



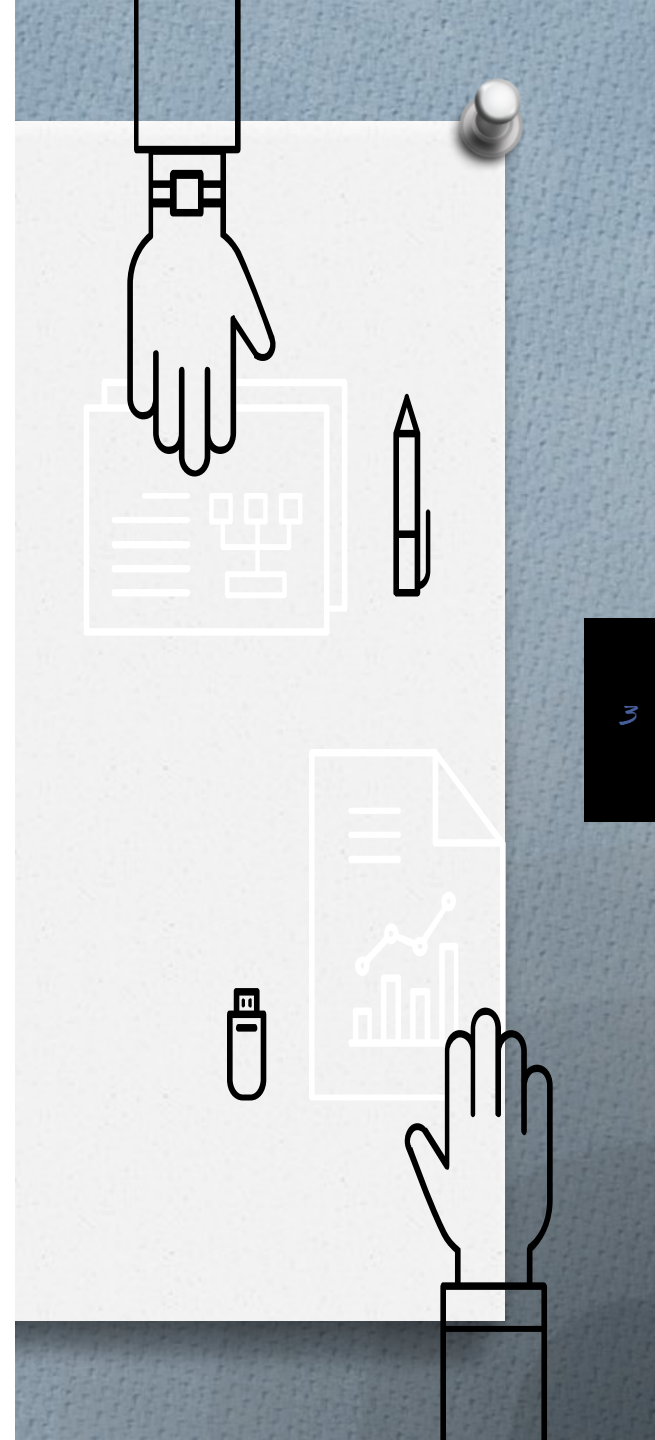
Tipe Data, Variabel, Operator

Elemen Dasar

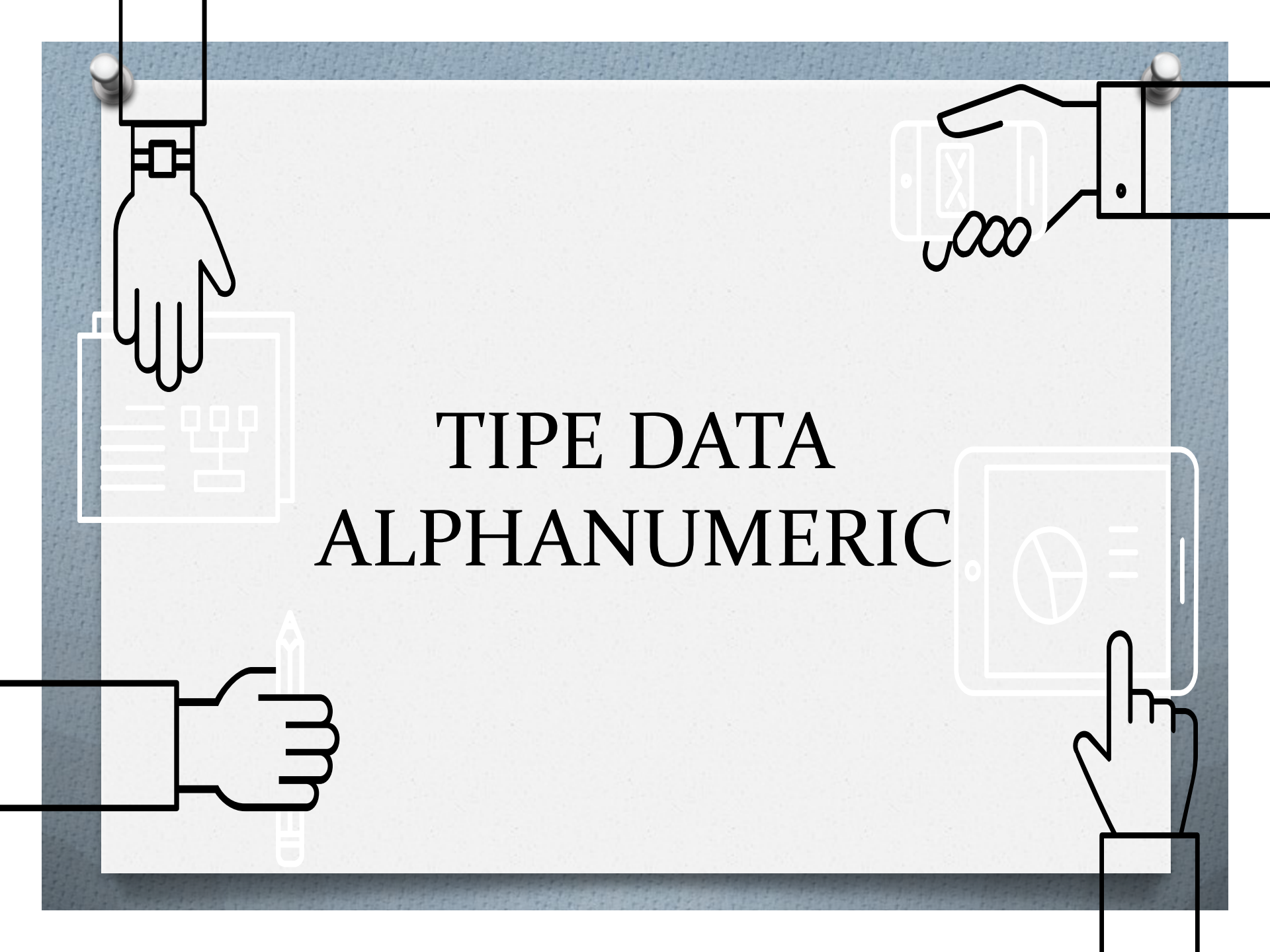
- *Identifier* (pengenal) adalah suatu nama yang biasa dipakai dalam pemrograman untuk menyatakan variabel, konstanta bernama, tipe data, fungsi, label, objek. Identifikasi dilakukan untuk mempermudah proses penanganan data atau nilai.
- Ketentuan penulisan identifier:
 - a. Identifier tidak boleh diawali dengan karakter yang berupa angka.
 - b. Identifier tidak boleh mengandung spasi.
 - c. Identifier tidak boleh menggunakan karakter-karakter simbol (#, @, ?, \$, dll).
 - d. Identifier tidak boleh menggunakan kata kunci (keyword) yang terdapat pada bahasa pemrograman
 - e. Nama identifier sebaiknya disesuaikan dengan kebutuhannya

TIPE DATA JAVA

- ▷ Java mempunyai dua jenis tipe data :
- ▷ 1. Tipe Data Primitive
- ▷ 2. Tipe Data Buatan
- ▷ Tipe data primitive dibedakan menjadi 3 :
- ▷ Tipe data Alphanumeric
- ▷ Tipe data alphabetic
- ▷ Tipe data boolean

