

**Laporan Tugas Kecil 3 IF2211 Strategi Algoritma**

**Semester II tahun 2022/2023**

**Implementasi Algoritma UCS dan A\* untuk Menentukan Lintasan Terpendek**



Disusun oleh:

Louis Caesa Kesuma

13521069

Dewana Gustavus Haraka Otang

13521173

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2023**

# **Daftar Isi**

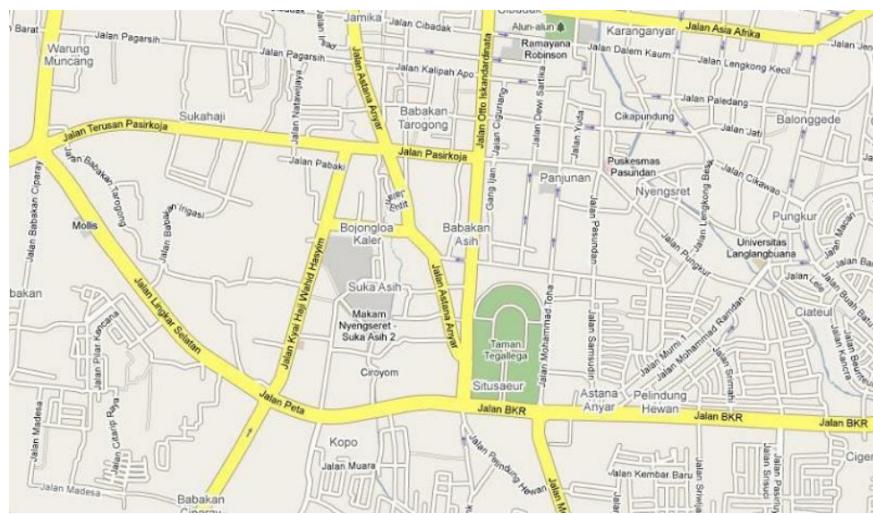
<b>Daftar Isi</b>	<b>2</b>
<b>Bab 1</b>	<b>3</b>
<b>    Deskripsi Persoalan</b>	<b>3</b>
1.1. Deskripsi Tugas	3
1.2. Spesifikasi GUI	4
1.3. Spesifikasi Wajib	5
<b>Bab 2</b>	<b>6</b>
<b>    Landasan Teori</b>	<b>6</b>
2.1. Dasar Teori	6
2.1.1. Uniform Cost Search (UCS)	6
2.1.2. A*	7
<b>Bab 3</b>	<b>8</b>
<b>    Implementasi Program</b>	<b>8</b>
3.1. Implementasi GUI	8
3.1.1. index.html	8
3.1.2. indexmap.html	8
3.1.3. styles.css	8
3.1.4. mapstyles.css	8
3.2. Implementasi Algoritma	8
3.2.1. button.js	8
3.2.2. map.js	8
3.2.3. path.js	8
3.2.4. queue.js	9
3.2.5. util.js	9
3.2.6. visualize.js	9
<b>Bab 4</b>	<b>10</b>
<b>    Eksperimen</b>	<b>10</b>
4.1. Eksperimen Graf	10
4.2. Eksperimen Map	10
<b>Bab 5</b>	<b>12</b>
<b>    Kesimpulan dan Saran</b>	<b>12</b>
5.1. Kesimpulan	12
5.2. Saran	12
5.3. Refleksi	12
5.4. Tanggapan	12
<b>Lampiran</b>	<b>13</b>
<b>Daftar Pustaka</b>	<b>14</b>

## Bab 1

## Deskripsi Persoalan

## 1.1. Deskripsi Tugas

Algoritma UCS (Uniform cost search) dan A\* (atau A star) dapat digunakan untuk menentukan lintasan terpendek dari suatu titik ke titik lain. Pada tugas kecil 3 ini, anda diminta menentukan lintasan terpendek berdasarkan peta Google Map jalan-jalan di kota Bandung. Dari ruas-ruas jalan di peta dibentuk graf. Simpul menyatakan persilangan jalan (simpang 3, 4 atau 5) atau ujung jalan. Asumsikan jalan dapat dilalui dari dua arah. Bobot graf menyatakan jarak (m atau km) antar simpul. Jarak antar dua simpul dapat dihitung dari koordinat kedua simpul menggunakan rumus jarak Euclidean (berdasarkan koordinat) atau dapat menggunakan ruler di Google Map, atau cara lainnya yang disediakan oleh Google Map.



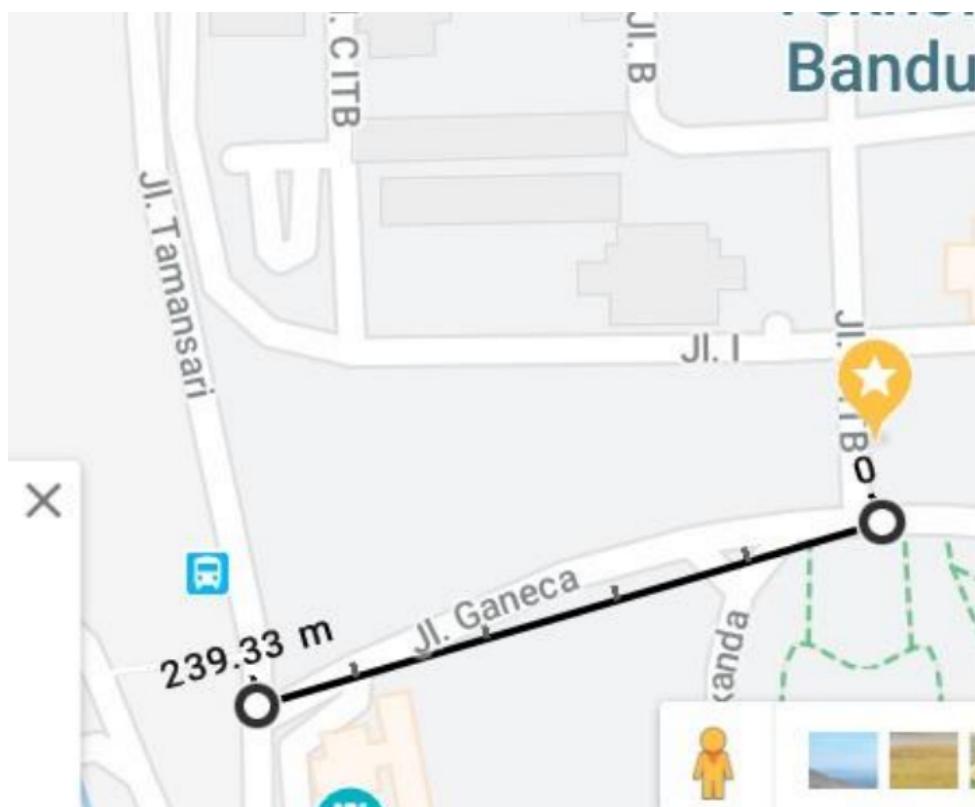
*Gambar 1.1.1 Ilustrasi Map pada Google API*

Langkah pertama di dalam program ini adalah membuat graf yang merepresentasikan peta (di area tertentu, misalnya di sekitar Bandung Utara/Dago). Berdasarkan graf yang dibentuk, lalu program menerima input simpul asal dan simpul tujuan, lalu menentukan lintasan terpendek antara keduanya menggunakan algoritma UCS dan A\*. Lintasan terpendek dapat ditampilkan pada peta/graf (misalnya jalan-jalan

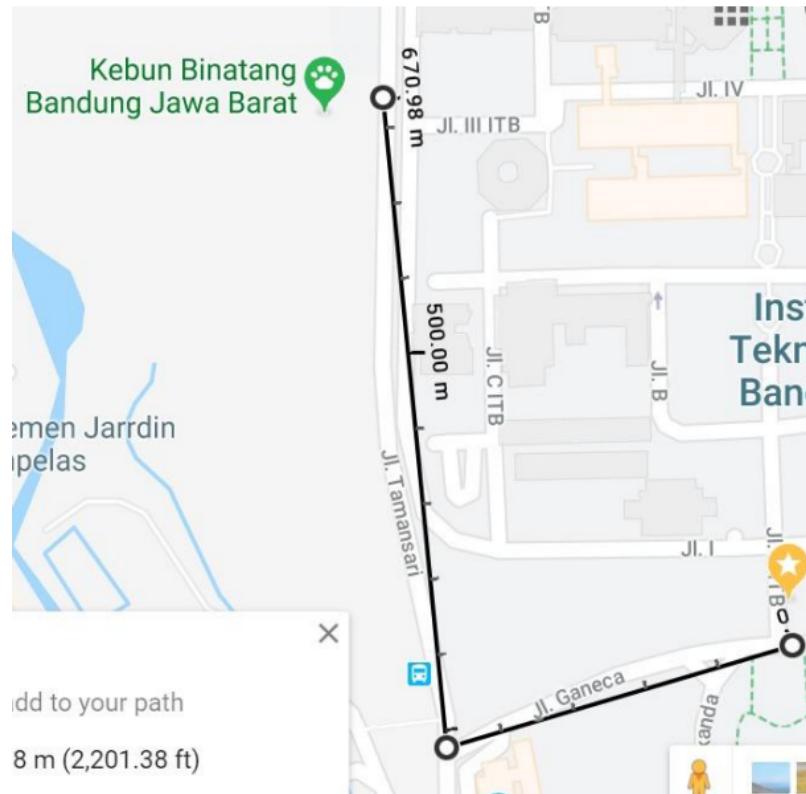
yang menyatakan lintasan terpendek diberi warna merah). Nilai heuristik yang dipakai adalah jarak garis lurus dari suatu titik ke tujuan.

## 1.2. Spesifikasi GUI

Bonus nilai diberikan jika dapat menggunakan Google Map API untuk menampilkan peta, membentuk graf dari peta, dan menampilkan lintasan terpendek di peta (berupa jalan yang diberi warna). Simpul graf diperoleh dari peta (menggunakan API Google Map) dengan mengklik ujung jalan atau persimpangan jalan, lalu jarak antara kedua simpul dihitung langsung dengan rumus Euclidean.



Gambar 1.2.1 Contoh Menghitung Jarak dengan Ruler di Google Map



*Gambar 1.2.2 Contoh output program untuk Gambar 1*

**Catatan:** Tampilan diatas hanya berupa contoh yang disediakan pada spesifikasi tugas saja, tampilan yang diimplementasikan bisa jadi berbeda dengan yang ada di spek namun fungisionalitasnya masih sama.

### 1.3. Spesifikasi Wajib

Spesifikasi program:

1. Program menerima input file graf (direpresentasikan sebagai matriks ketetanggaan berbobot), jumlah simpul minimal 8 buah.
2. Program dapat menampilkan peta/graf
3. Program menerima input simpul asal dan simpul tujuan.
4. Program dapat menampilkan lintasan terpendek beserta jaraknya antara simpul asal dan simpul tujuan.
5. Antarmuka program bebas, apakah pakai GUI atau command line saja.

## Bab 2

### Landasan Teori

#### 2.1. Dasar Teori

##### 2.1.1. Uniform Cost Search (UCS)

UCS adalah salah satu dari banyak algoritma pencarian rute yang populer. UCS sangatlah populer dalam mencari solusi rute jika data yang ada tidak memiliki informasi apapun (melakukan penjelajahan secara *uninformed*). UCS adalah salah satu variasi dari algoritma Dijkstra. Implementasi UCS menggunakan *priority queue* yang dimana prioritasnya ditentukan oleh jarak yang ditempuh. *Priority queue* tersebut akan terus-terusan ditambahkan dengan simpul yang bisa diakses dari simpul aktif. Tentunya simpul-simpul tersebut akan ditambahkan ke dalam *priority queue* sesuai dengan jarak yang telah ditempuh ketika menuju simpul tersebut. Algoritma UCS akan terus mengulangi langkah-langkah tersebut hingga simpul tujuan telah dicapai.

Fungsi evaluasi dari UCS adalah sebagai berikut:

$$f(n) = g(n)$$

Dimana  $f(n)$  adalah estimasi total jarak yang ditempuh dari simpul sekarang ke simpul tujuan, dan  $g(n)$  adalah total jarak yang telah ditempuh dari simpul awal ke simpul sekarang.

##### 2.1.2. A\*

A\* adalah salah satu jenis algoritma pencarian rute terpendek antara 2 buah titik di dalam sebuah grafik, baik terarah maupun tidak terarah. Algoritma A\* menggunakan heuristik untuk mempercepat pencarian jalur terpendek yang ada di dalam grafik tersebut. Pada setiap langkah pencarian, algoritma A\* mengevaluasi setiap simpul di sekitar simpul yang sedang dieksplorasi dan memperkirakan jarak yang tersisa hingga mencapai titik tujuan.

