

LAPORAN PRAKTIKUM

“LANGKAH PRAKTIKUM 10: GRAF”

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Matematika Diskrit yang di
ampu oleh:

Nur Rochmah Dyah PA, S.T., M.Kom



Disusun Oleh:

Mohammad Farid Hendianto 2200018401

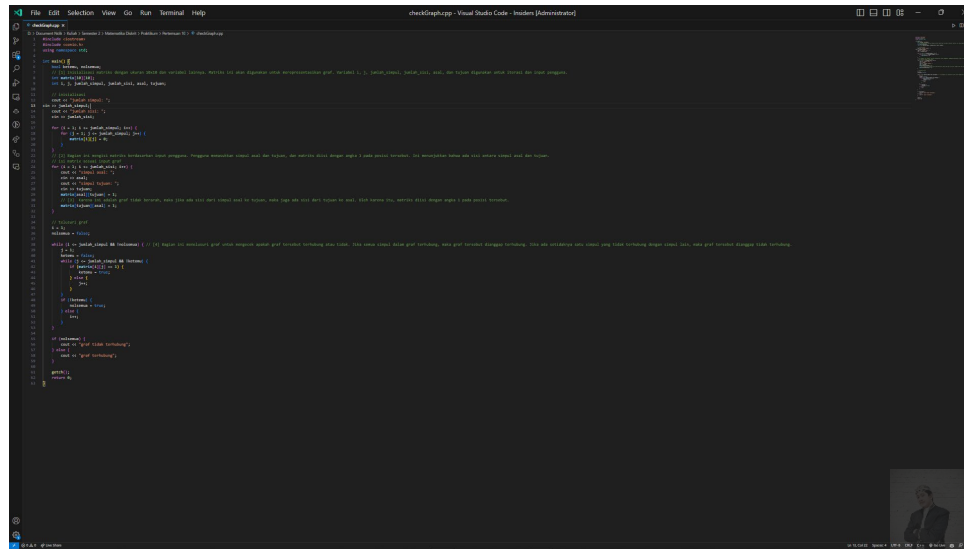
Selasa 12.00-13.30

PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

Berikut adalah kodingan di visual studio code:

Gambar 1 checkGraph.cpp. (Sumber: Penulis)

Berikut kodingan full sourcenya:



Gambar 2 full source checkGraph.cpp. (Sumber: Penulis)

Kode ini adalah program untuk mengecek apakah sebuah graf terhubung atau tidak. Graf yang terhubung adalah graf yang setiap simpulnya dapat dijangkau dari simpul lainnya. Program ini meminta input jumlah simpul dan jumlah sisi dari graf, kemudian meminta input simpul asal dan simpul tujuan untuk setiap sisi. Program akan menyimpan informasi graf dalam bentuk matriks adjacency, di mana setiap elemen matriks menunjukkan apakah ada sisi yang menghubungkan dua simpul atau tidak. Setelah itu, program akan menelusuri graf dengan mencari simpul yang dapat dijangkau dari simpul pertama. Jika semua simpul dapat dijangkau, maka graf dianggap terhubung. Jika tidak, maka graf dianggap tidak terhubung.

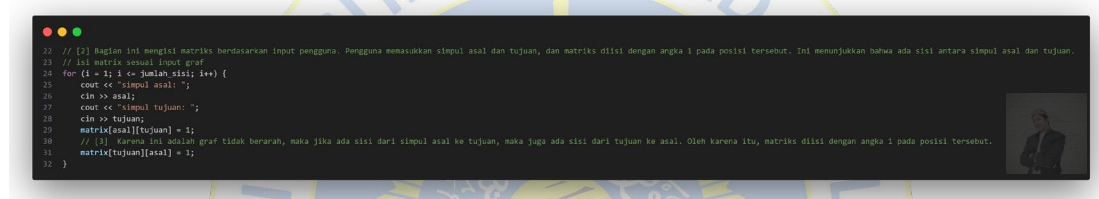
Program ini adalah implementasi dari struktur data graf menggunakan matriks adjacensi dalam bahasa pemrograman C++. Graf yang digunakan dalam program ini adalah graf tidak berarah, yang berarti bahwa setiap sisi menghubungkan dua simpul dan dapat dilalui dalam kedua arah.

Berikut adalah penjelasan lebih rinci tentang bagian-bagian kode:

- 1) Program ini dimulai dengan mendeklarasikan variabel-variabel yang diperlukan, termasuk matriks 10x10 yang akan digunakan untuk merepresentasikan graf.

