

PRAKTIKUM 1

PENDAHULUAN Pemrograman JAVA

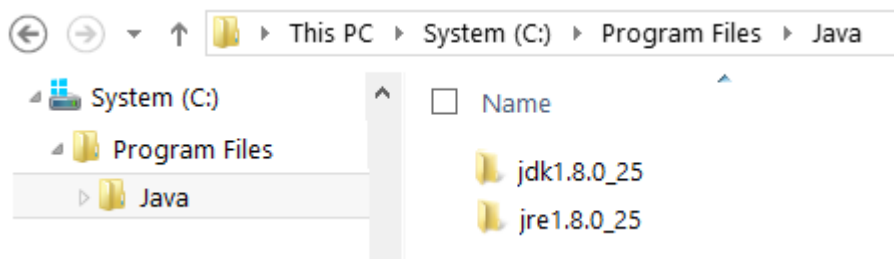
A. TUJUAN PEMBELAJARAN

1. Mengetahui dan mempersiapkan lingkungan kerja Java
2. Mengetahui penamaan identifier dan tipe dasar
3. Membuat, mengkompilasi dan menjalankan program Java
4. Memahami penggunaan berbagai macam jenis operator yang ada di Java
5. Menganalisa beberapa problem yang terjadi saat pemrograman dan memberikan solusi

B. DASAR TEORI

Bahasa pemrograman java merupakan salah satu bahasa yang bersifat *open source* yang dikeluarkan oleh Sun Microsystem (sekarang bagian dari Oracle). Informasi detail dapat dilihat pada laman <http://www.oracle.com/technetwork/java/index.html>. Java Development Kit (JDK) memuat Java Runtime Environment (JRE) dan perangkat pengembangan (seperti compiler dan debugger). JRE digunakan untuk eksekusi program java. Oleh karena itu, JDK dan JRE memiliki peran penting dalam mengkompilasi dan eksekusi program java. Java SE dapat diunduh pada tautan berikut <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Silakan lakukan instalasi dengan klik 2 kali pada program installer, contoh "jdk-8u{xx}-windows-x64.exe".

Proses instalasi JDK secara default akan tersimpan pada C:\Program Files\Java\



Selanjutnya adalah melakukan pengaturan PATH dan CLASSPATH di dalam sistem. Penyetingan PATH sangat berguna untuk memberitahu sistem dimana kita meletakkan file-file utama Java (diantaranya java, javac, jdb, dan lain-lain). Langkah pengaturan PATH environment variable pada Windows XP/Vista/7/8/10:

1. Buka "Control Panel" ⇒ "System" ⇒ klik "Advanced system settings".
2. Pilih tab "Advanced" ⇒ "Environment Variables".
3. Dalam "System Variables", cari "Path" ⇒ "Edit..."

4. Untuk Windows 10: kamu lihat tabel yang memuat PATH. Click "New" ⇒ masukan alamat "c:\Program Files\Java\jdk1.8.0_xx\bin" (ganti xx dengan versi update java!!!).

Untuk memastikan sistem anda sudah mengenali JAVA, buka CMD (klik "Start" ⇒ run... ⇒ enter "cmd"; Atau dari "Start" ⇒ All Programs ⇒ Accessories ⇒ Command Prompt. Lalu ketikan `javac -version`. Jika keluar versi java, maka sistem sudah mengenali

```
C:\Users\informatika>javac -version
javac 1.8.0_25
```

Identifier digunakan untuk memberikan nama variabel, method atau class. Penamaan identifier harus diawali dengan karakter, tanda \$ (dollar) atau tanda _ (underscore). Penamaan identifier ini bersifat case-sensitive dan tidak boleh mengandung spasi atau diawali angka.

Java mempunyai 59 kata kunci yang tidak bisa dipakai sebagai identifier termasuk di dalamnya 3 kata literal, yaitu *true*, *false* dan *null*, yang juga tidak bisa dipakai untuk penamaan identifier dan juga terdapat 2 reserved words yang tidak bisa digunakan sebagai nama identifier yaitu *const* dan *goto*.

