

## Menghitung Ketepatan Jawaban Soal Ujian Essay dengan Penerapan Algoritma Boyer-Moore

Imam Teguh Agustianto<sup>1\*</sup>, Lis Utari<sup>2</sup>

Teknik Informatika

STIKOM Binaniaga

Email: [lis\\_utari@yahoo.co.id](mailto:lis_utari@yahoo.co.id)

---

### ABSTRACT

*Education plays an important role in preparing its students to become quality human resources. To measure a student's learning achievement, it is necessary to do a test or test to test each subject that has been done during the teaching and learning process. There are also various types of tests, such as written tests, oral tests, and practice tests. In written tests there are several types of questions, one of which is essay. Essay is a matter where students have to answer questions with sentences that are remembered or made by themselves, this makes the possibility of the emergence of answers with many variations that are very high, this makes the lecturer need a lot of time and energy. The need for a system that can be used by lecturers to determine the accuracy of answers to essay questions. Boyer-Moore algorithm is an algorithm used to search strings with a search pattern from right to left, the Boyer-Moore algorithm is considered to be the most efficient string matching algorithm in commonly used applications, such as a text editor application for the "search" and " substitute ". This study applies the Boyer-Moore algorithm to find the answer key that the lecturer thinks is correct in the student's answer by the way the lecturer input the answer key in its entirety then it will be broken down into words and the lecturer gives points to each word to be counted as a value. Students only need to input the correct answer and the Boyer-Moore algorithm will look for the answer key inputted by the lecturer at the student's answer. The output will be in the form of a percentage of accuracy per question and the value of all students who take the course. The results of the questionnaire that was distributed to lecturers and system experts showed very decent results for both respondents.*

**Keywords:** Boyer-Moore, test, essay.

### ABSTRAK

*Pendidikan memegang peranan penting dalam mempersiapkan mahasiswanya menjadi sumber daya manusia yang berkualitas. Untuk mengukur prestasi belajar seorang mahasiswa perlu dilakukan sebuah tes atau ujian untuk menguji setiap mata kuliah yang telah dilakukan pada saat proses belajar mengajar. Jenis tes pun ada beberapa macam, seperti tes tertulis, tes lisan, dan tes praktek. Pada tes tertulis pun terdapat beberapa jenis soal, salah satunya adalah soal berbentuk essay. Soal essay adalah soal dimana mahasiswa harus menjawab soal dengan kalimat yang diingat atau dibuat sendiri, hal tersebut membuat kemungkinan munculnya jawaban dengan variasi yang banyak sangat tinggi, hal ini membuat dosen membutuhkan waktu dan tenaga yang banyak. Diperlukannya sebuah sistem yang bisa digunakan oleh dosen untuk mengetahui ketepatan jawaban soal essay. Algoritma Boyer-Moore adalah algoritma yang digunakan untuk pencarian string dengan pola pencarian dari kanan ke kiri, Algoritma Boyer-Moore dianggap sebagai algoritma pencocokan string yang paling efisien dalam aplikasi-aplikasi yang biasa digunakan, seperti aplikasi text editor untuk fungsi "search" dan "substitute". Penelitian ini menerapkan algoritma Boyer-Moore untuk mencari kunci jawaban yang dosen anggap benar pada jawaban mahasiswa dengan cara dosen menginputkan kunci jawaban secara utuh lalu akan dipecah menjadi perkata dan dosen memberi poin pada setiap kata untuk dihitung sebagai nilai. Mahasiswa*

*hanya perlu menginputkan jawaban yang dianggap benar dan algoritma Boyer-Moore akan mencari kunci jawaban yang diinputkan oleh dosen pada jawaban mahasiswa. Output yang dihasilkan akan berupa persentase ketepatan per soal dan nilai seluruh mahasiswa yang mengikuti mata kuliah. Hasil dari kuesioner yang disebarakan kepada dosen dan ahli sistem menunjukkan hasil sangat layak pada kedua responden.*

