

 <p> PRODI INFORMATIKA FAK. SAINS DAN TEKNOLOGI UNIV. SANATA DHARMA </p>	<p> SOAL UJIAN TENGAH SEMESTER SEMESTER GASAL TA 2022/2023 </p> <p> Mata Kuliah : Struktur Data Linier D Kode / SKS : ALGO 104 / 4 Hari / Tanggal : Rabu / 19 Oktober 2022 Waktu Ujian : 120 Menit Sifat : Buku terbuka Pengampu : Rosalia Arum Kumalasanti,S.T.,M.T. </p>
--	--

Nama : Johanes Yogtan Wicaksono Raharja

NIM : 215314105

A. Kerjakan soal-soal di bawah ini pada file docx / pdf. dan upload pada folder LMS.
File disimpan dengan nama UTS1_NIM.docx /.pdf

1. Jelaskan perbedaan dari Struktur data statis dengan Struktur data dinamis !

Jawaban :

Struktur data statis ukurannya tidak dapat berkurang atau bertambah saat waktu eksekusi, setelah suatu ukuran ditetapkan, baik itu sebelum maupun saat di eksekusi, ukuran tidak dapat berubah. Sedangkan **Struktur data dinamis** ukurannya dapat berubah atau bertambah saat waktu eksekusi, setelah suatu ukuran ditetapkan, baik itu sebelum maupun saat di eksekusi, ukuran dapat berubah.

Perbedaan Keuntungannya

Struktur data statis	Struktur data dinamis
Ukuran tipe data ditetapkan	Alokasi memori dilakukan selama eksekusi program.
Alokasi memori sebelum eksekusi program.	Timbunan digunakan.
Tumpukan digunakan	Lebih efisien.
Kurang efisien daripada struktur data dinamis.	Penggunaan kembali memori.
Tidak ada penggunaan kembali memori.	Terjadi luapan.
Overflow tidak mungkin.	Ukuran tipe data tidak tetap

2. Jelaskan apa yang dimaksud dengan Abstract Data Type !

Jawaban : **Pengertian** Abstract Data Type adalah sebuah konsep yang perilakunya ditentukan oleh sekumpulan nilai dan serangkaian operasi. Jadi menjadikan sebuah konsep yang dimana pengguna lebih tertarik pada fungsional/operasi yang disediakan daripada bagaimana fungsional/operasi itu diimplementasikan, oleh karena itu disebut “**abstract**” karena mempunyai implementasi sendiri dalam sebuah operasi.

Contohnya ADT dari struktur data tumpukan, maka user tau data tumpukan memiliki operasi push dan pop, tetapi tidak perlu mengerti bagaimana operasi itu dikerjakan dan bagaimana data itu disimpan

Bagian abstract data types yaitu ada int di komputer yang merupakan *representasi* abstract dari bilangan integer di matematika, terus juga ada operasi yang dilakukan yaitu penambahan, pengurangan, perkalian.

Saya juga mengetahui adanya beberapa Abstract Data Type dari yang saya pelajari ada stack dan juga queues.

3. Buatlah data menggunakan tanggal lahir dengan cara menambah tiap angka tanggal lahir dengan data 1,2,2,3,3,1 secara berurutan. Misal tanggal lahir anda 14-02-04 (tanggal 14 bulan Februari tahun 2004) maka data yang terbentuk (1+1),(2+4),(2+0),(3+2),(3+0),(1+4). Perhatikan angka yang dicetak tebal adalah tanggal lahir anda yaitu 1,4,0,2,0,4. Data yang terbentuk : 2,6,2,5,3,5. Tunjukkan Langkah ini saat membuat data. Urutkan data di atas menggunakan metode pengurutan secara ascending :

Screenshot Program :

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int [] array;

    System.out.println("Masukkan Jumlah Tanggal!");
    System.out.println("*Contoh : 13-03-03, berarti ada 6");
    System.out.print("Masukkan : ");
    int n = sc.nextInt();
    array = new int[n];
    System.out.println("-----");
    System.out.println("Masukkan Tanggal Lahir!");
    System.out.println("*Contoh : 13-03-03, berarti\n"
        + "Data 1 : 1\n"
        + "Data 2 : 3\n"
        + "Dst...");
    System.out.print("Masukkan : ");
    for(int i = 0; i < array.length; i++){
        System.out.print("Data "+(i+1)+" : ");
        if (i == 0) array[i] = sc.nextInt()+1;
        else if (i == 1) array[i] = sc.nextInt()+2;
        else if (i == 2) array[i] = sc.nextInt()+2;
        else if (i == 3) array[i] = sc.nextInt()+3;
        else if (i == 4) array[i] = sc.nextInt()+3;
        else if (i == 5) array[i] = sc.nextInt()+1;
    }
    System.out.println("-----");

