

# Optimalisasi Algoritma Linear Search Berdasarkan Pola Akses Pengguna Dan Komparasi Metode Implementasi

<sup>1</sup>Jajang Didin, <sup>2</sup>Nabilla Azzahra, <sup>3</sup>Isad Firdaus

<sup>1</sup>Universitas Sebelas April

<sup>2</sup>Politeknik Negeri Bandung

<sup>3</sup>Universitas Satu Bandung

Jl. Angrek Situ No.19, Sumedang, Jawa Barat 45323

email : [1240160121050@student.unsap.ac.id](mailto:1240160121050@student.unsap.ac.id), [nabilla.azzahra.ksy23@polban.ac.id](mailto:nabilla.azzahra.ksy23@polban.ac.id), [isad.firdaus@satu.ac.id](mailto:isad.firdaus@satu.ac.id)

## ABSTRACT

Data search efficiency is a crucial aspect of information system performance, particularly in applications handling dynamic data volumes within the public and private sectors. This study aims to analyze optimization strategies for the Linear Search algorithm through a user access pattern approach and to compare the technical performance between iterative and recursive implementation methods. The research methodology employs a quantitative approach involves computational experiments on datasets ranging from 1 million to 100 million records, alongside qualitative case studies on various existing information systems such as libraries, retail, and civil registry services. Test results indicate that despite sharing the same asymptotic time complexity, the iterative method proves to be far superior to the recursive method, demonstrating a significant execution time difference (0.0133 seconds versus 28.38 seconds on 100 million records) and more efficient memory usage. Furthermore, this study finds that the effectiveness of Linear Search can be drastically improved through the application of access pattern-based heuristic strategies, such as Move-to-Front for popular data and Timestamp Ordering for chronological data. The conclusion of this research recommends the use of the iterative Linear Search method combined with adaptive data position management as the optimal solution for systems characterized by dynamic and unordered data.

**Keywords** – Search Algorithms, Computational Efficiency, Linear Search, User Access Patterns

## 1. Introduction

Dalam era transformasi digital saat ini, kecepatan dan akurasi temu kembali informasi (*information retrieval*) telah menjadi indikator utama kualitas layanan sebuah sistem informasi. Berbagai organisasi, mulai dari sektor pemerintahan hingga ritel, menghadapi tantangan dalam mengelola volume data yang terus bertumbuh. Di tengah keberagaman algoritma pencarian yang kompleks seperti *hashing* atau *indexing* berbasis *tree*, algoritma *Linear Search* atau pencarian sekuensial tetap mempertahankan relevansinya. Algoritma ini sering menjadi pilihan utama untuk dataset berskala kecil hingga menengah atau pada kondisi data yang tidak terurut karena kemudahan implementasinya dan tidak adanya kebutuhan akan pra-pemrosesan pengurutan (*sorting*) yang memakan sumber daya komputasi. Namun, implementasi yang kurang tepat sering kali menyebabkan inefisiensi sistem, seperti antrean layanan yang panjang atau waktu respons aplikasi yang lambat [1,2].

Permasalahan nyata terlihat pada berbagai studi kasus di lapangan. Di sektor perpustakaan, proses pendataan dan sirkulasi manual sering kali lambat dan tidak akurat, menghambat layanan peminjaman [3]. Begitu pula di sektor ritel, manajemen inventaris yang tidak efisien dapat menyebabkan kesalahan stok dan kehilangan peluang penjualan [4]. Di sisi lain, instansi pemerintahan seperti Dinas Kependudukan dan Pencatatan Sipil (Disdukcapil) menghadapi kendala antrean panjang akibat

lambatnya verifikasi data penduduk. Masalah-masalah ini menuntut adanya solusi algoritma pencarian yang tidak hanya fungsional tetapi juga dioptimalkan sesuai dengan perilaku akses pengguna dan karakteristik data yang dikelola[5].

