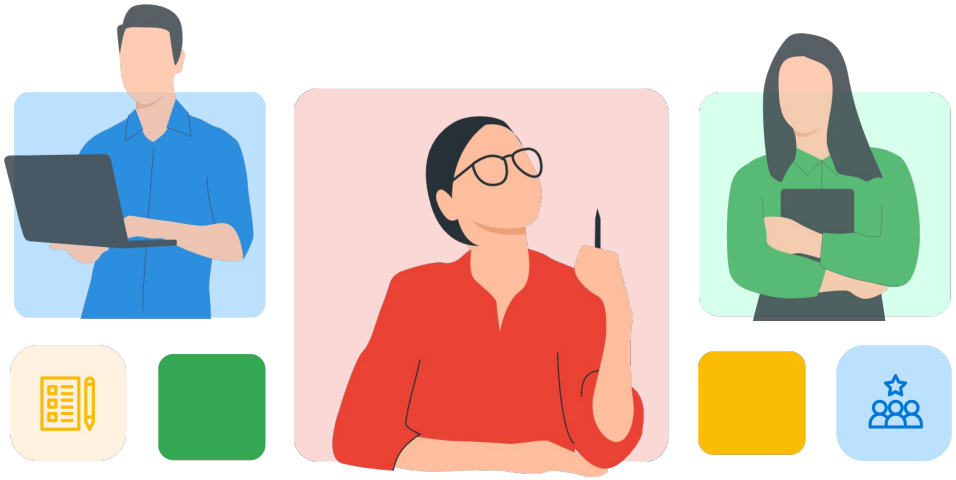
Tujuh Sembilan
Always Improving You

79 Bootcamp Digitalization

Introduction to Java Naming Convention

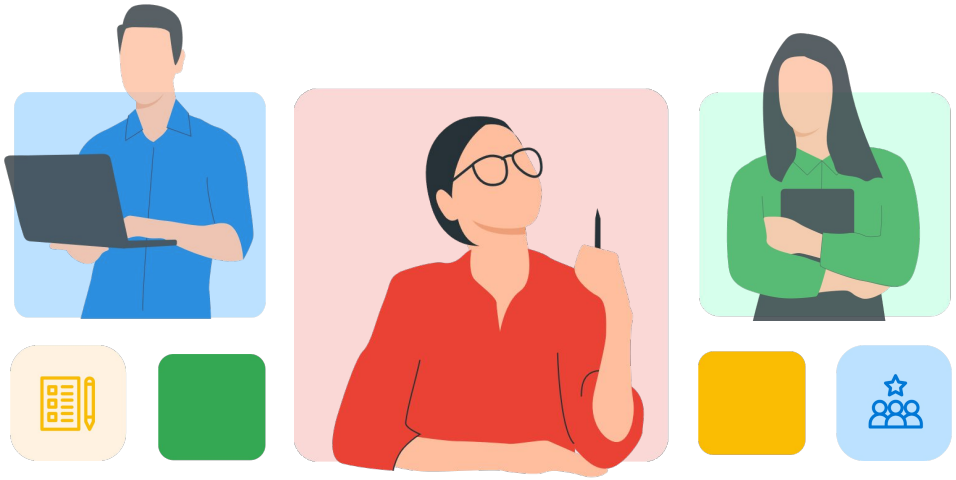
Tujuh Sembilan
Always Improving You



1. Java Naming Convention

Pengantar:

- Java Naming Convention adalah pedoman untuk penamaan **kelas**, **variabel**, **metode**, **konstanta**, dan **paket** dalam bahasa pemrograman **Java**.
- Mengikuti konvensi ini membantu membuat kode lebih mudah dibaca, dipahami, dan dipelihara oleh pengembang lain.



2. Naming Conventions for Packages, Classes, and Interfaces

Paket (Package) :

