



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2021-2022 / Genap
Modul : 9 - Tugas Besar

Naskah Soal Tugas Besar Tipe 1

Soal ini dikerjakan oleh kelompok dengan nomor ganjil

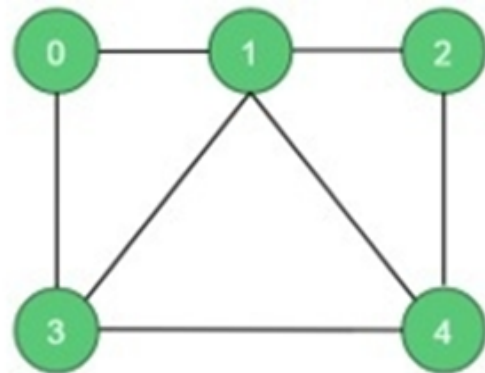
Pembuat Naskah: Dismas Widyanto

Graphs merupakan struktur data yang merepresentasikan sekumpulan objek yang terhubung. Obyek tersebut biasa disebut dengan *vertex*. Penghubung dari setiap *vertex* biasa disebut dengan *edges*. Salah satu operasi yang dapat dilakukan pada *Graphs* adalah mencari *path* dari suatu *vertex* ke *vertex* tertentu dengan jumlah *edges* yang dilalui sesuai dengan masukan dari pengguna.

Tugas ini meminta anda untuk mencari setiap jalur dari *vertex* asal ke *vertex* tujuan yang terdapat pada suatu *Graphs*. Jalur tersebut harus dapat mencapai tujuan dengan *edges* yang dilalui berjumlah tepat sesuai dengan masukan pengguna. *Graphs* yang digunakan direpresentasikan dalam bentuk *adjacent matrix* yang terdapat pada *file* eksternal. *Adjacent matrix* ini merupakan representasi *graph* dalam bentuk matriks sebesar banyak *vertex* dengan isi yang menunjukkan hubungan antar *vertex*.

Sebagai contoh

0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
1	1	0	0	1
0	1	1	1	0



Gambar di sebelah kanan adalah *graph* dan gambar di sebelah kiri adalah *adjacent matrix*. Matriks tersebut merepresentasikan masing-masing *vertex* dan angka 1 menunjukkan adanya hubungan antara *vertex* tersebut dengan *vertex* lainnya. Lebih jelasnya *adjacent matrix* akan berbentuk seperti tabel berikut.

VERTEX	0	1	2	3	4
0	0	1	0	1	0
1	1	0	1	1	1
2	0	1	0	0	1
3	1	1	0	0	1
4	0	1	1	1	0

Kemudian program mencari semua kombinasi jalur yang terdapat pada *Graphs* tersebut lalu mencetak jumlahnya pada layar. Selain itu, program juga akan menghasilkan *file* eksternal yang berisi seluruh kombinasi jalur yang ada.

Program yang dibuat diharapkan akan memiliki sifat seperti berikut, diurutkan berdasarkan prioritasnya

1. Memberikan solusi yang valid ketika solusi ada
2. Memberikan semua kemungkinan jalur yang ada
3. Dapat memberikan solusi secepat mungkin (memiliki *time complexity* serendah mungkin)
4. Dapat memberikan solusi dengan kebutuhan memori serendah mungkin (memiliki *space complexity* serendah mungkin)

Segala hal yang tidak dicantumkan pada naskah soal ini dapat diasumsikan atau ditanyakan kepada asisten pembimbing kelompok.

Ketentuan Input Output

Input dari program ini berupa sebuah *file* eksternal, vertex asal, vertex tujuan, dan jumlah edges yang diberikan oleh pengguna ketika program dijalankan. *File* eksternal tersebut berisi data banyaknya *vertex* beserta *adjacent matrix*. Struktur *file* eksternal yang digunakan dapat dilihat pada keterangan berikut.

```
<Banyak vertex>  
<adjacent matrix>
```

Berikut merupakan contoh *file* eksternal yang digunakan.

```
5  
0,1,0,1,0  
1,0,1,1,1  
0,1,0,0,1  
1,1,0,0,1  
0,1,1,1,0
```

File tersebut memiliki *vertex* berjumlah lima dengan hubungan antar *vertex* digambarkan pada *adjacent matrix*. Perhatikan bahwa proses *input* pada program harus bisa mengenali apabila *file* eksternal yang diberikan tidak ada, tidak sesuai format, atau kosong.

Output dari program ini adalah:

1. Jumlah jalur pada graphs yang diberikan
2. Semua kombinasi jalur pada graphs yang ditampilkan pada layar
3. *File* eksternal yang berisi seluruh kombinasi jalur pada *graphs*

File eksternal yang dihasilkan akan memiliki format tampilan persis seperti pada layar. Nama *file* eksternal yang dihasilkan akan memiliki format jalur-file sumber.txt.

```
#jalur-file1.txt  
<isi kombinasi jalur>
```

Contoh Eksekusi Program

Anda tidak harus mengikuti contoh eksekusi program ini. Silakan atur format eksekusi program selama masih memenuhi ketentuan *input output* di atas serta mudah dibaca oleh pengguna. *Output* dari program ini tidak akan diberikan pada *autograder*. *Underline* menunjukkan *input* dari *pengguna*.

#file1.txt

4

0,1,1,1

1,0,1,1

1,1,0,1

1,1,1,0

#file2.txt

5

0,1,1,0,0

1,0,0,0,0

1,0,0,1,1

0,0,1,0,1

0,0,1,1,0

#file3.txt

5

0,1,1,0,0

1,0,1,0,1

1,1,0,1,1

0,0,1,0,1

0,1,1,1,0

#1

Masukkan Graphs: file1.txt

Masukkan Vertex Asal: 2

Masukkan Vertex Tujuan: 4

Masukkan Batas Edges: 2

Banyak Jalur: 2

Jalur:

2-1-4

2-3-4

#2

Masukkan Graphs: file2.txt

Masukkan Vertex Asal: 1

Masukkan Vertex Tujuan: 5

Masukkan Banyak Edges: 5

Banyak Jalur: 0
Jalur:
Tidak Ada Jalur Yang Memenuhi

#3

Masukkan Graphs: file3.txt
Masukkan Vertex Asal: 1
Masukkan Vertex Tujuan: 4
Masukkan Banyak Edges: 3

Banyak Jalur: 3
Jalur:
1-2-3-4
1-2-5-4
1-3-5-4

#4

Masukkan Graphs: file_tidak_ada.txt

Error: file invalid!

#5

Masukkan Graphs: file_format_salah.txt

Error: file invalid!

#6

Masukkan Graphs: file_kosong.txt

Error: file empty!

#jalur-file1.txt

2-1-4
2-3-4

#jalur-file2.txt

Tidak Ada Jalur Yang Memenuhi

#jalur-file3.txt

1-2-3-4
1-2-5-4
1-3-5-4