

LAPORAN TUGAS KECIL 2
IF2211 STRATEGI ALGORITMA



Disusun oleh :

Kadek Dwi Bagus Ananta Udayana

13519057

Teknik Informatika

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2021

ALGORITMA TOPOLOGICAL SORT

Topological sort sudah menjadi problem yang cukup populer. Kebanyakan orang sering menyingkat topological sort menjadi toposort. Problem topological sort ini termasuk dalam problem graph. Lebih khususnya lagi, problem ini membahas pengurutan node pada *directed acyclic graph* (DAG). Untuk lebih mengenal toposort, pertama-tama harus mengenal DAG terlebih dahulu.

DAG merupakan singkatan dari *Directed Acyclic Graph*. Direct itu artinya berarah, acyclic berarti tidak memiliki loop. Intinya DAG adalah graph berarah yang tidak memiliki loop. Terdapat banyak sumber yang sudah membahas DAG. Algoritma *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. Algoritma ini berbeda dengan *divide and conquer* yang memproses semua upa-persoalan dan menggabungkan semua solusi setiap sub-persoalan. Algoritma *decrease and conquer* terdiri dari dua tahapan. Pertama *decrease* yaitu mereduksi persoalan menjadi beberapa persoalan yang lebih kecil (biasanya dua upa-persoalan). Kedua *conquer* yaitu memproses satu upa-persoalan secara rekursif. Terdapat 3 varian *decrease and conquer* yaitu *decrease by a constant*, *decrease by a constant factor*, dan *Decrease by a variable size*.

Pada persoalan Topological Sort, penyelesaian dilakukan dengan menggunakan varian *decrease by a variable size* dimana ukuran instant persoalan direduksi bervariasi pada setiap iterasi algoritma. Penulis memberikan nama program topological sort ini dengan nama *Find Course App*. Adapun algoritma topological sort yang diimplementasikan melalui tahap-tahap sebagai berikut.

1. Membaca masukan dari file txt dan membuat representasi graf dengan menggunakan array 2 dimensi. Dalam hal ini penulis menggunakan bahasa python, sehingga lebih mudah jika menyimpan data dengan array 2 dimensi. Penulis menyimpan program ini di dalam fungsi `makeListCourse`
2. Setelah itu, penulis mencari semua array dalam array 2 dimensi yang telah dibentuk tadi, yang memiliki panjang array 1. Ini berarti, pelajaran (dalam hal ini anggota) dalam array tersebut tidak memiliki syarat yang harus dipenuhi lagi. Proses ini nantinya akan dilakukan secara rekursif sehingga proses ini merupakan penerapan algoritma *conquer*.
3. Selanjutnya setelah mendapat semua array yang memiliki panjang array 1, maka program menyimpan semua anggota array itu ke dalam array baru yang dinamakan array pelajaran (memiliki anggota unik) dan setelahnya penulis mereduksi atau pop semua array yang memiliki panjang 1. Proses reduksi ini merupakan penerapan algoritma *decrease*
4. Setelah mendapatkan array pelajaran, penulis mereduksi atau pop semua anggota array di dalam array 2 dimensi awal yang memiliki anggota sama dengan array pelajaran. Jadi dalam

hal ini, program akan mereduksi semua syarat yang telah terpenuhi. Proses reduksi ini merupakan penerapan algoritma *decrease*

5. Ulangi langkah 2, 3, dan 4 hingga array 2 dimensi tidak memiliki anggota (penerapan proses conquer).
6. Kasus khusus. Pertama ketika tidak mendapatkan array yang memiliki panjang array 1 di dalam array 2 dimensi, maka program akan mengeluarkan pesan error bahwa tidak bisa mengambil pelajaran pada semester selanjutnya hingga semester 8. Kedua ketika array 2 dimensi masih memiliki nilai, tetapi telah melakukan 8 iterasi (melewati 8 semester). Maka program akan menampilkan pesan error bahwa masih terdapat pelajaran yang tersisa.

