

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



Disusun Oleh:

Mochammad Fatchur Rochman 13519009

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

BAB I

Algoritma Topological Sort dan kaitannya dengan pendekatan Decrease and Conquer

Topological Sorting

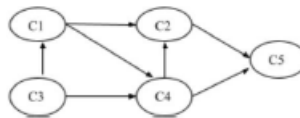
Adalah suatu metode sorting pada Directed Acyclic Graph untuk menentukan keterurutan setiap simpul dari simpul yang paling diutamakan sampai yang paling tidak diutamakan. Dalam permasalahan penyusunan rencana kuliah yang membutuhkan pre-requisite kaitannya dengan Decrease and Conquer adalah untuk setiap vertex/simpul yang dalam permasalahan ini adalah matakuliah yang memiliki degree-in sama dengan 0 atau dengan kata lain tidak memiliki pre-requisite akan dihilangkan dari DAG penyusunan kuliah, lalu disimpan didalam suatu list, dan untuk setiap simpul yang keluar dari simpul tersebut akan dikurangi degree nya sebanyak 1.

Untuk ilustrasi, misalnya ada 5 kode kuliah beserta pre-requisitenya

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

Pendekatan Topological Sorting dengan Decrease and Conquer (mengurangi source removal) :

DAG untuk kode kuliah dan pre-requisitenya



a. Dari graf (DAG) yang terbentuk, hitung semua derajat-masuk (*in-degree*) setiap simpul, yaitu banyaknya busur yang masuk pada simpul tersebut. Pada contoh kasus di Gambar 2, maka derajat-masuk tiap simpul adalah sebagai berikut.

C1 : 1
C2 : 2
C3 : 0
C4 : 2
C5 : 2

b. Pilih sembarang simpul yang memiliki derajat-masuk 0. Pada kasus Gambar 2, pilih simpul C3.

c. Ambil simpul tersebut, dan hilangkan simpul tersebut beserta semua busur yang keluar dari simpul tersebut pada graf, dan kurangi derajat simpul yang berhubungan dengan simpul tersebut dengan 1.

Setelah simpul C3 dipilih, maka derajat simpul yang lain menjadi sebagai berikut.

C1 : 0
C2 : 2
C4 : 1
C5 : 2

Ulangi langkah (b) dan (c) hingga semua simpul pada DAG terpilih. Untuk kasus pada Gambar 2, setelah simpul terakhir dipilih rencana kuliah yang dihasilkan adalah sebagai berikut.


Semester I : C3
Semester II : C1
Semester III : C4
Semester IV : C2
Semester V : C5

Kebetulan untuk contoh ini, satu semester hanya ada 1 kuliah.

BAB II

Source Code Program

main_13519009.py

 main_13519009.py - D:\STIMA\TUCIL\Tucil2_13519009\src\main_13519009.py (3.9.2)

File Edit Format Run Options Window Help

```
1 # NIM>Nama      : 13519009/Mochammad Fatchur Rochman
2 # Program      : Program Aplikasi Topology Sorting dengan Decrease and Conquer
3
4
5 # import file
6 import fungsi_13519009 as f
7
8 # ===== Main Program =====
9 listAdja = f.makeListAdja()
10 listVertex = f.makeListVertex()
11 AdjaMatrix = f.makeAdjaMatrix(listAdja,listVertex)
12
13 # Print Soal dari file test.txt yang ada di folder test
14 f.printSoal()
15 # Print Hasil data file test.txt setelah dilakukan Topological Sorting
16 print("Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting : ")
17 print(f.TopologicalSorting(listAdja,listVertex))
18 print()
19 # Print Solusi Penyusunan Mata Kuliah yang memungkinkan
20 print("Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut : ")
21 Solusi = f.TopologicalSorting(listAdja,listVertex)
22 f.printSolusi(Solusi)
23 print()
24
```

