

LAPORAN PRAKTIKUM 2
ALGORITMA PEMOGRAMAN DAN
KOMPUTER 2 “Linear Search dan Binary
Search”



Disusun oleh

Jiryan Farokhi/5002221102

Asisten Laboratorium

Muhammad Andhika Reswara/5002211086 dan Komang
Ryaandhi Suandita/5002211109

Dosen Pengampu

Dr. Budi Setiyono, S.Si, MT / 19720207 199702 1 001

DEPARTEMEN
MATEMATIKA FAKULTAS
SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH
NOPEMBER SURABAYA
2022

BAB 1 DESKRIPSI

1. Diberikan sebuah input dengan bertipe char dalam bentuk array yaitu {A, B, C, E, F, B, A, D, F, G, F}. Di sini programmer diminta untuk menampilkan banyaknya satu tipe data char yang diinputkan dan letak tipe data char tersebut dalam array. Alur logika dalam pengerjaan soal ini adalah dengan membuat sebuah method main dan method dua method bertipe int array untuk menyimpan lokasi data yang diinputkan dan method int untuk menampilkan jumlah tipe data yang diinputkan dalam array tersebut. Pada method main terdapat sebuah masukan *Scanner* untuk menginput data dari user. Kemudian masukan tersebut disimpan dalam variabel bertipe char bernama key. Selanjutnya menginisiasi tipe data integer array bernama indeks untuk menampilkan lokasi dari data yang diinputkan dengan method *LinearSearch* dan input berupa data array dan tipe data char yang diinputkan. Setelah itu di-print sebuah string yang sesuai dengan output yang diharapkan dan ditambahkan variabel inputan dan panjang indeks untuk menampilkan banyak kemunculan dari data input, serta *looping* dari i pada indeks yang disebutkan pada method *LinearSearch*. Pada method *LinearSearch*, diinisiasi sebuah tipe data array indeks dengan panjang adalah data char yang diinputkan. Otomatis pada array indeks tidak ada sama sekali data array. Kemudian diinisiasi count, i dengan nilai nol. Selanjutnya *loop* dengan metode *while* agar pengulangan tetap berjalan walaupun data sudah ditemukan. Dalam loop *while* tersebut diberikan kondisi apabila key atau tipe data yang diinputkan sama dengan data ke – i pada pengulangan maka indeks (i+1) disimpan dalam array indeks pada posisi count saat ini. count ditambah untuk menunjukkan slot berikutnya yang tersedia di array indeks. Selanjutnya dibuat array integer baru bernama hasil dengan panjang count yang menunjukkan jumlah kemunculan sebenarnya. Elemen array dari indeks akan dipindahkan ke array hasil. Array hasil ditampilkan dengan berisi indeks semua kemunculan data yang diinputkan.
2. Diberikan sebuah input berupa array bertipe char yaitu (a, c, e, g, i, k, o, q, r). Di sini programmer diminta untuk menampilkan letak posisi data input yaitu r pada array tersebut. Alur logika dalam pengerjaan soal ini pertama-pertama dengan method main untuk menampilkan hasil dan method pencarianBinary untuk alur *logic* dari *binarysearch*. Pada method main diinisiasi sebuah tipe data char seperti yang dijelaskan sebelumnya dan diinisiasi tipe data char key berisi input angka r. Diinisiasi integer indeks untuk menyimpan hasil indeks input data pada method pencarianBinary. Diberikan sebuah kondisi apabila nilai indeks -1 maka data tidak ditemukan apabila tidak maka data ditemukan. Pada method pencarianBinary, diinisiasi integer bawah dan atas dengan bawah bernilai nol dan atas bernilai panjang data – 1. Atas dikurangi satu karena panjang indeks selalu berkurang satu apabila dibaca oleh komputer. Selanjutnya diberikan *looping* dengan kondisi jika bawah kurang atas maka diberikan beberapa kondisi. Tetapi sebelumnya ada sebuah nilai awal tengah. Dalam looping ini sistem binary akan membagi data menjadi dua dan nilai tengah akan selalu dibulatkan ke bawah. Jika kata yang dicari berada pada bagian atas atau bawah maka data akan dibagi dua lagi menjadi atas atau bawah dengan nilai atas atau bawah akan selalu berubah sampai input data yang dicari akan menjadi nilai dari tengah itu sendiri. Maka dari itu data bawah akan bernilai tengah + 1 jika data input lebih dari data tengah mula-mula dan atas akan bernilai tengah – 1 jika data input kurang dari data tengah mula-mula.

