

# **LAPORAN STRUKTUR DATA**

## **“PROGRAM GRAF TIDAK BERARAH”**



**Disusun oleh :**

Muhammad Fikri Ramadhana (21091397033)

**KELAS 2021 A**

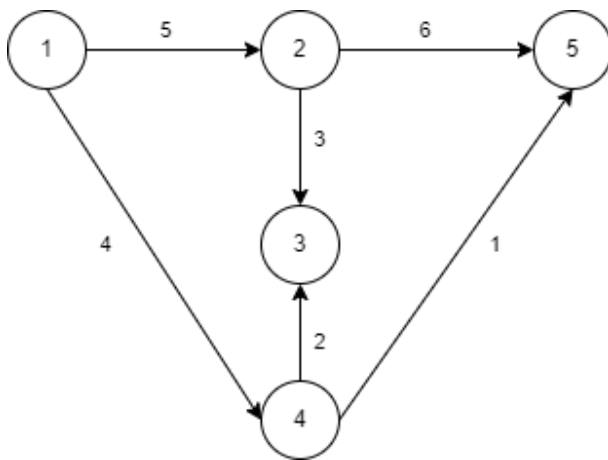
**PRODI D4 MANAJEMEN INFORMATIKA**

**UNIVERSITAS NEGERI SURABAYA**

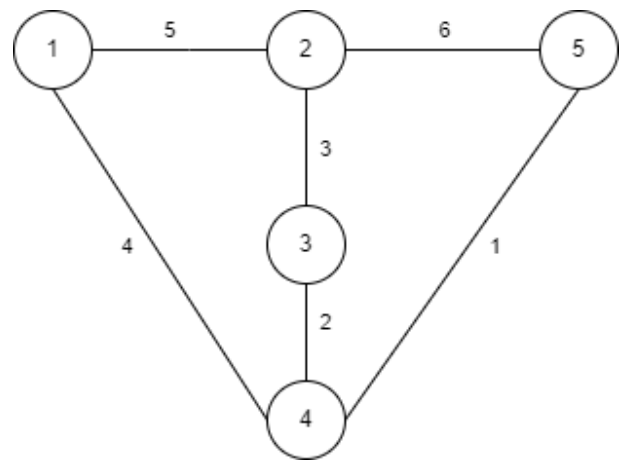
**2021/2022**

## A. Hasil Analisis dan Identifikasi

Graph merupakan sekumpulan dari simpul (*vertices*) dan busur (*edge*) yang dapat dinotasikan dengan  $G = (V,E)$ . Dengan sebuah garis (*adjacent*) yang menghubungkan antara kedua simpul yang dinyatakan sebagai  $u$  dan  $v$ . Pada graph berarah (*directed graph*), setiap garis memiliki arah dari simpul  $u$  mengarah ke simpul  $v$  dengan notasi  $u \rightarrow v$ . Kemudian pada graph tak berarah (*undirected graph*), setiap garis tidak memiliki arah yang menentukan diantara garis kedua simpul  $u$  dan  $v$  dengan notasi  $u < - > v$ . Implementasi dari tipe graph yaitu dapat menggambarkan pemetaan dari jalur jaringan dan peta jalan.



