1. ALGORITMA TOPOLOGICAL SORT DAN KAITANNYA DENGAN DECREASE AND CONQUER

Topological sort untuk *directed acrylic graph*(DAG) adalah pengurutan linear dari node-node pada DAG sehingga pada sisi(u,v) node u selalu berada sebelum node v. Langkah-langkah algoritma topological sort yang dilakukan :

1. Membaca isi file yang berisi mata kuliah dan syarat nya, sehingga di konversi menjadi list of mata\_kuliah dimana mata\_kuliah adalah [<kodematakuliah>|<kodematakuliahsyarat>]
2. Lalu array diurutkan menaik berdasarkan jumlah mata kuliah syarat.
3. Jika list mata\_kuliah kosong print kosong
4. Mata\_kuliah yang memiliki matakuliahsyarat berjumlah 0 akan dihapus dari list mata\_kuliah, dan list matakuliahsyarat dan dimasukkan ke dalam sebuah list jawaban
5. Print setiap anggota list jawaban
6. Lalu fungsi topological sort dipanggil Kembali dengan list mata\_kuliah yang sudah dihapus isinya

Hubungan dari algoritma ini dengan decrease and conquer ada di bagian nomor 4, yaitu adanya proses decrease dengan mengurangi persoalan menjadi hanya mata\_kuliah yang memiliki jumlah matakuliahsyarat sebanyak 0 buah.

1. SOURCE CODE

import re

def OpenFile(filename) :

    f = open(filename, "r")

    a = ""

    for x in f :

        a+=x

    a = cleanString(a)

    a = cleanArray(a)

    a = sortByLen(a)

    b = topSort(a,0)

    f.close()

    return b

#fungsi untuk memasukkan tiap line ke dalam array dengan menghilangkan koma(,) dan titik(.)

def cleanString(x) :

    x = x.split(".\n") #setiap line selalu diakhiri dengan titik(.)

    for i in range(len(x)) :

        x[i] = re.split("[,|\s+|.]",x[i]) #asumsi format file selalu ada spasi setelah koma dan tidak ada spasi sebelum koma

    return x

def cleanArray(x) :

    finalArray = []

    for i in range (len(x)) :

        array = []

        for j in range (len(x[i])) :

            if x[i][j] != "" :

                array.append(x[i][j])

        finalArray.append(array)

    return finalArray

def sortByLen(x) :

    for i in range(len(x)) :

        for j in range(len(x) - 1) :

            if (len(x[j]) > len(x[j+1])) :

                temp = x[j]

                x[j] = x[j+1]

                x[j+1] = temp

    return x

def isIn(x,y) :

    found = False

    i = 0

    if (len(x) > 1) :

        while(i < len(x) and found == False) :

            if (x[i] == y) :

                x.pop(i)

                found = True

            else :

                i+=1

    return x

def topSort(x,n) :

    sortedArray = []

    thisLevel = []

    if (len(x) == 0) :

        return []

    elif (len(x) == 1) :

        sortedArray.append(x[0])

        formatPrint(sortedArray,n+1)

        return sortedArray[0]

    else :

        i = 0

        while(i <len(x) and len(x[i]) == 1) :

            i+=1

        k = 0

        while(k<i) :

            j = 1

            while (j < len(x)) :

                isIn(x[j], x[0][0])

                j+=1

            if (len(x) > 0) :

                thisLevel.append(x[0][0])

                x.pop(0)

            x = sortByLen(x)

            k+=1

    sortedArray.append(thisLevel)

    formatPrint(sortedArray,n+1)

    sortedArray.append(topSort(x,n+1))

    return sortedArray

def formatPrint(x,i) :

    print("Semester " + str(i) + " : " , end="")

    for i in range (len(x)) :

        for j in range(len(x[i])) :

            if (j == len(x[i]) -1) :

                print(str(x[i][j]))

            else :

