

Laporan Penyusunan Rencana Kuliah dengan *Topological Sort (Penerapan *Decrease and Conquer*)*

TUGAS KECIL 2 STRATEGI ALGORITMA

Oleh:

Marcello Faria

13519086



PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

A. Algoritma *Topological Sort* dengan pendekatan *Decrease and Conquer*

Decrease and conquer memiliki definisi sebuah metode perancangan algoritma dengan mereduksi suatu permasalahan/persoalan menjadi dua / lebih upa-persoalan atau *sub-problem* yang lebih kecil dari permasalahan sebelumnya, setelah itu dilanjutkan dengan memproses salah satu dari kedua upa-persoalan tersebut yang lebih mendekati solusi. Tahapan dari decrease and conquer adalah:

1. Decrease: mereduksi persoalan tersebut menjadi beberapa persoalan yang lebih (secara umum 2 buah persoalan)
2. Conquer: memproses salah satu upa-persoalan secara rekursif

Dalam penerapannya, *Decrease and Conquer* memiliki tiga buah varian, yaitu *Decrease by a constant*, *Decrease by a constant factor* dan *Decrease by a variable size*.

Salah satu persoalan yang dapat digunakan dengan menggunakan pendekatan *Decrease and Conquer* adalah *Topological Sort*. *Topological Sort* merupakan penyusunan dari simpul yang tidak berkontradiksi terhadap *dependencies*-nya. Persoalan *Topological Sort* ini dapat diselesaikan melalui pendekatan *Decrease and Conquer* dengan varian *Decrease by a Constant*. Pada varian tersebut, ukuran dari sebuah instansi akan dikurangi dengan konstanta yang sama setiap iterasi dari algoritma. Secara umum, konstanta tersebut bernilai satu, walaupun terkadang terdapat konstanta dengan nilai lain.

Berbagai persoalan dengan permasalahan Directed Acyclic Graph (DAG) dapat kita selesaikan dengan mengimplementasikan *Topological Sort* ini. Salah satu permasalahannya adalah rekomendasi penyusunan rencana kuliah dengan beberapa mata kuliah yang memiliki prasyarat tertentu. Berikut Langkah-langkah yang merupakan garis besar dari implementasi program di dokumen ini dengan memanfaatkan algoritma *Decrease and Conquer*:

1. Aplikasi menerima input berupa nama file yang dijadikan string, file tersebut berisi Daftar mata kuliah beserta prasyaratnya dan dengan format yang telah ditentukan yaitu antara kode mata kuliah satu dan yang lain dipisahkan dengan tanda koma (,) dan diakhiri dengan tanda titik (.), dari file teks yang telah diinput, lalu akan diproses mata kuliah apa saja yang dapat diambil di semester 1, 2 dan seterusnya.
2. Proses yang dilakukan yaitu memetakan setiap mata kuliah menjadi string, dan dimasukkan kedalam vector of vector string. Sebelum dimasukkan, setiap kolom pertama (mata kuliah yang memiliki prasyarat) akan dikaitkan dulu dengan null, untuk menandai setiap baris pada kolom pertama.
3. Setelah dipetakan, terdapat suatu variable `total_size` yang menampung seluruh jumlah prerequisite dari suatu mata kuliah. Setelah itu, dilakukan pemanggilan terhadap fungsi `topo_sort()`.
4. Pada fungsi ini, dilakukan iterasi setiap baris untuk mencari dimana mata kuliah yang telah berhubungan dengan null yang berarti bahwa mata kuliah tersebut dapat diambil pada semester ini, karena tidak ada prasyarat mata kuliah lain. Lalu mata kuliah tersebut ditandai dengan `visited` dan dimasukkan terhadap vector of vector `erased` (vector 2 dimensi yang menampung semester yang dapat diambil pada semester ke `i` dengan `i` menandakan baris) dan juga dimasukkan kedalam `currMK` (vector mata kuliah yang dapat diambil pada semester ini).

5. Setelah itu dilakukan eliminasi mata kuliah dengan fungsi `topo_sortUtil`, yaitu menghapus semua prasyarat yang terdapat pada setiap baris mata kuliah (karena sudah diambil pada semester ini). Pada fungsi `topo_sort` ini, dilakukan metode rekursif setelah penghapusan, yaitu dengan melakukan pengecekan apabila vector of vector yang diinput diawal dari file teks telah berjumlah sama dengan null/ jumlah baris (kasus dasar), jika iya, maka dilakukan pengembalian (return) dan fungsi `topo_sort` telah selesai, namun jika tidak, `topo_sort` kembali dipanggil sampai menemukan kasus dasar.

