

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



oleh

Andres Jerriel Sinabutar / 13519218

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2021

1. Algoritma *Topological Sort* dan kaitannya dengan pendekatan *Decrease and Conquer*

Algoritma *Decrease and Conquer* adalah metode perancangan algoritma dengan mereduksi persoalan menjadi dua sub persoalan yang lebih kecil, tetapi selanjutnya hanya memproses satu sub persoalan saja. Sesuai dengan namanya, algoritma ini terdiri dari dua tahap, yaitu *decrease* dan *conquer*. *Decrease* adalah tahapan untuk mereduksi persoalan menjadi beberapa persoalan yang lebih kecil dan *conquer* adalah tahapan untuk memproses satu sub persoalan secara rekursif.

Topological Sort merupakan sebuah algoritma yang digunakan untuk mengurutkan sebuah *directed acyclic graph* (DAG) sampai membentuk suatu urutan linear setiap simpul dimana setiap uv yang saling terhubung secara langsung (*direct*), simpul u muncul sebelum simpul v dalam suatu urutan. Algoritma ini banyak digunakan untuk melakukan *job scheduling* yang memiliki dependensi satu dengan lainnya. Aplikasi dari algoritma ini dapat kita temukan pada *build project systems*, *advanced-packaging tool* (apt-get), *task scheduling*, dan permasalahan-permasalahan yang berkaitan dengan *pre-requisite*.

Penulis menyelesaikan persoalan penyusunan rencana kuliah ini dengan bahasa pemrograman Java. Algoritma *Topological Sort* yang diimplementasikan oleh penulis dalam menyelesaikan penyusunan rencana kuliah ini, yaitu:

1. Membaca masukan sebuah file .txt dan membentuknya menjadi sebuah graf. Graf direpresentasikan sebagai sebuah *adjacency list* yang menggunakan tipe data Map. *Adjacency list* disini menyimpan sebuah mata kuliah bertipe *string* sebagai *key* dan seluruh mata kuliah prasyaratnya bertipe *list of string* sebagai *value*. Tipe data Map digunakan karena akan mempermudah akses dan pencarian setiap mata kuliah sehingga tidak perlu melakukan perulangan untuk mendapatkan indeks dari elemen yang dicari.
2. Menghitung jumlah mata kuliah prasyarat dari setiap mata kuliah yang direpresentasikan sebagai jumlah busur yang masuk pada setiap simpul. Jumlah mata kuliah prasyarat dapat dihitung berdasarkan jumlah elemen dalam *value* dari sebuah mata kuliah (*key*).
3. Memilih mata kuliah yang akan dieliminasi/dihapus dari graf. Disini penulis membagi persoalan menjadi sub-sub persoalan yang lebih kecil dengan memilih mata kuliah yang akan dieliminasi/dihapus, yaitu mata kuliah yang tidak memiliki prasyarat atau memiliki jumlah busur yang masuk adalah nol. Mata kuliah yang akan dieliminasi pada tiap semesternya akan dimasukkan ke dalam sebuah *list of string*.
4. Menghapus mata kuliah yang terpilih dari graf. Mata kuliah yang sudah dimasukkan dalam *list*, yaitu yang tidak memiliki mata kuliah prasyarat akan dihapus dari graf. Ini menunjukkan bahwa dalam penyelesaian masalah ini setiap upa-persoalan direduksi beragam pada setiap interaksi algoritma (*decrease by a variable size*) karena variabel reduksi dalam persoalan ini adalah jumlah mata kuliah yang sudah tidak memiliki prasyarat atau jumlah busurnya nol dan kondisi ini bisa dimiliki lebih dari satu mata kuliah dalam satu kali rekursi. Setelah menghapus seluruh mata kuliah yang terdapat dalam *list*, *list* mata kuliah yang sudah tereliminasi ini akan dimasukkan ke dalam *list* hasil yang bertipe *list of list of string*. *List* hasil ini yang akan menyimpan seluruh mata

kuliah yang harus diambil tiap semester sebagai hasil *topological sort* yang sudah dilakukan.

- Langkah 2-4 akan dilakukan secara rekursif graf kosong dan seluruh elemen mata kuliah sudah dipindahkan ke dalam *list* hasil. Setelah proses rekursif selesai, *list* hasil yang sudah berisi mata kuliah yang akan diambil tiap semester akan diiterasi dan ditampilkan sebagai output hasil dari proses *topological sort*. Output yang dihasilkan memiliki format “Semester XX: A, B, C, dst” dengan XX sebagai urutan semester dan A, B, C sebagai mata kuliah yang akan diambil pada semester XX.

