

LAPORAN TUGAS KECIL II
IF2211 STRATEGI ALGORITMA
Semester II Tahun 2020/2021



Disusun oleh:

Nama	Arjuna Marcelino
NIM	13519021
Kelas	01

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

Daftar Isi

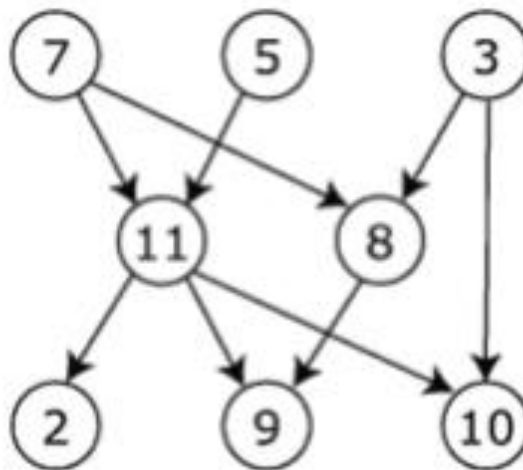
I. Algoritma <i>Topological Sort</i>	2
II. <i>Source Program</i>	5
III. Eksperimen	10
IV. Komponen Penilaian	18
Lampiran	19

I. Algoritma *Topological Sort*

Topological sorting atau *topological ordering* adalah suatu algoritma untuk mengurutkan secara linier semua simpul dari sebuah graf berarah, di mana setiap sisi uv , simpul u mendahului v pada hasil pengurutannya. Setiap simpul dari graf dapat dianggap sebagai pekerjaan yang telah selesai atau tempat yang sudah dikunjungi. Sisi yang ada menandakan perintah kerja yang harus diselesaikan terlebih dahulu sebelum tugas lainnya atau juga menandakan persinggahan yang harus dilewati terlebih dahulu sebelum menuju tempat lainnya. Dalam hal ini, *topological sorting* atau *topological ordering* adalah urutan perencanaan yang valid dari jadwal pekerjaan atau rute perjalanan tersebut. *Topological sorting* dapat berlangsung tepat bila graf yang dipertimbangkan tidak mempunyai siklus langsung, artinya graf tersebut merupakan *Directed Acyclic Graph* (DAG) atau graf langsung tanpa siklus. Setiap DAG harus memiliki setidaknya satu keterurutan secara topologi dan algoritma *topological sorting* digunakan untuk menentukan urutan topologi DAG secara linier.

Hanya graf tidak sirkuler yang terhubung dapat memiliki perluasan linier, karena beberapa sirkuler terhubung adalah sebuah kontradiksi yang melekat pada sebuah urutan linier dari tugas-tugas. Setiap DAG dapat menjadi *topologically sorted*, jadi harus selalu paling sedikit satu jadwal untuk untuk beberapa hal yang beralasan membatasi seluruh pekerjaan. DAG khusus memberikan beberapa jadwal, khususnya ketika ada beberapa kendala. Pertimbangkan n buah pekerjaan tanpa kendala. $N!$ permutasi dari pekerjaan tersebut merupakan sebuah perluasan linier yang valid.

Contoh kererurutan parsial banyak dijumpai dalam kehidupan sehari-hari, misalnya dalam suatu kurikulum, suatu mata pelajaran mempunyai *prerequisite* (prasyarat) mata pelajaran lain. Urutan linier adalah urutan untuk seluruh mata pelajaran dalam kurikulum.



Gambar 1 Contoh *Directed Acyclic Graph* (DAG)

Salah satu keterurutan topologi dari DAG di atas adalah: 5, 7, 3, 8, 11, 10, 9, 2. Keterurutan ini dilihat dari sedikitnya sisi yang masuk ke suatu simpul.

Berikut adalah algoritma dari topological sorting secara umum:

```
Algoritma TopologicalSort(G)  
  H ← G // copy G ke H  
  n ← G.numVertices()  
  while H tidak kosong do  
    v ← temukan simpul dengan  
    tidak ada sisi yang outgoing  
    Label v ← n  
    n ← n - 1  
    Hapus v dari H
```

Gambar 2 Algoritma dari Topological Sorting

Topological sorting menerapkan algoritma traversal yang diterapkan pada metode searching DFS (*depth first search*) dan disajikan dalam bentuk digraph/direct graph.

