## Pertanyaan

### 2.1.3 Pertanyaan Percobaan

1. Sebutkan beberapa jenis (minimal 3) algoritma yang menggunakan dasar Graph, dan apakah kegunaan algoritma-algoritma tersebut?

**Jawab :**

- **Algoritma Floyd**, untuk menentukan adanya jalur pada Graph
- **Algoritma Brent**, untuk menentukan adanya jalur pada Graph
- **Algoritma Hopcroft-Karp**, untuk penjodohan maksimum
- **Algoritma Hungaria**, untuk penjodohan sempurna

2.Pada class Graph terdapat array bertipe LinkedList, yaitu LinkedList list[]. Apakah tujuan pembuatan variabel tersebut?

**Jawab :**

Tujuannya ialah untuk memanggil method **Linkedlist** dan membuat objek bernama **list** yang gunanya untuk mengintansiasi list yang berupa vertex pada linked list tersebut.

3.Apakah alasan pemanggilan method **addFirst()** untuk menambahkan data, bukan method add jenis lain pada linked list ketika digunakan pada method addEdge pada **class Graph?**

**Jawab :**

alasan pemanggilan method **addFirst()** untuk menambahkan data, bukan method add jenis lain pada linked list ketika digunakan pada method addEdge pada **class Graph?** ialah untuk mengenalkan vertex tersebut dan koneksinya

4.Bagaimana cara mendeteksi prev pointer pada saat akan melakukan penghapusan suatu edge pada graph?

**Jawab :**

Cara mendeteksi prev pointer pada saat akan melakukan penghapusan suatu edge pada graph, yakni ketika `i == destination`, maka akan dilihat source dari i. Jadi, jika vertex lebih besar dari **"i"** dan **destination** sama dengan **"i"**, maka edge akan secara otomatis dihapus.

5.Kenapa pada praktikum 2.1.1 langkah ke-12 untuk menghapus path yang bukan merupakan lintasan pertama kali menghasilkan output yang salah? Bagaimana solusinya?

```
graph.removeEdge(1,3);
grap.printGraph();
```

**Jawab :**

yakni dengan mengubah isi dari vertex dan edges serta source dan destintion yang ada.

### 2.2.3 Pertanyaan Percobaan

1.Apakah perbedaan degree/derajat pada *directed* dan *undirected graph?*

**jawab :**

perbedaan degree/derajat pada **directed** dan **undirected graph?** adalah :

- jika **directed** `degreeIn` dan `degreeOut`nya berbeda, tetapi pada **undirected graph?** `degreeIn` dan `degreeOut`nya sama.
- Pada **directed** degree dapat mempengaruhi bobot pada edge antar vertex, misalnya pada X ke Y bobotnya 4, tetapi Y ke X bobotnya belum tentu sama 4. Sedangkan pada **undirected graph?** degree tidak dapat mempengaruhi bobot.

2.Pada implementasi graph menggunakan adjacency matriks. Kenapa jumlah vertices harus ditambahkan dengan 1 pada indeks array berikut?

```
  public graphArray(int v){
    vertices = v;
    twoD_array = new int[vertices + 1][vertices + 1];
  }
```

**Jawab :**
Karena indexnya dimulai dari 0, maka vertexnya perlu ditambahkan 1.

3.Apakah kegunaan method **getEdge()**?
**jawab :**
Kegunaan method **getEdge()** ialah untuk menampilkan suatu lintasan yang diperlukan.

4.Termasuk jenis graph apakah uji coba pada praktikum 2.2?
**Jawab :**
Termasuk jenis **Directed Graph**.

5.Mengapa pada method main harus menggunakan *try-catch Exception* ?
**Jawab :**
Pada method main harus menggunakan *try-catch Exception* supaya dapat menangani error pada saat program dijalankan.