

Pemanfaatan Algoritma Sequential Search Dalam Penerapan Aplikasi Inventaris

AR Walad Mahfuzhi¹, Sandhy Fernandez², Baharudin Rabbani³

^{1,3} Universitas Muhammadiyah Bengkulu, Bengkulu, Indonesia

² Institut Teknologi Telkom Purwokerto, Purwokerto, Indonesia

Article Info

Article history:

Diterima xx Bulan 20xx

Revisi xx Bulan 20xx

Diterbitkan xx Bulan, 20xx

Keywords:

Inventaris

Sequential Search

Aplikasi

Data

ABSTRAK

Inventaris di SMA XYZ mengalami ketidakefisienan karena melibatkan entri data satu per satu secara manual ke dalam buku inventaris. Hal ini mengakibatkan kesulitan yang berkelanjutan dalam memproses data inventaris, terutama dalam menyajikan informasi inventaris yang komprehensif, yang mengarah pada pelaporan yang tidak efektif. Selain itu, pemrosesan data secara manual sering kali menyebabkan keterlambatan dan kesalahan dalam menghasilkan laporan. Oleh karena itu, ada kebutuhan mendesak akan sistem rekapitulasi yang lebih baik untuk meningkatkan efisiensi pengumpulan dan pengambilan data, meminimalkan risiko kesalahan pencatatan, dan mencegah kehilangan data. Implementasi algoritma pencarian berurutan bertujuan untuk melakukan pencarian secara sistematis dan berurutan dengan membandingkan kunci dengan elemen yang tersimpan di dalam basis data, yang diorganisasikan dalam bentuk array data. Pembuatan sistem yang menggabungkan algoritma Sequential Search, bahasa pemrograman PHP, dan basis data MySQL, telah dirancang dengan cermat untuk penerapan yang efektif. Sistem baru ini menggantikan sistem sebelumnya, mengatasi keterbatasan sistem sebelumnya yang tidak memiliki alur kerja atau algoritma yang ditampilkan dalam penelitian ini.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Sandhy Fernandez,

Institut Teknologi Telkom Purwokerto, DI Panjaitan No.128, Purwokerto 53147, Indonesia

Email: sandhy@ittelkom-pwt.ac.id

1. PENDAHULUAN

Sistem informasi mengacu pada berbagai informasi yang saling berhubungan dan interaktif dalam satu kesatuan bentuk untuk mencapai suatu proses pencarian informasi tertentu [1]. Inventaris adalah daftar yang berisi semua peralatan kantor yang digunakan dalam menjalankan tugas. Pengelolaan inventaris yang tepat sangat penting karena secara tidak langsung berdampak pada proses pencapaian tujuan organisasi. Selain memberikan wawasan tentang aset organisasi, hal ini juga memastikan keamanan dan pelestarian aset tersebut. Tujuan inventaris memainkan peran penting dalam perencanaan, pengadaan, peralatan, dan kebutuhan sumber daya, yang pada akhirnya mendukung pencapaian tujuan institusi secara efektif [2],[3].

Saat ini, sistem pencatatan di SMA XYZ tidak efisien, dengan data yang dicatat secara manual satu per satu di buku inventaris. Hal ini menimbulkan berbagai masalah, termasuk kesulitan dalam menyajikan data inventaris secara efektif, sehingga mengakibatkan pengelolaan informasi inventaris menjadi tidak efisien. Selain itu, pengolahan data masih dilakukan dengan cara tradisional sehingga menghambat pengumpulan data dan dapat menimbulkan masalah [4],[5].

Oleh karena itu, ada kebutuhan akan aplikasi pencatatan yang lebih baik untuk membangun manajemen dan pencarian data yang efisien dan efektif, mengurangi risiko kesalahan pencatatan dan kehilangan data. Di antara berbagai algoritma pencarian, pencarian berurutan dipilih. Algoritma ini sering disebut sebagai algoritma pencarian berurutan, dimana algoritma ini mencari data dengan cara membandingkan secara berurutan antara data target dengan seluruh data yang ada pada suatu dataset, mulai dari awal hingga ditemukan kecocokan [6],[7].

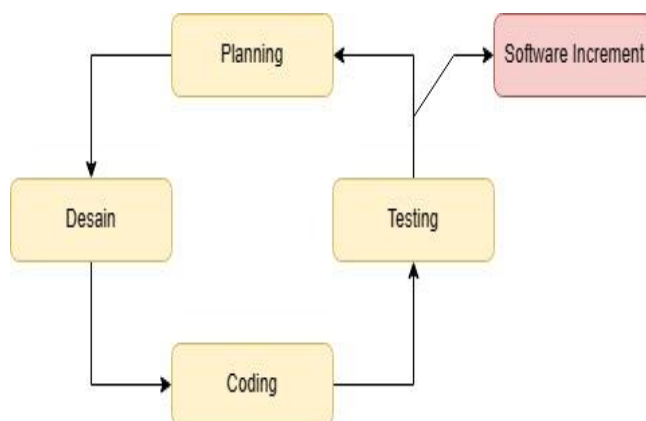
Untuk mengatasi tantangan ini, sebuah solusi telah dikembangkan untuk menyederhanakan tugas-tugas personalia. Solusi ini berupa perancangan aplikasi untuk mengelola dan mengambil data inventaris di SMA XYZ. Aplikasi inventaris berbasis web ini memanfaatkan algoritma pencarian data Sequential Search. Tujuan dari implementasi algoritma Sequential Search adalah untuk melakukan pencarian berurutan secara linier dengan cara membandingkan key dengan elemen yang tersimpan di dalam database, dimana elemen-elemen tersebut disimpan dalam bentuk array data. Proses algoritma Sequential Search melibatkan tiga langkah: membaca larik data, menentukan data target, dan kemudian membandingkannya dengan larik data sampai hasil yang diinginkan ditemukan [3].

