LAPORAN TUGAS KECIL 2 MATA KULIAH IF2211 STRATEGI ALGORITMA

Penyusunan Rencana Kuliah dengan *Topological Sort* (Penerapan *Decrease and Conquer*)

Disusun untuk Memenuhi Tugas Kecil 2 Mata Kuliah IF2211 Strategi Algoritma

Nama	NIM
Prana Gusriana	**(13519195)**

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2021

DAFTAR ISI

BAB I

DESKRIPSI MASALAH

I. Deskripsi Tugas

Pada tugas kali ini, mahasiswa diminta **membuat aplikasi sederhana** yang dapat menyusun rencana pengambilan kuliah, dengan memanfaatkan algoritma **Decrease and Conquer**. Penyusunan Rencana Kuliah diimplementasikan dengan menggunakan pendekatan *Topological Sorting*. Berikut akan dijelaskan tugas yang dikerjakan secara detail.

1. Aplikasi akan menerima daftar mata kuliah beserta prasyarat yang harus diambil seorang mahasiswa sebelum mengambil mata kuliah tersebut. Daftar mata kuliah tersebut dituliskan dalam suatu file teks dengan format:

```
<kode_kuliah_1>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode
kuliah prasyarat - 3>.
<kode_kuliah_2>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>.
<kode_kuliah_3>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode
kuliah prasyarat - 3>, <kode kuliah prasyarat - 4>.
<kode_kuliah_4>.
.
.
.
```

Gambar 1. Format File Teks untuk Masukan Daftar Kuliah

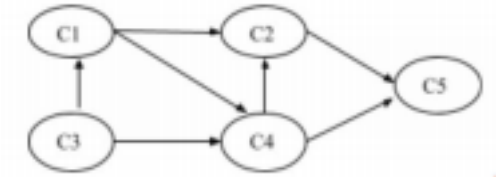
Sebuah kode_kuliah mungkin memiliki nol atau lebih prasyarat kuliah. Kode_kuliah bisa diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil di semester sebelumnya (tidak harus 1 semester sebelumnya). Asumsi semua kuliah bisa diambil di sembarang semester, baik semester ganjil maupun semester genap.

Sebagai contoh, terdapat 5 kuliah yang harus diambil seorang mahasiswa dengan daftar prerequisite dalam file teks sebagai berikut. Dari Gambar 2 terlihat bahwa kuliah C3 tidak memiliki prerequisite.

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

Gambar 2. Contoh sebuah berkas masukan Daftar Kuliah

Asumsi untuk persoalan ini, kuliah dan prerequisite nya pasti berupa Directed Acyclic Graph (DAG), dan untuk contoh pada Gambar 2, dapat dilihat representasi DAG pada gambar 3.



Gambar 3. DAG dari daftar kuliah pada Gambar 2

2. Dari file teks yang telah diterima, ditentukan kuliah apa saja yang bisa diambil di semester 1, semester 2, dan seterusnya. Sebuah kuliah tidak mungkin diambil pada semester yang sama dengan prerequisitenya. Untuk menyederhanakan persoalan, tidak ada Batasan banyaknya kuliah yang bisa diambil pada satu semester.

Dapat dilihat bahwa kasus penyusunan rencana kuliah ini sebagai salah satu implementasi *topological sorting*. Aplikasi harus dapat menyusun rencana kuliah dengan pendekatan *topological sorting* sebagai salah satu contoh penerapan *Decrease and Conquer*. Penjelasan tentang *topological sorting* dapat dibaca pada buku Levitin sub bab 4.2 dan video di YouTube:
https://www.youtube.com/watch?v=eL-KzMXSXXI

Pendekatan Topological Sorting

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.

3. Sediakan data uji sendiri, yang menjamin DAG jika diubah ke dalam representasi graf.

BAB II

PEMBAHASAN

1. **Algoritma *Decrease and Conquer***

   Decrease and conquer adalah metode perancangan algoritma dengan mereduksi persoalan menjadi dua upa-persoalan (*sub-problem*) yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja. Algoritma *decrease and conquer* terdiri dari dua tahapan yaitu *decrease* (mereduksi persoalan menjadi beberapa persoalan yang lebih kecil) dan *conquer* (memproses satu upa persoalan secara rekursif). Dalam *decrease and conquer* tidak terdapat tahap combine karena hanya satu upa-persoalan yang diselesaikan.

