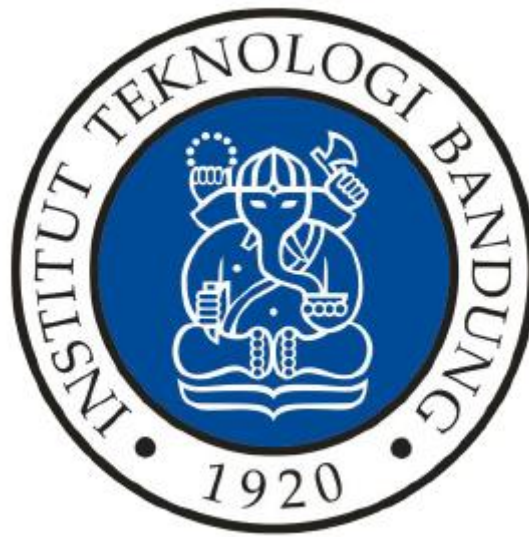


**Laporan Tugas Kecil II IF2211 Strategi Algoritma
Semester 2 Tahun Akademik 2020/2021**

PRaKTiS

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



Disusun Oleh:

**Wilson Tandy
13519209
K-04**

**Program Studi S1 Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021**

A. Algoritma *Topological Sort* dengan Pendekatan *Decrease and Conquer*

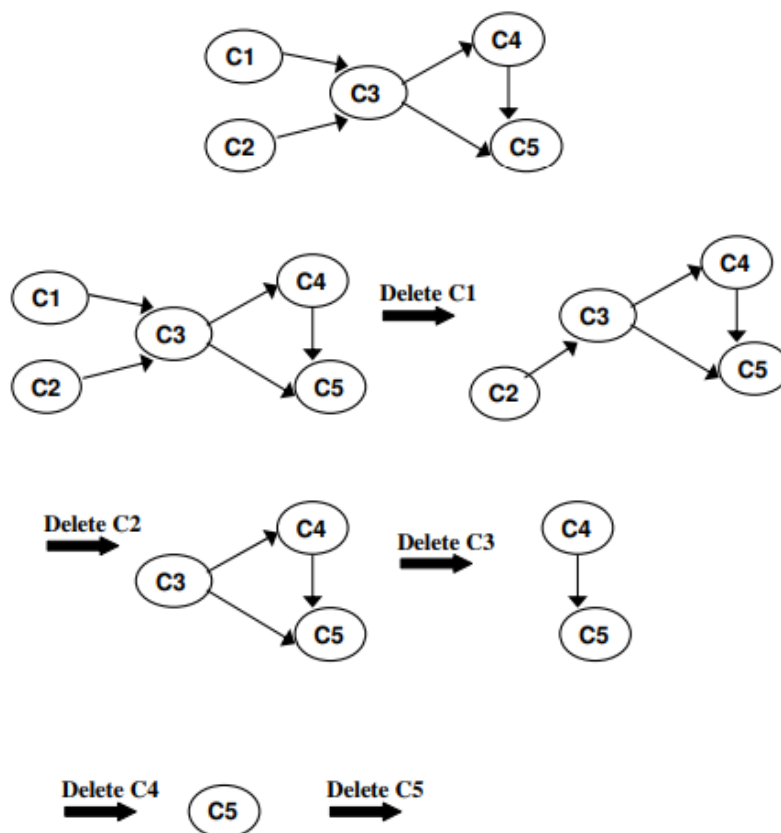
Topological Sort atau yang juga dikenal dengan *Topological Ordering* dari suatu graf berarah merupakan urutan dari simpul-simpulnya, sedemikian sehingga untuk setiap *edge* berarah uv , u selalu berada sebelum v dalam urutannya. Pada tugas kecil 2, hal ini diaplikasikan pada pengambilan mata kuliah yang memiliki *prerequisite*.

Langkah-langkah algoritma *topological sort*:

1. Akan dibuat DAG (*Directed Acyclic Graph*) dari file input .txt yang berisi daftar mata kuliah beserta prasyaratnya.
2. Memilih mata kuliah yang diambil pada semester 1, semester 2, dst. berdasarkan derajat-masuk dari mata kuliah yang direpresentasikan dengan simpul.
3. Menghapus simpul mata kuliah yang terpilih beserta semua busur yang keluar dari simpul tersebut, dan mengurangi derajat-masuk dari simpul-simpul yang ditunjuk tersebut.
4. Ulangi langkah (2) dan (3) hingga semua simpul terpilih.

Kaitannya dengan *decrease and conquer* adalah dengan menggunakan *topological sort* ini, diambil sebuah simpul dengan *in-degree* 0 lalu diselesaikan *sorting* pada simpul tersebut, dan kembali diulangi terus hingga semua simpul telah terpilih.

Ilustrasi *topological sorting* dengan pendekatan *decrease and conquer* (dengan asumsi DAG telah dibentuk) :



