

Analisis Komprehensif Algoritma Binary Search: Kompleksitas, Implementasi, dan Perbandingan Kinerja

Dariah Jaudah¹, Putri Cahya Pratiwi^{2*}

^{1,2}STMIK Kaputama, Binjai, Indonesia
ptrichy24@gmail.com^{1*}

Abstrak

Pencarian data merupakan salah satu operasi fundamental dalam ilmu komputer dan rekayasa perangkat lunak. Efisiensi algoritma pencarian sangat menentukan kinerja sistem secara keseluruhan, terutama ketika menangani himpunan data berskala besar. Jurnal ini menyajikan analisis komprehensif mengenai algoritma *Binary Search* (Pencarian Biner), mencakup prinsip kerja, analisis kompleksitas waktu dan ruang, serta perbandingannya dengan algoritma *Sequential Search* (Pencarian Sekuensial). Melalui kajian literatur, ditunjukkan bahwa *Binary Search* menawarkan efisiensi yang jauh lebih tinggi dengan kompleksitas waktu $O(\log n)$ dibandingkan $O(n)$ pada *Sequential Search*, meskipun mensyaratkan data yang telah terurut. Jurnal ini juga membahas implementasi algoritma dalam berbagai studi kasus praktis, seperti aplikasi kamus dan sistem informasi.

Kata Kunci: *Binary Search, Algoritma Pencarian, Kompleksitas Waktu, Sequential Search, Struktur Data*

1. Pendahuluan

Dalam era digital saat ini, volume data yang dihasilkan dan diproses oleh sistem informasi terus meningkat secara eksponensial. Kemampuan untuk menemukan informasi spesifik secara cepat dan akurat dari sekumpulan data yang besar menjadi sangat krusial [1]. Algoritma pencarian (*search algorithm*) adalah metode sistematis yang digunakan untuk menemukan posisi atau keberadaan suatu elemen dalam struktur data.

Terdapat berbagai jenis algoritma pencarian, di mana dua yang paling umum digunakan adalah *Sequential Search* (atau *Linear Search*) dan *Binary Search*. *Sequential Search* bekerja dengan memeriksa setiap elemen satu per satu dari awal hingga akhir, yang menjadikannya sederhana namun tidak efisien untuk data berskala besar. Sebaliknya, *Binary Search* menggunakan pendekatan *divide and conquer* (membagi dan menaklukkan) yang secara signifikan mengurangi ruang pencarian pada setiap iterasinya [2].

Meskipun *Binary Search* sangat efisien, penerapannya memiliki prasyarat mutlak: himpunan data harus dalam keadaan terurut (*sorted*) [3]. Jurnal ini bertujuan untuk mengkaji secara mendalam mekanisme kerja *Binary Search*, menganalisis kompleksitas algoritmanya, membandingkan kinerjanya dengan *Sequential Search*, serta meninjau penerapannya dalam berbagai aplikasi dunia nyata berdasarkan literatur akademik terkini.

2. Tinjauan Pustaka

2.1. Konsep Dasar Binary Search

Binary Search adalah algoritma pencarian yang beroperasi pada ruang pencarian yang terurut secara monoton (baik menaik maupun menurun). Algoritma ini bekerja dengan membandingkan nilai yang dicari (*target*) dengan elemen tengah (*middle element*) dari himpunan data. Jika nilai target sama dengan elemen tengah, pencarian selesai. Jika nilai target lebih kecil, pencarian dilanjutkan pada separuh pertama himpunan data. Sebaliknya, jika lebih besar, pencarian dilanjutkan pada separuh kedua. Proses ini diulang dengan membagi dua ruang pencarian hingga elemen ditemukan atau ruang pencarian habis [4].

2.2. Perbandingan dengan Sequential Search

Beberapa penelitian telah membandingkan kinerja *Binary Search* dan *Sequential Search*. Markuci dkk. (2025) menemukan bahwa *Binary Search* cenderung lebih stabil dan membutuhkan waktu yang jauh lebih sedikit dalam proses pencarian data dibandingkan *Sequential Search*, menjadikannya lebih cocok untuk aplikasi dengan basis data besar [5]. Studi lain oleh Putri (2025) dan Religia (2025) juga mengkonfirmasi efisiensi *Binary Search* yang signifikan dibandingkan *Sequential Search* [1], [2].

Namun, penting untuk dicatat bahwa *Sequential Search* memiliki keunggulan dalam kesederhanaan implementasi dan tidak memerlukan data yang terurut terlebih dahulu [6]. Sebuah penelitian oleh Balogun (2020) bahkan menunjukkan bahwa untuk himpunan data yang sangat kecil, *Linear Search* (nama lain dari *Sequential Search*) bisa lebih cepat karena *overhead* yang lebih rendah dibandingkan *Binary Search* [10]. Meskipun demikian, secara umum, *Binary Search* jauh lebih efisien untuk data yang terurut dan berukuran besar.

