

BAB III

METODE PENELITIAN

3.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah pemanfaatan algoritma Jaro-Winkler *Distance* dalam menciptakan sebuah aplikasi yang dapat membandingkan dan menyeleksi soal yang akan diambil untuk dijadikan sebuah paket soal. Agar tidak terjadi kesamaan antar soal dalam sebuah paket soal.

3.2 Tempat dan Waktu Penelitian

Penelitian dilakukan dilaboraturium multimedia Jurusan Teknik Elektro Universitas Negeri Jakarta sejak bulan Mei 2014 hingga November 2014.

3.3 Definisi Kebutuhan(*Requirements Definition*)

Pemanfaatan algoritma Jaro-Winkler *Distance* dalam penelitian ini untuk membantu sebuah institusi pendidikan dalam hal ini yaitu sebuah web repositori soal untuk membandingkan dan menyeleksi soal yang akan diambil untuk dijadikan sebuah paket soal dengan melihat tingkat kemiripan soal yang akan dibandingkan. Adapun kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam mengembangkan aplikasi pembanding dan penyeleksi soal adalah sebagai berikut:

1. Processor Intel® Core™ i5-2430M.
2. Memory 4096MB RAM.
3. Operating system windows 7 ultimate 32-bit.
4. NetBeans IDE.
5. XAMPP

3.4 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah salah satu metode pengembangan perangkat lunak, yaitu metode pengembangan sistem Waterfall. Menurut Roger S. Pressman (1992: 39), model Waterfall adalah metode klasik yang bersifat sistematis, berurutan dalam membangun software. Dimana aplikasi akan diimplementasikan ke dalam *website* repositori soal, yang dibuat menggunakan sebuah algoritma Jaro-Winkler *Distance* dengan menggunakan bahasa pemrograman PHP. Dalam pembuatan aplikasi ini, uji coba dilakukan dengan menggunakan soal-soal yang sudah dimasukkan ke dalam *website* oleh *admin website* repositori soal. Seorang guru akan memilih beberapa soal yang akan dibangkitkan menjadi sebuah paket soal. Aplikasi akan membuat sebuah paket soal yang didalamnya tidak ada soal yang sama karena soal-soal tersebut telah melalui tahap perbandingan kemiripan dan penyeleksian soal. Adapun tahapan dalam pengembangan aplikasi adalah:



Gambar 3.1 Tahapan Pengembangan Aplikasi

3.4.1 Analisis Sistem (*System Analysis*)

Tahap analisis sistem dalam pengembangan aplikasi pembandingan dan penyeleksi soal pilihan ganda sebagai bagian dari aplikasi pembangkit soal adalah menentukan langkah-langkah rancangan aplikasi untuk mendapatkan nilai kemiripan antara dua soal yang berbeda untuk dibandingkan isi antara kedua soal tersebut dengan memanfaatkan algoritma Jaro-Winkler *Distance* dalam perhitungannya.

1. Menganalisa kebutuhan aplikasi.
2. Perancangan sistem aplikasi.
3. Melakukan *pre-processing* sebelum perhitungan algoritma Jaro-Winkler *Distance* seperti (*read file, casefolding, sorting*).
4. Mengimplentasikan algoritma Jaro-Winkler *Distance* untuk perhitungan soal yang akan dibandingkan dalam menentukan nilai kemiripan dua soal tersebut.

5. Mendesain *graphical user interface*, untuk mempermudah percobaan dan pemakaian aplikasi.
6. Melakukan uji coba aplikasi dalam menentukan kemiripan isi teks soal untuk mendapatkan presentase nilai kemiripan antara soal yang sudah ada di dalam *databases website*.
7. Mengevaluasi aplikasi pemanfaatan algoritma Jaro-Winkler *Distance* dalam menentukan tingkat kemiripan soal sebagai bagian dari aplikasi pembangkit paket soal pilihan ganda.

