

**LAPORAN TUGAS KECIL 2**  
**IF2211 - STRATEGI ALGORITMA**  
**PENYUSUNAN RENCANA KULIAH DENGAN TOPOLOGICAL SORT**  
**(PENERAPAN DECREASE AND CONQUER)**  
**SEMESTER II TAHUN 2020/2021**

Oleh:  
Aurelius Marcel Candra (13519198)

Prodi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
2020

## **BAGIAN 1**

### **PENJELASAN ALGORITMA**

Langkah-langkah yang dilakukan oleh program untuk memperoleh solusi adalah sebagai berikut:

1. Mengorganisasikan penyimpanan data dari file yang dibaca. Pada Langkah pertama, program membaca elemen pertama pada tiap baris sebagai node, sedangkan elemen-elemen setelahnya digabungkan dalam sebuah list sebagai prerequisite dari node tersebut. Hal ini dilakukan untuk memudahkan seleksi dan penghapusan elemen.
2. Mencari semua node dengan list prerequisite kosong. Proses dilakukan dengan menelusuri seluruh node dalam list, jika elemen memiliki list prerequisite kosong, pindahkan elemen tersebut ke dalam list solusi.
3. Menghapus node yang sudah dipilih dari list prerequisite node lain. Dengan dilakukan penelusuran ke setiap elemen prerequisite dari setiap node, jika terdapat elemen yang sudah dipilih (dihapus dari list), elemen tersebut dihapus dari list prerequisite.
4. Ulangi langkah 2 dan 3 hingga list menjadi kosong.

Kaitan topological sort dengan algoritma decrease and conquer terletak pada jumlah elemen yang diperiksa untuk setiap perulangan. Walaupun topological sort tidak selalu dapat membagi persoalan menjadi bagian-bagian yang lebih kecil, aspek 'decrease' pada algoritma ini terlihat jelas pada jumlah elemen yang perlu diperiksa yang terus berkurang pada setiap perulangan dikarenakan adanya penghapusan elemen yang diseleksi. Dengan demikian, pada topological sort, algoritma decrease and conquer dikategorikan decrease by a variable size. Aspek 'conquer' pada algoritma topological sort terletak pada tahap seleksi node yang tidak memiliki prerequisite, dilanjutkan dengan penghapusan node serupa pada list prerequisite node lain.

## BAGIAN 2

### SOURCE CODE

```
import os

def topological_sort(file_name):
    # 1. Akses File
    file_sample = open(file_name, 'r')
    Lines = file_sample.readlines()

    container = []
    # > List untuk isi dari file

    for line in Lines:
        container.append(line.strip().split(','))

    # 2. Reform Bentuk List
    reform_container = []
    # > List dengan struktur elemen: [0] = elemen, [1] = list prerequisite | [ [ 'i' , [ 'j' , 'k' ] ] , ... ]

    for element in container:
        body = []
        # > temp var untuk menyimpan list prerequisite ([1])

        for i in range(len(element)):
            if (i == 0) :
                head = ''.join(character for character in element[i] if character.isalnum() or character=='_')
                # > temp var untuk menyimpan elemen ([0])
            else:
                body.append(''.join(character for character in element[i] if character.isalnum() or character=='_'))
```

```

    reform_container.append([head,body])

# 3. Seleksi dan Pop Elemen
sorted_container = []
# > List untuk hasil akhir, memiliki struktur elemen berupa himpunan elemen yang dapat dipilih | [ [ 'i' , 'j' ] , [ 'k' ] , ...
]

while (len(reform_container) != 0):
# > Loop while dilakukan hingga semua elemen pada reform_container terpilih, dengan asumsi bahwa soal selalu dapat diselesaikan

    popped_element = []
    # > List untuk menyimpan elemen reform_container yang di-pop (untuk menghilangkan elemen yang sudah terdapat dalam solusi)
    element_name = []
    # > List untuk menyimpan nama elemen yang tidak memiliki prerequisite

# 3.A. Mencari index elemen yang tidak memiliki prerequisite dan kemudian di-pop
i = 0

while (i < len(reform_container)):
    if (len(reform_container[i][1]) == 0):
        element_name.append(reform_container[i][0])
        popped_element.append(reform_container.pop(i))
        i = 0
    else:
        i += 1

sorted_container.append(element_name)

# 3.B. Menghapus dari list prerequisite elemen lain, elemen yang sudah berada dalam solusi
for l in range(len(element_name)):

