                        #nilai f(n)
        self.g = 0                        #nilai g(n)
        self.h = tabHn[int(self.idx)]

    #mengembalikan hasil copy dari node
    def copy(self):
        return deepcopy(self)

    #menampilkan info dari node
    def print(self):
        print("Nama      : " + self.name)
```

```
print("Indeks      : " + str(self.idx))
print("Jejak       : " + self.track)
print("F(n)        : " + str(self.f))
print("G(n)        : " + str(self.g))
print("H(n)        : " + str(self.h) + '\n')

#mendapatkan indeks parent dari node
def getParent(self):
    idxs = self.track.split()
    return idxs[len(idxs) - 1]

#mendapatkan string berupa urutan nama-nama dari simpul yang telah
dilewati hingga mencapai simpul yang sekarang
def getRekamJejak(self, tabNama):
    if self.track == "null":
        return "Tidak ada rekam jejak"
    ancestors = self.track.split()
    jejak = ""
    i = 0
    while i < len(ancestors):
        jejak += findName(int(ancestors[i]), tabNama) + " -> "
        i+=1
    jejak += self.name
    return jejak

#mendapatkan list indeks dari urutan simpul yang telah dilewati hingga
mencapai simpul yang sekarang
def getListJejak(self):
    if self.track == "null":
        return []
    ancestors = self.track.split()
    listJejak = []
    for x in ancestors:
        listJejak.append(int(x))
```

```
listJejak.append(self.idx)

return listJejak

#mendapatkan list indeks tetangga dari sebuah simpul node
def getListAdjIdx(self, tabAdj):
    listAdj=[]
    numNode = len(tabAdj[0])
    for i in range(numNode):
        if (tabAdj[int(self.idx)][i]):
            if not self.isVisited(i):
                listAdj.append(i)
    return listAdj

#mendapatkan list simpul tetangga dari sebuah simpul node
def getListAdjNode(self, tabAdj, tabNama, tabKoor):
    listAdj = self.getListAdjIdx(tabAdj)
    listNode = self.getListNextNode(listAdj, tabNama, tabKoor)
    return listNode

#menambahkan indeks node ke dalam track dari simpul dan mengganti nama
simpul dengan simpul selanjutnya
def nextNode(self, nextIdx):
    temp = self.copy()
    if temp.track == "null":
        temp.track = str(temp.idx)
    else: # node.track != "null"
        temp.track = temp.track + " " + str(temp.idx)
    temp.idx = nextIdx
    temp.name = findName(nextIdx)
    temp.g += getEuclideanDistance(temp.getParent(), nextIdx)
    temp.f = temp.g + temp.h
    return temp
```

```
#mengembalikan list berisi Node yang merupakan hasil next dari simpul
node terhadap indeks pada listIdx
```

```
def getListNextNode(self, listIdx, tabNama, tabKoor):
    listNode = []
    for x in listIdx:
        temp = deepcopy(self)
        if temp.track == "null":
            temp.track = str(temp.idx)
        else: #node.track != "null"
            temp.track = temp.track + " " + str(temp.idx)
        temp.idx = x
        temp.name = findName(x, tabNama)
        temp.g += getEuclideanDistance(temp.getParent(), x, tabKoor)
        temp.f = temp.g + temp.h
        listNode.append(temp)
    return listNode
```

```
def cleanListNode(self, listNode):
    for node in listNode:
        if node.f >= self.f:
            listNode.remove(node)
```

```
#mengecek apakah simpul pada checkIdx sudah dikunjungi oleh simpul node
atau belum
```

```
def isVisited(self, checkIdx):
    if self.track == "null":
        return False
    else:
        idxs = self.track.split()
        for i in range(len(idxs)):
            if idxs[i] == str(checkIdx):
                return True
        return False
```

```
#memasukkan simpul apabila listNode kosong atau apabila simpul memiliki
nilai f yang lebih kecil daripada elemen listNode

def appendCheck(self,listNode):
    if len(listNode)== 0:
        listNode.append(self)
    else:
        if self.f < listNode[0].f:
            listNode.clear()
            listNode.append(self)
        #else: do nothing

#mendapatkan nama dari simpul sesuai dengan indeksinya
def findName(idx,tabNama):
    return tabNama[int(idx)]

#mendapatkan indeks dari simpul sesuai dengan namanya
def findIdx(name,tabNama):
    for i in range(len(tabNama)):
        if tabNama[i]==name:
            return i

#mendapatkan jarak euclidean antara simpul start dan simpul goal
def getEuclideanDistance(startIdx,goalIdx,tabKoor):
    diffX = abs(tabKoor[int(goalIdx)][0] - tabKoor[int(startIdx)][0])
    diffY = abs(tabKoor[int(goalIdx)][1] - tabKoor[int(startIdx)][1])
    result = (diffX**2 + diffY**2)**0.5
    #print("jarak dari " + str(startIdx) + " ke " + str(goalIdx) + " = " +
    str(result))
    return(result)

#mengisi tabHn dengan nilai heuristik yang dihitung berdasarkan jarak simpul
ke simpul goal
def generateHn(goalIdx, tabHn, tabKoor):
    for nodeIdx in range(len(tabHn)):
```



```
tabHn[nodeIdx] = getEuclideanDistance(nodeIdx, goalIdx, tabKoor)

#mengembalikan indeks dari node dengan f(n) terkecil
def getBestNodeIdx(listNode):
    idxmin = 0
    min = listNode[0].f
    for i in range(len(listNode)):
        if min > listNode[i].f:
            min = listNode[i].f
            idxmin = i
    return idxmin

#mengembalikan indeks dari simpul goal pada listNode, mengembalikan -1 bila
simpul goal tidak ditemukan
def findGoalIdx(listNode, goalIdx):
    for i in range(len(listNode)):
        if int(listNode[i].idx) == int(goalIdx):
            return i
    #apabila tidak ditemukan
    return -1
```

3. AStar.py

