Jurnal Pendidikan dan Teknologi Indonesia (JPTI)
Vol. 4, No. 12, Desember 2024, Hal. 603-611
DOI: https://doi.org/10.52436/1.jpti.521
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Optimisasi Pagination dan Error Handling pada Portal Minat untuk Meningkatkan Performa Aplikasi Menggunakan Whitebox Testing

Ahmad Farhan[*1], Abdul Wahab[2], Faisal Ri'fai[3], Frederick Nehemia M[4], Hapifuddin Alman Faluti[5], Maya Oktarina[6]

[1,2,3,4,5,6]Universitas Indo Global Mandiri, Indonesia
Email: [1]ahmadfarhan.28af@gmail.com, [2]wahababdul513@gmail.com, [3]faisalrifai42@gmail.com, [4]frederick.mboeik@gmail.com, [5]hapifuddin.alman@gmail.com, [6]oktarinamaya66@gmail.com

Abstrak

Peningkatan jumlah pendaftar melalui Portal Minat memunculkan tantangan dalam pengelolaan data besar yang berdampak pada penurunan kinerja sistem. Penelitian ini bertujuan untuk mengimplementasikan pagination pada aplikasi Portal Minat guna mengoptimalkan kinerja sistem dan meningkatkan efisiensi waktu pemuatan data. Metode Whitebox Testing digunakan untuk mengevaluasi efektivitas pagination dan kemampuan sistem dalam menangani error handling saat memproses data besar. Hasil penelitian menunjukkan bahwa pagination berhasil mengurangi waktu pemuatan data dari 30 detik menjadi kurang dari 10 detik, serta mencegah kegagalan sistem saat menangani data besar. Penelitian ini merekomendasikan penerapan pagination pada seluruh fitur untuk meningkatkan stabilitas aplikasi.

Kata kunci: *error handling, filter data, pagination, performa sistem, whitebox testing*

Optimization of Pagination and Error Handling in Portal Minat to Improve Application Performance Through Whitebox Testing

Abstract

The increase in the number of registrants through the Interest Portal has created challenges in managing large-scale data, leading to a decline in system performance. This study aims to implement pagination in the Interest Portal application to optimize system performance and improve data loading efficiency. The Whitebox Testing method was employed to evaluate the effectiveness of pagination and the system's ability to handle error occurrences when processing large-scale data. The results show that pagination successfully reduced data loading time from 30 seconds to less than 10 seconds and prevented system failures when handling large datasets. This study recommends the implementation of pagination across all features to enhance application stability.

Keywords: *data filter, error handling failure, pagination, system performance, whitebox testing*

1. PENDAHULUAN

Dalam era digital, institusi pendidikan mulai mengadopsi sistem pendaftaran online untuk mempermudah manajemen data siswa. Aplikasi Portal Minat, yang awalnya memuat seluruh data pendaftar langsung dari database tanpa pagination, mengalami perlambatan signifikan ketika jumlah data meningkat. Pagination membantu membagi data besar menjadi bagian lebih kecil, Sehingga dapat mengurangi waktu pemuatan dan dapat meningkatkan pengalaman pengguna (*User Experience*) [1].

Penggunaan pagination terbukti efektif dalam mengurangi waktu komputasi dan beban CPU, sebagaimana ditunjukkan oleh Kartika dan Rinartha pada Sistem Informasi Manajemen Nursery Anggrek [2]. Pagination dan optimasi query data mampu menurunkan penggunaan memory footprint dan CPU secara signifikan, meningkatkan efisiensi pengelolaan data besar [2]. Seperti ditunjukkan oleh penelitian Wahyuni, sistem informasi berbasis web dengan fitur seperti pagination dan filtering dapat meningkatkan efisiensi pengelolaan data secara signifikan, memastikan transparansi dan stabilitas sistem [3]. Pada penelitian sebelumnya berfokus pada pengoptimalan kinerja API server menggunakan pagination dan caching, termasuk mitigasi kerentanan keamanan dan peningkatan kecepatan melalui scroll-induced pagination [4]. Namun, penelitian ini menekankan pada penerapan pagination untuk meningkatkan stabilitas sistem dalam pengelolaan data besar pada aplikasi pendidikan.