   Terdapat tiga varian *decrease and conquer* yaitu:

   a. *Decrease by a constant* yaitu ukuran instans persoalan direduksi sebesar konstanta yang sama setiap iterasi algoritma. Biasanya konstanta sama dengan satu. Contoh persoalan yang dapat diselesaikan dengan *decrease by a constant* yaitu perpangkatan a pangkat n, *selection sort*, dan *insertion sort*.
   b. *Decrease by a constant* factor yaitu ukuran instans persoalan direduksi sebesar faktor konstanta yang sama setiap iterasi algoritma. Biasanya faktor konstanta sama dengan dua. Contoh persoalan yang dapat diselesaikan dengan *decrease by a constant factor* adalah *binary search* dan mencari koin palsu.
   c. *Decrease by a variable size* yaitu ukuran instans persoalan direduksi bervariasi pada setiap iterasi algoritma. Contoh persoalan yang menggunakan *decrease by a variable size* adalah *interpolation search* dan mencari nilai median.

2. **Strategi Penyelesaian**

   Untuk menyelesaikan persoalan penyusunan rencana kuliah dengan *topological sort*, pendekatan yang saya lakukan adalah

   a. Membentuk *directed acyclic graph* dari file inputan dengan aturan input <kode_kuliah_1>,<kode_kuliah_prasyarat1>,<kode_kuliah_prasyarat2>. serta dipastikan input merupakan *directed acyclic graph*

   b. Setelah terbentuk DAG, cari simpul graf yang memiliki derajat masuk dengan jumlah nol lalu simpul tersebut dimasukkan kedalam solusi (mata kuliah yang dapat diambil persemesternya), lalu simpul tersebut dihapus dari graf dan juga semua

busur yang keluar dari simpul tersebut.

c. Ulangi langkah (b) sampai DAG menjadi kosong

d. Setelah itu tampilkan solusi yang diperoleh

Dalam implementasinya saya mengerjakannya dengan menggunakan bahasa pemrograman python. Saya membuat *directed acyclic graph* menjadi sebuah kelas dan akan diimplementasikan sebuah objek *directed acyclic graph* yang dibentuk berdasarkan input dari file. Objek tersebut akan diproses sehingga dapat menampilkan solusi dalam bentuk Semester <x> : <matakuliah1> <matakuliah2> <matakuliah3>. Penjelasan setiap fungsi yang saya kerjakan dapat dilihat langsung pada *source code* yang dapat diakses di github (dapat diakses mulai dari Senin, 1 Maret 2021 pukul 13.00) dan gdrive.

Algoritma *topological search* yang saya kerjakan ini berkaitan dengan *decrease and conquer*. Algoritma tersebut merupakan *decrease by a constant* dengan konstanta 1. Hal tersebut disebabkan karena ketika mencari node dengan derajat masuk nol, DAG dibagi menjadi dua upa persoalan, upa persoalan yang pertama hanya satu elemen simpul dan upa persoalan kedua berukuran n-1 elemen simpul. Upa-persoalan dengan satu elemen akan dicek apakah derajat masuknya nol atau bukan, jika nol maka simpul tersebut akan dimasukkan kedalam array solusi dan simpul tersebut akan dihapus beserta busur yang keluar dari simpul tersebut. Sedangkan upa-persoalan dengan n-1 elemen akan diproses secara rekursif. Proses pencarian simpul dengan derajat masuk nol tersebut akan terus dilakukan sampai DAG menjadi kosong.

Kompleksitas waktu untuk menemukan simpul dengan derajat masuk nol lalu menghapusnya adalah $T(n) = 0$ untuk $n = 0$ dan $T(n) = T(n-1) + 1 = O(n)$. n tersebut merupakan jumlah simpul (Vertex) dari DAG sehingga bisa dikatakan $O(n) = O(V)$. Karena pencarian tersebut dilakukan sampai DAG menjadi kosong dan pada setiap iterasi jumlah elemen DAG akan terus berkurang maka kompleksitas waktu totalnya adalah $O(V+E)$ dengan V adalah banyak simpul (*vertex*) dan E adalah banyak sisi (*edge*).

