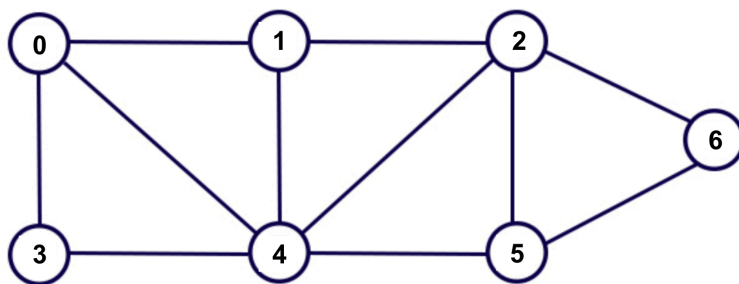


# Individual Project

## Soal 1 - Breadth First Search (BFS) Traversal

Seorang wisatawan ingin mengunjungi 5 tempat wisata (diberi label 0–4) sehingga total jarak yang ditempuh seminimal mungkin untuk menghubungkan seluruh tempat wisata (bukan rute pulang-pergi, melainkan jaringan koneksi minimal). Anda diminta membantu wisatawan tersebut dengan menulis program Python untuk membangun *Minimum Spanning Tree* (MST) dengan algoritma Kruskal. Data tempat wisata harus disimpan dalam *weighted undirected graph* dengan 5 *vertices* dan *weighted edge* yang merepresentasikan jarak antar tempat wisata dalam satuan kilometer (km).



0 - JAKARTA  
1 - SEMARANG  
2 - SURABAYA  
3 - BANDUNG  
4 - YOGYAKARTA  
5 - MALANG  
6 - DENPASAR

## Deskripsi

- Terdapat 7 kota, diberi nomor dari 0 hingga 6.
- Jarak antar pasangan tempat wisata harus disimpan dalam *list of tuple*  $(u, v)$ , di mana  $u$  dan  $v$  adalah nomor kota.
- Graph bersifat *undirected* dan *unweighted* di mana edge  $(u, v)$  sama dengan  $(v, u)$ .
- Tujuan: Lakukan BFS traversal dan print titik yang dilewati dengan menggantinya dengan nama kotak.

## Spesifikasi Input

- Pilih salah satu:
  - Membuat graph dengan memanggil *function* `add_edge(u, v)` satu per satu. (tidak ada tambahan point)
  - Membuat graph dengan memanggil *function* `add_edge(u, v)` dari *list of tuple*. (point: 5)
  - Membuat graph dengan memanggil *function* `add_edge(u, v)` dari *list of tuple* tetapi *list of tuple* dihasilkan dari *function* `read_file()` yang membaca data dari text file. (point: 10).

## Output

```
Enter user: JAKARTA
```

```
BFS Traversal:
```

```
JAKARTA SEMARANG BANDUNG YOGYAKARTA SURABAYA MALANG DENPASAR
```

```
Enter user: SEMARANG
```

```
BFS Traversal:
```

```
SEMARANG JAKARTA SURABAYA YOGYAKARTA BANDUNG MALANG DENPASAR
```

## Rubrik Penilaian

- |  |                   |
|--|-------------------|
| 1. Menggunakan <i>function</i> BFS yang disediakan dengan benar                                | 20 points         |
| 2. Melengkapi bagian main dengan benar dan efisien:  | 20 points         |
| 3. Teknik pembuatan <i>graph</i> dengan memanggil <i>function</i> <code>add_edge(u,v)</code> : |                   |
| a. Tanpa menggunakan <i>list of tuple</i> :  | 0 points          |
| b. Menggunakan <i>list of tuple</i> tetapi tidak menggunakan <i>text file</i> :                | 5 points          |
| c. Menggunakan <i>list of tuple</i> dengan membaca dari <i>text file</i> :                     | 10 points         |
| 4. Menghasilkan <i>output</i> daftar BSF traversal:  | 50 points         |
| <b>Total</b>   | <b>100 points</b> |

--- oOo ---