Untuk merepresantasikan digraph, maka digunakan adjacency matrix dan adjacency list. Adjacency matrix, menggambarkan hubungan antar simpul dalam bentuk matriks. Sedangkan, Adjacency list, menggambarkan hubungan antar simpul yang diwakili oleh edge. Adjacency matrix dan adjacency list merupakan dasar dari algoritma DFS untuk digraph. Penerapan algoritma DFS untuk graph pada umumnya dalam bentuk stack penelusuran dan pohon DFS.

Decrease and conquer merupakan metode perancangan algoritma dengan mereduksi persoalan menjadi dua upa-persoalan (*sub-problem*) yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja. Berbeda dengan *divide and conquer* yang memproses semua upa-persoalan dan menggabungkan semua solusi setiap sub-persoalan. Di dalam literatur lama, semua algoritma yang membagi persoalan menjadi dua upa-persoalan yang lebih kecil dimasukkan ke kategori *divide and conquer*. Meskipun demikian, tidak kedua upa-persoalan hasil pembagian diselesaikan. Jika hanya satu upa-persoalan yang diselesaikan, maka tidak tepat dimasukkan sebagai algoritma *divide and conquer*. Mereka dikategorikan sebagai *decrease and conquer*.

Algoritma *decrease and conquer* terdiri dari dua tahapan. Pertama, *Decrease* yang merupakan tahap mereduksi persoalan menjadi beberapa persoalan yang lebih kecil (biasanya dua upa-persoalan). Kedua, *Conquer* yang merupakan tahap memproses satu upa-persoalan secara rekursif. Tidak ada tahap *combine* dalam *decrease and conquer*, karena hanya satu upa-persoalan yang diselesaikan.

Strategi *decrease and conquer* juga diterapkan pada algoritma *Topological Sort*. Pada prosesnya, akan dihitung semua derajat-masuk setiap simpul (banyaknya busur yang masuk pada simpul tersebut). Kemudian dari hasil perhitungan, dipilih sebuah simpul yang memiliki derajat masuk adalah 0. Simpul tersebut diambil dan dihilangkan beserta semua busur yang keluar dari simpul tersebut pada graf. Hal ini akan mengakibatkan derajat simpul yang berhubungan dengan simpul tersebut juga akan berkurang sebanyak 1 derajat. Kemudian setiap simpul akan memiliki derajat masuk yang baru dan akan dilakukan proses yang sama hingga semua simpul pada DAG terpilih.

Pada permasalahan ini, *topological sort* digunakan untuk menyusun rencana kuliah yang dapat diambil seorang mahasiswa pada tiap semester. Ada beberapa mata kuliah yang memiliki mata kuliah lain sebagai pra-syaratnya. Daftar mata kuliah secara keseluruhan merupakan representasi dari DAG yang akan kita selesaikan. Kemudian mata kuliah adalah tiap simpul pada DAG. Tiap sisi uv pada graf berarti mata kuliah u merupakan prasyarat untuk mata kuliah v . Mata kuliah yang sudah terpenuhi prasyaratnya akan diambil dan disimpan sebagai mata kuliah yang bisa diambil. Kemudian proses ini akan diulang hingga pada semester 8 sehingga akan didapat urutan linier mata kuliah yang dapat diambil dari semester 1 sampai semester 8.

