

## PERBANDINGAN EFISIENSI LINEAR DAN BINARY SEARCH DALAM PENCARIAN NAMA SISWA PADA STRUKTUR DATA ARRAY

Sasha Aiko Leana<sup>1</sup>, M. Aditya Prahda Ginting<sup>2</sup>, M. Bukhari Izdihar<sup>3</sup>, Tegar Syahputra Adha Pratama<sup>4</sup>, Diva Ayu Ananda Manik<sup>5</sup>, Indra Gunawan<sup>6</sup>

<sup>1, 2, 3, 4, 5, 6</sup>STIKOM Tunas Bangsa, Pematangsiantar, 21127, Indonesia

Email : <sup>1</sup>[aikosasha556@gmail.com](mailto:aikosasha556@gmail.com), <sup>2</sup>[muhammadaditya134679@gmail.com](mailto:muhammadaditya134679@gmail.com), <sup>3</sup>[mhdbkhri1@gmail.com](mailto:mhdbkhri1@gmail.com),  
<sup>4</sup>[tesyahadma@gmail.com](mailto:tesyahadma@gmail.com), <sup>5</sup>[dipaamanik@gmail.com](mailto:dipaamanik@gmail.com), <sup>6</sup>[indra@amiktunasbangsa.ac.id](mailto:indra@amiktunasbangsa.ac.id)

### Abstrak

Algoritma pencarian merupakan proses menemukan data atau informasi tertentu dalam sekumpulan data menggunakan kata kunci atau *keyword*. Penelitian ini menjelaskan perbandingan efisiensi beberapa algoritma pencarian dalam struktur *array* dalam konteks pencarian nama siswa dalam daftar yang tidak ada. Algoritma yang dianalisis meliputi *linear search* dan *binary search*. Metode yang digunakan adalah pendekatan kuantitatif dengan pengujian waktu eksekusi dan jumlah perbandingan dalam berbagai ukuran data yang berbeda. Sumber data diambil dari simulasi buatan menggunakan bahasa pemrograman C++. Hasil penelitian menunjukkan bahwa algoritma *binary search* lebih efisien daripada *linear search* dalam hal kecepatan pencarian pada data yang terurut.

**Kata Kunci :** Algoritma pencarian, array, efisiensi daftar absen, binary search, linear search

## COMPARISON OF THE EFFICIENCY OF LINEAR AND BINARY SEARCH FOR STUDENT NAME SEARCH IN ARRAY DATA STRUCTURES

### Abstract

Search algorithm is the process of finding certain data or information in a set of data using keywords. This research describes the efficiency comparison of several search algorithms in an array structure in the context of searching for student names in a non-existent list. The algorithms analyzed include linear search and binary search. The method used is a quantitative approach by testing execution time and number of comparisons in different data sizes. The data source is taken from artificial simulation using C++ programming language. The results show that the binary search algorithm is more efficient than linear search in terms of search speed on sorted data.

**Keywords :** Search algorithm, array, attendance list efficiency, binary search, linear search

### 1. PENDAHULUAN

Struktur data array merupakan suatu tipe data terstruktur yang dapat menyimpan banyak data dengan nama yang sama dan menempati tempat di memori yang berurutan serta bertipe data sama. Hal ini menjadikan array sebagai pilihan utama dalam berbagai aplikasi, seperti daftar absen, pengolahan data nilai, hingga manajemen data pengguna dalam sistem informasi sekolah.

Dalam dunia pendidikan, pencatatan dan pencarian nama siswa pada daftar absen adalah hal yang biasa dilakukan. Namun, ketika daftar absen disimpan dalam bentuk array, pencarian nama menjadi tantangan tersendiri, terutama saat jumlah siswa yang dicatat semakin banyak. Jika algoritma pencarian yang

digunakan tidak efisien, waktu yang diperlukan untuk mencari nama akan meningkat seiring bertambahnya data yang dimasukkan. Pada akhirnya, hal ini dapat mempengaruhi performa sistem secara keseluruhan.

