**PENYUSUNAN RENCANA KULIAH DENGAN TOPOLOGICAL SORT (PENERAPAN DECREASE AND CONQUER)**

**LAPORAN TUGAS KECIL 2**

Diajukan sebagai laporan dari tugas kecil dua mata kuliah Strategi Algoritma IF2211 pada Semester II Tahun Akademik 2020-2021

Logo

Description automatically generated

Nama : Epata Tuah

NIM : 13519120

Kelas : K-03

Bahasa yang dipilih : Python

**TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2021**

1. **Algoritma *Topological Sort***

Algoritma *topological sort* merupakan salah satu penerapan dari pendekatan *Decrease and Conquer* , yakni algoritma pengurutan *node* pada graf non-sirkuler terarah atau dapat disebut *Directed Acyclic Graph* (DAG) dengan pemisalan *node* A menuju *node* B, *node* A muncul terlebih dahulu sebelum *node* B pada pengurutan. Berikut merupakan langkah-langkah algoritma *topological sort* pada program.

1. Simpan informasi semua mata kuliah yang tersedia pada array courseList
2. Simpan informasi *prerequisite* dari semua mata kuliah pada array coursePlan, indeks sesuai dengan mata kuliah yang ada pada array courseList (pendekatan Directed Acyclic Graph).
3. Inisialisasi sebuah array courseFix yang akan menjadi array mata kuliah yang sudah disusun
4. Cek apakah derajat masuk (Panjang dari coursePlan ke i) adalah nol selama isi array coursePlan belum habis (Masih ada simpul mata kuliah). Jika tidak nol, lewati pemrosesan pada coursePlan tersebut.
5. Jika nol, tambahkan mata kuliah tersebut ke array courseFix, hilangkan mata kuliah tersebut pada array courseList (dan courseList2) dan hilangkan juga *prerequisite*-nya pada array coursePlan
6. Lakukan perulangan dengan *range* panjang array coursePlan terbaru untuk mengecek apakah ada mata kuliah yang baru saja ditambahkan pada courseFix pada *prerequisite* coursePlan tersebut. Jika ada, hapus mata kuliah *prerequisite* tersebut (artinya sudah mengambil mata kuliah ini dan menghapus busur yang keluar dari mata kuliah ini)
7. Ulangi Langkah (4), (5), dan (6) hingga isi array coursePlan nol atau dapat dikatakan semua simpul pada DAG terpilih (Pada langkah ini, pendekatan rekursif dilakukan)

Algoritma *topological sort* menggunakan variasi *Decrease by a Constant* pada pendekatan *Decrease and Conquer* karena pada algoritma ini terjadi penghilangan suatu simpul dan semua busur yang keluar dari simpul tersebut. Algoritma ini akan mengurangi simpul-simpul graf setiap kali ditemukan simpul dengan banyaknya busur yang masuk adalah nol.