Fungsi evaluasi dari A\* adalah sebagai berikut:

$$f(n) = g(n) + h(n)$$

Dimana  $f(n)$  adalah estimasi total jarak yang ditempuh dari simpul sekarang ke simpul tujuan,  $g(n)$  adalah total jarak yang telah ditempuh dari simpul awal ke simpul sekarang, dan  $h(n)$  adalah estimasi total jarak yang akan ditempuh dari simpul sekarang ke simpul tujuan.

## Bab 3

### Implementasi Program

#### 3.1. Implementasi GUI

##### 3.1.1. index.html

```
<!DOCTYPE html>
<meta charset="utf-8" />
<link rel="stylesheet" href="styles.css">

<head>
    <h1> Tucil 3 - Tanpa Maps</h1>
    <script src="https://d3js.org/d3.v4.min.js"></script>
    <script src = "util.js" defer></script>
    <script src = "queue.js" defer></script>
    <script src = "path.js" defer></script>
    <script src = "visualize.js" defer></script>
    <script src = "button.js" defer></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Miriam+Libre:wght@700&family=Source+Sans+Pro&display=swap" rel="stylesheet">
</head>

<body>
    <div class=" sidenav">
        <div class="inlinetoggle">
            <div class="text">Pakai maps?</div>
            <label class="switch">
                <input type="checkbox" id="gotomaps" onclick="return gotomaps();">
                <span class="slider round"></span>
            </label>
        </div>
    </div>

    <div class="sidenav2">
        <div class="getfilebutton">
            <input type="button" id="get_file" value="Masukkan file">
            <input type="file" id="my_file">
        </div>

        <div class="inputform">
            <input type="text" id="start" placeholder="Start Node" class="inputspace"><br><br>
            <input type="text" id="end" placeholder="Destination Node" class="inputspace"><br><br>
        </div>
    </div>
</body>
```

```

        </div>

        <div class="algoritma" id="algoritma">
            <div class="radiobutton">
                <input type="radio" id="UCS" name="algo_choice" value="UCS"
onclick="using('UCS')">
                <label for="UCS">UCS</label><br>
            </div>
            <div class="radiobutton">
                <input type="radio" id="A*" name="algo_choice" value="A*"
onclick="using('A*')">
                <label for="A*">>A*</label><br>
            </div>
        </div>

        <div class=" sidenav3">
            <div id="output-section">
                The cost is:<br> <span id="output-jarak"></span>
            </div>
        </div>
        <svg width="1400" height="800"></svg>
</body>

```

### 3.1.2. indexmap.html

```

<!DOCTYPE html>
<meta charset="utf-8" />
<link rel="stylesheet" href="mapstyles.css">

<head>
    <h1> Tucil 3 - Dengan Maps</h1>
    <script src="https://d3js.org/d3.v4.min.js"></script>
    <script src = "util.js" defer></script>
    <script src = "queue.js" defer></script>
    <script src = "path.js" defer></script>
    <script src = "button.js" defer></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
    href="https://fonts.googleapis.com/css2?family=Miriam+Libre:wght@700&family=So
    urce+Sans+Pro&display=swap" rel="stylesheet">
    <link rel="stylesheet"
    href="https://unpkg.com/leaflet@1.8.0/dist/leaflet.css" />
    <link rel="stylesheet"
    href="https://unpkg.com/leaflet-routing-machine@latest/dist/leaflet-routing-ma
    chine.css" />
</head>

<body>
    <div class="sidenav">
        <div class="inlinetoggle">

```

```

        <div class="text">Pakai maps?</div>
        <label class="switch">
            <input type="checkbox" id="gotowithoutmaps" onclick="return
gotowithoutmaps();">
            <span class="slider2 round"></span>
        </label>
    </div>
</div>

<div class="sidenav2" >
    <div class="judul">
        Masukkan Edge
    </div>
    <br>
    <div class="inputform">
        <input type="text" id="source" placeholder="From"
class="inputspace"><br><br>
        <input type="text" id="target" placeholder="To"
class="inputspace"><br><br>
        <button class="submitbutton" onclick="addEdge () ">Tambah</button>
    </div>
</div>

<div class="sidenav3">
    <div class="inputform">
        <input type="text" id="start" placeholder="Start Node"
class="inputspace"><br><br>
        <input type="text" id="end" placeholder="Destination Node"
class="inputspace"><br><br>
    </div>

    <div class="algoritma" id="algoritma">
        <div class="radiobutton">
            <input type="radio" id="UCS" name="algo_choice" value="UCS"
onclick="using('UCS')">
            <label for="UCS">UCS</label><br>
        </div>
        <div class="radiobutton">
            <input type="radio" id="A*" name="algo_choice" value="A*"
onclick="using('A*')">
            <label for="A*">>A*</label><br>
        </div>
    </div>
</div>

<div class="sidenav4">
    <div id="output-section">
        The cost is:<br> <span id="output-jarak"></span>
    </div>
</div>

<div id="map"></div>
<script src="https://unpkg.com/leaflet@1.8.0/dist/leaflet.js"></script>
<script>

```

```

src="https://unpkg.com/leaflet-routing-machine@latest/dist/leaflet-routing-mac
hine.js"></script>
<script src="map.js"></script>
</body>

```

### 3.1.3. styles.css

```

/* style buat body */
body {
    background-color: #15181f;
    background-image: linear-gradient(to bottom right, #212429, #20232a);
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
}

/* style buat header 1 */
h1 {
    text-align: center ;
    color: white;
    position: relative;
    padding-bottom: 25px;
}

/* style buat svg */
svg {
    background-color: #2b2b33;
    border-style: solid;
    border-width: 0px;
    border-radius: 25px;
    position: relative;
    left: 19%;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    background-image: linear-gradient(to bottom right, #292a2c, #23262d);
}

/* buat nodes */
.circle {
    fill: black;
}

.links line {
    stroke-opacity: 0.6;
}

.nodes circle {
    stroke: #fff;
    stroke-width: 3px;
}

.node-label {
    font-size: 12px;
    font-weight: bold;
}

```

```

        fill: white;
    }

marker {
    opacity: 0.6;
}

.link-label {
    opacity: 0.6;
    font-size: 12px;
    fill: white;
}

/* input file */
#my_file {
    display: none;
}

#get_file {
    background: #f9f9f9;
    width: 120px;
    border: beige;
    padding: 15px;
    border-radius: 5px;
    margin: 10px;
    cursor: pointer;
    position: relative;
    left: 23%;
    background-image: linear-gradient(to right, #f904b4, #7335e7);
    margin-bottom: 70px;
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
    font-size: 20;
    font-weight: 500;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    color: #e1e2fa;
}

/* side navbar untuk toggle */
.sidenav {
    height: 60px;
    width: 250px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #292a2f;
    overflow-x: hidden;
    overflow-y: hidden;
    padding-top: 20px;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    background-image: linear-gradient(to bottom right, #292a2c, #23262d);
    margin-top: 170px;
    margin-left: 30px;
}

```

```

        border-style: solid;
        border-width: 0px;
        border-radius: 25px;
    }

/* side navbar untuk input file dan node */
.sidenav2 {
    height: 360px;
    width: 250px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #292a2f;
    overflow-x: hidden;
    overflow-y: hidden;
    padding-top: 20px;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    background-image: linear-gradient(to bottom right, #292a2c, #23262d);
    margin-top: 310px;
    margin-left: 30px;
    border-style: solid;
    border-width: 0px;
    border-radius: 25px;
}

/* untuk output jarak */
.sidenav3 {
    height: 60px;
    width: 250px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #292a2f;
    overflow-x: hidden;
    overflow-y: hidden;
    padding-top: 20px;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    background-image: linear-gradient(to bottom right, #292a2c, #23262d);
    margin-top: 750px;
    margin-left: 30px;
    border-style: solid;
    border-width: 0px;
    border-radius: 25px;
    color: white;
    text-align: center;
}

/* toggle pakai maps */
.inlinetoggle {
    width: 150px;
    position: relative;
    left: 10%;
}

```

```
height: 70px;

}

.text {
    text-align: center;
    vertical-align: middle;
    color: white;
    position: relative;
    left: 10%;
    margin-top: 10px;
}

.switch {
    margin-bottom: 100px;
    position: relative;
    display: inline-block;
    width: 60px;
    height: 34px;
    left: 20%;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
}

.text, .switch {
    display: inline-block;
}

.switch input {
    opacity: 0;
    width: 0;
    height: 0;
}

.slider {
    position: absolute;
    cursor: pointer;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: #ccc;
    -webkit-transition: .4s;
    transition: .4s;
}

.slider:before {
    position: absolute;
    content: "";
    height: 26px;
    width: 26px;
    left: 4px;
    bottom: 4px;
    background-color: white;
    -webkit-transition: .4s;
}
```

```

        transition: .4s;
    }

input:checked + .slider {
    background-image: linear-gradient(to right, #f904b4, #7335e7);
}

input:checked + .slider:before {
    -webkit-transform: translateX(26px);
    -ms-transform: translateX(26px);
    transform: translateX(26px);
}

.slider.round {
    border-radius: 34px;
}

.slider.round:before {
    border-radius: 50%;
}

.slider2 {
    position: absolute;
    cursor: pointer;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-image: linear-gradient(to right, #f904b4, #7335e7);
    -webkit-transition: .4s;
    transition: .4s;
}

.slider2:before {
    position: absolute;
    content: "";
    height: 26px;
    width: 26px;
    left: 30px;
    bottom: 4px;
    background-color: white;
    -webkit-transition: .4s;
    transition: .4s;
}

input:checked + .slider2 {
    -webkit-transform: translateX(26px);
    -ms-transform: translateX(26px);
    transform: translateX(26px);
}

input:checked + .slider2:before {
    background-image: linear-gradient(to right, #f904b4, #7335e7);
}

```

```

.slider2.round {
    border-radius: 34px;
}

.slider2.round:before {
    border-radius: 50%;
}

.inputform {
    position: relative;
    left: 15%;
    margin-top: 5px;
}

.submitbutton {
    background: #f9f9f9;
    width: 120px;
    border: beige;
    padding: 15px;
    border-radius: 5px;
    margin: 10px;
    cursor: pointer;
    position: relative;
    left: 9%;
    background-image: linear-gradient(to right, #f904b4, #7335e7);
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
    font-size: 20;
    font-weight: 500;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
}

.inputspace {
    position: relative;
    border-style: solid;
    border-width: 0px;
    border-radius: 10px;
    width: 180px;
    height: 30px;
    right: 6%;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    padding-left: 20px;
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
    font-size: 20;
    font-weight: 500;
}

/* buat algoritma */
.radioButton {
    float: left;
    margin: 5px;
    width: 100px;
}

```

```

height: 40px;
position: relative;
left: 6%;
overflow: hidden;
text-align: center;
vertical-align: middle;
color: white;
}

.radioButton label,
.radioButton input {
    display: block;
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
}

.radioButton input[type="radio"] {
    opacity: 0;
    z-index: 100;
}

.radioButton input[type="radio"]:checked + label {
    background-image: linear-gradient(to right, #f904b4, #7335e7);
    border-radius: 4px;
}

.radioButton label {
    cursor: pointer;
    z-index: 90;
    line-height: 1.8em;
}

.algoritma {
    margin-top: 40px;
}

```

### 3.1.4. mapstyles.css

```

/* style buat body */
body {
    background-color: #15181f;
    background-image: linear-gradient(to bottom right, #212429, #20232a);
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
}

/* style buat header 1 */
h1 {
    text-align: center ;

```

```

        color: white;
        position: relative;
        padding-bottom: 25px;
    }

/* buat nodes */
.circle {
    fill: black;
}

.links line {
    stroke-opacity: 0.6;
}

.nodes circle {
    stroke: #ffff;
    stroke-width: 3px;
}

.node-label {
    font-size: 12px;
    font-weight: bold;
    fill: white;
}

marker {
    opacity: 0.6;
}

.link-label {
    opacity: 0.6;
    font-size: 12px;
    fill: white;
}

/* input file */
#my_file {
    display: none;
}

#get_file {
    background: #f9f9f9;
    width: 120px;
    border: beige;
    padding: 15px;
    border-radius: 5px;
    margin: 10px;
    cursor: pointer;
    position: relative;
    left: 23%;
    background-image: linear-gradient(to bottom right, #fa07ad, #6230df);
    margin-bottom: 70px;
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
}

