

Universitas Pembangunan Jaya

Kelompok Algoritma : Dijkstra

Desain dan Analisis Algoritma Part 3

Anggota

- ✦ Muhammad Iqbal Fattah (2021071002)
 - ✦ Khairullah Nurfitri Ramadhan (2021071009)
 - ✦ Iqbal Ramadhan Syahputra (2021071010)
 - ✦ Haibibi Ar-Farrezy (2021071031)
-

Problems

1) Pengguna aplikasi peta tersesat sehingga mengalami insiden kecelakaan karena aplikasi salah pengambilan rute dan tujuan yang dimaksudkan atau aplikasi tidak mengetahui besar volume kendaraan dari pengguna.

2) Pengguna sering tersesat saat mencari rute tujuan dengan jalan terefisien mungkin.



Problems Support

✦ **Waze dikritik oleh pengguna karena mengarahkan pengguna ke jalan curam.**

Berdasarkan peristiwa tersebut, Adams dan penduduk setempat telah memberi sebuah feedback kepada pihak Waze untuk menghapus Baxter Street dari pemilihan rute.

Sumber :
<https://www.liputan6.com/tekno/read/3435066/waze-diprotes-gara-gara-arahkan-pengguna-lewat-jalan-curam>

✦ **Pengemudi roda 4 tersesat dan masuk ke dalam hutan karena mengikuti arahan google map.**

Šuryadi pengemudi mobil Daihatsu Xenia berumur 31 tahun tersesat di Hutan Pacet, Kabupaten Mojokerto pada Hari Kamis, 12/11/2020. Pengemudi tidak sadar sudah berada di kawasan jauh dengan penduduk dan jalanan penuh dengan batu sehingga tidak bisa dilewati roda 4 ataupun roda 2.

Sumber : <https://regional.kompas.com/read/2020/11/13/20334651/ikuti-google-maps-pengemudi-mobil-ini-dibuat-merinding-tersesat-di-hutan?page=all>

✦ **Minibus tersesat di sekitaran kuburan**

Pengemudi asal bekasi mengalami insiden kendaraan tersesat dan mobil selip setelah mengikuti arahan google dari arah Cikole, Kabupaten Bandung Barat ke wilayah Bekasi, Hari Minggu (12/06/22).

Sumber :
<https://news.okezone.com/read/2022/06/15/525/2612131/kronologi-mobil-tersesat-di-kuburan-gegara-ikuti-google-maps-sopir-syok>

Literature Review and Related Work

✦ Penerapan Algoritma Grafik pada Sistem Peta Digital.

Aplikasi peta terkadang menyebabkan pengguna tersasar atau celaka karena arahan dari instruksi peta yang salah. Terkadang mengarahkan pengguna ke tempat tujuan berbeda ataupun rutenya yang terus berubah sehingga membuat pengendara merasa bingung sehingga banyak kejadian kendaraan yang terperosok, tersangkut, tersesat, dan kasus kriminal.

✦ Implementasi dan Analisis Algoritma Pencarian Rute Terpendek di Kota Surabaya.

Pencarian rute terpendek adalah salah satu masalah yang paling banyak dibicarakan dalam transportasi. Dalam masalah transportasi, perhitungan rute terpendek menjadi penting karena harus dilakukan dalam waktu yang sangat singkat dan dengan distorsi yang sekecil mungkin.

✦ Pencarian Jalur Terpendek dengan Algoritma Dijkstra.

Dinamika perkotaan penting dalam memahami bagaimana masyarakat berfungsi. Jumlah rute yang diambil merupakan masalah tersendiri, karena dapat menyebabkan tujuan terlewatkan. Penerapan Algoritma Dijkstra akan membantu dalam hal menentukan jalur terpendek. Nilai pada sisi grafik dapat dinyatakan sebagai jarak antar kota. Jalur terpendek dapat dipahami sebagai proses meminimalkan bobot pada jalur.

Literature Review and Related Work

- ✦ Penggunaan Algoritma Dijkstra dalam Pencarian Rute Tercepat dan Rute Terpendek (Studi kasus pada jalan raya antara wilayah blom M dan kota).

Kemacetan lalu lintas Jakarta sudah menjadi pemandangan biasa. Penggunaan algoritma Dijkstra karena algoritma ini dipastikan dapat menemukan solusi terbaik dan memiliki kompleksitas yang lebih rendah dibandingkan dengan algoritma sejenis. Banyak langkah yang telah dilakukan pemerintah untuk mengurangi kemacetan, seperti pembangunan jalan layang dan underpass, pengoperasian jalur busway, penerapan tree in one hour, dan sebagainya. penggunaan metode model spiral dan kodenya ditulis dalam framework Code Igniter (CI).

Sources Slides

- ✦ Fauzi, Imron. 2011, "Penggunaan algoritma djikstra dalam pencarian rute tercepat dan rute terpendek: studi kasus pada jalan raya antara wilayah blok M dan kota". Skripsi, 2011.
- ✦ Harahap, Muhammad Khoiruddin. Khairina, Nurul. Oktober 2017, "Pencarian Jalur Terpendek dengan Algoritma Dijkstra". Jurnal dan Penelitian Tehknik Informatika Vol. 2 No. 2, Pencarian Jalur Terpendek dengan Algoritma Dijkstra, Oktober 2017.
- ✦ Purwananto, Yudhi. Purwitasari, Diana. Wibowo, Agung Wahyu. Desember 2005, IMPLEMENTASI DAN ANALISIS ALGORITMA PENCARIANRUTE TERPENDEK DI KOTA SURABAYA" , Jurnal Penelitian dan Pengembangan Telekomunikasi, Jurnal .