II. Source Program

```
# Arjuna Marcelino
# 13519021
# K1

'''
Prosedur Kalimat Pembuka / Opening
Menampilkan hiasan di awal sebagai pembuka program

Input :
Output :
'''

def pembuka():
    print("#####")
    print("#          --      --          --          --          #")
    print("#      | | | |      | |      | |      | |      #")
    print("#      | | | |      | |      | |      | |      #")
    print("#      |  --  |      | |      | |      | |      #")
    print("#      |  --  |      | |      | |      | |      #")
    print("#      | | | |      | |      | |      --      #")
    print("#      | | | |      | |      | |      /\      #")
    print("#      --      --          --          \      #")
    print("#####")
    print("#                                          #")
    print("#          PENYUSUNAN    RENCANA    KULIAH      #")
    print("#                                          #")
    print("#      ARJUNA MARCELINO          13519021      K01      #")
    print("#                                          #")
    print("#####")
    print("")

'''

Fungsi Load File

Input : Nama file teks
Output : Return Adjacency List

Format penulisan isi file adalah sebagai berikut :
A, B.
C, D, E.
F.
G, H, I, J.

File teks dianggap sudah valid (memenuhi syarat mata kuliah dengan prasyaratnya).
Tidak akan ada proses validasi, jika isi file salah, maka program akan error (karena terjadinya rekurens yang tak hingga).
```

Pastikan file teks anda sudah benar.

Beberapa kesalahan umum :

1. Mata kuliah prasyarat tidak terdaftar di mata kuliah.
 2. Mata kuliah prasyarat dan mata kuliahnya terjadi pada semester yang sama
- ...

```
def load() :  
    #input nama file  
    print("Program hanya akan menerima file yang berekstensi .txt (harap mencantumkan ekstensi file juga)")  
    nama = input("Masukkan nama File Daftar Mata Kuliah : ")  
  
    #membuka dan membaca file yang sudah diletakkan di folder test  
    DataMataKuliah = open ("../test/"+nama,"r")  
  
    #menyimpan ke dalam array  
    MatKul = [[num for num in line.split(',')]] for line in DataMataKuliah  
  
    #menghapus elemen tidak penting  
    MatKul = [[m.replace("\n","") for m in n] for n in MatKul]  
    MatKul = [[m.replace(" ", "") for m in n] for n in MatKul]  
    MatKul = [[m.replace(".", "") for m in n] for n in MatKul]  
  
    return MatKul
```

...

Fungsi Topological Sorting

Metode sorting pada Directed Acyclic Graph untuk menentukan keterurutan setiap simpul dari

simpul yang paling diutamakan sampai yang paling tidak diutamakan.

Input : Adjacency List

Output : List hasil sorting

...

```
def topsort(AdList, hasil):  
    #menyimpan panjang dari AdList atau banyak matkul  
    n = len(AdList)  
    curr =[]  
  
    #basis  
    if (n==0):  
        #do nothing  
        {}  
  
    #rekurens  
    elif (n != 0):  
        visited = False  
  
        for i in AdList:
```

```
#mengambil yang memiliki derajat keluar = 0 atau list yang hanya terdiri
i dari dirinya sendiri (1 elemen)
if(len(i)==1):
    visited = True
    curr.append(i[0])

#menghapus matkul yang sudah diambil
if visited:
    j = 0
    while j < len(AdList):
        for k in curr:
            if k in AdList[j]:
                AdList[j].remove(k)

        if len(AdList[j]) == 0:
            AdList.remove(AdList[j])
            j-=1

        j+=1

#menambahkan matkul pada variabel semester
semester = ""
for j in range (len(curr)):
    if j == len(curr)-1 :
        semester += curr[j]
    elif curr[j]!="":
        semester = semester+curr[j]+", "

#menambahkan semester pada list hasil
hasil.append(semester)

#pemanggilan fungsi topsort
topsort(AdList,hasil)

return hasil

'''
Prosedur Rencana Kuliah
Menampilkan hasil penyusunan rencana kuliah

Input :
Output :
'''
def RencanaKuliah(lisTopo):
    n_sem = len(lisTopo)
    for i in range (n_sem):
        if i == 0:
            print("Semester I : "+lisTopo[i], end='')
```



```
elif i == 1:
    print("\nSemester II : "+lisTopo[i], end='')
elif i == 2:
    print("\nSemester III : "+lisTopo[i], end='')
elif i == 3:
    print("\nSemester IV : "+lisTopo[i], end='')
elif i == 4:
    print("\nSemester V : "+lisTopo[i], end='')
elif i == 5:
    print("\nSemester VI : "+lisTopo[i], end='')
elif i == 6:
    print("\nSemester VII : "+lisTopo[i], end='')
elif i == 7:
    print("\nSemester VIII : "+lisTopo[i], end='')
else:
    #program tidak akan menampilkan mata kuliah yang masih membutuhkan syarat di atas mata kuliah semester 8
    #do nothing
    {}

print(".")

'''
Fungsi Main
Fungsi utama yang memanfaatkan fungsi cabang yang sudah dibuat untuk menghasilkan solusi
Program dapat melakukan pengulangan tergantung pada masukan pengguna di akhir persoalan
dengan hasil akhir menampilkan solusi yang didapat di layar
'''
if __name__ == "__main__":
    #panggil prosedur pembuka
    pembuka()

    #looping sampai pengguna tidak menjawab 'Y'
    while(True):
        current=[]

        #manggil fungsi load
        Matkul = load()

        #manggil fungsi Topological Sorting
        hasil = topsort(Matkul, current)

        #manggil prosedur Rencana Kuliah
        print()
        print("Rencana kuliah yang dihasilkan adalah sebagai berikut.")
        RencanaKuliah(hasil)
```

```
print()

#menanyakan pengguna apakah ingin mengulang program
ulang = input("Apakah ingin melakukan ulang? (Y/N) : ")
print()

#berhenti apabila pengguna tidak menjawab dengan 'Y'
if(ulang!='Y'):
    break
```