```

```

        font-size: 20;
        font-weight: 500;
        box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    }

/* side navbar untuk toggle */
.sidenav {
    height: 60px;
    width: 250px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #292a2f;
    overflow-x: hidden;
    overflow-y: hidden;
    padding-top: 20px;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    background-image: linear-gradient(to bottom right, #292a2c, #23262d);
    margin-top: 170px;
    margin-left: 30px;
    border-style: solid;
    border-width: 0px;
    border-radius: 25px;
}

/* side navbar untuk input edge dari node yang ada pada map */
.sidenav2 {
    height: 220px;
    width: 250px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #292a2f;
    overflow-x: hidden;
    overflow-y: hidden;
    padding-top: 20px;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    background-image: linear-gradient(to bottom right, #292a2c, #23262d);
    margin-top: 270px;
    margin-left: 30px;
    border-style: solid;
    border-width: 0px;
    border-radius: 25px;
    color: #ffffff;
}

.judul {
    text-align: center;
}

/* side navbar untuk input file dan node */
.sidenav3 {

```

```

height: 210px;
width: 250px;
position: fixed;
z-index: 1;
top: 0;
left: 0;
background-color: #292a2f;
overflow-x: hidden;
overflow-y: hidden;
padding-top: 20px;
box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
background-image: linear-gradient(to bottom right, #292a2c, #23262d);
margin-top: 540px;
margin-left: 30px;
border-style: solid;
border-width: 0px;
border-radius: 25px;
}

/* untuk output jarak */
.sidenav4 {
    height: 60px;
    width: 250px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #292a2f;
    overflow-x: hidden;
    overflow-y: hidden;
    padding-top: 20px;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    background-image: linear-gradient(to bottom right, #292a2c, #23262d);
    margin-top: 800px;
    margin-left: 30px;
    border-style: solid;
    border-width: 0px;
    border-radius: 25px;
    color: white;
    text-align: center;
}

/* toggle pakai maps */
.inlinetoggle {
    width: 150px;
    position: relative;
    left: 10%;
    height: 70px;
}

.text {
    text-align: center;
    vertical-align: middle;
}

```

```

        color: white;
        position: relative;
        left: 10%;
        margin-top: 10px;
    }

.switch {
    margin-bottom: 100px;
    position: relative;
    display: inline-block;
    width: 60px;
    height: 34px;
    left: 20%;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
}

.text, .switch {
    display: inline-block;
}

.switch input {
    opacity: 0;
    width: 0;
    height: 0;
}

.slider {
    position: absolute;
    cursor: pointer;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: #ccc;
    -webkit-transition: .4s;
    transition: .4s;
}

.slider:before {
    position: absolute;
    content: "";
    height: 26px;
    width: 26px;
    left: 4px;
    bottom: 4px;
    background-color: white;
    -webkit-transition: .4s;
    transition: .4s;
}

input:checked + .slider {
    background-image: linear-gradient(to bottom right, #fa07ad, #6230df);
}

```

```

input:checked + .slider:before {
    -webkit-transform: translateX(26px);
    -ms-transform: translateX(26px);
    transform: translateX(26px);
}

.slider.round {
    border-radius: 34px;
}

.slider.round:before {
    border-radius: 50%;
}

.slider2 {
    position: absolute;
    cursor: pointer;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-image: linear-gradient(to bottom right, #fa07ad, #6230df);
    -webkit-transition: .4s;
    transition: .4s;
}

.slider2:before {
    position: absolute;
    content: "";
    height: 26px;
    width: 26px;
    left: 30px;
    bottom: 4px;
    background-color: white;
    -webkit-transition: .4s;
    transition: .4s;
}

input:checked + .slider2 {
    -webkit-transform: translateX(26px);
    -ms-transform: translateX(26px);
    transform: translateX(26px);
}

input:checked + .slider2:before {
    background-image: linear-gradient(to bottom right, #fa07ad, #6230df);
}

.slider2.round {
    border-radius: 34px;
}

.slider2.round:before {
    border-radius: 50%;
}

```

```

}

.inputform {
    position: relative;
    left: 15%;
}

.submitbutton {
    background: #f9f9f9;
    width: 120px;
    border: beige;
    padding: 15px;
    border-radius: 5px;
    margin: 10px;
    cursor: pointer;
    position: relative;
    top: -20%;
    left: 8%;
    background-image: linear-gradient(to bottom right, #fa07ad, #6230df);
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
    font-size: 20;
    font-weight: 500;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
}

.inputspace {
    position: relative;
    border-style: solid;
    border-width: 0px;
    border-radius: 10px;
    width: 180px;
    height: 30px;
    right: 6%;
    box-shadow: 7px 10px 10px rgb(0 0 0 / 0.2);
    padding-left: 20px;
    font-family: 'Miriam Libre', sans-serif;
    font-family: 'Source Sans Pro', sans-serif;
    font-size: 20;
    font-weight: 500;
}

/* buat map */
#map {
    width: 1400px;
    height: 800px;
    position: relative;
    left: 19%;
    border-style: solid;
    border-radius: 25px;
}

/* buat algoritma */
.radioButton {

```

```

        float: left;
        margin: 5px;
        width: 100px;
        height: 40px;
        position: relative;
        left: 6%;
        overflow: hidden;
        text-align: center;
        vertical-align: middle;
        color: white;
    }

.radioButton label,
.radioButton input {
    display: block;
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
}

.radioButton input[type="radio"] {
    opacity: 0;
    z-index: 100;
}

.radioButton input[type="radio"]:checked + label {
    background-image: linear-gradient(to right, #f904b4, #7335e7);
    border-radius: 4px;
}

.radioButton label {
    cursor: pointer;
    z-index: 90;
    line-height: 1.8em;
}

.algoritma {
    margin-top: 40px;
}

```

## 3.2. Implementasi Algoritma

### 3.2.1. button.js

```

// graph input variable
var names;
var adjMatrix = null;

// inisialisasi algo yang digunakan
var algo = "";

```

```

// input node awal dan akhir dari form input
var inputstart = document.getElementById("start");
var inputend = document.getElementById("end");
let startberubah = false;
let endberubah = false;
var startNode;
var endNode;

// realtime update value start
inputstart.addEventListener("input", () => {
    startberubah = true;
    updateValues();
});

// realtime update value end
inputend.addEventListener("input", () => {
    endberubah = true;
    updateValues();
});

var delay;
function updateValues() {
    startNode = inputstart.value;
    endNode = inputend.value;

    // jika waktu diubah valuenya jadi kosong
    if (startNode == "" || endNode == "") {
        // Reset the changed variables
        startberubah = false;
        endberubah = false;
    }

    // run otomatis abis diubah
    if (startNode == "" || endNode == "" || algo == "" ||
        (document.getElementById('gotomaps') && adjMatrix == null) ||
        (document.getElementById('gotowithoutmaps') && markers.length == 0)) {
        // kalo masih ada yang kosong skip
        return;
    } else {
        clearTimeout(delay);
        if(document.getElementById('gotomaps'))delay = setTimeout(() =>
shortestPath();}, 1000);
        else delay = setTimeout(() => {mapVisualize();}, 1000);
    }
}

document.getElementById('get_file').onclick = function() {
    document.getElementById('my_file').click();
};

document.getElementById('my_file').onchange = async function() {
    const file = this.files[0];
    try {

```

```

    // get input
    input = await inputFile(file);

    // set variable and visualize
    names = input[0];
    adjMatrix = input[1];
    nodes = nodeNameToNodeList(names);
    links = adjMatrixToList(names, adjMatrix);
    setScale(adjMatrix);
    graphVisualize();
} catch(err) {
    alert(err);
}
};

function gotoMaps() {
    if(document.getElementById('gotoMaps').checked) {
        window.location='indexmap.html';
        return false;
    }
    return true;
}

function goToWithoutMaps() {
    if(document.getElementById('goToWithoutMaps').checked) {
        window.location='index.html';
        return false;
    }
    return true;
}

function using(value) {
    if (value == "UCS") {
        algo = "ucs";
    } else if(value == "A*") {
        algo = "astar";
    }

    if(document.getElementById('gotoMaps')) { // if index.html
        if(adjMatrix == null)alert("Input file before run");
        else if(startNode !== "" && endNode !== "") { // if start and end is
not empty, execute right after
            shortestPath();
        }
    } else { // if indexmap.html
        if(markers.length == 0){alert("Choose node before run");}
        else if(startNode !== "" && endNode !== "") { // if start and end is
not empty, execute right after
            mapVisualize();
        }
    }
}

function shortestPath() {

```

```

try{
    if (algo == "")throw "Select algorithm";
    if (startNode == "")throw "Start node is empty";
    if (endNode == "")throw "End node is empty";
    if (!names.includes(startNode))throw "Start node is not exist";
    if (!names.includes(endNode))throw "End node is not exist";

    const startIndex = names.indexOf(startNode);
    const endIndex = names.indexOf(endNode);
    if (startIndex == endIndex)throw "Select nodes that are different";

    var euclidArray = new Array(names.length).fill(0);
    if(algo == "astar")euclidArray = makeEuclidArrayFile(endIndex);
    const path = pathFinding(adjMatrix, startIndex, endIndex,
euclidArray);
    if (path.length === 0)throw "End node is not reachable from Start
node";
    links2 = pathToList(names, adjMatrix, path);
    const cost = pathCost(path, adjMatrix);

    graphVisualize();
    updateJarak(cost);
} catch(err){
    alert(err);
}
}

// button util
function updateJarak(jarak) {
    var hasiljarak = document.getElementById('output-jarak');
    hasiljarak.textContent = jarak;
}

function getEuclidDistance(index1, index2) {
    const x1 = nodes[index1].x;
    const x2 = nodes[index2].x;
    const y1 = nodes[index1].y;
    const y2 = nodes[index2].y;
    const dx = Math.abs(x1 - x2);
    const dy = Math.abs(y1 - y2);
    const dist = Math.sqrt(dx*dx + dy*dy) / 100;
    return dist;
}

function makeEuclidArrayFile(endIndex) {
    var euclidArray = new Array(names.length).fill(0);
    for(let i = 0 ; i < names.length ; i++) {
        const euclidDistance = getEuclidDistance(i, endIndex);
        if(euclidDistance == 0)euclidArray[i] = 0; // bikin kasus khusus biar
gak keluar domain scale
        else euclidArray[i] = scale.invert(euclidDistance);
    }
    console.log(euclidArray);
    return euclidArray;
}

```

```
}
```

### 3.2.2. map.js

```
// inisialisasi map dan id nodenya
var map = L.map('map').setView([-6.892, 107.612], 16);
var ctrmarker = 0; // id node
var bluePath = [];

L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: 'Map data © <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors'
}).addTo(map);

var adjacencyMatrix = [];
var markers = [] // daftar marker

// onclick add marker
map.on('click', function(event) {
    var url = 'https://router.project-osrm.org/nearest/v1/driving/' +
    event.latlng.lng + ',' + event.latlng.lat;
    fetch(url).then(function(response) {
        return response.json();
    }).then(function(json) {
        // buat mengubah marker yang ada di gedung dll jadi di jalan terdekat
        var nearestPoint = L.latLng(json.waypoints[0].location[1],
        json.waypoints[0].location[0]);
        ctrmarker+=1;

        // label marker
        var myIcon = L.divIcon({
            className: 'my-icon',
            html: '<div>' + ctrmarker + '</div>',
            iconSize: [40, 40],
        });
        // masukkan marker beserta dengan atribut-atributnya
        var marker = L.marker(nearestPoint, {
            title: ctrmarker,
            icon: myIcon,
        }).addTo(map);

        var marker = L.marker(nearestPoint).addTo(map);
        // tambahkan marker ke dalam array
        markers.push(marker);

        // untuk membuat/mengupdate adjacency matrix
        var temp = [];
        for (let i = 0; i < markers.length-1; i++) {
            adjacencyMatrix[i].push(0)
        }

        for (let i = 0; i < markers.length; i++) {
```

```

        temp.push(0);
    }

    adjacencyMatrix.push(temp);
});
});

// fungsi untuk memvisualisasikan map
function mapVisualize() {
    try{
        if (algo == "")throw "Select algorithm";
        if (startNode == "")throw "Start node is empty";
        if (endNode == "")throw "End node is empty";

        const startIndex = parseInt(startNode) - 1;
        if(isNaN(startIndex))throw "Start node is not an integer";
        const endIndex = parseInt(endNode) - 1;
        if(isNaN(endIndex))throw "End node is not an integer";

        if (startIndex < 0 || startIndex >= ctrmarker)throw "Start node is not exist";
        if (endIndex < 0 || endIndex >= ctrmarker)throw "End node is not exist";
        if (startIndex == endIndex)throw "Select nodes that are different";

        var euclidArray = new Array(markers.length).fill(0);
        if(algo == "astar")euclidArray = makeEuclidArrayMap(endIndex);
        const path = pathFinding(adjacencyMatrix, startIndex, endIndex,
        euclidArray);
        if (path.length === 0)throw "End node is not reachable from Start
node";
        const cost = pathCost(path, adjacencyMatrix);

        mapVisualizePath(path);
        updateJarak(cost);
    }catch(err){
        alert(err);
    }
}

// util
function mapVisualizePath(path) {
    for(let i = 0 ; i < bluePath.length ; i++){
        map.removeControl(bluePath[i]);
    }
    bluePath = [];

    const pathLength = path.length;
    for(let i = 1; i < pathLength ; i ++){
        var router = L.Routing.control({
            waypoints: [
                markers[path[i-1]].getLatLng(),
                markers[path[i]].getLatLng()
            ],

