

Tugas Kecil 2 IF2211 Strategi Algoritma

**Penyusunan Rencana Kuliah dengan *Topological Sort*  
(Penerapan *Decrease and Conquer*)**



Nama : Jusuf Junior Athala  
NIM : 13519174

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
2021

## A. Algoritma Topological Sort dalam Decrease and Conquer

Decrease and conquer: metode perancangan algoritma dengan mereduksi persoalan menjadi dua upa- persoalan (sub-problem) yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja. Algoritma decrease and conquer terdiri dari dua tahapan:

1. Decrease: mereduksi persoalan menjadi beberapa persoalan yang lebih kecil (biasanya dua upa-persoalan).
2. Conquer: memproses satu upa-persoalan secara rekursif.

Topological sort untuk Directed Acyclic Graph (DAG) adalah pengurutan linear dari simpul-simpul untuk setiap hubungan dari simpul u menuju simpul v, simpul u diproses terlebih dahulu sebelum simpul v dalam pengurutan. Untuk topological sort, DAG harus memiliki simpul awal atau simpul yang tidak memiliki derajat masuk. Algoritma ini akan mengurangi ukuran dari graph sebanyak satu setiap satu simpul dihilangkan.

## B. Deskripsi Tugas

Pada tugas kali ini, mahasiswa diminta membuat aplikasi sederhana yang dapat menyusun rencana pengambilan kuliah, dengan memanfaatkan algoritma Decrease and Conquer. Penyusunan Rencana Kuliah 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

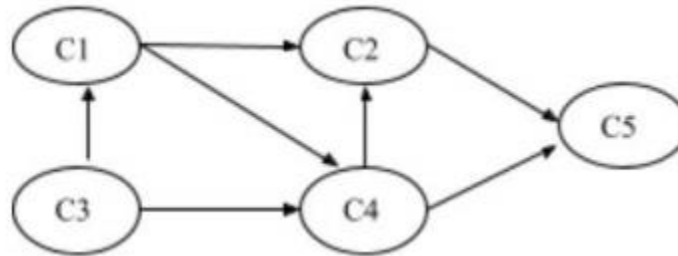
Sebuah kode\_kuliah mungkin memiliki nol atau lebih prasyarat kuliah. Kode\_kuliah bisa diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil di semester sebelumnya (tidak harus 1 semester sebelumnya). Asumsi semua kuliah bisa diambil di sembarang semester, baik semester ganjil maupun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

Asumsi untuk persoalan ini, kuliah dan prerequisite nya pasti berupa Directed Acyclic Graph (DAG), dan untuk contoh pada Gambar 2, dapat dilihat representasi DAG pada gambar 3.



Gambar 3. DAG dari daftar kuliah pada Gambar 2

2. Dari file teks yang telah diterima, ditentukan kuliah apa saja yang bisa diambil di semester 1, semester 2, dan seterusnya. Sebuah kuliah tidak mungkin diambil pada semester yang sama dengan prerequisitenya. Untuk menyederhanakan persoalan, tidak ada Batasan banyaknya kuliah yang bisa diambil pada satu 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 sebagai salah satu contoh penerapan Decrease and Conquer. Penjelasan tentang topological sorting dapat dibaca pada buku Levitin sub bab 4.2 dan video di YouTube: <https://www.youtube.com/watch?v=eL-KzMXSXXI>

#### Pendekatan 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

Ulangi langkah (b) dan (c) hingga semua simpul pada DAG terpilih. Untuk kasus pada Gambar 2, setelah simpul terakhir dipilih rencana kuliah yang dihasilkan adalah sebagai berikut.

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

Kebetulan untuk contoh ini, satu semester hanya ada 1 kuliah.