**Kata Kunci:** *Boyer-Moore, ujian, soal essay*

---

---

## PENDAHULUAN

Sekolah Tinggi Ilmu Komputer Binaniaga Bogor adalah sebuah lembaga yang bergerak di bidang pendidikan khususnya dalam rumpun ilmu komputer. Bidang pendidikan memegang peranan penting dalam mempersiapkan pelajar untuk menjadi sumber daya manusia yang berkualitas. Tinggi atau rendahnya kualitas sumber daya manusia dalam bidang pendidikan dapat dilihat dari prestasi belajar. Untuk mengukur prestasi belajar seorang pelajar perlunya dilakukan sebuah tes untuk menguji setiap pelajaran yang telah dilakukan pada saat proses belajar mengajar, baik itu dalam bentuk tes tertulis ataupun tes praktek.

Adapun tes praktek dilakukan untuk menguji kebiasaan seorang pelajar dalam menerapkan semua yang telah dipelajari di dalam kelas ke kasus yang lebih real. Sedangkan tes tertulis dilakukan untuk mengasah daya ingat, daya pikir logis, kreatifitas, dan ketelitian.

Tes dalam bentuk tertulis memiliki banyak jenis soal, antara lain adalah soal pilihan ganda, soal mencocokkan, dan soal *essay*. Soal pilihan ganda adalah soal dimana pelajar harus menjawab soal dengan memilih beberapa jawaban yang sudah disediakan pada setiap butir soal. Soal mencocokkan adalah soal dimana pelajar harus menjawab soal dengan mencocokkan soal dengan jawaban yang paling tepat, jumlah jawaban pun terkadang dlebihihkan dari jumlah soal yang ada untuk mengecoh pelajar agar lebih teliti, dan soal *essay* adalah soal dimana pelajar harus menjawab soal dengan kalimat yang diingat atau dibuat sendiri.

Dengan kondisi dimana pelajar harus menjawab soal berbentuk *essay* dengan kalimat yang dibuat sendiri, maka kemungkinan munculnya variasi jawaban yang berbeda-beda sangat tinggi. Hal tersebut menyebabkan pemeriksaan soal sangat menghabiskan banyak waktu dan tenaga karena pengajar harus memeriksa soal satu persatu untuk mengecek ketepatan dari setiap jawaban.

Hal ini membuat penyusun berpikir untuk melakukan penelitian untuk membuat sistem pengecekan jawaban soal *essay* yang berbentuk kalimat agar tepat dengan jawaban yang pengajar inginkan dengan menggunakan Algoritma Boyer Moore pada platform website menggunakan framework PHP CodeIgniter.

## Rumusan Masalah

Proses pemeriksaan soal yang dilakukan saat ini di STIKOM Binaniaga Bogor belum memiliki sistem, sehingga dosen masih memeriksa benar atau salahnya jawaban mahasiswa satu persatu. Jika jumlah soal yang banyak ditambah jumlah mahasiswa yang banyak juga proses pemeriksaan soal ini akan memakan banyak waktu.

Dari indikator diatas, perlunya ada kebutuhan teknologi yang dapat membantu dosen untuk memeriksa seberapa tepat jawaban dari setiap soal essay tanpa menghilangkan hak bebas dosen untuk memberi poin pada setiap jawaban.

Tabel 1. Point Waktu

| Waktu       | Point |
|-------------|-------|
| T ≤ 1 Menit | 1     |
| T ≤ 2 Menit | 2     |
| T ≤ 3 Menit | 3     |
| T > 3 Menit | 4     |

Pada Tabel 1.1 penyusun mengubah data waktu pada Kuesioner Lampiran 1 berdasarkan bobot skor seperti pada skala Likert (Sugiyono, 2010:134).

