

**TUGAS KECIL 2**  
**IF2211 STRATEGI ALGORITMA**  
**Penyusunan Rencana Kuliah dengan *Topological Sort***  
**(Penerapan *Decrease and Conquer*)**

**IF2211 STRATEGI ALGORITMA**  
**Semester II Tahun Ajaran 2020/2021**

Dosen Pengampu : Ir. Rila Mandala, M.Eng., Ph.D.



Disusun oleh :

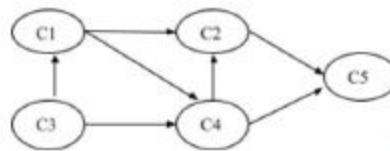
Ridho Daffasyah                      13519038

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

## I. ALGORITMA TOPOLOGICAL SORT DENGAN PENDEKATAN DECREASE AND CONQUER

Di dalam tugas kali ini, penyelesaian dari program harus menggunakan algoritma *Topological Sort* dengan pendekatan *decrease and conquer*. Algoritma *Topological Sort* adalah pengurutan atau pengaturan secara topologi dari graf berarah yang dimana pengurutan simpulnya dilakukan secara linear sehingga untuk setiap arah uv dari vertex u ke vertex v, u datang sebelum v. Simpul dari grafik dapat mewakili tugas yang harus dilakukan, dan tepi dapat mewakili batasan bahwa satu tugas harus dilakukan sebelum tugas lainnya.

*Topological Sort* selalu dilakukan dari simpul tanpa derajat masuk. Contoh, misalkan terdapat graf seperti :



Dari graf diatas, *Topological Sort* dilakukan dari simpul C3 yang merupakan simpul tanpa derajat masuk yang berarti hasil dari pengurutannya adalah C3, C1, C4, C2, C5. Untuk mendapatkan hasil pengurutan tersebut dapat dilakukan dengan menggunakan algoritma *decrease and conquer*. Berikut langkah-langkahnya dalam membuat program ini :

1. Dari inputan atau masukan berupa :

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

Ubah inputan diatas menjadi graf dengan representasi array 2D, seperti :

Array = [[C1,C3], [C2,C1,C4],[C3],[C4,C1,C3],[C5,C2,C4]]

2. Lalu, hitung semua derajat-masuk setiap simpul, yaitu banyaknya busur yang masuk pada simpul. Simpul direpresentasikan sebagai elemen dari baris 1 dan kolom 1 atau Array[0][0] dari array 2D dan derajat masuk dapat dihitung dengan melakukan pengecekan apakah panjang dari elemen baris ke i adalah 1 atau bukan. Karena jika panjang tersebut sama dengan 1 maka tandanya elemen tersebut tidak punya derajat masuk.
3. Catat simpul atau elemen yang tidak punya derajat masuk ke dalam array hasil.
4. Dan hilangkan simpul atau elemen tersebut beserta dengan semua busur yang keluar dari simpul tersebut pada graf dan kurangi derajat simpul dari elemen baris i yang merupakan elemen yang tidak punya derajat masuk. Contoh, dari inputan

diatas dapat ditentukan elemen baris i yang tidak punya derajat masuk (panjangnya sama dengan satu) adalah C3, lalu catat C3 ke dalam array hasil, dan hilangkan elemen array[i][j] yang sama dengan C3 sehingga didapat array baru yang berisikan [[C1], [C2,C1,C4],[C4,C1],[C5,C2,C4]].

5. Ulangi langkah 2,3,4 hingga semua simpul terpilih. Jika dilakukan dengan benar maka array hasil akan berisikan [C3, C1, C4, C2, C5]

