

## **TUGAS KECIL III**

**Algoritma A\* Pada Pencarian Rute Terpendek**

**IF2211 – Strategi Algoritma**

**Oleh**

**Ariya Adinatha 13519048**

**Nathaniel Jason 13519108**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2021**

## Kode Program

```
import math

def getIndexCity(city, nodes, adj):
    index = 0
    for i in nodes:
        if (i["name"] == city):
            return index
    index += 1


def getAdj(city, nodes, adj):
    indexCity = getIndexCity(city, nodes, adj)
    indexAdj = 0
    arrayAdj = []
    for i in adj[indexCity]:
        if (i != 0):
            # arrayAdj.append(nodes[indexAdj]["name"])
            arrayAdj.append(nodes[indexAdj])
        indexAdj += 1
    return arrayAdj


def getAdjIndex(city, nodes, adj):
    indexCity = getIndexCity(city, nodes, adj)
    indexAdj = 0
    arrayAdj = []
    for i in adj[indexCity]:
        if (i != 0):
            # arrayAdj.append(nodes[indexAdj]["name"])
            arrayAdj.append(indexAdj)
        indexAdj += 1
    return arrayAdj


def getDistance(lng1, lat1, lng2, lat2):
    distLon = (lng2 - lng1) * math.pi / 180.0
    distLat = (lat2 - lat1) * math.pi / 180.0
    lat1 = (lat1) * math.pi / 180.0
```

```

lat2 = (lat2) * math.pi / 180.0

a = (((math.sin(distLon/2))**2) *
      math.cos(lat1) * math.cos(lat2)) + (math.sin(distLat/2)**2))
c = 2 * math.asin(math.sqrt(a))
dist = c * 6371
return dist


def getDistanceBetweenCity(nodeKotaAwal, nodeKotaTujuan):
    distance = getDistance(
        nodeKotaAwal["longitude"], nodeKotaAwal["latitude"],
        nodeKotaTujuan["longitude"], nodeKotaTujuan["latitude"])
    return distance

def getProcessMinimumFn(nodeProcessArray):
    minProcess = min(nodeProcessArray, key = lambda el: el["fn"])

    return minProcess


def getMinimum(arrayRute):
    for i in arrayRute:
        if (i != 0):
            minimum = i
            break
    for i in range(len(arrayRute)):
        minimum = arrayRute[i]
    return minimum

def hn(kotaAwal, kotaTujuan):
    return getDistanceBetweenCity(kotaAwal, kotaTujuan)

def gn(nodeProcessAwal, kotaNext, nodes, adj):
    nameKotaAwal = nodeProcessAwal["node"]["name"]
    indexKotaAwal = getIndexCity(nameKotaAwal, nodes, adj)
    nodeKotaAwal = nodes[indexKotaAwal]

    indexkotaNext = getIndexCity(kotaNext, nodes, adj)

```

```

nodekotaNext = nodes[indexkotaNext]

    return getDistanceBetweenCity(nodeKotaAwal, nodekotaNext) +
nodeProcessAwal["gn"]

def fn(nodeProcessAwal, kotaNext, kotaTujuan, nodes, adj):
    # list node yg sudah dikunjungi
    # fn
    # node itu sendiri
    #
    #

    indexKotaTujuan = getIndexCity(kotaTujuan, nodes, adj)
    nodeKotaTujuan = nodes[indexKotaTujuan]

    nodeProcessNext = {}

    indexkotaNext = getIndexCity(kotaNext, nodes, adj)
    nodekotaNext = nodes[indexkotaNext]
    nodeProcessNext["node"] = nodekotaNext

    nameKotaAwal = nodeProcessAwal["node"]["name"]
    indexKotaAwal = getIndexCity(nameKotaAwal, nodes, adj)
    nodeProcessNext["visited"] = [indexKotaVisited for indexKotaVisited in
nodeProcessAwal["visited"]]
    nodeProcessNext["visited"].append(indexKotaAwal)

    f = 0

    h = hn(nodekotaNext, nodeKotaTujuan)
    nodeProcessNext["hn"] = h
    f += h

    g = gn(nodeProcessAwal, kotaNext, nodes, adj)
    nodeProcessNext["gn"] = g
    f += g

    nodeProcessNext["fn"] = f

```

```
return nodeProcessNext

def isPathAvailable(kotaAwal, kotaTujuan, nodes, adj):
    indexKotaAwal = getIndexCity(kotaAwal, nodes, adj)
    indexKotaTujuan = getIndexCity(kotaTujuan, nodes, adj)

    kotaVisited = [False for node in nodes]
    nextKotaVisitTargetIndex = []

    kotaCurrent = kotaAwal
    indexKotaCurrent = getIndexCity(kotaCurrent, nodes, adj)
    nodeKotaCurrent = nodes[indexKotaCurrent]

    tetangga = getAdjIndex(kotaCurrent, nodes, adj)

    for indexTetangga in tetangga:
        if (not(kotaVisited[indexTetangga])):
            kotaVisited[indexTetangga] = True
            nextKotaVisitTargetIndex.append(indexTetangga)

    while (kotaCurrent != kotaTujuan and len(nextKotaVisitTargetIndex) != 0):

        # re assign kota current
        indexKotaCurrent = nextKotaVisitTargetIndex[0]
        nodeKotaCurrent = nodes[indexKotaCurrent]
        kotaCurrent = nodeKotaCurrent["name"]

        del nextKotaVisitTargetIndex[0]

        tetangga = getAdjIndex(kotaCurrent, nodes, adj)

        for indexTetangga in tetangga:
            if (not(kotaVisited[indexTetangga])):
                kotaVisited[indexTetangga] = True
                nextKotaVisitTargetIndex.append(indexTetangga)
```

```

    return kotaCurrent == kotaTujuan


def aStarPath(kotaAwal, kotaTujuan, nodes, adj):
    if (isPathAvailable(kotaAwal, kotaTujuan, nodes, adj)):
        indexKotaAwal = getIndexCity(kotaAwal, nodes, adj)
        indexKotaTujuan = getIndexCity(kotaTujuan, nodes, adj)

        nextNodesProcess = []

        kotaCurrent = kotaAwal
        indexKotaCurrent = getIndexCity(kotaCurrent, nodes, adj)
        nodeKotaCurrent = nodes[indexKotaCurrent]
        nodeProcessCurrent = {
            "node": nodeKotaCurrent,
            "gn": 0,
            "visited": []
        }

    while (kotaCurrent != kotaTujuan):
        tetangga = getAdjIndex(kotaCurrent, nodes, adj)

        for indexTetangga in tetangga:
            nodeTetangga = nodes[indexTetangga]
            kotaTetangga = nodeTetangga["name"]

            # fn
            nodeTetanggaProcess = fn(nodeProcessCurrent, kotaTetangga,
kotaTujuan, nodes, adj)
            nextNodesProcess.append(nodeTetanggaProcess)

        # get the minimum value of fn
        minProcess = getProcessMinimumFn(nextNodesProcess)
        indexMinProcess = nextNodesProcess.index(minProcess)
        del nextNodesProcess[indexMinProcess]

        # reassign kota current state
        kotaCurrent = minProcess["node"]["name"]
        indexKotaCurrent = getIndexCity(kotaCurrent, nodes, adj)
        nodeKotaCurrent = nodes[indexKotaCurrent]

```

```
    nodeProcessCurrent = minProcess

    nodeProcessCurrent["path"] = [nodeIndex for nodeIndex in
nodeProcessCurrent["visited"]]
    nodeProcessCurrent["path"].append(indexKotaCurrent)

    nodeProcessCurrent["error"] = False

    return nodeProcessCurrent
else :
    return {"error": True}
```