Tabel 2. Rumusan Masalah

| No.                     | Bentuk Soal Essay | Dosen   | Waktu memeriksa soal perbutir |
|-------------------------|-------------------|---------|-------------------------------|
| 1.                      | Terbuka           | Dosen 1 | 3                             |
| 2.                      |                   | Dosen 2 | 4                             |
| 3.                      |                   | Dosen 7 | 4                             |
| 4.                      |                   | Dosen 8 | 2                             |
| Rata-rata soal terbuka  |                   |         | 3.25                          |
| 1.                      | Tertutup          | Dosen 2 | 3                             |
| 2.                      |                   | Dosen 3 | 1                             |
| 3.                      |                   | Dosen 4 | 4                             |
| 4.                      |                   | Dosen 5 | 1                             |
| 5.                      |                   | Dosen 6 | 4                             |
| 6.                      |                   | Dosen 8 | 1                             |
| 7.                      |                   | Dosen 9 | 3                             |
| Rata-rata soal tertutup |                   |         | 2.43                          |

Dapat dilihat dari Tabel 2 bahwa seorang dosen rata-rata memerlukan waktu memeriksa jawaban soal essay perbutir sekitar ≤ 3 menit untuk soal essay dengan bentuk terbuka, dan sekitar ≤ 2 menit untuk soal essay dengan bentuk tertutup. Adapun dosen yang mengeluarkan kedua bentuk soal essay diatas dengan waktu memeriksa jawaban soal perbutir yang berbeda. Waktu tersebut terbilang lama untuk satu butir soal.

Dari penjelasan diatas dapat dirumuskan permasalahan sebagai berikut:

1. Dosen membutuhkan ketelitian dan konsentrasi yang tinggi untuk memeriksa ketepatan dan akurasi jawaban soal berbentuk essay.
2. Dosen membutuhkan waktu yang lama dalam memeriksa jawaban soal berbentuk essay.

## **Maksud dan Tujuan Penelitian**

### **1. Maksud Penelitian**

Maksud penelitian ini untuk menerapkan Algoritma Boyer Moore untuk menghitung ketepatan dan akurasi dari jawaban soal essay.

### **2. Tujuan Penelitian**

- a. Mempermudah proses pemeriksaan ketepatan dan akurasi jawaban soal essay berdasarkan jawaban yang dosen anggap tepat.
- b. Dosen dapat mendapatkan nilai secara real time sesaat setelah ujian dilaksanakan.

## **Tinjauan Pustaka**

Penelitian yang menjadi acuan penulisan adalah penelitian yang dilakukan oleh Vina Sagita dan Maria Irimina Prasetiyowati (2013) dalam jurnal “Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, dan Tuned Boyer-Moore dalam Pencarian String” yang menjelaskan bahwa string searching merupakan suatu proses yang umum dilakukan dalam proses-proses yang dilakukan komputer karena teks merupakan bentuk utama penyimpanan data. Terdapat beberapa macam cara yang dapat dilakukan untuk mencari sebuah string pada kumpulan string lain yang lebih besar. Beberapa diantaranya adalah algoritma Boyer-Moore, Turbo Boyer-Moore dan Tuned Boyer-Moore. Guna mengetahui bagaimana performa algoritma-algoritma tersebut, terutama di bidang waktu yang diperlukan, maka dibuatlah aplikasi yang dapat digunakan untuk mengetahui waktu yang diperlukan untuk mencari suatu pattern dalam text. Aplikasi dibangun menggunakan metode prototyping dan menggunakan Microsoft Visual Studio dengan bahasa C# untuk pembangunannya. Aplikasi ini mendukung pencarian dengan penggunaan tiga algoritma (Boyer-Moore, Turbo Boyer-Moore, Tuned Boyer-Moore), pengubah kata (replace), highlight kata yang dicari, dan pemberian informasi waktu yang dibutuhkan masing-masing algoritma untuk pencarian serta algoritma mana yang membutuhkan waktu paling sedikit untuk pencarian. Dari penelitian yang dilakukan, dapat disimpulkan bahwa algoritma Boyer-Moore adalah algoritma yang paling cepat dalam pencarian string.