## II. SOURCE CODE PROGRAM

### 1. Source Code 13519038-toposort.py

```
#Nama : Ridho Daffasyah
#NIM : 13519038
#Kelas : 01
#Tugas Kecil 2 - IF2211 Strategi Algoritma

global hasilTopo, simpan_matkul
hasilTopo = []
simpan_matkul = []

def pilihHapusSimpulDerajatNol(list_matkul):

    derajat_simpul = []
    x = 0
    while (x < len(list_matkul)):
        if (len(list_matkul[x]) == 1):
            derajat_simpul = 0
            #derajat simpul nol dapat dihitung dari (len(x))
            x+=1

    simpul_derajat_nol = []

    i = 0
    while(i < len(list_matkul)):
        if (len(list_matkul[i]) == derajat_simpul+1):
            #simpan simpul berderajat nol
            simpul_derajat_nol.append(list_matkul[i][0])
            #tempat simpan matkul sementara
            simpan_matkul.append(list_matkul[i][0])
            i+=1

    #Hapus simpul yang berderajat nol (matakuliah tanpa prereq) dari list
    matakuliah
    j = 0
```

```

while (j < len(simpul_derajat_nol)):
    list_matkul.remove([simpul_derajat_nol[j]])
    j+=1

#Catat mata kuliah yang derajat nol ke array
hasilTopo.append(simpul_derajat_nol)

def pendekatanTopologicalSort(list_matkul):
    #Representasi graf pada array 2D
    #[[simpul 1, derajat_masuk 1],
    # [simpul 2, derajat_masuk 1, derajat_masuk 2], dst]

    #kode_kuliah= simpul
    #kode kuliah prasyarat = derajat_masuk

    #Pendekatan Topological Sorting
    # 1. Dari graf (DAG) yang terbentuk, hitung semua derajat-masuk (in-degree)
    setiap simpul,
    # yaitu banyaknya busur yang masuk pada simpul tersebut
    # 2. Pilih sembarang simpul yang memiliki derajat-masuk 0.
    pilihHapusSimpulDerajatNol(list_matkul)

    # 3. 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.
    #Hapus simpul yang bersisian dengan simpul berderajat nol yang dipilih
    j = 0
    while (j < len(simpan_matkul)):
        y = 0
        while (y < len(list_matkul)):
            if (list_matkul[y].count(simpan_matkul[j]) == 1):
                list_matkul[y].remove(list_matkul[y][list_matkul[y].index(simpan_matkul[j])])
                y += 1
            j += 1

    #basis jika sudah tidak ada matakul yang harus diambil
    if (len(list_matkul) == 0):
        return
    else :
        #rekursif
        #Ulangi langkah (2) dan (3) hingga semua simpul pada DAG terpilih.
        pendekatanTopologicalSort(list_matkul)

```

## 2. Source Code 13519038-main.py

```
#Nama : Ridho Daffasyah
#NIM : 13519038
#Kelas : 01
#Tugas Kecil 2 - IF2211 Strategi Algoritma

import string
import os.path
topoSort = __import__('13519038-toposort')

def bacaInput(inputFile):

    #Input File
    f = open(os.path.dirname(__file__) + '/../test/' + inputFile)
    #Deklarasi list matkul dan prereq
    daftar_matkul_dan_prereq = []

    for x in f:
        #Menjadikan list matkul dan prereq ke list 2D
        daftar_matkul_dan_prereq.append(x.strip().split('>'))

    x = daftar_matkul_dan_prereq
    #Menghilangkan punctuation
    x = [''.join(c for c in s if c not in string.punctuation) for s in x]

    daftar_matkul_dan_prereq2 = []
    for i in x :
        #Menghilangkan "\n"
        daftar_matkul_dan_prereq2.append(i.strip())

    daftar_matkul_dan_prereq3 = []
    for x in daftar_matkul_dan_prereq2:
        #Menghilangkan "." dan memisahkan elemen dengan ","
        y = x.replace(".", "").replace(" ", "").split(",")
        daftar_matkul_dan_prereq3.append(y)

    #return array matkul final
    return daftar_matkul_dan_prereq3

def printRencanaKuliah():
```

```

i = 0
while (i < len(topoSort.hasilTopo)):
    print("Semester", i+1, ":", end="")
    j = 0
    while(j < len(topoSort.hasilTopo[i])):
        print(topoSort.hasilTopo[i][j], end=" ")
        j += 1
    i += 1
    print("\r")

def mainProgram():
    print("Masukkan nama file : ", end="")
    x = input()
    matkul = bacaInput(x)
    topoSort.pendekatanTopologicalSort(matkul)
    printRencanaKuliah()

mainProgram()

