**LAPORAN**

**Tugas Kecil 2 IF2121 Strategi Algoritma**

**DISUSUN OLEH**

M. Reyhanullah Budiaman 13519045

**INSTITUT TEKNOLOGI BANDUNG**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**TEKNIK INFORMATIKA**

**2020/2021**

**BAB I**

**Deskripsi Masalah**

Pada tugas kali ini, mahasiswa diminta **membuat aplikasi sederhana** yang dapat menyusun rencana pengambilan kuliah, dengan memanfaatkan algoritma **Decrease and Conquer**. Penyusunan Rencana Kuliah diimplementasikan dengan menggunakan pendekatan *Topological Sorting*. Berikut akan dijelaskan tugas yang dikerjakan secara detail.

1. Aplikasi akan menerima daftar mata kuliah beserta prasyarat yang harus diambil seorang mahasiswa sebelum mengambil mata kuliah tersebut. Daftar mata kuliah tersebut dituliskan dalam suatu file teks dengan format:

<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

Sebuah kode\_kuliah mungkin memiliki nol atau lebih prasyarat kuliah. Kode\_kuliah bisa diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil di semester sebelumnya (tidak harus 1 semester sebelumnya). Asumsi semua kuliah bisa diambil di sembarang semester, baik semester ganjil maupun semester genap.

2. Dari file teks yang telah diterima, ditentukan kuliah apa saja yang bisa diambil di semester 1, semester 2, dan seterusnya. Sebuah kuliah tidak mungkin diambil pada semester yang sama dengan prerequisitenya. Untuk menyederhanakan persoalan, tidak ada Batasan banyaknya kuliah yang bisa diambil pada satu semester.

**Pendekatan Topological Sorting**

1. a. Dari graf (DAG) yang terbentuk, hitung semua derajat-masuk (*in-degree*) setiap simpul, yaitu banyaknya busur yang masuk pada simpul tersebut. Pada contoh kasus di Gambar 2, maka derajat-masuk tiap simpul adalah sebagai berikut.

C1 : 1

C2 : 2

C3 : 0

C4 : 2

C5 : 2

1. b. Pilih sembarang simpul yang memiliki derajat-masuk 0. Pada kasus Gambar 2, pilih simpul C3.
2. c. Ambil simpul tersebut, dan hilangkan simpul tersebut beserta semua busur yang keluar dari simpul tersebut pada graf, dan kurangi derajat simpul yang berhubungan dengan simpul tersebut dengan 1.

Setelah simpul C3 dipilih, maka derajat simpul yang lain menjadi sebagai berikut.

C1 : 0

C2 : 2

C4 : 1

C5 : 2

Ulangi langkah (b) dan (c) hingga semua simpul pada DAG terpilih. Untuk kasus pada Gambar 2, setelah simpul terakhir dipilih rencana kuliah yang dihasilkan adalah sebagai berikut.

Semester I : C3

Semester II : C1

Semester III : C4

Semester IV : C2

Semester V : C5.

Kebetulan untuk contoh ini, satu semester hanya ada 1 kuliah.