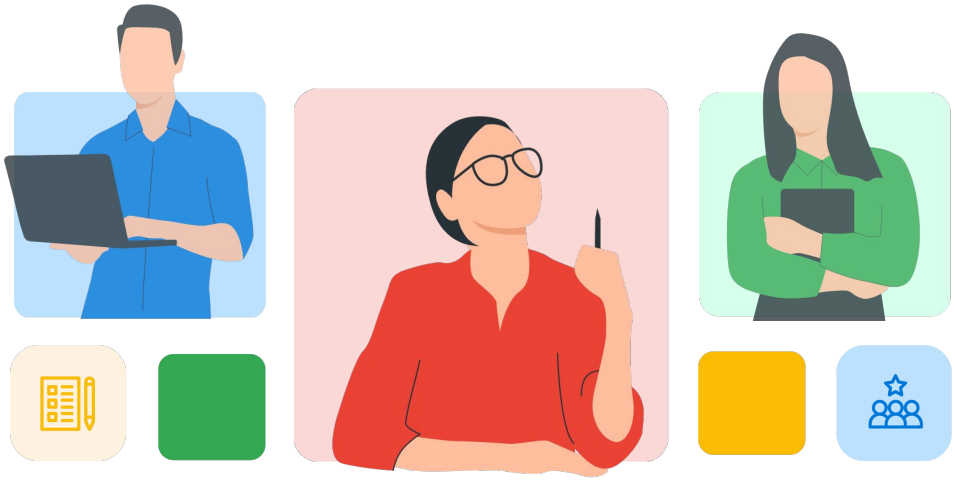
- Nama paket (package) **dimulai dengan huruf kecil** dan **menggunakan huruf kecil untuk semua kata yang mengikutinya**.
- Contoh: **com.example.myproject**

Kelas (Class) :

- Nama kelas **dimulai dengan huruf kapital** dan **menggunakan gaya CamelCase**.
- Nama **kelas harus berupa kata benda** dan jika terdiri dari beberapa kata, awalan setiap kata harus huruf kapital.
- Contoh: **Customer**, **EmployeeDetails**

Interface :

- Nama interface juga dimulai dengan huruf kapital dan menggunakan gaya CamelCase
- Contoh: **Serializable**, **Comparable**



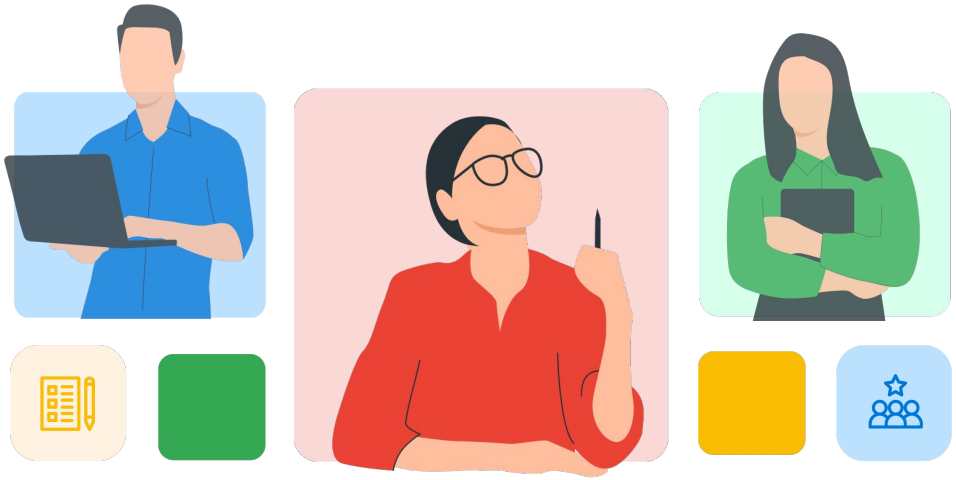
3. Naming Conventions for Methods and Variables

Metode (Method) :

- Nama **metode/method dimulai dengan huruf kecil** dan **menggunakan gaya camelCase**.
- Nama **metode/method** harus berupa **kata kerja** atau **frasa kerja**.
- Contoh: **calculateTotal()**, **getUserInfo()**

Variabel (Variable) :

- Nama **variabel** dimulai dengan **huruf kecil** dan **menggunakan gaya CamelCase**.
- Nama **variabel** sebaiknya **deskriptif, singkat dan jelas**, sehingga mudah dipahami.
- Contoh: **firstName**, **orderQuantity**



