

**TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA**

**SEMESTER II TAHUN 2020/2021**

**Penyusunan Rencana Kuliah dengan *Topological Sort***

**(Penerapan *Decrease and Conquer*)**

Oleh

Arsa Daris Gintara – 13519037



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

**2020/2021**

# Algoritma *Topological Sort* dan kaitan pendekatan *Decrease and Conquer*

## Algoritma *Topological Sort*

1. Program menerima masukan file .txt yang berisi daftar mata kuliah dan prerequisitenya jika ada berupa graf berarah yang terletak di direktori /test/ untuk kemudian dibaca.
2. Kemudian dalam daftar mata kuliah tersebut, dibentuk list mata kuliah unik yang ada dan dibentuk graf berarah dalam representasi list dua dimensi. List yang terbentuk memiliki anggota berupa list juga yang memiliki format elemen pertama sebagai simpul dan elemen berikutnya merupakan derajat masuk menuju simpul tersebut.
3. Kemudian akan di iterasi dalam graf berarah ini untuk dicari simpul yang memiliki derajat nol akan dimasukkan ke dalam solusi.
4. Simpul yang berderajat nol tadi juga dihapus di dalam graf dan dihapus juga sebagai derajat masuk ketika simpul tersebut menjadi derajat masuk simpul lain artinya prerequisite terpenuhi.
5. Setelah semua iterasi selesai dan graf berarah sudah kosong akan dihasilkan keluaran berupa daftar mata kuliah yang harus diambil pada setiap semester dengan catatan graf berupa DAG dan memenuhi untuk 8 semester saja.

## Kaitan dengan *Decrease and Conquer*

Algoritma *Topological Sort* memiliki kaitan dengan pendekatan *Decrease and Conquer*. Dalam praktiknya algoritma ini membagi persoalan menjadi bagian yang lebih kecil namun kemudian hanya memproses satu sub-persoalan saja. Dalam hal ini memproses mata kuliah yang tidak memiliki/sudah terpenuhi *prerequisite*-nya. Hal ini selaras dengan konsep *Decrease and Conquer* yang membagi persoalan menjadi bagian lebih kecil dan memproses satu sub-persoalan saja.

## Source Code Program

```
# load text dalam file
def toGraph(filename):
    theString = []
    with open(filename, 'r') as file:
        for line in file:
            theString.append(line.replace(',', ' ').replace(' ', ' ').replace('.', '').replace('\n', ''))

    return theString

# membuat list daftar mata kuliah yang ada (unik) menggunakan set
def nodeList(theList):
    theString = ''
    for i in theList:
        theString += i + ' '

    toSet = theString.split()
    result = list(set(toSet))
```

```

result.sort()

return result

# merepresentasikan graph berarah dalam bentuk list 2 dimensi. Dalam setiap list
# Index pertama sebagai Matakuliah, indeks berikutnya sebagai prerequisite dari mata kuliah tersebut
def graphInList(theList, listOfNode):
    result = []
    for every in theList:
        tmpEvery = every.split()
        tmp = []
        for inside in tmpEvery:
            for i in range(len(listOfNode)):
                if inside == listOfNode[i]:
                    tmp.append(i)

        result.append(tmp)

    return result

# pendekatan topological sort
def topSort(graphRepre):
    theOrder = []
    deg0 = []

    # mencari mata kuliah yang memiliki derajat masuk 0 atau tidak memiliki prerequisite
    # termasuk jika prerequisite sudah terpenuhi
    for i in range(len(graphRepre)):
        if len(graphRepre[i]) == 1:
            deg0 += graphRepre[i]

    # masukkan ke hasil dan menghapus mata kuliah yang berderajat masuk 0 dalam graf
    if len(deg0) > 0:
        for j in range(len(deg0)):
            graphRepre.remove([deg0[j]])

        for every in graphRepre:
            if (len(every) > 1 and deg0[j] in every):
                every.remove(deg0[j])

        theOrder.append(deg0)
    else:
        return None

