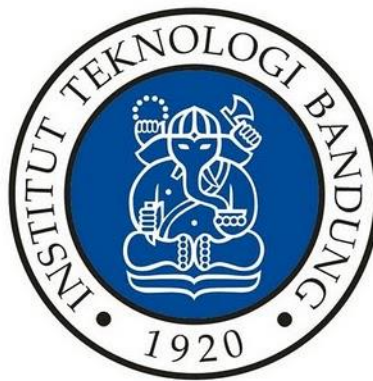


Laporan Tugas Kecil 2 IF2211 Strategi Algoritma
Penyusunan Rencana Kuliah dengan *Topological Sort*
(Penerapan *Decrease and Conquer*)

Oleh :

Muhammad Fikri. N

13519069



INSTITUT TEKNOLOGI BANDUNG
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
TEKNIK INFORMATIKA
2020/2021

BAB I

DESKRIPSI MASALAH

Penyusunan rencana pengambilan mata kuliah adalah proses pengambilan mata kuliah yang akan dilaksanakan mahasiswa pada semester yang akan datang. Mata kuliah dapat memiliki prerequisite mata kuliah yang harus diambilnya sebelum mengambil mata kuliah tersebut. Permasalahan penyusunan rencana pengambilan mata kuliah dapat dilakukan dengan memanfaatkan algoritma *Decrease and Conquer*. *Topological sort* merupakan salah satu implementasi untuk menyelesaikan permasalahan penyusunan rencana pengambilan mata kuliah

Contoh :

1. Input file teks berisi :

C1, C3.

C2, C1, C4.

C3.

C4, C1, C3.

C5, C2, C4.

Solusinya adalah :

Semester 1 : C3.

Semester 2 : C1.

Semester 3 : C4.

Semester 4 : C2 .

Semester 5 : C5.

2. Input file teks berisi :

C1, C2.

C2.

Solusinya adalah :

Semester 1 : C2.

Semester 2 : C1.

3. Input file teks berisi :

C1, C2.

Solusinya adalah :

Tidak ada urutan yang mungkin.

Spesifikasi :

Membuat aplikasi sederhana yang dapat menyusun rencana pengambilan kuliah, dengan memanfaatkan algoritma Decrease and Conquer dalam bahasa C/C++/Java/Python. Penyusunan rencana kuliah diimplementasikan dengan menggunakan pendekatan Topological Sorting. Batasan untuk aplikasi ini adalah untuk setiap semester tidak terdapat batasan pengambilan mata kuliah yang bisa diambil dan pembatasan 8 semester dalam penyusunan rencana mata kuliah. Namun, untuk batasan 8 semester, aplikasi ini memberikan opsi pilihan untuk tetap menampilkan hasil penyusunan rencana mata kuliah atau tidak.

Input :

File teks dengan format di bawah ini :

```
<kode_kuliah_1>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>.  
<kode_kuliah_2>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>.  
<kode_kuliah_3>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>, <kode kuliah prasyarat - 4>.  
<kode_kuliah_4>.  
.  
.  
.
```

Gambar 1. Format File Teks untuk Masukan Daftar Kuliah

<Sumber : File Spek Tupil 2>

Contoh isi file teks:

C1, C3.

C2, C1, C4.

C3.

C4, C1, C3.

C5, C2, C4.

Output:

Semester 1 : C3.

Semester 2 : C1.

Semester 3 : C4.

Semester 4 : C2 .

Semester 5 : C5.

BAB II

ALGORITMA TOPOLOGICAL SORT

Topological sorting merupakan suatu algoritma yang digunakan untuk memproses suatu Directed Acyclic Graph (DAG) sehingga mendapatkan keterurutan dari setiap simpul.

Bahasa pemrograman yang digunakan adalah python.

