# **BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM**

* 1. **Batasan Implementasi**

Dalam penerapan perangkat lunak, perlengkapan perangkat keras dan perangkat lunak yang memadai sangat diperlukan untuk menyokong terus-menerus selama proses pembuatan program. Komponen yang digunakan saat melakukan implementasi meliputi:

* + 1. **Komputer perangkat keras**

Adapun komputer yang digunakan memiliki detail teknis sebagai berikut :

Tabel 4. 1 Komputer perangkat keras

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Keras** | **Spesifikasi** |
| 1 | Processor | Intel(R) Core(TM) i3-1005G1 2 CPU @ 1.20GHz |
| 2 | Harddisk | SSSTC CL1-4D512 512GB |
| 3 | Ram | 12 GB DDR4 2666 MHz |
| 4 | VGA | NVIDIA GeForce MX330 2 GB |
| 5 | Monitor | 15.6 inch |
| 6 | Lan Card | Intel(R) Wireless-AC 9560 |

* + 1. **Perangkat lunak**

Untuk mengimplementasikan rancangan yang telah disusun, dibutuhkan sejumlah perangkat lunak untuk operasionalisasi program aplikasi, yang mencakup:

Tabel 4. 2 Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Lunak** | **Spesifikasi** |
| 1 | Sistem Operasi | Windows 11 Pro v22621.1778 |
| 2 | Bahasa Pemrograman | PHP v8.1.12 |
| 3 | Database | MariaDB v15.1 |
| 4 | Server | Apache v2.4.54 |
| 5 | Browser | Microsoft Edge v13.0.1774.57 |

* 1. **Implementasi Antarmuka**

Tahap implementasi sistem mengacu pada tahap di mana desain yang dibuat berdasarkan analisis ditransformasikan ke dalam bahasa pemrograman tertentu serta aplikasi perangkat lunak yang diciptakan diterapkan pada lingkungan yang sebenarnya. Sistem Perbandingan String Matching ini diterapkan dalam kerangka sistem Berbasis Web.

1. **Halaman Testing**

Berikut ini adalah contoh pengujian string matching dengan kata kunci Aditya Degnan menggunakan 250.000 data. Deskripsi seperti di bab 3.4

A screenshot of a computer

Description automatically generated with medium confidence

Gambar 4. 1 Halaman Testing Tabel 10.000-100.000 Data

Dalam gambar diatas memperlihatkanhasil implementasi dari perancangan sebelumnya, dalam pengujian tersebut digunakan Pattern “Aditya” dengan 250000 Data.

A screenshot of a graph

Description automatically generated with low confidence

Gambar 4. 2 Halaman Testing Diagram Garis Kecepatan Waktu dan Penggunaan Memory

* + 1. **Halaman Daftar Hasil Testing**

Halaman ini menampilkan daftar testing yang sudah disimpan pada testing sebelumnya. Untuk gambar nya bisa di lihat di bawah.

**A screenshot of a computer

Description automatically generated with medium confidence**

Gambar 4. 3 Halaman Daftar Hasil Testing

* + 1. **Halaman Detail Hasil Testing Sebelumnya**

Berikut ini adalah halaman data hasil pengujian string matching yang sudah tersimpan dengan kata kunci Aditya Degnan menggunakan 250.000 data.

A screenshot of a computer

Description automatically generated with medium confidence

Gambar 4. 4 Halaman Detail Testing Sebelmunya - Tabel 100.000-250.000 Data

Dalam gambar diatas memperlihatkan data hasil pengujian yang telah dilakukan sebelumnya.

A screenshot of a graph

Description automatically generated with low confidence

Gambar 4. 5 Halaman Testing Sebelumnya - Diagram Garis Kecepatan Waktu dan Penggunaan Memory

* 1. **Pengujian Sistem**

Langkah pengujian merupakan langkah yang ditempuh setelah penyelesaian tahap implementasi. Fungsi dari bagian ini adalah untuk memverifikasi kesesuaian antara sistem/aplikasi yang telah dikembangkan dengan niat perancangan awal. Pengujian sistem merangkum serangkaian uji coba yang dilakukan pada masing-masing komponen dalam sebuah sistem/aplikasi. Alasan diadakannya evaluasi terhadap suatu aplikasi adalah untuk menemukan dan memperbaiki kesalahan atau kekurangan yang berpotensi terjadi pada aplikasi tersebut. Metode pengujian yang diterapkan dalam studi ini melibatkan uji hitam alias *blackbox* dan pengujian beta.

