

# **Implementasi Algoritma A\* untuk Menentukan Lintasan Terpendek**

**Laporan Tugas Kecil 3 IF2211 Strategi Algoritma**

**Disusun sebagai syarat tugas kecil 3 mata kuliah IF2211 Strategi Algoritma**

**Semester II tahun 2020/2021**

**Disusun oleh:**

**Ferdy Irawan Firdaus (13519030)**

**Muhammad Rizal Muhamimin (13519136)**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO & INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2021**

## **BAB I**

### **ALGORITMA A\* (A Star)**

Fungsi evaluasi heuristik yang digunakan adalah jarak euclidean dari dua buah titik yang diketahui titik latitude dan longitude nya, dimana menerapkan algoritma haversine yang menerima masukan parameter berupa empat buah koordinat sudut dari dua buah titik yang akan dicari jaraknya. Kemudian untuk algoritma A star sendiri pertama menambahkan simpul asal ke dalam list hasil kemudian dicari tetangga simpul dari node asal dan diambil satu buah simpul yang memiliki nilai  $f_n$  paling kecil yang mana  $f_n = g_n + h_n$ ,  $g_n$  merupakan jarak antara simpul asal ke simpul tetangganya dan  $h_n$  merupakan jarak yang diperoleh dari fungsi haversine dari simpul tetangga tersebut ke simpul tujuan. Setelah diambil satu simpul yang memiliki nilai  $f_n$  paling kecil maka simpul asal diassign lagi menjadi simpul sekarang atau simpul yang terpilih (memiliki  $f_n$  paling minimal). Langkah ini diulangi sampai simpul asal sama dengan simpul tujuan.

## BAB II

### SOURCE CODE PROGRAM

**Main.py**

```
import pythonfile.BacaSqlite as BS
import pythonfile.TucilAstar as TA
import pythonfile.JarakEulidean as JE
import pythonfile.MakeHTML as MH
import pythonfile.MakeSqlite as MS
import os

def Tampilan_Awal():
    print(
        """
=====
Implementasi Algoritma A* untuk Menentukan Lintasan Terpendek
Tugas Kecil 2 IF2211 Strategi Algoritma
Semester II tahun 2020/2021
=====\\n"""
    )

def List_file_test():
    #membaca list file yang ada di file test
    print("\nDaftar file test :\n")
    path = "../test"
    Dir_test = os.listdir(path)
    for namefile in Dir_test:
        print(namefile)

def input_file():
    nfile = input("\nMasukkan Nama File\n:> ")
    try:
        path = "../test/"+str(nfile)
        MS.FiletoSql(path)
    except:
        print("\n===== =====\n")
        print("Maaf Nama File salah")
        print("\n===== =====\n")

def Daftar_Simpul():
    count = 1
    print("\nDaftar Lokasi : ")
    for text in BS.getSimpulAll():
        print(str(count) + str("."),text)
```

```

        count += 1

def Input_Asal_Tujuan():
    print("\n")
    NameSimpulAsal = input("Masukkan Nama Lokai Awal: > ")
    NameSimpulTujuan = input("Masukkan Nama Lokai Tujuan: > ")
    MH.ExportHTML(NameSimpulAsal, NameSimpulTujuan)
    print("Silahkan Buka file 'index.html' untuk melihat hasil")

#MAIN PROGRAM
Tampilan_Awal()
List_file_test ()
input_file ()
Daftar_Simpul()
Input_Asal_Tujuan()

```

### **TucilAstar.py**

```

import pythonfile.BacaSqlite as BS
import pythonfile.JarakEulidean as JE

def A_Star_Algoritma(NameSimpulAsal, NameSimpulTujuan):
    listKunjungi = []
    ListJawaban = [NameSimpulAsal]
    ListTetangga = BS.getTetangga(NameSimpulAsal)
    listResult = []

    #print(ListTetangga)
    gn = 0
    count = 0

    found = False
    while not found:

        for simpul in ListTetangga:

            ListSimpulJarak = [simpul, gn +
JE.distance(BS.getLitLng(ListJawaban[len(ListJawaban)-1]),BS.getLitLng(simpul)) +
JE.distance(BS.getLitLng(simpul),BS.getLitLng(NameSimpulTujuan))]
            listKunjungi.append (ListSimpulJarak)

            #print(listKunjungi)
            ListMin = []
            for i in range(len(listKunjungi)):
                if i == 0 :
                    ListMin = listKunjungi[i]

