

## **LAPORAN 2**

### **Pemrograman Berioorientasi Objek**

**“ Tipe Primitive dan Wrapper””**



**Nama : Panji Akbar**  
**NIM : 60200112059**  
**Kelas : F**

**LABORATORIUM KOMPUTER TERPADU**  
**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI ALAUDDIN**  
**MAKASSAR**  
**2015**

## LEMBAR PENILAIAN PRAKTIKUM

Nama : Panji Akbar .....  
NIM : 60200112059 .....  
Praktikum : Pemrograman berorientasi objek...  
Judul praktikum : Tipe Primitive dan Wrapper .....

No	Komponen penilaian	Nilai
I.	Nilai Kehadiran (100)	
II.	Nilai Pelaksanaan praktikum (100)	
	A. Kedisiplinan (25)	
	B. Keterampilan menggunakan Komputer/software/jaringan (40)	
	C. Keaktifan (25)	
	D. Dan lain ... lain (10)	
III.	Nilai laporan praktikum (100)	
TOTAL (300)		

Tanggal praktikum : .....  
Dikumpulkan tanggal : .....  
Dikoreksi tanggal : .....

Asisten Praktikum,

.....

## LEMBAR PENGESAHAN/ACC PRAKTIKUM

Laporan praktikum pemrograman berorientasi objek dengan judul Tipe Primitive dan Wrapper yang disusun oleh :

Nim : 60200112059

Nama : Panji Akbar

Kelas : C

Kelompok :

Telah diperiksa dan dikonsultasikan kepada Asisten I dan Asisten II  
maka dinyatakan diterima

Makassar, .....

Asisten I

Asisten II

Gunawan, S.Kom

Mudassir

Mengetahui,  
Dosen Penanggung Jawab

Nur Afif, S.T., M.T

## 1. Tujuan Praktikum

- ☐ Mengetahui cara penamaan identifier
- ☐ Mengetahui tipe primitif dan wrapper

## 2. Dasar Teori Kelas Wrapper pada Java

### Pengenalan Kelas Wrapper

- Java mengenal 8 buah tipe data primitif dan tidak dapat dibuat objek. Karena hal ini mengalami kesulitan dalam penggunaannya karena beberapa class di library Java hanya dapat berinteraksi dengan objek. Untuk mengatasi masalah ini, Java menyediakan tipe data class untuk tipe data primitif. Class ini membungkus tipe data primitif agar dapat digunakan sebagai objek. Class ini disebut dengan type Wrapper.

### Fungsi Utama Kelas Wrapper

1. Menyediakan mekanisme untuk membungkus data primitif ke dalam obyek sehingga data primitif ini dapat menggunakan operasi-operasi yang khusus diperuntukkan untuk obyek, seperti penambahan ke dalam koleksi atau pengembalian dari sebuah *method* yang mengembalikan sebuah obyek
2. Menyediakan satu set fungsi *utility* untuk data primitif. Sebagian besar dari fungsi-fungsi ini berhubungan dengan konversi data primitif dari satu bentuk ke bentuk yang lainnya, sebagai contoh, mengubah data primitif ke dalam bentuk String dan sebaliknya, mengubah data primitif ke dalam

TIPE DATA PRIMITIF	KELAS WRAPPER	ARGUMEN KONSTRUKTOR
boolean	Boolean	boolean atau String
byte	Byte	byte atau String
char	Character	char
short	Short	short atau String
int	Integer	int atau String
long	Long	long atau String
float	Float	float atau String
double	Double	double atau String

- Setiap data primitif dalam Java memiliki kelas *wrapper* yang bersesuaian. Sebagai contoh, kelas *wrapper* untuk **int** adalah **Integer**, untuk **float** adalah **Float**, dan sebagainya. Yang patut diingat adalah nama dari kelas *wrapper* suatu data primitif adalah sama dengan tipe data primitif yang bersesuaian dengan huruf pertamanya dalam bentuk huruf besar, kecuali untuk tipe data **char** dengan kelas *wrapper* **Character**, dan **int** dengan kelas *wrapper* **Integer**.
- Obyek *wrapper* bersifat *immutable*, yang berarti bahwa sekali diberikan nilai pada sebuah obyek *wrapper*, nilainya tidak dapat diubah.
- Semua kelas *wrapper*, (kecuali **Character**), menyediakan dua buah konstruktor: satu konstruktor menggunakan nilai primitif yang bersesuaian, dan konstruktor yang lain menggunakan String yang merupakan representasi dari data primitif yang akan dibungkus di dalam kelas *wrapper*.
- Setiap objek wrapper mempunyai konstanta `MAX_VALUE` dan `MIN_VALUE`. `MAX_VALUE` : nilai numerik terbesar yang dapat ditampung oleh objek dengan tipe data class wrapper ini. `MIN_VALUE` : nilai numerik

terkecil yang dapat ditampung oleh objek dengan tipe data class wrapper ini.