DFD digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail.

a. Diagram Konteks (level 0)

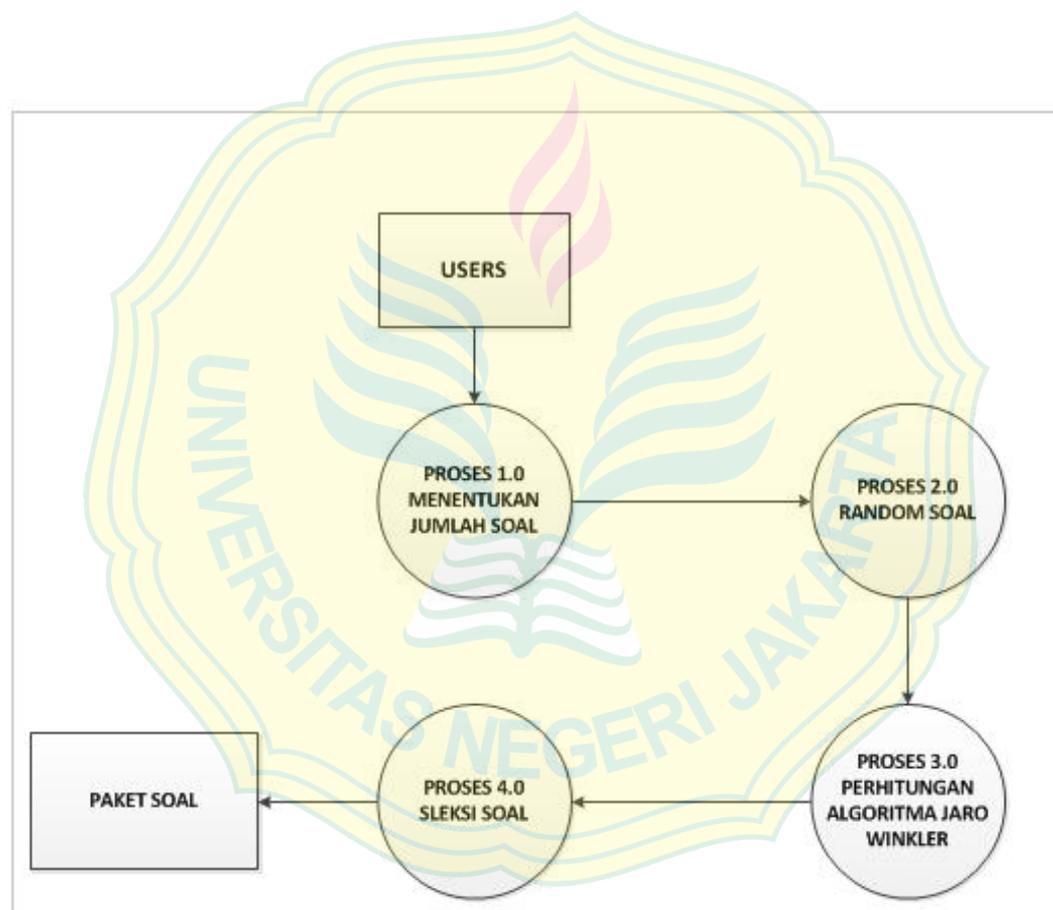
Diagram konteks menggambarkan sistem yang akan dibuat sebagai suatu *entitas* tunggal yang berinteraksi dengan orang maupun sistem lain. Diagram konteks dari sistem rancangan aplikasi ditunjukkan pada gambar 3.2.



Gambar 3.2 Diagram Konteks (Level 0)

Diagram diatas menunjukkan Sistem rancangan aplikasi. Pada diagram konteks, terdapat dua *entitas* yaitu *user* dan paket soal, serta terdapat 2 proses yaitu *random* soal dengan perhitungan algoritma Jaro-Winkler *Distance*, selain itu terdapat 2 arus data yang saling berhubungan yaitu penentuan jumlah soal yang akan dibangkitkan dengan paket soal yang ingin dibuat.

b. Diagram Zero (level 1)