- 2) Program kemudian meminta pengguna untuk memasukkan jumlah simpul dan sisi dalam graf. Setelah itu, matriks diinisialisasi dengan semua elemen bernilai 0, yang berarti bahwa tidak ada sisi antara simpul-simpul pada awalnya.
- 3) Program kemudian meminta pengguna untuk memasukkan sisi-sisi dalam graf. Untuk setiap sisi, pengguna memasukkan simpul asal dan tujuan, dan matriks diisi dengan angka 1 pada posisi yang sesuai. Karena graf ini adalah graf tidak berarah, matriks diisi dengan angka 1 pada posisi yang sesuai untuk sisi dari simpul tujuan ke asal juga.
- 4) Setelah matriks telah diisi, program menelusuri matriks untuk mengecek apakah graf tersebut terhubung atau tidak. Jika semua simpul dalam graf terhubung, maka graf tersebut dianggap terhubung. Jika tidak, maka graf tersebut dianggap tidak terhubung.
- 5) Program kemudian menampilkan pesan yang menunjukkan apakah graf tersebut terhubung atau tidak, dan kemudian berakhir.

Perlu dicatat bahwa dalam program ini, indeks matriks dan simpul graf dimulai dari 1, bukan 0, yang biasanya adalah standar dalam pemrograman.

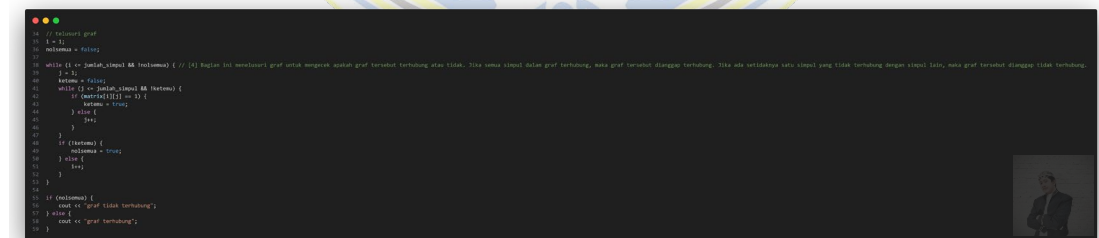


```

22 // [2] Bagian ini mengisi matriks berdasarkan input pengguna. Pengguna memasukkan simpul asal dan tujuan, dan matriks diisi dengan angka 1 pada posisi tersebut. Ini menunjukkan bahwa ada sisi antara simpul asal dan tujuan.
23 // Isi matriks sesuai input graf
24 for (i = 1; i <= jumlah_sisi; i++) {
25     cout << "simpul asal: ";
26     cin >> asal;
27     cout << "simpul tujuan: ";
28     cin >> tujuan;
29     matrix[asal][tujuan] = 1;
30     // [3] Karena ini adalah graf tidak berarah, maka jika ada sisi dari simpul asal ke tujuan, maka juga ada sisi dari tujuan ke asal. Oleh karena itu, matriks diisi dengan angka 1 pada posisi tersebut.
31     matrix[tujuan][asal] = 1;
32 }
    
```

Gambar 3 Mengisi simpul asal dan tujuan

Pada bagian ini, program mengisi matriks berdasarkan input pengguna. Pengguna diminta untuk memasukkan simpul asal dan tujuan, dan matriks diisi dengan angka 1 pada posisi tersebut. Ini menunjukkan bahwa ada sisi antara simpul asal dan tujuan. Kemudian, karena ini adalah graf tidak berarah, maka jika ada sisi dari simpul asal ke tujuan, maka juga ada sisi dari tujuan ke asal. Oleh karena itu, matriks diisi dengan angka 1 pada posisi tersebut.



```

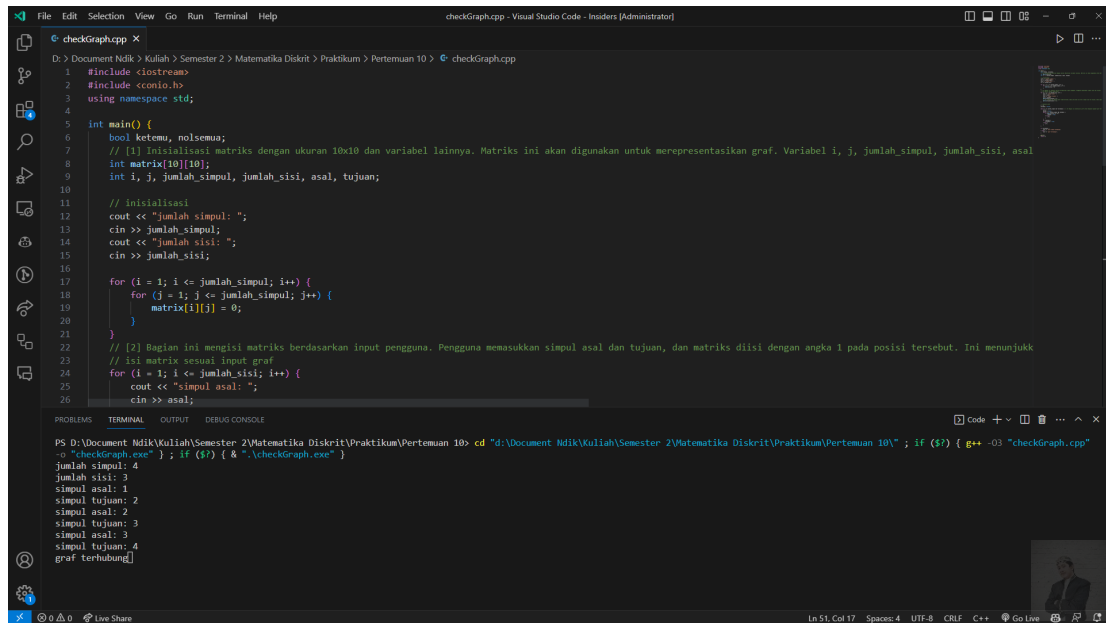
33 // [4] Bagian ini menelusuri graf untuk mengecek apakah graf tersebut terhubung atau tidak. Jika semua simpul dalam graf terhubung, maka graf tersebut dianggap terhubung. Jika ada setidaknya satu simpul yang tidak terhubung dengan simpul lain, maka graf tersebut dianggap tidak terhubung.
34 while (i <= jumlah_simpul && terhubung) { // [5] Bagian ini menelusuri graf untuk mengecek apakah graf tersebut terhubung atau tidak. Jika semua simpul dalam graf terhubung, maka graf tersebut dianggap terhubung. Jika ada setidaknya satu simpul yang tidak terhubung dengan simpul lain, maka graf tersebut dianggap tidak terhubung.
35     j = 1;
36     terhubung = false;
37     while (j <= jumlah_simpul && terhubung) {
38         if (matrix[i][j] == 1)
39             terhubung = true;
40         j++;
41     }
42     if (!terhubung)
43         terhubung = true;
44     i++;
45 }
46 if (!terhubung) {
47     cout << "graf tidak terhubung";
48 } else {
49     cout << "graf terhubung";
50 }
    
```