```

```

        for i in range(len(reform_container)):
            for j in range(len(reform_container[i][1])):
                if (reform_container[i][1][j] == element_name[1]):
                    reform_container[i][1].remove(element_name[1])
                    break

# 4. Print Soal dan Solusi
print("Persoalan:")
for line in Lines:
    print(" ■ " + line.strip())

print("\nSolusi:")
for i in range(len(sorted_container)):
    print("Semester {} \t: {}".format(i + 1, ', '.join(c for c in sorted_container[i])))

if __name__ == '__main__':
    current_dirr = os.path.dirname(__file__)
    parent_dirr = os.path.split(current_dirr)[0]
    file_path = os.path.join(parent_dirr, 'test')

    for file_name in os.scandir(file_path):
        topological_sort(file_name)
    print()

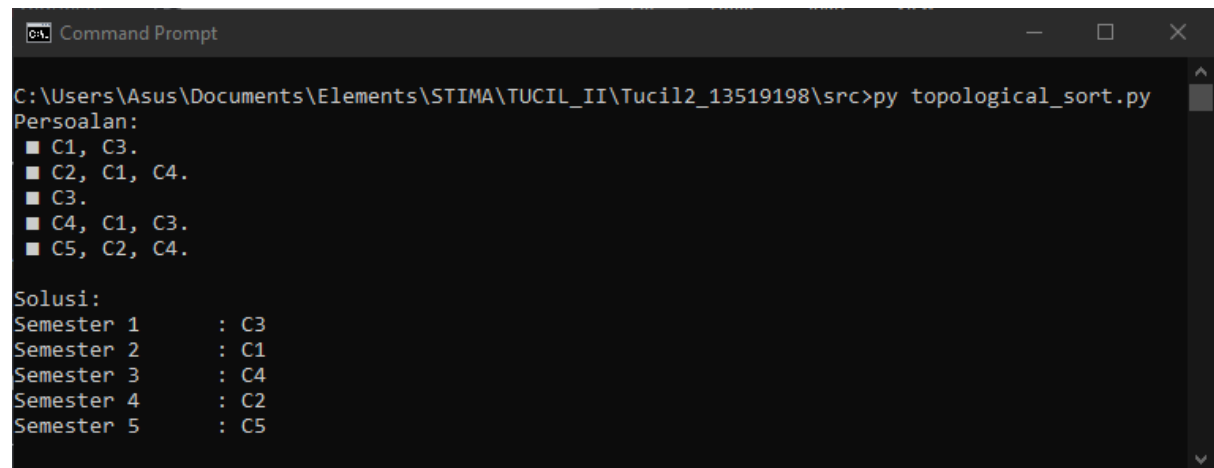
```

### BAGIAN 3 HASIL PERCOBAAN

**INPUT: sample0.txt (template)**

C1, C3.  
C2, C1, C4.  
C3.  
C4, C1, C3.  
C5, C2, C4.