2. Source Code dalam Bahasa Pemrograman Java

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;

public class RencanaKuliah {

    // Representasi graf sebagai adjacency list
    private Map<String, List<String>> adjMataKuliah;

    // default ctor
    RencanaKuliah() {
        this.adjMataKuliah = new HashMap<>();
    }

    // Method untuk menambahkan mata kuliah dari graf.
    void addMataKuliah(String matkul) {
        adjMataKuliah.putIfAbsent(matkul, new ArrayList<>());
    }

    // Method untuk menghapus mata kuliah dari graf dan sisi yang berhubungan
    dengannya.
    // Prekondisi: matkul pasti terdapat dalam graf.
    void removeMataKuliah(String matkul) {
        adjMataKuliah.values().stream().forEach(i -> i.remove(matkul));
        adjMataKuliah.remove(matkul);
    }

    // Method untuk menambah sebuah matkul menjadi prasyarat matkul lainnya
    // Prekondisi: matkul pasti terdapat dalam graf. MatkulPrasyarat terdefinisi.
    void addMatkulsbgPrasyarat(String matkul, String matkulPrasyarat) {
        adjMataKuliah.get(matkul).add(matkulPrasyarat);
    }

    // Menghapus matkul sebagai prasyarat dari matkul lainnya.
    // Prekondisi: matkul dan matkulPrasyarat pasti terdapat dalam graf.
    void removeMatkulsbgPrasyarat(String matkul, String matkulPrasyarat) {
        List<String> listOfPrasyarat = adjMataKuliah.get(matkul);
        if (listOfPrasyarat != null) {
            listOfPrasyarat.remove(matkulPrasyarat);
        }
    }

    // Mengembalikan sebuah list of String yang berisi seluruh matkul prasyarat dari
    sebuah masukan matkul tertentu.
    // Prekondisi: matkul pasti terdapat dalam graf.
    List<String> getAllMatkulPrasyarat(String matkul) {
```

```

        return adjMataKuliah.get(matkul);
    }

    // Mengembalikan jumlah matkul prasyarat dari sebuah masukan matkul tertentu.
    // Prekondisi: matkul pasti terdapat dalam graf.
    int getJumlahMatkulPrasyarat(String matkul) {
        return adjMataKuliah.get(matkul).size();
    }

    // Mengecek apakah graf kosong. Mengembalikan true jika graf kosong, dan false
    // jika sebaliknya
    boolean isEmpty() {
        return adjMataKuliah.size() == 0;
    }

    // Membuat Rencana Mata Kuliah dari masukan File txt
    static RencanaKuliah createNewFromFile () {
        RencanaKuliah rencanaKuliah = new RencanaKuliah();

        try {
            Scanner scanner = new Scanner(System.in);
            String filename = scanner.next();
            File myObj = new File(filename);
            Scanner myReader = new Scanner(myObj);
            while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                String[] arr = data.split(",");
                for (int i = 0; i < arr.length; i++) {
                    arr[i] = arr[i].replace(" ", "");
                    arr[i] = arr[i].replaceAll("\\.", "");
                    if (i == 0) {
                        rencanaKuliah.addMataKuliah(arr[i]);
                    } else {
                        rencanaKuliah.addMataKuliah(arr[i]);
                        rencanaKuliah.addMatkulsbgPrasyarat(arr[0], arr[i]);
                    }
                }
            }
            myReader.close();
        } catch (FileNotFoundException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
        return rencanaKuliah;
    }

    // Implementasi Topological Sort
    static void TopologicalSort(RencanaKuliah rencanaKuliah, ArrayList<List<String>>
    result) {
        while (!rencanaKuliah.isEmpty()) {
            ArrayList<String> currSmt = new ArrayList<>(); // List of string untuk
            // menyimpan mata kuliah yang akan dieliminasi
            String[] keys = rencanaKuliah.adjMataKuliah.keySet().toArray(new
            String[0]); // Array of string yang berisi seluruh mata kuliah dalam graf

            for (String matkul : keys) {
                if (rencanaKuliah.getJumlahMatkulPrasyarat(matkul) == 0) { //
                // Pemilihan mata kuliah yang tidak memiliki prasyarat
                currSmt.add(matkul); // Mata kuliah yang tidak memiliki
                // prasyarat dimasukkan ke dalam list
                }
            }
        }
    }

```