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

BAB III

IMPLEMENTASI PROGRAM

Dalam pengerjaan tugas kecil 2 untuk mempersiapkan rencana kuliah dengan *topological sort* ini saya menggunakan bahaya python dan saya mendekomposisi persoalannya menjadi tiga bagian, yaitu untuk memproses file (persoalan membaca input file atau menyimpan ke file eksternal), kelas *directed acyclic graph*, dan main program. Semua source code terdapat dalam satu folder yang sama yaitu folder src. *Source code* untuk memproses file yaitu FileProcesses.py, *source code* untuk kelas directed acyclic graph adalah DirectedAcyclicGraph.py, dan *source code* untuk main programnya yaitu topologikeun.py. Program yang saya buat memiliki nama "Topologikeun". *Source code*, hasil kompilasi, dan juga dokumentasi dapat diakses di github (dapat diakses mulai dari Senin, 1 Maret 2021 pukul 13.00) dan gdrive berikut ini. Berikut saya tuliskan *source code* yang sudah saya kerjakan.

1. FileProcesses.py

```
"""
    FUNGSI readInputFromFile menerima parameter namaFile berupa string yang me
rupakan nama dari file yang akan dibaca
    Filenya terdapat pada folder test
    Input file:
        <kode_kuliah_1>,<kode_kuliah_prasyarat_1>,<kode_kuliah_prasyarat2>.
        <kode_kuliah_2>,<kode_kuliah_prasyarat_1>.
        <kode_kuliah_3>.
    Keluaran dari fungsi ini adalah sebuah array
"""
def readInputFromFile(namaFile):
    file = open("../test/"+namaFile, "r")
    file_input = file.readlines()
    i = 0
    while(i < len(file_input)):
        j = 0
        strRet = ""
        tempstr = file_input[i]
        while (j < len(tempstr)):
            if(tempstr[j] != "\n"):
                strRet += tempstr[j]
            j += 1
        file_input[i] = strRet
        i += 1
    file.close()
    return file_input
```

```
"""
    Fungsi saveFile: untuk menyimpan solusi pada file eksternal dengan nama fi
le yaitu namaFile dan isi dari filenya adalah file_output
"""
def saveFile(file_output, namaFile):
    file = open("../test/"+namaFile, "w")
    file.write(file_output)
    file.close()
```

