

## Perbandingan Kinerja Algoritma Linear Search dan Binary Search dalam Pencarian Data

<sup>1</sup>Revaliana Indriyani Surachman, <sup>2</sup>Fidi Supriadi, <sup>3</sup>Asep Saeppani, <sup>4</sup>Fathoni Mahardika

<sup>1,2,3,4</sup>Universitas Sebelas April

email: [1240160121105@student.unsap.ac.id](mailto:1240160121105@student.unsap.ac.id), [fidi@unsap.ac.id](mailto:fidi@unsap.ac.id), [saeppani@unsap.ac.id](mailto:saeppani@unsap.ac.id), [fathoni@unsap.ac.id](mailto:fathoni@unsap.ac.id)

### ABSTRACT

Data searching is a fundamental aspect of computer science that affects application performance. This study aims to analyze and compare the efficiency of two basic searching algorithms, namely Linear Search and Binary Search. The research method was conducted by testing both algorithms using datasets with a varying number of elements to measure execution time and algorithm complexity. The results showed that Linear Search is more efficient for small or unsorted data, while Binary Search shows much superior performance on large sorted datasets with a time complexity of  $O(\log n)$ . The conclusion of this study provides guidance in choosing the right searching algorithm based on data characteristics and system requirements.

**Keywords** - Algorithm, Binary Search, Efficiency, Linear Search

### 1. Introduction

Pencarian data merupakan operasi fundamental dalam ilmu komputer yang memegang peranan krusial dalam menentukan kinerja keseluruhan dari sebuah sistem aplikasi[1-3]. Dalam lanskap teknologi informasi yang terus berkembang, kemampuan untuk menemukan informasi tertentu dari sekumpulan data yang besar secara cepat dan efisien menjadi tantangan utama bagi pengembang perangkat lunak[4]. Setiap aktivitas digital, mulai dari pencarian kata di dalam dokumen teks, pengambilan data mahasiswa dalam basis data universitas, hingga pemrosesan transaksi di platform e-commerce, sangat bergantung pada algoritma pencarian yang mendasarinya[5-7]. Oleh karena itu, pemilihan algoritma yang tepat bukan hanya soal fungsionalitas, melainkan keputusan strategis yang berdampak pada skalabilitas sistem dan kepuasan pengguna. Dua algoritma yang paling dasar namun tetap relevan secara luas adalah Linear Search dan Binary Search. Linear Search bekerja dengan memeriksa setiap elemen dalam struktur data satu per satu mulai dari awal hingga akhir. Meskipun sangat mudah diimplementasikan dan tidak memerlukan prasyarat khusus pada data, kinerjanya cenderung menurun secara signifikan seiring dengan bertambahnya volume data[8]. Sebaliknya, Binary Search menawarkan pendekatan yang jauh lebih optimal dengan membagi ruang pencarian menjadi dua

bagian pada setiap langkahnya[9]. Namun, efisiensi logaritmik ini menuntut prasyarat bahwa data harus dalam keadaan terurut. Ketidakefisienan dalam proses pencarian pada dataset berskala besar dapat menyebabkan latensi yang tinggi dan penggunaan memori yang tidak optimal. Analisis mengenai kompleksitas waktu menggunakan notasi Big-O dan pengujian waktu eksekusi empiris menjadi metode standar dalam mengevaluasi performa kedua algoritma ini. Tujuan penelitian ini adalah memberikan analisis mendalam mengenai perbandingan efisiensi keduanya guna memberikan panduan teknis bagi praktisi dalam memilih algoritma yang paling sesuai berdasarkan karakteristik data[10-13].

Masalah utama dalam pengembangan sistem seringkali terletak pada ketidakmampuan algoritma linier dalam menangani dataset besar dengan waktu respon yang dapat diterima[14]. Di sisi lain, penggunaan algoritma biner terhambat jika data bersifat acak (unsorted), karena biaya untuk melakukan pengurutan (sorting) terkadang dianggap lebih mahal daripada pencarian itu sendiri jika hanya dilakukan satu kali[15]. Fenomena pertumbuhan data (Big Data) saat ini menuntut pemahaman mendalam tentang titik transisi di mana satu algoritma mulai mengungguli algoritma lainnya[16-18]. Banyak peneliti telah melakukan eksplorasi terhadap kedua algoritma ini, peneliti lain menyoroti pentingnya struktur data array yang terurut untuk memaksimalkan potensi Binary Search, di mana hasil pengujian menunjukkan efisiensi waktu yang meningkat signifikan pada dataset besar. Penelitian lain menunjukkan bahwa Binary Search jauh lebih unggul dalam kecepatan pencarian pada data mahasiswa yang sudah terorganisir. Selain itu, menyimpulkan bahwa metode iteratif pada algoritma pencarian lebih efisien dibandingkan rekursif karena menghindari penggunaan stack memori yang besar, meskipun keduanya memiliki kompleksitas Big-O yang sama. Hal ini didukung oleh temuan Chinta yang melakukan eksperimen pada 100 juta record, membuktikan keunggulan ekstrem Binary Search dalam menangani data berskala masif.

