

Laporan Tugas Kecil 2
IF2211 Strategi Algoritma
Semester II tahun 2020/2021



Nama : Thomas Ferdinand Martin
NIM : 13519099
Kelas : 02
Institut Teknologi Bandung

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.	✓	

1. Algoritma Topological Sort dan kaitannya dengan pendekatan Decrease and Conquer

Topological sort :

Graf berarah adalah graf yang memiliki sisi berarah pada setiap simpulnya, baik simpul masuk maupun simpul keluar. *Directed acyclic graph* atau biasa disebut DAG adalah sebuah graf berarah dimana setiap simpulnya tidak memiliki siklus artinya tidak ada lintasan dari simpul itu untuk kembali ke simpul yang sama. Model DAG dapat dilihat pada permasalahan pengambilan mata kuliah yang memiliki prasyarat. Contohnya apabila seseorang ingin mengambil mata kuliah dengan kode IF2001 orang tersebut harus terlebih dahulu mengambil mata kuliah dengan kode IF2000 di semester sebelumnya. Apabila ada banyak mata kuliah yang ingin diambil dan tiap mata kuliah memiliki prasyarat akan cukup sulit untuk mengetahui kuliah mana dulu yang harus diambil, oleh karena itu dapat kita gunakan topological sort.

Topological sort dapat digunakan untuk menyelesaikan masalah pengambilan mata kuliah dengan melakukan pemodelan DAG pada jenis-jenis mata kuliah dan prasyaratnya. Tiap mata kuliah dilambangkan dengan simpul dan tiap kuliah prasyarat memiliki sisi keluar menuju simpul mata kuliah yang dapat diambil setelah menyelesaikan mata kuliah prasyarat. Topological sorting dilakukan dengan mengurutkan simpul-simpul yang ada sehingga untuk setiap sisi pada graf, simpul yang memiliki sisi keluar berada pada index lebih kecil dibandingkan dengan tujuan simpul masuk dari sisi keluar tadi. Salah satu pendekatan dari topological sort adalah dengan algoritma decrease and conquer.

Decrease and Conquer:

Kategori : Decrease by constant yaitu 1 karena pada tiap iterasi dilakukan pengecekan simpul, apabila simpul tidak memiliki sisi masuk, maka dihapus dari graf beserta dengan sisi keluarnya.

Algoritma :

- Tiap fase, cari tiap simpul dengan *in degree* sama dengan 0, kemudian hapus simpul itu (decrease by constant).
- Simpan simpul yang baru saja dihapus ke dalam sebuah array multidimensi dengan index array mewakili semester dimana mata kuliah tersebut diambil
- Cari sisi keluar dari simpul-simpul yang baru saja dihapus kemudian hapus sisi itu sehingga derajat *in degree* simpul tujuan sisi keluar menjadi berkurang.
- Pemanggilan rekursif dari tahap (a) hingga kondisi graf saat itu tidak memiliki simpul lagi.

2. Source code program

```
# Thomas Ferdinand Martin
# 13519099
# Tugil 2 : Decrease and Conquer

# run : python 13519099_versi2.py
# berhasil di run menggunakan python versi 3.9.0

# Definisi class beserta method dan atribut
class Graf:
    def __init__(self):
        # Graf = menyimpan list sisi yang masuk, indeks ke i pada graf
        berkorespondensi dengan simpul ke i pada simpul
        # Bentuk fisik :
        # graf = [[C2,C3], [C3], []]
        # simpul = [C1, C2, C3]
        # Menunjukkan C1 bersisian dengan C2 dan C3 dengan C2 dan C3 adalah
        sisi yang masuk ke C1. C3 tidak memiliki sisi masuk

        # List pada graf[i] = sisi yang masuk pada simpul[i]
        self.graf = []
        # List simpul-simpul (kode mata kuliah) yang ada pada graf
        self.simpul = []
        # Hasil dari topological sort
        self.result = []

    # Fungsi menambahkan simpul pada graf
    def addSimpul(self, simpul):
        self.simpul.append(simpul)
        self.graf.append([])

    # Fungsi menambahkan sisi masuk pada suatu simpul
    def addSisiMasuk(self, simpul, simpul_masuk):
        idx = self.simpul.index(simpul)
        self.graf[idx].append(simpul_masuk)

    # Fungsi menghapus suatu simpul dari graf
    def hapusSimpul(self, simpul):
        idx = self.simpul.index(simpul)
        self.simpul.remove(self.simpul[idx])

        sisi = self.graf[idx]
```

```

        self.graf.remove(sisi)

# Menghapus sisi yang menghubungkan simpul dan simpul_masuk dari graf
def hapusSisi(self, simpul, simpul_masuk):
    idx = self.simpul.index(simpul)
    self.graf[idx].remove(simpul_masuk)

# Mengembalikan true apabila graf kosong
def isEmpty(self):
    return self.simpul == []

# Untuk keperluan testing
def printGraf(self):
    print(self.graf)
    print(self.simpul)

# Fungsi-fungsi utama topologicalSort
# Fungsi menambahkan matkul yang saat itu pre requisitnya terpenuhi
atau seolah-olah tidak memiliki pre requisit ke dalam list mata kuliah per
semester
# Fungsi kemudian mengubah kondisi graf dengan menghapus mata kuliah
yang memiliki derajat masuk
def hapusZeroInDegree(self):
    # List berisi mata kuliah yang harus diambil pada semester tertentu
    current_semester = []

    # Untuk setiap data mata kuliah dengan pre requisitnya
    i = 0
    for elm in self.graf:
        # Jika saat itu mata kuliah tidak memiliki pre requisit (jumlah
        derajat masuk = 0), maka tambahkan ke dalam array mata kuliah yang bisa
        diambil dalam satu semester itu (current_semester)
        if (len(elm) == 0):
            current_semester.append(self.simpul[i])
            i+=1

    # Hapus mata kuliah yang diambil pada semester tersebut dari daftar
    mata kuliah yang masih perlu diambil (hapus simpul dari graf)
    for matkul in current_semester:
        self.hapusSimpul(matkul)

# Tambahkan data mata kuliah semester tersebut ke dalam array hasil
self.result.append(current_semester)

