

**TUGAS KECIL 2**  
**IF2211 STRATEGI ALGORITMA**  
**Penyusunan Rencana Kuliah dengan *Topological Sort***  
**(Penerapan *Decrease and Conquer*)**  
**SEMESTER 2 TAHUN 2020/2021**

oleh

Irvin Andryan Pratomo / 13519162



**PROGRAM STUDI TEKNIK INFORMATIKA**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**BANDUNG**  
**2021**

## **I. Algoritma *Topological Sort***

### **1. Penjelasan *Topological Sort* dan Kaitannya dengan *Decrease and Conquer***

*Topological sort* adalah proses pengurutan yang dilakukan dengan memperhatikan prasyarat / ketergantungan pengurutan suatu elemen dengan elemen lainnya. Proses sorting pada *topological sort* dilakukan dengan cara menghitung *in-degree* atau derajat masuk dari sebuah node pada *Directed Acrylic Graph* (DAG). Untuk mendapatkan hasil, sebuah DAG harus memiliki node sumber / node awal, yang memiliki *in-degree* = 0, artinya node awal adalah node tanpa prasyarat yang tidak bergantung kepada node lain.

Algoritma *topological sort* akan menghitung *in-degree* dari setiap node yang ada dan mencatat node-node yang menjadi prasyarat suatu node pada graph. Ketika ditemukan sebuah node yang memiliki *in-degree* = 0, misalkan benrama node “A”, maka node tersebut ditambahkan ke dalam list node hasil pengurutan. Setelah itu, node “A” dihapus dari graph, sehingga node lain yang sebelumnya memiliki prasyarat berupa node “A”, *in-degree*-nya akan berkurang. Proses ini dilakukan secara rekursif sampai semua node pada graph berhasil diurutkan pada list node hasil pengurutan.

Algoritma *topological sort* merupakan salah satu penerapan dari metode *Decrease and Conquer*. Hal ini dapat dijelaskan sebagai berikut. Pada setiap pengulangan pembacaan graph, ketika ditemukan node yang sudah memiliki *in-degree* = 0, node tersebut langsung dimasukkan ke dalam list hasil pengurutan, dihapus dari graph, dan jumlah *in-degree* node lain dengan prasyarat node yang dihapus tersebut berkurang (*Decrease*). Setelah itu akan dilakukan pembacaan lagi terhadap graph (*Conquer*), namun dengan jumlah node yang harus ditngani lebih sedikit daripada sebelumnya.

### **2. Penjelasan Langkah Kerja Program**

- a. Program pertama-tama akan membaca file masukkan menggunakan fungsi `openFile(namaFile)` (tipe parameter `namaFile` adalah sebuah string), yang membuka file masukkan, membaca isi file masukkan, dan menyimpan hasil pembacaan isi file masukkan di dalam sebuah list 2 dimensi bernama `listMataKuliah` (tanda baca yang tidak diperlukan seperti titik dan koma akan tidak akan dimasukkan ke dalam list).
- b. Contoh list yang berisi hasil pembacaan isi file mata kuliah adalah sebagai berikut :

**[[C1, C3], [C2, C1, C4], [C3], [C4, C1, C3], [C5, C2, C4]]**

Dari list tersebut, artinya adalah, mata kuliah C1 prasyaratnya adalah C3; mata kuliah C2 prasyaratnya adalah C1 dan C4; mata kuliah C3 tidak memiliki prasyarat; mata kuliah C4 prasyaratnya adalah C1 dan C3; mata kuliah C5 prasyaratnya adalah C2 dan C4.

- c. Setelah diperoleh list berisi semua mata kuliah dan prasyaratnya, program kemudian akan memanggil fungsi `getUrutanMatkul(listMataKuliah)` (tipe parameter `listMataKuliah` adalah list), yang berisi algoritma topological sorting untuk menyelesaikan persoalan.
- d. Berdasarkan `listMataKuliah` yang dimasukkan ke dalam fungsi `getUrutanMatkul`, program akan melakukan iterasi terhadap list tersebut untuk mencari `listMataKuliah[i]` yang memiliki `Panjang = 1`. `listMataKuliah[i]` dengan `Panjang = 1` artinya tidak memiliki mata kuliah prasyarat. Jika dilihat dari contoh list di atas, `listMataKuliah[i]` dengan `Panjang = 1` adalah `listMataKuliah[2]` yang isinya adalah list [3].
- e. Saat sudah menemukan `listMataKuliah[i]` dengan `Panjang = 1`, program akan memasukkan `listMataKuliah[i]` tersebut ke dalam sebuah list lain bernama `currentSemester`, yaitu list yang berisi mata kuliah yang akan diambil di semester yang sedang diproses (satu kali iterasi `listMataKuliah` adalah satu semester). Kemudian program akan melanjutkan sampai selesai membaca elemen terakhir dari `listMataKuliah`.
- f. Setelah program selesai membaca elemen terakhir `listMataKuliah`, program akan menghapus semua mata kuliah yang terdapat di dalam list `currentSemester` dari `listMataKuliah`. Mata kuliah tersebut dihapus baik sebagai mata kuliah itu sendiri maupun sebagai prasyarat mata kuliah lainnya, sehingga sekarang mata kuliah yang sebelumnya memiliki prasyarat mata kuliah yang sudah diambil akan berkurang derajat masuknya. Penghapusan ini tidak dilakukan segera setelah mata kuliah dimasukkan ke dalam list `currentSemester` untuk mencegah suatu mata kuliah diambil di semester yang sama dengan mata kuliah prasyaratnya.