BAB 2

SOURCE CODE

Soal 1

```
import java.util.Scanner;

public class SequentialSearch {

    public static void main(String[] args) {
        char[] data = {'A', 'B', 'C', 'E', 'F', 'B', 'A', 'D', 'G', 'F'};
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkkan karakter yang akan dicari : ");
        char key = input.next().charAt(0);
        int[] indeks = linearSearch(data, key);
        System.out.print("Huruf " + key + " ditemukan sebanyak " +
indeks.length + " kali yaitu pada urutan ke ");
        for(int i = 0; i < indeks.length; i++){
            if(i != indeks.length - 1 ) {
                System.out.print(indeks[i] + ", ");
            } else {
                System.out.print(indeks[i]);
            }
        }

    }

    public static int[] linearSearch(char [] data, char key){
        int[] indeks = new int[data.length];
        int i = 0;
        int count = 0;
        while (i < data.length) {
            if (key == data[i]) {
                indeks[count] = i + 1;
                count++;
            }
            i++;
        }
        int[] hasil = new int[count];
        for(int j = 0; j < count; j++){
            hasil[j] = indeks[j];
        }
        return hasil;
    }

}
```

Soal 2

```
public class BinarySearch {
```

```

public static void main(String[] args) {
    final long START = System.nanoTime();
    char[] data = {'a','c','e','g','i','k','o','q','r'};
    char key = 'r';
    int indeks;
    indeks = pencarianBinary(data, key);
    if (indeks == -1)
        System.out.println("Huruf " + key + " Tidak ditemukan ");
    else
        System.out.println("Huruf " + key + " berada pada indeks ke - " +
(indeks));
    final long END = System.nanoTime();
    System.out.println("Time taken : " + ((END - START) / 1e+9) + "
seconds");
}

public static int pencarianBinary(char[] data, char key) {
    int bawah = 0;
    int atas = data.length - 1;

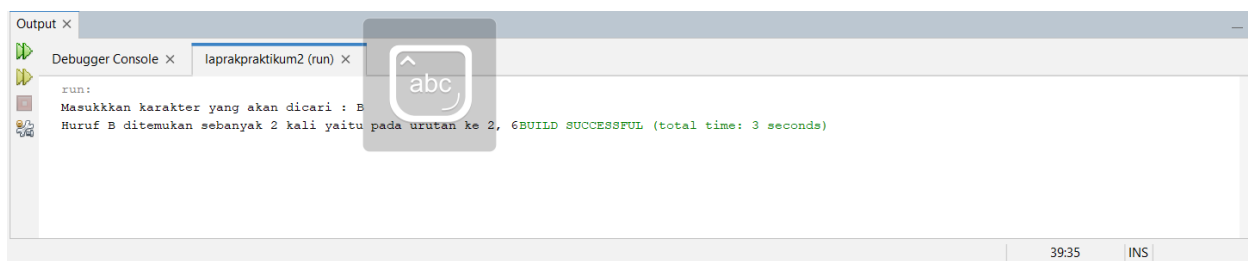
    while (bawah <= atas) {
        int tengah = (bawah + atas) / 2;
        if (key == data[tengah]) {
            return tengah;
        } else if (key > data[tengah]) {
            bawah = tengah + 1;
        } else {
            atas = tengah - 1;
        }
    }

    return -1;
}
}

```

BAB 3 OUTPUT PROGRAM

Soal 1



Soal 2