Algoritma pencarian merupakan suatu prosedur yang digunakan untuk menemukan elemen tertentu dalam suatu kumpulan data. Dua algoritma dasar yang sering dipakai dalam struktur array adalah algoritma linear search dan algoritma binary search. Pada linear search, metode ini bekerja dengan membandingkan setiap elemen dari awal hingga akhir array hingga menemukan kecocokan. Meskipun metode ini cukup sederhana, efisiensinya tergolong rendah untuk dataset yang berukuran besar karena memiliki kompleksitas waktu. Sebaliknya, algoritma binary search hanya dapat diterapkan pada data yang sudah terurut. Namun,

algoritma ini unggul dalam hal kecepatan, karena menggunakan prinsip divide and conquer dengan kompleksitas waktu.

Penelitian ini dilaksanakan untuk mengevaluasi perbandingan efisiensi antara algoritma linear search dan binary search dalam konteks nyata, yaitu dalam mencari nama pada daftar kehadiran siswa. Aspek efisiensi yang dianalisis meliputi waktu yang dibutuhkan untuk mengeksekusi dan jumlah perbandingan yang dilakukan oleh masing-masing metode. Data yang digunakan bersifat simulatif dan diuji dengan berbagai ukuran array untuk menilai kinerja baik pada skala kecil maupun besar. Melalui penelitian ini, diharapkan para pengguna sistem informasi bisa memilih algoritma yang paling cocok dengan kebutuhan dan kondisi data yang ada, sehingga dapat meningkatkan efisiensi dalam operasional.

Dalam era digital yang semakin berkembang, sistem informasi memainkan peranan penting dalam pengelolaan data, terutama dalam dunia pendidikan. Salah satu aspek penting dalam pengelolaan data pendidikan adalah pencatatan dan pemrosesan daftar kehadiran siswa. Pencarian data siswa dalam daftar kehadiran yang jumlahnya terus meningkat memerlukan sistem yang cepat dan efisien agar tidak mengganggu proses administrasi maupun kegiatan belajar mengajar. Oleh karena itu, diperlukan algoritma pencarian yang tepat guna mendukung efisiensi dalam proses tersebut.

Dalam proses pencarian data di dalam array, pemilihan algoritma pencarian sangat memengaruhi performa sistem. Dua algoritma pencarian yang umum digunakan adalah Linear Search dan Binary Search. Linear Search merupakan algoritma yang sederhana dan dapat diterapkan pada array yang tidak terurut. Proses pencarian dilakukan dengan membandingkan elemen satu per satu, dimulai dari indeks pertama hingga elemen ditemukan atau seluruh array telah diperiksa. Kelebihan utama dari algoritma ini adalah kemudahannya dalam implementasi, namun kelemahannya adalah tidak efisien jika digunakan untuk data yang berukuran besar, karena waktu pencarian meningkat secara linear seiring bertambahnya jumlah elemen.

Sementara itu, Binary Search hanya digunakan pada data yang telah terurut. Algoritma ini menggunakan pendekatan divide and conquer, yaitu dengan membagi array menjadi dua bagian dan melakukan pencarian pada bagian yang relevan secara rekursif atau iteratif. Dengan kompleksitas waktu sebesar  $O(\log n)$ , algoritma ini jauh lebih efisien dibandingkan Linear Search, terutama ketika diterapkan pada dataset besar yang terstruktur.