III. Eksperimen

Contoh 1

```
Input
uji - Notepad
File Edit Format View
C1, C3.
C2, C1, C4.
C3.
C4, C1, C3.
C5, C2, C4.
```


Output

```
C:\Users\OMEN\OneDrive\Desktop\Tucil2_13519021\src>python main.py
#####
#          --      --           --          --              #
#         | |   | |           | |           | |             #
#         | |   | |           | |           | |             #
#        --    --            --            --              #
#         | |   | |           | |           | |             #
#         | |   | |           | |           | |             #
#         | |   | |           | |           / \            #
#         --    --            --            V               #
#####
#                                                         #
#                PENYUSUNAN     RENCANA       KULIAH        #
#                                                         #
#      ARJUNA MARCELINO          13519021        K01        #
#                                                         #
#####


Masukkan nama File Daftar Mata Kuliah : uji.txt

Rencana kuliah yang dihasilkan adalah sebagai berikut.
Semester I : C3
Semester II : C1
Semester III : C4
Semester IV : C2
Semester V : C5.
```


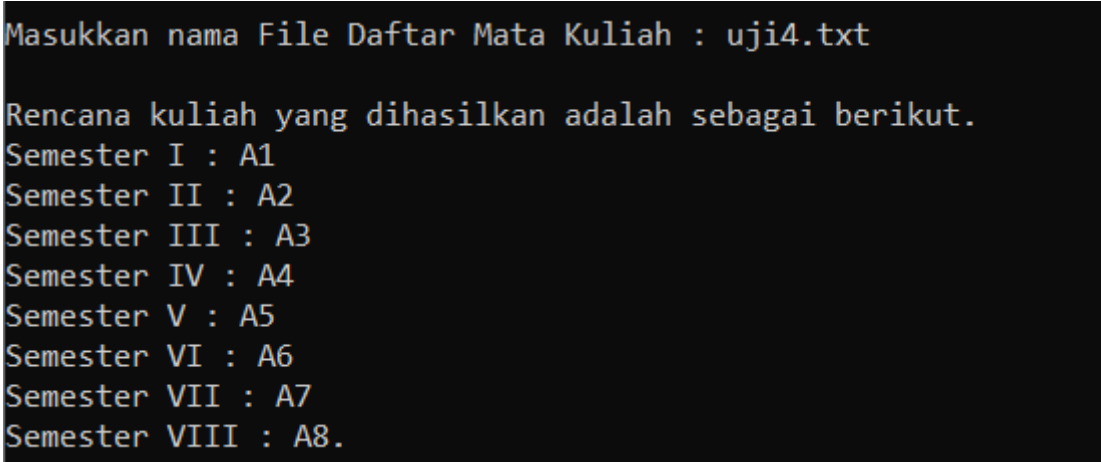
Contoh 2

Input
 uji2 - Notepad File Edit Format View Help KU1102. IF1210, KU1102. IF2110, IF1210, KU1102. IF2210, IF2110, IF1210. IF2211, IF2110, IF1210.
Output
<pre>Apakah ingin melakukan ulang? (Y/N) : Y Masukkan nama File Daftar Mata Kuliah : uji2.txt Rencana kuliah yang dihasilkan adalah sebagai berikut. Semester I : KU1102 Semester II : IF1210 Semester III : IF2110 Semester IV : IF2210, IF2211.</pre>


