

PRAKTIKUM 1 - STRUKTUR DATA

ARRAYS

Learning outcomes:

1. Mampu menjelaskan konsep dan implementasi array pada program
2. Mampu melakukan manipulasi data array: menambahkan item, melakukan pencarian, dan menghapus item pada array
3. Mampu mengimplementasikan *ordered array* pada program.
4. Mampu mengimplementasikan *binary search* pada *ordered array*.
5. Mampu menyimpan dan manipulasi objek pada array.

A. PENDAHULUAN

1. Format penulisan code yang digunakan untuk mendeklarasikan sebuah array adalah:

`TipeData namaVariable[] = new TipeData[panjang/ukuranArray];`

Sedangkan format penulisan code untuk menambahkan item pada array yang telah dideklarasikan adalah:

`namaVariable[index]= value;`

Tulis dan jalankan listing program berikut:

```
public class classArray {

    public static void main(String[] args) {
        int[] array = new int[10];
        array[0] = 10;
        array[1] = 20;
        array[2] = 30;
        array[3] = 40;
        array[4] = 50;

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

Pada listing program tersebut, ukuran array yang dideklarasikan adalah 10. *Insert* item dilakukan hingga index ke-4, artinya hanya terdapat 5 item.

Tuliskan output program tersebut dan jelaskan kenapa demikian!

jawaban

2. Tambahkan baris code berikut ini pada listing program nomer 1

```
...

array = new int[20];

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

} //akhir method main
} //akhir class
```

Jalankan program tersebut, apa output yang dihasilkan?

jawaban

Apakah item yang awal dimasukkan (pada listing no 1) masih tersimpan didalam array? Jelaskan kenapa demikian?

jawaban

Hingga tahap ini, yang dapat disimpulkan adalah:

Ukuran array bersifat *fixed* / *not fixed* *)

*)coret salah satu

3. Lakukan *experiment* menggunakan listing nomer 1 untuk menjawab pertanyaan berikut (beri keterangan benar/salah untuk soal berupa statemen dan tulis jawaban untuk soal isian)

Insert item pada array hanya bisa dilakukan secara berurutan mulai index ke-0.	
Insert item pada array hanya bisa dilakukan hingga ukuran array – 1.	
Cell array untuk semua tipe data <i>primitive</i> yang belum diberi value secara <i>default</i> bernilai 0.	
Cell array untuk tipe data <i>String</i> yang belum diberi value secara <i>default</i> bernilai <i>null</i> .	
Keterangan yang muncul jika memasukkan item melebihi ukuran array adalah	

4. Lengkapi listing berikut:

```

public class ClassArray {

    public static void main(String[] args) {
        int[] array = new int[100];
        int nElemen = 0;
        array[0] = 30;
        array[1] = 20;
        array[2] = 60;
        array[3] = 70;
        array[4] = 50;
        array[5] = 10;
        [REDACTED]

        for (int i = 0; i < [REDACTED]; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println("");
    }
}

```

run:
30 20 60 70 50 10
BUILD SUCCESSFUL (total time: 1 second)

Jalankan dan tuliskan penjelasan dari listing yang telah Anda lengkapi!

jawaban

5. Berikut ini adalah listing program array yang dituliskan dalam bentuk *object oriented programming*. Class *HighArray* memiliki method untuk manipulasi array, yaitu *insert*, *find/search*, dan *delete* serta method *display* untuk menampilkan isi array. Method dalam class *HighArray* tersebut dipanggil dan dijalankan pada class *HighArrayApp*.
Pahami listing berikut dengan menulis dan menjalankannya, kemudian tuliskan penjelasan tiap barisnya!

class HighArray {	Awal sebuah kelas bernama HighArray
private int[] arr;	Deklarasi variable integer bertipe array bernama "arr" dengan akses private
private int nElemen;	
public HighArray(int max) {	
arr = new int[max];	
nElemen = 0;	
}	
public void insert(int value) {	
arr[nElemen] = value;	
nElemen++;	
}	
public boolean find(int key) {	
int i;	
for (i = 0; i < nElemen; i++) {	
if (arr[i] == key) {	
break;	
}	
}	
if (i == nElemen) {	
return false;	
} else {	
return true;	
}	
}	
public boolean delete(int value) {	

