

Laporan Tugas Kecil 3
IF 2211 Strategi Algoritma



Pencarian Lintasan Terpendek dengan Algoritma A*

Daniel Mario Reynaldi 13519031

Imam Nurul Hukmi 13519150

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Kode Program

Implementasi Priority Queue dan Graph

```
class PrioQueue():
    def __init__(self):
        self.queue = []

    def isEmpty(self):
        return len(self.queue) == 0

    def put(self, data):
        self.queue.append(data)

    def pop(self):
        minIdx = 0
        for i in range(len(self.queue)):
            if (self.queue[i][1] < self.queue[minIdx][1]):
                minIdx = i

        temp = self.queue[minIdx]
        del self.queue[minIdx]
        return temp
```

```
class Node:
    name: str
    lng: float
    lat: float

    def __init__(self, name, lng, lat):
        self.name = name
        self.lng = lng
        self.lat = lat

    def distance(self, other):
        R = 6371
        dLat = math.radians(float(self.lat)-float(other.lat))
        dLong = math.radians(float(self.lng)-float(other.lng))
        a = (
            math.sin(dLat/2) * math.sin(dLat/2) +
            math.cos(math.radians(self.lat)) * math.cos(math.radians(other.lat)) *
            math.sin(dLong/2) * math.sin(dLong/2)
        )
        c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))
        d = R * c
        return d
```

```
class Graph:
    def __init__(self, nodes, adjMatrix):
        self.nodes = nodes
        self.adjList = {}
        for i in range(len(nodes)):
            self.adjList[nodes[i]] = []
            for j in range(len(adjMatrix[0])):
                if(adjMatrix[i][j] != 0):
                    current = (nodes[j], adjMatrix[i][j])
                    self.adjList[nodes[i]].append(current)

    def getNeighbour(self, node):
        neighbour = []
        for i in range(len(self.adjList[node])):
            neighbour.append(self.adjList[node][i][0])
        return neighbour

    def getWeight(self, node1, node2):
        neighbour = self.adjList[node1]
        for i in range(len(neighbour)):
            if (neighbour[i][0] == node2):
                return neighbour[i][1]
        return 0
```

Implementasi Algoritma A*

```
def shortestPath(self, startNode, endNode):
    open = PrioQueue()
    open.put((startNode, 0))
    prevNode = {}
    costArr = {}
    prevNode[startNode] = None
    costArr[startNode] = 0

    while not (open.isEmpty()):
        top = open.pop()
        top_node = top[0]

        if (top_node == endNode):
            break

        for node in self.getNeighbour(top_node):
            cost = costArr[top_node] + self.getWeight(top_node, node)
            if (node not in costArr) or (cost < costArr[node]):
                costArr[node] = cost
                cost += node.distance(endNode)
                open.put((node, cost))
                prevNode[node] = top_node

    return prevNode, costArr

def tracePath(self, trace, startNode, endNode):
    path = []
    current = endNode
    while (current != startNode):
        path.append(current)
        current = trace[current]
    path.append(startNode)
    path.reverse()

    return path
```