1. **Pengujian Blackbox**

Uji *blackbox* merupakan teknik evaluasi yang menekankan pada kepuasan kriteria fungsi dari sebuah aplikasi. Pembuktian dilaksanakan dengan memasukkan sebuah rangkaian proses input dan kemudian mengevaluasi hasil output dari proses input tersebut.

Berikut adalah hasil dari pemanfaatan uji *blackbox* pada Sistem Perbandingan String Matching ini.

* + 1. Halaman Testing

Berdasarkan pengujian dengan metode *blackbox* yang dilakukan terhadap halaman testing, maka diperoleh hasil sebagai berikut.

Tabel 4. 3 Hasil pengujian blackbox pada halaman Testing

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Fungsi** | **Hasil Pengujian** |
| Pattern | Untuk menampilkan inputan pattern | Sesuai |
| Jumlah Data Testing | Untuk menampilkan dropdown jumlah input data testing | Sesuai |
| Tombol Testing | Untuk Memulai Testing kemudian menampilkan hasil | Sesuai |
| Tombol Simpan | Untuk menyimpan hasil testing yang sudah dilakukan sebelumnya | Sesuai |

* + 1. Halaman Daftar Testing

Berdasarkan pengujian dengan metode *blackbox* yang dilakukan terhadap halaman daftar testing, maka diperoleh hasil sebagai berikut.

Tabel 4. 4 Halaman Daftar Testing

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Fungsi** | **Hasil Pengujian** |
| Daftar Testing Sebelumnya | Untuk menampilkan daftar testing yang sudah dilakukan sebelumnya | Sesuai |
| Hapus Data Testing | Untuk menghapus data testing yang sudah di lakukan sebelumnya | Sesuai |
| Link Detail Data Testing | Untuk Berpindah halaman melihat detail data testing | Sesuai |

* + 1. Halaman Detail Testing Sebelumnya

Berdasarkan pengujian dengan metode *blackbox* yang dilakukan terhadap halaman daftar testing, maka diperoleh hasil sebagai berikut.

Tabel 4. 5 Halaman Detail Testing Sebelumnya

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Fungsi** | **Hasil Pengujian** |
| Detail Testing | Menampilkan Data Testing yang sudah tersimpan sebelumnya. | Sesuai |

1. **Pengujian Beta**

Dalam penelitian ini, pengujian beta dilakukan dengan cara membiarkan pengguna mencoba aplikasi, kemudian pengisian kuesioner oleh pengguna. Kuesioner ini bertujuan untuk mengumpulkan dan mengatur data yang telah diperoleh yang nantinya akan diolah untuk mendukung hasil penelitian dan memungkinkan penarikan kesimpulan.

Kuesioner ini berjudul Perbandingan Performansi Metode String Matching Menggunakan Metode Naive String Matching, Knuth Morris Pratt, Boyer-Moore, Rabin Karp dan SQL Query Like Untuk Pencarian Data Konsumen. Kuesioner ini akan disajikan kepada responden dengan 5 pertanyaan. Jawaban akan mengikuti skala Likert dari 1 hingga 5. Berikut adalah penjelasan rinci tentang skala Likert:

Tabel 4. 6 Skala likert dan interval

|  |  |  |
| --- | --- | --- |
| **Jawaban** | **Skor** | **Interval Penilaian** |
| Sangat Setuju | 5 | Index 80 % - 100 % |
| Setuju | 4 | Index 60 % - 79,99 % |
| Netral | 3 | Index 40 % - 59,99 % |
| Tidak Setuju | 2 | Index 20 % - 39,99 % |
| Sangat Tidak Setuju | 1 | Index 0 % - 19,99 % |

Untuk menghitung index persentase, digunakan rumus di bawah.

Untuk menghitung skor maksimum, digunakan rumus di bawah.

Skor Maksimum

= 20 x 5

= 100

Skor maksimum pada kasus ini adalah 100.

Untuk pertanyaan yang diajukan kepada para responder tertera pada tabel di bawah ini.

Tabel 4. 7 Daftar pertanyaan

|  |  |
| --- | --- |
| **No** | **Skor** |
| 1 | Apakah tampilan dari Sistem yang dibangun menarik? |
| 2 | Apakah tampilan Sistem yang dibangun sesuai dengan fungsinya? |
| 3 | Apakah Sistem yang dibangun mudah untuk digunakan? |
| 4 | Apakah Sistem yang dibangun berjalan lancar dan tidak ada error saat digunakan? |
| 5 | Apakah aplikasi Sistem yang dibangun sudah layak dipublikasikan dan siap digunakan? |

Kuesioner ini diberikan kepada 20 orang yang berasal dari berbagai macam bidang, mulai dari orang yang berhubungan dengan teknologi dan juga orang umum. Hasil perhitungan index persentase tiap jawaban bisa dilihat di bawah ini.

