

LAPORAN UAS STRUKTUR DATA



Rendi Nicolas Mahendra - 21091397071

**FAKULTAS VOKASI
PRODI D4 MANAJEMEN INFORMATIKA
UNIVERSITAS NEGERI SURABAYA
TAHUN AJARAN 2021/2022**

- Undirect Graph

```
// Program C++ yang merepresentasi undirected dan weighted graph
// ke dalam bentuk adjacency-list.
#include <bits/stdc++.h>
#include <iostream>
using namespace std;

// function untuk menambahkan edge dan deklarasi variabel untuk
// edge
void addEdge(vector <pair<int, int> > adj[], int u, int v, int wt)
{
    adj[u].push_back(make_pair(v, wt));
    adj[v].push_back(make_pair(u, wt));
}

// Function untuk merubah graph menjadi adjacency list
void printGraph(vector<pair<int,int> > adj[], int V) {

    // deklarasi variabel
    int v, w;

    // perulangan bersarang untuk adjacency-list
    // perulangan untuk setiap baris nya
    for (int u = 1; u < V; u++) {
        cout << u;

        // perulangan untuk menampilkan nilai edge dalam array
        for (auto it = adj[u].begin(); it!=adj[u].end(); it++) {
            v = it->first;
            w = it->second;

            // tampilan dari struktur program
            cout << " => " << "[" << v << "," << w << "]"";
        }
        cout << "\n";
    }
}
```

```

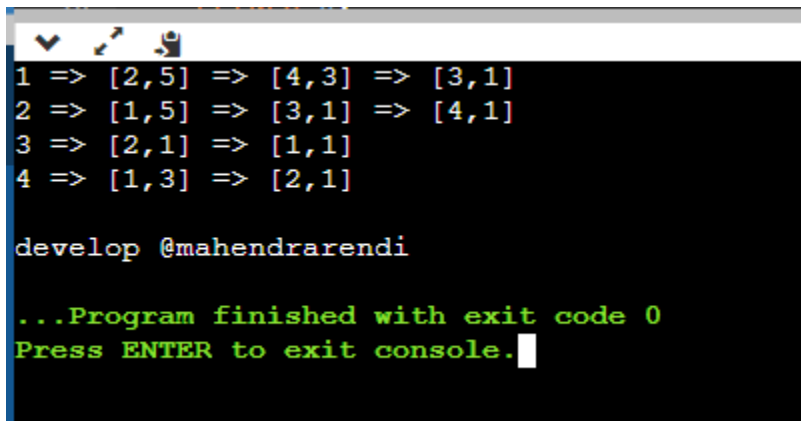
// function untuk adjacency-list
int main() {
    int V = 5;
    vector<pair<int, int> > adj[V];
    //menambahkan nilai edge
    addEdge(adj, 1, 2, 5);
    addEdge(adj, 2, 3, 1);
    addEdge(adj, 4, 1, 3);
    addEdge(adj, 2, 4, 1);
    addEdge(adj, 3, 1, 1);

    // mencetak graph
    printGraph(adj, V);

    cout << "\ndevelop @mahendrarendi";
    return 0;
}

```

HASIL DEBUG



```

1 => [2,5] => [4,3] => [3,1]
2 => [1,5] => [3,1] => [4,1]
3 => [2,1] => [1,1]
4 => [1,3] => [2,1]

develop @mahendrarendi

...Program finished with exit code 0
Press ENTER to exit console.

