

LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA
SEMESTER GENAP – 2020/2021
13519028 – HAFID ABI DANISWARA

1. PENJELASAN ALGORITMA TOPOLOGICAL SORT DAN KAITANYA DENGAN DECREASE AND CONQUER

a. Topological Sort

Topologi sort merupakan cara / algoritma untuk mengurutkan Directed Acyclic Graph. Dalam topologi sort ini merupakan pengurutan yang hasilnya tidak unik. Dalam algoritma ini graf diurutkan dengan cara memilih semua node yang tidak memiliki panah yang mengarah ke node tersebut, kemudian node tersebut ditampilkan dan node tersebut dihapus beserta dengan panah yang berasal dari node tersebut, kemudian proses ini dilakukan terus menerus sampai semua node ditampilkan atau tidak tersisa node lagi.

b. Penggunaan Decrease and Conquer dalam penyelesaian topological sort

Decrease and conquer merupakan algoritma yang mereduksi persoalan menjadi 2 sub masalah yang lebih kecil namun selanjutnya yang di proses hanya satu persoalan saja. Dalam varian decrease and conquer ada tiga tipe yaitu, decrease by constant, decrease by constant factor, dan decrease by variable size.

Pada persoalan ini lebih menggunakan decrease by variable size, karena pada pola ini setiap rekursif ukuran variable berkurang. Pada penerapan program ini pada setiap rekursi melakukan penghapusan node-node atau mata kuliah yang sudah tidak memiliki prequisite sampai ketemu basis dari prosedur topological sort. Basis pada persoalan ini adalah ketika sudah tidak ada upagraf lagi atau graf berbentuk sirkular dan tidak ada node yang bisa dihapus (untuk kasus ini akan menghasilkan error).

2. SOURCE CODE DALAM BAHASA PYTHON

a. Class untuk Graf

```
class Graf :
    def __init__(self,ArrayData):
        '''
        CLASS CONSTRUCTOR
        :param ArrayData: berisikan list of list hasil konversi dari txt menuju
        ke list of list
        '''
        self.data = self.__extractData(ArrayData)
        self.jmlhElmen = self.getJumlahData()

    def __extractData(self,data):
        '''
        FUNGSI UNTUK MERUBAH DATA MENJADI SEBUAH
        FORMAT GRAF (DICTIONARY BERISIKAN ID NODE DAN GRAF YANG MENGARAH KE
        NODE BERSANGKUTAN)
        :param data: BERUPA LIST OF LIST,FUNCTION INI PRIVATE
        :return: SEBUAH GRAF DALAM BENTUK DICTIONARY BERISIKAN ID NODE DAN
        GRAF YG MENGARAH KE NODE BERSANGKUTAN
        '''
        r = []
        for x in data:
            temp = {}
            if(len(x)>0):
                temp["id"] = x[0]
                temp["preq"] = []
                for y in range(1,len(x)):
                    temp["preq"].append(x[y])
                r.append(temp)
        return r

    def removeGraf(self,key):
        '''
        UNTUK MENGHAPUS GRAF YANG TIDAK MEMILIKI PREC YG MENGARAH DAN
        MENGHAPUS PREC DARI SEMUA
        NODE YANG BERASAL DARI GRAF YANG DIHAPUS
        :param key: ID Node yang ingin dihapus
        :return: mengembalikan boolean apakah ID nodal yang di input
        diparameter apakah berhasil dihapus dari graf
        '''
        newdata = []
        status = True
        for x in self.data:
            if(x["id"]==key and len(x["preq"])>0):
                status = 1==2
            elif(x["id"]!=key):
                if(key in x["preq"]):
                    x["preq"].remove(key)
                    newdata.append(x)
        if(status):
            self.data = newdata
        return status

    def getNoPreq(self):
        '''
        UNTUK MENDAPATKAN GRAF YANG TIDAK MEMILIKI PREC / PANAH YANG
        MENGARAH KE SUATU NODE
        :return: LIST OF NO PREC NODE
        '''
        ret = []
        for x in self.data:
            if(len(x["preq"])==0):
                ret.append(x["id"])
        return ret

    def getJumlahData(self):
        '''
        :return: mengembalikan jumlah nodal
        '''
        return len(self.data)
```