fungsi_13519009.py

```
fungsi_13519009.py - D:\STIMA\TUCIL\Tucil2_13519009\src\fungsi_13519009.py (3.9.2)
File Edit Format Run Options Window Help

# CATATAN : Jika ingin mengubah file.txt yang akan diuji pada program ini
#           dapat mengubah '../test/test5.txt' yang terdapat pada
#           bagian PATH, ubah test5.txt dengan file.txt lainnya yang
#           terdapat pada folder test.

# PATH
path = '../test/test3.txt'

# ===== Fungsi - Fungsi yang berhubungan dengan Graph =====

# Fungsi Buka File Text dan Membuat List Adjacencynya
def makeListAdja():
    listAdjacency = []
    with open(path, 'r') as file:
        for datas in file:
            listTemp = []
            stringTemp = ''
            data = datas.replace('\n', '')
            for data_i in data:
                if(data_i == ','):
                    listTemp = listTemp + [stringTemp]
                    stringTemp = ''
                elif(data_i == '.'):
                    listTemp = listTemp + [stringTemp]
                    listAdjacency = listAdjacency + [listTemp]
                    listTemp = []
                elif(data_i != ' '):
                    stringTemp = stringTemp + data_i

            return listAdjacency

# Fungsi Mencari Panjang list
def listLength(lis):
    leng = 0
    for data in lis:
        leng += 1
    return leng

# Fungsi Membuat list Vertex
def makeListVertex():
    listAdja = makeListAdja()
    lengList = listLength(listAdja)
    listVertex = []

    for i in range(lengList):
        listVertex = listVertex + [listAdja[i][0]]

    return listVertex

# Fungsi Menghitung Vertex
def countVertex(V):
    count = 0
    for i in V:
        count += 1
    return count
```

```

# Fungsi Membuat Adjacency Matrix
def makeAdjaMatrix(listAdja, listVertex):
    lengMatrix = listLength(listVertex)
    AdjaMatrix = [[0 for j in range(lengMatrix)] for i in range(lengMatrix)]

    lengListAdja = listLength(listAdja)
    lengListVertex = listLength(listVertex)

    for vi in listVertex:
        i_loop = 0
        found = False
        i = 0
        while(i_loop < lengListAdja and (not found)):
            if(vi == listAdja[i_loop][0]):
                found = True
                i = i_loop
                i_loop = 0
            else:
                i_loop += 1
        found = False

        # isi Adjacency Matrix
        lengListAdjai = listLength(listAdja[i])
        for j in range(1, lengListAdjai):
            for k in range(lengListVertex):
                if(listAdja[i][j] == listVertex[k]):
                    AdjaMatrix[k][i] += 1

    return AdjaMatrix

# Fungsi Menghitung degree-in (din) pada suatu Vertex
def dinV(j, AdjaMatrix):
    lengAdjaMatrix = listLength(AdjaMatrix)
    din = 0
    for i in range(lengAdjaMatrix):
        din += AdjaMatrix[i][j]

    return din

```

```

# Fungsi untuk Mengetahui Apakah keduanya tetangga-an
def isTetangga(i,j):
    listAdj = makeListAdja()
    listVer = makeListVertex()
    adjaMatrix = makeAdjaMatrix(listAdj, listVer)

    if(adjaMatrix[i][j] == 1 or adjaMatrix[j][i] == 1):
        return True
    else:
        return False

# Fungsi untuk mengetahui index dari suatu Vertex
def getIndexVertex(V):
    listVer = makeListVertex()
    index = -1
    x = 0
    for vertex in listVer:
        if(V == vertex):
            index = x
            x += 1
    if(index == -1):
        print("\nSimpul tersebut tidak ada di list Vertex ini")
    else:
        return index

# Algoritma Topological Sorting dengan Decrease and Conquer
def TopologicalSorting(listAdja,listVertex):

    AdjaMatrix = makeAdjaMatrix(listAdja,listVertex)
    lengAdjaMatrix = listLength(AdjaMatrix)
    lengListVertex = listLength(listVertex)

    listSorted = []
    listVertexIn = [False for j in range(lengListVertex)]

    allVertexListed = False
    vertexIn = 0
    nVertex = countVertex(listVertex)

    while(not allVertexListed):
        for j in range (lengListVertex):
            # cek apakah sudah dimasukkan ke list Sorted
            In = listVertexIn[j]
            if(not In):
                din = dinV(j,AdjaMatrix)
                if(din == 0):
                    listSorted = listSorted + [listVertex[j]]
                    listVertexIn[j] = True
                    vertexIn += 1
                    # Menghapus Vertex listVertex[j] pada AdjaMatrix
                    for i in range(lengListVertex):
                        AdjaMatrix[j][i] = 0

            # Cek apakah semua vertex sudah masuk ke listSorted
            if(vertexIn == nVertex):
                allVertexListed = True

    return listSorted