## **METODE**

### **Algoritma Boyer-Moore**

Menurut (Sjukani, 2014:1) Algoritma adalah langkah-langkah yang diambil dalam menyelesaikan suatu pekerjaan. Algoritma, pada dasarnya, adalah alur pikiran dalam menyelesaikan suatu pekerjaan, yang dituangkan dalam bentuk tertulis yang dapat dimengerti oleh orang lain.

Menurut (Juneadi, 2007:2) Dalam menyelesaikan masalah komputer apapun, diperlukan suatu algoritma untuk membantu mencari solusi permasalahan tersebut. Dalam bidang komputer, algoritma didefinisikan sebagai cara memecahkan masalah dalam waktu yang terbatas dan jumlah langkah yang terbatas, sehingga dalam pembuatan algoritma pasti ada titik awal sebagai dasar untuk memulai dan titik akhir yang menunjukkan akhir dari suatu algoritma.

Algoritma terbagi menjadi beberapa jenis, salah satunya adalah algoritma pencocokan string. Algoritma pencocokan string adalah komponen dasar yang digunakan dalam implementasi software-software yang berada di bawah kebanyakan sistem operasi. Salah satu contoh algoritma pencocokan string adalah Algoritma Boyer-Moore.

Menurut (Charras dkk., 2004:91) Algoritma Boyer-Moore dianggap sebagai algoritma pencocokan string yang paling efisien dalam aplikasi-aplikasi yang biasa digunakan. Versi yang sudah disederhanakan atau keseluruhan algoritma biasanya diimplementasikan dalam text editor untuk perintah “search” dan “subtitute”.

Algoritma Boyer-Moore memindai karakter-karakter dengan pola dari kanan ke kiri dimulai dari yang paling kanan. Dalam kasus ketidakcocokan (atau cocok dengan seluruh bagian pattern) akan digunakan dua fungsi precomputed untuk menggeser pola ke kanan. Dua fungsi pergeseran ini disebut good-suffix shift (disebut juga matching shift) dan bad-character shift (disebut juga the occurrence shift).

Adapun fitur-fitur utama yang dimiliki Algoritma Boyer-Moore adalah :

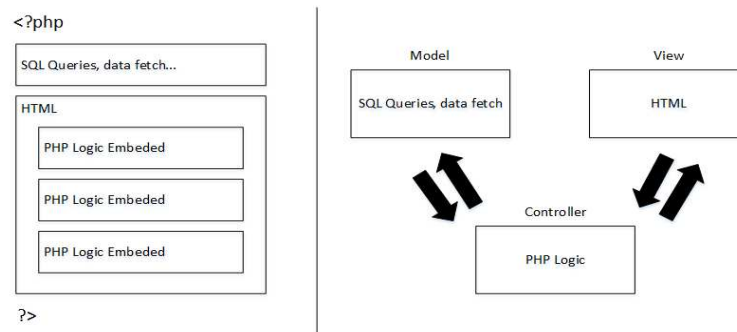
1. Melakukan perbandingan dari kanan ke kiri.
2. Fase pra proses dalam kompleksitas waktu dan ruang  $O(m + \sigma)$ .
3. Fase pencarian dalam kompleksitas waktu  $O(m \times n)$ .
4. Perbandingan karakter teks  $3n$  dalam kasus terburuk dalam pencarian untuk pola non periodic.
5. Performa terbaik dalam  $O(n / m)$ .

## **Codeigniter**

Menurut (Ardhana, 2013:1) PHP atau *PHP Hypertext Preprocessor* merupakan bahasa pemrograman yang berjalan disisi server. Ketika user melakukan akses ke sebuah alamat situs dengan mengetikkan alaman URLnya, browser akan mengirimkan request/permintaan ke webserver.

CodeIgniter adalah aplikasi open source yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP, CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. Perbedaan PHP native dengan metode MVC CodeIgniter dapat dilihat pada Gambar 1.

