

IF34348 - PEMROGRAMAN LANJUT

TIPE DATA DAN VARIABLE

03

Oleh : Andri Heryandi, M.T.

TIPE DATA PRIMITIF

IF34348 - Pemrograman Lanjut

- **Tipe Bilangan Bulat** (`byte`, `short`, `int`, **dan** `long`)
- **Tipe Bilangan Pecahan** (`float` **dan** `double`)
- **Tipe Text** (`char`)
- **Tipe Logika** (`boolean`)



Oleh : Andri Heryandi, M.T.

TIPE BILANGAN BULAT

IF34348 - Pemrograman Lanjut

Tipe	Panjang	Range	Contoh
byte	8 bit	-2^7 to $2^7 - 1$ (-128 to 127)	2 -114 0b10 (biner)
short	16 bit	-2^{15} to $2^{15} - 1$ (-32,768 s.d 32,767)	2 -32699
int (Default)	32 bit	-2^{31} to $2^{31} - 1$ (-2,147,483,648 to 2,147,483,647)	2 147334778 123_456_678
long	64 bit	-2^{63} to $2^{63} - 1$ (-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)	2 -2036854775808L 1L

- Gunakan akhiran huruf “l” atau “L” jika ingin mengisi variable bertipe long.

Oleh : Andri Heryandi, M.T.



TIPE BILANGAN BULAT

IF34348 - Pemrograman Lanjut

```
public class TestInteger {  
    public static void main(String[] args) {  
        byte b;  
        short s;  
        int i;  
        long l;  
        b=120;  
        s=32767;  
        i=1_000_000_000;  
        l=900000000000L;  
        System.out.println("byte    : "+b);  
        System.out.println("short   : "+s);  
        System.out.println("int     : "+i);  
        System.out.println("long    : "+l);  
        System.out.println("byte    : 0b"+Integer.toBinaryString(b));  
        System.out.println("short   : 0"+Integer.toOctalString(s));  
        System.out.println("int     : 0x"+Integer.toHexString(i));  
    }  
}
```

Underscore (_) digunakan sebagai pemisah ribuan.



Oleh : Andri Heryandi, M.T.

TIPE DATA PECAHAN

IF34348 - Pemrograman Lanjut

Tipe	Panjang	Contoh
float	32 bit	99F -32745699.01F 4.2E6F (notasi untuk $4.2 * 10^6$)
double (Default)	64 bit	-1111 2.1E12 99970132745699.999

- Gunakan akhiran “F” jika anda ingin mengisi nilai berupa float. Jika sebuah angka pecahan tidak diakhiri dengan “F” maka akan dianggap bertipe double.

TIPE BILANGAN PECAHAN

IF34348 - Pemrograman Lanjut

```
public class TestPecahan {  
    public static void main(String[] args) {  
        float f;  
        double d;  
        f=1234567890.123456789F;  
        d=1_234_567_890.123456789;  
        System.out.println("Float      : "+f);  
        System.out.println("Double    : "+d);  
    }  
}
```

Perhatikan akurasi
tiap tipe data



TIPE DATA TEKS

IF34348 - Pemrograman Lanjut

- Satu-satunya tipe data primitif teks adalah `char`.
- Digunakan untuk sebuah karakter (16 bit).
- Contoh :

```
public char jenisKelamin= 'L';
```



Oleh : Andri Heryandi, M.T.

TIPE BILANGAN TEKS

IF34348 - Pemrograman Lanjut

```
public class TestChar {  
    public static void main(String[] args) {  
        char c;  
        char tab='\t';  
        char newline='\n';  
        c='A';  
        System.out.println("Hasil "+newline+"Char : "+  
tab+"berisi "+c);  
    }  
}
```

Simbol Lain :

\' = simbol kutip 1 (')

\" = simbol kutip 2 (")

\\ = simbol backslash (\)



TIPE DATA LOGIKA

IF34348 - Pemrograman Lanjut

- Tipe data untuk menampung data logika bisa menggunakan tipe `boolean`.
- Hanya dapat menampung nilai `true` atau `false`.



Oleh : Andri Heryandi, M.T.

TIPE BILANGAN LOGIKA

IF34348 - Pemrograman Lanjut

```
public class TestBoolean {  
    public static void main(String[] args) {  
        int umur=19;  
        boolean dewasa;  
        dewasa= (umur >= 17);  
        System.out.println("Status Dewasa : "+dewasa);  
    }  
}
```



PENAMAAN VARIABLE

IF34348 - Pemrograman Lanjut

■ Aturan

- Nama variable harus diawali oleh huruf, garis_bawah (_) atau tanda dolar (\$).
- Tidak boleh memiliki spasi, atau tanda baca
- Tidak boleh menggunakan keyword java, di bawah ini

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

* not used

** added in 1.2

*** added in 1.4

Olel **** added in 5.0



PENAMAAN VARIABLE

IF34348 - Pemrograman Lanjut

Panduan lain dalam penulisan variable adalah :

- Awali nama variable dengan huruf kecil. Untuk kata ke dua dan selanjutnya awali dengan huruf besar. (contoh: `myVariable`).
- Pilihlah nama yang mengindikasikan isi variabelnya. Jangan hanya x, y atau z.