2. DirectedAcyclicGraph.py

```
import FileProcesses

"""
CLASS DirectedAcyclicGraph : digunakan untuk membuat objek DAG yang direpresen
tasikan dengan adjacency list dengan bantuan type dictionary
-> Mempunyai atribut:
    - DAG : adjacency list dengan menggunakan dictionary
    - course : array untuk menyimpan course apa saja yang tersedia
    - semester : array untuk menyimpan solusi
    - visited: array untuk menyimpan informasi apakah suatu node telah dikunju
ngi atau belum
-> Mempunyai method:
    - __init__ : konstruktor untuk membentuk objek
    - buildDAG : membuat directed acyclic graph dari input file
    - delNode : untuk menghapus node dan semua sisi yang keluar dari node ters
ebut
    - topologiin : untuk mengecek graf yang memiliki derajat masuk = 0 lalu me
masukkan ke solusi dan simpul serta sisi yang keluar dari simpul tersebut diha
pus dengan delNode
    - topologikeun : untuk mencari solusinya sampai graf DAG kosong
    - printInput : untuk menampilkan persoalan apa yang akan diselesaikan
    - printSolusi : untuk menampilkan solusi yaitu mata kuliah yang dapat diam
bil setiap semesternya
-> CATATAN:
    - Input file diasumsikan merupakan DAG (Directed Acyclic Graph) dikarenaka
n tidak ada penanganan kasus jika inputnya bukan DAG
    - Input file mengikuti aturan:
        <kode_kuliah_1>,<kode_kuliah_prasyarat_1>,<kode_kuliah_prasyarat2>.
        <kode_kuliah_2>,<kode_kuliah_prasyarat_1>.
        <kode_kuliah_3>.
      Jika tidak maka akan terjadi error
      Semua matkul harus ditulis meskipun tidak memiliki prerequisite untuk me
nghindari error.
"""

class DirectedAcyclicGraph:
    def __init__(self):
```

```
        self.DAG = {}
        self.course = []
        self.semester = [[] for _ in range(8)]
        self.visited = []

    # memiliki parameter berupa string inputFile yang merupakan nama file inpu
t yang berada pada folder test
    def buildDAG(self, inputFile):
        fileInput = FileProcesses.readInputFromFile(inputFile)
        i = 0
        while (i < len(fileInput)):
            j = 0
            course = []
            prereq = []
            koma = 0
            strret = ""
            # Memproses input file yang berbentuk <kode_kuliah_1>,<kode_kuliah
_prasyarat_1>,<kode_kuliah_prasyarat2>.
            while (j < len(fileInput[i])):
                if (fileInput[i][j] != '.'):
                    if (fileInput[i][j] != ','):
                        strret += fileInput[i][j]
                    else:
                        if (koma > 0):
                            prereq += [strret]
                        else:
                            course += [strret]
                        koma += 1
                        strret = ""
                else:
                    if (koma > 0):
                        prereq += [strret]
                    else:
                        course += [strret]
                j += 1
            # Membuat simpul course dan sisi yang masuk dari prerequisite ke s
impul course
            self.DAG[course[0]]=prereq
            i += 1
        for i in self.DAG: # Penambahan semua course yang ada kedalam atribut
course
            self.course += [i]

    # Untuk menampilkan DAG, course, dan semester, untuk keperluan debug
    def printGraf(self):
        print(self.DAG, "DAG")
        print(self.course, "COURSE")
        print(self.semester, "SEMESTER")
```

```
    # Menerima input parameter berupa node yang akan dihapus
    def delNode(self, node):
        del self.DAG[node] # Hapus simpul
        # Hapus sisi yang keluar dari simpul yang dihapus
        for i in self.DAG:
            temparr = []
            for j in self.DAG[i]:
                if j != node:
                    temparr += [j]
            self.DAG[i] = temparr

    # Mencari solusi (mencari simpul yang derajat masuknya 0 lalu memasukkanny
a kedalam array semester dengan indeks i)
    # parameter n merupakan indeks dari array course
    # akan terus rekursif sampai n >= banyak course
    def topologiin(self, n, i):
        if (n < len(self.course)):
            if (self.visited[n] == False):
                if (len(self.DAG[self.course[n]]) == 0):
                    self.semester[i] += [self.course[n]]
                    self.visited[n] = True
                    self.topologiin(n+1, i)
                    self.delNode(self.course[n])
                else:
                    self.topologiin(n+1, i)
            else:
                self.topologiin(n+1, i)

    # Mencari solusi (sampai graf DAG kosong)
    def topologikeun(self):
        i = 0
        n = 0
        self.visited = [False for i in range(len(self.course))]
        while (len(self.DAG) != 0):
            self.topologiin(n, i)
            i += 1

    # menampilkan persoalan dari yang berbentuk <kode_kuliah_1>,<kode_kuliah_p
rasyarat_1>,<kode_kuliah_prasyarat2>.
    # menjadi berbentuk <kode_kuliah_1> <- <kode_kuliah_prasyarat_1> <- <kode_
kuliah_prasyarat2>
    def printInput(self, file_output):
        strout = ""
        for i in self.DAG:
            stret = ""
            stret += str(i)
            for j in self.DAG[i]:
```

```
                stret += " <- " + j
            strout += stret + "\n"
            print(stret)
        file_output[0] += strout

    # Menampilkan solusi yang berupa Semester x : matkul1 matkul2 matkul3
    def printSolusi(self, file_output):
        s = 1
        strout = ""
        for i in self.semester:
            strs = ""
            if (len(i) > 0):
                strs += "Semester " + str(s) + " : "
                for j in i:
                    strs += str(j) + " "
                print(strs)
                strout += strs + "\n"
                s += 1
        file_output[0] += strout
```