```
import Node as n

def aStar(startIdx, goalIdx, initListHold, initListGoal, initCurrNode,
tabAdj, tabNama, tabKoor):
    listHold = initListHold.copy()          #menyimpan list simpul yang masih
berpotensi untuk menghasilkan solusi terbaik

    listGoal = initListGoal.copy()          #menyimpan simpul goal terbaik
apabila ditemukan (list dengan maks 1 elemen)

    currNode = initCurrNode.copy()          #menyimpan simpul yang saat ini
sedang diperiksa

    #mengembalikan simpul start apabila startIdx == goalIdx
    if startIdx == goalIdx:
```

```
print("Simpul awal sama dengan simpul akhir.")

return currNode

#melanjutkan pemeriksaan terhadap simpul-simpul tetangga dari currNode
listAdjNode = currNode.getListAdjNode(tabAdj,tabNama,tabKoor)

if len(listAdjNode) == 0:

    #apabila tidak ditemukan simpul tetangga, pemeriksaan dilakukan
    terhadap simpul yang di-hold

    if len(listHold) > 0:

        currNode = listHold[0].copy()

        listHold.remove(listHold[0])

        return aStar(startIdx, goalIdx, listHold, listGoal,
currNode,tabAdj,tabNama,tabKoor)

    #apabila tidak ditemukan simpul tetangga dan simpul yang di-hold,
    mengembalikan elemen pertama listGoal

    if len(listGoal) > 0:

        return listGoal[0]

    #apabila tidak ditemukan simpul tetangga, simpul yang di-hold, dan
    simpul goal, maka solusi tidak ditemukan

    return -1          #mengembalikan kode error -1

#mencari apakah ditemukan simpul goal pada listAdjNode

foundIdx = n.findGoalIdx(listAdjNode,goalIdx)

if foundIdx == -1:

    #apabila tidak ditemukan, dilakukan pemeriksaan terhadap simpul
    dengan f(n) terkecil

    idxBest = n.getBestNodeIdx(listAdjNode)

    currNode = listAdjNode[idxBest].copy()

    del listAdjNode[idxBest]

    listHold += listAdjNode

return
aStar(startIdx,goalIdx,listHold,listGoal,currNode,tabAdj,tabNama,tabKoor)
```

```
    else:
        #simpul goal ditemukan
        goalNode = listAdjNode[foundIdx].copy()
        listAdjNode.remove(listAdjNode[foundIdx])

        #memasukkan simpul tetangga lain ke dalam listHold
        listHold += listAdjNode

        #menghapus simpul-simpul yang di-hold dan memiliki nilai f(n) >=
        nilai f(n) simpul goal
        goalNode.cleanListNode(listHold)

        #memperbaharui listGoal apabila goalNode lebih efektif
        goalNode.appendCheck(listGoal)

        if len(listHold) == 0:
            #apabila tidak ada lagi simpul yang di-hold, mengembalikan
            goalNode
            return listGoal[0]

        else:
            #apabila masih ada, dilakukan pemeriksaan terhadap elemen pertama
            listHold

            currNode = listHold[0].copy()
            listHold.remove(listHold[0])

            return
            aStar(startIdx, goalIdx, listHold, listGoal, currNode, tabAdj, tabNama, tabKoor)
```

4. Main.py

```
import Node as n

def aStar(startIdx, goalIdx, initListHold, initListGoal, initCurrNode,
          tabAdj, tabNama, tabKoor):
```

```
listHold = initListHold.copy()          #menyimpan list simpul yang masih
berpotensi untuk menghasilkan solusi terbaik

listGoal = initListGoal.copy()          #menyimpan simpul goal terbaik
apabila ditemukan (list dengan maks 1 elemen)

currNode = initCurrNode.copy()          #menyimpan simpul yang saat ini
sedang diperiksa

#mengembalikan simpul start apabila startIdx == goalIdx
if startIdx == goalIdx:
    print("Simpul awal sama dengan simpul akhir.")
    return currNode

#melanjutkan pemeriksaan terhadap simpul-simpul tetangga dari currNode
listAdjNode = currNode.getListAdjNode(tabAdj,tabNama,tabKoor)
if len(listAdjNode) == 0:

    #apabila tidak ditemukan simpul tetangga, pemeriksaan dilakukan
    terhadap simpul yang di-hold
    if len(listHold) > 0:
        currNode = listHold[0].copy()
        listHold.remove(listHold[0])

        return aStar(startIdx, goalIdx, listHold, listGoal,
currNode,tabAdj,tabNama,tabKoor)

    #apabila tidak ditemukan simpul tetangga dan simpul yang di-hold,
    mengembalikan elemen pertama listGoal
    if len(listGoal) > 0:
        return listGoal[0]

    #apabila tidak ditemukan simpul tetangga, simpul yang di-hold, dan
    simpul goal, maka solusi tidak ditemukan
    return -1          #mengembalikan kode error -1

#mencari apakah ditemukan simpul goal pada listAdjNode
foundIdx = n.findGoalIdx(listAdjNode,goalIdx)
if foundIdx == -1:
```

```
        #apabila tidak ditemukan, dilakukan pemeriksaan terhadap simpul
        dengan f(n) terkecil

        idxBest = n.getBestNodeIdx(listAdjNode)

        currNode = listAdjNode[idxBest].copy()

        del listAdjNode[idxBest]

        listHold += listAdjNode

                                                                 return
aStar(startIdx,goalIdx,listHold,listGoal,currNode,tabAdj,tabNama,tabKoor)

    else:

        #simpul goal ditemukan

        goalNode = listAdjNode[foundIdx].copy()

        listAdjNode.remove(listAdjNode[foundIdx])

        #memasukkan simpul tetangga lain ke dalam listHold

        listHold += listAdjNode

        #menghapus simpul-simpul yang di-hold dan memiliki nilai f(n) >=
        nilai f(n) simpul goal

        goalNode.cleanListNode(listHold)

        #memperbaharui listGoal apabila goalNode lebih efektif

        goalNode.appendCheck(listGoal)

        if len(listHold) == 0:

            #apabila tidak ada lagi simpul yang di-hold, mengembalikan
            goalNode

            return listGoal[0]

        else:

            #apabila masih ada, dilakukan pemeriksaan terhadap elemen pertama
            listHold

            currNode = listHold[0].copy()

            listHold.remove(listHold[0])

                                                                 return
aStar(startIdx,goalIdx,listHold,listGoal,currNode,tabAdj,tabNama,tabKoor)
```

II. Peta Input

Terdapat enam file test yang digunakan dalam program ini, yang ditampilkan sebagai berikut.

1. File uji test.txt: berdasarkan peta jalan sekitar kampus ITB

test1 - Notepad

File Edit Format View Help

10

ITB

Masjid Salman

Simpang Tamansari x Ganesha

Kebun Binatang

Simpang Ganesha x Juanda

RS Borromeus

SPBU Pertamina Dago

Simpang Juanda x Siliwangi

Simpang Siliwangi x Sumur

Warunk Upnormal

-6.893089484901214 107.61043794566532

-6.893506877790332 107.6112998313408

-6.893868617985648 107.6084479006883

-6.890268275293758 107.60807889305418

-6.893749393160202 107.61288043412759

-6.893742874298343 107.61303802613446

-6.8901705244788625 107.61327441414511

-6.885217441587917 107.61366972135883

-6.885176697969291 107.61320843642146

-6.885447235548346 107.61304591966437

0 1 1 0 1 1 0 0 0 0

1 0 1 0 1 1 0 0 0 0

1 1 0 1 1 1 0 0 0 0

0 0 1 0 0 0 0 0 0 0