Gambar 1.1 Graph berarah



Gambar 1.2 Graph tak berarah

Namun program yang dibuat berikut menggunakan konsep *undirected graph* dengan representasi *adjacency matrix* input nilai simpul (*vertices*) dan busur (*edge*). Berikut tampilan kodingan yang telah dikerjakan :

```
Uas SD 2.cpp UAS_STRUKDAT_1.cpp
1 //Program undirected graph UAS SD
2 #include <iostream>
3 #include <stdio.h>
4 #include <conio.h>
5 #define N 100
6 using namespace std;
7
8 int main()
9 {
10     //inisialisasi variabel data
11     int graph[N][N];
12     int vertex, edge;
13     int vertex_1, vertex_2, bobot;
14
15     printf("=====\n");
16     printf(" Program Undirected Graph \n");
17     printf("=====\n\n");
18
19     //input data vertex dan edge yang diinginkan
20     printf("Masukkan vertex & edge : \n");
21     scanf("%d%d", &vertex, &edge);
22
23     //proses untuk pengulangan graph
24     for(int i=0; i<vertex; ++i)
25     {
26         for(int j=0; j<vertex; ++j)
27         {
28             graph[i][j]=0;
29         }
30     }
31
32     //inisialisasi vertex_1 = u, vertex_2 = v, dan bobot = w kemudian input datanya
33     printf("Masukkan data (u v w) : \n");
34     //proses pengulangan berdasarkan edge yang dimasukkan
35     for(int i=0; i<edge; ++i)
36     {
37         scanf("%d%d%d", &vertex_1, &vertex_2, &bobot);
38         //proses untuk fungsi undirected graph
39         graph[vertex_1][vertex_2] = graph[vertex_2][vertex_1] = bobot;
40     }
41
42     //mencetak hasil output dari proses program graph tidak berarah
43     printf("\n");
44     printf("Hasil Output : \n");
45     for(int i = 0; i < vertex; ++i)
46     {
47         for(int j = 0; j < vertex; ++j)
48         {
49             printf("%d ", graph[i][j]);
50         }
51         printf("\n");
52     }
53
54     return 0;
55     getch();
56 }
```

```
Uas SD 2.cpp UAS_STRUKDAT_1.cpp
19 //input data vertex dan edge yang diinginkan
20 printf("Masukkan vertex & edge : \n");
21 scanf("%d%d", &vertex, &edge);
22
23 //proses untuk pengulangan graph
24 for(int i=0; i<vertex; ++i)
25 {
26     for(int j=0; j<vertex; ++j)
27     {
28         graph[i][j]=0;
29     }
30 }
31
32 //inisialisasi vertex_1 = u, vertex_2 = v, dan bobot = w kemudian input datanya
33 printf("Masukkan data (u v w) : \n");
34 //proses pengulangan berdasarkan edge yang dimasukkan
35 for(int i=0; i<edge; ++i)
36 {
37     scanf("%d%d%d", &vertex_1, &vertex_2, &bobot);
38     //proses untuk fungsi undirected graph
39     graph[vertex_1][vertex_2] = graph[vertex_2][vertex_1] = bobot;
40 }
41
42 //mencetak hasil output dari proses program graph tidak berarah
43 printf("\n");
44 printf("Hasil Output : \n");
45 for(int i = 0; i < vertex; ++i)
46 {
47     for(int j = 0; j < vertex; ++j)
48     {
49         printf("%d ", graph[i][j]);
50     }
51     printf("\n");
52 }
53
54 return 0;
55 getch();
56 }
```

Hasil Ouput :

```
C:\Users\pico\Documents\FILE DEV C++\UAS_STRUKDAT_!.exe
=====
Program Undirected Graph
=====

Masukkan vertex & edge :
4 3
Masukkan data (u v w) :
1 2 4
2 4 4
1 4 3

Hasil Output :
0 0 0 0
0 0 4 0
0 4 0 0
0 0 0 0

-----
Process exited after 61.61 seconds with return value 0
Press any key to continue . . .
```

Berdasarkan kodingannya yang dimulai dari memasukkan file header yang diperlukan untuk menjalankan sistem operasi, kemudian memasuki int main diawali dengan inialisasi variabel data. Setelah melakukan inialisasi, langkah selanjutnya yaitu membuat fungsi input data vertex dan edge, kemudian dilakukan fungsi pengulangan agar dapat melakukan perulangan input dari nilai data yang sesuai dari jumlah input data vertex dan edge. Langkah berikutnya berupa membuat fungsi input nilai data dari u, v, dan w (vertex\_1, vertex\_2, dan bobot) yang dimasukkan dalam proses perulangan dengan batas jumlah edge yang dimasukkan, kemudian memasukkan proses untuk melakukan fungsi *undirected graph* dengan cara nilai  $\text{graph}[\text{vertex\_1}][\text{vertex\_2}] = \text{graph}[\text{vertex\_2}][\text{vertex\_1}] = \text{bobot}$ . Setelah itu membuat proses fungsi untuk mencetak nilai graph dari sistem operasi yang telah dilakukan sebelumnya dengan cara perulangan ditumpuk untuk menghasilkan bentuk matriks dan memasukkan variabel  $\text{graph}[i][j]$  sebagai hasil nilai dari proses yang telah dilakukan seperti yang ada digambar hasil output. Berikut gambar algoritma dari graph yang telah dibuat :