SOURCE CODE

Dalam pembuatan source code untuk membuat aplikasi penyelesaian Topological sorting ini. Penulis membuat program *find course* dengan menggunakan bahasa *python* dan membuatnya menjadi modular dengan membaginya ke dalam beberapa fungsi dan prosedur. Berikut fungsi dan prosedur yang penulis buat beserta implementasinya.

1. Prosedur printWelcome

Menampilkan output ucapan selamat datang saat program pertama kali dijalankan

```
##### VOID UNTUK MEMBUAT TAMPILAN WELCOME #####
def printWelcome() :
    print("#####")
    print("##          WELCOME TO          ##")
    print("##          FIND COURSE APP        ##")
    print("##                                ##")
    print("#####")
    print("\n")
#####
```

2. Prosedur printThankyou

Menampilkan output terima kasih saat program telah selesai dijalankan

```
##### VOID UNTUK MEMBUAT TAMPILAN THANKYOU #####
def printThankyou() :
    print("#####")
    print("##          NOW YOU CAN CHOOSE YOUR COURSE          ##")
    print("##                      HAVE A NICE DAY                      ##")
    print("##                      THANK YOU                      ##")
    print("#####")
#####
```

3. Prosedur printStillHaveCourse

Menampilkan output terima kasih dan memberi tahu bahwa masih terdapat pelajaran yang belum diambil

```
##### VOID UNTUK MEMBUAT TAMPILAN THANKYOU #####
def printStillHaveCourse(listPelajaran) :
    print("#####")
    print("##      NOW YOU CAN CHOOSE YOUR COURSE      ##")
    print("[" | BUT YOU STILL HAVE COURSE ", end=""]
    pelajaran = []
    for i in range(len(listPelajaran)) :
        for j in range(len(listPelajaran[i])) :
            if (i == 0) and (j == 0) : # jika di indeks awal maka tidak usah print , didepannya
                pelajaran.append(listPelajaran[i][j])
            else :
                tidakAda = True
                for k in range (len(pelajaran)) :
                    if pelajaran[k] == listPelajaran[i][j] :
                        tidakAda = False
                if tidakAda :
                    pelajaran.append(listPelajaran[i][j])
    for i in range(len(pelajaran)) :
        if (i == 0) :
            print(pelajaran[i], end="")
        else :
            print(", " , pelajaran[i], end="")
    print(" AND YOU CAN'T PICK      ", end="\n")
    print("##                          THANK YOU      ##")
    print("#####")
#####
```

4. Prosedur printInputFalse

Menampilkan output *error* saat program menerima input yang tidak bisa diselesaikan dengan menggunakan algoritma topological sort

```
##### VOID UNTUK MEMBUAT TAMPILAN THANKYOU #####
def printInputFalse() :
    print("#####")
    print("##      PLEASE, COMPLETE YOUR COURSE IN THE PREVIOUS SEMESTER      ##")
    print("##                                OR                                ##")
    print("##      CHECK YOUR INPUT TEXT      ##")
    print("#####")
#####
```

5. Fungsi makeListCourse

Fungsi yang menerima hasil bacaan file eksternal dan mengembalikan ListPelajaran beserta syarat-syaratnya.

```
##### MEMASUKKAN FILE EKSTERNAL KE DALAM ARRAY #####
def makeListCourse (tucil) :
    listPelajaran = []
    for baris in tucil:
        listSyarat = []
        pelajaran = ""
        for i in range(len(baris)):
            if not baris[i] == ',' and not baris[i] == '.' and not baris[i] == ' ' and not baris[i] == '\n': #merangkai string pelajaran
                pelajaran += baris[i]
            elif (baris[i] == ',' or baris[i] == '.') and pelajaran != "" : # akhir rangkaian string pelajaran
                listSyarat.append(pelajaran)
                pelajaran = "" # kembalikan pelajaran menjadi string kosong
        listPelajaran.append(listSyarat)
    return listPelajaran
#####
```