Pengujian

↳ Masukkan alamat file: /content/ITB.txt

1. ITB
2. SBM
3. Salman
4. Borromeus
5. Teuku_Umar
6. BonBin
7. MCD
8. Dipati_Ukur
9. Dago
10. Siliwangi
11. Tamansari

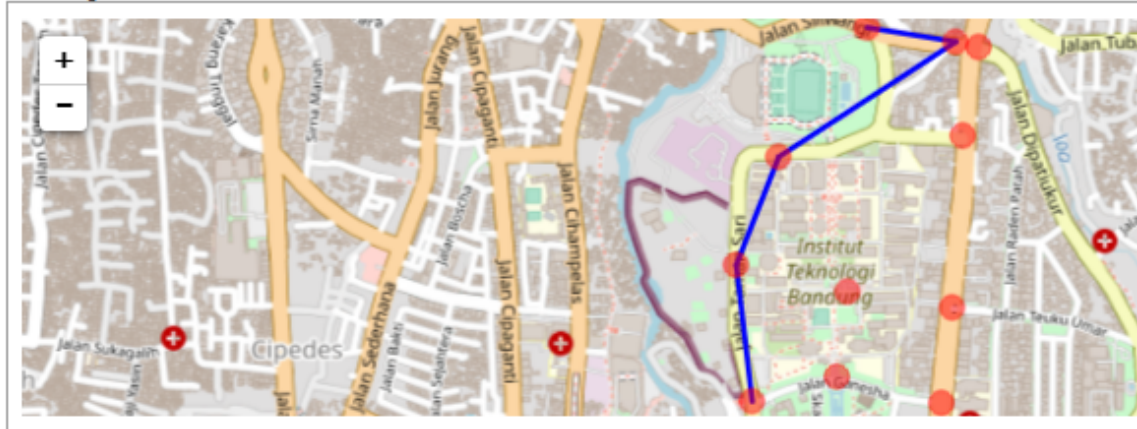
Pilih simpul asal: 11

Pilih simpul tujuan: 10

Rute: Tamansari -> BonBin -> SBM -> MCD -> Siliwangi

(Siliwangi : simpul ujung atas, Tamansari : simpul ujung bawah)

Total jarak = 987.85 m



Masukkan alamat file: /content/alun_alun.txt

1. Museum
2. BRI
3. Alun-Alun
4. Masjid-Raya
5. Cikapundung
6. Jend-Sudirman
7. Braga
8. 0KM

Pilih simpul asal: 8

Pilih simpul tujuan: 2

Rute: 0KM -> Museum -> BRI

(BRI : simpul ujung atas, 0KM : simpul ujung bawah)

Total jarak = 382.39 m



Masukkan alamat file: /content/bogor.txt

1. Botani
2. Lippo
3. Sempur
4. SMA-Regina-Pacis
5. Katedral
6. BTM
7. Surken
8. MCD
9. Kebun Raya

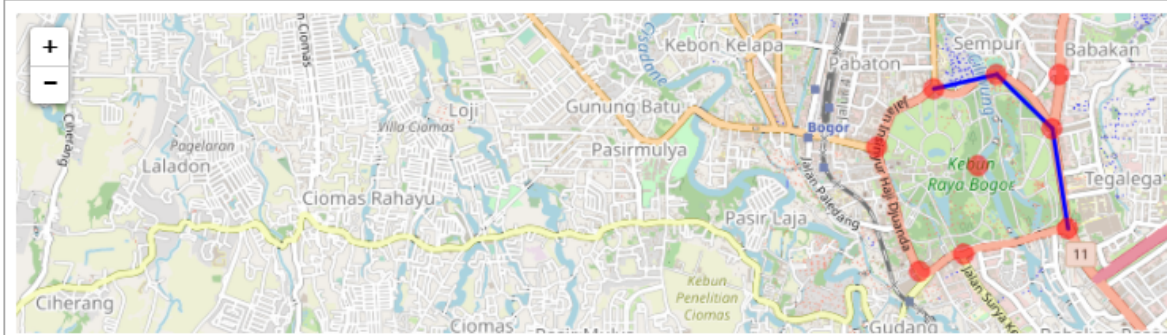
Pilih simpul asal: 1

Pilih simpul tujuan: 4

Rute: Botani -> Lippo -> Sempur -> SMA-Regina-Pacis

(SMA-Regina-Pacis : simpul ujung atas, Botani : simpul ujung bawah)

