

## **LAPORAN TUGAS KECIL 2**

### **IF2211 – STRATEGI ALGORITMA**

**Penyusunan Rencana Kuliah dengan Topological Sort**

**(Penerapan Decrease and Conquer)**



**Oleh :**

**Clarisa Natalia Edelin 13519213**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2021**

## 1. Algoritma Topological Sort (Penerapan Decrease and Conquer)

Decrease and Conquer merupakan sebuah metode perancangan algoritma dengan mereduksi persoalan menjadi dua upa-persoalan atau sub-problem yang lebih kecil, namun selanjutnya hanya memproses satu sub-persoalan saja. Perbedaan decrease and conquer dengan divide and conquer adalah divide and conquer memproses semua upa – persoalan dan menggabungkan semua solusi setiap sub-persoalan, sedangkan decrease and conquer hanya memproses salah satu dari hasil pembagian upa-persoalan. Algoritma decrease and conquer dibagi menjadi 2 tahapan, yaitu decrease, dimana tahapan ini mereduksi persoalan menjadi beberapa persoalan yang lebih kecil, dan conquer, dimana tahapan ini yang memproses satu upa-persoalan secara rekursif.

Topological sorting merupakan salah satu pendekatan untuk mengurutkan objek atau kegiatan secara terjadwal. Keterurutan tersebut diperlukan karena adanya suatu persyaratan atau keharusan suatu objek atau kegiatan harus dilakukan atau dieksekusi terlebih dahulu dibandingkan yang lain. Maka, pendekatan topological sorting mengeksekusikan objek atau kegiatan yang tidak memiliki prasyarat terlebih dahulu dan menghapusnya dari daftar sehingga objek atau kegiatan lain bisa dikerjakan setelah prasyaratnya telah terpenuhi. Setelah itu, pada daftar akan ditemukan kembali sebuah objek atau kegiatan yang tidak memerlukan prasyarat, lalu dieksekusi dan dihapus dari daftar, seperti itu seterusnya sampai daftar kosong. Oleh karena itu, topological sorting termasuk ke dalam decrease and conquer dengan varian decrease by constant.

Berikut adalah langkah – langkah algoritma Topological Sort dengan menggunakan pendekatan Decrease and Conquer:

1. Program membaca input dari file.txt yang di dalamnya terdapat sebuah soal yang terdiri dari sebuah daftar matakuliah yang diikuti dengan matakuliah prasyaratnya. File tersebut akan diolah sehingga daftar tersebut akan disimpan dalam program dalam bentuk dictionary, dengan key adalah mata kuliah utama dan value adalah mata kuliah prasyarat.
2. Program menginisialisasi sebuah variabel dengan nama indeks, yaitu sebuah indeks yang mengindikasikan proses topological sort yang akan dilakukan dan berkoresponden dengan semester waktu mata kuliah dapat diambil. Inisialisasi indeks adalah 1.
3. Program akan memanggil sebuah fungsi rekursif, yaitu `hasiltolosort(soal,indeks)`, dimana soal adalah dictionary hasil pembacaan file. Pada fungsi tersebut, program akan membuat sebuah list dengan isi semua key yang dimiliki oleh dictionary soal, menginisialisasi sebuah dictionary `hasilsort` sebagai dictionary hasil akhir, dan menginisialisasi sebuah array `arrayhasilpersemester` sebagai array yang akan menyimpan indeks dari mata kuliah tanpa prasyarat di dalam list key.
4. Program akan mencari matakuliah yang tidak memiliki prasyarat, atau yang nilainya nol dan menyimpan indeks key tempat matakuliah tanpa prasyarat itu berada ke dalam sebuah array.
5. Program akan menghapus mata kuliah yang tidak memiliki prasyarat tersebut dari seluruh value yang ada di dictionary.

