

Panduan Karya Tulis

**STRATEGI ALGORITMA
ALGORITMA PENCARIAN**

IF4005

KELAS TI IIIB

Oleh:

HANIFAH ALIFIA

A22100058



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS SEBELAS APRIL
UNSAP SUMEDANG
2022**

1. Pendahuluan

Pencarian (searching) merupakan suatu pekerjaan yang sering dikerjakan dalam kehidupan sehari – hari. Ada kalanya kita mencari sesuatu dengan tujuan hanya untuk mengetahui apakah data tersebut ada dalam sekumpulan data atau tidak, sementara dilain waktu mungkin kita menginginkan posisi dari data yang dicari tersebut.

Dalam Algoritma pencarian (searching,) ada Beberapa metoda pencarian

Seperti:

- Metode linier (Linear Search.) adalah teknik pencarian data dimana data dicari secara urut dari depan ke belakang atau dari awal sampai akhir secara Sequen
- Pencarian biner (Binary Search) adalah sebuah teknik untuk menemukan nilai tertentu dalam sebuah larik (array) , Prinsip pencarian Biner yaitu dengan membagi data atas dua bagian kemudian membandingkannya.
- Pencarian interpolasi (Interpolation Search) adalah menentukan posisi yang diestimasi dari sisa rekaman yang belum diperiksa

Masing – masing algoritma memiliki prasyarat dan cara serta waktu pelaksanaan yang berbeda. Pemilihan atas metoda pencarian dilakukan berdasarkan keadaan dan keinginan pengguna metoda yang biasanya tergantung pada jumlah data, jenis data dan struktur data yang digunakan.

Berdasarkan uraian di atas, penulis bermaksud untuk merancang perangkat sebuah Aplikasi untuk mensimulasikan metoda pencarian linier, biner dan interpolasi dari data - data numerik dan text yang di-input sebagai media pembelajaran bagi mahasiswa.

2. Dasar Teori

Algoritma pencarian (searching algorithm) adalah algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (successful) atau tidak ditemukan (unsuccessful).

Ada dua macam teknik pencarian yaitu pencarian sekuensial (sequential search) dan pencarian biner (binary search). Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak teratur. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

1. PENCARIAN BERURUTAN (SEQUENTIAL SEARCH)

Algoritma pencarian dapat dijelaskan sebagai berikut : pencarian dimulai dari data paling awal, kemudian ditelusuri dengan menaikkan indeks data, apabila data sama dengan kunci pencarian dihentikan dan diberikan nilai pengembalian true, apabila sampai indeks terakhir data tidak ditemukan maka diberikan nilai pengembalian false.

Ilustrasi dari algoritma pencarian biner adalah sebagai berikut :

Kunci=3

i=0	↓	12	35	9	11	3	17	23	15	31	20
i=1	↓	12	35	9	11	3	17	23	15	31	20
i=2	↓	12	35	9	11	3	17	23	15	31	20
i=3	↓	12	35	9	11	3	17	23	15	31	20
i=0	↓	12	35	9	11	3	17	23	15	31	20

Data[4] = 3 sama dengan kunci=3 maka data ditemukan dan diberikan nilai pengembalian i (posisi) dan proses dihentikan. Apabila data tidak ditemukan, maka fungsi akan mengembalikan nilai -1

3. Deskripsi Masalah

Perpustakaan masih banyak yang menggunakan teknik pembukuan secara manual. Hal ini mengakibatkan pencarian data sebuah buku menjadi tidak efisien. Penulis ingin membuat aplikasi yang dapat mendata seluruh buku yang ada di perpustakaan, untuk mempercepat proses pencarian buku.

