



Java Fundamental | Bootcamp

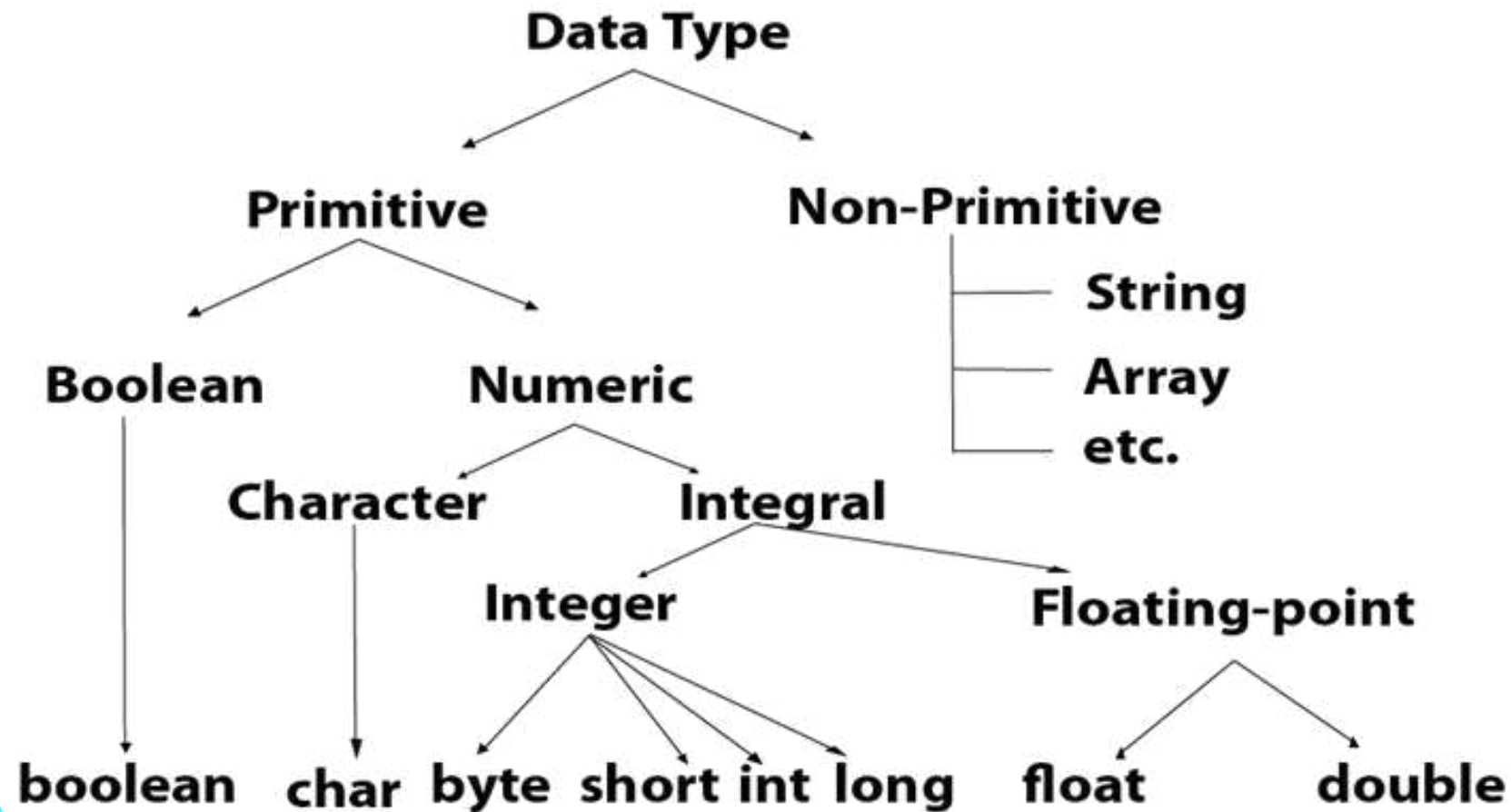
Data Type & Variable

DATA TYPE

Data Type dapat menentukan ukuran dan nilai berbeda yang akan disimpan kedalam suatu variable. Dalam java terdapat 2 tipe data :

1. Primitive data types
2. Non-primitive data types

PRIMITIVE DATA TYPES



PRIMITIVE DATA TYPES - BOOLEAN

Boolean tipe data yang digunakan untuk menyimpan hanya 2 value, 'true' dan 'false'. Boolean bisa digunakan untuk mengetahui benar atau salah tergantung kondisinya.

```
boolean benar = true;  
Boolean salah = false;
```

Primitive Type

Reference Type

PRIMITIVE DATA TYPES - BYTE

Byte adalah tipe data seperti int, hanya saja ukuran maksimal byte jauh lebih kecil daripada int, rentang nilai byte adalah -128 s/d 127, secara default nilainya 0.

```
byte k = 10;  
byte h = 129;
```

