

**TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA
PENYELESAIAN PENYUSUNAN RENCANA KULIAH DENGAN TOPOLOGICAL
SORT (PENERAPAN DECREASE AND CONQUER)**



Disusun oleh:

**Nama: Muhammad Furqon
NIM: 13519184**

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

Daftar Isi

1. Algoritma Topological Sort	3
2. SourceCode(Python).....	4
3. Tangkapan Layar Input Output	7
4. Tautan	11

1. Algoritma Topological Sort

Algoritma Topological Sort dengan pendekatan decrease and conquer

1. Bentuk graf dengan bentuk adjacency list antar simpul. Buat daftar atau list simpul yang merupakan mata kuliah yang belum diambil.
2. Proses simpul dalam daftar simpul yang belum diproses, cek apakah simpul memiliki jumlah *prerequisite* (derajat masuk) sama dengan nol, yaitu simpul tidak memiliki simpul lain yang menunjuk simpul tersebut. Jika memenuhi syarat, pilih simpul-simpul tersebut.
3. Simpul yang telah dipilih akan dihapus sisi-sisinya. Kurangi derajat masuk dari simpul-simpul yang ditunjuk simpul terpilih tersebut sebanyak satu.
4. Simpul yang terpilih dihilangkan dari daftar atau list simpul mata kuliah yang belum diambil. List simpul yang belum diproses adalah list simpul setelah proses ini.
5. Ulangi langkah kedua sampai langkah keempat hingga list simpul mata kuliah yang belum diambil sudah kosong atau telah mencapai delapan semester.

Pendekatan algoritma Topological sort ini adalah decrease and conquer dengan decrease sebanyak constant. Graf yang telah dibuat pada setiap iterasinya akan dikurangi menjadi sebuah upagraf yang lebih kecil. Jika n adalah banyak elemen pada graf dan s adalah simpul yang terpilih maka graf akan berkurang menjadi upagraf sebanyak $n-s$. Setelah pengurangan iterasi berikutnya akan diproses lagi sampai upagraf kosong atau semester telah mencapai semester 8.

Contoh:

Terdapat mata kuliah dengan nama mata kuliah dan *prerequisite*.

IF1,IF2.

IF2,IF3,IF4.

IF3.

IF4,IF3.

IF5,IF2,IF3,IF4.

Loop dalam pengerjaan disimbolkan dengan semester

IF1 : 1

IF2 : 2

IF3 : 0

IF4 : 1

IF5 : 3

Semester1: Pilih simpul-simpul dengan *prerequisite* sebanyak nol. IF3 dipilih, IF2,IF4, dan IF5 berkurang *prerequisite* sebanyak 1.

Semester2: Pilih simpul-simpul dengan *prerequisite* sebanyak nol. IF4 dipilih, IF2 dan IF5 berkurang *prerequisite* sebanyak 1.

Semester3: Pilih simpul-simpul dengan *prerequisite* sebanyak nol. IF2 dipilih, IF1 dan IF5 berkurang *prerequisite* sebanyak 1.

Semester4: Pilih simpul-simpul dengan *prerequisite* sebanyak nol. IF1,IF5 dipilih.

Program berakhir karena semua mata kuliah telah diambil.

2. SourceCode(Python)

Program MatkulDecider terbagi ke dalam beberapa bagian. Dalam penerapannya digunakan kelas Simpul dan representasi graf dalam bentuk adjacency list dalam dictionary. Fungsi terdiri dari inisialisasiFile untuk file, buatGraph untuk membuat graf, topologicalSearch untuk melakukan topological sort pada simpul, dan kurangiPrereq untuk menghapus simpul dan sisinya.