1. ***Source Code***
2. 13519120-graphnfunc.py

|  |
| --- |
| # Nama : Epata Tuah  # NIM : 13519120  # Kelas : K-03  # Mata Kuliah : Strategi Algoritma  # Deskripsi : TUGAS KECIL 2 (Penyusunan Rencana Kuliah dengan Topological Sort)  # implementasi 13519120-functions.py  #FUNGSI-FUNGSI ESENSIAL YANG DIGUNAKAN PADA 13519120-main.py dan 13519120-topsort.py  #Fungsi convert(arrayOfChar) menyatukan isi arrayOfChar menjadi sebuah kata  def convert(arrayOfChar):  new = ""  for x in arrayOfChar:  new += x  return new  #REPRESENTASI DAG (DIRECTED ACYCLIC GRAPH)  #Fungsi preqCourse(lineSemester) mengembalikan array preqList dari suatu line kata  #array preqList berisi simpul-simpul mata kuliah prerequisite tujuan dari suatu simpul mata kuliah  def preqCourse(lineSemester):  preqList = []  word = []  i = 0  for character in lineSemester:  if (character == ' '):  continue  if (character == ',' or character == '.'):  i+=1  if (i==1):  word = []  continue  else:  preqList.append(convert(word))  word = []  continue  word.append(character)  return(preqList)  #Fungsi indeksAdaCourse(erasedCourse, coursePlan) mengecek apakah terdapat kata pada array coursePlan yang  #merupakan kata erasedCourse  def indeksAdaCourse(erasedCourse, coursePlan):  for word in coursePlan:  if (word == erasedCourse):  return True  return False  #Prosedur hapusCourse(erasedCourse, coursePlan) menghapus mata kuliah pada array coursePlan berisi  #prerequisite course yang mata kuliahnya adalah erasedCourse (menghapus busur masuk pada simpul mata kuliah  #yang terhubung pada simpul mata kuliah erasedCourse yang telah dihapus)  def hapusCourse(erasedCourse, coursePlan):  for i in range(len(coursePlan)):  if (coursePlan[i] == erasedCourse):  coursePlan.pop(i)  break  #OUTPUT  #Prosedur outputPembagianMatkul(courseFix, jumlahSemester) digunakan untuk mengeluarkan output mata kuliah yang  #harus diambil per semester  def outputPembagianMatKul(courseFix, jumlahSemester):  if (jumlahSemester == 8):  matkulPerSemester = int(len(courseFix)/jumlahSemester)  iSemester = 1  iHitungCourse = 0  iCurrCourse = 0  while (iSemester <= 8):  hitungKoma = 0  print("Semester", iSemester, ": ", end='')  while (iCurrCourse<(matkulPerSemester+iHitungCourse)):  if (matkulPerSemester>1):  if (hitungKoma==matkulPerSemester-1):  print(courseFix[iCurrCourse], end='')  else:  print(courseFix[iCurrCourse], end='')  print(', ', end='')  hitungKoma+=1  else:  print(courseFix[iCurrCourse])  iCurrCourse+=1  iHitungCourse += matkulPerSemester  iSemester+=1  if (matkulPerSemester>1):  print() |

1. 13519120-topsort.py

|  |
| --- |
| # Nama : Epata Tuah  # NIM : 13519120  # Kelas : K-03  # Mata Kuliah : Strategi Algoritma  # Deskripsi : TUGAS KECIL 2 (Penyusunan Rencana Kuliah dengan Topological Sort)  # implementasi 13519120-topsort.py  #import other py  func = \_\_import\_\_("13519120-graphnfunc")  #Algoritma Topological Sort  def topSort(courseFix, coursePlan, courseList, courseList2):  #Inisialisasi i dan j  i = 0  j = 0  #Cek apakah semua simpul pada DAG terpilih  if (len(coursePlan) == 0): #Jika semua simpul telah terpilih  func.outputPembagianMatKul(courseFix, 8)  else: #Jika ada simpul yang belum terpilih  while (i < len(coursePlan)) : #selama ketemu simpul yang derajat masuknya bukan nol, dilakukan perulangan while  if (len(coursePlan[i]) == 0): #Jika derajat masuk = 0 (prerequisite nol)  tempCourse = courseList[i] #simpan nama mata kuliah tersebut di tempCourse (sementara)  #terdapat courseList dan courseList2 untuk kasus input mata kuliah prerequisite pada file .txt  #yang memiliki spasi setelah mata kuliah non-prerequisite  courseFix.append(courseList2[i]) #Tambahkan mata kuliah tersebut di array courseFix  courseList2.pop(i) #Hilangkan simpul mata kuliah tersebut pada array courseList2  courseList.pop(i) #Hilangkan simpul mata kuliah tersebut pada array courseList  coursePlan.pop(i) #Hilangkan himpunan kosong ([] karena prerequisite simpul mata kuliah tersebut nol) pada array coursePlan  #Hapus semua busur yang keluar dari simpul mata kuliah tersebut  for i in range(len(coursePlan)):  if (func.indeksAdaCourse(tempCourse, coursePlan[i])):  func.hapusCourse(tempCourse,coursePlan[i])  break #stop perulangan karena sudah menemui simpu yang derajat masuk = 0  i+=1  #Panggil kembali fungsi topSort (rekursif) hingga semua simpul DAG terpilih (array coursePlan habis)  topSort(courseFix, coursePlan, courseList, courseList2) |