3. Sediakan data uji sendiri, yang menjamin DAG jika diubah ke dalam representasi graf.

### **C. Langkah-langkah Penyusunan Rencana Kuliah dengan Topological Sort**

1. Pisahkan setiap mata kuliah beserta mata kuliah prerequisite masing-masing.
2. Buat list mata kuliah yang muncul, kemudian membuat dictionary dengan kode mata kuliah atau simpul sebagai key dan jumlah mata kuliah prerequisite atau derajat masuk sebagai value dari key.
3. Pilih mata kuliah yang memiliki value 0.
4. Hilangkan mata kuliah dari dictionary lalu kurangi jumlah value pada mata kuliah lain yang berhubungan dengan mata kuliah tersebut.
5. Ulangi langkah (3) dan (4) hingga semua kode mata kuliah terpilih.

### **D. Source Code Program**

```
# Nama          : Jusuf Junior Athala
# NIM           : 13519174
# Mata kuliah   : IF2211 - Strategi Algoritma
# Kelas         : K-04
# Tugas         : Tugas Kecil 02

#fungsi untuk membaca file dan mengembalikan isi dari file dalam bentuk string
def bacaFile():

    fileinput=input("Masukkan nama file (contoh: 'input1.txt'): ")
    f=open("../test/"+fileinput, "r")
    contents =f.read()
    return contents

#fungsi membuat dictionary dari matakuliah yang muncul
def makeDictionary(matkul,sentence):
    separator =', '
    mydict = { }
    for mat in matkul:
        count = 0
```

```

        value = 0
        temp=mat
        for sen in sentence :
            if (sen.find(mat)==0):
                #jika matkul berada pada
                posisi awal, matkul lainnya akan menjadi value untuk dictionary dari matkul
                posisi awal
                value = sen.count(separator)
                #, matkul lainnya akan
                menjadi value untuk dictionary dari matkul posisi awal
                mydict[mat]=value
        return mydict

#fungsi topoogical Sort
def topoSort(mydict,key_list,val_list,sentence,matkul):
    count=1
    while (len(mydict)>0):
        first = True
        print ("Semester ",count," :", end = " ")
        for i in range(len(val_list)):
            if (val_list[i]==0):
                #Mencari mata kuliah dengan
                value prerequisite bernilai 0
                if (first) :
                    print(key_list[i], end = "")
                    first = False
                else :
                    print(", ",key_list[i], end="")

            temp=key_list[i]
            #Mengakses nama kunci pada
            index i

            for sen in sentence:
                #Jika mata kuliah merupakan
                prerequisite dari matkul lain, value dari matkul lain tersebut berkurang 1
                if (sen.find(temp)>1):
                    for mat in matkul:
                        if (sen.find(mat)==0):
                            mydict[mat]= mydict.get(mat)-1

            del mydict[temp]
            #Penghapusan kode matkul dari
            dictionary
            count+=1
            print("")
            key_list = list(mydict.keys())
            #Mengupdate list kode matkul
            dan value prerequisite setelah dilakukan
            val_list = list(mydict.values())
            #pengurangan value dan
            penghapusan kode matkul dari dictionary

if __name__ == '__main__':
    #Memulai program dengan membaca masukan file
    contents = bacaFile()

    #contents yang berupa string akan diolah dan dipisahkan
    sentence = contents.replace(".", "").split("\n")
    #setiap line dipisah
    menjadi bentuk <kode_kuliah_1>,<kode kuliah prasyarat - 1>, <kode kuliah
    prasyarat - 2>, dst.
    muncul = contents.replace (" , ", "\n").replace(".", "").split("\n")
    #penghapusan karakter yang tidak digunakan dan setiap kode kuliah akan
    dipisahkan masing-masing