PENGATURAN TAMPILAN ISI VARIABLE

IF34348 - Pemrograman Lanjut

Pengaturan tampilan isi variable bisa dilakukan dengan menampilkan data dengan menggunakan perintah `System.out.printf`. Format selalu diawali dengan tanda `%`. Format yang bisa digunakan adalah :

Format	Keterangan
d	Format untuk bilangan bulat
f	Format untuk bilangan pecahan
x.y	Format menampilkan pecahan dimana x adalah angka yang menunjuk berapa digit yang akan ditampilkan, sedangkan y adalah berapa digit pecahan yang akan ditampilkan.
+	Format untuk menampilkan tanda positif atau negatif
,	Format untuk menampilkan kelompok ribuan.
n	Format untuk menambahkan new line (pindah baris)
-	Format untuk menampilkan data terformat rata kiri.



PENGATURAN TAMPILAN ISI VARIABLE

IF34348 - Pemrograman Lanjut

```
import java.util.Locale;

public class TestFormating {
    public static void main(String[] args) {
        int i=3546764;
        int iminus=-i;
        System.out.println("i                : "+i);
        System.out.printf("%%d                : %d%n",i);
        System.out.printf("%%10d             : %10d%n",i);
        System.out.printf("%%+10d          : %+10d%n",i);
        System.out.printf("%%+10d          : %+10d%n",iminus);
        System.out.printf("%%,10d          : %,10d%n",i);
        System.out.printf("%%-10d          : %-10d%n",i);
        double f=5675482.982;
        System.out.println("f                : "+f);
        System.out.printf("%%f            : %f%n",f);
        System.out.printf("%%.2f          : %.2f%n",f);
        System.out.printf("%%12.2f         : %12.2f%n",f);
        System.out.printf("%%,12.2f        : %,12.2f%n",f);
        System.out.printf(new Locale("in","id"),"%%,12.2f      : %,12.2f%n",f);
    }
}
```



Oleh : Andri Heryandi, M.T.

OPERATOR-OPERATOR

IF34348 - Pemrograman Lanjut

- Operator Aritmatika
- Operator Relasional
- Operator Bit
- Operator Logika
- Operator Penugasan



Oleh : Andri Heryandi, M.T.

OPERATOR ARITMATIKA

IF34348 - Pemrograman Lanjut

■ Operator Aritmatika

- + Penambahan
- - Pengurangan
- * Perkalian
- / Pembagian
- % Modulus (siswa hasil bagi)
- ++ Penambahan 1
- -- Pengurangan 1



Oleh : Andri Heryandi, M.T.

OPERATOR ARITMATIKA

IF34348 - Pemrograman Lanjut

```
public class OprAritmatika {  
    public static void main(String args[]) {  
        int a = 10;  
        int b = 20;  
        int c = 25;  
        int d = 25;  
        System.out.println("a + b = " + (a + b) );  
        System.out.println("a - b = " + (a - b) );  
        System.out.println("a * b = " + (a * b) );  
        System.out.println("b / a = " + (b / a) );  
        System.out.println("b % a = " + (b % a) );  
        System.out.println("c % a = " + (c % a) );  
        System.out.println("a++      = " + (a++) );  
        System.out.println("b--      = " + (a--) );  
        System.out.println("d++      = " + (d++) );  
        System.out.println("++d      = " + (++d) );  
    }  
}
```



Oleh : Andri Heryandi, M.T.

OPERATOR RELASIONAL

IF34348 - Pemrograman Lanjut

■ Operator-Operator relasional

- == Perbandingan sama dengan
- > Perbandingan lebih besar dari
- < Perbandingan lebih kecil dari
- >= Perbandingan lebih besar atau sama dengan dari
- <= Perbandingan lebih kecil atau sama denan dari
- != Perbandingan tidak sama dengan



OPERATOR RELASIONAL

IF34348 - Pemrograman Lanjut

```
public class OprRelasional {  
    public static void main(String args[]) {  
        int a = 10;  
        int b = 20;  
        System.out.println("a == b = " + (a == b) );  
        System.out.println("a != b = " + (a != b) );  
        System.out.println("a > b = " + (a > b) );  
        System.out.println("a < b = " + (a < b) );  
        System.out.println("b >= a = " + (b >= a) );  
        System.out.println("b <= a = " + (b <= a) );  
    }  
}
```



OPERATOR BIT

IF34348 - Pemrograman Lanjut

- Operator bit digunakan untuk melakukan perhitungan secara bit. Operator yang digunakan :
 - `&` : Operasi and
 - `|` : Operasi or
 - `^` : Operasi xor
 - `~` : Operasi not
 - `<<` : Operator shift left (geser bit ke kiri)
 - `>>` : Operasi shift right (geser bit kanan)