Contoh 3

Input
 uji3 - Notepad File Edit Format View Help A. B. C. D. E. F, G, H. G, H, H.
Output
<pre>Masukkan nama File Daftar Mata Kuliah : uji3.txt Rencana kuliah yang dihasilkan adalah sebagai berikut. Semester I : A, B, C, D, E, H Semester II : G Semester III : F.</pre>


Contoh 4

Input
 uji4 - Notepad File Edit Format View Help A1. A2, A1. A3, A2, A1. A4, A3, A2, A1. A5, A4, A3, A2, A1. A6, A5, A4, A3, A2, A1. A7, A6, A5, A4, A3, A2, A1. A8, A7, A6, A5, A4, A3, A2, A1. A9, A8, A7, A6, A5, A4, A3, A2, A1.
Output
 <pre>Masukkan nama File Daftar Mata Kuliah : uji4.txt Rencana kuliah yang dihasilkan adalah sebagai berikut. Semester I : A1 Semester II : A2 Semester III : A3 Semester IV : A4 Semester V : A5 Semester VI : A6 Semester VII : A7 Semester VIII : A8.</pre>


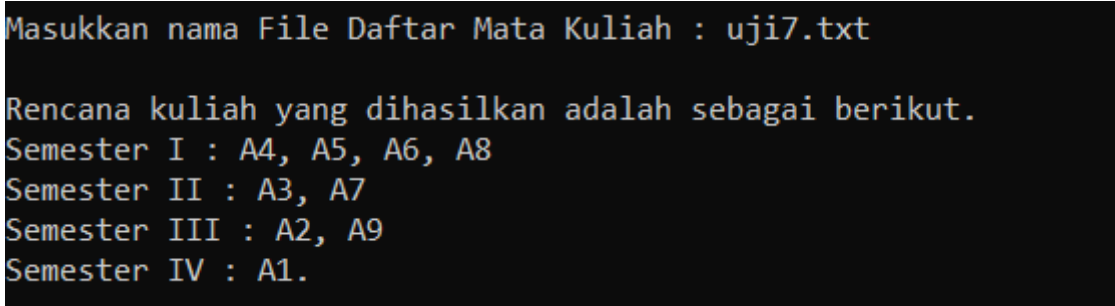
Contoh 5

Input
 uji5 - Notepad File Edit Format View Help KewirausahaanDasar, PengantarBisnis. Kewirausahaan, KewirausahaanDasar. PengantarBisnis. StudiKelayakanBisnis, ManajemenKeuangan, ManajemenPemasaran, KewirausahaanDasar. ManajemenKeuangan. ManajemenPemasaran. PenguatanKelembagaanBisnis, Kewirausahaan. SimulasiBisnis, Kewirausahaan. SeminarKewirausahaan, PenguatanKelembagaanBisnis, SimulasiBisnis.
Output
Masukkan nama File Daftar Mata Kuliah : uji5.txt Rencana kuliah yang dihasilkan adalah sebagai berikut. Semester I : PengantarBisnis, ManajemenKeuangan, ManajemenPemasaran Semester II : KewirausahaanDasar Semester III : Kewirausahaan, StudiKelayakanBisnis Semester IV : PenguatanKelembagaanBisnis, SimulasiBisnis Semester V : SeminarKewirausahaan.