```

```

# ===== Tulis =====

# Prosedur Tulis Semester
def printSemester(i_semester, matkul):
    listSemester = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII']
    if(i_semester == 0 or i_semester == 4):
        print("Semester", listSemester[i_semester], ' : ', matkul)
    elif(i_semester == 1 or i_semester == 3 or i_semester == 5):
        print("Semester", listSemester[i_semester], ' : ', matkul)
    elif(i_semester == 2 or i_semester == 6):
        print("Semester", listSemester[i_semester], ' : ', matkul)
    else:
        print("Semester", listSemester[i_semester], ': ', matkul)

# Prosedur Tulis Solusi
def printSolusi(Solusi):
    lengSolusi = listLength(Solusi)
    nSemester = 8
    listMatkul = makeListVertex()

    semester_i = 0
    i_listSolusi = 0
    i_v = getIndexVertex(Solusi[i_listSolusi])
    j_v = getIndexVertex(Solusi[i_listSolusi+1])
    countMatkul = 0
    complete = False

    while(semester_i < nSemester and (not complete)):
        strMatkulSemester_i = Solusi[i_listSolusi]
        i_listSolusi += 1
        countMatkul += 1
        tetangga = isTetangga(i_v, j_v)
        # Melakukan pendataan mata kuliah (Vertex) apa yang bisa dimasukkan pada semester ini
        while(not tetangga and (not complete)):
            strMatkulSemester_i = strMatkulSemester_i + ', ' + listMatkul[j_v]
            countMatkul += 1
            if(countMatkul == lengSolusi):
                complete = True
            else:
                i_listSolusi += 1
                j_v = getIndexVertex(Solusi[i_listSolusi])
                tetangga = isTetangga(i_v, j_v)
        # Melakukan pencetakan ke layar matkul semester ini
        printSemester(semester_i, strMatkulSemester_i)
        # melakukan pengecekan untuk anggota listSolusi selanjutnya
        if(lengSolusi - countMatkul >= 2):
            i_v = getIndexVertex(Solusi[i_listSolusi])
            j_v = getIndexVertex(Solusi[i_listSolusi+1])
        else:
            if(countMatkul == lengSolusi):
                complete = True

        semester_i += 1

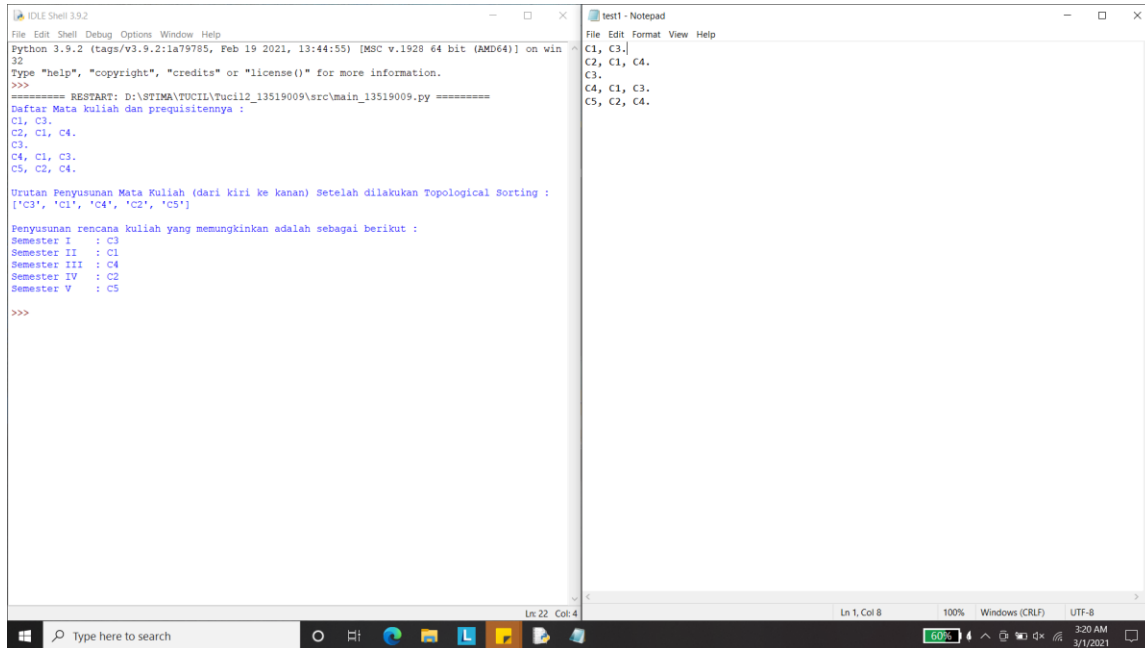
# Prosedur Tulis Soal
def printSoal():
    print("Daftar Mata kuliah dan prequisitenya : ")
    with open(path, 'r') as file:
        soal = file.read()
    print(soal, '\n')

```


BAB III

Tangkapan layar dari input dan output

test1.txt



The screenshot shows two windows: IDLE Shell 3.9.2 and Notepad. The IDLE Shell window displays the output of a Python script. The script defines a list of courses and their prerequisites, performs a topological sort, and outputs the sorted list and a semester plan. The Notepad window shows the input data for the script.