```

```

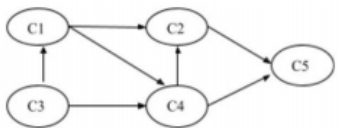
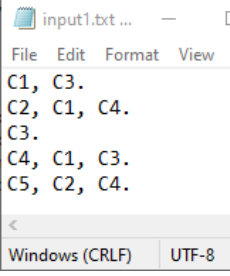
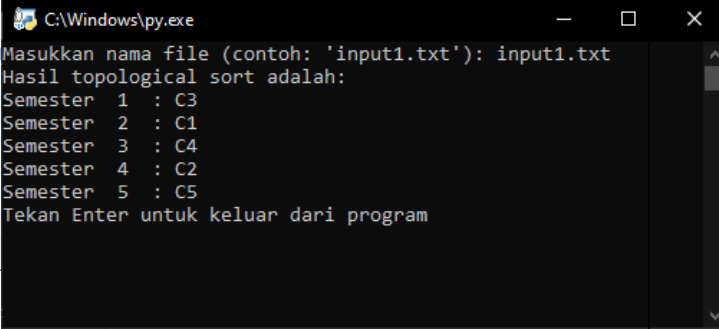
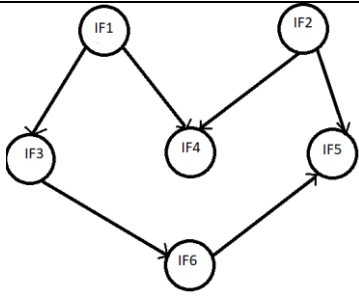
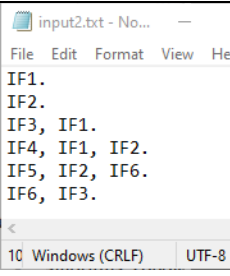
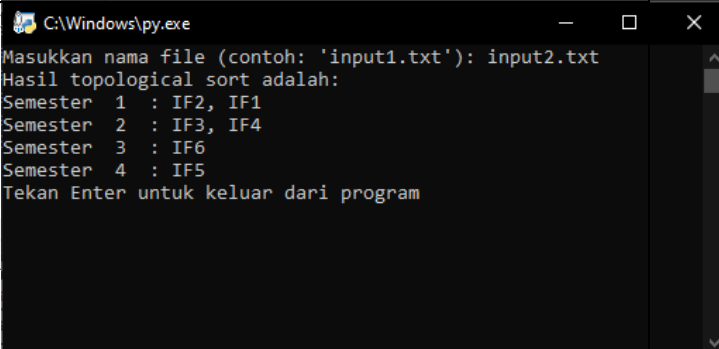
    matkul = set(muncul)          #menghapus duplikat dari mata kuliah yang
muncul menggunakan set
    matkul = list(matkul)        #mengubah set menjadi list

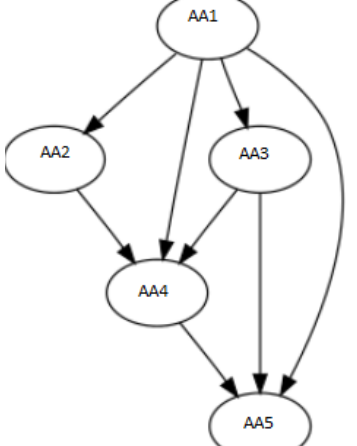
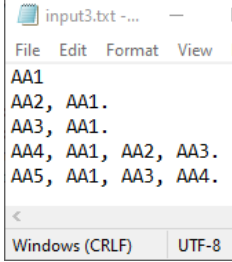
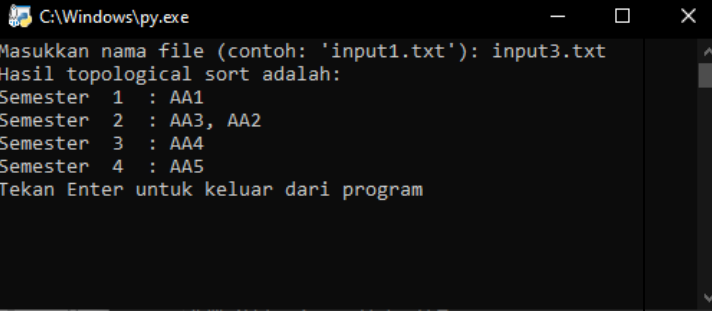
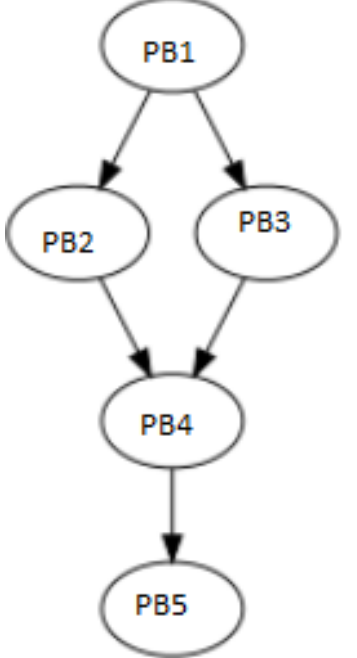
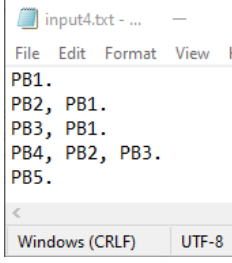
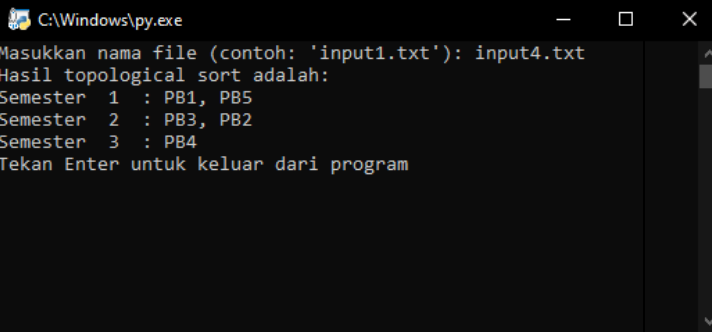
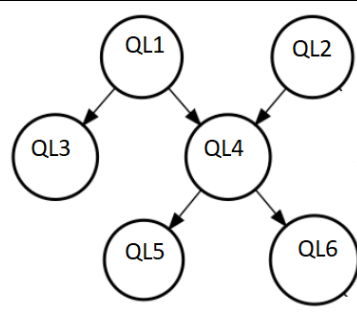
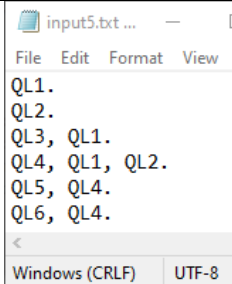
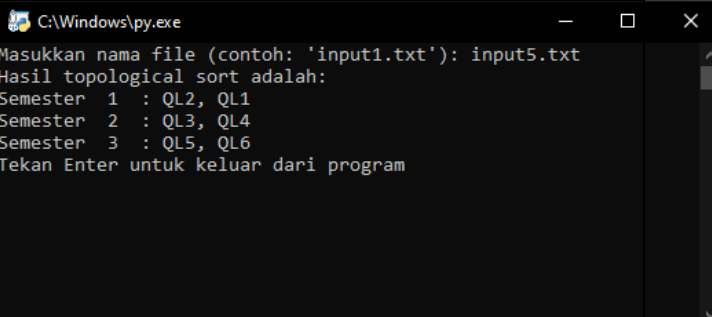
#Membuat list dan dictionary dari hasil pengolahan string
    mydict = makeDictionary(matkul,sentence)
    key_list = list(mydict.keys()) #kode matkul dalam bentuk list
    val_list = list(mydict.values()) #value prerequisite dari setiap kode
matkul dalam bentuk list
                                #list kode matkul dan value dari setiap
kode matkul sesuai

    print ("Hasil topological sort adalah: ")
#Melakukan topological sort
    topoSort(mydict,key_list,val_list,sentence,matkul)
    selesai = input('Tekan Enter untuk keluar dari program')

```

## E. Contoh Eksekusi Program (Input dan Output)

No	Soal	Input file	Output
1			
2			

3			
4			
5			

6			
7			
8			
9			

## F. Checklist

Berikut adalah checklist penilaian.



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

Berikut adalah link untuk file program :

[https://github.com/jusufjathala/Tucil2\\_13519174](https://github.com/jusufjathala/Tucil2_13519174)