Gambar 3.3 Diagram Level Zero

Keterangan :

a. Proses 1.0

Pada Proses 1.0 *Users* menentukan jumlah soal yang akan dibangkitkan dengan melihat indikator apa saja yang akan dipilih.

b. Proses 2.0

Pada Proses 2.0 Aplikasi me-*random* soal yang akan dipilih untuk dijadikan sebuah paket soal.

c. Proses 3.0

Proses 3.0 Setelah aplikasi me-*random* data yang akan diambil, setelah data yang dipilih dibandingkan tingkat kemiripan soal tersebut, menggunakan perhitungan algoritma jaro-winkler.

d. Proses 4.0

Proses 4.0 Setelah soal di-*random* dan dihitung dengan menggunakan algoritma *Jaro-winkler Distance* maka, akan terlihat tingkat kemiripan soal yang baru dipilih dengan soal yang sudah dipilih oleh aplikasi.

3.4.2. Perancangan Sistem

Perancangan sistem pada aplikasi ini adalah untuk merancang atau mendesain suatu sistem, yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur untuk mendukung operasi sistem. Adapun tujuan utama perancangan sistem ini adalah :

1. Untuk memenuhi kebutuhan para pemakai sistem.
2. Untuk memberi gambaran yang jelas dan rancang bangun yang lengkap kepada programmer.

Kedua tujuan ini berfokus pada perancangan atau desain sistem yang terinci yaitu pembuatan rancang bangun yang jelas dan lengkap yang nantinya digunakan untuk pembuatan program komputer.

Program sistem menggunakan algoritma Jaro-Winkler *Distance* dalam pengembangan aplikasi ini. Adapun input dan output dalam sistem aplikasi adalah:

1. *Input* sistem dalam aplikasi ini adalah *user* meng-*input* beberapa soal yang akan di bangkitkan dalam sebuah paket soal.
2. *Output* yang dihasilkan dalam sistem aplikasi ini berupa presentase kemiripan antar soal yang akan di bangkitkan menjadi sebuah paket soal yang utuh, serta mencegah agar soal yang dijadikan sebuah paket soal tidak ada yang sama.

3.4.3. Desain

Dalam aplikasi ini *user interface* dirancang untuk memudahkan user dalam penggunaan aplikasi. Berikut adalah rancangan *user interface* aplikasi:

Bayu Yudo Utomo

Generate Soal

Bank Soal

Log out

Generate Paket Soal

Mata Pelajaran

Sistem Operasi

Id	Indikator	Total Soal yang Tersedia	Banyaknya Soal yang Ingin Disertakan
42	Menjelaskan pengertian dan fungsi Sistem Operasi	0	0
43	Menjelaskan struktur sistem operasi	0	0
44	Menjelaskan macam-macam sistem operasi berbasis GUI	6	0
45	Menjelaskan macam-macam sistem operasi berbasis Text	6	0
46	Menjelaskan jenis-jenis metode instalasi sistem operasi	7	0
47	Menjelaskan Metode instalasi sistem operasi Clean Instal	0	0
48	Menjelaskan Metode instalasi sistem operasi Upgrade	0	0
49	Menjelaskan metode Multibooting	0	0
50	Menjelaskan metode Virtualisasi	0	0
51	Menjelaskan langkah-langkah instalasi sistem operasi berbasis GUI	7	0
52	Menjelaskan langkah-langkah instalasi driver	5	0
53	Menjelaskan langkah-langkah instalasi driver Motherboard (disk drive, DVD/CD ROM drives, Keyboard, Mouse,	5	0
56	Menjelaskan langkah-langkah instalasi software aplikasi,	3	0
57	Menjelaskan cara menginstalasi software Aplikasi	4	0
58	Menjelaskan cara menggunakan software aplikasi	7	0
59	Menjelaskan langkah-langkah instalasi software utility (partition magic, antivirus)	3	0
60	Menjelaskan langkah-langkah menginstalasi software utility (antivirus)	2	0
61	Menjelaskan langkah-langkah mengupdate antivirus	0	0
62	Menjelaskan tentang peripheral komputer	0	0
63	Menjelaskan cara menghubungkan peripheral dengan komputer	8	0
64	Menjelaskan fungsi driver pada peripheral	9	0
65	Menjelaskan cara menginstal driver	4	0
66	Menjelaskan kelengkapan peripheral	8	0
67	Menjelaskan cara menguji peripheral yang telah diinstall	9	1

