

Laporan Tugas Kecil 2

Topological Sort

Oleh :

Denilsen Axel Candiasa / 13519059

Algoritma Topological Sort :

1. Nyatakan graf DAG (*Directed Acyclic Graph*) ke dalam senarai ketetanggaan yang menyatakan setiap simpul dengan simpul awalnya (simpul yang menjadi *pre-requisitenya*).
2. Pilih sembarang simpul yang memiliki derajat-masuk 0 (dalam hal ini tidak memiliki simpul awal / *pre-requisite*), kemudian masukkan ke dalam himpunan solusi.
3. Kemudian, hapus simpul yang sudah dipilih beserta seluruh busur yang keluar dari simpul tersebut, kurangi pula simpul tersebut dari senarai ketetanggaan simpul yang bertetanggaan dengannya.
4. Ulangi terus langkah (2) dan (3) sampai senarai ketetanggaan graf tersebut kosong (semua simpul dalam DAG terpilih). Himpunan solusi terakhir ketika semua simpul dalam DAG terpilih sudah pasti terurut

Kaitan antara algoritma Topological Sort diatas dengan pendekatan Decrease and Conquer adalah dalam algoritma Topological Sort diatas, terdapat reduksi jumlah simpul DAG dalam setiap langkah (2) dan (3) yang kemudian akan dicari solusi dari setiap iterasinya, dalam hal ini solusi dari setiap iterasi adalah simpul “terkecil” di iterasi tersebut. Adapun varian dari algoritma Topological Sort diatas adalah *decrease by a variable size*, karena untuk setiap iterasinya, jumlah simpul yang direduksi bervariasi, bisa saja dalam satu iterasi terdapat 1 simpul saja yang direduksi, bisa juga terdapat 2 atau lebih simpul yang direduksi. Adapun kompleksitas dari algoritma ini adalah $O(V + E)$, dimana V adalah jumlah simpul, dan E adalah jumlah total derajat simpul yang ada dalam graf.

Source Code Program (dalam Python) :

Berikut *screenshot* dari source code program SmortPRS dalam bahasa Python :

```
import Utility

solution = {}
semester = 1

def findAndDelete(courseDict, delCourse):
    # Fungsi untuk mencari mata kuliah yang merupakan pre-
    # requisite suatu mata kuliah
    # lalu menghapusnya dari list pre-requisite mata kuliah tersebut

    # Looping di setiap mata kuliah
    for course in courseDict:
        # Jika delCourse ada dalam list pre-requisite, hapus dari list
        if(delCourse in courseDict[course]):
            courseDict[course].remove(delCourse)

def topoSort(courseDict):
    # Algoritma Topological Sort untuk memperoleh solusi

    global solution
    global semester
    selectedCourse = []

    # Berhenti jika semua mata kuliah sudah masuk ke dalam himpunan solusi
    if(len(courseDict) == 0):
        return

    # Mencari mata kuliah yang semua pre-requisitenya sudah diambil
    for course in courseDict:
        if(len(courseDict[course]) == 0):
            selectedCourse.append(course)

    # Menghapus mata kuliah tersebut dari list pre-
    # requisite mata kuliah lain
    for course in selectedCourse:
        findAndDelete(courseDict, course)
        courseDict.pop(course)

    # Memasukkan mata kuliah tersebut ke dalam himpunan solusi
    solution.update({semester: selectedCourse})
```

```

semester += 1

topoSort(courseDict)

def printSolution():
    # Fungsi Utilitas untuk mencetak himpunan solusi yang berisi urutan pe-
    ngambilan mata kuliah

    global solution
    for semester in solution:
        print("Semester " + Utility.numToRoman(semester) + "\t: ", end="")
        for course in solution[semester]:
            print(course + " ", end="")
        print()

def run(fileName):
    # Fungsi yang digunakan di GUI untuk memproses solusi sebelum ditampil-
    kan ke GUI

    lineArr = Utility.readFile(fileName)
    courseDict = Utility.createCourseDict(lineArr)
    topoSort(courseDict)

def clear():
    # Fungsi untuk meng-
    clear solusi untuk menunggu input berikutnya dari GUI

    global solution
    global semester
    solution.clear()
    semester = 1

# path = "../test/"
# fileName = input("Masukkan Nama File: ")
# run(path + fileName)
# printSolution()