```

```

# rekursif sekaligus checker apakah DAG
if (len(graphRepre) > 0):
    if (len(deg0) != 0):
        nextOrder = topSort(graphRepre)
        if (nextOrder != None):
            theOrder += nextOrder
        else:
            return None

    if (len(theOrder) > 8):
        return None

    return theOrder

# mencetak hasil sort
def printResult(resultList, listOfNode):
    print(" ,-,--. .=-.-. ,-.--. ")
    print(" ,-.'- _\ /==/_ /...-. /==, .' ")
    print("/==/_ ,_.'|==|, | \==\ -\ /= - / ")
    print("\==\ \ |==| | \==\ `-' ,/ ")
    print(" \==\ -\ |==|- | |==|, - | ")
    print(" _\==\ ,\ |==| ,| /==/ , \ ")
    print("/==/\ / _ ||==|- | /==/, .-- , - \ ")
    print("\==\ - , //==/. / \==\ - \ /= , / ")
    print(" `--`--`--' `--`--' `--`--' `--` ")

    print()

    for i in range(len(resultList)):
        print('Semester {}: '.format(i+1), end='')
        for j in range(len(resultList[i])):
            print('{}'.format(listOfNode[resultList[i][j]]), end='')
            if (j != len(resultList[i])-1):
                print(', ', end='')
        print()

#main program
if __name__ == "__main__":
    txtfile = input('Masukkan nama file yang terletak di folder /test/ (contoh: 1.txt) : ')
    source = 'test/'+txtfile

    try:
        open(source)
    except:
        source = '../test/'+txtfile
    try:

```

```
    open(source)
except:
    source = 'Tucil2_13519037/test/'+txtfile

listFromFile = toGraph(source)
listOfNode = nodeList(listFromFile)

graphRepre = graphInList(listFromFile, listOfNode)
result = topSort(graphRepre)

# masukan valid jika graf berupa DAG dan hasilnya tidak melebihi 8 semester(
menurut QNA)
if (result == None):
    print('Maaf, masukan tidak valid')
else:
    printResult(result, listOfNode)
```

## Screenshot Program

Tucil2\_13519037 > test > 1.txt

You, 5 hours ago | 1 author (You)

1 C1, C3.

2 C2, C1, C4.

3 C3.

4 C4, C1, C3.

5 C5, C2, C4.

You, 5 hours ago • add topological sort algorithm

TERMINAL

PROBLEMS

OUTPUT

DEBUG CONSOLE

+ CategoryInfo : ObjectNotFound: (1.txt:String) [], CommandNotFoundException

+ FullyQualifiedErrorId : CommandNotFoundException

PS D:\arsa\kuampus\semester\_4\Strategi Algoritma\tucil 2> & C:/Users/hp/AppData/Local/Programs/Python/Python39-6/Python.exe i Algoritma/tucil 2/Tucil2\_13519037/src/13519037.py"

Masukkan nama file yang terletak di folder /test/ (contoh: 1.txt) : 1.txt

```
graph TD
    A[1] --> B[2]
    A --> C[3]
    B --> D[4]
    B --> E[5]
    C --> E
    D --> F[6]
    E --> F
    F --> G[7]
    F --> H[8]
    G --> I[9]
    H --> I
    I --> J[10]
    I --> K[11]
    J --> L[12]
    K --> L
```

Semester 1: C3  
Semester 2: C1  
Semester 3: C4  
Semester 4: C2  
Semester 5: C5

13519037.py 2.txt 8.txt

Tucil2\_13519037 > test > 2.txt

You, 3 minutes ago | 1 author (You)