Banyaknya soal yang diinginkan: 1

Threshold

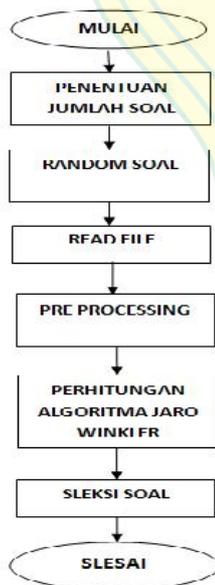
GENERATE

Gambar 3.4 User Interface 1

1. Generate soal : Berfungsi untuk membangkitkan paket soal yang akan dibuat.
2. Bank soal : Tampilan soal-soal yang ada di dalam *website*.
3. Log out : Keluar dari *website*.
4. Threshold : User menentukan nilai *threshold*.
5. Indiator : Indikator dari soal-soal yang sudah tersedia pada *databases*
6. Generate : Membangkitkan soal untuk menjadi sebuah paket soal.

3.4.4. Alur Kerja Sistem

Alur kerja sistem yang terjadi pada penelitian ini terdiri dari empat tahap, yaitu pemilihan berapa soal yang akan di bangkitkan, *random* soal yang akan diambil, *read file*, *pre-processing*, perhitungan algoritma Jaro-Winkler *Distance*, dan seleksi soal untuk dijadikan sebuah paket soal. Berikut gambar 3.5 alur sistem:



Gambar 3.5 Alur Sistem

Sistem pertama me-*random* soal yang akan dimasukan ke dalam sebuah paket soal. Tahapan selajutnya adalah *pre-proccesing* yang meliputi: *read file*, *case folding*, dan *sorting*. Kemudian dilakukan perhitungan *string* menggunakan algoritma Jaro-Winkler *Distance*, dan tahap yang terakhir adalah seleksi soal, jadi bila soal yang *diproses* memiliki presentase *thresholdnya* rendah maka akan masuk ke dalam sebuah paket soal. Jika presentase kemiripannya tinggi maka soal tersebut akan dibuang dan tidak diijinkan masuk ke dalam paket soal yang ingin dibuat.

3.4.4.1.Pre-processing

Pre-processing adalah proses untuk mendapatkan kata kunci yang digunakan untuk pencocokan kata. Proses *pre-processing* yang digunakan pada aplikasi yang dibuat meliputi : *read file*, *case folding*, *analizing*.

1) *Read file*

Proses *read file* pada tahap ini adalah membaca soal yang akan diambil untuk dibandingkan.

2) *Case folding*

Case folding adalah tahap pengubahan huruf kedalam dokumen menjadi huruf kecil ('a' sampai 'z'). Proses ini dilakukan untuk menyetarakan semua karakter dokumen menjadi huruf kecil. Karena penulisan karakter yang berbeda, seperti "Bayu" dengan "bayu" maka karakter B dengan b dianggap berbeda.

3) *Sorting*

Sorting tahap pengurutan kata secara *ascending* agar pencocokan isi teks dokumen dilakukan pada data yang terurut. Setelah tahap ini, maka dilakukan proses akhir, yaitu *processing*.

Berikut contoh kasus yang telah melalui proses *preprocessing*.

Tahap pertama *read file*:

1. **Smadav dibuat oleh?**
2. **Avast diciptakan oleh?**

Tahap pertama *case folding*:

1. **Smadav dibuat oleh?**
2. **Avast diciptakan oleh?**

Tahap pertama *sorting*:

1. **dibuatolehsmadav?**
2. **Avastdiciptakanoleh?**

3.4.5. Pengujian Aplikasi

Kebutuhan yang harus dipenuhi oleh aplikasi ini adalah dapat mendeteksi kesamaan *string* yang terdapat pada soal yang ingin dibangkitkan menjadi sebuah paket soal, agar didalam sebuah paket soal tidak terjadi kesamaan antar soal yang sudah dibangkitkan menjadi paket soal.

