

Nama : Febi Rahmadia Putri

Nim : 20220021

Prak. Desain Analisis dan Algoritma

Kode Python yang diberikan sudah benar dan dapat digunakan untuk mengetahui jarak terpendek dari node A ke node E pada grafik yang diberikan. Berikut adalah hasil dari kode Python tersebut:

```
import heapq

def dijkstra(graph, start, end):
    # Inisialisasi jarak ke setiap node dengan nilai infinity
    distances = {node: float('inf') for node in graph}
    # Jarak ke node awal diatur ke 0
    distances[start] = 0
    # Setiap node yang sudah dievaluasi disimpan di set visited
    visited = set()
    # Queue untuk menyimpan node yang akan dievaluasi berikutnya
    heap = [(0, start)]
    while heap:
        # Ambil node dengan jarak terdekat dari heap
        (distance, current) = heapq.heappop(heap)
        # Jika node tersebut belum dievaluasi sebelumnya
        if current not in visited:
            # Tandai node tersebut sebagai dievaluasi
            visited.add(current)
            # Loop melalui semua node yang terhubung dengan node tersebut
            for neighbor, weight in graph[current].items():
                # Hitung jarak baru ke node tersebut
```

```
# Jika jarak baru lebih pendek dari jarak sebelumnya ke node tersebut
if new_distance < distances[neighbor]:
    # Update jarak ke node tersebut
    distances[neighbor] = new_distance
    # Tambahkan node tersebut ke heap
    heapq.heappush(heap, (new_distance, neighbor))

# Kembalikan jarak terpendek ke node tujuan
return distances[end]

# Definisikan grafik
graph = {
    'A': {'B': 30, 'C': 50},
    'B': {'C': 10, 'D': 10},
    'C': {'D': 20, 'E': 30},
    'D': {'E': 50},
    'E': {}
}

# Hitung jarak terpendek dari A ke E
shortest_distance = dijkstra(graph, 'A', 'E')

# Cetak jarak terpendek
print(f"Jarak terpendek dari A ke E: {shortest_distance}")
```

Output yang dihasilkan adalah:

```
Jarak terpendek dari A ke E: 70
```

Penjelasannya : Jadi, jarak terpendek dari node A ke node E pada grafik yang diberikan adalah 90.