LAPORAN TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA

Semester II tahun 2020/2021

Penyusunan Rencana Kuliah dengan Topological Sort

(Penerapan Decrease and Conquer)

OLEH :

M. IBNU SYAH HAFIZH
13519177
K-04

INSTITUT TEKNOLOGI BANDUNG
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA
PROGRAM STUDI TEKNIK INFORMATIKA

BAB I
ALGORITMA TOPOLOGICAL DAN KAITANNYA DENGAN PENDEKATAN DECREASE AND CONQUER

Topological sort dari graf berarah adalah urutan linear dari simpul-simpulnya sedemikian rupa sehingga untuk setiap tepi berarah uv dari simpul u ke simpul v, u berada sebelum v dalam urutannya. Topological sort dapat diselesaikan dengan pendekatan algoritma decrease and conquer. Algoritma decrease and conquer adalah algoritma yang menyelesaikan sebuah persoalan dengan membaginya menjadi dua buah upa persoalan yang lebih kecil dan menyelesaikan salah satu dari dua buah upa persoalan tersebut untuk mendapatkan solusi global yang optimal dari persoalan awal.

Langkah-langkah dalam menyelesaikan topological sort dengan pendekatan algoritma decrease and conquer yaitu yang pertama menentukan derajat masuk tiap simpul dari graf (DAG). Kemudian mereduksi graf tersebut berdasarkan derajat masuk tiap simpul sehingga didapatkan dua buah upa persoalan dari persoalan awal yaitu kumpulan simpul yang berderajat masuk nol dan simpul yang berderajat masuk lebih dari nol. Dikarenakan dikurangi dengan simpul yang berderajat masuk satu maka derajat tiap simpul dari upa persoalan simpul yang berderajat lebih dari nol akan berkurang sebanyak simpul berderajat masuk nol sebelumnya yang berhubungan dengan simpul tersebut. Lalu langkah ini diulangi untuk upa persoalan yang terdiri atas simpul-simpul berderajat lebih dari nol. Dengan menyelesaikan secara rekursif pada satu buah upa persoalan saja maka akan didapatkan solusi optimal untuk persoalan awal.

Pada program ini yaitu program yang dapat menyusun rencana pengambilan mata kuliah diimplementasikan dengan menggunakan pendekatan topological sorting dan memanfaatkan algoritma decrease and conquer. Mata kuliah direpresentasikan sebagai simpul dengan persyaratan (prerequisite) sebagai derajat masuk. Untuk menentukan pengambilan mata kuliah yang tepat tiap semesternya dapat diselesaikan dengan langkah-langkah penyelesaian topological sort dengan memanfaatkan algoritma decrease and conquer. Program ini memanfaatkan fungsi decrease dan prosedur conquer yaitu fungsi yang mengembalikan sebuah upa persoalan yang mana sudah diredukasi dengan simpul berderajat nol dan prosedur yang menyelesaikan satu upa persoalan secara rekursif. Pada akhirnya akan didapatkan solusi optimal untuk persoalan di awal. Dapat dilihat bahwa hanya dengan menyelesaikan salah satu upa persoalan tanpa menyelesaikan upa persoalan lainnya dapat dihasilkan solusi yang optimal untuk persoalan besar di awal. Hal ini membuktikan bahwa algoritma decrease and conquer cukup efektif dan efisien dalam menyelesaikan persoalan terutama persoalan topological sort.