Error, karena ukuran data melebihi batas
127

Type mismatch: cannot convert from int to byte
2 quick fixes available:
[Add cast to 'byte'](#)
[Change type of 'h' to 'int'](#)
Press 'F2' for focus

```
10  
11  
12  
13  
14
```

```
byte k = null;  
Byte h = 0;  
  
Byte d = null;  
byte e = 0;
```

PRIMITIVE DATA TYPES - SHORT

Short adalah tipe data yang ukurannya 16-bit dengan rentang nilai **-32,768 s/d 32,767**.

```
short k = 32767;|  
short h = 32768;
```

Type mismatch: cannot convert from int to short
2 quick fixes available:
[Add cast to 'short'](#)
[Change type of 'h' to 'int'](#)
Press 'F2' for focus

Meskipun hanya selisih 1 tetap akan error karna batasnya : **32767**

```
10      short k = null;  
11      Short h = 0;  
12  
13      Short d = null;  
14      short e = 0;
```

PRIMITIVE DATA TYPES - INT

Int adalah tipe data yang ukurannya 32-bit dengan rentang nilai - **2.147.483.648 s/d 2.147.483.647** dan default nilainya adalah 0.

```
int k = 2147483647;  
int h = 2147483648;
```

The literal 2147483648 of type int is out of range
Press 'F2' for focus

Meskipun hanya selisih 1
tetap akan **error**

```
int k = null;  
Integer o = 0;  
  
int d = 0;  
Integer e = null;
```

int tidak bisa diisi nilai **hampa** (null),
Integer bisa diisi 0 karena 0 masih
dianggap sebagai suatu **nilai**

PRIMITIVE DATA TYPES - LONG

Long adalah tipe data yang ukurannya 64-bit dengan rentang nilai - **9.223.372.036.854.775.808 (-9 Kuantiliun) s/d**

9.223.372.036.854.775.807 (9 Kuantiliun) dan default nilainya adalah 0.

```
long k = null;  
Long o = 0;
```

```
long d = 0;  
Long e = null;
```

long tidak bisa diisi **null** (hampa), dan **Long** tidak bisa diisi **0**

PRIMITIVE DATA TYPES - FLOAT

Float adalah tipe data yang digunakan untuk menyimpan data angka dengan koma, float adalah floating point IEEE 745 dengan size 32-bit.

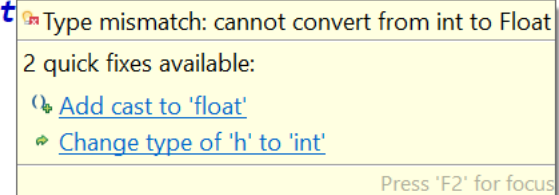
Terdapat 2 penulisan dari data type ini yakni, **Float** dan **float**.



```
float k = null;  
float h = 0;
```



```
Float k = null;  
Float h = 0;  
System.out
```



PRIMITIVE DATA TYPES - DOUBLE

Sama dengan float, double juga digunakan untuk menyimpan data angka dengan koma, namun double memiliki ukuran 64-bit dua kali lebih besar dari float. Default nilainya 0.0d.

Terdapat 2 penulisan dari data type ini yakni, **Double** dan **double**.

```
double doubleKecil = 0.0;  
Double doubleBesar = null;
```



```
18  
19  
20
```

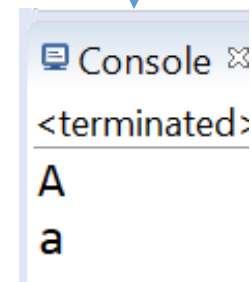
```
double doubleKecil2 = null;  
Double doubleBesar2 = 0.0;
```

PRIMITIVE DATA TYPES - CHAR

Tipe data char digunakan untuk menyimpan karakter tunggal dengan ukurannya 16-bit sedangkan rentrang nilainya antara '\ u0000' (atau 0) hingga '\ uffff' (65.535)

```
char karakter1= 'a';
```

```
System.out.println(Character.toUpperCase(karakter1));  
System.out.println(karakter1);
```



NON-PRIMITIVE DATA TYPES

NonPrimitive disebut dengan reference type, karena dia lebih merujuk ke object. Ada 4 tipe data yang tergolong tipe nonprimitive.