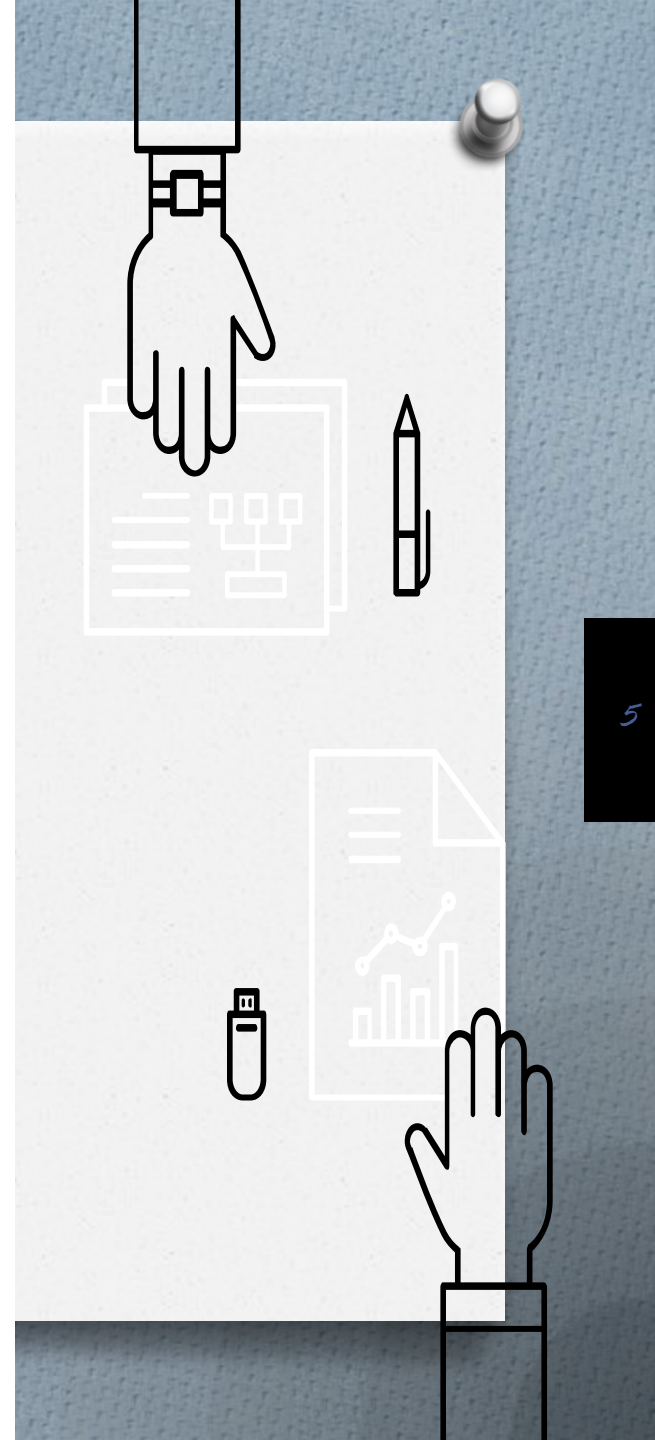


TIPE DATA ALPHANUMERIC



BILANGAN BULAT

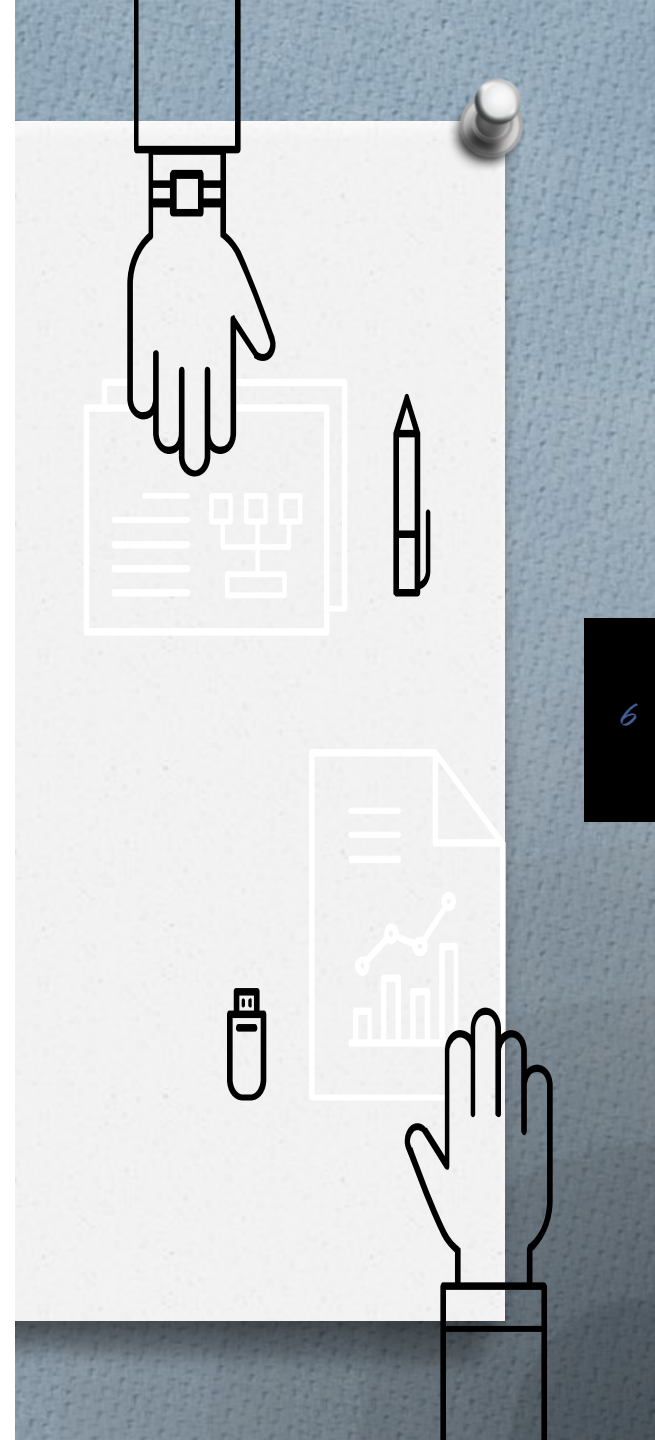
Tipe data yang masuk menjadi bagian ini adalah byte, short, int dan long. Semua tipe data ini bersifat *Signed*, yaitu bisa mempresentasikan nilai positif dan negatif.



BYTE

Type *byte* umumnya digunakan pada saat kita bekerja dengan sebuah data *stream* dari suatu file maupun jaringan, yaitu untuk keperluan proses membaca/menulis. Selain itu, tipe ini juga digunakan saat bekerja dengan data biner yang tidak kompatibel dengan tipe-tipe lain yang didefinisikan di dalam Java.

