

## RESPONSI 2

### PRAKTIKUM STRUKTUR DATA DAN ALGORITMA

2024

#### IDENTITAS

Nama : **Mohammad Nazhiif Al-Ghoni**  
NIM : **L0123084**  
Judul Program : **England Roadmap**  
Deskripsi Program : **Program untuk mencari indeks kota secara terurut dan mencari jarak terpendek antara kota di Inggris**

#### DOKUMENTASI PROGRAM

*(masukkan berbagai penjelasan program yang lebih detail di sini, seperti instruksi pemakaian, fitur-fitur, screenshot, dan lain-lain)*

##### • Mencari indeks kota dari daftar kota yang terurut

- Program menyediakan daftar kota-kota (London, Bristol, Manchester, Liverpool, Birmingham) yang diurutkan secara alfabetis menggunakan algoritma Quick Sort.
- Pengguna dapat memasukkan nama kota yang ingin dicari.
- Program akan mencari indeks dari kota tersebut dalam daftar kota yang terurut menggunakan algoritma Binary Search.
- Jika kota ditemukan, program akan menampilkan indeks dari kota tersebut dalam daftar kota.
- Jika kota tidak ditemukan, program akan memberikan pesan bahwa kota tidak ditemukan.

##### • Mencari jarak terpendek antar kota

- Program menyediakan graf yang merepresentasikan koneksi antar kota beserta jarak (bobot) antar kota.
- Pengguna dapat memasukkan indeks kota awal.
- Program akan menghitung jarak terpendek dari kota awal ke semua kota lainnya menggunakan algoritma Dijkstra.
- Program akan menampilkan jarak terpendek dari kota awal ke setiap kota lainnya.

#### Instruksi Pemakaian:

1. Jalankan program.
2. Program akan menampilkan daftar kota beserta indeksnya.
3. Pilih opsi program yang diinginkan:
  - Opsi 1: Cari indeks kota terurut.
  - Opsi 2: Cari jarak terdekat.
4. Ikuti instruksi pada program untuk memasukkan input yang diperlukan (nama kota atau indeks kota awal).
5. Program akan menampilkan hasil sesuai dengan opsi yang dipilih.

### Fitur:

- Penggunaan struktur data adjacency list untuk merepresentasikan graf.
- Implementasi algoritma Quick Sort untuk mengurutkan daftar kota secara alfabetis.
- Implementasi algoritma Binary Search untuk mencari indeks kota dalam daftar kota yang terurut.
- Implementasi algoritma Dijkstra untuk mencari jarak terpendek antar kota dalam graf.

```
Daftar kota:
0. London
1. Bristol
2. Manchester
3. Liverpool
4. Birmingham

Pilih opsi program:
1. Cari indeks kota terurut.
2. Cari jarak terdekat.
Pilih opsi (1-2): 1

Cari indeks kota secara terurut
Input nama kota: Bristol
Kota Bristol ditemukan di indeks 1
```

```
Daftar kota:
0. London
1. Bristol
2. Manchester
3. Liverpool
4. Birmingham

Pilih opsi program:
1. Cari indeks kota terurut.
2. Cari jarak terdekat.
Pilih opsi (1-2): 2
Input indeks kota awal: 2

Jarak terpendek dari Manchester ke kota-kota lain:
Ke London: 3
Ke Bristol: 1
Ke Manchester: 0
Ke Liverpool: 8
Ke Birmingham: 3
```

***Tabel implementasi materi struktur data dan algoritma***

<b>No.</b>	<b>MATERI</b>	<b>PERAN</b>
1.	Adjacency List (unordered_map dan vector):	Digunakan untuk merepresentasikan graf yang menghubungkan kota-kota beserta bobot (jarak) antar kota. Struktur ini memungkinkan akses yang efisien untuk mendapatkan daftar kota-kota yang terhubung dengan suatu kota tertentu, serta bobot (jarak) antar kota.
2.	Quick Sort (quickSort dan partition):	Digunakan untuk mengurutkan daftar kota-kota secara alfabetis. Algoritma ini memberikan efisiensi waktu rata-rata $O(n \log n)$ , yang lebih baik daripada metode pengurutan sederhana seperti bubble sort atau insertion sort untuk daftar kota yang panjang.
3	Binary Search (binarySearch):	Digunakan untuk mencari indeks suatu kota dalam daftar kota yang sudah terurut secara efisien dengan kompleksitas waktu $O(\log n)$ . Algoritma ini memanfaatkan sifat terurut dari daftar kota yang telah dihasilkan oleh Quick Sort.
4	Dijkstra (dijkstra):	Digunakan untuk mencari jarak terpendek dari suatu kota awal ke semua kota lainnya dalam graf. Algoritma ini menggunakan prinsip greedy dan memanfaatkan struktur data priority queue untuk memastikan efisiensi waktu pencarian jarak terpendek.
5	Priority Queue (priority_queue):	Digunakan oleh algoritma Dijkstra untuk mengatur urutan pengolahan kota-kota berdasarkan jarak terpendek yang telah ditemukan saat itu. Struktur data ini memungkinkan pengambilan elemen dengan jarak terpendek sementara secara efisien.
6	vector:	Digunakan untuk menyimpan daftar kota-kota dan juga untuk menyimpan jarak terpendek dari kota awal ke setiap kota lainnya yang ditemukan oleh algoritma Dijkstra.