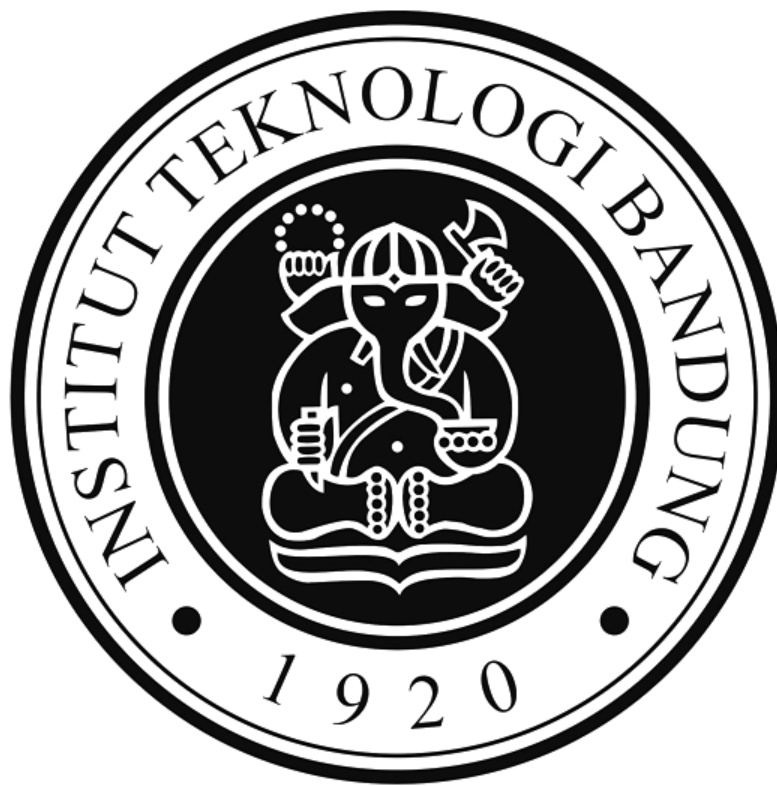


Tugas Kecil 2 IF2211 Strategi Algoritma
Semester II tahun 2020/2021

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



Oleh:

Mhd. Hiro Agayeff Muslion - 13519070

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

Bab I

Deskripsi Masalah

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

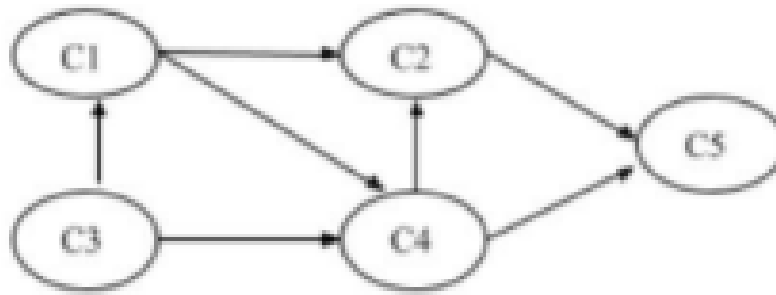
Sebuah kode_kuliah mungkin memiliki nol atau lebih prasyarat kuliah. Kode_kuliah bisa diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil di semester sebelumnya (tidak harus 1 semester sebelumnya). Asumsi semua kuliah bisa diambil di sembarang semester, baik semester ganjil maupun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

Algoritma Decrease and Conquer

Decrease and conquer adalah metode desain algoritma dengan mereduksi persoalan menjadi beberapa sub-persoalan yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja. Metode ini memiliki dua tahapan, antara lain:

1. Decrease, yaitu mereduksi persoalan menjadi beberapa persoalan yang lebih kecil (biasanya dua sub-persoalan).
2. Conquer, yaitu memproses satu sub-persoalan secara rekursif.

Topological Sorting merupakan sebuah metode penyortiran pada Directed Acyclic Graph (DAG) untuk menentukan keterurutan setiap simpul dari simpul yang paling diutamakan sampai yang paling tidak diutamakan. Pada kasus ini, penyortiran dilakukan untuk menemukan matakuliah yang dapat diambil oleh mahasiswa berdasarkan urutannya dari semester paling awal sampai semester akhir.