**OUTPUT:**



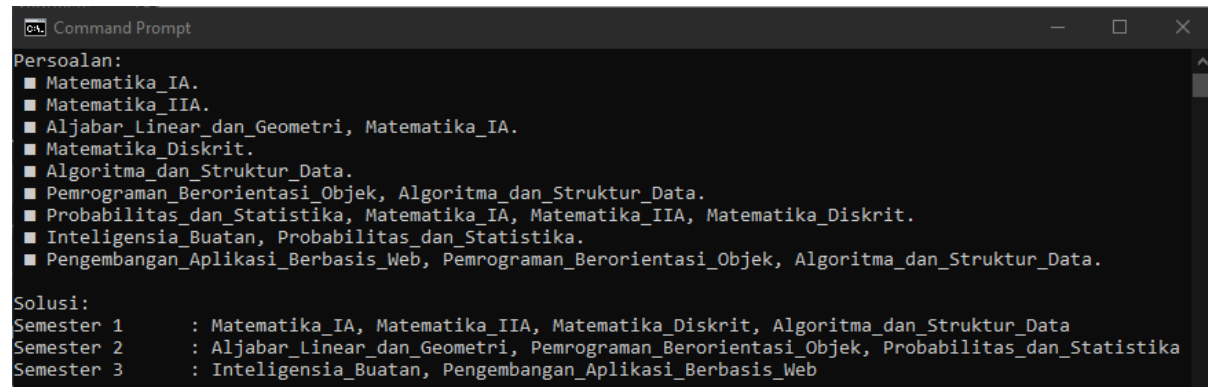
```
C:\Users\Asus\Documents\Elements\STIMA\TUCIL_II\Tucil2_13519198\src>py topological_sort.py
Persoalan:
■ C1, C3.
■ C2, C1, C4.
■ C3.
■ C4, C1, C3.
■ C5, C2, C4.

Solusi:
Semester 1      : C3
Semester 2      : C1
Semester 3      : C4
Semester 4      : C2
Semester 5      : C5
```

### INPUT: sample1.txt (berisi teks nama mata kuliah)

```
Matematika_IA.  
Matematika_IIA.  
Aljabar_Linear_dan_Geometri, Matematika_IA.  
Matematika_Diskrit.  
Algoritma_dan_Struktur_Data.  
Pemrograman_Berorientasi_Objek, Algoritma_dan_Struktur_Data.  
Probabilitas_dan_Statistika, Matematika_IA, Matematika_IIA, Matematika_Diskrit.  
Inteligensia_Buatan, Probabilitas_dan_Statistika.  
Pengembangan_Aplikasi_Berbasis_Web, Pemrograman_Berorientasi_Objek,  
Algoritma_dan_Struktur_Data.
```

### OUTPUT:

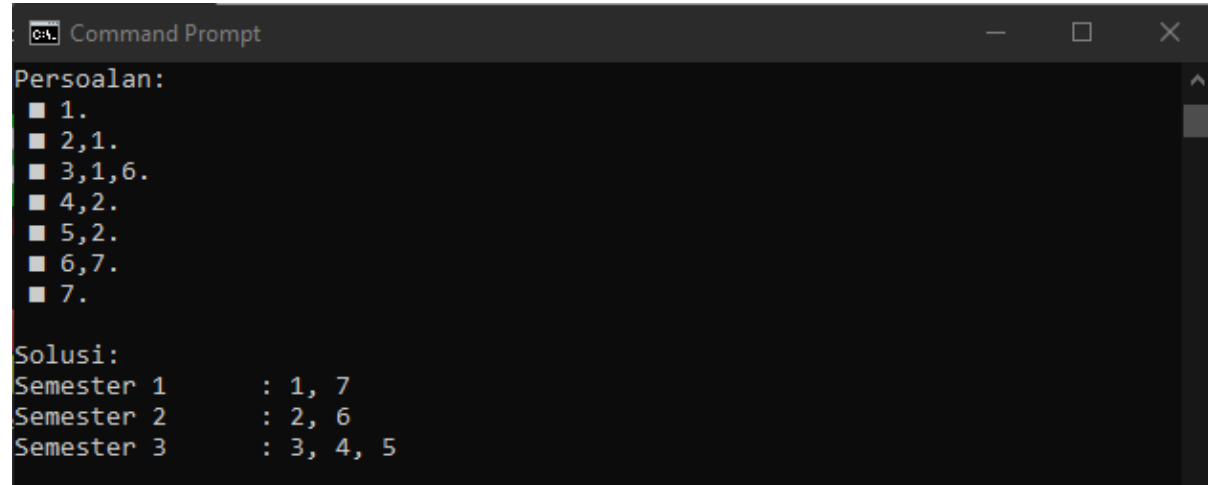


```
ca Command Prompt  
Persoalan:  
■ Matematika_IA.  
■ Matematika_IIA.  
■ Aljabar_Linear_dan_Geometri, Matematika_IA.  
■ Matematika_Diskrit.  
■ Algoritma_dan_Struktur_Data.  
■ Pemrograman_Berorientasi_Objek, Algoritma_dan_Struktur_Data.  
■ Probabilitas_dan_Statistika, Matematika_IA, Matematika_IIA, Matematika_Diskrit.  
■ Inteligensia_Buatan, Probabilitas_dan_Statistika.  
■ Pengembangan_Aplikasi_Berbasis_Web, Pemrograman_Berorientasi_Objek, Algoritma_dan_Struktur_Data.  
Solusi:  
Semester 1      : Matematika_IA, Matematika_IIA, Matematika_Diskrit, Algoritma_dan_Struktur_Data  
Semester 2      : Aljabar_Linear_dan_Geometri, Pemrograman_Berorientasi_Objek, Probabilitas_dan_Statistika  
Semester 3      : Inteligensia_Buatan, Pengembangan_Aplikasi_Berbasis_Web
```

