



Project Based Internship

Java Data Types, Variable, Expressions and Method

- Memahami data types, variable, dan expressions pada Java
- Memahami cara membuat Method pada Java
- Memahami cara memanggil Method pada Java

Daftar Isi

A. Memahami data types, variable, dan expressions pada Java	3
B. Memahami cara membuat Method pada Java	7
C. Memahami cara memanggil Method pada Java	9
References	11

A. Memahami data types, variable, dan expressions pada

Java

Java merupakan bahasa pemrograman yang berorientasi objek. Namun meskipun Java berorientasi pada objek, tidak semua tipe data java adalah objek. Sebab Java dibangun di atas tipe variabel dasar yang disebut dengan primitif.

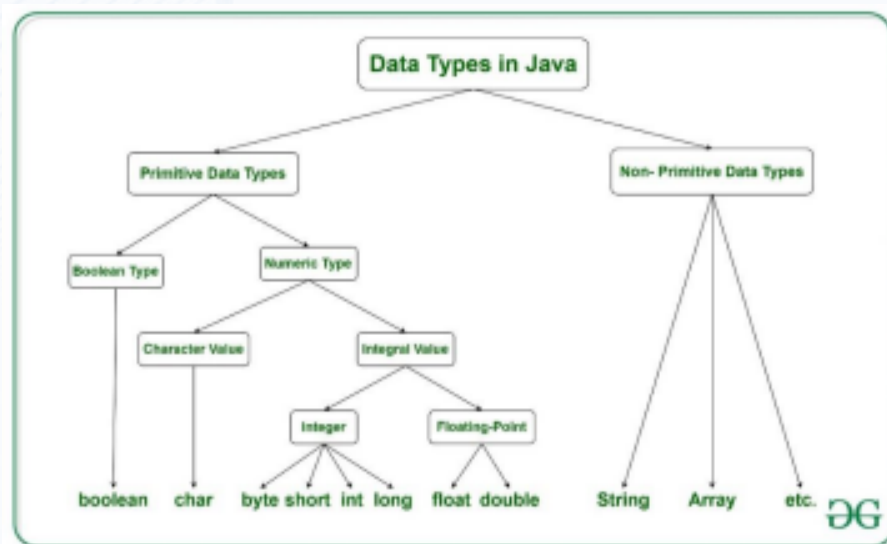
Tipe data adalah berbagai ukuran dan nilai yang dapat disimpan dalam variabel yang dibuat sesuai kenyamanan dan keadaan untuk menutupi semua kasus pengujian. Juga, mari kita tutupi penyakit penting lainnya yang sebagian besar ada dua jenis bahasa yaitu sebagai berikut:

Pertama, salah satunya adalah bahasa yang diketik secara Statis di mana setiap variabel dan jenis ekspresi sudah diketahui pada waktu kompilasi. Setelah variabel dinyatakan sebagai tipe data tertentu, variabel tersebut tidak dapat menyimpan nilai tipe data lainnya. Misalnya C, C++, Java.

Yang lainnya adalah bahasa yang diketik secara dinamis. Bahasa-bahasa ini dapat menerima tipe data yang berbeda dari waktu ke waktu. Misalnya Ruby, Python.

Java memiliki dua kategori di mana tipe data dipisahkan

1. Tipe Data Primitif: seperti boolean, char, int, short, byte, long, float, dan double.
2. Tipe Data Non-Primitif atau tipe Data Objek: seperti String, Array, dll.



Data primitif hanyalah nilai tunggal dan tidak memiliki kemampuan khusus. Ada 8 tipe data primitif. Mereka digambarkan di bawah ini dalam format tabular di bawah ini sebagai berikut

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	����	16 bits	'a', '����', '����', 'W', '�', 'w', '�', '�'	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

Dalam menuliskan code pada Java, Anda harus mendefinisikan setiap variabelnya sebelum digunakan. Hal ini karena Java adalah salah satu bahasa pemrograman yang bersifat kuat (strong typed language). Berikut ini penjelasan setiap variabel Java dan tipe data Java:

Numbers atau Angka

Tipe data Java yang pertama adalah Numbers. Tipe data Numbers dideklarasikan dan ditetapkan dengan sintaks sebagai berikut:

```
int angka;  
angka = 10;
```

Namun tentu saja ada cara lain untuk mendeklarasikannya, yaitu dengan menggabungkan sintaks dan angka yang dideklarasikan seperti di bawah ini:

```
int angka = 10;
```

Sementara itu, untuk mendeklarasikan numbers yang merupakan bilangan floating dengan point ganda, gunakan sintaks berikut ini:

```
double nilai = 9.5;  
nilai = 5.0;
```

Untk mendeklarasikan bilangan float tunggal, Anda dapat menggunakan:

```
float nomor = (float) 11.5;
```

atau Anda dapat menuliskan sintaknya seperti di bawah ini:

```
float nomor = 11.5f;
```

Char dan String

Di dalam bahasa pemrograman Java, char atau karakter merupakan tipe tersendiri, bukan sekadar angka. Sehingga tidak umum jika Anda memasukkan nilai di dalamnya, telah terdapat sintaks khusus untuk karakter, seperti berikut ini:

```
char nama = 'Jason';
```

Untuk tipe data string sendiri, tidak termasuk dalam tipe primitif. String merupakan tipe data Java yang nyata, akan tetapi Java memiliki perlakuan khusus untuk tipe data satu ini. Berikut ini cara yang dapat Anda lakukan untuk menggunakan tipe data string:

```
String namadpn = new String("Jason");
```

```
String namablkg = "Putro";
```

```
String hasil = namadpn + namablkg;
```

Di dalam bahasa pemrograman Java, tidak ada operator yang overload atau kelebihan beban. Namun terdapat pengecualian dalam aturannya, yaitu untuk variabel tipe data string, sebab Java mendukung overload operator pada string.

