

LAPORAN TUGAS KECIL II IF2211 STRATEGI ALGORITMA
SEMESTER II TAHUN 2020/2021
PENYUSUNAN RENCANA KULIAH DENGAN *TOPOLOGICAL SORT*
(PENERAPAN *DECREASE AND CONQUER*)



OLEH
RAYHAN ALGHIFARI FAUZTA (13519039)

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

I. Algoritma Topological Sort

Tugas kecil kali ini meminta mahasiswa untuk memecahkan penyusunan rencana kuliah dengan memanfaatkan algoritma *topological sort*. Pada tugas ini, program dibuat menggunakan bahasa pemrograman Python 3 dengan memanfaatkan beberapa struktur data yang sudah tersedia seperti *dictionary* dan *list*.

Alur program adalah sebagai berikut:

1. Program dimulai dengan meminta masukan file test case yang akan diuji dari pengguna.
2. Menghilangkan tanda titik (.) dan koma (,) dari file masukan kemudian dimasukkan ke dalam list dengan setiap elemen dipisahkan oleh *newline*.
3. Pembuatan *adjacency list* dari list masukan file dengan memanfaatkan *dictionary* Python.
4. Inisialisasi *list of list* kosong dengan panjang 8 beserta sebuah integer bernilai 0 sebagai penanda jumlah iterasi
5. Mencari solusi dengan algoritma *topological sort* yang akan dijalankan secara rekursif. Perjalanan dimulai dengan mengunjungi node dengan derajat masuk 0 dan mengurangi derajat masuk node lain yang *incoming edge*-nya mengandung node awal tersebut. Langkah diulangi hingga seluruh node graf telah dikunjungi.
6. Selama iterasi, node yang sudah dikunjungi akan dimasukkan ke dalam *list of list* dengan urutan menaik dari derajat masuknya.
7. Mencetak urutan pengambilan mata kuliah sesuai format berdasarkan *list of list* mata kuliah yang dihasilkan dalam *topological sort* sebelumnya.

Secara singkat, *topological sorting* adalah proses pengurutan node graf sedemikian rupa sehingga untuk setiap *edge* berarah dari node u ke node v , u berada sebelum v dalam urutannya. *Topological sort* hanya dapat diaplikasikan jika graf berupa *Directed Acyclic Graph* (DAG) atau tidak memiliki siklus.

Jika dikaitkan dengan pendekatan *decrease and conquer*, algoritma *topological sort* termasuk ke dalam varian *decrease by constant*. Algoritma ini akan mengurangi ukuran graf sebanyak satu node dalam setiap penerapannya hingga node tersebut dihapus. Dalam varian *decrease by constant*, konstanta pengurangan yang digunakan biasanya adalah satu dan jumlah ini tidak berubah di tiap pengulangannya. Oleh karena itu, dapat disimpulkan *topological sort* termasuk dalam *decrease and conquer* varian *decrease by constant*.

II. Source Code

```
"""
Course Organizer w/ Toposort
Author: Rayhan Alghifari Fauzta (13519039)
"""

"""
Read from file
Input: -
Process: remove comma and stop from input file
Output: list of courses
"""

def read_file():
    filename = input('Masukkan test case (tanpa ".txt"): ')
    with open("../test/" + filename + ".txt") as f:
        lines = f.read().strip().split("\n")

    clear_comma = []
    for l in lines:
        clear_comma.append(l.replace(",", ' '))

    clear_stop = []
    for l in clear_comma:
        clear_stop.append(l.replace(".", ''))

    return clear_stop

"""
Create adjacency list
Input: list of courses
Process: take first element of course as key and its prerequisites as values
Output: adjacency list (dictionary)
"""

def make_adj_list(courses):
    adj = {}
    for course in courses:
        ls = course.split()
        adj[ls[0]] = ls[1:]

    return adj
```

```

'''
Topological sorting algorithm
Input: adjacency list (dict), empty list of list, current iteration number (start from 0)
Process:
    1. find any node with zero incoming edge
    2. add the node to list with index according to iteration number
    3. make a copy of adjacency list, delete previously visited node as well as any edges containing its value
    4. repeat until all nodes are visited
Output: list of courses per semester (list of list)
'''

def toposort(adj, list_smt, cur_smt):
    if (adj):
        for k, v in adj.items():
            if (len(adj[k]) == 0):
                list_smt[cur_smt].append(k)

        adj2 = adj.copy()
        for element in list_smt:
            for course in element:
                for k in list(adj2):
                    if k == course:
                        del adj2[k]
                for v in adj2.values():
                    for val in v:
                        if course in val:
                            v.remove(course)

        toposort(adj2, list_smt, cur_smt + 1)
    return list_smt

```

```

'''
Print result with the following style:
    Semester 1: a, b,
    Semester 2: e, f, g, h
    Semester 3: i
    and so on
Input: courses per semester (list of list)
Process: printing result according to specified format
Output: result printed
'''

def print_result(result):
    sem = 1
    for courses in result:
        if courses:
            course = ", ".join(courses)
            print("Semester {}: ".format(sem), course)
            sem += 1

''' main program '''
def main():
    f = read_file()
    adj = make_adj_list(f)
    list_smt = [[] for _ in range(8)]
    cur_smt = 0
    result = toposort(adj, list_smt, cur_smt)
    print_result(result)

if __name__ == "__main__":
    main()