1. Apakah tampilan dari Sistem yang dibangun menarik?

Tabel 4. 8 Kuesioner pertanyaan 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kategori jawaban** | **Skor** | **Frekuensi Jawaban** | **Total Skor** | **Nilai Index Persentase** |
| Sangat Setuju | 5 | 6 | 30 | = 80 / 100 x 100 %  = 80 % |
| Setuju | 4 | 9 | 36 |
| Netral | 3 | 4 | 12 |
| Tidak Setuju | 2 | 1 | 2 |
| Sangat Tidak Setuju | 1 | 0 | 0 |
| Jumlah | | 20 | 80 |

Berdasarkan perhitungan pada tabel di atas, total skor yang didapat adalah 80 skor. Sedangkan hasil dari nilai persentase responden 80 % yang masuk pada interval penilaian sangat setuju. Maka dapat disimpulkan bahwa tampilan dari aplikasi sistem pendukung keputusan ini sudah menarik.

1. Apakah tampilan Sistem yang dibangun sesuai dengan fungsinya?

Tabel 4.9 Kuesioner pertanyaan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kategori jawaban** | **Skor** | **Frekuensi Jawaban** | **Total Skor** | **Nilai Index Persentase** |
| Sangat Setuju | 5 | 9 | 45 | = 89 / 100 x 100 %  = 89 % |
| Setuju | 4 | 11 | 44 |
| Netral | 3 | 0 | 0 |
| Tidak Setuju | 2 | 0 | 0 |
| Sangat Tidak Setuju | 1 | 0 | 0 |
| Jumlah | | 20 | 89 |

Berdasarkan perhitungan pada tabel di atas, total skor yang didapat adalah 89 skor. Sedangkan hasil dari nilai persentase responden 89 % yang masuk pada interval penilaian sangat setuju. Maka dapat disimpulkan bahwa tampilan dari aplikasi sistem pendukung keputusan ini sudah sesuai dengan fungsinya.

1. Apakah Sistem yang dibangun mudah untuk digunakan?

Tabel 4.10 Kuesioner pertanyaan 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kategori jawaban** | **Skor** | **Frekuensi Jawaban** | **Total Skor** | **Nilai Index Persentase** |
| Sangat Setuju | 5 | 9 | 45 | = 86 / 100 x 100 %  = 86 % |
| Setuju | 4 | 8 | 32 |
| Netral | 3 | 3 | 9 |
| Tidak Setuju | 2 | 0 | 0 |
| Sangat Tidak Setuju | 1 | 0 | 0 |
| Jumlah | | 20 | 86 |

Berdasarkan perhitungan pada tabel di atas, total skor yang didapat adalah 86 skor. Sedangkan hasil dari nilai persentase responden 86 % yang masuk pada interval penilaian sangat setuju. Maka dapat disimpulkan bahwa aplikasi sistem pendukung keputusan ini mudah digunakan.

1. Apakah Sistem yang dibangun berjalan lancar dan tidak ada error saat digunakan?

Tabel 4.11 Kuesioner pertanyaan 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kategori jawaban** | **Skor** | **Frekuensi Jawaban** | **Total Skor** | **Nilai Index Persentase** |
| Sangat Setuju | 5 | 6 | 30 | = 84 / 100 x 100 %  = 84 % |
| Setuju | 4 | 12 | 48 |
| Netral | 3 | 2 | 6 |
| Tidak Setuju | 2 | 0 | 0 |
| Sangat Tidak Setuju | 1 | 0 | 0 |
| Jumlah | | 20 | 84 |

Berdasarkan perhitungan pada tabel di atas, total skor yang didapat adalah 84 skor. Sedangkan hasil dari nilai persentase responden 84 % yang masuk pada interval penilaian sangat setuju. Maka dapat disimpulkan bahwa aplikasi sistem pendukung keputusan ini sudah berjalan lancar dan tidak ada error.

1. Apakah aplikasi Sistem yang dibangun sudah layak dipublikasikan dan siap digunakan?

Tabel 4.12 Kuesioner pertanyaan 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kategori jawaban | Skor | Frekuensi Jawaban | Total Skor | Nilai Index Persentase |
| Sangat Setuju | 5 | 6 | 30 | = 84 / 100 x 100 %  = 84 % |
| Setuju | 4 | 12 | 48 |
| Netral | 3 | 2 | 6 |
| Tidak Setuju | 2 | 0 | 0 |
| Sangat Tidak Setuju | 1 | 0 | 0 |
| Jumlah | | 20 | 84 |

Berdasarkan perhitungan pada tabel di atas, total skor yang didapat adalah 84 skor. Sedangkan hasil dari nilai persentase responden 84 % yang masuk pada interval penilaian sangat setuju. Maka dapat disimpulkan bahwa aplikasi sistem pendukung keputusan ini sudah layak dipublikasikan dan siap digunakan.