    System.out.println("DATA SEBELUM DIURUTKAN");
    cetak(array);
}
```

Output Program

```
run:
Masukkan Jumlah Tanggal!
*Contoh : 13-03-03, berarti ada 6
Masukkan : 6
-----
Masukkan Tanggal Lahir!
*Contoh : 13-03-03, berarti
Data 1 : 1
Data 2 : 3
Dst...
Masukkan :
Data 1 : 1
Data 2 : 3
Data 3 : 0
Data 4 : 3
Data 5 : 0
Data 6 : 3
-----
DATA SEBELUM DIURUTKAN
2 5 2 6 3 4
DATA SESUDAH DIURUTKAN
2 2 3 4 5 6
```

a. Buble Sort

Screenshot Program :

```
        bubbleSortAsc(array);
        System.out.println("\nDATA SESUDAH DIURUTKAN");
        cetak(array);
        System.out.println("");
    }

    static void bubbleSortAsc(int array[]) {
        int jum;
        int n = array.length;
        for (int i = 0; i < n - 1; i++)
            for (int j = 0; j < n - i - 1; j++)
                if (array[j] > array[j + 1]) {
                    jum = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = jum;
                }
    }

    public static void cetak(int array[]) {
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
    }
}
```

Ilustrasi

Iterasi 1

X[0]	Dengan	X[1]	(2 dengan 5)	Tak Berubah
X[1]	Dengan	X[2]	(5 dengan 2)	Ditukar
X[2]	Dengan	X[3]	(5 dengan 6)	Tak Berubah
X[3]	Dengan	X[4]	(6 dengan 3)	Ditukar
X[4]	Dengan	X[5]	(6 dengan 4)	Ditukar

Iterasi 2

X[0]	Dengan	X[1]	(2 dengan 2)	Tak Berubah
X[1]	Dengan	X[2]	(2 dengan 5)	Tidak Berubah
X[2]	Dengan	X[3]	(5 dengan 3)	Ditukar
X[3]	Dengan	X[4]	(5 dengan 4)	Ditukar
X[4]	Dengan	X[5]	(5 dengan 6)	Tidak Berubah

Iterasi 3 **SAMPAI** Iterasi 5

X[0]	Dengan	X[1]	(2 dengan 2)	Tak Berubah
X[1]	Dengan	X[2]	(2 dengan 3)	Tidak Berubah
X[2]	Dengan	X[3]	(3 dengan 4)	Tidak Berubah
X[3]	Dengan	X[4]	(4 dengan 5)	Tidak Berubah
X[4]	Dengan	X[5]	(5 dengan 6)	Tidak Berubah

Penjelasan

Seperti iterasi 1 dan 2 diatas, indek 0 dan 1 di bandingkan, ternyata 3 lebih besar dari 2, sehingga data ditukar, dilanjutkan indek 1 dan 2, ternyata 3 tidak lebih besar dari 5, sehingga data tidak ditukar, dilanjutkan indek 2 dan 3, ternyata 5 tidak lebih besar dari 7, sehingga data tidak ditukar, dilanjutkan indek 2 tadi yang sudah dipindahkan untuk membandingkan, indek 3 dan 4, ternyata 7 tidak lebih besar dari 8, sehingga data tidak ditukar, dilanjutkan indek 4 dan 5, ternyata 8 tidak lebih besar dari 9, sehingga data tidak berubah. Begitu seterusnya hingga perulangan 1 selesai dan data tidak ditukar.

	[0]	[1]	[2]	[3]	[4]	[5]	Jumlah Perbandingan (jumlah pertukaran yang terjadi)
Awal	2	5	2	6	3	4	-
Iterasi 1	2	2	5	3	4	6	3
Iterasi 2	2	2	3	4	5	6	2
Iterasi 3	2	2	3	4	5	6	0
Iterasi 4	2	2	3	4	5	6	0
Iterasi 5	2	2	3	4	5	6	0

b. Insertion Sort

Screenshot