```

```

        lineOptions: {
            styles: [{color: 'blue', opacity: 0.7, weight: 5}]
        },
        fitSelectedRoutes: true,
        draggableWaypoints: false,
        routeWhileDragging: false,
        createMarker: function() { return null; },
        addWaypoints: false,
    })
    router.addTo(map);
    bluePath.push(router);
}
}

function makeEuclidArrayMap(endIndex) { // get all euclid distance from marker
to end node
    var euclidArray = new Array(markers.length).fill(0);
    for(let i = 0 ; i < markers.length ; i++) {
        euclidArray[i] = markers[i].getLatLng()

.distanceTo(markers[endIndex].getLatLng());
    }
    console.log(euclidArray);
    return euclidArray;
}

// fungsi yang digunakan untuk menambah edge atau rute pada map
function addEdge() {
    try {
        var source = document.getElementById("source").value;
        var target = document.getElementById("target").value;

        if (source == "" && target == "") throw "Select source node and target
node";
        if (source == "") throw "Select source node";
        if (target == "") throw "Select target node";

        source = parseInt(source);
        target = parseInt(target);
        if (isNaN(source)) throw "Source node input is not an integer";
        if (isNaN(target)) throw "Target node input is not an integer";
        if (source > ctrmarker && target > ctrmarker) throw "Select node that
already exists for source node and target node";
        if (source > ctrmarker) throw "Select node that already exists for
source node";
        if (target > ctrmarker) throw "Select node that already exists for
target node";
        if (source <= 0 && target <= 0) throw "Source node and target node
input must be > 0";
        if (source <= 0) throw "Source node and input must be > 0";
        if (target <= 0) throw "Target node and input must be > 0";
        if (source == target) throw "Select nodes that are different";
        if (adjacencyMatrix[source - 1][target - 1] != 0) throw "Edge already
exist, select a different one"
    }
}

```

```

var router = L.Routing.control({
    waypoints: [
        markers[source - 1].getLatLng(),
        markers[target - 1].getLatLng()
    ],
    fitSelectedRoutes: true,
    draggableWaypoints: false,
    routeWhileDragging: false,
    createMarker: function() { return null; },
    addWaypoints: false
});

router.addTo(map);
router.on('routesfound', function(e) {
    var route = e.routes[0];
    var distance = route.summary.totalDistance; // distance in meters
    adjacencyMatrix[source - 1][target - 1] = parseInt(distance);
});
console.log(adjacencyMatrix);
} catch (err) {
    alert(err);
}
}
}

```

### 3.2.3. path.js

```

class activeNode {
    node;
    path;
    cost;
    priority;
    constructor(node, path, cost, priority) {
        this.node = node;
        this.path = path;
        this.cost = cost;
        this.priority = priority;
    }
}

/*
    input : weighted adj matrix, start node index, target node index
    if aStarArray is all zero, then the algorithm is UCS,
    else (aStarArray is not zero) algorithm is A*
    input format : 0-indexed
    output : arrays of pair, shortest path from start to target with ucs
algorithm
    ex output : path = [1,2,3,4] means the path is 1->2->3->4
    if there is no path from start to end or start==end path will be []
*/
function pathFinding(adjMatrix, start, target, aStarArray) {
    var isAstar = (aStarArray.length !== 0)

```

```

// Input Output variable
const nodeAmount = adjMatrix.length;
var path = [];

// Search variable
const visited = Array(nodeAmount).fill(false);
const comparator = (a, b) => {
    return a.priority - b.priority;
};
const queue = new PriorityQueue(comparator);

// start search from start node
const startNode = new activeNode(start, [start], 0, 0);
visited[start] = true;
queue.enqueue(startNode);

// start searching
var found = false;
while (!found && !queue.isEmpty()) {
    const currentNode = queue.dequeue();
    visited[currentNode.node] = true;
    if (currentNode.node === target) { // reach target
        found = true; // found
        path = currentNode.path; // assign path
    } else { // continue search
        for (let next = 0; next < nodeAmount; next++) {
            if (visited[next]) continue; // skip if visited
            if (adjMatrix[currentNode.node][next] === 0) continue; // skip
if no path

                // set new value
                const nextpath = [...currentNode.path, next];
                const nextcost = currentNode.cost +
adjMatrix[currentNode.node][next];
                const nextpriority = nextcost + aStarArray[next];
                const nextNode = new activeNode(next, nextpath, nextcost,
nextpriority);
                // console.log(nextpriority);

                // queueing to search next node
                queue.enqueue(nextNode);
            }
        }
    }
}
return path;
}

```

### 3.2.4. queue.js

```

class PriorityQueue {
    array;
    compareFunction;
}

```

```

constructor(compareFunction) {
    this.array = [];
    this.compareFunction = compareFunction;
}
enqueue(val) {
    this.array.push(val);
    this.array.sort(this.compareFunction); // ? masih O(n) mungkin bisa di
optimize
}
dequeue() {
    const val = this.front();
    this.array.shift();
    return val;
}
isEmpty() {
    return this.array.length === 0;
}
size() {
    return this.array.length;
}
front() {
    return this.array[0];
}
}

```

### 3.2.5. util.js

```

/*
    input : File that get from input file form
    output : if valid get {node names, adj matrix} else get exception error
*/
function inputFileFrom(file) {
return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.readAsText(file);
    const nodeName = [];
    const adjMatrix = [];
    reader.onload = function () {
        // get file input
        var contents = reader.result?.toString();
        if (!contents) reject("file is empty"); // ? asumsi undefined karena
empty
        contents = contents.replaceAll("\r", "");
        const contentLines = contents.split("\n");

        // get node amount
        const nodeAmount = parseInt(contentLines[0]);
        if (isNaN(nodeAmount)) reject("Invalid Format, node amount is not
integer");
        if (contentLines.length < (1 + 2 * nodeAmount)) reject("Invalid Format,
too few line");
        if (contentLines.length > (1 + 2 * nodeAmount)) reject("Invalid Format,
too many line");
    }
})
}

```

```

extra trailing line");

    // get node name
    for (let i = 1; i < 1 + nodeAmount; i++) {
        if(contentLines[i] === '')reject("Invalid Format, node name in
line " + i + " is empty");
        if(nodeName.includes(contentLines[i]))reject("Invalid Format, node
name in line " + i + " already used before");
        nodeName.push(contentLines[i]);
    }

    // get adjacency matrix
    for (let i = 1 + nodeAmount; i < 1 + 2 * nodeAmount; i++) {
        const line = contentLines[i].split(" ");
        const currentLine = i - nodeAmount;
        const errorMessage = "Invalid Format, adjacency matrix line " +
currentLine;
        if (line.length < nodeAmount)reject(errorMessage + " have few
column");
        if (line.length > nodeAmount)reject(errorMessage + " have extra
column");

        const numLines = [];
        for (let j = 0; j < nodeAmount; j++) {
            const num = parseFloat(line[j]);
            if (isNaN(num))reject(errorMessage + " column " + (j + 1) + "
value is not number");
            if (num < 0)reject(errorMessage + " column " + (j + 1) + "
value is negative");
            numLines.push(num);
        }
        adjMatrix.push(numLines);
    }
    resolve([nodeName, adjMatrix]);
};

reader.onerror = function() {
    reject(reader.error);
};
});
}
}

function nodeNameToNodeList(names) {
    var list = [];
    const nodeAmount = names.length; // ? pengen pake foreach malah gajelas
    for (let i = 0; i < nodeAmount ; i++) {
        list.push({"name" : names[i]});
    }
    return list;
}

function adjMatrixToList(names, adjMatrix) {
    var list = [];
    const nodeAmount = names.length;

```

```

        for (let i = 0; i < nodeAmount ; i++){
            for (let j = 0; j < nodeAmount ; j++){
                if(adjMatrix[i][j] === 0)continue;
                list.push({source : names[i], target : names[j], distance :
adjMatrix[i][j]});
            }
        }
        return list;
    }

    function pathToList(names, adjMatrix, path){
        var list = [];
        const pathLength = path.length;
        for(let i = 1; i < pathLength;i++){
            list.push({source : names[path[i-1]], target : names[path[i]],
distance : adjMatrix[path[i-1]][path[i]]});
        }
        return list;
    }

    function pathCost(path, adjMatrix) {
        var cost = 0;
        const pathLength = path.length;
        for (let i = 1; i < pathLength; i++){
            cost += adjMatrix[path[i-1]][path[i]];
            curr = path[i];
        }
        return cost;
    }
}

```

### 3.2.6. visualize.js

```

//initialize svg
var svg = d3.select("svg");
var width = svg.attr("width");
var height = svg.attr("height");
var scale = d3.scaleLog()
    .domain([1, 100])
    .range([2, 10]);

// JIKA ELEMEN INI ISINYA DIUBAH, MAKA OTOMATIS VISUALISASI JUGA AKAN BERUBAH
// inisialisasi semua elemen-elemen graf
var nodes = [
    {name: "1"}, 
    {name: "2"}, 
    {name: "3"}, 
    {name: "4"}, 
    {name: "5"}, 
    {name: "6"}, 
    {name: "7"}, 
    {name: "8"}, 
    {name: "9"}, 
    {name: "10"}, 
    {name: "11"}, 
    {name: "12"}, 
    {name: "13"}, 
    {name: "14"}, 
    {name: "15"}, 
    {name: "16"}, 
    {name: "17"}, 
    {name: "18"}, 
    {name: "19"}, 
    {name: "20"}, 
    {name: "21"}, 
    {name: "22"}, 
    {name: "23"}, 
    {name: "24"}, 
    {name: "25"}, 
    {name: "26"}, 
    {name: "27"}, 
    {name: "28"}, 
    {name: "29"}, 
    {name: "30"}, 
    {name: "31"}, 
    {name: "32"}, 
    {name: "33"}, 
    {name: "34"}, 
    {name: "35"}, 
    {name: "36"}, 
    {name: "37"}, 
    {name: "38"}, 
    {name: "39"}, 
    {name: "40"}, 
    {name: "41"}, 
    {name: "42"}, 
    {name: "43"}, 
    {name: "44"}, 
    {name: "45"}, 
    {name: "46"}, 
    {name: "47"}, 
    {name: "48"}, 
    {name: "49"}, 
    {name: "50"}, 
    {name: "51"}, 
    {name: "52"}, 
    {name: "53"}, 
    {name: "54"}, 
    {name: "55"}, 
    {name: "56"}, 
    {name: "57"}, 
    {name: "58"}, 
    {name: "59"}, 
    {name: "60"}, 
    {name: "61"}, 
    {name: "62"}, 
    {name: "63"}, 
    {name: "64"}, 
    {name: "65"}, 
    {name: "66"}, 
    {name: "67"}, 
    {name: "68"}, 
    {name: "69"}, 
    {name: "70"}, 
    {name: "71"}, 
    {name: "72"}, 
    {name: "73"}, 
    {name: "74"}, 
    {name: "75"}, 
    {name: "76"}, 
    {name: "77"}, 
    {name: "78"}, 
    {name: "79"}, 
    {name: "80"}, 
    {name: "81"}, 
    {name: "82"}, 
    {name: "83"}, 
    {name: "84"}, 
    {name: "85"}, 
    {name: "86"}, 
    {name: "87"}, 
    {name: "88"}, 
    {name: "89"}, 
    {name: "90"}, 
    {name: "91"}, 
    {name: "92"}, 
    {name: "93"}, 
    {name: "94"}, 
    {name: "95"}, 
    {name: "96"}, 
    {name: "97"}, 
    {name: "98"}, 
    {name: "99"}, 
    {name: "100"}]