Penelitian ini bertujuan untuk mengeksplorasi strategi optimalisasi algoritma *Linear Search* agar mampu memberikan kinerja maksimal. Fokus utama penelitian mencakup analisis pengaruh pola akses pengguna seperti pencarian berdasarkan identitas unik, kata kunci semantik, atau data terkini terhadap kecepatan penemuan data. Selain itu, penelitian ini juga akan membandingkan secara teknis efisiensi antara pendekatan pemrograman iteratif dan rekursif untuk menentukan metode implementasi terbaik. Melalui analisis mendalam ini, diharapkan dapat dirumuskan rekomendasi teknis yang aplikatif untuk meningkatkan efisiensi sistem informasi di berbagai domain tanpa harus beralih ke struktur data yang lebih kompleks.

## 2. Research Method

Penelitian ini menerapkan pendekatan campuran (*mixed methods*) yang menggabungkan analisis eksperimental kuantitatif dan studi kasus kualitatif. Tahap pertama melibatkan tinjauan literatur sistematis terhadap tujuh studi kasus implementasi algoritma *Linear Search* di berbagai platform, termasuk sistem perpustakaan web, aplikasi *e-commerce*, ensiklopedia kesehatan *mobile*, dan sistem administrasi kependudukan. Analisis ini difokuskan untuk mengidentifikasi pola akses pengguna, kendala implementasi, dan dampak penggunaan algoritma terhadap efisiensi operasional[1,6].

Tahap kedua adalah eksperimen komputasi untuk mengukur kinerja teknis algoritma. Pengujian dilakukan dengan membandingkan dua paradigma implementasi utama, yaitu metode iteratif dan metode rekursif. Parameter yang diukur meliputi waktu eksekusi (*execution time*) dan penggunaan memori (*memory usage*) pada variasi ukuran dataset, mulai dari 1 juta, 10 juta, hingga 100 juta elemen data integer. Data kinerja ini dianalisis untuk membuktikan hipotesis mengenai efisiensi memori dan kecepatan proses antara kedua metode tersebut, serta untuk memahami batasan skalabilitas dari algoritma *Linear Search* dalam skenario *worst-case* dan *average-case*.

## 3. Result & Analysis

### Analisis Pola Akses Pengguna

Efektivitas algoritma *Linear Search* sangat bergantung pada posisi data dalam *array* pencarian, yang secara langsung dipengaruhi oleh bagaimana pengguna mengakses data tersebut. Berdasarkan analisis terhadap studi kasus, ditemukan tiga pola akses dominan yang mempengaruhi strategi optimalisasi. Pola pertama adalah akses identitas unik (*Unique ID Access*), di mana pengguna mencari entitas spesifik seperti Kode Buku atau Nomor Induk Kependudukan (NIK). Dalam kasus Perpustakaan Adventure Book, pencarian berdasarkan kode buku yang unik memungkinkan algoritma berhenti segera setelah data ditemukan, menghasilkan waktu pencarian rata-rata yang sangat cepat, yaitu sekitar 10,64 milidetik. Hal serupa terjadi pada sistem absensi karyawan yang mencocokkan ID *fingerprint*, di mana pencarian bersifat eksak dan deterministik untuk keperluan penggajian.

Pola kedua adalah akses eksploratif dan semantik, yang umumnya terjadi pada aplikasi yang membutuhkan pencarian berbasis teks atau kategori, seperti pada Ensiklopedia Dermatologi dan Toko Sepatu Barok. Pengguna cenderung menggunakan kata kunci gejala penyakit atau nama produk yang mungkin menghasilkan banyak kecocokan (*multiple matches*) atau membutuhkan pencarian parsial. Pada skenario ini, algoritma sering kali harus menelusuri seluruh *database* untuk memastikan semua data relevan ditemukan, yang secara teoritis meningkatkan beban komputasi mendekati *worst-case* ( $O(n)$ ). Namun, implementasi pada kasus toko sepatu menunjukkan bahwa meskipun harus menelusuri data produk, sistem digital ini tetap mampu meningkatkan kecepatan layanan hingga 30% dibandingkan

pencarian manual, membuktikan bahwa kecepatan komputasi *Linear Search* masih sangat memadai untuk inventaris skala menengah [4].