```

        int i;
        for (i = 0; i < nElemen; i++) {
            if (value == arr[i]) {
                break;
            }
        }
        if (i == nElemen) {
            return false;
        } else {
            for (int j = i; j < nElemen; j++) {
                arr[j] = arr[j + 1];
            }
            nElemen--;
            return true;
        }
    }

    public void display() {
        for (int i = 0; i < nElemen; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println("");
    }
}

public class HighArrayApp {

    public static void main(String[] args) {
        int maxSize = 100;
        HighArray arr;
        arr = new HighArray(maxSize);

        arr.insert(70);
        arr.insert(80);
        arr.insert(75);
        arr.insert(55);
        arr.insert(85);
        arr.insert(25);
        arr.insert(30);
        arr.insert(00);
        arr.insert(90);
        arr.insert(40)

        arr.display();
    }
}

```

int key = 25;	
if (arr.find(key)) {	
System.out.println(key +	
" ditemukan");	
} else {	
System.out.println(key +	
" tidak ditemukan");	
}	
arr.delete(00);	
arr.delete(80);	
arr.delete(55);	
arr.display();	
}	
}	

Output program tersebut adalah....

jawaban

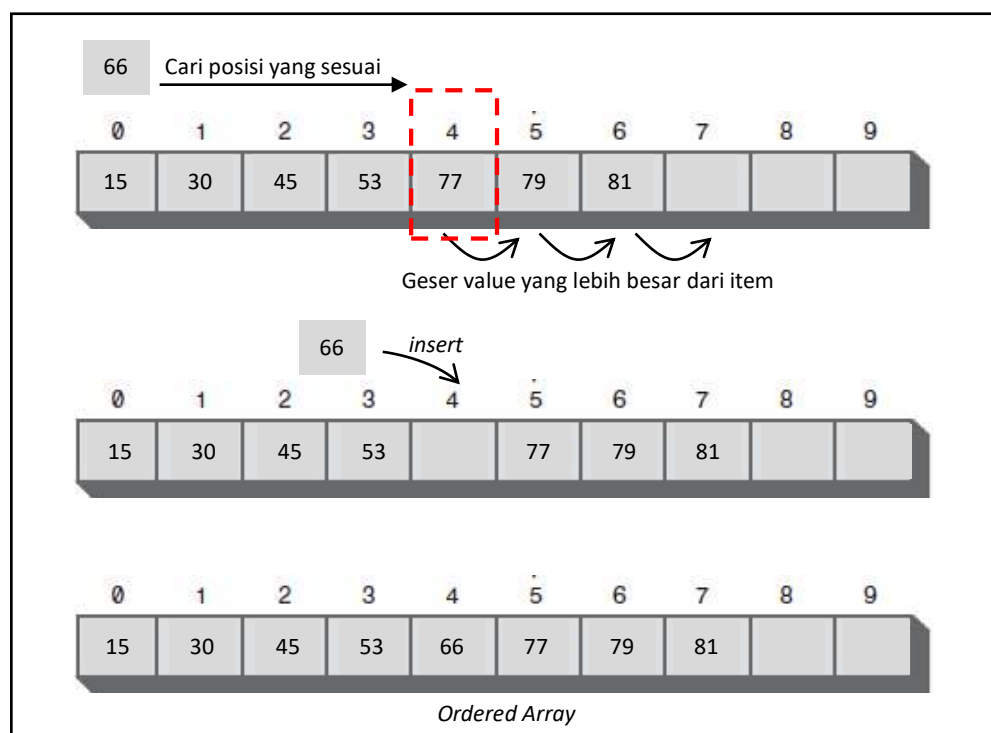
6. Tambahkan sebuah method *size* pada class *HighArray* yang mampu mengembalikan nilai jumlah elemen array. Panggil method tersebut pada class *HighArrayApp* untuk menampilkan jumlah elemen.
Tulis code dan penjelasannya!

jawaban

B. PRAKTIKUM

1. Pada listing nomer 5 (tugas pendahuluan), method *insert* digunakan untuk menambahkan item pada *cell* yang belum terisi tanpa memberhatikan *value* item yang ditambahkan sehingga elemen pada array disimpan secara tidak berurutan (*unordered*).

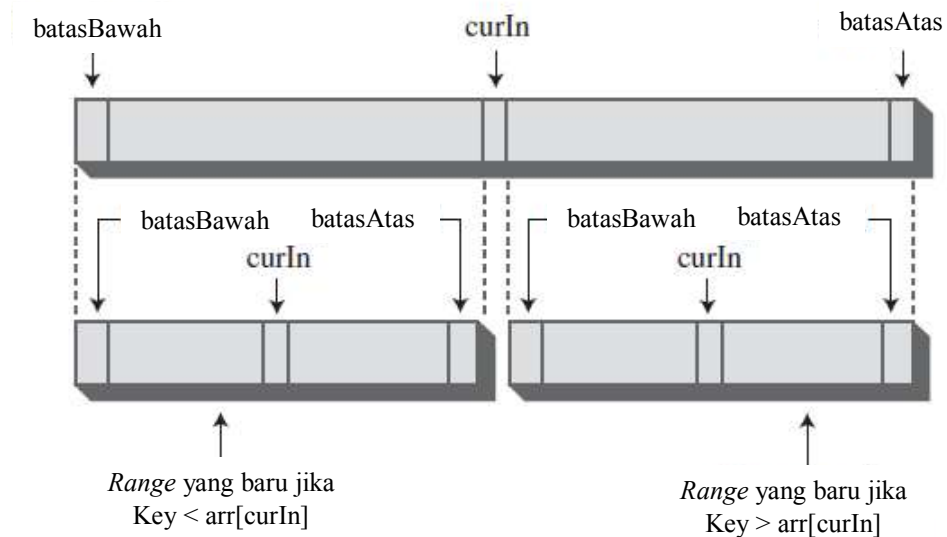
Agar item dapat disimpan pada urutan yang sesuai dengan *value*-nya maka perlu dilakukan pencarian posisi *cell* yang tepat bagi item yang akan dimasukkan dengan cara membandingkan tiap item pada *cell* dengan item yang akan dimasukkan, yaitu pencarian secara *linier*. Setelah *cell* tepat ditemukan, langkah selanjutnya adalah menyiapkan *cell* tersebut untuk diisi