1 1 1 0 0 1 1 1 0 0

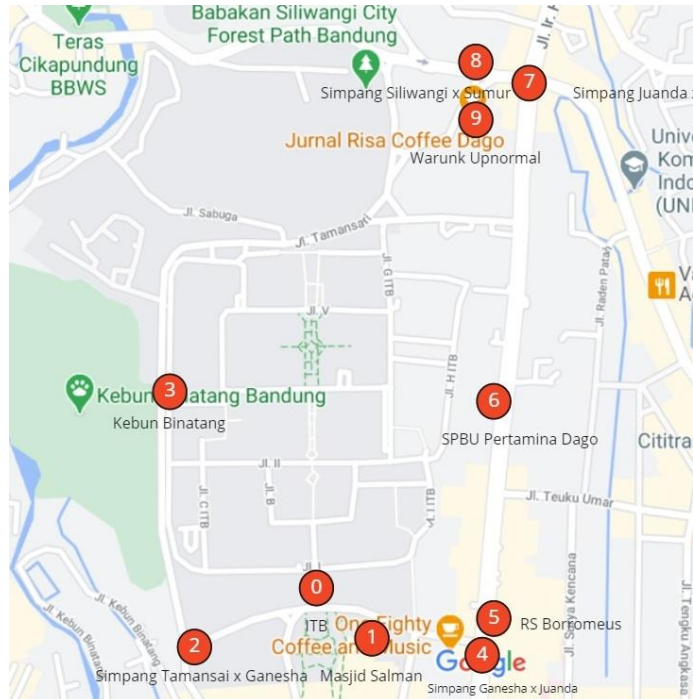
1 1 1 0 1 0 1 1 0 0

0 0 0 0 1 1 0 1 0 0

0 0 0 0 1 1 1 0 1 0

0 0 0 0 0 0 0 1 0 1

0 0 0 0 0 0 0 0 1 0



2. File uji test2.txt: berdasarkan peta jalan sekitar Alun-Alun Bandung

test2 - Notepad

File Edit Format View Help

11

IDNGRAFIS

Jl. Asia Afrika 59

MBDC Bandung

OBOSS KREASINDO

toko computer newpelangi

Jl. Dalem Kaum 56

Jl. Belakang Factory 2

Nasi uduk ayam goreng bakar maranti

Baso Tahu Mang Jangkung

Wajan Neng Incessss

Pojok Hepi Shopee Bandung

-6.9211766220697 107.60758414404644

-6.921246771042323 107.60775380663335

-6.9225655570645515 107.60764487478365

-6.922533566834937 107.60751597418738

-6.921010754132276 107.60647057918314

-6.922395132079498 107.60638015136941

-6.919973943339997 107.60661502459642

-6.920021624321239 107.60700226158842

-6.920069669442445 107.60740715798342

-6.9202266918317275 107.60816942756514

-6.921235129976102 107.60803618500614

0 1 0 1 1 0 0 0 0 0 1

1 0 1 0 0 0 0 0 0 0 1

0 1 0 1 0 0 0 0 0 0 0

1 0 1 0 0 1 0 0 0 0 0

1 0 0 0 0 0 1 0 0 0 0

0 0 0 1 0 0 0 0 0 0 0

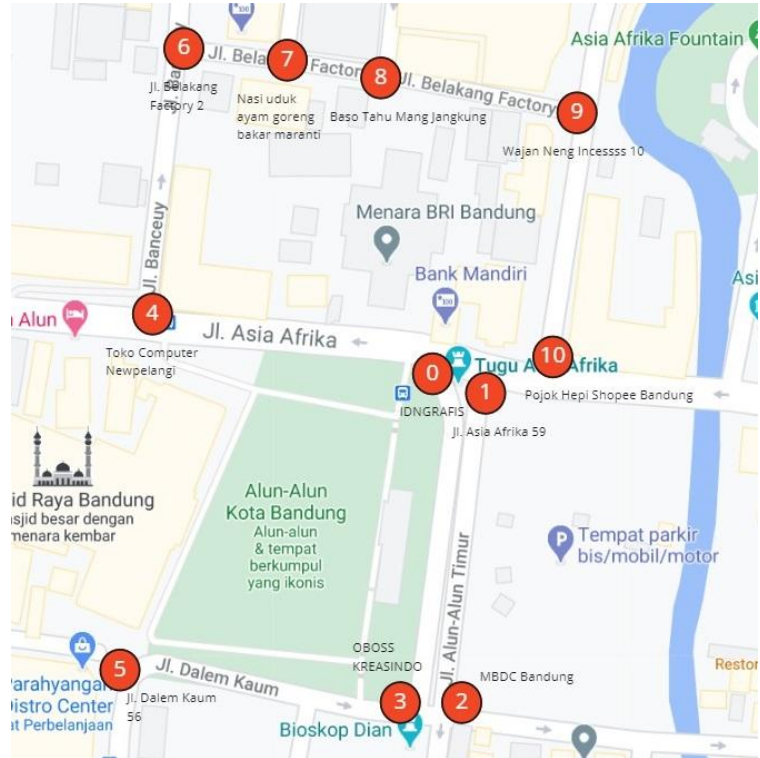
0 0 0 0 1 0 0 1 0 0 0

0 0 0 0 0 0 1 0 1 0 0

0 0 0 0 0 0 0 1 0 1 0

0 0 0 0 0 0 0 0 1 0 1

