

LAPORAN PRAKTIKUM 2
ALGORITMA PEMOGRAMAN DAN
KOMPUTER 2 “Sorting”



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. Tujuan dari program ini adalah untuk mencari angka tertentu dalam sebuah array bilangan bulat menggunakan teknik Insertion Sort dan pencarian Binary Search. Pada *method* main langkah-langkahnya adalah sebagai berikut. Pertama, mendeklarasikan array bilangan bulat data. Kedua, meminta pengguna memasukkan angka yang ingin dicari. Ketiga, memanggil metode InsertSort untuk mengurutkan array. Keempat, Mencetak array yang telah diurutkan. Kelima, Memanggil metode pencarianBinary untuk mencari angka yang diinput. Keenam, mencetak hasil pencarian (apakah angka ditemukan dan indeks posisinya, atau tidak ditemukan). Pada *method* InsertSort, mengimplementasikan algoritma Insertion Sort untuk mengurutkan array secara ascending. Untuk lebih jelasnya sebagai berikut. `for (int i = 1; i < n; i++)`: Melakukan iterasi dari elemen kedua (indeks 1) hingga elemen terakhir (indeks n-1) array. `v = arr[i]`: Menyimpan nilai elemen array saat ini di variabel v. `j = i-1`: Menginisialisasi variabel j dengan indeks elemen sebelumnya. `while(j >= 0 && arr[j]>v)`: Melakukan perulangan selama j lebih besar atau sama dengan 0 dan nilai elemen pada indeks j lebih besar dari v. `arr[j+1] = arr[j]`: Menggeser elemen array pada indeks j satu posisi ke kanan. `j = j-1`: Mengurangi nilai j untuk memeriksa elemen sebelumnya. `arr[j+1] = v`: Menyisipkan nilai v ke array pada indeks j+1. Algoritma InsertSort mengimplementasikan Pada *method* pencarianBinary, mengimplementasikan algoritma Binary Search untuk mencari angka dalam array yang telah diurutkan. Untuk lebih jelasnya sebagai berikut. `int bawah = 0`: Menginisialisasi variabel bawah dengan indeks awal array (0). `int atas = data.length-1`: Menginisialisasi variabel atas dengan indeks akhir array (n-1). `while (bawah <= atas)`: Melakukan perulangan selama bawah lebih kecil atau sama dengan atas. `int tengah = (bawah + atas) / 2`: Menghitung indeks tengah array. `if (key == data[tengah])`: Memeriksa apakah angka yang dicari (key) sama dengan nilai elemen pada indeks tengah. `return tengah+1`: Mengembalikan indeks elemen yang ditemukan. `else if (key > data[tengah])`: Memeriksa apakah angka yang dicari (key) lebih besar dari nilai elemen pada indeks tengah. `bawah = tengah + 1`: Memperbarui nilai bawah dengan indeks tengah + 1. `else`: Angka yang dicari (key) lebih kecil dari nilai elemen pada indeks tengah. `atas = tengah - 1`: Memperbarui nilai atas dengan indeks tengah - 1. `return -1`: Angka yang dicari (key) tidak ditemukan dalam array.
2. Tujuan dari program ini dirancang untuk menampilkan 5 nilai terbesar dari sebuah array bilangan bulat menggunakan Bubble Sort. Pada *method* main langkah-langkahnya adalah sebagai berikut. Pertama, mendeklarasikan array bilangan bulat data. Kedua, memanggil *method* bubblesort untuk mengurutkan array secara descending (menurun). Ketiga, mencetak array yang telah diurutkan. Keempat memanggil metode tampil5terbesar untuk menampilkan 5 nilai terbesar dari array. Pada *method* bubblesort, mengimplementasikan algoritma Bubble Sort untuk mengurutkan array secara descending. Untuk penjelasan lebih lengkapnya sebagai berikut. `int n = arr.length`: Menyimpan jumlah elemen array. `int temp = 0`: Menyimpan nilai sementara untuk pertukaran elemen. `for(int i = 0; i < n; i++)`: Melakukan perulangan sebanyak n kali untuk mengurutkan seluruh elemen. `for(int j = 1; j < n; j++)`: Melakukan perulangan di dalam loop i untuk membandingkan elemen berdekatan. `if(arr[j-1] < arr[j])`: Membandingkan nilai elemen pada indeks j-1 dan j. `temp = arr[j-1]`: Menyimpan nilai elemen j-1 ke dalam variabel temp. `arr[j-1] = arr[j]`: Mengubah nilai elemen j-1 dengan nilai elemen j. `arr[j] = temp`: Mengubah nilai elemen j dengan nilai