3. Metodologi dan Analisis Algoritma

3.1. Mekanisme Kerja Algoritma

Langkah-langkah sistematis dari algoritma *Binary Search* dapat diuraikan sebagai berikut:

1. Tentukan batas awal (*low*) dan batas akhir (*high*) dari himpunan data (array).
2. Hitung indeks tengah (*mid*) dengan rumus: $\text{mid} = \text{low} + (\text{high} - \text{low}) / 2$.
3. Bandingkan elemen pada indeks *mid* dengan nilai target:
 - Jika $\text{array}[\text{mid}] == \text{target}$, kembalikan indeks *mid*.
 - Jika $\text{array}[\text{mid}] < \text{target}$, perbarui batas awal: $\text{low} = \text{mid} + 1$.
 - Jika $\text{array}[\text{mid}] > \text{target}$, perbarui batas akhir: $\text{high} = \text{mid} - 1$.
4. Ulangi langkah 2 dan 3 selama $\text{low} \leq \text{high}$.
5. Jika perulangan selesai dan elemen tidak ditemukan, kembalikan nilai yang mengindikasikan kegagalan (misalnya -1).

3.2. Analisis Kompleksitas

Kompleksitas algoritma diukur berdasarkan waktu eksekusi (*Time Complexity*) dan penggunaan memori (*Space Complexity*).

Kompleksitas Waktu (Time Complexity):

- **Kasus Terbaik (Best Case):** $O(1)$. Terjadi ketika elemen yang dicari tepat berada di tengah array pada iterasi pertama.
- **Kasus Rata-rata (Average Case):** $O(\log n)$. Secara rata-rata, ruang pencarian dibagi dua pada setiap langkah.
- **Kasus Terburuk (Worst Case):** $O(\log n)$. Terjadi ketika elemen berada di ujung array atau tidak terdapat dalam array sama sekali [7].

Kompleksitas Ruang (Space Complexity):

- **Implementasi Iteratif:** $O(1)$. Hanya membutuhkan beberapa variabel tambahan untuk menyimpan indeks batas dan tengah.
- **Implementasi Rekursif:** $O(\log n)$. Membutuhkan ruang tambahan pada *call stack* memori untuk setiap pemanggilan fungsi rekursif [4].

Tabel Perbandingan Kinerja

Tabel berikut merangkum perbandingan karakteristik antara *Binary Search* dan *Sequential Search*.

Tabel 1: Perbandingan		
Karakteristik	Binary Search	Sequential Search
Prasyarat Data	Harus terurut (<i>sorted</i>)	Tidak perlu terurut
Pendekatan	<i>Divide and Conquer</i>	Pengecekan berurutan
Kompleksitas Waktu (Worst Case)	$O(\log n)$	$O(n)$
Kompleksitas Waktu (Best Case)	$O(1)$	$O(1)$
Efisiensi untuk Data Besar	Sangat Tinggi	Rendah
Kompleksitas Implementasi	Sedang	Sangat Sederhana

4. Implementasi dan Penerapan Praktis

Algoritma *Binary Search* telah diimplementasikan secara luas dalam berbagai sistem perangkat lunak untuk mengoptimalkan proses pencarian. Berdasarkan tinjauan literatur, berikut adalah beberapa contoh penerapannya:

1. **Aplikasi Kamus Digital:** Pencarian kata dalam kamus digital sangat diuntungkan oleh *Binary Search* karena entri kamus secara alami terurut berdasarkan alfabet. Penggunaan algoritma ini memastikan respons pencarian yang instan meskipun kamus memuat ratusan ribu kata [5]. Penelitian oleh Religia (2025) dan Fitriyani (2026) menunjukkan bahwa *Binary Search* mampu mempercepat proses pencarian kata secara signifikan dalam aplikasi kamus [2], [9].
2. **Sistem Informasi Manajemen:** Dalam sistem seperti pencarian data jemaat gereja atau data potensi investasi daerah, *Binary Search* digunakan untuk mempercepat penemuan rekam data spesifik berdasarkan ID atau nama yang telah diindeks dan diurutkan [8]. Sebuah studi kasus pada sistem *e-tracking* pengajuan surat menunjukkan bahwa *Binary Search* memberikan waktu pencarian rata-rata yang lebih cepat dibandingkan *Sequential Search* untuk data yang terurut [11].
3. **Sistem Supply Barang:** Penerapan pada sistem inventaris memungkinkan pencarian kode barang secara efisien, mengurangi waktu tunggu pengguna saat melakukan pengecekan stok [9].
4. **Big Data:** Meskipun *Binary Search* umumnya diterapkan pada data yang terurut, penelitian juga mengeksplorasi penerapannya pada *big data* dengan analisis kinerja yang menunjukkan efisiensi yang baik setelah data diurutkan [12].

Studi eksperimental telah mengukur perbedaan kinerja secara kuantitatif. Sebagai contoh, sebuah penelitian yang melakukan 50 kali pengujian dari 500 data menunjukkan bahwa *Binary Search* menghasilkan waktu rata-rata 0.0402524 ms, sementara *Sequential Search* membutuhkan 0.0519893 ms [13]. Penelitian lain yang membandingkan kedua algoritma pada 507 data dengan 100 pencarian menemukan bahwa *Sequential Search* memiliki waktu rata-rata 5.35 milidetik dan *Binary Search* 6.16 milidetik, menunjukkan bahwa untuk kasus tertentu dengan data yang tidak terlalu besar dan mungkin tidak selalu teratur, *Sequential Search* bisa lebih efisien [11].