3. Sediakan data uji sendiri, yang menjamin DAG jika diubah ke dalam representasi graf.

**BAB II**

**IMPLEMENTASI PROGRAM**

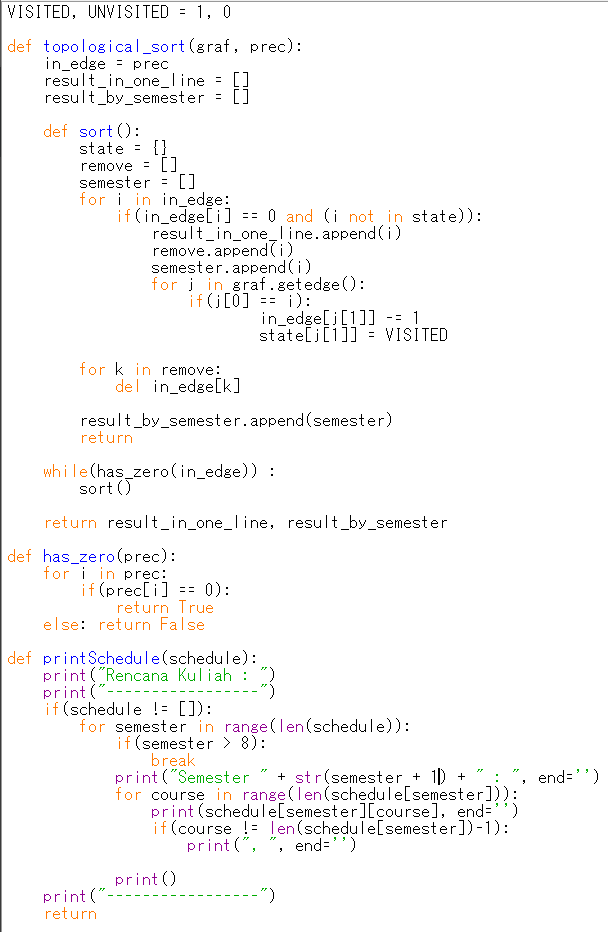
Bahasa pemograman yang digunakan adalah python. Data mata kuliah direpresentasikan ke dalam bentuk DAG(Directed Acyclic Graph) atau grap berarah non-siklus. Tiap-tiap matakuliah direpresentasikan dengan node dan sisi merepresentasikan suatu matakuliah meruapakan prerequirement matakuliah tertentu. Algoritma yang digunakan adalah algoritma topological sort yang berkaitan dengan pendekatan decrease and conquer.

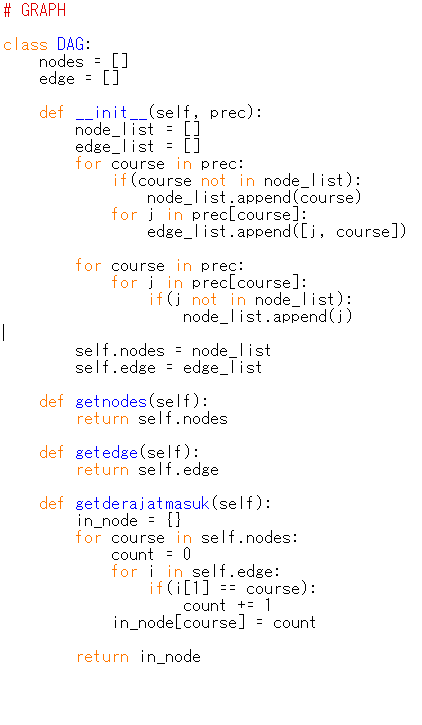
Algoritma decrease and conquer:

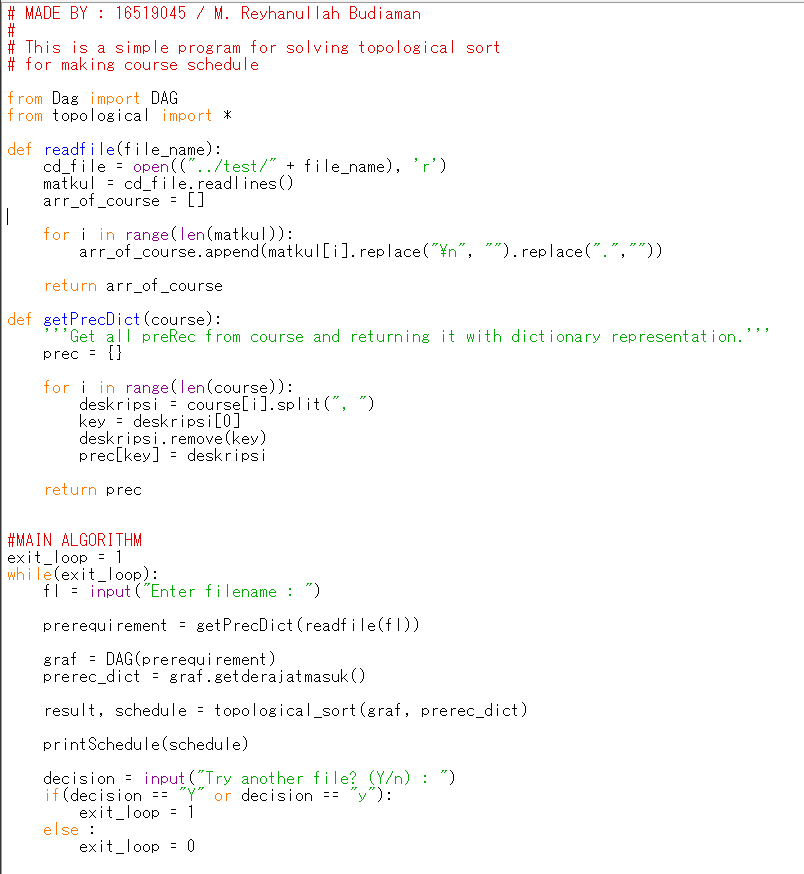
1. Hitung derajat masuk setiap simpul pada graf (N).
2. Eliminasi *n* simpul dengan derajat = 0 dan simpan simpul tersebut pada sebuah tuple solusi. Untuk setiap simpul yang dieliminasi, lakukan pengurangan derajat masuk dari simpul yang bersisian dari simpul yang dieliminasi (next node )
3. Ulangi proses 1-2 terhadap N-n simpul yang tersisa.

Jumlah iterasi yang dilakukan, dengan memperhitungkan iterasi untuk mengurangi derajat masuk, adalah *T(n)* = *n2* untuk worst case scenarionya, dan best case scenarionya adalah *T(n)* = 1. Sehingga kompleksitas algoritmanya adalah O(n2) yang tentunya lebih baik dibandingkan dengan menggunakan algoritma brute-force dengan kompleksitas O(nn)

Berikut merupakan Source code dari program yang telah dibuat dalam bahasa python.







Fungsi – fungsi yang ada :

* read\_code(file\_name) : fungsi ini membaca daftar mata kuliah yang ada didalam file dan mengembalikannya dalam bentuk array
* getPrecDict(course) : fungsi ini mengubah membentuk dictionary dengan mata kuliah sebagai key
* topological\_sort(graf, prec) : Menyusun rencana kuliah berdasarkan algoritma decrease and conquer
* printSchedule(schedule) : menampilkan solusi

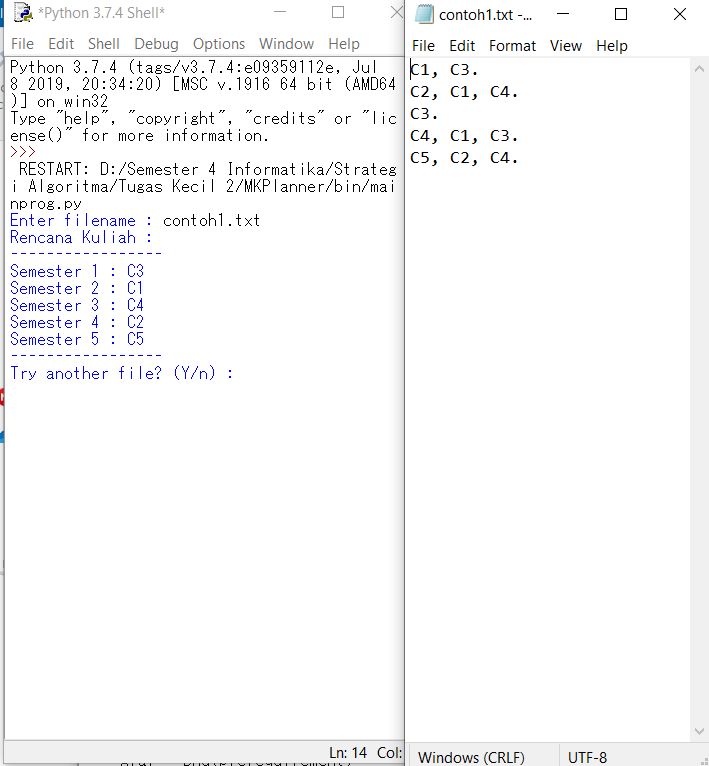
Class yang ada :