BAB II
SOURCE PROGRAM

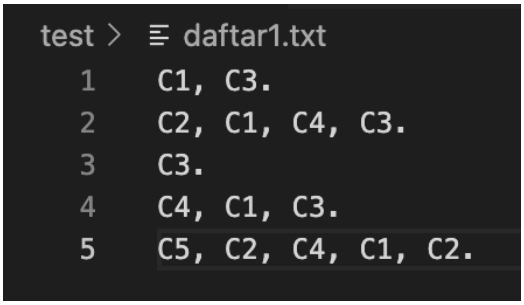
```
# Decrease and Conquer (tugas kecil 2)
# M. Ibnu Syah Hafizh
# 13519177


def inputfile(): # Fungsi untuk input file dan mengembalikan nilai berupa array yang elemennya adalah baris dari isi file
    namafile = str(input("Masukkan nama file (tanpa .txt): "))
    file = open('./test/'+namafile+'.txt', 'r')
    Lines = file.readlines()
    wordlist = []
    for line in Lines:
        clear = line.rstrip('\n').rstrip('.').replace(",","").split(' ')
        wordlist.append(clear)
    return wordlist

matkul = [] #list untuk menampung hasil mata kuliah tiap semester

# Fungsi decrease dengan parameter X bertipe list adalah fungsi untuk mengeleminasi
# simpul GAD yang berderajat masuk 0 dan memasukkan simpul tersebut ke dalam list matkul
# Fungsi deacrease mengembalikan nlai berupa list baru tanpa simpul yang tereleminasi
def decrease(X):
    Xbaru = [] #list untuk menampung elemen yang bukan berderajat masuk 0
    derajat_masuk = [] #list untuk menampung derajat masuk tiap simpul
    for i in X:
        derajat_masuk.append(len(i)-1)
    group = [] #list untuk menampung simpul yang berderajat masuk 0
    for i in range (len(derajat_masuk)):
        if (derajat_masuk[i] == 0):
            group.append(X[i][0]) #Menambahkan simpul yang berderajat masuk 0 ke dalam list group
        else:
            continue
    for i in range (len(derajat_masuk)):
        if (derajat_masuk[i] == 0):
            awal = X[i][0] #menyimpan simpul yang berderajat masuk 0
            break
        else:
            continue
    X.remove([awal]) #menghapus simpul yang berderajat masuk 0 dari list X
    matkul.append(group) #memasukkan list group ke dalam list matkul, nantinya elemen yang berada dalam satu group akan menjadi satu kesatuan (matkul pada 1 semester)
    for i in range (len(X)):
        if(awal in X[i]):
            (X[i]).remove(awal) #menghapus matkul yang dieleminasi dari prerequisite tiap mata kuuliah
            Xbaru.append(X[i])
        else:
            Xbaru.append(X[i])
    return (Xbaru) #mengembalikan list baru tanpa simpul berderajat masuk 0
```

```
# Prosedur conquer dengan parameter X bertipe list adalah prosedur yang digunakan untuk menyelesaikan persoalan
# topological sort dengan mereduksi sebuah persoalan menjadi 2 buah upa-persoalan dan menyelesaikan salah satu upa-persoalan tersebut
# hingga mendapat solusi untuk persoalan awal. Pada hal ini sebuah persoalan direduksi dengan konstanta konstan 1.
def conquer(X):
    if (len(X) == 1): #jika hanya terdapat satu simpul maka akan memasukkan simpul tersebut (kode mata kuliah) ke dalam list matkul
        matkul.append([X[0][0]])
    else: #jika terdapat lebih dari 1 simpul maka akan dieleminiasi satu persatu simpul yang berderajat masuk 0 sampai bersisa satu simpul
        conquer(decrease(X))

# MAIN PROGRAN
listmatkul = inputfile() #membaca input file
conquer(listmatkul) #menyelesaikan persoalan dari file masukkan

# Menghapus duplikat matkul pada tiap semester.
# Tanpa prosedur di bawah ini maka akan terjadi duplikasi kode mata kuliah di beberapa semester.
# Misalnya pada semester pertama terdapat matkul C1 dan di semester kedua terdpaat pula matkul C1
for i in range(1, len(matkul)):
    dual = []
    duplikat = matkul[i][0]
    if (duplikat in matkul[i-1]):
        (matkul[i]).remove(duplikat) #
    j = len(matkul[i])
    for k in range (i):
        for l in range (len(matkul[k])):
            dual.append(matkul[k][l])
    while j > 0:
        if (matkul[i][j-1] in dual):
            (matkul[i]).remove(matkul[i][j-1])
        j = j- 1
while ([] in matkul):
    matkul.remove([])

# Menampilkan hasil program pada layar dengan format "Semester <angka> : <kode mata kuliah>, <kode mata kuliah>." (banyaknya
# kode mata kuliah  tergantung elemen dalam list matkul)
for i in range(len(matkul)):
    print("Semester", i+1, ": ", end='')
    for j in range(len(matkul[i])):
        if (j == len(matkul[i])-1):
            print(matkul[i][j] + '.')
        else:
            print(matkul[i][j], end=', ')
```