B. Source Code

13519086.cpp

```
1  #include "13519086-topo_sort.hpp"
2  using namespace std;
3
4  int main()
5  {
6      // menjalankan program parsing
7      mains();
8  }
9
```

Main cpp yang memanggil fungsi parsing

13519086-topo_sort.hpp

```
1  #ifndef TOPO_SORT_HPP
2  #define TOPO_SORT_HPP
3  using namespace std;
4  #include <bits/stdc++.h>
5
6  typedef struct {
7      map< string, list< string > > adjList; // vektor untuk menampung mata kuliah dari file teks
8      vector<vector<string>> erased; // vektor untuk menampung mata kuliah yang sudah diambil, baris menandai semester
9      vector<string> currMK, solved; // vektor solved untuk menampung mata kuliah akhir yang sudah dihapus saat iterasi terakhir
10     int total_size = 0, semester = 0;
11     bool vis[10000];
12 } MataKuliah;
13
14 void topo_sortUtil(MataKuliah * M);
15
16 void topo_sort(MataKuliah * M);
17
18 string ConvertToRomawi(int n);
19
20 void mains();
21
22 #endif
23
```

File untuk membuat struktur data mata kuliah

135319086-topo_sort.cpp

```

1  #include "135319086-topo_sort.hpp"
2
3  void topo_sortUtil(MataKuliah * M)
4  {
5      //program untuk mengeliminasi node yang terdapat dalam semester sekarang
6      for(auto i : M->currMK)
7      {
8          for(auto& val : M->adjList)
9          {
10             auto& adjV = val.second;
11             auto itr = adjV.begin();
12             while (itr != adjV.end())
13             {
14                 if(*itr == i)
15                 {
16                     // hapus node jika node sekarang terdapat dalam list semester sekarang
17                     adjV.remove(i);
18                     M->total_size--;
19                     break;
20                 }
21                 ++itr;
22             }
23         }
24     }
25     return;
26 }
27
28 void topo_sort(MataKuliah * M)
29 {
30     M->currMK.clear();
31     for(auto val = M->adjList.begin(); val != M->adjList.end(); val++)
32     {
33         string v = (*val).first;
34         int idx = distance(M->adjList.begin(), val);
35         // disini jika kolom pertama tidak dikaitkan dengan null, maka akan dilewati oleh iterasi vektor
36         // dicek juga apabila sudah visited/belum jika sudah maka dilewati
37         if ((*val).second.size() == 1 && !M->vis[+idx])
38         {
39             M->solved.push_back(v);
40             M->vis[idx] = true; // mengeset idx sekarang menjadi visited
41             M->erased[M->semester].push_back(v);
42             M->currMK.push_back(v);
43         }
44         else
45         {
46             ++val;
47         }
48     }
49
50     topo_sortUtil(M);
51     M->semester++;
52     if(M->total_size != M->adjList.size())
53     {
54         // jika jumlah matkul sekarang belum sama dengan jumlah null maka akan dilakukan
55         // pemanggilan ulang topo_sort
56         topo_sort(M);
57     }
58     else
59     {
60         // kasus basis, jika jumlah matkul sekarang = jumlah null maka return
61         return;
62     }
63 }
64
65 string ConvertToRomawi(int n)
66 {
67     // program untuk mengonversi integer ke romawi
68     string arrRomawi[] = {"X", "IX", "V", "IV", "I"};
69     int val[] = {10, 9, 5, 4, 1};
70     string res = "";
71     for (int i = 0; i < 5; i++)
72     {
73         while(n - val[i] >= 0)

```

```

72     while(n - val[i] > -1)
73     {
74         n -= val[i];
75         res = res + arrRomawi[i];
76     }
77 }
78 return res;
79 }
80
81 void mains() {
82     fstream input_stream;
83     ofstream output_stream;
84     MataKuliah M;
85     string filename, str, v1, v2; // filename untuk menampung string nama file, str untuk menampung array of string per baris
86                                     // v1 dan v2 untuk menampung vertices
87     vector<string> input_data; // input_data untuk vector 2 dimensi dari file input
88     cout<<"Masukkan input nama file : ";
89     cin >> filename;
90
91     memset(M.vis, false, sizeof(M.vis));
92     input_stream.open(filename, ios::in);
93     output_stream.open("output.txt");
94     // Menginput per baris, lalu dimasukkan ke vector 2 dimensi
95     while(getline(input_stream, str))
96     {