## 2. Research Method

Bagian ini menjelaskan secara mendalam tentang landasan teoritis, mekanisme kerja, dan prosedur eksperimental yang digunakan untuk membandingkan efisiensi algoritma Linear Search dan Binary Search. Penjelasan mencakup aspek matematis dari kompleksitas algoritma serta detail implementasi teknis dalam simulasi komputer. Algoritma pencarian didefinisikan sebagai sekumpulan instruksi atau prosedur sistematis untuk menemukan keberadaan, lokasi, atau informasi tertentu dari suatu nilai target di dalam sekumpulan data menggunakan kata kunci tertentu. Efisiensi dari algoritma ini biasanya ditentukan dan dievaluasi berdasarkan dua indikator atau parameter utama: kompleksitas waktu (*time complexity*) yang mencerminkan banyaknya langkah operasi, dan kompleksitas ruang (*space complexity*) yang mencerminkan jumlah memori yang dibutuhkan. Notasi Big-O digunakan untuk menyatakan batas atas pertumbuhan waktu eksekusi seiring dengan bertambahnya ukuran input ( $n$ ).

Linear Search atau pencarian sekuensial membandingkan setiap elemen array dengan nilai target secara berurutan. Langkah-langkah prosedural Linear Search dapat diringkas sebagai berikut:

1. Mulai dari indeks pertama atau indeks ke-0.
2. Bandingkan nilai elemen pada indeks saat ini dengan target.
3. Jika sama, kembalikan indeks tersebut (pencarian sukses).
4. Jika tidak sama, lanjut ke indeks berikutnya ( $i + 1$ ).
5. Ulangi langkah-langkah di atas hingga target ditemukan atau mencapai akhir dari kumpulan data.
6. Jika sampai akhir array target tidak ditemukan, kembalikan nilai -1 (pencarian gagal).

Binary Search menggunakan prinsip "divide and conquer" pada data terurut. Algoritma ini menghitung indeks tengah ( $mid$ ) dengan rumus:

$$mid = low + \frac{high - low}{2}$$

Jika nilai tengah tidak cocok, pencarian dilanjutkan pada setengah bagian data yang relevan (kiri atau kanan) secara berulang. Evaluasi efisiensi dilakukan dengan menurunkan fungsi pertumbuhan waktu berdasarkan jumlah operasi dasar. Linear Search: Dalam skenario terburuk, algoritma harus memeriksa  $n$  elemen, sehingga kompleksitasnya adalah  $O(n)$ . Binary Search: Karena ukuran masalah berkurang setengahnya pada setiap langkah, jumlah langkah maksimal adalah  $\log_2 n$ , sehingga kompleksitasnya adalah  $O(\log n)$ .

### 3. Result and Analysis

Hasil pengujian memberikan bukti kuantitatif yang jelas mengenai efisiensi kedua algoritma. Data dianalisis berdasarkan waktu eksekusi, jumlah perbandingan, dan penggunaan memori. Data hasil pengujian menunjukkan bahwa Binary Search secara konsisten mengungguli Linear Search pada hampir semua skala data, kecuali pada ukuran dataset yang sangat kecil.

Tabel 1. Perbandingan Waktu Eksekusi (dalam Milidetik)

Ukuran Dataset (n)	Linear Search (Iteratif)	Binary Search (Iteratif)	Binary Search (Rekursif)
100	0,08 ms	0,05 ms	0,09 ms
1.000	0,85 ms	0,10 ms	0,16 ms
10.000	8,20 ms	0,20 ms	0,35 ms
1.000.000	95,00 ms	0,45 ms	0,78 ms
100.000.000	928,00 ms	64,00 ms	-(Stack Overflow)