Pemecahan masalah penyusunan rencana kuliah menggunakan algoritma topological sort :

1. Baca file teks kemudian hasilnya disimpan ke array yang berisi matkul beserta *prerequisite* nya ([elm_1,elm_2,elm_3, ... , elm_N]) dengan N merupakan jumlah baris dari file teks dan untuk setiap elm_i dengan $1 \leq i \leq N$ akan berbentuk [matkul,prereq_1,prereq_2, ... , prereq_M] sebut saja array Matkul. M mungkin 0 yang berarti suatu matkul tidak memiliki prerequisite.
2. Dibentuk representasi graf menggunakan tipe data dictionary. dict dengan key yaitu matkul dan value berbentuk dictionary juga. value memiliki dua key yaitu N_preq dan Arr_preq. N_preq yaitu banyaknya *prerequisite* yang blm diambil dan Arr_preq adalah array dari *prerequisite*.
3. Dibentuk array of array dgn nama Semester yang akan menampung mata kuliah tiap semester Semester = [[semester1],[semester2],[semester3], ... , [semester40]] Dalam hal ini, diinisialisasi maksimal menampung 40 semester atau dengan indeks 0 sampai 39. Pada penulisan hasil, akan dilakukan pengecekan agar maksimal 8 semester saja. Namun, dapat memilih opsi tetap menampilkan hasil penyusunan jika ingin melihat hasil walaupun lebih dari 8 semester.
4. Setelah itu, dilakukan pengecekan apakah ada N_preq yang bernilai 0. (Jika tidak berarti mata kuliah dan *prerequisite* nya tidak berbentuk Directed Acyclic Graph)
5. Jika langkah 4 = True, maka akan dilakukan proses selama dictionary tdk kosong ({ }) dan langkah 4 = True.
6. Proses nya adalah dilakukan iterasi untuk semua elemen pada Matkul, jika terdapat dictionary yang key nya adalah elm_i[0] dengan $1 \leq i \leq N$ atau dengan kata lain adalah elemen matkul (refer ke poin 1) dan key dari valuenya adalah N_preq yang bernilai 0. Maka matkul tersebut akan dimasukkan ke array temporary untuk menampung matkul yang sudah diambil. Setelah itu, dilakukan penghapusan *record* dictionary yang memiliki key elm_i[0] tadi yang telah diambil. Kemudian, juga dilakukan pengurangan N_preq untuk setiap *record* dictionary yang *prerequisite* nya adalah elm_i[0] tersebut dan penghapusan array Matkul yang matkul = elm_i[0].
7. Lakukan pengecekan pada langkah 5 hingga langkah 5 = False
8. Setelah itu, diperoleh array Semester yang telah berisi matkul-matkul hasil penyusunan matkul per semester. Jika semester tidak melebihi 8 semester maka akan langsung diberikan hasilnya, jika tidak maka akan ada pilihan apakah tetap akan ditampilkan atau tidak dan kemudian memberikan saran untuk melakukan pengaturan pilihan matkul.
9. Pada langkah 8, jika semester tidak melebihi 8 maka akan dilakukan pengecekan jika dict kosong ({ }) berarti semua matkul dapat diambil dan jika dictionary tdk memiliki matkul yang N_preq sama dengan 0 berarti proses berhenti di tengah jalan / tidak memasuki proses sama sekali (tidak terdapat lagi DAG sehingga tidak dapat diproses).
10. Setiap testcase dengan nama filename akan memberikan output file dengan format filename_sol.txt pada directory (./test/solution).