```

```

97         input_data.push_back(str);
98     }
99     M.erased.resize(2001, vector<string>(input_data.size()));
100     for(int i=0; i<input_data.size(); i++)
101     {
102         int j = 0;
103         int count = 0;
104         while(j < input_data[i].size())
105         {
106             int k = j;
107             vector<char> mk; //Vektor untuk menampung node mata kuliah
108             while(k < input_data[i].size())
109             {
110                 // jika menemukan titik atau koma, maka lanjut ke node baru
111                 if(input_data[i][k] == ',' || input_data[i][k] == '.')
112                 {
113                     string s(mk.begin(), mk.end());
114                     if(count == 0)
115                     {
116                         // setiap baris kolom pertama akan dikaitkan dengan null untuk menandai
117                         // pada vektor dan digunakan untuk kasus basis pada rekursif
118                         v1 = s;
119                         v2 = "null";
120                     }
121                     else
122                     {
123                         v2 = s;
124                     }
125                     M.adjList[v1].push_back(v2);
126                     count++;
127                     j = k+1;
128                     break;
129                 }
130                 mk.push_back(input_data[i][k]);
131                 k++;
132             }
133             j++;
134         }
135     }
136     for(auto itr : M.adjList)
137     {
138         //total_
139         M.total_size += itr.second.size();
140     }
141     clock_t tStart = clock();
142     float time;
143     topo_sort(&M);
144 }

```

```

145     for(auto i = M.adjList.begin(); i!= M.adjList.end();)
146     {
147         auto it = find (M.solved.begin(), M.solved.end(), (*i).first);
148         // ini merupakan iterasi untuk menambahkan vektor mata kuliah pada semester akhir
149         // karena tidak dilewati oleh iterasi (kondisi basis sudah ditemukan dahulu)
150         if(it == M.solved.end())
151         {
152             M.erased[M.semester].push_back((*i).first);
153         }
154         i++;
155     }
156     for(int i =0 ;i<=M.semester;i++)
157     {
158         cout<<"Semester "<<ConvertToRomawi(i+1)<<" : ";
159         output_stream<<"Semester "<<ConvertToRomawi(i+1)<<" : ";
160         for(int j=0;j<M.erased[i].size();j++)
161         {
162             if(M.erased[i][j] == "")
163                 continue;
164             cout<<M.erased[i][j];
165             output_stream<<M.erased[i][j]<<endl;
166             if(j!=M.erased[i].size()-1)
167             {
168                 cout<<",";
169                 output_stream<<",";
170             }
171             else
172             {
173                 cout<<".";
174                 output_stream<<".";
175             }
176         }
177         cout<<endl;
178     }
179     time=(double)(clock() - tStart)/CLOCKS_PER_SEC;
180     cout << "Waktu yang dibutuhkan : " << time << " detik" << endl;
181     output_stream << "Waktu yang dibutuhkan : " << time << " detik" << endl;
182 }

```

C. INPUT/OUTPUT

INPUT

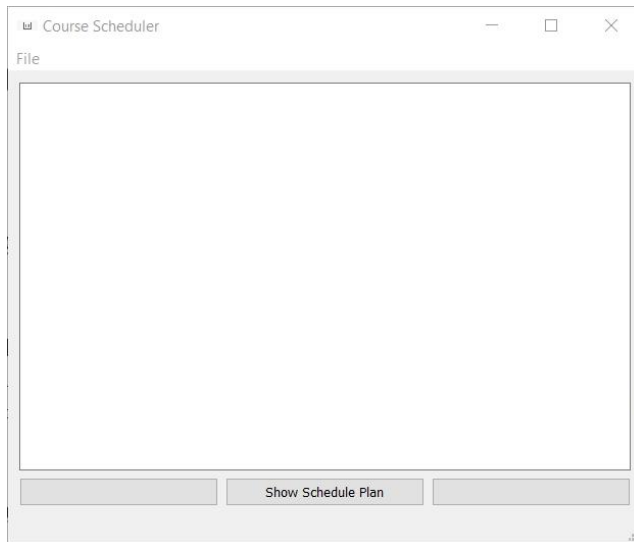
Output

C1. C2. C3. C4, C1, C2. C5, C4. C6. C7, C2, C6. C8, C7. C9. C10, C2, C9.	Masukkan input nama file : 1.txt Semester I : C1, C2, C3, C6, C9. Semester II : C10, C4, C7. Semester III : C5, C8. Waktu yang dibutuhkan : 0.003 detik
C1. C2, C1. C3. C4, C2, C3. C5, C3, C4. C6. C7, C1, C2, C4. C8. C9, C1, C7. C10, C5.	Masukkan input nama file : 2.txt Semester I : C1, C3, C6, C8. Semester II : C2. Semester III : C4. Semester IV : C5, C7. Semester V : C10, C9. Waktu yang dibutuhkan : 0.003 detik
C1. C2. C3, C1. C4. C5, C2. C6. C7, C3, C6. C8, C1, C3, C4. C9, C3. C10, C5, C9.	Masukkan input nama file : 3.txt Semester I : C1, C2, C4, C6. Semester II : C3, C5. Semester III : C7, C8, C9. Semester IV : C10. Waktu yang dibutuhkan : 0.005 detik
C1. C2, C1. C3. C4, C1. C5, C2, C3. C6. C7, C5. C8. C9, C3, C7. C10, C1.	Masukkan input nama file : 4.txt Semester I : C1, C3, C6, C8. Semester II : C10, C2, C4. Semester III : C5. Semester IV : C7. Semester V : C9. Waktu yang dibutuhkan : 0.006 detik

