

Worksheet pertemuan 10 – 1

Algoritma dan Struktur Data

Graf

NIM : 23523195

Nama : Bintang Rizqi Anugrah

A. Membuat Folder Untuk Menyimpan Hasil Praktikum

1. Siapkan folder kosong dengan nama menggunakan NIM masing-masing. Jika folder NIM pada pertemuan sebelumnya mau dimanfaatkan, jangan lupa pindahkan dulu isinya ke folder lain sebagai arsip.
2. Folder ini akan dijadikan tempat untuk menyimpan semua pdf dari worksheet ini beserta fail praktikum lainnya.

B. Membuat Class Edge

1. Buat sebuah class dengan nama **Edge**
2. Kemudian salin tempel kode pogram di bawah ini

```
public class Edge {  
  
    int dest, weight;  
  
    public Edge(int dest, int weight){  
        this.dest = dest;  
        this.weight = weight;  
    }  
  
}
```

C. Membuat Class Graf

1. Siapkan class ArrayList yang dulu pernah dibuat, kita akan menggunakannya lagi untuk kelas Graf ini
2. Buat sebuah class dengan nama **Graph**
3. Kemudian salin tempel kode pogram di bawah ini
4. Perhatikan di class tersebut ada 3 method addEdge yang berbeda, pelajari apa bedanya.

```
public class Graph {  
  
    private ArrayList<ArrayList<Edge>> adj = new ArrayList< ArrayList<Edge>>  
>();  
  
    public Graph(int N) {  
        for(int i=0;i<N;i++){  
            adj.add(new ArrayList<Edge>());  
        }  
    }  
  
}
```

```

// method utama addEdge
public void addEdge(int source, int dest, int weight, boolean directed){
    ArrayList<Edge> adjU = (ArrayList<Edge>) adj.get(source);
    adjU.add(new Edge(dest, weight));
    if(directed==false){
        ArrayList<Edge> adjV = (ArrayList<Edge>) adj.get(dest);
        adjV.add(new Edge(source, weight));
    }
}

// untuk unweighted directed graph
public void addEdge(int u, int v, boolean directed){
    this.addEdge(u, v, 1, directed);
}

// untuk unweighted graph
public void addEdge(int u, int v){
    this.addEdge(u, v, 1, false);
}

public void draw(){
    for(int i=0;i<adj.size();i++){
        System.out.print(i+ " : ");
        ArrayList<Edge> adjacent = (ArrayList<Edge>) adj.get(i);
        for(int j=0;j<adjacent.size();j++){
            Edge v = (Edge) adjacent.get(j);
            System.out.print("(" +v.dest+", " +v.weight+" )");
        }
        System.out.println("");
    }
}
}

```

D. Membuat dan Menjalankan Main Method

1. Silakan buat sebuah main method class dengan nama **Main**
2. Kemudian salin tempel kode program di bawah ini

```

public static void main(String[] args) {
    Graph mygraph = new Graph(5);
    mygraph.addEdge(0, 1);
    mygraph.addEdge(1, 2);
    mygraph.addEdge(3, 4, 12, false);
    mygraph.addEdge(0, 4, true);
}

```

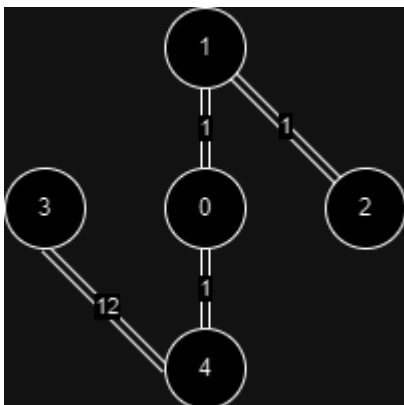
3. Jalankan *main method* tersebut. Tambahkan baris kode program untuk memanggil method `draw()` pada akhir program. Tuliskan apa output dari program di bawah

```

PS C:\Users\ROG STRIX\Documents\ASD\Week_10.1\Graf>
\ROG STRIX\Documents\ASD\Week_10.1\Graf\bin' 'Main'
0 : (1, 1) (4, 1)
1 : (0, 1) (2, 1)
2 : (1, 1)
3 : (4, 12)
4 : (3, 12)
PS C:\Users\ROG STRIX\Documents\ASD\Week_10.1\Graf>

```

4. Ilustrasikan graf yang telah dibuat tersebut di bawah. Gambar bisa menggunakan tulisan tangan yang difoto atau dengan software ilustrator seperti <https://app.diagrams.net/>



***Catatan**

- Jangan lupa simpan juga fail worksheet ini (yang sudah diisi) sebagai fail pdf di folder NIM anda.
- Sertakan juga fail **Main.java** di dalam folder yang Anda gunakan
- Kompres folder ini sebagai fail ZIP kemudian kumpulkan di classroom atau ruang pengumpulan lain di kelas masing-masing.