Hasil persentase rata-rata nilai index persentase dari setiap pertanyaan adalah seperti perhitungan di bawah.

Rata-rata index persentase = = 84,6 %

Rata-rata index persentase dari setiap pertanyaan adalah 84,6 % yang masuk pada interval penilaian sangat setuju. Maka dapat disimpulkan bahwa sistem Perbandingan Performansi Metode String Matching Menggunakan Metode Naive String Matching, Knuth Morris Pratt, Boyer-Moore, Rabin Karp dan SQL Query Like Untuk Pencarian Data Konsumen, akurat dan objektif berdasarkan sekenario pengujian yang sudah ditentukan.

1. **Hasil Penelitian**

Hasil penelitian ini diambil dari sekenario pengujian yang sudah di deskripsikan sebelumnya. Hasil dari sekenario pengujian sebagai berikut:

1. **Uji Coba Kasus Rata-rata (Average Case)**

Pada sekenario ini akan dilakukan pengujian dengan menggunakan kata kunci “ANDY” selama 5 kali dengan jumlah data 250.000, hasil dari pengujian sebagai berikut:

* **Kecepatan**: Pengujian ini menggunakan satuan milidetik sebagai berikut.

Tabel 4. 13 Hasil Pengujian dengan sekenario kasus rata-rata (Kecepatan)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pengujian** | **Kecepatan (ms)** | | | | |
| **SQL** | **NS** | **KMP** | **BM** | **RK** |
| 1 | 355 | 292 | 368 | 167 | 553 |
| 2 | 363 | 311 | 370 | 208 | 566 |
| 3 | 364 | 277 | 366 | 173 | 530 |
| 4 | 341 | 294 | 357 | 177 | 542 |
| 5 | 343 | 312 | 363 | 232 | 514 |
| Rata-Rata | 353.2 | 297.2 | 364.8 | 191.4 | 541 |

Dalam tabel diatas bisa diketahui hasil dari 5 kali pengujian algoritma Booyer More lebih cepat dibanding algoritma yang lain dengan selisih 161,8 dengan SQL Query Like atau 45,8% lebih cepat.

Gambar 4. 6 Hasil Pengujian dengan sekenario kasus rata-rata (Kecepatan)

Dalam gambar diagram garis diatas bisa diketahui bahwa algoritma Booyer Moore 64,6% lebih cepat disbanding algoritma Rabin Karp dengan selisih 349,6 Milidetik lebih cepat.

* **Memori**: Pengujian ini menggunakan satuan bytes sebagai berikut.

Tabel 4. 14 Hasil Pengujian dengan sekenario kasus rata-rata (Memori)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pengujian** | **Memory (bytes)** | | | | |
| **SQL** | **NS** | **KMP** | **BM** | **RK** |
| 1 | 382056 | 21664 | 21664 | 21664 | 21664 |
| 2 | 382056 | 21664 | 21664 | 21664 | 21664 |
| 3 | 382056 | 21664 | 21664 | 21664 | 21664 |
| 4 | 382056 | 21664 | 21664 | 21664 | 21664 |
| 5 | 382056 | 21664 | 21664 | 21664 | 21664 |
| Rata-Rata | 382056 | 21664 | 21664 | 21664 | 21664 |

Dalam tabel diatas bisa diketahui hasil dari 5 kali pengujian algoritma rata-rata memori yang digunakan SQL lebih banyak dibandingkan algoritma yang lain.

Gambar 4. 7 Hasil Pengujian dengan sekenario kasus rata-rata (Memori)

1. **Pengujian Dengan Karakteristik Teks Khusus**

Untuk pengujian dengan karakteristik teks khusus ini menggkunakan pattern/Kata kunci “123” dengan jumlah pengujian selama 5 kali dengan jumlah data 250.000, hasil dari pengujian sebagai berikut:

Gambar 4. 8 Pengujian Dengan Karakteristik Teks Khusus (Kecepatan)

Dari hasil pengujian dalam gambar diatas bisa diketahui bahwa algoritma Booyer Moore lebih baik dari segi kecepatan dengan selisih 430ms atau 74.6% lebih cepat dibandingkan algoritma Rabin karp.

Gambar 4. 9 Pengujian Dengan Karakteristik Teks Khusus (Memory)

Dalam gambar diatas bisa diketahui hasil dari pengujian algoritma memori yang digunakan SQL lebih banyak dibandingkan algoritma yang lain.