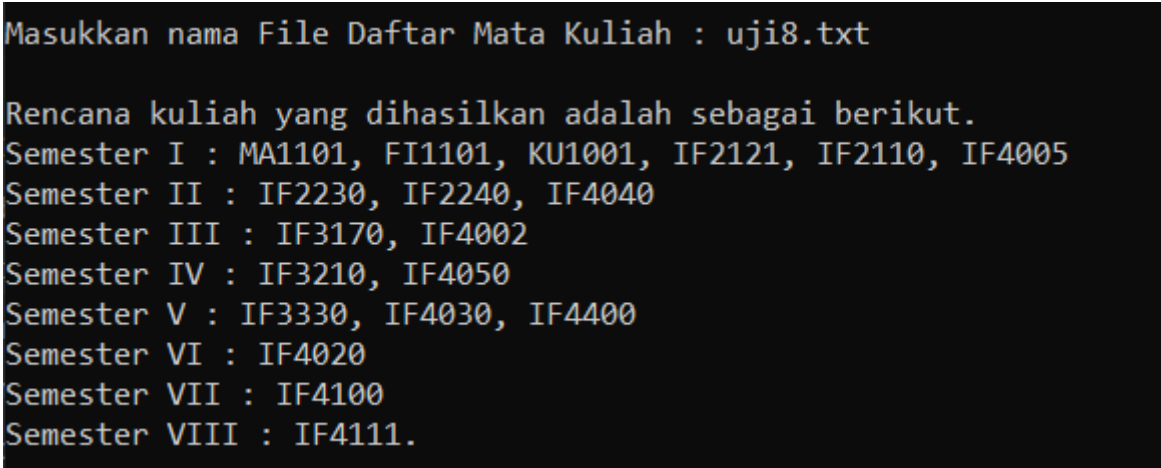
Contoh 6

Input
 uji6 - Notepad File Edit Format View Help MatematikaIA. MatematikaIIA, MatematikaIA. MatematikaDiskrit, MatematikaIIA, MatematikaIA. Strategi Algoritma, MatematikaDiskrit. Algeo, MatematikaIIA. ProbStat, MatematikaDiskrit.
Output
<pre>Masukkan nama File Daftar Mata Kuliah : uji6.txt Rencana kuliah yang dihasilkan adalah sebagai berikut. Semester I : MatematikaIA Semester II : MatematikaIIA Semester III : MatematikaDiskrit, Algeo Semester IV : StrategiAlgoritma, ProbStat.</pre>

Contoh 7

Input	
	 uji7 - Notepad <u>F</u> ile <u>E</u> dit <u>F</u> ormat <u>V</u> iew <u>H</u> elp A1, A2, A3. A2, A3. A3, A4, A5. A4. A5. A6. A7, A8, A5. A9, A7. A8.
Output	
	

Contoh 8

Input	
	 uji8 - Notepad File Edit Format View Help MA1101. FI1101. KU1001. IF2121. IF2110. IF2230, IF2121. IF2240, IF2110. IF3170, IF2230. IF3210, IF3170. IF3330, IF3210. IF4030, IF3210. IF4020, IF4030. IF4002, IF4040. IF4040, IF4005. IF4005. IF4050, IF4002. IF4400, IF4050. IF4100, IF4020. IF4111, IF4100.
Output	
 <pre>Masukkan nama File Daftar Mata Kuliah : uji8.txt Rencana kuliah yang dihasilkan adalah sebagai berikut. Semester I : MA1101, FI1101, KU1001, IF2121, IF2110, IF4005 Semester II : IF2230, IF2240, IF4040 Semester III : IF3170, IF4002 Semester IV : IF3210, IF4050 Semester V : IF3330, IF4030, IF4400 Semester VI : IF4020 Semester VII : IF4100 Semester VIII : IF4111.</pre>	

IV. Komponen Penilaian

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

Lampiran

File/repository dari Tugas Kecil 2 ini dapat juga diakses di pranala berikut ini :

https://github.com/arjunamarcelino/IF2211-Tucil2_13519021.git

atau dapat diakses pada pranala berikut :

<https://drive.google.com/drive/folders/1i2j5l29Au0zlc1qvwmRICk1FJB8kHd6W?usp=sharing>