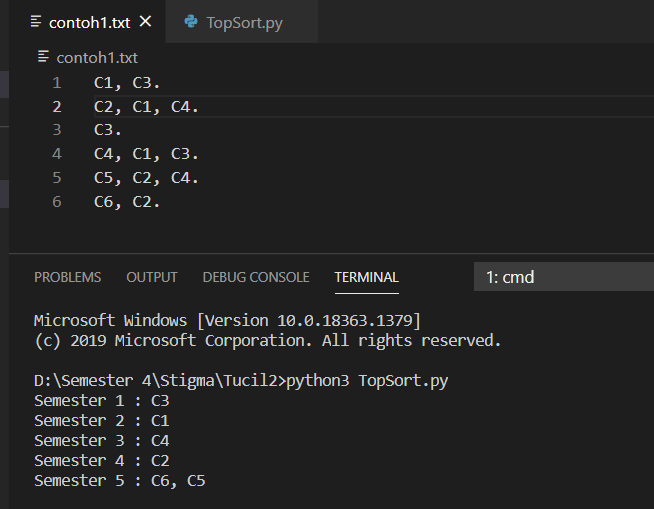
                print(str(x[i][j])+", ",end = "")

x = OpenFile("contoh5.txt")

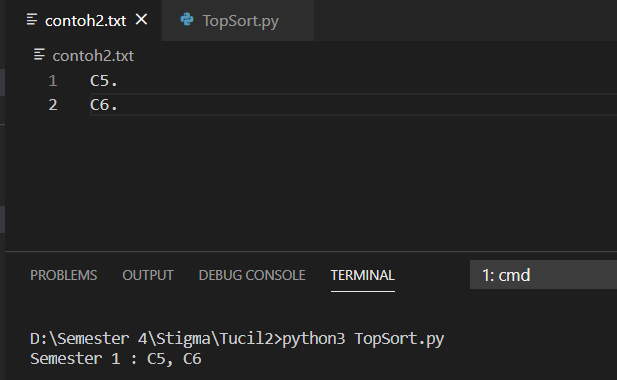
1. SCREENSHOT HASIL

Screenshot output :