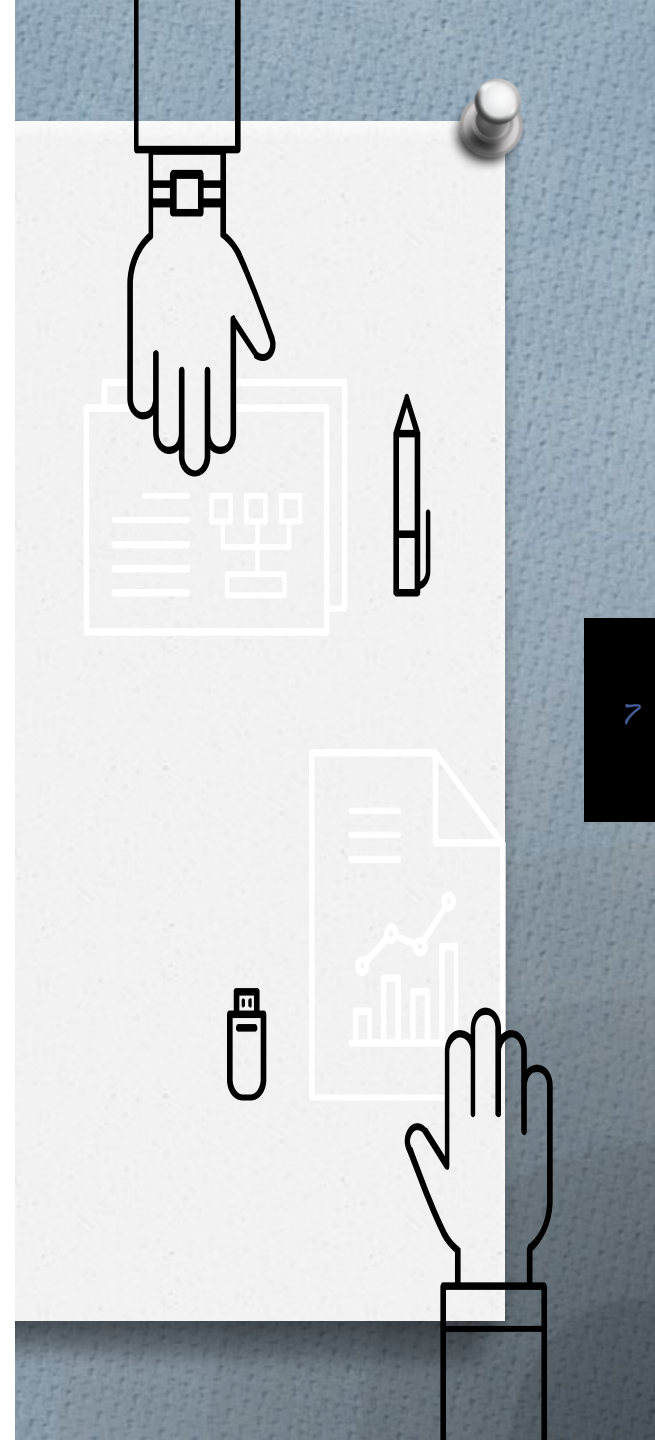
Range nilai : -128 sampai 127



SHORT

Pada umumnya diaplikasikan pada komputer-komputer 16-bit, yang saat ini semakin jarang keberadaanya.

Range nilai : -32768 sampai 32767



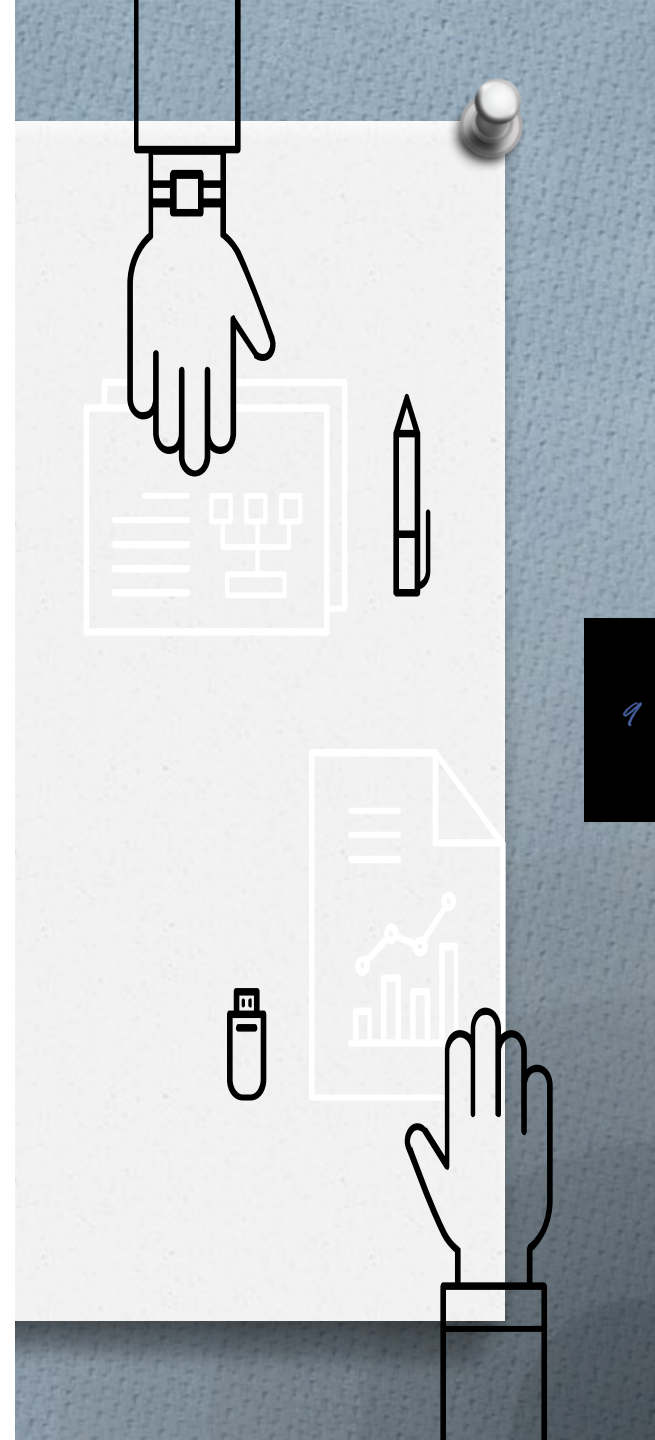
INT

- ▷ Tipe ini merupakan tipe yang paling banyak dipakai dalam merepresentasikan angka dalam Java, dikarenakan dianggap paling efisien dibandingkan dengan tipe-tipe integer lainnya.
- ▷ Range nilai :
-2147483648 sampai 2147483647



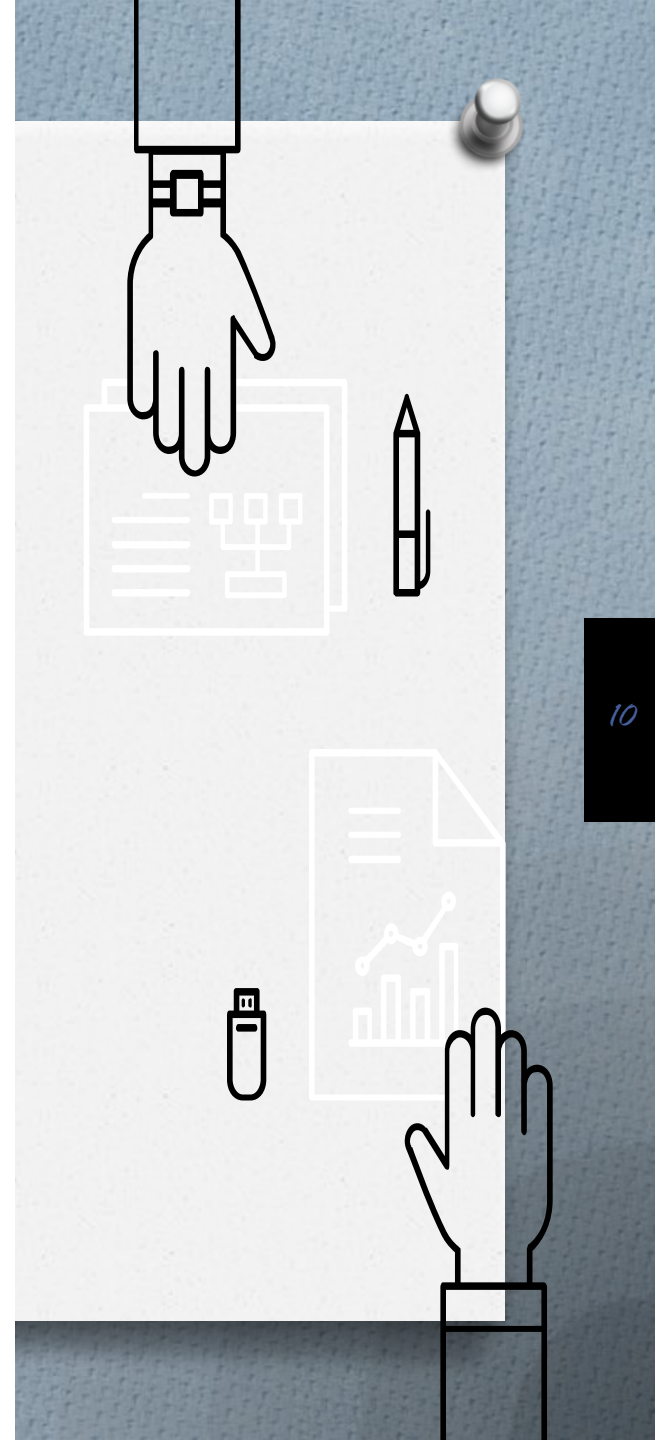
INT

- ▷ Tipe *Int* banyak digunakan untuk indeks dalam struktur pengulangan maupun dalam konstruksi sebuah *array*. Selain itu, secara teori setiap ekspresi yang melibatkan tipe integer *byte*, *short*, *int*, *long*) semuanya akan dipromosikan ke *int* terlebih dahulu sebelum dilakukan proses perhitungan.