Menurut (Ardhana, 2013:19) MVC adalah konsep dasar yang harus dipelajari sebelum mengenal CodeIgniter. MVC merupakan sebuah pattern atau teknik pemrograman yang memisahkan antar pengembang aplikasi berdasarkan komponen utama pada sebuah aplikasi, seperti manipulasi data, user interface dan bagian yang menjadi kontrol aplikasi.



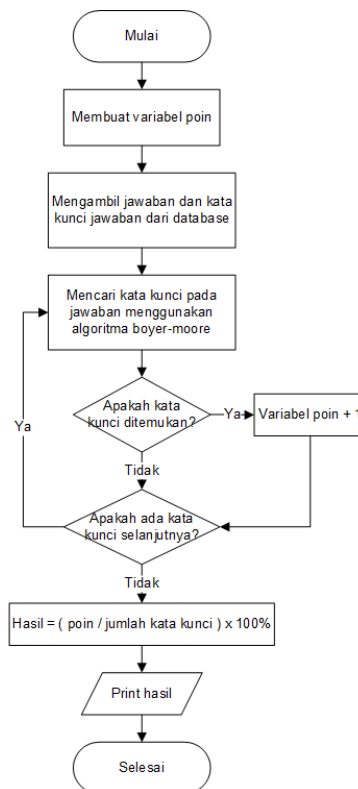
Gambar 1. Perbedaan PHP Native dan Codeigniter

## HASIL DAN PEMBAHASAN

### Hasil

#### 1. Penerapan Algoritma Boyer-Moore

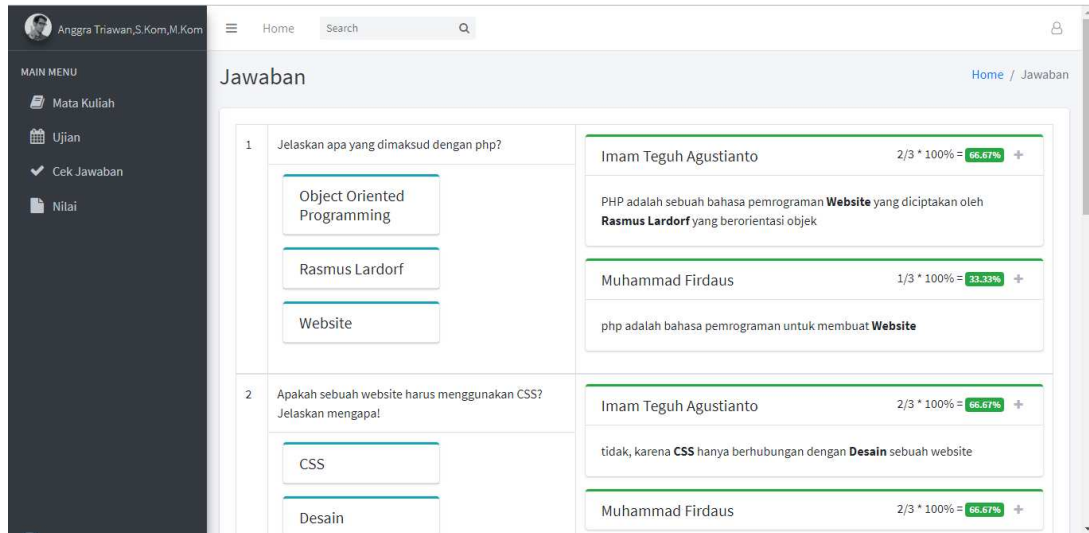
Algoritma Boyer-Moore pada penelitian ini diterapkan pada fitur Cek Jawaban pada user Dosen untuk melihat ketepatan jawaban per soal dari semua mahasiswa yang menjawab. Alur penerapan algoritma Boyer-Moore bisa dilihat pada flowchart pada Gambar 2.