```

```

    elif ListMin[1] > listKunjungi[i][1]:
        ListMin = listKunjungi[i]

    ListJawaban.append(ListMin[0])
    #print("\n",ListMin, "\n")
    listKunjungi.remove(ListMin)
    #print(listKunjungi)

    if ListMin[0] == NameSimpulTujuan:
        found = True
    else:
        # Pencarian gn
        ListTetangga = BS.getTetangga(ListMin[0])
        gn = 0
        CopyList = ListJawaban.copy()
        Back = CopyList.pop()
        ABack = CopyList.pop()
        while ABack not in BS.getTetangga(Back) : ABack = CopyList.pop()

        while len(CopyList) != 0:
            gn += JE.distance(BS.getLitLng(ABack),BS.getLitLng(Back) )
            Back = ABack
            if len(CopyList) != 0:
                ABack = CopyList.pop()

    #simpul yang bakal dikunjungi selanjutnya
    ListTetangga = BS.getTetangga(ListMin[0])

gn = 0
#print(ListJawaban)
Back = ListJawaban.pop()
ListTetangga = BS.getTetangga(Back)
listResult.insert(0,Back)
while Back != NameSimpulAsal:
    for text in ListJawaban:
        if text in ListTetangga:
            #print(ListTetangga)
            gn += JE.distance(BS.getLitLng(text),BS.getLitLng(Back) )
            Back = text
            #print(text)
            listResult.insert(0,Back)
            ListTetangga = BS.getTetangga(text)
            break

listReturn = [listResult, gn]
return listReturn

```

```
def getLitLngHasil(NameSimpulAsal, NameSimpulTujuan):
    ListHasil = A_Star_Algoritma(NameSimpulAsal, NameSimpulTujuan) [0]
    listLitLngHasil = []
    for text in ListHasil:
        listLitLngHasil.append([float(BS.getLit(text)), float(BS.getLng(text))])
    return listLitLngHasil
```

# BAB III

## SCREENSHOT INPUT DAN OUTPUT

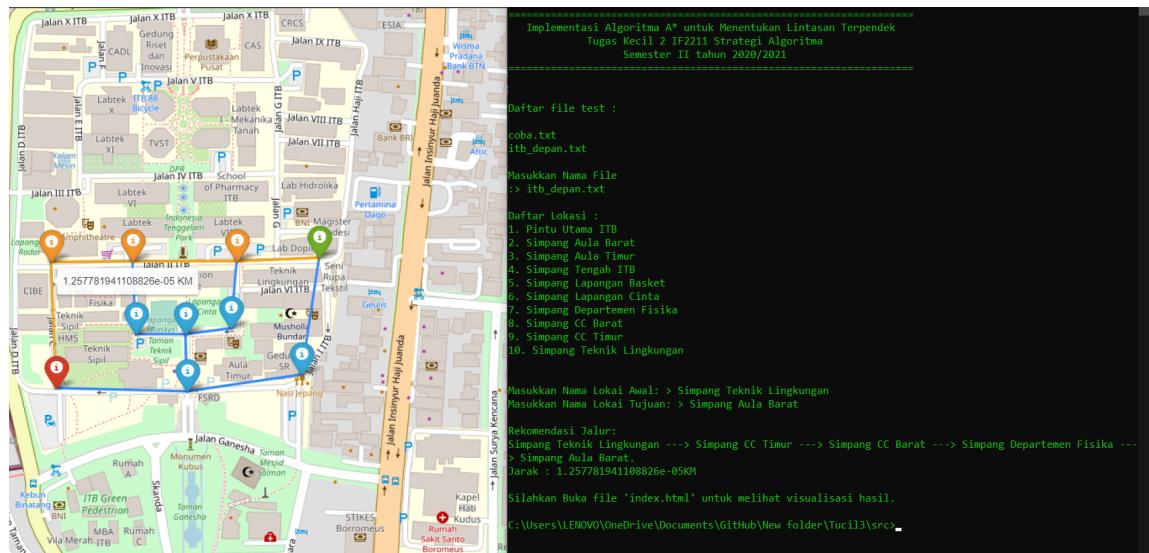
### 1. File Test “itb\_depat.txt”

#### 1.1. Input

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

Lokasi Awal: Simpang Teknik Lingkungan (berwarna hijau di map)  
Lokasi Tujuan : Simpang Aula Barat (berwarna merah di map)

#### 1.2. Output



Rekomendasi Jalur:

Simpang Teknik Lingkungan ---> Simpang CC Timur ---> Simpang CC Barat  
---> Simpang Departemen Fisika ---> Simpang Aula Barat.