```
insertionSortAsc(array);
System.out.println("\nDATA SESUDAH DIURUTKAN");
cetak(array);
System.out.println("");
}

static void insertionSortAsc(int array[]) {
    int n = array.length;
    for (int j = 1; j < n; j++) {
        int key = array[j];
        int i = j - 1;
        while ((i > -1) && (array[i] > key)) {
            array[i + 1] = array[i];
            i--;
        }
        array[i + 1] = key;
    }
}

public static void cetak(int array[]) {
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + " ");
    }
}
```

Ilustrasi

Iterasi 1

X[0]	Dengan	X[1]	(2 dengan 5)	Tidak Berubah
X[1]	Dengan	X[2]	(5 dengan 2)	Ditukar
X[2]	Dengan	X[3]	(2 dengan 6)	Tidak Berubah
X[2]	Dengan	X[3]	(6 dengan 3)	Ditukar
X[4]	Dengan	X[5]	(3 dengan 4)	Tidak Berubah

Iterasi 2

X[1]	Dengan	X[2]	(2 dengan 5)	Tidak Berubah
X[2]	Dengan	X[3]	(5 dengan 3)	Ditukar
X[3]	Dengan	X[4]	(3 dengan 6)	Tidak Berubah
X[3]	Dengan	X[4]	(6 dengan 4)	Ditukar
X[4]	Dengan	X[5]	(26 dengan 31)	Tidak Berubah

Iterasi 3

X[2]	Dengan	X[3]	(3 dengan 5)	Tidak Berubah
X[3]	Dengan	X[4]	(5 dengan 4)	Ditukar
X[4]	Dengan	X[5]	(4 dengan 6)	Tidak Berubah

Penjelasan

	[0]	[1]	[2]	[3]	[4]	[5]	Jumlah Perbandingan (jumlah pertukaran yang terjadi)
Awal	2	5	2	6	3	4	-
Iterasi 1	2	2	5	3	6	4	2
Iterasi 2	2	2	3	5	4	6	2
Iterasi 3	2	2	3	4	5	6	1
Iterasi 4	2	2	3	4	5	6	0
Iterasi 5	2	2	3	4	5	6	0

c. Selection Sort

Screenshot

```
selectionSort(array);
System.out.println("\nDATA SESUDAH DIURUTKAN");
cetak(array);
System.out.println("");
}

static void selectionSort(int array[]) {
    int n = array.length;
    for (int i = 0; i < n - 1; i++) {
        int indek = i;
        for (int j = i + 1; j < n; j++) {
            if (array[j] < array[indek]) {
                indek = j;
            }
        }
        int jum = array[indek];
        array[indek] = array[i];
        array[i] = jum;
    }
}

public static void cetak(int array[]) {
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + " ");
    }
}
```

Ilustrasi

X[0]	Dengan	X[1]	(2 dengan 5)	Tidak Berubah
X[1]	Dengan	X[2]	(2 dengan 2)	Tidak Berubah
X[2]	Dengan	X[3]	(2 dengan 6)	Tidak Berubah
X[3]	Dengan	X[4]	(2 dengan 3)	Tidak Berubah
X[4]	Dengan	X[5]	(2 dengan 4)	Tidak Berubah

Iterasi 2

X[0]	Dengan	X[1]	(5 dengan 2)	Ditukar
X[1]	Dengan	X[2]	(5 dengan 6)	Tidak Berubah
X[2]	Dengan	X[3]	(5 dengan 3)	Ditukar
X[3]	Dengan	X[4]	(5 dengan 4)	Ditukar

Iterasi 3

X[0]	Dengan	X[1]	(6 dengan 3)	Ditukar
X[1]	Dengan	X[2]	(6 dengan 4)	Ditukar
X[2]	Dengan	X[3]	(6 dengan 5)	Ditukar

Penjelasan

	[0]	[1]	[2]	[3]	[4]	[5]	Jumlah Perbandingan (jumlah pertukaran yang terjadi)
Awal	2	5	2	6	3	4	-
Iterasi 1	2	5	2	6	3	4	
Iterasi 2	2	2	6	3	4	5	3
Iterasi 3	2	2	3	4	5	6	3
Iterasi 4	2	2	3	4	5	6	0
Iterasi 5	2	2	3	4	5	6	0

Tuliskan pertukaran data untuk setiap langkah yang dilakukan. Kelengkapan dan kejelasan setiap Langkah menjadi dasar penilaian

- Setelah data terurut pada nomor 3, lakukan pencarian data pada urutan terakhir menggunakan algoritma **binary search**. Tuliskan data yang akan dicetak dari perintah yang digaris bawah

