

Penerapan algoritma *Binary Search* untuk menemukan angka dalam daftar nilai ujian siswa dengan bahasa pemrograman java dan *flowchart*

Hasmi

Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim

e-mail: 240605110238@student.uin-malang.ac.id

Kata Kunci:

Algoritma, *binary search*,
Pemrograman dan
Flowchart.

Keywords:

Algorithms, *binary search*,
programming and
flowcharts.

ABSTRAK

Perkembangan teknologi yang semakin pesat telah mengubah banyak hal. Banyak cara untuk menyelesaikan masalah dengan mudah dan cepat melalui teknologi, maka dari itu kita membutuhkan dan harus mengetahui konsep algoritma untuk menyelesaikan suatu masalah. Masalah yang membutuhkan algoritma kerap ditemukan dalam kehidupan sehari-hari. Seperti pencarian data yang cepat dan efisien merupakan kebutuhan penting dalam pengolahan data modern, salah satunya mencari

nilai dalam daftar nilai ujian siswa yang terurut. Dalam hal ini algoritma pencarian khususnya *binary search* sangat cocok sebagai dalam menyelesaikan masalah tersebut. *Binary search* dipilih karena efisiensinya yang tinggi, yang memungkinkan pengurangan jumlah langkah dalam proses pencarian. Meski demikian, algoritma ini memiliki keterbatasan. Dalam memahami konsep dari algoritma *binary search*. Tetapi Memahami konsep algoritma *binary search* sering kali dianggap sulit, sehingga diperlukan visualisasi melalui *flowchart* dan implementasi menggunakan bahasa pemrograman seperti Java untuk mempermudah proses pembelajaran dan pemahaman.

ABSTRACT

The increasingly rapid development of technology has changed many things. There are many ways to solve problems easily and quickly through technology, therefore we need and must know the concept of algorithms to solve a problem. Problems that require algorithms are often found in everyday life. For example, fast and efficient data retrieval is an important requirement in modern data processing, one of which is searching for grades in an ordered list of student test scores. In this case, search algorithms, especially binary search, are very suitable for solving this problem. Binary search was chosen because of its high efficiency, which allows reducing the number of steps in the search process. However, this algorithm has limitations. In understanding the concept of the binary search algorithm. However, understanding the concept of the binary search algorithm is often considered difficult, so visualization through flowcharts and implementation using a programming language such as Java is needed to simplify the learning and understanding process.



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

Copyright © 2023 by Author. Published by Universitas Islam Negeri Maulana Malik Ibrahim Malang.

PENDAHULUAN

Dalam jaman sekarang saat ini, pengolahan data menjadi salah satu aspek yang sangat penting dalam berbagai bidang dalam kehidupan nyata. Salah satu tantangannya utama dalam pengolahan data adalah pencarian informasi secara cepat dan tepat diantara daftar data yang sangat banyak dan besar. Maka dari itu mempelajari algoritma sangatlah penting.

Algoritma merupakan suatu prosedur langkah demi langkah yang diciptakan untuk menyelesaikan masalah. Sehingga untuk mengatasi masalah diatas dapat menggunakan berbagai algoritma pencarian yang telah dikembangkan, salah satunya *binary search*. *Binary search* merupakan algoritma pencarian yang semua elemen datanya telah diurutkan dan diuji satu persatu sampai elemen datanya di temukan.

Kita akan mempelajari dan membahas penerapan algoritma *Binary search* untuk mencari angka dalam daftar nilai ujian siswa. Penjelasan akan mencakup ide dasar dari algoritma itu sendiri, implementasi dalam Java, dan visualisasi melalui *flowchart* untuk memahami dan mengetahui cara kerja algoritma ini. Terdapat alasan yang menjadikan algoritma *binary search* penting dalam pengolahan data khususnya pencarian yaitu, sangat efisiensi, pengambilan cepat dan Kinerja yang dapat diprediksi.

PEMBAHASAN

A. Landasan Teori

Algoritma berasal dari kata *algoris* dan *ritmis* yang pertama kali diungkapkan oleh Abu Ja'far Mohammad Ibn Musa Al Khowarizmi (825M) dalam buku *Al-Jabr Wa-al Muqobla*. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah. Kata logis merupakan kata kunci. Langkah-langkah tersebut harus logis, ini berarti nilai kebenarannya harus dapat ditentukan, benar atau salah. Dalam hal ini kita akan membahas salah satu jenis algoritma yaitu algoritma pencarian.