Jarak : 1.257781941108826e-05KM

## 2. File Test "itb\_luar.txt"

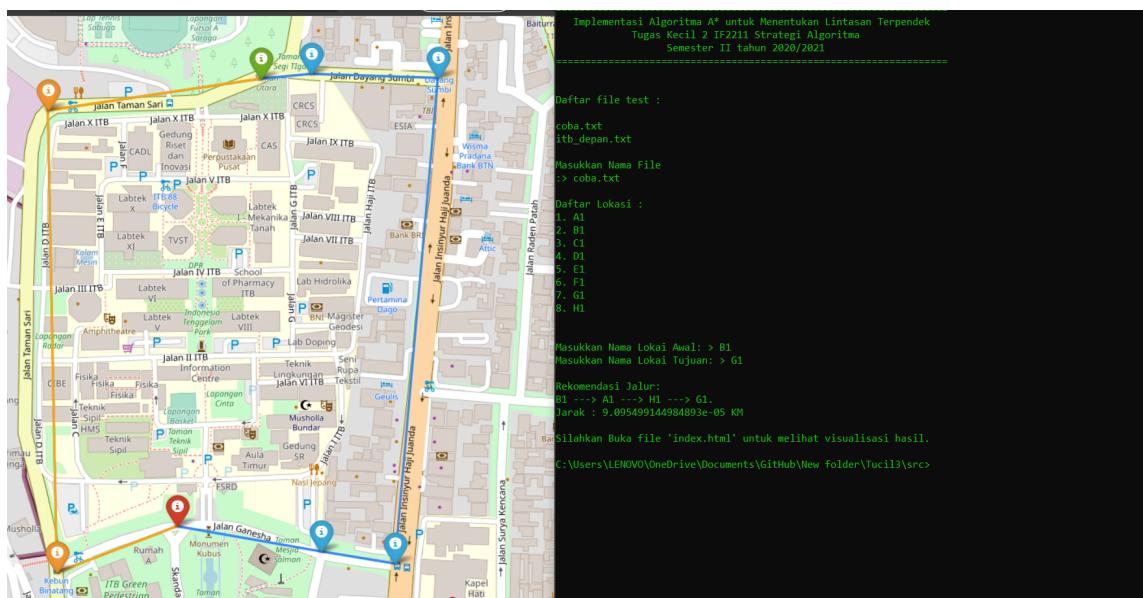
### 2.1. Input

```
8
-6.8878211 107.6083408 A1
-6.8873970 107.6111497 B1
-6.8873381 107.6118125 C1
-6.8873937 107.6134839 D1
-6.8937472 107.6129183 E1
-6.8935881 107.6119443 F1
-6.8932563 107.6100426 G1
-6.8938630 107.6084604 H1
0 1 0 0 0 0 0 1
1 0 1 0 0 0 0 0
0 1 0 1 0 0 0 0
0 0 1 0 1 0 0 0
0 0 0 1 0 1 0 0
0 0 0 0 1 0 1 0
0 0 0 0 0 1 0 1
1 0 0 0 0 0 1 0
```

Lokasi Awal: B1 (berwarna hijau di map)

Lokasi Tujuan : H1 (berwarna merah di map)

### 2.2. Output



Rekomendasi Jalur:

B1 ---> A1 ---> H1 ---> G1.