Topological order yang terbentuk adalah **C1, C2, C3, C4, C5**

B. Source code program PRaKTiS dalam bahasa pemrograman *Python*

```
import os

#Membaca file
nama_file = input("Masukkan nama file: ")
if os.name=='nt':
    file_path = os.path.join("../test", nama_file)
else:
    file_path = os.path.join("test", nama_file)
f = open(file_path, "r")

A=f.read().split('\n')

#Menyimpan list matkul unik pada matkul []
#Menyimpan dalam bentuk DA Graph pada edges []
matkul = []
edges = []
for line in range(len(A)):
    j=0
    #menghitung berapa koma yang telah dilewati
    #bila 0 maka menjadi simpul tujuan dari <matkul> pada baris tersebut)
    koma = 0
    #menyimpan kode matkul
    temp=""
    #mengecek apakah masih ada <matkul_x> pada line
    ada = True
    while(ada):
        #Bila ditemui tanda koma, maka akan melompat 2 karakter
        #Karena format pada .txt <matkul_1>, <matkul_2>, .....
        if (A[line][j] == ','):
            if temp not in matkul:
                matkul.append(temp)
            if (koma == 0):
                dest = temp
            else:
                edges.append((temp,dest))
            temp = ""
            j+=2
            koma+=1

        #Bila ditemui tanda titik akan dilanjutkan ke baris berikutnya
        elif (A[line][j] == '.'):
            if temp not in matkul:
                matkul.append(temp)
            #prevent bila dalam 1 baris hanya ada 1 mata kuliah
            if (koma != 0):
                edges.append((temp,dest))
            ada = False
        else:
            temp+=(A[line][j])
            j+=1

#Membuat indegreelist berdasarkan urutan mata kuliah di list matkul[]
indegreelist = [0 for i in range (len(matkul))]
for i in range (len(matkul)):
    for (src,dest) in edges:
        if (matkul[i] == dest):
            indegreelist[i]+=1
```

```

semester = 1
#sisaMatkul diperlukan untuk matkul sisa yang belum diambil(saat edges sudah kosong)
sisaMatkul = matkul.copy()
#dituliskan maksimum 8 semester asumsinya
while (edges != [] and semester<8):
    print("Semester", semester, ':')
    #tempIndegree diperlukan agar saat penghapusan edge tidak langsung ter-update
    #dapat membuat course dan pre req course diambil bersamaan
    tempIndegree = indegreelist.copy()
    for counter, i in enumerate(indegreelist):
        if (i == 0) :
            #penanda agar tidak terbaca 2 kali (diberi -999)
            tempIndegree[counter] -= 999
            #Menampilkan matkul yang diambil
            print(matkul[counter], end=' ')
            #Hapus matkul yang telah diambil
            sisaMatkul.remove(matkul[counter])
            for (src,dest) in edges:
                if (src == matkul[counter]):
                    #Hapus edge dari list
                    indexEdges = 0
                    for (src,dest) in edges:
                        if (src == matkul[counter]):
                            #Kurangi indegree dest yang akan di-pop
                            for j in range (len(matkul)):
                                if(dest == matkul[j]):
                                    tempIndegree[j] -= 1
                            #Decrease list edges
                            edges.pop(indexEdges)
                            indexEdges += 1
    print("")
    semester+=1
    #update indegreelist
    indegreelist = tempIndegree


#menampilkan sisa matkul yang butuh diambil pada semester terakhir
print("Semester", semester, ':')
for i in sisaMatkul:
    print(i, end=' ')

print("")

```

C. Tangkapan Layar Hasil *Testing* Program PRaKTiS (*Input & Output*)

Test Case 1


 test1 - Notepad

File Edit Format

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

```
Masukkan nama file: test1.txt
Semester 1 :
C3
Semester 2 :
C1
Semester 3 :
C4
Semester 4 :
C2
Semester 5 :
C5
```

Test Case 2


 test2 - Notepad

File Edit Format

IF1, IF3.
IF2, IF1, IF4.
IF3.
IF4, IF3.

```
Masukkan nama file: test2.txt
Semester 1 :
IF3
Semester 2 :
IF1 IF4
Semester 3 :
IF2
```

Test Case 3

 test3 - Notepad

File Edit Format

MA1, MA2.


MA2, MA3.

MA3, MA4.

MA4.

```
Masukkan nama file: test3.txt
Semester 1 :
MA4
Semester 2 :
MA3
Semester 3 :
MA2
Semester 4 :
MA1
```

Test Case 4

 test4 - Notepad

File Edit Format

IF1, II1.

II1, IF2.


IF2, IF3, II3.

II3.

IF3.

```
Masukkan nama file: test4.txt
Semester 1 :
IF3 II3
Semester 2 :
IF2
Semester 3 :
II1
Semester 4 :
IF1
```

Test Case 5

 test5 - Notepad

File Edit Format

FI1, FI3.


FI2, FI4.

FI1, FI3.

FI4.

```
Masukkan nama file: test5.txt
Semester 1 :
FI3 FI4
Semester 2 :
FI1 FI2
```

Test Case 6

 test6 - Notepad

File Edit Format View Help

KU2000, KU3030.


IF2030, KU2000, KU3000.

KU3000.

IF3000, KU3030.

```
Masukkan nama file: test6.txt
Semester 1 :
KU3030 KU3000
Semester 2 :
KU2000 IF3000
Semester 3 :
IF2030
```


Test Case 7

 test7 - Notepad

```
File Edit Format View Help
IF1010, IF1011, IF1012, IF1013.
IF1020, IF1021.
IF1030, IF1031.
IF2010, IF1010.
IF2020, IF1020, IF1030.
IF3030, IF2010, IF2020.
```

```
Masukkan nama file: test7.txt
Semester 1 :
IF1011 IF1012 IF1013 IF1021 IF1031
Semester 2 :
IF1010 IF1020 IF1030
Semester 3 :
IF2010 IF2020
Semester 4 :
IF3030
```

Test Case 8 :)

 test8 - Notepad

```
File Edit Format
W.
T, I.
AN, L.
D, S.
Y, O, N.
A, I.
```

```
Masukkan nama file: test8.txt
Semester 1 :
W I L S O N
Semester 2 :
T A N D Y A
```


D. *Link* kode program PRaKTiS

<https://github.com/WilsonTandya/PRaKTiS>

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.	✓	