

Laporan Tugas Kecil II IF2211 Strategi Algoritma Semester 2
Tahun 2020/2021

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



Disusun oleh :
Ryandito Diandaru
13519157

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

Algoritma *Topological Sort* dan Kaitannya dengan Pendekatan *Decrease and Conquer*

Topological sort adalah algoritma pengurutan sebuah *Directed Acyclic Graph* (DAG) yang diurutkan berdasarkan *predecessor-nya*, dalam kata lain berdasarkan pengambilan *node* yang memiliki nilai derajat masuk 0 karena sudah tidak memiliki *predecessor*. Pada saat derajat masuk sebuah *node* sudah mencapai nilai 0, *node* tersebut akan dihilangkan dari graf, dan proses akan berlanjut sampai semua *node* dalam graf dihilangkan. Pendekatan seperti ini menunjukkan aplikasi dari strategi ***Decrease and Conquer*** yang berarti mengurangi besar input/permasalahan untuk setiap laju pemrosesan. Dalam kasus ini diberikan sebuah daftar mata kuliah yang dapat diambil oleh seorang mahasiswa dalam program bernama “ENEM : Program Pemilihan Rencana Studi”. Masukan dari program berupa text file yang berisi daftar mata kuliah beserta prerequisitenya dalam format seperti contoh sebagai berikut:

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

dengan C1, C2, C3, C4, C5 adalah nama mata kuliah bertipe data *string*. Program akan membuat **objek matkul** yang berisi informasi tentang mata kuliah dan sebuah **objek graph** untuk menyimpan objek-objek matkul yang sudah dibuat untuk setiap matkul yang berada dalam file .txt. Program mengimplementasikan *Topological Sorting* dengan algoritma rekursif yang akan menerima sebuah objek graph dan sebuah integer untuk menandakan semester. Fungsi rekursif yang dibuat memiliki basis ketika semua matkul dalam *graph* sudah memiliki derajat masuk 0, dalam kasus itu, program akan mencetak nomor semester beserta semua matkul dalam *graph* dan menghentikan rekursi. Bagian rekurens dari fungsi akan mencetak dan men-*delete* matkul-matkul yang sudah memiliki derajat masuk 0, diikuti dengan mengurangi derajat masuk dari objek-objek matkul lain yang mejadikan matkul tersebut *prerequisite*. Selanjutnya, program akan memanggil fungsi itu sendiri dengan parameter graf yang sudah dipangkas dan semester yang di-*increment* nilainya dengan 1. Dengan begitu, algoritma rekursif yang diimplementasikan merupakan penerapan strategi ***Decrease and Conquer by a variable size***, karena pengurangan jumlah *node* dalam *graph* bervariasi tergantung *prerequisite* masing-masing objek matkul.