```

Berikut *screenshot* dari source code program Utility dalam bahasa Python :

```
def readFile(fileName):
```

```

# Fungsi untuk membaca input dari file eksternal

f_input = open(fileName, "r")
line_input = f_input.read().splitlines()
f_input.close()
return line_input

def createCourseDict(lineArr):
    # Fungsi untuk membuat dictionary mata kuliah beserta list mata kuliah
    pre-requisitenya
    # Menerima input list of strings dimana tiap elemen list merupakan bar
    is dalam input file .txt

    courseDict = {}

    # Looping di setiap baris
    for line in lineArr:
        courseCount = 0
        courseID = ""
        currCourse = ""
        currPrerequisite = []
        i = 0

        # Parsing string menjadi dictionary mata kuliah
        for i in range(len(line)):
            if(line[i] == ',' or line[i] == '.'):
                if(courseCount > 0):
                    currPrerequisite.append(courseID)
                else:
                    currCourse = courseID
                    courseID = ""
                    courseCount += 1
            elif(line[i] != ' '):
                courseID += line[i]
            i += 1

        courseDict.update({currCourse: currPrerequisite})

    return courseDict

def numToRoman(num):
    # Fungsi utilitas untuk mengubah angka arab menjadi angka romawi

```

```

val = [10, 9, 5, 4, 1]
roman_val = ["X", "IX", "V", "IV", "I"]
ans = ""
i = 0
while num > 0:
    for _ in range(num // val[i]):
        ans += roman_val[i]
        num -= val[i]
    i += 1
return ans

```

Berikut *screenshot* dari source code program SmortPRS_GUI dalam bahasa Python :

```

import SmortPRS
import Utility
import PySimpleGUI as sg
import os.path

# Pengaturan layout window dalam 2 kolom

file_list_column = [
    [
        sg.Text("Silahkan pilih folder tempat file disimpan!!"),
    ],
    [
        sg.In(size=(20, 1), enable_events=True, key="-FOLDER-"),
        sg.FolderBrowse(), ],
    [
        sg.Listbox(
            values=[], enable_events=True, size=(30, 20), key="-FILE LIST-"
        ),
    ],
]

solution_viewer_column = [
    [sg.Text("Silahkan pilih file .txt yang berisi list mata kuliah dan pr
e-requisitenya")],
    [sg.Text(size=(40, 1), key="-TOUT-")],
    [sg.Listbox(values=[], enable_events=True,
                size=(150, 20), key="-SOLUTION-")],
    [sg.Button("Clear")],
]

```

```

layout = [
    [
        sg.Column(file_list_column),
        sg.VSeperator(),
        sg.Column(solution_viewer_column),
    ]
]

window = sg.Window("SmortPRS", layout)

# Array untuk menyimpan list kuliah
output = []

while True:
    event, values = window.read()
    if event == "Exit" or event == sg.WIN_CLOSED:
        break

    # Nama folder sudah dimasukkan, selanjutnya akan menampilkan list file
    # yang bisa dipilih
    if event == "-FOLDER-":
        folder = values["-FOLDER-"]
        try:
            # Mengambil list file dari folder
            file_list = os.listdir(folder)
        except:
            file_list = []

        fnames = [
            f
            for f in file_list
            if os.path.isfile(os.path.join(folder, f))
            and f.lower().endswith(".txt")
        ]
        window["-FILE LIST-"].update(fnames)

    elif event == "-FILE LIST-": # Sebuah file dipilih oleh user
        try:
            filename = os.path.join(
                values["-FOLDER-"], values["-FILE LIST-"][0]
            )

            window["-TOUT-"].update(filename)

            SmortPRS.run(filename)

```

```

        # Memasukkan solution ke dalam list output
        for semester in SmortPRS.solution:
            currSemester = "Semester "
            currSemester += Utility.numToRoman(semester)
            currSemester += "\t: "
            for course in SmortPRS.solution[semester]:
                currSemester += course
                currSemester += " "
            output.append(currSemester)

        window["-SOLUTION-"].update(output)
    except:
        pass

    elif event == "Clear": # Akan mereset dict solution dan list output u
ntuk menampung solusi lain
        SmortPRS.clear()
        output.clear()
        window["-SOLUTION-"].update(output)

window.close()

```

Screenshoot Input dan Output Program :

1. Input :

soal1 - Notepad

File Edit Format View Help

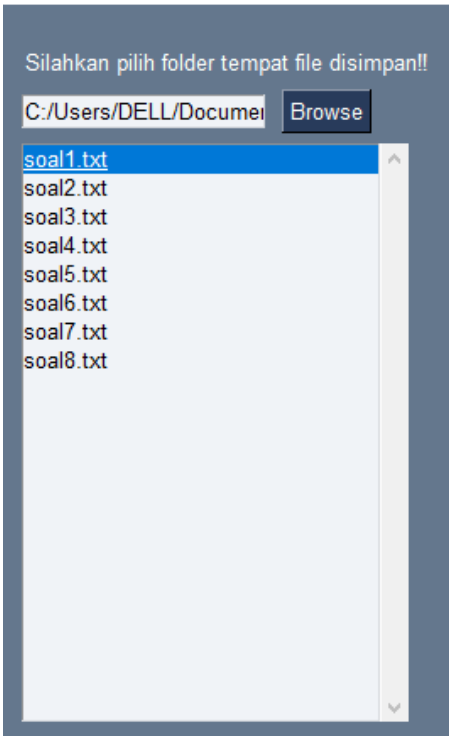
C1, C3.