```

```

    {name: "8"},  

    {name: "9"},  

    {name: "10"},  

    {name: "12"},  

];  
  

// daftar hubungan antar node  

var links = [  

    {source: "1", target: "2", distance: "40"},  

    {source: "1", target: "5", distance: "40"},  

    {source: "1", target: "6", distance: "40"},  

    {source: "2", target: "3", distance: "40"},  

    {source: "2", target: "7", distance: "40"},  

    {source: "3", target: "4", distance: "40"},  

    {source: "8", target: "3", distance: "40"},  

    {source: "4", target: "5", distance: "40"},  

    {source: "4", target: "9", distance: "40"},  

    {source: "5", target: "10", distance: "40"},  

    {source: "10", target: "5", distance: "40"},  

    {source: "12", target: "10", distance: "140"},  

];  
  

// daftar hubungan antar node yang merupakan shortest path  

var links2 = [  

    {source: "12", target: "10", distance: "140"},  

];  
  

function graphVisualize() {  

    // reset svgnya terlebih dahulu  

    svg.selectAll("*").remove();  
  

    // simulasikan  

    var simulation = d3.forceSimulation(nodes)  

        .force("link", d3.forceLink().links(links).id(function(d) { return d.name; }))  

        .distance(function(d) {  

            return scale(d.distance) * 100; }))  

        .force("charge", d3.forceManyBody().strength(-30))  

        .force("center", d3.forceCenter(width / 2, height / 2))  

        .on("tick", ticked);  
  

    var link = svg  

        .append("g")  

        .attr("class", "links")  

        .selectAll("line")  

        .data(links)  

        .enter()  

        .append("line")  

        .attr("stroke-width", function(d) {  

            return 3;  

        })  

        .attr("stroke", function(d) {  

            if (isItInList(links2, d)) {  


```

```

        return "blue"; // jika linknya adalah antara node yang merupakan
shortest path
    } else {
        return "#999"; // jika default
    }
});

var linkLabels = svg.selectAll(".link-label")
.data(links)
.enter()
.append("text")
.attr("class", "link-label")
.text(function(d) { return d.distance; });

var node = svg
.append("g")
.attr("class", "nodes")
.selectAll("circle")
.data(nodes)
.enter()
.append("circle")
.attr("r", 22)
.attr("fill", "red")
.call(
d3
.drag()
.on("start", dragstarted)
.on("drag", dragged)
.on("end", dragended)
);
}

svg.append("defs").append("marker")
.attr("id", "arrow-blue")
.attr("viewBox", "0 -5 10 10")
.attr("refX", 19)
.attr("refY", 0)
.attr("markerWidth", 8)
.attr("markerHeight", 8)
.attr("orient", "auto-start-reverse")
.append("path")
.attr("fill", "blue")
.attr("d", "M0,-5L10,0L0,5");

svg.append("defs").append("marker")
.attr("id", "default")
.attr("viewBox", "0 -5 10 10")
.attr("refX", 19)
.attr("refY", 0)
.attr("markerWidth", 10)
.attr("markerHeight", 7)
.attr("orient", "auto-start-reverse")
.append("path")
.attr("fill", "grey")
.attr("d", "M0,-5L10,0L0,5");

```

```

var nodelabel = svg.selectAll(".node-label")
  .data(nodes)
  .enter()
  .append("text")
  .attr("class", "node-label")
  .text(function(d) { return d.name; });

// buat mengatur posisi elemen" di simulasi
simulation.on("tick", function() {
  link
    .attr("x1", function(d) { return d.source.x; })
    .attr("y1", function(d) { return d.source.y; })
    .attr("x2", function(d) { return d.target.x; })
    .attr("y2", function(d) { return d.target.y; })
    .attr("marker-end", function(d) {
      if (isItInList(links2, d)) {
        return "url(#arrow-blue)";
      } else {
        return "url(#default)";
      }
    });
  node
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
  nodelabel
    .attr("x", function(d) { return d.x + 26; })
    .attr("y", function(d) { return d.y + 5; });

  linkLabels
    .attr("x", function(d) { return (d.source.x + d.target.x) / 2; })
    .attr("y", function(d) { return (d.source.y + d.target.y) / 2; })
    .attr("dx", function(d) {
      var angle = Math.atan2(scale(d.target.y)*100 - scale(d.source.y)*100,
      scale(d.target.x)*100 - scale(d.source.x)*100);
      return Math.cos(angle) * 30;
    })
    .attr("dy", function(d) {
      var angle = Math.atan2(scale(d.target.y)*100 - scale(d.source.y)*100,
      scale(d.target.x)*100 - scale(d.source.x)*100);
      return Math.sin(angle) * 30;
    })

    // dx dan dy untuk mendekatkan label dengan ujung panah
  });
}

function ticked() {
  link
    .attr("x1", function(d) {
      return d.source.x;
    })
    .attr("y1", function(d) {

```

```

        return d.source.y;
    })
    .attr("x2", function(d) {
        return d.target.x;
    })
    .attr("y2", function(d) {
        return d.target.y;
    });
}

node
    .attr("cx", function(d) {
        return d.x;
    })
    .attr("cy", function(d) {
        return d.y;
    });
}

// efek drag
function dragstarted(d) {
    if (!d3.event.active) simulation.alphaTarget(0.3).restart();
    d.fx = d.x;
    d.fy = d.y;
}

function dragged(d) {
    d.fx = d3.event.x;
    d.fy = d3.event.y;
}

function dragended(d) {
    if (!d3.event.active) simulation.alphaTarget(0);
    d.fx = null;
    d.fy = null;
}
}

// utility functions
function isItInList(links, data){
    for (let i = 0; i < links.length; i++) {
        if (links[i].source === data.source.name && links[i].target ===
data.target.name && links[i].distance === data.distance) {
            return true;
        }
    }
    return false;
}

function setScale(adjMatrix) {
    var minVal = Number.MAX_VALUE;
    var maxVal = 0;
    const nodeAmount = adjMatrix.length;
    for(let i = 0 ; i < nodeAmount ; i++) {
        for(let j = 0 ; j < nodeAmount ; j++) {

```

```
        if(adjMatrix[i][j] == 0) continue;
        minVal = Math.min(minVal, adjMatrix[i][j]);
        maxVal = Math.max(maxVal, adjMatrix[i][j]);
    }
}
scale = d3.scaleLog()
    .domain([minVal, maxVal])
    .range([1, 5]);
}
```

Bab 4

# Eksperimen

#### 4.1. Eksperimen Graf

#### **4.1.1. Peta jalan sekitar kampus ITB/Dago/Bandung Utara**

*input*

Tucil 3 - Tanpa Maps

Pakai map?

Menentukan Tujuan

Start Node:

Destination Node:

UCS A\*

The cost is:

*output UCS*

Tucil 3 - Tanpa Maps

Pakai map?

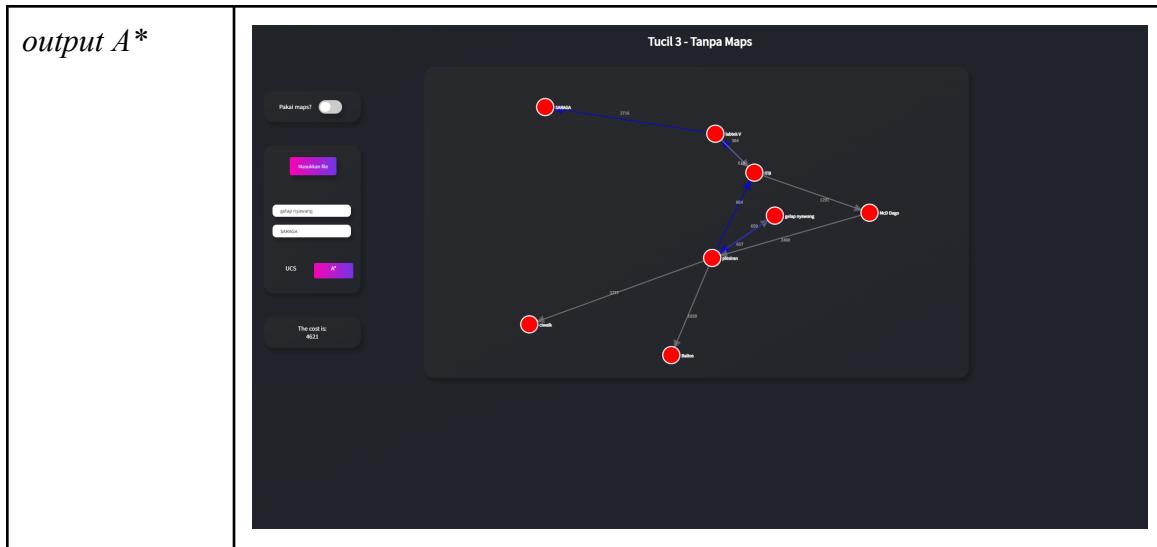
Menentukan Tujuan

Start Node:

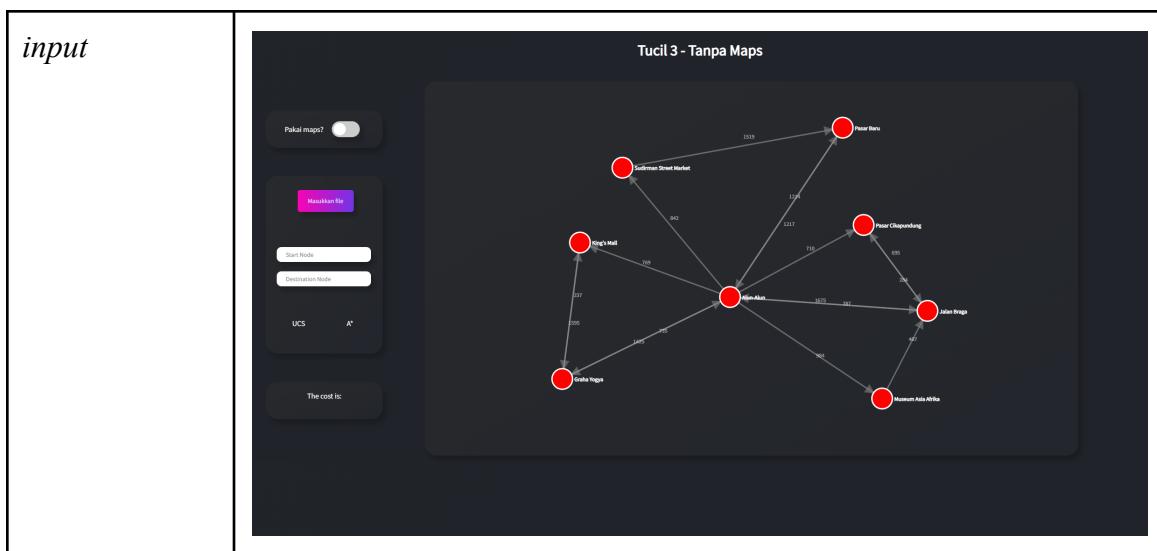
Destination Node:

UCS A\*

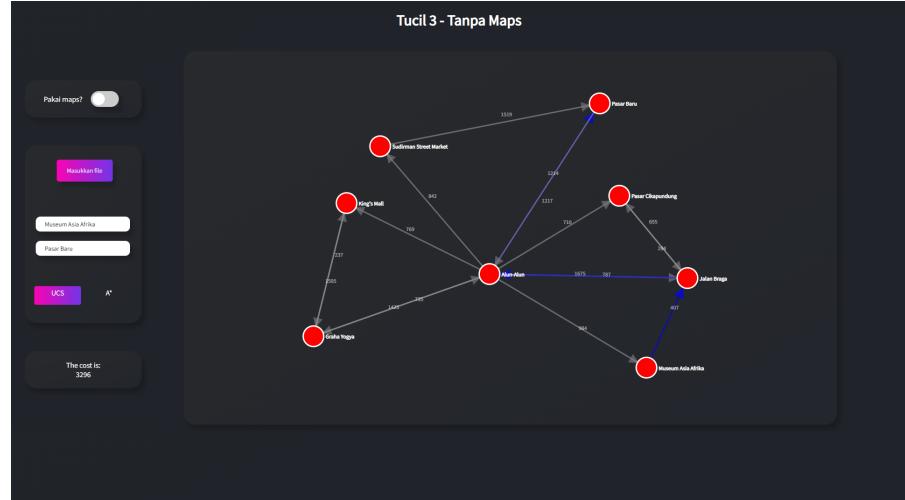
The cost is:



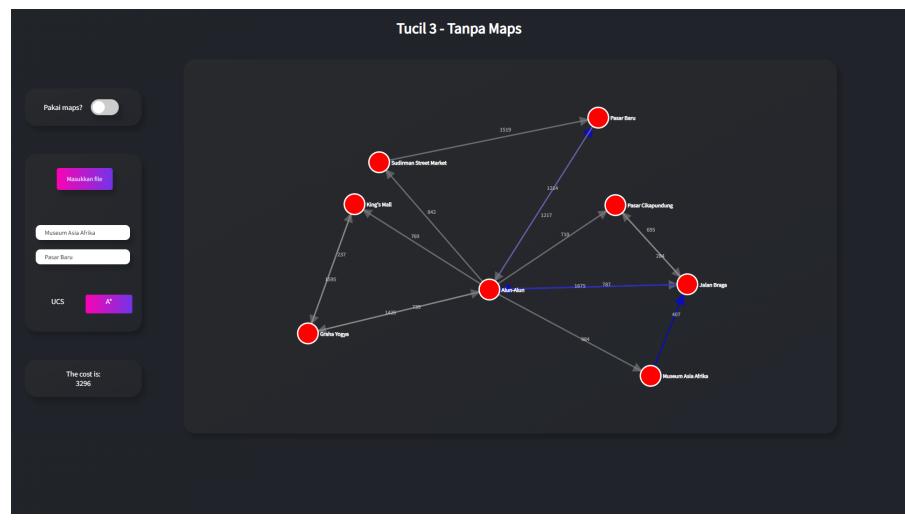
#### **4.1.2. Peta jalan sekitar Alun-alun Bandung**