- g. Setelah itu, list `currentSemester` akan di `append` ke dalam list `urutanMatkul`, yang berisi urutan mata kuliah hasil *topological sorting*. Contoh list `urutanMatkul` adalah sebagai berikut :

**[[C1, C7], [C2], [C3, C4], [C5], [C6]]**

Cara membaca list tersebut adalah, mata kuliah C1 dan C7 diambil di semester 1; mata kuliah C2 diambil di semester 2; mata kuliah C3 dan C4 diambil di semester 3; mata kuliah C5 diambil di semester 4; mata kuliah C6 diambil di semester 5.

- h. Proses pada poin d sampai g akan dilakukan secara berulang hingga list `MataKuliah` sudah kosong (semua mata kuliah sudah terurut di dalam list `urutanMatkul`).
- i. Setelah proses selesai, program akan memanggil prosedur `printUrutanMatkul(listUrutanMatkul)` (parameter `listUrutanMatkul` bertipe list), yang akan menjalankan proses untuk menampilkan luaran ke layar.

## II. Source Code

Program dibuat menggunakan bahasa Python 3.

### 1. Modul program untuk membaca file masukan (`openFile.py`)

```
import os

'''
Fungsi untuk membuka file input dan memasukkannya ke dalam list 2 dimensi bernama listOfContents
fungsi ini juga sudah menghilangkan tanda titik dan koma yang tidak akan terpakai
hasilnya adalah listOfContents[i][j]
contoh : listOfContents[1][0] = "IF1111"
listOfContents[1][1], listOfContents[1][2], ... adalah mata kuliah prasyarat dari mata kuliah "IF1111"
'''

def openFile(namaFile):
    os.chdir("../")
    os.chdir("test")
    with open(namaFile) as inputFile:
        listOfContents = [line.split() for line in inputFile]
    #membuka file dan hasilnya dimasukkan ke dalam list 2 dimensi bernama listOfContents
    #hasil yang dimasukkan sudah terbagi menjadi per baris, contoh : [[baris1], [baris2], [baris3]]

    for i in range(len(listOfContents)):
```

```

for j in range(len(listOfContents[i])):
    listOfContents[i][j] = listOfContents[i][j].strip(",.")
    #menghapus tanda koma dan titik.
return listOfContents

```

## 2. Modul program untuk *topological sort* (topoSort.py)

'''

Fungsi untuk menyelesaikan persoalan urutan mata kuliah dengan topological sort,  
 fungsi ini memiliki parameter bertipe list yang berisi nama mata kuliah dan nama mata kuliah prasyaratnya  
 fungsi ini memiliki return value berupa list 2 dimensi yang berisi urutan mata kuliah tiap semesternya,  
 contoh return : [[a,b,c], [d,e,f], [g,h,i]] maka a,b,c adalah mata kuliah  
 yang bisa diambil di semester 1; d,e,f di semester 2; dan g,h,i di semester 3