1 1 0 0 0 0 0 0 0 1 0



3. File uji test3.txt: berdasarkan peta jalan sekitar Buahbatu

test3 - Notepad

File Edit Format View Help

12

Jl. Neptunus Barat 17-21

Jl. Neptunus Barat No. 21

EZ Street Food

Jl. Neptunus Timur I

Jl. Neptunus Tengah V

Jl. Neptunus Barat II 6

Jl. Neptunus Barat II

Zian Makeup Artist Bandung

Jl. Neptunus Barat IV

Jl. Neptunus Raya Barat

ohid social bar

Jl. Neptunus Tengah IV

-6.947228457829187 107.65604878163839

-6.947816198194652 107.65594366118093

-6.947845468356845 107.65626831550973

-6.947839363998204 107.65666188455923

-6.947216280955274 107.65666471861364

-6.9480852339858625 107.65586212201187

-6.948023046520824 107.65561153188033

-6.947638947289663 107.65562995762528

-6.947510914142878 107.65535725659977

-6.947215373178532 107.65540251103648

-6.947248676455137 107.65570961889735

-6.947494865346758 107.65667076768666

0 1 0 0 0 0 0 0 0 0 0 0

1 0 1 0 0 1 0 0 0 0 0 0

0 1 0 1 1 0 0 0 0 0 0 0

0 0 1 0 0 0 0 0 0 0 0 1

0 0 1 0 0 0 0 0 0 0 0 1

0 1 0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 1 0 1 0 0 0 0

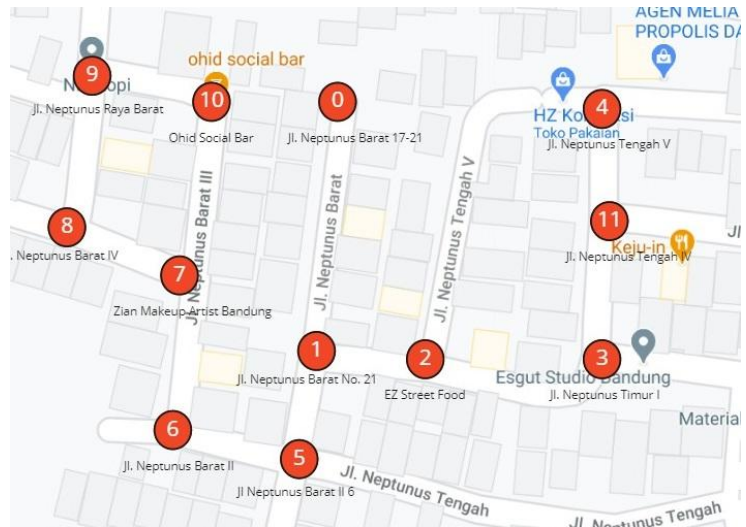
0 0 0 0 0 0 1 0 1 0 1 0

0 0 0 0 0 0 0 1 0 1 0 0

0 0 0 0 0 0 0 0 1 0 1 0

0 0 0 0 0 0 0 1 0 1 0 0