Pola ketiga adalah akses layanan publik massal yang bersifat *real-time*, seperti yang terlihat pada sistem pendaftaran Disdukcapil . Masyarakat cenderung memantau status permohonan terkini, menciptakan pola akses yang berfokus pada data terbaru (*recency*). Dalam kondisi ini, urutan penyimpanan data menjadi krusial. Jika data baru disimpan di akhir *database* dan pencarian dilakukan dari awal, sistem akan selalu mengalami kondisi terburuk. Studi pada Disdukcapil juga mengungkapkan bahwa penggunaan *query* SQL langsung ke *database* (0.291 ms) jauh lebih cepat dibandingkan *Linear Search* yang dilakukan di sisi aplikasi melalui ORM (7.837 ms), menyoroti pentingnya mempertimbangkan *overhead* abstraksi dalam implementasi algoritma pada pola akses massal[5] .

### Evaluasi Kinerja: Iteratif vs Rekursif

Perdebatan mengenai metode implementasi terbaik antara pendekatan iteratif (menggunakan *loop*) dan rekursif (pemanggilan fungsi diri sendiri) dijawab melalui data eksperimen yang menunjukkan disparitas kinerja yang ekstrem. Hasil pengujian pada dataset berukuran 100 juta data menunjukkan bahwa metode iteratif hanya membutuhkan waktu eksekusi 0,0133 detik, sementara metode rekursif membutuhkan waktu hingga 28,38 detik . Perbedaan ini sangat signifikan, di mana metode rekursif berjalan ribuan kali lebih lambat akibat *overhead* mekanisme pemanggilan fungsi yang berulang, yang memaksa prosesor untuk terus melakukan *context switching* dan alokasi *stack frame* baru.

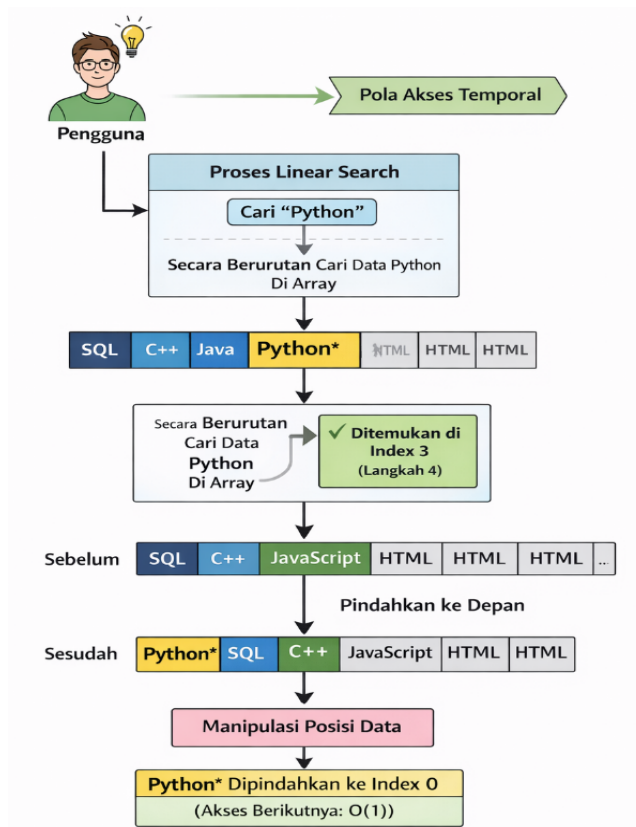
Tabel 1. Perbandingan Waktu Eksekusi Metode Iteratif vs Rekursif

Ukuran Dataset	Metode Iteratif (Detik)	Metode Rekursif (Detik)	Analisis Performa
1 Juta Data	0,0143 s	25,5147 s	Iteratif ±1.784x lebih cepat
10 Juta Data	0,0160 s	26,8470 s	Iteratif ±1.677x lebih cepat
<b>100 Juta Data</b>	<b>0,0133 s</b>	<b>28,3863 s</b>	<b>Iteratif ±2.134x lebih cepat</b>

Dari sisi penggunaan memori, metode rekursif juga terbukti tidak efisien. Pada pengujian yang sama, metode rekursif mengonsumsi memori yang lebih besar dibandingkan iteratif karena tumpukan *call stack* yang terus bertambah seiring kedalaman rekursi . Meskipun selisih memori mungkin terlihat kecil pada dataset kecil, risiko utama dari pendekatan rekursif adalah terjadinya *Stack Overflow Error* pada dataset besar atau pada lingkungan dengan memori terbatas seperti perangkat *mobile* . Temuan ini menegaskan bahwa untuk sistem produksi yang mengutamakan stabilitas dan kecepatan, seperti sistem pengajian atau aplikasi layanan publik , implementasi *Linear Search* wajib menggunakan pendekatan iteratif[2].