- String
- Class
- Array
- Interfaces

NON-PRIMITIVE DATA TYPES - STRING

String merupakan sekelompok karakter yang untuk mendeklarasikannya diawali dan diakhiri dengan tanda (" "). String biasa digunakan untuk menyimpan suatu teks.

```
String company = "kodehive";  
System.out.println(company);
```

NON-PRIMITIVE DATA TYPES - CLASS

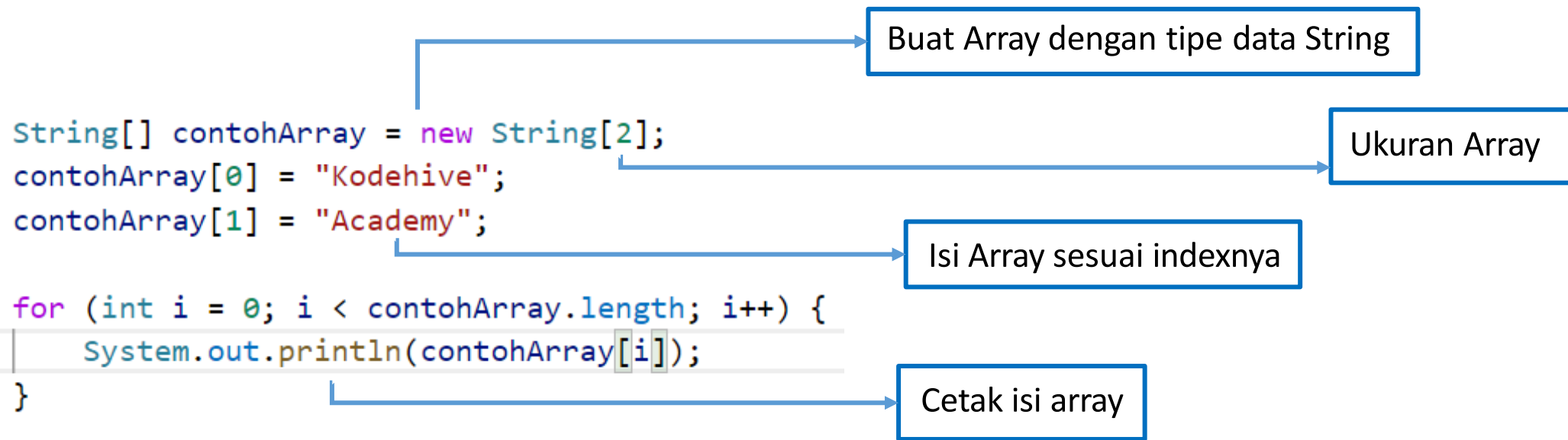
Class memungkinkan kita membuat suatu objek, karena java sendiri adalah bahasa pemrograman berbasis objek.

```
public class KodehiveAcademy {
```

```
}
```

NON-PRIMITIVE DATA TYPES - ARRAY

Array memungkinkan kita untuk menampung sejumlah nilai-nilai tetap dalam satu variable.



NON-PRIMITIVE DATA TYPES - INTERFACES

Interface memungkinkan kita dapat membuat abstract method / method yang tidak berisi.

```
public interface Kodehive {  
  
}
```

Local Variable

```
package day2;

public class LocalVariable {

    public void metodeSatu() {
        String warna = "Biru";
        int berat = 65;
    }

    public void metodeDua() {
        String warna = "Merah";
        int berat = 50;
    }

    // Terdapat 2 var local yaitu warna dan berat
    // var local atau local variable dlm bahasa inggrisnya
    // variable tsb posisinya di dalam sebuah method
    // jika variable tsb berada dalam method, warnanya akan berbeda
    // dengan variable yang berada diluar method
}
```

Global Variable

```
package day2;

public class GlobalVariable {
    String nama = "WILIAM";
    int usia = 27;
    // var nama dan usia disebut dengan global
    // karena var global posisi syntax var tsb berada diluar
    // diluar metodeSatu dan metodeDua

    public void metodeSatu() {

    }

    public void metodeDua() {

    }
}
```

Global & Local Variable

```
public class GlobalAndLocal {  
  
    String bahasaPersatuan = "Bahasa Indonesia"; // var global  
    int pancasila = 5; // var global  
  
    public void sukuJawa() {  
        String bahasaDaerah = "Bahasa Jowo"; // var local  
        String kataSandangPria = "Mas"; // var local  
        String bahasaNasional = bahasaPersatuan; // var Local nilainya dari var global  
    }  
  
    public void sukuBatak() {  
        String bahasaDaerah = "Bahasa Batak"; // var local  
        String kataSandangPria = "Ucok"; // var local  
        int falsafah = pancasila; // var local nilainya dari var global  
    }  
}
```

Konversi Variable – Sesudah Konversi

```
public class KonversiVariable {  
    Run | Debug  
    public static void main(String[] args) {  
        String angkaString = "900";  
        int angkaInteger = 100;  
        int perkalian = Integer.valueOf(angkaString) * angkaInteger;  
        System.out.println(perkalian);  
    }  
}
```

Target
Convert

Data yang
akan di
Convert

Operator Aritmatika

No	Operasi	Operator
1	Penjumlahan	+
2	Pengurangan	-
3	Perkalian	*
4	Pembagian	/
5	Increment	++
6	Decrement	--
7	Modulus / Sisa bagi	%
8	Assignment Operator	=

Table disamping adalah operator aritmatika yang biasa digunakan dalam programming.