MatkulDecider.py

```
#Muhammad Furqon (13519184)
#Decrease and conquer (Topological Sort)
#13519184.py
#MatkulDecider
import os

#class untuk simpul terdiri dari kode dan prereq
class Simpul:
    #prereq:derajat masuk ; kode:kode matkul
    def __init__(self,kode,prereq):
        self.kode=kode
        self.prereq=int(prereq)

#Dari list kurangi graph yang sudah diproses
def kurangiPrereq(List, Simpul, Graph):
    sisi=Graph[Simpul.kode]
    #hilangkan sisi ke simpul lain dan kurangi prereq
    for x in sisi:
        for elemen in List:
            if(elemen.kode==x):
                elemen.prereq-=1
    #hilangkan dari dictionary
    Graph.pop(Simpul.kode)

#Print isi list objek, debugging purpose
def printListObj(List):
    first=True
    for obj in List:
        print("Kode matkul: "+ obj.kode)
        print("Jumlah prereq: "+ str(obj.prereq))
        first=False
    print("-"*10)

#Menggunakan topological sort
```

```

def topologicalSort(ListObj,ListResult,Graph):
    for matkul in ListObj:
        #mengecek adakah matkul yang prereq 0
        if matkul.prereq==0:
            #list hasil dari satu loop
            indeks=ListObj.index(matkul)
            ListResult.append(ListObj[indeks])

    for matkul_hasil in ListResult:
        #panggil fungsi mengurangi prereq yang membutuhkan matkul tersebut
        kurangiPrereq(ListObj,matkul_hasil,Graph)
        #buang dari list objek
        indeks=ListObj.index(matkul_hasil)
        ListObj.pop(indeks)

def inisialisasiFile(list1,dictionary):
    list_main=[]
    # List setiap barisnya
    os.chdir("..")
    cur_dir=os.getcwd()
    os.chdir(cur_dir+"\\test")
    nama_file = input("Masukkan nama file masukan: ")
    cur_dir=os.getcwd()
    path=cur_dir+ "\\ " + nama_file
    file = open(path,"r")
    #membaca dari file lalu mengubah menjadi bentuk array
    for line in file:
        line=line.strip()
        line=line.replace(" ","")
        line=line.strip(".")
        line.split(",")
        list_main.append(line)

    for x in list_main:
        x= list(x.split(","))
        #bagian head(depan) nama, tail berisi prereq
        head=x[0]
        tail=x[1:]
        temp_obj = Simpul(head,len(tail))
        list1.append(temp_obj)
        #dimasukkan ke dalam dictionary temporary yang akan digunakan untuk graph
        dictionary.update({head:tail})

def buatGraph(dictionary):
    #format key=simpul lalu value=sisi

```

```

#contoh {'IF1': [], 'IF2': ['IF1', 'IF5'], 'IF4': ['IF2', 'IF5'], 'IF5': []}
temp_dict={}
for key in dictionary:
    temp_arr=[]
    for key2 in dictionary:
        temp=dictionary[key2]
        if(key in temp):
            temp_arr.append(key2)
    temp_dict[key]=temp_arr
return temp_dict

#MAIN PROGRAM
#list objek Simpul
list_objek=[]
#list menyimpan hasil pada semester tersebut
list_hasil=[]
#list hasil total, debugging purpose
list_hasil_all=[]
#list nama semester
list_semester=["I","II","III","IV","V","VI","VII","VIII"]
#temporary dictionary untuk membuat graph
dict_awal={}
#graph yang digunakan dalam memilih matkul
graph={}

#inisialisasi dari file
inisialisasiFile(list_objek,dict_awal)

#buat graph
graph = buatGraph(dict_awal)

semester=0
while(len(list_objek)!=0 and semester<=7):
    #Melakukan topological sort
    topologicalSort(list_objek,list_hasil,graph)
    print("Semester "+ list_semester[semester]+" :",end=" ")
    first=True
    for matkul_diambil in list_hasil:
        if(first):
            print(matkul_diambil.kode,end="")
            first=False
        else:
            print(", "+matkul_diambil.kode,end="")
    print()
    list_hasil_all.append(list_hasil)

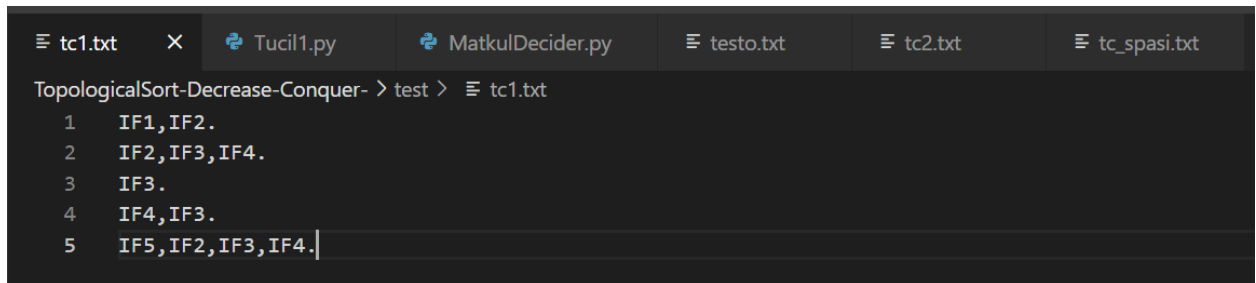
```

```
list_hasil=[]
semester+=1

input("Tekan enter untuk mengakhiri program")
```