Gambar 2. Flowchart Penerapan Algoritma Boyer-Moore

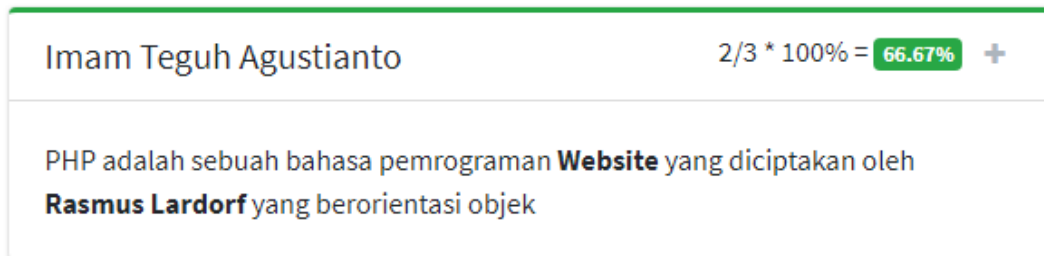
## 2. Penerapan dalam Aplikasi

Gambar 3 memuat inputan untuk soal dan kata kunci jawaban yang bisa ditambah sesuai dengan kebutuhan dosen dan juga tombol simpan yang menyimpan semua soal dan kata kunci jawaban bersamaan. Proses penginputan soal dilakukan per mata kuliah



Gambar 3. Tampilan Hasil

Tampilan hasil pada Gambar 3 merupakan hasil yang memuat jawaban mahasiswa per soal, dimana kata kunci yang terdapat pada jawaban yang diberikan mahasiswa dicari menggunakan Algoritma Boyer-Moore.



Gambar 4. Detail Jawaban

Gambar 4 merupakan detail dari jawaban mahasiswa dimana kata kunci yang terdapat pada jawaban dibuat menjadi huruf tebal, dan juga persentase ketepatan dari jawaban tersebut dimana dihitung dengan rumus

$$\text{Persentase Ketepatan} = \frac{\text{Jumlah kata kunci ditemukan}}{\text{Jumlah kata kunci keseluruhan}} \times 100\%$$

## Pembahasan

Aplikasi ini dilihat dan dicoba digunakan oleh pengguna dan ahli sistem untuk menilai apakah sistem ini sudah sesuai atau sudah layak digunakan untuk menghitung ketepatan jawaban soal essay di

STIKOM Binaniaga Bogor, adapun penilaian yang dilakukan adalah dengan cara penyebaran kuesioner kepada pengguna dan ahli sistem. Berikut penilaian kuesioner tersebut:

1. Uji Kelayakan Pengguna

Bobot skor yang digunakan untuk perhitungan pada kuesioner kelayakan sistem untuk pengguna ini menggunakan skala likert seperti yang sudah ditunjukkan pada Tabel 3.