Tabel 1. Kata kunci, literal dan reserved word

abstract	boolean	break	byte	byvalue	case
cast	catch	char	class	const	continue
default	do	double	else	extends	false
final	finally	float	for	future	generic
goto	if	implements	import	inner	instanceof
int	interface	long	native	new	null
operator	outer	package	private	protected	public
rest	return	short	static	super	switch
synchronized	this	throw	throws	transient	true
try	var	void	volatile	while	

Java mempunyai 8 tipe dasar, yaitu boolean, char, byte, short, int, long, float, dan double. Dari 8 tipe data dasar tersebut bisa dikelompokkan menjadi 4 kelompok data yaitu, integer, floating point, character, dan logical. Yang termasuk dalam kelompok data integer atau bilangan bulat adalah byte, short, int dan long. Nilai default untuk kelompok data integral adalah int. Yang termasuk dalam kelompok data floating point adalah float dan double dengan nilai default double. Sedang yang termasuk dalam tipe data character adalah char yang direpresentasikan dengan Unicode leter. Dan kelompok data yang terakhir adalah logical yaitu boolean dengan

hanya dua buah nilai saja yang diijinkan yaitu “true” dan “false”. Spesifikasi panjang bit dan range serta nilai default untuk masing-masing tipe data bisa dilihat pada Tabel 2.

Casting diperlukan untuk mengkonversi dari suatu tipe ke tipe data yang lebih kecil panjang bitnya. Sedangkan promotion terjadi pada saat mengkonversi dari suatu tipe data ke tipe data yang lebih panjang bitnya.

Contoh : `int p = (int) 10L;`

`long i = 10;`

Tabel 2 Spesifikasi panjang bit dan nilai default tipe data dasar

Tipe	Panjang bit	Range	Nilai Default
boolean	16	-	False
char	16	$0 - 2^{16}-1$	'\u0000'
byte	8	$-2^7 - 2^7-1$	0
short	16	$-2^{15} - 2^{15}-1$	0
int	32	$-2^{31} - 2^{31}-1$	0
long	64	$-2^{63} - 2^{63}-1$	0L
float	32	-	0.0F
double	64	-	0.0

Operator dapat diklasifikasikan menjadi 2 bentuk, yaitu unary operator dan binary operator. Unary operator adalah operator yang hanya melibatkan 1 operan. Sedangkan binary operator adalah operator yang melibatkan 2 operan. Java mempunyai berbagai macam jenis operator yang dapat digolongkan menjadi operator aritmatika, increment-decrement, bitwise, boolean, logik, shift (geser), penugasan, kombinasi dan kondisi.

Arithmetic operator (operator aritmatika) adalah operator yang berfungsi untuk operasi aritmatika. Yang termasuk dalam arithmetic operator bisa dilihat pada Tabel 3

Tabel 3 Arithmetic operator

Arithmetic Operator	Keterangan
+	Operasi penambahan
-	Operasi pengurangan
*	Operasi perkalian
/	Operasi pembagian
%	Operasi modulus

Increment – decrement operator adalah operator yang berguna untuk menaikkan 1 nilai (increment) dan menurunkan 1 nilai (decrement). Yang termasuk increment-decrement operator bisa dilihat pada Tabel 4. Berdasarkan urutan eksekusi penaikan dan penurunan nilainya, increment-decrement operator ini dapat diklasifikasikan menjadi 2 macam, yaitu pre-increment/decrement dan post-increment/decrement.

Tabel 4. Arithmetic operator

Increment-Decrement Operator	Keterangan
++	increment
--	decrement

Bitwise operator adalah operator yang dipakai untuk operasi bit pada nilai operan. Yang termasuk bitwise operator bisa dilihat pada Tabel 5.

Tabel 5. Bitwise Operator

Bitwise Operator	Keterangan
~	Operasi complement
&	Operasi AND
	Operasi OR
^	Operasi XOR

Boolean operator (operator boolean) adalah operator yang mengharuskan operannya bertipe boolean (true atau false). Yang termasuk boolean operator bisa dilihat pada Tabel 6.

Tabel 6 . Boolean Operator

Logical Operator	Keterangan
!	Operasi negasi (NOT)
&	Operasi AND
	Operasi OR
^	Operasi XOR
&&	Operasi AND (short circuit)
	Operasi OR (short circuit)

Operator !, &, | dan ^ mempunyai implementasi yang sama sebagaimana ketika ia menjadi bitwise operator. Hanya saja di logical operator, operan yang dilibatkan disini harus bertipe boolean, yang hanya mempunyai nilai true atau false.

Logical operator (operator logika) adalah operator yang sering dipakai untuk operasi perbandingan dan selalu menghasilkan suatu nilai bertipe boolean (true atau false). Yang termasuk logical operator bisa dilihat pada Tabel 7.

