**TUGAS KECIL 2**

**IF2211 STRATEGI ALGORITMA**

**Penyusunan Rencana Kuliah dengan *Topological Sort***

**(Penerapan *Decrease and Conquer*)**

**SEMESTER 2 TAHUN 2020/2021**

oleh

Irvin Andryan Pratomo / 13519162

Logo

Description automatically generated

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2021**

1. **Algoritma *Topological Sort***

Algoritma *brute force* yang digunakan pada program ini pada dasarnya adalah dengan melakukan *iterate* angka-angka pada *range* tertentu. *Range* angka tersebut ditentukan berdasarkan jumlah huruf unik yang diperoleh dari kata pada operan dan hasil.

Langkah-langkah kerja :

1. Program akan membaca input dari text file yang berisi operan dan hasil, kemudian menyimpan isi dari text file tersebut ke dalam sebuah list.
2. Program “membersihkan” karakter yang tidak perlu seperti ‘+’ dan ‘-----‘, kemudian menyimpan hasilnya ke dalam sebuah listOfKata, yaitu list berisi kata-kata yang terdapat di dalam text file input.
3. Dari listOfKata, program kemudian mengambil huruf-huruf yang terdapat di dalam kata-kata tersebut dan menyimpannya ke dalam list baru bernama hurufUnik. Selain itu program juga mencatat huruf pertama yang digunakan pada kata dan menyimpannya di list baru bernama hurufAwal.
4. Program melakukan looping yang dimulai dari sebuah angka yang ditentukan berdasarkan banyak huruf yang digunakan. Sebuah variabel stringOfDigit menyimpan string ‘1234567890’, Ketika ada 6 huruf yang digunakan, loop akan dimulai dari angka 123456 hal ini untuk mengurangi waktu yang tidak perlu jika looping dilakukan dari angka nol. Batas akhir loop adalah 10^(banyak huruf unik yang digunakan).
5. Ketika program looping dan sedang berada pada indeks dengan digit yang tidak berulang (missal 273098), maka digit-digit tersebut akan dimasukkan ke dalam list bernama angkaUnik [2, 7, 3, 0, 9, 8]. Kemudian huruf yang ada di list hurufUnik akan di-*assign* sesuai dengan angka di dalam list angkaUnik yang indeksnya sama. Missal hurufUnik = [H, R, E, U, P, S], maka huruf H = 2, R = 7, E = 3, U = 0, P = 9, S = 8. Namun hal ini akan batal jika ada angka nol dan huruf yang di-*assign* dengan angka nol adalah anggota dari list hurufAwal (karena leading 0 tidak boleh). Jika batal, maka program akan melanjutkan looping dan mengulangi langkah nomor 5 ini. Angka nol tidak akan mungkin untuk di assign pada huruf di indeks pertama hurufUnik, karena huruf itu sudah pasti adalah huruf awal dari kata di operan pertama (hal ini juga menjadi alasan loop bisa dilakukan tidak dari nol).
6. Ketika berhasil dengan tidak ada huruf pertama yang di assign dengan angka nol, program kemudian menggunakan listOfKata dan angka-angka tersebut akan disusun berdasarkan kemunculan huruf yang bersesuaian, missal kata PURE akan membentuk 9073.
7. Program kemudian menjumlahkan angka-angka tersebut, jika hasil operan + operan + … + operan (n) = hasil, maka akan menuliskannya pada layar, jika gagal, maka kembali ke langkah nomor 5.
8. **Source Code**

Program dibuat menggunakan bahasa Python 3.

'''

Fungsi untuk membuka file input dan memasukkannya ke dalam list 2 dimensi bernama listOfContents

fungsi ini juga sudah menghilangkan tanda titik dan koma yang tidak akan terpakai

hasilnya adalah listOfContents[i][j]

contoh : listOfContents[1][0] adalah matakuliah "IF1111"

listOfContents[1][1], listOfContents[1][2], ... adalah mata kuliah prasyarat dari mata kuliah "IF1111"

'''

def openFile(namaFile):

with open(namaFile) as inputFile:

listOfContents = [line.split() for line in inputFile]

#membuka file dan hasilnya dimasukkan ke dalam list 2 dimensi bernama listOfContents

#hasil yang dimasukkan sudah terbagi menjadi per baris, contoh : [[baris1], [baris2], [baris3]]

for i in range(len(listOfContents)):

for j in range(len(listOfContents[i])):

listOfContents[i][j] = listOfContents[i][j].strip(",.")