Tabel 3. Rekapitulasi Kuisisioner Pengguna

| <b>Pernyataan</b>                                       | <b>R1</b>  | <b>R2</b>  | <b>R3</b>  | <b>R4</b>  | <b>R5</b>  |
|---------------------------------------------------------|------------|------------|------------|------------|------------|
| Kelengkapan informasi yang disajikan oleh sistem        | 5          | 5          | 5          | 4          | 4          |
| Ketepatan informasi yang disajikan oleh sistem          | 4          | 5          | 5          | 4          | 3          |
| Kehandalan informasi yang disajikan oleh sistem         | 4          | 4          | 5          | 4          | 4          |
| Bentuk keluaran yang disajikan oleh sistem              | 3          | 5          | 5          | 4          | 5          |
| Kemudahan dalam menggunakan sistem                      | 4          | 5          | 5          | 5          | 4          |
| Integrasi sistem dengan data yang sudah ada             | 4          | 4          | 5          | 5          | 3          |
| Kecepatan sistem untuk merespon                         | 4          | 5          | 5          | 5          | 4          |
| Pemulihan kesalahan                                     | 4          | 4          | 5          | 4          | 4          |
| Kemudahan dalam mengakses sistem                        | 4          | 4          | 5          | 5          | 4          |
| Bahasa yang digunakan di dalam sistem                   | 3          | 5          | 5          | 5          | 4          |
| Kesesuaian sistem dengan prosedur yang ada              | 4          | 5          | 5          | 4          | 3          |
| Terpenuhinya kebutuhan pengguna oleh sistem             | 4          | 4          | 5          | 4          | 4          |
| Kesigapan dalam melayani pengguna                       | 3          | 4          | 5          | 5          | 4          |
| Waktu penggunaan harian                                 | 4          | 3          | 5          | 5          | 3          |
| <b>Pernyataan</b>                                       | <b>R1</b>  | <b>R2</b>  | <b>R3</b>  | <b>R4</b>  | <b>R5</b>  |
| Frekuensi penggunaan                                    | 4          | 3          | 5          | 5          | 3          |
| Kemaksimalan kerja sistem                               | 4          | 4          | 5          | 4          | 4          |
| Pengunjungan ulang                                      | 4          | 3          | 5          | 5          | 3          |
| Kecepatan menyelesaikan tugas dengan menggunakan sistem | 4          | 4          | 5          | 5          | 4          |
| Kinerja sistem dalam menjalankan perintah               | 4          | 5          | 5          | 5          | 4          |
| Efektifitas pekerjaan dengan menggunakan sistem         | 4          | 5          | 5          | 4          | 4          |
| <b>Jumlah</b>                                           | <b>78</b>  | <b>86</b>  | <b>100</b> | <b>91</b>  | <b>75</b>  |
| <b>Nilai Tertinggi</b>                                  | <b>100</b> | <b>100</b> | <b>100</b> | <b>100</b> | <b>100</b> |

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = \frac{78 + 86 + 100 + 91 + 75}{100 + 100 + 100 + 100 + 100} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = 86\%$$

**Kesimpulan**



Persentase kelayakan yang didapat sebesar 86%, maka dapat dikategorikan ke dalam interpretasi yang sangat layak

## 2. Uji Kelayakan Ahli

Bobot skor yang digunakan untuk perhitungan pada kuesioner kelayakan sistem untuk ahli sistem ini menggunakan skala likert seperti yang sudah ditunjukkan pada Tabel 4.

Tabel 4. Rekapitulasi Kuisisioner Ahli

| Aspek Penilaian        | Indikator           | R1        | R2        |
|------------------------|---------------------|-----------|-----------|
| Komponen Input         | User interface      | 5         | 4         |
|                        | Interaksi Sistem    | 5         | 3         |
| Aspek Penilaian        | Indikator           | R1        | R2        |
| Komponen Model         | Prosedur Sistem     | 4         | 5         |
|                        | Logika Program      | 4         | 5         |
| Komponen Output        | Ketepatan Informasi | 4         | 5         |
|                        | Kekinian Informasi  | 5         | 4         |
| Komponen Teknologi     | Waktu Respon        | 4         | 4         |
|                        | Keluwesannya Sistem | 4         | 4         |
| Komponen Kontrol       | Keamanan Sistem     | 4         | 4         |
| Komponen Basis Data    | Konten Basis Data   | 4         | 3         |
| <b>Jumlah</b>          |                     | <b>43</b> | <b>41</b> |
| <b>Nilai Tertinggi</b> |                     | <b>50</b> | <b>50</b> |

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = \frac{43 + 41}{50 + 50} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = 84\%$$

### Kesimpulan

Persentase kelayakan yang didapat sebesar 84%, maka dapat dikategorikan ke dalam interpretasi sangat layak

## SIMPULAN DAN SARAN

### Simpulan

Berdasarkan hasil penelitian yang diuraikan pada BAB IV dapat ditarik kesimpulan :

1. Algoritma Boyer-Moore dapat digunakan untuk menghitung ketepatan jawaban soal essay sehingga dosen tidak memerlukan ketelitian yang terlalu tinggi untuk memeriksa jawaban soal essay.
2. Dosen dapat mendapatkan nilai secara real time sesaat setelah ujian dilaksanakan sehingga mahasiswa tidak perlu menunggu lama untuk mengetahui hasil ujiannya.