Hubungan algoritma topological sort dengan pendekatan algoritma *decrease and conquer* adalah dengan penghapusan vertex yang telah diambil dan pengurangan N_preq / N_masuk jika berhubungan dengan vertex yang dihapus tadi. Varian *decrease and conquer* yang digunakan pada topological sort adalah *decrease by a constant*

BAB III

SOURCE PROGRAM

File 13519069_util.py

```
'''
Author : Muhammad Fikri. N
NIM : 13519069
Topik : Topological Sort
'''

# Untuk menghapus string kosong
def delete_arr_empty(arr):
    while (" " in arr):
        arr.remove(" ")
    return arr

# Untuk menghapus elemen arr / matkul (berisi matkul beserta prereq) yang
telah diambil (arrTakes)
def delete_arr(arr, arrTakes):
    while (len(arrTakes) > 0):
        for content in arr:
            course = content[0]
            for take in arrTakes:
                if course == take:
                    arr.remove(content)
                    arrTakes.remove(take)

    return arr

# Untuk menghapus record dictionary yang memiliki kunci = key
def delete_dict(dict, key):
    for course in dict.keys():
        if course == key:
            dict.pop(key)
    return dict

# Untuk mengurangi banyaknya prerequisite yang blm diambil
def minus_Npreq(dict, arrTakes):
    for course, info_course in dict.items():
        for take in arrTakes:
            if take in info_course['Arr_preq']:
                info_course["N_preq"] -= 1

# Untuk mengetahui apakah ada matkul yang banyak prerequisite yang blm
diambil = 0
def is_Npreq_have_zero(dict):
    bool = False
    for course, info_course in dict.items():
        if info_course["N_preq"] == 0:
            return True

    return bool

# Untuk convert array to string
def arrayToString(arr):
    string = ""
    for i in range(len(arr)):
        if (i < (len(arr)-1)):
            string += arr[i]
            string += ", "
        else:
```

```

        string += arr[i]
        string += "."
    return string

```

File 13519069_main.py

```

'''
Author : Muhammad Fikri. N
NIM : 13519069
Topik : Topological Sort
'''
import os
ut = __import__('13519069_util')

print("
print("
print("
print("
print("
print("
print("

-----")
print("| Silahkan input nama file |")
print("| (eg : tc) |")
print("| (Tanpa ekstensi file) |")
print("-----")

print()

filename = input("Input nama file : ")
path = "../test/" + filename + ".txt"

isfile = os.path.isfile(path)

# Melakukan pengecekan apakah filename terdapat pada folder test atau tidak
while (not isfile):
    filename = input("Ulangi input nama file : ")
    path = "../test/" + filename + ".txt"
    isfile = os.path.isfile(path)

test = open(path, 'r')

temp = test.read().splitlines() # baca file teks (dengan readlines yang
sekaligus hapus \n)

# delete string kosong (ketika file .txt memiliki banyak enter tp tidak ada
record)
ut.delete_arr_empty(temp)

temps = []

for i in range(len(temp)):
    elm = temp[i].replace('.', '')
    elm = elm.replace(" ", "")
    temps.append(elm)

arr = []

for i in range(len(temps)):
    # mengubah array of string menjadi array of array string dgn delimiter

```

```

koma (,)
    content = temps[i].split(',')

    arr.append(content)

dict = {}

for i in range(len(arr)):
    N_preq = (len(arr[i][1:len(arr[i])])) # N_preq adalah banyaknya
prerequisite yang blm diambil
    Arr_preq = (arr[i][1:len(arr[i])])
    dict[arr[i][0]] = { # Key = Matkul
        "N_preq": N_preq, # Value = dictionary (key1 = N_preq
        (banyak prerequisite yang blm diambil)
        "Arr_preq": Arr_preq # key2 = Arr_preq (array dari
        matkul prerequisite)
    }

semester = [[] for i in range(40)] # Asumsi maksimal 40 semester yang dapat
ditampung

temporary = []
i = -1

print()
print("-----")
print("| Hasilnya adalah |")
print("| ↓ ↓ ↓ ↓ ↓ ↓ |")
print("-----")
print()

name_output = filename + "_sol.txt"
outfile = open("../test/solution/" + name_output, 'w')

if ut.is_Npreq_have_zero(dict):
    while (dict != {} and ut.is_Npreq_have_zero(dict)):
        i += 1

        # untuk mengulang array temporary kembali kosong (berisi matkul
yang diambil)
        temporary = []

        for content in arr:
            if dict[content[0]]["N_preq"] == 0:
                take = content[0]
                semester[i].append(take)
                temporary.append(take)

                # Menghapus course yang telah diambil pada dict
                ut.delete_dict(dict, take)

        ut.minus_Npreq(dict, temporary)
        ut.delete_arr(arr, temporary)

semester = [x for x in semester if x] # Untuk menghapus empty array dari
sebuah array

if len(semester) <= 8:
    if (dict == {}):
        for i in range(len(semester)):
            print("Semester " + str(i+1) + ": ", end = "")