## Peta/Graf Input

### Test1

```
nodes = [
  {
    "longitude": 107.6084493845701,
    "latitude": -6.893831423890319,
    "name": "Kebun Binatang"
  },
  {
    "longitude": 107.61043690145013,
    "latitude": -6.8932512622009625,
    "name": "Monumen Kubus"
  },
  {
    "longitude": 107.61294895783067,
    "latitude": -6.893787986679996,
    "name": "Rumah Sakit Boromeus"
  },
  {
    "longitude": 107.61037856340408,
    "latitude": -6.891037787385059,
    "name": "Information Center"
  },
  {
    "longitude": 107.61036951094864,
    "latitude": -6.889920394744195,
    "name": "DPR"
  },
  {
    "longitude": 107.61157147586343,
    "latitude": -6.889900090593812,
    "name": "Kantin Bengkok"
  },
  {
    "longitude": 107.61150576174259,
    "latitude": -6.8878869796945565,
    "name": "CRCS"
  },
  {
    "longitude": 107.61212803423406,
    "latitude": -6.891001173429032,
    "name": "Lab Doping"
  },
]
```

```

{
  "longitude": 107.6093492656946,
  "latitude": -6.887952219533602,
  "name": "CADL"
},
{
  "longitude": 107.60869715362789,
  "latitude": -6.8910907111892215,
  "name": "CIBE"
},
{
  "longitude": 107.60879170149563,
  "latitude": -6.8926744283312615,
  "name": "Parkir Sipil"
},
{
  "longitude": 107.6118289679289,
  "latitude": -6.892560259744101,
  "name": "Parkir SR"
},
{
  "longitude": 107.61041980236767,
  "latitude": -6.8926551228578035,
  "name": "Jalan ITB"
},
{
  "longitude": 107.60904148221015,
  "latitude": -6.89012443476463,
  "name": "GKUB"
}
]

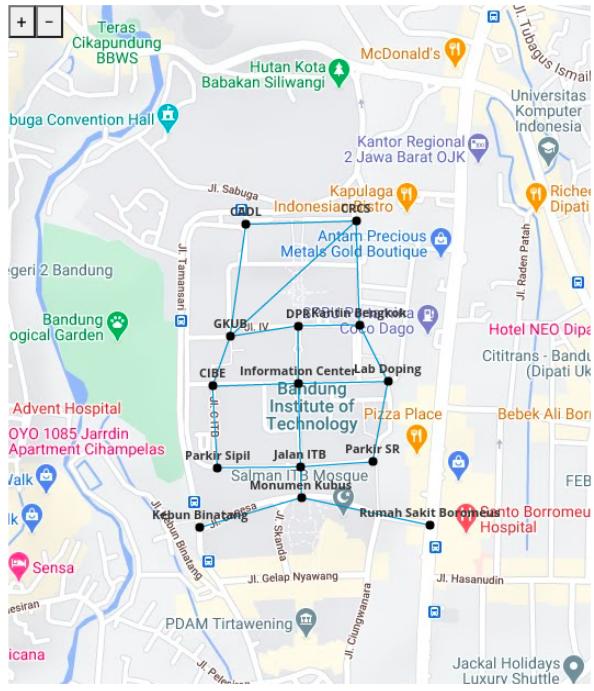
```

```

adj = [
  [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, 1, 0],
  [0, 1, 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, 1, 0, 0],
  [0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
  [0, 0, 0, 0, 1, 0, 1, 1, 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, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0],
  [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
  [0, 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, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0]
]
```

```
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0],  
[0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0],  
[0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0]
```

]



# Adjacency Matrix

	01.	02.	03.	04.	05.	06.	07.	08.	09.	10.	11.	12.	13.	14.
01.	0	1	0	0	0	0	0	0	0	0	0	0	0	0
02.	1	0	1	0	0	0	0	0	0	0	0	0	1	0
03.	0	1	0	0	0	0	0	0	0	0	0	0	0	0
04.	0	0	0	0	1	0	0	1	0	1	0	0	1	0
05.	0	0	0	1	0	1	0	0	0	0	0	0	0	1
06.	0	0	0	0	1	0	1	1	0	0	0	0	0	0
07.	0	0	0	0	0	0	0	0	1	0	0	0	0	1
08.	0	0	0	1	0	1	0	0	0	0	0	1	0	0
09.	0	0	0	0	0	0	1	0	0	0	0	0	0	1
10.	0	0	0	1	0	0	0	0	0	0	1	0	0	1
11.	0	0	0	0	0	0	0	0	0	1	0	0	1	0
12.	0	0	0	0	0	0	0	1	0	0	0	0	1	0
13.	0	1	0	1	0	0	0	0	0	0	1	1	0	0
14.	0	0	0	0	1	0	0	0	1	1	0	0	0	0

## Node List

- |     |   |  |
|-----|---|--|
| 01. | X | Kebun Binatang  107.6084493845701, -6.893831423890319        |
| 02. | X | Monumen Kubus  107.61043690145013, -6.8932512622009625       |
| 03. | X | Rumah Sakit Boromeus  107.61294895783067, -6.893787986679996 |
| 04. | X | Information Center  107.61037856340408, -6.891037787385059   |
| 05. | X | DPR  107.61036951094864, -6.889920394744195                  |
| 06. | X | Kantin Bengkok  107.61157147586343, -6.889900090593812       |
| 07. | X | CRC  107.61150576174259, -6.8878869796945565                 |
| 08. | X | Lab Doping  107.61212803423406, -6.891001173429032           |
| 09. | X | CADL  107.6093492656946, -6.887952219533602                  |
| 10. | X | CIBE  107.60869715362789, -6.8910907111892215                |
| 11. | X | Parkir Sipil  107.60879170149563, -6.8926744283312615        |
| 12. | X | Parkir SR  107.6118289679289, -6.892560259744101             |
| 13. | X | Jalan ITB  107.61041980236767, -6.8926551228578035           |
| 14. | X | GKUB  107.60904148221015, -6.89012443476463                  |