b. Function untuk membaca file dari txt dengan parameter nama file

```
def readfiles(flname):
    '''
    memproses file txt menjadi sebuah list of list
    :param flname: nama file txt yang terdapat pada folder test
    :return: list of list hasil pemrosesan file txt
    '''
    flname = flname + ".txt" if ".txt" in flname else flname + ".txt"
    txt = open("test/" + flname, 'r').readlines()
    for x in txt:
        if "." not in x: return []
    data = [x.strip().replace('.', '').replace(" ", "").split(sep=",") for x
in txt]
    return data
```

c. Function untuk melakukan topologi sort dengan parameter objek graf

```
def runTopologiSort(objekGraf, semester=1):
    '''
    melakukan topologi sort dengan menghapus node/matkul yang udah tidak
    memiliki prequisite lagi scr rekursif
    :param objekGraf: objek dari class graf
    :param semester: semester pengambilan, default = 1
    :return: merupakan prosedur rekursif, sehingga tidak ada return
    '''
    if objekGraf.getJumlahData() == 0:
        if semester == 1:
            print("Tidak ada data")
        else:
            print("topologi sort selesai!")
    else:
        noPreq = objekGraf.getNoPreq()
        if (len(noPreq) == 0):
            print("Graf mengandung sirkular sehingga tidak bisa lanjutkan
topologi sort")
        else:
            print("Semester "+str(semester)+": ", end='')
            for id in noPreq:
                print(str(id) + " ", end='')
                objekGraf.removeGraf(id)
            print()
            runTopologiSort(objekGraf, semester+1)
```

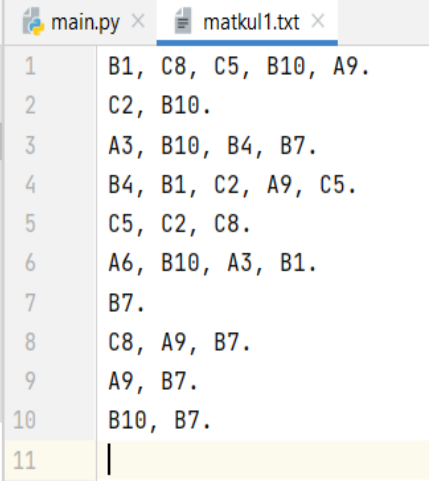
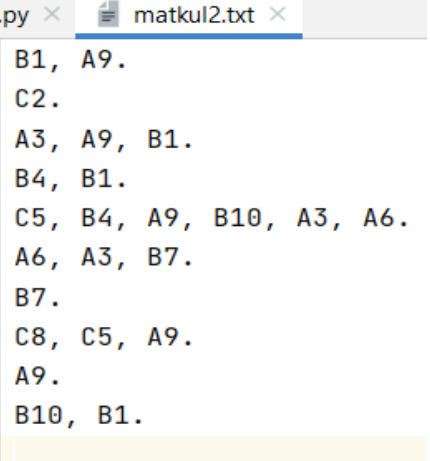
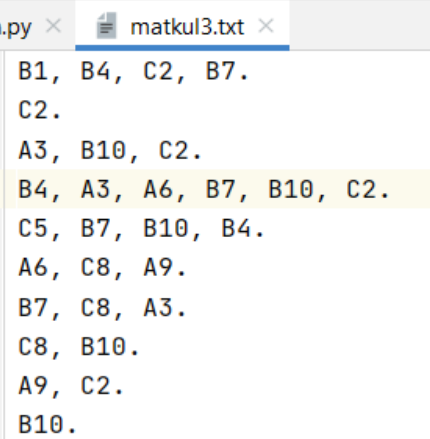
d. Function validate untuk memastikan txt sesuai dengan format dengan parameter list hasil output dari function readfiles

```
def validate(listMatkul):
    '''
    memvalidasi apakah txt sudah sesuai dengan format yang ditentukan asisten
    :param listMatkul: list of list hasil pembacaan file txt
    :return: boolean apakah valid atau tidak
    '''
    if (len(listMatkul) == 0): return False
    z = ""
    for x in listMatkul:
        z = z + str(x)
    z = z.replace("[", "").replace("]", "").replace("'", "").replace(",",
    "").lower()
    x = 0
    while (x < len(z)):
        if not ( (z[x] >= 'a' and z[x] <= 'z') or (z[x] >= '0' and z[x] <= '9') or
z[x] == " " or z[x] == "-" or z[x] == "_"):
            return False
        else: x = x + 1
    return True
```