```

```

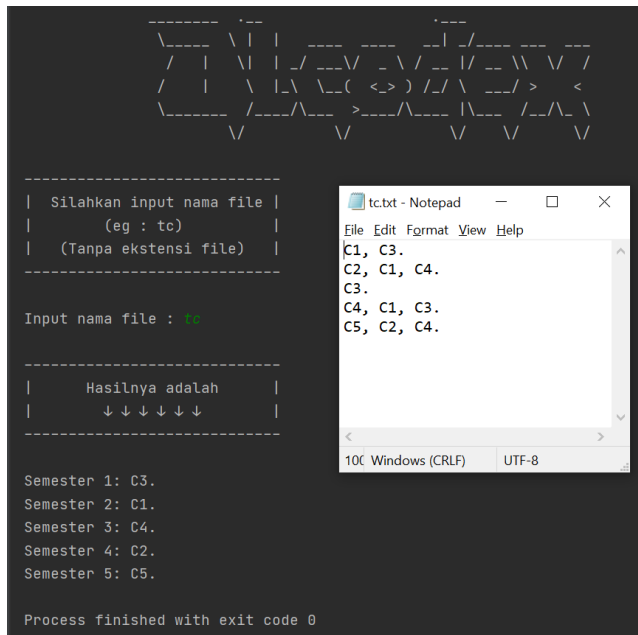
        outfile.write("Semester " + str(i+1) + ": ")
        print(ut.arrayToString(semester[i]))
        outfile.write(ut.arrayToString(semester[i]))
        outfile.write("\n")
    elif (not ut.is_Npreq_have_zero(dict)):
        for i in range(len(semester)):
            print("Semester " + str(i+1) + ": ", end="")
            outfile.write("Semester " + str(i + 1) + ": ")
            print(ut.arrayToString(semester[i]))
            outfile.write(ut.arrayToString(semester[i]))
            outfile.write("\n")
        print("Tidak ada urutan yang mungkin lagi")
        outfile.write("Tidak ada urutan yang mungkin lagi\n")
        print("Pastikan kuliah dan prerequisite nya berupa Directed Acyclic
Graph (DAG)")
        outfile.write("Pastikan kuliah dan prerequisite nya berupa Directed
Acyclic Graph (DAG)\n")
        print("Silahkan atur pilihan matkul kamu lagi, FIGHTING (~▽~)~")
        outfile.write("Silahkan atur pilihan matkul kamu lagi, FIGHTING
^_^")
else:
    print("Hasil proses melebihi 8 semester !!!")
    outfile.write("Hasil proses melebihi 8 semester !!!\n")
    print("Tidak boleh melebihi 8 semester :(")
    outfile.write("Tidak boleh melebihi 8 semester :(\n")
    print("Apakah tetap akan ditampilkan atau tidak perlu ?")
    print("-----")
    print("|           Opsi input           |")
    print("|           1. YA                 |")
    print("|           2. TIDAK              |")
    print("-----")

    pil = int(input("Input pilihan : "))
    while (pil != 1 and pil != 2):
        pil = int(input("Ulangi input pilihan, pastikan pilihan 1 atau 2 :
"))
    if pil == 1:
        for i in range(len(semester)):
            print("Semester " + str(i + 1) + ": ", end="")
            outfile.write("Semester " + str(i + 1) + ": ")
            print(ut.arrayToString(semester[i]))
            outfile.write(ut.arrayToString(semester[i]))
            outfile.write("\n")
        print("\nJangan lupa atur pilihan matkul kamu lagi, FIGHTING (~▽
~)~")
        outfile.write("\nJangan lupa atur pilihan matkul kamu lagi,
FIGHTING ^_^")
    else:
        print("\nSilahkan atur pilihan matkul kamu lagi, FIGHTING (~▽~)
~")
        outfile.write("\nSilahkan atur pilihan matkul kamu lagi, FIGHTING
^_^")

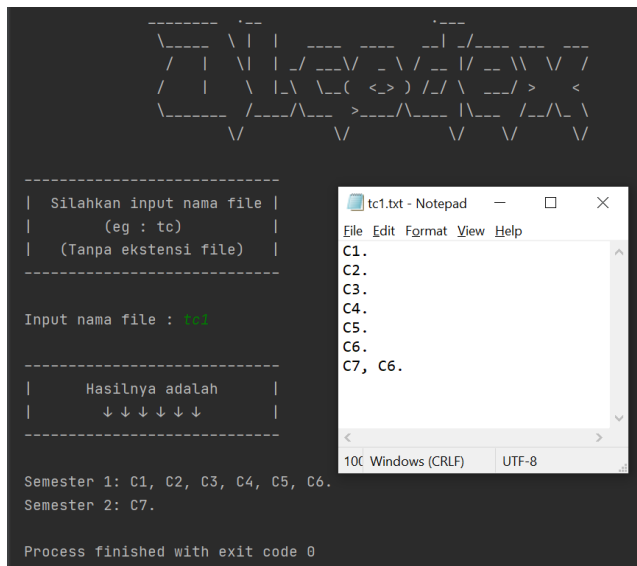
```