Hasil Test1

**Submit**

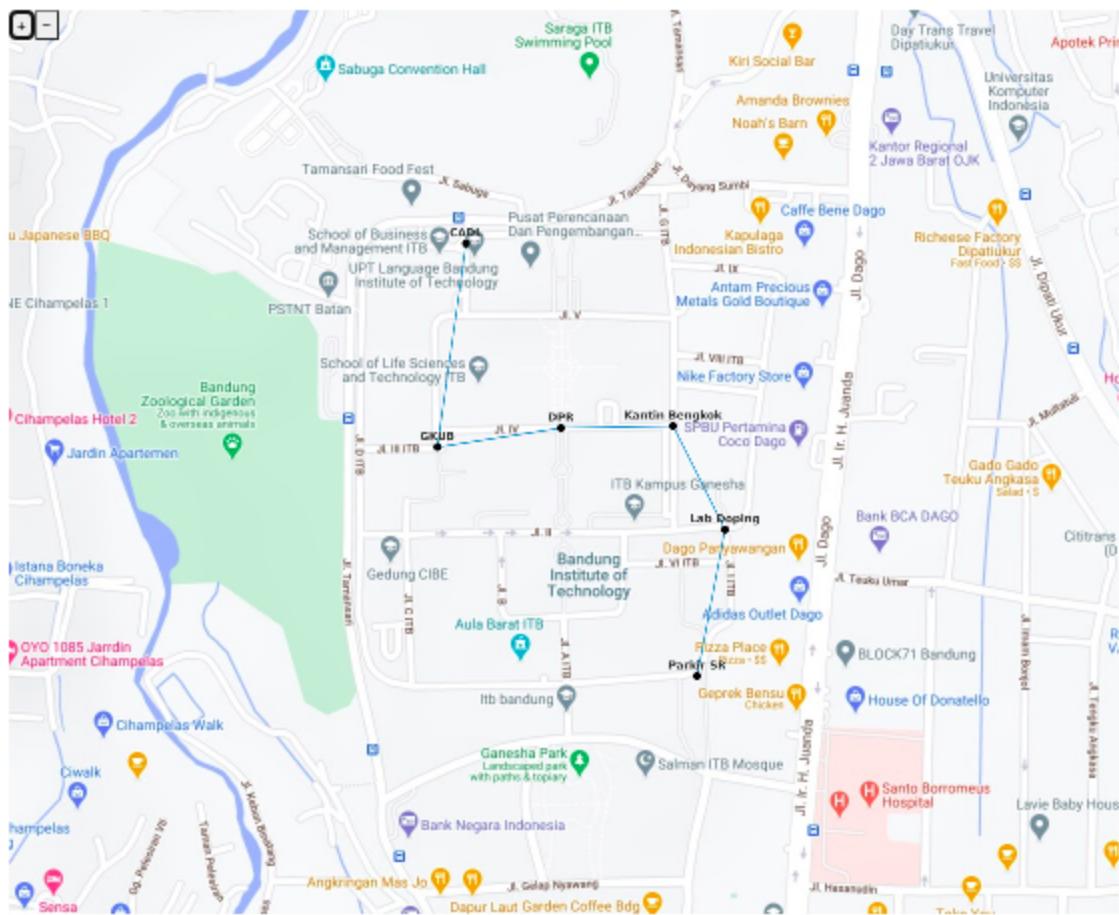
**Node Start : CADL**

**Node Destination : Parkir SR**

**Find Path**

**Path Result**

Jarak lintasan: 0.8384364084834666 KM



## Test2

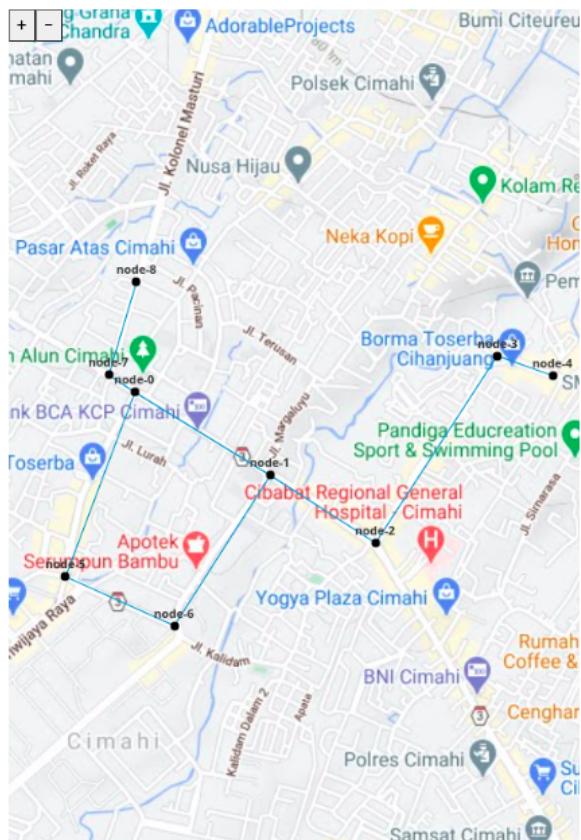
```
nodes = [
  {
    "longitude": 107.54198661068948,
    "latitude": -6.873510118150591,
    "name": "node-0"
  },
  {
    "longitude": 107.54613830448444,
    "latitude": -6.876028983371896,
    "name": "node-1"
  },
  {
    "longitude": 107.54936864336027,
    "latitude": -6.878077142567463,
    "name": "node-2"
  },
  {
    "longitude": 107.55309836817929,
    "latitude": -6.872406473525132,
    "name": "node-3"
  },
  {
    "longitude": 107.55479661121187,
    "latitude": -6.8730063081909805,
    "name": "node-4"
  },
  {
    "longitude": 107.539861935881,
    "latitude": -6.879103181123938,
    "name": "node-5"
  },
  {
    "longitude": 107.5431973964604,
    "latitude": -6.880586220342579,
    "name": "node-6"
  },
  {
    "longitude": 107.54118694268483,
    "latitude": -6.872989988584507,
    "name": "node-7"
  },
  {
```

```

    "longitude": 107.54203729437575,
    "latitude": -6.87016899070025,
    "name": "node-8"
}
]

adj = [
    [0, 1, 0, 0, 0, 1, 0, 1, 0],
    [1, 0, 1, 0, 0, 0, 1, 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, 0, 0, 0],
    [1, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 1, 0, 0, 0, 1, 0, 0, 0],
    [1, 0, 0, 0, 0, 0, 0, 0, 1],
    [0, 0, 0, 0, 0, 0, 0, 1, 0]
]

```



	01.	02.	03.	04.	05.	06.	07.	08.	09.
01.	0	1	0	0	0	1	0	1	0
02.	1	0	1	0	0	0	1	0	0
03.	0	1	0	1	0	0	0	0	0
04.	0	0	1	0	1	0	0	0	0
05.	0	0	0	1	0	0	0	0	0
06.	1	0	0	0	0	0	1	0	0
07.	0	1	0	0	0	1	0	0	0
08.	1	0	0	0	0	0	0	0	1
09.	0	0	0	0	0	0	0	1	0

Adjacency Matrix

## Node List

- |     |   |   |
|-----|---|---|
| 01. | X | node-0  107.54198661068948, -6.873510118150591  |
| 02. | X | node-1  107.54613830448444, -6.876028983371896  |
| 03. | X | node-2  107.54936864336027, -6.878077142567463  |
| 04. | X | node-3  107.55309836817929, -6.872406473525132  |
| 05. | X | node-4  107.55479661121187, -6.8730063081909805 |
| 06. | X | node-5  107.539861935881, -6.879103181123938    |
| 07. | X | node-6  107.5431973964604, -6.880586220342579   |
| 08. | X | node-7  107.54118694268483, -6.872989988584507  |
| 09. | X | node-8  107.54203729437575, -6.87016899070025   |

