

Laporan Tugas Kecil 2 IF 2211 Strategi Algoritma  
Semester II Tahun 2020/2021

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



**Disusun Oleh:**

Aulia Adila  
13519100

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
SEMESTER II TAHUN 2020/2021**

## Pendahuluan

Sebuah mata kuliah mungkin memiliki nol atau lebih prasyarat kuliah. Mata kuliah tersebut dapat diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil di semester sebelumnya (tidak harus 1 semester sebelumnya). Salah satu cara penyusunan rencana pengambilan kuliah ini dapat memanfaatkan algoritma decrease and conquer, dengan implementasi *topological sort*. *Topological Sort* merupakan sebuah pengurutan linear yang dilakukan terhadap graf berarah, di mana pada setiap busur berarah  $uv$  dari simpul  $u$  menuju simpul  $v$ , maka  $u$  akan diurutkan sebelum  $v$ .

### **Algoritma *Topological Sort* dan kaitannya dengan pendekatan *Decrease and Conquer***

Decrease and conquer adalah metode perancangan algoritma dengan mereduksi persoalan menjadi sub-problem yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja. Decrease, artinya adalah mereduksi persoalan menjadi beberapa persoalan yang lebih kecil. Conquer, artinya adalah memproses satu-upa persoalan secara rekursif. Dalam algoritma ini, tidak terdapat tahap combine, sebab hanya satu upa-persoalan yang diselesaikan.

Algoritma Topological Sort yang diimplementasikan oleh penulis dalam persoalan penyusunan rencana kuliah terdiri atas beberapa tahap sebagai berikut.

1. Membaca masukan berupa file txt dan membuat representasi graf dengan menggunakan dictionary sesuai dengan mata kuliah dan prasyaratnya  
Penulis menggunakan bahasa pemrograman Python, sehingga penyimpanan data yang cocok menurut penulis adalah dengan menggunakan dictionary. Dictionary menyimpan data dalam pasangan key dan value. Dalam persoalan ini, key diisi oleh setiap mata kuliah (secara unik), dan value setiap key diisi dengan mata kuliah prasyarat. Penyimpanan data menggunakan dictionary dianggap mempermudah akses mata kuliah, karena tidak diperlukan index yang menyatakan keterurutan key.  
Nama fungsi terkait : `makedict(f)` dan `main`
2. Menghitung jumlah busur yang masuk dalam setiap simpul, di mana simpul merepresentasikan mata kuliah tertentu, sedangkan busur berarah menyatakan keterhubungan urutan prasyarat antar mata kuliah

Jumlah busur dihitung dengan cara menghitung jumlah elemen yang terdapat dalam list value dari setiap key (mata kuliah).

Nama fungsi terkait : `count(dict)`

3. Memilih mata kuliah yang akan di eliminasi

Langkah ini adalah implementasi dari algoritma **conquer**, di mana penulis membagi persoalan besar menjadi upa-persoalan yang kecil, yaitu key (mata kuliah) yang akan di eliminasi. Mata kuliah ini merupakan mata kuliah yang boleh di ambil pada semester pertama yaitu mata kuliah yang tidak memiliki prasyarat, atau memiliki nol jumlah busur, atau memiliki nol jumlah elemen yang terdapat di dalam value dari key tersebut.

Nama fungsi terkait : `delkey(dict1,dict_c)`

4. Menghapus mata kuliah terpilih

Mata kuliah yang tidak memiliki prasyarat dikenakan aksi menghapus, yang merupakan implementasi dari algoritma **decrease**. Satu upa-persoalan ini direduksi bervariasi pada setiap interaksi algoritma (*decrease by a variable size*) sebab parameter reduksi dalam persoalan adalah saat jumlah busur atau jumlah elemen value dari key bernilai nol, yang mana kondisi ini mungkin dicapai oleh lebih dari satu mata kuliah (dalam bentuk simpul/key) dalam satu waktu.

Nama fungsi terkait : `delkey(dict1,dict_c)`

5. Memperbarui/melakukan update terhadap kondisi graf

Graf akan diperbarui setiap pergantian semester, dengan syntax update terhadap dictionary. Selain dictionary untuk merepresentasikan graf, terdapat dictionary yang merepresentasikan jumlah busur yang masuk untuk setiap simpul mata kuliah yang akan di update setiap semester. Update untuk dictionary ini dilakukan dengan method clear dictionary lama, lalu copy dictionary graf yang baru dan mengubah value setiap key menjadi jumlah elemen value setiap key (menggunakan `len`).