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL2_13519009\src\main_13519009.py =====
Daftar Mata kuliah dan prequisitenya :
C1, C3.
C2, C1, C4.
C3.
C4, C1, C3.
C5, C2, C4.

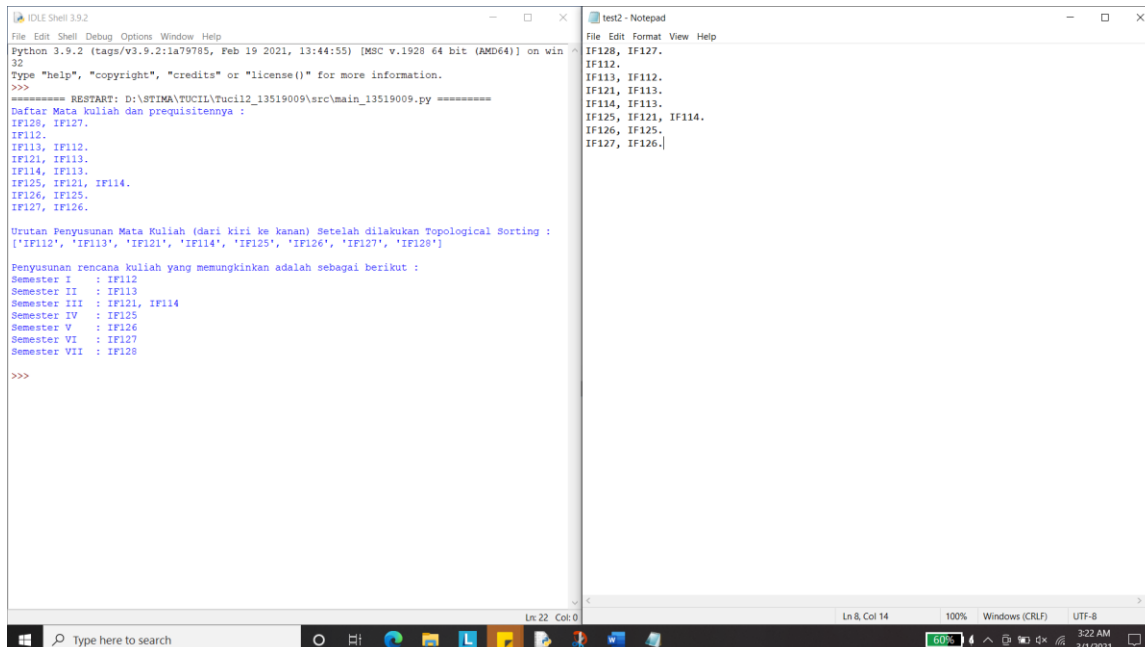
Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['C3', 'C1', 'C4', 'C2', 'C5']

Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut :
Semester I : C3
Semester II : C1
Semester III : C4
Semester IV : C2
Semester V : C5
>>>
```

test1 - Notepad

```
File Edit Format View Help
C1, C3.
C2, C1, C4.
C3.
C4, C1, C3.
C5, C2, C4.
```

test2.txt



The screenshot shows two windows: IDLE Shell 3.9.2 and Notepad. The IDLE Shell window displays the output of a Python script. The script defines a list of courses and their prerequisites, performs a topological sort, and outputs the sorted list and a semester plan. The Notepad window shows the input data for the script.

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL2_13519009\src\main_13519009.py =====
Daftar Mata kuliah dan prequisitenya :
IF128, IF127.
IF112.
IF113, IF112.
IF121, IF113.
IF114, IF113.
IF125, IF121, IF114.
IF126, IF125.
IF127, IF126.

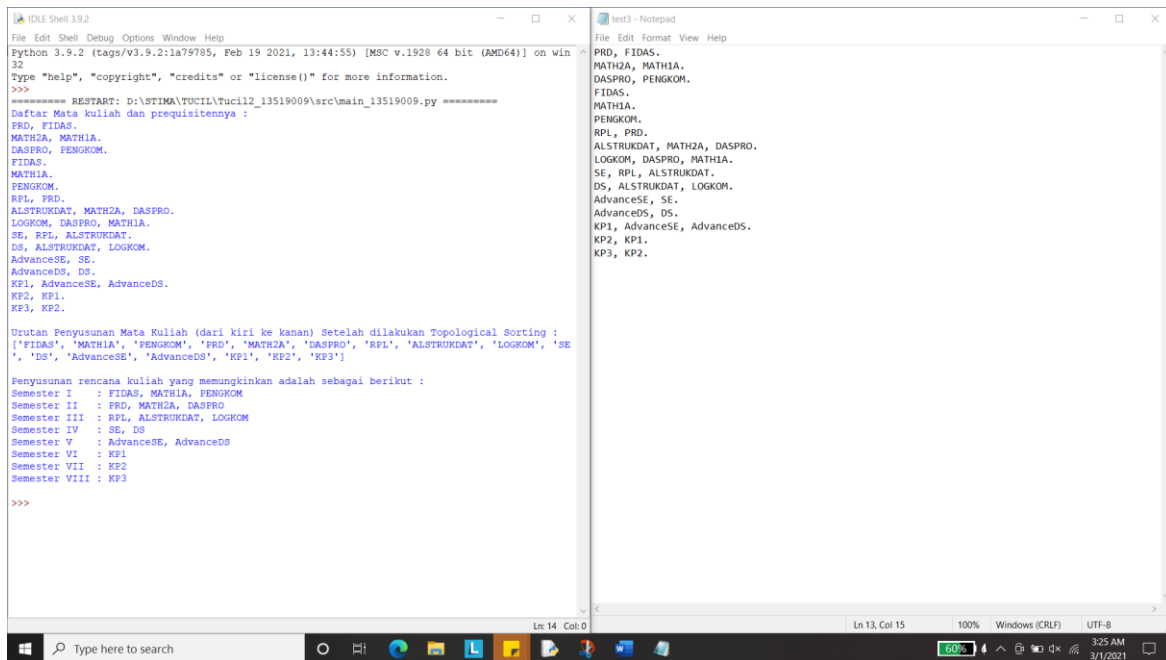
Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['IF112', 'IF113', 'IF121', 'IF114', 'IF125', 'IF127', 'IF126']

Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut :
Semester I : IF112
Semester II : IF113
Semester III : IF121, IF114
Semester IV : IF125
Semester V : IF126
Semester VI : IF127
Semester VII : IF128
>>>
```