Hasil Test2

Submit

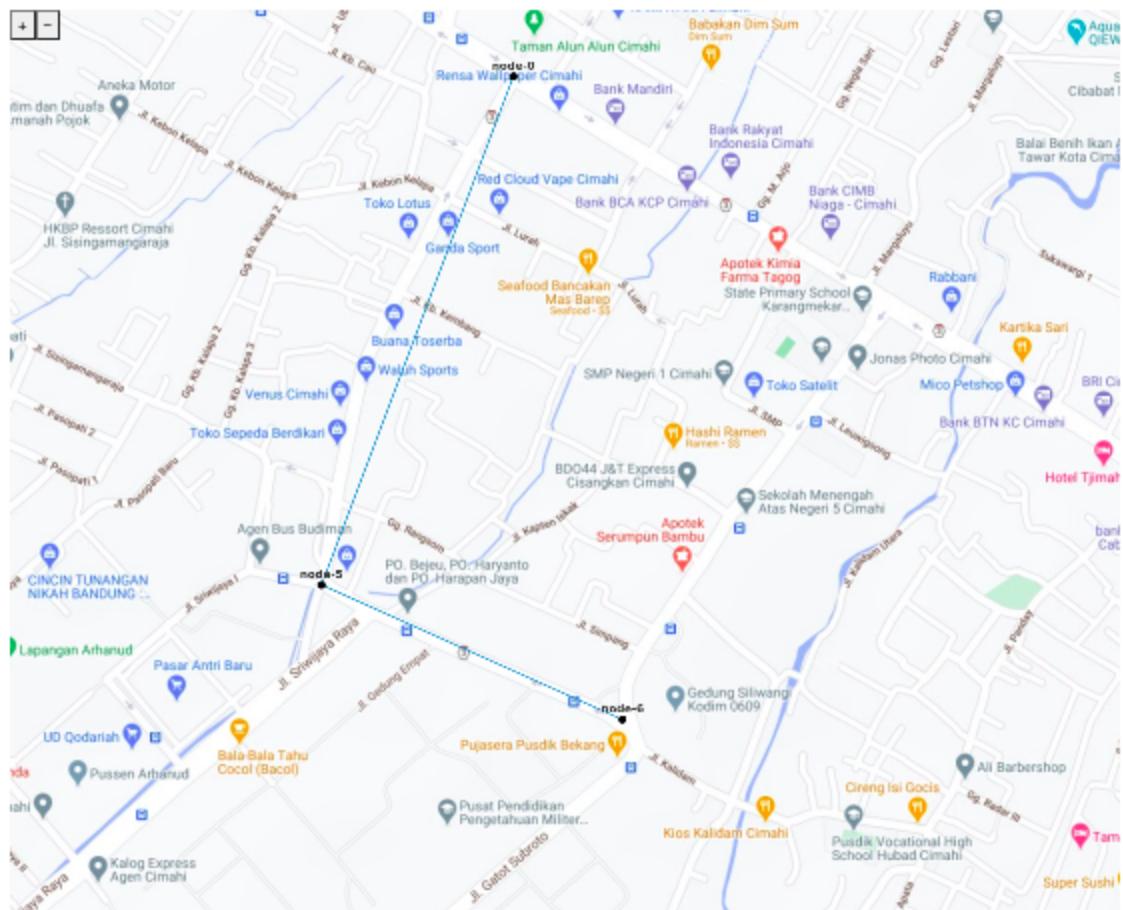
Node Start : node-0

Node Destination : node-6

Find Path

### Path Result

Jarak lintasan: 1.0681367974811202 KM



### Test3

```
nodes = [
  {
    "longitude": 107.65927442576984,
    "latitude": -6.942072697375892,
    "name": "node-0"
  },
  {
    "longitude": 107.65927442576984,
    "latitude": -6.942072697375892,
    "name": "node-1"
  },
  {
    "longitude": 107.65927442576984,
    "latitude": -6.942072697375892,
    "name": "node-2"
  },
  {
    "longitude": 107.65927442576984,
    "latitude": -6.942072697375892,
    "name": "node-3"
  },
  {
    "longitude": 107.65927442576984,
    "latitude": -6.942072697375892,
    "name": "node-4"
  },
  {
    "longitude": 107.65927442576984,
    "latitude": -6.942072697375892,
    "name": "node-5"
  },
  {
    "longitude": 107.65927442576984,
    "latitude": -6.942072697375892,
    "name": "node-6"
  }
]
```

```
"longitude": 107.6591411470107,  
"latitude": -6.943466952474225,  
"name": "node-1"  
},  
{  
    "longitude": 107.65992496160887,  
    "latitude": -6.94358830220726,  
    "name": "node-2"  
},  
{  
    "longitude": 107.65991998646375,  
    "latitude": -6.943958290525941,  
    "name": "node-4"  
},  
{  
    "longitude": 107.66129776458288,  
    "latitude": -6.944114057014474,  
    "name": "node-5"  
},  
{  
    "longitude": 107.65945702923541,  
    "latitude": -6.947042305836376,  
    "name": "node-6"  
},  
{  
    "longitude": 107.65912364028033,  
    "latitude": -6.946985246981754,  
    "name": "node-7"  
},  
{  
    "longitude": 107.65923285390356,  
    "latitude": -6.946038068983839,  
    "name": "node-8"  
},  
{  
    "longitude": 107.65737510606941,  
    "latitude": -6.945889494021216,  
    "name": "node-9"  
},  
{  
    "longitude": 107.65762316599084,  
    "latitude": -6.943629362080301,  
    "name": "node-10"  
},
```

```

{
  "longitude": 107.65816118464389,
  "latitude": -6.943636215575268,
  "name": "node-11"
},
{
  "longitude": 107.65818992507104,
  "latitude": -6.943390860674555,
  "name": "node-12"
},
{
  "longitude": 107.65828514036842,
  "latitude": -6.941949228399977,
  "name": "node-13"
},
{
  "longitude": 107.65827347966398,
  "latitude": -6.940266763457316,
  "name": "node-14"
},
{
  "longitude": 107.65948110376118,
  "latitude": -6.939967069503822,
  "name": "node-15"
},
{
  "longitude": 107.66247795316042,
  "latitude": -6.942262530791154,
  "name": "node-16"
},
{
  "longitude": 107.66125520458608,
  "latitude": -6.942047988821244,
  "name": "node-17"
}
]

```

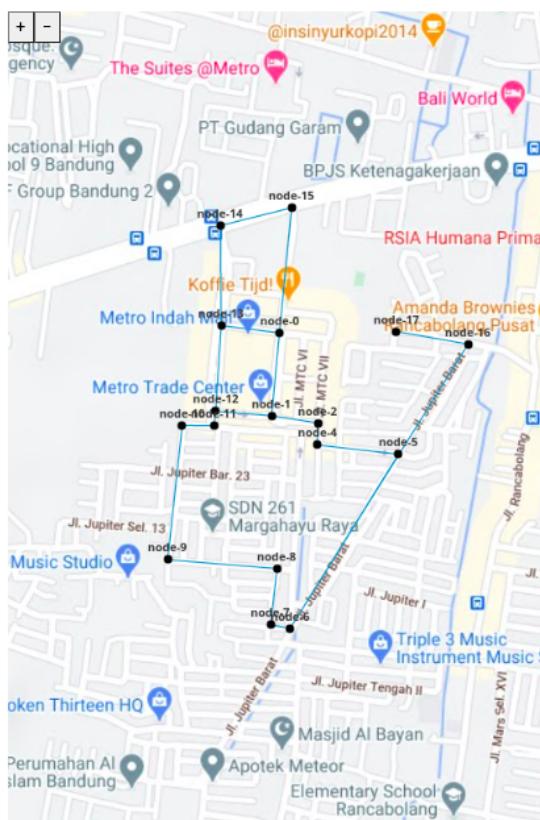
```

adj = [
  [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0],
  [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
  [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, 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, 1, 0],

```

```
[0, 0, 0, 0, 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, 1, 0, 1, 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, 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, 1, 0, 1, 0, 0],  
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0],  
[1, 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, 1, 0, 1, 0, 0],  
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
```

]



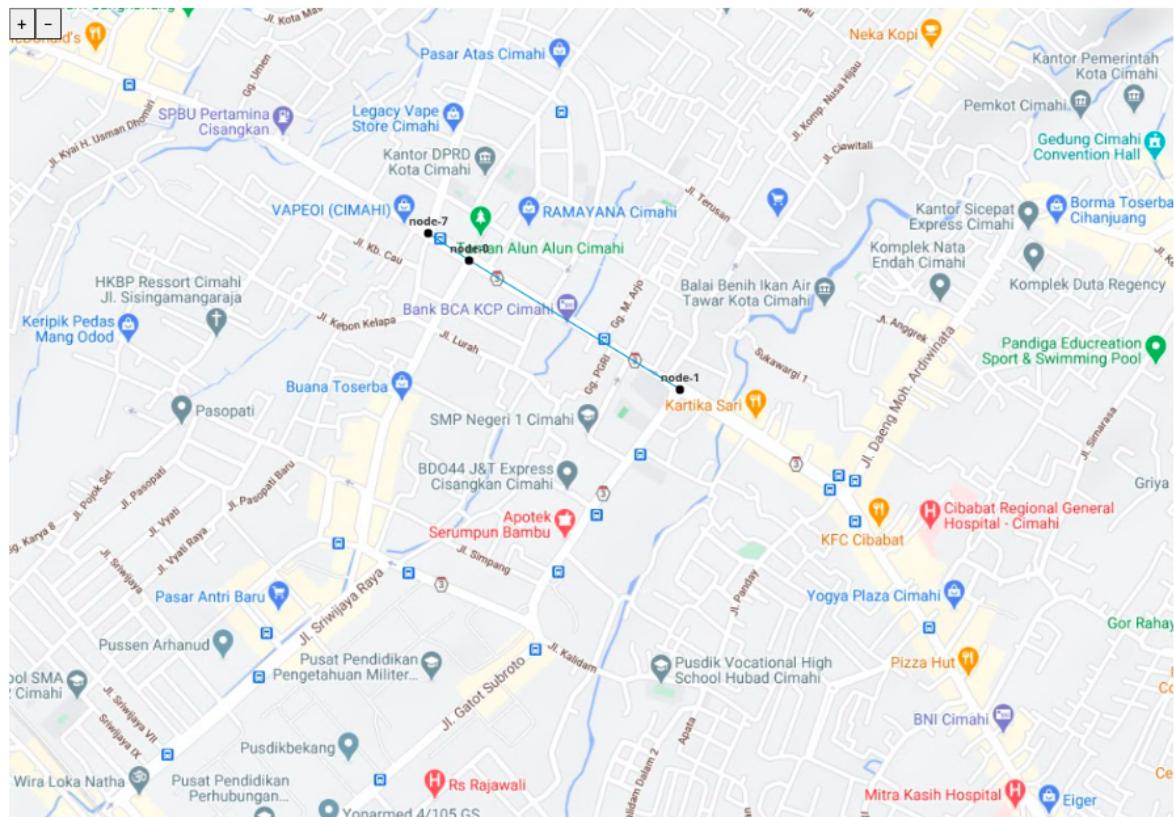
## Adjacency Matrix

## Node List

01. X node-0| 107.65927442576984, -6.942072697375892
02. X node-1| 107.6591411470107, -6.943466952474225
03. X node-2| 107.65992496160887, -6.94358830220726
04. X node-4| 107.65991998646375, -6.943958290525941
05. X node-5| 107.66129776458288, -6.944114057014474
06. X node-6| 107.65945702923541, -6.947042305836376
07. X node-7| 107.65912364028033, -6.946985246981754
08. X node-8| 107.65923285390356, -6.946038068983839
09. X node-9| 107.65737510606941, -6.945889494021216
10. X node-10| 107.65762316599084, -6.943629362080301
11. X node-11| 107.65816118464389, -6.943636215575268
12. X node-12| 107.65818992507104, -6.943390860674555
13. X node-13| 107.65828514036842, -6.941949228399977
14. X node-14| 107.65827347966398, -6.940266763457316
15. X node-15| 107.65948110376118, -6.939967069503822
16. X node-16| 107.66247795316042, -6.942262530791154
17. X node-17| 107.66125520458608, -6.942047988821244