```
public int binarySearch( double key) {
    int low = 0; // low end of the search area
    int high = this.data.length - 1; // high end of the search area
    int middle = (low + high ) / 2; // middle element
    int location = -1; // return value; -1 if not found

    do {
        System.out.println(data[middle]+" ; ");
        if (key == data[middle]) {
            location = middle; // location is the current middle
        }
        else if (key < this.getData()[middle]) { // middle element is
too high
            high = middle - 1; // eliminate the higher half
        }
        else { // middle element is too low
            low = middle + 1; // eliminate the lower half
        }
        middle = (low + high ) / 2; // recalculate the middle
    } while ((low <= high) && (location == -1));
    return location; // return location of search key
}
```


Screenshot Output

```
package array;
import java.util.Scanner;
public class Latihan3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int [] array;

        System.out.println("Masukkan Jumlah Tanggal!");
        System.out.println("*Contoh : 13-03-03, berarti ada 6");
        System.out.print("Masukkan : ");
        int n = sc.nextInt();
        array = new int[n];
        System.out.println("-----");
        System.out.println("Masukkan Tanggal Lahir!");
        System.out.println("*Contoh : 13-03-03, berarti\n"
            + "Data 1 : 1\n"
            + "Data 2 : 3\n"
            + "Dst...");
        System.out.println("Masukkan : ");
        for(int i = 0; i < array.length; i++){
            System.out.print("Data "+(i+1)+" : ");
            if (i == 0) array[i] = sc.nextInt()+1;
            else if (i == 1) array[i] = sc.nextInt()+2;
            else if (i == 2) array[i] = sc.nextInt()+2;
            else if (i == 3) array[i] = sc.nextInt()+3;
            else if (i == 4) array[i] = sc.nextInt()+3;
            else if (i == 5) array[i] = sc.nextInt()+1;
        }
        System.out.println("-----");

        System.out.println("DATA SEBELUM DIURUTKAN");
        cetak(array);

        selectionSort(array);
        System.out.println("\nDATA SESUDAH DIURUTKAN");
        cetak(array);
        System.out.println("");

        System.out.print("\nMasukkan Data Pencarian : ");
        int cari = sc.nextInt();

        binarySearch(array, 0, array.length, cari);
    }
}
```

```

static void selectionSort(int array[]) {
    int n = array.length;
    for (int i = 0; i < n - 1; i++) {
        int indek = i;
        for (int j = i + 1; j < n; j++) {
            if (array[j] < array[indek]) {
                indek = j;
            }
        }
        int jum = array[indek];
        array[indek] = array[i];
        array[i] = jum;
    }
}

public static void cetak(int array[]) {
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + " ");
    }
}

public static int binarySearch(int array[], int indekAwal, int indekAkhir, int cari) {
    int indekTengah = 0;
    while (indekAwal <= indekAkhir) {
        indekTengah = (indekAwal + indekAkhir) / 2;
        if (array[indekTengah] == cari) {
            System.out.println("Data Ditemukan di Indeks " + indekTengah);
            return indekTengah;
        } else if (array[indekTengah] < cari) {
            indekAwal = indekTengah + 1;
        } else {
            indekAkhir = indekTengah - 1;
        }
    }
    System.out.println("Data Tidak Ditemukan");
    return -1;
}
}

```

Screenshot Output

```
run:
Masukkan Jumlah Tanggal!
*Contoh : 13-03-03, berarti ada 6
Masukkan : 6
-----
Masukkan Tanggal Lahir!
*Contoh : 13-03-03, berarti
Data 1 : 1
Data 2 : 3
Dst...
Masukkan :
Data 1 : 1
Data 2 : 3
Data 3 : 0
Data 4 : 3
Data 5 : 0
Data 6 : 3
-----
DATA SEBELUM DIURUTKAN
2 5 2 6 3 4
DATA SESUDAH DIURUTKAN
2 2 3 4 5 6

Masukkan Data Pencarian : 6
Data Ditemukan di Indek 5
```