'''

```
def getUrutanMatkul(listMataKuliah):
```

```
    done = False
```

```
    #inisiasi boolean yang akan digunakan sebagai tanda selesainya proses
```

```
    urutanMatkul = []
```

```
    #inisiasi list yang akan menjadi list 2 dimensi berisi mata kuliah terurut per semester
```

```
    #list ini adalah nilai yang akan di-return oleh fungsi ini
```

```
    while (not done):
```

```
        currentSemester = []
```

```
        #inisiasi list yang akan diisi mata kuliah yang diambil di semester
```

```
        #yang sedang diproses, nantinya akan di-append ke list 'urutanMatkul'
```

```
        for i in range(len(listMataKuliah)):
```

```
            if (len(listMataKuliah[i]) == 1):
```

```
                #jika hanya ada satu elemen pada listMataKuliah[i],
```

```
                #artinya mata kuliah pada listMataKuliah[i][0] adalah
```

```
                #mata kuliah yang tidak ada prasyaratnya lagi (bisa diambil)
```

```
                currentSemester.append(listMataKuliah[i][0])
```

```
                #tambahkan mata kuliah yang sudah tidak ada prasyaratnya ke list 'currentSemester'
```

```
        for i in range(len(currentSemester)):
```

```
            for j in range(len(listMataKuliah)):
```

```
                if (currentSemester[i] in listMataKuliah[j]):
```

```
                    listMataKuliah[j].remove(currentSemester[i])
```

```
                    #menghapus mata kuliah yang diambil di semester ini dari listMataKuliah.
```

```
                    #dihapus baik sebagai mata kuliah itu sendiri maupun sebagai prasyarat mata kuliah lain.
```

#penghapusan dilakukan di sini (tidak segera setelah mata kuliah semester ini ditemukan) untuk mencegah suatu mata kuliah diambil pada semester yang sama dengan mata kuliah prasyaratnya

```
urutanMatkul.append(currentSemester)
#menambahkan mata kuliah pada listCurrentSemester ke list urutanMatkul
matkulHabis = True
#boolean untuk mengecek apakah sudah tidak ada mata kuliah yang belum diambil di listMatakuliah
for k in range(len(listMataKuliah)):
    if (len(listMataKuliah[k]) != 0):
        matkulHabis = False #jika ada yang panjangnya > 0 masih ada matkul yang belum diambil
if (matkulHabis == True): #jika sudah tidak ada matkul yang belum diambil, selesai
    done = True
return urutanMatkul
```

### 3. Modul program untuk mengeluarkan tampilan ke layar (printMatkul.py)

'''

Prosedur untuk print urutan mata kuliah yang diambil ke layar. contoh hasil :

Semester 1 : C1.

Semester 2 : C3, C5.

Semester 3 : C4.

Semester 4 : C2.

'''

```
def printUrutanMatkul(listUrutanMatkul):
    for i in range(len(listUrutanMatkul)):
        print("Semester", i+1, "- ", end = "")
        for j in range(len(listUrutanMatkul[i])):
            if (j < len(listUrutanMatkul[i]) - 1):
                print(listUrutanMatkul[i][j], end = ', ')
            else:
                print(listUrutanMatkul[i][j], end = '.')
        print("")
```

### 4. Main program (Tucil2\_13519162.py)

```
from openFile import openFile
from topoSort import getUrutanMatkul
from printMatkul import printUrutanMatkul
```

'''

### Main Program

'''

programRun = True #selama true program akan berjalan terus

while (programRun):

namaFile = str(input("Masukkan nama file : ")) #nama file input

listMataKuliah = openFile(namaFile) #membuka file dan memasukkan isinya ke listMataKuliah

listUrutanMatkul = getUrutanMatkul(listMataKuliah) #mendapatkan urutan matkul, dan disimpan di

listUrutanMatkul

printUrutanMatkul(listUrutanMatkul) #print urutan matkul ke layar

print("")

print("Apakah ingin memasukkan file baru?")

fileBaru = str(input("Masukkan pilihan [Y / N] : "))

while(fileBaru != 'N' and fileBaru != 'n' and fileBaru != 'Y' and fileBaru != 'y'):

print("Masukkan salah!")

fileBaru = str(input("Masukkan pilihan [Y / N] : "))

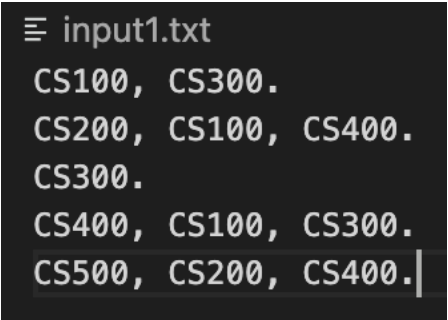
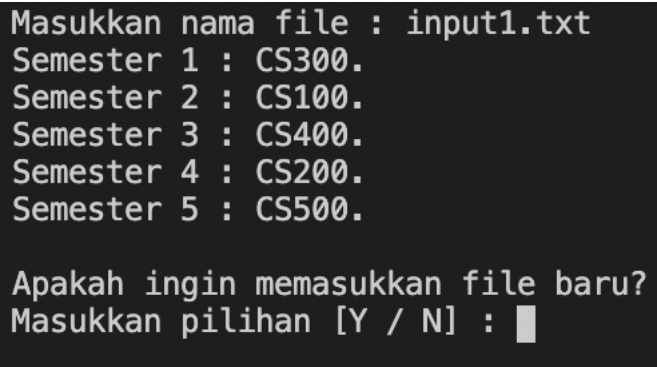
if (fileBaru == 'N' or fileBaru == 'n'):

