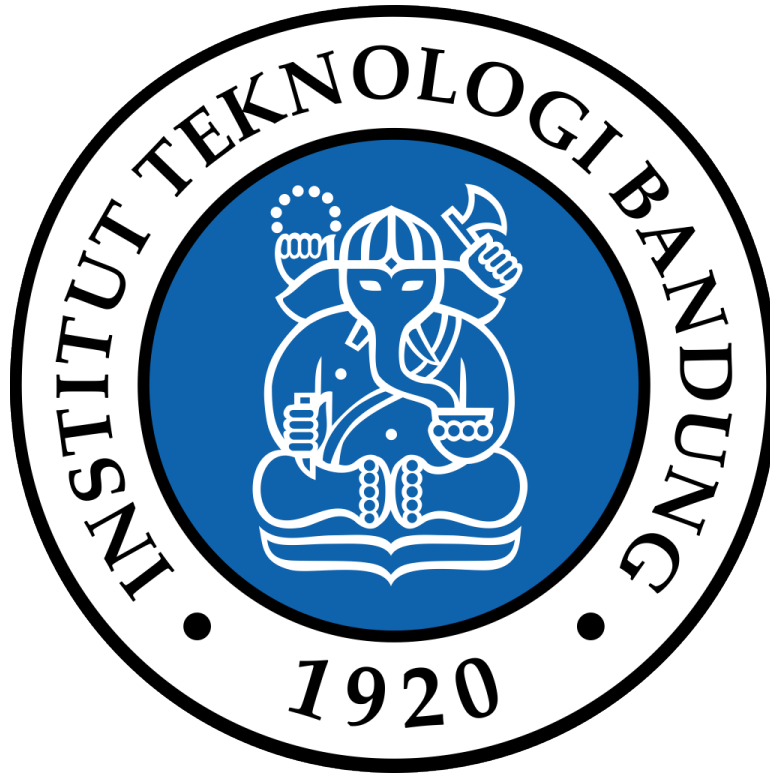


**Tugas Kecil 2 IF2121**

**Strategi Algoritma**

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

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



**Nama : Rayhan Asadel**

**NIM : 13519196**

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

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

*Topological Sort* adalah suatu algoritma *sorting* yang dapat mengurutkan suatu elemen/node yang sebelumnya memiliki dependencies dan harus dikerjakan terlebih dahulu. Algoritma ini dalam kehidupan sehari-hari dapat dimisalkan seperti pengambilan prerequisite pada mata pelajaran/ mata kuliah, penjadwalan event scheduling, dan lain sebagainya.

Berikut adalah langkah-langkah dari *topological sort* yang dilakukan dalam tugas ini.

1. Menerima input file .txt dan mengubahnya menjadi representasi graph dengan matriks, dimana elemen baris adalah elemen dari matkul atau node, dan elemen kolom adalah elemen dari prerequisite.
2. Dengan matriks tersebut, akan dihitung dan dicari *in degree* tiap matkul, dan dipilih matkul yang memiliki *in degree* = 0.
3. Matkul atau *node* yang terpilih tersebut berikutnya akan dihapus beserta busur yang keluar dari *node* tersebut, dan node tersebut akan disimpan dalam larik yang berisi urutan dari hasil *sorting*.
4. Ulangi langkah 2 sampai 3, hingga semua *node* atau elemen sudah dikunjungi

Dalam implementasinya, karena setiap iterasi ukuran matriks yang dicek selalu berkurang, pendekatan *topological sort* ini termasuk varian *Decrease and Conquer* dengan *decrease by a variable size*, karena setiap iterasi matriks yang diproses akan berkurang, dan berkurangnya tergantung dengan jumlah *node* yang memiliki *in degree* = 0 pada iterasi tersebut.

