

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kurikulum menjadi komponen acuan oleh setiap satuan pendidikan. Kurikulum berkembang sejalan dengan perkembangan teori dan praktek pendidikan, selain itu juga bervariasi sesuai dengan aliran atau teori pendidikan yang dianut pemangku kebijakan. Kurikulum memiliki kedudukan yang sangat sentral dalam keseluruhan proses pendidikan. Kurikulum juga mengarahkan segala bentuk aktivitas pendidikan kepada tercapainya tujuan-tujuan pendidikan. Sehingga kurikulum menjadi elemen pokok dalam sebuah layanan program pendidikan. Kurikulum juga memiliki peranan penting dalam pendidikan, kaitannya yaitu dengan penentuan arah, isi, dan proses pendidikan yang pada akhirnya menentukan macam dan kualifikasi lulusan suatu lembaga pendidikan. Dengan kata lain kurikulum menjadi syarat mutlak dari pendidikan dan kurikulum merupakan bagian yang tak terpisahkan dari pendidikan dan pengajaran. Sehingga sangatlah sulit dibayangkan bagaimana bentuk pelaksanaan suatu pendidikan tanpa adanya kurikulum.

Pada dasarnya kurikulum tidak hanya berisikan tentang petunjuk teknis materi pembelajaran. Kurikulum merupakan sebuah program terencana dan menyeluruh, yang secara tidak langsung menggambarkan manajemen pendidikan suatu bangsa. Dengan begitu otomatis kurikulum memegang peran yang sangat penting dan strategis dalam kemajuan dunia pendidikan suatu negara.

Efektifitas implementasi dan pengembangan kurikulum di lapangan sangatlah bergantung pada kompetensi sumber daya yang tersedia di universitas untuk memfasilitasi pengajar dalam mengartikulasi topik-topik yang termuat dalam kurikulum. Pengajar yang menjalankan segala sesuatu yang terjadi dalam kelasnya. Sehingga keberhasilan pengembangan kurikulum juga bergantung pada manajemen dari setiap pengajar. Kurikulum sendiri pada setiap satuan pendidikan sebagai alat penggerak pendidikan. Dengan kesesuaian dan ketepatan setiap komponen yang ada dalam kurikulum diharapkan sasaran dan tujuan pendidikan akan tercapai secara maksimal.

Dikarenakan peran kurikulum sendiri sangatlah penting dalam upaya pencapaian tujuan pendidikan nasional, maka pemerintah Indonesia telah melakukan berbagai macam upaya untuk merevisi, mengembangkan dan menyempurnakan desain kurikulum pendidikan nasional Indonesia untuk bisa menghasilkan proses dan produk pendidikan yang bermutu dan kompetitif. Kurikulum tidak bersifat statis, sehingga munculnya kurikulum disesuaikan dengan perkembangan zaman dan tuntutan kemajuan kehidupan dalam masyarakat. Kurikulum memang selalu berkembang dan menyesuaikan diri dengan kemajuan zaman. Untuk itu pengembangan kurikulum berupa proses yang dinamis dan integratif yang memang perlu diupayakan melalui langkah-langkah yang sistematis, profesional dan melibatkan seluruh aspek yang terkait dalam tercapainya tujuan pendidikan nasional. Namun jika kita melihat di lapangan perubahan kurikulum yang dirasa menjadi suatu siklus yang ekstrem malah menunjukkan banyak masalah karena perubahan kurikulum itu sendiri yang terlalu sering. Setiap pergantian kepemimpinan atau perubahan menteri pendidikan sendiri hampir bisa dipastikan akan terjadi perubahan kurikulum yang akhirnya membuat para aktor di bidang pendidikan mendapat kurikulum yang tidak konsisten.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan, rumusan masalah pada penelitian ini adalah:

1. Tujuan diubahnya kurikulum lama (2013) ke kurikulum baru (2018).
2. Alasan diubahnya kurikulum lama (2013) ke kurikulum baru (2018).
3. Dampak yang terjadi dengan adanya perubahan kurikulum.

## 1.3 Tujuan

Berdasarkan rumusan masalah di atas, maka tujuan dari penelitian ini adalah:

1. Mahasiswa mengetahui tentang tujuan perubahan kurikulum.
2. Mahasiswa mengetahui tentang alasan diubahnya kurikulum lama ke kurikulum baru.
3. Mahasiswa mengetahui tentang dampak perubahan kurikulum.

## 1.4 Batasan Masalah

Adapun batasan masalah yang didapat dari tujuan dan rumusan masalah di atas adalah:

1. Perangkat lunak yang dikembangkan akan berbasis web.
2. Tampilan pada web hanya dibuat dalam bentuk graf.
3. Perangkat lunak yang dikembangkan hanya berlaku untuk mahasiswa Teknologi Informatika.