Tabel 7. Logical Operator

Logical Operator	Keterangan
= =	Operasi perbandingan sama dengan
!=	Operasi perbandingan tidak sama dengan
>	Operasi perbandingan lebih besar
>=	Operasi perbandingan lebih besar sama dengan
<	Operasi perbandingan lebih kecil
<=	Operasi perbandingan lebih kecil sama dengan

Shift operator (operator geser) adalah operator yang berfungsi untuk menggeser susunan bit pada suatu nilai. Yang termasuk dalam shift operator dapat dilihat pada Tabel 8.

Tabel 8. Shift Operator

Shift Operator	Keterangan
>>	right shift
>>>	unsigned right shift
<<	left shift

Combination operator (operator kombinasi) adalah operator yang terdiri dari gabungan 2 operator. Biasanya combination operator ini dipakai untuk mempersingkat waktu penulisan program. Yang termasuk operator combination bisa dilihat pada Tabel 9.

Conditional operator (operator kondisional) adalah operator yang dipakai untuk operasi kondisi (persyaratan), sama sebagaimana if-then-else dan hanya berlaku untuk pernyataan tunggal. Operator ini mengembalikan suatu nilai yang benar sesuai dengan kondisi yang diberikan. Conditional operator (operator kondisional) ini hanya ada 1 macam, yaitu ? disertai dengan tanda : (titik dua). Jika kondisi persyaratan yang terletak di sebelah kiri tanda ? bernilai benar, maka pernyataan yang berada di sebelah kiri tanda : yang akan diambil. Demikian juga sebaliknya, jika kondisi persyaratan bernilai salah, maka pernyataan yang berada di sebelah kanan tanda : yang akan diambil.

Tabel 9. Combination Operator

Combination Operator	Keterangan
<code>+=</code>	Gabungan dari operator <code>=</code> dan <code>+</code>
<code>-=</code>	Gabungan dari operator <code>=</code> dan <code>-</code>
<code>*=</code>	Gabungan dari operator <code>=</code> dan <code>*</code>
<code>/=</code>	Gabungan dari operator <code>=</code> dan <code>/</code>
<code>%=</code>	Gabungan dari operator <code>=</code> dan <code>%</code>
<code>>>=</code>	Gabungan dari operator <code>=</code> dan <code>>></code>
<code>>>>=</code>	Gabungan dari operator <code>=</code> dan <code>>>></code>
<code><<=</code>	Gabungan dari operator <code>=</code> dan <code><<</code>
<code>&=</code>	Gabungan dari operator <code>=</code> dan <code>&</code>
<code> =</code>	Gabungan dari operator <code>=</code> dan <code> </code>
<code>^=</code>	Gabungan dari operator <code>=</code> dan <code>^</code>

C. TUGAS PENDAHULUAN

1. Buatlah makalah yang berisi tentang perkembangan teknologi Java dan uraikan berbagai macam teknologi Java serta aplikasinya saat ini.
2. Apakah yang dimaksud dengan casting (narrowing conversion) ?
3. Apakah yang dimaksud dengan konversi (widening conversion) ?

D. PERCOBAAN

Percobaan 1 : Instalasi JDK

Pilih dan tentukan file instalasi JDK sesuai dengan operating system yang dipakai oleh komputer anda. Jalankan file instalasi JDK dan ikuti proses instalasi tahap demi tahap. Pilihlah direktori penginstallan secara default (biasanya ini akan membuat direktori baru atau `c:\jdkxxx` atau `c:\j2sdkxxx` dimana xxx adalah versi JDK yang di-install.

Percobaan 2 : Pengesetan PATH

Masuk ke system setting OS anda melalui control panel dan lakukan penambahan PATH dengan cara sebagai berikut:

```
set PATH=%PATH%;%JAVA_HOME%\bin
```

Percobaan 3 : Pengesetan CLASSPATH

Masuk ke system setting OS anda melalui control panel dan lakukan penambahan atau membuat baru CLASSPATH dengan cara sebagai berikut:

set CLASSPATH=.;%JAVA_HOME%\lib\tools.jar

Percobaan 4 : Menampilkan suatu tulisan ke layar

ApaKabar.java

```
public class ApaKabar {  
    public static void main(String args[]) {  
        System.out.println("Apa Kabar...");  
    }  
}
```

Percobaan 5 : Melibatkan class lain dalam program

Buatlah class UjiSelamat dan Selamat dalam dua file sumber berbeda.