3. topologikeun.py

```
import DirectedAcyclicGraph
import FileProcesses
import os.path

# PRINT LOGO DARI APLIKASI TOPOLOGIKEUN

print("d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db
   dD d88888b db    db d8b   db")
print("`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88
 ,8P' 88'     88    88 888o  88")
print("   88    88    88 88oodD' 88    88 88      88    88 88         88    88
,8P   88ooooo 88    88 88V8o 88")
print("   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88
`8b   88~~~~~ 88    88 88 V8o88")
print("   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88
 `88. 88.     88b  d88 88  V888")
print("   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP
   YD Y88888P ~Y8888P' VP   V8P")
print("="*102)
print(" "*9, "Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGI
KEUN aja !!!")
print("="*102)
print("")

# INPUT NAMA FILE, JIKA TIDAK TERDAPAT FILE DENGAN NAMA YANG DIINPUTKAN MAKA A
KAN DIULANG SAMPAI MENEMUKAN YANG SESUAI
```

```
N = input("Masukkan nama file: ")
while (not(os.path.isfile("../test/"+N))):
    print("File tidak ditemukan silahkan masukkan kembali input nama file,")
    N = input("Masukkan nama file: ")

file_output = [""] # INISIALISASI UNTUK KEPERLUAN SAVE FILE (ISI DARI FILE YAN
G AKAN DISIMPAN)

# INISIALISASI OBJEK DAG
DAG = DirectedAcyclicGraph.DirectedAcyclicGraph()

# MEMBENTUK DAG BERDASARKAN FILE INPUTAN YAITU N
DAG.buildDAG(N)
print("="*30)
print("          Persoalan: ")
print("="*30)
file_output[0] += "="*30 + "\n" + "          Persoalan: \n" + "="*30 + "\n"

# MENAMPILKAN PERSOALAN
DAG.printInput(file_output)

# MENYELESAIKAN PERSOALAN DENGAN MENGGUNAKAN TOPOLOGY SORT
DAG.topologikeun()
print("="*30)
print("           Solusi: ")
print("="*30)
file_output[0] += "="*30 + "\n" + "           Solusi: \n" + "="*30 + "\n"

# MENAMPILKAN SOLUSI DARI PERSOALAN (MATA KULIAH YANG DAPAT DIAMBIL SETIAP SEM
ESTERNYA)
DAG.printSolusi(file_output)
file_output[0] += "\nBy: Prana Gusriana, 13519195, IF2211 Strategi Algoritma 2
021 (TUCIL 2)\n"

# UNTUK MENANYAKAN APAKAH SOLUSINYA AKAN DISIMPAN DI FILE EKSTERNAL
print("="*55)
P = input("Apakah solusi akan disimpan di file eksternal? (Y/N) ")
while (not(P!="Y" or P!="N")):
    print("Maaf hanya menerima input (Y/N), ")
    P = input("Apakah solusi akan disimpan di file eksternal? (Y/N) ")

if (P == "Y"):
    O = input("Masukkan nama file: ")
    while (os.path.isfile("../test/"+O)): # HANYA DAPAT MENYIMPAN KE FILE EKST
ERNAL DENGAN NAMA YANG BELUM ADA
        print("Nama file telah digunakan silahkan masukkan kembali input nama
file,")
        O = input("Masukkan nama file: ")
```

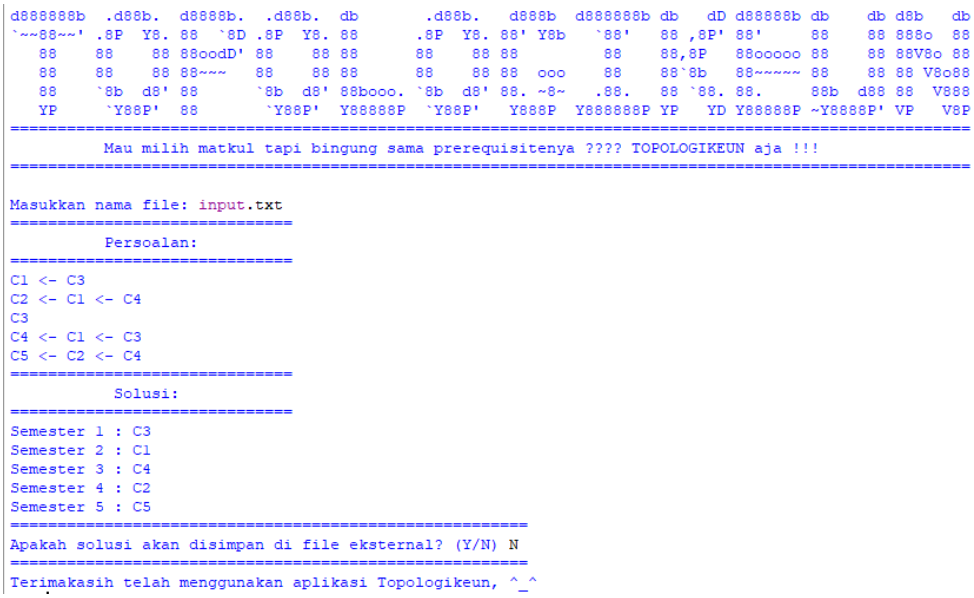
```
    FileProcesses.saveFile(file_output[0], 0) # MENYIMPAN SOLUSI PADA FILE EKS
TERNAL
print("="*55)
print("Terimakasih telah menggunakan aplikasi Topologikeun, ^_^")
```

BAB IV

EKSPERIMEN

Untuk gambar atau hasil kompilasi dapat dilihat difolder gambar dan doc, pada file submisi, github (dapat diakses mulai dari Senin, 1 Maret 2021 pukul 13.00), gdrive

1. Hasil kompilasi dengan input.txt: (input pada folder test dengan nama input.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)