Total jarak = 1030.15 m

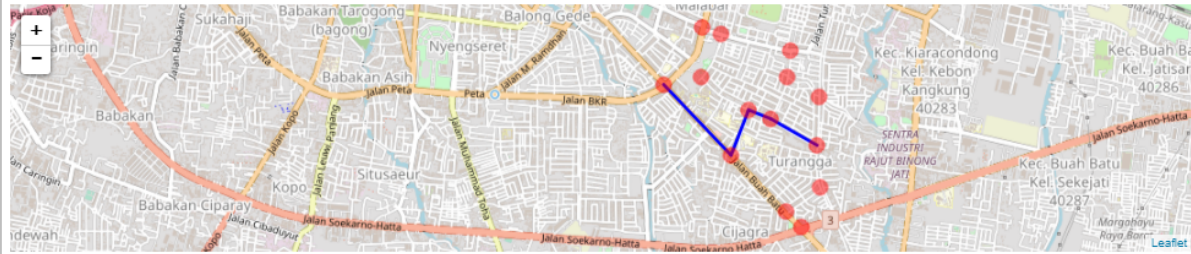


Masukkan alamat file: /content/buahbatu.txt

1. BKR-Buah
2. Pelajar-Marta
3. Masku-Marta
4. Masku-Kara
5. Kara-Klini
6. Klini-Buah
7. Buah-Guntur
8. Buah-Soek
9. Guntur
10. Guntur-Sale
11. Sale
12. Sale-Reog
13. Reog-Marta
14. Reog-Kara

Pilih simpul asal: 1
Pilih simpul tujuan: 10

Rute: BKR-Buah -> Klini-Buah -> Kara-Klini -> Reog-Kara -> Guntur-Sale
(BKR-Buah : simpul ujung atas, Guntur-Sale : simpul ujung bawah)
Total jarak = 1904.3 m




Masukkan alamat file: /content/test2.txt

1. Kue
2. Kupul
3. Gae
4. Mine
5. Mate
6. Bise
7. Ebon
8. Bisb
9. Bismat
10. Matpul
11. Mato
12. Minpul
13. Garpul
14. Kenar

Pilih simpul asal: 2
Pilih simpul tujuan: 6

Rute: Kupul -> Garpul -> Minpul -> Matpul -> Bismat -> Bisb -> Bise
(Bise : simpul ujung atas, Kupul : simpul ujung bawah)
Total jarak = 125.27 m



Masukkan alamat file: /content/IPB.txt

1. Kamper-Meranti
2. Kamper
3. Kamper-Ulin
4. Bunder4
5. Tj-Huni
6. Huni-Ulin
7. Ulin
8. Tj-Ulin-Meranti
9. Meranti

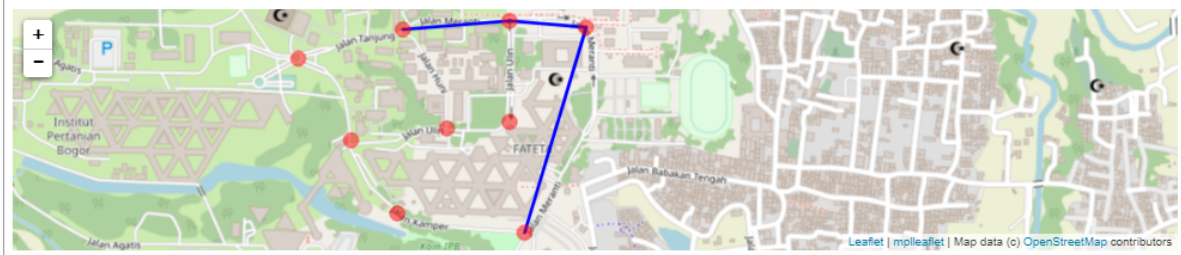
Pilih simpul asal: 1

Pilih simpul tujuan: 5

Rute: Kamper-Meranti -> Meranti -> Tj-Ulin-Meranti -> Tj-Huni

(Tj-Huni : simpul ujung atas, Kamper-Meranti : simpul ujung bawah)

Total jarak = 877.29 m



GitHub: <https://github.com/DanielMariooR/Doogle-Map>