

MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

ARRAY, STRING

Deskripsi Singkat

Praktikum pemrograman berorientasi objek adalah praktikum yang menggunakan bahasa Java sebagai bantuan dalam memahami konsep pemrograman berorientasi objek. Materi praktikum berisi teori, latihan dan soal pemrograman.

Tujuan

1. Membuat dan menggunakan array satu dimensi
2. Membuat dan menggunakan array multi dimensi
3. Menyalin elemen array
4. Memahami perbedaan String, StringBuffer dan StringBuilder.
5. Menggunakan String, StringBuffer dan StringBuilder pada pemrograman berorientasi objek.

Prasyarat

Siswa telah melakukan praktikum 1-2.

Materi 1 : Array Satu Dimensi

Array adalah sekumpulan variabel yang memiliki tipe data yang sama dan dinyatakan dengan nama yang sama. Array memungkinkan untuk menyimpan data maupun referensi objek dalam jumlah banyak dan terindeks. Array menggunakan indeks integer untuk menentukan urutan elemen-elemennya, dimana elemen pertama dimulai dari indeks 0, elemen kedua berindeks 1, dan seterusnya.

Cara mendeklarasikan variable array:

```
tipedata[] variable; atau typedata variable [];
```

Cara mendefinisikan array:

```
variable = new typedata[ukuran];
```

Contoh:

```
int[] bilangan;  
bilangan = new int[10]; }
```

Dapat diringkas menjadi:

```
int[] bilangan = new int[10];
```

Elemen di dalam array dimasukkan dan diakses dengan menggunakan indeks.

```
indeks = 2;  
bilangan[indeks] = 10;  
System.out.println(bilangan[indeks]);
```

Array juga dapat diinisialisasi dan diberi nilai langsung, contoh:

```
int [] nilai = {34, 56, 23, 12, 78};
```

Materi 2 : Array Dua Dimensi

Array satu dimensi hanya memiliki satu kurungan siku []. Jika array multi dimensi menggunakan kurungan siku [] sebanyak dimensi yang diinginkan. Umumnya dimensi yang sering digunakan adalah dimensi dua dan dimensi tiga. Contoh:

```
int [][] dnilai = new int [3][4];
```

dnilai merupakan array dua dimensi. Dimana dimensi ke-1 bisa menyimpan 3 elemen, dan dimensi ke-2 bisa menyimpan 4 elemen. Pada kasus array dua dimensi, dimensi ke-1 juga sering disebut sebagai baris dan dimensi ke-2 disebut sebagai kolom.

Materi 3 : Panjang Array

Panjang dari suatu array yang telah dibuat, kita dapat memakai atributnya yaitu length. Sebab setiap kita membuat array, kita selalu mendefinisikan ukuran array. Ukuran itulah yang disimpan pada atribut **length**. Berikut format untuk menggunakan length:

var_array.length : total elemen array pada dimensi 1

var_array[i].length : total elemen array pada dimensi 2 untuk indeks ke-i pada dimensi 1

var_array[i][j].length : total elemen array pada dimensi 3 untuk indeks ke-i pada dimensi 1 dan indeks ke-j pada dimensi 2

Contoh:

```
int [] nilai = {34, 56, 23, 12, 78};
```

```
int x = nilai.length;
```

Maka x akan berisi ukuran dari nilai yaitu 5.

Materi 4 : Menyalin Array

Isi dari suatu array dapat kita kopi pada array yang lain dengan memanfaatkan method `arraycopy()` pada class `System`. Format penulisannya sebagai berikut:

```
System.arraycopy(array1, p1, array2, p2, n);
```

array1 = array asal/sumber pengkopian

array2 = array tujuan pengkopian

p1 = posisi indeks awal pengkopian pada array asal

p2 = posisi indeks awal pengkopian pada array tujuan

n = banyaknya elemen array yang akan dikopi

Materi 5 : String

Objek dari String sering sekali kita gunakan sehari-hari, bahkan mungkin di setiap class dari Java kita terdapat rutin yang menangani String. Class String merepresentasikan suatu teks. Class String berisi string yang tetap (immutable string). Artinya sekali instance String dibuat maka isinya tidak bisa diubah. Kelas String memiliki 13 konstruktor yang memungkinkan kita membuat obyek String dan menginisialisasi nilainya dengan menggunakan berbagai macam sumber data yang berbeda. Cara membuat obyek string adalah sebagai berikut:

```
String str = "abc";
```

Sama dengan

```
char data[] = {'a', 'b', 'c'};
```

```
String str = new String(data);
```

Class String memiliki method untuk menangani individual karakter dari suatu string, membandingkan string, mencari string, mendapatkan substring, membuat kopi isi string dan dijadikan lowercase atau uppercase. Java menyediakan operator spesial untuk menggabungkan dua string yaitu operator plus (+). Panjang string dapat diketahui dengan menggunakan method **`length()`**.