Class DAG : class untuk merekonstruksi graf

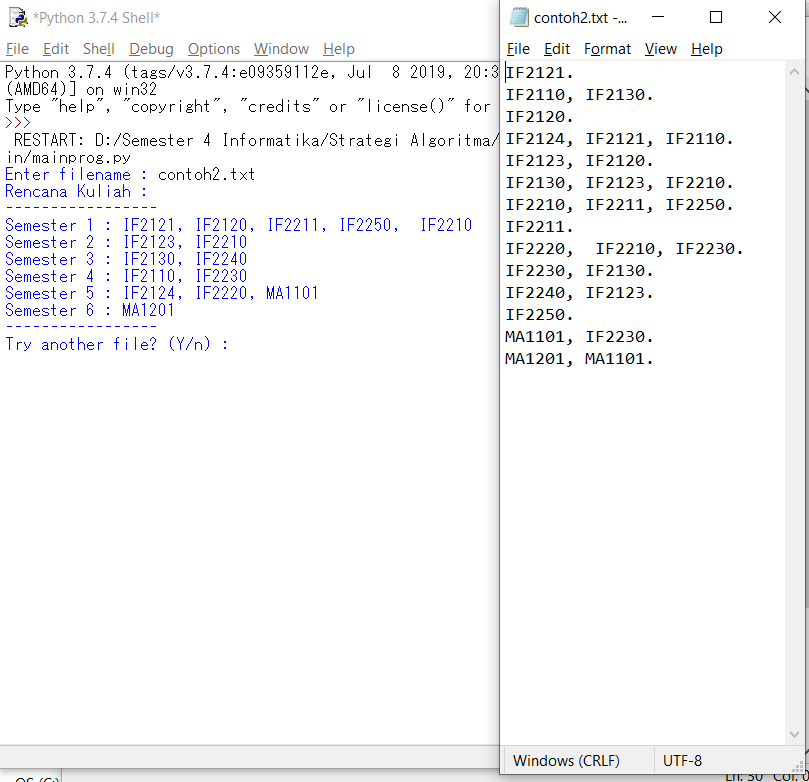
**BAB III**

**EKSPERIMEN**

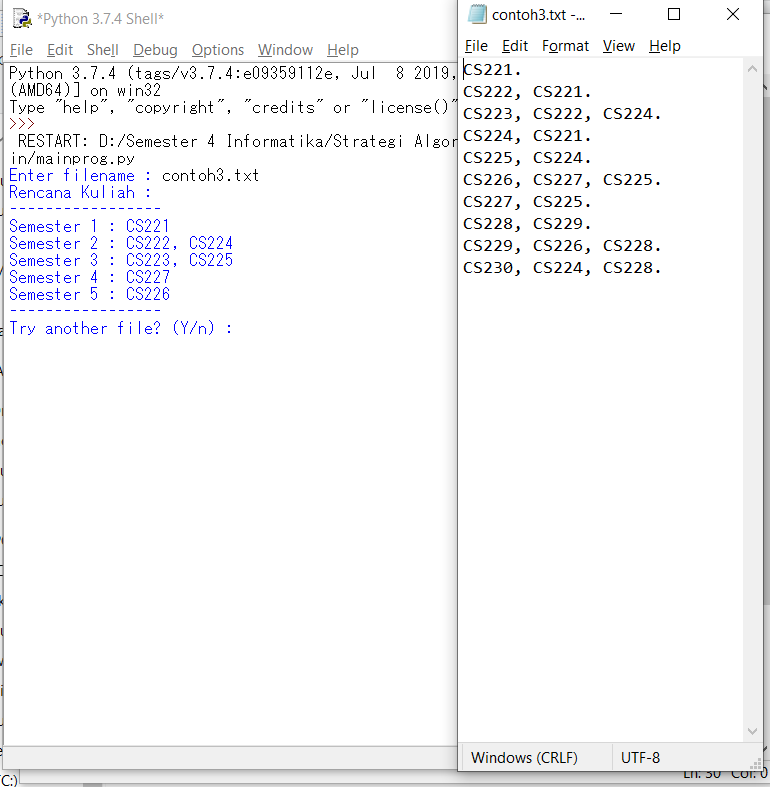
Contoh 1 (contoh1.txt):