2. METODE

2.1 Metode Pengembangan Sistem

Menjelaskan Aplikasi inventori pada penelitian ini dirancang dan dibangun dengan menggunakan model pengembangan sistem linier sekuensial [8]. Aplikasi inventori dibangun dengan menggunakan metode pengembangan sistem Extreme Programming (XP). Metode ini cocok digunakan untuk membangun sistem dengan kebutuhan yang berubah-ubah karena menekankan pada pengembangan yang responsif terhadap kebutuhan pengguna [9],[10].

Aplikasi dapat dibangun dengan cepat menggunakan XP karena metode ini memungkinkan pemrograman berpasangan. Hal ini membuat proses pengembangan menjadi lebih efisien dan mengurangi kesalahan. Selain itu, XP juga menyederhanakan tahapan pengembangan, membuat pengembangan perangkat lunak menjadi lebih fleksibel dan adaptif terhadap perubahan kebutuhan pengguna. Gambar 1 menggambarkan tahapan XP.



Gambar 1. Diagram Alir Desain Sistem

Proses awal dalam XP dimulai dengan perencanaan, yang melibatkan pemahaman langkah-langkah dan menjelaskan fasilitas dan tujuan dari setiap fasilitas. Pada tahap ini, fokus utamanya adalah menentukan fungsi yang akan dimiliki oleh sistem yang sedang dikembangkan. Pernyataan kebutuhan fungsional digunakan untuk menggambarkan fasilitas atau fitur yang dibutuhkan dalam konteks input tertentu dan situasi tertentu. Identifikasi masalah merupakan dasar untuk mendapatkan fungsionalitas, sehingga kita dapat memahami masalah dan kendala yang dihadapi oleh pengguna [11]-[13].

Selanjutnya, tahap selanjutnya adalah desain, dimana pada tahap ini dilakukan pemodelan untuk memvisualisasikan sistem agar dapat dipahami dengan mudah. Perancangan ini meliputi pembuatan struktur menu aplikasi, yang merupakan rancangan bagian-bagian dari menu aplikasi yang akan dikembangkan. Tujuan dari struktur menu aplikasi adalah untuk menentukan urutan akses bagian-bagian tersebut setelah aplikasi selesai dibuat [14],[15].

Langkah selanjutnya adalah implementasi melalui proses coding, dimana desain yang telah disusun diterjemahkan ke dalam bahasa pemrograman dengan menggunakan compiler untuk membuat aplikasi. Aplikasi inventori yang dikembangkan dibangun dengan menggunakan text editor seperti Adobe Dreamweaver CS6 dan notepad++, sedangkan XAMPP digunakan untuk membuat server lokal dan MySQL sebagai basis data melalui fasilitas phpMyAdmin. Bahasa pemrograman yang digunakan dalam proses ini adalah PHP [16].

Langkah terakhir melibatkan pengujian atau tes. Sebelum sistem digunakan secara aktif, disarankan untuk melakukan pengujian terhadap sistem untuk memastikan bahwa sistem dapat beroperasi sesuai dengan harapan pengguna. Pada penelitian ini, pengujian dilakukan dengan mengukur kecepatan algoritma sequential search untuk proses pencarian. Pengukuran kecepatan ini menggunakan page load time, yang menunjukkan berapa lama waktu yang dibutuhkan untuk menampilkan seluruh konten aplikasi. Untuk menghitung page load time,

penelitian ini menggunakan fungsi microtime, sebuah fungsi pada PHP yang menyediakan nilai timestamp saat ini dengan tingkat detail hingga mikro detik [17]-[19].

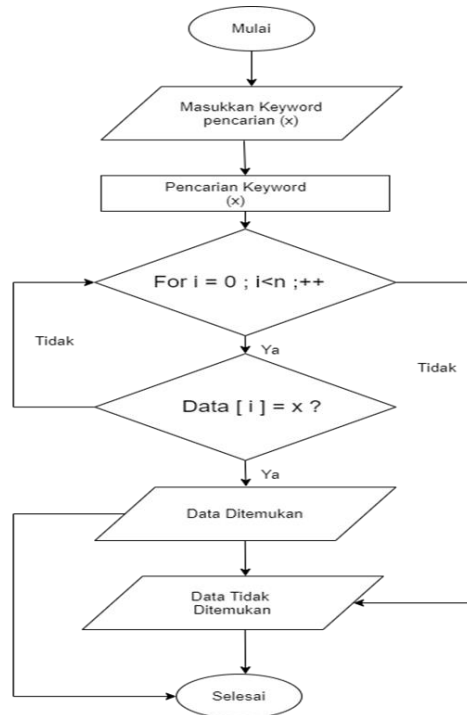
2.2 Algoritma Sequential Search

Algoritma Sequential Search adalah algoritma pencarian data yang dilakukan secara berurutan dari awal hingga akhir atau dari depan ke belakang, tergantung pada kunci yang dicari. Salah satu keuntungan dari metode ini adalah data dapat dengan cepat ditemukan ketika berada di dekat bagian depan [19]. Metode ini merupakan salah satu metode pengolahan data sederhana yang dapat diimplementasikan untuk pencarian data [19].

