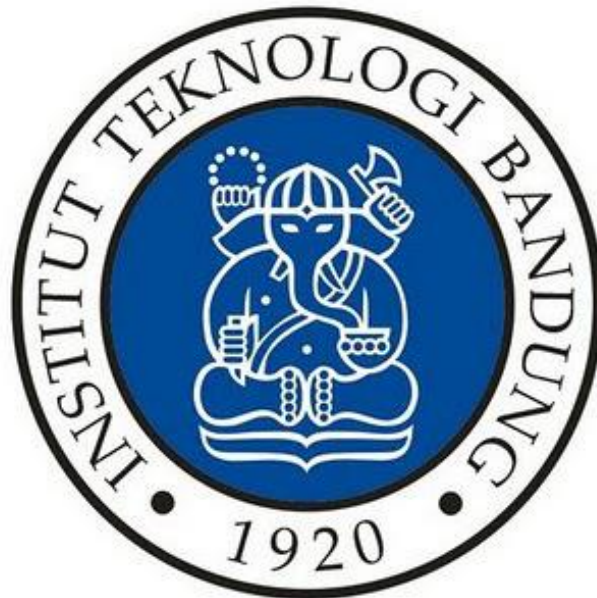


## **Tugas Kecil 2 IF2211 Strategi Algoritma**

*Penyusunan Rencana Kuliah dengan Topological Sort*

**Allief Nuriman**  
**13519221**



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

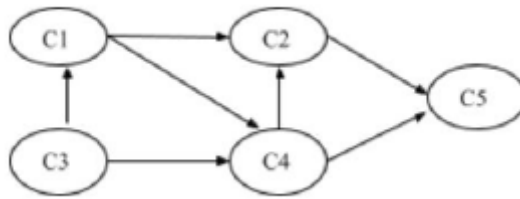
## Daftar Isi

A. Algoritma Topological Sort.....	3
B. Source Code Program.....	3
C. Contoh Masukan dan Keluaran .....	6
D. Checklist.....	7
E. Reposit.....	7

## A. Algoritma Topological Sort

Pendekatan Topological Sorting adalah sebagai berikut:

Misal kita menerima suatu graf berbentuk seperti di bawah:



Gambar A.1

1. Dari graf tersebut, kita hitung semua (*in-degree*) setiap simpul. Yaitu banyaknya busur yang masuk pada simpul tersebut.
2. Pilih sembarang simpul yang memiliki *in-degree* 0. Misalnya C3
3. Ambil simpul tersebut & hilangkan beserta semua busur yang keluar dari simpul tersebut pada graf, kurangi derajat simpul yang berhubungan dengan simpul tersebut sebanyak 1
4. Ulangi langkah ke-2 dan ke-3 sampai tidak ada lagi yang tersisa

Kaitan hal ini dengan Decrease and Conquer adalah, ide dari algoritma ini ialah mereduksi persoalan hingga kita dapat memecahkannya lebih mudah.

## B. Source Code Program

13519221.py

```
# File: 13519221.py
# 13519221 Allief Nuriman
# IF2211 2020-2
# Main program, jalankan ini

import graph_13519221 as dag
file = open("matkul.txt", 'r')
cc = file.read().splitlines()
m = dag.readfile(cc) # Baca file lalu masukkan ke m
dag.topSort(m)
file.close()
```

graph\_13519221.py

```
# File: graph_13519221.py
# 13519221 Allief Nuriman
# IF2211 2020-2
# Modul untuk [Pseudo] Graf

def readfile(m): # Akan membaca file lalu menyimpan di matriks dua dimensi, representasi, contoh: m[0][0] ialah matkul, m[0][i] dengan i > 0 ialah prasyarat-prasyarat matkul m[0][0]
    for i in range(len(m)): # Iterasi per line
        m[i] = m[i].replace(',', '') # Delete tanda koma, ganti kosong
        m[i] = m[i].replace('.', '') # Delete tanda titik, ganti kosong

    for i in range(len(m)):
```

```

        m[i] = m[i].split(" ") # Karena sekarang tiap baris cuma beda tan
da " ", kita pisah lagi, sehingga dia bakal jadi array dua dimensi dengan
m[0][i], i > 0 ialah prasyarat-prasyarat matkul m[0][0]

    return m

def tampilkanData(m): # Pada dasarnya ini dibuat untuk melakukan uji coba
    apakah isi file masuk ke variabel
    for i in range(len(m)):
        if (len(m[i]) == 1):
            print("Mata kuliah",end=' ')
            print(m[i][0],end=' ')
            print("tidak mempunyai prasyarat apapun")
        else:
            print("Mata kuliah",end=' ')
            print(m[i][0],end=' ')
            print("mempunyai prasyarat sebagai berikut:")
            for j in range(1,len(m[i])):
                print(m[i][j])
    print()

# Algoritma melakukan topologicalSorting, pada dasarnya
# variabel m berisi array dua dimensi
# m[i][0] untuk i >= 0 merepresentasikan mata kuliah
# dan m[i][j] untuk i >= 0 dan j > 0 merepresentasikan mata kuliah prasy
arat m[i][0]
def topSort(m):
    sem = 1
    while (len(m) != 0): # Fungsi berjalan hingga m sudah berisi 0 elemen
        print("Semester",end=' ')
        print(sem,end='')
        print(":",end=' ')
        terdelete = [] # Array ini digunakan untuk menyimpan semua mata ku
liah yang
        for z in range(len(m)): # mempunyai "busur masuk" sebanyak 0, den
gan kata lain hanya berisi m[0][0]
            if (len(m[z]) == 1):
                terdelete.append(m[z])
                print(m[z][0],end=' ')
        print()
        j = 0
        while (j < len(m)): # Dibuat untuk menghapus mata kuliah dgn busu
r masuk 0
            if (len(m[j]) == 1): # len(m[j]) == 1 mengimplikasikan hanya
berisi m[0][0]
                m.pop(j)
            else:
                j += 1