**INPUT: sample2.txt (berupa kumpulan node angka)**

```
1.  
2,1.  
3,1,6.  
4,2.  
5,2.  
6,7.  
7.
```

**OUTPUT:**



The screenshot shows a Windows Command Prompt window with the title "C:\> Command Prompt". The window contains the following text:

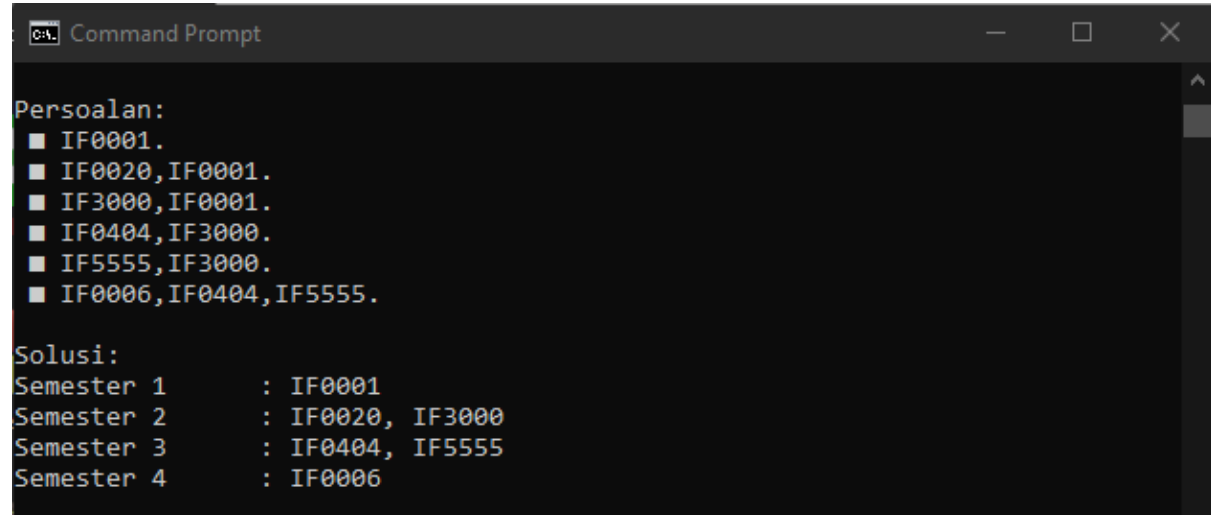
```
Persoalan:  
■ 1.  
■ 2,1.  
■ 3,1,6.  
■ 4,2.  
■ 5,2.  
■ 6,7.  
■ 7.  
  
Solusi:  
Semester 1      : 1, 7  
Semester 2      : 2, 6  
Semester 3      : 3, 4, 5
```



