
JAVA FUNDAMENTAL BOOTCAMP

WELCOME TO JAVA JUNGLE! 🌴



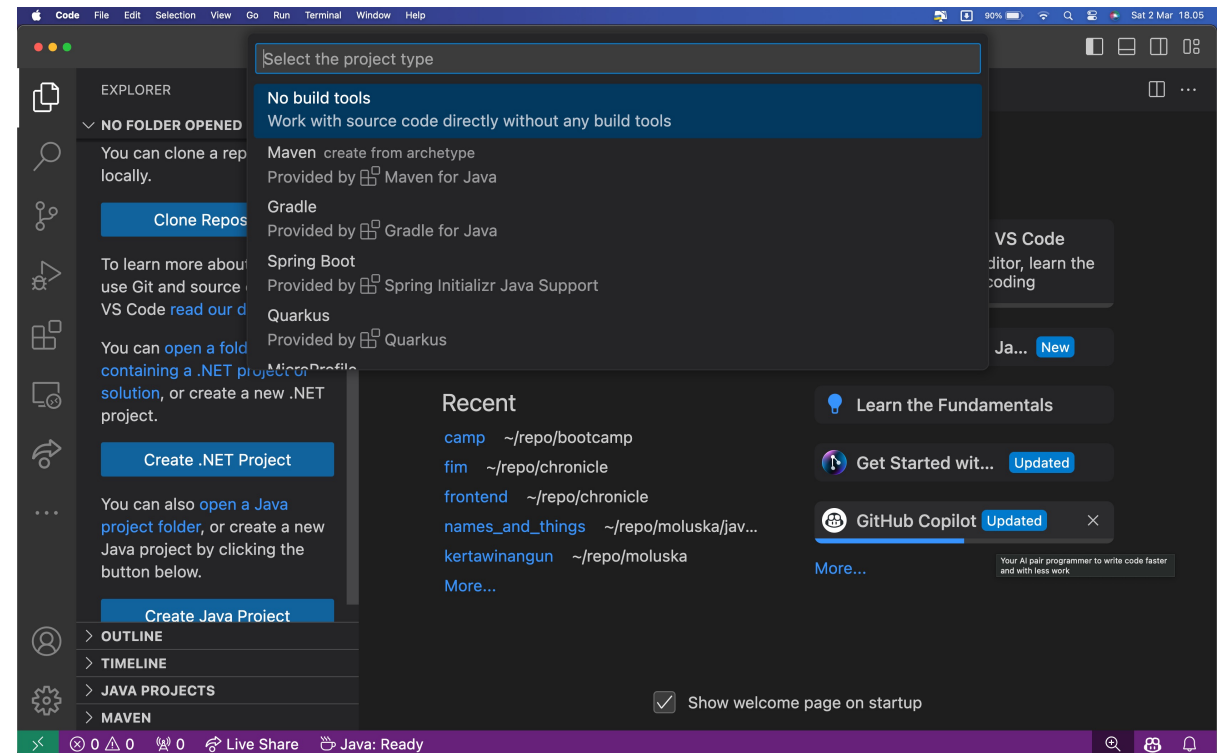
SETTING UP CAMP

TOOLS & TRADE

- Java Development Kit (JDK)
 - JDK adalah Lingkungan Pengembangan (Development Environment) yang digunakan untuk membangun aplikasi dan komponen-komponen menggunakan bahasa pemrograman Java.
 - Pada bootcamp ini kita akan menggunakan JDK 21 yang dapat di unduh pada link berikut;
<https://www.oracle.com/java/technologies/downloads/#java21>
- Integrated Development Environment (IDE)
 - IDE adalah software berbasis GUI yang digunakan sebagai tempat menuliskan setiap baris kode Bahasa pemrograman.
 - Untuk IDE pada bootcamp ini kita akan menggunakan Visual Studio Code (VSCode) dan beberapa extensionnya yang dibutuhkan.
 - VSCode bisa di unduh pada link berikut; <https://code.visualstudio.com/download>
 - Extension yang dibutuhkan bisa di unduh pada link berikut;
<https://marketplace.visualstudio.com/items?itemName=vscjava.vscode-java-pack>

FIRST ENCOUNTER

- Menulis dan menjalankan program java pertamamu.
 - Untuk menulis program kita akan menggunakan VSCode.
 - Tekan Tombol “**Create Java Project**” dan lanjutkan dengan memilih project type “**No build tools**” lalu setelah itu pilih folder dimana project akan disimpan.
 - VSCode akan membuatkan beberapa file dan struktur folder. Kita akan fokus pada folder “**src**” dimana kode akan kita tulis.
 - Untuk menjalankan program, kita bisa menekan tombol **F5** atau pilih menu **Run** pada VSCode.