## Path Result

Jarak lintasan: 0.6426716972566198 KM



## Test4

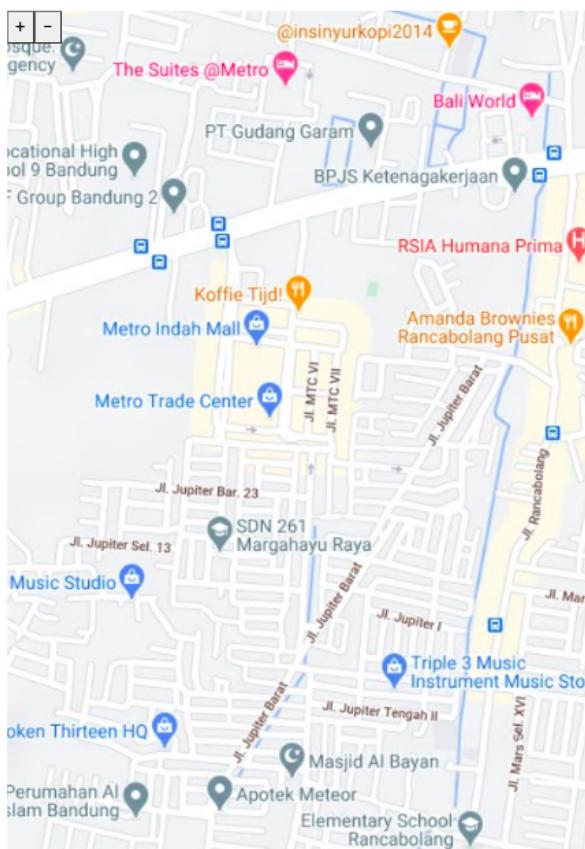
```
nodes = [
{
    "longitude": 115.23385360836984,
    "latitude": -8.673158801771962,
    "name": "Lapangan Puputan"
},
{
    "longitude": 115.24488419294359,
    "latitude": -8.6734677057039,
    "name": "Monumen Bundaran Renon"
},
{
    "longitude": 115.24070531129837,
    "latitude": -8.664531927149241,
    "name": "Klinik Utama Padma Bahtera"
}
```

```
},
{
  "longitude": 115.25957331061363,
  "latitude": -8.674365249163287,
  "name": "Sanur"
},
{
  "longitude": 115.22169515490533,
  "latitude": -8.672562205476154,
  "name": "Florence Bakery"
},
{
  "longitude": 115.22210016846657,
  "latitude": -8.666776498034608,
  "name": "Dinas Pendapatan"
},
{
  "longitude": 115.2385099232197,
  "latitude": -8.667373103516823,
  "name": "Ayam Betutu Bu Nia"
},
{
  "longitude": 115.23861318826678,
  "latitude": -8.664653900630867,
  "name": "Merdeka Medical Center"
},
{
  "longitude": 115.21550461649895,
  "latitude": -8.669486405094972,
  "name": "Kimia Farma"
},
{
  "longitude": 115.21517738699916,
  "latitude": -8.679039864756348,
  "name": "Pertamina"
},
{
  "longitude": 115.21597869694233,
  "latitude": -8.679234086748295,
  "name": "Gang Kuntul"
},
{
  "longitude": 115.22177293896675,
  "latitude": -8.669791334705451,
```

```
        "name": "Raya Puputan"  
    }  
]
```

```
adj = [
    [0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [1, 0, 1, 1, 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, 1, 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, 1],
    [0, 0, 0, 0, 0, 0, 1, 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, 1, 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, 1, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0]
```

]



## Adjacency Matrix

+	01.	02.	03.	04.	05.	06.	07.	08.	09.	10.	11.	12.
01.	0	1	0	0	1	0	0	0	0	0	0	0
02.	1	0	1	1	0	0	0	0	0	0	0	0
03.	0	1	0	0	0	0	0	1	0	0	0	0
04.	0	1	0	0	0	0	0	0	0	0	0	0
05.	1	0	0	0	0	0	0	0	0	0	0	1
06.	0	0	0	0	0	0	1	0	0	0	0	1
07.	0	0	0	0	0	1	0	1	0	0	0	0
08.	0	0	1	0	0	0	1	0	0	0	0	0
09.	0	0	0	0	0	0	0	0	0	1	0	1
10.	0	0	0	0	0	0	0	0	1	0	1	0
11.	0	0	0	0	0	0	0	0	0	1	0	0
12.	0	0	0	0	1	1	0	0	1	0	0	0

## Node List

01. X Lapangan Puputan| 115.23385360836984, -8.673158801771962
02. X Monumen Bundaran Renon| 115.24488419294359, -8.6734677057039
03. X Klinik Utama Padma Bahtera| 115.24070531129837, -8.664531927149241
04. X Sanur| 115.25957331061363, -8.674365249163287
05. X Florence Bakery| 115.22169515490533, -8.672562205476154
06. X Dinas Pendapatan| 115.22210016846657, -8.666776498034608
07. X Ayam Betutu Bu Nia| 115.2385099232197, -8.667373103516823
08. X Merdeka Medical Center| 115.23861318826678, -8.664653900630867
09. X Kimia Farma| 115.21550461649895, -8.669486405094972
10. X Pertamina| 115.21517738699916, -8.679039864756348
11. X Gang Kuntul| 115.21597869694233, -8.679234086748295
12. X Raya Puputan| 115.22177293896675, -8.669791334705451

Submit

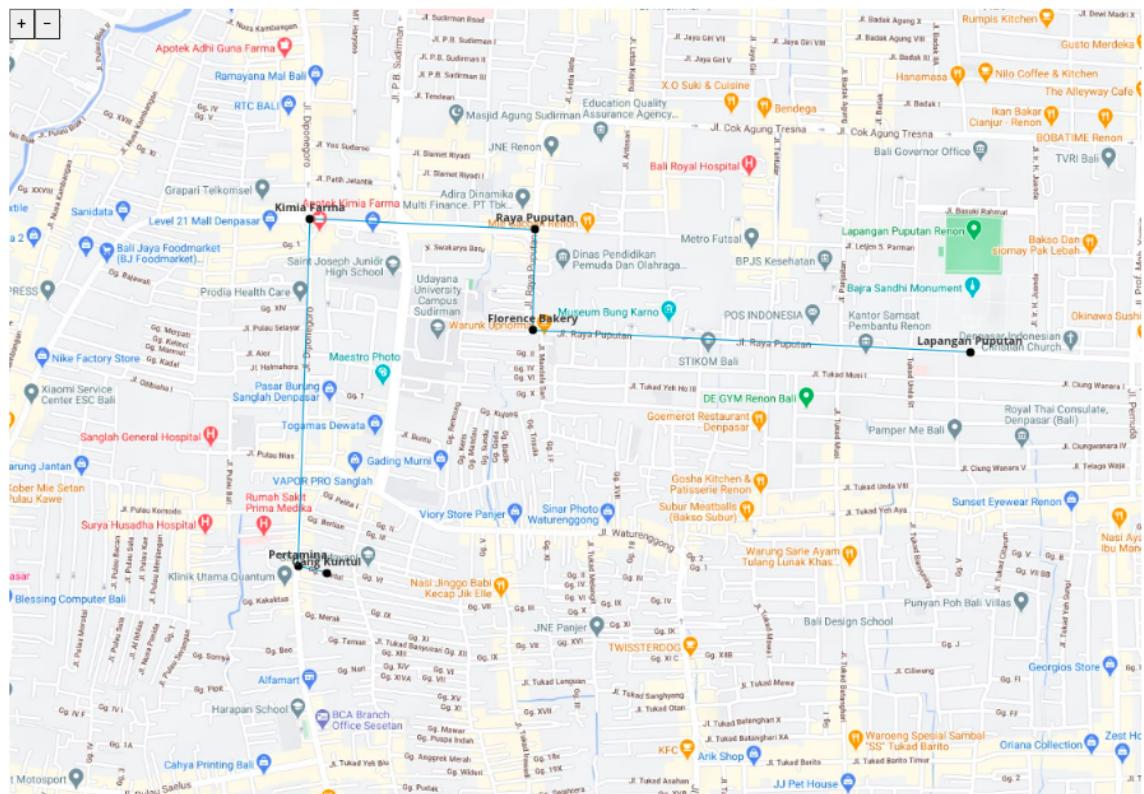
## **Node Start : Lapangan Puputan**

**Node Destination : Gang Kuntul**

## Find Path

## Path Result

Jarak lintasan: 3.489840690857947 KM



## Test5

```
nodes = [
    {
        "longitude": 44.457,
        "latitude": 26.093,
        "name": "Bucharest"
    },
    {
        "longitude": 46.181,
        "latitude": 21.312,
        "name": "Arad"
```