Jarak : 9.095499144984893e-05 KM

### 3. File Test “alunalun.txt”

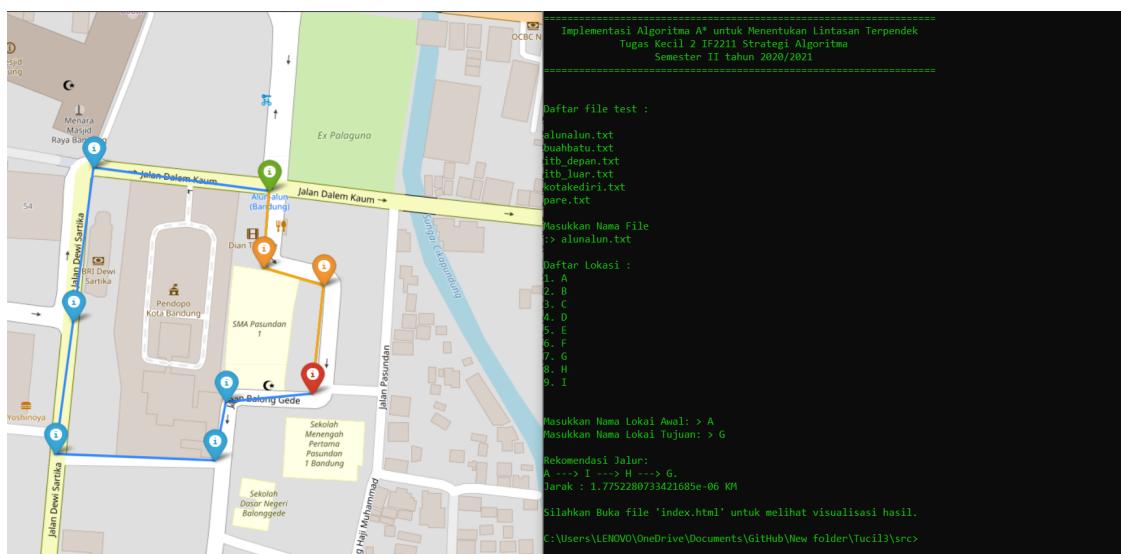
#### 3.1. Input

```
9
-6.9225673 107.6076125 A
-6.9224129 107.6064538 B
-6.9234204 107.6063224 C
-6.9242838 107.6062027 D
-6.9243300 107.6072524 E
-6.9239429 107.6073282 F
-6.9238897 107.6078955 G
-6.9231811 107.6079750 H
-6.9230689 107.6075733 I
0 1 0 0 0 0 0 0 1
1 0 1 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0
0 0 1 0 1 0 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 1 0
0 0 0 0 0 0 1 0 1
1 0 0 0 0 0 0 1 0
```

Lokasi Awal: A (berwarna hijau di map)

Lokasi Tujuan : G (berwarna merah di map)

#### 3.2. Output



Rekomendasi Jalur:

A ---> I ---> H ---> G.

Jarak : 1.7752280733421685e-06 KM

## 4. File Test “buahbatu.txt”

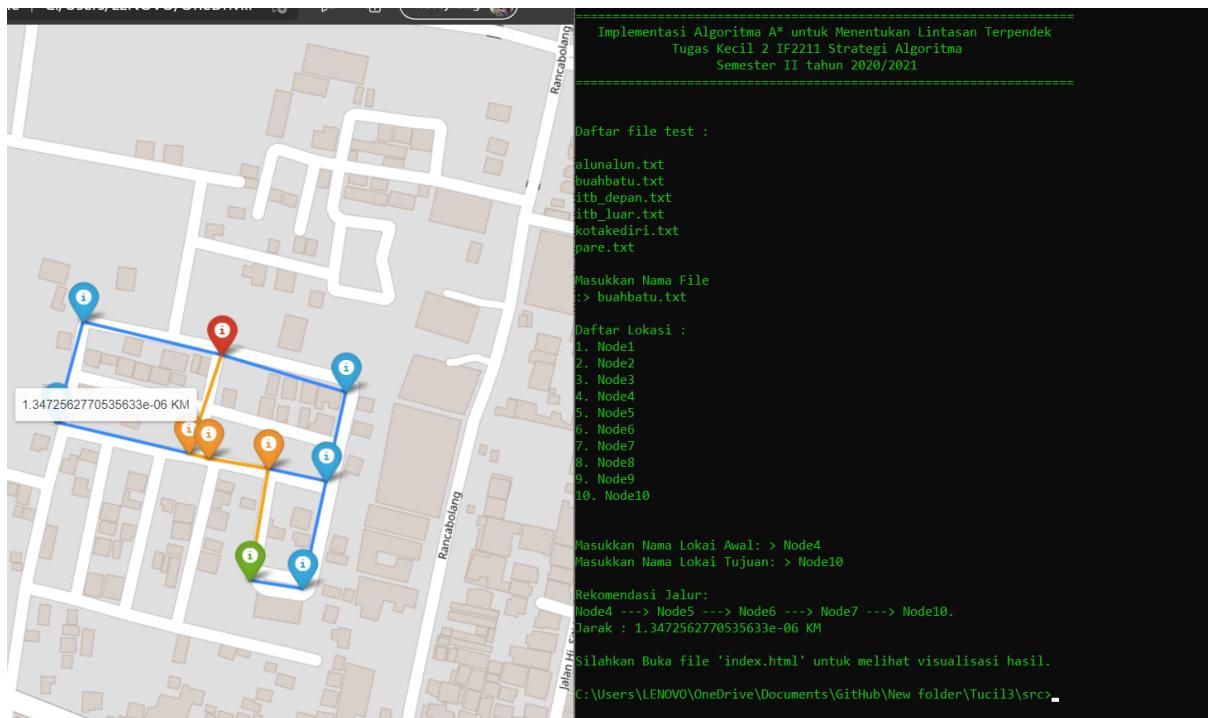
### 4.1. Input

```
10
-6.9566251 107.6596554 Node1
-6.9570851 107.6595525 Node2
-6.9576372 107.6594268 Node3
-6.9575969 107.6591552 Node4
-6.9570202 107.6592487 Node5
-6.9569656 107.6589420 Node6
-6.9569373 107.6588360 Node7
-6.9567799 107.6581494 Node8
-6.9562637 107.6582929 Node9
-6.9564324 107.6590100 Node10
0 1 0 0 0 0 0 0 0 1
1 0 1 0 1 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 1 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 0 1
0 0 0 0 0 0 1 0 1 0
0 0 0 0 0 0 0 1 0 1
1 0 0 0 0 0 1 0 1 0
```

