

## Tugas II IF211 Strategi Algoritma

### Penyusunan Rencana Kuliah dengan Memanfaatkan DFS dan BFS

**Batas pengumpulan :** Rabu, 21 Maret 2018 sebelum pukul 13.00 WIB

**Arsip pengumpulan :** - *CD/DVD* yang berisi *Source* dan *Exe* program disertai *readme.txt*  
- Laporan (*hard copy*)

**Mekanisme pengumpulan :** Diumumkan kemudian oleh asisten

#### Deskripsi tugas :

Pada tugas kali ini, mahasiswa diminta **membuat aplikasi berbasis GUI (Graphical User Interface)** yang dapat menyusun rencana pengambilan kuliah, dengan memanfaatkan algoritma **DFS** dan **BFS**. Penyusunan Rencana Kuliah dengan memanfaatkan DFS dan BFS diimplementasikan dengan menggunakan pendekatan *Topological Sorting*. Berikut akan dijelaskan tugas yang dikerjakan secara detail.

1. Aplikasi akan menerima daftar mata kuliah beserta prasyarat yang harus diambil seorang mahasiswa sebelum mengambil mata kuliah tersebut. Daftar mata kuliah tersebut dituliskan dalam suatu file teks dengan format:

```
<kode_kuliah_1>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>.  
<kode_kuliah_2>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>.  
<kode_kuliah_3>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>, <kode kuliah prasyarat - 4>.  
<kode_kuliah_4>.
```

```
.  
. .  
.
```

Gambar 1. Format File Teks untuk Masukan Daftar Kuliah

Sebuah kode\_kuliah mungkin memiliki nol atau lebih prasyarat kuliah. Kode\_kuliah bisa diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil di semester sebelumnya (tidak harus 1 semester sebelumnya).

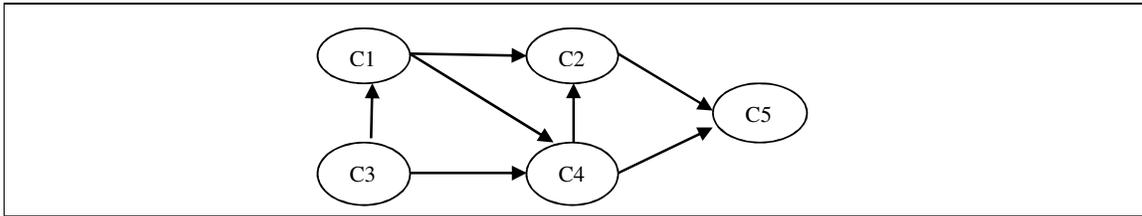
Sebagai contoh, terdapat 5 kuliah yang harus diambil seorang mahasiswa dengan daftar prerequisite dalam file teks sebagai berikut. Dari Gambar 2 terlihat bahwa kuliah C3 tidak memiliki prerequisite.

```
C1, C3.  
C2, C1, C4.  
C3.  
C4, C1, C3.  
C5, C2, C4.
```

Gambar 2. Contoh sebuah berkas masukan Daftar Kuliah

2. Dari file teks yang telah diterima, aplikasi harus dapat menampilkan visualisasi graf berarah tidak bersiklus (Directed Acyclic Graph/ DAG). Proses visualisasi ini boleh memanfaatkan

pustaka atau kakas yang tersedia. Contoh visualisasi graf untuk daftar mata kuliah pada Gambar 2, dapat dilihat pada Gambar 3.



Gambar 3. Visualisasi DAG dari contoh kasus pada gambar 2

3. Dari DAG yang sudah dibentuk, aplikasi menyusun jadwal pengambilan kuliah dengan memanfaatkan DFS dan BFS untuk penelusuran graf. Tidak ada batasan banyaknya kuliah yang dapat diambil pada suatu semester, tapi pengambilan suatu mata kuliah tidak boleh bersamaan dalam satu semester dengan kuliah prasyaratnya. Selain itu diasumsikan semua mata kuliah ditawarkan di setiap semester.