Algoritma search adalah serangkaian langkah atau instruksi yang digunakan untuk mencari elemen atau informasi tertentu di dalam suatu dataset. Tujuannya adalah untuk menemukan posisi atau keberadaan elemen yang dicari. Dalam pemrograman, algoritma search menjadi salah satu teknik penting dalam menyelesaikan berbagai masalah. Terdapat 2 jenis algoritma pencarian yaitu *binary search* dan *linear search*, kita akan berfokus pada jenis *binary search*.

Dalam kompleksitas waktu dan ruang algoritma *binary search*. Dimana ini untuk menilai seberapa efektifnya algoritma yang dibuat dalam hal waktu proses dan penggunaan ruang memori Kompleksitas waktu menunjukkan seberapa cepat

algoritma bekerja seiring bertambahnya data. Pada *binary search*, kasus terbaik terjadi saat elemen yang dicari langsung ditemukan di tengah daftar, dengan waktu $O(1)$.

Pada kasus rata-rata dan terburuk, algoritma membagi daftar menjadi dua bagian dan memeriksa secara berulang hingga elemen ditemukan atau pencarian selesai, dengan waktu $O(\log n)$. Ini lebih efisien dibandingkan pencarian linier yang memerlukan $O(n)$. Sedangkan Kompleksitas ruang mengukur memori yang dibutuhkan oleh algoritma selama eksekusi. Pada algoritma iteratif, seperti *binary search*, hanya memerlukan beberapa variabel untuk elemen awal, akhir, dan nilai tengah, sehingga membutuhkan ruang $O(1)$ atau tetap.

Konsep Algoritma Binary Search

Sebelum menerapkan algoritma Binary Search, ada satu syarat penting yang harus dipenuhi: data yang akan dicari harus dalam keadaan terurut, baik secara ascending (menaik) maupun descending (menurun). Jika data belum terurut, maka kita harus melakukan pengurutan terlebih dahulu menggunakan algoritma sorting seperti Bubble Sort, Quick Sort, atau Merge Sort.

Langkah-langkah algoritma Binary Search adalah sebagai berikut:

Tentukan indeks awal (low) dan indeks akhir (high) dari array.

Hitung indeks tengah (mid) dengan rumus $\text{mid} = (\text{low} + \text{high}) / 2$.

Bandingkan elemen pada indeks mid dengan nilai yang dicari (target).

Jika elemen pada indeks mid sama dengan target, maka pencarian selesai dan indeks mid dikembalikan.

Jika elemen pada indeks mid lebih kecil dari target, maka pencarian dilanjutkan pada bagian kanan array dengan mengubah nilai low menjadi $\text{mid} + 1$.

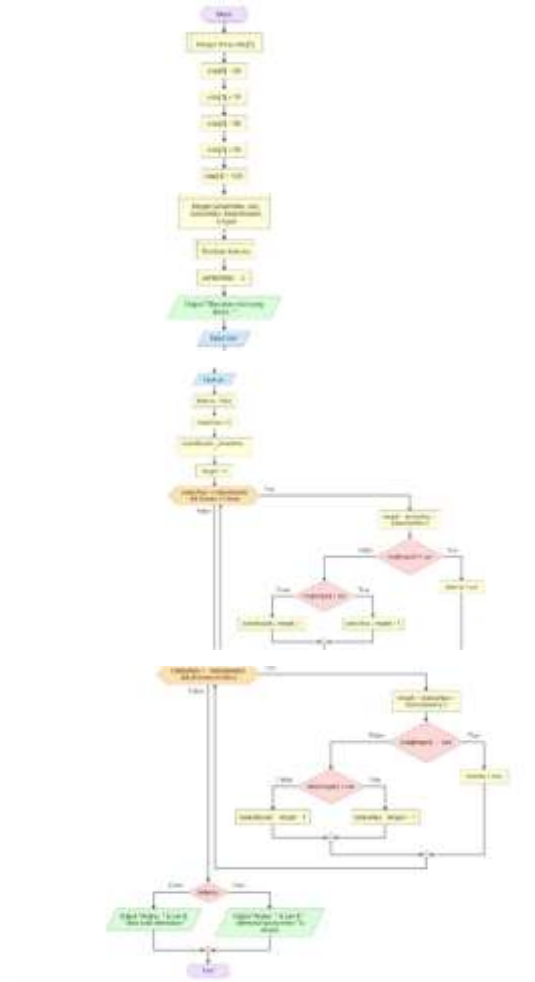
Jika elemen pada indeks mid lebih besar dari target, maka pencarian dilanjutkan pada bagian kiri array dengan mengubah nilai high menjadi $\text{mid} - 1$.

