

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

4.1.1. Hasil Pengujian

Pengujian dilakukan untuk mengetahui kinerja dari aplikasi yang sudah dibuat. Selain itu pengujian juga digunakan untuk melihat apa saja kelemahan yang terdapat didalam aplikasi. Aplikasi diuji dengan dua cara yang berbeda dimana cara yang pertama yaitu dengan perbandingan pasangan soal dan cara yang kedua yaitu dengan pengujian paket soal. Berikut hasil percobaan dua cara tersebut:

1. Perbandingan Pasangan Soal

Aplikasi diuji dengan cara mengambil sebuah soal yang akan dijadikan sebuah soal master dan akan diuji kemiripannya dengan kelima soal dengan tingkat kemiripan yang berbeda-beda sesuai dengan tahapan soal, yaitu dari soal sama sampai soal yang tidak sama. Uji coba dilakukan sebanyak seratus kali dimana:

A. Percobaan Pertama

Pada percobaan pertama aplikasi dijalankan tanpa menggunakan fungsi *threshold*. Berikut hasil tabel kemunculan soal dengan menghilangkan fungsi *threshold*:

Tabel 4.1 Hasil Percobaan Tanpa Threshold

| Nomor soal | Kemunculan | Persentase Kemunculan |
|------------|------------|-----------------------|
| 1 | 20 | 20% |
| 2 | 18 | 18% |
| 3 | 19 | 19% |
| 4 | 22 | 22% |
| 5 | 21 | 21% |

Bisa dilihat dari tabel 4.1 bahwa fungsi *random* yang dilakukan oleh aplikasi ini cukup baik karena persentase kemunculan soal yang dibandingkan dengan soal pengecoh cukup merata. Selain itu pada tabel 3.1 bisa di lihat jika aplikasi tidak menggunakan fungsi *threshold* maka didalam sebuah paket soal akan muncul sebuah soal yang sama.

B. Percobaan Kedua

Pada percobaan kedua uji coba dilakukan dengan menggunakan fungsi *threshold* 60 %. Berikut hasil tabel kemunculan soal dengan menggunakan fungsi *threshold* 60 %:

Tabel 4.2 Hasil Percobaan Dengan Menggunakan *Threshold* 60 %

| Nomor soal | Kemunculan | Persentase kemunculan |
|------------|------------|-----------------------|
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 45 | 45% |
| 5 | 55 | 55% |

Bisa dilihat dari tabel 4.2 bahwa dengan menggunakan fungsi *threshold* 60 % aplikasi kurang baik untuk menyeleksi soal yang akan dijadikan sebuah paket soal. Karena aplikasi sangat selektif dalam menyeleksi soal. Contohnya seperti soal pada nomor tiga seharusnya dapat ditampilkan karena pada soal nomor tiga tidak hanya berbeda strukturnya saja dengan soal master tetapi berbeda makna dari isi soal tersebut.

C. Percobaan Ketiga

Pada percobaan ketiga uji coba dilakukan dengan menggunakan fungsi *threshold* 75 %. Berikut hasil tabel kemunculan soal dengan menggunakan fungsi *threshold* 75 %:

Tabel 4.3 Hasil Percobaan Dengan Menggunakan *Threshold* 75 %

| Nomor soal | Kemunculan | Persentase kemunculan |
|------------|------------|-----------------------|
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 30 | 30% |
| 4 | 37 | 37% |
| 5 | 33 | 33% |

Dari hasil tabel 4.3 bisa di lihat aplikasi cukup efektif dalam menyeleksi sebuah paket soal. Karena aplikasi dapat menyeleksi soal nomor satu dan nomor dua dimana soal nomor satu adalah soal yang sama dengan soal master. Sedangkan soal nomor dua adalah soal yang hampir sama dengan soal nomor satu namun hanya berbeda satu katanya saja. Pada soal ini yang berbeda katanya yaitu *avira* dengan *smadav*.

D. Percobaan Keempat

Pada percobaan keempat uji coba dilakukan dengan menggunakan fungsi *threshold* 85 %. Berikut hasil tabel kemunculan soal dengan menggunakan fungsi *threshold* 85 %:

Tabel 4.4 Hasil Percobaan Dengan Menggunakan *Threshold* 85 %

| Nomor soal | Kemunculan | Persentase kemunculan |
|------------|------------|-----------------------|
| 1 | 0 | 0 |
| 2 | 26 | 26% |
| 3 | 23 | 23% |
| 4 | 27 | 27% |
| 5 | 24 | 24% |

Dari hasil tabel 4.4 bisa dilihat aplikasi kurang efektif dalam menyeleksi sebuah paket soal. Karena aplikasi tidak menyeleksi soal nomor dua dimana soal nomor dua merupakan sebuah soal yang hampir sama dengan soal nomor satu, namun hanya berbeda satu katanya saja yaitu *avira* dengan *smadav*.

2. Pengujian Paket Soal

aplikasi diuji dengan dua cara. Cara pertama 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.