Lokasi Awal: Node4 (berwarna hijau di map)

Lokasi Tujuan : Node10 (berwarna merah di map)

### 4.2. Output



Rekomendasi Jalur:

Node4 ---> Node5 ---> Node6 ---> Node7 ---> Node10.

Jarak : 1.3472562770535633e-06 KM

## 5. File Test “kotakediri.txt”

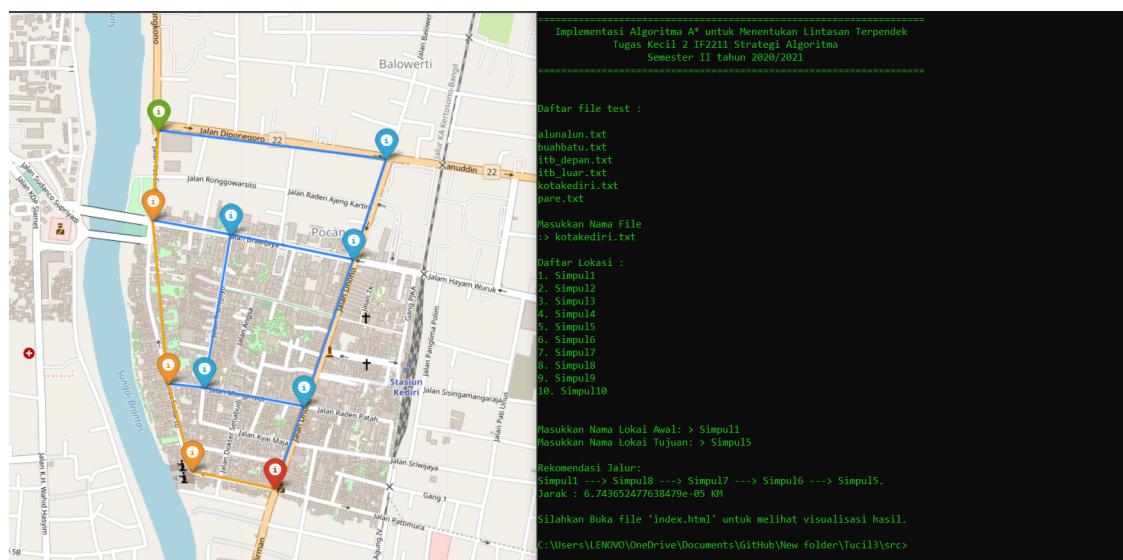
### 5.1. Input

```
10
-7.810977 112.008886 Simpul1
-7.8117433 112.0149795 Simpul2
-7.8144053 112.0140826 Simpul3
-7.8182976 112.0127539 Simpul4
-7.8204925 112.0119851 Simpul5
-7.8200105 112.0097686 Simpul6
-7.8177103 112.0091400 Simpul7
-7.8133716 112.0087279 Simpul8
-7.8137370 112.0108060 Simpul9
-7.8178123 112.0100992 Simpul10
0 1 0 0 0 0 0 1 0 0
1 0 1 0 0 0 0 0 0 0
0 1 0 1 0 0 0 0 1 0
0 0 1 0 1 0 0 0 0 1
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 0 1
1 0 0 0 0 0 1 0 1 0
0 0 1 0 0 0 0 1 0 1
0 0 0 1 0 0 1 0 1 0
```