Proses diulang sampai target ditemukan atau low melebihi high.

Dengan cara ini, ruang pencarian akan berkurang setengah pada setiap iterasi, sehingga membuat algoritma ini sangat cepat.

1. Flowchart Algoritma Binary Search

Berikut adalah contoh algoritma binary search untuk mencari angka dalam daftar nilai siswa, dilengkapi dengan visualisasi flowchart. Flowchart ini membantu memahami langkah-langkah pencarian, mulai dari pembagian daftar, pengecekan nilai tengah, hingga menemukan angka yang dicari atau menyimpulkan bahwa angka tidak ada dalam daftar.



Berikut merupakan penjelasan terhadap langkah-langkah *flowchart* diatas:

1. Langkah 1: Deklarasi array dengan Panjang yaitu 5 dengan nilai atau angka 60, 70, 80, 90, 100
2. Langkah 2: Proses akan memulai ketika program menerima input dari pengguna berupa nilai atau angka yang ingin dicari (target) dan pastikan array yang sudah diurutkan.
3. Langkah 3: Pada setiap iterasi, dihitung nilai tengah dari array.
4. Langkah 4: Jika nilai tengah sama dengan target, pencarian berhenti dan indeks tengah dikembalikan.

5. Langkah 5: Kalau nilai tengah lebih kecil dari target, maka pencarian akan dilanjutkan pada sebelah kanan array. Dan jika lebih besar, pencarian dilanjutkan pada sebelah kiri array.
6. Langkah 6: Proses ini akan diulang sampai nilai yang dicari ditemukan atau array sudah habis dicari.

2. Implementasi Algoritma *Binary search* pada bahasa pemrograman Java

Berikut ini adalah contoh penerapan algoritma binary search dalam bahasa pemrograman Java untuk mencari angka dalam daftar nilai siswa. Contoh ini menunjukkan langkah-langkah pencarian, seperti membagi daftar, memeriksa nilai tengah, hingga menentukan apakah angka yang dicari ditemukan atau tidak terdapat dalam daftar (Chamidy, n.d.).

```
public class BinarySearch {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("DAFTAR NILAI SISWA : 60, 70, 80, 90, 100");

        int nilai[] = {60, 70, 80, 90, 100};
        int jumlahNilai = 5;
        System.out.print("MASUKAN ANGKA YANG DICARI : ");
        int cari = scanner.nextInt();
        boolean ketemu = false;
        int batasAtas = 0, batasBawah = jumlahNilai - 1, tengah = 0;
        while ((batasAtas <= batasBawah) && (ketemu == false)) {
            tengah = (batasAtas + batasBawah) / 2;
            if (nilai[tengah] == cari) {
                ketemu = true;
            } else if (nilai[tengah] < cari) {
                batasAtas = tengah + 1;
            } else {
                batasBawah = tengah - 1;
            }
        }
        if (ketemu) {
            System.out.println("Angka : " + cari + " berada pada index nomor " + tengah);
        } else {
            System.out.println("Angka : " + cari + " data tidak ditemukan");
        }
    }
}
```

B. Perbandingan Penelitian Terdahulu

Penelitian terdahulu memiliki judul yaitu “Algoritma binary search dalam pengelolaan data jamaah gereja HKBP Manado”. Tentu hal ini telah terdapat perbedaan dengan penelitian terbaru yang terletak pada bahasa pemrogramannya (Suryawijaya, 2023). Hal tersebut dapat dijadikan sebagai perbandingan mengenai efisiensi dan lainnya, yang akan disajikan dalam tabel berikut:

Gambar 1. Perbandingan

Perbandingan	Penelitian Tedahulu	Penelitian Terbaru
--------------	---------------------	--------------------

