

LAPORAN PRAKTIKUM
MODUL 14
GRAPH



Nama :

Muhammad Rifqi Al Baqi Ananta (2311104005)

Dosen :

Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. GUIDED

1. Soal 1

Code:

```
#include <iostream>
#include <vector>
#include <string>
#include <iomanip>
using namespace std;

class PengolahanJarak {
private:
    vector<string> namaSimpul;
    vector<vector<int>> matriksJarak;

public:
    void inputSimpul() {
        int jumlahSimpul;
        cout << "Masukkan jumlah simpul: ";
        cin >> jumlahSimpul;

        namaSimpul.resize(jumlahSimpul);
        cout << "Masukkan nama simpul:\n";
        for (int i = 0; i < jumlahSimpul; i++) {
            cout << "Simpul ke-" << (i + 1) << ": ";
            cin >> namaSimpul[i];
        }
    }

    void hitungJarak() {
        int jumlahSimpul = namaSimpul.size();
        matriksJarak.resize(jumlahSimpul, vector<int>(jumlahSimpul, 0));

        cout << "Masukkan jarak antar simpul:\n";
        for (int i = 0; i < jumlahSimpul; i++) {
            for (int j = 0; j < jumlahSimpul; j++) {
                if (i == j) {
                    matriksJarak[i][j] = 0;
                } else {
                    cout << "Jarak dari simpul " << namaSimpul[i]
                        << " ke simpul " << namaSimpul[j] << ": ";
                    cin >> matriksJarak[i][j];
                }
            }
        }
    }

    void tampilkanMatriksJarak() {
        cout << "\nMatriks Jarak Antar Simpul:\n";

        cout << setw(15) << " ";
        for (const auto& simpul : namaSimpul) {
            cout << setw(15) << simpul;
        }
        cout << endl;

        for (size_t i = 0; i < namaSimpul.size(); i++) {
            cout << setw(15) << namaSimpul[i];
            for (size_t j = 0; j < namaSimpul.size(); j++) {
                cout << setw(15) << matriksJarak[i][j];
            }
            cout << endl;
        }
    }
};

int main() {
    PengolahanJarak perhitungan;

    perhitungan.inputSimpul();
    perhitungan.hitungJarak();

    perhitungan.tampilkanMatriksJarak();

    return 0;
}
```

Output:

```
Masukkan jumlah simpul: 2
Masukkan nama simpul:
Simpul ke-1: BALI
Simpul ke-2: PALU
Masukkan jarak antar simpul:
Jarak dari simpul BALI ke simpul PALU: 3
Jarak dari simpul PALU ke simpul BALI: 4

Matriks Jarak Antar Simpul:
```

	BALI	PALU
BALI	0	3
PALU	4	0

2. Soal 2

Code:

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

class Graf {
private:
    int jumlahSimpul;
    vector<vector<int>> matriksKetetanggaan;

public:
    Graf(int simpul) {
        jumlahSimpul = simpul;
        matriksKetetanggaan.resize(simpul, vector<int>(simpul, 0));
    }

    void tambahSisi(int simpul1, int simpul2) {
        simpul1--;
        simpul2--;

        matriksKetetanggaan[simpul1][simpul2] = 1;
        matriksKetetanggaan[simpul2][simpul1] = 1;
    }

    void cetakMatriksKetetanggaan() {
        cout << "Adjacency Matrix:\n";
        for (int i = 0; i < jumlahSimpul; i++) {
            for (int j = 0; j < jumlahSimpul; j++) {
                cout << matriksKetetanggaan[i][j] << " ";
            }
            cout << endl;
        }
    }
};

int main() {
    int jumlahSimpul, jumlahSisi;

    cout << "Masukkan jumlah simpul: ";
    cin >> jumlahSimpul;

    Graf graf(jumlahSimpul);

    cout << "Masukkan jumlah sisi: ";
    cin >> jumlahSisi;

    cout << "Masukkan pasangan simpul:\n";
    for (int i = 0; i < jumlahSisi; i++) {
        int simpul1, simpul2;
        cin >> simpul1 >> simpul2;
        graf.tambahSisi(simpul1, simpul2);
    }

    graf.cetakMatriksKetetanggaan();

    return 0;
}
```

Output:

```
Masukkan jumlah simpul: 4
Masukkan jumlah sisi: 4
Masukkan pasangan simpul:
1 2
1 3
2 4
3 4
Adjacency Matrix:
0 1 1 0
1 0 0 1
1 0 0 1
0 1 1 0
```

II. PENJELASAN

Graph adalah struktur data yang digunakan untuk memodelkan hubungan antar elemen, yang disebut simpul (vertices), dengan sisi (edges) yang menghubungkannya. Setiap sisi dapat memiliki arah (pada graf berarah) atau tidak (pada graf tak berarah), dan dalam beberapa kasus, sisi juga dapat memiliki bobot yang menunjukkan biaya atau jarak antar simpul. Graph sering digunakan dalam berbagai aplikasi, seperti dalam representasi jaringan komputer, peta rute, dan masalah optimasi. Struktur ini dapat direpresentasikan dalam berbagai bentuk, seperti adjacency matrix atau adjacency list, yang masing-masing memiliki kelebihan dan kekurangan tergantung pada kebutuhan pemrograman atau perhitungan yang dilakukan..