```

III. Testing

Data Uji	Hasil
C1, C3. C2, C1, C4. C3. C4, C1, C3. C5, C2, C4.	ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/ \$ python 13519039.py Masukkan test case (tanpa ".txt"): test1 Semester 1: C3 Semester 2: C1 Semester 3: C4 Semester 4: C2 Semester 5: C5
C1, C2. C2, C4. C3. C4, C3. C5, C3. C6.	ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/ \$ python 13519039.py Masukkan test case (tanpa ".txt"): test2 Semester 1: C3, C6 Semester 2: C4, C5 Semester 3: C2 Semester 4: C1
C1, C3. C2, C3, C5, C7. C3. C4, C1, C3. C5. C6, C5, C3. C7, C4.	ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/ \$ python 13519039.py Masukkan test case (tanpa ".txt"): test3 Semester 1: C3, C5 Semester 2: C1, C6 Semester 3: C4 Semester 4: C7 Semester 5: C2
MA1201, MA1101. FI1201, FI1101. IF1210, KU1102. KU1202, KU1102. KI1002, KU1011. EL1200, FI1101. KU1102. MA1101. FI1101. KU1011.	ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/college/stima/tuci \$ python 13519039.py Masukkan test case (tanpa ".txt"): test4 Semester 1: KU1102, MA1101, FI1101, KU1011 Semester 2: MA1201, FI1201, IF1210, KU1202, KI1002, EL1200
MA1101. MA1201, MA1101. IF2130. IF2110. IF2121. IF2124, IF2120, IF2110. IF2120. IF2220, MA1101, MA1201, IF2120. IF2211. IF3170, IF2121, IF2124, IF2220, IF2211. IF3210, IF2130, IF2110. IF3270, IF3170, IF2110.	ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/college/stima/tuci \$ python 13519039.py Masukkan test case (tanpa ".txt"): test5 Semester 1: MA1101, IF2130, IF2110, IF2121, IF2120, IF2211 Semester 2: MA1201, IF2124, IF3210 Semester 3: IF2220 Semester 4: IF3170 Semester 5: IF3270

MA1101. EL1200, MA1101. IF2110. IF2130. IF2123, MA1101. IF2210, IF2110. IF2220, MA1101, IF2230. IF2240. IF2250. IF3170, IF2220, IF3110, IF2210, IF2110. IF3130, IF2230. IF3141, IF2240, IF2250. IF3150, IF2250. IF3151, IF2250. IF3210, IF2130, IF2110. IF3270, IF3170, IF2110. IF3230, IF3130. IF3250, IF3150, IF2250. IF3260, IF2130, IF2110, IF2123.	<pre> ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/college/stima/tucil/tucil 2/src \$ python 13519039.py Masukkan test case (tanpa ".txt"): test6 Semester 1: MA1101, IF2110, IF2130, IF2230, IF2240, IF2250 Semester 2: EL1200, IF2123, IF2210, IF2220, IF3130, IF3141, IF3150, IF3151, IF3210 Semester 3: IF3170, IF3110, IF3230, IF3250, IF3260 Semester 4: IF3270 </pre>
MA1101. MA1201. EP2091, MA1101, MA1201. EL1200. EL2001, EL1200. EP2076, EL2001. MA2072. EP2094, MA2072, EL2001. EL2005, EL2001. EL2006, MA2072, FI1201. FI1201. EP3071, EL2001, EL2006. EP3073, MA1101, MA1201. EP3075, EL2001. EP3095, EL2006. TI3004, MA1101, MA1201.	<pre> ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/college/stima/tucil/ \$ python 13519039.py Masukkan test case (tanpa ".txt"): test7 Semester 1: MA1101, MA1201, EL1200, MA2072, FI1201 Semester 2: EP2091, EL2001, EL2006, EP3073, TI3004 Semester 3: EP2076, EP2094, EL2005, EP3071, EP3075, EP3095 </pre>

MA1101. FI1101. KU1001. KU1102. KU1011. KU1024. MA1201, MA1101. FI1201, FI1101. IF1210, KU1102. KU1202, KU1102. KI1002, KU1011. EL1200, FI1101. IF2121, IF1210, MA1101, MA1201. IF2110, KU1102, IF1210. IF2120, MA1201, MA1101. IF2124, EL1200. IF2123, MA1201. IF2130, KU1202. IF2210, IF2110. IF2211, IF2110. IF2220, MA1101, MA1201, IF2120. IF2230, IF2130. IF2240, IF2121, IF2120. IF2250, KU1202, IF2110. IF3170, IF2121, IF2124, IF2220, IF2211. IF3110, IF2210, IF2110. IF3130, IF2230. IF3141, IF2240, IF2250. IF3150, IF2250. IF3140, IF2240. IF3151, IF2250. IF3210, IF2110, IF2130, IF3110. IF3270, IF2210, IF3170. IF3230, IF3130. IF3250, IF2250, IF3150. IF3260, IF2123, IF2110, IF2130, IF3151. IF3280, IF3151, IF3150. IF4090, IF3280. IF4091, IF3280. IF4092, IF4091.	<pre> ASUS@LAPTOP-GQF2KGUJ MINGW64 ~/Documents/college/stima/tucil/tucil \$ python 13519039.py Masukkan test case (tanpa ".txt"): test8 Semester 1: MA1101, FI1101, KU1001, KU1102, KU1011, KU1024 Semester 2: MA1201, FI1201, IF1210, KU1202, KI1002, EL1200 Semester 3: IF2121, IF2110, IF2120, IF2124, IF2123, IF2130 Semester 4: IF2210, IF2211, IF2220, IF2230, IF2240, IF2250 Semester 5: IF3170, IF3110, IF3130, IF3141, IF3150, IF3140, IF3151 Semester 6: IF3210, IF3270, IF3230, IF3250, IF3260, IF3280 Semester 7: IF4090, IF4091 Semester 8: IF4092 </pre>
--	--

IV. Alamat Source Code

Source code program dapat diakses di repositori GitHub berikut:

<https://github.com/rayfazt/stima-toposort>

V. Checklist

No	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	✓	