

LAPORAN TUGAS KECIL 2

IF2211 Strategi Algoritma

Penyusunan Rencana Kuliah dengan *Topological Sort*

(Penerapan *Decrease and Conquer*)



Daffa Ananda Pratama Resyaly

13519107

Kelas 02

Sekolah Teknik Elektro dan Informatika

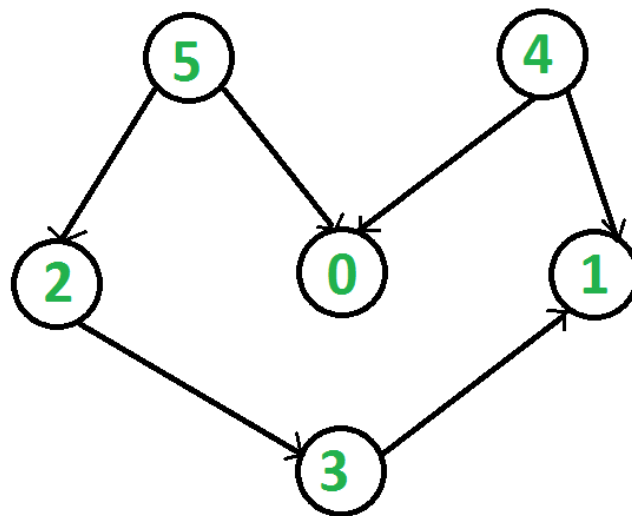
Program Studi Teknik Informatika

Tahun Ajaran 2020/2021

I. Algoritma Topological Sorting dan Kaitannya dengan Pendekatan “Decrease and Conquer”

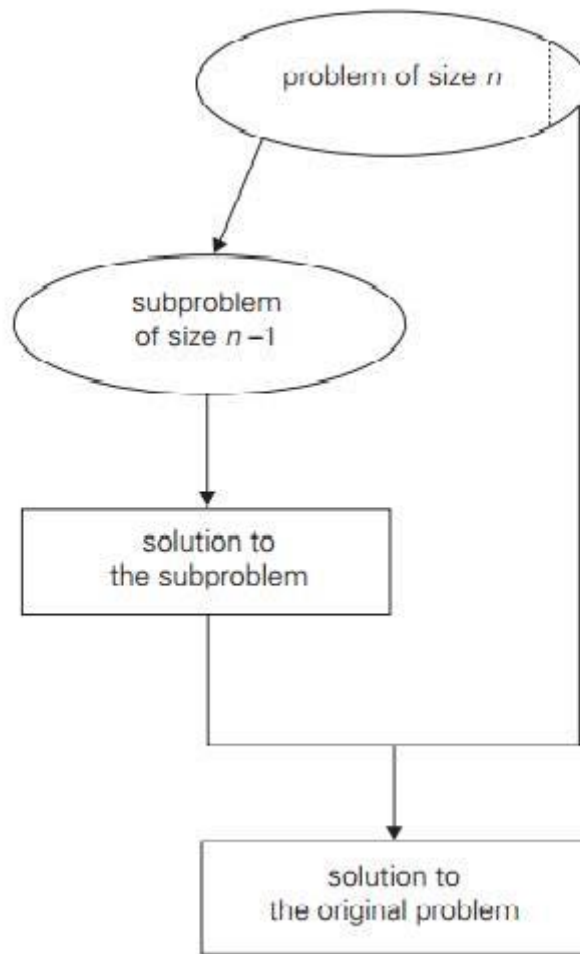
Topological Sorting adalah suatu metode pengurutan simpul pada graf berarah dimana untuk setiap sisi berarah dari simpul A ke simpul B, simpul A pasti muncul sebelum simpul B muncul di dalam pengurutan. Hasil dari *Topological Sorting* tidaklah unik, yang berarti kemungkinan terdapat lebih dari satu hasil *Topological Sorting* untuk setiap graf berarah. Selain itu, algoritma ini juga tidak dapat diaplikasikan pada segala macam graf, melainkan terbatas hanya pada graf yang tidak memiliki *cycle*. Tipe graf yang valid diimplementasikan untuk *Topological Sorting* hanyalah *Directed Acyclic Graph* (DAG) yang merupakan sejenis graf berarah yang tidak memiliki *cycle* di dalamnya. Segala macam *tree* juga cocok untuk diimplementasikan untuk *Topological Sorting* karena *tree* tidak mungkin mempunyai *cycle* di dalamnya.