### Saran

Pencarian string yang dijalankan oleh algoritma Boyer-Moore adalah pencarian string yang identik, jadi jika kata yang dicari berbeda sedikit saja kata tidak dapat ditemukan, adapun dalam kasus

jawaban soal essay dimana jawaban diinputkan langsung oleh mahasiswa dan kesalahan menetik sering terjadi, maka dari itu penyusun menyarankan untuk menggunakan algoritma untuk mendeteksi kesalahan pengetikan, adapun dari sisi keakuratan diperlukan metode lain seperti metode intelegent bolt untuk mencari kesamaan tema antara jawaban dosen dan jawaban mahasiswa.

## DAFTAR RUJUKAN

- Ardhana, YM Kusuma. 2013. *Pemrograman PHP : CodeIgniter Black Box*. Jasakom. Jakarta.
- Argakusumah, Kencana Wulan. 2014. *IMPLEMENTASI ALGORITMA BOYER- MOORE PADA APLIKASI KAMUS KEDOKTERAN BERBASIS ANDROID*. Universitas Multimedia Nusantara.
- Arikunto, Suharsimi. 2015. *Manajemen Penelitian*. Edisi Revisi. Rineka Cipta. Jakarta.
- Charass, Christian. Lecroq, Thierry. 2004. *Handbook of Exact String Matching Algorithms* [PDF].  
Ada di: [https://www.researchgate.net/publication/220693416\\_Handbook\\_of\\_Exact\\_String\\_Matching\\_Algorithms](https://www.researchgate.net/publication/220693416_Handbook_of_Exact_String_Matching_Algorithms) [Diakses tgl 16 Maret 2018].
- CodeIgniter. [https://www.codeigniter.com/user\\_guide/index.html](https://www.codeigniter.com/user_guide/index.html) [Diakses tgl 26 Maret 2018].
- Januardi, Andri. 2013. *ANALISA PERBANDINGAN ALGORITMA BRUTE FORCE DAN BOYER MOORE DALAM PENCARIAN WORD SUGGESTION MENGGUNAKAN METODE PERBANDINGAN EKSPONENSIAL*. STMIK Budi Darma Medan.
- Junaedi, Fajar EP. 2007. *Algoritma dan Pemrograman*, Salemba Infotek. Jakarta.
- Kristanto, Steven G. Rachmat, Antonius C. Santosa, R Gunawan. 2013. *IMPLEMENTASI ALGORITMA BOYER-MOORE PADA PERMAINAN WORD SEARCH PUZZLE*. Universitas Kristen Duta Wacana.
- Lestari, Citra Puji. Hasibuan, Nelly Astuti. Ginting, Guidio Leonarde. 2016. *PERANCANGAN APLIKASI KAMUS ISTILAH MEDIS BERBASIS ANDROID DENGAN ALGORITMA BOYER-MOORE*. STMIK Budi Darma Medan.
- Pressman, Ph.D, R. S. 2012. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Rahmanita, Eza, S.T., M.T. 2014. *PENCARIAN STRING MENGGUNAKAN ALGORITMA BOYER MOORE PADA DOKUMEN*, Vol.1/No.1. Universitas Trunojoyo.
- Sagita, Vina. Prasetiyowati, Maria Irminda. 2013. *Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, Tuned Boyer-Moore dalam Pencarian String*. Vol.4/No.1. Universitas Multimedia Nusantara.
- Sjukani, Moh. 2014. *ALGORITMA: (Algoritma dan Struktur Data 1) dengan C, C++, dan Java – Teknik-Teknik Dasar Pemrograman Komputer*. Edisi 9. Penerbit Mitra Wacana Media. Jakarta.
- Sugiyono. 2014. *METODE PENELITIAN KUANTITATIF, KUALITATIF DAN R&D*. Alfabeta. Bandung.