**TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA**

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

Oleh

Muhammad Fawwaz Naabigh - 13519206



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2020/2021**

**Algoritma *Topological Sort* dan kaitannya dengan pendekatan *Decrease and Conquer***

***Source Code***

|  |
| --- |
| import re, time, os  def bacaFile(namafile):  '''  Membaca masukan dari file.txt lalu membaca tiap baris  dan mengembalikan array matakuliah yang juga berisi sekumpulan array (sub-array)  dengan elemen pertama adalah kode kuliah dan sisanya adalah kuliah prasyarat.  Sub-array tersebut adalah data per baris yang dibaca dari file.  [['kode kuliah1', 'prasyarat1', 'prasyarat2'], ['kode kuliah2']]  '''  try:  directory = os.path.abspath(f'../test/{namafile}.txt')  f = open(directory, "r")  lines = f.readlines()  courses = []  for line in lines:  course = []  x = re.sub(r'[^\w\s]', '', line)  y = re.sub(r'\n', '', x)  if y!="":  y = y.split(' ')  for matkul in y:  course.append(matkul)  courses.append(course)  return courses  except:  printFailed()  return quit()  def hitungPrasyarat(arrayOfCourses):  '''  Nilai yang dikembalikan adalah dictionary dengan key berupa nama course (simpul)  dan value adalah prasyarat (busur masuk).  {'kode kuliah1': ['prasyarat1', 'prasyarat2'], 'kode kuliah2': []}    Bila direpresentasikan dengan graf.  - key -> simpul  - value -> simpul yang masuk (busur) ke simpul key. berupa array of course ['C1', 'C2', 'C6'].  value berupa array kosong ([]) berarti course tersebut tidak memiliki prasyarat.  '''  courses = {}  for course in arrayOfCourses:  matkul = course[0]  prasyarat = course[1:]  courses[matkul] = prasyarat  return courses  def topSort(dictOfCourses):  '''  Nilai yang dikembalikan adalah array yang berisi subarray  dengan index array adalah semester - 1 dan subarray berisi course yang dapat diambil pada semester tersebut.    [['C1', 'C2'], ['C3']]  Misal, index ke-0 berisi ['C1', 'C2'],  berarti pada semester 1 mata kuliah yang dapat diambil adalah course C1 dan C2.  Begitu seterusnya.  '''  coursePerSemester = [] # Berisi course dengan semester = index+1  while(len(dictOfCourses)!=0):  selectedCourse = [] # Menampung course (simpul) dengan busur masuk = 0  for course, prereq in dictOfCourses.items():  if(len(prereq)==0):  selectedCourse.append(course)  coursePerSemester.append(selectedCourse)  # Menghapus course (simpul) dengan prasyarat (busur masuk) = 0  for c in selectedCourse:  del dictOfCourses[c]  # Menghapus busur yang keluar dari selectedCourse (course (simpul) dengan busur masuk = 0)  for course, prereq in dictOfCourses.items():  for matkul in selectedCourse:  if matkul in prereq:  prereq.remove(matkul)  return coursePerSemester  def printFailed():  '''  Dipanggil jika masukan salah,  bisa disebabkan jika file tidak terbaca (entah user salah input dan sebagainya)  atau jika isi file kosong.  '''  print("\n\tMasukan tidak terbaca.")  print("\tPastikan file Anda:")  print("\t - Berekstensi .txt")  print("\t - Memiliki nama file yang sesuai dengan yang Anda input. Nama file ditulis tanpa .txt")  print("\t - Berada pada directory yang sama dengan program ini")  print("\t - Tidak berisi file kosong dan memiliki format yang benar")  print("\t - Contoh format yang benar adalah")  print('''\t\tC1, C3.  C2, C1, C4.  C3.  C4, C1, C3.  C5, C2, C4.  ''')  print("\t Tiap baris berisi course dan diikuti prerequisite dari course tersebut")  print()  input("Press ENTER to exit")  def printHasil(arrayOfCourses):  '''  Dipanggil untuk menampilkan hasil ke layar  '''  if(len(topsort) == 0):  printFailed()  else:  print("\n\tMata kuliah yang dapat Anda ambil tiap semester adalah:")  for i in range(len(arrayOfCourses)):  print(f'\tSemester {i+1} : {", ".join(topsort[i])}')  print()  input("Press ENTER to exit")  a = input("Masukkan nama file Anda (tanpa ekstensi .txt): ")    topsort = topSort(hitungPrasyarat(bacaFile(a)))  printHasil(topsort) |

***Screenshot* hasil percobaan**

|  |  |
| --- | --- |
| Masukan | Luaran |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

*Link Source Code*

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