2. Hasil kompilasi dengan input2.txt: (input pada folder test dengan nama input2.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

3. Hasil kompilasi dengan input3.txt: (input pada folder test dengan nama input3.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

```
d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db   dD d88888b db    db d8b   db
`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88 ,8P' 88'     88    88 888o  88
   88    88    88 88oodD' 88    88 88      88    88 88         88    88,8P   88ooooo 88    88 88V8o 88
   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88`8b   88~~~~~ 88    88 88 V8o88
   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88 `88. 88.     88b  d88 88  V888
   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP   YD Y88888P ~Y8888P' VP   V8P
================================================================================================================
         Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGIKEUN aja !!!
================================================================================================================

Masukkan nama file: input3.txt
==============================
         Persoalan:
==============================
MATDIS <- MAT1A <- MAT2A
ORKOM <- DASPRO
ALSTRUKDAT <- DASPRO
PENGKOM
DASPRO <- PENGKOM
MAT1A
MAT2A <- MAT1A
==============================
          Solusi:
==============================
Semester 1 : PENGKOM MAT1A
Semester 2 : DASPRO MAT2A
Semester 3 : MATDIS ORKOM ALSTRUKDAT
=========================================================
Apakah solusi akan disimpan di file eksternal? (Y/N) Y
Masukkan nama file: output3.txt
=========================================================
Terimakasih telah menggunakan aplikasi Topologikeun, ^_^
```

4. Hasil kompilasi dengan input4.txt: (input pada folder test dengan nama input4.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

```
d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db   dD d88888b db    db d8b   db
`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88 ,8P' 88'     88    88 888o  88
   88    88    88 88oodD' 88    88 88      88    88 88         88    88,8P   88ooooo 88    88 88V8o 88
   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88`8b   88~~~~~ 88    88 88 V8o88
   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88 `88. 88.     88b  d88 88  V888
   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP   YD Y88888P ~Y8888P' VP   V8P
================================================================================================================
         Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGIKEUN aja !!!
================================================================================================================

Masukkan nama file: input4.txt
==============================
         Persoalan:
==============================
MATDIS <- MAT1A <- MAT2A
ORKOM <- DASPRO
ALSTRUKDAT <- DASPRO
PENGKOM
DASPRO <- PENGKOM
MAT1A
MAT2A <- MAT1A
OS <- ORKOM
STIMA <- ALSTRUKDAT
PBO <- ALSTRUKDAT <- DASPRO
==============================
          Solusi:
==============================
Semester 1 : PENGKOM MAT1A
Semester 2 : DASPRO MAT2A
Semester 3 : MATDIS ORKOM ALSTRUKDAT
Semester 4 : OS STIMA PBO
=========================================================
Apakah solusi akan disimpan di file eksternal? (Y/N) Y
Masukkan nama file: input4.txt
Nama file telah digunakan silahkan masukkan kembali input nama file,
Masukkan nama file: output4.txt
=========================================================
Terimakasih telah menggunakan aplikasi Topologikeun, ^_^
```

5. Hasil kompilasi dengan input5.txt: (input pada folder test dengan nama input5.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

```
d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db   dD d88888b db    db d8b   db
`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88 ,8P' 88'     88    88 888o  88
   88    88    88 88oodD' 88    88 88      88    88 88         88    88,8P   88ooooo 88    88 88V8o 88
   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88`8b   88~~~~~ 88    88 88 V8o88
   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88 `88. 88.     88b  d88 88  V888
   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP   YD Y88888P ~Y8888P' VP   V8P
================================================================================================================
          Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGIKEUN aja !!!
================================================================================================================

Masukkan nama file: input5.txt
==============================
          Persoalan:
==============================
MAT1A
FIS1A
PENGKOM
TTKI
BING
OR
==============================
           Solusi:
==============================
Semester 1 : MAT1A FIS1A PENGKOM TTKI BING OR
======================================================
Apakah solusi akan disimpan di file eksternal? (Y/N) Y
Masukkan nama file: output5.txt
======================================================
Terimakasih telah menggunakan aplikasi Topologikeun, ^_^
```