Topological Sorting merupakan sebuah aplikasi dari metode algoritma *Decrease and Conquer* dimana permasalahan graf tersebut dapat dipecah dengan mencari derajat setiap simpul, dimana hal ini merupakan banyaknya busur yang masuk pada suatu simpul. Setelah mendapat derajat masing-masing simpul, lalu akan dipilih salah satu penyelesaian secara sembarang dengan memilih derajat masuk paling kecil, yaitu 0. Simpul tersebut akan dihilangkan beserta semua busur yang keluar dari simpul tersebut pada graf, lalu derajat simpul yang berhubungan dengan simpul yang dihilangkan tersebut akan dikurangi dengan 1. Langkah-langkah tersebut akan diulangi sampai semua derajat pada simpul menjadi 0. Metode ini merupakan salah satu aplikasi dari algoritma *Decrease and Conquer* karena algoritma ini menerapkan 2 tahapan pada algoritma *Decrease and Conquer*, yaitu pereduksian permasalahan menjadi permasalahan yang lebih kecil dan memproses sub-permasalahan tersebut secara rekursif.

Bab III

Implementasi pada program Matkul Counter menggunakan bahasa python, dengan source code terlampir dibawah berikut.

```
#function to make list to string separated by comma
def listtostring(x):
    temp = ", ".join(x)
    return temp

#logo matkul counter
strip1 = ("-----")
strip2 = ("-----")
strip3 = ("-----")
print("")
print(str(strip1))
print(str(strip2))
print(str(strip3))
print("-----• | • . . . . . -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ | ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print("-----■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ -----")
print(str(strip3))
print(str(strip2))
print(str(strip1))
print("")

#input file
finput = input("Input file (.txt): ")
file = open(finput, "r")
content = file.read()
content = content.replace(".", "") #remove .
content = content.split("\n") #split list indicate by \n

#store matkul beserta preqnya di list content separated by comma
for i in range(len(content)):
    content[i] = content[i].split(", ") #contoh list content = ["matkul1","prereq1"],["matkul2","prereq2"], dst]
```

```

prq = [] #store count prerequisite tiap matkul, list contoh prq = [1,0,3,1]
arrprq = [] #store prerequisite tiap matkul, list contoh arrprq = [["IF1"],[],["IF2","IF3","IF4"],["IF6"]]

#proses mengisi prq dengan count prerequisite masing-masing matkul
#proses mengisi arrprq dengan prerequisite masing-masing matkul, jika prerequisite = 0, maka akan diisi dengan list kosong ([])
for i in range(len(content)):
    arr1 = [] #prerequisite matkul indeks saat ini
    j = len(content[i]) - 1
    for k in range(j):
        l = 1 + k #digunakan agar pada content[i][0] tidak masuk
        arr1.append(content[i][l])
    tmp = len(arr1)
    prq.append(tmp)
    arrprq.append(arr1)

semester = 1 #value output semester pada print output
done = [] #list yang berisikan matkul yang sudah dihitung menjadi prerequisite
# ide algoritma:
#     - mencari prerequisite yang 0 di list prq
#     - memasukkan matkul yang memiliki 0 prerequisite ke list done (yang dimasukkan adalah content[i][0])
#     - mengupdate list prq dan arrprq dengan menghilangkan matkul yang sudah dihitung
#         sebagai prerequisite pada proses sebelumnya
#     - ketika sum(prq) = 0 (ketika list prq sudah berbentuk [0,0,0,0,...,0])
#         , maka
#         akan di print matkul terakhir yang TIDAK MENJADI prerequisite di matkul manapun
while (sum(prq) != 0):
    matkulsmt = [] #list yang digunakan untuk mengoutput matkul yang eligible pada semester ke-n (n = 1,2,...)
    #proses mengisi list done dan matkulsmt
    for i in range(len(prq)):
        if (prq[i] == 0 and content[i][0] not in done):
            done.append(content[i][0])
            matkulsmt.append(content[i][0])
    #proses update list prq dan arrprq (penghilangan matkul yang sudah dihitung menjadi prerequisite)
    for i in range(len(matkulsmt)):
        x = matkulsmt[i]
        for j in range(len(prq)):
            if x in arrprq[j]:
                prq[j] -= 1
                arrprq[j].remove(x)

    #output
    temp = listtoString(matkulsmt) #convert list to string