HELLO, WORLD!

The screenshot displays the Visual Studio Code interface with a project named 'camp'. The Explorer sidebar on the left shows the file structure: `.vscode`, `bin`, `lib`, `src`, `App.java`, and `README.md`. The `App.java` file is open in the editor, showing the following code:

```
src > J App.java > App
1 public class App {
2     public static void main(String[] args) throws Exception {
3         System.out.println(x:"Hello, World!");
4     }
5 }
```

The bottom panel contains the TERMINAL view, showing a series of commands and their outputs:

```
~/r/b/camp ls
README.md bin lib src
~/r/b/camp
* History restored
~/r/b/camp
```

On the right side of the terminal, there are three status messages indicating successful Python executions:

- ✓ < base Py < at 05:37:34 PM
- ✓ < base Py < at 05:37:38 PM
- ✓ < base Py < at 06:08:02 PM

The status bar at the bottom indicates the current cursor position is at Line 5, Column 2, with 4 spaces, UTF-8 encoding, LF line endings, and the Java language mode. It also shows that Prettier is installed and ready.

HELLO, WORLD!

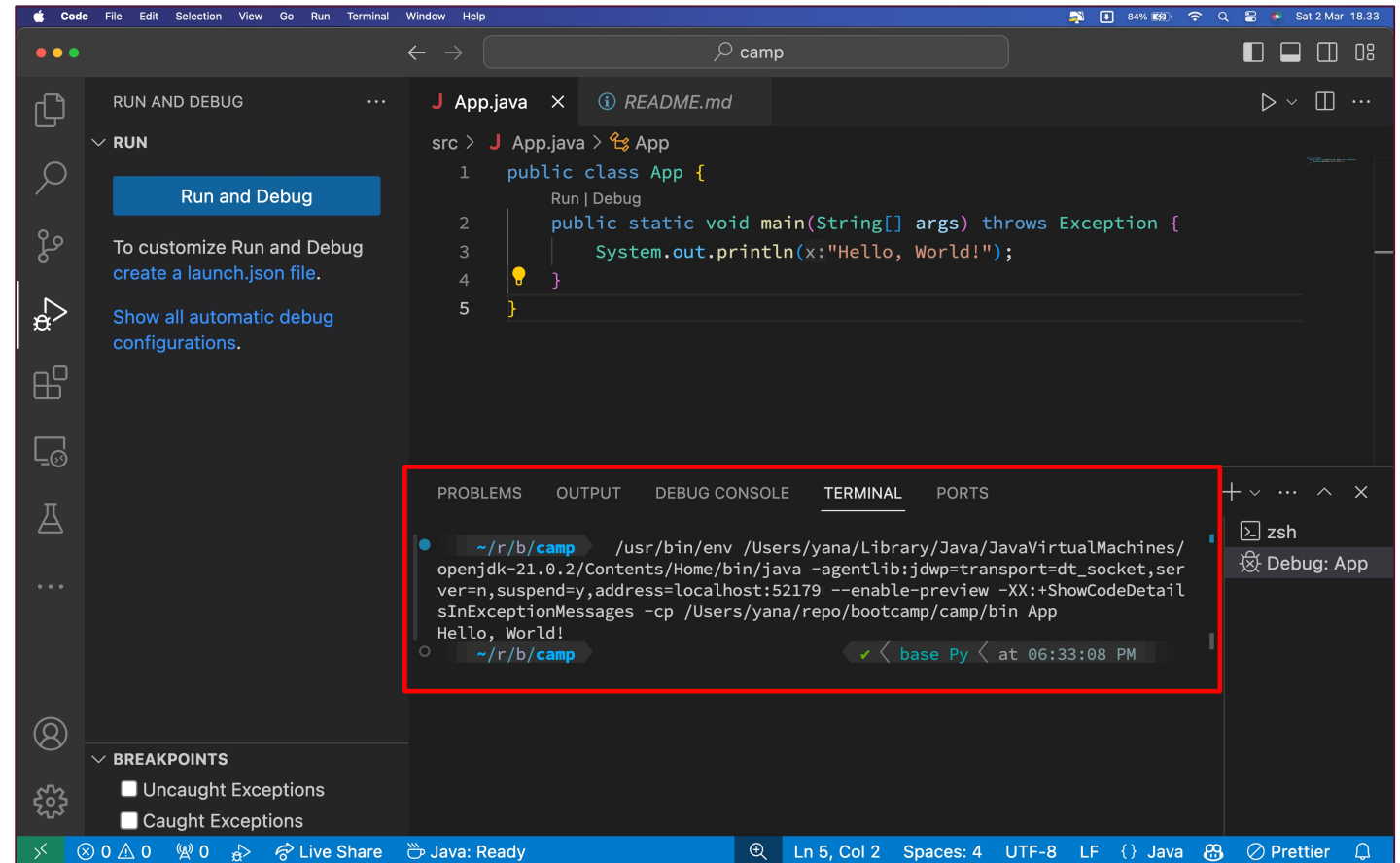
- `public class App {...}`
- Pada Bahasa pemrograman Java, setiap aplikasi akan dimulai oleh sebuah class definition dan nama class definition harus selalu sama dengan nama file dalam hal ini nama classnya adalah `App` dan nama filenya adalah `App.java`
- Class adalah prototype, atau blueprint, atau rancangan yang mendefinisikan variable dan method-methode pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan di jalankan, di dalamnya berisi atribut / type data dan method untuk menjalankan suatu program.

