

# **MODUL 7 PRAKTIKUM**

## **STRUKTUR DATA**

**C/C++**

**Oleh:**

**Hermawan ST, M.Kom.**

**Informatika 2023**

**Uniersitas Trunojoyo Madura**

## Adjacency List dalam Teori Graph

### ❖ Pengertian Graph

Dalam bidang ilmu komputer, sebuah graf dapat dinyatakan sebagai sebuah struktur data, atau secara spesifik dinamakan sebagai ADT(abstract data type) yang terdiri dari kumpulan simpul dan sisi yang membangun hubungan antar simpul. Dengan kata lain graph adalah kumpulan noktah (simpul) di dalam bidang dua dimensi yang dihubungkan dengan sekumpulan garis (sisi). Graph dapat digunakan untuk merepresentasikan objek - objek diskrit dan hubungan antara objek-objek tersebut. Ada beberapa istilah yang biasa digunakan pada suatu graph:

#### a. Vertex (simpul / node)

Sebuah graph dibentuk dari kumpulan titik yang dihubungkan dengan garis -garis. Titik titik tersebut disebut vertex.

#### b. Edge (busur)

Merupakan garis penghubung antara dua vertex.

#### c. Adjacent (bertetangga)

Pada graph tak berarah (indirected graph) dua buah vertex / vertex disebut adjacent jika ada edge yang menghubungkan dua buah vertex. Sedangkan pada graph berarah (directed graph) sebuah vertex a disebut adjacent dengan vertex b jika ada vertex dari b ke a.

#### d. Weight

Apabila setiap edge mempunyai sebuah nilai (dapat berupa jarak, waktu atau biaya) yang menyatakan hubungan antara kedua buah vertex maka edge tersebut dikatakan memiliki bobot. Graph yang memiliki bobot dapat disebut sebagai graph berbobot atau weighted graph.

#### e. Path (lintasan)

Path adalah serangkaian vertex yang berbeda yang adjacent secara berturut turut dari vertex satu ke vertex berikutnya.

#### f. Direct (arah)

Merupakan arah suatu graph. Pada graph berarah (directed graph atau digraph) urutan vertex memiliki arti

### ❖ Pengertian Adjacency List

Adjacency List Dalam teori graf, adjacency list merupakan bentuk representasi dari seluruh sisi atau busur dalam suatu graf sebagai suatu senarai. Simpul-simpul yang dihubungkan sisi atau busur tersebut dinyatakan sebagai simpul yang saling terkait.

### ❖ Penerapan dari Adt Graph

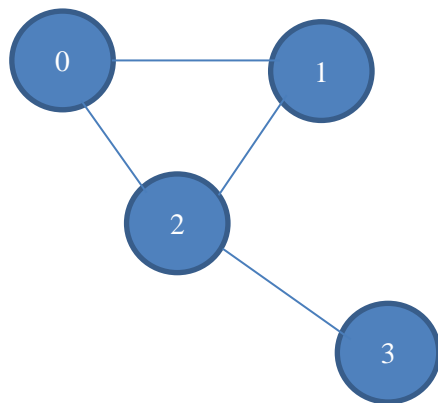
Dalam bidang ilmu komputer, sebuah graph dapat dinyatakan sebagai sebuah struktur data, atau secara spesifik dinamakan sebagai ADT(abstract datatype) yang terdiri dari kumpulan vertex dan sisi yang membangun hubungan antarvertex. Konsep ADT graph ini merupakan turunan konsep graph dari bidang kajian matematika.

Secara umum terdapat dua macam representasi dari struktur data graph yang dapat diimplementasi. Pertama disebut adjacency list, dan diimplementasi dengan menampilkan masing-masing vertex sebagai sebuah struktur data yang mengandung senarai dari semua vertex yang saling berhubungan. Yang kedua adalah representasi berupa adjacency matrix dimana baris dan kolom dari matriks (jika dalam konteks implementasi berupa senarai dua dimensi) tersebut merepresentasikan vertex awal dan vertex tujuan dan sebuah entri di dalam senarai yang menyatakan apakah terdapat sisi di antara kedua vertex tersebut.

Graph tidak berarah dalam representasi adjd matrik:

$$V = \{0,1,2,3\}$$

$$E = \{\{0,1\},\{0,2\},\{1,2\},\{2,3\}\}$$



Membuat code Adjacency List

```
#include <iostream>
#include <list>
using namespace std;

class Graph{
    int V;
    //array of list
    list<int> *l;
public:
    Graph(int V){
        this->V = V;
        l = new list<int> [V];
    }
    void addEdge(int x, int y){
        l[x].push_back(y);
        l[y].push_back(x);
    }
    void printGraph(){
        for (int i=0;i<V;i++){
            cout << "Vertex " << i << "->";
            for(int num : l[i]){
                cout << num << ",";
            };
            cout << endl;
        }
    }
};

int main (){
    Graph g(4);
    g.addEdge(0,1);
    g.addEdge(0,2);
    g.addEdge(2,3);
    g.addEdge(1,2);
    g.printGraph();
    return 0;
}
```

```
Vertex 0->1,2,  
Vertex 1->0,2,  
Vertex 2->0,3,1,  
Vertex 3->2,
```

## 1.2. Instruksi Praktikum

1. Buat Program sesuai dengan tutorial
2. Lakukan pengujian script sesuai dengan tutorial

## 1.3. Tugas Praktikum

1. Buatlah code untuk membuat Adt Graph tidak berarah!  
 $V = \{0,1,2,3,4,5\}$   
 $E = \{\{0,1\},\{0,5\},\{1,2\},\{1,5\},\{2,4\},\{3,4\},\{3,5\}\}$