3. Tangkapan Layar Input Output

1. tc1.txt

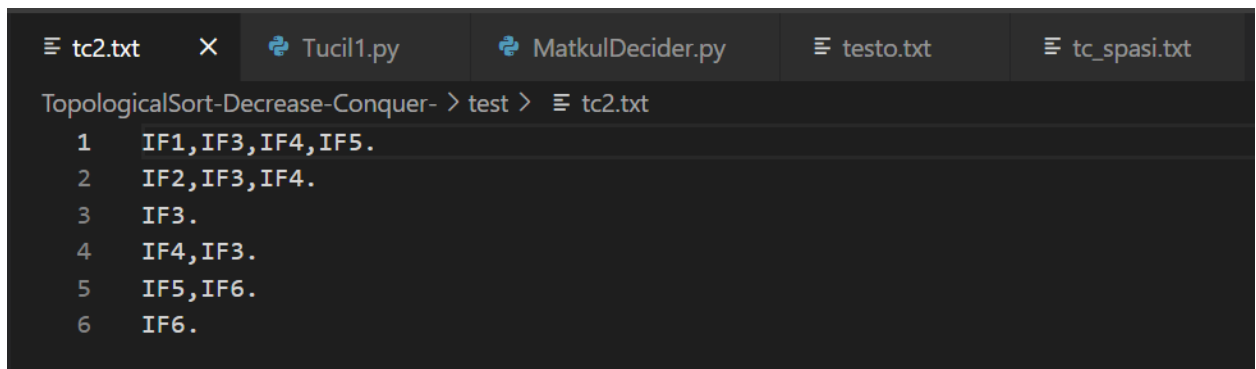


```
TopologicalSort-Decrease-Conquer- > test > tc1.txt
1 IF1,IF2.
2 IF2,IF3,IF4.
3 IF3.
4 IF4,IF3.
5 IF5,IF2,IF3,IF4.
```

Gambar

```
Masukkan nama file masukan: tc1.txt
Semester I : IF3
Semester II : IF4
Semester III : IF2
Semester IV : IF1, IF5
```

2. tc2.txt



```
TopologicalSort-Decrease-Conquer- > test > tc2.txt
1 IF1,IF3,IF4,IF5.
2 IF2,IF3,IF4.
3 IF3.
4 IF4,IF3.
5 IF5,IF6.
6 IF6.
```

```
Masukkan nama file masukan: tc2.txt
Semester I : IF3, IF6
Semester II : IF4, IF5
Semester III : IF1, IF2
```

3. tc3.txt

```
≡ tc3.txt  X  ≡ tc2.txt  MatkulDecider.py  ≡ tc1.txt
TopologicalSort-Decrease-Conquer- > test > ≡ tc3.txt
1  IF1,IF2,IF3,IF4.
2  IF2.
3  IF3.
4  IF4.
5  IF5.
Masukkan nama file masukan: tc3.txt
Semester I : IF2, IF3, IF4, IF5
Semester II : IF1
```