6. Hasil kompilasi dengan input6.txt: (input pada folder test dengan nama input6.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

```
d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db   dD d88888b db    db d8b   db
`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88 ,8P' 88'     88    88 888o  88
   88    88    88 88oodD' 88    88 88      88    88 88         88    88,8P   88ooooo 88    88 88V8o 88
   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88`8b   88~~~~~ 88    88 88 V8o88
   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88 `88. 88.     88b  d88 88  V888
   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP   YD Y88888P ~Y8888P' VP   V8P
================================================================================================================
          Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGIKEUN aja !!!
================================================================================================================

Masukkan nama file: input6.txt
==============================
          Persoalan:
==============================
PROBSTAT <- MAT1A <- MAT2A <- MATDIS
STIMA <- ALSTRUKDAT <- MATDIS
TBFO <- DASPRO
LOGKOM <- DASPRO <- PENGKOM
PENGKOM
AI <- LOGKOM <- TBFO <- PROBSTAT <- STIMA
DASPRO <- PENGKOM
MATDIS <- MAT1A <- MAT2A
MAT1A
MAT2A <- MAT1A
ALSTRUKDAT <- DASPRO
==============================
           Solusi:
==============================
Semester 1 : PENGKOM MAT1A
Semester 2 : DASPRO MAT2A
Semester 3 : TBFO LOGKOM MATDIS ALSTRUKDAT
Semester 4 : PROBSTAT STIMA
Semester 5 : AI
======================================================
Apakah solusi akan disimpan di file eksternal? (Y/N) Y
Masukkan nama file: output6.txt
======================================================
Terimakasih telah menggunakan aplikasi Topologikeun, ^_^
```

7. Hasil kompilasi dengan input7.txt: (input pada folder test dengan nama input7.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

```
d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db   dD d88888b db    db d8b   db
`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88 ,8P' 88'     88    88 888o  88
   88    88    88 88oodD' 88    88 88      88    88 88         88    88,8P   88ooooo 88    88 88V8o 88
   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88`8b   88~~~~~ 88    88 88 V8o88
   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88 `88. 88.     88b  d88 88  V888
   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP   YD Y88888P ~Y8888P' VP   V8P
================================================================================================================
          Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGIKEUN aja !!!
================================================================================================================

Masukkan nama file: input7.txt
==============================
          Persoalan:
==============================
FIS1A
FIS2A <- FIS1A
MAT1A
MAT2A <- MAT1A
PENGKOM
DASPRO <- PENGKOM
TBFO <- DASPRO
MATDIS <- MAT1A <- MAT2A
LOGKOM <- DASPRO <- PENGKOM
ALGEO <- MAT1A <- MAT2A
ALSTRUKDAT <- DASPRO
ORKOM <- DASPRO
PROBSTAT <- MAT1A <- MAT2A <- MATDIS
STIMA <- ALSTRUKDAT <- MATDIS
PBO <- ALSTRUKDAT
AI <- LOGKOM <- TBFO <- PROBSTAT <- STIMA
WEB <- PBO <- ALSTRUKDAT
OS <- ORKOM <- ALSTRUKDAT
JARKOM <- OS
BASDAT <- LOGKOM
SI <- BASDAT
GRAFIKAKOM <- ORKOM <- ALSTRUKDAT <- ALGEO
ML <- AI <- ALSTRUKDAT
==============================
           Solusi:
==============================
Semester 1 : FIS1A MAT1A PENGKOM
Semester 2 : FIS2A MAT2A DASPRO
```

```
Semester 3 : TBFO MATDIS LOGKOM ALGEO ALSTRUKDAT ORKOM
Semester 4 : PROBSTAT STIMA PBO OS BASDAT GRAFIKAKOM
Semester 5 : AI WEB JARKOM SI
Semester 6 : ML
=======================================================
Apakah solusi akan disimpan di file eksternal? (Y/N) Y
Masukkan nama file: output7.txt
=======================================================
Terimakasih telah menggunakan aplikasi Topologikeun, ^_^
```