jika sudah ada item yang tersimpan pada *cell* tersebut. Hal ini bisa dilakukan dengan cara menggeser item yang memiliki *value* lebih besar dari item yang akan dimasukkan, dengan demikian terdapat *cell* kosong untuk diisi dengan item baru. Langkah-langkah *insert* item pada *ordered array* ditunjukkan pada Gambar 1.1 berikut ini.



Gambar 1.1 Langkah *insert* item pada *ordered array*

Tuliskan listing untuk method *insert* untuk menyimpan elemen array secara berurutan (*ordered*)!

2. Pencarian pada method *find* listing nomer 5 (tugas pendahuluan) menggunakan *linier search*, artinya terhadap key yang dicari, program akan melakukan pencarian pada array secara berurutan mulai dari elemen pertama hingga elemen terakhir. Hal ini tidak efisien. Pada *ordered array*, dapat dilakukan pencarian menggunakan *binary search* yang lebih efisien dibandingkan dengan *linier search*. Pada *binary search*, *range* elemen array dibagi dua secara berulang-ulang. Hal ini menjadikan *range* pencarian semakin kecil dan terpusat pada item yang memiliki *value* mendekati *key* pencarian. Pembagian *range* pencarian pada *binary search* ditunjukkan pada Gambar 1.2 berikut ini.

Gambar 1.2 Pembagian *range* pencarian pada *binary Search*

Tuliskan listing untuk method *find* yang mengimplementasikan *binary search*!

3. *Storing object*

Item data pada *real world* tidak direpresentasikan dalam bentuk data *primitive* tapi berupa *record* yang merupakan kombinasi dari beberapa *field*. Misalkan untuk *record* personal, kita dapat menyimpan nama, tempat tanggal lahir, nomer telpon, email, dsb. Untuk data mahasiswa, kita dapat menyimpan nim, nama, jurusan, asal, dsb. Dalam java, *record* data biasanya direpresentasikan dengan sebuah *class object*.

Berikut ini listing yang menunjukkan implementasi *storing object*. Terdapat tiga class, yaitu class “Mahasiswa”, “DataArray”, dan “DataArrayApp”. *Record* yang disimpan adalah data mahasiswa yang terdiri dari *field* nim, nama, dan asal. *Record* mahasiswa ini direpresentasikan dalam sebuah *class object* dengan nama “Mahasiswa”.

Tulis dan pahami listing program untuk menyimpan object berikut ini.