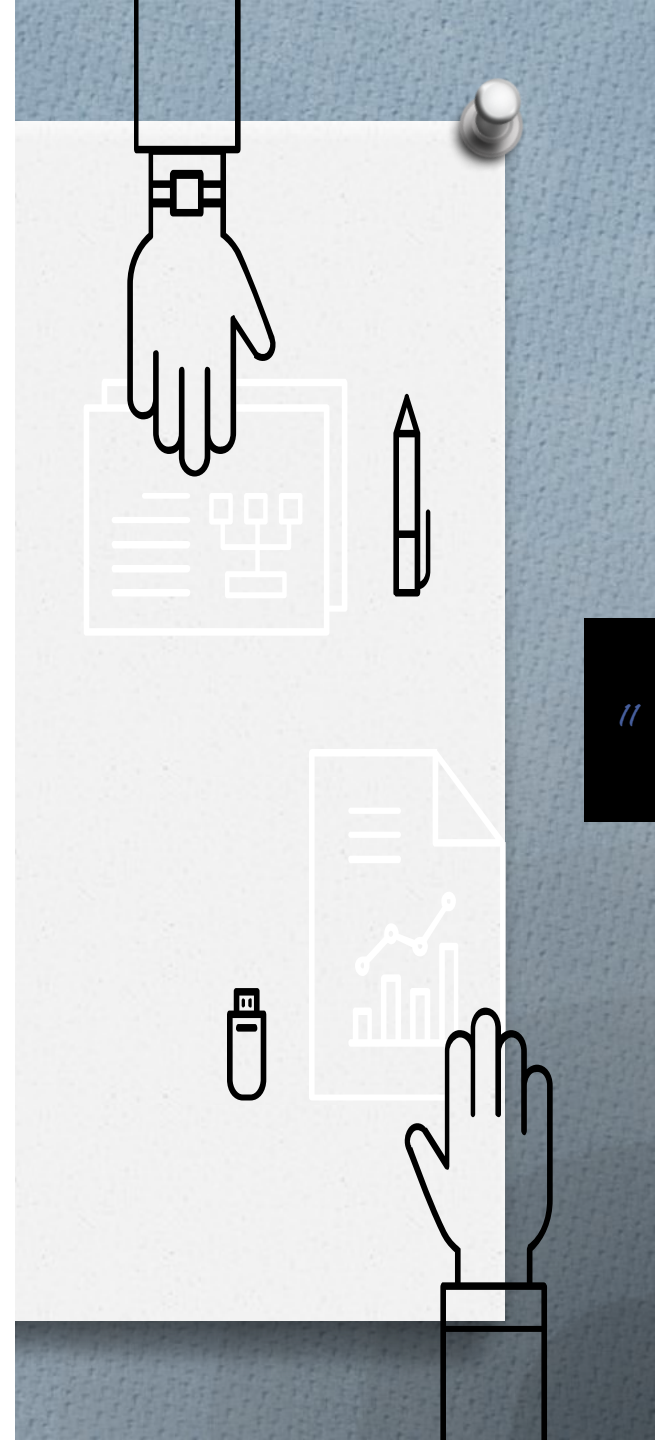
LONG

- ▷ Tipe ini digunakan untuk kasus-kasus tertentu yang nilainya berada di luar rentang tipe *int*, karna tipe ini punya range paling tinggi dibanding *Integer* lainnya. Dengan kata lain, tipe *long* terpaksa digunakan jika data memiliki range diluar range *int*.
- ▷ Range nilai :
-9232372036854775808 sampai
9232372036854775808



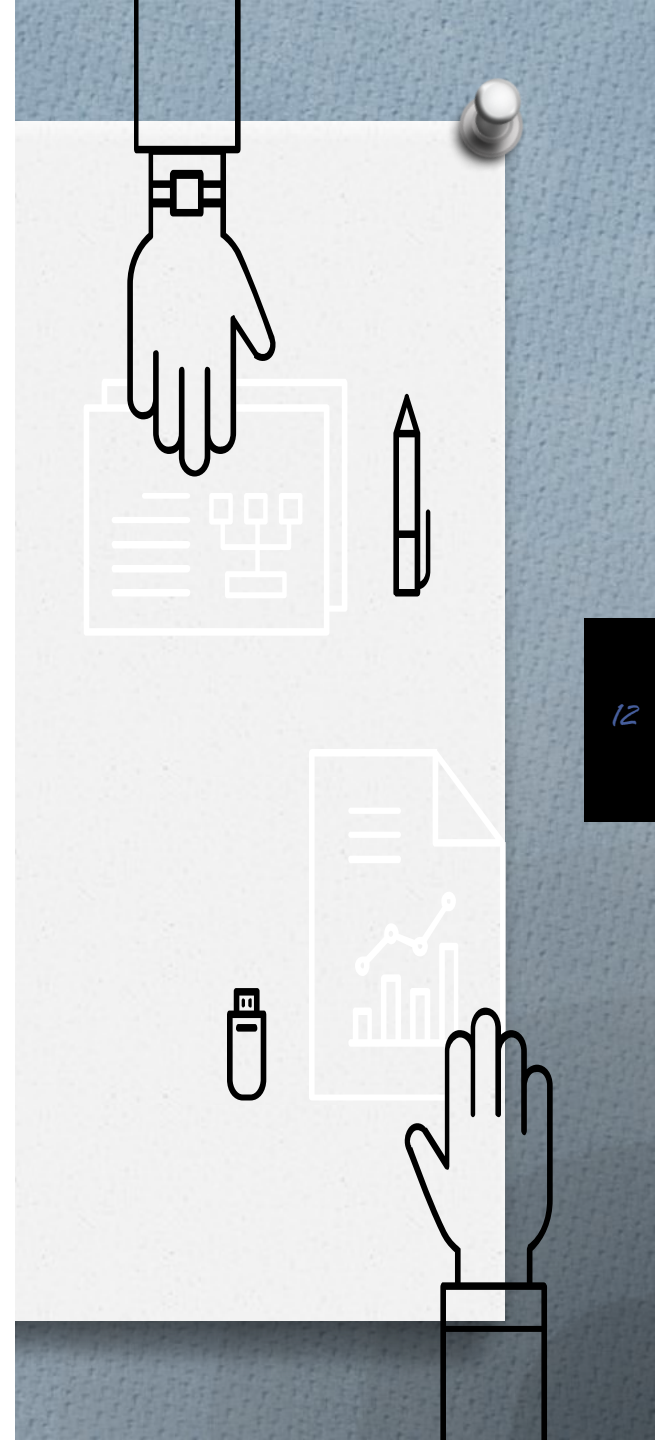
BILANGAN PECAHAN

- ▷ digunakan untuk merepresentasikan nilai-nilai yang mengandung pecahan atau angka decimal di belakang koma, seperti 3.1416, 5.25, dan sebagainya. Bilangan semacam ini disebut sebagai bilangan riil. Dalam Java tipe ini dibedakan menjadi dua jenis, yaitu *float*, dan *double*