test2 - Notepad

```
File Edit Format View Help
IF128, IF127.
IF112.
IF113, IF112.
IF121, IF113.
IF114, IF113.
IF125, IF121, IF114.
IF126, IF125.
IF127, IF126.
```

test3.txt



```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL\tucil2_13519009\src\main_13519009.py =====
Daftar Mata kuliah dan prequisitenya :
PRD, FIDAS.
MATH2A, MATH1A.
DASPRO, PENGKOM.
FIDAS.
MATH1A.
PENGKOM.
RPL, PRD.
ALSTURKDAT, MATH2A, DASPRO.
LOGKOM, DASPRO, MATH1A.
SE, RPL, ALSTURKDAT.
DS, ALSTURKDAT, LOGKOM.
AdvanceSE, SE.
AdvanceDS, DS.
KP1, AdvanceSE, AdvanceDS.
KP2, KP1.
KP3, KP2.

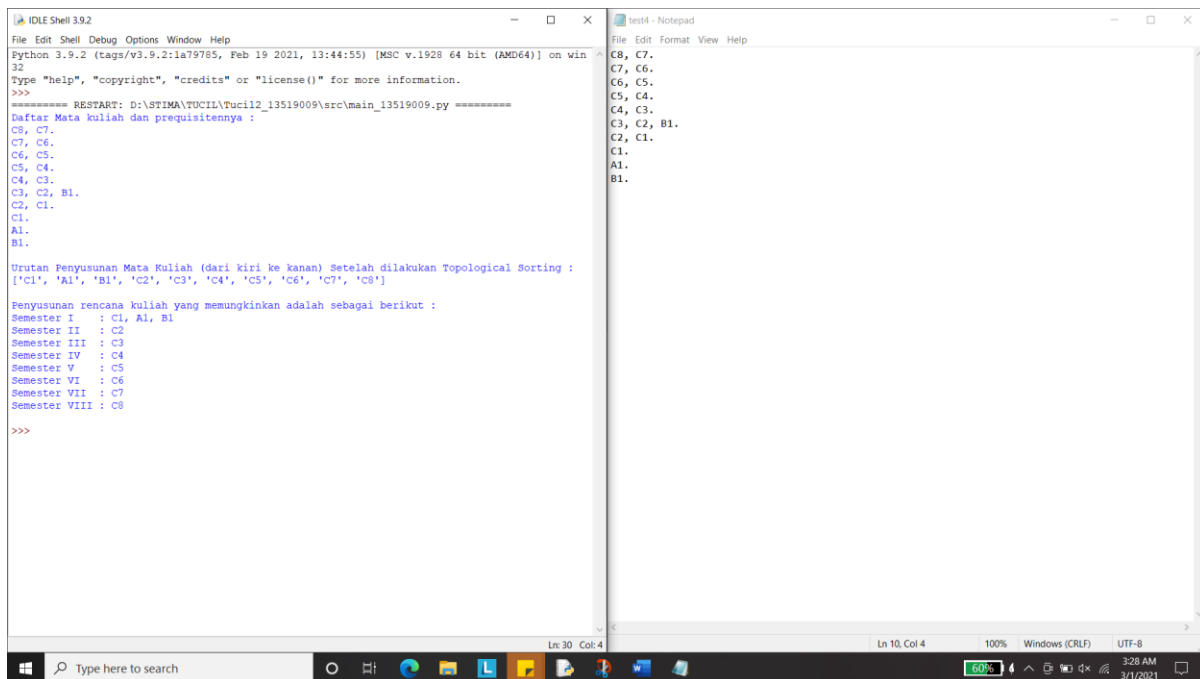
Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['FIDAS', 'MATH1A', 'PENGKOM', 'PRD', 'MATH2A', 'DASPRO', 'RPL', 'ALSTURKDAT', 'LOGKOM', 'SE',
, 'DS', 'AdvanceSE', 'AdvanceDS', 'KP1', 'KP2', 'KP3']

Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut :
Semester I   : FIDAS, MATH1A, PENGKOM
Semester II  : PRD, MATH2A, DASPRO
Semester III : RPL, ALSTURKDAT, LOGKOM
Semester IV  : SE, DS
Semester V   : AdvanceSE, AdvanceDS
Semester VI  : KP1
Semester VII : KP2
Semester VIII: KP3

>>>
```

```
PRD, FIDAS.
MATH2A, MATH1A.
DASPRO, PENGKOM.
FIDAS.
MATH1A.
PENGKOM.
RPL, PRD.
ALSTURKDAT, MATH2A, DASPRO.
LOGKOM, DASPRO, MATH1A.
SE, RPL, ALSTURKDAT.
DS, ALSTURKDAT, LOGKOM.
AdvanceSE, SE.
AdvanceDS, DS.
KP1, AdvanceSE, AdvanceDS.
KP2, KP1.
KP3, KP2.
```