## Source Code Program

```
#Kamus Fungsi dan Prosedur

def listToString(s):
    #Fungsi yang mengubah suatu list s menjadi string str1

    str1 = ""

    for ele in s:
        str1 += ele

    return str1

def inputfiletoArray():
    #prosedur yang menerima dan mengolah input file text menjadi matriks adjacency (representasi DAG dengan adjacency list)
    print()
    filename = str(input("Masukan Nama File (Nama.txt): ")) #Masukan nama file dengan format "Nama.txt"
    print()
    with open('../test/'+filename, 'r') as file:

        #Konversi isi file txt menjadi string yang bersih dari karakter2 pemisah dan "\n"
        ListKiri = [] #Listkiri untuk menampung sementara hasil dari file text

        for line in file:
            if line.startswith('-'):
                continue
            else:
                ListKiri.append(line.replace(',', '').replace(' ', '').replace('C', '').replace('\n', ' '))

        Str = listToString(ListKiri)
        Str = Str.split(' ')
        panjangArr = len(Str)
        Str = listToString(Str)
        panjangString = len(Str)

        #arrayInt merupakan adjacency list, dengan baris sebagai simpul (matkul), dan kolom sebagai prerequisite
        arrayInt = [[0 for x in range(panjangArr)] for y in range(panjangArr)]
```

```

j = 0
baris = 0
kolom = 0
for i in range(1, panjangString):
    Angka = Str[j:i]
    j = j+1

    if (Angka == '.'):
        baris = baris+1
        kolom = 0
    else:
        Angka = int(Angka)
        for n in range (panjangArr+1):
            if (n == Angka):
                if (Angka-1==baris):
                    arrayInt[baris][n-1] = 0
                    kolom = kolom+1
                else:
                    arrayInt[baris][n-1] = 1
                    kolom = kolom+1

        return arrayInt          ## mereturn matriks dengan elemen[n][0] adalah
matkul n, dan elemen[n][1..n] adalah prereq dari matkul n

def findLowestInDegree(matriks):
    #Mencari elemen dengan in degree 0, pada matriks, dan mengeluarkan posisi
    barisnya
    min = 0
    sum = 0
    indeksMatkul = -1;

    for i in range(1):
        sum = 0
        for j in range(len(matriks)):
            if (matriks[i][j]!=-1):
                sum = sum + matriks[i][j]

    counter = 0
    for i in range(len(matriks)):
        sum = 0
        counter = 0
        for j in range(len(matriks)):
            if (matriks[i][j]!=-1):
                sum = sum + matriks[i][j]
            else:
                counter = counter + matriks[i][j]

    if (sum == 0) and (counter!=(len(matriks)*-1)) :

```

```

        indeksMatkul = i

    return indeksMatkul

def deleteVertexAndConnection(matriks, indeksMatkul):
    #Menghapus simpul dengan posisi baris indeksMatkul, dan semua busur yang k
    eluar dari simpul tersebut
    for i in range(len(matriks)):
        for j in range(len(matriks)):
            if (i==indeksMatkul):
                matriks[indeksMatkul][j] = -1
            elif (j==indeksMatkul) and (matriks[i][indeksMatkul]==1) :
                matriks[i][indeksMatkul] = -1

    return matriks

def findCountZeroDegree(matriks):
    #Mencari ada berapa simpul yang memiliki in degree = 0
    nZeroDegrees = 0
    counter = 0
    for i in range(len(matriks)):
        sum = 0
        counter = 0
        for j in range(len(matriks)):
            if (matriks[i][j]!=-1):
                sum = sum + matriks[i][j]
            else:
                counter = counter + matriks[i][j]

        if (sum == 0) and (counter!=(len(matriks)*-1)) :
            nZeroDegrees = nZeroDegrees + 1

    return nZeroDegrees