Class String terdapat di dalam package java.lang sehingga tidak perlu di-import. Class String memiliki sangat banyak method yang dapat digunakan untuk memproses data string yang anda miliki. Detil setiap method tersebut dapat dilihat pada Java API.

Method di bawah digunakan untuk membandingkan kesamaan dua string. Kedua method tersebut akan mengembalikan nilai boolean.

- **equals(String s)**
Dengan fungsi ini, maka objek string yang bersangkutan akan dibandingkan dengan objek string s, pada parameter fungsi ini, dengan membedakan antara huruf besar dengan huruf kecil.
- **equalsIgnoreCase(String s)**
Dengan fungsi ini, maka objek string yang bersangkutan akan dibandingkan dengan objek string s, pada parameter fungsi ini, dengan tanpa memperdulikan perbedaan antara huruf besar dengan huruf kecil.

Class String memiliki method untuk mengetahui string berawalan atau berakhiran dengan string tertentu. Method tersebut akan mengembalikan nilai Boolean. Berikut deskripsinya.

- **startsWith(String s)**
Dengan fungsi ini, maka objek String yang bersangkutan akan diperiksa, apakah diawali oleh objek String s, pada parameter fungsi ini.
- **endsWith(String s)**
Dengan fungsi ini, maka objek string yang bersangkutan akan diperiksa, apakah diakhiri oleh objek string s, pada parameter fungsi ini.

Class String memiliki method untuk membandingkan urutan string yaitu dengan method **compareTo()**. Method ini membandingkan karakter-karakter pada String secara berurutan dari awal String. Misalnya string pertama bernilai "a" dan string kedua bernilai "b", maka apabila Stringpertama.compareTo(Stringkedua) akan menghasilkan nilai negatif (<0) dan apabila dilakukan sebaliknya akan menghasilkan nilai positif (>0). Nilai 0 akan dihasilkan apabila string pertama dan kedua sama.

Class String memiliki method untuk mengembalikan posisi karakter atau string tertentu di dalam string. Method-method tersebut adalah:

- **indexOf(char karakter)**
Memerlukan argumen berupa karakter dan akan mengembalikan nilai posisi indeks dari karakter yang dicari. Posisi yang dikembalikan adalah posisi pertama dari karakter yang ditemukan. Bila karakter tidak ditemukan, maka akan mengembalikan nilai -1.
- **indexOf(char karakter, int indeks)**
Sama dengan sebelumnya, tetapi memerlukan argumen tambahan, yaitu indeks posisi awal pencarian dalam integer.
- **indexOf(String Str)**
Penggunaan dan fungsi sama dengan method untuk char.

- **indexOf(String str, int indeks)**
Penggunaan dan fungsi sama dengan method untuk char.
- **lastIndexOf(String str)**
Mengembalikan index di dalam string dimana sub string terakhir kali ditemukan, substring berada di ujung string, index paling kanan (rightmost)
- **charAt(int index)**
Mengembalikan karakter pada index tertentu. Index mempunyai range dari 0 sampai length() - 1. Karakter pertama dalam deretan (string) berada pada index 0, index berikutnya 1 dan seterusnya.

Substring dari suatu string juga dapat dicari dengan menggunakan method yang ada dalam kelas String. Method-method tersebut adalah:

- **substring(int indeks)**
Sebuah string akan dihasilkan oleh fungsi ini, yaitu karakter pertama dari string yang dihasilkan adalah karakter ke-index dari objek string yang menggunakan fungsi ini.
- **substring(int indeksAwal, int indexAkhir)**
Sebuah string akan dihasilkan oleh fungsi ini, yaitu karakter pertama dari string yang dihasilkan adalah karakter ke-indeksAwal dari objek string yang menggunakan fungsi ini dan karakter terakhir dari string yang dihasilkan adalah karakter ke-indexAkhir dikurangi 1 dari string yang menggunakan fungsi ini.