Source Code Program dalam Bahasa Python

```
#ENEM : Program Pemilihan Rencana Studi
#Ryandito Diandaru
#13519157
#K03

#class node matkul
class matkul:
    def __init__(self, raw):#inisialisasi node matkul
        self.name = raw[0]          #nama matkul
        self.prereq = raw[1:len(raw)] #nama-nama matkul prerequisite
        self.inDegree = len(raw)-1   #jumlah derajat masuk

    def changeInDegree(self, n):#method mengubah indegree
        self.inDegree += n

#class graph
class graph:
    def __init__(self):#inisialisasi graph
        self.nodes = []

    #menambah node yang berupa objek matkul
    def addNode(self, nod):
        self.nodes.append(nod)

    #menghilangkan course dari graph, asumsi matkul sudah diambil
    def popNode(self, nod):
        for i in range(len(self.nodes)):
            #mengambil indeks dari matkul yang akan dihilangkan dari graph
            if(self.nodes[i].name == nod.name):
                todelete = i
            #mengurangi in degree dari matkul matkul yang
            #menjadikan matkul sekarang prereq
            if (nod.name in self.nodes[i].prereq):
                self.nodes[i].changeInDegree(-1)
        #mendelete matkul dari graph karena asumsi sudah diambil
        self.nodes.pop(todelete)

    #mengecek apakah semua matkul yang ada di graf
    #sudah memiliki in degree 0 untuk basis rekursi
    def checkAllZeroInDegree(self):
        i = 0
        while(i < len(self.nodes)):
            if(self.nodes[i].inDegree != 0):
                return False
            i += 1
        return True
```

```

#membuat objek graph
graeph = graph()
#membuka file
namafile = input("Masukkan nama file :")
file = open("../test/"+namafile, "r")
for c in file:
    #menghapus karakter yang tidak perlu
    node = c.replace(' ', '').replace('.', '').replace('\n', '').split(",")
    #memasukkan node matkul ke dalam graph
    graeph.addNode(matkul(node))

#Decrease and conquer
def topoSort(graf, sem):
    #basis
    if(graf.checkAllZeroInDegree()):
        #Mengeprint semua matkul karena semua inDegree sudah 0
        print("Semester ", sem, "\t:", end='')
        for i in graf.nodes:
            print(i.name, end = ', ')
        print("\b\b", end='')
        print(" ")

    #rekurens
    else:
        #inisialisasi dan cetak "Semester"
        i = 0
        todelete = graph()
        print("Semester ", sem, "\t:", end='')

        #mencetak matkul yang memiliki inDegree 0
        while(i < len(graf.nodes)):
            if(graf.nodes[i].inDegree == 0):
                print(graf.nodes[i].name, end=', ')
                #pencatatan matkul ke todelete jika
                #inDegree 0 karena matkul akan didelete
                todelete.addNode(graf.nodes[i])
            i += 1
        print("\b\b", end='')
        print(" ")

        #mendelete semua matkul yang ada dalam todelete
        for d in todelete.nodes:
            graf.popNode(d)

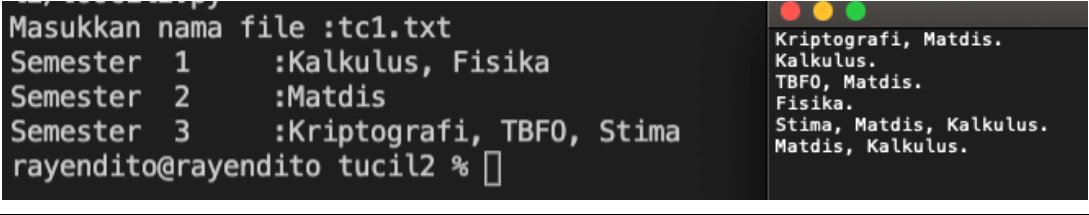
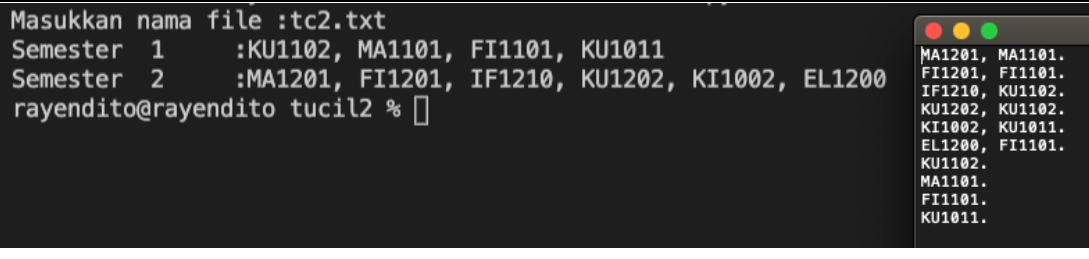
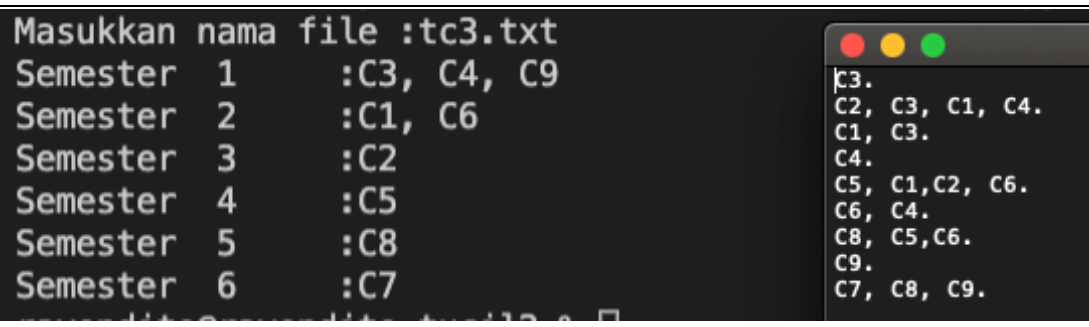
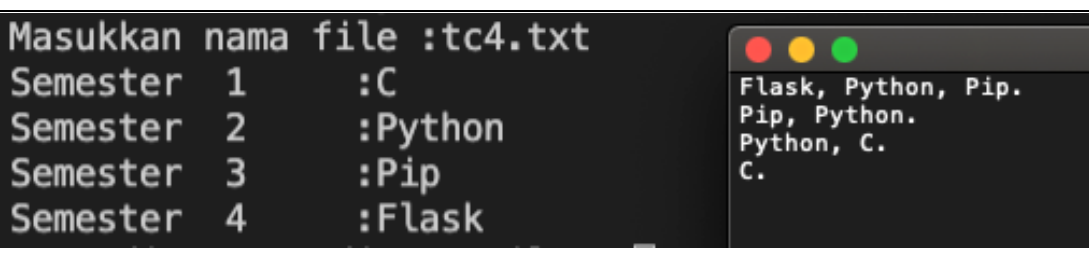
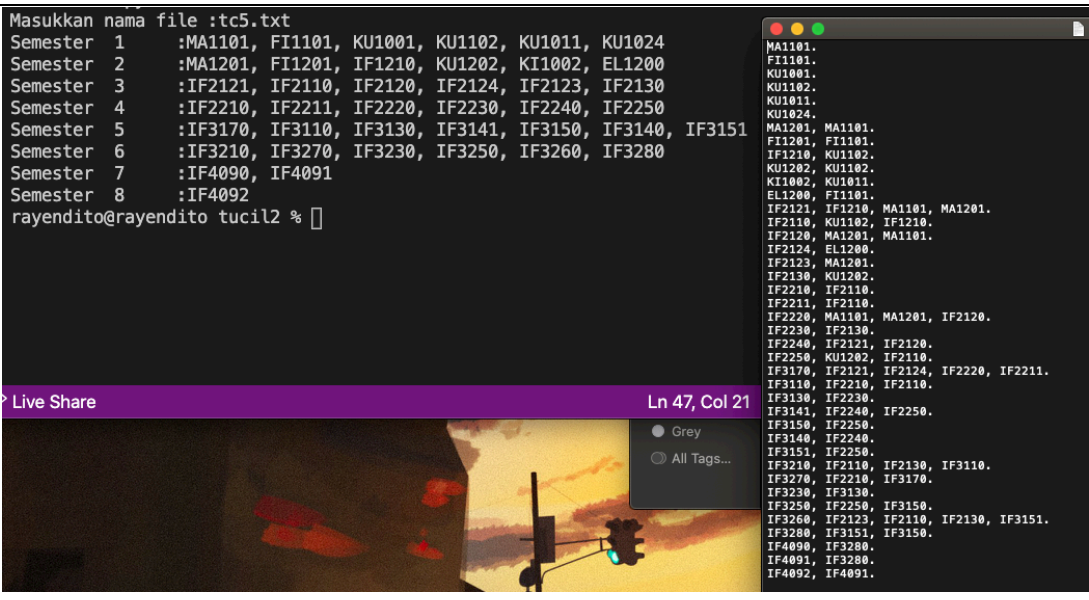
        #pemanggilan fungsi rekursi dengan graf baru yang ukurannya telah
        didecrease
        #variabel "sem" untuk menunjukkan nomor semester
        topoSort(graf, sem+1)

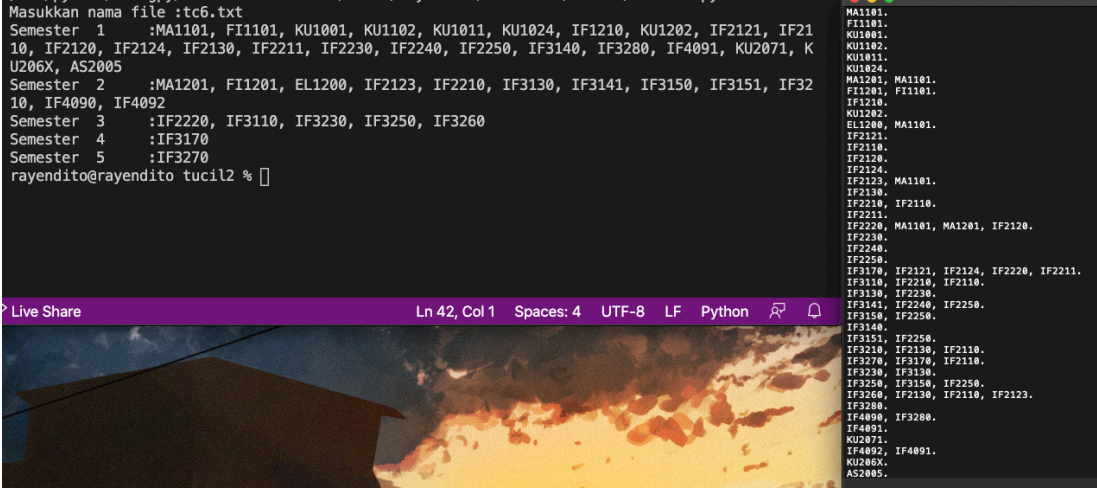
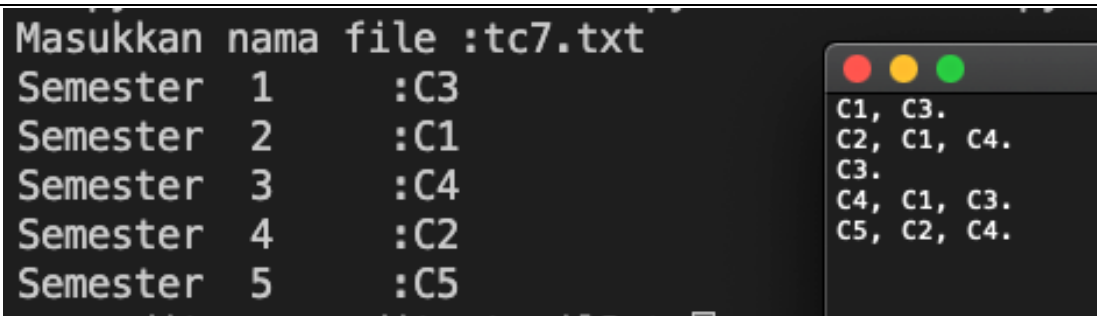
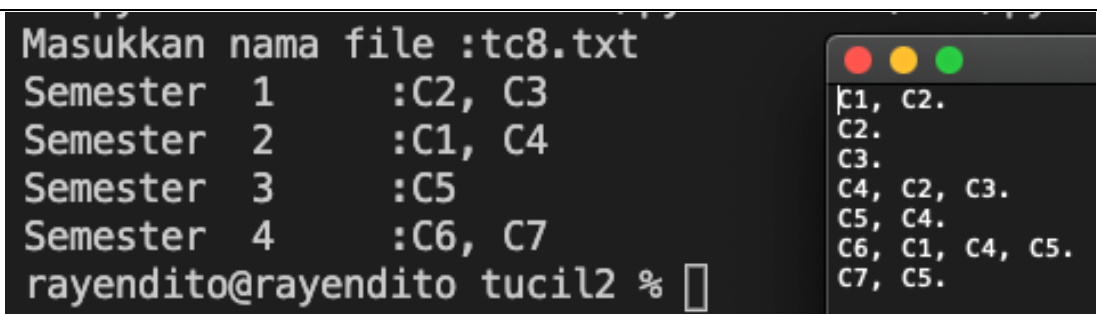
#pemanggilan fungsi
topoSort(graeph, 1)