Algoritma *Topological Sorting* memiliki banyak aplikasi dalam dunia nyata karena banyak hal dalam dunia nyata dapat direpresentasikan dalam bentuk graf berarah dimana suatu kejadian harus terjadi sebelum kejadian berikutnya terjadi. Contoh dari pengaplikasian *Topological Sorting* pada dunia nyata adalah penyusunan rencana kuliah, *program dependencies*, penjadwalan suatu acara, dan instruksi dalam bahasa assembly. Algoritma pengurutan menggunakan *Topological Sorting* memiliki kompleksitas waktu $O(V+E)$, dimana V adalah jumlah simpul dan E adalah jumlah sisi pada suatu graf berarah.



Gambar 1.1. Contoh *Directed Acyclic Graph* untuk *Topological Sorting*
Sumber : <https://www.geeksforgeeks.org/topological-sorting/>

Decrease and Conquer merupakan suatu strategi metode perancangan algoritma dengan mereduksi persoalan menjadi dua upa persoalan yang lebih kecil, kemudian hanya satu upa persoalan dari kedua upa persoalan tersebut yang diproses atau diselesaikan. Di dalam algoritma *Decrease and Conquer*, tidak ada tahap penggabungan upa persoalan (tahap *combine*) karena yang diproses hanyalah satu upa persoalan saja. Algoritma *Decrease and Conquer* memiliki tiga variasi, yaitu *decrease by a constant* (ukuran instans persoalan direduksi sebesar konstanta yang sama, biasanya 1, pada setiap iterasi algoritma), *decrease by a constant factor* (ukuran instans direduksi sebesar faktor konstanta yang sama, biasanya 2, pada setiap iterasi algoritma), dan *decrease by a variable size* (ukuran instans persoalan direduksi bervariasi pada setiap iterasi algoritma).



Gambar 1.2. Contoh variasi dari *Decrease and Conquer* (*Decrease by a constant*)

Sumber : http://www.brainkart.com/article/Decrease-and-Conquer_8015/

Kaitan algoritma *Topological Sorting* dengan pendekatan *Decrease and Conquer* adalah algoritma *Topological Sorting* mereduksi suatu persoalan bervariasi tergantung dengan ukuran ruang lingkup persoalannya. Secara umum, algoritma *Topological Sorting* tersebut adalah sebagai berikut. Pertama, akan dipilih simpul secara acak dari graf yang belum pernah dikunjungi. Lalu, dimulai dari simpul yang terpilih tersebut, lakukan *Depth First Search* (DFS) yang mengunjungi setiap simpul tetangga yang belum pernah dikunjungi sebelumnya. Kemudian, DFS akan dipanggil kembali secara rekursif. Dalam pemanggilan rekursif tersebut, simpul tersebut ditambahkan ke list *ordering* pada posisi terakhir.

II. Source Code (Python)

Pada *source code* dari program yang telah dibuat, terdapat sebuah file python yang berisi tiga fungsi utama dan satu program utama untuk menerapkan algoritma *Topological Sorting* untuk menyusun daftar mata kuliah dari file yang ada, satu fungsi tambahan untuk memperindah penjalanan program pada CLI, dan satu program utama untuk mengetes jalannya program.

2.1. Fungsi Utama

2.1.1. Fungsi untuk membaca file teks

```
# Read input from text file
# And change it to DAG (containing list of course and its prerequisite)
def readfile(filename):
    # Open the text file and read it line-by-line
    fOpen = open(filename)
    fRead = fOpen.read().split('\n')

    listCourse = []

    # Clean the blank newline
    fRead = filter(lambda x: x.strip(), fRead)
    for line in fRead:
        # Clean each line from unwanted characters
        # e.g. From "C1, C2, C3." to "C1,C2,C3"
        line = line.strip()
        line = line.replace('.', '')
        # Insert each course ID that was separated by comma to listCourse
        # e.g. From "C1,C2,C3" to "['C1', 'C2', 'C3']"
        line = line.split(',')
        listCourse.append(line)

    return listCourse
```

