**TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA**

**SEMESTER II TAHUN 2020/2021**

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

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

Oleh

Arsa Daris Gintara – 13519037

A black and white drawing of a cartoon character

Description automatically generated with low confidence

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2020/2021**

# Algoritma *Topological Sort* dan kaitan pendekatan *Decrease and Conquer*

1. Pertam
2. Tes

# *Source Code* Program

*# load text dalam file*

def toGraph(*filename*):

  theString = []

  with open(filename,'r') as file:

    for line in file:

      theString.append(line.replace(',',' ').replace(', ', ' ')

      .replace('.', '').replace('\n', ''));

  return theString

*# membuat list daftar mata kuliah yang ada (unik) menggunakan set*

def nodeList(*theList*):

  theString = ''

  for i in theList:

    theString += i + ' '

  toSet = theString.split()

  result = list(set(toSet))

  result.sort()

  return result

*# merepresentasikan graph berarah dalam bentuk list 2 dimensi. Dalam setiap list*

*# Index pertama sebagai Matakuliah, indeks berikutnya sebagai prerequisite dari mata kuliah tersebut*

def graphInList(*theList*, *listOfNode*):

  result = []

  for every in theList:

    tmpEvery = every.split()

    tmp = []

    for inside in tmpEvery:

      for i in range(len(listOfNode)):

        if inside == listOfNode[i]:

          tmp.append(i)

    result.append(tmp)

  return result

*# pendekatan topological sort*

def topSort(*graphRepre*):

  theOrder = []

  deg0 = []

*# mencari mata kuliah yang memiliki derajat masuk 0 atau tidak memiliki prerequisite*

*# termasuk jika prerequisite sudah terpenuhi*

  for i in range(len(graphRepre)):

    if len(graphRepre[i]) == 1:

      deg0 += graphRepre[i]

*# menghapus mata kuliah yang berderajat masuk 0 dalam graf*

  if len(deg0) > 0:

    for j in range(len(deg0)):

      graphRepre.remove([deg0[j]])

      for every in graphRepre:

        if (len(every) > 1 and deg0[j] in every):

          every.remove(deg0[j])

    theOrder.append(deg0)

*# jika dalam 1 semester memiliki lebih dari satu mata kuliah yang tidak memiliki prerequisite*

  if (len(graphRepre) > 0):

    nextOrder = topSort(graphRepre)

    theOrder += nextOrder

  return theOrder

*# mencetak hasil sort*

def printResult(*resultList*, *listOfNode*):

  print("   ,-,--.   .=-.-.         ,-.--, ")

  print(" ,-.'-  \_\ /==/\_ /.--.-.  /=/, .' ")

  print("/==/\_ ,\_.'|==|, | \==\ -\/=/- /   ")

  print("\==\  \   |==|  |  \==\ `-' ,/    ")

  print(" \==\ -\  |==|- |   |==|,  - |    ")

  print(" \_\==\ ,\ |==| ,|  /==/   ,   \   ")

  print("/==/\/ \_ ||==|- | /==/, .--, - \  ")

  print("\==\ - , //==/. / \==\- \/=/ , /  ")

  print(" `--`---' `--`-`   `--`-'  `--`   ")

  print()

  for i in range(len(resultList)):

    print('Semester {}: '.format(i+1), *end*='')

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

      print('{}'.format(listOfNode[resultList[i][j]]), *end*='')

      if (j != len(resultList[i])-1):

        print(', ', *end*='')

    print()

*#main program*

if \_\_name\_\_ == "\_\_main\_\_":

  txtfile = input('Masukkan nama file yang terletak di folder /test/ (contoh: 1.txt) : ')

  source = 'test/'+txtfile

  try:

    open(source)

  except:

    source = '../test/'+txtfile

  try:

    open(source)

  except:

    source = 'Tucil2\_13519037/test/'+txtfile

  listFromFile = toGraph(source)

  listOfNode = nodeList(listFromFile)

  graphRepre = graphInList(listFromFile, listOfNode)

  result = topSort(graphRepre)

  printResult(result, listOfNode)

# *Screenshot* Program

|  |
| --- |
| Text  Description automatically generated |
| Text  Description automatically generated |
| Text  Description automatically generated |
| Text  Description automatically generated |
| Text  Description automatically generated |
| Text  Description automatically generated |
| Text  Description automatically generated |
| Text  Description automatically generated |

# Alamat Program

<https://github.com/arsa-dg/coursehouse>

# Checklist

|  |  |  |
| --- | --- | --- |
| 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 | √ |  |