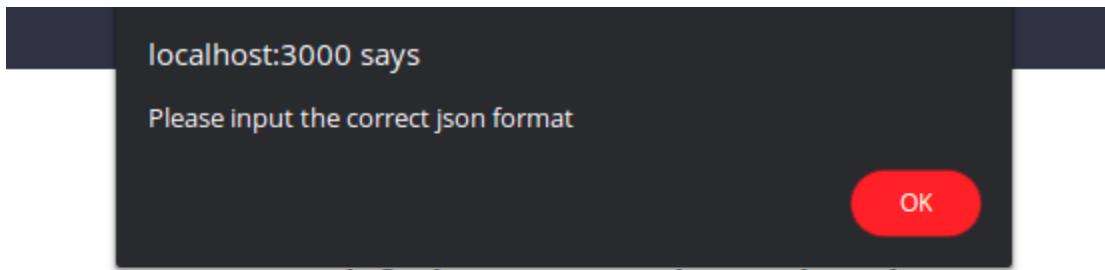
```
},
{
  "longitude": 46.624,
  "latitude": 21.518,
  "name": "Zerind"
},
{
  "longitude": 47.089,
  "latitude": 21.907,
  "name": "Oradea"
},
{
  "longitude": 45.794,
  "latitude": 24.128,
  "name": "Sibiu"
},
{
  "longitude": 45.842,
  "latitude": 24.973,
  "name": "Fagaras"
},
{
  "longitude": 45.756,
  "latitude": 21.231,
  "name": "Timisoara"
},
{
  "longitude": 45.691,
  "latitude": 21.903,
  "name": "Lugoj"
},
{
  "longitude": 44.904,
  "latitude": 22.365,
  "name": "Mehadia"
},
{
  "longitude": 44.639,
  "latitude": 22.659,
  "name": "Drobeta"
},
{
  "longitude": 44.319,
  "latitude": 23.794,
```

```
        "name": "Cralova"
    },
    {
        "longitude": 45.099,
        "latitude": 24.369,
        "name": "Rimnicu Vilcea"
    },
    {
        "longitude": 44.856,
        "latitude": 24.869,
        "name": "Pitesti"
    },
    {
        "longitude": 43.905,
        "latitude": 25.969,
        "name": "Giurgiu"
    },
    {
        "longitude": 44.718,
        "latitude": 26.645,
        "name": "Urziceni"
    },
    {
        "longitude": 47.056,
        "latitude": 26.506,
        "name": "Neamt"
    },
    {
        "longitude": 47.158,
        "latitude": 27.598,
        "name": "Iasi"
    },
    {
        "longitude": 46.641,
        "latitude": 27.728,
        "name": "Vaslui"
    },
    {
        "longitude": 44.69,
        "latitude": 27.945,
        "name": "Hirsova"
    },
    {
        "longitude": 44.049,
```

```
        "latitude": 28.653,
        "name": "Eforie"
    }
]

adj = [
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 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, 1, 0, 1, 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, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 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, 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, 1, 0, 1, 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, 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, 1, 0, 1, 1, 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, 1, 0, 1, 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, 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, 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, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
]
```

## TEST FILE jika format salah



### Route / path finder app using the A\* algorithm

nenerima input dari file ataupun langsung dari map-nya. Anda dapat mengclick it. Anda juga dapat menghapus koordinat yang sudah ambil pada nodes list. A menggunakan adjacency matrix

~~This page was last updated on 2023-09-11.~~ Halaman ini dibuat oleh Nathaniel Iason (12510108) dan Ariva Adinatha (12510015)

#### Test6

```
nodes = [
  {
    "longitude":107.608454,
    "latitude": -6.893874,
    "name":A
  },
  {
    "longitude":107.609908,
    "latitude": -6.893294,
    "name":B
  },
  {
    "longitude":107.610450,
    "latitude": -6.893230,
    "name":C
  },
  {
    "longitude":107.611949,
    "latitude": -6.893592,
    "name":D
  },
  {
    "longitude":107.612972,
    "latitude": -6.893754,
    "name":E
  }
]
```

```

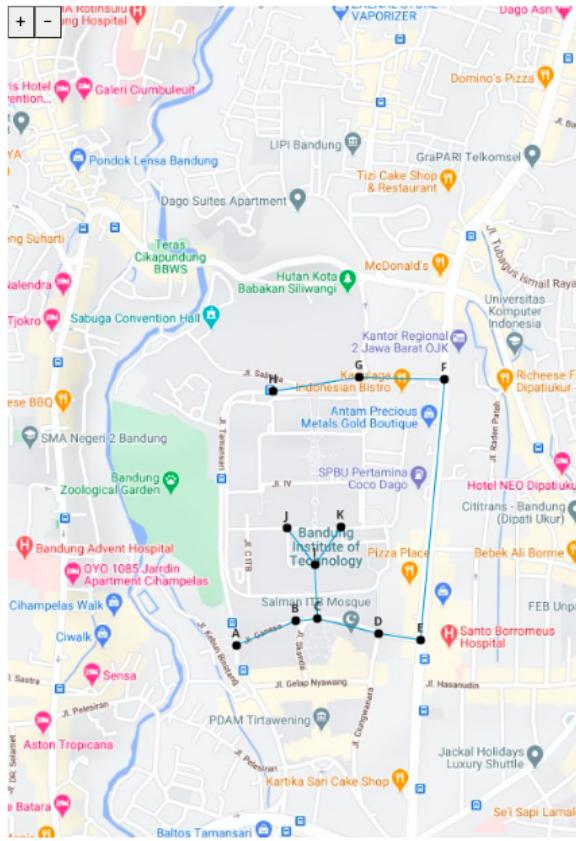
},
{
  "longitude":107.613567,
  "latitude": -6.887412,
  "name":F
},
{
  "longitude":107.611469,
  "latitude": -6.887363,
  "name":G
},
{
  "longitude":107.609365,
  "latitude": -6.887703,
  "name":H
},
{
  "longitude":107.610386,
  "latitude": -6.891923,
  "name":I
},
{
  "longitude":107.609701,
  "latitude": -6.891034,
  "name":J
},
{
  "longitude":107.611022,
  "latitude": -6.891013,
  "name":K
}
]

```

```

adj = [
  [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, 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, 0, 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, 0, 0, 0, 0, 1],
  [0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
]
```

[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]  
]



**Adjacency Matrix**

	01.	02.	03.	04.	05.	06.	07.	08.	09.	10.	11.
01.	0	1	0	0	0	0	0	0	0	0	0
02.	1	0	1	0	0	0	0	0	0	0	0
03.	0	1	0	1	0	0	0	0	1	0	0
04.	0	0	1	0	1	0	0	0	0	0	0
05.	0	0	0	1	0	1	0	0	0	0	0
06.	0	0	0	0	1	0	1	0	0	0	0
07.	0	0	0	0	0	1	0	1	0	0	0
08.	0	0	0	0	0	0	1	0	0	0	0
09.	0	0	1	0	0	0	0	0	0	1	1
10.	0	0	0	0	0	0	0	0	1	0	0
11.	0	0	0	0	0	0	0	0	1	0	0

## Node List

01.	X	A  107.608454, -6.893874
02.	X	B  107.609908, -6.893294
03.	X	C  107.61045, -6.89323
04.	X	D  107.611949, -6.893592
05.	X	E  107.612972, -6.893754
06.	X	F  107.613567, -6.887412
07.	X	G  107.611469, -6.887363
08.	X	H  107.609365, -6.887703
09.	X	I  107.610386, -6.891923
10.	X	J  107.609701, -6.891034
11.	X	K  107.611022, -6.891013