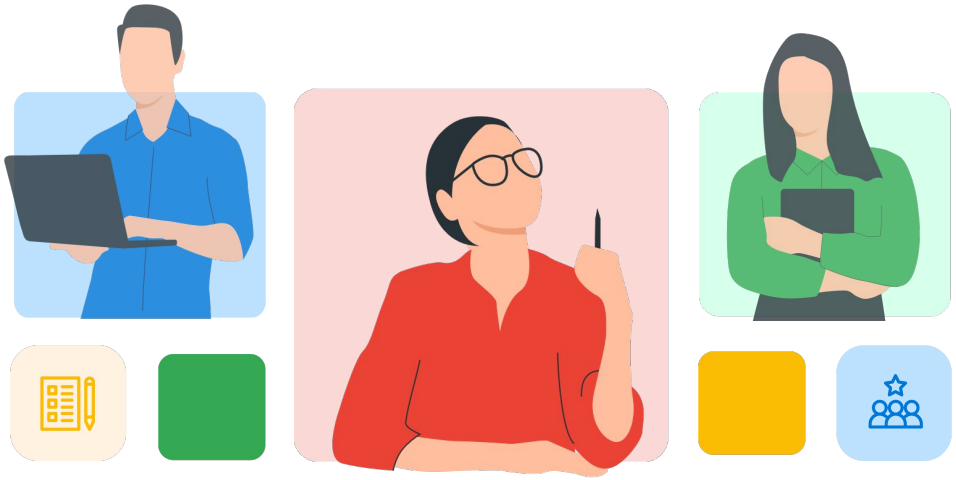
4. Naming Conventions for Constants and Parameters

Konstanta (Constant) :

- Nama **konstanta/final** ditulis dengan **huruf besar semua**.
- Kata-kata dalam nama konstanta dipisahkan dengan garis bawah (_).
- Contoh: **MAX_SIZE**, **PI_VALUE**

Method Parameter :

- Nama parameter metode sama dengan variabel.
- Contoh: calculateArea(double **radius**), printMessage(String **message**), printName(String **firstName**, String **lastName**)



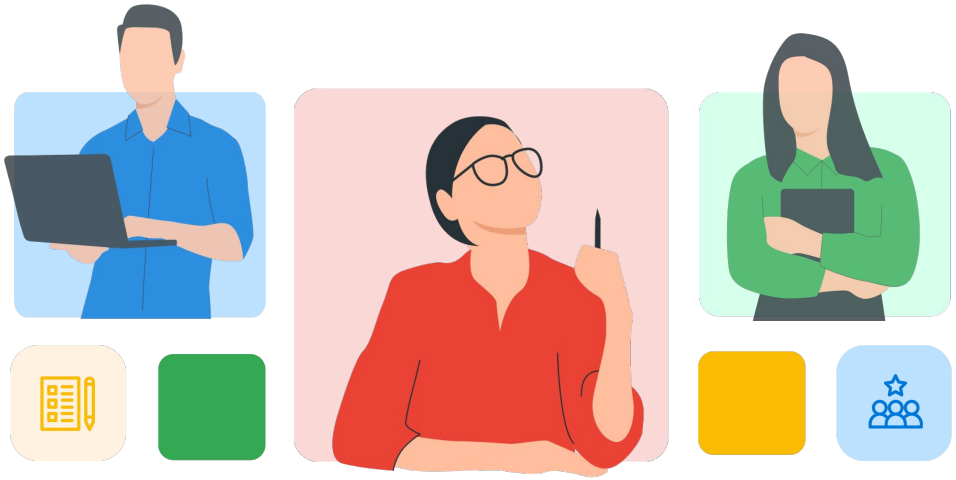
5. Additional Naming Conventions

Konstanta (Constant) :

- Nama konstanta **enumerasi/enum** harus ditulis dengan **huruf besar semua**.
- Kata-kata dalam nama enumerasi dipisahkan dengan garis bawah (_).
- Contoh: **Color.RED**, **DayOfWeek.MONDAY, Level.MEDIUM**

Penamaan Boolean :

- Nama **variabel boolean harus menunjukkan pertanyaan yang dijawabnya**.
- Sebaiknya menggunakan **kata kerja atau frasa kerja sebagai awalan**, diikuti oleh **kata benda** yang menjawab pertanyaan.
- Contoh: **isReady**, **hasPermission**



6. The art of naming variables

Penamaan itu terkadang Sulit. Tetapi hasil dari kesulitan itu akan terbayar ketika kita memberikan Nama yang Sesuai dan Tetap. Sehingga kodingan yang kita buat mudah untuk dibaca dan dipahami orang lain.