*output UCS*

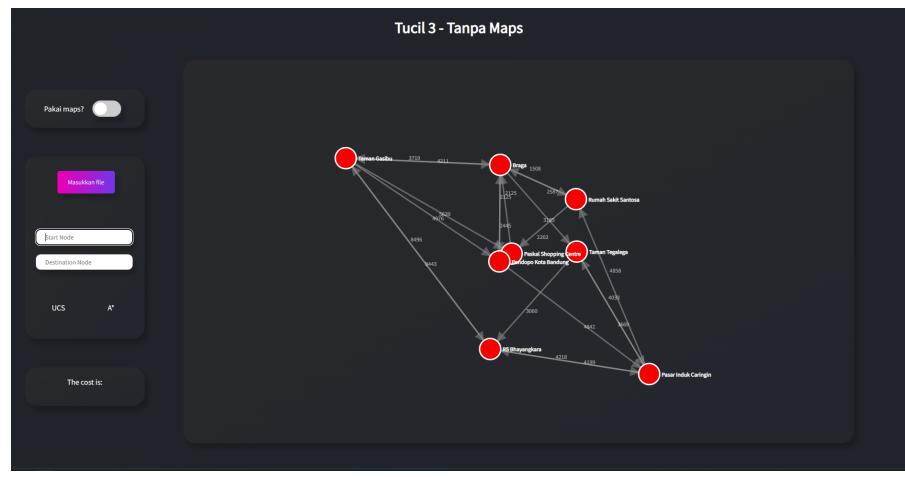


*output A\**

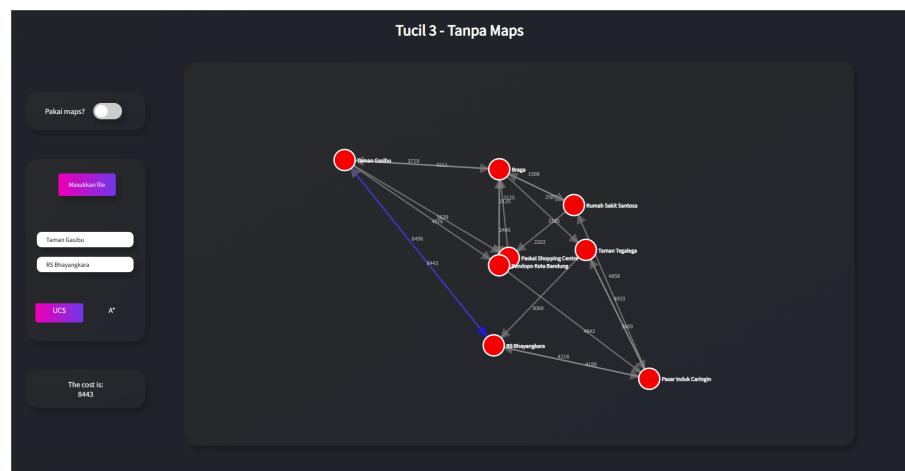


#### 4.1.3. Peta jalan sekitar Buahbatu atau Bandung Selatan

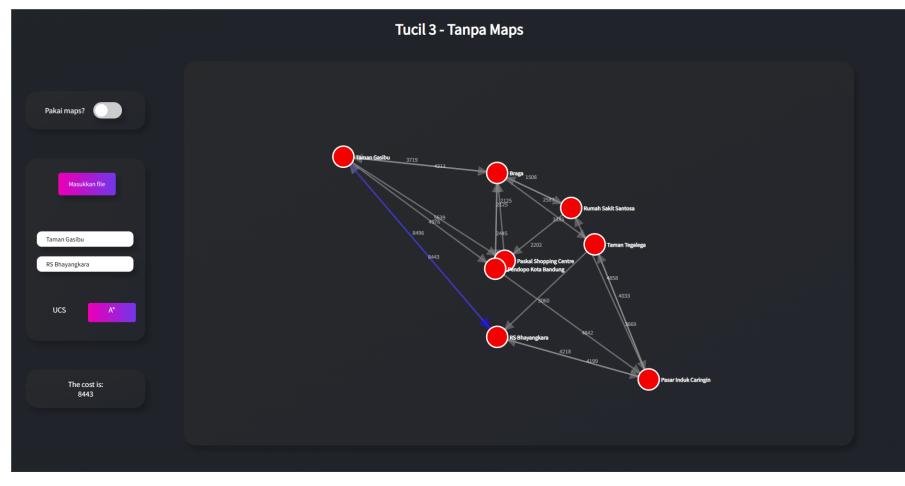
*input*



*output UCS*

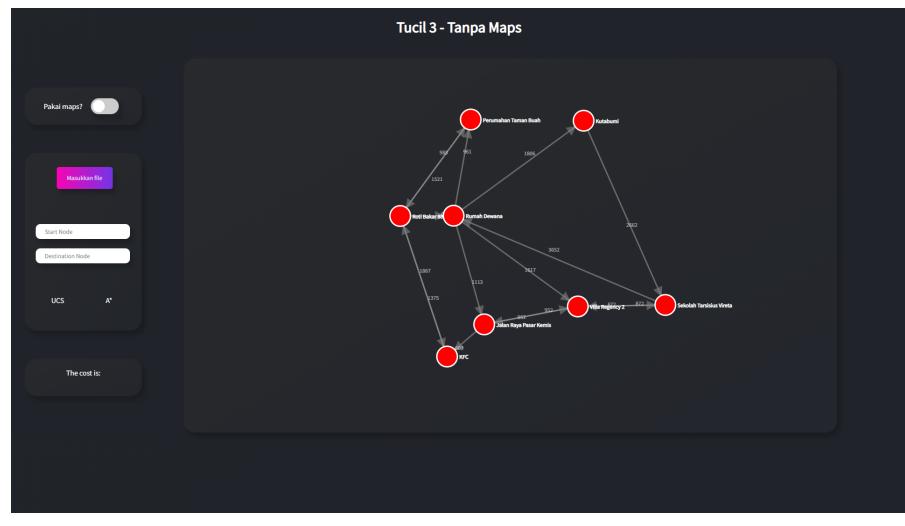


*output A\**

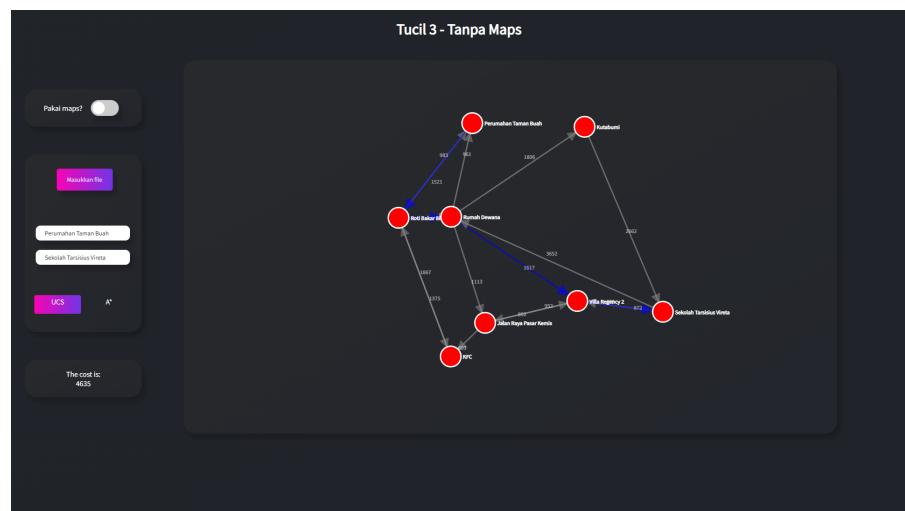


#### 4.1.4. Peta jalan sebuah kawasan di kabupaten Tangerang

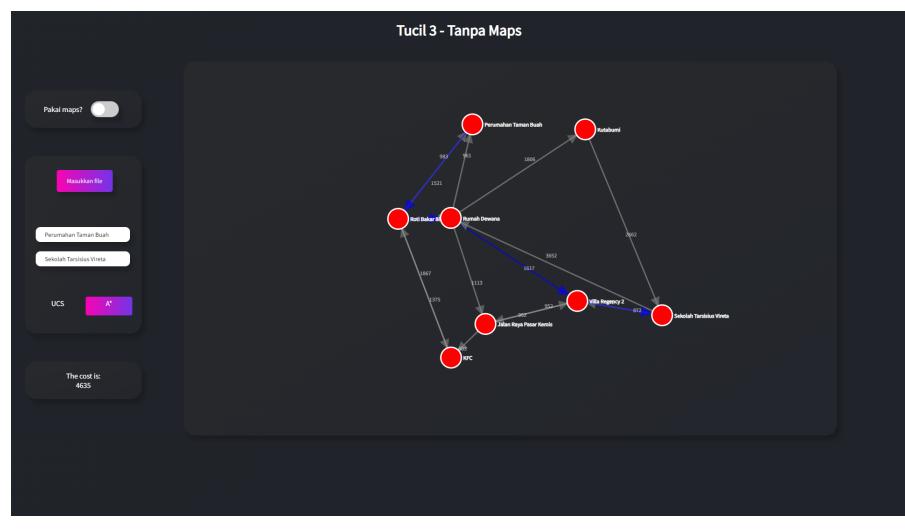
*input*



*output UCS*

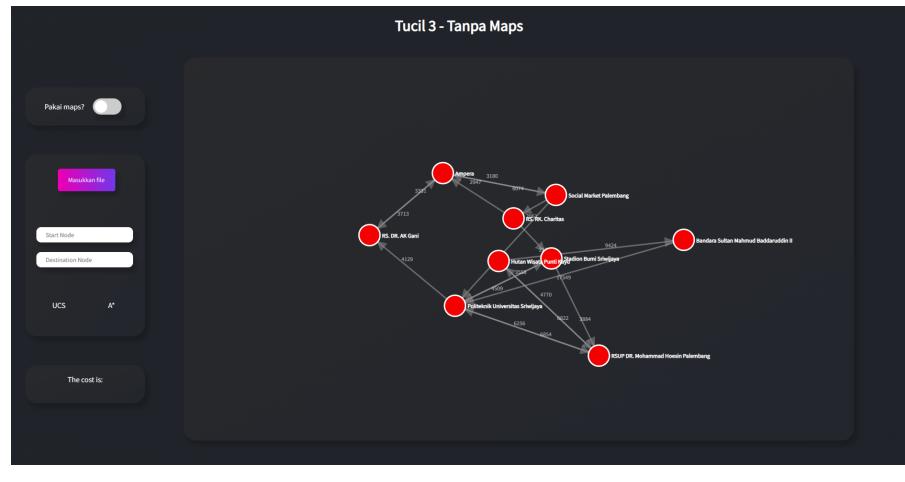


*output A\**



#### 4.1.5. Peta jalan sebuah kawasan di kota Palembang

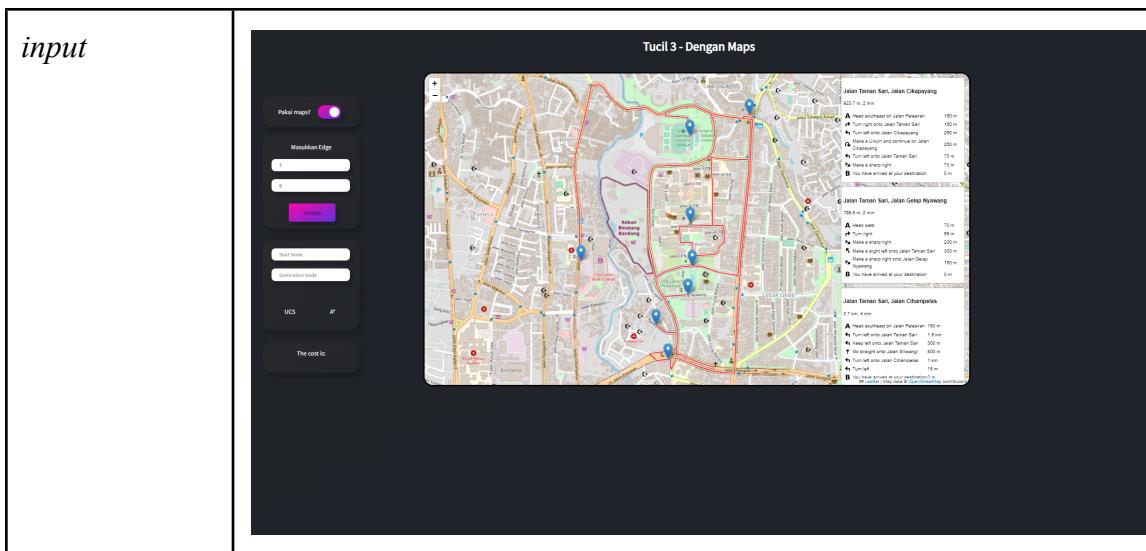
*input*



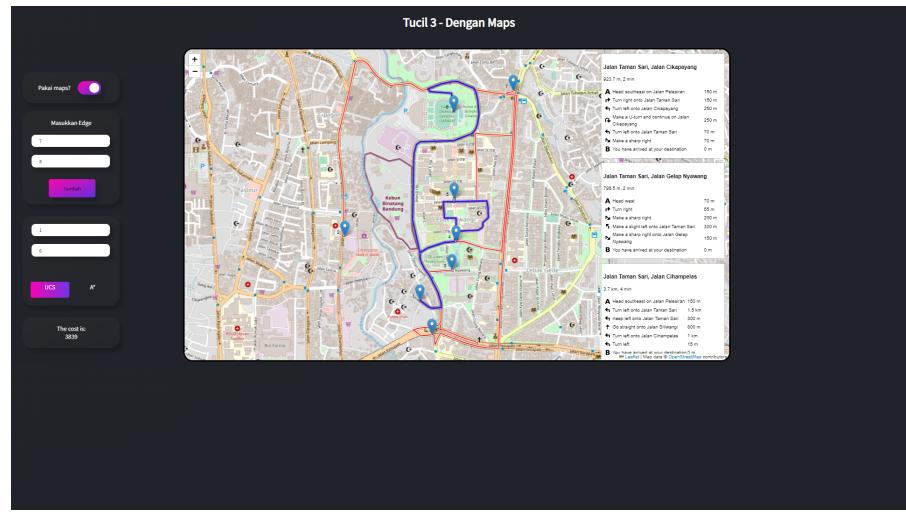


## 4.2. Eksperimen Map

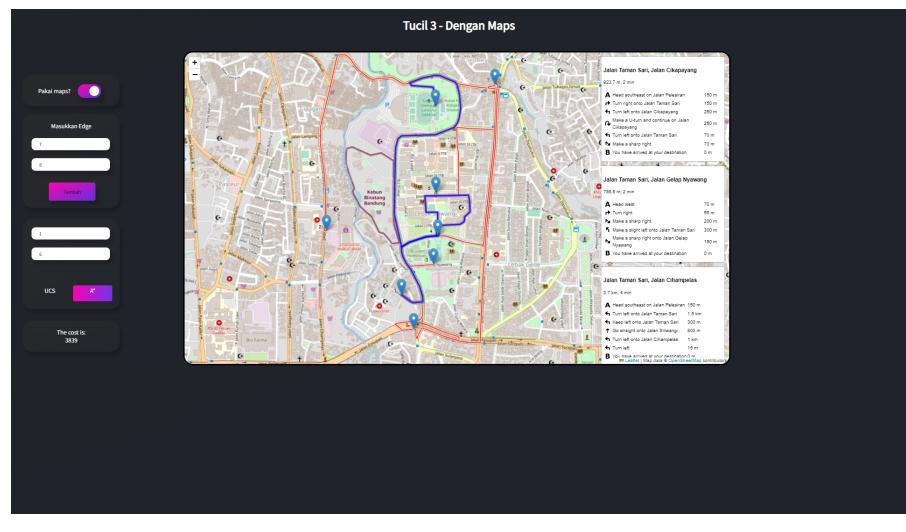
### 4.2.1. Peta jalan sekitar kampus ITB/Dago/Bandung Utara