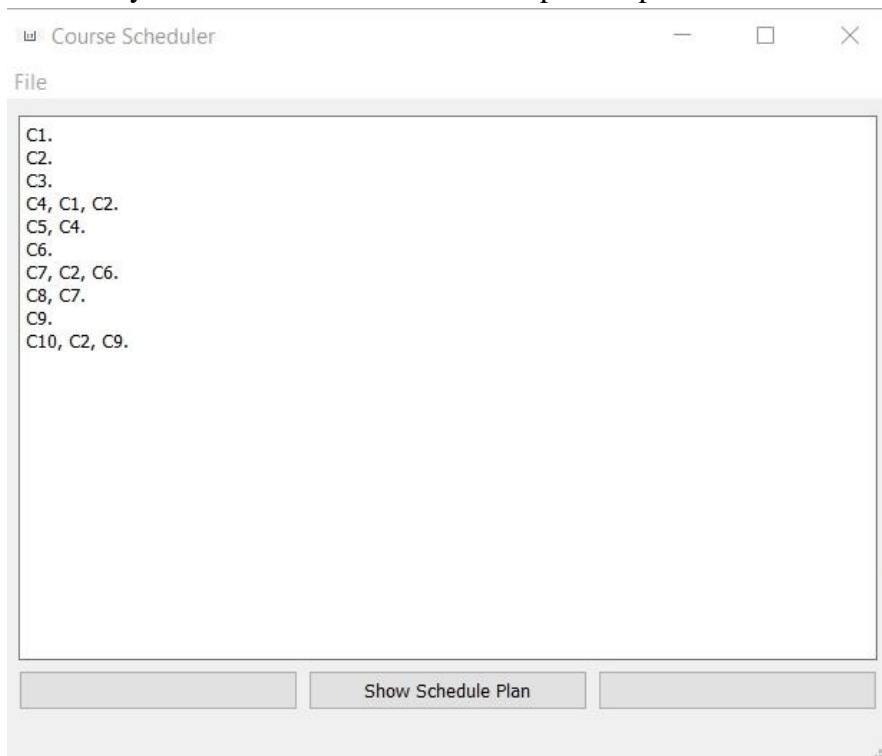
<p>Fisika Dasar. Statistika Elementer. Kalkulus 1. Bahasa Inggris. Logika. Pemecahan Masalah Mat. Pemrograman Komputer. Aljabar Matriks. Kalkulus 2, Kalkulus 1. Estetika. Agama Katolik/Fenomonologi Agama. Matematika Diskrit, Pemecahan Masalah Mat. Komputasi Statistika, Pemrograman Komputer, Statistika Elementer. Etika. Kalkulus Vektor, Kalkulus 2. Teori Peluang, Kalkulus 2. Teori Suku Bunga, Kalkulus 2. Metoda Matematika, Kalkulus 2. Aljabar Linear, Aljabar Matriks. Komputasi Matematika, Statistika Elementer, Kalkulus Vektor, Aljabar Matriks. Optimasi, Aljabar Matriks, Kalkulus Vektor. Persamaan Differensial Biasa, Aljabar Matriks, Kalkulus 2. Statistika Matematika, Teori Peluang. Metoda Numerik, Komputasi Matematika. Fungsi Kompleks, Kalkulus Vektor. Proses Stokastik, Teori Peluang. Analisis Real, Fungsi Kompleks. Pemodelan Matematika, Komputasi Statistika, Komputasi Matematika, Aljabar Matriks, Kalkulus Vektor, Persamaan Differensial Biasa.</p>	<p>Masukkan input nama file : 5.txt Semester I : Agama Katolik/Fenomonologi Agama, Aljabar Matriks, Bahasa Inggris, Estetika, Etika, Fisika Dasar, Kalkulus 1, Logika, Pemecahan Masalah Mat, Pemrograman Komputer, Statistika Elementer. Semester II : Aljabar Linear, Kalkulus 2, Komputasi Statistika, Matematika Diskrit. Semester III : Kalkulus Vektor, Metoda Matematika, Persamaan Differensial Biasa, Teori Peluang, Teori Suku Bunga. Semester IV : Fungsi Kompleks, Komputasi Matematika, Optimasi, Proses Stokastik, Statistika Matematika. Semester V : Analisis Real, Metoda Numerik, Pemodelan Matematika. Waktu yang dibutuhkan : 0.011 detik</p>
<p>C1, C3. C2, C1, C4. C3. C4, C1, C3. C5, C2, C4.</p>	<p>Masukkan input nama file : 6.txt Semester I : C3. Semester II : C1. Semester III : C4. Semester IV : C2. Semester V : C5. Waktu yang dibutuhkan : 0.006 detik</p>
<p>Kalkulus Peubah Banyak, Kalkulus Differensial, Kalkulus Integral. Kalkulus Integral, Kalkulus Differensial. Kalkulus Differensial. Analisis Vektor, Kalkulus Differensial, Kalkulus Integral. Geometri Analitik Ruang, Geometri Ruang, Geometri Analitik Bidang. Geometri Ruang, Geometri Bidang. Geometri Analitik Bidang, Geometri Bidang. Geometri Bidang. Program Linear, Aljabar Matriks, Aljabar Linear. Aljabar Matriks. Aljabar Linear.</p>	<p>Masukkan input nama file : 7.txt Semester I : Aljabar Linear, Aljabar Matriks, Geometri Bidang, Kalkulus Differensial. Semester II : Geometri Analitik Bidang, Geometri Ruang, Kalkulus Integral, Program Linear. Semester III : Analisis Vektor, Geometri Analitik Ruang, Kalkulus Peubah Banyak. Waktu yang dibutuhkan : 0.005 detik</p>
<p>TIF5209, TIF5101. TIF5101. TIF5212, TIF5103. TIF5103. TIF5213, UNIS17104. UNIS17104. TIF5315, TIF5214. TIF5214. TIF5321, TIF5210. TIF5210. TIF5425, TIF5317. TIF5317. TIF5638, TIF5425. TIF5744, TIF5638.</p>	<p>Masukkan input nama file : 8.txt Semester I : TIF5101, TIF5103, TIF5210, TIF5214, TIF5317, UNIS17104. Semester II : TIF5209, TIF5212, TIF5213, TIF5315, TIF5321, TIF5425. Semester III : TIF5638. Semester IV : TIF5744. Waktu yang dibutuhkan : 0.006 detik</p>