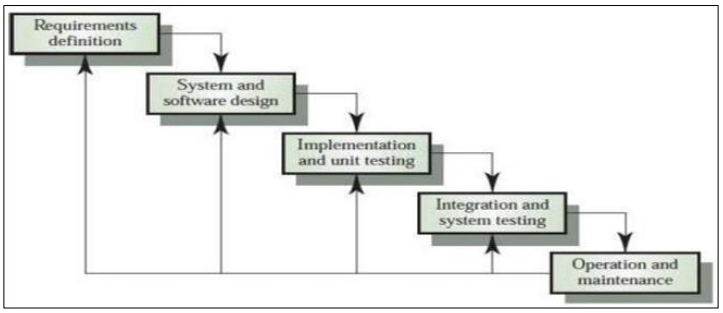
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Penelitian ini bertujuan untuk menerapkan pagination pada Portal Minat guna mengurangi beban komputasi di halaman home dan meningkatkan kinerja sistem. Pagination membatasi pemuatan hingga 100 entri data terbaru, mengurangi waktu pemrosesan, dan meringankan beban server. Selain itu, Whitebox Testing digunakan untuk menganalisis efektivitas pagination dan mengukur kemampuan sistem dalam menangani Error Handling Failure pada data besar. White box Testing adalah metode pengujian yang memeriksa dan menganalisis kode program untuk menemukan kesalahan atau kekurangan dalam suatu aplikasi atau perangkat lunak [5]. Metode ini melibatkan pemeriksaan modul secara detail untuk memastikan keberfungsiannya [6]. Whitebox Testing memastikan setiap jalur eksekusi fitur pagination dan filter diuji secara menyeluruh [7].

Penelitian ini diharapkan dapat memberikan solusi konkret untuk optimisasi kinerja Portal Minat serta meningkatkan stabilitas sistem, khususnya dalam memuat data besar dan fitur filter tahun ajaran yang belum mendukung pagination.

2. METODE PENELITIAN

Penelitian ini menggunakan metode Waterfall, merupakan model proses dasar yang memiliki aktivitas terdiri dari spesifikasi (specification), pengembangan (development), validasi (validation), dan evolusi (evolution) [8]. Sebagai model proses linier, Waterfall dipilih karena tahap-tahapnya terstruktur dengan jelas [9]. Meskipun perbaikan hanya dapat dilakukan setelah sistem selesai, metode ini ideal untuk proyek dengan kebutuhan yang telah terdefinisi secara jelas [10].

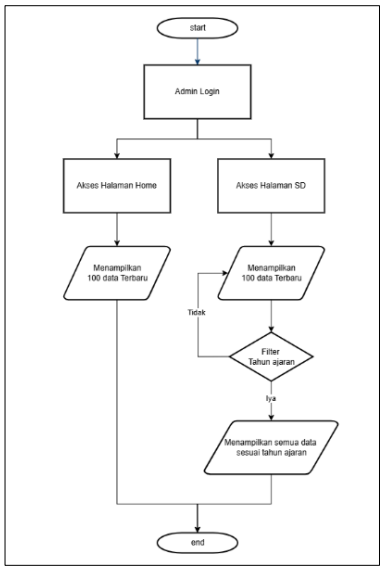


Gambar 1. Metode Waterfall

1. Tahap Analisis Kebutuhan

Pada tahap ini, Identifikasi menunjukkan bahwa tanpa pagination, aplikasi Portal Minat memuat seluruh data pendaftar saat halaman Home dibuka, meningkatkan waktu komputasi dan menyebabkan Error Handling Failure pada data besar. Pengujian dilakukan menggunakan Laptop Intel Core i5, RAM 16GB, OS Windows 11, dan Visual Studio Code.

2. Tahap Perancangan Sistem



Gambar 2. Flowchart proses pagination dan filter tahun ajar

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Pagination pada aplikasi dirancang untuk membatasi jumlah data yang dimuat pada halaman Home menjadi 100 entri terbaru. Selain itu, filter tahun ajaran yang digunakan untuk menampilkan data berdasarkan tahun pendaftaran belum dilengkapi dengan pagination, sehingga berpotensi menyebabkan perlambatan saat memproses data dalam jumlah besar.