6. Fungsi isSyarat

Fungsi yang menerima inputan array pelajaran dan array listPelajaran beserta syarat-syaratnya. Lalu fungsi akan mengembalikan array listPelajaran baru yang syarat-syaratnya telah di pop.

```
##### APAKAH COURSE YANG TELAH DI AMBIL MENJADI SYARAT DI COURSE LAIN #####
def isSyarat(pelajaran, listPelajaran) :
    for i in range (len(listPelajaran)) :
        j = 0
        while j < len(listPelajaran[i]) :
            if (j >= len(listPelajaran[i])) : #jika indeks melewati panjang array, karena telah di pop
                break
            k = 0
            while k < len(pelajaran) :
                if listPelajaran[i][j] == pelajaran[k] :#jika pelajaran merupakan syarat, maka di pop pada array
                    listPelajaran[i].pop(j)
                    j = 0
                    k = 0
                    k +=1
                j +=1
            j +=1
    return listPelajaran
#####
```


7. Prosedur printCourse

Prosedur ini bisa dibilang sebagai prosedur yang paling utama. Prosedur ini akan menampilkan output pelajaran apa saja yang bisa diambil setiap 1 semester.

```
##### PRINT COURSE SETIAP SEMESTER #####
def printCourse(listPelajaran):
    semester = 1
    isPrintThankyou = True #jika inputan masih di anggap benar
    while len(listPelajaran) != 0 and semester < 9 :
        pelajaran = [] # list pelajaran pada setiap semester
        tidakAdaPrint = True
        print("semester", semester, ": ", end="")
        i = 0;
        while i < len(listPelajaran) :
            if i >= len(listPelajaran) : #jika indeks melebihi panjang array, karena telah di pop
                break
            if len(listPelajaran[i]) == 1 : #jika pelajaran sudah tidak ada syarat
                pelajaran.append(listPelajaran[i][0])
                if len(pelajaran) == 1 :
                    print(pelajaran[len(pelajaran) - 1], end="")
                else :
                    print(", ", pelajaran[len(pelajaran) - 1], end="")
                tidakAdaPrint = False
                listPelajaran.pop(i)
                i -=1;
            i +=1
        if tidakAdaPrint : #jika inputan salah
            print("There are no course that you can choose for this semester")
            print("\n")
            printInputFalse()
            isPrintThankyou = False
            break
        else :
            print("\n")
            listPelajaran = isSyarat(pelajaran, listPelajaran)
            semester +=1
    if isPrintThankyou :
        if len(listPelajaran) == 0 :
            printThankyou()
        else :
            printStillHaveCourse(listPelajaran)
#####
```

8. Prosedur mainProgram

Prosedur ini akan menjalankan program dari awal hingga akhir.

```
##### MAIN PROGRAM #####
def mainProgram(tucil) :
    printWelcome()
    listPelajaran = makeListCourse(tucil)
    printCourse(listPelajaran)
#####
```

HASIL PERCOBAAN

1. INPUT (Kasus banyak pelajaran diambil dalam 1 semester)

```
II2130.  
II2110.  
II2111.  
TI3005.  
IF2140.  
IF2111.  
II2250, IF2140.  
II2260, II2130.  
II2230, II2130.  
II2220, TI3005.  
II2240, TI3005, IF2111.  
IF2212, IF2111.  
II3150, II2260, II2230.  
II3160, II2260, II2130.  
II3120, II2260, II2220.  
II3131, II2230, IF2212.  
II3121, II2220, TI3005.  
IF3152, II2220, TI3005, II2240.  
II3260, IF2111, IF2212, II3150.  
II3230, II2230, II3131.  
II3211, IF3152, II2220.  
II3220, II3121.  
II3240, IF3152, II3160.  
II4090, IF2111, IF2212, II3211.  
II4091, II3211.  
II4370, II3230.  
II4371, II3240, II2110.  
II4092, II4091.  
II4472, II4370, II2260.
```