Materi 6 : StringBuffer

Class StringBuffer mirip dengan String tetapi bersifat mutable, atau dapat diubah atau dimodifikasi dengan menggunakan beberapa method yang dimilikinya. String buffer aman digunakan oleh beberapa thread. Method-methodnya bersifat synchronized sehingga beberapa operasi yang terjadi pada suatu obyek string buffer akan diselesaikan secara serial sesuai urutan pemanggilan.

Operasi utama class StringBuffer adalah method append dan insert. Method append selalu menambahkan teks diakhir string, sedang method insert menambahkan teks di posisi tertentu. Sebagai contoh:

```
StringBuffer sb1 = new StringBuffer("start");  
  
maka isi sb1 adalah "startle" sb1.append("le");  
  
StringBuffer sb2 = new StringBuffer("start");  
  
maka isi sb1 adalah "starlet" sb2.insert(4,"le");
```

Secara umum, jika sb adalah obyek dari StringBuffer, maka sb.append(x) sama dengan operasi sb.insert(sb.length(),x).

Materi 7 : StringBuilder

StringBuilder adalah string yang bersifat mutable. Operasi yang dimiliki class StringBuilder mirip dengan class StringBuffer. Perbedaannya adalah StringBuilder tidak mendukung sinkronisasi.

LATIHAN 1

Ketik program berikut:

```
public class Array1 {  
    public static void main(String args[]) {  
        int nilai[]=new int[3];  
        nilai[0]=70;  
        nilai[1]=80;  
        nilai[2]=65;  
        double ratarata=0.0;  
  
        for(int i=0; i<nilai.length; i++)  
            ratarata+=nilai[i];  
  
        ratarata/=nilai.length;  
        System.out.println("Nilai rata-rata = " + ratarata);  
    }  
}
```

Simpan, kompilasi dan jalankan program tersebut dan lihat hasilnya. Pahami mengapa hasilnya seperti itu?

LATIHAN 2

Ketik program berikut:

```
/**  
 * Program berikut untuk mencoba array dua dimensi  
 */  
public class Array2 {  
    public static void main(String args[]) {
```

Praktikum 3
Array, String

```
//Array 2 dimensi
int[][] arrx = new int[3][3];
arrx[0][0] = 1;
arrx[0][1] = 2;
arrx[0][2] = 3;
arrx[1][0] = 4;
arrx[1][1] = 5;
arrx[1][2] = 6;
arrx[2][0] = 7;
arrx[2][1] = 8;
arrx[2][2] = 9;
System.out.println("Nilai arrx[0][0] : " + arrx[0][0]);
System.out.println("Nilai arrx[0][1] : " + arrx[0][1]);
System.out.println("Nilai arrx[0][2] : " + arrx[0][2]);
System.out.println("Nilai arrx[1][0] : " + arrx[1][0]);
System.out.println("Nilai arrx[1][1] : " + arrx[1][1]);
System.out.println("Nilai arrx[1][2] : " + arrx[1][2]);
System.out.println("Nilai arrx[2][0] : " + arrx[2][0]);
System.out.println("Nilai arrx[2][1] : " + arrx[2][1]);
System.out.println("Nilai arrx[2][2] : " + arrx[2][2]);
    }
}
```

Simpan dan kompilasi program di atas. Perhatikan hasilnya.

LATIHAN 3

Ubah class Array2, tambahkan kode untuk menghitung jumlah semua data dalam arrx.

LATIHAN 4

Ketik program berikut:

```
/**
 * Program berikut untuk melihat panjang array
 */
public class CariPanjangElemen {
    public static void main(String args[]) {
        int x[][][][]=new int[2][][][];
```

```
x[0]=new int[1][][];
x[0][0]=new int[2][];
x[0][0][0]=new int[3];
x[0][0][1]=new int[2];
x[1]=new int[2][][];
x[1][0]=new int[1][];
x[1][0][0]=new int[2];
x[1][1]=new int[2][];
x[1][1][0]=new int[1];
x[1][1][1]=new int[3];

System.out.println(x.length);
System.out.println(x[0].length);
System.out.println(x[0][0].length);
System.out.println(x[0][0][0].length);
System.out.println(x[0][0][1].length);
System.out.println();
System.out.println(x[1].length);
System.out.println(x[1][0].length);
System.out.println(x[1][0][0].length);
System.out.println(x[1][1].length);
System.out.println(x[1][1][0].length);
System.out.println(x[1][1][1].length);
    }
}
```

Simpan, kompilasi dan jalankan program tersebut. Perhatikan output yang dihasilkan.