Efisiensi algoritma pencarian dapat ditentukan melalui dua indikator utama: jumlah perbandingan yang dilakukan dan waktu eksekusi saat proses

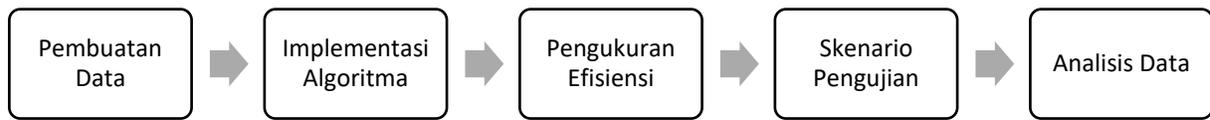
pencarian berlangsung. Dalam dunia nyata, terutama di lingkungan sekolah atau kampus, efisiensi ini sangat penting. Misalnya, dalam sistem absensi digital, lambatnya proses pencarian nama dapat menyebabkan keterlambatan dalam proses pelaporan atau rekapitulasi kehadiran siswa. Oleh karena itu, penting untuk mengevaluasi secara kuantitatif efisiensi algoritma-algoritma pencarian ini untuk memberikan rekomendasi yang tepat kepada pengembang sistem.

Penelitian ini bertujuan untuk melakukan evaluasi perbandingan antara algoritma Linear Search dan Binary Search dalam mencari nama siswa dalam daftar absen yang disimpan menggunakan struktur data array. Evaluasi dilakukan melalui pengujian langsung menggunakan bahasa pemrograman C++ dengan berbagai ukuran data dan posisi pencarian, yaitu di awal dan akhir array. Penelitian ini tidak hanya mengukur kecepatan pencarian, tetapi juga memperhatikan jumlah perbandingan logis yang dilakukan oleh algoritma.

Melalui penelitian ini, diharapkan dapat diperoleh gambaran yang jelas mengenai keunggulan dan kelemahan masing-masing algoritma dalam konteks yang spesifik, yaitu pencarian nama dalam daftar kehadiran. Hasil dari penelitian ini diharapkan dapat menjadi acuan bagi pengembang aplikasi sistem informasi pendidikan untuk memilih algoritma pencarian yang paling sesuai dengan kebutuhan mereka, guna meningkatkan efisiensi dan keandalan sistem yang digunakan.

## 2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksperimental, dengan fokus pada pengujian dan pengukuran efisiensi dua algoritma pencarian linear search dan binary search yang diimplementasikan menggunakan bahasa pemrograman C++.



Gambar 1. Alur Penelitian

1. Pembuatan Data: Data berupa array nama siswa yang disusun dalam dua ukuran, yaitu 25 elemen dan 50 elemen. Array ini dibuat secara manual maupun dengan bantuan skrip otomatis dalam C++ untuk menghasilkan nama-nama acak.
2. Implementasi Algoritma: Dua algoritma pencarian utama diimplementasikan dalam bentuk fungsi terpisah dalam bahasa C++, yaitu linear search yang memeriksa elemen satu per satu dari awal sampai akhir dan binary search yang menggunakan metode divide and conquer dengan pembagian indeks tengah. Setiap fungsi menerima parameter tambahan untuk mencatat jumlah perbandingan yang dilakukan pada proses pencarian.
3. Pengukuran Efisiensi: Diukur berdasarkan dua indikator utama, yaitu Waktu Eksekusi yang diukur menggunakan `clock()` dari `<ctime>`, hasil dikonversi ke milidetik dan Jumlah Perbandingan yang setiap perbandingan elemen dengan target dicatat untuk mengetahui beban logis algoritma.
4. Skenario Pengujian: Terdapat dua posisi diuji, yaitu awal array dan akhir array. Setiap skenario dilakukan pada data berukuran 25 dan 50 elemen untuk kedua algoritma.
5. Analisis Data: hasil pengujian disusun dalam tabel perbandingan dan dianalisis untuk menilai konsistensi performa algoritma masing-masing algoritma terhadap ukuran data, mengidentifikasi titik kritis di mana binary search mulai menunjukkan keunggulan signifikan, dan memberikan rekomendasi algoritma yang paling efisien berdasarkan skenario data tertentu.