6. Program akan menghapus mata kuliah tanpa prasyarat dari daftar key yang ada di dictionary. Sehingga dictionary soal sekarang hanya memiliki daftar mata kuliah yang sebelumnya memiliki prasyarat.
7. Dengan menggunakan array yang berisi indeks key tempat matakuliah tanpa prasyarat berada, program akan menambahkan mata kuliah dapat diambil pada semester tersebut ke dalam dictionary hasil.
8. Program akan menampilkan hasil pencarian dengan dengan menampilkan dictionary hasil pada layar.
9. Jika panjang dari list key dari array soal yang sudah dikurangi dengan mata kuliah yang telah ditampilkan di layar masih belum nol, program akan menambahkan indeks sebanyak satu dan memanggil fungsi `hasiltoposort(soal,indeksss)` lagi dan mengulang proses nomor 3 sampai 9.

## 2. Source Code Program

```

1  #Nama      : Clarisa Natalia Edelin
2  #NIM       : 13519213
3  #TUCIL 2 STIMA : Topological Sort dengan metode Decrease and Conquer
4
5  def bacafiletxt():
6      with open("toposort1.txt") as A: #mengubah nama file sesuai dengan nama file yang diinginkan disini
7          lines = A.read().strip().split("\n")
8          lis = {}
9          for line in lines:
10             b = line.replace(" ", "").split(",")
11             lis[b[0]] = b[1:]
12
13         return(lis)
14
15 def hapusmatkultertentudaridict(soals,matkulygmaudiapus,keys):
16     for q in range(len(soals)):
17         #print(q)
18         if soals.get(keys[q]) != None:
19             if(len(soals.get(keys[q])) > 0):
20                 for j in range(len(soals.get(keys[q]))):
21                     #print(q,j)
22                     if((soals.get(keys[q])[j]) == matkulygmaudiapus):
23                         #print('yas')
24                         soals.get(keys[q]).remove(matkulygmaudiapus)
25                         break
26
27     return soals
28
29 def hasiltoposort(soal,indeksss):
30     key = list(soal.keys())
31     hasilsort = {}
32     arrayhasilpersemester = []
33
34     for i in range(len(key)):
35         if len(soal.get(key[i])) == 0:
36             arrayhasilpersemester.append(i)
37
38     for aaa in range(len(arrayhasilpersemester)):
39         hapusmatkultertentudaridict(soal,key[arrayhasilpersemester[aaa]],key)
40
41     for aaa in range(len(arrayhasilpersemester)):
42         del soal[key[arrayhasilpersemester[aaa]]]
43
44     hasilsort = {indeksss:[]}
45     for aaa in range(len(arrayhasilpersemester)):
46         if(hasilsort.get(indeksss) == []):
47             hasilsort[indeksss] = [key[arrayhasilpersemester[aaa]]]
48         else:
49             hasilsort[indeksss].append(key[arrayhasilpersemester[aaa]])
50
51     keyhasilsort = list(hasilsort.keys())
52     print("semester", keyhasilsort[0], " : " , hasilsort.get(keyhasilsort[0]))
53
54
55     keysekarang = list(soal.keys())
56

```

```

57     if(len(keysekarang)) != 0:
58         indeksss += 1
59         hasiltoposort(soal,indeksss)
60
61
62 #program utama
63 soal = bacafiletxt()
64 indeksss = 1
65
66 hasiltoposort(soal,indeksss)
67