```

### III. SCREENSHOT INPUT OUTPUT

#### 1. test1.txt

Input :	Output :
	Masukkan nama file : test1.txt
C1, C3.	Semester 1 :C3
C2, C1, C4.	Semester 2 :C1
C3.	Semester 3 :C4
C4, C1, C3.	Semester 4 :C2
C5, C2, C4.	Semester 5 :C5
	>>>

#### 2. test2.txt

Input :	Output :
	Masukkan nama file : test2.txt
C1, C2.	Semester 1 :C2
C2.	Semester 2 :C1
C3, C1, C4.	Semester 3 :C4
C4, C1, C2.	Semester 4 :C3
C5, C3, C4.	Semester 5 :C5
	>>>

### 3. test3.txt

Input :	Output :
C1.	Masukkan nama file : test3.txt
C2, C1.	Semester 1 :C1
C3, C1.	Semester 2 :C2 C3
C4, C2.	Semester 3 :C4 C5
C5, C2.	>>>

### 4. test4.txt

Input :	Output :
MA1101.	Masukkan nama file : test4.txt
MA1201, MA1101.	Semester 1 :MA1101
IF2120, MA1201, MA1101.	Semester 2 :MA1201 IF2123
IF2123, MA1101.	Semester 3 :IF2120
IF2220, IF2120, MA1201, MA1101.	Semester 4 :IF2220
	>>>

### 5. test5.txt

Input :	Output :
C1.	Masukkan nama file : test5.txt
C2, C1.	Semester 1 :C1
C3, C1.	Semester 2 :C2 C3
C4, C2.	Semester 3 :C4 C6
C5, C2, C6.	Semester 4 :C5
C6, C3.	Semester 5 :C7
C7, C4, C5, C6.	>>>

### 6. test6.txt

Input :	Output :
C1.	Masukkan nama file : test6.txt
C2, C1.	Semester 1 :C1
C3, C1.	Semester 2 :C2 C3
C4, C1, C2, C3.	Semester 3 :C4
C5, C1, C3, C4.	Semester 4 :C5
	>>>

## 7. test7.txt

Input :

C1.  
C2.  
C3.  
C4.  
C12, C1, C2.  
C13, C1, C3.  
C14, C1, C4.  
C23, C2, C3.  
C24, C2, C4.  
C34, C3, C4.

Output :

```
Masukkan nama file : test7.txt
Semester 1 :C1 C2 C3 C4
Semester 2 :C12 C13 C14 C23 C24 C34
>>>
```

## 8. test8.txt

Input :

C1, C2, C3.  
C2, C3, C4.  
C3, C10.  
C4, C7.  
C5, C4, C6, C7, C3, C1.  
C6, C7, C8, C9.  
C7, C9.  
C8, C10.  
C9, C8, C10.  
C10.

Output :

```
Masukkan nama file : test8.txt
Semester 1 :C10
Semester 2 :C3 C8
Semester 3 :C9
Semester 4 :C7
Semester 5 :C4 C6
Semester 6 :C2
Semester 7 :C1
Semester 8 :C5
>>>
```

## 9. test9.txt

Input :

C1, C2, C3.  
C2.  
C3, C5, C6.  
C4, C3.  
C5.  
C6.  
C7, C5, C6.

Output :

```
Masukkan nama file : test9.txt
Semester 1 :C2 C5 C6
Semester 2 :C3 C7
Semester 3 :C1 C4
>>>
```



#### 10. test10.txt

Input :

C0, C1.  
C1, C2, C3, C4.  
C2, C5.  
C3, C9.  
C4, C6.  
C5, C7, C8.  
C6, C7.  
C7, C9.  
C8.  
C9, C10, C11.  
C10, C12, C13.  
C11, C13.  
C12.  
C13.

Output :

```
Masukkan nama file : test10.txt
Semester 1 :C8 C12 C13
Semester 2 :C10 C11
Semester 3 :C9
Semester 4 :C3 C7
Semester 5 :C5 C6
Semester 6 :C2 C4
Semester 7 :C1
Semester 8 :C0
>>>
```

#### IV. ALAMAT DRIVE KODE PROGRAM

[https://github.com/ridhodaffasyah/Tucil2\\_13519038.git](https://github.com/ridhodaffasyah/Tucil2_13519038.git)

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