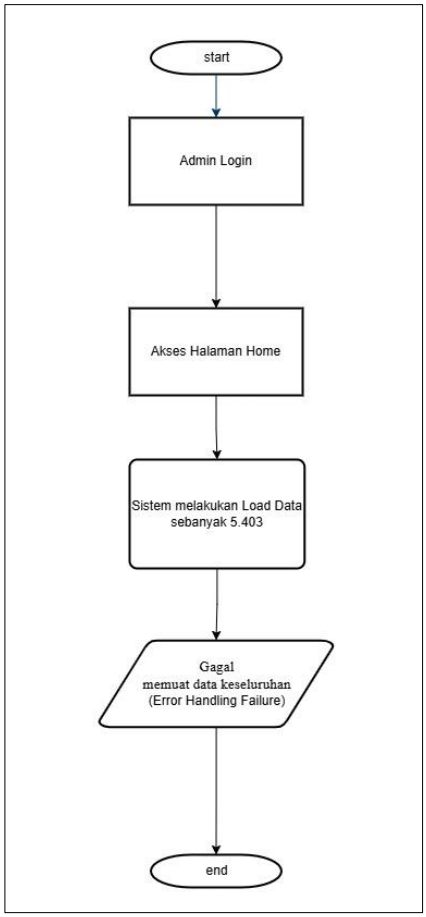
Flowchart ini menggambarkan alur proses login admin, akses halaman HOME, pengecekan filter tahun ajaran, dan implementasi pagination di Portal Minat. Pagination memungkinkan data ditampilkan bertahap, sedangkan tanpa pagination, sistem berisiko mengalami Error Handling Failure saat memuat data besar.

3. Tahap Implementasi

Pagination diterapkan pada halaman utama Portal Minat dengan batas tampilan data 100 entri per halaman. Selain itu, pengujian dilakukan untuk mengidentifikasi perbaikan waktu komputasi dengan pagination dibandingkan tanpa pagination. Pengujian ini bertujuan untuk memastikan efektivitas pagination dalam mengurangi waktu pemuatan data dan menangani Error Handling Failure saat menangani data besar.

4. Tahap Pengujian

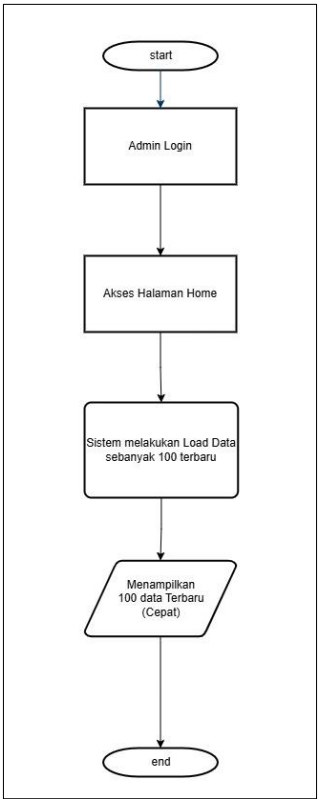
Pengujian dilakukan dengan metode Whitebox Testing untuk mengamati struktur dan alur proses pagination di Portal Minat serta memantau kemampuan sistem dalam menangani Error Handling Failure pada data besar. Pengujian ini bertujuan untuk mengukur efektivitas pagination dalam meningkatkan performa, mengurangi waktu pemuatan data, dan meminimalkan Error Handling Failure saat halaman Home diakses admin. Whitebox Testing dipilih karena kemampuannya mengidentifikasi galat pada kode dengan menghapus baris yang tidak diperlukan, serta memberikan cakupan pengujian maksimal untuk memastikan kualitas aplikasi [11]. Penerapan pagination membantu membatasi data per halaman, yang secara signifikan mengurangi beban server dan mempercepat waktu respon aplikasi. Pengujian dilakukan pada tiga skenario: tanpa pagination, dengan pagination, dan filter tahun ajaran tanpa pagination.