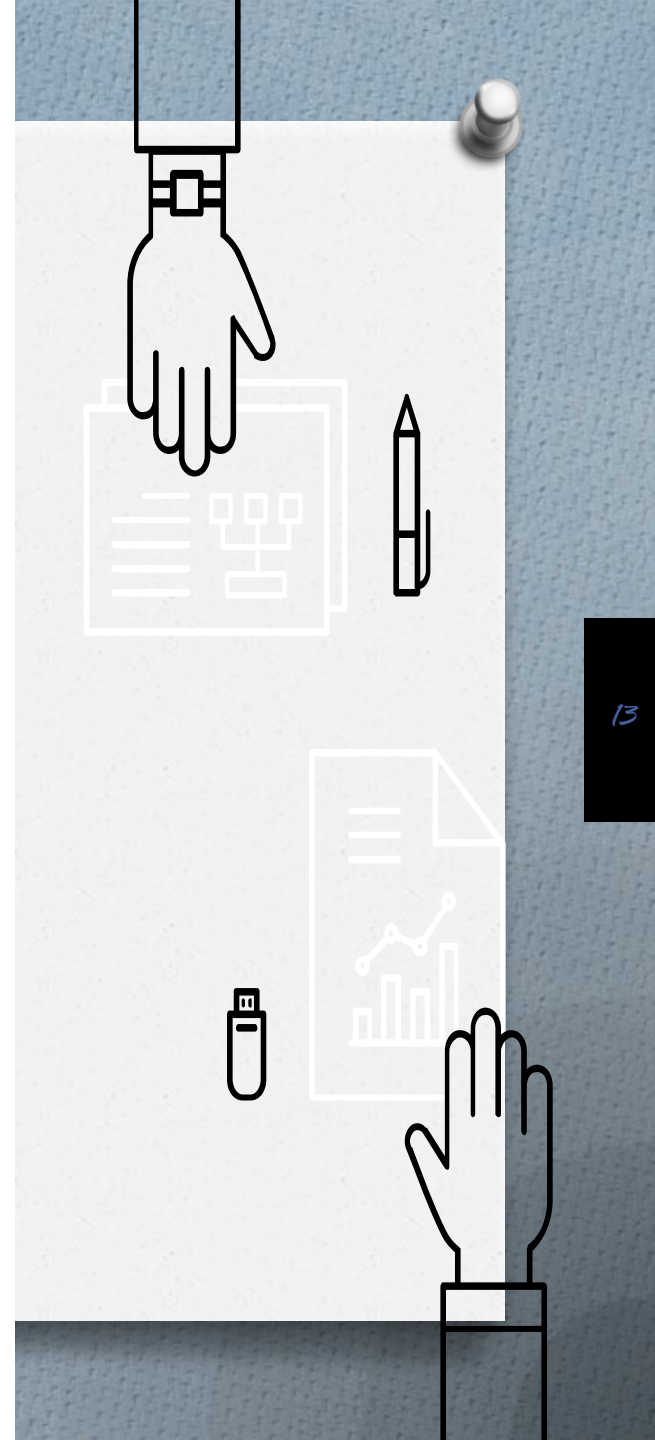
FLOAT

- ▷ Tipe ini digunakan untuk menandakan nilai-nilai yang mengandung presisi atau ketelitian tunggal (*single-precision*) yang menggunakan ruang penyimpanan 32-bit. Presisi tunggal biasanya lebih cepat untuk processor-processor tertentu dan memakan ruang penyimpanan setengah kali lebih sedikit dibandingkan presisi ganda (*double precision*).
- ▷ Range nilai : 10^{-38} sampai 10^{38}



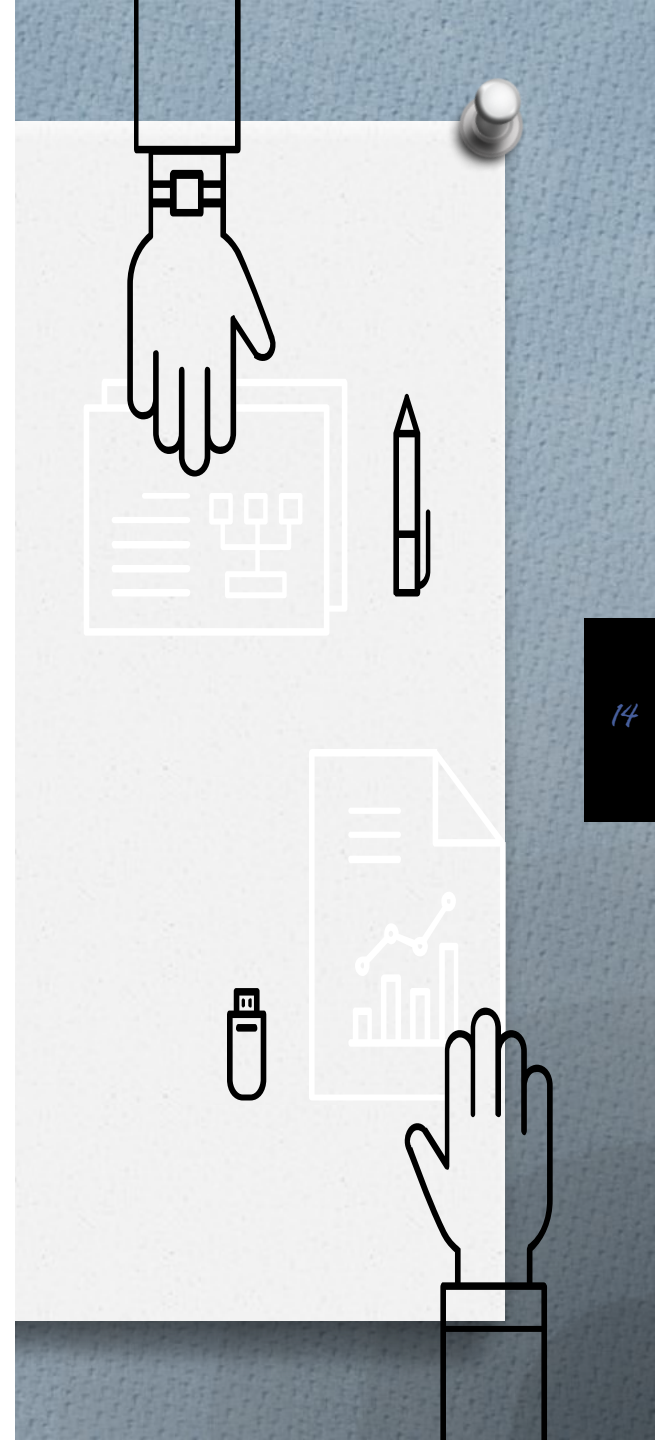
FLOAT

- ▶ Permasalahan yang timbul dari pemakaian tipe *float* untuk nilai-nilai yang terlalu kecil atau justru terlalu besar, karena nilai yang dihasilkan akan menjadi tidak akurat



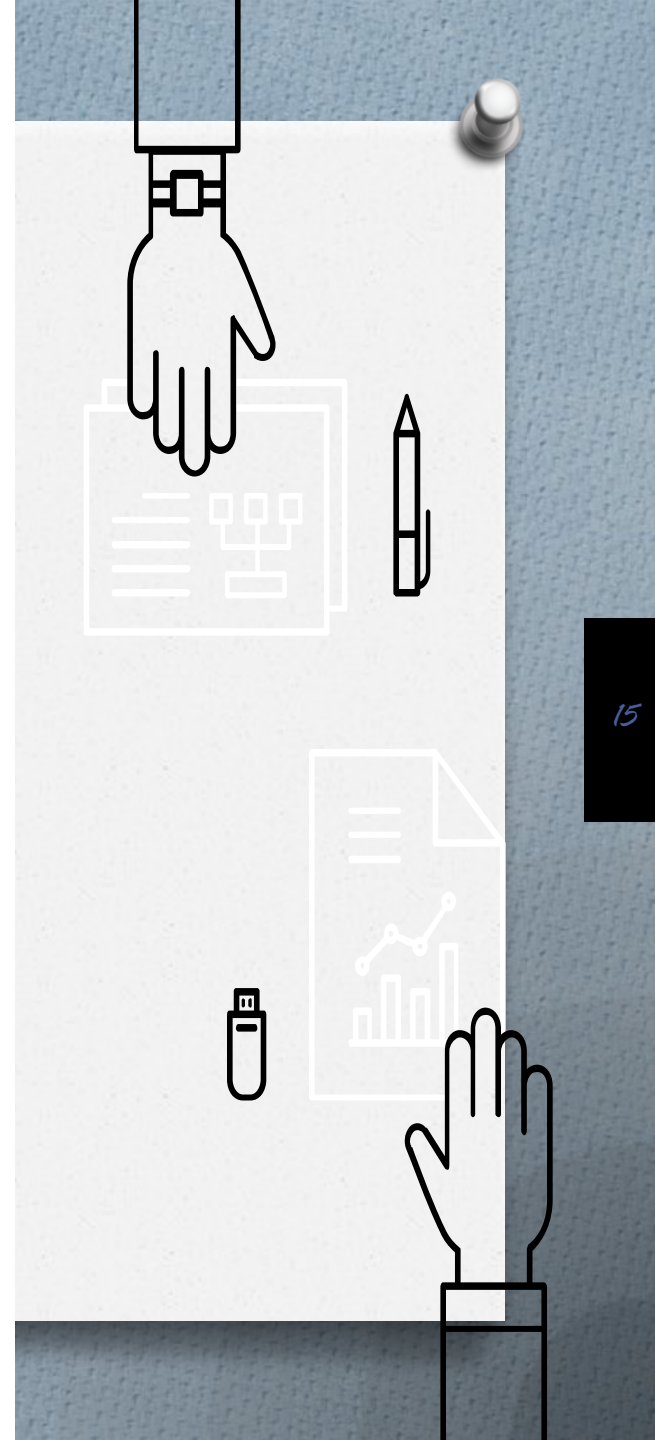
DOUBLE

- ▷ Tipe ini mengandung tingkat ketelitian ganda atau presisi ganda (*double precision*) dan menggunakan ruang penyimpanan 64-bit untuk menyimpan nilai. Tipe *double* tentu lebih cepat untuk melakukan perhitungan-perhitungan matematis daripada tipe *float*.
- ▷ Range nilai : 10^{-308} sampai 10^{308}