UjiSelamat.java

```
public class UjiSelamat {  
    public static void main (String[] args) {  
        Selamat hello = new Selamat();  
        hello.greet();  
    }  
}
```

Selamat.java

```
public class Selamat {  
    public void greet() {  
        System.out.println("hi");  
    }  
}
```

Percobaan 6 : Memberikan nilai ke suatu tipe data

```
public class Assign {  
    public static void main(String args[]) {  
        boolean b = true;  
        System.out.println("Value b = " + b);  
        char c = 'C'; System.out.println("Value c = " + c);  
        byte bt = 10; System.out.println("Value bt = " + bt);  
        short s = 20; System.out.println("Value s = " + s);  
        int i = 30; System.out.println("Value i = " + i);  
        long l = 40L; System.out.println("Value l = " + l);  
        float f = 3.14F; System.out.println("Value f = " + f);  
        double d = 3.14; System.out.println("Value d = " + d);  
    }  
}
```


Percobaan 7 : Mencetak nilai default dari tipe dasar

```
public class DefaultValue { static boolean b;
    static char c;
    static byte bt; static short s; static int
    i; static long l; static float f; static
    double d;
    public static void main(String args[]) {
        System.out.println("Default value b = " + b);
        System.out.println("Default value c = " + c);
        System.out.println("Default value bt = " + bt);
        System.out.println("Default value s = " + s);
        System.out.println("Default value i = " + i);
        System.out.println("Default value l = " + l);
        System.out.println("Default value f = " + f);
        System.out.println("Default value d = " + d);
    }
}
```

Percobaan 8 : Mengamati hasil perubahan nilai dari suatu operasi matematis

```
public class Divide {
    public static void main(String[] arguments) { float
        number1 = 15;
        float number2 = 6;
        float result = number1 / number2; float
        remainder = number1 % number2;
        System.out.println(number1 + " divided by " + number2);
        System.out.println("\nResult\tRemainder");
        System.out.println(result + "\t" + remainder);
    }
}
```

Percobaan 9 : Mengamati hasil perubahan nilai dari suatu operasi matematis

```
class Invest {
    public static void main(String[] arguments) { float
        total = 14000;
        System.out.println("Original investment: $" + total);

        // Increases by 40 percent the first year total = total +
        (total * .4F); System.out.println("After one year: $" +
        total);
        // Loses $1,500 the second year
        total = total - 1500F;
        System.out.println("After two years: $" + total);
        // Increases by 12 percent the third year total =
        total + (total * .12F);
        System.out.println("After three years: $" + total);
    }
}
```

Percobaan 10 : Menampilkan bilangan oktal ke format desimal

```
public class Octal {
    public static void main(String args[]) {
        int six=06;
        int seven=07; int ight=010;
        int nine=011;
        System.out.println("Octal six = " + six);
        System.out.println("Octal seven = " + seven);
        System.out.println("Octal eight = " + eight);
        System.out.println("Octal nine = " + nine);
    }
}
```

Percobaan 11 : Menampilkan bilangan heksadesimal ke format desimal

```
public class Hexadecimal {
    public static void main(String args[]) { int
        x=0x0001;
        int y=0x7fffffff; int
        z=0xDeadCafe;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("z = " + z);
    }
}
```

Percobaan 12 : Mengamati perubahan nilai pada suatu tipe

```
public class Plus {
    public static void main(String args[]) { int x;
        int y;
        x=2147483647; //(2^31)-1
        System.out.println("x = " + x);
        y=x+1;
        System.out.println("y = " + y);
    }
}
```

Percobaan 13 : Memahami pemakaian Unicode

```
public class CobaUnicode {
    public static void main(String args[]) { ch\u0061r a='a';
        char \u0062 = 'b'; char c= '\u0063';
        String kata="\u0061\u0062\u0063";
        System.out.println("a: " + a); System.out.println("a: " + b);
        System.out.println("a: " + c); System.out.println("kata: " +
        kata);
    }
}
```

Percobaan 14 : Melakukan increment dan decrement nilai

```
class IncDec {
    public static void main (String args[]) { int x = 8, y = 13;
        System.out.println("x = " + x); System.out.println("y = "
            + y); System.out.println("x = " + ++x);
        System.out.println("y = " + y++); System.out.println("x =
            " + x--); System.out.println("y = " + --y);
    }
}
```

Percobaan 15 : Melakukan operasi bit

```
class Bitwise {
    public static void main (String args[]) { int x = 5, y = 6;
        System.out.println("x = " + x); System.out.println("y = "
            + y);
        System.out.println("x & y = " + (x & y)); System.out.println("x |
            y = " + (x | y)); System.out.println("x ^ y = " + (x ^ y));
    }
}
```