**INPUT: sample3.txt (berupa kode mata kuliah acak)**

```
IF0001.  
IF0020,IF0001.  
IF3000,IF0001.  
IF0404,IF3000.  
IF5555,IF3000.  
IF0006,IF0404,IF5555.
```

**OUTPUT:**

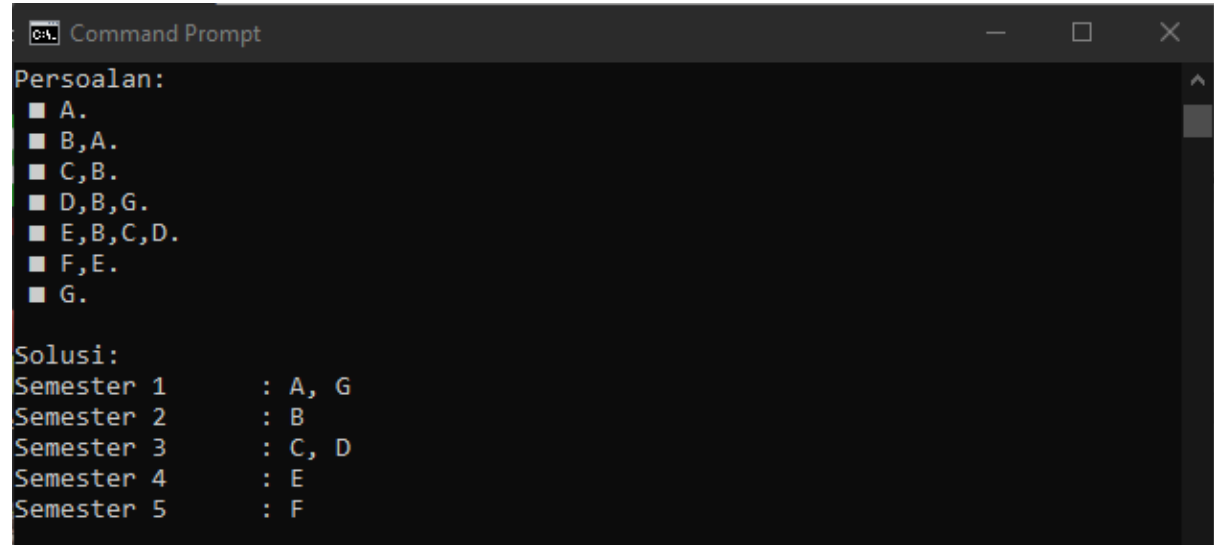


```
Command Prompt  
Persoalan:  
■ IF0001.  
■ IF0020,IF0001.  
■ IF3000,IF0001.  
■ IF0404,IF3000.  
■ IF5555,IF3000.  
■ IF0006,IF0404,IF5555.  
  
Solusi:  
Semester 1      : IF0001  
Semester 2      : IF0020, IF3000  
Semester 3      : IF0404, IF5555  
Semester 4      : IF0006
```

**INPUT: sample4.txt (berupa kumpulan node alfabet)**

A.  
B,A.  
C,B.  
D,B,G.  
E,B,C,D.  
F,E.  
G.

**OUTPUT:**



The screenshot shows a Windows Command Prompt window with the title "C:\> Command Prompt". The window contains the following text:

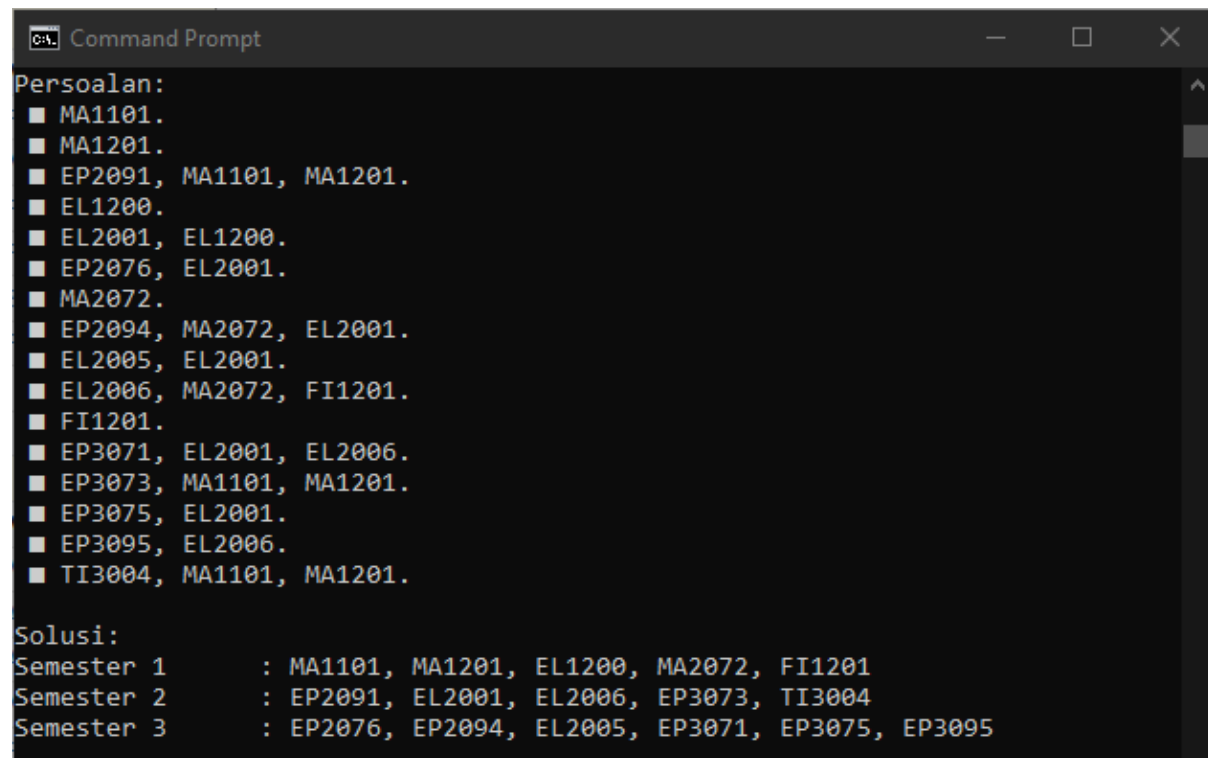
```
Persoalan:
■ A.
■ B,A.
■ C,B.
■ D,B,G.
■ E,B,C,D.
■ F,E.
■ G.

Solusi:
Semester 1      : A, G
Semester 2      : B
Semester 3      : C, D
Semester 4      : E
Semester 5      : F
```

**INPUT: sample5.txt (kumpulan kode mata kuliah EL)**

```
MA1101.  
MA1201.  
EP2091, MA1101, MA1201.  
EL1200.  
EL2001, EL1200.  
EP2076, EL2001.  
MA2072.  
EP2094, MA2072, EL2001.  
EL2005, EL2001.  
EL2006, MA2072, FI1201.  
FI1201.  
EP3071, EL2001, EL2006.  
EP3073, MA1101, MA1201.  
EP3075, EL2001.  
EP3095, EL2006.  
TI3004, MA1101, MA1201.
```