```

```
print("Semester " + str(semester) + ": " + str(temp))
#next loop semester + 1
semester += 1

#proses output matkul terakhir yang tidak menjadi prerequisite matkul manapun
matkulsisa = [] #matkul terakhir yg tidak menjadi prerequisite (tidak ada di l
ist arrprq)
for i in range(len(content)):
    if (content[i][0] not in done):
        matkulsisa.append(content[i][0])
temp = listtostring(matkulsisa) #convert list to string
#output
print("Semester " + str(semester) + ": " + str(temp))
```


Eksekusi Program

Untuk menguji program Matkul Counter, penulis menggunakan 12 contoh berkas input (berupa ekstensi .txt), dengan hasil masing-masing eksekusi sebagai berikut.

1. Tampilan awal

[illegible]

2. File masukan 1 beserta luarannya

```
test > ≡ ex1.txt
1    C1, C3.
2    C2, C1, C4.
3    C3.
4    C4, C1, C3.
5    C5, C2, C4.
```

```
PS C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2> python -u "c:\Users\Hiro A  
-----  
-----  
      HARKU  
      GUNTER  
-----  
-----
```

```
Input file (.txt): \Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2\test\ex1.txt  
Semester 1: C3  
Semester 2: C1  
Semester 3: C4  
Semester 4: C2  
Semester 5: C5  
PS C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2>
```

3. File masukan 2 beserta luarannya

```
test > ex2.txt
1  A, B, C.
2  B, D, E, F.
3  C, F, G.
4  D, H, I, J.
5  E, J, K, L, M.
6  F, M, N, O.
7  G, O, P, V.
8  H.
9  I, Q, R, S.
10 J, S.
11 K, S.
12 L, T.
13 M.
14 N, U, V.
15 O.
16 P.
17 Q.
18 R.
19 S.
20 T, U.
21 U.
22 V.
```

```
PS C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2> python -u "c:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2\main.py" "c:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2\test\ex2.txt"

-----
          NATURAL
          COMPUTER
          -----

Input file (.txt): \Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2\test\ex2.txt
Semester 1: H, M, O, P, Q, R, S, U, V
Semester 2: G, I, J, K, N, T
Semester 3: D, F, L
Semester 4: C, E
Semester 5: B
Semester 6: A
PS C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2>
```

4. File masukan 3 beserta luarannya

```
test > ex3.txt
1  ID1.
2  ID2, ID1.
3  ID3, ID1, ID2.
4  ID4, ID2, ID3.
5  ID5, ID2, ID4.
6  ID6, ID4.
7  ID7, ID3, ID4.
8  ID8, ID5, ID6, ID7.
```

[illegible]

5. File masukan 4 beserta luarannya

```
test > ex4.txt
1    IF01, IF03, IF05.
2    IF02, IF03.
3    IF03.
4    IF04, IF01, IF02, IF05.
5    IF05.
6    IF06, IF05.
7    IF07, IF04.
8    IF08, IF04.
9    IF09, IF07, IF08.
10   IF10, IF06, IF08, IF11.
11   IF11, IF06.
12   IF12, IF09, IF10.
13   IF13, IF10.
```

[illegible]

6. File masukan 5 beserta luarannya

[illegible]

7. File masukan 6 beserta luarannya

```
test > ≡ ex6.txt
```

1	C1, C3.
2	C2, C3, C5, C7.
3	C3.
4	C4, C1, C3.
5	C5.
6	C6, C5, C3.
7	C7, C4.

```
P5 C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2> python -U "c:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2\test\ex6.txt"
```

```
-----  
-----  
-----  
MATRIAL  
LOUTER  
-----  
-----  
-----
```

```
Input file (.txt): \Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2\test\ex6.txt  
Semester 1: C3, C5  
Semester 2: C1, C6  
Semester 3: C4  
Semester 4: C7  
Semester 5: C2  
P5 C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2>
```