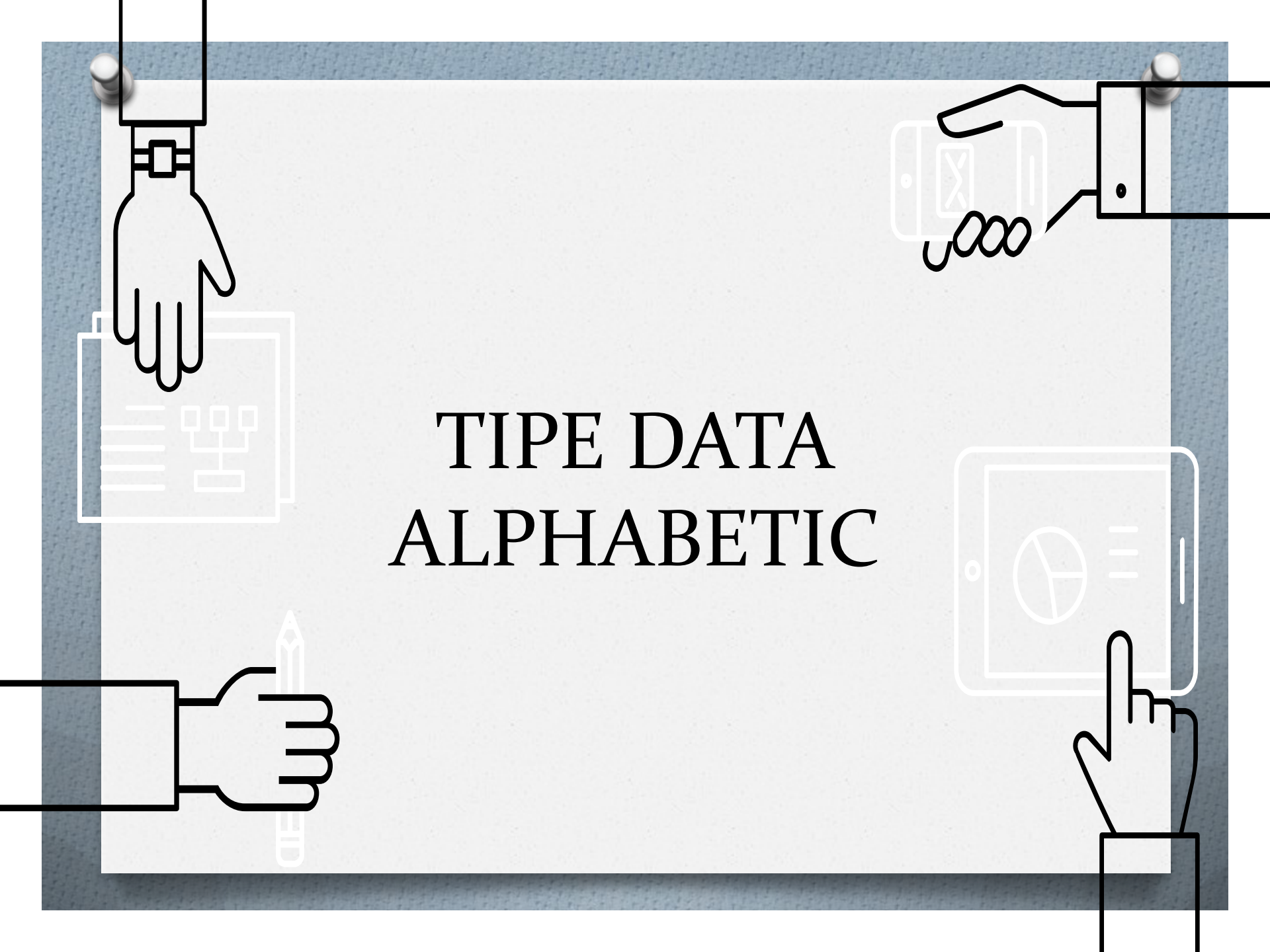


DOUBLE (LANJUTAN)

- ▶ Untuk perhitungan yang bersifat bilangan riil dan menghasilkan hasil yang lebih akurat, maka lebih baik menggunakan tipe *double*.

