LAPORAN HASIL TUGAS KECIL 2

Penyusunan Rencana Kuliah dengan *Topological Sort* "SUPREMO PLAN" dalam Bahasa *C++*

DIAJUKAN UNTUK MEMENUHI TUGAS MATA KULIAH

IF2211 – Strategi Algoritma

SEMESTER II TAHUN 2020/2021

Disusun oleh :

Karel Renaldi

13519180

K04

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2021

I. Algoritma *Topological Sort*

1. Buat representasi DAG(*Directed Acyclic Graph*) dan representasi untuk menyimpan nilai derajat masuk dari setiap simpul / *vertices*
2. Pilih semua simpul / *vertices* yang memiliki derajat masuk sebesar 0 lalu buang simpul-simpul tersebut dari graf.
3. Lakukan perubahan derajat masuk kepada simpul / vertices dengan mengurangi derajat simpul-simpul tersebut yang berhubungan dengan simpul yang dibuang dari graf.
4. Ulangi langkah (2) dan (3) hingga graf kosong, jika graf sudah kosong maka algoritma pengurutan sudah selesai.

Kaitan algoritma *topological sort* ini dengan pendekatan Decrease and Conquer adalah pada pengimplementasian *topological sort*, Algoritma yang saya buat ini adalah mereduksi tiap” simpul yang memiliki derajat masuk 0 di graf diulangi terus menerus sampai semua simpul sudah dikunjungi dan terurut.