Gambar 3. Flowchart pengujian tanpa *pagination*

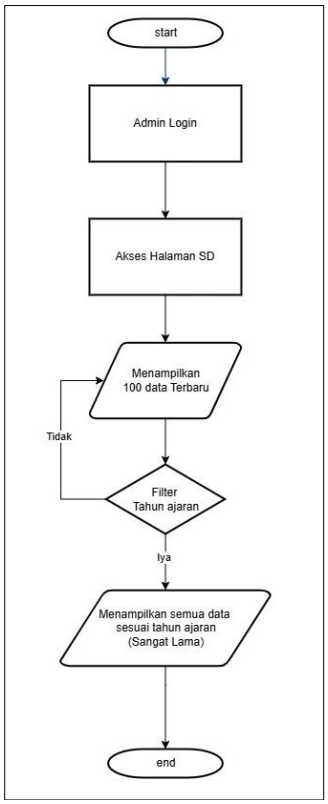
Flowchart ini menggambarkan alur pemrosesan data pada Portal Minat sebelum implementasi pagination, di mana seluruh data dimuat sekaligus, menyebabkan perlambatan kinerja.

p-ISSN: 2775-4227
e-ISSN: 2775-4219



Gambar 4. Flowchart pengujian dengan *pagination*

Flowchart ini menunjukkan pengoptimalan alur sistem setelah pagination diterapkan, membatasi data yang dimuat dalam jumlah lebih kecil untuk meningkatkan efisiensi.



Gambar 5. Flowchart pengujian filter tahun tanpa pagination

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Flowchart ini menggambarkan proses penerapan filter tahun pada Portal Minat sebelum pagination diterapkan. Seluruh data yang sesuai dengan kriteria filter dimuat sekaligus, yang menyebabkan perlambatan kinerja ketika data yang harus diproses berjumlah besar.

Tabel 1. Tabel Pengujian Pagination

No	Skenario Pengujian	Jumlah Data	Waktu Komputasi	Observasi
1	Tanpa pagination, halaman Home	5.403 (Semua Data)	*Error Handling Failure*	Gagal memuat data keseluruhan
2	Dengan pagination, halaman Home	100	< 10 detik	Data ditampilkan lebih cepat
3	Filter tahun ajaran tanpa pagination	1000	*Error Handling Failure*	Data Tampil sangat lama

Tanpa pagination, sistem mengalami Error Handling Failure saat memuat 5.403 data di halaman Home. Dengan pagination, waktu komputasi untuk 100 entri berkurang menjadi kurang dari 10 detik. Fitur filter tahun ajaran tanpa pagination juga gagal pada 1000 data, menegaskan pentingnya pagination untuk menjaga stabilitas aplikasi.

5. Tahap Pemeliharaan dan Evaluasi

Tahap pemeliharaan dan evaluasi berfokus pada peningkatan kinerja aplikasi Portal Minat, terutama dengan mengoptimalkan fitur pagination dan mengatasi kendala pada filter tahun ajaran yang masih memuat seluruh data. Pengembangan lebih lanjut diperlukan untuk memastikan aplikasi dapat menangani data besar secara efisien tanpa mengorbankan kecepatan dan stabilitas.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Penelitian

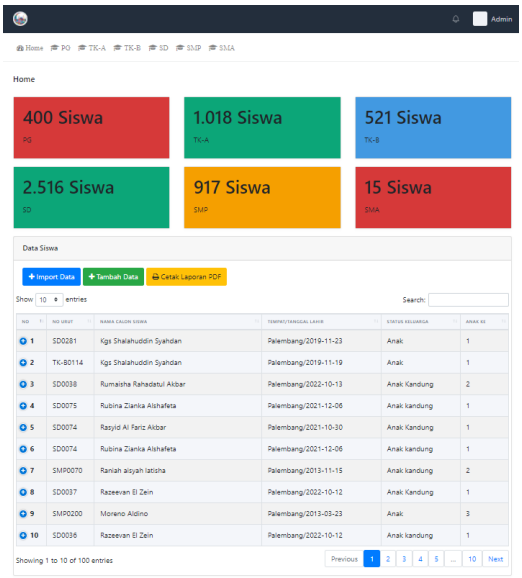
Tabel 2. Hasil Pengujian Pagination Halaman Home.