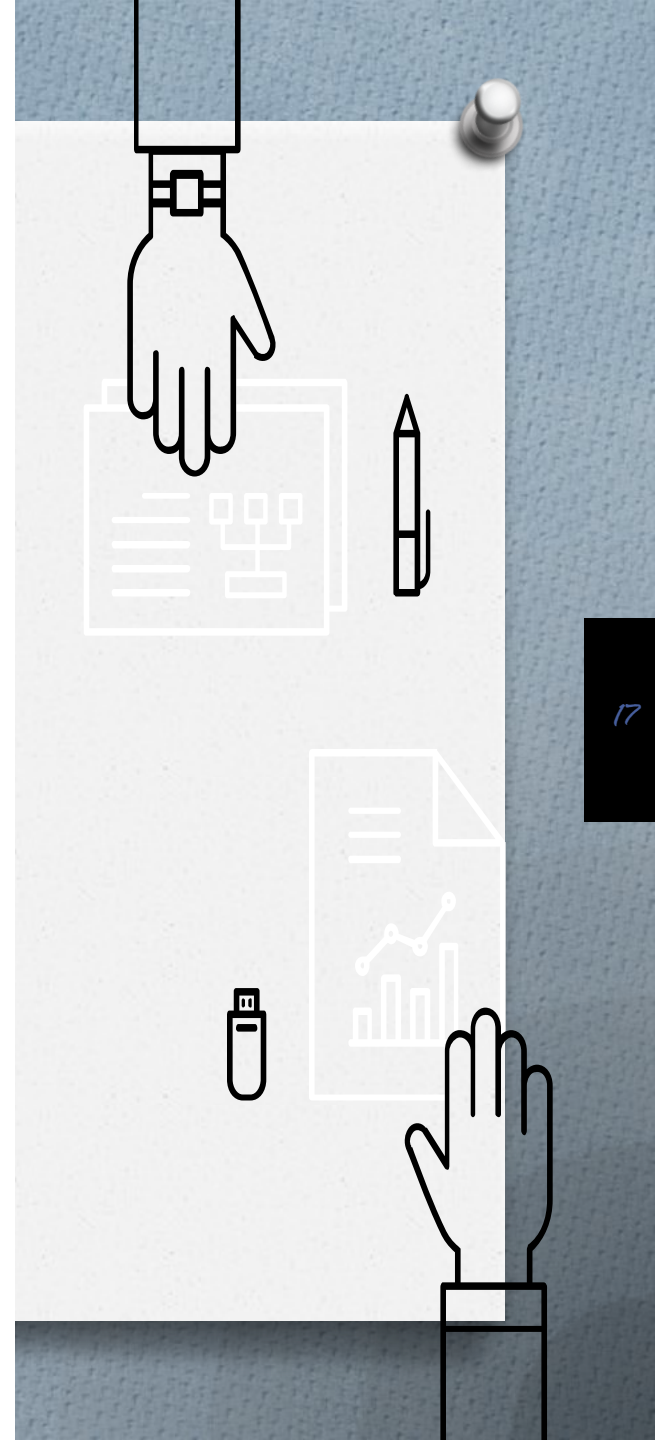


TIPE DATA ALPHABETIC



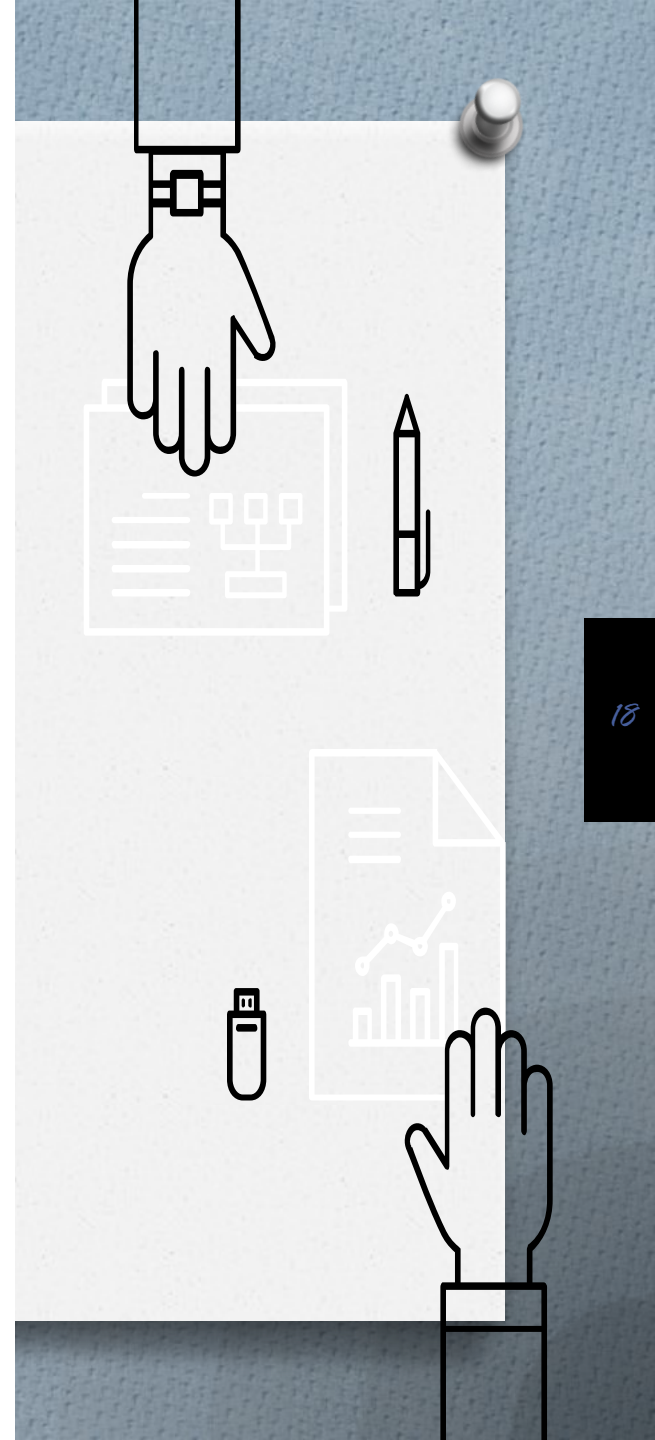
CHAR

- ▷ Tipe data char merupakan tipe untuk menyatakan sebuah karakter.
- ▷ *Unicode* ialah sekumpulan karakter yang terdapat pada semua bahasa, seperti bahasa Latin, Arab, Yunani dan lain-lainnya.

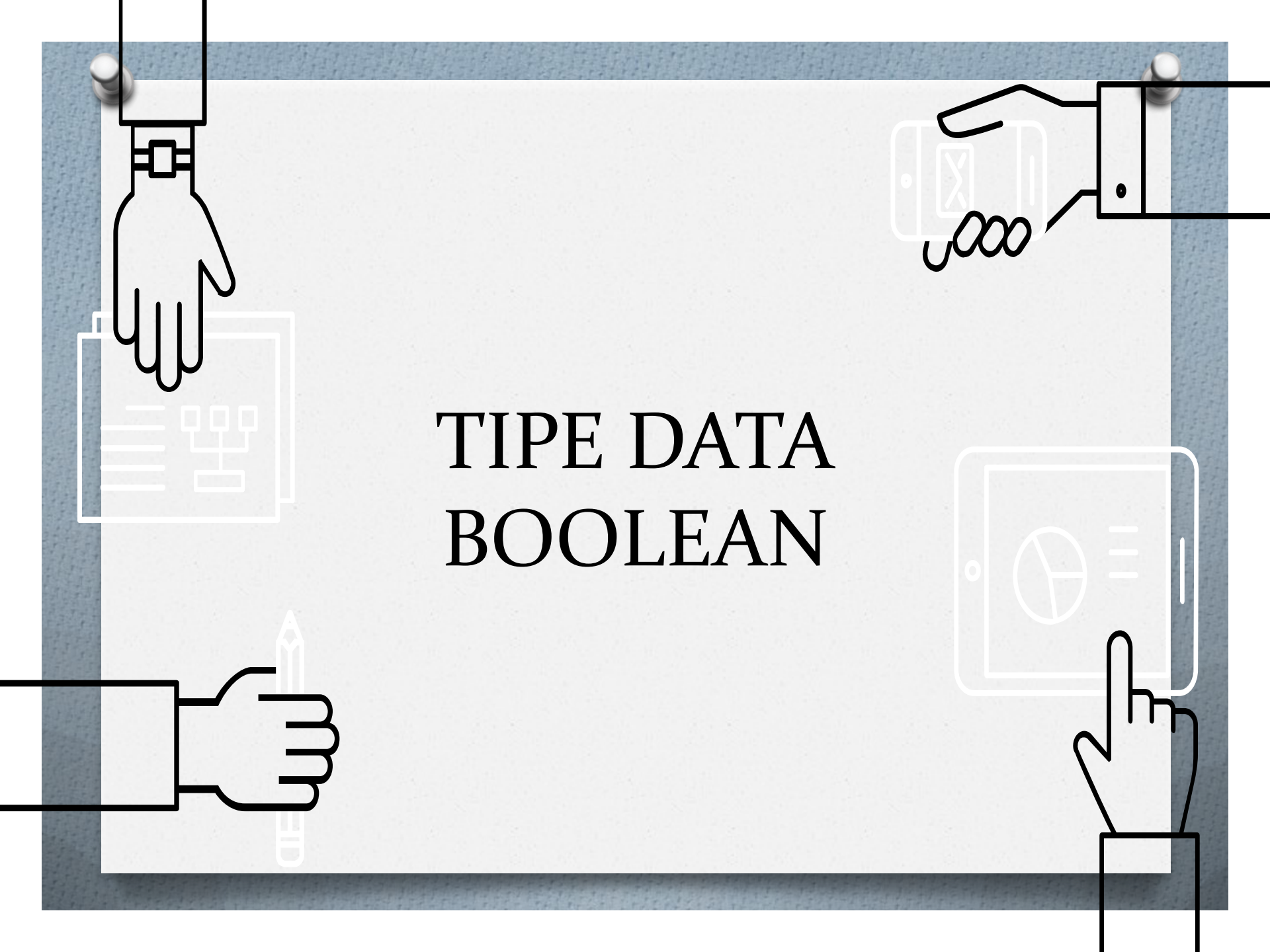


CHAR (LANJUTAN)

- ▶ Untuk karakter-karakter yang tidak dapat diketikkan secara langsung melalui keyboard, java menyediakan beberapa *escape sequence* (pasangan karakter yang dianggap sebagai karakter tunggal). *Escape sequence* tidak dianggap sebagai *String*, melainkan tetap sebagai tipe karakter khusus.