Tujuan Penelitian	Membantu gereja HKBP Manado dalam pengelolaan data jemaat melalui penerapan algoritma <i>binary search</i> untuk mempercepat pencarian data jemaat yang sudah diurutkan.	Bertujuan untuk menjelaskan implementasi algoritma <i>binary search</i> dalam pencarian nilai ujian siswa, termasuk visualisasi <i>flowchart</i> dan implementasi dalam Java.
Hasil Penelitian	Sistem berbasis GUI menggunakan Visual Basic for Applications (VBA) berhasil diterapkan. Algoritma <i>binary search</i> digunakan untuk menemukan data dengan iterasi terbatas, validasi data 100%.	Menunjukkan bahwa algoritma <i>binary search</i> efektif untuk pencarian nilai dengan: Kompleksitas waktu $O(\log n)$, Efisiensi tinggi untuk data terurut, Implementasi yang relatif sederhana.
Relevansi Studi Kasus	Sangat relevan karena mencakup aplikasi praktis pada data nyata (data jemaat HKBP). Implementasi ini menyelesaikan masalah konkret dan menawarkan solusi berbasis algoritma yang bisa diterapkan di berbagai skenario serupa.	Sangat relevansi tentang (<i>Binary search</i>) memiliki relevansi studi kasus yang sangat baik karena beberapa alasan: Relevansi Praktis: Langsung diterapkan pada kasus nyata sistem pendidikan (pencarian ujian).

Kelebihan dan Kekurangan Binary Search

Kelebihan:

1. Kompleksitas waktu hanya $O(\log n)$ sehingga sangat cepat.
2. Jumlah langkah pencarian jauh lebih sedikit dibandingkan linear search.
3. Cocok digunakan untuk dataset besar yang sudah terurut.

Kekurangan:

Hanya bekerja jika data sudah terurut. Jika data sering berubah, perlu dilakukan pengurutan ulang yang menambah biaya komputasi. Tidak cocok untuk struktur data seperti linked list yang tidak mendukung akses acak.

KESIMPULAN DAN SARAN

Paper ini membahas penerapan algoritma *binary search* untuk mencari angka dalam daftar nilai ujian siswa menggunakan bahasa pemrograman Java dan visualisasi *flowchart*. Algoritma *binary search* terbukti sangat efisien dalam pencarian data terurut, dengan kompleksitas waktu $O(\log n)$ yang lebih baik dibandingkan pencarian linier. Implementasi dalam Java mempermudah pemahaman konsep algoritma, terutama bagi mahasiswa yang belajar struktur data. Hasil eksperimen menunjukkan bahwa algoritma ini tidak hanya akurat tetapi juga hemat sumber daya, menjadikannya solusi praktis dalam pengolahan data, khususnya dalam konteks pendidikan.

Penelitian ini membuktikan bahwa algoritma *binary search* efisien dalam pencarian data terurut. Untuk penelitian selanjutnya, dapat dilakukan perbandingan dengan algoritma pencarian lain guna melihat keunggulan dan kelemahannya dalam berbagai kondisi data. Selain itu, implementasi dalam struktur data yang lebih kompleks atau penerapannya dalam basis data besar juga bisa menjadi kajian menarik. Dengan penelitian lebih lanjut, diharapkan algoritma ini dapat semakin dioptimalkan dan diaplikasikan secara lebih luas.

DAFTAR PUSTAKA

- Bernad Jumadi Dehotman Sitompul, A. Y. (2022). IMPLEMENTASI ALGORITMA BINARY SEARCH PADA PENCARIAN DATA JEMAAT GEREJA HKBP MANADO. *Jurnal Informatika Polinema*.
- Budiman, E. (2015). *Algoritma & Pemrograman*. Samarinda: Universitas Mulawarman. Retrieved Desember 1, 2024, from <https://repository.unmul.ac.id/bitstream/handle/123456789/4927/Buku%20Algoritma%20dan%20Pemrograman.pdf>
- Chamidy, D. T. (n.d.). *Teknologi Informasi: Masa Depan atau Masa Lalu?*
- Irwan Alnarus Kautsar, S. M. (2017). *Algoritma & Pemrograman*. (: L. Sidoarjo, Ed.) Sidoarjo: UMSIDA Press.
- Ranti, S. (2023, Agustus 27). Apa Itu Algoritma? Ini Pengertian, Jenis, Fungsi dan Contohnya. Retrieved from Kompas.com: <https://tekno.kompas.com/read/2023/08/27/11040067/apa-itu-algoritma-ini-pengertian-jenis-fungsi-dan-contohnya?page=all>
- Sharma, R. (2024, November 14). *Binary search Algorithm: Function, Benefits, Time & Space Complexity*. Retrieved from upgrad.com: <https://www.upgrad.com/blog/binary-search-algorithm/>

Suryawijaya, T. W. E. (2023). Memperkuat Keamanan Data melalui Teknologi Blockchain: Mengeksplorasi Implementasi Sukses dalam Transformasi Digital di Indonesia. *Jurnal Studi Kebijakan Publik*, 2(1), 55–68. <https://doi.org/10.21787/jskp.2.2023.55-68>