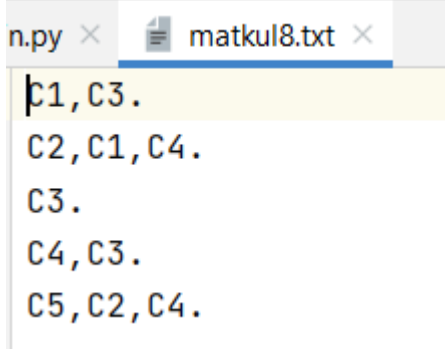
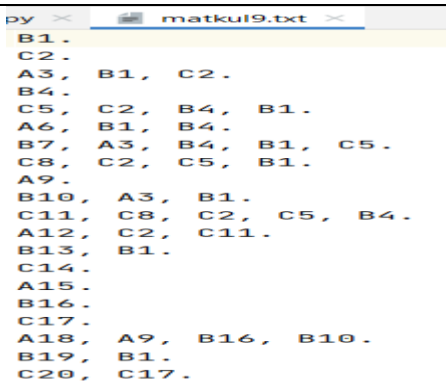
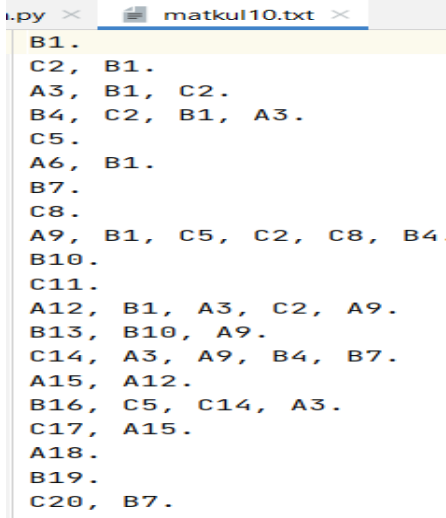
e. Main program

```
if __name__ == "__main__":
    namafile = input("masukkan nama file test = ")
    data = readfiles(namafile)
    if validate(data):
        runTopologiSort(Graf(data))
    else:
        print("data tidak valid, pastikan format di file txt telah benar dan
diakhir baris dikasih titik (format kelas hanya mengandung alfabet, numerik,-
,dan _) contoh: Stima_01")
```

3. SCREENSHOT DARI INPUT DAN OUTPUT

No	Input	Output
1	 <pre> 1 B1, C8, C5, B10, A9. 2 C2, B10. 3 A3, B10, B4, B7. 4 B4, B1, C2, A9, C5. 5 C5, C2, C8. 6 A6, B10, A3, B1. 7 B7. 8 C8, A9, B7. 9 A9, B7. 10 B10, B7. 11 </pre>	<p>masukkan nama file test = <i>matkul1</i></p> <p>Semester 1: B7</p> <p>Semester 2: A9 B10</p> <p>Semester 3: C2 C8</p> <p>Semester 4: C5</p> <p>Semester 5: B1</p> <p>Semester 6: B4</p> <p>Semester 7: A3</p> <p>Semester 8: A6</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>
2	 <pre> 1 B1, A9. 2 C2. 3 A3, A9, B1. 4 B4, B1. 5 C5, B4, A9, B10, A3, A6. 6 A6, A3, B7. 7 B7. 8 C8, C5, A9. 9 A9. 10 B10, B1. 11 </pre>	<p>masukkan nama file test = <i>matkul2.txt</i></p> <p>Semester 1: C2 B7 A9</p> <p>Semester 2: B1</p> <p>Semester 3: A3 B4 B10</p> <p>Semester 4: A6</p> <p>Semester 5: C5</p> <p>Semester 6: C8</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>
3	 <pre> 1 B1, B4, C2, B7. 2 C2. 3 A3, B10, C2. 4 B4, A3, A6, B7, B10, C2. 5 C5, B7, B10, B4. 6 A6, C8, A9. 7 B7, C8, A3. 8 C8, B10. 9 A9, C2. 10 B10. 11 </pre>	<p>masukkan nama file test = <i>matkul3.txt</i></p> <p>Semester 1: C2 B10</p> <p>Semester 2: A3 C8 A9</p> <p>Semester 3: A6 B7</p> <p>Semester 4: B4</p> <p>Semester 5: B1 C5</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>