Danyakanya soal yang diinginkan:

Threshold

GENERALE

Paket Soal

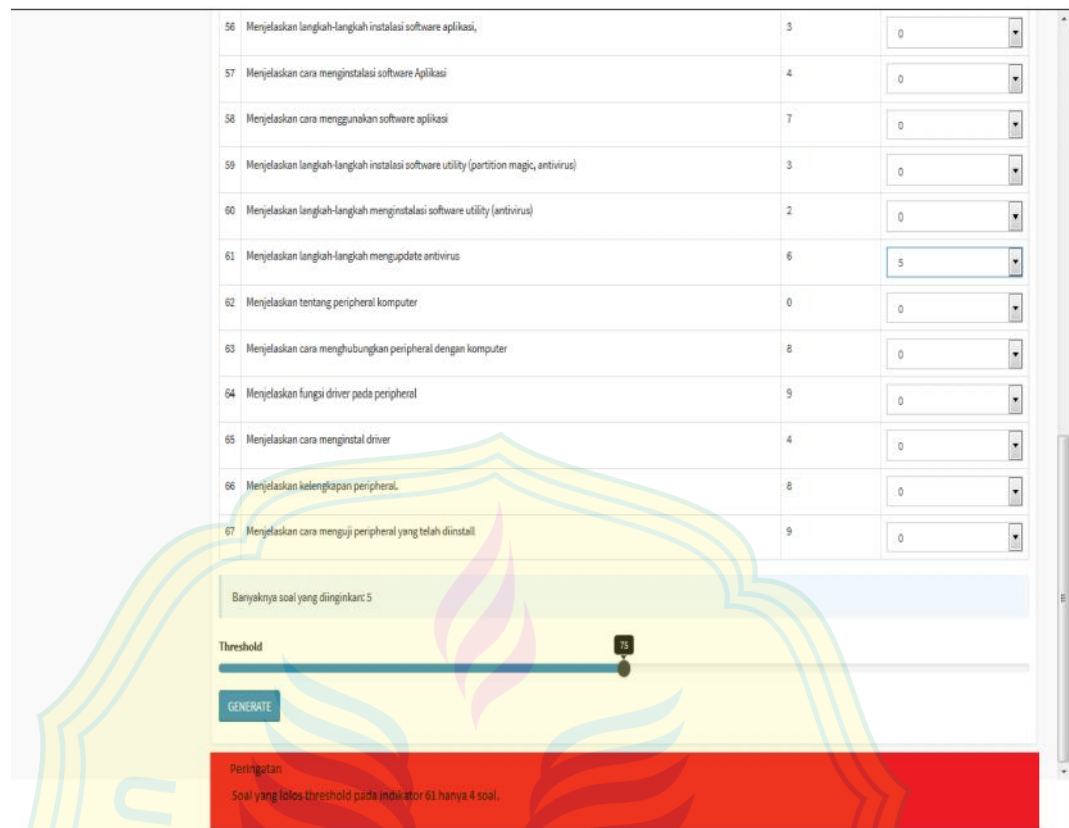
Threshold: 75

| No. | Indikator | Soal | A. | B. | C. | D. | E. |
|-----|-----------|---|------------|-----------|-----------|----------|------------|
| 1 | 65 | Windows Server 2003 mulai dikerjakan pada tahun | Akhir 2002 | Awal 2003 | awal 2001 | 2003 | Akhir 2000 |
| 2 | 65 | Berapa kecepatan akses data pada sebuah server yang memiliki standar kode IEEE 802.11g? | 216Mb/s | 54Mb/s | 432Mb/s | 11Mb/s | 108Mb/s |
| 3 | 65 | Nama domain host / server www.asus.com berlokasi di? | Singapura | Taiwan | Thailand | Swiss | China |
| 4 | 61 | Kelebihan aplikasi smadav yaitu? | tidak jitu | jitu | jitu | jitu | gdg |
| 5 | 61 | Jenis jenis OS Android dibawah ini? | asasasas | asafadfad | hdhfdg | ghghghgh | juuuu |
| 6 | 61 | Smadav dibuat oleh? | hfhfjfu | hfhfh | hnhh | bbbbb | hhhhj |

Buat PDF

Gambar 4.1 Proses Percobaan Paket Soal Pertama

Pada percobaan kedua aplikasi diuji 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. Berikut gambar hasil dari percobaan kedua:



| | | | |
|----|---|---|---|
| 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 | 6 | 5 |
| 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 | 0 |

Banyaknya soal yang diinginkan: 5

Threshold: 75

GENERATE

Peringatan:
Soal yang lolos threshold pada indikator 61 hanya 4 soal.

Gambar 4.2 Proses Percobaan Paket Soal Kedua

4.2. Pembahasan

Pada percobaan perbandingan pasangan soal bisa dilihat bahwa fungsi *random* pada aplikasi cukup berjalan dengan baik. Bisa di lihat dari tabel 4.1 bahwa persentase kemunculan dari soal nomor 1 sampai nomor lima hampir merata. Selain itu pada percobaan perbandingan pasangan soal dapat dilihat ketika fungsi *threshold* tidak diaktifkan maka soal yang sama dengan soal yang kemiripan soalnya tinggi masih sering keluar. Ini menunjukkan bahwa perlu adanya suatu fungsi yang dapat membatasi agar soal-soal yang sama dan soal yang kemiripannya tinggi tidak muncul di dalam satu paket soal. Lalu ketika fungsi *threshold* dinyalakan bisa dilihat bahwa soal yang sama dan soal yang

kemiripannya tinggi tidak muncul didalam sebuah paket soal. Dan pada percobaan pengujian paket soal. Aplikasi dapat berjalan dengan baik. Pada percobaan yang pertama bisa dilihat dari gambar 4.1 aplikasi dapat membuat sebuah paket soal yang di dalamnya tidak ada soal yang sama atau mirip. Pada percobaan yang kedua dapat dilihat dari gambar 4.2 bahwa aplikasi mempunyai suatu kelemahan yaitu aplikasi tidak dapat berfungsi ketika *user* ingin membangkitkan sebuah soal tetapi soal yang dapat lolos dari penyeleksian menggunakan *threshold* kurang dari soal yang diinginkan oleh *user*. Maka aplikasi membuat sebuah peringatan dengan menampilkan sebuah pesan bahwa pada indikator yang ingin dibangkitkan. Soal yang dapat lolos *threshold* kurang dari soal yang ingin dibangkitkan oleh *user*.

