

NOTES

```
Class Hello World {  
    Public Static Void Main (String[] a){  
        System.out.println("Hello World") ;  
    }  
}
```

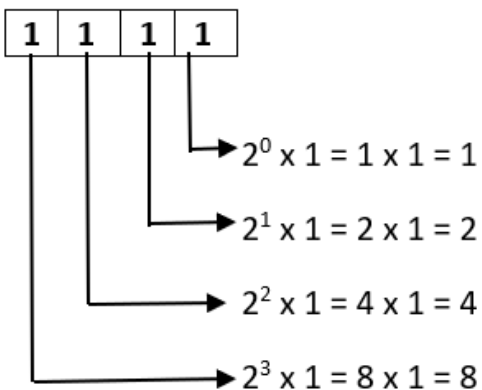
===== Tidak semua class pake, hanya agar computer tau yang mana yang harus dirun dulu. Public Static Void = Agar bisa dirun (secara singkat).

- Selalu teliti Syntax sama Besar Kecil => Syntax error or Compiler error = Ada problem di syntaxnya atau di compiler
- Parser = Membaca kode dan urutan membacanya dari kiri ke kanan dan dari atas ke bawah
- Variabel = Jadi container untuk data yang biasanya berubah - ubah
- Tipe data INTEGER obv **GAK BISA** masukin data STRING dan gak bisa diubah jadi STRING
- Fungsi Tipe Data =
 - Bisa ngasik tau computer berapa memori yang harus dialokasikan untuk data tersebut
 - Biar bisa ngasik tau harus ditaruh dimana (currently ini ditaruh di Stack)
 - Stack = Operational sehari - sehari secara cepat, Isinya Primitive Data Type
 - Heap = For Non Primitive data type
 - Ngasik tau data, apa aja hitungan yang bisa kamu lakukan
- Garbage Collection = Membersihkan alokasi memori yang tidak terpakai
- High Level Language (R, Java ,dkk) = Punya fungsionalitasnya banyak
- Astrix (*) = Kali
- / = Cuma mbagi bagian angka yang bulat, tapi kalo kamu kasik float ya hasilnya decimal juga
- **Batas angka INT -> -2^{31} to $2^{31} - 1$** = Kapasitasnya 32 bit (semua bit dilist = pangkat untuk batas angka - 1)
 - Knp kok gitu jadinya cuma 31 ?
 - Bit terbatas = Kode plus minus (0 = Positif, 1 = Negatif)
 - Ada - 1 nya karena ada perhitungan dimulai dari 0
- Double = More accurate than Float
- Koma = Pake titik JANGAN LUPA !!!
- Hexadecimal = Basisnya 16 (f = 15)

LIST PRIMITIVE DATA TYPE

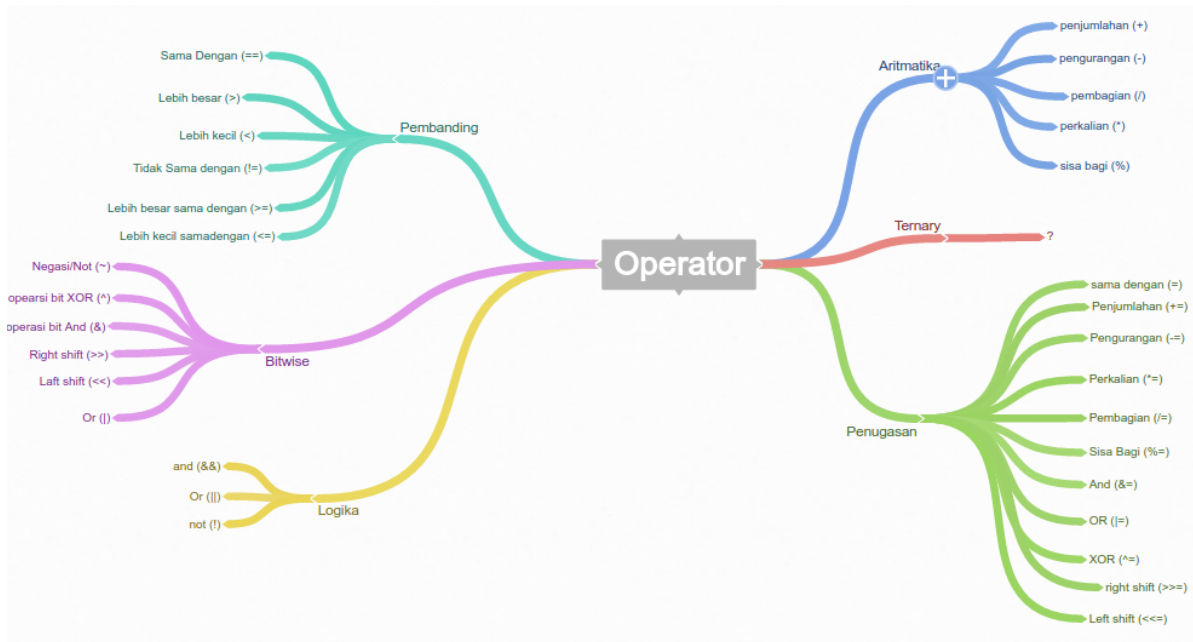
TYPE	DESCRIPTION	DEFAULT	SIZE	EXAMPLE LITERALS	RANGE OF VALUES
boolean	true or false	false	1 bit	true, false	true, false
byte	twos complement integer	0	8 bits	(none)	-128 to 127
char	unicode character	\u0000	16 bits	'a', '\u0041', '\101', '\l', '\l', '\n', '\b'	character representation of ASCII values 0 to 255
short	twos complement integer	0	16 bits	(none)	-32,768 to 32,767
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2	-2,147,483,648 to 2,147,483,647
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F	upto 7 decimal digits
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d	upto 16 decimal digits