Perhatikan Contoh Berikut ini:

```
String[] arrX = {"One", "Two", "Three"};
String[] z = "";

for(String x : arrX){
    z = z + x + ", ";
}

System.out.print(z);
```

Arrays

```
// bad
String[] fruit = {'apple', 'banana', 'cucumber'};

// okay
String[] fruitArr = {'apple', 'banana', 'cucumber'};

// good
String[] fruits = {'apple', 'banana', 'cucumber'};

// great - "names" implies strings
String[] fruitNames = {'apple', 'banana', 'cucumber'};
```

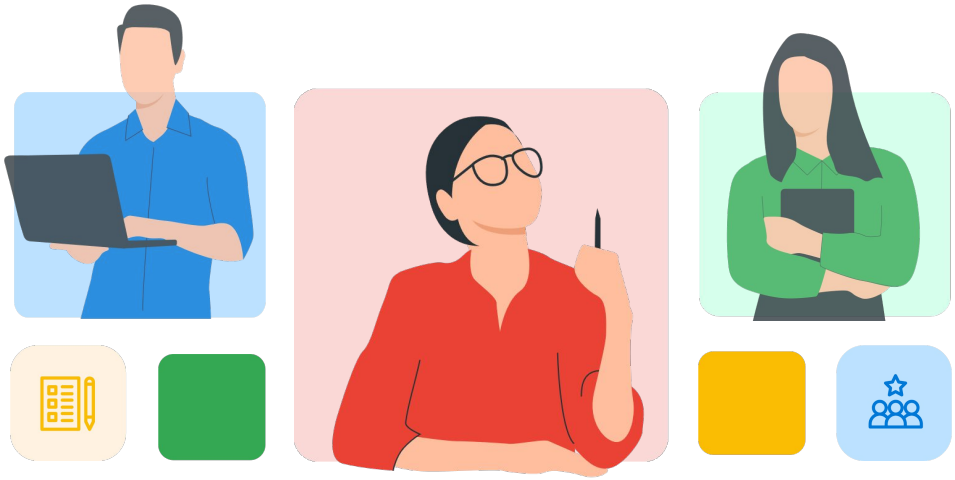
Booleans

```
// bad
boolean open = true;
boolean write = true;
boolean fruit = true;

// good
boolean isOpen = true;
boolean canWrite = true;
boolean hasFruit = true;
```

```
// good
public static boolean checkHasFruit(){
    //Code Here

}

boolean hasFruit = checkHasFruit();
```

Functions

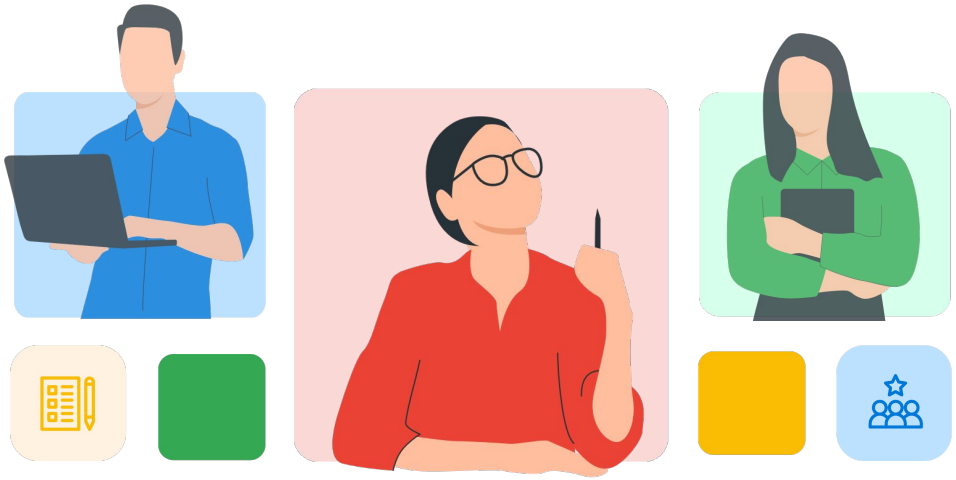
```
// bad
userData(userId)
userDataFunc(userId)
totalOfItems(items)

// good
getUser(userId);
calculateTotal(items);

A common convention I've seen used for transforming values is prefixing function names with to.

// I like it!
toDollars('euros', 20);
toUppercase('a string');

Another common naming pattern I like is when iterating over items. When receiving the argument inside
the function, use the singular version of the array name.
```



6. Benefits of Java Naming Convention

Benefits of Java Naming Convention

- Java Naming Convention adalah seperangkat aturan penamaan dalam Java.
- Kode lebih mudah dibaca dan dipahami.
- Mempermudah kolaborasi dengan pengembang lain.
- Memudahkan pemeliharaan kode di masa depan.



7. Kesimpulan

Kesimpulan:

- Mengikuti Java Naming Convention penting untuk membuat kode Java yang mudah dibaca, dipahami, dan dipelihara.
- Dengan mempraktikkan konvensi ini, kita dapat meningkatkan produktivitas dan kualitas pengembangan perangkat lunak.

Tujuh Sembilan
Always Improving You

Sekian,
Terima Kasih