programRun = False #program selesai

else:

continue #program terus berjalan

### III. Testing

INPUT	OUTPUT
 <pre>≡ input1.txt CS100, CS300. CS200, CS100, CS400. CS300. CS400, CS100, CS300. CS500, CS200, CS400.</pre>	 <pre>Masukkan nama file : input1.txt Semester 1 : CS300. Semester 2 : CS100. Semester 3 : CS400. Semester 4 : CS200. Semester 5 : CS500.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] : █</pre>

<pre> ≡ input2.txt CS100. CS200, CS100. CS300, CS200. CS400, CS200, CS700. CS500, CS200, CS300, CS400. CS600, CS500. CS700. </pre>	<pre> Masukkan nama file : input2.txt Semester 1 : CS100, CS700. Semester 2 : CS200. Semester 3 : CS300, CS400. Semester 4 : CS500. Semester 5 : CS600.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] :  </pre>
<pre> ≡ input3.txt CS100. CS200, CS100. CS300, CS200, CS600. CS400, CS200. CS500, CS400. CS600, CS400, CS500. </pre>	<pre> Masukkan nama file : input3.txt Semester 1 : CS100. Semester 2 : CS200. Semester 3 : CS400. Semester 4 : CS500. Semester 5 : CS600. Semester 6 : CS300.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] :  </pre>
<pre> ≡ input4.txt CS270. CS381, CS270, CS150. CS390, CS381. CS150. CS350, CS361. CS170. CS250, CS150. CS361, CS250, CS270. CS355, CS250, CS381. </pre>	<pre> Masukkan nama file : input4.txt Semester 1 : CS270, CS150, CS170. Semester 2 : CS381, CS250. Semester 3 : CS390, CS361, CS355. Semester 4 : CS350.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] :  </pre>



<pre> ≡ input5.txt CS100. CS150. CS250, CS151. CS220, CS100, CS150. CS151, CS150. CS230, CS151. CS466, CS230. CS360, CS220. CS490, CS360, CS477. CS477, CS220. </pre>	<pre> Masukkan nama file : input5.txt Semester 1 : CS100, CS150. Semester 2 : CS220, CS151. Semester 3 : CS250, CS230, CS360, CS477. Semester 4 : CS466, CS490.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] :  </pre>
<pre> ≡ input6.txt CS100. MA200. MA600. MA250, MA200, MA600. CS110, CS100. CS120, CS110, MA200. CS510, MA250, CS120. CS160, CS120. CS270, CS120. CS180, CS160, CS510. </pre>	<pre> Masukkan nama file : input6.txt Semester 1 : CS100, MA200, MA600. Semester 2 : MA250, CS110. Semester 3 : CS120. Semester 4 : CS510, CS160, CS270. Semester 5 : CS180.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] :  </pre>
<pre> ≡ input7.txt CS100. CS200. CS300, CS100, CS200. CS400. CS500, CS200, CS400. CS600, CS400. CS700, CS400. CS800, CS600, CS700. CS900, CS800. CS110, CS300, CS900. </pre>	<pre> Masukkan nama file : input7.txt Semester 1 : CS100, CS200, CS400. Semester 2 : CS300, CS500, CS600, CS700. Semester 3 : CS800. Semester 4 : CS900. Semester 5 : CS110.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] :  </pre>

<pre> ≡ input8.txt CS100. CS200. CS300, CS400. CS400, CS100, CS200. CS500, CS200. CS600, CS200. CS700, CS600. MA100, CS400. MA200, CS400. MA300, CS400, CS600. MA400, CS400, CS500, CS600. MA500, CS400, MA400. SE100, MA300. SE200, MA200, SE300. SE300, MA300. SE400, MA300, MA400. SE500, SE400. </pre>	<pre> Masukkan nama file : input8.txt Semester 1 : CS100, CS200. Semester 2 : CS400, CS500, CS600. Semester 3 : CS300, CS700, MA100, MA200, MA300, MA400. Semester 4 : MA500, SE100, SE300, SE400. Semester 5 : SE200, SE500.  Apakah ingin memasukkan file baru? Masukkan pilihan [Y / N] : █ </pre>
--	---

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

Repository berisi source code :

<https://github.com/irvinandryan/Decrease-Conquer>