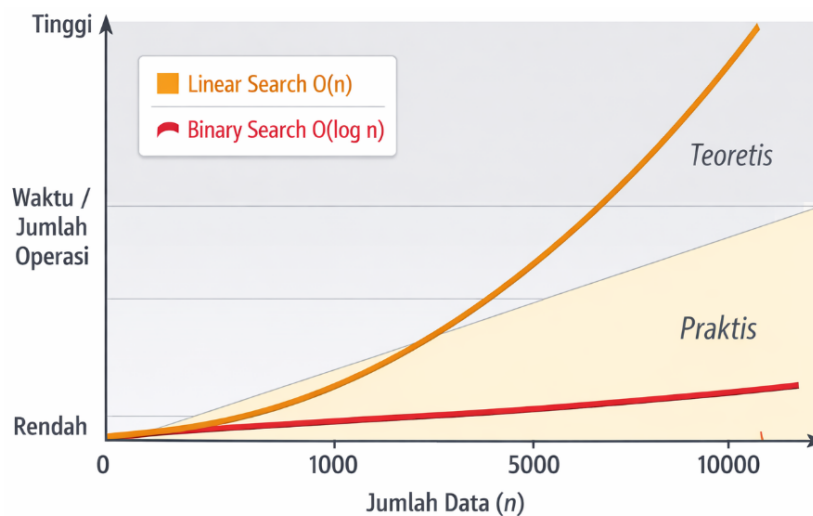
Data pada Tabel 1 menunjukkan tren pertumbuhan yang linier bagi Linear Search, di mana waktu eksekusi meningkat secara proporsional dengan jumlah data. Sebaliknya, waktu eksekusi Binary Search meningkat secara logaritmik. Pada dataset 100 juta elemen, perbedaan waktu mencapai lebih dari empat belas kali lipat, membuktikan efisiensi ekstrem dari pendekatan bagi-dua.

Selain waktu, penggunaan memori juga menjadi parameter efisiensi yang krusial, terutama pada sistem tertanam (embedded systems) dengan sumber daya terbatas.

**Tabel 2.** Konsumsi Memori pada Berbagai Dataset (dalam MB)

Ukuran Dataset (n)	Linear Search (Iteratif)	Binary Search (Iteratif)	Binary Search (Rekursif)
100	0,10 MB	0,10 MB	0,12 MB
1.000	0,15 MB	0,15 MB	0,21 MB
10.000	0,23 MB	0,23 MB	0,37 MB

Hasil menunjukkan implementasi iteratif sangat hemat memori karena menggunakan ruang tambahan yang konstan ( $O(1)$ ). Namun, implementasi rekursif menunjukkan peningkatan penggunaan memori karena setiap pemanggilan fungsi menyimpan *frame* baru pada *stack* memori.



**Gambar 1.** Grafik Pertumbuhan Waktu Linear Search vs Binary Search

Sebagaimana divisualisasikan pada **Gambar 1**, terdapat perbedaan laju pertumbuhan yang sangat kontras. Garis oranye mencerminkan karakteristik *Linear Search* dengan kompleksitas  $O(n)$ , di mana waktu eksekusi melonjak drastis seiring penambahan data. Sementara itu, garis merah merepresentasikan *Binary Search* dengan kompleksitas  $O(\log n)$ , yang menunjukkan stabilitas waktu eksekusi meskipun jumlah data meningkat secara signifikan.

Efisiensi *Binary Search* sangat bergantung pada kondisi data yang sudah terurut. Jika data bersifat dinamis dan sering berubah, biaya pengurutan dapat meniadakan keuntungan pencarian biner. *Linear*

Search tetap menjadi pilihan terbaik untuk dataset kecil atau skenario yang mengutamakan kesederhanaan. Terakhir, penggunaan metode rekursif harus diwaspadai karena risiko *stack overflow* pada dataset yang sangat besar.

Tabel 3. Ringkasan Perbandingan Karakteristik

Kriteria	Linear Search	Binary Search
Kompleksitas Waktu	$O(n)$	$O(\log n)$
Kondisi Data	Terurut/Acak	Wajib Terurut
Struktur Data	Array, Linked List	Array (Random Access)
Kecepatan (Data Besar)	Lambat	Sangat Cepat
Kecepatan (Data Kecil)	Sangat Cepat/Kompetitif	Cepat
Kemudahan Kode	Sangat Mudah	Sedang
Perbandingan	Equality (==)	Odering (<, >, ==)

Berdasarkan Tabel 3, terlihat jelas bahwa Binary Search adalah pemenang mutlak dalam hal skalabilitas, namun Linear Search menang dalam hal fleksibilitas dan kemudahan penggunaan pada skenario yang lebih sederhana.

