

Laporan Tugas Kecil 2
IF2211 Strategi Algoritma
Penyusunan Rencana Kuliah dengan Topological Sort
(Penerapan Decrease and Conquer)



Disusun Oleh:

Zaidan Naufal Sudrajat / 13518021

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

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

Topological Sort adalah suatu metode pengurutan simpul yang digunakan pada *directed graph* agar pada setiap simpul u yang memiliki sisi berarah ke simpul v , simpul u ada pada posisi sebelum simpul v pada urutan. Pada tugas kecil ini topological sort di aplikasikan menggunakan pendekatan algoritma *Decrease and Conquer*. Algoritma *Decrease and Conquer* memiliki pendekatan mengurangi masalah menjadi masalah yang lebih kecil lalu menyelesaikan masalah yang kecil tersebut. Berikut merupakan algoritma yang digunakan pada tugas kecil ini :

1. Persiapan Data

Data Graf pada persoalan ini diimplementasikan sebagai ADT *Dictionary* Bahasa Python , yang menggambarkan *dictionary key* sebagai Ujung Simpul, dan *dictionary values* sebagai *list* dari simpul yang memiliki sisi berarah dari dari ujung simpul. Pada tucil ini *dictionary key* akan disebut sebagai mata kuliah, *list* pada *dictionary values* akan disebut sebagai mata kuliah prasyarat, dan *dictionary* akan disebut sebagai graf.

2. Pengambilan simpul yang memiliki derajat 0.

```
def ambil_matkul(graph_mk_prasyarat):
    """ Meminta input graph, mengeluarkan graph baru tanpa matkul yang diambil, dan matkul yang diambil list"""
    yg_diambil = []
    new_graph_mk_prasyarat = graph_mk_prasyarat.copy()
    for mk,prasyarat in graph_mk_prasyarat.items():
        if len(prasyarat) == 0:
            new_graph_mk_prasyarat = hapus_mk_dan_prasyarat(new_graph_mk_prasyarat,mk)
            yg_diambil.append(mk)

    return new_graph_mk_prasyarat, yg_diambil
```

Data yang diinputkan akan diperiksa satu persatu apabila derajat (jumlah dari list pada mata kuliah prasyarat) dari mata kuliah sama dengan 0 maka mata kuliah tersebut akan dikeluarkan dari graf dan di append pada suatu list mata kuliah yang diambil.

```
def hapus_mk_dan_prasyarat(graph_mk_prasyarat,mk):
    """ Menghapus matkul dari graph"""
    graph_mk_prasyarat.pop(mk)
    for matkul,prasyarat in graph_mk_prasyarat.items():
        if mk in prasyarat:
            prasyarat = [x for x in prasyarat if x!=mk]
            graph_mk_prasyarat[matkul] = prasyarat
    return graph_mk_prasyarat
```

Graf yang sudah digunakan akan diganti dengan graf yang baru yang tidak berisi mata kuliah yang sudah diambil

3. Operasi diulang pada graf terbaru hingga graf kosong.

```
def topological_sort(graph_mk_prasyarat):
    """ Meminta input graph, mengeluarkan order matkul yang bisa diambil berupa list """
    order_diambil = []
    while True:
        graph_mk_prasyarat, mk_diambil = ambil_matkul(graph_mk_prasyarat)
        order_diambil.append(mk_diambil)
        if not (graph_mk_prasyarat):
            break
    return order_diambil
```

B. Source Code dalam Bahasa Python

```
import sys
import os
def topological_sort(graph_mk_prasyarat):
    """ Meminta input graph, mengeluarkan order matkul yang bisa diambil berupa list """
    order_diambil = []
    while True:
        graph_mk_prasyarat, mk_diambil = ambil_matkul(graph_mk_prasyarat)
        order_diambil.append(mk_diambil)
        if not (graph_mk_prasyarat):
            break
    return order_diambil

def ambil_matkul(graph_mk_prasyarat):
    """ Meminta input graph, mengeluarkan graph baru tanpa matkul yang diambil, dan matkul yang diambil list """
    yg_diambil = []
    new_graph_mk_prasyarat = graph_mk_prasyarat.copy()
    for mk, prasyarat in graph_mk_prasyarat.items():
        if len(prasyarat) == 0:
            new_graph_mk_prasyarat = hapus_mk_dan_prasyarat(new_graph_mk_prasyarat, mk)
            yg_diambil.append(mk)

    return new_graph_mk_prasyarat, yg_diambil

def hapus_mk_dan_prasyarat(graph_mk_prasyarat, mk):
    """ Menghapus matkul dari graph """
    graph_mk_prasyarat.pop(mk)
    for matkul, prasyarat in graph_mk_prasyarat.items():
        if mk in prasyarat:
            prasyarat = [x for x in prasyarat if x != mk]
            graph_mk_prasyarat[matkul] = prasyarat
    return graph_mk_prasyarat

def keluaran(order):
    """ fungsi output """
    romawi = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII']
    print("Urutan mata kuliah yang dapat diambil :")
    order = ["", " ".join(x for x in order)]
    for i in range(min(len(order), 8)):
        print("SEMESTER {:<3} : {}".format(romawi[i], str(order[i])))
    if len(order) > 8:
        print("Matkul lainnya tidak dapat diambil dalam 8 semester")

def clean(line):
```

```

""" fungsi membersihkan input """
line = line.split(",")
line = [x.strip("\n,") for x in line]
return line[0], line[1:]

if __name__ == "__main__":
    graph = {}
    path = "./test/"
    namafile = sys.argv[1]
    with open(os.path.join(path, namafile), "r") as f:
        for line in f:
            mk, prasyarat = clean(line)
            graph[mk] = prasyarat
    f.close()
    order = topological_sort(graph)
    keluaran(order)

```

C. Hasil *Screenshot* input dan output

1. Uji 1

Input :

```

CS001.
CS002.
CS003, CS002.
CS004
CS005, CS004.
CS006, CS003, CS005.
CS007, CS006, CS003, CS004.
CS008, CS005, CS004.
CS009, CS002, CS005.
CS010, CS001.