## 1.5 Metodologi Penelitian

Dalam penyusunan skripsi ini mengikuti langkah-langkah metodologi penelitian sebagai berikut:

1. Melakukan studi pustaka untuk dijadikan referensi dalam pembangunan perangkat lunak pohon kurikulum.
2. Melakukan studi tentang penggunaan vis.js untuk visualisasi pohon kurikulum.
3. Melakukan studi tentang cara penggunaan DOT *Language*

## 1.6 Sistematika Penulisan

Keseluruhan bab yang disusun dalam penelitian ini terbagi kedalam bab-bab sebagai berikut:

1. Bab 1 Pendahuluan Bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.
2. Bab 2 Dasar Teori Bab ini membahas mengenai pengertian graf, apa itu DOT *Language*, dan visualisasi menggunakan viz.js.
3. Bab 3 Analisis

- 
- 1     4. Bab 4 Perancangan Bab ini akan membahas mengenai perancangan struktur pohon kurikulum  
2         untuk mahasiswa, di mana nanti di dalamnya akan berisi mata kuliah, syarat tempuh, dan  
3         syarat lulus.
  - 4     5. Bab 5 Implementasi dan Pengujian Bab ini akan membahas mengenai pengujian, implementasi  
5         kode program untuk membuat pohon kurikulum.
  - 6     6. Bab 6 Kesimpulan dan Saran Bab ini akan membahas mengenai kesimpulan dari penelitian  
7         yang telah dilakukan dan saran-saran untuk pengembangan lebih lanjut dari penelitian ini.



## BAB 2

### DASAR TEORI

Pada bab ini akan diuraikan teori-teori yang berhubungan dengan pembangunan pohon kurikulum. Teori-teori tersebut adalah teori tentang pengertian graph, DOT language, dan visualisasi pohon menggunakan *vis.js*.

## 2.1 Graph

### 2.1.1 Definisi Graph

Suatu graph didefinisikan oleh himpunan verteks dan himpunan sisi (edge). Verteks menyatakan entitas-entitas data dan sisi menyatakan keterhubungan antara verteks. Biasanya untuk suatu graph  $G$  digunakan notasi matematis.

$G=(V,E)$

$G = \text{Graph}$

$V = \text{Simpul atau vertex, atau node, atau titik}$

$E = \text{Sisi atau garis, atau Edge}$

$V$  adalah himpunan *verteks* dan  $E$  himpunan sisi yang terdefinisi antara pasangan-pasangan verteks. Sebuah sisi antara verteks  $x$  dan  $y$  ditulis  $x, y$ . Suatu graph  $H = (V1,E1)$  disebut subgraph dari graph  $G$  jika  $V1$  adalah himpunan bagian dari  $V$  dan  $E1$  himpunan bagian dari  $E$ .

### 2.1.2 Istilah dalam Graph

1. *Incident* Jika  $e$  merupakan busur dengan simpul-simpulnya adalah  $v$  dan  $w$  yang ditulis  $e=(v,w)$ , maka  $v$  dan  $w$  disebut "terletak" pada  $e$ , dan  $e$  disebut incident dengan  $v$  dan  $w$ .

2. *Degree* Di dalam Graph ada yang disebut dengan *Degree*, *Degree* mempunyai 3 jenis antara lain :

- Degree dari suatu verteks  $x$  dalam undigraph adalah jumlah busur yang incident dengan simpul tersebut.
- Indegree dari suatu verteks  $x$  dalam digraph adalah jumlah busur yang kepalanya incident dengan simpul tersebut, atau jumlah busur yang "masuk" atau menuju simpul tersebut.
- Outdegree dari suatu verteks  $x$  dalam digraph adalah jumlah busur yang ekornya incident dengan simpul tersebut, atau jumlah busur yang "keluar" atau berasal dari simpul tersebut.

3. *Adjacent* Pada graph tidak berarah, 2 buah simpul disebut adjacent bila ada busur yang menghubungkan kedua simpul tersebut. Simpul  $v$  dan  $w$  disebut adjacent. Pada graph berarah, simpul  $v$  disebut adjacent dengan simpul  $w$  bila ada busur dari  $w$  ke  $v$ .

4. *Successor dan Predecessor* Pada graph berarah, bila simpul  $v$  adjacent dengan simpul  $w$ , maka simpul  $v$  adalah *successor* simpul  $w$ , dan simpul  $w$  adalah *predecessor* dari simpul  $v$ .

1	<b>2.2 DOT Language</b>
2	2.2.1 Dasar Menggambar Grafik
3	2.2.2 Atribut Menggambar
4	Bentuk dan Label
5	Bentuk Grafik
6	Menggambar Ukuran dan Jarak
7	Penempatan Sisi dan Simpul
8	2.2.3 Node Ports
9	Pengelompokan
10	<b>2.3 Visualisasi Graph dengan Viz.js</b>

# LAMPIRAN A

## KODE PROGRAM

Listing A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Listing A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }
```





## LAMPIRAN B

### HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4