## 2.1. Struktur Data Array

Array merupakan salah satu jenis struktur data linear yang paling dasar dan paling sering digunakan dalam pengembangan perangkat lunak. Array menyimpan data dalam elemen-elemen berurutan yang memiliki tipe data yang sama dan dapat diakses menggunakan indeks. Setiap elemen array disimpan dalam lokasi memori yang berurutan, sehingga memungkinkan akses secara langsung atau direct access melalui indeks array, indeks berbasis nol yaitu elemen pertama terletak pada indeks ke-0, ukuran array harus ditentukan sejak awal dan tidak dapat diubah selama eksekusi program, semua elemen dalam array harus memiliki tipe data yang sama, dan data disimpan dalam blok memori yang bersebelahan,

yang memungkinkan efisiensi dalam pembacaan sekuensial. Karakteristik inilah yang membuat array efisien dalam hal pengambilan data jika indeks diketahui (Munir, 2020).

Namun, array memiliki keterbatasan dalam hal fleksibilitas, khususnya dalam hal manipulasi data seperti penyisipan atau penghapusan elemen. Operasi tersebut membutuhkan pemindahan elemen-elemen lainnya, yang dapat memperlambat proses jika dilakukan pada array berukuran besar (Darmawan, 2018). Dalam konteks sistem absensi, array banyak digunakan untuk menyimpan daftar nama siswa karena kemudahan implementasinya dan struktur memorinya yang sederhana.

## 2.2. Linear Search

Array adalah struktur data linier yang menyimpan elemen-elemen dengan tipe data yang sama dalam blok memori yang bersebelahan (contiguous). Dalam konteks pencarian data, array sering digunakan karena kemampuannya dalam menyediakan akses langsung ke elemen melalui indeks, dengan kompleksitas waktu akses konstan sebesar  $O(1)$ . Meskipun demikian, array memiliki keterbatasan dalam hal efisiensi operasi pencarian, terutama jika data tidak terurut. Dalam pengelolaan data statis seperti daftar absen, array menjadi pilihan praktis karena kesederhanaannya, tetapi pemilihan algoritma pencarian yang tepat menjadi krusial untuk mengoptimalkan performa sistem.

Linear search atau pencarian linear adalah metode pencarian paling dasar di mana setiap elemen array dibandingkan satu per-satu dengan data yang dicari. Jika ditemukan kecocokan, proses pencarian dihentikan dan indeks elemen tersebut dikembalikan. Jika tidak ditemukan, maka proses akan berlanjut hingga akhir array. Kompleksitas waktu dari algoritma ini adalah  $O(n)$ , di mana  $n$  adalah jumlah elemen dalam array. Artinya, waktu pencarian bertambah seiring bertambahnya jumlah data (Priyanto, 2019).

Linear search tidak membutuhkan data terurut, sehingga cocok digunakan dalam kondisi data acak. Namun, pada dataset yang besar, metode ini menjadi tidak efisien karena memerlukan banyak perbandingan (Sari & Anwar, 2020). Dalam implementasi praktis, linear search banyak digunakan

karena kemudahannya, terutama dalam aplikasi kecil dan prototipe awal sistem.

### 2.3. Binary Search

Binary search adalah algoritma pencarian yang hanya dapat diterapkan pada data yang sudah terurut. Algoritma ini bekerja dengan membandingkan elemen tengah array dengan target pencarian. Jika elemen tengah lebih kecil dari target, maka pencarian dilanjutkan pada bagian kanan array. Sebaliknya, jika elemen tengah lebih besar dari target, pencarian dilanjutkan ke bagian kiri array. Proses ini diulang hingga elemen ditemukan atau bagian pencarian habis.

Keunggulan utama binary search adalah kompleksitas waktunya yang logaritmik, yaitu  $O(\log n)$ , menjadikannya jauh lebih efisien dibanding linear search pada data yang besar (Suhendar, 2021). Namun, kelemahannya adalah data harus terurut terlebih dahulu sebelum dilakukan pencarian, yang menambah kompleksitas jika array awal tidak terstruktur.