```

    }

    if (currSmt.isEmpty()) { // Error handle jika graf yang dimasukkan
        bukanlah DAG
        throw new Error("\nRencana kuliah yang diberikan tidak memenuhi
        prekondisi. \nKuliah dan prerequisitenya tidak berupa Directed Acyclic Graph.");
    }

    for (String matkul : currSmt) { // Mengeliminasi seluruh mata kuliah
        yang tidak memiliki prasyarat
        rencanaKuliah.removeMataKuliah(matkul);
    }
    result.add(currSmt); // Menambahkan list of mata kuliah yang tidak
    memiliki prasyarat ke dalam list hasil
    TopologicalSort(rencanaKuliah, result);
}

// Menampilkan logo selamat datang saat memulai program
static void printLogo() {
    System.out.print("
\n" +
            "      8 8888      ,88' 8 8888          .8.           b.                8
8 8888888888o.      8 8888888888      d8888888o. 88888888 888888888888 8 8888 b.
8 \n" +
            "      8 8888      ,88' 8 8888          .888.         888o.              8
8 8888      ^888.   8 8888      .`8888:' `88.       8 8888      8 8888 888o.
8 \n" +
            "      8 8888      ,88' 8 8888          :88888.        Y888888o.          8
8 8888      `88. 8 8888      8.`8888.   Y8      8 8888      8 8888 Y888888o.
8 \n" +
            "      8 8888      ,88' 8 8888          .`88888.        .`Y8888888o.       8
8 8888      `88 8 8888      `8.`8888.     8 8888      8 8888 .`Y8888888o.
8 \n" +
            "      8 8888      ,88' 8 8888          .8. `88888.      8o. `Y8888888o. 8
8 8888      88 8 8888888888888888 `8.`8888.     8 8888      8 8888 8o.
`Y8888888o. 8 \n" +
            "      8 8888      88'      8 8888          .8`8. `88888.      8`Y8o. `Y888888o8
8 8888      88 8 8888      `8.`8888.     8 8888      8 8888 8`Y8o.
`Y888888o8 \n" +
            "      8 888888<      8 8888          .8' `8. `88888.      8 `Y8o. `Y8888
8 8888      ,88 8 8888      `8.`8888.     8 8888      8 8888 8 `Y8o.
`Y8888 \n" +
            "      8 8888      `Y8.      8 8888          .8' `8. `88888.      8 `Y8o. `Y8
8 8888      ,88' 8 8888      8b `8.`8888.     8 8888      8 8888 8 `Y8o.
`Y8 \n" +
            "      8 8888      `Y8.      8 8888          .8888888888. `88888.      8 `Y8o.`
8 8888      ,o88P' 8 8888      `8b. ;8.`8888     8 8888      8 8888 8
`Y8o.` \n" +
            "      8 8888      `Y8. 8 8888888888888888 .8' `8. `88888. 8 `Yo
8 8888888888P'      8 8888888888888888 `Y8888P ,88P'      8 8888      8 8888 8
`Yo \n");
    System.out.println(" ____
\n" +
            "      / _ \\
+
            "/ /_\\ \\ \\ ____ - - - - - - - - - |
|\n" +
            "| | _ | / __||| ||| '_ \\ / _ \\| '| / _ `||'_ \\ ||'_ \\ ||
|\n" +

```



```

    }
    while (input >= 4) {
        s += "IV";
        input -= 4;
    }
    while (input >= 1) {
        s += "I";
        input -= 1;
    }
    return s;
}