Alur pencarian informasi pada sistem pembelajaran dapat memanfaatkan algoritma pencarian berurutan untuk mempercepat proses dan mempermudah kegiatan pembelajaran, begitu juga bagi mereka yang membutuhkan bibliografi untuk menemukan materi yang diperlukan. Implementasi alur ini merupakan salah satu contoh penerapannya dalam konteks pendidikan [21],[22].

Algoritma pencarian berurutan dapat digunakan untuk mencari data yang diurutkan dan tidak diurutkan. Dalam aliran pencarian berurutan, proses pencarian data melibatkan perbandingan data satu per satu dengan data awal hingga akhir aliran. Jika data cocok, maka data ditemukan. Jika data tidak ditemukan, itu berarti data yang dicari tidak tersedia dalam dataset [23], [24].

Algoritma yang akan digunakan adalah algoritma Sequential Search, di mana teknik yang digunakan untuk mencari data melibatkan pencarian secara berurutan dari depan (awal) ke belakang (akhir) dalam sebuah larik berdasarkan kata kunci atau istilah pencarian, tanpa perlu melakukan pengurutan. Iterasi terus berlanjut hingga tidak ada data yang sama, dan kemudian pencarian berhenti [25]-[29]. Langkah-langkah dalam proses algoritma pencarian berurutan adalah sebagai berikut, seperti yang digambarkan pada Gambar 1 di bawah ini.



Gambar 2. Diagram Alir Algoritma Sequential Search [28]-[30]

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, pengembangan aplikasi inventori menggunakan metode *Sequential Search*. Metode ini mengatur proses pengembangan aplikasi ke dalam serangkaian tahap yang berurutan dan terstruktur, di mana setiap tahap dijalankan secara berurutan, bergantung pada penyelesaian tahap sebelumnya. Pendekatan ini diadopsi untuk memastikan kelancaran dan keberhasilan pengembangan aplikasi. Tahapan dalam metode *Sequential Search* dimulai dengan analisis kebutuhan, diikuti dengan desain sistem, implementasi, dan pengujian. Mengikuti proses berurutan ini diantisipasi untuk memastikan perkembangan pengembangan aplikasi yang lancar dan efektif, menghasilkan hasil yang memuaskan.

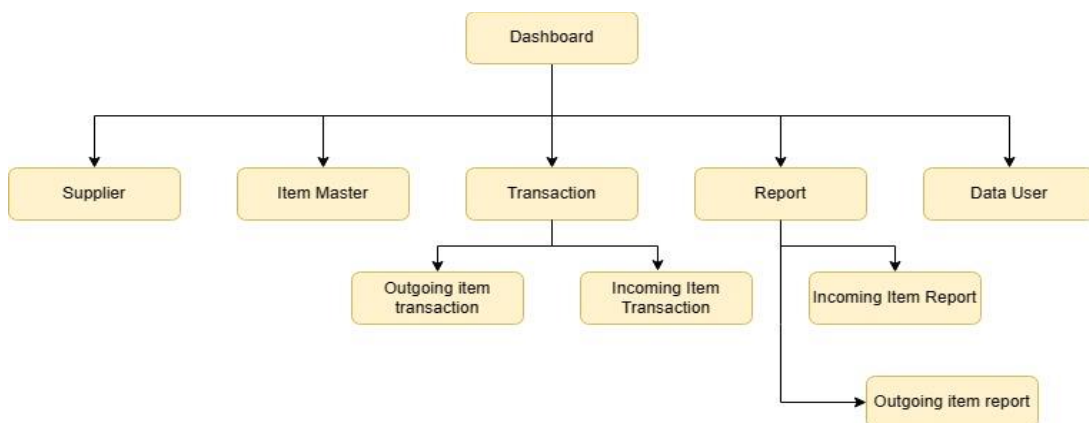
3.1 Perencanaan

Untuk menentukan kebutuhan fungsional aplikasi inventori, dilakukan wawancara di SMA XYZ. Berdasarkan data yang diperoleh, dapat disimpulkan bahwa kebutuhan fungsional aplikasi inventori yang akan dikembangkan antara lain:

- Sistem harus dapat mengelola pemasok, sehingga pengguna dapat melihat, menambah, mengedit, dan menghapus data pemasok.
- Sistem harus dapat mengelola item, memungkinkan pengguna untuk melihat, menambah, mengedit, dan menghapus data item.
- Sistem harus dapat mengelola transaksi barang yang masuk.
- Sistem harus dapat menghasilkan laporan inventaris dalam format PDF.