#menghapus tanda koma dan titik dari elemen-elemen listOfContents

return listOfContents

'''

Fungsi untuk menyelesaikan persoalan urutan mata kuliah dengan topological sort,

fungsi ini memiliki parameter bertipe list yang berisi nama mata kuliah dan nama mata kuliah prasyaratnya

fungsi ini memiliki return value berupa list 2 dimensi yang berisi urutan mata kuliah tiap semesternya,

contoh return : [[a,b,c], [d,e,f], [g,h,i]] maka a,b,c adalah mata kuliah

yang bisa diambil di semester 1; d,e,f di semester 2; dan g,h,i di semester 3

'''

def getUrutanMatkul(listMataKuliah):

done = False

#inisiasi boolean "done" yang akan digunakan sebagai tanda selesainya proses

urutanMatkul = []

#inisiasi list "urutanMatkul" yang akan menjadi list 2 dimensi berisi mata kuliah terurut per semester

#list ini adalah nilai yang akan di-return oleh fungsi ini

while (not done):

currentSemester = []

#inisiasi list "currentSemester" yang akan diisi mata kuliah yang diambil di semester

#yang sedang diproses, nantinya list ini di-append ke list urutanMatkul

for i in range(len(listMataKuliah)):

if (len(listMataKuliah[i]) == 1):

#cek apakah panjang listMataKuliah[i] = 1, jika iya, maka hanya ada satu elemen

#pada listMataKuliah[i], yang artinya mata kuliah pada listMataKuliah[i][0] adalah

#mata kuliah yang prasyaratnya sudah terpenuhi atau tidak punya prasyarat

currentSemester.append(listMataKuliah[i][0])

#tambahkan mata kuliah yang sudah tidak ada prasyaratnya tersebut ke dalam

#list currentSemester

for i in range(len(currentSemester)):

for j in range(len(listMataKuliah)):

if (currentSemester[i] in listMataKuliah[j]):

listMataKuliah[j].remove(currentSemester[i])

#menghapus mata kuliah yang diambil di semester ini dari listMataKuliah.

#dihapus baik sebagai mata kuliah itu sendiri maupun sebagai prasyarat mata kuliah lain.

urutanMatkul.append(currentSemester)

#menambahkan mata kuliah pada listCurrentSemester ke dalam list urutanMatkul

matkulHabis = True

#boolean untuk mengecek apakah sudah tidak ada mata kuliah yang belum diambil di listMatakuliah

for k in range(len(listMataKuliah)):

if (len(listMataKuliah[k]) != 0):

matkulHabis = False #jika masih ada yang panjangnya > 0 artinya masih ada matkul yang belum diambil

if (matkulHabis == True): #jika sudah tidak ada matkul yang belum diambil, maka selesai

done = True

return urutanMatkul

'''

Prosedur untuk print urutan mata kuliah yang diambil ke layar. contoh hasil :

Semester 1 : C1

Semester 2 : C3 C5

Semester 3 : C4

Semester 4 : C2

'''

def printUrutanMatkul(listUrutanMatkul):

for i in range(len(listUrutanMatkul)):

print("Semester", i+1, ": ", end = '')

for j in range(len(listUrutanMatkul[i])):

print(listUrutanMatkul[i][j], end = ' ')

print('')

'''

Main Program

'''

namaFile = str(input("Masukkan nama file : ")) #nama file input

listMataKuliah = openFile(namaFile) #membuka file dan memasukkan isinya ke listMataKuliah

listUrutanMatkul = getUrutanMatkul(listMataKuliah) #mendapatkan urutan matkul dan memasukkan hasilnya ke listUrutanMatkul

printUrutanMatkul(listUrutanMatkul) #print urutan matkul ke layar

1. **Input / Output**

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| **Graphical user interface, text, application  Description automatically generated** | **A picture containing text, clock, green, time  Description automatically generated** |
| **Graphical user interface, text, application  Description automatically generated** | **A picture containing text, clock, meter, device  Description automatically generated** |
| **Graphical user interface, text, application  Description automatically generated** | **A screenshot of a video game  Description automatically generated with low confidence** |
| **Graphical user interface, text, application  Description automatically generated** | **A picture containing text, clock, clock radio, meter  Description automatically generated** |
| **Graphical user interface, text, application  Description automatically generated** | **A picture containing text, clock, clock radio, meter  Description automatically generated** |
| **Graphical user interface, text, application  Description automatically generated** | **A picture containing text, clock, meter, clock radio  Description automatically generated** |
| **Graphical user interface, text, application  Description automatically generated** | **A screenshot of a video game  Description automatically generated with low confidence** |
| **Graphical user interface, text, application  Description automatically generated** | **A picture containing text, clock, green, meter  Description automatically generated** |

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi | √ |  |
| 1. Program berhasil running | √ |  |
| 1. Program dapat membaca berkas input dan menuliskan output | √ |  |
| 1. Luaran sudah benar untuk semua kasus input | √ |  |

Repository berisi source code :

<https://github.com/irvinandryan/Tucil1_Stima>