Percobaan 16 : Melakukan operasi komplemen

```
class BitwiseComplement {
    public static void main (String args[]) { int x = 8;
        System.out.println("x = " + x); int y = ~x;
        System.out.println("y = " + y);
    }
}
```

Percobaan 17 : Melakukan operasi shift

```
class Shift {
    public static void main (String args[]) {
        int x = 7;
        System.out.println("x = " + x);
        System.out.println("x >> 2 = " + (x >> 2));
        System.out.println("x << 1 = " + (x << 1));
        System.out.println("x >>> 1 = " + (x >>> 1));
    }
}
```

Percobaan 18 : Menggunakan logical operator

```
class LogicalOperator {
    public static void main (String args[]) { int x =
        7, y = 11, z = 11; System.out.println("x = "
            + x); System.out.println("y = " + y);
```

```

        System.out.println("z      =      "      +      z);
        System.out.println("x < y = " + (x < y));
        System.out.println("x > z = " + (x > z));
        System.out.println("y <= z = " + (y <= z));
        System.out.println("x >= y = " + (x >= y));
        System.out.println("y == z = " + (y == z));
        System.out.println("x != y = " + (x != z));
    }
}

```

Percobaan 19 : Menggunakan operator boolean and

```

public class BooleanAnd {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a<2) & (b++<10)) b+=2;
        System.out.println(b);
    }
}

```

Percobaan 20 : Menggunakan operator boolean and short-circuit

```

public class ShortCircuitBooleanAnd { public static void
    main(String args[]) {
        int a=5, b=7;
        if ((a<2) && (b++<10)) b+=2;
        System.out.println(b);
    }
}

```

Percobaan 21 : Menggunakan boolean or

```

public class BooleanOr {
    public static void main(String args[]) { int a=5, b=7;
        if ((a>2) | (b++<10)) b+=2;
        System.out.println(b);
    }
}

```

Percobaan 22 : Menggunakan boolean or short-circuit

```

public class ShortCircuitBooleanOr {
    public static void main(String args[]) { int a=5, b=7;
        if ((a>2) || (b++<10)) b+=2;
        System.out.println(b);
    }
}

```

Percobaan 23 : Menggunakan operator kondisi

```
class Conditional {  
    public static void main (String args[]) { int x = 0;  
        boolean isEven = false; System.out.println("x = " + x);  
        x = isEven ? 4 : 7; System.out.println("x = " + x);  
    }  
}
```

E. LATIHAN

Latihan 1 : Menganalisa dan membenahi kesalahan pada program

Tulislah program berikut ini dan simpanlah dengan nama Test.java

Test.java

```
public class Testing {  
    public static void main(String[] args) {  
        System.out.println("Ada Apa Dengan Program ini?");  
    }  
}
```

Lakukan kompilasi pada file tersebut dan amati hasilnya. Kenapa terjadi kegagalan pada saat kompilasi?. Benahilah kesalahan di atas sehingga program tersebut dapat berjalan dengan baik.

Latihan 2 : Menganalisa dan membenahi kesalahan pada program

Tulislah program dibawah ini dan simpanlah dengan nama tertentu sesuai pilihan anda.

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("Ada Apa Dengan Program ini?");  
    }  
}  
  
public class TestAnother {  
    public static void main(String[] args) {  
        System.out.println("Ada Apa Dengan Program ini?");  
    }  
}
```

Lakukan kompilasi pada file tersebut dan amati hasilnya. Kenapa terjadi kegagalan pada saat kompilasi? Benahilah kesalahan diatas sehingga program tersebut dapat berjalan dengan baik.

Latihan 3 : Menganalisa dan membenahi kesalahan pada program

Tulislah program berikut ini dan simpanlah.

```
public class Test {
    public static void main(String args) {
        System.out.println("Ada Apa Dengan Program ini?");
    }
}
```

Lakukan kompilasi pada program tersebut dan jalankan. Kenapa terjadi kesalahan pada saat menjalankan program tersebut. Benahilah kesalahan diatas sehingga program tersebut dapat berjalan dengan baik.

Latihan 4 : Menganalisa dan membenahi kesalahan pada program

Tulislah program berikut ini dan simpanlah.