OPERATOR BIT

IF34348 - Pemrograman Lanjut

```
public class OprBit {  
    public static void main(String args[]) {  
        int a = 60;          /* 60 = 0011 1100 */  
        int b = 13;          /* 13 = 0000 1101 */  
        int c = 0;  
        c = a & b;            /* 12 = 0000 1100 */  
        System.out.println("a & b = " + c );  
        c = a | b;            /* 61 = 0011 1101 */  
        System.out.println("a | b = " + c );  
        c = a ^ b;            /* 49 = 0011 0001 */  
        System.out.println("a ^ b = " + c );  
        c = ~a;               /* -61 = 1100 0011 */  
        System.out.println("~a = " + c );  
        c = a << 2;           /* 240 = 1111 0000 */  
        System.out.println("a << 2 = " + c );  
        c = a >> 2;           /* 15 = 0000 1111 */  
        System.out.println("a >> 2 = " + c );  
    }  
}
```



OPERATOR LOGIKA

IF34348 - Pemrograman Lanjut

- Operator logika digunakan untuk melakukan operasi logika seperti AND, OR dan NOT. Adapun operator yang digunakan adalah :
 - && : Operasi logika AND
 - || : Operasi logika OR
 - ! : Operasi logika NOT



OPERATOR LOGIKA

IF34348 - Pemrograman Lanjut

```
public class OprLogika {  
    public static void main(String args[]) {  
        boolean a = true;  
        boolean b = false;  
        System.out.println("a && b      = " + (a&&b));  
        System.out.println("a || b      = " + (a||b));  
        System.out.println("!a          = " + !a);  
        System.out.println("! (a && b) = " + !(a && b));  
    }  
}
```



OPERATOR PENUGASAN

IF34348 - Pemrograman Lanjut

- **Operator penugasan adalah operator yang bertugas untuk melakukan pengisian data. Operatornya adalah :**
 - **= : Pengisian nilai**
 - **+=, -=, *=, /=, %= : Operasi aritmatika dengan nilai tertentu**
 - **^=, !=, &=, |= : Operasi bit dengan nilai tertentu**
 - **<<=, >>= : Operasi geser bit dengan nilai tertentu**



OPERATOR PENUGASAN

IF34348 - Pemrograman Lanjut

```
public class OprPenugasan {  
    public static void main(String args[]) {  
        int a = 10;  
        int b = 20;  
        int c = 0;  
        c = a + b;  
        System.out.println("c = a + b = " + c );  
        c += a ;  
        System.out.println("c += a  = " + c );  
        c -= a ;  
        System.out.println("c -= a = " + c );  
        c *= a ;  
        System.out.println("c *= a = " + c );  
        a = 10;  
        c = 15;  
        c /= a ;  
        System.out.println("c /= a = " + c );  
    }  
}
```



OPERATOR PENUGASAN

IF34348 - Pemrograman Lanjut

```
a = 10;
c = 15;
c %= a ;
System.out.println("c %= a = " + c );
c <<= 2 ;
System.out.println("c <<= 2 = " + c );
c >>= 2 ;
System.out.println("c >>= 2 = " + c );
c >>= a ;
System.out.println("c >>= a = " + c );
c &= a ;
System.out.println("c &= 2 = " + c );
c ^= a ;
System.out.println("c ^= a = " + c );
c |= a ;
System.out.println("c |= a = " + c );
}
}
```



LATIHAN 1

IF34348 - Pemrograman Lanjut

Sebuah perusahaan mempunyai peraturan perhitungan gaji sebagai berikut :

- Tunjangan Keluarga (TK) : 20% dari gaji Pokok
- Tunjangan Anak (TA) : 5% dari gaji pokok / anak
- Gaji Kotor : Gaji Pokok + TK + TA
- Pajak : 15% dari gaji kotor
- Gaji Bersih : Gaji Kotor – Pajak

Hitunglah gaji seorang pegawai jika gaji pokoknya 1.000.000 dengan banyak anak adalah 3 orang.

Tampilkan tunjangan keluarga, tunjangan anak, gaji kotor, pajak dan gaji bersih. Tampilkan datanya dengan rapi rata kanan (currency) lengkapi dengan simbol Rp. dan terformat ribuan.



CONTOH HASIL RUN

LATIHAN 1

IF34348 - Pemrograman Lanjut

Gaji Pokok	:	Rp.	1.000.000
Banyak Anak	:		3
Tunjangan Keluarga	:	Rp.	200.000
Tunjangan Anak	:	Rp.	150.000
Gaji Kotor	:	Rp.	1.350.000
Pajak (15%)	:	Rp.	202.500
Gaji Bersih	:	Rp.	1.147.500



Oleh : Andri Heryandi, M.T.

LATIHAN 2

IF34348 - Pemrograman Lanjut

- Hitunglah lama waktu rental komputer. Biaya rental per jam adalah 5000.
- Contoh :
 - Jam Masuk : 9
 - Menit Masuk : 50
 - Jam Keluar : 12
 - Menit Keluar : 30

Output :

Lama Rental adalah : 2 Jam 40 menit (2.67 Jam) dengan Biaya Rp. 13.350