Test4.txt



```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL\tucil2_13519009\src\main_13519009.py =====
Daftar Mata kuliah dan prequisitenya :
C8, C7.
C7, C6.
C6, C5.
C5, C4.
C4, C3.
C3, C2, B1.
C2, C1.
C1.
A1.
B1.

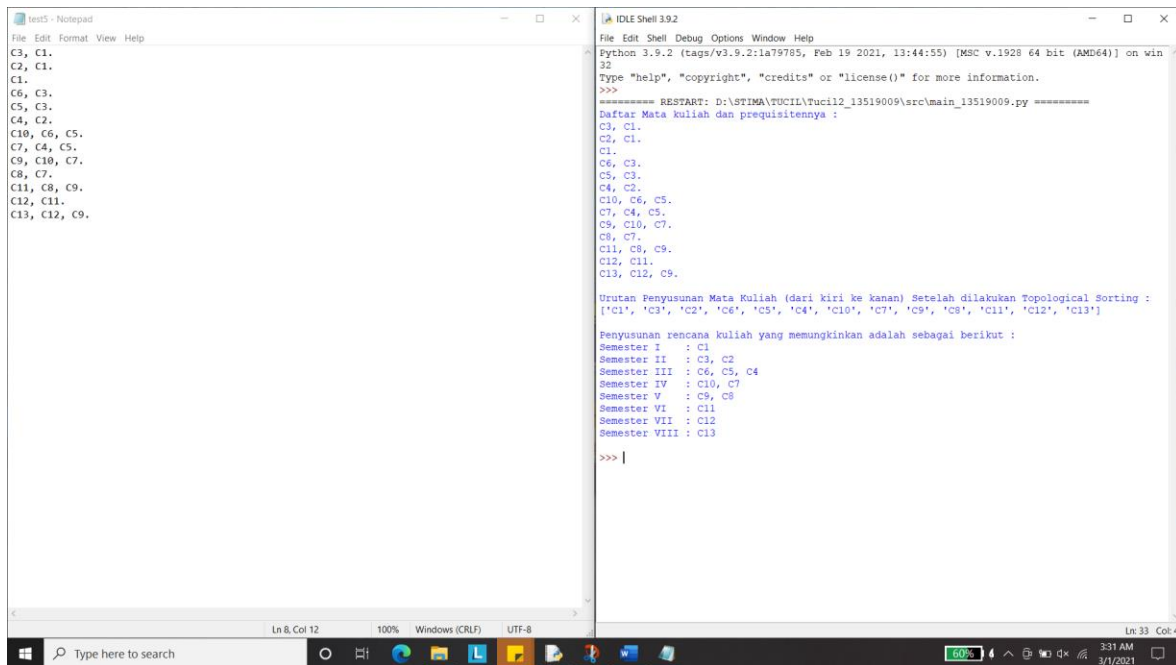
Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['C1', 'A1', 'B1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8']

Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut :
Semester I   : C1, A1, B1
Semester II  : C2
Semester III : C3
Semester IV  : C4
Semester V   : C5
Semester VI  : C6
Semester VII : C7
Semester VIII: C8

>>>
```

```
C8, C7.
C7, C6.
C6, C5.
C5, C4.
C4, C3.
C3, C2, B1.
C2, C1.
C1.
A1.
B1.
```