0 0 0 1 1 0 0 0 0 0 0 0



4. File uji test4.txt: berdasarkan peta jalan kecamatan Balige

test4 - Notepad

File Edit Format View Help

11

Gothil Khakiku

Clinton Sepatu & Winda Tas

Lumban Dolok Haume Bange

Perempatan Jl. DR. Term. Simatupang x Jl. Raja Paindoan

CV. Aek Raja

Jl. Patuan Anggi No. 3

Marlina Toko Kain

Optik Yosua Silaen

BALIGE Motor

Jl. Raja Paindoan 5

Willi Cell

2.3332035609386295 99.06710456334507

2.3323738872971904 99.06705644410307

2.332352772768501 99.06633791734448

2.333170215531011 99.06611287793179

2.334136250864292 99.06506385311643

2.333724593808109 99.06613639786927

2.333867889533341 99.06633070088252

2.334133825508186 99.06487514201837

2.3338315566954027 99.06711195136373

2.333118337249984 99.0653305809592

2.331757192127559 99.06699048640358

0 1 0 1 0 0 0 0 1 0 0

1 0 1 0 0 0 0 0 0 0 1

0 1 0 1 0 0 0 0 0 0 0

1 0 1 0 0 0 1 0 0 0 1 0

0 0 0 0 0 1 0 1 0 0 0

0 0 0 1 1 0 1 0 0 0 0

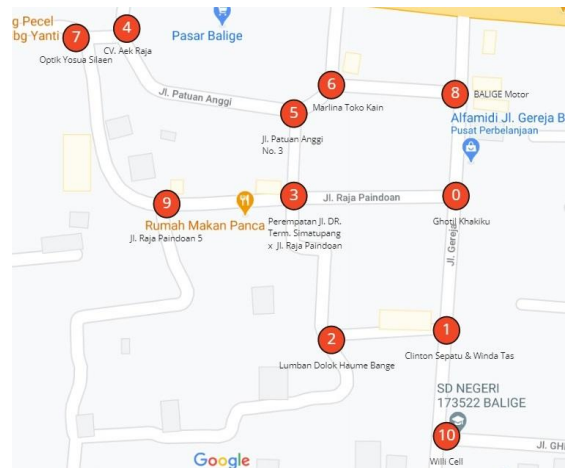
0 0 0 0 0 1 0 0 1 0 0

0 0 0 0 1 0 0 0 0 0 1 0

1 0 0 0 0 0 1 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



5. File uji test5.txt: berdasarkan peta jalan sekitaran kampus USU

test5 - Notepad

File Edit Format View Help

8

Jl. Sei Selayang 36-28