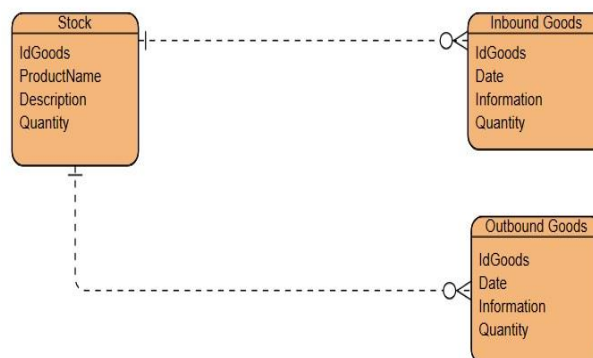
3.2 Desain

Perancangan struktur menu aplikasi meliputi perencanaan bagian-bagian menu aplikasi yang akan dibangun serta penentuan urutan akses setelah aplikasi selesai dibangun. Gambar 3 terlampir merupakan rancangan arsitektur menu aplikasi yang sedang dikembangkan.



Gambar 3. Arsitektur Menu

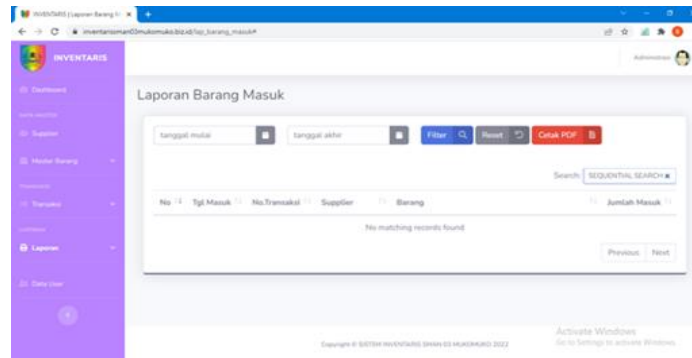
Sebelum melanjutkan ke tahap pengembangan, kita akan membahas tahapan-tahapan Entity Relationship Diagram (ERD), seperti yang digambarkan pada Gambar 4.



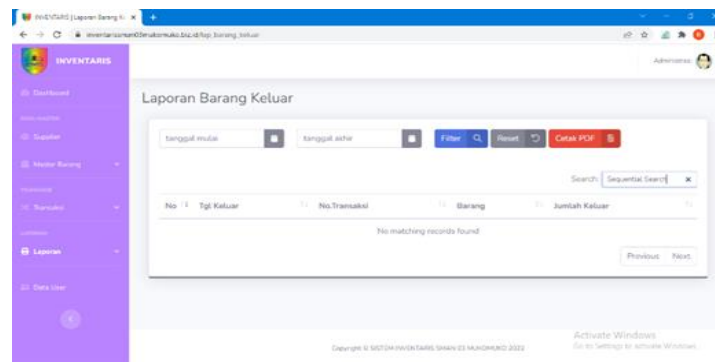
Gambar 4. ERD Aplikasi Inventaris

3.3 Pengkodean

Penelitian ini mengimplementasikan Algoritma *Sequential Search* dalam pencarian data inventaris di SMA XYZ, dimana proses pencarian dilakukan secara berurutan dari depan ke belakang atau dari awal hingga akhir. Pada laporan barang masuk dan keluar akan muncul menu pencarian pada gambar 5 dan 6.

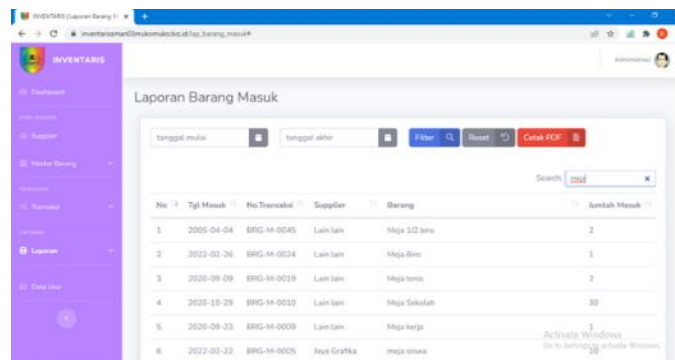


Gambar 5. Mencari Laporan Barang Masuk Menggunakan Algoritma *Sequential Search*

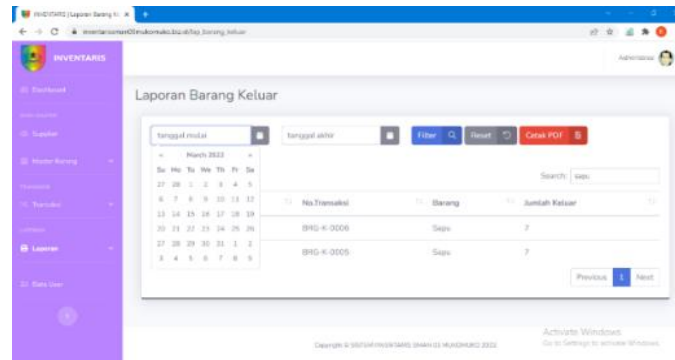


Gambar 6. Mencari Data Laporan Barang Keluar Menggunakan Algoritma *Sequential Search*