1 Mat.

2 Fis.

3 TTKI.

4 Pengkom.

5 Inggris.

6 Or.

7 Matdis, Mat.

8 Algeo, Mat, Fis.

9 TBFO, Mat.

10 Orkom, Or.

11 Alstrukdat, Pengkom, Mat, Fis.

12 Logkom, Pengkom, Mat.

TERMINAL

PROBLEMS

OUTPUT

DEBUG CONSOLE

Semester 2: C1  
Semester 3: C4  
Semester 4: C2  
Semester 5: C5

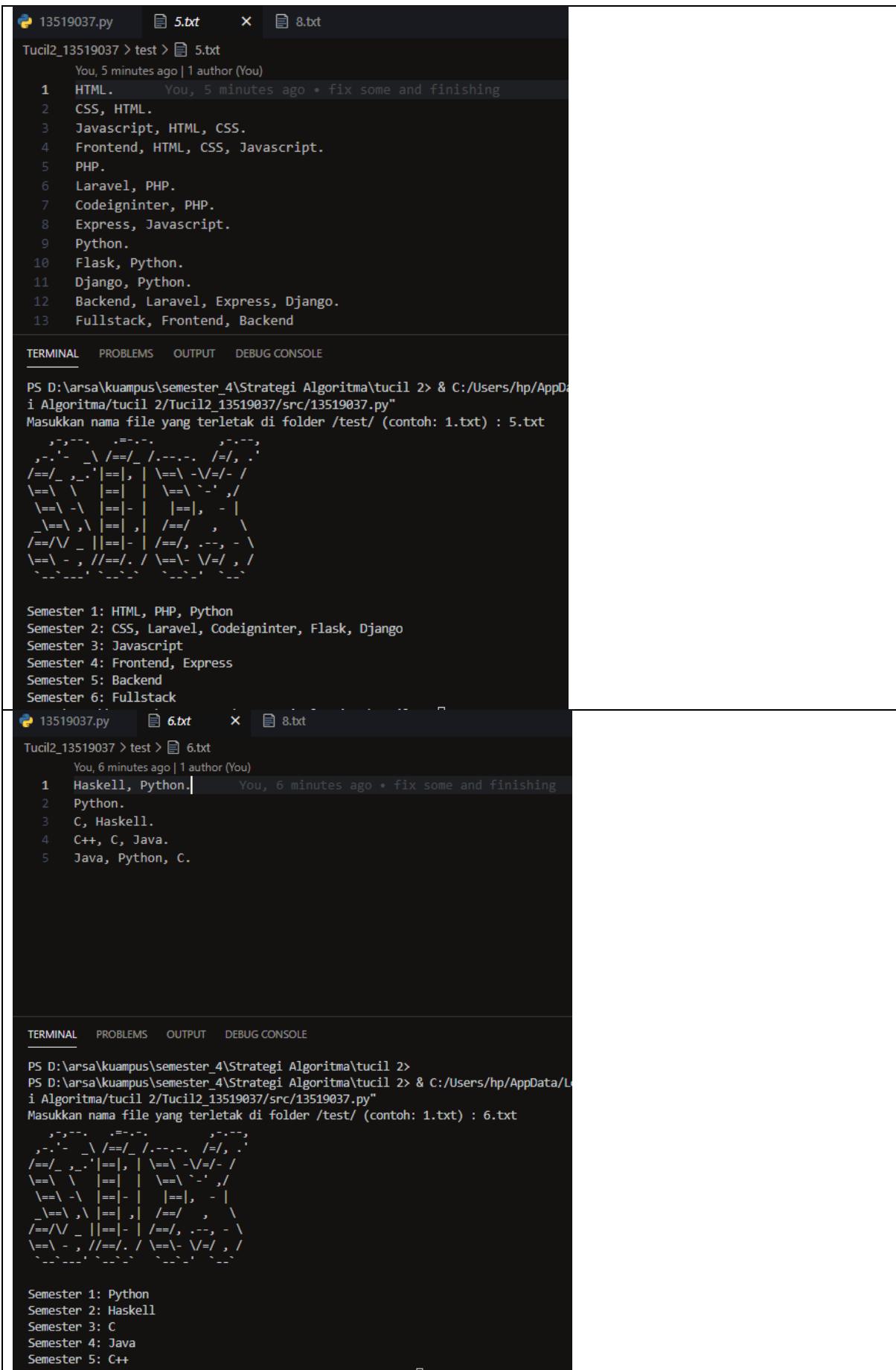
PS D:\arsa\kuampus\semester\_4\Strategi Algoritma\tucil 2&gt; &amp; C:/Users/hp/AppData/Local/Programs/Python/Python39-6/Python.exe i Algoritma/tucil 2/Tucil2\_13519037/src/13519037.py"

Masukkan nama file yang terletak di folder /test/ (contoh: 1.txt) : 2.txt

```
graph TD
    A[1] --> B[2]
    A --> C[3]
    B --> D[4]
    B --> E[5]
    C --> E
    D --> F[6]
    E --> F
    F --> G[7]
    F --> H[8]
    G --> I[9]
    H --> I
    I --> J[10]
    I --> K[11]
    J --> L[12]
    K --> L
```

Semester 1: Mat, Fis, TTKI, Pengkom, Inggris, Or  
Semester 2: Matdis, Algeo, TBFO, Orkom, Alstrukdat, Logkom









## Alamat Program

<https://github.com/arsa-dg/coursehouse>

## Checklist

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	√	