4	<pre> .py x matkul4.txt x B1, B10, B4, B7, A6. C2. A3, C2, B7. B4. C5. A6, C8, B4. B7, A9. C8, B10, A3. A9. B10, C2, A3, A9. </pre>	<p>masukkan nama file test = <i>matkul4.txt</i></p> <p>Semester 1: C2 B4 C5 A9</p> <p>Semester 2: B7</p> <p>Semester 3: A3</p> <p>Semester 4: B10</p> <p>Semester 5: C8</p> <p>Semester 6: A6</p> <p>Semester 7: B1</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>
5	<pre> .py x matkul5.txt x B4, A3. C5. A6, C2, C5. B7, A6. C8. A9. B10. C11, A3, A9. A12. B13, C8, C5, A9, B10, B4. C14, A12, B7, A6, B13. A15. B16. C17. A18, B16, A12, C11. B19, B7. C20. </pre>	<p>masukkan nama file test = <i>matkul5.txt</i></p> <p>Semester 1: B1 A3 C5 C8 A9 B10 A12 A15 B16 C17 C20</p> <p>Semester 2: C2 B4 C11</p> <p>Semester 3: A6 B13 A18</p> <p>Semester 4: B7</p> <p>Semester 5: C14 B19</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>
6	<pre> 9028.py x matkul6.txt x B1. C2. A3. B4, A3, C2. C5, C2. A6, C5. B7. C8, B4, B1. A9, B1, A3, C5, C2, B4. B10. C11. A12, B7. B13, B10, A12, B7, B1. C14. A15, C8, A6, C2, C14, A9, B7, C11, A3, B13. B16. C17. A18, B1, C2, C17, A9, A15, B16, B4, A6, A12. B19, C14, A9, A18, B13, C17, C5, A12, A3, B1. C20. </pre>	<p>masukkan nama file test = <i>matkul6</i></p> <p>Semester 1: B1 C2 A3 B7 B10 C11 C14 B16 C17 C20</p> <p>Semester 2: B4 C5 A12</p> <p>Semester 3: A6 C8 A9 B13</p> <p>Semester 4: A15</p> <p>Semester 5: A18</p> <p>Semester 6: B19</p> <p>topologi sort selesai!</p>
7	<pre> .py x matkul7.txt x B1, C5, B7, C2. C2, B7. A3, A9, B1, B10, B10. B4. C5. A6, A3, A9, A3, B1. B7, B4, C5. C8, B4, A6, A9, A3. A9, A3, A6, A3. B10, A9, B4, B7, B7, A9. </pre>	<p>masukkan nama file test = <i>matkul7.txt</i></p> <p>Semester 1: B4 C5</p> <p>Semester 2: B7</p> <p>Semester 3: C2</p> <p>Semester 4: B1</p> <p>Graf mengandung sirkular sehingga tidak bisa lanjutkan topologi sort</p> <p>Process finished with exit code 0</p>

8		<p>masukkan nama file test = <i>matkul8.txt</i></p> <p>Semester 1: C3</p> <p>Semester 2: C1 C4</p> <p>Semester 3: C2</p> <p>Semester 4: C5</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>
9		<p>masukkan nama file test = <i>matkul9.txt</i></p> <p>Semester 1: B1 C2 B4 A9 C14 A15 B16 C17</p> <p>Semester 2: A3 C5 A6 B13 B19 C20</p> <p>Semester 3: B7 C8 B10</p> <p>Semester 4: C11 A18</p> <p>Semester 5: A12</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>
10		<p>masukkan nama file test = <i>matkul10.txt</i></p> <p>Semester 1: B1 C5 B7 C8 B10 C11 A18 B19</p> <p>Semester 2: C2 A6 C20</p> <p>Semester 3: A3</p> <p>Semester 4: B4</p> <p>Semester 5: A9</p> <p>Semester 6: A12 B13 C14</p> <p>Semester 7: A15 B16</p> <p>Semester 8: C17</p> <p>topologi sort selesai!</p> <p>Process finished with exit code 0</p>

4. LINK SOURCE CODE

<https://github.com/hafidabid/tucil2-stima-13519028>

5. CHECKLIST

Poin	Ya	Tidak
1. Program berhasil dikompilasi	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2. Program berhasil <i>running</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3. Program dapat menerima berkas input dan menuliskan output.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4. Luaran sudah benar untuk semua kasus input.	<input checked="" type="checkbox"/>	<input type="checkbox"/>