- How many Unicode di char ? 65,535 => hexadecimal
- Char vs Short = Mewakili hal yang berbeda dan batasnya beda (Char dikonversikan jadi huruf sedangkan Short jadi bilangan bulat biasa)

HOW TO READ BINARY (Mungkin berguna untuk memahami gambar ibunya)

Resultant decimal number= $1+2+4+8 = 15$

- How to declare
String foo;
int x;
long aLongNumber;
boolean **isStudent**;
- Camel Case : Huruf untuk kata berikutnya dibesarin, no _ (T^T TIDAKKK)[BERLAKU UNTUK SEGALA PENAMAAN, biasanya class diawali pake huruf besar]
- **OPERATORS IN JAVA**



OPERATOR	USAGE EXAMPLE	RETURNS "TRUE" VALUE IF...
>	a > b	a is greater than b
>=	a >= b	a is greater than or equal to b
<	a < b	a is less than b
<=	a <= b	a is less than or equal to b
==	a == b	a is equal to b
!=	a != b	a isn't equal to b
&&	a && b	a and b are true values, b is evaluated relatively (if a is false, b isn't evaluated)
	a b	a or b is true, b is evaluated relatively (if a is true, b isn't evaluated)
!	!a	a is false
&	a & b	a and b are true, b is evaluated in any case
	a b	a or b is true, b is evaluated in any case

(&& dan ||, yang di depan dicek dulu, kalau false, gak lanjut ke belakang, kalau true, baru dicek yang belakang)

(& dan |, langsung dicek dua - duanya)

- BILANGAN BULAT PASTI INT
- Kalo operator keduanya bulat, hasilnya bulat. Kalo salah satu decimal, nanti hasilnya decimal
- Kalau kamu declare secara eksplisit tipe datanya, Kalau kamu tulis `Float a = 2 => 5 / 2` jawabannya 2
- Kalo ndak kamu declare dan kamu tulis angka decimal => Double (DEFAULT FOR ALL DECIMALS)
- Btw ndak ada pembulatan, jadi kayak $5/2 = 2$, bukan 3 :D
- Kalau mau nulis package, tulisannya harus huruf kecil semua
- Kalau mau nulis comment, pake `/* blab la blab la */`

```
System.out.println(5<<1); /* 5 geser 1 bit ke kiri */
System.out.println(5>>1); /* 5 geser 1 bit ke kanan */
```
- Kalo masih bingung bisa dibayangkan pake hitungan binary yang tadi
- Operan = angkanya, Operator = Ya + - / *
- Khusus Operator Manipulasi Bit or Bitwise, Operan di depan adalah objek yang dikenai, operan belakang = berapa banyak
 - Shift right 1 = Dibagi 2
 - Shift right 2 = Dibagi 4
 - Shift right 3 = Dibagi 8
 - Shift right 4 = Dibagi 16
 - Shift left 1 = Dikalikan 2
 - Shift left 2 = Dikalikan 4
 - Shift left 3 = Dikalikan 8
 - Shift left 4 = Dikalikan 16
- Terkadang kalau hasil operasinya terlalu besar jadinya ngebug, kita pakai manipulasi bit ini agar gak ngebug dan lebih mudah ditranslate sama computer
- Assignment Operator
 - Nilai yang di **kanan dimasukkan ke kiri**
 - Jadi kalau `a = b => sama` 5, tapi kalau `b = a => sama` 3
 - `(a = b) == (b = a) ?? False`
 - `==` itu operator logika dan jawabannya sebatas TRUE or FALSE
 - Kalau untuk masukin nilai ke variable, pake `=`
- Operator Precedence
 - Kayak aturan ngitung matematika !
 - Tanda kurung dulu, trus kali bagi, baru + -
 - Kalau ada yang sama, diitung dari kiri ke kanan
- Kalau tipe data string, trus kamu kasih operator +, nanti langsung kegabung tanpa spasi. Jadi kamu harus tambahkan " " di antara kedua variable itu
- Kalau kamu `String + Int`, ya hasilnya `StringInt`

- Ah yea, itu kata + angka + angka, sama java masih dianggep string semua, bukan int. Jadi kamu perlu declare atau kamu hitung dulu dalam variable :D
- Bedakan variable dan value ya 😊
-

CODE OF THE DAY

```
public class operator {  
    no usages  
    public static void main (String[] args){  
        System.out.println(5/2);  
        System.out.println(5.0/2.0);  
        System.out.println(5.0/2);  
        System.out.println(5/2.0);  
  
        float b = 2;  
        System.out.println(5/b);  
  
        System.out.println(true & false);  
        System.out.println(true | false);  
        System.out.println(true && false);  
        System.out.println(true || false);  
  
        System.out.println(2 == 2);  
        System.out.println(2 != 2);  
        System.out.println(!false);  
  
        System.out.println(5<<1); /* 5 geser 1 bit ke kiri */  
        System.out.println(5>>1); /* 5 geser 1 bit ke kanan */  
  
        /* cara buat comment pakek ini boss */  
    }  
}
```

```
String c = "saya";  
String d = "makan";  
  
System.out.println(c + " " + d);  
System.out.println(c + " " + 3);
```

```
int a = 3;  
int b = 5;  
  
a = b;  
b = a;  
  
System.out.println(a);  
System.out.println(b);
```

```
int e = 3;  
int f = 5;  
System.out.println(c + " " + e + f);  
System.out.println(c + " " + (e+f));  
System.out.println(e + f + " " + c);
```

```
saya 35  
saya 8  
8 saya
```