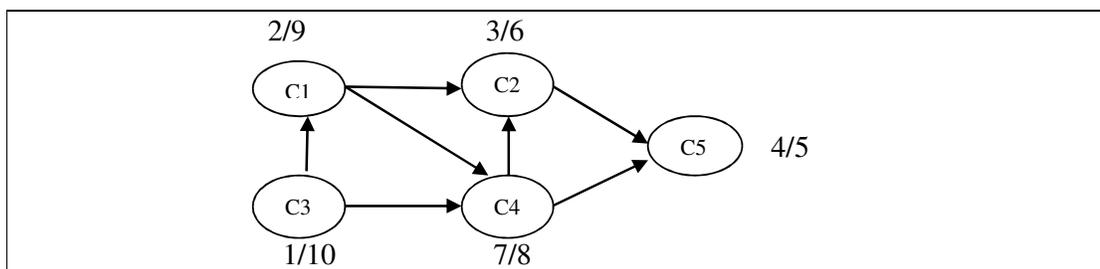
Dapat dilihat bahwa kasus penyusunan rencana kuliah ini sebagai salah satu implementasi *topological sorting*. Aplikasi harus dapat menyusun rencana kuliah dengan pendekatan *topological sorting* dan memanfaatkan DFS dan BFS untuk sortingnya. Video *topological sorting* dapat dilihat pada laman *Youtube* berikut:

<https://www.youtube.com/watch?v=yYW6lQ1ajx4&feature=youtu.be>

**Pendekatan DFS untuk Topological Sorting**

- a. Dari graf yang terbentuk, lakukan penelusuran terhadap DAG yang terbentuk pada langkah sebelumnya. Setiap kali melakukan penelusuran, catat waktu selesai (*time stamp*) penelusuran pada setiap simpul.

Untuk kasus pada Gambar 2, maka penelusuran DFS dapat dimulai dari C3 sehingga C3 dimulai pada waktu ke-1, dan waktu selesainya 10, yaitu setelah semua simpul selesai ditelusuri. Pada Gambar 4, tiap simpul dilengkapi dengan informasi [waktu\_mulai/waktu\_selesai]. Sebagai contoh, simpul C3 adalah simpul yang pertama ditelusuri karena tidak memiliki busur yang masuk ke simpul C3. Simpul C3 selesai ditelusuri setelah waktu ke-10, yaitu ketika semua simpul sudah selesai ditelusuri. Contoh lain untuk simpul C5, waktu mulai ditelusurinya adalah time stamp ke 4 (setelah C3, C1, C2). Waktu selesai C5 adalah 5, karena setelah C5 tidak ada lagi ‘anak’ dari C5 yang bisa ditelusuri, sehingga kembali ke C2. Waktu selesai C2 adalah 6, dan kembali ke C1, demikian seterusnya.

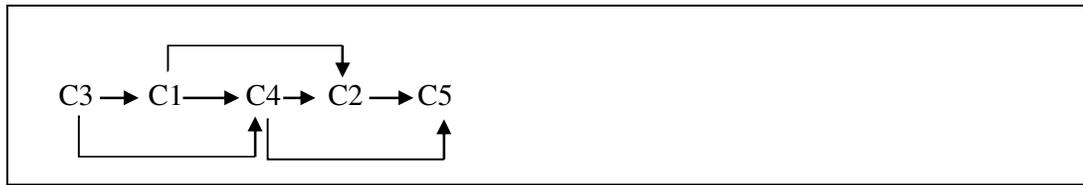


Gambar 4. Penelusuran DFS dari DAG daftar mata kuliah

- b. Lakukan pengurutan waktu penyelesaian secara menurun untuk menghasilkan urutan semua simpul yang ada pada DAG. Berdasarkan Gambar 4, maka urutan yang dihasilkan adalah:

C3    C1    C4    C2    C5

- c. Berdasarkan DAG yang sudah dihasilkan sebelumnya, maka busur yang menghubungkan semua simpul di langkah (b), dan hasilnya urutan adalah sebagai berikut.



Gambar 5. Hasil topological Sort untuk Kasus 5 mata kuliah pada gambar 2

Berdasarkan pada Gambar 5, maka rencana kuliah yang dihasilkan adalah sebagai berikut.

Semester I : C3  
 Semester II : C1  
 Semester III : C4  
 Semester IV : C2  
 Semester V : C5.

### Pendekatan BFS untuk Topological Sorting

- a. Dari graf (DAG) yang terbentuk, hitung semua derajat-masuk (*in-degree*) setiap simpul, yaitu banyaknya busur yang masuk pada simpul tersebut. Pada contoh kasus di Gambar 2, maka derajat-masuk tiap simpul adalah sebagai berikut.

C1 : 1  
 C2 : 2  
 C3 : 0  
 C4 : 2  
 C5 : 2

- b. Pilih sembarang simpul yang memiliki derajat-masuk 0. Pada kasus Gambar 2, pilih simpul C3.  
 c. Ambil simpul tersebut, dan hilangkan simpul tersebut beserta semua busur yang keluar dari simpul tersebut pada graf, dan kurangi derajat simpul yang berhubungan dengan simpul tersebut dengan 1.

Setelah simpul C3 dipilih, maka derajat simpul yang lain menjadi sebagai berikut.