2.1.2. Fungsi untuk melakukan Topological Sort

```
def topologicalsort(listCourse, result):
    # Do process recursively until the list of course has no more element
    if len(listCourse) != 0:
        available = []
        # Find course which has no prerequisite in list of course to be processed first
        # Insert that course to list of available courses
        for courses in listCourse:
            if (len(courses) == 1):
                available.append(courses[0])
        i = 0
        while (i < len(listCourse)):
            # Delete node of the course which is in list 'available' on each course in listCourse,
            # if it's one of the prerequisites of relevant course
            for course in available:
                if course in listCourse[i]:
                    listCourse[i].remove(course)
            # Delete node that has been removed before
            # (the one(s) on list 'available')
            # Size of listCourse is reduced by 1
            if (len(listCourse[i]) == 0):
                listCourse.remove(listCourse[i])
                i-=1
            i+=1

        # Format of output if it's possible to take more than one course in a semester
        # e.g. "Semester IV : IF2211, IF2212"
        # Append to result list first
        courseID = ""
        for i in range(len(available)):
            if (i+1 == len(available)):
                courseID += available[i]
            else:
                courseID += available[i]
                courseID += ', '
        result.append(courseID)
        # Recursive call of topological sort
        topologicalsort(listCourse, result)
    return result
```

2.1.3. Fungsi untuk menampilkan rencana studi yang telah disusun ke CLI

```
# Show the recommended course plan out to CLI
def courseplan(result):
    for i in range (len(result)):
        # Show the output in format :
        # "Semester <which semester> : <courseID, courseName>"
        print("Semester", end=' ')
        # Determine semester based on i (same as switch case in C)
        # Assumption : Maximum semester is 8
        semester = {
            0:"I",
            1:"II",
            2:"III",
            3:"IV",
            4:"V",
            5:"VI",
            6:"VII",
            7:"VIII"
        }[i]
        print(semester, end='')
        print("\t: " + result[i], end='')

        # End every semester except last semester with newline
        # End last semester with '.'
        if (i+1 != len(result)):
            print()
        else:
            print('.')

```

2.2. Fungsi Tambahan

2.2.1. Fungsi untuk menampilkan awalan dari program ketika dijalankan

```
# Intro
def home():
    print("\n-----Welcome to Course Plan Organizer-----")
    print("by Daffa Ananda Pratama Resyaly -- 13519107\n")
    print("Pick your choice: (Enter/Exit)")

```

2.3. Main Program

```
# Main Program
# Welcome
if __name__ == "__main__":
    home()
    option = str(input())
    option = option.lower()
    # Determine action based on option
    if (option == 'enter'):
        # Initialization to count the execution time
        start_time = time.time()
        # option == Enter => Start Program
        # Initialize test file with its path
        path = os.path.dirname(Path(__file__).absolute().parent)
        filename = "test1.txt"
        ffilename = os.path.join(path, 'test', filename)

        print('\nReading from file "%s"\n' % filename)
        print('Recommended Course Plan:')
        print('-----')

        # Start reading file and determine the output
        f = readfile(ffilename)
        res = []
        test2 = topologicalsort(f, res)
        courseplan(test2)

        # Show the program execution time
        print("\nProgram execution time : %f detik" % (time.time() - start_time))
        print("Goodbye~")
    else:
        # option == Exit => Exit Program
        exit()
```

III. Screenshot dari Input dan Output

3.1.

Input

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

Output

```
-----Welcome to Course Plan Organizer-----  
by Daffa Ananda Pratama Resyaly -- 13519107  
  
Pick your choice: (Enter/Exit)  
enter  
  
Reading from file "test1.txt"  
  
Recommended Course Plan:  
-----  
Semester I      : C3  
Semester II     : C1  
Semester III    : C4  
Semester IV     : C2  
Semester V      : C5.  
  
Program execution time : 0.010987 detik  
Goodbye~
```


3.2.

Input

```
MA1101.
FI1101.
KU1001.
KU1102.
KU1011.
KU1024.

MA1201.
FI1201.
IF1210.
KU1202.
KI1002.
EL1200.

IF2121.
IF2110.
IF2120.
IF2124.
IF2123.
IF2130.

IF2210, IF2110.
IF2211.
IF2220, MA1101, MA1201, IF2120.
IF2230.
IF2240.
IF2250.

IF3170, IF2121, IF2124, IF2220, IF2211.
IF3110, IF2210, IF2110.
IF3130, IF2230.
IF3141, IF2240, IF2250.
IF3150, IF2250
IF3140.
IF3151, IF2250.

IF3210, IF2130, IF2110.
IF3270, IF3170, IF2110.
IF3230, IF3130.
IF3250, IF3150, IF2250.
IF3260, IF2130, IF2110, IF2123.
IF3280.

IF4090, IF3280.
IF4091.

IF4092, IF4091.
```