No	Kode yang Diuji	Deskripsi Fungsi	Hasil Pengujian	Kesimpulan
1	$data['title'] = 'Home'; $data['user'] = $this->db->get_where('user', ['email' => $this->session->userdata('email')])->row_array();	Mengatur judul halaman dan mengambil data pengguna dari sesi aktif untuk ditampilkan di halaman Home.	Judul halaman dan data pengguna ditampilkan dengan benar.	Berhasil, fungsi sesuai harapan.
2	$this->load->library('pagination');	Memuat library pagination bawaan CodeIgniter untuk digunakan dalam membatasi tampilan data.	Library pagination berhasil dimuat tanpa error.	Berhasil, pagination siap digunakan.
3	$config['base_url'] = base_url('index.php/admin/index');	Menentukan URL dasar untuk pagination di halaman HOME agar dapat diakses melalui segmen URL yang benar.	URL pagination sesuai, navigasi dapat diakses.	Berhasil, URL pagination diatur dengan benar.
4	$config['total_rows'] = $this->db->count_all('siswa');	Menghitung jumlah total data siswa dari database untuk menentukan batasan pagination.	Jumlah data dihitung sesuai total entri dalam tabel siswa.	Berhasil, data total sesuai harapan.
5	$config['per_page'] = 100; $config['uri_segment'] = 3;	Mengatur batas tampilan data per halaman menjadi 100 entri, dan menentukan segmen URI yang digunakan untuk pagination.	Pagination membatasi 100 entri per halaman.	Berhasil, konfigurasi per halaman sesuai.
6	$this->pagination->initialize($config);	Menginisialisasi pagination dengan konfigurasi yang sudah diatur agar bisa	Pagination terinisialisasi tanpa error.	Berhasil, pagination siap untuk digunakan.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

No	Kode yang Diuji	Deskripsi Fungsi	Hasil Pengujian	Kesimpulan
		digunakan di halaman HOME.		
7	$page = ($this->uri->segment(3)) ? $this->uri->segment(3) : 0; $this->db->order_by('verifikasi_date', 'DESC'); $this->db->order_by('no_urut', 'ASC'); $data['siswa'] = $this->db->get('siswa', $config['per_page'], $page)->result_array();	Menentukan halaman data saat ini dan mengambil data siswa sesuai pagination, diurutkan berdasarkan tanggal verifikasi dan no_urut.	Data siswa ditampilkan sesuai halaman dan urutan yang benar.	Berhasil, data diambil sesuai konfigurasi.
8	$data['pagination'] = $this->pagination->create_links();	Membuat link navigasi pagination untuk berpindah halaman pada data siswa yang ditampilkan di halaman HOME.	Link pagination berhasil dibuat dan navigasi dapat diakses.	Berhasil, navigasi pagination sesuai.

Hasil pengujian menunjukkan bahwa pagination pada halaman home bekerja sesuai harapan. Sistem berhasil membatasi jumlah data yang dimuat menjadi 100 entri, mengurangi waktu komputasi dari 30 detik menjadi kurang dari 10 detik. Pagination digunakan untuk membatasi data yang ditampilkan per halaman, sehingga mengurangi beban server dan mempercepat waktu respon aplikasi [12].



Gambar 6. Tampilan 100 Data dengan Pagination pada Halaman Home

Pada gambar 3 tampilan ini menunjukkan bagaimana data yang dibatasi pagination dapat dimuat dengan cepat dan efisien.

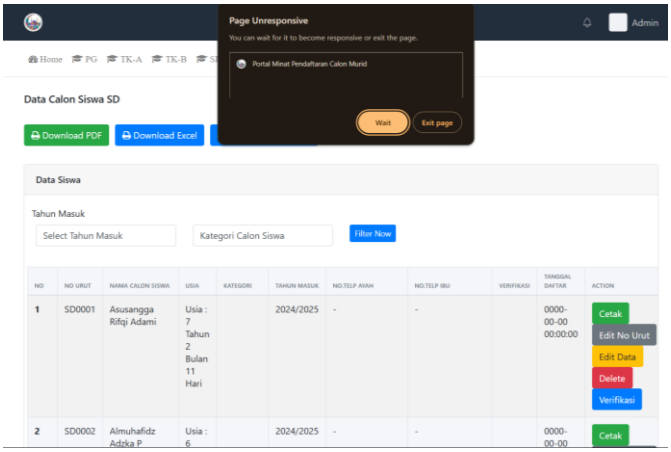
Tabel 3. Hasil Pengujian Filter Tahun Ajaran Halaman SD.