```

### 3. Hasil Test Case Program

Input	Output
C2, C1, C3 C3, C1 C1 C4, C3, C2	<pre> C:\Users\ELIA\Documents\test&gt; python 01_01_01.py semester 1 : ['C1'] semester 2 : ['C3'] semester 3 : ['C2'] semester 4 : ['C4'] </pre>
C1, C3 C2, C1, C4 C3 C4, C1, C3 C5, C2, C4	<pre> C:\Users\ELIA\Documents\test&gt; python 01_01_01.py semester 1 : ['C3'] semester 2 : ['C1'] semester 3 : ['C4'] semester 4 : ['C2'] semester 5 : ['C5'] </pre>
IF4051, IF3211, IF3130 IF3211, IF2130, IF2110 IF3130, IF2230 IF2130 IF2110 IF2230	<pre> C:\Users\ELIA\Documents\test&gt; python 01_01_01.py semester 1 : ['IF2130', 'IF2110', 'IF2230'] semester 2 : ['IF3211', 'IF3130'] semester 3 : ['IF4051'] </pre>
c1, c2 c2, c3 c3, c4 c4, c5 c5, c6 c6, c7 c7, c8 c8	<pre> C:\Users\ELIA\Documents\test&gt; python 01_01_01.py semester 1 : ['c8'] semester 2 : ['c7'] semester 3 : ['c6'] semester 4 : ['c5'] semester 5 : ['c4'] semester 6 : ['c3'] semester 7 : ['c2'] semester 8 : ['c1'] </pre>
ET4047, ET3103 ET3103, ET2208 ET2208, ET2101 ET2101, MA1204 MA1204 ET4111, ET3101, ET3103 ET3101, ET2109, ET2204 ET2109, MA1204 ET2204, ET2103 ET2103, EL1200 EL1200, MA1104 MA1104	<pre> C:\Users\ELIA\Documents\test&gt; python 01_01_01.py semester 1 : ['MA1204', 'MA1104'] semester 2 : ['ET2101', 'ET2109', 'EL1200'] semester 3 : ['ET2208', 'ET2103'] semester 4 : ['ET3103', 'ET2204'] semester 5 : ['ET4047', 'ET3101'] semester 6 : ['ET4111'] </pre>

<p>EL3016, EL2004, EL2007  EL2004, MA1201  EL2007, EL1200  MA1201  EL1200, MA1101  MA1101  EL3010, EL2007  EL2007, EL1200</p>	<pre>PS C:\Users\ELIA\Documents\test&gt; python test.py semester 1 : ['MA1201', 'MA1101'] semester 2 : ['EL2004', 'EL1200'] semester 3 : ['EL2007'] semester 4 : ['EL3016', 'EL3010'] PS C:\Users\ELIA\Documents\test&gt;</pre>
<p>Matematika IA  Matematika IIA, Matematika IA  Matematika Diskrit, Matematika IA, Matematika IIA  Pengenalan Komputasi  Dasar Pemrograman, Pengenalan Komputasi  Aljabar linier dan Geometri, Matematika IA, Matematika IIA  Algoritma dan Struktur Data, Dasar Pemrograman  Probabilitas dan Statistika, Matematika IA, Matematika IIA, Matematika Diskrit  Pengembangan Aplikasi Berbasis Web, Pemrograman Berorientasi Objek  Pemrograman Berorientasi Objek, Algoritma dan Struktur Data</p>	<pre>PS C:\Users\ELIA\Documents\test&gt; python test.py semester 1 : ['MatematikaIA', 'PengenalanKomputasi'] semester 2 : ['MatematikaIIA', 'DasarPemrograman'] semester 3 : ['MatematikaDiskrit', 'AljabarlinierdanGeometri', 'AlgoritmadanStrukturData'] semester 4 : ['ProbabilitasdanStatistika', 'PemrogramanBerorientasiObjek'] semester 5 : ['PengembanganAplikasiBerbasisWeb']</pre>
<p>Termodinamika Teknik Kimia, Fisika Dasar IIA  Neraca Massa dan Energi Dasar IIA  Kimia Dasar Ia  Fisika Dasar IA  Fisika Dasar IIA, Fisika Dasar IA  Teknik Reaksi Kimia, Termodinamika Teknik Kimia, Neraca Massa dan Energi Dasar IIA  Operasi Perpindahan Kalor, Termodinamika Teknik Kimia, Mekanika Fluida dan Partikel  Mekanika Fluida dan Partikel</p>	<pre>PS C:\Users\ELIA\Documents\test&gt; python test.py semester 1 : ['NeracaMassadanEnergiDasarIIA', 'KimiaDasarIa', 'FisikaDasarIA', 'MekanikaFluidadanPartikel'] semester 2 : ['FisikaDasarIIA'] semester 3 : ['TermodinamikaTeknikKimia'] semester 4 : ['TeknikReaksiKimia', 'OperasiPerpindahanKalor']</pre>

**Link pengumpulan :**

<https://github.com/ClarisaNatalia/Tucil-2-Stima>

Poin	Ya	Tidak
1. Program berhasil dikompilasi	✓	
2. Program berhasil running	✓	
3. Program dapat menerima berkas input dan menuliskan output	✓	
4. Luaran sudah benar untuk semua kasus input	✓	