Output

```
-----Welcome to Course Plan Organizer-----
by Daffa Ananda Pratama Resyaly -- 13519107

Pick your choice: (Enter/Exit)
enter

Reading from file "test2.txt"

Recommended Course Plan:
-----
Semester I      : MA1101, FI1101, KU1001, KU1102, KU1011, KU1024, MA1201, FI1201, IF1210, KU1202, KI1002, EL1200, IF2121, IF2110, IF2120, IF2124, IF2123, IF2130, IF2211
, IF2230, IF2240, IF2250, IF3140, IF3280, IF4091
Semester II     : IF2210, IF2220, IF3130, IF3141, IF3150, IF3151, IF3210, IF3260, IF4090, IF4092
Semester III    : IF3170, IF3110, IF3230, IF3250
Semester IV     : IF3270.

Program execution time : 0.003003 detik
Goodbye~
```

3.3.

Input

```
MA1101.  
FI1101.  
KU1001.  
KU1102.  
KU1011.  
KU1024.  
  
MA1201.  
FI1201.  
IF1210.  
KU1202.  
KI1002.  
EL1200.  
  
II2130, IF1210.  
II2110, MA1201.  
II1111, MA1201.  
TI3005.  
IF2140.  
IF2111.  
  
II2250, IF2140.  
II2260, II2130, IF1210.  
II2230, II2130.  
II2220, TI3005.  
II2240.  
IF2212.  
  
II3150, II2110.  
II3160.  
II3120, TI3005.  
II3131.  
II3121, II2240.  
IF3152.  
  
II3260, IF3152.  
II3230, II2230, II3131.  
II2221, TI3005.  
II3220, II3121.  
II3240, IF3152.  
  
II4090.  
II4091.  
II4370.  
II4371.  
II4092, II4091.  
II4472.  
KU2071.  
BI2001.
```

Output

```
-----Welcome to Course Plan Organizer-----  
by Daffa Ananda Pratama Resyaly -- 13519107  
Pick your choice: (Enter/Exit)  
enter  
Reading from file "test3.txt"  
Recommended Course Plan:  
-----  
Semester I : MA1101, FI1101, KU1001, KU1102, KU1011, KU1024, MA1201, FI1201, IF1210, KU1202, KI1002, EL1200, TI3005, IF2140, IF2111, II2240, IF2212, II3160, II3131,  
IF3152, II4090, II4091, II4370, II4371, II4472, KU2071, BI2001  
Semester II : II2130, II2110, II1111, II2250, II2220, II3120, II3121, II3260, II2221, II3240, II4092  
Semester III : II2260, II2230, II3150, II3220  
Semester IV : II3230.  
Program execution time : 0.015996 detik  
Goodbye~
```

3.4.

Input

```
Bangun tidur.  
Mandi, Bangun tidur.  
Makan, Bangun tidur, Mandi.  
Cium tangan orang tua, Bangun tidur, Mandi, Makan.  
Berangkat sekolah, Cium tangan orang tua, Makan, Mandi, Bangun tidur.  
Belajar di kelas, Berangkat sekolah.  
Makan di kantin, Berangkat sekolah, Belajar di kelas.  
Pulang sekolah, Berangkat sekolah, Belajar di kelas, Makan di kantin.
```

Output

```
-----Welcome to Course Plan Organizer-----  
by Daffa Ananda Pratama Resyaly -- 13519107  
  
Pick your choice: (Enter/Exit)  
enter  
  
Reading from file "test4.txt"  
  
Recommended Course Plan:  
-----  
Semester I      : Bangun tidur  
Semester II     : Mandi  
Semester III    : Makan  
Semester IV     : Cium tangan orang tua  
Semester V      : Berangkat sekolah  
Semester VI     : Belajar di kelas  
Semester VII    : Makan di kantin  
Semester VIII   : Pulang sekolah.  
  
Program execution time : 0.025119 detik  
Goodbye~
```

3.5.