*output UCS*

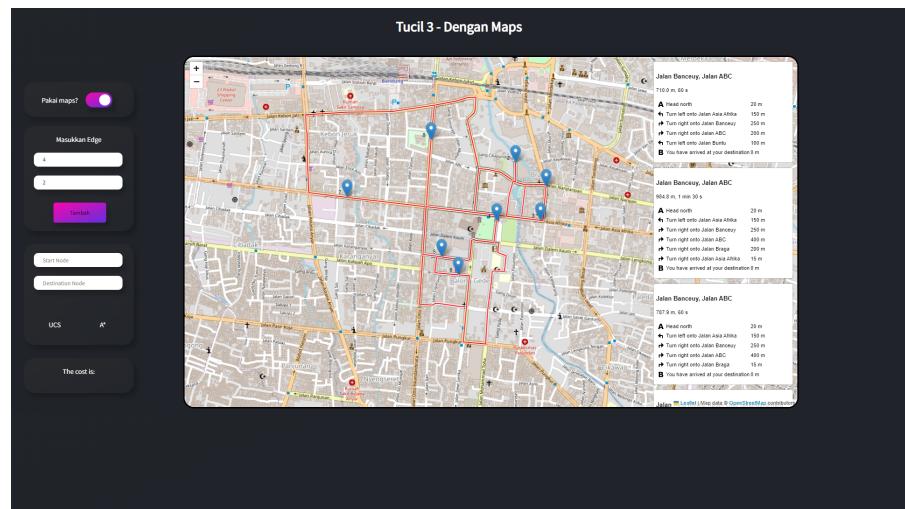


*output A\**

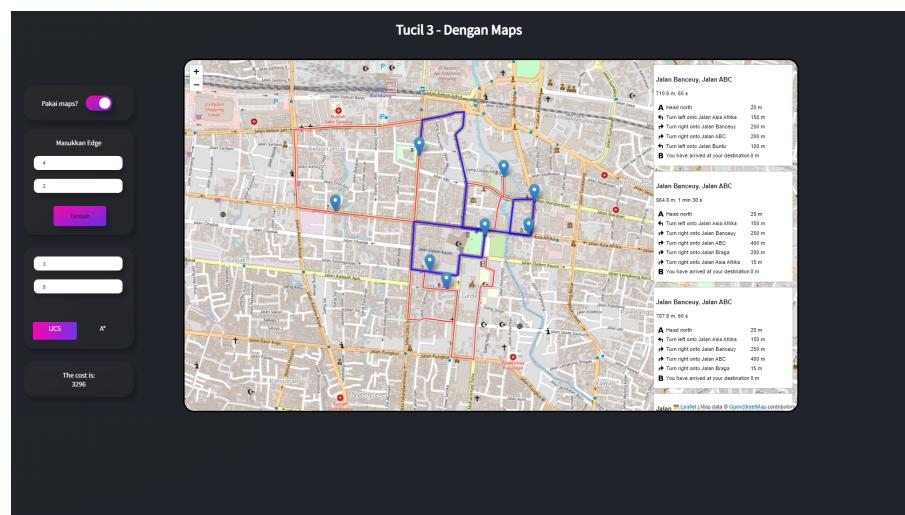


#### 4.2.2. Peta jalan sekitar Alun-alun Bandung

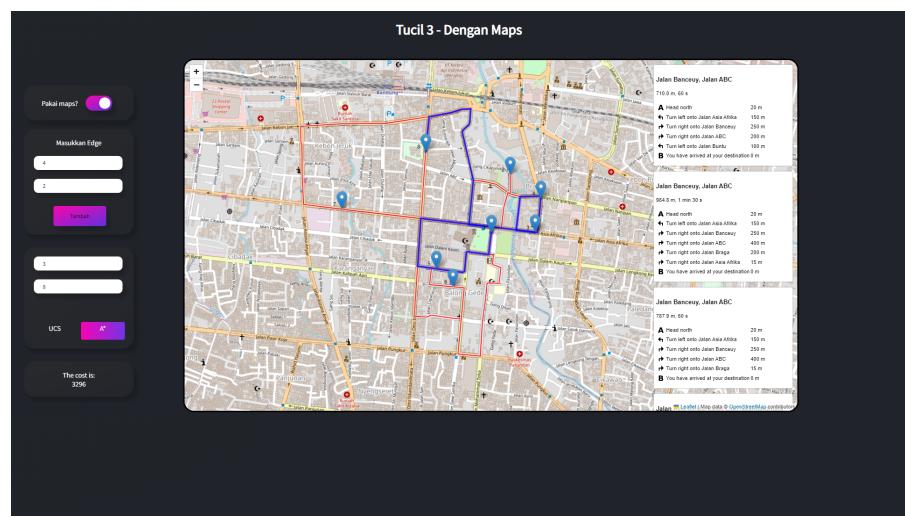
*input*



*output UCS*



*output A\**

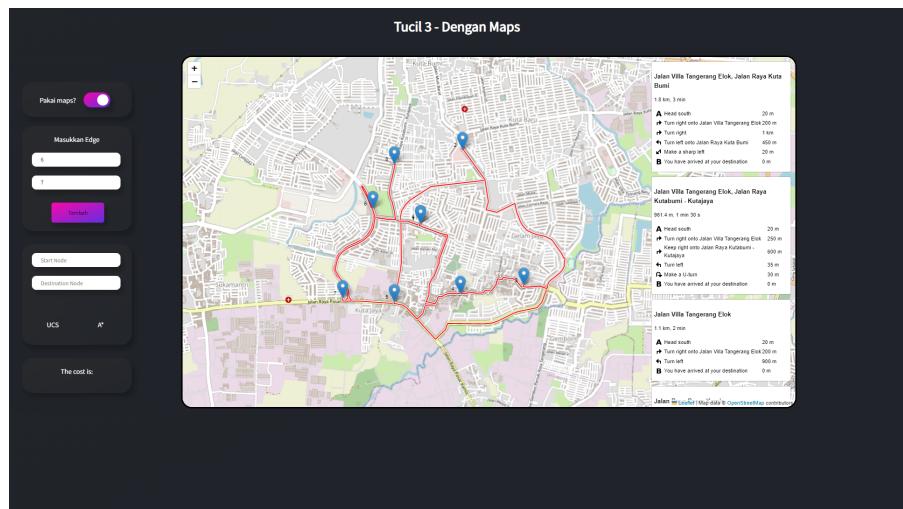


#### 4.2.3. Peta jalan sekitar Buahbatu atau Bandung Selatan

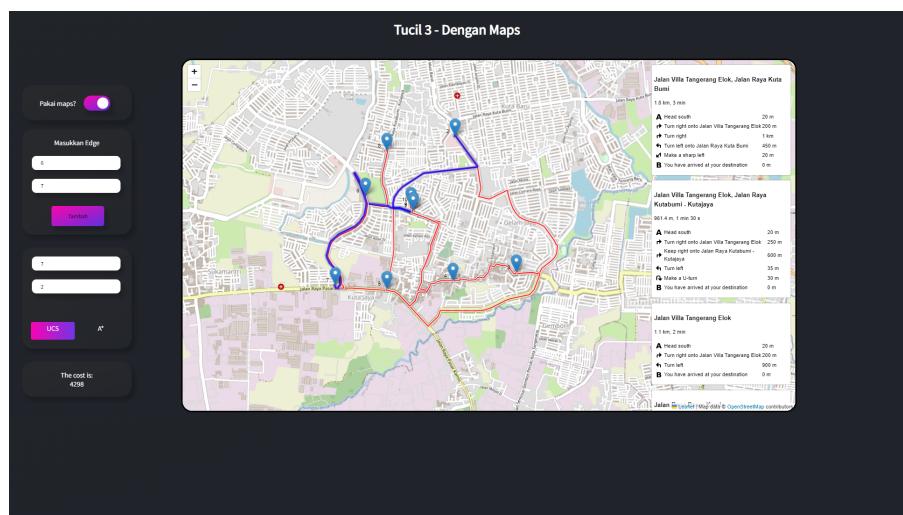
<i>input</i>	<p>Tucil 3 - Dengan Maps</p> <p>Jalan Kebon Jati, Jalan ABC 2.1 km, 2 min 30 s</p> <ul style="list-style-type: none"> <li>A Head east on Jalan Kebon Jati 1 km</li> <li>↗ Turn right onto Jalan Otto Iskandarindata 250 m</li> <li>↗ Turn left onto Jalan ABC 300 m</li> <li>↗ Turn left onto Jalan Banowijaya 15 m</li> <li>↗ Turn right onto Jalan ABC 400 m</li> <li>↗ Turn left onto Jalan Braga 40 m</li> <li>B You have arrived at your destination 0 m</li> </ul> <p>Jalan Sumiraja, Jalan Otto Iskandarindata 3.2 km, 4 min</p> <ul style="list-style-type: none"> <li>A Head north on Jalan Braga 350 m</li> <li>↗ Turn left to stay on Jalan Sumiraja 200 m</li> <li>↗ Turn left to stay on Jalan Sumiraja 350 m</li> <li>↗ Turn left onto Jalan Otto Iskandarindata 2 km</li> <li>↗ Turn left 100 m</li> <li>↗ Turn right 150 m</li> <li>↗ Turn right 30 m</li> <li>B You have arrived at your destination 0 m</li> </ul> <p>Jalan Mohammad Toha 3.1 km, 5 min</p> <ul style="list-style-type: none"> <li>A Head north 30 m</li> <li>↗ Make a sharp right 200 m</li> <li>↗ Turn left 50 m</li> <li>↗ Turn right onto Jalan Mohammad Toha 2.5 km</li> <li>↗ Turn left 70 m</li> <li>B You have arrived at your destination 0 m</li> </ul>
<i>output UCS</i>	<p>Tucil 3 - Dengan Maps</p> <p>Jalan Kebon Jati, Jalan ABC 2.1 km, 2 min 30 s</p> <ul style="list-style-type: none"> <li>A Head east on Jalan Kebon Jati 1 km</li> <li>↗ Turn right onto Jalan Otto Iskandarindata 250 m</li> <li>↗ Turn left onto Jalan ABC 300 m</li> <li>↗ Turn left onto Jalan Banowijaya 15 m</li> <li>↗ Turn right onto Jalan ABC 400 m</li> <li>↗ Turn left onto Jalan Braga 40 m</li> <li>B You have arrived at your destination 0 m</li> </ul> <p>Jalan Sumiraja, Jalan Otto Iskandarindata 3.2 km, 4 min</p> <ul style="list-style-type: none"> <li>A Head north on Jalan Braga 350 m</li> <li>↗ Turn left to stay on Jalan Sumiraja 200 m</li> <li>↗ Turn left to stay on Jalan Sumiraja 350 m</li> <li>↗ Turn left onto Jalan Otto Iskandarindata 2 km</li> <li>↗ Turn left 100 m</li> <li>↗ Turn right 150 m</li> <li>↗ Turn right 30 m</li> <li>B You have arrived at your destination 0 m</li> </ul> <p>Jalan Mohammad Toha 3.1 km, 5 min</p> <ul style="list-style-type: none"> <li>A Head north 30 m</li> <li>↗ Make a sharp right 200 m</li> <li>↗ Turn left 50 m</li> <li>↗ Turn right onto Jalan Mohammad Toha 2.5 km</li> <li>↗ Turn left 70 m</li> <li>B You have arrived at your destination 0 m</li> </ul>
<i>output A*</i>	<p>Tucil 3 - Dengan Maps</p> <p>Jalan Kebon Jati, Jalan ABC 2.1 km, 2 min 30 s</p> <ul style="list-style-type: none"> <li>A Head east on Jalan Kebon Jati 1 km</li> <li>↗ Turn right onto Jalan Otto Iskandarindata 250 m</li> <li>↗ Turn left onto Jalan ABC 300 m</li> <li>↗ Turn left onto Jalan Banowijaya 15 m</li> <li>↗ Turn right onto Jalan ABC 400 m</li> <li>↗ Turn left onto Jalan Braga 40 m</li> <li>B You have arrived at your destination 0 m</li> </ul> <p>Jalan Sumiraja, Jalan Otto Iskandarindata 3.2 km, 4 min</p> <ul style="list-style-type: none"> <li>A Head north on Jalan Braga 350 m</li> <li>↗ Turn left to stay on Jalan Sumiraja 200 m</li> <li>↗ Turn left to stay on Jalan Sumiraja 350 m</li> <li>↗ Turn left onto Jalan Otto Iskandarindata 2 km</li> <li>↗ Turn left 100 m</li> <li>↗ Turn right 150 m</li> <li>↗ Turn right 30 m</li> <li>B You have arrived at your destination 0 m</li> </ul> <p>Jalan Mohammad Toha 3.1 km, 5 min</p> <ul style="list-style-type: none"> <li>A Head north 30 m</li> <li>↗ Make a sharp right 200 m</li> <li>↗ Turn left 50 m</li> <li>↗ Turn right onto Jalan Mohammad Toha 2.5 km</li> <li>↗ Turn left 70 m</li> <li>B You have arrived at your destination 0 m</li> </ul>