No	Kode yang Diuji	Deskripsi Fungsi	Hasil Pengujian	Kesimpulan
1	$data['title'] = 'Data Siswa SD'; $data['user'] = $this->db->get_where('user', ['email' => $this->session->userdata('email')])->row_array();	Mengatur judul halaman dan mengambil data pengguna dari sesi aktif untuk halaman filter tahun ajaran SD.	Judul halaman dan data pengguna ditampilkan dengan benar.	Berhasil, fungsi sesuai harapan.
2	$key = $this->input->post('key'); $kategori = $this->input->post('kategori');	Mengambil input pengguna dari form POST untuk nilai tahun masuk dan kategori siswa.	Input POST berhasil diterima.	Berhasil, data input diterima dengan benar.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

No	Kode yang Diuji	Deskripsi Fungsi	Hasil Pengujian	Kesimpulan
3	if (!empty($key)) { $this->db->where('tahun_masuk', $key); } if (!empty($kategori)) { $this->db->where('kategori_siswa', $kategori); }	Menyaring data berdasarkan tahun masuk dan kategori siswa jika input tersedia.	Data berhasil disaring sesuai input tahun dan kategori.	Berhasil, filtering sesuai input.
4	$this->db->where('grade', 'SD');	Menambahkan filter untuk data siswa yang berada di grade "SD" saja.	Data terbatas pada grade SD	Berhasil, filter grade bekerja dengan benar.
5	$this->db->order_by('no_urut', 'ASC');	Mengurutkan data siswa berdasarkan no_urut secara ascending.	Data siswa diurutkan dengan benar.	Berhasil, data diurutkan sesuai harapan.
6	$data['sd'] = $this->db->get('siswa')->result_array();	Mengambil data siswa yang sudah difilter untuk ditampilkan di halaman filter SD.	Data berhasil diambil, namun gagal untuk data besar.	Gagal, pagination diperlukan untuk data besar.

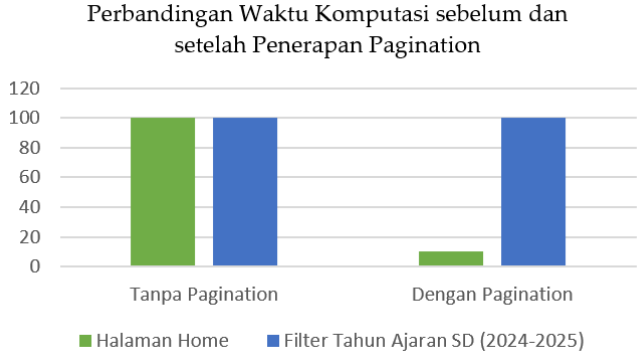
Pada fitur filter tahun ajaran, pagination belum diterapkan. Data yang difilter berdasarkan tahun ajaran diambil secara penuh dari database, menyebabkan sistem mengalami *Error Handling Failure* saat memproses data besar, khususnya pada 1.000 entri atau lebih. Whitebox Testing menunjukkan bahwa sistem gagal menangani data besar tanpa pagination, menegaskan perlunya implementasi pagination pada fitur ini untuk meningkatkan stabilitas aplikasi.



Gambar 7. Tampilan Filter Tahun Ajaran Tanpa Pagination

Pada gambar 4 tampilan ini menunjukkan kegagalan sistem saat filter tahun ajaran digunakan tanpa pagination.

3.2. Pembahasan



Gambar 8. Grafik Perbandingan Waktu Komputasi sebelum dan setelah Penerapan Pagination

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Grafik pada Gambar 8 memperlihatkan bahwa penerapan pagination secara signifikan mengurangi waktu komputasi, dari 30 detik menjadi kurang dari 10 detik untuk 100 entri. Namun, pada data besar (5.403 entri) di halaman Home dan fitur filter tahun ajaran dengan 1.000 data, sistem mengalami *Error Handling Failure*, dengan waktu komputasi lebih dari 100 detik. Hal ini menunjukkan ketidakmampuan sistem untuk memuat seluruh data secara bersamaan tanpa pagination.