```

```

## ALGORITMA UTAMA

hasilSortArray = []      #Sebagai array penerima hasil sorting
matriks = inputfiletoArray()
sizeMatriks = len(matriks)

print("Ini Adalah Representasi Matriks dari File Text Anda: ")
print("Dengan Dimensi:",end=' ')
print(sizeMatriks)

for i in range(len(matriks)):
    for j in range(len(matriks)):
        print(matriks[i][j],end='')      #menampilkan matriks
    print()
print()

i = 0
j = 0
ParamLoop = 0
while (i<sizeMatriks):      #topological sort, dengan makin lama baris matriks
    yang dicek semakin sedikit (decrease by a variable size)
    ParamLoop = findCountZeroDegree(matriks)
    semester = []

    for j in range(ParamLoop):      #loop untuk menentukan >1 matkul pada 1 se
mester
        target = findLowestInDegree(matriks)
        matriks = deleteVertexAndConnection(matriks,target)
        semester = semester + ['C'+str(target+1)]
    i = i + ParamLoop
    hasilSortArray.append([semester])

#endwhile

panjangHasilSort = len(hasilSortArray)

if (panjangHasilSort>8):
    print("Solusi tidak dapat ditampilkan, perlu lebih dari 8 semester")
else:
    print("Berikut Adalah Rencana Kuliah Anda")
    for i in range(0,panjangHasilSort):
        print("Semester ",i+1," : ",hasilSortArray[i])

```

## Contoh Input dan Output

Berikut adalah beberapa hasil pengetesan beserta masukannya.

- |   |             |
|---|-------------|
| 1 | C1, C3.     |
| 2 | C2, C1, C4. |
| 3 | C3.         |
| 4 | C4, C1, C3. |
| 5 | C5, C2, C4. |

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C3']]
Semester 2 : [['C1']]
Semester 3 : [['C4']]
Semester 4 : [['C2']]
Semester 5 : [['C5']]
```
- |   |             |
|---|-------------|
| 1 | C1.         |
| 2 | C2, C5.     |
| 3 | C3.         |
| 4 | C4, C1, C3. |
| 5 | C5, C4.     |

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C3', 'C1']]
Semester 2 : [['C4']]
Semester 3 : [['C5']]
Semester 4 : [['C2']]
```
- |   |                 |
|---|-----------------|
| 1 | C1, C3.         |
| 2 | C2, C3, C5, C7. |
| 3 | C3.             |
| 4 | C4, C1, C3.     |
| 5 | C5.             |
| 6 | C6, C5, C3.     |
| 7 | C7, C4.         |

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C5', 'C3']]
Semester 2 : [['C6', 'C1']]
Semester 3 : [['C4']]
Semester 4 : [['C7']]
Semester 5 : [['C2']]
```
- |   |             |
|---|-------------|
| 1 | C1.         |
| 2 | C2, C1.     |
| 3 | C3, C1.     |
| 4 | C4, C2, C3. |
| 5 | C5, C2, C4. |

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C1']]
Semester 2 : [['C3', 'C2']]
Semester 3 : [['C4']]
Semester 4 : [['C5']]
```
- |   |             |
|---|-------------|
| 1 | C1.         |
| 2 | C2, C1.     |
| 3 | C3, C1.     |
| 4 | C4, C2, C3. |

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C1']]
Semester 2 : [['C3', 'C2']]
Semester 3 : [['C4']]
```

6.

```
1 C1.
2 C2, C1.
3 C3, C1.
4 C4, C2, C3.
5 C5, C4.
6 C6, C4.
```

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C1']]
Semester 2 : [['C3', 'C2']]
Semester 3 : [['C4']]
Semester 4 : [['C6', 'C5']]
```

7.

```
1 C1.
2 C2, C1.
3 C3, C1.
4 C4, C2, C3.
5 C5, C4.
6 C6, C4.
7 C7, C5, C6.
```

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C1']]
Semester 2 : [['C3', 'C2']]
Semester 3 : [['C4']]
Semester 4 : [['C6', 'C5']]
Semester 5 : [['C7']]
```

8.

```
1 C1.
2 C2, C1.
3 C3, C2, C4.
4 C4, C2.
5 C5, C4.
```

```
Berikut Adalah Rencana Kuliah Anda
Semester 1 : [['C1']]
Semester 2 : [['C2']]
Semester 3 : [['C4']]
Semester 4 : [['C5', 'C3']]
```



## Tabel Checklist dan Link File Tugas

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

Link Google Drive :

<https://drive.google.com/drive/folders/1GIjudaHV4tydC3MNC7TAR5M9qSQJIglW?usp=sharing>