Input

```
Pre-Calculus.  
IT Hardware & System Software.  
IS Applications.  
Statistics.  
  
Calculus 1, Pre-Calculus.  
Comp Sci & Programming, Pre-Calculus.  
ELET 2300 C++ Program, Pre-Calculus.  
  
IS Application Development, Comp Sci & Programming, ELET 2300 C++ Program, Calculus 1.  
Internet Applications, Comp Sci & Programming, ELET 2300 C++ Program.  
IS Infrastructure & Networks, IT Hardware & System Software.  
System Analysis & Design, IS Applications, Statistics.  
Fundamentals of Info Security.  
  
Adv IS Development, IS Application Development.  
Integrated IS, IS Application Development, Internet Applications, IS Infrastructure & Networks.  
Database Management, System Analysis & Design.  
  
Enterprise Application Development, Adv IS Development, Integrated IS, Database Management.  
Database Admin & Implementation, Database Management.  
  
Project Management & Practice, Enterprise Application Development, Database Admin & Implementation.
```

Output

```
-----Welcome to Course Plan Organizer-----  
by Daffa Ananda Pratama Resyaly -- 13519107  
  
Pick your choice: (Enter/Exit)  
enter  
  
Reading from file "test5.txt"  
  
Recommended Course Plan:  
-----  
Semester I      : Pre-Calculus, IT Hardware & System Software, IS Applications, Statistics, Fundamentals of Info Security  
Semester II     : Calculus 1, Comp Sci & Programming, ELET 2300 C++ Program, IS Infrastructure & Networks, System Analysis & Design  
Semester III    : IS Application Development, Internet Applications, Database Management  
Semester IV     : Adv IS Development, Integrated IS, Database Admin & Implementation  
Semester V      : Enterprise Application Development  
Semester VI     : Project Management & Practice.  
  
Program execution time : 0.019059 detik  
Goodbye~
```

3.6.

Input

```
Algebra.  
Compilers, Software Engineering.  
Computers, Calculus.  
Geometry, Algebra.  
Operating Systems, Programming Languages.  
Programming Languages, Computers.  
Robotics, Geometry.  
Robotics, Programming Languages.  
Software Engineering, Programming Languages.  
Software Engineering, Databases.
```

Output

```
-----Welcome to Course Plan Organizer-----  
by Daffa Ananda Pratama Resyaly -- 13519107  
  
Pick your choice: (Enter/Exit)  
enter  
  
Reading from file "test6.txt"  
  
Recommended Course Plan:  
-----  
Semester I      : Algebra  
Semester II     : Geometry  
Semester III    : Robotics  
Semester IV     : Programming Languages  
Semester V      : Operating Systems, Computers, Software Engineering  
Semester VI     : Compilers, Calculus, Databases.  
  
Program execution time : 0.017000 detik  
Goodbye~
```

3.7.

Input

```
Wooden log.
Epic log, Wooden log.
Super log, Epic log.
Mega log, Super log.
Hyper log, Mega log.
Ultra log, Hyper log.

Fish.
Golden fish, Fish.
Epic fish, Golden fish.

Apple.
Banana, Apple.

Ruby.

Wolf skin.
Zombie eye.
Unicorn horn.
Mermaid hair.
Chip.

Wooden sword, Epic log, Wooden log.
Fish armor, Fish, Wooden log.
Fish sword, Golden fish, Epic log.
Wolf armor, Wooden log, Epic log, Wolf skin.
Apple sword, Apple, Wooden log, Super log.
Eye armor, Zombie eye, Wooden log, Super log.
Zombie sword, Wooden log, Apple, Super log, Zombie eye.
Banana armor, Wooden log, Banana, Super log.
Ruby sword, Ruby, Mega log, Wooden log.
Epic armor, Epic log, Epic fish.
```

Output

```
-----Welcome to Course Plan Organizer-----
by Daffa Ananda Pratama Resyaly -- 13519107

Pick your choice: (Enter/Exit)
enter

Reading from file "test7.txt"

Recommended Course Plan:
-----
Semester I       : Wooden log, Fish, Apple, Ruby, Wolf skin, Zombie eye, Unicorn horn, Mermaid hair, Chip
Semester II      : Epic log, Golden fish, Banana, Fish armor
Semester III     : Super log, Epic fish, Wooden sword, Fish sword, Wolf armor
Semester IV      : Mega log, Apple sword, Eye armor, Zombie sword, Banana armor, Epic armor
Semester V       : Hyper log, Ruby sword
Semester VI      : Ultra log.

Program execution time : 0.018258 detik
Goodbye~
```

3.8.