Submit

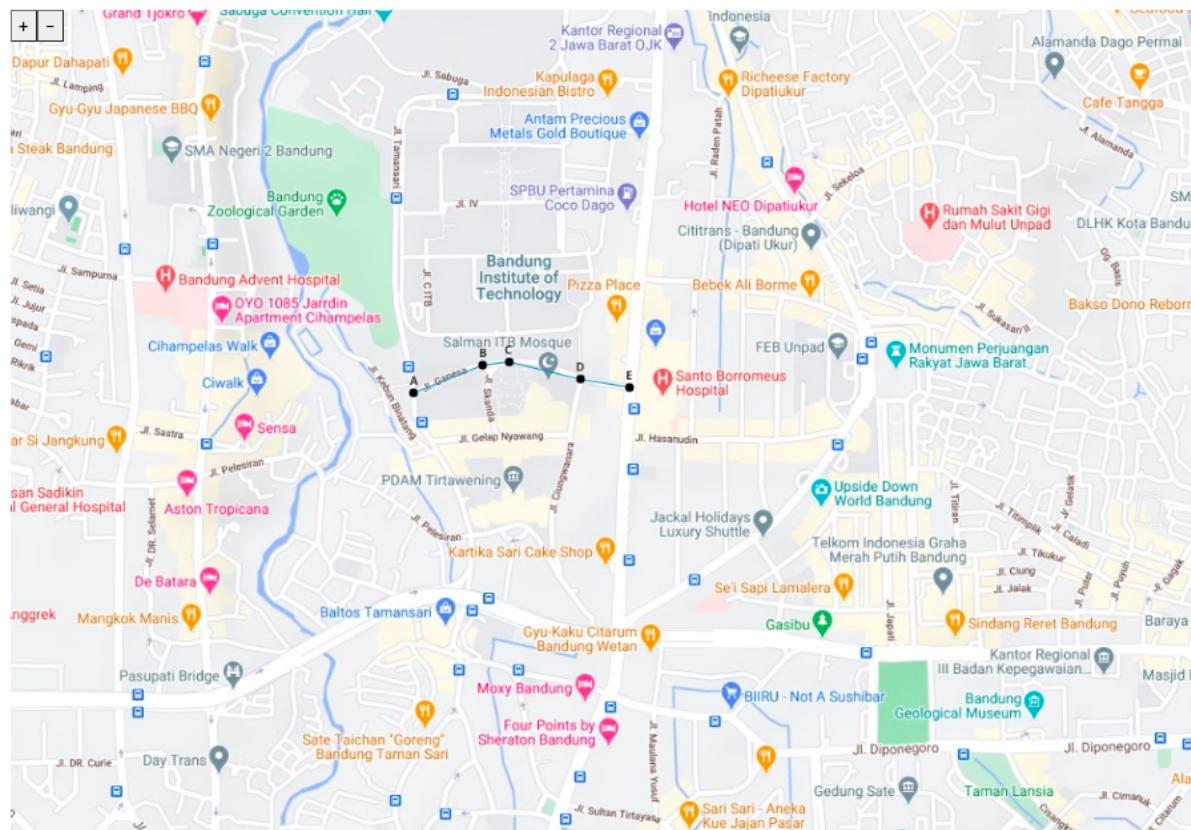
Node Start : A

Node Destination : E

Find Path

### Path Result

Jarak lintasan: 0.5178939864407455 KM



## Screenshot Peta

ITB

## Select Node

**Node Start**

 | ▾

**Node Destination**

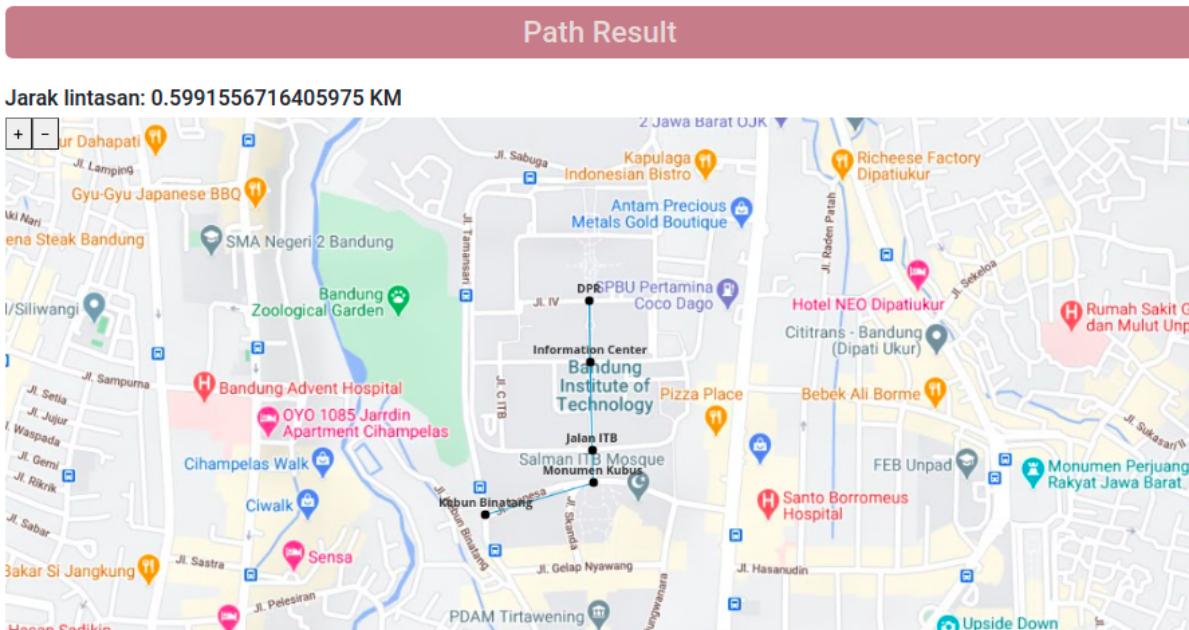
 | ▾

**Submit**

Node Start : Kebun Binatang

Node Destination : DPR

**Find Path**



## Alun Alun Kota Bandung

Select Node

Node Start

node-1

Node Destination

node-7

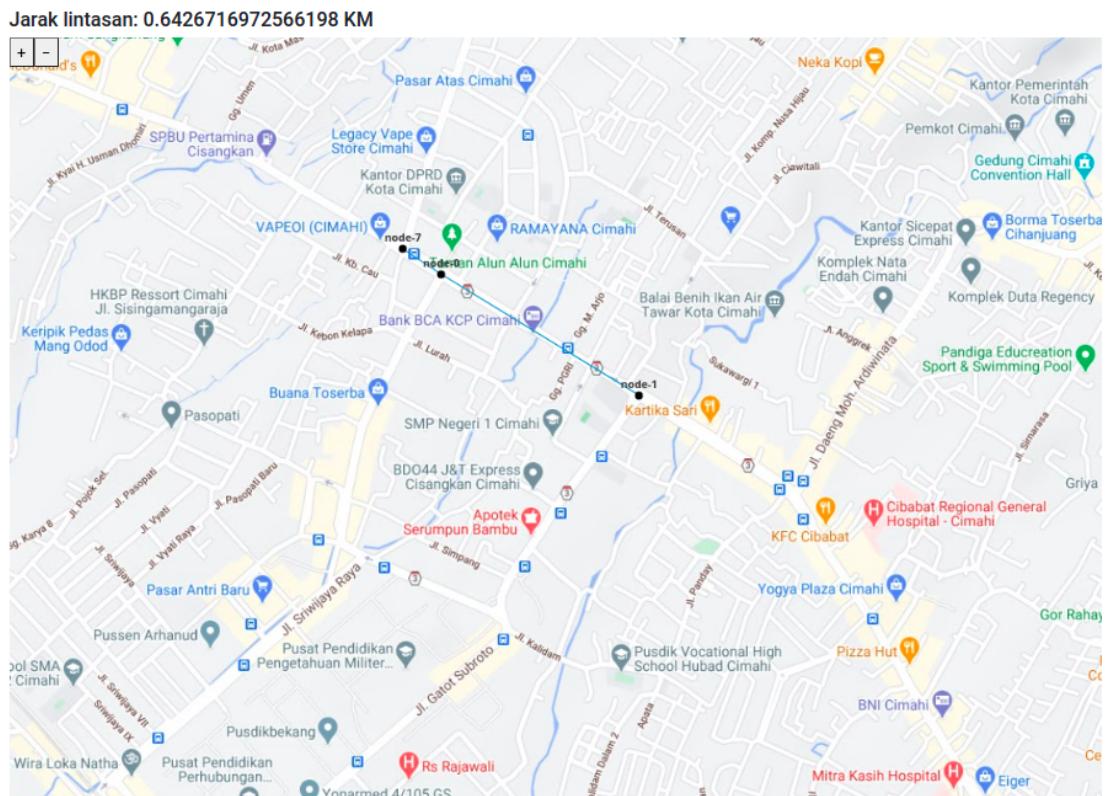
Submit

Node Start : node-1

Node Destination : node-7

Find Path

Path Result



## Buah

**Submit**

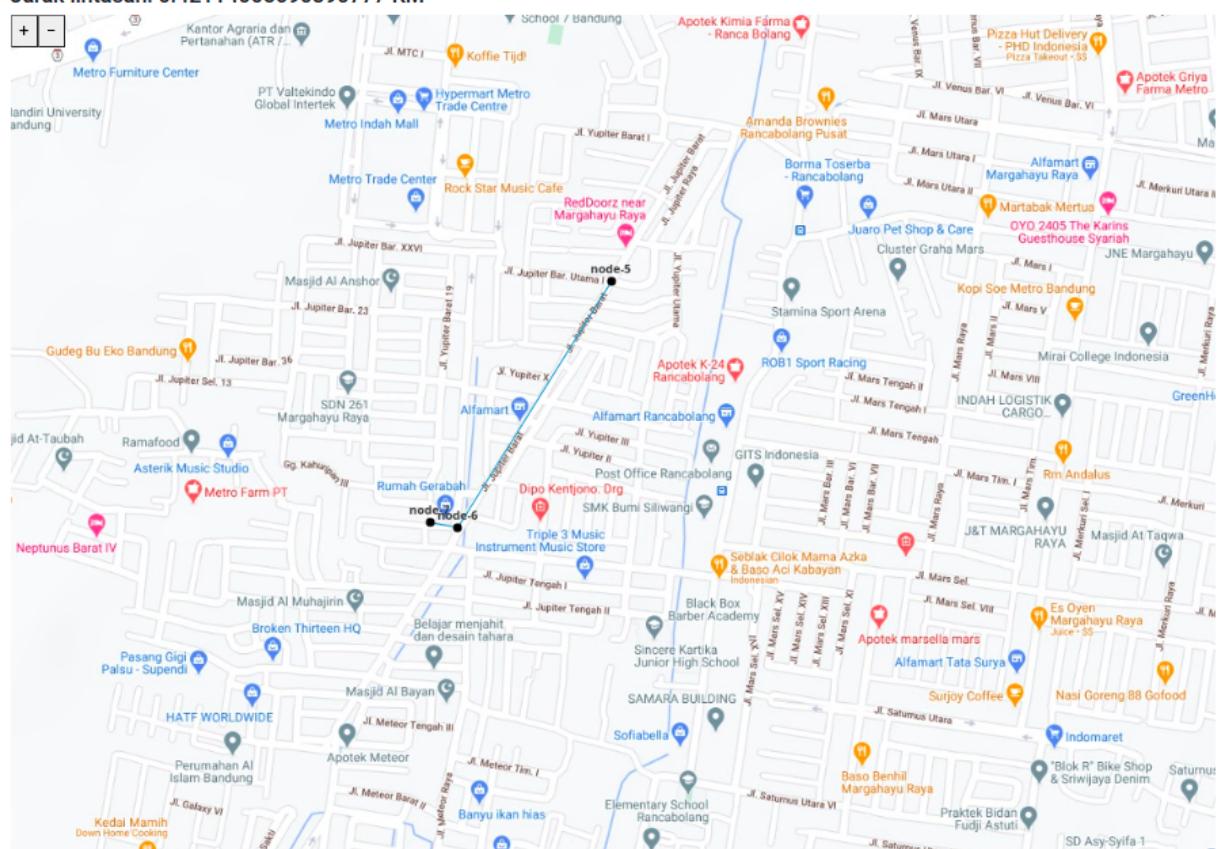
**Node Start : node-5**

**Node Destination : node-7**

## Find Path

## Path Result

Jarak lintasan: 0.42114005690890777 KM