Hal ini menggarisbawahi pentingnya mempertimbangkan karakteristik data dan konteks implementasi.

Penting untuk dicatat bahwa efisiensi *Binary Search* harus dipertimbangkan bersama dengan biaya komputasi untuk mengurutkan data (*sorting*). Jika data sangat dinamis (sering mengalami penambahan atau penghapusan), biaya untuk mempertahankan keterurutan data mungkin melebihi keuntungan dari pencarian yang cepat. Dalam kasus seperti ini, struktur data alternatif seperti *Hash Table* atau *Binary Search Tree* (BST) mungkin lebih sesuai.

5. Kesimpulan

Algoritma *Binary Search* merupakan metode pencarian yang sangat efisien dengan kompleksitas waktu logaritmik $O(\log n)$, menjadikannya pilihan utama untuk pencarian pada himpunan data berskala besar. Keunggulannya terletak pada kemampuannya mereduksi ruang pencarian secara eksponensial melalui pendekatan *divide and conquer*. Meskipun mensyaratkan data yang telah teratur, kinerja *Binary Search* jauh melampaui *Sequential Search* dalam sebagian besar skenario praktis, seperti pada aplikasi kamus digital dan sistem basis data. Pemilihan antara *Binary Search* dan algoritma lain harus didasarkan pada karakteristik data, terutama tingkat kedinamisan dan ukuran himpunan data yang dikelola.

Referensi

- [1] R. N. M. Putri, "Implementasi dan Analisis Algoritma Binary Search," *Universitas Islam Negeri Maulana Malik Ibrahim Malang*, 2025. [Online]. Available: <https://urj.uin-malang.ac.id/index.php/mij/article/view/14003>
- [2] Y. Religia, "Analisis Algoritma Sequential Search Dan Binary Search," *Pelita Teknologi: Jurnal Ilmiah Informatika, Arsitektur dan Lingkungan*, 2025. [Online]. Available: <https://jurnal.mdp.ac.id/index.php/msc/article/view/11237>
- [3] Wikipedia, "Binary search," [Online]. Available: https://en.wikipedia.org/wiki/Binary_search
- [4] GeeksforGeeks, "Binary Search," 2026. [Online]. Available: <https://www.geeksforgeeks.org/dsa/binary-search/>
- [5] D. Markuci et al., "Analisis Perbandingan Penggunaan Algoritma Sequential Search dan Binary Search pada Aplikasi SuratPerjalanan Dinas," *JATI (Jurnal Mahasiswa Teknik Informatika)*. [Online]. Available: <https://ejournal.itn.ac.id/index.php/jati/article/download/4569/3037>
- [6] "Sequential Search vs Binary Search Comparison: A Study of Search Algorithms," *SSRN*, 2023. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4451861
- [7] Stack Overflow, "Time Complexity of Binary Search," 2021. [Online]. Available: <https://stackoverflow.com/questions/69931471/timecomplexity-of-binary-search>
- [8] "Penerapan Algoritma Binary Search Dalam Pencarian Data Potensi Investasi Di Kabupaten Seluma Dengan Smartphone," *Neliti*. [Online]. Available: <https://www.neliti.com/publications/439366/penerpan-algoritma-binary-search-dalam-pencarian-data-potensiinvestasi-di-kabup>
- [9] A. Fitriyani, "Penerapan Algoritma Binary Search Pencarian Data Pada Sistem Supply Barang," *Jurnal Sistem Informasi Universitas Suryadarma*. [Online]. Available: <https://journal.universitassuryadarma.ac.id/index.php/jsi/article/view/1743>
- [10] G. B. Balogun, "A Comparative Analysis of the Efficiencies of Binary and Linear Search Algorithms," *African Journal of Computing & ICT*, vol. 13, no. 1, pp. 10-15, 2020. [Online]. Available: <https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=20061781&AN=152880323&h=OjMSGkSg5384lvW9U%2Ft3iU1%2FmX2eLPg>
- [11] "Perbandingan Algoritma Sequential Search dan Binary Search Pada Website E-Tracking Pengajuan Surat," *Semantic Scholar*. [Online]. Available: <https://pdfs.semanticscholar.org/de6a/e895fa418adf5dcc362a944a735c2f8d805d.pdf>
- [12] "Analisis Algoritma Sequential Search dan Binary Search pada Big Data," *Academia.edu*. [Online]. Available: https://www.academia.edu/16000508/ANALISIS_ALGORITMA_SEQUENTIAL_SEARCH_DAN_BINARY_SEARCH_PADA_BIG_DATA
- [13] "Perbandingan Algoritma Binary Search dan Sequential Search pada Aplikasi Pencarian Data," *Journal LPPPM Unindra*. [Online]. Available: <https://www.journal.lppmunindra.ac.id/index.php/STRING/article/view/16475>