OUTPUT

```
#####  
##          WELCOME TO          ##  
##          FIND COURSE APP      ##  
##                               ##  
#####  
  
semester 1 : II2130, II2110, II2111, TI3005, IF2140, IF2111  
  
semester 2 : II2250, II2260, II2230, II2220, II2240, IF2212  
  
semester 3 : II3150, II3160, II3120, II3131, II3121, IF3152  
  
semester 4 : II3260, II3230, II3211, II3220, II3240  
  
semester 5 : II4090, II4091, II4370, II4371  
  
semester 6 : II4092, II4472  
  
#####  
##          NOW YOU CAN CHOOSE YOUR COURSE      ##  
##          HAVE A NICE DAY          ##  
##          THANK YOU                ##  
#####
```


2. INPUT (Kasus terdapat pelajaran sisa yang belum diambil selama 8 semester)

```
AA1.  
AA2, AA1.  
AA3, AA1.  
AA4, AA2, AA1.  
AA5, AA2, AA4, AA7.  
AA6, AA4.  
AA7, AA6, AA3.  
AA8, AA5.  
AA9, AA8, AA5.  
AA10, AA9, AA2.
```

OUTPUT

```
#####  
##                WELCOME TO                ##  
##                FIND COURSE APP            ##  
##                                           ##  
#####  
  
semester 1 : AA1  
  
semester 2 : AA2, AA3  
  
semester 3 : AA4  
  
semester 4 : AA6  
  
semester 5 : AA7  
  
semester 6 : AA5  
  
semester 7 : AA8  
  
semester 8 : AA9  
  
#####  
##                NOW YOU CAN CHOOSE YOUR COURSE                ##  
##                BUT YOU STILL HAVE COURSE AA10 AND YOU CAN'T PICK                ##  
##                THANK YOU                ##  
#####
```

3. INPUT (Kasus bahwa inputan tidak bisa diselesaikan dengan topological sort)

```
1  A1.  
2  A2, A1.  
3  A3, A1.  
4  A4, A2, A3, A5.  
5  A5, A2, A4.  
6  A6, A4, A3.
```

OUTPUT

```
#####
##                WELCOME TO                ##
##                FIND COURSE APP            ##
##                                           ##
#####

semester 1 : A1

semester 2 : A2, A3

semester 3 : There are no course that you can choose for this semester

#####
##          PLEASE, COMPLETE YOUR COURSE IN THE PREVIOUS SEMESTER          ##
##                                OR                                         ##
##          CHECK YOUR INPUT TEXT                                           ##
#####
PS D:\ITB\Tingkat2\Tingkat2_semester2\Strategi Algoritma\TUGAS_KFCTL\tug
```

4. INPUT (Kasus banyak pelajaran diambil dalam 1 semester)

```
Logkom.
Alstrukdat.
RPL, TBFO.
AI, Logkom, Algeo, Stima.
WBD, RPL.
Orkom.
OOP, Alstrukdat.
Stima, Matdis.
Probststat, Matdis, Algeo.
OS, Orkom.
Basdat, Alstrukdat.
MachineLearning, AI, Probststat, Algeo, Stima.
Sispar, OS, Orkom, Jarkom.
PPL, RPL, MRPL.
Grafkom, IMK, Algeo.
SocioIF, AI, MRPL.
Jarkom, OS, Orkom.
SI, Basdat.
MRPL, RPL.
MBD, Basdat.
IMK, OOP, Probststat.
PDA, WBD, MRPL.
Matdis.
TBFO.
Algeo.
KP, Alstrukdat, OOP, AI, PDA.
TA1, SocioIF.
TA2, TA1.
```