## **SOAL & Jawaban**

### **A. SOURCE KODE JAVA**

```
/**
 *
 */
package praktikum_2;

/**
 * @author panji coy
 *
 */
public class praktikum_2 {
    static boolean bool;
    static byte by;
    static char ch;
    static double d;
    static float f;
    static int i;
    static long l;
    static short sh;
    static String str;

    /**
     * @param args
     */
    public static void main(String[] args) {
        bool = true;
        ch = 'C';
        by = 100;
        sh = 10000;
        i = 26;
        int decVal = i;
        int octVal = 032;
        int hexVal = 0X1a;
        int binVal = 0b11010;
        double d1 = 123.4;
    }
}
```

```

double d2 = 1.234e2;
float f1 = 123.4f;

System.out.println("bool = " + bool);
System.out.println("by = " + by);
System.out.println("ch = " + ch);
System.out.println("d = " + d);
System.out.println("f = " + f);
System.out.println("i = " + i);
System.out.println("l = " + l);
System.out.println("sh = " + sh);
System.out.println(" str = " + str);

System.out.println("----- ");

System.out.println("decimal = " + decVal);
System.out.println("oktal = " + octVal);
System.out.println("heksa = " + hexVal);
System.out.println("binery = " + binVal);

System.out.println("----- ");

System.out.println("d1 = " + d1);
System.out.println("d2 = " + d2);
System.out.println("f1 = " + f1);

// System.out.println("");
// TODO Auto-generated method stub

    }

}

```

Problems @ Javadoc Declaration Console

<terminated> praktikum\_2 [Java Application] C:\Program Files\Java\jre1.8.0\_25\bin\javaw.exe (May 5, 2015, 1:48:40 AM)

```

by =100
ch =C
d =0.0
f =0.0
i =26
l =0
sh =10000
str =null

-----

decimal =26
oktal =26
heksa =26
binery =26
-----

d1 =123.4
d2 =123.4
f1 =123.4

/**
 *
 */
package praktikum_2;

/**
 * @author panji coy
 *
 */
public class praktikum22 {
    static boolean bool;
    static byte by;
    static char ch;
    static double d;
    static float f;
    static int i;
    static long l;
    static short sh;
    static String str;

    /**
     * @param args
     */
    public static void main(String[] args) {

```

```

        Boolean refBool = new Boolean(true);
        boolean bool = refBool.booleanValue();
    }
}

```



```

        System.out.println("objek boolean = " + bool);
        System.out.println("-----");

        Byte refByte = new Byte((byte)123);
        byte by = refByte.byteValue();
        System.out.println("objek byte = " + by);
        System.out.println("-----");

        Character refChar = new Character('X');
        char c= refChar.charValue();
        System.out.println("objek char = " + c );
        System.out.println("-----");

        Short refShort = new Short((short)123);
        short s= refShort.shortValue();
        System.out.println("objek short = " + s );
        System.out.println("-----");

        Integer refInt = new Integer(123);
        int i= refInt.intValue();
        System.out.println("objek Integer = " + i );
        System.out.println("-----");

        Long refLong = new Long(123L);
        long l= refLong.longValue();
        System.out.println("objek Long = " + l );
        System.out.println("-----");

        Float refFloat = new Float(12.3F);
        float f = refFloat.floatValue();
        System.out.println("objek Float = " + f );
        System.out.println("-----");

        Double refDouble = new Double(12.3D);
        double d = refDouble.doubleValue();
        System.out.println("objek Integer = " + d );
        System.out.println("-----");
        // System.out.println("decimal = " + decVal);
        // TODO Auto-generated method stub

    }

}

```

```
Problems @ Javadoc Declaration Console
<terminated> praktikum22 [Java Application] C:\Program Files\Java\jre1.8.0_25\bin\javaw.exe (May 5, 2015, 1:49:10 AM)
objek boolean = true
-----
objek byte = 123
-----
objek char = X
-----
objek short = 123
-----
objek Integer = 123
-----
objek Long = 123
-----
objek Float = 12.3
-----
objek Integer = 12.3
-----

/**
 *
```