8. File masukan 7 beserta luarannya

```
test > ex7.txt
1 MA1101.
2 FI1101.
3 KU1001.
4 KU1102.
5 KU1011.
6 KU1024.
7 MA1201, MA1101.
8 FI1201, FI1101.
9 IF1210, KU1102.
10 KU1202, KU1102.
11 KI1002, KU1011.
12 EL1200, FI1101.
13 IF2121, IF1210, MA1101, MA1201.
14 IF2110, KU1102, IF1210.
15 IF2120, MA1201, MA1101.
16 IF2124, EL1200.
17 IF2123, MA1201.
18 IF2130, KU1202.
19 IF2210, IF2110.
20 IF2211, IF2110.
21 IF2220, MA1101, MA1201, IF2120.
22 IF2230, IF2130.
23 IF2240, IF2121, IF2120.
24 IF2250, KU1202, IF2110.
25 IF3170, IF2121, IF2124, IF2220, IF2211.
26 IF3110, IF2210, IF2110.
27 IF3130, IF2230.
28 IF3141, IF2240, IF2250.
29 IF3150, IF2250.
30 IF3140, IF2240.
31 IF3151, IF2250.
32 IF3210, IF2110, IF2130, IF3110.
33 IF3270, IF2210, IF3170.
34 IF3230, IF3130.
35 IF3250, IF2250, IF3150.
36 IF3260, IF2123, IF2110, IF2130, IF3151.
37 IF3280, IF3151, IF3150.
38 IF4090, IF3280.
```

```
PS C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2> python -u "c:\Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2\TUCIL 2.py"
```

MATRIX COLLATERA

Input file (.txt): \Users\Hiro Agayeff\Documents\0KULIAH\SMT_4\STRATEGI ALGORITMA\TUCIL 2\test\ex7.txt

Semester 1: MA1101, FI1101, KU1001, KU1102, KU1011, KU1024

Semester 2: MA1201, FI1201, IF1210, KU1202, KI1002, EL1200

Semester 3: IF2121, IF2110, IF2120, IF2124, IF2123, IF2130

Semester 4: IF2210, IF2211, IF2220, IF2230, IF2240, IF2250

Semester 5: IF3170, IF3110, IF3130, IF3141, IF3150, IF3140, IF3151

Semester 6: IF3210, IF3270, IF3230, IF3250, IF3260, IF3280

Semester 7: IF4090, IF4091

Semester 8: IF4092

PS C:\Users\Hiro Agayeff\Documents\0KULIAH\SMT 4\STRATEGI ALGORITMA\TUCIL 2>

9. File masukan 8 beserta luarannya

```
test > ex8.txt
```

```
1 T1, T5, T9, T10.
```

```
2 T2, T1.
```

```
3 T3, T1, T4.
```

```
4 T4, T9.
```

```
5 T5, T4.
```

```
6 T6, T2, T13.
```

```
7 T7, T6, T8, T11.
```

```
8 T8, T3.
```

```
9 T9.
```

```
10 T10.
```

```
11 T11, T4, T10.
```

```
12 T12, T7.
```

```
13 T13.
```

```
PS C:\Users\Hiro Agayeff\Documents\ØKULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2> python -U "c:\Users\Hiro A
```

```
-----  
-----  
-----  
HAND  
COUNTA  
-----  
-----  
-----  
-----  
-----
```

```
Input file (.txt): \Users\Hiro Agayeff\Documents\ØKULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2\test\ex8.txt  
Semester 1: T9, T10, T13  
Semester 2: T4  
Semester 3: T5, T11  
Semester 4: T1  
Semester 5: T2, T3  
Semester 6: T6, T8  
Semester 7: T7  
Semester 8: T12  
PS C:\Users\Hiro Agayeff\Documents\ØKULIAH\SMT_4\STRATEGI ALGORITMA\tUCIL 2>
```

10. File masukan 9 beserta luarannya

```
test > ≡ ex9.txt
1  A1.
2  B1, A1.
3  C1, B1.
4  D1, B1, G1.
5  E1, C1, D1.
6  F1, E1.
7  G1.
```


12. File masukan 11 beserta luarannya

```
test > ≡ ex11.txt
1      A.
2      B, A, J.
3      C, A.
4      D.
5      E, A, D.
6      F, D, K.
7      G, C, E, K.
8      H, A, B, C, I, L.
9      I, F.
10     J.
11     K.
12     L.
```

[illegible]

13. File masukan 12 beserta luarannya

```
test > ≡ ex12.txt
1  A.
2  B.
3  C.
4  D, C.
5  E, A, B, C.
6  F, G.
7  G, E, H.
8  H, A, D.
```


Bab V

Lain-Lain

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

Alamat yang berisi kode program,

Github: <https://github.com/rdx199/Tugas-Kecil-2-Strategi-Algoritma-Topological-Sort.git>