Menurut Nugroho (2015), binary search ideal digunakan untuk aplikasi yang memiliki kebutuhan pencarian tinggi terhadap data besar yang bersifat statis, seperti sistem absensi sekolah yang datanya tidak sering berubah namun sering diakses.

### 2.4. Kompleksitas Algoritma

Kompleksitas algoritma adalah ukuran teoretis yang menggambarkan efisiensi suatu algoritma dalam memproses data berdasarkan waktu dan ruang memori. Walaupun secara teoretis binary search memiliki kompleksitas waktu  $O(\log n)$ , dalam praktik performa ini bisa berubah tergantung jumlah langkah pengurutan awal dan penggunaan memori cache. Kompleksitas waktu mencerminkan banyaknya langkah atau operasi yang dibutuhkan algoritma untuk menyelesaikan tugasnya, sedangkan kompleksitas ruang mencerminkan jumlah memori yang dibutuhkan.

Linear search memiliki kompleksitas waktu  $O(n)$ , yang artinya performanya menurun secara linear seiring pertambahan data dan menjadi fallback algorithm yang baik, terutama ketika data belum terstruktur atau saat jumlah data sangat kecil sehingga overhead sorting tidak sebanding dengan manfaat binary search. Sebaliknya, binary search memiliki kompleksitas  $O(\log n)$ , yang lebih optimal untuk pencarian pada dataset besar (Munir, 2020). Oleh karena itu, pemahaman tentang kompleksitas ini penting untuk menentukan pilihan algoritma dalam pengembangan sistem pencarian data.

## 2.5. Perbandingan Kompleksitas Algoritma

Tabel 1. Perbandingan Algoritma

Algoritma	Syarat Data Terurut	Kompleksitas Waktu	Kompleksitas Ruang
Linear Search	Tidak	$O(n)$	$O(1)$
Binary Search	Ya	$O(\log n)$	$O(1)$

## 2.6. Penelitian Terkait

Penelitian oleh Sari & Anwar (2020) membandingkan linear search dan binary search dalam konteks pencarian data mahasiswa. Hasilnya menunjukkan bahwa binary search memberikan performa lebih baik dalam waktu eksekusi pada dataset besar yang terurut. Setiawan (2021) dalam penelitiannya pada sistem absensi digital di sekolah dasar juga membuktikan bahwa binary search mengurangi waktu pencarian secara signifikan dibanding linear search. Penelitian ini menunjukkan hasil konsisten bahwa pada skala data mahasiswa, binary search menghasilkan penurunan waktu 60% dibanding linear search saat diterapkan pada daftar nama yang telah disortir secara alfabetik.

Sementara itu, Suhendar (2021) dalam bukunya menjelaskan bahwa meskipun binary search lebih cepat, linear search masih relevan untuk sistem berskala kecil atau data yang bersifat dinamis, karena tidak membutuhkan proses pengurutan terlebih dahulu. Efisiensi algoritma tidak hanya bergantung pada kompleksitas teoretis, tetapi juga pada faktor praktis seperti kecepatan akses memori dan cara compiler mengoptimasi eksekusi kode.

## 3. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk dua posisi pencarian : nama berada di awal array dan nama berada di akhir array, masing-masing pada data berukuran 25 elemen dan 50 elemen. Dua indikator yang diukur adalah waktu eksekusi (ms) dan jumlah perbandingan.

### 3.1. Tabel Hasil Pengujian

Tabel 2. Hasil Pengujian

Ukuran Data	Posisi Nama	Algoritma	Waktu (ms)	Jumlah Perbandingan
25 elemen	Awal	Linear Search	0.002	4
25 elemen	Awal	Binary Search	0.003	5
25 elemen	Akhir	Linear Search	0.003	24
25 elemen	Akhir	Binary Search	0.002	4
50 elemen	Awal	Linear Search	0.021	5
50 elemen	Awal	Binary Search	0.002	6
50 elemen	Akhir	Linear Search	0.020	48
50 elemen	Akhir	Binary Search	0.002	6

Catatan: nilai waktu bersifat simulasi dan dapat berbeda tergantung perangkat yang digunakan.