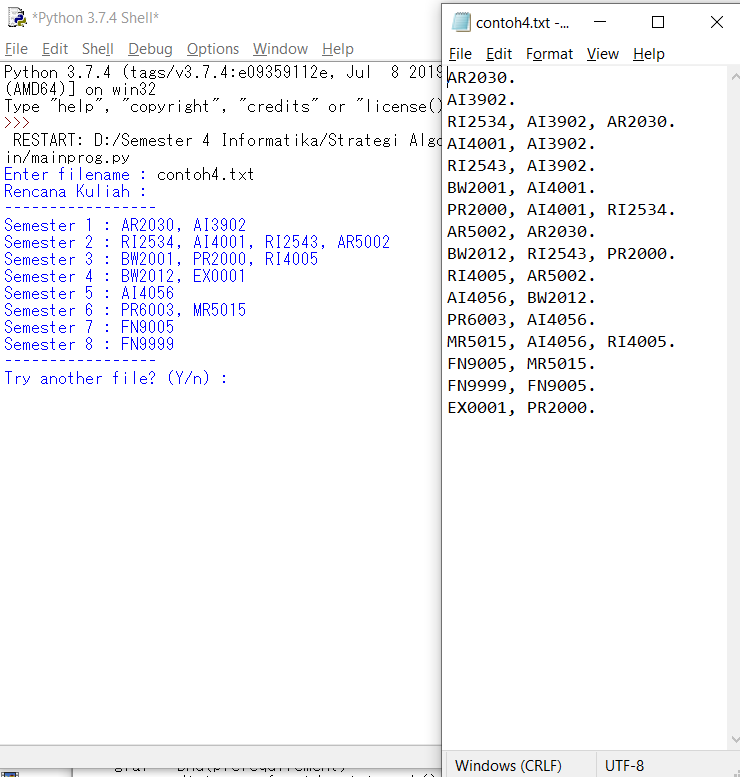
Contoh 2 (contoh2.txt):