Nama fungsi terkait : `delkey(dict1,dict_c)` dan `update(idx,value)`

6. Melakukan iterasi dari prosedur-prosedur yang dijelaskan sebelumnya hingga dictionary graf initial (dictionary graf dari masukan teks yang pertama) sudah menjadi empty dictionary ( { } )

Nama fungsi terkait : *main*

7. Menampilkan output daftar mata kuliah setiap semester

Nama fungsi terkait : `printd(dict)`

Dalam program, terdapat tambahan *user interaction* yang diberikan untuk setiap user.

Saat user menjalankan program, user dapat memasukkan nama, kemudian program akan menampilkan pilihan untuk menjalankan program atau tidak (y/n). Tampilan personalisasi ini dibungkus dalam sebuah fungsi `welcome()`.

*Source program* dalam bahasa pemrograman yang dipilih (Python)

```
#MAKE DICTIONARY
#Fungsi untuk membuat dictionary graf dari
# masukan teks yang diterima pertama kali
#Key: mata kuliah (unik);
#Value: mata kuliah prasyarat terkait
def makedict(f):
    dict = {}
    for line in f:
        i = 0
        for word in line.split():
            word1 = word.rstrip(",.")
            i += 1
            if (i==1):
                key = word1
                dict[key] = []
            else:
                dict[key].append(word1)
    return dict
```

```
#COUNT VALUE PER KEYS
#Fungsi untuk membuat dictionary
# count mata kuliah prasyarat
#Key: mata kuliah (unik);
#Value: jumlah mata kuliah prasyarat
def count(dict):
    dict_c = dict.copy()
    for k in dict_c:
        dict_c[k] = len(dict.get(k))
    return dict_c
```

```
#SELECT KEY TO BE DELETED
#Fungsi untuk menghapus mata kuliah yang
# tidak memiliki mata kuliah prasyarat
#Akan mengembalikan dictionary graf yang baru
# serta array berisi list mata kuliah yang
# di delete (memiliki nol prasyarat)
def delkey (dict1,dict_c):
    eachsem = []
    for k in dict_c:
        if dict_c.get(k) == 0:
            delete = k
            eachsem.append(delete)
            dict1.pop(delete)

            for v in iter(dict1.values()):
                for element in v:
                    if element == delete:
                        v.remove(element)

    count(dict1)
    dict_c.clear() #works!
    return dict1,eachsem
```

```
#UPDATE THE PRINT
#Membentuk dictionary dari array
# list mata kuliah dengan nol prasyarat
def update(idx,value):
    prereq = {}
    key = "Semester " + str(idx)
    prereq[key] = value
    return prereq

#PRINT DICTIONARY
#Print dictionary sesuai output yang diinginkan
def printd(dict):
    for k,v in dict.items():
        print(k," : ",end="")
        count = 0
        for item in v:
            print(item,end="")
            count += 1
            if count != len(v):
                print(", ",end="")
        print("\n")
```

```

#USER INTERACTION
def welcome():
    name = input("Hallo, what's your name? ")
    yesno = input("Do you have any trouble in defining your subjects prerequisites? (Y/N) ")
    printname = name+", "

    return yesno, printname

```

```

#MAIN
yesno = welcome()

if ((yesno[0] == "Y") and (yesno[0] == "y")):
    print("Well", yesno[1], "I'm here to help you!", "\n")
    #Meminta masukan file teks
    file = input("Insert your list of subjects in filename.txt format : ")
    print("\n")

    f = open("../test/"+file, "r")
    dict = makedict(f)

    print("=====", yesno[1][:-1].upper()+"'S", "LIST OF SUBJECTS =====", "\n")

    emptydict = {}
    prereq = {}
    idx = 1

    #Iterasi hingga dictionary graf awal menjadi empty ( {} )
    while dict != emptydict:
        dict_c = count(dict)
        new = delkey(dict, dict_c)
        updating = update(idx, new[1])
        prereq.update(updating)
        idx += 1

    printd(prereq)

else:
    print("Good then! See u next time!!")