Pada penelitian ini, algoritma *Sequential Search* digunakan untuk pencarian data inventaris di SMA XYZ, dimana proses pencarian data dilakukan secara berurutan dari awal (indeks pertama) hingga akhir (indeks terakhir) dalam sebuah larik. Pencarian dimulai dari elemen pertama pada array dan dilanjutkan secara berurutan sampai data ditemukan atau sampai akhir array tercapai. Dalam fungsi ini, pencarian berurutan diproses. Ketika pengguna memasukkan istilah pencarian di kotak pencarian, aplikasi memproses dan bekerja untuk menampilkan data item yang masuk dan keluar yang dicari oleh pengguna. Ini mencari di dalam database, dan sistem menyajikan data seperti yang ditunjukkan pada Gambar 7 dan 8.



Gambar 7. Hasil Pencarian di Laporan Barang Masuk Menggunakan Algoritma *Sequential Search*



Gambar 8. Hasil Pencarian di Laporan Barang Keluar Menggunakan Algoritma *Sequential Search*

Proses pencarian di *Sequential Search* dimulai ketika pengguna memasukkan kata kunci yang ingin dicari di kotak pencarian. Aplikasi kemudian memproses filter dan menampilkan hasil pencarian jika data cocok dengan yang ada di database.

3.4 Tes

Pengujian kecepatan hasil pencarian yang dilakukan oleh algoritma *Sequential Search* dilakukan dengan mengamati waktu muat halaman dengan menggunakan fungsi *microtime*. Fungsi *microtime* digunakan untuk mengukur waktu yang dibutuhkan halaman untuk memproses pencarian. Algoritma *Sequential Search* dapat menemukan kata yang Anda cari secara real-time, dengan waktu rata-rata sekitar 0,01 ms. Dalam hal ini, kata-kata yang dicari disusun menurut abjad, sehingga memudahkan algoritma untuk bekerja.

Selain itu, terdapat 568 kata dalam database item, yang menunjukkan bahwa jumlah data relatif terbatas atau tidak terlalu banyak. Hasil pengujian menunjukkan beberapa kelebihan dari algoritma *Sequential Search*, antara lain: Pencarian dapat dilakukan dengan cepat untuk data yang terbatas atau tidak terlalu banyak, *Sequential Search* merupakan algoritma yang sederhana dan tidak rumit.

Namun demikian, perlu dicatat bahwa algoritma *Sequential Search* melakukan pencarian satu per satu secara berurutan. Oleh karena itu, jika data yang Anda cari berada di belakang atau di bagian paling akhir, proses pencarian bisa jadi akan memakan waktu lebih lama. Selain itu, perlu dilakukan uji coba penggunaan algoritma *Sequential Search* pada data dalam jumlah besar untuk mengukur kinerjanya secara lebih komprehensif.

3.5 Algoritma *Sequential Search*

Pencarian dalam aplikasi berbasis web ini, yang mengimplementasikan algoritma *Sequential Search*, dilakukan secara real-time. Aplikasi ini menggunakan event untuk mendeteksi apakah pengguna telah selesai mengetikkan kata kunci di kotak pencarian dan segera memulai pencarian kata kunci tersebut. Proses ini membuat pencarian menjadi lebih efektif dan efisien. Kode untuk algoritma pencarian berurutan yang diimplementasikan pada aplikasi dapat dilihat pada Gambar 9.

```

public function index()
{
    $data['title'] = 'Supplier';
    $data['supplier'] = $this->supplier_model->data()->result();

    $data = $this->input->post('search');
    $query = $this->db->get('barang_masuk')->result();
    for ($i = 0; $i < count($query); $i++) {
        if ($data == $query[$i]->id_barang_masuk) {
            $data['search'] = $query[$i];
        } else {
            $data['search'] = "Data Tidak Ditemukan!";
        }
    }

    $this->load->view('templates/header', $data);
    $this->load->view('supplier/index', $data);
    $this->load->view('templates/footer');
}

```

Gambar 9. Kode Algoritma Sequential Search Kode Algoritma *Sequential Search*

Pada Gambar 9, terdapat dua komponen utama dalam algoritma Sequential Search: larik dan kunci. Larik berfungsi sebagai wadah untuk sekumpulan data yang tidak terpisahkan, sedangkan kunci merepresentasikan istilah pencarian atau data yang akan digunakan sebagai referensi untuk mencari di dalam larik. Selain itu, ada metode, yang merupakan kumpulan instruksi atau perintah dalam program dengan nama tertentu.

Data dalam database diekstrak dan kemudian disimpan dalam sebuah array. Di dalam array, hanya nama item yang disimpan karena data akan dicari berdasarkan nama item, seperti yang telah direncanakan sebelumnya. Alur kerja dalam algoritma pencarian berurutan adalah sebagai berikut:

- Mengatur jumlah data yang akan ditempatkan ke dalam larik menggunakan variabel int ukuran = panjang larik.
- Mengulangi data menggunakan perulangan FOR untuk setiap data dan melakukan pencarian berurutan pada data yang tersimpan di dalam larik, menggunakan perulangan for (for int i = 0; i < size; i++).
- Menggunakan pernyataan IF untuk memeriksa kecocokan antara data input dalam larik dengan kata kunci unik yang dicari, dengan menggunakan kondisi if (array[i] == key).
- Jika kecocokan ditemukan selama pencarian, pencarian akan dihentikan, dan nilai tersebut dikembalikan sebagai hasil pencarian, dengan menggunakan pernyataan return array[i].
- Jika tidak ada data yang cocok ditemukan selama pencarian, maka akan mengembalikan nilai -i sebagai indikasi bahwa data tersebut tidak ada, dengan menggunakan pernyataan return -i.