Implementation Algorithm w/ Python

```
def dijkstra(current, nodes, distances):
    # These are all the nodes which have not been visited yet
    unvisited = {node: None for node in nodes}
    # It will store the shortest distance from one node to another
    visited = {}
    # It will store the predecessors of the nodes
    currentDistance = 0
    unvisited[current] = currentDistance
    # Running the loop while all the nodes have been visited
    while True:
        # iterating through all the unvisited node
        for neighbour, distance in distances[current].items():
            # Iterating through the connected nodes of current_node (for
            # example, a is connected with b and c having values 10 and 3
            # respectively) and the weight of the edges
            if neighbour not in unvisited: continue
            newDistance = currentDistance + distance
            if unvisited[neighbour] is None or unvisited[neighbour] > newDistance:
                unvisited[neighbour] = newDistance
        # Till now the shortest distance between the source node and target node
        # has been found. Set the current node as the target node
        visited[current] = currentDistance
        del unvisited[current]
        if not unvisited: break
        candidates = [node for node in unvisited.items() if node[1]]
        print(sorted(candidates, key = lambda x: x[1]))
        current, currentDistance = sorted(candidates, key = lambda x: x[1])[0]
    return visited
```

```
nodes = ('A', 'B', 'C', 'D', 'E')
distances = {
    'A': {'B': 2, 'C': 3, 'D': 6},
    'B': {'A': 2, 'D': 4},
    'C': {'A': 3, 'D': 3},
    'D': {'B': 4, 'E': 7},
    'E': {'B': 7, 'D': 2}}
current = 'A'
```

```
print(dijkstra(current, nodes, distances))
```


Analysis Data

w/ Python

```
nodes = ('A', 'B', 'C', 'D', 'E')
distances = {
    'A': {'B': 2, 'C': 3, 'D': 6},
    'B': {'A': 2, 'D': 4},
    'C': {'A': 3, 'D': 3},
    'D': {'B': 4, 'E': 7},
    'E': {'B': 7, 'D': 2}}
current = 'A'

print(dijkstra(current, nodes, distances))
```

```
[('B', 2), ('C', 3), ('D', 6)]
[('C', 3), ('D', 6)]
[('D', 6)]
[('E', 13)]
{'A': 0, 'B': 2, 'C': 3, 'D': 6, 'E': 13}
```

```
nodes = ('A', 'B', 'C', 'D', 'E')
distances = {
    'A': {'B': 2, 'C': 3, 'D': 6},
    'B': {'A': 2, 'D': 4},
    'C': {'A': 3, 'D': 3},
    'D': {'B': 4, 'E': 7},
    'E': {'B': 7, 'D': 2}}
current = 'B'

print(dijkstra(current, nodes, distances))
```

```
[('A', 2), ('D', 4)]
[('D', 4), ('C', 5)]
[('C', 5), ('E', 11)]
[('E', 11)]
{'B': 0, 'A': 2, 'D': 4, 'C': 5, 'E': 11}
```

Analysis Data w/ Python

```
nodes = ('A', 'B', 'C', 'D', 'E')
distances = {
    'A': {'B': 2, 'C': 3, 'D': 6},
    'B': {'A': 2, 'D': 4},
    'C': {'A': 3, 'D': 3},
    'D': {'B': 4, 'E': 7},
    'E': {'B': 7, 'D': 2}}
current = 'D'
```

```
print(dijkstra(current, nodes, distances))
```

```
[('B', 4), ('E', 7)]
[('A', 6), ('E', 7)]
[('E', 7), ('C', 9)]
[('C', 9)]
{'D': 0, 'B': 4, 'A': 6, 'E': 7, 'C': 9}
```

```
nodes = ('A', 'B', 'C', 'D', 'E')
distances = {
    'A': {'B': 2, 'C': 3, 'D': 6},
    'B': {'A': 2, 'D': 4},
    'C': {'A': 3, 'D': 3},
    'D': {'B': 4, 'E': 7},
    'E': {'B': 7, 'D': 2}}
current = 'C'
```

```
print(dijkstra(current, nodes, distances))
```

```
[('A', 3), ('D', 3)]
[('D', 3), ('B', 5)]
[('B', 5), ('E', 10)]
[('E', 10)]
{'C': 0, 'A': 3, 'D': 3, 'B': 5, 'E': 10}
```

```
nodes = ('A', 'B', 'C', 'D', 'E')
distances = {
    'A': {'B': 2, 'C': 3, 'D': 6},
    'B': {'A': 2, 'D': 4},
    'C': {'A': 3, 'D': 3},
    'D': {'B': 4, 'E': 7},
    'E': {'B': 7, 'D': 2}}
current = 'E'
```

```
print(dijkstra(current, nodes, distances))
```

```
[('D', 2), ('B', 7)]
[('B', 6)]
[('A', 8)]
[('C', 11)]
{'E': 0, 'D': 2, 'B': 6, 'A': 8, 'C': 11}
```

Thank
you!