Jl. Sei Asahan 14-12

Ujung Gg. Siput

Jl. Sei Bulan 3

Laundry Coin 77

Jl. Sei Selayang 41

Jl. Sei Padang No. 75

RNTOKOMEDAN

3.5718052776700766 98.65581567099132

3.5717383526979263 98.65669811767333

3.5711256158192035 98.6561466418129

3.5711331875207675 98.6566473463256

3.570312737981298 98.65651060138232

3.5710189836158848 98.6557332490844

3.5702078519227656 98.65667738663316

3.570979981434506 98.65721651244807

0 1 0 0 0 1 0 0

1 0 0 1 0 0 0 0

0 0 0 1 0 0 0 0

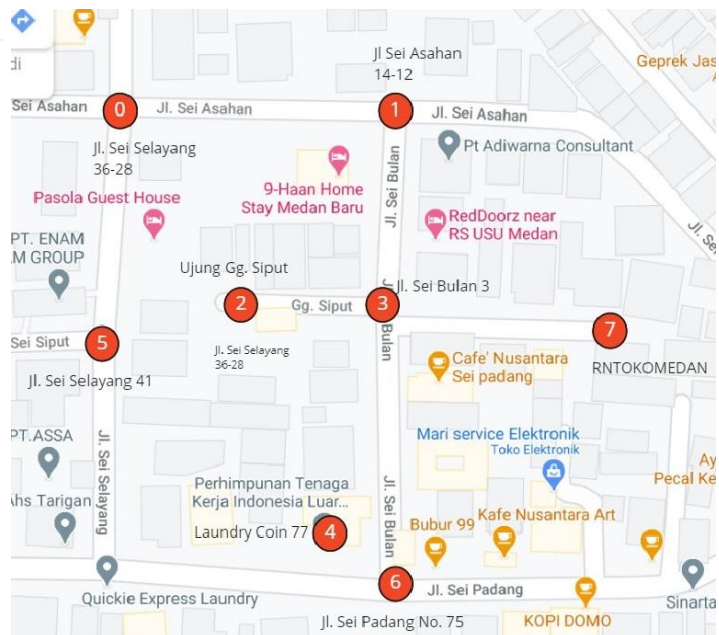
0 1 1 0 0 0 1 1

0 0 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 1 0 0 0 0



6. File uji test6.txt: berdasarkan peta jalan sekitaran kampus UI

test6 - Notepad

File Edit Format View Help

8

Jl. Juragan Sinda I 37-38

Ayam Bakar Wong Solo

Jl. Juragan Sinda Raya 2-39

Nasi goreng sedap malam bang im

Jl. Juragan Sinda Raya No. 52

Wisma Regita

Jl. Juragan Sinda II No. 46

Jl. Juragan Sinda I No. 34

-6.356983662645149 106.82074764847125

-6.3570956227443505 106.82087907670049

-6.357235588409989 106.82053196664761

-6.357585825760623 106.8202837718399

-6.35752229849499 106.82045089687988

-6.357700969426745 106.82066193793365

-6.357910322364068 106.8207082221744