HELLO, WORLD!

- `public static void main(String[] args) throws Exception { ... }`
- Ini adalah method utama.
- Setiap program pada Java harus memiliki method utama (main method).
- Java compiler (yang merupakan salah satu komponen dalam JDK) akan menerjemahkan dan menyusun kode Java dimulai dari method utama (main method).
- **Kompilator** atau **kompiler** (*compiler*) adalah sebuah program komputer yang berguna untuk menafsirkan program komputer yang ditulis dalam bahasa pemrograman tertentu (bahasa *asal*) menjadi program yang ditulis dalam bahasa pemrograman lain (bahasa *sasaran*).
- Dalam hal ini, Bahasa Java akan di-*compile* menjadi Bahasa *Assembly* yang dimengerti oleh komputer.
- `throws Exception`. Di Java, Exception memungkinkan kita untuk menulis kode berkualitas baik di mana kesalahan diperiksa pada waktu kompilasi.

HELLO, WORLD!

- `System.out.println("Hello,World!");`
- Kode ini merupakan print statement untuk mencetak pada console/terminal. Dalam hal ini, kode tersebut akan mencetak text “Hello,World!” pada terminal.



SURVIVAL SKILLS (JAVA SYNTAX AND TYPES)

THE LANGUAGE OF THE JUNGLE

VARIABLES AND LITERALS

- Variable adalah lokasi dalam memori untuk menyimpan sebuah data.
- Untuk menunjukkan tempat penyimpanannya, setiap variabel harus diberi nama unik (identifier).
- Penamaan variable harus mengikuti standard yang sudah ditentukan oleh Java;
 - Java adalah Bahasa yang case sensitive yang berarti jika ada dua buah variable semisal `usia` dan `USIA` maka keduanya adalah merupakan dua data yang sama sekali berbeda.
 - Sebuah variable harus dimulai dengan huruf, underscore atau dollar sign \$
 - Variable tidak boleh dimulai dengan angka.
 - Variable tidak boleh mengandung whitespace (spasi)
 - Saat menulis sebuah variable sangat disarankan untuk memilih nama variable yang masuk akal seperti `usia` untuk menyimpan data usia, `nama` untuk menyimpan data nama. Tidak disarankan memilih nama yang tidak terlalu masuk akal seperti `u` untuk usia atau `n` untuk nama.

VARIABLES AND LITERALS

- Literals adalah data yang digunakan untuk mewakili nilai tetap. Literal dapat digunakan langsung dalam kode.
 - Boolean Literals, adalah literals yang digunakan untuk menyimpan dua buah nilai yaitu true dan false.
 - Integer Literals, adalah nilai numerik (terkait dengan angka) tanpa bagian pecahan atau eksponensial.
 - Ada 4 jenis literal integer di Java:
 - binary (base 2)
 - decimal (base 10)
 - octal (base 8)
 - hexadecimal (base 16)
 - Floating-point Literals, adalah literals numerik yang memiliki bentuk pecahan atau eksponensial.
 - Character Literals, adalah sebuah karakter unicode yang diapit dalam tanda kutip tunggal.
 - String Literals, adalah rangkaian karakter yang diapit tanda kutip ganda.