8. Hasil kompilasi dengan input8.txt: (input pada folder test dengan nama input8.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

```
d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db   dD d88888b db    db d8b   db
`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88 ,8P' 88'     88    88 888o  88
   88    88    88 88oodD' 88    88 88      88    88 88         88    88,8P   88ooooo 88    88 88V8o 88
   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88`8b   88~~~~~ 88    88 88 V8o88
   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88 `88. 88.     88b  d88 88  V888
   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP   YD Y88888P ~Y8888P' VP   V8P
================================================================================================================
         Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGIKEUN aja !!!
================================================================================================================

Masukkan nama file: input
File tidak ditemukan silahkan masukkan kembali input nama file,
Masukkan nama file: input8.txt
==============================
         Persoalan:
==============================
FIS1A
FIS2A <- FIS1A
MAT1A
TTKI
BING
OR
MAT2A <- MAT1A
PENGKOM
DASPRO <- PENGKOM
TBFO <- DASPRO
MATDIS <- MAT1A <- MAT2A
LOGKOM <- DASPRO <- PENGKOM
ALGEO <- MAT1A <- MAT2A
ALSTRUKDAT <- DASPRO
ORKOM <- DASPRO
PROBSTAT <- MAT1A <- MAT2A <- MATDIS
STIMA <- ALSTRUKDAT <- MATDIS
PBO <- ALSTRUKDAT
AI <- LOGKOM <- TBFO <- PROBSTAT <- STIMA
WEB <- PBO <- ALSTRUKDAT
OS <- ORKOM <- ALSTRUKDAT
JARKOM <- OS
BASDAT <- LOGKOM
SI <- BASDAT
GRAFIKAKOM <- ORKOM <- ALSTRUKDAT <- ALGEO
ML <- AI <- ALSTRUKDAT
```



```
SISTER <- ORKOM <- JARKOM
TA1 <- TBFO <- MATDIS <- LOGKOM <- ALGEO <- ALSTRUKDAT <- ORKOM <- PROBSTAT <- STIMA <- PBO <- AI <- WEB <- OS <- JARKOM <- BASDAT <- SI <- GRAFIKAKOM <- ML <- SISTER
TA2 <- TA1
==============================
          Solusi:
==============================
Semester 1 : FIS1A MAT1A TTKI BING OR PENGKOM
Semester 2 : FIS2A MAT2A DASPRO
Semester 3 : TBFO MATDIS LOGKOM ALGEO ALSTRUKDAT ORKOM
Semester 4 : PROBSTAT STIMA PBO OS BASDAT GRAFIKAKOM
Semester 5 : AI WEB JARKOM SI
Semester 6 : ML SISTER
Semester 7 : TA1
Semester 8 : TA2
========================================================
Apakah solusi akan disimpan di file eksternal? (Y/N) Y
Masukkan nama file: output8.txt
========================================================
Terimakasih telah menggunakan aplikasi Topologikeun, ^_^
```


9. Hasil kompilasi dengan input9.txt: (input pada folder test dengan nama input9.txt) (inputan dapat dilihat pada screen shoot di bagian Persoalan:)

```
d888888b  .d88b.  d8888b.  .d88b.  db       .d88b.   d888b  d888888b db   dD d88888b db    db d8b   db
`~~88~~' .8P  Y8. 88  `8D .8P  Y8. 88      .8P  Y8. 88' Y8b   `88'   88 ,8P' 88'     88    88 888o  88
   88    88    88 88oodD' 88    88 88      88    88 88         88    88,8P   88ooooo 88    88 88V8o 88
   88    88    88 88~~~   88    88 88      88    88 88  ooo    88    88`8b   88~~~~~ 88    88 88 V8o88
   88    `8b  d8' 88      `8b  d8' 88booo. `8b  d8' 88. ~8~   .88.   88 `88. 88.     88b  d88 88  V888
   YP     `Y88P'  88       `Y88P'  Y88888P  `Y88P'   Y888P  Y888888P YP   YD Y88888P ~Y8888P' VP   V8P
================================================================================================================
         Mau milih matkul tapi bingung sama prerequisitenya ???? TOPOLOGIKEUN aja !!!
================================================================================================================

Masukkan nama file: input9.txt
==============================
         Persoalan:
==============================
PENGKOM
==============================
          Solusi:
==============================
Semester 1 : PENGKOM
========================================================
Apakah solusi akan disimpan di file eksternal? (Y/N) Y
Masukkan nama file: output9.txt
========================================================
Terimakasih telah menggunakan aplikasi Topologikeun, ^_^
```

BAB V
SARAN, KESIMPULAN, DAN REFLEKSI

1. Kesimpulan

Tugas kecil 2 mata kuliah IF2211 Strategi Algoritma ini sudah bisa diselesaikan sesuai spesifikasi.

2. Saran

Semoga tugas-tugas selanjutnya bisa dikerjakan dengan lebih baik lagi.

3. Refleksi

Awalnya bingung gimana nerapin algoritma decrease and conquernya soalnya referensi youtube yang terdapat pada spesifikasi itu kalau tidak salah menggunakan DFS. Setelah membaca kembali spesifikasinya lalu dicoba dan ternyata berhasil. Semoga tugas selanjutnya dapat dikerjakan dengan lebih baik lagi.