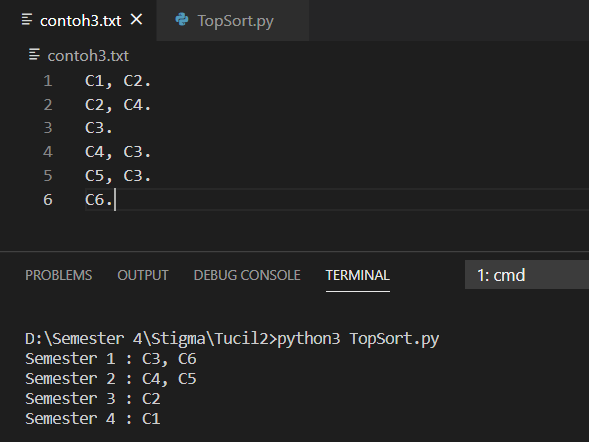
Test1



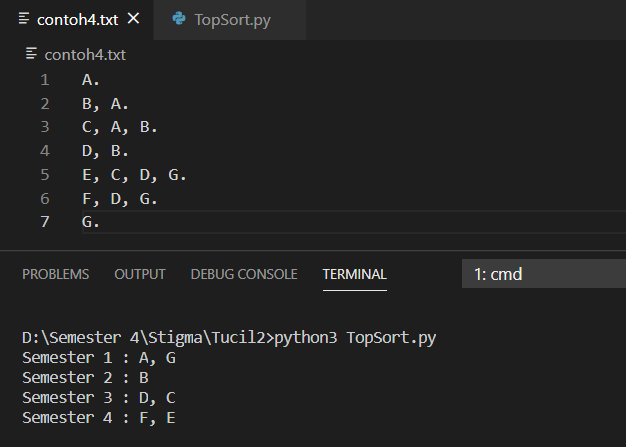
Test2



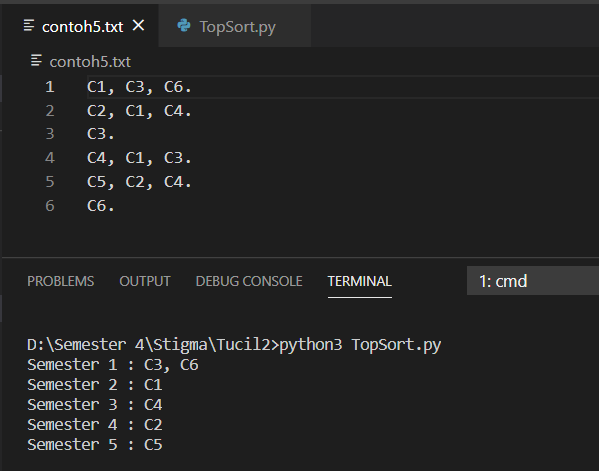
Test3



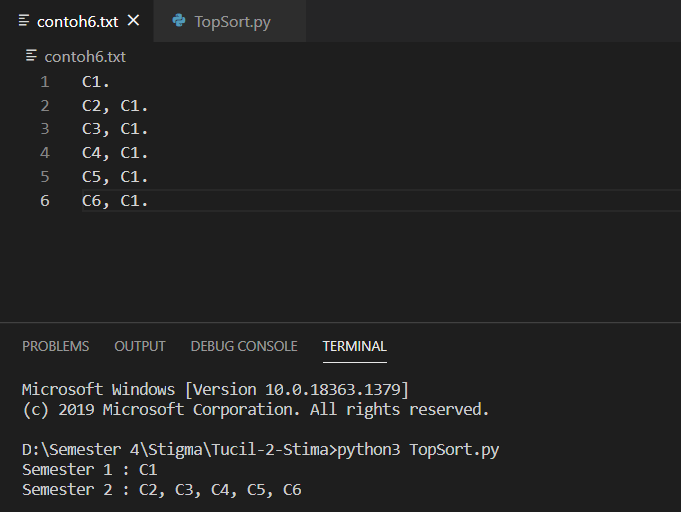
Test4



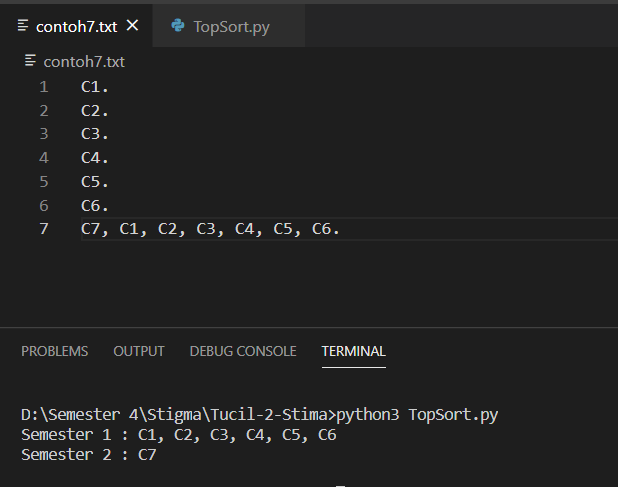
Test5



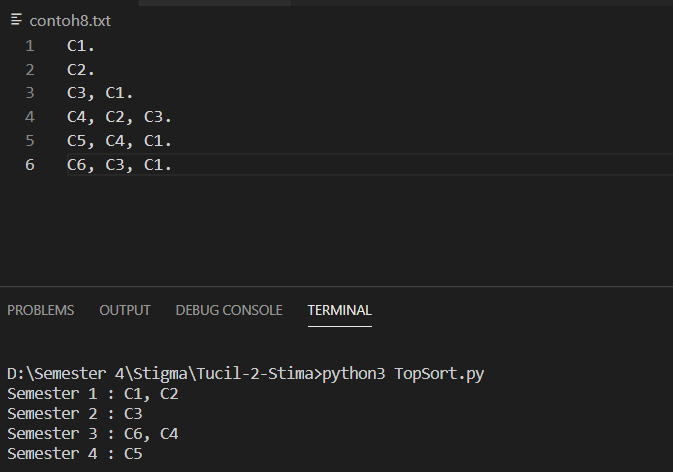
Test6



Test7



Test8



1. ALAMAT GITHUB

https://github.com/rehagana16/Tucil-2-Stima

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program Berhasil di kompilasi | ✔ |  |
| 2. Program berhasil *running* | ✔ |  |
| 3. Program dapat menerima berkas input dan menuliskan output | ✔ |  |
| 4. Luaran sudah benar untuk seluruh kasus input | ✔ |  |