4. Implementasi

A. Implementasi pencarian dengan metode sequential search

```
*/
public static void main(String[] args) {
    // TODO code application logic here

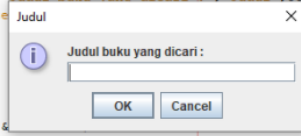
    //Scanner input = new Scanner(System.in);
    JFrame f;
    f=new JFrame();
    String input = JOptionPane.showInputDialog(null,"Judul buku yang dicari :", "Judul",JOptionPane.INFORMATION_MESSAGE);
    String[][] names = {{ "A21", "PKN"}, { "A22", "Matematika"}, { "A23", "Agama"}, { "A24", "Fisika"}, { "A25", "Kimia"}, {"A

    String firstLetter = input;
    boolean found = false;
    for (int i = 0; i < names.length; i++) {
        for (int j = 0; j < names[i].length; j++) {
            if (names[i][j].startsWith(firstLetter) && !found) {
                //System.out.println("Buku ditemukan " + names[i][1]);
                JOptionPane.showMessageDialog(f,"Buku ditemukan "+names[i][1], "Hasil",JOptionPane.INFORMATION_MESSAGE);
                found = true;
                break;
            }
        }
    }
    if (!found) {
        //System.out.println("Buku Tidak Ditemukan");
        JOptionPane.showMessageDialog(f,"Buku tidak ditemukan ", "Hasil",JOptionPane.INFORMATION_MESSAGE);
    }
}
```

```
String[] args) {
    // logic here

    // Scanner(System.in);

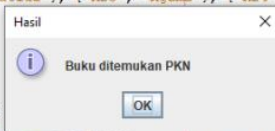
    JOptionPane.showInputDialog(null, "Judul buku yang dicari : ", "Judul", JOptionPane.INFORMATION_MESSAGE, null, null, null, null);
    String input = JOptionPane.showInputDialog("Judul");
    if (input != null) {
        for (int i = 0; i < names.length; i++) {
            if (names[i].startsWith(input)) {
                JOptionPane.showMessageDialog(f, "Buku ditemukan " + names[i], "Hasil", JOptionPane.INFORMATION_MESSAGE);
                return true;
            }
        }
    }
}
```



```
String[] args) {
    // logic here

    // Scanner(System.in);

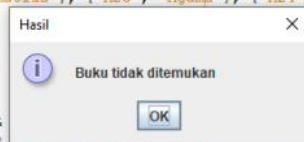
    JOptionPane.showInputDialog(null, "Judul buku yang dicari : ", "Judul", JOptionPane.INFORMATION_MESSAGE, null, null, null, null);
    String input = JOptionPane.showInputDialog("Judul");
    if (input != null) {
        for (int i = 0; i < names.length; i++) {
            if (names[i].startsWith(input)) {
                JOptionPane.showMessageDialog(f, "Buku ditemukan " + names[i], "Hasil", JOptionPane.INFORMATION_MESSAGE);
                return true;
            }
        }
    }
}
```



```
String[] args) {
    // logic here

    // Scanner(System.in);

    JOptionPane.showInputDialog(null, "Judul buku yang dicari : ", "Judul", JOptionPane.INFORMATION_MESSAGE, null, null, null, null);
    String input = JOptionPane.showInputDialog("Judul");
    if (input != null) {
        for (int i = 0; i < names.length; i++) {
            if (names[i].startsWith(input)) {
                JOptionPane.showMessageDialog(f, "Buku ditemukan " + names[i], "Hasil", JOptionPane.INFORMATION_MESSAGE);
                return true;
            }
        }
    }
}
```



5. Kesimpulan

Pada karya tulis ini, penulis menggunakan program java sederhana dalam implementasi algoritma searching (pencarian) dengan metode sequential searching. Data buku tersebut terdiri dari judul dan kode buku yang selanjutnya pengguna akan menginput judul atau kode buku tersebut. Kemudian hasilnya akan terbagi menjadi dua kemungkinan, yaitu jika data yang dicari ada dalam list maka data ditemukan dan jika data yang dicari tidak ada dalam list maka data tidak ditemukan.

Berdasarkan hal tersebut, ditemukan bahwa algoritma searching dengan metode sequential searching sangat efektif digunakan dalam program perpustakaan, karena algoritma ini memiliki kecepatan dalam melakukan pencarian. Sehingga proses pengolahan data buku khususnya dalam proses peminjaman buku akan berlangsung secara efisien. Penggunaan algoritma searching (pencarian) dalam pengolahan data

buku dapat dikembangkan lagi menjadi sistem yang lebih kompleks.

6. Daftar Pustaka

- [1] Wahyuni, W. S., Andryana, S., & Rahman, B. (2022). PENGGUNAAN ALGORITMA SEQUENTIAL SEARCHING PADA APLIKASI PERPUSTAKAAN BERBASIS WEB. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7(2), 294-302.
- [2] Muttaqin, I. (2022). E-LIBRARY BERBASIS WEBSITE MENGGUNAKAN METODE ALGORITMA APRIORI DAN SEQUENTIAL SEARCH. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 9(2), 1223-1232.