1. 13519120-main.py

|  |
| --- |
| # Nama : Epata Tuah  # NIM : 13519120  # Kelas : K-03  # Mata Kuliah : Strategi Algoritma  # Deskripsi : TUGAS KECIL 2 (Penyusunan Rencana Kuliah dengan Topological Sort)  # implementasi 13519120-main.py  #import other py  TS = \_\_import\_\_('13519120-topsort')  func = \_\_import\_\_("13519120-graphnfunc")  #Pembuka Program  print(" \_\_\_\_ \_ \_\_ \_ \_ \_ ")  print("| \_ \\ \_\_\_ \_ \_\_ \_\_\_ \_\_ \_ \_ \_\_ \_\_ \_ | |/ / \_| (\_) \_\_ \_| |\_\_ ")  print("| |\_) / \_ \\ '\_ \\ / \_\_/ \_` | '\_ \\ / \_` | | ' / | | | | |/ \_` | '\_ \\ ")  print("| \_ < \_\_/ | | | (\_| (\_| | | | | (\_| | | . \\ |\_| | | | (\_| | | | |")  print("|\_| \\\_\\\_\_\_|\_| |\_|\\\_\_\_\\\_\_,\_|\_| |\_|\\\_\_,\_| |\_|\\\_\\\_\_,\_|\_|\_|\\\_\_,\_|\_| |\_|")  print("ASUMSI JUMLAH MATA KULIAH YANG DIAMBIL DIBAGI RATA KE 8 SEMESTER,")  print(" JUMLAH MATA KULIAH HARUS KELIPATAN ANGKA 8")  print(" KETIK FILE .txt YANG BERISI INPUT MATA KULIAH")  print(" BESERTA PREREQUISITE-NYA (misal 'test1' tanpa tanda petik)")  #input file yang akan disusun mata kuliahnya  inputAwal = str(input(" Nama File : "))  fname = "../test/" + inputAwal + ".txt"  #Menyimpan daftar mata kuliah dengan memedulikan spasi (' ')  with open(fname,'r') as f:  courseList = []  for line in f:  word = []  for character in line:  if (character == ' '):  continue  if (character == ',' or character == '.'):  break  word.append(character)  courseList.append(func.convert(word))  #Menyimpan daftar mata kuliah tanpa memedulikan spasi (' ')  with open(fname,'r') as f:  courseList2 = []  for line in f:  word = []  for character in line:  if (character == ',' or character == '.'):  break  word.append(character)  courseList2.append(func.convert(word))  #Menyimpan daftar mata kuliah prerequisite tanpa memedulikan spasi (' ')  with open(fname, 'r') as f:  coursePlan = []  for line in f:  coursePlan.append(func.preqCourse(line))  #Cek apakah jumlah mata kuliah pada file adalah kelipatan 8  if (len(courseList) % 8 != 0): #Jika tidak  print("Jumlah mata kuliah bukan kelipatan 8! Silahkan ubah rencana mata kuliah Anda kembali.")  else: #Jika ya  courseFix = []  print()  print("MATA KULIAH YANG DAPAT DIAMBIL PER SEMESTER:")  #Memanggil prosedur topSort  TS.topSort(courseFix, coursePlan, courseList, courseList2) |

1. **Hasil Masukan dan Luaran**

|  |  |  |
| --- | --- | --- |
| No. | Masukan | Luaran |
| 1. | Representasi Graf: |  |
| 2. | Representasi Graf: |  |
| 3. | Representasi Graf: |  |
| 4. | Representasi Graf: |  |
| 5. | Representasi Graf: |  |
| 6. | Representasi Graf: |  |
| 7. | Representasi Graf: |  |
| 8. | Representasi Graf: |  |

1. **Alamat *drive***

Link Repository Github: https://github.com/epata/Tucil-1-Strategi-Algoritma---13519120

Link Drive Alternatif: <https://drive.google.com/drive/u/0/folders/1K8Je2A0MtOPB7cFfovHMxX-fpC1uHGYf>

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