3.6 Pengujian Aplikasi

Aplikasi yang telah dirancang diuji untuk mengetahui kesesuaiannya dengan implementasi algoritma *Sequential Search*. Pengujian aplikasi dilakukan dengan menggunakan proses *black-box* testing untuk memastikan bahwa hasil yang ditampilkan sesuai dengan rancangan aplikasi. Hasil pengujian dapat dilihat pada Tabel 1 berikut ini:

Tabel 1. Pengujian Aplikasi

Unit/Acara	Hasil Pengujian	Kesimpulan
Tambahkan Produk	Berhasil menampilkan data dan berhasil menambahkan produk	Valid
Edit Produk	Berhasil menampilkan data dan berhasil mengedit produk.	Valid

Menghapus Produk	Berhasil menampilkan data dan berhasil menghapus produk	Valid
Satuan Barang	Berhasil membuat data unit barang, menambahkan data, mengedit, dan menghapus	Valid
Kategori produk	Berhasil menampilkan data kategori barang, deskripsi, data yang diedit, dan data yang dihapus	Valid
Informasi produk	Berhasil menampilkan semua produk dan stok yang tersedia	Valid
Transaksi Barang Masuk	Berhasil menampilkan penambahan item, pengeditan item, dan penghapusan item	Valid
Transaksi barang keluar	Berhasil menampilkan penambahan item, pengeditan item, dan penghapusan item.	Valid
Laporan barang masuk	Berhasil menampilkan pencarian, pengaturan ulang, dan mencetak laporan	Valid
Laporan barang keluar	Berhasil menampilkan pencarian, pengaturan ulang, dan mencetak laporan	Valid

Setelah melalui serangkaian pengujian black box yang komprehensif, dapat dipastikan bahwa aplikasi ini telah berhasil melewati ujian dan sesuai dengan spesifikasi yang telah ditentukan. Semua fungsi yang diuji, meliputi pencarian data, penambahan, pengeditan, dan penghapusan entitas seperti pemasok dan barang, serta proses transaksi, memberikan hasil yang sesuai dengan ekspektasi. Kinerja aplikasi selama fase pengujian menunjukkan responsif dan stabilitas. Oleh karena itu, aplikasi ini dapat diandalkan untuk memenuhi persyaratan fungsional dan menawarkan pengalaman pengguna yang memuaskan. Hasil yang baik yang diperoleh dari pengujian kotak hitam ini menanamkan kepercayaan diri dalam kesiapan aplikasi untuk tahap selanjutnya, termasuk pengujian lebih lanjut dan potensi penyebaran ke pengguna akhir.

Dalam menguji kinerja waktu pencarian di situs web, fokus utamanya adalah pada efisiensi sistem dalam menyajikan hasil pencarian kepada pengguna. Waktu yang dibutuhkan untuk menemukan dan menampilkan data dapat memberikan wawasan tentang seberapa cepat dan responsif situs web dalam memenuhi kebutuhan pengguna. Pengujian ini membantu mengidentifikasi potensi peningkatan dalam proses pencarian data, memastikan bahwa pengalaman pengguna dapat dioptimalkan. Dengan mengevaluasi waktu pencarian, kita dapat menilai keefektifan sistem dan memastikan bahwa pengguna dapat dengan cepat dan efisien menemukan informasi yang mereka cari. Berikut adalah hasil dari uji kecepatan yang dilakukan pada hasil pencarian yang dipamerkan seperti yang disajikan pada Tabel 2.

Tabel 2. Pengujian Kecepatan

Unit/Acara	Waktu
Hasil pencarian di Laporan Barang Masuk.	0,01 ms
Hasil pencarian di Laporan Barang Keluar.	0,015 ms

4. KESIMPULAN

Rancangan sistem dengan implementasi algoritma *Sequential Search* yang berhasil dilakukan, menggunakan pemrograman PHP, dan memanfaatkan database MySQL telah diaplikasikan dan dapat diimplementasikan. Algoritma *Sequential Search* dapat menemukan kata yang dicari secara real-time, dengan waktu rata-rata sekitar 0,01 ms. Dalam hal ini, kata-kata yang dicari disusun menurut abjad, sehingga memudahkan algoritma untuk bekerja. Terdapat 568 kata dalam database item, yang menunjukkan bahwa jumlah data relatif terbatas atau tidak terlalu banyak. Hasil menunjukkan beberapa kelebihan dari algoritma *Sequential Search*, yaitu pencarian dapat dilakukan dengan cepat untuk data yang terbatas atau tidak terlalu banyak dan juga algoritma *Sequential Search* bekerja dengan cara mencocokkan setiap elemen data secara berurutan dari awal hingga akhir, sehingga cocok digunakan ketika data tidak terlalu besar.