C2, C1, C4.

C3.

C4, C1, C3.

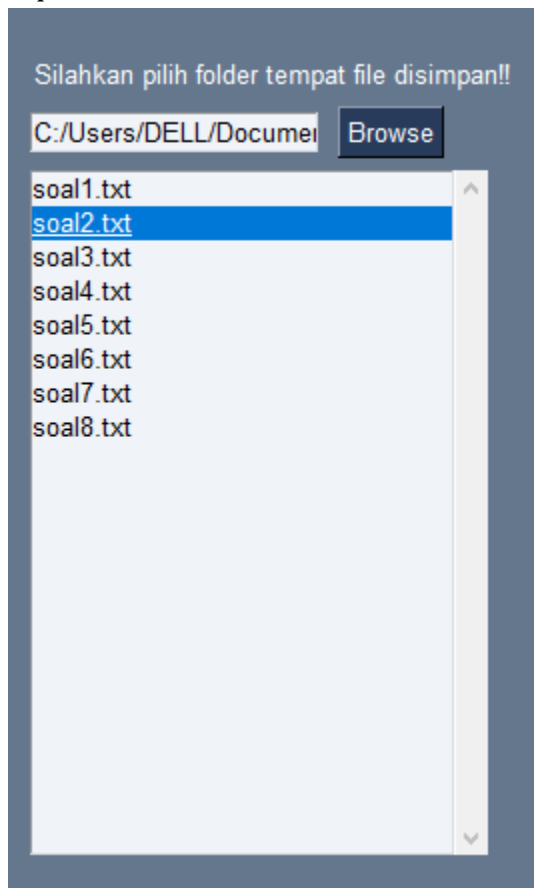
C5, C2, C4.



Output :



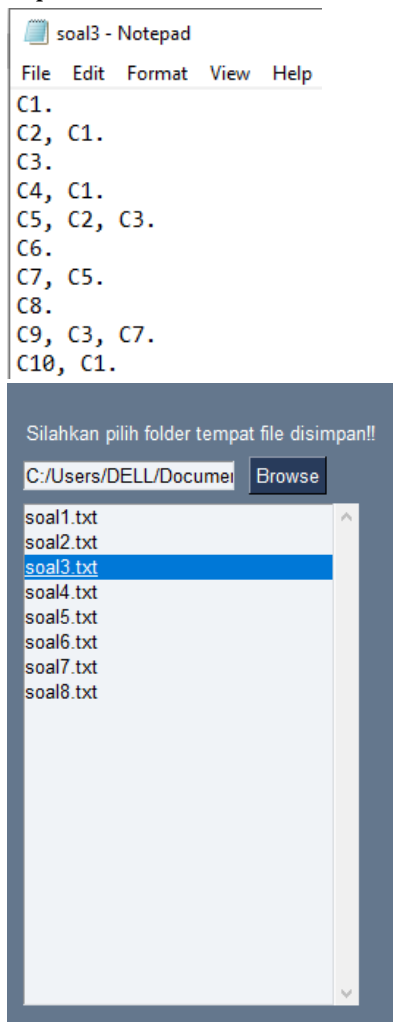
2. Input :



Output :



3. Input :



Output :