II. *Source Code* Program

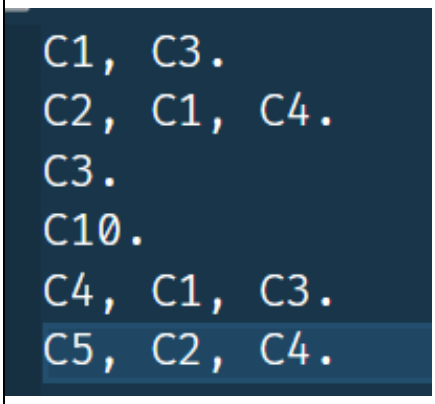
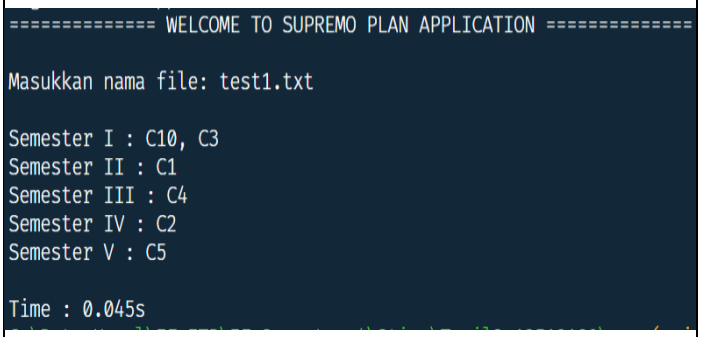
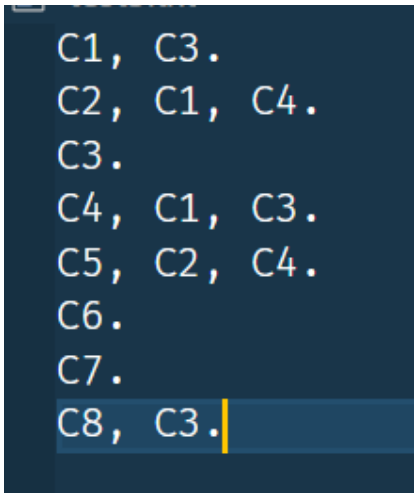
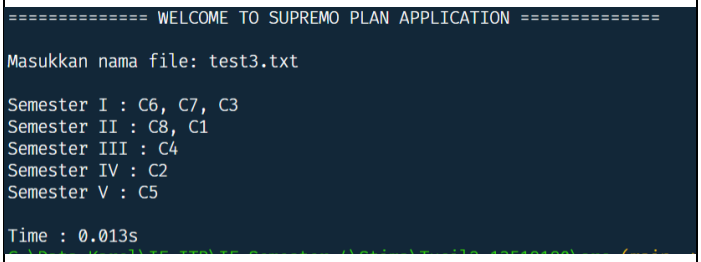
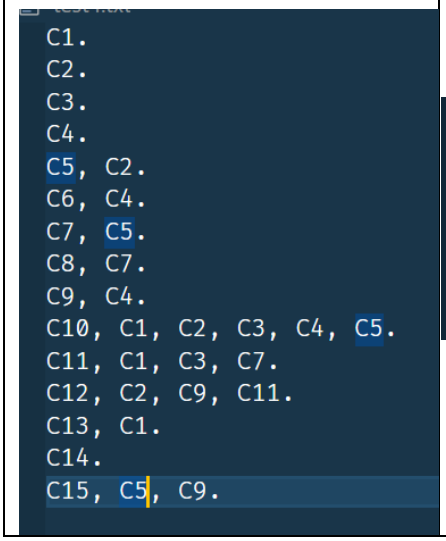
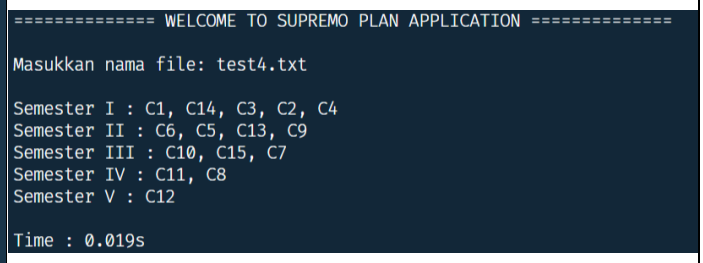
Dibawah ini adalah kode program dari aplikasi “SUPREMO PLAN”:

```
// Karel Renaldi / 13519180 / K-04 / Tucil Stima 2
#include <bits/stdc++.h>
using namespace std;

// Global variabel
int semester = 0;

void split(string const &str, const char *delim, vector<string> &out)
{
    char *token = strtok(const_cast<char *>(str.c_str()), delim);
    while (token != nullptr)
    {
        out.push_back(string(token));
        token = strtok(nullptr, delim);
    }
}

string convert_roman_numeral(int &number)
{
    int standard_numeral[] = {1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500,
 900, 1000};
    string roman_numeral[] = {"I", "IV", "V", "IX", "X", "XL", "L", "XC"
, "C", "CD", "D", "CM", "M"};

    int remainder = number, curr_pointer = 12; // roman_numeral (arr siz
e - 1)
    string res = "";

    while (remainder > 0)
    {
        int div_result = remainder / standard_numeral[curr_pointer];
        while (div_result--)
            res += roman_numeral[curr_pointer];
        remainder = remainder % standard_numeral[curr_pointer];
        curr_pointer--;
    }

    return res;
}

void create_graph(string filename, unordered_map<string, int> &vertices_
degree_in, unordered_map<string, unordered_set<string>> &graph)
{
    string text;
    ifstream file(filename);

    while (getline(file, text))
```

```
    {
        vector<string> parse_text;
        unordered_set<string> vertices;

        text.pop_back();
        split(text, ", ", parse_text);

        graph.insert({parse_text[0], vertices});

        for (int i = 1; i < parse_text.size(); i++)
            graph[parse_text[0]].insert(parse_text[i]);
    }

    for (auto adj : graph)
        vertices_degree_in[adj.first] = adj.second.size();
}

void topological_sort(unordered_map<string, int> &vertices_degree_in, un
ordered_map<string, unordered_set<string>> &graph)
{
    vector<string> erase_key;

    semester++;
    cout << "Semester " << convert_roman_numeral(semester) << " : ";
    for (auto vertices = vertices_degree_in.begin(); vertices != vertice
s_degree_in.end(); vertices++)
    {
        if (vertices->second == 0)
        {
            erase_key.push_back(vertices->first);
            cout << vertices->first << ", ";
        }
    }
    cout << "\b\b"
         << " " << endl;

    // Decrease
    for (string key : erase_key)
    {
        vertices_degree_in.erase(key);
        for (auto adj : graph)
            if (adj.first != key && adj.second.count(key) != 0)
                vertices_degree_in[adj.first]--;
    }

    // Recursive
    if (vertices_degree_in.size() != 0)
        topological_sort(vertices_degree_in, graph);
}

int main()
{
    unordered_map<string, int> vertices_degree_in;
```

```
    unordered_map<string, unordered_set<string>> graph;
    string filename = "test1.txt", path = "../test/";

    cout << "============== WELCOME TO SUPREMO PLAN APPLICATION ========
======" << endl
         << endl;

    cout << "Masukkan nama file: ";
    cin >> filename;
    cout << endl;

    // Parsing text to graph
    create_graph(path + filename, vertices_degree_in, graph);

    clock_t start = clock();
    double duration;

    // Implement topological sort algorithm
    topological_sort(vertices_degree_in, graph);

    duration = (clock() - start) / (double)CLOCKS_PER_SEC;

    cout << endl
         << "Time : " << duration << "s";

    return 0;
}
```