Tabel 4. Perbandingan Waktu Komputasi sebelum dan setelah Penerapan Pagination

Kondisi	Jumlah Data	Waktu Komputasi tanpa Pagination	Waktu Komputasi dengan Pagination
Halaman Home	100	30 detik	< 10 detik
Halaman Home	5.403	Error Handling Failure (>100 Detik)	Error Handling Failure (pagination belum diterapkan (>100 Detik))
Filter Tahun Ajaran	1000	Error Handling Failure (>100 Detik)	Error Handling Failure (pagination belum diterapkan (>100 Detik))

Tabel 4 menunjukkan perbandingan waktu komputasi sebelum dan setelah penerapan pagination. Pada data kecil (100 entri), penerapan pagination berhasil menurunkan waktu komputasi dari 30 detik menjadi kurang dari 10 detik, membuktikan efisiensinya dalam mengelola data kecil. Namun, pada data besar (5.403 entri) di halaman Home dan fitur filter tahun ajaran dengan 1.000 data, sistem masih mengalami *Error Handling Failure* tanpa pagination, dengan waktu komputasi lebih dari 100 detik. Hal ini mengindikasikan bahwa pagination sangat diperlukan pada fitur filter untuk menghindari kegagalan sistem dan memastikan stabilitas aplikasi saat menangani data dalam jumlah besar.

Penerapan pagination terbukti meningkatkan stabilitas aplikasi Portal Minat pada halaman Home. Pagination membantu mengurangi beban server, mempercepat waktu respon, dan mencegah Error Handling Failure pada data kecil (100 entri). Namun, tanpa pagination pada fitur filter tahun ajaran, sistem tidak mampu memproses data besar secara efisien, menegaskan kebutuhan mendesak untuk implementasi pagination pada semua fitur filter.

Hasil penelitian ini sejalan dengan studi oleh Mardiani dan Irmayanti, yang menemukan bahwa optimasi query dan pembatasan data dapat secara signifikan menurunkan waktu akses pada sistem berbasis data besar [13]. Implementasi pagination di halaman Home mendukung temuan ini, menunjukkan efisiensi dalam menangani data besar dan meningkatkan performa aplikasi secara keseluruhan. Hasil penelitian ini juga sejalan dengan rekomendasi Riastanjung, yang menyebutkan bahwa penerapan pagination pada sistem web service dapat meningkatkan efisiensi waktu pemrosesan dan stabilitas aplikasi [14]. Hal ini mendukung temuan bahwa pagination di Portal Minat berhasil mengurangi waktu komputasi pada data kecil dan mencegah *Error Handling Failure* pada data besar. Sebagaimana *Google Looker Studio* meningkatkan stabilitas sistem dengan membatasi jumlah data yang divisualisasikan [15], pagination pada Portal Minat terbukti efektif dalam mengurangi beban server dan meningkatkan waktu respon aplikasi.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan pagination pada Portal Minat efektif dalam meningkatkan performa sistem dengan mengurangi waktu pemuatan data secara signifikan, dari 30 detik menjadi kurang dari 10 detik pada data kecil (100 entri). Namun, fitur filter tahun ajaran masih memerlukan optimisasi pagination untuk menghindari kegagalan sistem saat menangani data besar (5.403 entri). Rekomendasi pengembangan lebih lanjut mencakup penerapan pagination pada semua fitur filter dan integrasi dengan metode optimisasi query untuk meningkatkan efisiensi sistem secara menyeluruh. hasil penelitian ini dapat diterapkan pada sistem serupa untuk mendukung pengelolaan data besar, meningkatkan efisiensi operasional, dan menjaga keandalan aplikasi pendaftaran online dalam institusi pendidikan.

DAFTAR PUSTAKA

[1] J. T. Santoso, *ILMU DATA (Data Science)*, 1 ed., vol. 9. Semarang: Yayasanpat, 2023.

[2] L. G. S. Kartika dan K. Rinartha, "Pengaruh Pagination dan Kompleksitas Query Data Terhadap Aspek Kehijauan dari Sistem Informasi Manajemen," *JSI*, vol. Vol. 13, No. 2, hlm. 115–123, Mei 2019.