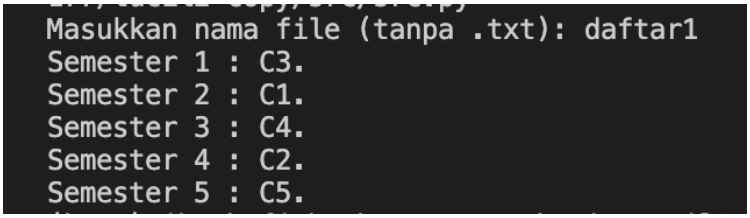
BAB III
HASIL PERCOBAAN

Kasus 1

a. Input

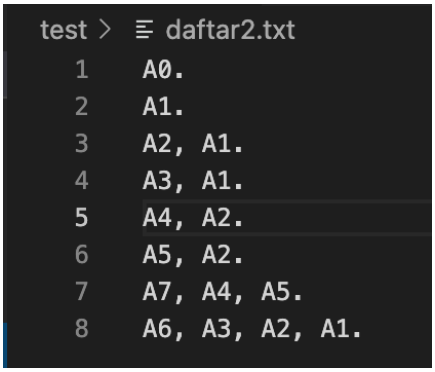


b. Output



Kasus 2

a. Input

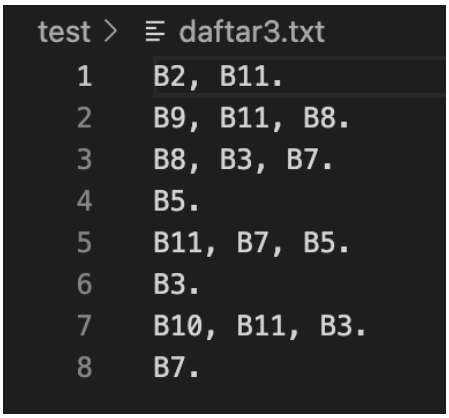


b. Output

```
Masukkan nama file (tanpa .txt): daftar2
Semester 1 : A0, A1.
Semester 2 : A2, A3.
Semester 3 : A4, A5.
Semester 4 : A6.
Semester 5 : A7.
```

Kasus 3

a. Input

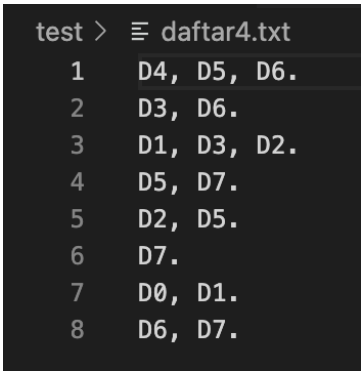


b. Output

```
Masukkan nama file (tanpa .txt): daftar3
Semester 1 : B5, B3, B7.
Semester 2 : B8, B11.
Semester 3 : B2, B9, B10.
```

Kasus 4

a. Input



b. Output

```
Masukkan nama file (tanpa .txt): daftar4
Semester 1 : D7.
Semester 2 : D5, D6.
Semester 3 : D2.
Semester 4 : D4, D3.
Semester 5 : D1.
Semester 6 : D0.
```

Kasus 5

a. Input

test > daftar5.txt

```
1  E1.
2  E2, E1.
3  E3, E2, E4.
4  E4, E2.
5  E5, E4.
```

b. Output

```
Masukkan nama file (tanpa .txt): daftar5
Semester 1 : E1.
Semester 2 : E2.
Semester 3 : E4.
Semester 4 : E3, E5.
```

Kasus 6

a. Input

test > daftar6.txt

```
1  F1.
2  F2, F1.
3  F3, F1.
4  F4, F2, F3.
5  F5, F4.
6  F6, F4.
7  F7, F5, F6.
```

b. Output

```
Masukkan nama file (tanpa .txt): daftar6
Semester 1 : F1.
Semester 2 : F2, F3.
Semester 3 : F4.
Semester 4 : F5, F6.
Semester 5 : F7.
```

Kasus 7

a. Input

test > daftar7.txt

```
1  G2, G1.
2  G1.
3  G3, G1.
4  G4, G2, G3.
5  G5, G4.
6  G6, G4.
```

b. Output

```
Masukkan nama file (tanpa .txt): daftar7
Semester 1 : G1.
Semester 2 : G2, G3.
Semester 3 : G4.
Semester 4 : G5, G6.
```

Kasus 8

a. Input

```
test > daftar8.txt
1   H1, H2, H3.
2   H3, H4, H5.
3   H2, H4.
4   H5.
5   H4.
6   H6, H1.
```

b. Output

```
Masukkan nama file (tanpa .txt): daftar8
Semester 1 : H5, H4.
Semester 2 : H3, H2.
Semester 3 : H1.
Semester 4 : H6.
```

Kasus 9

a. Input

```
test > daftar9.txt
1   J1, J3.
2   J2, J3, J5, J7, J6.
3   J3.
4   J4, J1, J3.
5   J5.
6   J6, J5, J4.
7   J7, J4.
```

b. Output

```
Masukkan nama file (tanpa .txt): daftar9
Semester 1 : J3, J5.
Semester 2 : J1.
Semester 3 : J4.
Semester 4 : J7.
Semester 5 : J6.
Semester 6 : J2.
```

Kasus 10

a. Input

```
test > ≡ daftar10.txt
1   K4, K1, K3.
2   K1.
3   K2, K5.
4   K3.
5   K5, K4.
```

b. Output

```
Masukkan nama file (tanpa .txt): daftar10
Semester 1 : K1, K3.
Semester 2 : K4.
Semester 3 : K5.
Semester 4 : K2.
```

BAB IV
ALAMAT SOURCE CODE

GitHub:
https://github.com/ibnuhafizzh/Tucil2_13519177.git

Google Drive:
https://drive.google.com/drive/folders/1RQrapQJo9fU5COkNQxFgLmDKycTYKtPb?usp=sharing

BAB V
CEK LIST

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	