**Studi Kasus**

Kita akan membuat sistem pemetaan untuk kota kecil yang memiliki beberapa tempat penting. Tempat-tempat ini akan direpresentasikan sebagai simpul (node) dalam graf, dan jalan di antara mereka akan menjadi sisi (edge). Kita ingin:

1. Menemukan semua tempat yang dapat dijangkau dari tempat tertentu (traversals).
2. Menemukan jalur terpendek dari satu tempat ke tempat lain.

**Tempat yang akan direpresentasikan**

* A: Rumah Sakit
* B: Sekolah
* C: Taman
* D: Mall
* E: Stasiun

**Koneksi antar tempat**

* A ke B (5)
* A ke C (10)
* B ke D (2)
* C ke D (1)
* D ke E (3)
* B ke E (7)

**Penjelasan Kode**

1. **Kelas Graph**:
   * Menyimpan adjacency list untuk graf.
   * Metode add\_edge untuk menambahkan sisi antara dua node.
2. **Metode BFS**:
   * Memulai dari simpul start.
   * Menggunakan antrean untuk mengunjungi setiap simpul.
   * Menyimpan simpul yang dikunjungi dalam traversal\_order.
3. **Metode Dijkstra**:
   * Menggunakan heap untuk menemukan jarak terpendek.
   * Menyimpan jarak terpendek dalam dictionary distances.
   * Mengupdate jarak jika ditemukan jalur yang lebih pendek dan menambahkannya ke heap.
   * Mengembalikan jarak dan pohon jalur terpendek.

**Output**

* **Traversal BFS**: Menunjukkan urutan simpul yang dikunjungi.
* **Jarak terpendek**: Menampilkan jarak dari simpul awal ke semua simpul lainnya.
* **Pohon jalur terpendek**: Menunjukkan koneksi antara simpul dalam jalur terpendek.

Dengan implementasi ini, kita dapat mengeksplorasi graf dan menemukan jalur terpendek dengan mudah.