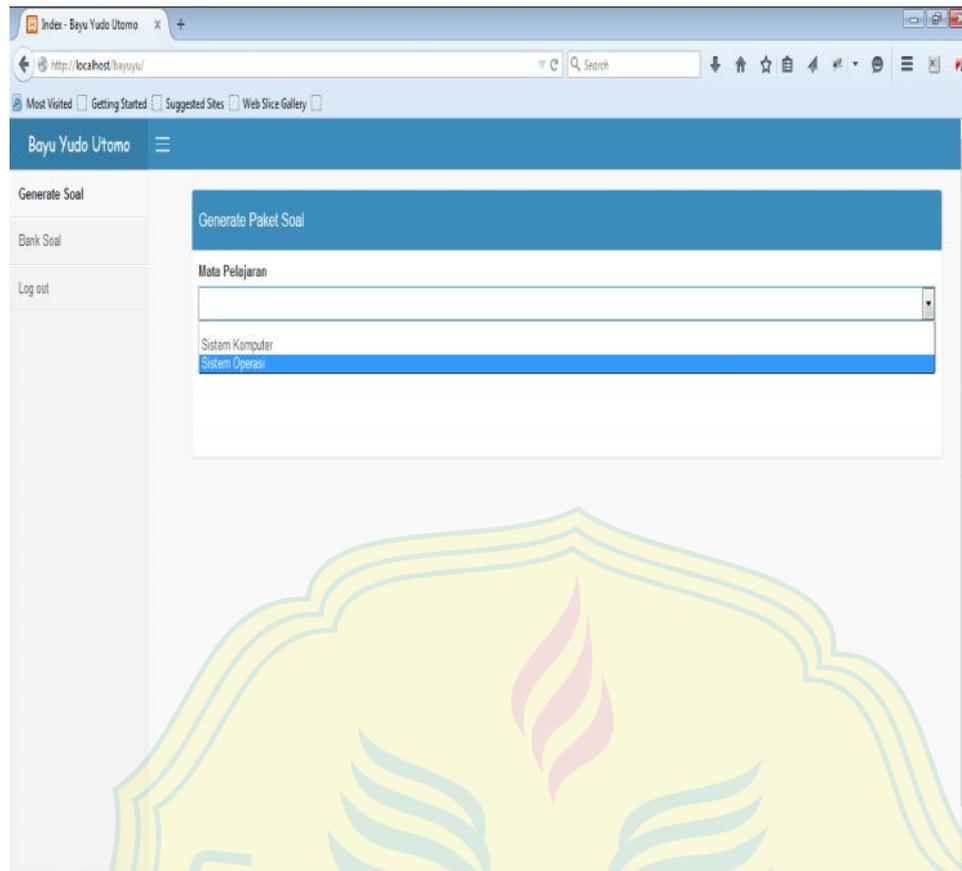
Pengujian aplikasi dilakukan untuk mengetahui seberapa bagus pemanfaatan algoritma Jaro-Winkler *Distance* dalam perbandingan *string* untuk dapat membandingkan dan menyeleksi soal yang akan dibangkitkan menjadi sebuah paket soal. Soal yang diuji adalah soal-soal untuk mata pelajaran Sistem Operasi.