SCREENSHOOT I/O

Test Case 1 (File tc.txt)



Test Case 2 (File tc1.txt)



Test Case 3 (File tc2.txt)

[illegible]

Test Case 4 (File tc3.txt)

[illegible]

Test Case 5 (File tc4.txt)

[illegible]

Test Case 6 (File tc5.txt)

```

      _-_-_   _-_-_   _-_-_   _-_-_   _-_-_   _-_-_   _-_-_
     /  \  /  \  /  \  /  \  /  \  /  \  /  \  /  \  /  \
    /    \|    \|    \|    \|    \|    \|    \|    \|    \|
   /      \|      \|      \|      \|      \|      \|      \|
  /        \|        \|        \|        \|        \|        \|
 /          \|          \|          \|          \|          \|
/            \|            \|            \|            \|
\            /            /            /            /
 \          /          /          /          /          /
  \        /        /        /        /        /        /
   \|      /       /       /       /       /       /
    \|    /      /      /      /      /      /      /
     \|  /      /      /      /      /      /      /
      \|/      /      /      /      /      /      /

```

| Silahkan input nama file |
| (eg : tc) |
(Tanpa ekstensi file)

Input nama file : **tc5**

| Hasilnya adalah |
↓↓↓↓↓↓

Semester 1: C5, C7, C3.
Semester 2: C11, C8.
Semester 3: C2, C9, C10.

Process finished with exit code 0

The screenshot shows a Windows Notepad window titled "tc5.txt - Notepad". The menu bar includes File, Edit, Format, View, and Help. The text area contains the following output:
C5.
C7.
C3.
C11, C5, C7.
C8, C7, C3.
C2, C11.
C9, C8, C11.
C10, C3, C11.
At the bottom of the window, there are two tabs: "10(Windows (CRLF)" and "UTF-8".