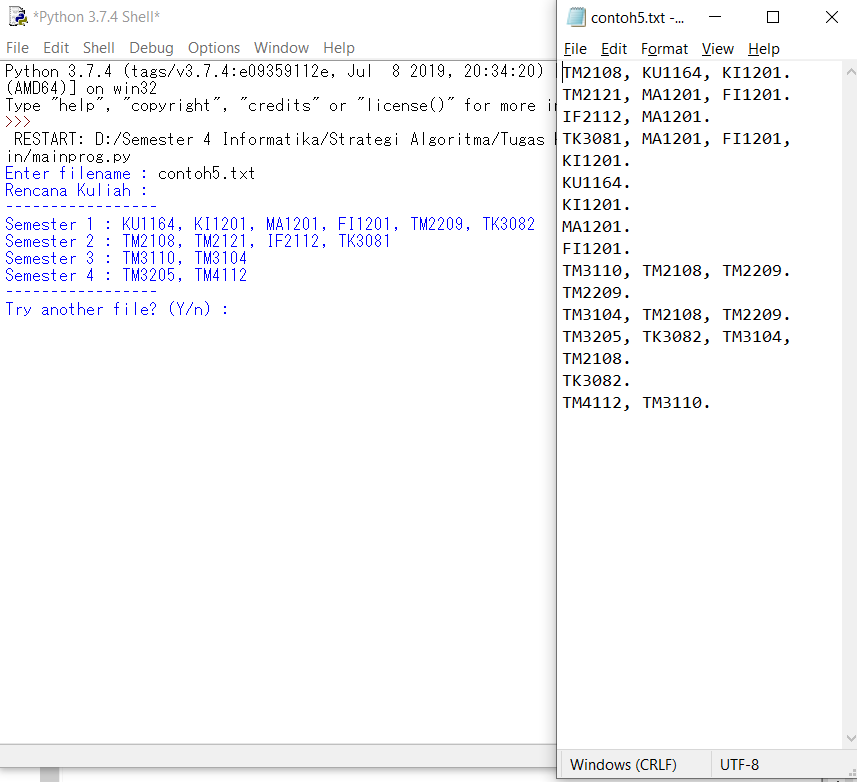


Contoh 3 (contoh3.txt):

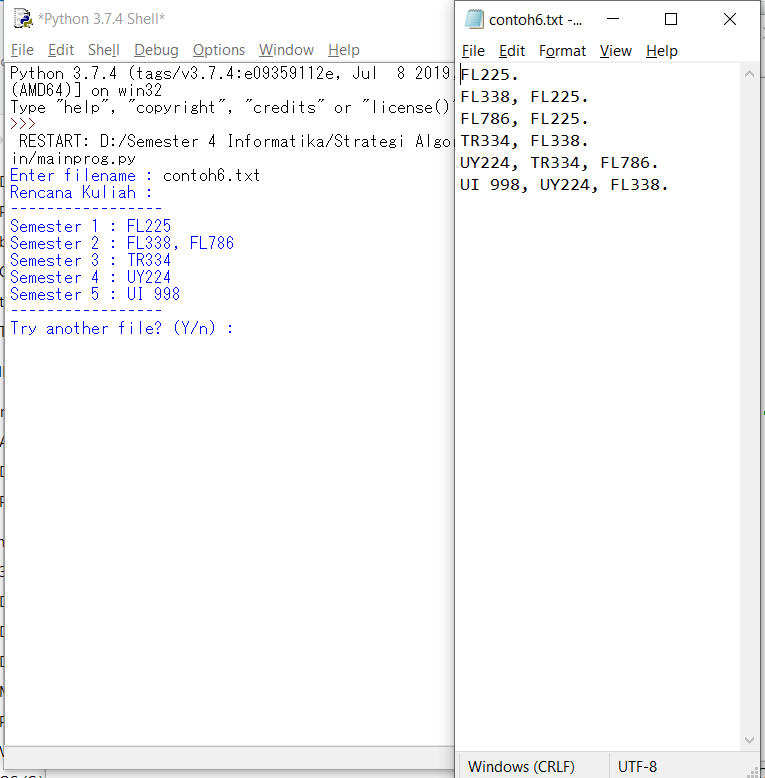


Contoh 4 (contoh4.txt):