test5.txt



The screenshot shows a Windows desktop with two windows. The 'test5 - Notepad' window on the left contains a list of course prerequisites: C3, C1; C2, C1; C1; C6, C3; C5, C3; C4, C2; C10, C6, C5; C7, C4, C5; C9, C10, C7; C8, C7; C11, C8, C9; C12, C11; C13, C12, C9. The 'IDLE Shell 3.9.2' window on the right shows the execution of a Python script. The script's output lists the prerequisites for each course, followed by a topological sort of the courses: ['C1', 'C3', 'C2', 'C6', 'C5', 'C4', 'C10', 'C7', 'C9', 'C8', 'C11', 'C12', 'C13']. It then provides a semester plan: Semester I: C1; Semester II: C3, C2; Semester III: C6, C5, C4; Semester IV: C10, C7; Semester V: C9, C8; Semester VI: C11; Semester VII: C12; Semester VIII: C13.

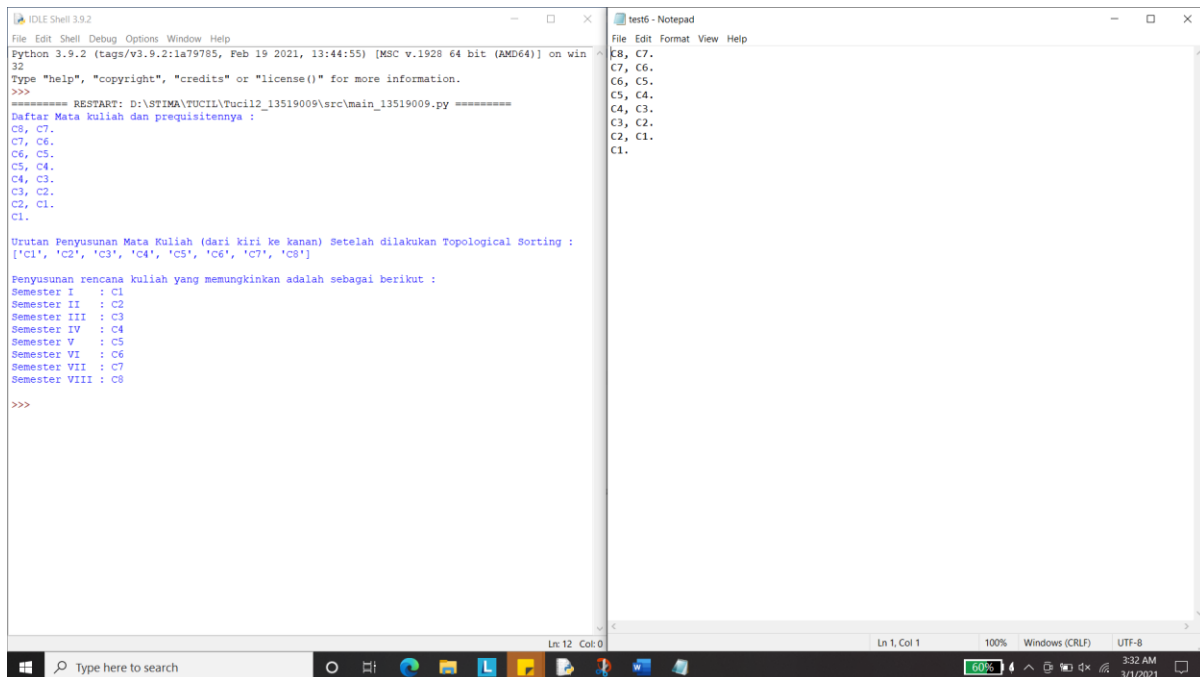
```
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL\Tucil2_13519009\src\main_13519009.py =====
Daftar Mata kuliah dan prasyaratnya :
C3, C1.
C2, C1.
C1.
C6, C3.
C5, C3.
C4, C2.
C10, C6, C5.
C7, C4, C5.
C9, C10, C7.
C8, C7.
C11, C8, C9.
C12, C11.
C13, C12, C9.

Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['C1', 'C3', 'C2', 'C6', 'C5', 'C4', 'C10', 'C7', 'C9', 'C8', 'C11', 'C12', 'C13']

Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut :
Semester I : C1
Semester II : C3, C2
Semester III : C6, C5, C4
Semester IV : C10, C7
Semester V : C9, C8
Semester VI : C11
Semester VII : C12
Semester VIII : C13

>>> |
```