Berikut adalah proses jalannya aplikasi serta *screen shoot* gambar proses jalannya aplikasi:



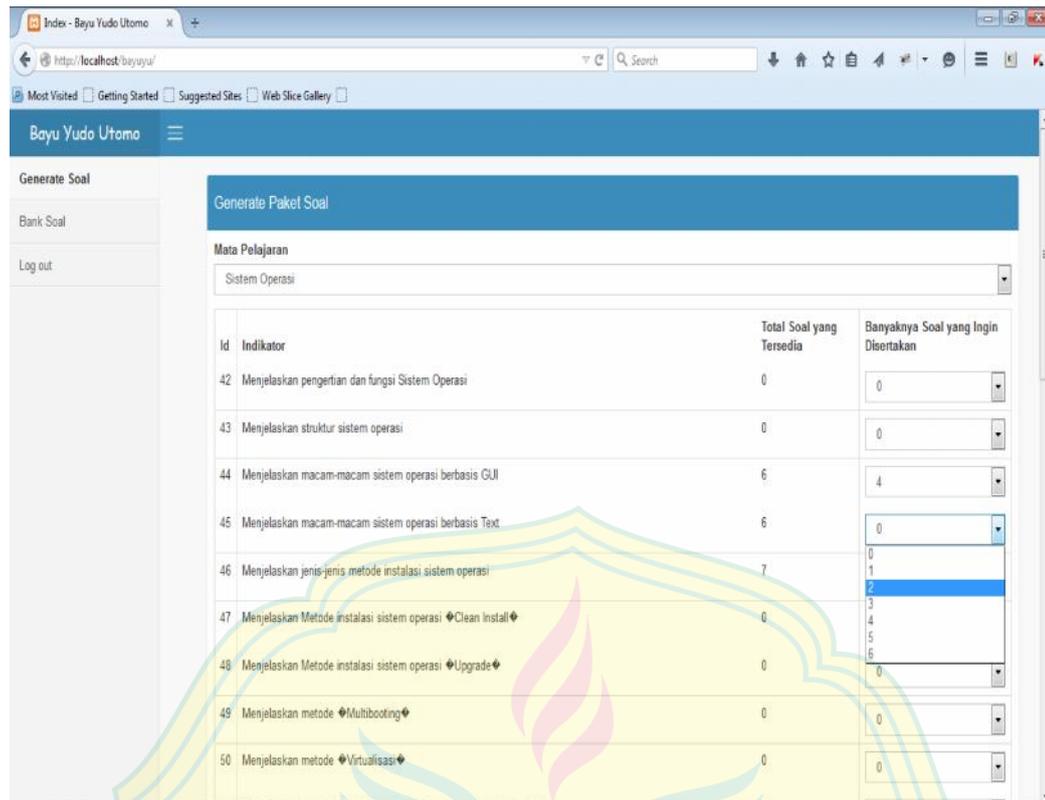
Gambar 3.6 Proses Jalannya Aplikasi





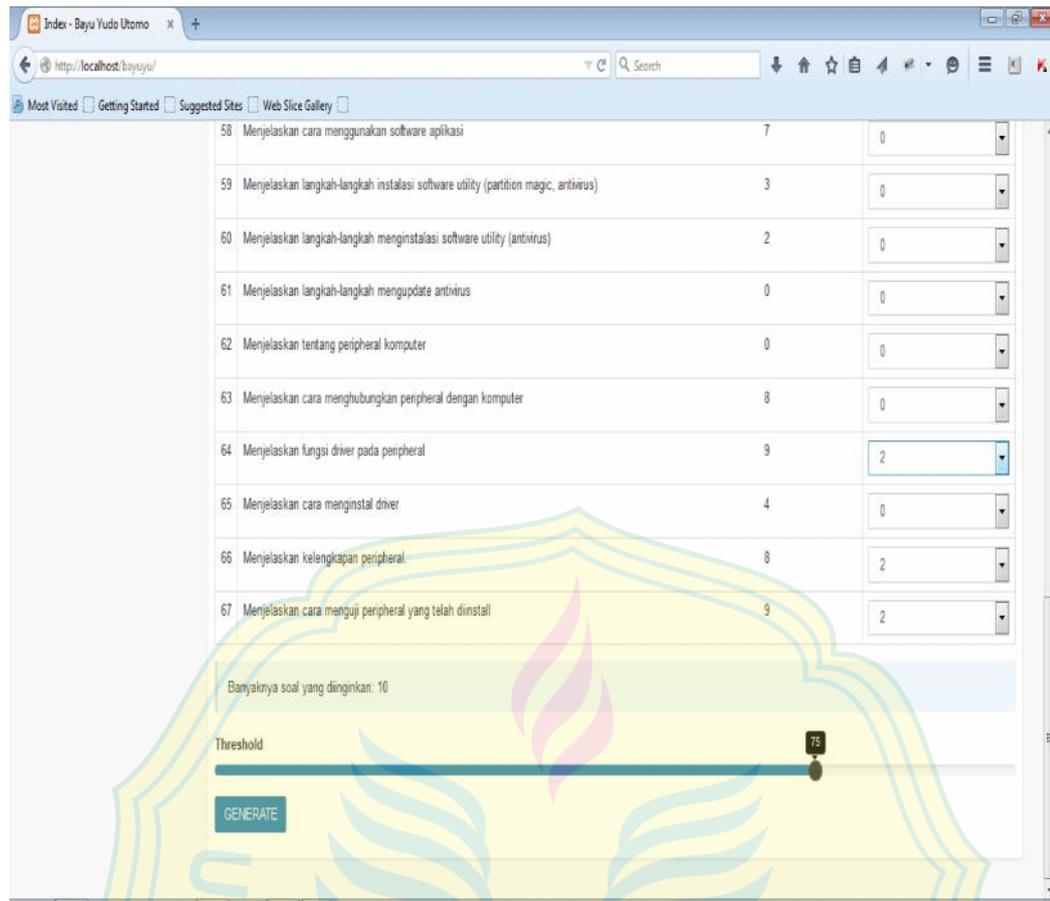
Gambar 3.7 User Interface 1

Pada gambar 3.7 *user* memilih mata pelajaran apa yang ingin dia pilih untuk membuat sebuah paket soal.



Gambar 3.8 User Interface 2

Pada gambar 3.8 aplikasi memberikan pilihan indikator serta total soal yang tersedia pada setiap indikator. *User* memilih berapa soal yang ingin dibangkitkan per indikator yang tersedia.



Gambar 3.9 User Interface 3

Pada gambar 3.9 *user* menentukan tingkat persentase kemiripan atau *Threshold* agar soal yang dibangkitkan oleh *user* tidak terjadi kesamaan antar soal yang dibangkitkan.

Soal master:

1. Smadav dibuat oleh?

Soal yang akan dibandingkan dengan soal master:

1. Smadav dibuat oleh?
2. Avira di buat oleh?
3. versi terbaru smadav adalah?
4. kelebihan aplikasi smadav yaitu?
5. Jenis jenis OS Android dibawah ini?