Test Case 7 (File tc6.txt)

```

-----
|          Silahkan input nama file          |
|          (eg : tc)                          |
|          (Tanpa ekstensi file)              |
|-----|
Input nama file : tc

-----
|          Hasilnya adalah                    |
|          ↓ ↓ ↓ ↓ ↓ ↓                      |
|-----|

Semester 1: C_M, C_N, C_P.
Semester 2: C_Q, C_O.
Semester 3: C_S.
Semester 4: C_R.
Semester 5: C_U, C_Y.
Semester 6: C_T, C_V.
Semester 7: C_X, C_W.
Semester 8: C_Z.

Process finished with exit code 0

```

Test Case 8 (File tc7.txt)

```

      \_____/
    /   |   \   /___\   /___\   /___\
  /    / \   /___V___ \   /___ \___ V___ /
 /    /   \   /___( <_> )___/___ \___ >   <
 \_____/___\___/___>___\___/___ \___/___ \___
          V               V               V
-----
| Silahkan input nama file |
|         (eg : tc)        |
| (Tanpa ekstensi file)   |
|-----|
Input nama file : tc7

-----
| Hasilnya adalah       |
|      ↓ ↓ ↓ ↓ ↓       |
|-----|

Semester 1: C_M, C_N, C_P.
Semester 2: C_Q, C_O.
Semester 3: C_S.

Tidak ada urutan yang mungkin lagi
Silahkan atur pilihan matkul kamu lagi, FIGHTING (~▽~)~

Process finished with exit code 0

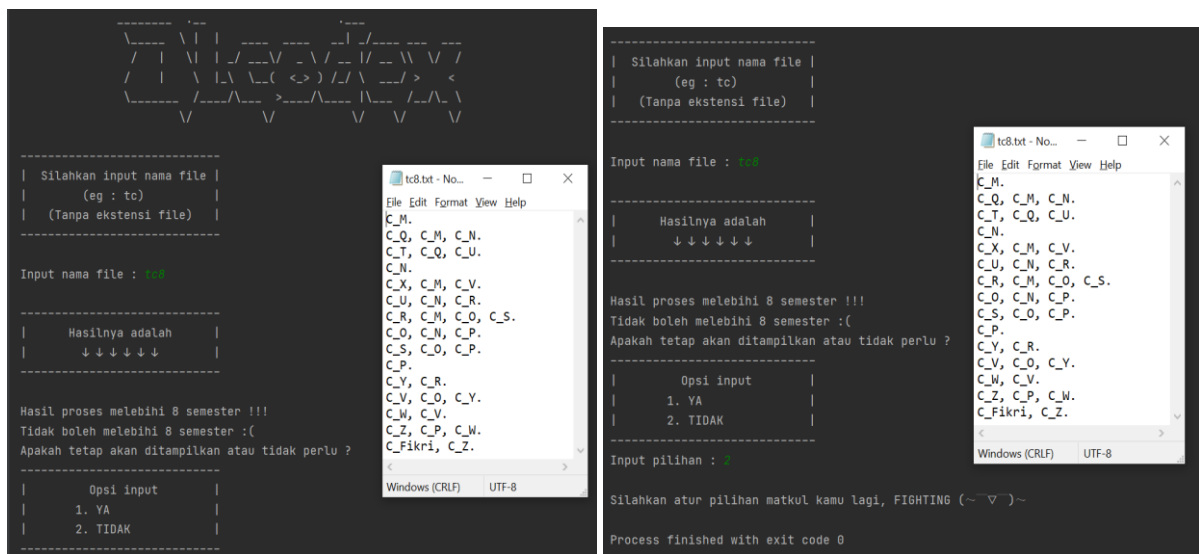
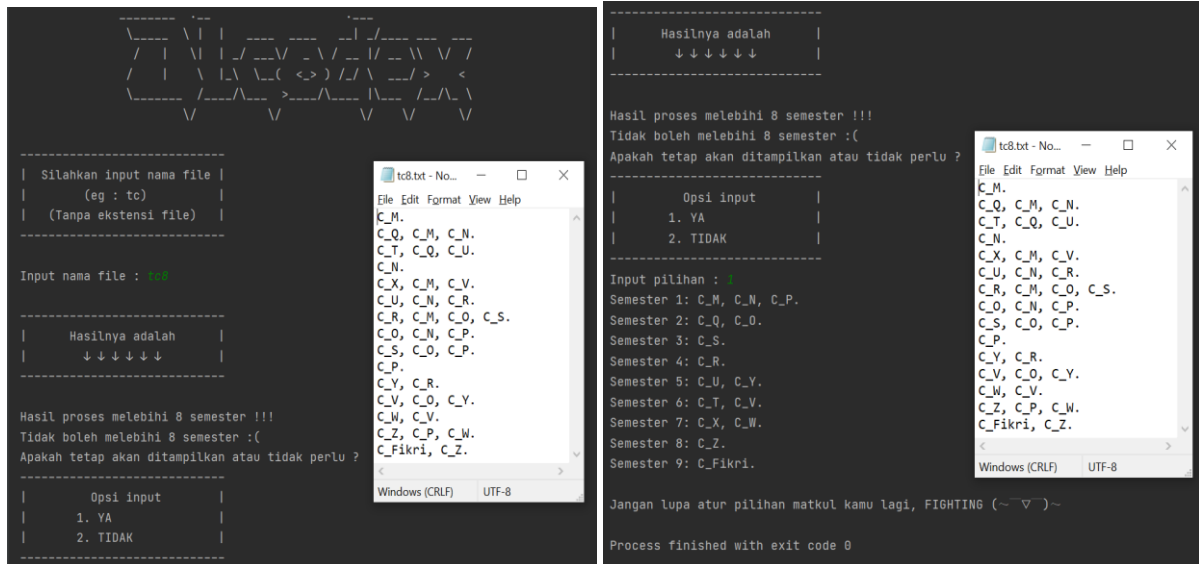
```