```

```

        i = 0 # Navigasi ulang array of array m, untuk menghapus matkul d
gn busur masuk 0 pd matkul yg membutuhkan m[0][0]
        while (i < len(m)): # Ini dilakukan karena m[0][0] sudah diambil
            j = 0
            while (j < len(m[i])):
                if (SyaratDiambil(m[i][j],terdelet)): # Karena terdelet a
dalam array of array dan m[i][j] adalah string, saya membuat
                    m[i].pop(j) # fungsi khusus untuk menangani ini
                    j = len(m[i])
                else:
                    j += 1
            if (not (j < len(m[i]))):
                i += 1
            terdelet = [] # Reset array of array yang menampung sekumpulan m[
i][0] untuk iterasi selanjutnya
            sem += 1

def SyaratDiambil(a,b): # Fungsi khusus untuk menangani apakah terdapat e
lemen array of array yang sama persis dengan string a
    bebas = False
    i = 0
    while (i < len(b) and not bebas):
        bebas = False
        if (len(a) == len(b[i][0])):
            j = 0
            while (j < len(a)):
                if (a[j] != b[i][0][j]):
                    bebas = False
                else:
                    if (j == len(a) - 1): # Misalkan semua char pada a da
n b sama persis
                        bebas = True # Maka kita set True
                    j += 1
            else:
                bebas = False
            i += 1
    return bebas # Lalu kirimkan

```

### C. Contoh Masukan dan Keluaran

Masukan	Keluaran
Kriptografi, Matdis. Kalkulus. TBFO, Matdis. Fisika. Stima, Matdis, Kalkulus Matdis, Kalkulus.	Semester 1: Kalkulus Fisika Semester 2: Matdis Semester 3: Kriptografi TBFO Stima
MA1201, MA1101. FI1201, FI1101. IF1210, KU1102. KU1202, KU1102. KI1002, KU1011. EL1200, FI1101. KU1102. MA1101. FI1101. KU1011.	Semester 1: KU1102 MA1101 FI1101 KU1011 Semester 2: MA1101 FI1101 IF1210 KU1102 KU1011 EL1
FI2201, FI1201. FI2202, FI1201. FI2203, FI1201. FI2204, FI2103. FI1201.	Semester 1: FI1201 Semester 2: FI1201 FI2202 FI2203 Semester 3: FI2204
Flask, Python, Pip. Pip, Python. Python, C. C.	Semester 1: C Semester 2: Python Semester 3: Pip Semester 4: Flask
SR2002, SR2001. SR2201, SR2101. SR2202, SR2102. SR2203. SR2204. SR2001. SR2101. SR2102.	Semester 1: SR2203 SR2204 SR2001 SR2101 SR2102 Semester 2: SR2001 SR2101 SR2102
C1, C3. C2, C1, C4. C3. C4, C1, C3. C5, C2, C4.	Semester 1: C3 Semester 2: C1 Semester 3: C4 Semester 4: C2 Semester 5: C5
Mouse. Ayam, Kentucky. Kentucky. Komputer, Mouse. Komputer, Ayam.	Semester 1: Mouse Kentucky Semester 2: Ayam Komputer Semester 3: Ayam

C1, C3. C2, C1, C4. C3. C6. C4, C1, C3. C5, C2, C4.	Semester 1: C3 C6 Semester 2: C1 Semester 3: C4 Semester 4: C2 Semester 5: C5
--	---

#### D. Checklist

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

#### E. Reposit

Pada tautan berikut: [allief876/course-scheduler: Tugas Kecil IF2211 Strategi Algoritma 2020-2 \(github.com\)](https://github.com/allief876/course-scheduler)