yang disimpan di temp. Pada *method* tampil5terbesar, menyalin 5 elemen pertama dari array yang diurutkan ke dalam array baru berukuran 5 dan mencetak isi array baru yang berisi 5 nilai terbesar. Untuk penjelasan lebih lengkapnya sebagai berikut. `int[] data = new int[5];` Membuat array baru dengan ukuran 5 untuk menampung 5 nilai terbesar. `for(int i = 0; i < 5; i++)`: Melakukan perulangan untuk menyalin 5 elemen pertama dari array yang diurutkan ke array baru. `data[i] = arr[i];` Menyalin nilai elemen i dari array arr ke array data. `System.out.println(Arrays.toString(data));` Mencetak isi array data yang berisi 5 nilai terbesar.

BAB 2

SOURCE CODE

Soal 1

```
import java.util.Arrays;
import java.util.Scanner;

public class No1 {

    public static void main(String[] args) {
        int[] data = {18, 28, 36, 38, 81, 82, 93};
        System.out.println("Masukkan angka yang akan di cari : ");
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        System.out.println("data setelah diurutkan : ");
        InsertSort(data);
        System.out.println(Arrays.toString(data));
        int indeks;
        indeks = pencarianBinary(data, n);
        if (indeks == -1)
            System.out.println("Huruf " + n + " Tidak ditemukan ");
        else
            System.out.println("Huruf " + n + " berada pada urutan ke - " +
(indeks));
    }

    public static void InsertSort (int[] arr){
        int n = arr.length;
        int v, j;
        for(int i = 1; i < n; i++){
            v = arr[i];
            j = i-1;

            while(j >= 0 && arr[j]>v){
                arr[j+1] = arr[j];
                j = j-1;
            }
            arr[j+1] = v;
        }
    }

    public static int pencarianBinary(int[] data, int key) {
```

```

int bawah = 0;
int atas = data.length-1;

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

return -1;
}

}

```

Soal 2

```

import java.util.Arrays;
import java.util.Scanner;

public class No2 {

    public static void main(String[] args) {
        int[] data = {15, -2, 0, 10, 59, 5, -10, 53, -25, 36};
        Scanner input = new Scanner(System.in);
        System.out.println("data setelah diurutkan : ");
        bubblesort(data);
        System.out.println(Arrays.toString(data));
        System.out.println("5 Nilai terbesar : ");
        tampil5terbesar(data);
    }

    public static void tampil5terbesar (int [] arr){
        int[] data = new int[5];
        for(int i = 0; i < 5; i++){
            data[i] = arr[i];
        }
        System.out.println(Arrays.toString(data));
    }

    public static void bubblesort(int[]arr){
        int n = arr.length;
        int temp = 0;
    }
}

```

```

        for(int i = 0; i < n; i++){
            for(int j = 1; j < n; j++){
                if(arr[j-1] < arr[j]){
                    temp = arr[j-1];
                    arr[j-1] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
}

```

BAB 3 OUTPUT PROGRAM

Soal 1

```

run:
Masukkan angka yang akan di cari :
81
data setelah diurutkan :
[18, 28, 36, 38, 81, 82, 93]
Huruf 81 berada pada urutan ke - 5
BUILD SUCCESSFUL (total time: 4 seconds)

```

Soal 2