The screenshot shows a Notepad window titled "t07.txt - Notepad". The menu bar includes File, Edit, Format, View, and Help. The text content lists several course codes grouped by semester:

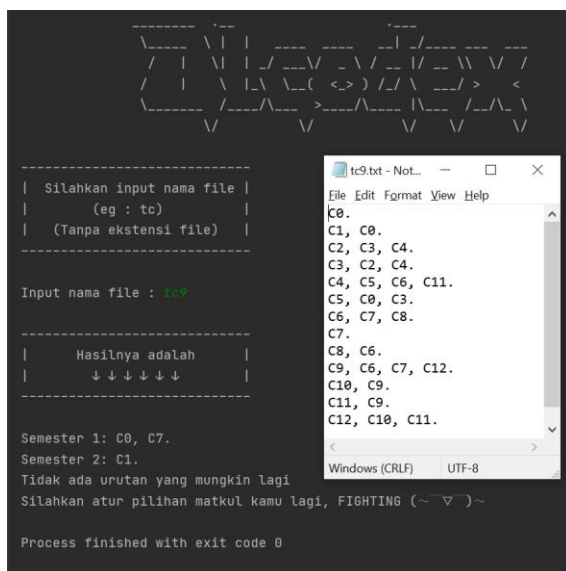
- C_M.
- C_Q, C_M, C_N.
- C_T, C_Q, C_U.
- C_N.
- C_X, C_M, C_V.
- C_U, C_N, C_R.
- C_R, C_M, C_O, C_S, C_Z.
- C_O, C_N, C_P.
- C_S, C_O, C_P.
- C_P.
- C_Y, C_R.
- C_V, C_O, C_Y.
- C_W, C_V.
- C_Z, C_P, C_W.

At the bottom of the Notepad window, there are three tabs or status indicators: "100%", "Windows (CRLF)", and "UTF-8".

Test Case 9 (File tc8.txt)



Test Case 10 (File tc9.txt)



BAB V
CEK LIST

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

Link Github : <https://github.com/mfikrin/TopSort>