```

Tangkapan Layar dari *Input* dan *Output*

Tabel 3.1, Hasil percobaan dengan 8 buah *test case*

No.	Hasil screenshot
1	
2	
3	
4	
5	

6	<pre> Masukkan nama file :tc6.txt Semester 1 :MA1101, FI1101, KU1001, KU1102, KU1011, KU1024, IF1210, KU1202, IF2121, IF21 10, IF2120, IF2124, IF2130, IF2211, IF2230, IF2240, IF2250, IF3140, IF3280, IF4091, KU2071, K U206X, AS2005 Semester 2 :MA1201, FI1201, EL1200, IF2123, IF2210, IF3130, IF3141, IF3150, IF3151, IF32 10, IF4090, IF4092 Semester 3 :IF2220, IF3110, IF3230, IF3250, IF3260 Semester 4 :IF3170 Semester 5 :IF3270 rayendito@rayendito tucil2 % </pre> 
7	<pre> Masukkan nama file :tc7.txt Semester 1 :C3 Semester 2 :C1 Semester 3 :C4 Semester 4 :C2 Semester 5 :C5 </pre> 
8	<pre> Masukkan nama file :tc8.txt Semester 1 :C2, C3 Semester 2 :C1, C4 Semester 3 :C5 Semester 4 :C6, C7 rayendito@rayendito tucil2 % </pre> 

LAMPIRAN

Alamat source code program:

<https://drive.google.com/drive/folders/1AG-PI2JE4h1G4TsbrmPwCrcOut2jMel4?usp=sharing>

Tabel 4.1, Checklist

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