#### 4.2.4. Peta jalan sebuah kawasan di kabupaten Tangerang

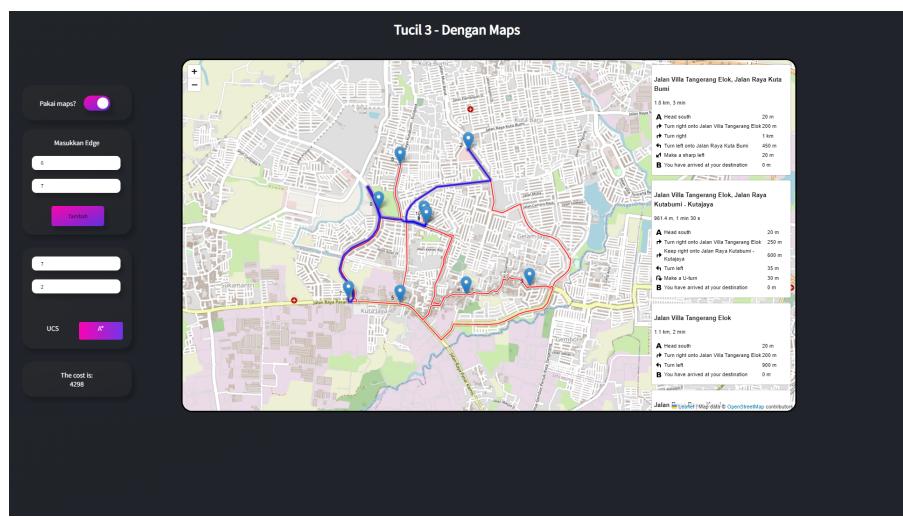
*input*



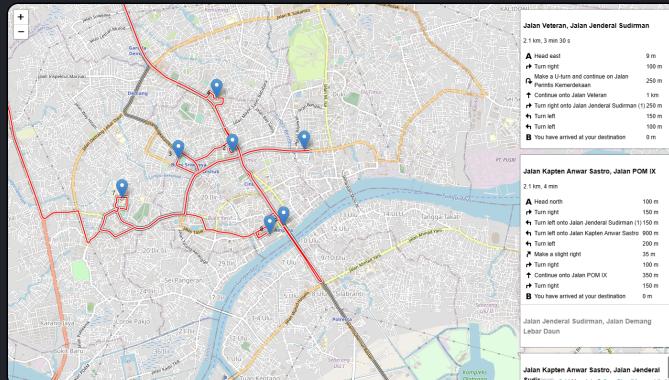
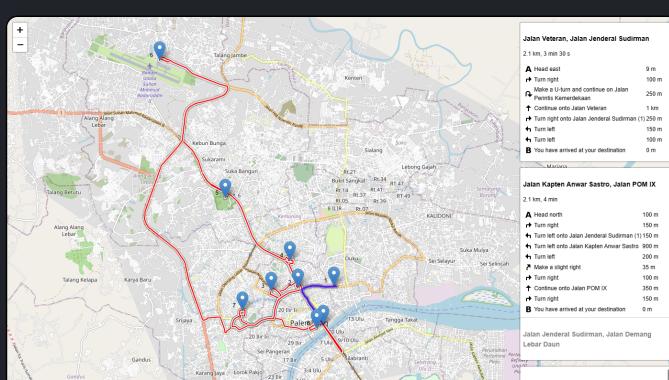
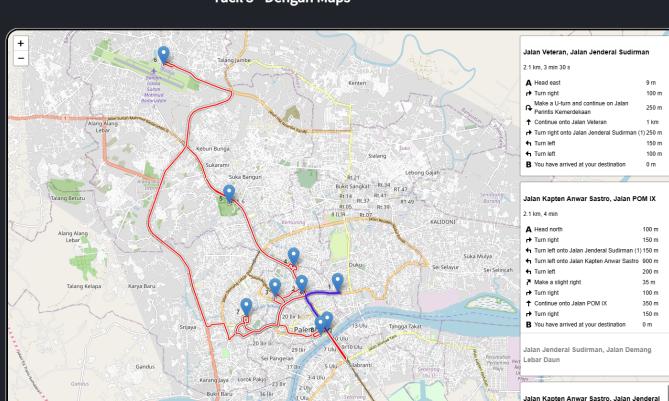
*output UCS*



*output A\**



#### 4.2.5. Peta jalan sebuah kawasan di kota Palembang

<p><i>input</i></p>	 <p>The cost is:</p>
<p><i>output UCS</i></p>	 <p>The cost is: 3180</p>
<p><i>output A*</i></p>	 <p>The cost is: 3180</p>

## Bab 5

### Kesimpulan dan Saran

#### 5.1. Kesimpulan

Dari hasil percobaan yang telah kami lakukan, didapat bahwa hasil dari algoritma UCS dan A\* itu adalah sama. Pada pengimplementasiannya, kedua algoritma tersebut sama yang membedakan hanyalah penggunaan nilai heuristik pada algoritma A\*. Karena persamaan tersebut kita bisa mengimplementasikan algoritma UCS dan A\* pada algoritma yang sama, dan yang membedakannya adalah nilai heuristik yang digunakan pada algoritma UCS adalah 0.

#### 5.2. Saran

Pada saat pengimplementasian visualisasi, akan lebih baik jika memilih bahasa pemrograman yang memiliki *library* visualisasi bawaan yang sudah bagus. Jika ingin agar *edge* yang divisualisasikan ukurannya tidak berbeda terlalu jauh dengan *edge* yang lain, maka bisa menggunakan skala untuk menormalisasikan panjang dari *edge* saat divisualisasikan. Banyak cara yang dapat digunakan untuk mendapatkan nilai skala tersebut, salah satunya adalah dengan menjadikan *edge* terkecil dan *edge* terbesar sebagai skala.

#### 5.3. Refleksi

Pada saat mengimplementasikan visualisasi, akan lebih baik jika memahami terlebih dahulu API yang digunakan memiliki fitur apa saja. Mengerti fitur-fitur yang ditawarkan oleh API tersebut dapat mempermudah dalam mengimplementasikan fitur-fitur yang diinginkan.

Selanjutnya, akan lebih baik jika proses *enqueue* yang digunakan pada *priority queue* dibuat lebih *optimize* dengan kompleksitas yang lebih kecil dari  $O(n)$ .

#### **5.4. Tanggapan**

Dalam pelaksanaan tugas kecil ini, banyak sekali tantangan yang didapat. Namun, kebanyakan dari tantangan tersebut sebenarnya adalah tantangan visualisasi. Ini adalah tantangan terberat karena kami menggunakan *platform* berbasis *web*, sehingga dibutuhkan eksplorasi lebih lanjut. Namun dengan adanya tugas kecil ini, ilmu yang kami dapatkan mengenai pengembangan visualisasi dalam *platform web* juga bertambah.

## Lampiran

Link github : [https://github.com/Ainzw0rth/Tucil3\\_13521069\\_13521173.git](https://github.com/Ainzw0rth/Tucil3_13521069_13521173.git)

Tabel penilaian:

1	Program dapat menerima input graf	✓
2	Program dapat menghitung lintasan terpendek dengan UCS	✓
3	Program dapat menghitung lintasan terpendek dengan A*	✓
4	Program dapat menampilkan lintasan terpendek serta jaraknya	✓
5	Bonus: Program dapat menerima input peta dengan Google Map API dan menampilkan peta serta lintasan terpendek pada peta	✓

## **Daftar Pustaka**

edunex.itb.ac.id (Diakses pada tanggal 9 April 2023)

Munir, Rinaldi. 2021. “Penentuan Rute (Route/Path Planning) Bagian 1: BFS, DFS, UCS, Greedy Best First Search”.

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian1-2021.pdf>, diakses 9 April 2023.

Munir, Rinaldi. 2021. “Penentuan Rute (Route/Path Planning) Bagian 2: Algoritma A\*”.

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf>, diakses 9 April 2023.

<https://www.redblobgames.com/pathfinding/a-star/implementation.html#python-dijkstra>

“Implementation of A\*”.

<https://www.redblobgames.com/pathfinding/a-star/implementation.html#python-dijkstra>, diakses pada 9 April 2023

GIS Solutions, Leaflet playlist.

<https://youtube.com/playlist?list=PLnHeemIZ2QILzh2FL0Nvo5X0RF0mF3Xor>