UCAPAN TERIMA KASIH

Terima kasih yang sebesar-besarnya kepada semua pihak yang telah terlibat dalam penelitian ini. Dukungan, bimbingan, dan kontribusi dari semua pihak telah membantu menjadikan penelitian ini berjalan dengan baik. Terima kasih atas dedikasi dan kerja keras yang telah diberikan. Semoga hasil penelitian ini bermanfaat bagi banyak orang dan dapat memberikan kontribusi positif bagi perkembangan ilmu pengetahuan.

REFERENSI

- [1] S. Fernandez, C. Prihantoro, and A. Kharisma Hidayah, "Implementasi Weighted Product Pada Pemilihan Dosen Terbaik Di Universitas Muhammadiyah Bengkulu," *Pseudocode*, vol. 8, no. 2, pp. 126-133, 2021.
- [2] R. Nur Annisa, M. Ugiarto, K. Gunung Kelua Samarinda, and K. Timur, "SISTEM INVENTARIS SARANA DAN PRASARANA DI FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS MULAWARMAN," *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*, vol. 2, no. 1, 2017, [Online]. Available: www.bsnp-indonesia.org/id/wp-
- [3] A. Srrahyu and F. Suryani, "Implementasi Algoritma Sequential Search Pada Aplikasi Ensiklopedia Dermatologi Berbasis Mobile," *INFOKES*, vol. 11, no. 1, 2021.
- [4] A. Muhazir, M. Fakhriza, and E. Sutejo, "Implementasi Metode Sequential Dalam Pencarian Pendistribusian Barang Pada Sistem Cargo Integration," *Jurnal & Penelitian Teknik Informatika*, vol. 2, no. 2, 2017.
- [5] A. Sonita and M. Sari, "IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING UNTUK PENCARIAN NOMOR SURAT PADA SISTEM ARSIP ELEKTRONIK," 2018. [Online]. Available: www.ejournal.unib.ac.id/index.php/pseudocode
- [6] Y. Rahmanto, J. Alfian, and R. Indra Borman, "Penerapan Algoritma Sequential Search pada Aplikasi Kamus Bahasa Ilmiah Tumbuhan 21," 2021.
- [7] S. K. Dirjen dkk., "Implementasi Sequential Search Pada Pencarian Data Tarif Aplikasi Perjalanan Dinas Karyawan PT Telkom Akses," *RESTI*, vol. 3, no. 2, pp. 202-209, 2019.
- [8] C. Tri Lestari and F. Latifah, "Aplikasi Pencatatan Keuangan Pribadi Dengan Analisa SWOT Menggunakan Algoritma Sequential Search Berbasis Mobile," *JISAMAR*, vol. 3, no. 2, 2019.
- [9] A. A. Rismayadi and L. Jamaliah, "IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING PADA APLIKASI E-OFFICE," 2019.
- [10] S. Informasi dkk., "SISTEM INFORMASI LOKASI SEJARAH PAHLAWAN SULAWESI SELATAN MENGGUNAKAN ALGORITMA SEQUENTIAL SEARCH BERBASIS ANDROID," 2020.
- [11] Y. Religia, "Analisis Algoritma Sequential Search Dan Binary Search Pada Big Data," *Pelita*, vol. 14, no. 1, pp. 74-79, 2019.
- [12] M. Y. Fathoni, R. F. Waliulu, A. Susanto, and M. Nishom, "Perancangan Aplikasi Penjualan Berbasis Client Server Pada Kedai WKWK Kota Purwokerto Menggunakan Metode Prototype," *J. Inform.*, vol. 7, no. 1, pp. 49-54, 2022.
- [13] W. M. Ashour, R. Z. Muqat, A. B. Alqazzaz, and S. R. Abdelnabi, "Meningkatkan skema algoritma sekuensial dasar dengan menggunakan algoritma koloni semut," *IEEE 7th Palest. Int. Conf. Electr. Comput. Eng. PICECE 2019*, 2019.
- [14] E. Mühendisli, "EEG İşaretlerinin Ardışık İleri Yönde Seçim Algoritmaları ile Öznitelik Seçimi ve Farklı Sınıflandırıcılar ile İncelenmesi: Fitir Seçimi ve Analizi Sinyal EEG dengan Algoritma Seleksi Maju Berurutan dan Pengklasifikasi yang Berbeda," hlm. 1-4, 2020.
- [15] C. Zhang dan Y. Zu, "Algoritma penambangan pola sekuensial paralel utilitas tinggi yang efisien," *Proc. - 21st IEEE Int. Conf. High Perform. Comput. Commun. 17th IEEE Int. Conf. Kota Cerdas 5th IEEE Int. Conf. Data Sci. Syst. HPC/SmartCity/DSS 2019*, pp. 2798-2803, 2019.
- [16] S. Shahzad, "Belajar dari Pengalaman: Analisis Proses Pemrograman Ekstrim," *Konferensi Internasional Keenam tentang Teknologi Informasi 2009: Generasi Baru*, Las Vegas, NV, USA, 2009, hal. 1405-1410, doi: 10.1109/ITNG.2009.299.
- [17] S. Shahzad, "Belajar dari Pengalaman: Analisis Proses Pemrograman Ekstrim," *Konferensi Internasional Keenam tentang Teknologi Informasi 2009: Generasi Baru*, Las Vegas, NV, USA, 2009, hal. 1405-1410, doi: 10.1109/ITNG.2009.299.
- [18] V. Jovanovic, T. Murphy dan A. Greca, "Penggunaan pemrograman ekstrim (XP) dalam pengajaran pemrograman pengantar," *32nd Annual Frontiers in Education*, Boston, MA, USA, 2002, hal. F1G-, doi: 10.1109/FIE.2002.1158141.
- [19] P. Sharma dan N. Hasteer, "Analisis metodologi pengembangan pemrograman sekuensial linier dan pemrograman ekstrim untuk aplikasi game," *2016 International Conference on Communication and Signal Processing (ICCS)*, Melmaruvathur, India, 2016, hlm. 1916-1920, doi: 10.1109/ICCS.2016.7754505.
- [20] A. M. K. Cheng dan P. Wu, "Work-in-Progress: Memasukkan Penjadwalan Berbasis Tenggat Waktu dalam Model Pemrograman Penugasan untuk Komputasi Paralel Skala Ekstrim," *2018 IEEE Real-Time Systems Symposium (RTSS)*, Nashville, TN, USA, 2018, hlm. 131-134, doi: 10.1109/RTSS.2018.00022.
- [21] L. Pan, "Merancang Model Pengajaran Berbasis Ekstrim untuk Mata Kuliah Pemrograman Pertama," *Konferensi Internasional Ilmu Komputasi dan Kecerdasan Komputasi (CSCI) 2021*, Las Vegas, NV, USA, 2021, hlm. 1091-1094, doi: 10.1109/CSCI54926.2021.00230.