## Bali

Submit

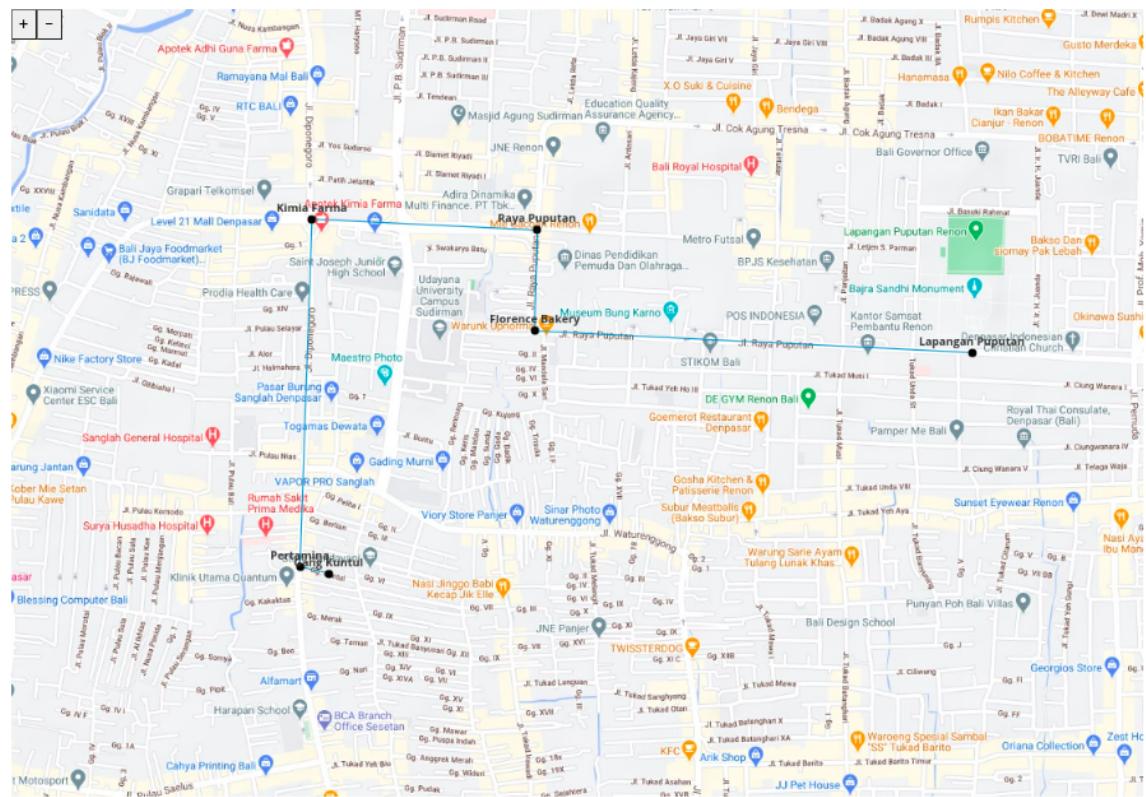
Node Start : Lapangan Puputan

Node Destination : Gang Kuntul

Find Path

Path Result

Jarak lintasan: 3.489840690857947 KM



## ITB 2

Submit

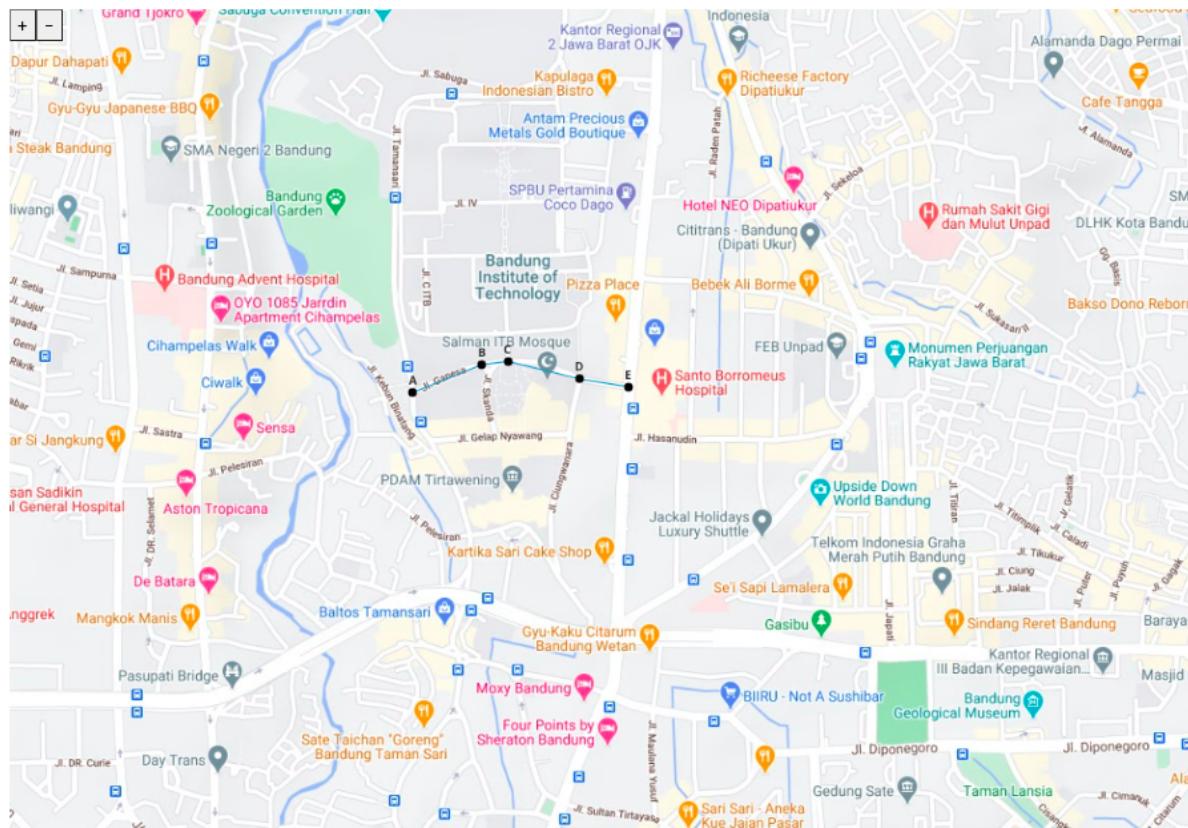
Node Start : A

Node Destination : E

Find Path

Path Result

Jarak lintasan: 0.5178939864407455 KM



**Tabel**

1	Program dapat menerima input graf	centang
2	Program dapat menghitung lintasan terpendek	centang
3	Program dapat menampilkan lintasan terpendek serta jaraknya	centang
4	Bonus: Program dapat menerima input peta dengan Google Map API dan menampilkan peta	centang

Link Source Code :

<https://github.com/ariyaadinatha/route-finder>