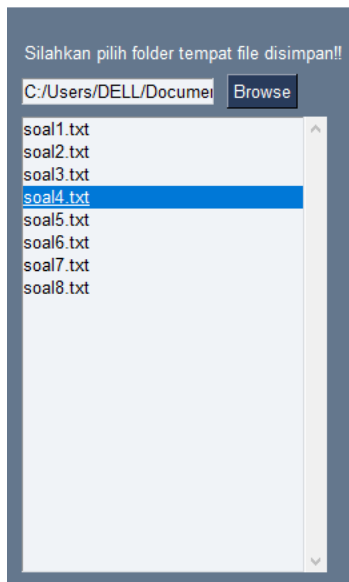


4. Input :

soal4 - Notepad

File Edit Format View Help

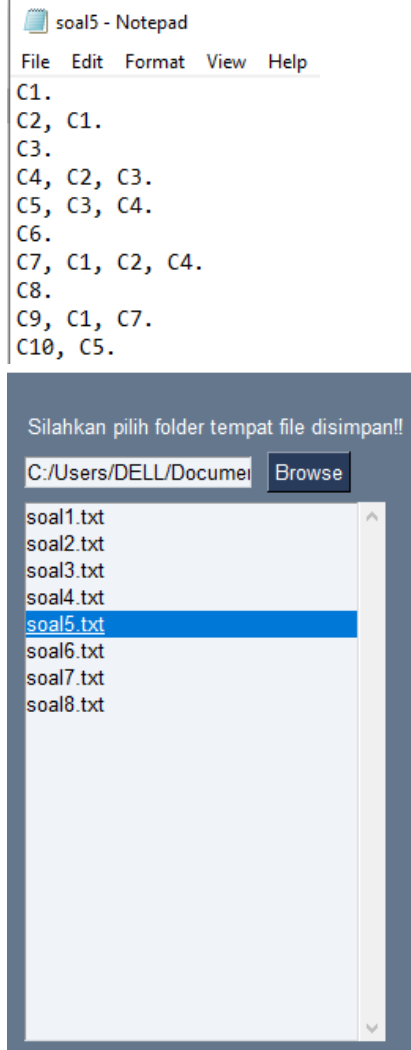
C1.
C2.
C3, C1.
C4.
C5, C2.
C6.
C7, C3, C6.
C8, C1, C3, C4.
C9, C3.
C10, C5, C9.



Output :



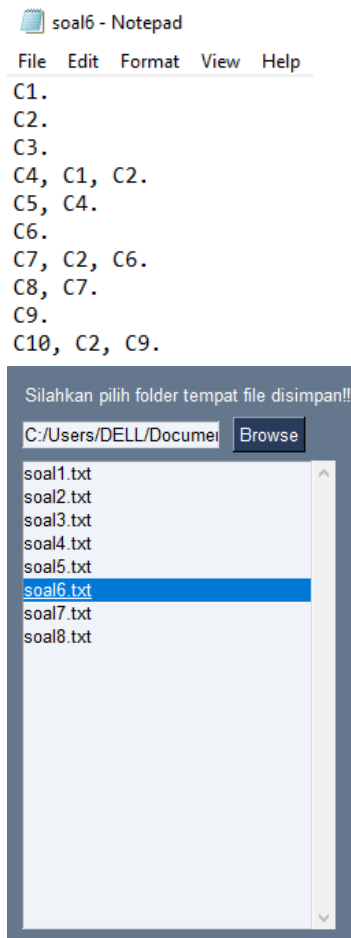
5. Input :



Output :



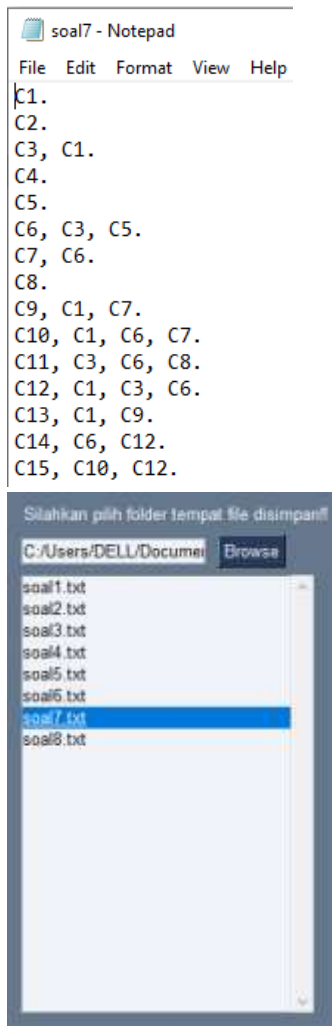
6. Input :



Output :



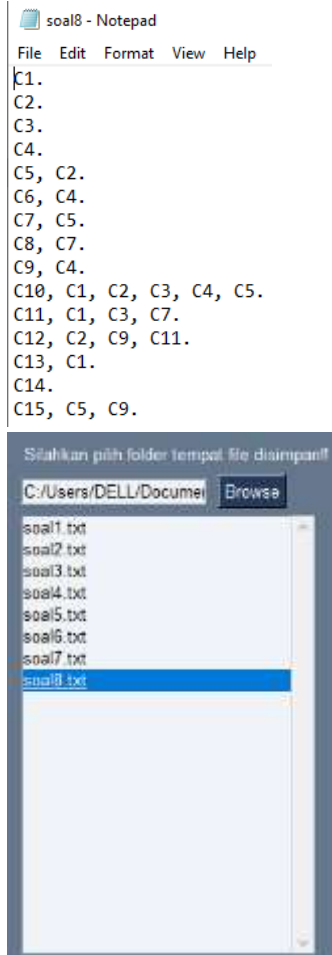
7. Input :



Output :



8. Input :



Output :



Alamat Source Code :

Dalam folder ../src, untuk alamat *executable file*-nya dalam folder ../bin

Tabel Ceklis :

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	√	