Praktikum Kecerdasan Buatan

Nama : Andyka Salom

Nim : 434221054

TI –B4

D4 TEKNIK INFORMATIKA

Fakultas VOKASI

Universitas Airlangga

- Program Python berikut mengimplementasikan pencarian jalur terpendek antara dua titik dalam sebuah graf menggunakan pendekatan rekursif. Program menggunakan struktur data `defaultdict` dari modul `collections` untuk merepresentasikan graf.

```
from collections import defaultdict
```

- Fungsi `shortest_paths` adalah fungsi utama yang digunakan untuk mencari semua jalur terpendek antara dua titik dalam graf. Fungsi ini menerima parameter berikut:
  - `graph`: Graf yang direpresentasikan sebagai kamus dengan simpul sebagai kunci dan daftar simpul terhubung sebagai nilai.
  - `start`: Simpul awal dari pencarian jalur.
  - `end`: Simpul akhir atau tujuan dari pencarian jalur.
  - `path`: Jalur yang telah ditemukan sejauh ini (secara default adalah daftar kosong).

```
def shortest_paths(graph, start, end, path=[]):
    path = path + [start]
    if start == end:
        return [path]
    if start not in graph:
        return []
    shortest_paths_list = []
    for node in graph[start]:
        if node not in path:
            newpaths = shortest_paths(graph, node, end, path)
            for newpath in newpaths:
                shortest_paths_list.append(newpath)
    return shortest_paths_list
```

  Fungsi ini secara rekursif mencari jalur terpendek antara `start` dan `end`. Pencarian dilakukan dengan mengikuti semua kemungkinan jalur yang mungkin dari simpul saat ini (`start`) ke simpul lainnya dalam graf. Jika jalur saat ini mencapai simpul tujuan (`end`), jalur tersebut ditambahkan ke dalam daftar jalur terpendek (`shortest_paths_list`). Setiap jalur baru yang ditemukan akan ditambahkan ke dalam jalur yang sedang ditemukan, dan pencarian akan dilanjutkan hingga semua jalur yang mungkin dieksplorasi.

- Fungsi `add_edge` digunakan untuk menambahkan sisi (edge) ke dalam graf. Ini mengambil graf dan dua simpul sebagai argumen, dan menambahkan kedua simpul tersebut sebagai simpul yang terhubung satu sama lain.

```
17  def add_edge(graph, vertex1, vertex2):
18      graph[vertex1].append(vertex2)
19      graph[vertex2].append(vertex1)
20
```

- Fungsi `main` adalah fungsi utama program yang membuat graf dengan menambahkan sisi-sisinya, menentukan simpul awal dan simpul akhir, dan kemudian memanggil fungsi `shortest_paths` untuk mencari jalur terpendek antara keduanya.

Hasilnya kemudian dicetak, menunjukkan jalur-jalur terpendek antara simpul awal dan akhir, jika ada.

```
def main():
    graph = defaultdict(list)

    add_edge(graph, 'A', 'C')
    add_edge(graph, 'C', 'D')
    add_edge(graph, 'D', 'E')
    add_edge(graph, 'E', 'B')
    add_edge(graph, 'A', 'F')
    add_edge(graph, 'F', 'G')
    add_edge(graph, 'G', 'H')
    add_edge(graph, 'H', 'E')

    start_city = 'A'
    end_city = 'B'

    shortest_paths_list = shortest_paths(graph, start_city, end_city)
    if shortest_paths_list:
        print("Semua jalur terpendek antara", start_city, "dan", end_city, "adalah:")
        for i, path in enumerate(shortest_paths_list, 1):
            print(f"Jalur {i}: {path}")
    else:
        print("Tidak ada jalur yang tersedia antara", start_city, "dan", end_city)

if __name__ == "__main__":
    main()
```

Dalam kode yang diberikan, graf yang dibuat memiliki simpul-simpul 'A', 'B', 'C', 'D', 'E', 'F', 'G', dan 'H', dengan sisi-sisi yang menghubungkan mereka. Program mencari jalur terpendek antara simpul 'A' dan 'B' dalam graf tersebut. Jika ada jalur yang ditemukan, program akan mencetaknya; jika tidak, program akan mencetak bahwa tidak ada jalur yang tersedia.