```

Output :

```

zaidanaufa1@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji1.txt
Urutan mata kuliah yang dapat diambil :
SEMESTER I   : CS001, CS002, CS004
SEMESTER II  : CS003, CS005, CS010
SEMESTER III : CS006, CS008, CS009
SEMESTER IV  : CS007

```

2. Uji 2

Input:

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

Output:

```
zaidanaufa1@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji2.txt  
Urutan mata kuliah yang dapat diambil :  
SEMESTER I      : C3  
SEMESTER II     : C1  
SEMESTER III    : C4  
SEMESTER IV     : C2  
SEMESTER V      : C5
```

3. Uji 3

Input:

```
CS001.  
CS002, CS001.  
CS003.  
CS004, CS002.  
CS005, CS001, CS002, CS004.  
CS006, CS002.  
CS007, CS005.  
CS008, CS003.  
CS009, CS006.  
CS010, CS002, CS006, CS007.
```

Output:

```
zaidanaufa1@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji3.txt  
Urutan mata kuliah yang dapat diambil :  
SEMESTER I      : CS001, CS003  
SEMESTER II     : CS002, CS008  
SEMESTER III    : CS004, CS006  
SEMESTER IV     : CS005, CS009  
SEMESTER V      : CS007  
SEMESTER VI     : CS010
```

4. Uji 4

Input:

```
CS001.  
CS002.  
CS003, CS001  
CS004, CS001.  
CS005, CS004.  
CS006, CS003, CS005.  
CS007.  
CS008.  
CS009, CS003, CS005, CS006, CS007.  
CS010, CS003.  
CS011, CS004, CS007.  
CS012, CS007.  
CS013, CS003.  
CS014, CS005, CS013.  
CS015, CS005, CS009, CS010.
```

Output:

```
zaidanaufa1@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji4.txt  
Urutan mata kuliah yang dapat diambil :  
SEMESTER I   : CS001, CS002, CS007, CS008  
SEMESTER II  : CS003, CS004, CS012  
SEMESTER III : CS005, CS010, CS011, CS013  
SEMESTER IV  : CS006, CS014  
SEMESTER V   : CS009  
SEMESTER VI  : CS015
```


5. Uji 5

Input:

```
CS001.  
CS002.  
CS003, CS002.  
CS004, CS002, CS003.  
CS005, CS004.  
CS006, CS002, CS003, CS005.  
CS007, CS003, CS005.  
CS008.  
CS009, CS002, CS004.  
CS010, CS002.
```

Output:

```
zaidanaufal@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji5.txt  
Urutan mata kuliah yang dapat diambil :  
SEMESTER I   : CS001, CS002, CS008  
SEMESTER II  : CS003, CS010  
SEMESTER III : CS004  
SEMESTER IV  : CS005, CS009  
SEMESTER V   : CS006, CS007
```

6. Uji 6

Input:

```
CS001.  
CS002.  
CS003.  
CS004, CS003.  
CS005.  
CS006.  
CS007, CS005.  
CS008, CS002, CS003, CS004.  
CS009, CS008, CS005.  
CS010, CS004, CS005.
```

Output:

```

zaidanaufai@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji6.txt
Urutan mata kuliah yang dapat diambil :
SEMESTER I   : CS001, CS002, CS003, CS005, CS006
SEMESTER II  : CS004, CS007
SEMESTER III : CS008, CS010
SEMESTER IV  : CS009

```

7. Uji 7

Input:

```

CS001.
CS002.
CS003.
CS004, CS001, CS003.
CS005.
CS006, CS001.
CS007.
CS008.
CS009, CS005, CS007.
CS010.
CS011, CS010, CS004, CS006.
CS012, CS002, CS003, CS006.
CS013, CS004, CS003.
CS014, CS005, CS003.
CS015, CS007, CS008, CS011.
CS016, CS015, CS013, CS014, CS001.
CS017, CS003, CS012, CS007, CS015.
CS018, CS011, CS013, CS005, CS014.
CS019, CS012, CS015, CS011, CS006.
CS020, CS003, CS017, CS006, CS001.

```

Output:

```

zaidanaufai@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji7.txt
Urutan mata kuliah yang dapat diambil :
SEMESTER I   : CS001, CS002, CS003, CS005, CS007, CS008, CS010
SEMESTER II  : CS004, CS006, CS009, CS014
SEMESTER III : CS011, CS012, CS013
SEMESTER IV  : CS015, CS018
SEMESTER V   : CS016, CS017, CS019
SEMESTER VI  : CS020

```


8. Uji 8

Input:

```
CS001.  
CS002.  
CS003.  
CS004, CS002, CS003.  
CS005, CS002.  
CS006, CS004, CS002.  
CS007, CS001.  
CS008, CS006, CS005.  
CS009, CS007, CS005, CS006.  
CS010, CS001, CS005, CS006.
```

Output:

```
zaidanaufa1@LAPTOP-7GE03F4H:/mnt/d/Coding/Stima/Tucil_Stima/tucil_2$ ./run.sh uji8.txt  
Urutan mata kuliah yang dapat diambil :  
SEMESTER I   : CS001, CS002, CS003  
SEMESTER II  : CS004, CS005, CS007  
SEMESTER III : CS006  
SEMESTER IV  : CS008, CS009, CS010
```

D. Link Source Code

Github : https://github.com/zaidannaufal/Tucil_Stima/tree/main/tucil_2

E. Kotak Penilaian Asisten

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