```

```

        return current_semester

    # Fungsi menghapus derajat masuk dari mata kuliah yang pra syaratnya
    # sudah diambil disemester tersebut atau yang ada pada array current_semester
    def hapusSimpulTetangga(self, current_semester):
        for matkul in current_semester:
            for sisa_matkul in self.graf:
                # Jika ditemukan matkul pra syarat yang sudah terambil maka
                # seolah-olah hapus sisi masuk graf
                if (sisa_matkul.count(matkul) == 1):
                    self.hapusSisi(self.simpul[self.graf.index(sisa_matkul)], matkul)

    # Topological sort dengan decrease and conquer
    def topologicalSort(self):
        # Basis ketika sudah tidak ada mata kuliah yang perlu diambil
        # (tersisa 0 simpul)
        if self.isEmpty():
            return
        else:
            # Pemanggilan fungsi penghapusan simpul derajat masuk 0
            current_semester = self.hapusZeroInDegree()
            # Pemanggilan fungsi penghapusan sisi yang bersisian dengan
            # simpul yang terhapus
            self.hapusSimpulTetangga(current_semester)
            # Pemanggilan rekursif
            self.topologicalSort()

    # Mengembalikan hasil topological sort
    def getResult(self):
        return self.result

    # Fungsi menulis hasil ke layar
    def printHasil(self):
        print("=====++++=====")
        for i in range(len(self.result)):
            print("Semester", i+1, ": ", end="")
            for matkul in self.result[i]:
                print(matkul, end=" ")
            print("\r")

    # Apabila jumlah semester yang harus diambil lebih dari 8

```

```

        if len(self.result) > 8:
            print("Sayang sekali sepertinya nanti kelulusan Anda harus
ditunda jika mau mengikuti seluruh mata kuliah ini.")
            print("====++++====")

#-----
#-----#
#-----
#-----#

# Fungsi membuka dan membaca file dan mengubahnya ke dalam bentuk graf
def readFile(namaFile):
    try:
        f = open("../test/" + namaFile, 'r')
    except:
        print("Salah memasukkan nama file")
        return

    graf_matkul = Graf()

    print("====++++====")
    for line in f:
        matkul = line.replace(" ", "").replace(".",
        "").replace("\n", "").split(",")
        simpul = matkul[0]

        print("Kode matakuliah :", simpul)
        print("Prasyarat : ", end="")

        for kode in matkul:
            if kode == simpul:
                graf_matkul.addSimpul(kode)
            else:
                graf_matkul.addSisiMasuk(simpul, kode)
                print(kode, end=" ")

        if len(matkul) == 1:
            print("-", end=" ")

        print("\r")

    return graf_matkul

```

```
# Program utama
# Input nama file
g = Graf()
x = input("Masukkan nama file [1-8].txt : ")
# Pembacaan file dan transformasi ke graf
try:
    g = readFile(x)
    g.topologicalSort()
    g.printHasil()
# Bila terjadi error
except:
    print("Pastikan input benar")
```

3. Tangkapan layar

```
C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan angka nama file [1-8].txt : 1.txt
=====++++=====
Kode mata kuliah : C1
Prasyarat : -
Kode mata kuliah : C4
Prasyarat : C1
Kode mata kuliah : C2
Prasyarat : C1
Kode mata kuliah : C3
Prasyarat : C2 C4
=====++++=====
=====++++=====
Semester 1 : C1
Semester 2 : C4 C2
Semester 3 : C3
=====++++=====

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>
```

```
C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan angka nama file [1-8].txt : 2.txt
=====++++=====
Kode mata kuliah : C1
Prasyarat : -
Kode mata kuliah : C3
Prasyarat : C1
Kode mata kuliah : C2
Prasyarat : C3
Kode mata kuliah : C4
Prasyarat : -
Kode mata kuliah : C5
Prasyarat : -
Kode mata kuliah : C6
Prasyarat : C5 C2
=====++++=====
=====++++=====
Semester 1 : C1 C4 C5
Semester 2 : C3
Semester 3 : C2
Semester 4 : C6
=====++++=====

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>
```



```

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan angka nama file [1-8].txt : 3.txt
=====++++=====
Kode mata kuliah : C1
Prasyarat : C4
Kode mata kuliah : C2
Prasyarat : C5
Kode mata kuliah : C3
Prasyarat : C1 C2
Kode mata kuliah : C4
Prasyarat : -
Kode mata kuliah : C5
Prasyarat : -
Kode mata kuliah : C0
Prasyarat : C5 C4
=====++++=====
Semester 1 : C4 C5
Semester 2 : C1 C2 C0
Semester 3 : C3
=====++++=====

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>