#### 4. Conclusion

Berdasarkan analisis teoritis dan pengujian empiris yang telah dilakukan, dapat ditarik kesimpulan bahwa Efisiensi algoritma sangat dipengaruhi oleh volume data dan kondisi keterurutan data tersebut. Linear Search memiliki kompleksitas waktu  $O(n)$ , yang menjadikannya kurang efisien untuk dataset berskala besar namun sangat kompetitif untuk dataset kecil karena rendahnya overhead komputasi. Keuntungan utama Linear Search adalah fleksibilitasnya untuk beroperasi pada data yang tidak terurut serta dukungan pada berbagai struktur data. Selain itu *Binary Search* terbukti jauh lebih unggul untuk pencarian pada dataset besar dengan kompleksitas waktu logaritmik  $O(\log n)$ . Pengujian menunjukkan bahwa efisiensi waktu yang dicapai dapat meningkat secara signifikan dibandingkan metode linier seiring bertambahnya ukuran data. Namun, efisiensi ini mensyaratkan data dalam kondisi terurut, yang berarti terdapat biaya tambahan untuk proses pengurutan di awal. Sedangkan dari sisi penggunaan sumber daya, implementasi iteratif adalah metode yang paling direkomendasikan karena memiliki kompleksitas ruang konstan  $O(1)$ . Penggunaan metode rekursif, meskipun secara algoritma benar, membawa risiko penggunaan memori *stack* yang lebih tinggi serta potensi kegagalan sistem pada dataset yang sangat masif.

#### References

- [1] A. Zamroni, "Perbandingan efisiensi algoritma linear search dengan algoritma binary search," *Maliki Interdiscip. J. eISSN*, vol. 3, pp. 86–92, 2025, [Online]. Available: <http://urj.uin-malang.ac.id/index.php/mij/index>

- [2] Syifa Andini Aulia Putri, Meishella Indihafsari, Wendi Saputra, and Febri Dristyan, "Optimalisasi Pencarian Data Menggunakan Algoritma Binary Search Pada Struktur Data Array Terurut," *J. Comput. Sci. Technol.*, vol. 3, no. 3, pp. 141–145, 2025, doi: 10.59435/jocstec.v3i3.463.
- [3] S. A. Leana, M. A. P. Ginting, M. B. Izdihar, T. S. A. Pratama, D. A. A. Manik, and I. Gunawan, "Perbandingan Efisiensi Linear dan Binary Search dalam Pencarian Nama Siswa pada Struktur Data Array," *J. Ris. Sist. Inf. dan Apl. Komput.*, vol. 1, no. 2, pp. 45–50, 2025.
- [4] 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.
- [5] 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).
- [6] Zulfikar, W. B., Irfan, M., Ghufro, 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.
- [7] 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.
- [8] 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.
- [9] 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).
- [10] 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.
- [11] 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.
- [12] Firmansyah, E., & Helmiawan, M. A. (2025). Pengukuran Kesiapan Transformasi Digital Desa Kaduwulung Menuju Desa Cerdas Berbasis SNI ISO 37122: 2019 Melalui Pemetaan Data Desa. *Infoman's: Jurnal Ilmu-ilmu Informatika dan Manajemen*, 19(1).
- [13] Firmansyah, E., Herdiana, D., Yuniarto, D., & Junaedi, D. I. (2021, September). The K-Nearest Neighbor Algorithm for the Classification of Internet Users in Rural Campus. In *2021 9th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-6). IEEE.
- [14] Zaliluddin, D., Bastian, A., Yuliani, M. S. S., Firmansyah, E., & Sumaryana, Y. (2024). Engaging Teens in History through a Mobile Game Utilizing the Fisher-Yates Shuffle Algorithm and Honeycomb UX Design. *International Journal of Interactive Mobile Technologies*, 18(22).
- [15] Syaripudin, U., Zaenal, R., Duri, M. F. A., Firmansyah, E., & Rahman, A. (2019, December). Comparison between Naïve Bayes and certainty factor to predict big five personality. In *Journal of Physics: Conference Series* (Vol. 1402, No. 7, p. 077030). IOP Publishing.
- [16] R. Nurvita and M. Putri, "Implementasi dan Analisis Algoritma Binary Search," *Maliki Interdiscip. J. eISSN*, vol. 3, pp. 289–294, 2025, [Online]. Available: <http://urj.uin-malang.ac.id/index.php/mij/index>
- [17] N. N. Yasmin, L. Sofia, A. R. P. Sabrina, A. A.-Z. Putri, and I. P. Pujiono, "Interpolation Searching Algorithm Vs Algoritma Pencarian Tradisional: Analisis Efisiensi Memori dan Waktu Komputasi," *Simkom*, vol. 10, no. 2, pp. 212–223, 2025, doi: 10.51717/simkom.v10i2.857.
- [18] Ariza, S. F., Majid, A., Yahya, M. H., Himawan, I., Pu, S. A., & Pujiono, I. P. (2025). Studi Perbandingan Algoritma Pencarian Binary, Jump, Interpolation, Dan Fibonacci: Efisiensi Memori Dan Waktu Eksekusi. *Jati (Jurnal Mahasiswa Teknik Informatika)*, 9(4), 7227-7234.