Gambar 4 Bagian proses pengecekan keterhubungan graf. (Sumber: Penulis)

Pada bagian kodingan tersebut, terdapat proses pengecekan keterhubungan graf. Pertama, program akan menelusuri graf dengan menggunakan perulangan while. Selama perulangan berlangsung, program akan mengecek apakah setiap simpul dalam graf terhubung atau tidak. Jika semua simpul dalam graf terhubung, maka graf tersebut dianggap terhubung. Jika ada setidaknya satu simpul yang tidak terhubung dengan simpul lain, maka graf tersebut dianggap

tidak terhubung. Proses pengecekan keterhubungan graf dilakukan dengan menggunakan matriks adjacency. Jika terdapat simpul yang terhubung dengan simpul lain, maka nilai pada matriks adjacency akan bernilai 1. Jika tidak, maka nilai pada matriks adjacency akan bernilai 0. Pada bagian kodingan tersebut, program akan mengecek nilai pada matriks adjacency untuk setiap simpul dalam graf. Jika nilai pada matriks adjacency bernilai 1, maka simpul tersebut terhubung dengan simpul lain. Jika tidak, maka program akan mengecek simpul berikutnya. Jika semua simpul dalam graf terhubung, maka program akan menampilkan pesan "graf terhubung". Jika tidak, maka program akan menampilkan pesan "graf tidak terhubung".

Berikut adalah outputnya:



```

1 #include <iostream>
2 #include <conio.h>
3 using namespace std;
4
5 int main() {
6     bool ketemu, nolsemua;
7     // [1] Inisialisasi matriks dengan ukuran 10x10 dan variabel lainnya. Matriks ini akan digunakan untuk merepresentasikan graf. Variabel i, j, jumlah_simpul, jumlah_sisi, asal
8     int matrix[10][10];
9     int i, j, jumlah_simpul, jumlah_sisi, asal, tujuan;
10
11     // inisialisasi
12     cout << "jumlah simpul: ";
13     cin >> jumlah_simpul;
14     cout << "jumlah sisi: ";
15     cin >> jumlah_sisi;
16
17     for (i = 1; i <= jumlah_simpul; i++) {
18         for (j = 1; j <= jumlah_simpul; j++) {
19             matrix[i][j] = 0;
20         }
21     }
22
23     // [2] Bagian ini mengisi matriks berdasarkan input pengguna. Pengguna memasukkan simpul asal dan tujuan, dan matriks diisi dengan angka 1 pada posisi tersebut. Ini menunjukk
24     // isi matriks sesuai input graf
25     for (i = 1; i <= jumlah_sisi; i++) {
26         cout << "simpul asal: ";
27         cin >> asal;
28     }
29 }

```

```

PS D:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 10> cd "d:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 10\" ; if ($?) { g++ -O3 "checkGraph.cpp"
-o "checkGraph.exe" } ; if ($?) { & ".\checkGraph.exe" }
jumlah simpul: 4
jumlah sisi: 3
simpul asal: 1
simpul tujuan: 2
simpul asal: 2
simpul tujuan: 3
simpul asal: 3
simpul tujuan: 4
graf terhubung

```