[3] N. K. T. Wahyuni, "SISTEM INFORMASI PENGELOLAAN DATA PENYEWA PADA KOS MOJA BIAUNG," Tugas Akhir, Politeknik Negeri Bali, Bukit Jimbaran, 2024. Diakses: 11 Desember 2024. [Daring]. Tersedia pada: http://repository.pnb.ac.id/id/eprint/12350

[4] P. R. Sudda, "OPTIMIZING PHP API CALLS WITH PAGINATION AND CACHING," Michigan

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Technological University, Houghton, Michigan, 2024. doi: 10.37099/mtu.dc.etdr/1752.

[5] R. I. Ndaumanu, "Pengujian Sistem Informasi Perpustakaan Berbasis Website dengan Basis Path Testing," *Justek : Jurnal Sains dan Teknologi*, vol. 6, no. 1, hlm. 123–134, Mar 2023, doi: 10.31764/justek.v6i1.13808.

[6] A. Andriyadi, Zulkarnaini, R. R. N. Fikri, dan E. F. Saputri, "Evaluasi Sistem Informasi Perpustakaan Institur Informatika Darmajaya Dengan WhiteBox Testing," *Journal of Innovation Research and Knowledge*, vol. 1, no. 8, hlm. 743–746, Jan 2022.

[7] D. Prasetyo *dkk.*, *Manajemen Proyek Perangkat Lunak*, 1 ed. Yogyakarta: Penamuda Media, 2024.

[8] Marwondo dan R. Melati, *Dasar-Dasar Pengembangan Perangkat Lunak dan Gim*, 1 ed. Jakarta: Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, 2023. [Daring]. Tersedia pada: https://buku.kemdikbud.go.id

[9] Y. S. Triana, A. Rochana, dan A. E. Saputri, "Implementasi Sequential Search Pada Pencarian Data Tarif Aplikasi Perjalanan Dinas Karyawan PT Telkom Akses," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. Vol. 3 No. 2, no. 3, hlm. 202–209, 2019.

[10] P. S. Ganney, S. Pisharody, dan E. Claridge, "Software engineering," *Clin Eng*, hlm. 131–168, Jan 2020, doi: 10.1016/B978-0-08-102694-6.00009-7.

[11] M. F. Landjo, "IMPLEMENTASI WHITE BOX TESTING DENGAN TEKNIK BASIS PATH PADA PENGUJIAN FORM LOGIN," *Jurnal Siliwangi Seri Sains dan Teknologi*, vol. 7, no. 2, hlm. 35–40, 2021, doi: https://doi.org/10.37058/jssainstek.v7i2.4086.

[12] R. Afrilia, Z. R. Mair, dan Juansyah, "Sistem Informasi Pengelolaan Data Alumni Pada UPT SMK Negeri 1 Musi Banyuasin," *Jurnal Nasional Ilmu Komputer*, vol. 2, no. 2, hlm. 112–134, Mei 2021.

[13] G. T. Mardiani dan H. Irmayanti, "ANALISIS OPTIMASI QUERY SQL MENGGUNAKAN TEKNIK HEURISTIC PADA KASUS DATA TRANSAKSI PELANGGAN YANG LAYAK MENDAPATKAN REKOMENDASI PRODUK," *Majalah Ilmiah UNIKOM*, vol. 16, no. 2, hlm. 133–143, Nov 2018, doi: 10.34010/miu.v16i2.1356.

[14] O. Riastanjung, "Pencegahan Man In The Middle Attack Pada Transmisi Data Web Service Menggunakan End To End Encryption (E2EE)," Skripsi(S1), Universitas Maritim Raja Ali Haji, Tanjungpinang, 2024. Diakses: 10 Desember 2024. [Daring]. Tersedia pada: http://repositori.umrah.ac.id/7808/

[15] F. Fitriyadi, F. Hari, S. Al Haris, dan A. Charolina, "PELATIHAN KETRAMPILAN GOOGLE LOOKER STUDIO UNTUK TENAGA KEBERSIHAN UNIVERSITAS SAHID SURAKARTA," *BUDIMAS*, vol. 06, no. 03, hlm. 1–8, 2024.