LATIHAN 5

Ketik program berikut:

```
/**
 * Program berikut untuk mendapatkan nilai argumen pada method main
 */
public class GetArguments {
    public static void main(String args[]) {
        System.out.println("Tanggal : " + args[0]);
        System.out.println("Bulan : " + args[1]);
    }
}
```



```
        System.out.println("Tahun : " + args[2]);  
    }  
}
```

Simpan, kompilasi. Kemudian jalankan Aplikasi dengan memasukkan argumen tanggal seperti contoh di bawah.

```
D:\contohJava>java GetArguments 23 maret 2017  
Tanggal : 23  
Bulan : maret  
Tahun : 2017
```

Method main memiliki parameter berupa array bertipe String. Berapapun argumen yang diketik saat running, akan disimpan dalam array tersebut dan diakses dengan menggunakan indeks.

LATIHAN 6

Ketik program berikut:

```
/**  
 * Program berikut untuk menyalin array ke array lain  
 */  
public class CopyArray {  
    public static void main(String args[]) {  
        int[] array1 = { 7, 4, 8, 1, 4, 1, 4 };  
        int[] array2 = new int[3];  
  
        System.arraycopy(array1, 0, array2, 0, 3);  
        System.out.print("Array1 : ");  
        for (int i=0; i<array1.length; i++)  
            System.out.print(array1[i] + " ");  
        System.out.println();  
  
        System.out.print("Array2 : ");  
        for (int i=0; i<array2.length; i++)  
            System.out.print(array2[i] + " ");  
    }  
}
```

Simpan, kompilasi dan jalankan program tersebut. Perhatikan output yang dihasilkan.

LATIHAN 7

Apakah perbedaan class String, StringBuffer dan StringBuilder?

Apakah yang dimaksud dengan sifat mutable dan immutable? Beri contoh!

Jelaskan operasi utama append dan insert yang dimiliki oleh StringBuffer!

LATIHAN 8

Ketik program berikut:

```
/**
 * Program berikut untuk menguji kesamaan string
 */
public class CopyArray {
    public static void main(String args[]) {
        String str1 = "Pemrograman";
        String str2 = "Java";
        String str3 = "PemrogramanJava";
        System.out.println("String 1:"+str1);
        System.out.println("String 2:"+str2);
        System.out.println("String 3:"+str3);
        System.out.println("String 1 = String 3 ==> "
            +str1.equals(str3));
        System.out.println("String 2 = String 3 ==> "
            +str2.equals(str3));
        System.out.println("String 1 + String 2 = String 3 ==> "
            +(str1+str2).equals(str3));
        String str4 = "Universitas Syiah Kuala";
        String str5 = "universitas syiah kuala";
        System.out.println("String 4:"+str4);
        System.out.println("String 5:"+str5);
        System.out.println("Perintah Pembandingan dibawah
mengabaikan antara huruf
besar dengan huruf kecil");
        System.out.println("String 4 = String 5 ==> "
            +str4.equalsIgnoreCase(str5));
        System.out.println("String 4 + String 5 = String 5 + String
4 ==> "
            +(str4+str5).equalsIgnoreCase(str5+str4));
    }
}
```

Simpan, kompilasi dan jalankan program tersebut. Perhatikan output yang dihasilkan.

SOAL LATIHAN 9

Ketik program berikut:

```
/**
 * Program berikut untuk menguji awalan atau akhir suatu string
 * berntebet
 */
public class CopyArray {
    public static void main(String args[]) {
        String str1 = "Informatika Universitas Syiah Kuala";
        System.out.println(str1.startsWith("Inf"));
        System.out.println(str1.endsWith("kuala"));
        System.out.println(str1.startsWith("Inf",1));
        System.out.println(str1.startsWith("Inf",2));
        System.out.println(str1.startsWith("Inf",3));
    }
}
```

SOAL-SOAL

1. Buatlah program untuk melakukan operasi penjumlahan dari sebuah matriks dua dimensi.
2. Buatlah program untuk melakukan operasi perkalian dari sebuah matriks dua dimensi.
3. Buatlah program untuk mendapatkan substring dari suatu string tertentu.