**OUTPUT:**

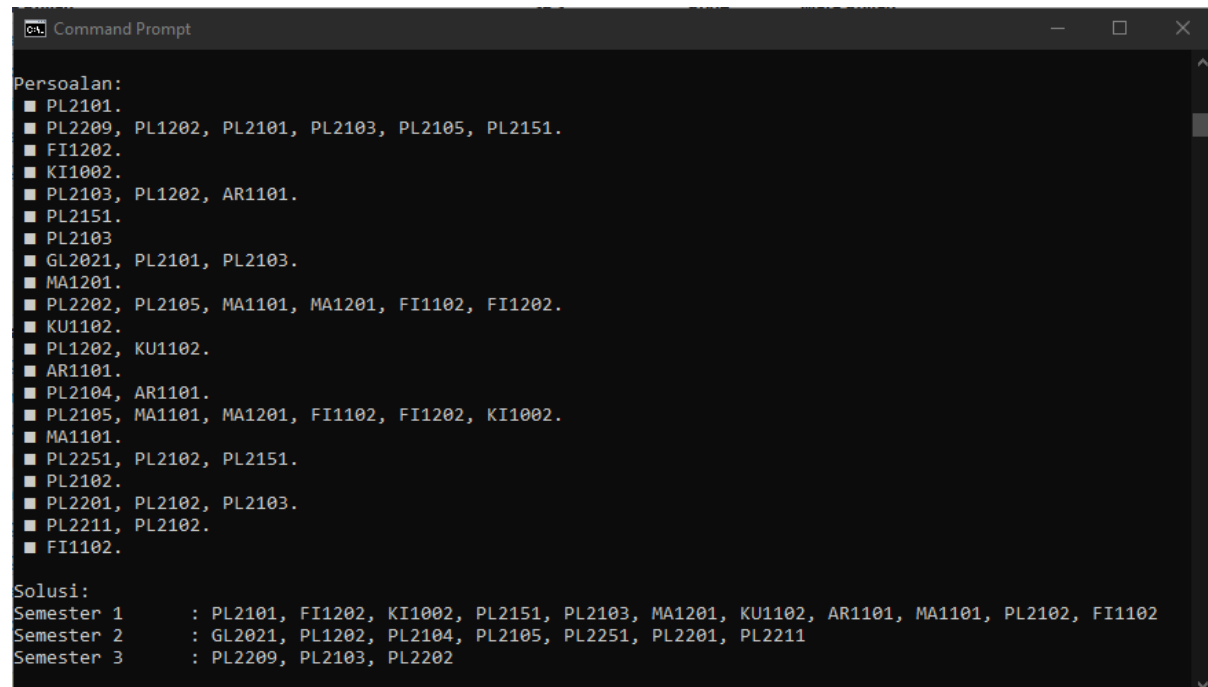


```
Command Prompt  
Persoalan:  
■ MA1101.  
■ MA1201.  
■ EP2091, MA1101, MA1201.  
■ EL1200.  
■ EL2001, EL1200.  
■ EP2076, EL2001.  
■ MA2072.  
■ EP2094, MA2072, EL2001.  
■ EL2005, EL2001.  
■ EL2006, MA2072, FI1201.  
■ FI1201.  
■ EP3071, EL2001, EL2006.  
■ EP3073, MA1101, MA1201.  
■ EP3075, EL2001.  
■ EP3095, EL2006.  
■ TI3004, MA1101, MA1201.  
  
Solusi:  
Semester 1      : MA1101, MA1201, EL1200, MA2072, FI1201  
Semester 2      : EP2091, EL2001, EL2006, EP3073, TI3004  
Semester 3      : EP2076, EP2094, EL2005, EP3071, EP3075, EP3095
```

### INPUT: sample6.txt (kumpulan kode mata kuliah PL)

```
PL2101.
PL2209, PL1202, PL2101, PL2103, PL2105, PL2151.
FI1202.
KI1002.
PL2103, PL1202, AR1101.
PL2151.
PL2103
GL2021, PL2101, PL2103.
MA1201.
PL2202, PL2105, MA1101, MA1201, FI1102, FI1202.
KU1102.
PL1202, KU1102.
AR1101.
PL2104, AR1101.
PL2105, MA1101, MA1201, FI1102, FI1202, KI1002.
MA1101.
PL2251, PL2102, PL2151.
PL2102.
PL2201, PL2102, PL2103.
PL2211, PL2102.
FI1102.
```