### Strategi Optimalisasi Berbasis Heuristik

Berdasarkan analisis pola akses dan kinerja teknis, optimalisasi *Linear Search* tidak harus dilakukan dengan mengganti algoritma, melainkan dengan memanipulasi posisi data agar sesuai dengan perilaku pengguna. Strategi pertama yang direkomendasikan adalah *Move-to-Front*. Strategi ini sangat efektif untuk pola akses berulang, seperti pada katalog produk atau buku perpustakaan yang memiliki item “best seller”. Dengan memindahkan data yang baru saja ditemukan ke posisi paling depan *array*, pencarian berikutnya untuk item yang sama akan menjadi instan ( $O(1)$ ). Strategi ini memanfaatkan prinsip *temporal locality* untuk secara drastis menurunkan waktu pencarian rata-rata[7-10].



Gambar 1. Strategi Move-to-Front

Strategi kedua adalah pengurutan berbasis waktu (*Timestamp Ordering*) atau LIFO (*Last-In, First-Out*) untuk pencarian status transaksi[11-13]. Pada sistem seperti Disdukcapil atau pelacakan pesanan, pengguna hampir selalu mencari data terbaru[14]. Oleh karena itu, mekanisme penyimpanan data harus memastikan data terbaru ditempatkan di indeks awal pencarian, atau algoritma pencarian dimodifikasi untuk menelusuri dari indeks terakhir ke awal. Penerapan strategi heuristik sederhana ini dapat mengubah kinerja sistem[15] dari yang sebelumnya lambat karena menelusuri data historis, menjadi sangat responsif terhadap kebutuhan informasi terkini pengguna[16].

**4. Conclusion**

Penelitian ini menyimpulkan bahwa algoritma *Linear Search* tetap menjadi solusi yang relevan dan andal untuk sistem informasi modern, khususnya yang mengelola data dinamis dan tidak terurut. Kunci keberhasilan implementasinya tidak hanya terletak pada kesederhanaan algoritma itu sendiri, tetapi pada kesesuaian teknik implementasi dengan karakteristik penggunaan sistem. Secara teknis, metode implementasi iteratif terbukti mutlak lebih unggul dibandingkan metode rekursif, baik dari segi kecepatan eksekusi maupun efisiensi memori, sehingga harus menjadi standar dalam pengembangan perangkat lunak produksi .

Lebih jauh, penelitian ini menyoroti bahwa kinerja *Linear Search* dapat ditingkatkan secara signifikan tanpa harus beralih ke algoritma kompleks (seperti *Binary Search* yang mewajibkan pengurutan) dengan menerapkan strategi manajemen data yang cerdas. Pemanfaatan heuristik pola akses pengguna, seperti strategi *Move-to-Front* untuk data populer dan prioritas data terbaru (*recency*), terbukti mampu meminimalkan kelemahan utama *Linear Search* dalam menangani dataset besar. Dengan demikian, pengembang sistem disarankan untuk selalu menganalisis pola perilaku pengguna terlebih dahulu sebelum merancang logika pencarian data demi mencapai efisiensi operasional yang optimal.