Lokasi Awal: Simpul1 (berwarna hijau di map)

Lokasi Tujuan : Simpul5 (berwarna merah di map)

### 5.2. Output



Rekomendasi Jalur:

Simpul1 ---> Simpul8 ---> Simpul7 ---> Simpul6 ---> Simpul5.

Jarak : 6.743652477638479e-05 KM

## 6. File Test “pare.txt”

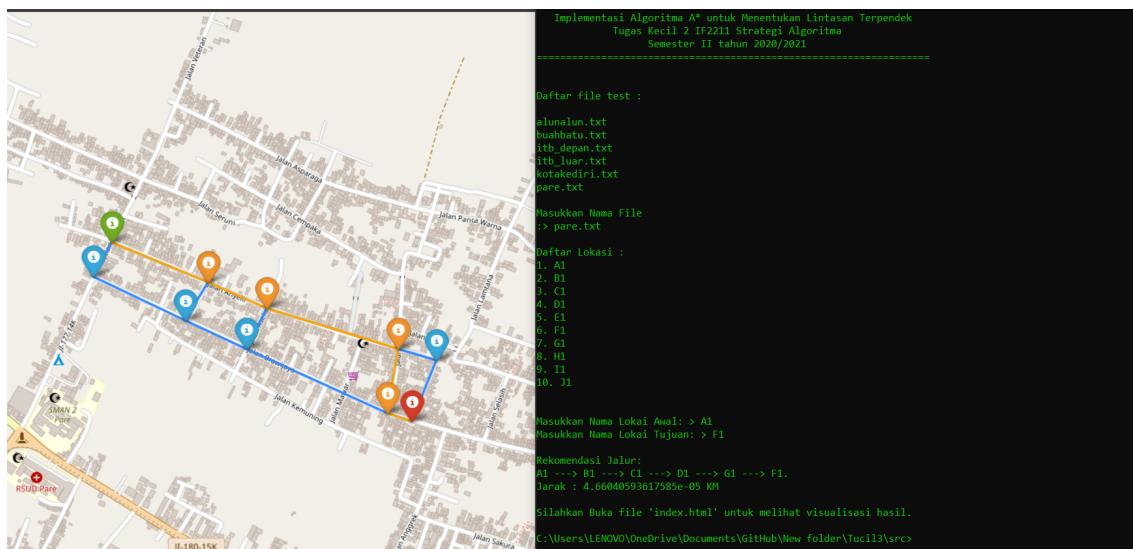
### 6.1. Input

```
10
-7.7537520 112.1778359 A1
-7.7547978 112.1803722 B1
-7.7554649 112.1819199 C1
-7.756535 112.185372 D1
-7.7568332 112.1863750 E1
-7.7584338 112.1857309 F1
-7.758226 112.185097 G1
-7.7565406 112.1813804 H1
-7.7558021 112.1797754 I1
-7.7546440 112.1773319 J1
0 1 0 0 0 0 0 0 0 1
1 0 1 0 0 0 0 0 1 0
0 1 0 1 0 0 0 1 0 0
0 0 1 0 1 0 1 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 1 0 1 0 1 0 0
0 0 1 0 0 0 1 0 1 0
0 1 0 0 0 0 0 1 0 1
1 0 0 0 0 0 0 0 1 0
```

Lokasi Awal: A1 (berwarna hijau di map)

Lokasi Tujuan : F1 (berwarna merah di map)

### 6.2. Output



Rekomendasi Jalur:

A1 ---> B1 ---> C1 ---> D1 ---> G1 ---> F1.

Jarak : 4.66040593617585e-05 KM

## **BAB IV**

### **ALAMAT REPOSITORY**

<https://github.com/ferdyirawanf/Tucil3.git>

Poin	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	✓