Masing-masing soal diuji sebanyak seratus kali dengan menggunakan tingkatan *threshold* yang berbeda-beda. Percobaan yang pertama soal diuji dengan menghilangkan fungsi *threshold*. Percobaan yang kedua soal diuji dengan menggunakan nilai *threshold* 60%. Percobaan yang ketiga soal diuji dengan menggunakan nilai *threshold* 75%. Percobaan yang keempat soal diuji dengan menggunakan nilai *threshold* 85%.

Cara kedua yaitu dengan menguji paket soal. Dalam pengujian paket soal dilakukan 2 percobaan. Percobaan yang pertama yaitu dengan membangkitkan 6 soal yang akan dijadikan sebuah paket soal. Soal-soal tersebut diambil dari dua indikator yang berbeda dengan membangkitkan 3 soal dari setiap indikator. Lalu pada percobaan kedua yaitu dengan membangkitkan soal lebih dari soal yang lolos *threshold* dalam sebuah

indikator. Contoh kali ini dengan membangkitkan 5 soal dari 6 soal yang tersedia pada indikator nomor 61 tentang menjelaskan langkah-langkah *mengupdate antivirus*. Soal dibangkitkan dengan *threshold* 75%. Sedangkan soal yang lolos dengan *threshold* 75% hanya 4 soal.