## References

- [1] H. Situmorang, "Analisa Algoritma Pada Metoda Pencarian Linier, Biner Dan Interpolasi," *J. Mahajana Inf.*, vol. 2, no. 2, pp. 31–41, 2018, doi: 10.51544/jurnalmi.v2i2.177.
- [2] E. Lutfina, N. Inayati, and G. W. Saraswati, "Analisis Perbandingan Kinerja Metode Rekursif dan Metode Iteratif dalam Algoritma Linear Search," *Komputika J. Sist. Komput.*, vol. 11, no. 2, pp. 143–150, 2022, doi: 10.34010/komputika.v11i2.5493.
- [3] W. S. Wahyuni, S. Andryana, and B. Rahman, "Penggunaan Algoritma Sequential Searching Pada Aplikasi Perpustakaan Berbasis Web," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 2, pp. 294–302, 2022, doi: 10.29100/jupi.v7i2.2646.
- [4] Alfath Aji Rizkyanto, Tri Ginanjar Laksana, and Mukhlis, "Penerapan Algoritma Linear Search dalam Sistem Informasi Penjualan Sepatu Berbasis Web: Studi Kasus Toko Barok," *J. Ilm. Multidisiplin*, vol. 2, no. 1, pp. 15–29, 2025, doi: 10.62282/juilmu.v2i1.15-29.
- [5] W. R. Theresa Sukmawaty Putri, Saepudin Nirwan, "Implementation of Linear Search Algorithm To Optimize Registration Search in Disdukcapil Subang District," vol. 3, no. 3, pp. 115–124, 2024.
- [6] A. Srirahayu and F. Suryani, "Implementasi Algoritma Sequential Search Pada Aplikasi Ensiklopedia Dermatologi berbasis Mobile," *Infokes*, vol. 11, no. 1, pp. 49–53, 2021.
- [7] H. Kurniawan and S. Rudiarto, "Aplikasi Penghitung Upah Lembur Berdasarkan Jumlah Waktu Lembur Dengan Menggunakan Algoritma Linear Search," *Petir*, vol. 12, no. 1, pp. 47–53, 2019, doi: 10.33322/petir.v12i1.417.
- [8] Rahmayani, R., Firmansyah, E., & Hikmah, H. U. (2025). Inovasi Layanan Antar Jemput Paket Surat PT Pos Indonesia Berdasarkan Penjualan dan Minat Beli. *JOVISHE: Journal of Visionary Sharia Economy*, 4(1), 33-47.
- [9] Firmansyah, E., Rahman, A. B. A., & Subiyakto, A. A. (2023). Pengukuran Kesiapan Kota Cerdas Berdasarkan SNI ISO 37122: 2019. *Infoman's: Jurnal Ilmu-ilmu Informatika dan Manajemen*, 17(2).
- [10] Zulfikar, W. B., Irfan, M., Ghufron, M., Jumadi, J., & Firmansyah, E. (2020). Marketplace affiliates potential analysis using cosine similarity and vision-based page segmentation. *Bulletin of Electrical Engineering and Informatics*, 9(6), 2492-2498.
- [11] Ramadhan, N. D., Fhatturohmah, S., Ramadhani, S., & Firmansyah, E. (2023). Analysis of Digital Wallet Usage on Consumptive Lifestyle. *Journal of Islamic Economics and Business*, 3(2), 118-136.
- [12] Sutara, B., & Firmansyah, E. (2021). Design and Build Student Attendance System Using Fingerprint. *J-Tin's-Jurnal Teknik Informatika*, 5(1).
- [13] Wahana, A., Firmansyah, E., Al Rosyid, H. I., Fuadi, R. S., & Maylawati, D. S. A. (2021). Fuzzy Tahani Method in the Recommendation System for Selecting Mountain Tourism Destinations in West Java.
- [14] Tamrin, M. A., Rizki, B., Nodas, A., Rahman, A., & Firmansyah, E. (2020). Perbandingan Penggunaan Metode Topsis dan Metode AHP dalam Penilaian Kinerja pada Karyawan (PT XYZ). *Infoman's: Jurnal Ilmu-ilmu Informatika dan Manajemen*, 14(1).
- [15] Firmansyah, E., Herdiana, D., & Yuniarto, D. (2020, October). Examining readiness of e-Learning implementation using information system readiness impact model. In *2020 8th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-5). IEEE.
- [16] Firmansyah, E., Rosmawati, R., Fuadi, R. S., Fauzy, D., & Ramdhani, M. A. (2019, December). Design of expert system to determine the proper diet using harmony search method. In *Journal of Physics: Conference Series* (Vol. 1402, No. 7, p. 077006). IOP Publishing.