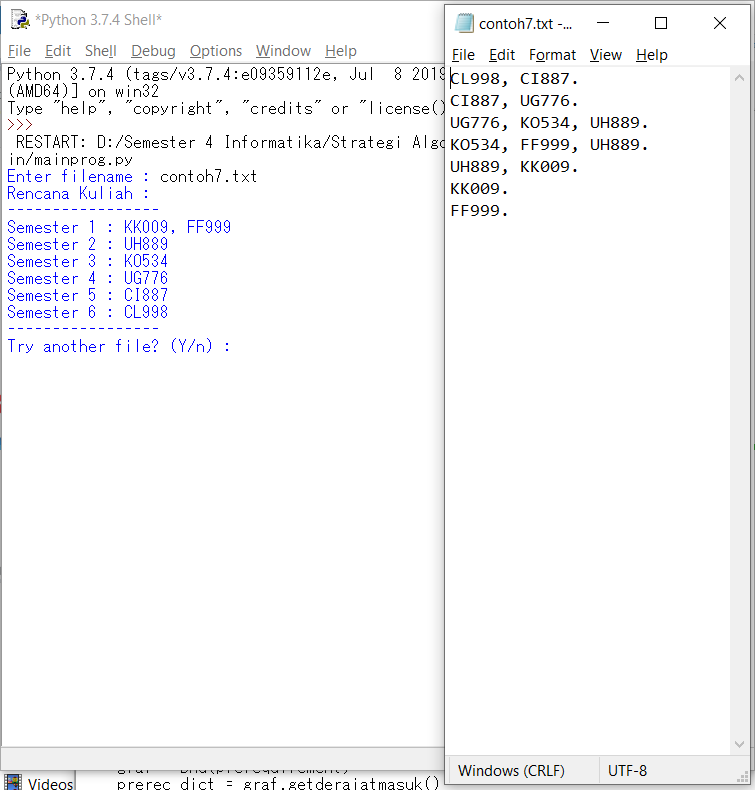


Contoh 5 (contoh5.txt):  


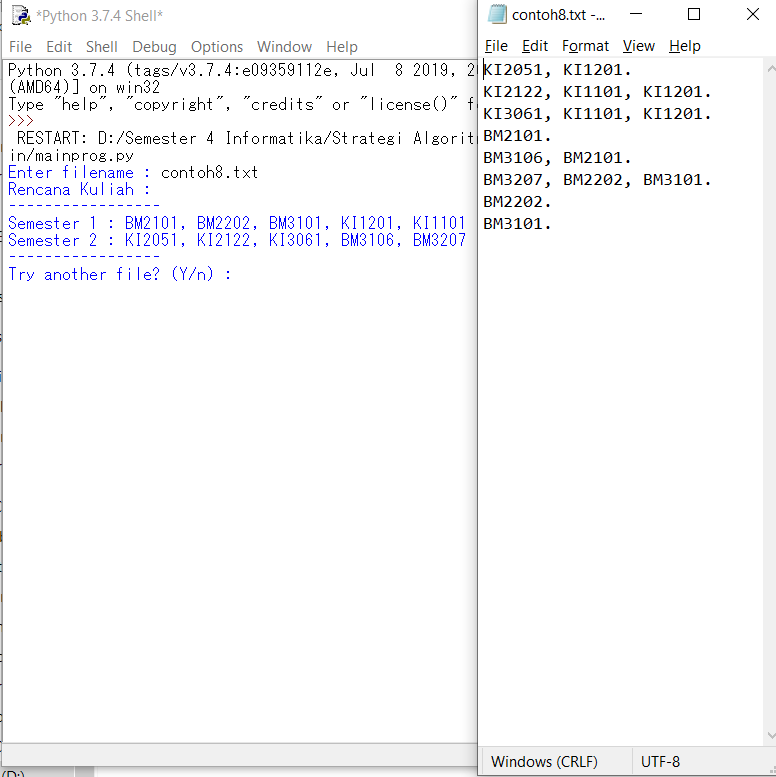
Contoh 6 (contoh6.txt):



Contoh 7 (contoh7.txt):



Contoh 8 (contoh 8.txt):



**BAB IV**

**LAMPIRAN**

**5.1 Alamat Repository**

https://github.com/reynull20/MKPlanner

**5.2 Penilaian**

|  |  |  |
| --- | --- | --- |
| 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. |  |  |