```
class Mahasiswa {
    private long nim;
    private String nama;
    private String asal;

    public Mahasiswa(long nim, String nama, String asal) {
        this.nim = nim;
        this.nama = nama;
        this.asal = asal;
    }

    public void displayMhs() {
        System.out.print("\tNIM: " + nim);
        System.out.print(", Nama: " + nama);
        System.out.println(", Asal: " + asal);
    }

    public long getNim() {
        return nim;
    }
}
```



```

public class DataArray {

    private Mahasiswa[] mhs;
    private int nElemen;

    public DataArray(int max) {
        mhs = new Mahasiswa[max];
        nElemen = 0;
    }

    public Mahasiswa find(long searchNim) {
        int i;
        for (i = 0; i < nElemen; i++) {
            if (mhs[i].getNim()==searchNim) {
                break;
            }
        }
        if (i == nElemen) {
            return null;
        } else {
            return mhs[i];
        }
    }

    public void insert(long nim, String nama, String asal) {
        mhs[nElemen] = new Mahasiswa(nim, nama, asal);
        nElemen++;
    }

    public boolean delete(long searchNim) {
        int i;
        for (i = 0; i < nElemen; i++) {
            if (mhs[i].getNim()==searchNim) {
                break;
            }
        }
        if (i == nElemen) {
            return false;
        } else {
            for (int j = i; j < nElemen; j++) {
                mhs[j] = mhs[j + 1];
            }
            nElemen--;
            return true;
        }
    }

    public void displayArray(){
        for (int i = 0; i < nElemen; i++) {
            mhs[i].displayMhs();
        }
    }
}

```

Objek mahasiswa disimpan dalam array. Class “DataArray” berisi method-method untuk manipulasi object mahasiswa, yaitu *insert*, *find*, dan *delete*, serta method untuk menampilkan array berisi object mahasiswa, yaitu *displayArray*.

Class yang digunakan untuk menjalankan program adalah class “DataArrayApp”. Class ini memiliki method main yang didalamnya terdapat listing untuk memanggil dan menjalankan fungsi-fungsi pada class DataArray yang telah dibuat.

```

public class DataArrayApp {

    public static void main(String[] args) {
        int maxSize = 100;
        DataArray arr;
        arr = new DataArray(maxSize);
        arr.insert(16650200, "Jundi", "Malang");
        arr.insert(16650210, "Ahmad", "Sidoarjo");
        arr.insert(16650220, "Ismail", "Banyuwangi");
        arr.insert(16650230, "Sofi", "Semarang");
        arr.insert(16650240, "Dinda", "Bandung");
        arr.insert(16650250, "Rais", "Ambon");
        arr.insert(16650260, "Helmi", "Madura");
        arr.insert(16650270, "Agung", "Madiun");
        arr.insert(16650280, "Arina", "Malang");

        arr.displayArray();

        long searchKey = 16650270;
        Mahasiswa mhs = arr.find(searchKey);
        if (mhs != null) {
            System.out.print("\nketemu");
            mhs.displayMhs();
        } else {
            System.out.println("ga ketemu " + searchKey);
        }

        searchKey=16650240;
        System.out.println("\nhapus nim: "+searchKey);
        arr.delete(searchKey);

        arr.displayArray();
    }
}

```

Output:

```

run:
NIM: 16650200, Nama: Jundi, Asal: Malang
NIM: 16650210, Nama: Ahmad, Asal: Sidoarjo
NIM: 16650220, Nama: Ismail, Asal: Banyuwangi
NIM: 16650230, Nama: Sofi, Asal: Semarang
NIM: 16650240, Nama: Dinda, Asal: Bandung
NIM: 16650250, Nama: Rais, Asal: Ambon
NIM: 16650260, Nama: Helmi, Asal: Madura
NIM: 16650270, Nama: Agung, Asal: Madiun
NIM: 16650280, Nama: Arina, Asal: Malang

ketemu NIM: 16650270, Nama: Agung, Asal: Madiun

hapus nim: 16650240
NIM: 16650200, Nama: Jundi, Asal: Malang
NIM: 16650210, Nama: Ahmad, Asal: Sidoarjo
NIM: 16650220, Nama: Ismail, Asal: Banyuwangi
NIM: 16650230, Nama: Sofi, Asal: Semarang
NIM: 16650250, Nama: Rais, Asal: Ambon
NIM: 16650260, Nama: Helmi, Asal: Madura
NIM: 16650270, Nama: Agung, Asal: Madiun
NIM: 16650280, Nama: Arina, Asal: Malang
BUILD SUCCESSFUL (total time: 1 second)

```

C. KESIMPULAN

Kesimpulan yang diperoleh dari pembahasan praktikum kali ini adalah:

1. Tentang *unordered arrays* dan *ordered arrays*

2. Tentang *linier search* dan *binary search*

3. Tentang menyimpan object (*storing object*)