OUTPUT

```
#####
##                WELCOME TO                ##
##                FIND COURSE APP            ##
##                                           ##
#####

semester 1 : Logkom, Alstrukdat, Orkom, Matdis, TBF0, Algeo
semester 2 : RPL, OOP, Stima, Probstat, OS, Basdat
semester 3 : AI, WBD, Jarkom, SI, MRPL, MBD, IMK
semester 4 : MachineLearning, Sispar, PPL, Grafkom, SocioIF, PDA
semester 5 : KP, TA1
semester 6 : TA2

#####
##                NOW YOU CAN CHOOSE YOUR COURSE                ##
##                HAVE A NICE DAY                                ##
##                THANK YOU                                       ##
#####
```

5. INPUT (Kasus terdapat pelajaran sisa yang belum diambil selama 8 semester)

```
C1.
C2, C1.
C3, C2, C9.
C4, C3.
C5, C4, C2.
C6, C5, C10, C4.
C7, C6.
C8, C7.
C9.
C10, C3, C2.
C11, C7, C10, C5.
C12. C7.
C13, C8.
C14, C13.
C15, C14. |
```

OUTPUT

```
#####
##                WELCOME TO                ##
##                FIND COURSE APP            ##
##                ##                          ##
#####

semester 1 : C1, C9

semester 2 : C2

semester 3 : C3

semester 4 : C4, C10

semester 5 : C5

semester 6 : C6

semester 7 : C7

semester 8 : C8, C11, C12

#####
##                NOW YOU CAN CHOOSE YOUR COURSE                ##
##                BUT YOU STILL HAVE COURSE C13, C14, C15 AND YOU CAN'T PICK                ##
##                THANK YOU                ##
#####
```

6. INPUT (Kasus Sederhana)

```
C1, C3.
C2, C1, C4.
C3.
C4, C1, C3.
C5, C2, C4.
C6.

```

OUTPUT

```
#####
##                WELCOME TO                ##
##                FIND COURSE APP            ##
##                                           ##
#####

semester 1 : C3, C6

semester 2 : C1

semester 3 : C4

semester 4 : C2

semester 5 : C5

#####
##                NOW YOU CAN CHOOSE YOUR COURSE                ##
##                HAVE A NICE DAY                                ##
##                THANK YOU                                     ##
#####
```

7. INPUT (Kasus bahwa inputan tidak bisa diselesaikan dengan topological sort)

```
A6, A1.
A7, A6.
A1, A2, A3, A4, A5, A7.
A2, A3, A4, A5.
A3, A4, A5.
A4, A5.
A5.
```


OUTPUT

```
#####
##          WELCOME TO          ##
##          FIND COURSE APP      ##
##                               ##
#####

semester 1 : A5

semester 2 : A4

semester 3 : A3

semester 4 : A2

semester 5 : There are no course that you can choose for this semester

#####
##          PLEASE, COMPLETE YOUR COURSE IN THE PREVIOUS SEMESTER  ##
##                               OR                               ##
##          CHECK YOUR INPUT TEXT                                ##
#####
```

8. INPUT (Kasus banyak pelajaran diambil dalam 1 semester)

```
1  FISIKA, KALKULUS.
2  KALKULUS, PENGKOM, PTI.
3  PTI.
4  PENGKOM.
5  KIMIA, PENGKOM, TTKI.
6  PAR, KALKULUS, KIMIA.
7  DASPRO, TTKI, KALKULUS.
8  TTKI.
```

OUTPUT

```
#####
##          WELCOME TO          ##
##          FIND COURSE APP      ##
##                               ##
#####

semester 1 : PTI, PENGKOM, TTKI

semester 2 : KALKULUS, KIMIA

semester 3 : FISIKA, PAR, DASPRO

#####
##          NOW YOU CAN CHOOSE YOUR COURSE          ##
##          HAVE A NICE DAY                          ##
##          THANK YOU                                ##
#####
```


LINK SOURCE CODE

Github : https://github.com/dwibagus154/find_course.git

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