Boolean

Tipe terakhir ialah Boolean. Setiap operator pembandingan yang terdapat dalam bahasa pemrograman Java akan mengembalikan setiap tipe boolean. Tidak seperti bahasa lain yang menerima lebih dari dua nilai khusus, Java hanya menerima dua nilai khusus yaitu true atau false. Contoh penulisan sintaks pada tipe data Java satu ini ialah:

```
boolean x = false;
```

```
x = true;
```

```
boolean aX = false;
```

Itulah pembahasan tentang 3 variabel dan tipe data Java khusus untuk Anda. Untuk Anda yang belajar Java pemula, ingatlah bahwa cara terbaik untuk memahami dan menjadi mahir adalah dengan berlatih sesering mungkin.

B. Memahami cara membuat Method pada Java

Sebuah method di Java memiliki struktur yang terdiri dari:

- Modifier berbagai statement yang menandakan hak akses method tersebut dari luar class. Terdiri dari statement public, private, dan protected
- Return type, statement yang menandakan tipe kembalian dari method. Ada int, void, string, double, dan berbagai objek kompleks di java
- Nama method, nama method yang biasanya terdiri dari alfabet saja dan ditulis dengan camel case

- Parameter, berbagai nilai yang diperlukan oleh method, dalam penulisannya harus diawali dulu dengan tipe datanya baru nama parameternya.
- Return statement, kembalian suatu method harus sesuai dengan tipe data method, jadi jika method berjenis int maka kembaliannya harus int. Khusus untuk tipe void tidak perlu menggunakan return statement
- Exception, kemungkinan error yang akan muncul ketika kode dieksekusi di dalam method, biasanya ditambahkan statement throws dan jenis exception yang akan ditangkap

Method di Java harus selalu berada di dalam class tidak dapat berdiri sendiri seperti di Python atau PHP.

Pada dasarnya sebuah method adalah function yang ada di dalam class. Semua fitur function bisa kita pakai ke dalam method, misalnya mengembalikan suatu nilai (return):

```
class Laptop {
    int harga = 7500000;
    String cekHarga () {
        return "Harga Laptop adalah "+harga;
    }
};

class BelajarJava {
    public static void main(String args[]){

        Laptop laptop = new Laptop();
        String hargaLaptop = laptop.cekHarga();
        System.out.println(hargaLaptop);

    }
}
```


Hasil kode program: Harga Laptop adalah 7500000

Di baris 1-6 class Laptop dengan property harga yang diisi angka 7500000. Setelah itu terdapat method cekHarga yang mengembalikan sebuah string di baris 7.

Class laptop kemudian di instansiasi pada baris 11 ke dalam object Laptop. Di bawahnya, variabel hargaLaptop akan menampung hasil pemanggilan method laptop.cekHarga().

Dengan cara ini, variabel hargaLaptop akan berisi string "Harga Laptop adalah 7500000", sesuai dengan hasil perintah return dari method laptop.cekHarga().

C. Memahami cara memanggil Method pada Java

Untuk kode yang agak kompleks, sebuah method bisa memanggil method lain dalam class yang sama.

```
class Laptop {
    String hidupkanLaptop(String pemilik, String merk) {
        return "Hidupkan Laptop "+merk+" milik "+pemilik;
    }

    String matikanLaptop(String pemilik, String merk) {
        return "Matikan Laptop "+merk+" milik "+pemilik;
    }

    void restartLaptop() {
        System.out.println(matikanLaptop("Rissa","Lenovo"));
        System.out.println(hidupkanLaptop("Rissa","Lenovo"));
    }
};

class BelajarJava {
    public static void main(String args[]){
        Laptop laptopAnto = new Laptop();
        laptopAnto.restartLaptop();
    }
}
```

Di dalam class Laptop sekarang ada 3 buah method: `hidupkanLaptop()`, `matikanLaptop()` dan `restartLaptop()`.

Method `hidupkanLaptop()` dan `matikanLaptop()` butuh 2 argument dan mengembalikan sebuah string. Kedua method ini diakses dari dalam method `restartLaptop()`.

Method `restartLaptop()` tidak mengembalikan nilai (`void`), dan langsung menampilkan hasil method `hidupkanLaptop()` dan `matikanLaptop()` dengan perintah `System.out.println()`.

Di dalam class `BelajarJava` object `laptopAnto` di-instansiasi dari class `Laptop`, lalu menjalankan method `restartLaptop()`.

References

<https://codekey.id/java/tipe-data-java/>

<https://www.geeksforgeeks.org/data-types-in-java/>

<https://www.duniaikom.com/tutorial-oop-java-cara-membuat-method-bahasa-java/>