-6.357330665488881 106.82124247311083

0 1 1 0 0 0 0

1 0 0 0 0 0 1 1

1 0 0 1 0 0 0 0

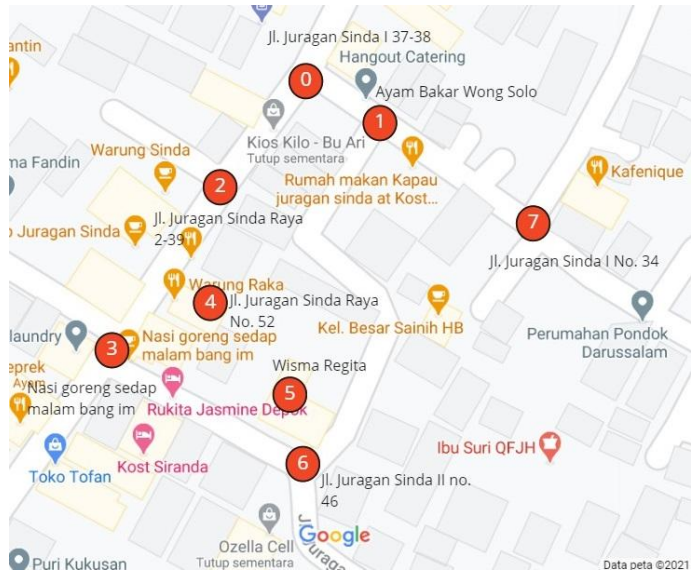
0 0 1 0 0 0 1 0

0 0 0 0 0 1 0 0

0 0 0 0 1 0 0 0

0 1 0 1 0 0 0 0

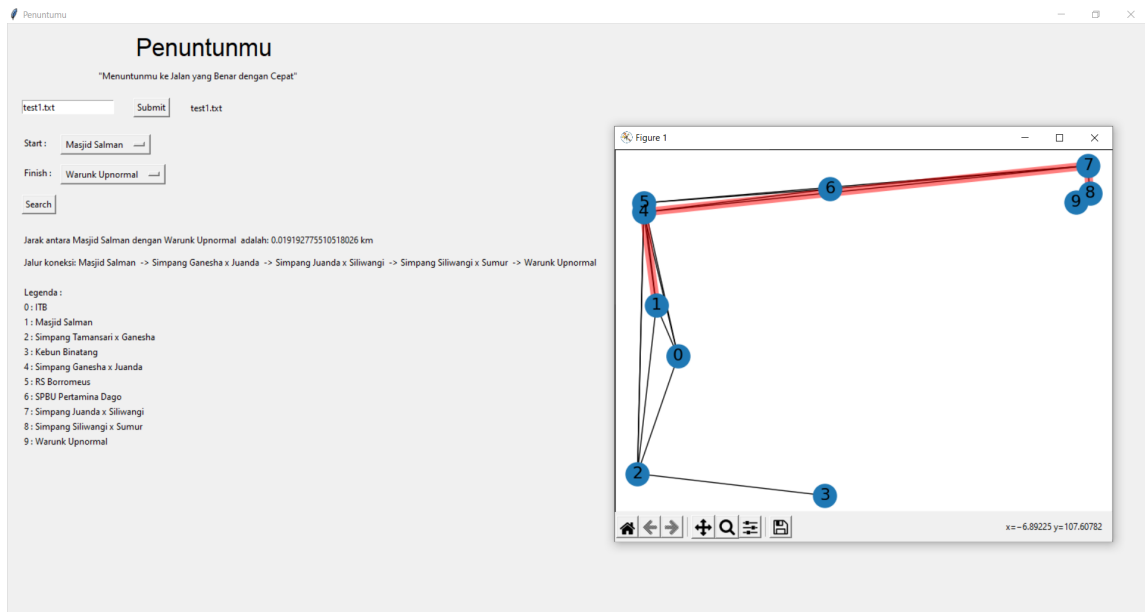
0 1 0 0 0 0 0 0



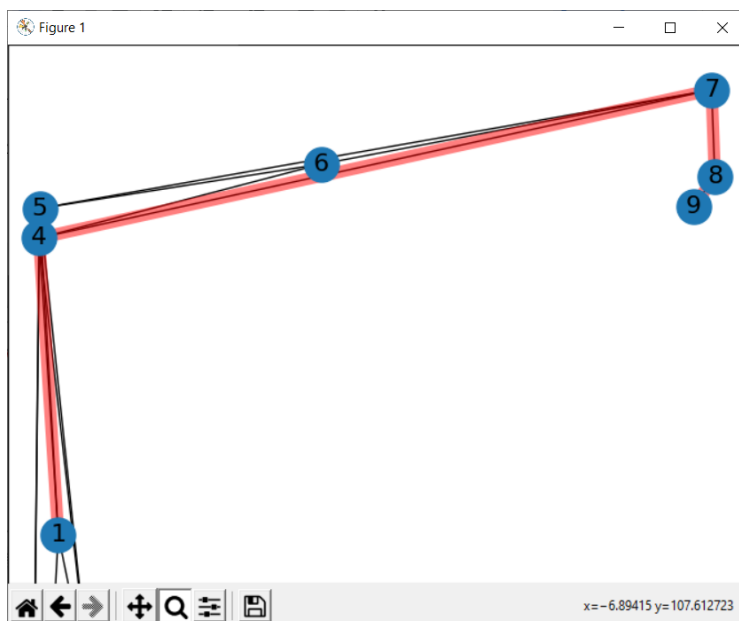
III. Hasil Eksekusi Program

Hasil eksekusi program adalah sebagai berikut untuk satu kasus setiap file uji ditampilkan sebagai berikut.

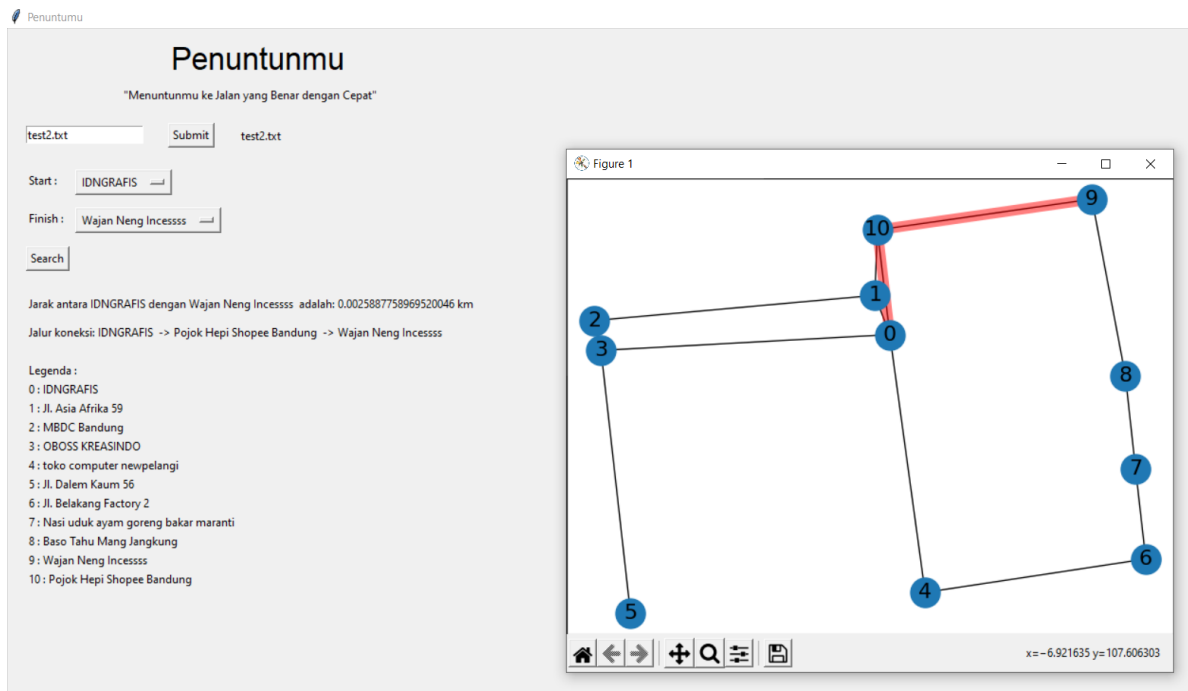
1. Menggunakan file uji test1.txt



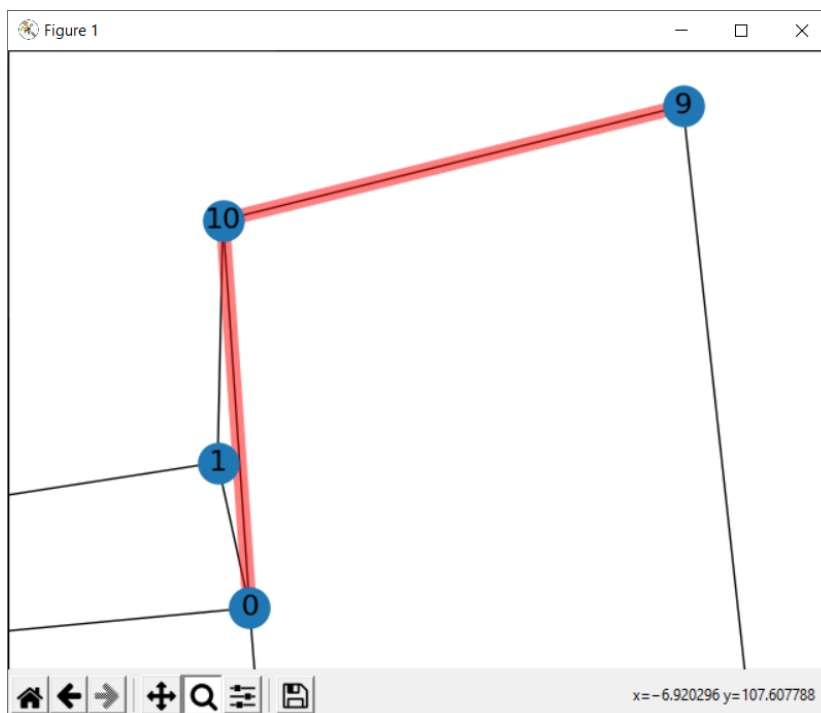
Visualisasi jalur koneksi dengan lebih jelas ditampilkan sebagai berikut.



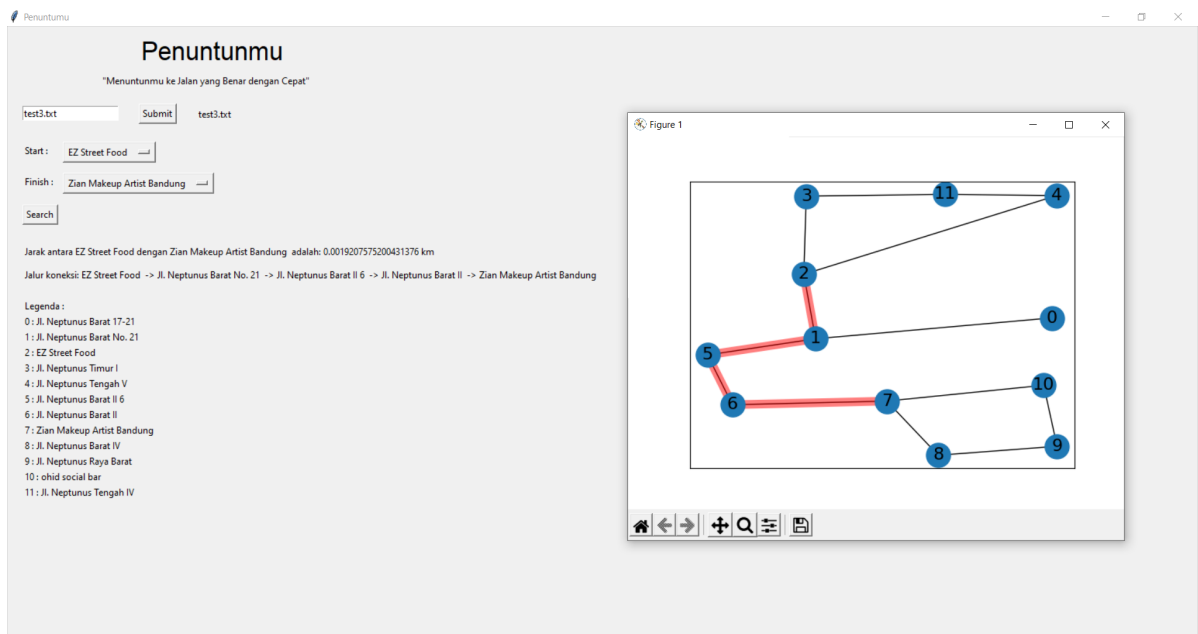
2. Menggunakan file uji test2.txt



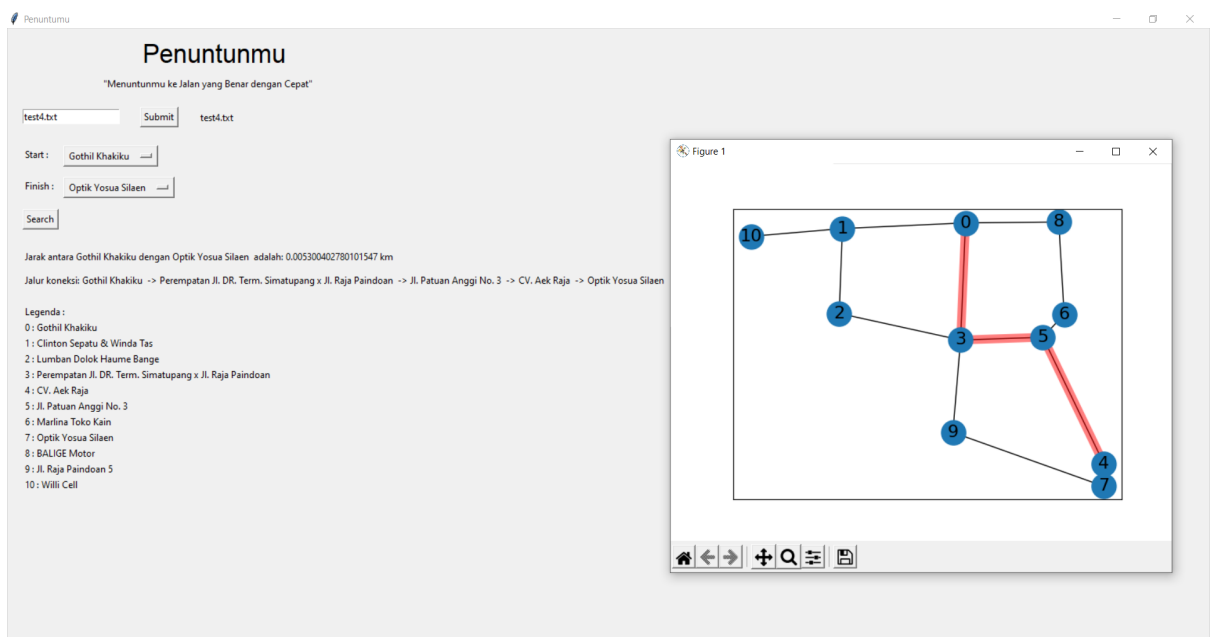
Visualisasi jalur koneksi dengan lebih jelas ditampilkan sebagai berikut.



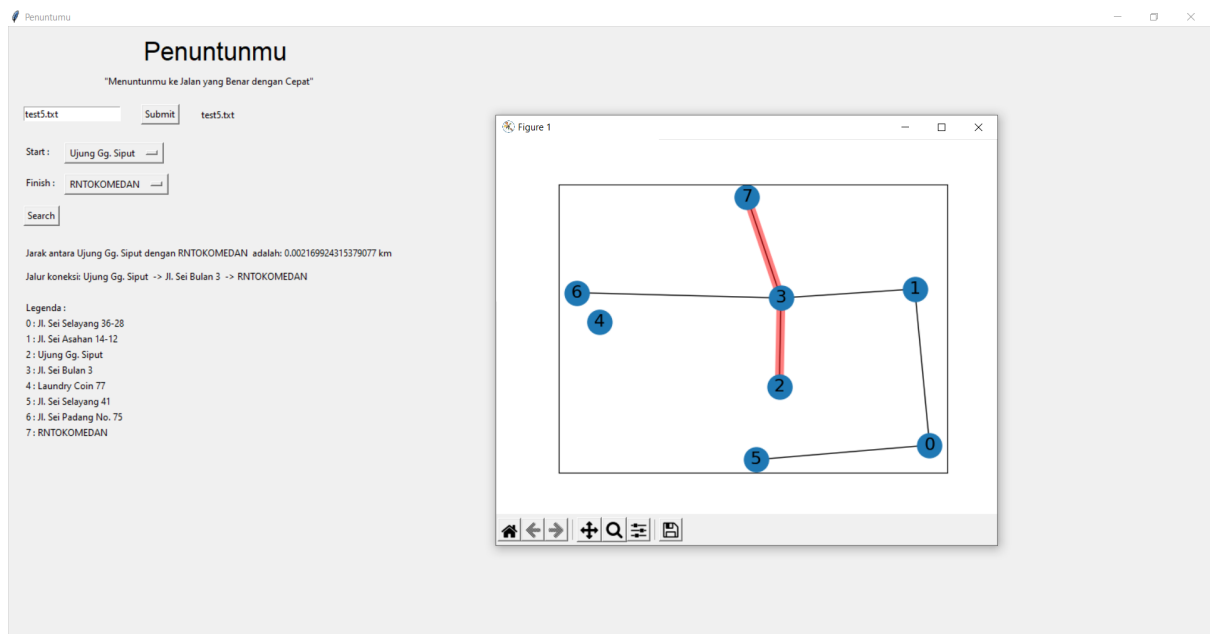
3. Menggunakan file uji test3.txt



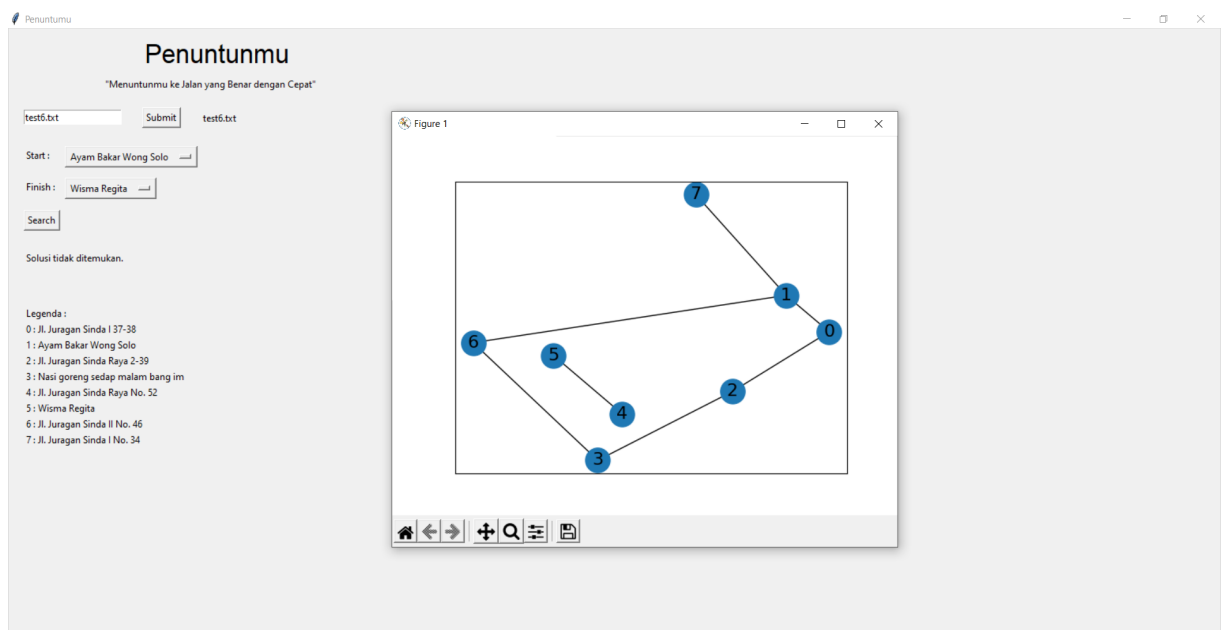
4. Menggunakan file uji test4.txt



5. Menggunakan file uji test5.txt



6. Menggunakan file uji test6.txt



IV. Komponen Penilaian

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		√

Alamat alternatif peletakan sumber kode program:

https://github.com/arjunamarcelino/IF2211-Tucil3_13519021