- [22] L. Liu dan Y. Lu, "Penerapan metode agile dalam sistem manajemen backstage situs web perusahaan: Praktik untuk pemrograman ekstrim," Konferensi Internasional ke-2 2012 tentang Elektronik Konsumen, Komunikasi dan Jaringan (CECNet), Yichang, Cina, 2012, hlm. 2412-2415, doi: 10.1109/CECNet.2012.6201545.
- [23] J. Kivi, D. Haydon, J. Hayes, R. Schneider dan G. Succi, "Pemrograman ekstrim: pengalaman desain tim universitas," Konferensi Kanada 2000 tentang Teknik Elektro dan Komputer. Prosiding Konferensi. Menavigasi ke Era Baru (Cat. No. 00TH8492), Halifax, NS, Kanada, 2000, hal. 816-820 vol.2, doi: 10.1109/CCECE.2000.849579.
- [24] V. Jovanovic, T. Murphy dan A. Greca, "Penggunaan pemrograman ekstrim (XP) dalam pengajaran pemrograman pengantar," 32nd Annual Frontiers in Education, Boston, MA, USA, 2002, hal. F1G-, doi: 10.1109/FIE.2002.1158141.
- [25] J. Choudhari dan U. Suman, "Siklus Hidup Pemeliharaan Iteratif Menggunakan Pemrograman eXtreme," Konferensi Internasional 2010 tentang Kemajuan Teknologi Terkini dalam Komunikasi dan Komputasi, Kottayam, India, 2010, hal. 401-403, doi: 10.1109/ARTCom.2010.52.
- [26] S. Bryant, "Masalah Ganda: Memadukan Metode Kualitatif dan Kuantitatif dalam Studi Programmer eXtreme," 2004 IEEE Symposium on Visual Languages - Human Centric Computing, Roma, Italia, 2004, hal. 55-61, doi: 10.1109/VLHCC.2004.20.
- [27] H. Li dan H. Wang, "Pendekatan pelatihan dua tingkat berdasarkan mesin pembelajaran otomatis mesin pembelajaran ekstrim untuk klasifikasi data," Konferensi Internasional ke-18 tentang Kecerdasan dan Keamanan Komputasi (CIS) 2022, Chengdu, Cina, 2022, hlm. 171-175, doi: 10.1109/CIS58238.2022.00043.
- [28] C. A. Wellington, "Mengelola kursus proyek menggunakan Pemrograman Ekstrim," Prosiding Konferensi Tahunan ke-35 Frontiers in Education, Indianapolis, IN, USA, 2005, hal. T3G-1, doi: 10.1109/FIE.2005.1611948.
- [29] Z. Li-li, H. Lian-feng dan S. Qin-ying, "Penelitian tentang Persyaratan untuk Model Pemrograman Ekstrim Berkualitas Tinggi," 2011 International Conference on Information Management, Innovation Management and Industrial Engineering, Shenzhen, China, 2011, hlm. 518-522, doi: 10.1109/ICIM.2011.132.
- [30] A. Vihavainen dan M. Luukkainen, "Hasil dari Transisi Tiga Tahun ke Metode Pemrograman Ekstrim," Konferensi Internasional IEEE ke-13 tentang Teknologi Pembelajaran Lanjutan, Beijing, Cina, 2013, hal. 336-340, doi: 10.1109/ICALT.2013.104.