// Print hasil sesuai dengan format yang ada
static void printSolution(ArrayList<List<String>> result) {
    System.out.println("Berikut adalah rencana kuliah Anda: ");
    for (int cnt = 0; cnt < result.size(); cnt++) {
        System.out.print("Semester " + IntegerToRomanNumeral(cnt + 1) + ": ");
        List<String> currSmt = result.get(cnt);
        for (int i = 0; i < currSmt.size(); i++) {
            if (i == 0) {
                System.out.print(currSmt.get(i));
            } else {
                System.out.print(", " + currSmt.get(i));
            }
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    printLogo();
    RencanaKuliah rencanaKuliah = createNewFromFile();
    ArrayList<List<String>> result = new ArrayList<>();
    TopologicalSort(rencanaKuliah, result);
    printSolution(result);
}
}

```

3. Input & Output

Isi file teks (.txt)	Output
C7,C4,C5. C6,C5,C3. C5,C3. C4,C2,C1. C3. C2. C1.	<pre> Selamat datang! Masukkan nama beserta direktori file daftar kuliah Anda: test1.txt Berikut adalah rencana kuliah Anda: Semester I: C3, C1, C2 Semester II: C4, C5 Semester III: C6, C7 </pre>

A. B,A. C,A,F. D,I,B. E,C,G. F,D. G. H,D,E. I,G. J,B,I.	<p>Selamat datang!</p> <p>Masukkan nama beserta direktori file daftar kuliah Anda:</p> <p><i>test2.txt</i></p> <p>Berikut adalah rencana kuliah Anda:</p> <p>Semester I: A, G</p> <p>Semester II: B, I</p> <p>Semester III: D, J</p> <p>Semester IV: F</p> <p>Semester V: C</p> <p>Semester VI: E</p> <p>Semester VII: H</p>
5,6,2. 7,4,5,6. 4,2. 6,3. 2,1. 3,1. 1.	<p>Selamat datang!</p> <p>Masukkan nama beserta direktori file daftar kuliah Anda:</p> <p><i>test3.txt</i></p> <p>Berikut adalah rencana kuliah Anda:</p> <p>Semester I: 1</p> <p>Semester II: 2, 3</p> <p>Semester III: 4, 6</p> <p>Semester IV: 5</p> <p>Semester V: 7</p>
D,B,G. G. C,B. B,A. E,C,B,D. F,E. A.	<p>Selamat datang!</p> <p>Masukkan nama beserta direktori file daftar kuliah Anda:</p> <p><i>test4.txt</i></p> <p>Berikut adalah rencana kuliah Anda:</p> <p>Semester I: A, G</p> <p>Semester II: B</p> <p>Semester III: C, D</p> <p>Semester IV: E</p> <p>Semester V: F</p>
D,B,C. B,A. E,B,D. F. C,F. A,C,F.	<p>Selamat datang!</p> <p>Masukkan nama beserta direktori file daftar kuliah Anda:</p> <p><i>test5.txt</i></p> <p>Berikut adalah rencana kuliah Anda:</p> <p>Semester I: F</p> <p>Semester II: C</p> <p>Semester III: A</p> <p>Semester IV: B</p> <p>Semester V: D</p> <p>Semester VI: E</p>

D,B,C. E,C. F,D,C,E. B,A. C,A. A.	Selamat datang! Masukkan nama beserta direktori file daftar kuliah Anda: <code>test6.txt</code> Berikut adalah rencana kuliah Anda: Semester I: A Semester II: B, C Semester III: D, E Semester IV: F
0. 1,0. 2,0. 3,0,2. 4,6,3. 5,0,3. 6,0. 7,8. 8. 9,6,4. 10,9. 11,9. 12,9,11.	Selamat datang! Masukkan nama beserta direktori file daftar kuliah Anda: <code>test7.txt</code> Berikut adalah rencana kuliah Anda: Semester I: 0, 8 Semester II: 1, 2, 6, 7 Semester III: 3 Semester IV: 4, 5 Semester V: 9 Semester VI: 11, 10 Semester VII: 12
1. 2,1. 3,2. 4,1. 5,4. 6,4. 7,6,2. 8,7,5. 9,4,5.	Selamat datang! Masukkan nama beserta direktori file daftar kuliah Anda: <code>test8.txt</code> Berikut adalah rencana kuliah Anda: Semester I: 1 Semester II: 2, 4 Semester III: 3, 5, 6 Semester IV: 7, 9 Semester V: 8
C0, C1. C1, C0. C2, C0. C3, C0, C2. C4, C6, C3. C5, C0, C3. C6, C0. C7, C8. C8. C9, C6, C4. C10, C9. C11, C9. C12, C9, C11.	Selamat datang! Masukkan nama beserta direktori file daftar kuliah Anda: <code>test9.txt</code> Exception in thread "main" java.lang.Error: Rencana kuliah yang diberikan tidak memenuhi prekondisi. Kuliah dan prerequisitenya tidak berupa Directed Acyclic Graph. at RencanaKuliah.TopologicalSort(RencanaKuliah.java:103) at RencanaKuliah.TopologicalSort(RencanaKuliah.java:110) at RencanaKuliah.TopologicalSort(RencanaKuliah.java:110) at RencanaKuliah.main(RencanaKuliah.java:221)

4. Alamat Github Kode Program

Berikut adalah pranala akses menuju repositori dari kode program ini:

<https://github.com/andresjerriels/Tucil2Stima>

5. *Check List Program*

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	✓	