4. tc4.txt

```
≡ tc4.txt  X  MatkulDecider.py  ≡ testo.txt  ≡ tc_spasi.txt
TopologicalSort-Decrease-Conquer- > test > ≡ tc4.txt
1  IF1,IF5,IF6.
2  IF2,IF5.
3  IF3,IF6.
4  IF4,IF2,IF3.
5  IF5.
6  IF6.
Masukkan nama file masukan: tc4.txt
Semester I : IF5, IF6
Semester II : IF1, IF2, IF3
Semester III : IF4
```

5. tc5.txt

```
≡ tc5.txt  X  MatkulDecider.py  ≡ tc4.txt
TopologicalSort-Decrease-Conquer- > test > ≡ tc5.txt
1  IF1.
2  IF2,IF1.
3  IF3,IF1.
4  IF4,IF2.
5  IF5,IF2,IF6.
6  IF6,IF3.
7  IF7,IF4,IF5,IF6.
Masukkan nama file masukan: tc5.txt
Semester I : IF1
Semester II : IF2, IF3
Semester III : IF4, IF6
Semester IV : IF5
Semester V : IF7
```

6. tc6.txt


```
tc6.txt  X  MatkulDecider.py  tc4.txt
TopologicalSort-Decrease-Conquer- > test > tc6.txt
1  IF1.
2  IF2.
3  IF3,IF1,IF2.
4  IF4,IF3,IF5.
5  IF5,IF2.

Masukkan nama file masukan: tc6.txt
Semester I : IF1, IF2
Semester II : IF3, IF5
Semester III : IF4
```

7. tc7.txt

```
tc7.txt  X  MatkulDecider.py
TopologicalSort-Decrease-Conquer- > test > tc7.txt
1  IF1.
2  IF2,IF1.
3  IF3,IF2.
4  IF4,IF2,IF7.
5  IF5,IF2,IF3,IF4.
6  IF6,IF5.
7  IF7.

Masukkan nama file masukan: tc7.txt
Semester I : IF1, IF7
Semester II : IF2
Semester III : IF3, IF4
Semester IV : IF5
Semester V : IF6
```

8. tc8.txt

```
≡ tc8.txt X MatkulDecider.py
TopologicalSort-Decrease-Conquer- > test > ≡ tc8.txt
1 IF2,IF1.
2 IF1.
3 IF3,IF1,IF2.
4 IF5,IF3,IF4.
5 IF6,IF4,IF5.
6 IF4,IF2,IF3.

Masukkan nama file masukan: tc8.txt
Semester I : IF1
Semester II : IF2
Semester III : IF3
Semester IV : IF4
Semester V : IF5
Semester VI : IF6
```

9. testo.txt

```
≡ testo.txt X ≡ tc5.txt MatkulDecider.py ≡ tc4.txt
TopologicalSort-Decrease-Conquer- > test > ≡ testo.txt
1 C1,C3.
2 C2,C1,C4.
3 C3.
4 C4,C1,C3.
5 C5,C2,C4.

Masukkan nama file masukan: testo.txt
Semester I : C3
Semester II : C1
Semester III : C4
Semester IV : C2
Semester V : C5
```

10. tc_spasi.txt

```
≡ tc_spasi.txt X ≡ testo.txt ≡ tc5.txt
TopologicalSort-Decrease-Conquer- > test > ≡ tc_spasi.txt
1 C1, C3.
2 C2, C1, C4.
3 C3.
4 C4, C1, C3.
5 C5, C2, C4.
```

```
Masukkan nama file masukan: tc_spasi.txt
Semester I : C3
Semester II : C1
Semester III : C4
Semester IV : C2
Semester V : C5
```

4. Tautan

Alamat kode, github repo

https://github.com/Purecon/Tucil2_13519184

Link Pengumpulan

<http://bit.ly/KumpulTucil2Stima>

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.	✓	