```

### Hasil *screenshots* dari masukan dan keluaran

```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? dil
Do you have any trouble in defining your subjects prerequisites? (Y/N) y
Well dil, I'm here to help you!

Insert your list of subjects in filename.txt format : 1.txt

===== DIL'S LIST OF SUBJECTS =====

Semester 1   : Logkom, Alstrukdat, Orkom, Matdis, TBFO, Algeo
Semester 2   : RPL, OOP, Stima, Probststat, OS, Basdat
Semester 3   : AI, WBD, Jarkom, SI, MRPL, MBD, IMK
Semester 4   : MachineLearning, Sispar, PPL, Grafkom, SocioIF, PDA
Semester 5   : KP, TA1
Semester 6   : TA2
```

```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? adila
Do you have any trouble in defining your subjects prerequisites? (Y/N) Y
Well adila, I'm here to help you!

Insert your list of subjects in filename.txt format : 2.txt

===== ADILA'S LIST OF SUBJECTS =====

Semester 1   : C3
Semester 2   : C1
Semester 3   : C4
Semester 4   : C2
Semester 5   : C5, C6
```

```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? aulia
Do you have any trouble in defining your subjects prerequisites? (Y/N) y
Well aulia, I'm here to help you!
```

Insert your list of subjects in filename.txt format : 3.txt

===== AULIA'S LIST OF SUBJECTS =====

Semester 1 : C5

Semester 2 : C4

Semester 3 : C3

Semester 4 : C2

Semester 5 : C1

```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? aul
Do you have any trouble in defining your subjects prerequisites? (Y/N) y
Well aul, I'm here to help you!
```

Insert your list of subjects in filename.txt format : 4.txt

===== AUL'S LIST OF SUBJECTS =====

Semester 1 : IF03, IF05

Semester 2 : IF01, IF02, IF06

Semester 3 : IF04, IF11

Semester 4 : IF07, IF08

Semester 5 : IF09, IF10

Semester 6 : IF12, IF13



```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? alya
Do you have any trouble in defining your subjects prerequisites? (Y/N) y
Well alya, I'm here to help you!
```

```
Insert your list of subjects in filename.txt format : 5.txt
```

```
===== ALYA'S LIST OF SUBJECTS =====
```

```
Semester 1   : Cc, Ce
```

```
Semester 2   : Ca, Cb, Cf
```

```
Semester 3   : Cd, Ck
```

```
Semester 4   : Cg, Ch
```

```
Semester 5   : Ci, Cj
```

```
Semester 6   : Cl, Cm
```

```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? hafiza
Do you have any trouble in defining your subjects prerequisites? (Y/N) y
Well hafiza, I'm here to help you!
```

```
Insert your list of subjects in filename.txt format : 6.txt
```

```
===== HAFIZA'S LIST OF SUBJECTS =====
```

```
Semester 1   : Ca
```

```
Semester 2   : Cb
```

```
Semester 3   : Cc
```

```
Semester 4   : Cd
```

```
Semester 5   : Ce, Cf, Cg
```

```
Semester 6   : Ch
```

```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? hai
Do you have any trouble in defining your subjects prerequisites? (Y/N) y
Well hai, I'm here to help you!

Insert your list of subjects in filename.txt format : 7.txt

===== HAI'S LIST OF SUBJECTS =====

Semester 1   : C3, C5
Semester 2   : C1, C6
Semester 3   : C4
Semester 4   : C7
Semester 5   : C2
```

```
D:\tucil\Stima_2\TucilStima2\src>python3 toposort.py
Hallo, what's your name? ola
Do you have any trouble in defining your subjects prerequisites? (Y/N) y
Well ola, I'm here to help you!

Insert your list of subjects in filename.txt format : 8.txt

===== OLA'S LIST OF SUBJECTS =====

Semester 1   : II2130, II2110, II2111, TI3005, IF2140, IF2111
Semester 2   : II2250, II2260, II2230, II2220, II2240, IF2212
Semester 3   : II3150, II3160, II3120, II3131, II3121, IF3152
Semester 4   : II3260, II3230, II3211, II3220, II3240
Semester 5   : II4090, II4091, II4370, II4371
Semester 6   : II4092, II4472
```

**Alamat drive yang berisi kode program (eksekusi program jika diperlukan oleh asisten)**

<https://github.com/auliaadila/TucilStima2>

### **Checklists**

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	√	