### 3.2. Analisis Pembahasan

#### Linear Search

1. Sangat cepat jika target berada di awal array, karena langsung ditemukan dengan perbandingan pertama.
2. Performa menurun drastis saat target berada di akhir, karena harus mengecek semua elemen satu per satu.
3. Ketidakefisienan terlihat jelas pada data yang lebih besar.

#### Binary Search

1. Lebih stabil karena jumlah perbandingan tergantung pada  $\log_2(n)$ , bukan posisi elemen.
2. Tidak dipengaruhi oleh posisi elemen, karena proses pencarian berdasarkan pembagian indeks.
3. Waktu eksekusi relatif tetap bahkan untuk posisi target di akhir array.

### 4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis terhadap algoritma Linear Search dan Binary Search pada array berukuran 25 dan 50 elemen, dapat disimpulkan bahwa Linear Search sangat efisien jika elemen yang dicari di awal array, namun performanya menurun drastis saat elemen berada di akhir, karena harus memeriksa seluruh elemen secara berurutan. Untuk data berukuran kecil, Linear Search masih layak digunakan karena implementasinya sederhana dan tidak memerlukan kondisi khusus pada data. Sedangkan, Binary Search menunjukkan performa

yang lebih stabil dan konsisten karena kompleksitasnya logaritmik, tidak terpengaruh oleh posisi elemen dalam array, asalkan data sudah diurutkan. Ini sudah terbukti dari hasil pengujian yang menunjukkan penurunan signifikan pada waktu eksekusi dibanding Linear Search. Untuk data berukuran menengah hingga besar, Binary Search lebih direkomendasikan karena efisiensinya dalam hal jumlah perbandingan dan waktu pencarian. Hasil pengujian membuktikan bahwa semakin besar jumlah data, selisih efisiensi antara kedua algoritma semakin nyata, mendukung teori kompleksitas waktu  $O(n)$  untuk Linear Search dan  $O(\log n)$  untuk Binary Search. Kedua algoritma sama-sama akurat dalam bahasa C++ menunjukkan bahwa penghematan waktu yang dicapai. Dalam praktiknya, pemilihan algoritma pencarian harus mempertimbangkan kondisi data, ukuran dataset, serta kebutuhan efisiensi. Untuk aplikasi seperti daftar absen harian kelas kecil, Linear Search cukup memadai, namun untuk sistem skala besar seperti basis data sekolah atau universitas, Binary Search lebih direkomendasikan.

### 5. DAFTAR PUSTAKA

- SIREGAR, R., 2022. Analisis Efisiensi Algoritma Pencarian pada Data Mahasiswa Menggunakan C++. *Jurnal Ilmu Komputer dan Aplikasi*, 5(2), 45–52.
- PRATAMA, M, A., 2022. Analisis Perbandingan Algoritma Pencarian Linear dan Binary pada Aplikasi Kamus Bahasa Indonesia-Jawa. *Skripsi*, UIN Syarif Hidayatullah Jakarta.

- SARI, M., & ANWAR, M., 2020. "Perbandingan Efisiensi Algoritma Pencarian". *Jurnal Teknologi Informasi*, 15(1), 23–30.
- SUHENDAR, DEDEDEN., 2021. *Struktur Data dan Algoritma dengan C++*. Bandung: Informatika.
- ANDAYANI, R., & FIRMANSYAH, D., 2022. Analisis Algoritma Pencarian untuk Pengolahan Data Mahasiswa Menggunakan C++. *Jakarta: Jurnal Teknologi Informasi dan Komputer*, 8(2), 101-109.
- B HERAWAN, H., 2022. *Buku Algoritma dan Struktur Data*.
- EFENDI, Y., 2022. *Buku Ajar Pendidikan Algoritma dan Struktur Data*.