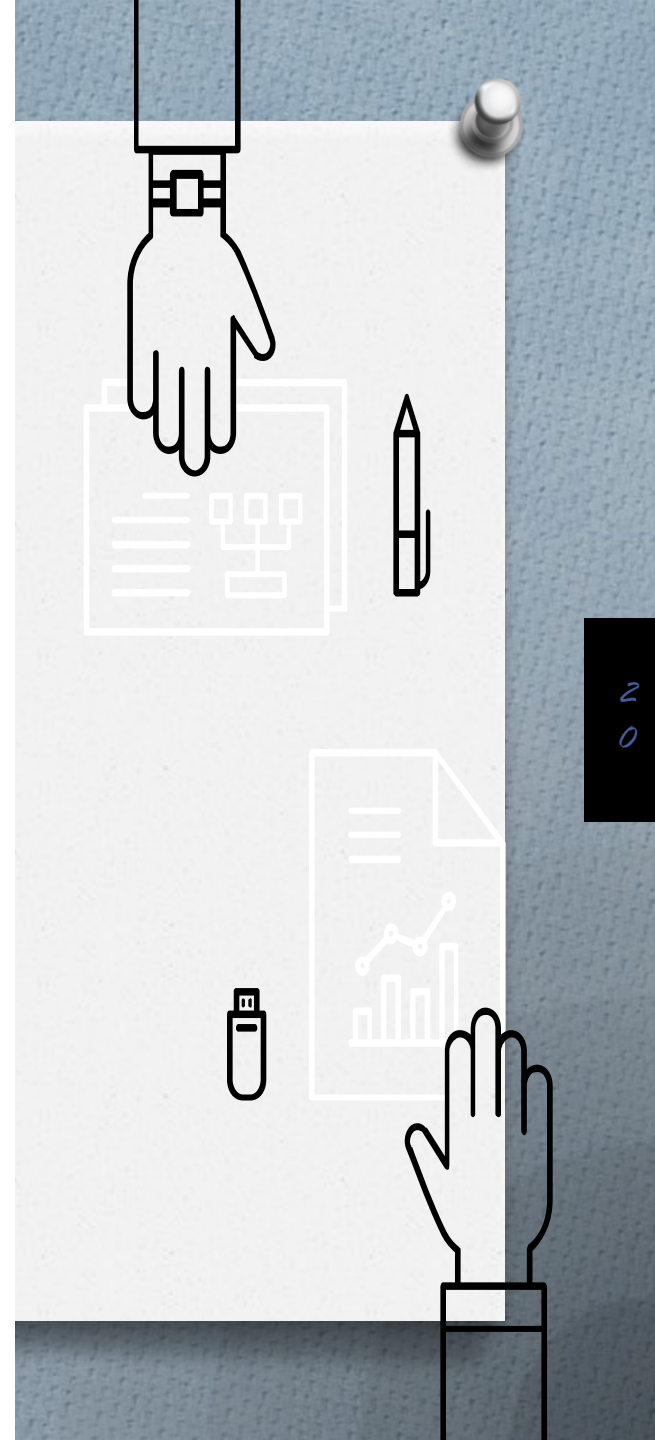


TIPE DATA BOOLEAN



BOOLEAN

- ▷ Tipe *boolean* adalah tipe data yang digunakan untuk menampung nilai logika, yaitu nilai yang hanya memiliki dua buah kemungkinan (benar atau salah). Tipe ini ditandai dengan kata kunci *Boolean*.
- ▷ Dalam bahasa Java, nilai benar dipresentasikan dengan kata kunci *true* dan nilai salah dengan kata kunci *false*



Variabel

- Variabel
 - sebuah nama yang mengacu ke sebuah lokasi pada memori yang digunakan untuk menyimpan suatu nilai data.
 - nilai yang ada, dapat diubah selama eksekusi program berlangsung.
 - digunakan berpasangan dengan tipe data, untuk membatasi nilai data apa yang bisa ditampung dalam sebuah variabel.
 - variabel harus dideklarasikan terlebih dahulu sebelum digunakan.
 - penamaan variabel bersifat case sensitive, dan nama variabel harus dimulai dengan huruf atau garis bawah, boleh mengandung huruf, angka, akan tetapi tidak boleh mengandung simbol

Tipe Data dan Variabel

- Deklarasi:

```
[tipedata] <var_name>;  
[tipedata] <var_name1>, ..., <var_nameN>;  
[tipedata] <var_name1> = <value>,  
<var_name2>;
```

- Contoh:

- `int bil1;`
- `int bil1, bil2, bil3;`
- `int bil1 = 10, bil2;`

Akan dibahas lebih lanjut setelah mempelajari Java

Operator

- o Operator Penugasan
- o Operator Aritmatika
- o Operator Hubungan (Perbandingan)
- o Operator Logika
- o Operator Unary

Operator Penugasan

Operator Penugasan (Assignment operator) dilambangkan dengan tanda sama dengan (“=”).

Contoh :

nilai = 80;

A = x * y;

Artinya : variable “nilai” diisi dengan 80 dan variable “A” diisi dengan hasil perkalian antara x dan y.

Operator Aritmatika

- Beberapa operator aritmatika, yaitu :
 - * : untuk perkalian
 - / : untuk pembagian
 - % : untuk sisa pembagian (modulus)
 - + : untuk pertambahan
 - : untuk pengurangan
- Catatan : operator % digunakan untuk mencari sisa hasil bagi antara dua bilangan.

Misalnya :

$$9 \% 2 = 1$$

$$9 \% 3 = 0$$

$$9 \% 5 = 4$$

$$9 \% 6 = 3$$

Operator Perbandingan

- Operator perbandingan digunakan untuk membandingkan hubungan antara dua buah operand (sebuah nilai atau variable).

Operator	Arti	Contoh	
<	Kurang dari	$x < y$	Apakah x kurang dari y
<=	Kurang dari sama dengan	$x \leq y$	Apakah x kurang dari sama dengan y
>	Lebih dari	$x > y$	Apakah x lebih dari y
>=	Lebih dari sama dengan	$x \geq y$	Apakah x lebih dari sama dengan y
==	Sama dengan	$x == y$	Apakah x sama dengan y
!=	Tidak sama dengan	$x != y$	Apakah x tidak sama dengan y

Operator Logika

- Jika operator perbandingan membandingkan hubungan antara dua buah operand, maka
- operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan.
- Operator logika ada tiga macam, yaitu :
 - && : Logika AND (DAN)
 - || : Logika OR (ATAU)
 - ! : Logika NOT (INGKARAN)

Operator logika (lanjutan)

A	B	!A	!B	A && B	A B
Benar	Benar	Salah	Salah	Benar	Benar
Benar	Salah	Salah	Benar	Salah	Benar
Salah	Benar	Benar	Salah	Salah	Benar
Salah	Salah	Benar	Benar	Salah	Salah

Operator Unary

- Operator Unary merupakan operator yang hanya membutuhkan satu operand saja.

Operator	Arti/Maksud	Letak	Contoh	Equivalen
-	Unary minus	Sebelum operator	$A + -B * C$	$A + (-B) * C$
++	Peningkatan dengan penambahan nilai 1	Sebelum dan sesudah	$A++$	$A = A + 1$
--	Penurunan dengan pengurangan nilai 1	Sebelum dan sesudah	$A--$	$A = A - 1$
sizeof	Ukuran dari operand dalam byte	Sebelum	<code>sizeof(I)</code>	-
!	Unary NOT	Sebelum	<code>!A</code>	-
~	Bitwise NOT	Sebelum	<code>~A</code>	-
&	Menghasilkan alamat memori operand	Sebelum	<code>&A</code>	-
*	Menghasilkan nilai dari pointer	Sebelum	<code>*A</code>	-

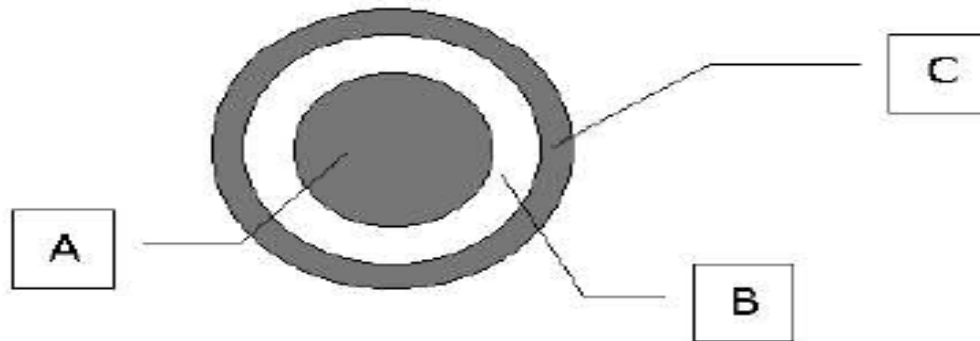
Tabel Precedence Operator

Operator	Precedence
<i>Postfix</i>	<i>expr++ expr--</i>
<i>Unary</i>	<i>++expr --expr +expr -expr ~ !</i>
Multiplikasi	<i>* / %</i>
Aditif	<i>+ -</i>
Pergeseran	<i><< >> >>></i>
Relasional	<i>< > <= >= instanceof</i>
Persamaan	<i>== !=</i>
bitwise AND	<i>&</i>
bitwise exclusive OR	<i>^</i>
bitwise inclusive OR	<i> </i>
Logika AND	<i>&&</i>
Logika OR	<i> </i>
<i>Ternary</i>	<i>? :</i>
Pemberian	<i>= += -= *= /= %= &= ^= = <<= >>= >>>=</i>

Latihan

1. Buatlah flowchart untuk mencari rata-rata 4 buah bilangan
2. Perhatikan gambar di bawah

Diketahui 3 buah lingkaran dengan posisi saling menindih seperti pada gambar.



Hitung luas lingkaran yang diarsir
(pseudocode) jika jari-jari $A < \text{jari-jari } B < \text{jari-jari } C$