Input

```
Coin.

Wooden log.
Epic log, Wooden log.
Super log, Epic log.
Mega log, Super log.
Hyper log, Mega log.
Ultra log, Hyper log.

Fish.
Golden fish, Fish.
Epic fish, Golden fish.

Apple.
Banana, Apple.

Ruby.

Wolf skin.
Zombie eye.
Unicorn horn.
Mermaid hair.
Chip.

Unicorn sword, Unicorn horn, Fish, Super log.
Ruby armor, Ruby, Unicorn horn, Super log, Mega log.
Hair sword, Mermaid hair, Mega log.
Coin armor, Coin, Hyper log.
Coin sword, Coin, Ruby, Hyper log.
Mermaid armor, Mermaid hair, Mega log, Golden fish, Fish.
Electronical sword, Chip, Hyper log.
Electronical armor, Chip, Mega log, Hyper log.
EDGY sword, Wooden log, Ultra log.
EDGY armor, Wolf skin, Zombie eye, Unicorn horn, Mermaid hair, Chip.
```

Output

```
-----Welcome to Course Plan Organizer-----
by Daffa Ananda Pratama Resyaly -- 13519107

Pick your choice: (Enter/Exit)
enter

Reading from file "test8.txt"

Recommended Course Plan:
-----
Semester I      : Coin, Wooden log, Fish, Apple, Ruby, Wolf skin, Zombie eye, Unicorn horn, Mermaid hair, Chip
Semester II     : Epic log, Golden fish, Banana, EDGY armor
Semester III    : Super log, Epic fish
Semester IV     : Mega log, Unicorn sword
Semester V      : Hyper log, Ruby armor, Hair sword, Mermaid armor
Semester VI     : Ultra log, Coin armor, Coin sword, Electronical sword, Electronical armor
Semester VII    : EDGY sword.

Program execution time : 0.020999 detik
Goodbye~
```

3.9.

Input

```
HP.

Life potion.

Wooden log.
Mega log, Wooden log.
Ultra log, Mega log.

Epic fish.

Ruby.

Wolf skin.
Zombie eye.
Unicorn horn.
Mermaid hair.
Chip.
Dragon scale.
Dragon essence.

Common lootbox.
Uncommon lootbox.
Rare lootbox.
Epic lootbox.
Edgy lootbox.
Omega lootbox.
Godly lootbox.

EDGY sword, Wooden log, Ultra log.
EDGY armor, Wolf skin, Zombie eye, Unicorn horn, Mermaid hair, Chip.
ULTRA-EDGY sword, EDGY sword, Ultra log, Epic fish, Dragon scale.
ULTRA-EDGY armor, EDGY armor, Ultra log, Ruby, Dragon scale.
OMEGA sword, ULTRA-EDGY sword, HP, Mega log, Dragon scale.
OMEGA armor, ULTRA-EDGY armor, Omega lootbox, Dragon scale.
ULTRA-OMEGA sword, OMEGA sword, Ultra log, Dragon scale.
ULTRA-OMEGA armor, OMEGA armor, Life potion, Dragon scale.

GODLY sword, ULTRA-OMEGA sword, ULTRA-OMEGA armor, Dragon essence, Godly lootbox, Omega lootbox.
```

Output

```
-----Welcome to Course Plan Organizer-----
by Daffa Ananda Pratama Resyaly -- 13519107
Pick your choice: (Enter/Exit)
enter
Reading from file "test9.txt"
Recommended Course Plan:
-----
Semester I      : HP, Life potion, Wooden log, Epic fish, Ruby, Wolf skin, Zombie eye, Unicorn horn, Mermaid hair, Chip, Dragon scale, Dragon essence, Common lootbox, U
ncommon lootbox, Rare lootbox, Epic lootbox, Edgy lootbox, Omega lootbox, Godly lootbox
Semester II     : Mega log, EDGY armor
Semester III    : Ultra log
Semester IV     : EDGY sword, ULTRA-EDGY armor
Semester V      : ULTRA-EDGY sword, OMEGA armor
Semester VI     : OMEGA sword, ULTRA-OMEGA armor
Semester VII    : ULTRA-OMEGA sword
Semester VIII   : GODLY sword.
Program execution time : 0.020996 detik
Goodbye~
```


3.10.

Input

```
x.
```

Output

```
-----Welcome to Course Plan Organizer-----  
by Daffa Ananda Pratama Resyaly -- 13519107  
  
Pick your choice: (Enter/Exit)  
enter  
  
Reading from file "test10.txt"  
  
Recommended Course Plan:  
-----  
Semester I      : x.  
  
Program execution time : 0.011155 detik  
Goodbye~
```

IV. Alamat tempat kode sumber program

Repositori Github:

https://github.com/slarkdarr/Tucil2_13519107.git

V. Checklist

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	✓	