

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

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

Disusun oleh :

Nama : Muhammad Jafar Gundari

NIM :13519197



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

A. Algoritma Topological Sort dan Kaitannya dengan Algoritma Decrease and Conquer

Topological sort adalah algoritma yang digunakan untuk mengurutkan directed acyclic graph (DAG). Adapun pada kasus **Penyusunan Rencana Kuliah dengan Topological Sort** pendekatannya adalah sebagai berikut.

Input program adalah berupa file yang berformat seperti pada gambar 1.

```
<kode_kuliah_1>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>.  
<kode_kuliah_2>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>.  
<kode_kuliah_3>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>, <kode kuliah prasyarat - 4>.  
<kode_kuliah_4>.  
.  
.  
.
```

Gambar 1. Format File Teks untuk Masukan Daftar Kuliah

Untuk itu dibuat sebuah kelas diberi nama *CourseType* yang membuat setiap mata kuliah menjadi representasi graf. Kelas *CourseType* mempunyai atribut nama mata kuliah, *array* mata kuliah prasyarat, dan derajat masuk. Masukan dari file kemudian dibuat menjadi *array of CourseType*. Fungsi *toposort*, dilakukan dengan mencari mata kuliah yang memiliki derajat masuk (*indegree*) bernilai 0, atau yang tidak memiliki prasyarat sebagai senarai baru (kemudian dinamakan dengan *array 0degree*) untuk mengelompokkan mata kuliah yang berada pada semester yang sama. Pada setiap mata kuliah yang berada pada *array 0Degree* hapus busur (*edge*) yang berasal dari mata kuliah tersebut dengan kata lain menghapus *array prerequisites* pada mata kuliah yang terhubung dengannya (mengurangi derajat masuk) dengan cara menelusuri *array of CourseType*. Setelah itu hapus mata kuliah *0degree* tersebut. Dengan begini, *array* mata kuliah sudah berkurang elemennya sebanyak *array 0degree*. Sebelum dihapus, tampilkan ke layar mata kuliah beserta semesternya. Ulangi Langkah dengan memanggil rekursif fungsi *toposort* sampai tidak ada mata kuliah tersisa pada senarai mata kuliah yang didapat saat awal pembacaan file. Kaitan dengan algoritma *decrease and conquer* terletak pada saat mengurangi *array* mata kuliah sebanyak mata kuliah yang memiliki *0degree*. Pada setiap langkahnya, hanya berfokus pada satu *array* yang persoalannya telah mengerucut.

B. Source program

```
class CourseType:
    # Constructor for class CourseType
    def __init__(self, name):
        self.name = name
        self.prereq = []
        self.inDegree = 0

    # Remove course item that exists in prereq, update indegree
    def removePrereq(self, course):
        self.prereq.remove(course)
        self.inDegree -= 1

    # Insert item as prereq
    def addPrereq(self, ListCourse):
        for course in ListCourse:
            self.prereq.append(course)
            self.inDegree += 1

    # Print all prerequisites
    def printPrereq(self):
        print("Prerequisites for", self.name, "is : ", end=" ")
        print(self.prereq, end=" ")
        print("In Degree :", self.inDegree)
```

```

# Insert all item as type CourseType into ListCourse
def addtoListCourse(arrayCourse):
    global ListCourse
    C = CourseType(arrayCourse[0])
    arrayCourse.pop(0)
    C.addPrereq(arrayCourse)
    ListCourse.append(C)

# Create an array that contains CourseType with indegree = 0
def insert0Degree():
    global ListCourse
    List0Degree = []
    for course in ListCourse:
        if (course.inDegree == 0):
            List0Degree.append(course)
    return List0Degree

# Delete CourseType prereq that equals course.name (removing edge that came from selected course)
def deleteEdge(course, List0Degree):
    global ListCourse
    for matkul in ListCourse:
        if(matkul not in List0Degree):
            if course.name in matkul.prereq:
                matkul.removePrereq(course.name)

```

```

# Toposort Algorithm
def toposort():
    global ListCourse
    global semester
    # Base, until ListCourse is empty
    if(len(ListCourse) == 0):
        return
    else:
        # Semester will assigned = 1 for the first time, outside this function, so it is safe
        print("Semester", semester, ": ", end=" ")
        # Remove all CourseType with indegree = 0, remove edge from that vertice and print them
        List0Degree = insert0Degree()
        for vertice in List0Degree:
            print(vertice.name, end=" ")
            deleteEdge(vertice, List0Degree)
            # Remove vertice, subtract size of ListCourse
            ListCourse.remove(vertice)
        print()
        # Call toposort recursively
        semester += 1
        toposort()

```


C. Masukan dan keluaran program

```

===== Know Your Lecture =====
===== By Jafar =====

Insert filename : tc1.txt

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

Do you want to add another file ? (yes/no) ☐

```

Gambar 2.1 tc1.txt

```
Insert filename : tc2.txt

Your result :
Semester 1 : C2; C3;
Semester 2 : C1; C4;
Semester 3 : C5;
Semester 4 : C6; C7;
=====
Do you want to add another file ? (yes/no) ☐
```

Gambar 2.2 tc2.txt

```
Insert filename : grafTest1.txt

Your result :
Semester 1 : Kalkulus; Fisika;
Semester 2 : Matdis;
Semester 3 : Kriptografi; TBF0; Stima;
=====
```

Gambar 2.3 grafTest1.txt

```
Insert filename : tc3.txt

Your result :
Semester 1 : C3; C4; C9;
Semester 2 : C1; C6;
Semester 3 : C2;
Semester 4 : C5;
Semester 5 : C8;
Semester 6 : C7;
=====

Do you want to add another file ? (yes/no) _
```

Gambar 2.4 tc3.txt

```
Insert filename : tc4.txt

Your result :
Semester 1 : C6; C10;
Semester 2 : C7; C5;
Semester 3 : C9; C8; C4;
Semester 4 : C3;
Semester 5 : C1;
Semester 6 : C2;
=====

Do you want to add another file ? (yes/no) _
```

Gambar 2.5 tc4.txt

```
Insert filename : grafTest3.txt

Your result :
Semester 1 : MA1101; FI1101; KU1001; KU1102; KU1011; KU1024; IF1210; KU1202; IF2121; IF2110; IF2120; IF2124; IF2130; IF2211; IF2230; IF2240;
IF2250; IF3140; IF3280; IF4091; KU2071; KU206X; AS2005;
Semester 2 : MA1201; FI1201; EL1200; IF2123; IF2210; IF3130; IF3141; IF3150; IF3151; IF3210; IF4090; IF4092;
Semester 3 : IF2220; IF3110; IF3230; IF3250; IF3260;
Semester 4 : IF3170;
Semester 5 : IF3270;
=====
```

Gambar 2.6 grafTest3.txt

```
Insert filename : grafTest4.txt

Your result :
Semester 1 : C;
Semester 2 : Python;
Semester 3 : Pip;
Semester 4 : Flask;
=====

Do you want to add another file ? (yes/no) _
```

Gambar 2.7 grafTest4.txt

```
Insert filename : grafTest2.txt

Your result :
Semester 1 : KU1102; MA1101; FI1101; KU1011;
Semester 2 : MA1201; FI1201; IF1210; KU1202; KI1002; EL1200;
=====
```

Gambar 2.8 grafTest2.txt

D. Drive source code

https://drive.google.com/drive/folders/1k5gz3tzt_YbN7dgQyYCxLd513durPshw?usp=sharing

E. Penilaian

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