```

```

Command Prompt

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan angka nama file [1-8].txt : 4.txt
=====++++=====
Kode mata kuliah : C2
Prasyarat : C11
Kode mata kuliah : C3
Prasyarat : -
Kode mata kuliah : C5
Prasyarat : -
Kode mata kuliah : C7
Prasyarat : -
Kode mata kuliah : C8
Prasyarat : C7 C3
Kode mata kuliah : C9
Prasyarat : C8 C11
Kode mata kuliah : C10
Prasyarat : C11 C3
Kode mata kuliah : C11
Prasyarat : C7 C5
=====++++=====
Semester 1 : C3 C5 C7
Semester 2 : C8 C11
Semester 3 : C2 C9 C10
=====++++=====

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>

```

```

Command Prompt

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan angka nama file [1-8].txt : 5.txt
=====+++++=====
Kode mata kuliah : MK002
Prasyarat : MK011
Kode mata kuliah : MK003
Prasyarat : -
Kode mata kuliah : MK005
Prasyarat : -
Kode mata kuliah : MK007
Prasyarat : MK003
Kode mata kuliah : MK008
Prasyarat : MK007
Kode mata kuliah : MK009
Prasyarat : MK008 MK011
Kode mata kuliah : MK010
Prasyarat : MK011 MK003 MK002
Kode mata kuliah : MK011
Prasyarat : MK005 MK007
=====+++++=====
Semester 1 : MK003 MK005
Semester 2 : MK007
Semester 3 : MK008 MK011
Semester 4 : MK002 MK009
Semester 5 : MK010
=====+++++=====

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>_

```

```

Select Command Prompt

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan angka nama file [1-8].txt : 6.txt
=====+++++=====
Kode mata kuliah : IF100
Prasyarat : IF103 IF107
Kode mata kuliah : IF101
Prasyarat : IF105 IF107
Kode mata kuliah : IF102
Prasyarat : IF101
Kode mata kuliah : IF103
Prasyarat : -
Kode mata kuliah : IF104
Prasyarat : IF101 IF103
Kode mata kuliah : IF105
Prasyarat : -
Kode mata kuliah : IF106
Prasyarat : IF101 IF100
Kode mata kuliah : IF107
Prasyarat : -
=====+++++=====
Semester 1 : IF103 IF105 IF107
Semester 2 : IF100 IF101
Semester 3 : IF102 IF104 IF106
=====+++++=====

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>_

```

```
Command Prompt

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan angka nama file [1-8].txt : 7.txt
=====++++=====
Kode mata kuliah : TF1001
Prasyarat : -
Kode mata kuliah : TF1002
Prasyarat : TF1001
Kode mata kuliah : TF1003
Prasyarat : TF1001 TF1004
Kode mata kuliah : TF1004
Prasyarat : TF1001 TF1002 TF1005
Kode mata kuliah : TF1005
Prasyarat : TF1002
Kode mata kuliah : TF1006
Prasyarat : TF1003 TF1004 TF1007
Kode mata kuliah : TF1007
Prasyarat : TF1004 TF1005
=====++++=====
Semester 1 : TF1001
Semester 2 : TF1002
Semester 3 : TF1005
Semester 4 : TF1004
Semester 5 : TF1003 TF1007
Semester 6 : TF1006
=====++++=====

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>_
```

```
Command Prompt

C:\Users\ACER\Documents\File Kuliah\Semester 4\Strategi Algoritma\Tugas\Tucil 2>python Tucil2_13519099.py
Masukkan nama file [1-8].txt : 8.txt
=====++++=====
Kode mata kuliah : A
Prasyarat : B C
Kode mata kuliah : B
Prasyarat : C D
Kode mata kuliah : C
Prasyarat : J
Kode mata kuliah : D
Prasyarat : G
Kode mata kuliah : E
Prasyarat : D F G C A
Kode mata kuliah : F
Prasyarat : G H I
Kode mata kuliah : G
Prasyarat : I
Kode mata kuliah : H
Prasyarat : J
Kode mata kuliah : I
Prasyarat : H J
Kode mata kuliah : J
Prasyarat : -
=====++++=====
Semester 1 : J
Semester 2 : C H
Semester 3 : I
Semester 4 : G
Semester 5 : D F
Semester 6 : B
Semester 7 : A
Semester 8 : E
=====++++=====
```

4. Alamat

Program berada pada **folder src**. Berhasil dijalankan pada python **versi 3.9.0**

Run di Windows : python 13519099.py

Github : <https://github.com/thomas-fm/Tucil2-Stima>