### OUTPUT:



```
Command Prompt

Persoalan:
■ PL2101.
■ PL2209, PL1202, PL2101, PL2103, PL2105, PL2151.
■ FI1202.
■ KI1002.
■ PL2103, PL1202, AR1101.
■ PL2151.
■ PL2103
■ GL2021, PL2101, PL2103.
■ MA1201.
■ PL2202, PL2105, MA1101, MA1201, FI1102, FI1202.
■ KU1102.
■ PL1202, KU1102.
■ AR1101.
■ PL2104, AR1101.
■ PL2105, MA1101, MA1201, FI1102, FI1202, KI1002.
■ MA1101.
■ PL2251, PL2102, PL2151.
■ PL2102.
■ PL2201, PL2102, PL2103.
■ PL2211, PL2102.
■ FI1102.

Solusi:
Semester 1 : PL2101, FI1202, KI1002, PL2151, PL2103, MA1201, KU1102, AR1101, MA1101, PL2102, FI1102
Semester 2 : GL2021, PL1202, PL2104, PL2105, PL2251, PL2201, PL2211
Semester 3 : PL2209, PL2103, PL2202
```

**INPUT: sample7.txt (kumpulan kode mata kuliah AK)**

AK4094, AK4093.  
AK2163, MA1101, MA1201.  
MA1101.  
AK4093, AK4185, AK3283.  
AK4091, AK3283, AK4185.  
MA2181, MA1101, MA1201, MA2081.  
MA2081.  
MA3131, MA2031.  
MA3281, MA2181, MA3181.  
MA4181, MA2181, MA3181.  
AK2281, MA2181.  
MA2271, MA1101, MA1201.  
AK4185, MA3181.  
MA1201.  
AK4285, AK4185.  
MA3182.  
AK3182, MA2181, MA3182.  
MA3181, MA2181.  
MA3071, MA2031, MA2021.  
MA2031.  
AK3283, MA3181.  
MA2021.  
AK2263, AK2163.  
MA2151, MA1101, MA1201.  
AK4183, AK3283.  
AK4092, AK4091.

## OUTPUT:

```
Command Prompt

Persoalan:
■ AK4094, AK4093.
■ AK2163, MA1101, MA1201.
■ MA1101.
■ AK4093, AK4185, AK3283.
■ AK4091, AK3283, AK4185.
■ MA2181, MA1101, MA1201, MA2081.
■ MA2081.
■ MA3131, MA2031.
■ MA3281, MA2181, MA3181.
■ MA4181, MA2181, MA3181.
■ AK2281, MA2181.
■ MA2271, MA1101, MA1201.
■ AK4185, MA3181.
■ MA1201.
■ AK4285, AK4185.
■ MA3182.
■ AK3182, MA2181, MA3182.
■ MA3181, MA2181.
■ MA3071, MA2031, MA2021.
■ MA2031.
■ AK3283, MA3181.
■ MA2021.
■ AK2263, AK2163.
■ MA2151, MA1101, MA1201.
■ AK4183, AK3283.
■ AK4092, AK4091.

Solusi:
Semester 1      : MA1101, MA2081, MA1201, MA3182, MA2031, MA2021
Semester 2      : AK2163, MA2181, MA3131, MA2271, MA3071, MA2151
Semester 3      : AK2281, AK3182, MA3181, AK2263
Semester 4      : MA3281, MA4181, AK4185, AK3283
Semester 5      : AK4093, AK4091, AK4285, AK4183
Semester 6      : AK4094, AK4092

C:\Users\Asus\Documents\Elements\STIMA\TUCIL_II\Tucil2_13519198\src>
```

**LINK SOURCE CODE:** [https://github.com/Ardovigus/TUCIL2\\_IF2211\\_13519198](https://github.com/Ardovigus/TUCIL2_IF2211_13519198)

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	✓	