```

- **Algoritma Naga Sakti**

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    // deklarasi variabel
    int jumlah;

    cout<<"* Jumlah kota yang berada di kerajaan Britan : "<<
endl;
    cin>>jumlah;

    char kota[jumlah];
    char jarak_kota[jumlah];
    char sisi_kota[jumlah];

    // perulangan untuk banyak_kota Britan
    for(int i = 1; i <= jumlah; ++i) {
        cout << "Masukkan nama kota ke-" << i << " : ";
        cin >> kota[i];
    }

    int banyak_kota = sizeof(kota)/sizeof(kota[0]);
    // for(int i = 1; i <= banyak_kota; ++i) cout << kota[i];

    //deklarasi graph
    //menampilkan graph yang terjadi
    cout<< "* Sisi-sisinya adalah : " << endl << endl;
    for(int i = 1; i <= jumlah; i++){
        for(int j = 1; j <= jumlah; j++){
            std::cout<<kota[i]<<kota[j]<<" ";
            sisi_kota[i] = i, j;
        }
    }
}
```

```

int banyak_sisi = sizeof(sisi_kota)/sizeof(sisi_kota[0]);
cout<< endl << "SISI KOTA ";
for(int i = 1; i <= banyak_sisi; ++i) cout << sisi_kota[i] <<
endl;

cout<< endl << endl << "* Panjang jalan antar kota : " <<
endl;
cout<<"* seluruh jalan yang ada dalam kerajaan britan dan
panjang jalannya : "<< endl;
for(int i = 1; i <= jumlah; i++){
    for(int j = 1; j <= jumlah; j++)
    {
        std::cout<< "Panjang " << kota[i] << kota[j] << " : "
<< endl;

        // cin >> jarak_kota[k];
        cin >> jarak_kota[i];

cout<<" ("<<kota[i]<<","<<kota[j]<<","<<jarak_kota[i]<<") ";
    }
}
// cout<<"panjang "<<kota1<<" ke "<<kota2<< ": "; cin>>
hasil1;

//menampilkan tempat pedagang berada
cout<<"* kota tempat pedagang sekarang berada : "<<endl<<endl;
cout<<kota[0];

cout<<endl<<endl;

//menampilkan kota yang diserang naga
cout<<"* kota yang diserang naga : "<<endl<<endl;
cout<<kota[2];

cout<<endl<<endl;

//menampilkan kota yang terdapat kastil
cout<<"* kota yang memiliki kastil : "<<endl<<endl;
int b = sizeof(kota) / sizeof(kota[0]);
int kota_terakhir = kota[b-1];

```

```

cout<<kota_terakhir;

cout<<endl<<endl;

//menampilkan vertex tercepat untuk selamat
cout<<"* jalur yang paling cepat ditempuh : "<<endl<<endl;
cout<<kota[0]<<"-"<<kota[1]<<"-"<<kota[2]<<endl;

cout<<endl<<endl;

//total edge yang harus ditempuh
cout<< "* dengan jarak : "<<endl<<endl;
int a = sizeof(jarak_kota) / sizeof(jarak_kota[0]);
int jarakkota_terakhir = jarak_kota[a-1];
cout<<jarak_kota[0]+jarakkota_terakhir<<endl<<endl;

cout<<"\ndevelop @mahendrarendi";
return 0;
}

```

HASIL DEBUG

```

* Jumlah kota yang berada di kerajaan Britan :
3
Masukkan nama kota ke-1 : 3
Masukkan nama kota ke-2 : 2
Masukkan nama kota ke-3 : 1
* Sisi-sisinya adalah :

33 32 31 23 22 21 13 12 11
SISI KOTA

* Panjang jalan antar kota :
* seluruh jalan yang ada dalam kerajaan britan dan panjang jalannya :
Panjang 33 :
4
(3,3,4) Panjang 32 :
5
(3,2,5) Panjang 31 :
2
(3,1,2) Panjang 23 :
3
(2,3,3) Panjang 22 :
4
(2,2,4) Panjang 21 :
5
(2,1,5) Panjang 13 :
5
(1,3,5) Panjang 12 :
2
(1,2,2) Panjang 11 :
2

```

```

(1,1,1) kota tempat pedagang berdagang berada :
?

* kota yang diserang naga :
3

* kota yang memiliki kastil :
51

* jalur yang paling cepat ditempuh :
? 2-3

* dengan jarak :
167

develop @mahendrarendi

```