D. Implementasi pada Gui:

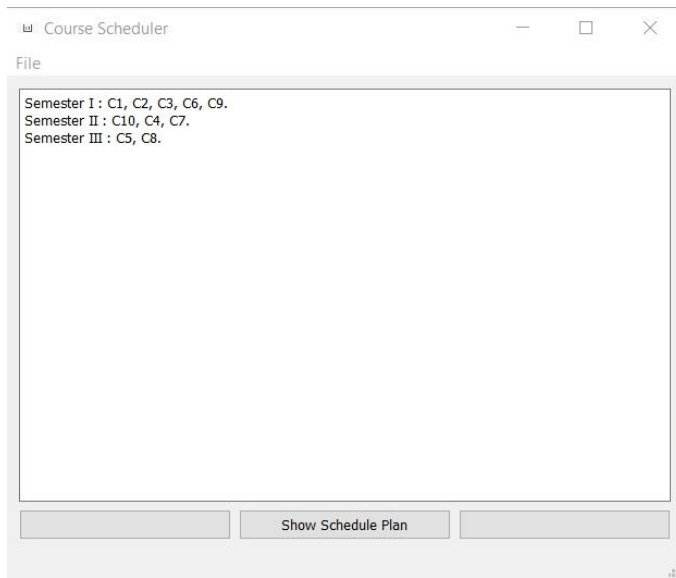
Berikut merupakan gui yang dibuat dengan menggunakan Qt creator agar memudahkan pengguna dalam melakukan testing dan melihat hasil :



Pada ujung kiri atas terdapat file, jika pengguna ingin memasukkan input file teks, dapat menekan file lalu open setelah itu pilih file teks berisi format yang telah dijelaskan sebelumnya. Setelah itu akan muncul tampilan seperti ini :



Setelah itu tekan tombol Show Schedule Plan dibawah, lalu akan muncul hasilnya :



E. Link Alamat Drive:

<https://drive.google.com/drive/folders/1YNLcNgvr0VzX29hJ9EGlh7X1RCRjgPri?usp=sharing>

F. Tabel Ceklist:

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. operand	√	