JAVA DATA TYPES (PRIMITIVE)

- Boolean, hanya dapat diisi dengan nilai true atau false
- Byte, dapat diisi dengan nilai -128 sampai 127 (8-bit)
- Short, dapat diisi dengan nilai -32768 sampai 32767 (16-bit)
- Int, dapat diisi dengan nilai -2^{31} sampai $2^{31}-1$ (32-bit)
- Long, dapat diisi dengan nilai -2^{63} sampai $2^{63}-1$ (64-bit)
- Double, double-precision 64-bit floating-point.
- Float, single-precision 32-bit floating-point.
- Char, 16-bit Unicode character.
- String, Java bisa mensupport tipe data string tapi string pada Java bukan tipe data primitive melainkan sebuah Objek.

JAVA OPERATORS

- Arithmetic Operators, adalah operator yang digunakan untuk operasi aritmatika

Operator	Operation
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus (Sisa pembagian dari bilangan terhadap bilangan lain)

JAVA OPERATORS

- Assignment Operators, digunakan di Java untuk menetapkan nilai kedalam variabel.

Operator	Contoh	Ekuivalen dengan
=	a = b;	a = b;
+=	a += b;	a = a + b;
-=	a -= b;	a = a - b;
*=	a *= b;	a = a * b;
/=	a /= b;	a = a / b;
%=	a %= b;	a = a % b;

JAVA OPERATORS

- Relational Operators, digunakan untuk memeriksa hubungan antara dua operan.

Operat or	Deskripsi	Contoh
==	Sama dengan	3 == 5 returns false
!=	Tidak sama dengan	3 != 5 returns true
>	Lebih dari	3 > 5 returns false
<	Kurang dari	3 < 5 returns true
>=	Lebih dari atau sama dengan	3 >= 5 returns false
<=	Kurang dari atau sama dengan	3 <= 5 returns true

JAVA OPERATORS

- Logical Operators, digunakan untuk memeriksa apakah suatu ekspresi benar atau salah. Operator ini digunakan dalam pengambilan keputusan.

Operator	Contoh	Meaning
&& (Logical AND)	expression1 && expression2	true hanya jika expression1 dan expression2 bernilai true
(Logical OR)	expression1 expression2	true jika salah satu dari expression1 atau expression2 bernilai true
! (Logical NOT)	!expression	true jika expression bernilai false dan vice versa

JAVA OPERATORS

- Unary Operators, digunakan hanya dengan satu operand. Misalnya, ++ adalah operator unary yang meningkatkan nilai variabel sebesar 1. Artinya, ++5 akan menghasilkan 6.

Operator	Meaning
+	Unary plus: Tidak dibutuhkan karena secara default sebuah bilangan akan bernilai positif
-	Unary minus: membalikkan tanda suatu ekspresi
++	Increment operator: menambahkan nilai dengan angka 1
--	Decrement operator: mengurangi nilai dengan angka 1
!	Logical complement operator: membalikkan nilai sebuah boolean

JAVA OPERATORS

- Bitwise Operators, Operator bitwise di Java digunakan untuk melakukan operasi pada bit individual.

Operator	Description
~	Bitwise Complement
<<	Left Shift
>>	Right Shift
>>>	Unsigned Right Shift
&	Bitwise AND
^	Bitwise exclusive OR

JAVA BASIC OUTPUT

- Dalam Bahasa Java kamu bisa menggunakan sintax ini untuk output: `System.out.println();` atau `System.out.print();` atau `System.out.printf();`
- `print()` - Mencetak string didalam sebuah double quotes.
- `println()` - sama dengan `print()` Mencetak string didalam sebuah double quotes. Lalu memindahkan kursor ke awal sebuah line selanjutnya.
- `printf()` digunakan untuk melakukan string formatting

JAVA BASIC INPUT

- Java memiliki banyak cara untuk mendapatkan input salah satunya dengan menggunakan **Scanner**. Contoh,
`Scanner input = new Scanner(System.in);`

JAVA EXPRESSIONS, STATEMENTS AND BLOCKS

- Didalam sebuah expression tergabung didalamnya berbagai variable, operator, literal dan method call.
- Setiap statement merupakan sebuah unit komplit (complete unit) dari setiap execution.
- Sebuah Block adalah sekelompok statement (satu atau lebih) yang ada didalam curly brace {}