```
public class Test {
    public void main(String args[]) {
        System.out.println("Ada Apa Dengan Program ini?");
    }
}
```

Lakukan kompilasi pada program tersebut dan jalankan. Kenapa terjadi kesalahan pada saat menjalankan program tersebut. Benahilah kesalahan diatas sehingga program tersebut dapat berjalan dengan baik.

Latihan 5 : Membuat formulasi proses casting dari tipe-tipe primitif

Lakukan percobaan casting antar tipe-tipe primitif. Lalu amati dan catatlah konversi dari suatu tipe ke tipe lainnya yang memerlukan suatu casting. Dari hasil analisa anda, buatlah suatu skema casting diantara tipe-tipe primitif.

Latihan 6 : Membuat formulasi proses promotion dari tipe-tipe primitif

Lakukan percobaan promotion antar tipe-tipe primitif. Lalu amati dan catatlah konversi dari suatu tipe ke tipe lainnya yang menyebabkan terjadinya suatu promotion. Dari hasil analisa anda, buatlah suatu skema promotion diantara tipe-tipe primitif.

Latihan 7 : Menampilkan representasi biner dari bilangan desimal bertipe int

Buatlah suatu program untuk menampilkan susunan bit dari suatu bilangan desimal. Nilai bilangan input yang dimasukkan adalah bertipe **int**.

Contoh tampilan:

Masukkan nilai desimal: 13

[illegible]

Masukkan nilai desimal: 612

Susunan bit dari 13 adalah 0000000000000000000000001001100100

Masukkan nilai desimal: -1

Susunan bit dari 13 adalah 11111111111111111111111111111111

Masukkan nilai desimal: -13

Susunan bit dari 13 adalah 11111111111111111111111111110011

F. TUGAS

Tugas 1 : Menghitung luas dan keliling lingkaran

Buatlah program untuk menghitung luas dan keliling lingkaran. Untuk nilai PI gunakan konstanta PI yang ada di java.lang.Math.PI

Tugas 2 : Mengkonversi suatu nilai dari Celcius ke Fahrenheit atau sebaliknya

Buatlah suatu program untuk mengkonversi suatu nilai dari Celcius ke Fahrenheit atau sebaliknya.

$$\text{Fahrenheit} = \frac{\text{Celcius} \times 9}{5} + 32$$

Rumus :

Tugas 3 : Menganalisa batasan maksimum dari suatu tipe

Amatilah dan tulislah program berikut ini:

```
public class BigInteger {  
    public static void main(String args[]) {  
        long p=2147483648;  
    }  
}
```

Lakukan kompilasi pada file tersebut dan amati pesan kesalahannya. Lakukan analisa mengapa bisa terjadi kesalahan padahal batasan nilai maksimum dari suatu bilangan bertipe long adalah $2^{63}-1$ (9223372036854775807)?. Kemudian berikanlah solusi yang tepat untuk mengatasi persoalan diatas.

Tugas 4 : Mencari panjang menit dari durasi waktu

Input: jam awal, menit awal, jam akhir, menit akhir

Output: durasi waktu (dalam menit) antara jam awal menit awal dengan jam akhir menit akhir.

Tugas 5 : Mencari representasi biner dari suatu bilangan

Tuliskan representasi bit dari nilai -19? Jelaskan.

Tugas 6 : Menganalisa pergeseran bit dari operasi shift

Jelaskan apa yang terjadi pada potongan program berikut ini:

```
Byte a=-1;
a=(byte) (a >>> 2);
```

Lampiran

Cara memasukkan input melalui JOptionPane .

```
import javax.swing.JOptionPane;
public class InputPane {
    public static void main(String args[]) {
        int nilai;
        String str = JOptionPane.showInputDialog("Masukkan nilai :");
        nilai = Integer.parseInt(str); System.out.println(nilai);
        System.exit(0);
    }
}
```

Cara memasukkan input melalui Scanner .

```
import java.util.Scanner;
public class Halo{
    public static void main(String args[]){
        Scanner userInput = new Scanner(System.in);
        System.out.print("Masukan nama Anda : ");
        String nama=userInput.next();
        System.out.println("Halo "+nama);
    }
}
```

LAPORAN RESMI

Kumpulkan hasil percobaan di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat. Setiap program ditambah author.

Contoh :

Percobaan 4

Analisa : mencetak tulisan Apa kabar

Hasil percobaan:

```
public class ApaKabar {
    public static void main(String args[]) {
        System.out.println("Author : 130303160021 Jamal");
        System.out.println("Apa Kabar...");
    }
}
```

```
C:\Users\informatika\Google Drive\
>java ApaKabar
Author : 130303160021 Jamal
Apa Kabar...
```