C1 : 0  
 C2 : 2  
 C4 : 1  
 C5 : 2

- d. Ulangi langkah (b) dan (c) hingga semua simpul pada DAG terpilih. Untuk kasus pada Gambar 2, setelah simpul terakhir dipilih maka urutan yang dihasilkan akan seperti pada Gambar 5.

4. Aplikasi harus dapat menunjukkan langkah-langkah proses penentuan urutan dengan memanfaatkan algoritma DFS dan BFS.
5. Aplikasi harus dapat menunjukkan hasil *topological sort* dengan memanfaatkan algoritma DFS dan BFS seperti pada Gambar 5.
6. Aplikasi menampilkan hasil penentuan rencana kuliah, berupa semester dan kuliah apa saja yang diambil pada setiap semester. Tampilan untuk fitur ini bebas, silakan dibuat sebagus mungkin dengan berbasis GUI.
7. Data Uji akan diberikan oleh asisten.

### Lain – lain :

1. Anda dapat menambahkan fitur-fitur lain yang menunjang program yang anda buat (unsur kreativitas).
2. Aplikasi dibuat dengan bahasa C# dengan platform .NET. Kakas pemrograman yang digunakan misalnya *Visual Studio*.
3. Tugas dikerjakan per kelompok dengan jumlah anggota minimal 2 orang dan maksimal 3 orang. Anggota kelompok diperbolehkan lintas kelas dan tidak boleh sama dengan kelompok Tubes sebelumnya.
4. Program harus modular dan mengandung komentar yang jelas.
5. Beri nama aplikasi anda tersebut dengan nama-nama yang menarik dan mudah diingat.
6. Dilarang menggunakan kode program *Topological Sorting* DFS dan BFS yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
7. Pengumpulan adalah hari Rabu tanggal 21 Maret 2018 sebelum pukul 13.00. Keterlambatan akan mengurangi nilai.
8. *Source, exe, dan readme.txt* disimpan di dalam *folder* StrAlgo2-xxxxx. Lima digit terakhir adalah NIM anggota terkecil.
9. **Bonus** (nilai maksimal 10): Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke *Youtube*. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video *Youtube* tersebut dan memutarinya di depan asisten sebelum memulai demo.
10. Semua pertanyaan menyangkut tugas ini harus dikomunikasikan melalui milis agar dapat dicermati oleh semua peserta kuliah IF2211.
11. Demo program akan dilaksanakan pada tanggal 21 Maret 2018 – 29 Maret 2018 kecuali ada pemberitahuan lebih lanjut dari asisten. Peserta mengisi jadwal demo yang disediakan.
12. Tiap anggota harus memahami proses pembuatan program, karena akan ada pertanyaan-pertanyaan yang harus dijawab per individu.
13. Pada saat demo, asisten akan memanggil per kelompok. Kelompok yang tidak berkepentingan dilarang masuk. Demo dilakukan di Lab IRK.

### Isi laporan :

*Cover*: *Cover* laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo “gajah” ganesha.

Bab 1: Deskripsi tugas (dapat menyalin spesifikasi tugas ini)

Bab 2: Dasar teori (Algoritma DFS dan BFS)

Bab 3: Analisis Pemecahan Masalah. Langkah-langkah pemecahan masalah ada di sini beserta contoh ilustrasi kasus lain yang berbeda dari contoh pada spesifikasi tugas ini, termasuk di dalamnya struktur data dan spesifikasi program.

Bab 4: Implementasi dan pengujian, berisi implementasi program dalam bahasa yang dipilih dan kakas yang digunakan (bukan berisi kode program), misalnya menampilkan nama file program (jika modular atau dalam betuk kelas), *screen-shot* antarmuka program, dan lain-lain. Di dalam pengujian dituliskan masukan daftar kuliah yang dicoba, lalu dianalisis hasil-hasil tersebut.

Bab 5: Kesimpulan dan saran.

Daftar Pustaka.

### Keterangan laporan :

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Laporan tidak perlu memakai *cover* mika atau dijilid. Cukup dibuat agar laporan tidak akan tercecer bila dibaca.

3. Laporan boleh menggunakan kertas riu, boleh bolak-balik, boleh dalam satu halaman kertas terdapat dua halaman tulisan asalkan masih terbaca.
4. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

**Penilaian :**

1. Kebenaran program (40%) : program mampu memproses data yang sudah disediakan dan data dari asisten.
2. Demo – pemahaman Anda dalam pembuatan program (30%)
3. Laporan (20%)
4. *Interface*, fitur-fitur program, dan unsur kreativitas (10%)

*-selamat mengerjakan-*