TUGAS PRAKTIKUM 8

STRUKTUR DATA

PENJELASAN DARI ALGORITMA BELLMAN-FORD

ILHAM FIRDAUS AZIZA

G.211.22.0079

CODING YANG SAYA GUNAKAN :

```
class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = []

    def add_edge(self, u, v, w):
        self.graph.append([u, v, w])

    def bellman_ford(self, src):
        # Inisialisasi jarak dari source ke semua vertex sebagai tak terhingga
        distance = [float("inf")] * self.V
        distance[src] = 0

        # Lakukan relaksasi untuk semua edge
        for _ in range(self.V - 1):
            for u, v, w in self.graph:
                if distance[u] != float("inf") and distance[u] + w < distance[v]:
                    distance[v] = distance[u] + w

        # Cek apakah terdapat siklus berbobot negatif
        for u, v, w in self.graph:
```

```
            if distance[u] != float("inf") and distance[u] + w < distance[v]:
                print("Graf berisi siklus berbobot negatif")
                return

        # Tampilkan jarak dari source ke setiap vertex
        print("Jarak dari source ke setiap vertex:")
        for i in range(self.V):
            print(f"{i} -> {distance[i]}")



# Contoh penggunaan
g = Graph(5)
g.add_edge(0, 1, -1)
g.add_edge(0, 2, 4)
g.add_edge(1, 2, 3)
g.add_edge(1, 3, 2)
g.add_edge(1, 4, 2)
g.add_edge(3, 2, 5)
g.add_edge(3, 1, 1)
g.add_edge(4, 3, -3)

source_vertex = 0
g.bellman_ford(source_vertex)
```

PENJELASANNYA :

Algoritma Bellman-Ford digunakan untuk mencari jarak terpendek dari suatu vertex (source) ke semua vertex lainnya dalam graf berbobot (graf dengan bobot pada setiap edge). Algoritma ini dapat mengatasi graf dengan bobot negatif, tetapi sensitif terhadap siklus berbobot negatif.

Langkah-langkah algoritma Bellman-Ford adalah sebagai berikut:

1. Inisialisasi Jarak: Inisialisasi jarak dari source ke semua vertex sebagai tak terhingga, kecuali jarak dari source ke dirinya sendiri yang diatur menjadi 0.

2. Relaksasi: Lakukan relaksasi pada setiap edge (u, v) dalam graf. Relaksasi dilakukan dengan memeriksa apakah jarak dari source ke vertex u ditambah dengan bobot edge (u, v) lebih kecil dari jarak saat ini dari source ke vertex v. Jika ya, maka perbarui jarak tersebut.

3. Iterasi: Lakukan langkah relaksasi sebanyak V-1 kali, di mana V adalah jumlah vertex dalam graf. Hal ini diperlukan karena jarak terpendek antara dua vertex paling banyak melibatkan V-1 edge.

4. Deteksi Siklus Berbobot Negatif: Periksa apakah terdapat siklus berbobot negatif dalam graf. Lakukan iterasi sekali lagi dan jika masih terdapat perbaikan jarak, maka graf berisi siklus berbobot negatif.

5. Output Jarak: Tampilkan jarak terpendek dari source ke setiap vertex setelah iterasi selesai.

Keunggulan dan Batasan:

*Keunggulan utama algoritma Bellman-Ford adalah kemampuannya untuk mengatasi graf dengan bobot edge negatif.

*Batasan utama adalah kinerjanya yang kurang efisien dibandingkan dengan algoritma Dijkstra untuk graf dengan bobot edge positif. Jika graf tidak memiliki bobot edge negatif, sebaiknya menggunakan algoritma Dijkstra karena lebih cepat.

*Algoritma Bellman-Ford juga cocok digunakan jika tujuan Anda adalah mendeteksi keberadaan siklus berbobot negatif dalam graf.