test6.txt



The screenshot shows a Windows desktop with two windows. The 'IDLE Shell 3.9.2' window on the left shows the execution of a Python script. The script's output lists the prerequisites for each course, followed by a topological sort of the courses: ['C1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8']. It then provides a semester plan: Semester I: C1; Semester II: C2; Semester III: C3; Semester IV: C4; Semester V: C5; Semester VI: C6; Semester VII: C7; Semester VIII: C8. The 'test6 - Notepad' window on the right contains a list of course prerequisites: C8, C7; C7, C6; C6, C5; C5, C4; C4, C3; C3, C2; C2, C1; C1.

```
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL\Tucil2_13519009\src\main_13519009.py =====
Daftar Mata kuliah dan prasyaratnya :
C8, C7.
C7, C6.
C6, C5.
C5, C4.
C4, C3.
C3, C2.
C2, C1.
C1.

Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['C1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8']

Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut :
Semester I : C1
Semester II : C2
Semester III : C3
Semester IV : C4
Semester V : C5
Semester VI : C6
Semester VII : C7
Semester VIII : C8

>>>
```

```
File Edit Format View Help
C8, C7.
C7, C6.
C6, C5.
C5, C4.
C4, C3.
C3, C2.
C2, C1.
C1.
```

test7.txt

```
test7 - Notepad
File Edit Format View Help
IL2, IL1.
IL3, IL1.
IL4, IL2, IL3.
IL6, IL4.
IL9, IL6.
IL10, IL6.
IL11, IL9, IL10.
IL12, IL11.
IL13, IL12.
IL1.

IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL\Tucil2_13519009\src\main_13519009.py =====
Daftar Mata Kuliah dan prekusitennya :
IL2, IL1.
IL3, IL1.
IL4, IL2, IL3.
IL6, IL4.
IL9, IL6.
IL10, IL6.
IL11, IL9, IL10.
IL12, IL11.
IL13, IL12.
IL1.

Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['IL1', 'IL2', 'IL3', 'IL4', 'IL6', 'IL9', 'IL10', 'IL11', 'IL12', 'IL13']

Penyusunan rencana Kuliah yang memungkinkan adalah sebagai berikut :
Semester I : IL1
Semester II : IL2, IL3
Semester III : IL4
Semester IV : IL6
Semester V : IL9, IL10
Semester VI : IL11
Semester VII : IL12
Semester VIII : IL13

>>>
```

test8.txt

```
test8 - Notepad
File Edit Format View Help
IL2, IL1.
IL3, IL1.
IL4, IL2, IL3.
IL6, IL4.
IL9, IL6.
IL10, IL6.
IL11, IL9, IL10.
IL12, IL11.
IL13, IL12.
IL1.
IL100.
IL101.
IL102.

IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\STIMA\TUCIL\Tucil2_13519009\src\main_13519009.py =====
Daftar Mata kuliah dan prekusitennya :
IL2, IL1.
IL3, IL1.
IL4, IL2, IL3.
IL6, IL4.
IL9, IL6.
IL10, IL6.
IL11, IL9, IL10.
IL12, IL11.
IL13, IL12.
IL1.
IL100.
IL101.
IL102.

Urutan Penyusunan Mata Kuliah (dari kiri ke kanan) Setelah dilakukan Topological Sorting :
['IL1', 'IL100', 'IL101', 'IL102', 'IL2', 'IL3', 'IL4', 'IL6', 'IL9', 'IL10', 'IL11', 'IL12',
, 'IL13']

Penyusunan rencana kuliah yang memungkinkan adalah sebagai berikut :
Semester I : IL1, IL100, IL101, IL102
Semester II : IL2, IL3
Semester III : IL4
Semester IV : IL6
Semester V : IL9, IL10
Semester VI : IL11
Semester VII : IL12
Semester VIII : IL13

>>>
```

BAB IV

Alamat tempat source code diletakkan dan Tabel CekList

Alamat tempat source code diletakkan : Tucil2_13519009/src/main.py

Tabel Checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi	v	
2. Program berhasil <i>running</i>	v	
3. Program dapat menerima berkas input dan menuliskan output	v	
4. Luaran sudah benar untuk semua kasus input.	V	

DAFTAR PUSTAKA

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Tugas-Kecil-2-\(2021\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Tugas-Kecil-2-(2021).pdf)