III. *Screenshot Input* dan *Output*

Input	Output
	
	
	

Input	Output
C1. C2. C3, C1. C4. C5. C6, C3, C5. C7, C6. C8. C9, C1, C7. C10, C1, C6, C7. C11, C3, C6, C8. C12, C1, C3, C6. C13, C1, C9. C14, C6, C12. C15, C10, C12.	========== WELCOME TO SUPREMO PLAN APPLICATION ========== Masukkan nama file: test5.txt Semester I : C1, C5, C8, C2, C4 Semester II : C3 Semester III : C6 Semester IV : C11, C12, C7 Semester V : C10, C14, C9 Semester VI : C15, C13
C1. C2. C3. C4, C1, C2. C5, C4. C6. C7, C2, C6. C8, C7. C9. C10, C2, C9.	========== WELCOME TO SUPREMO PLAN APPLICATION ========== Masukkan nama file: test6.txt Semester I : C6, C1, C9, C3, C2 Semester II : C4, C10, C7 Semester III : C5, C8 Time : 0.022s
C1. C2, C1. C3. C4, C2, C3. C5, C3, C4. C6. C7, C1, C2, C4. C8. C9, C1, C7. C10, C5.	========== WELCOME TO SUPREMO PLAN APPLICATION ========== Masukkan nama file: test7.txt Semester I : C6, C1, C8, C3 Semester II : C2 Semester III : C4 Semester IV : C5, C7 Semester V : C9, C10

```
C1.
C2.
C3, C1.
C4.
C5, C2.
C6.
C7, C3, C6.
C8, C1, C3, C4.
C9, C3.
C10, C5, C9.
```

```
============== WELCOME TO SUPREMO PLAN APPLICATION ==============

Masukkan nama file: test8.txt

Semester I : C6, C4, C1, C2
Semester II : C5, C3
Semester III : C9, C8, C7
Semester IV : C10

Time : 0.013s
```

```
C1.
C2, C1.
C3.
C4, C1.
C5, C2, C3.
C6.
C7, C5.
C8.
C9, C3, C7.
C10, C1.
```

```
============== WELCOME TO SUPREMO PLAN APPLICATION ==============

Masukkan nama file: test9.txt

Semester I : C6, C1, C8, C3
Semester II : C4, C10, C2
Semester III : C5
Semester IV : C7
Semester V : C9

Time : 0.02s
```

IV. Alamat *Source Code* Program

Kode sumber program terdapat pada folder *src* dengan nama file tucil2.cpp. Terdapat file *executable code* pada folder bin. Untuk membuat kode program menjadi file *executable* gunakan perintah “g++ tucil2.cpp -o ../bin/tucil2”. Untuk menjalankan file *executable* tersebut perintah yang harus dijalankan adalah “cd ./src” lalu “./tucil2”.

Untuk *backup* terdapat pada link Github berikut ini:

https://github.com/karelrenaldi/Tucil2_13519180

Poin	Ya	Tidak
1. Program berhasil dikompilasi	V	
2. Program berhasil *running*	V	
3. Program dapat menerima berkas input dan menuliskan output	V	
4. Luaran sudah benar untuk semua kasus input	V	