

2. Pengenalan Java

By : Mahendar Dwi Payana, S.ST., M.T

Prasyarat :

1. [Pengenalan Algoritma dan Flowchart](#)

Referensi :

1. Buku "Pemrograman Berorientasi Objek. Teori dan Implementasi Java"
2. Pengarang : Raden Budiarto
3. Tahun : 2018
4. Jenis : E-Book - [ebook play.books.com](http://ebook.play.books.com)

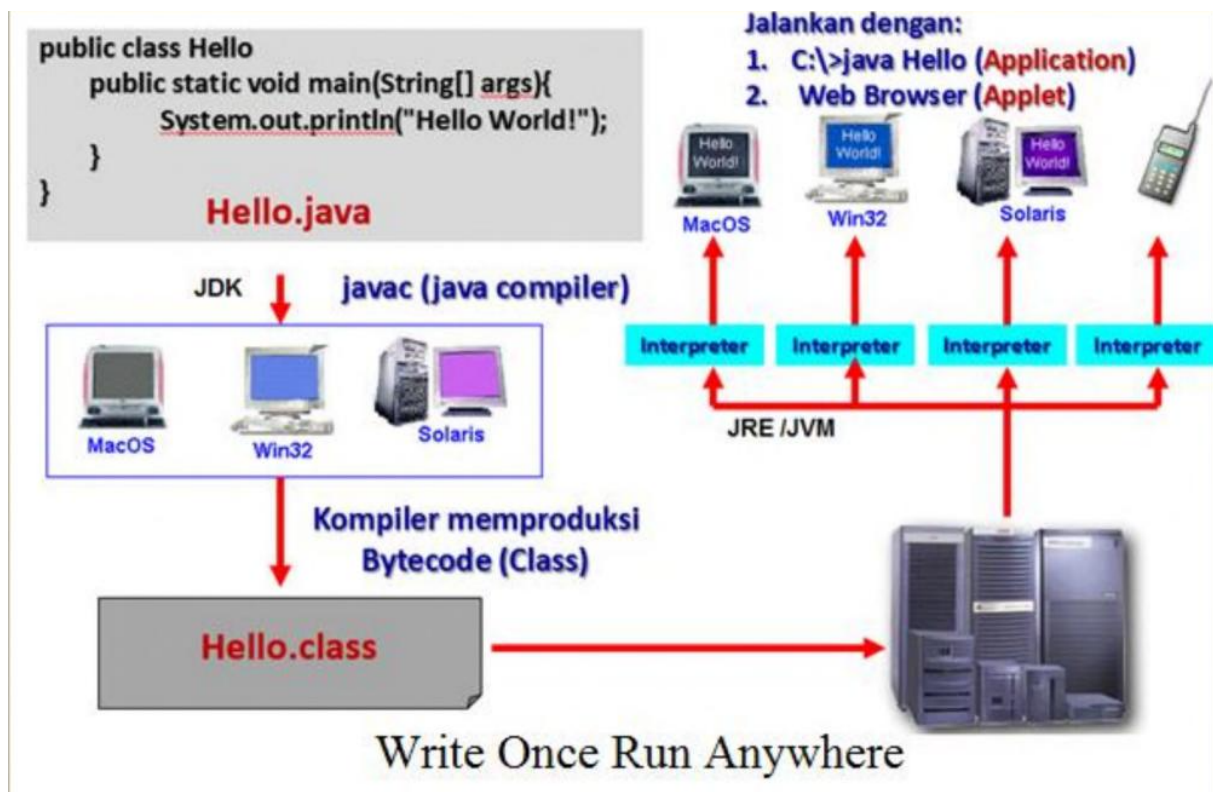
2.1 Pengenalan Java

Bahasa pemrograman **Java** merupakan bahasa pemrograman yang dikembangkan oleh **Sun Microsystems** pada tahun 1991. Bahasa pemrograman ini dikembangkan oleh **James Gosling** dan rekan-rekannya. Bahasa pemrograman ini merupakan bahasa pemrograman yang berorientasi objek. Bahasa pemrograman ini banyak digunakan untuk membuat aplikasi berbasis desktop, web, dan mobile. Bahasa pemrograman ini juga banyak digunakan untuk membuat aplikasi berbasis **Internet of Things (IoT)**.

2.1.1 Konsep Dasar Java

Pemrograman **Java** merupakan pemrograman compiler yang berarti kode program yang dibuat harus diubah menjadi kode mesin terlebih dahulu sebelum dieksekusi. Kode mesin yang dihasilkan oleh compiler Java disebut dengan **bytecode**. Program yang dihasilkan oleh compiler Java dapat dijalankan di berbagai platform yang mendukung Java Virtual Machine (JVM). **JAVA** dicompile menjadi **bytecode** oleh **Java Compiler** yaitu **javac**, kemudian **bytecode** tersebut diinterpretasikan oleh **Java Virtual Machine** yaitu **java**.

Untuk menjalankan program Java, kita membutuhkan beberapa perangkat lunak yaitu **JRE** atau *Java Runtime Environment*. Kemudian untuk dapat membuat aplikasi Java dibutuhkan **JDK** atau *Java Development Kit*. JDK terdiri dari beberapa perangkat lunak yaitu **JRE**, **Java Compiler**, dan **Java Debugger**. Seluruh perangkat lunak untuk keperluan pengembangan menggunakan Java dapat diunduh di <https://www.oracle.com/java/technologies/javase-downloads.html>.



Gambar 2.1 Cara Kerja Java

2.1.2 Struktur Program Java

Program Java terdiri dari beberapa bagian yaitu :

- **Package** : Bagian ini berisi nama package yang digunakan untuk mengelompokkan kelas-kelas yang dibuat.
- **Import** : Bagian ini berisi nama package yang digunakan untuk mengimport package lain yang digunakan dalam program.
- **Class** : Bagian ini berisi kelas-kelas yang dibuat dalam program.
- **Method** : Bagian ini berisi method-method yang dibuat dalam program.
- **Statement** : Bagian ini berisi statement-statement yang dibuat dalam program.
- **Expression** : Bagian ini berisi expression-expression yang dibuat dalam program.
- **Comment** : Bagian ini berisi komentar-komentar yang dibuat dalam program.
- **Whitespace** : Bagian ini berisi spasi, tab, dan baris baru yang digunakan untuk memudahkan pembacaan program.
- **Identifier** : Bagian ini berisi nama-nama yang digunakan dalam program.
- **Literals** : Bagian ini berisi nilai-nilai yang digunakan dalam program.
- **Keyword** : Bagian ini berisi kata-kata yang digunakan dalam program.
- **Operator** : Bagian ini berisi operator-operator yang digunakan dalam program.
- **Separator** : Bagian ini berisi separator-separator yang digunakan dalam program.
- **Semicolon** : Bagian ini berisi tanda titik koma yang digunakan dalam program.
- **Block** : Bagian ini berisi blok-blok yang digunakan dalam program.
- **Statement Termination** : Bagian ini berisi tanda titik koma yang digunakan dalam program.

- **Case Sensitivity** : Bagian ini berisi sensitivitas huruf besar dan huruf kecil yang digunakan dalam program.
- **Reserved Word** : Bagian ini berisi kata-kata yang tidak boleh digunakan dalam program.

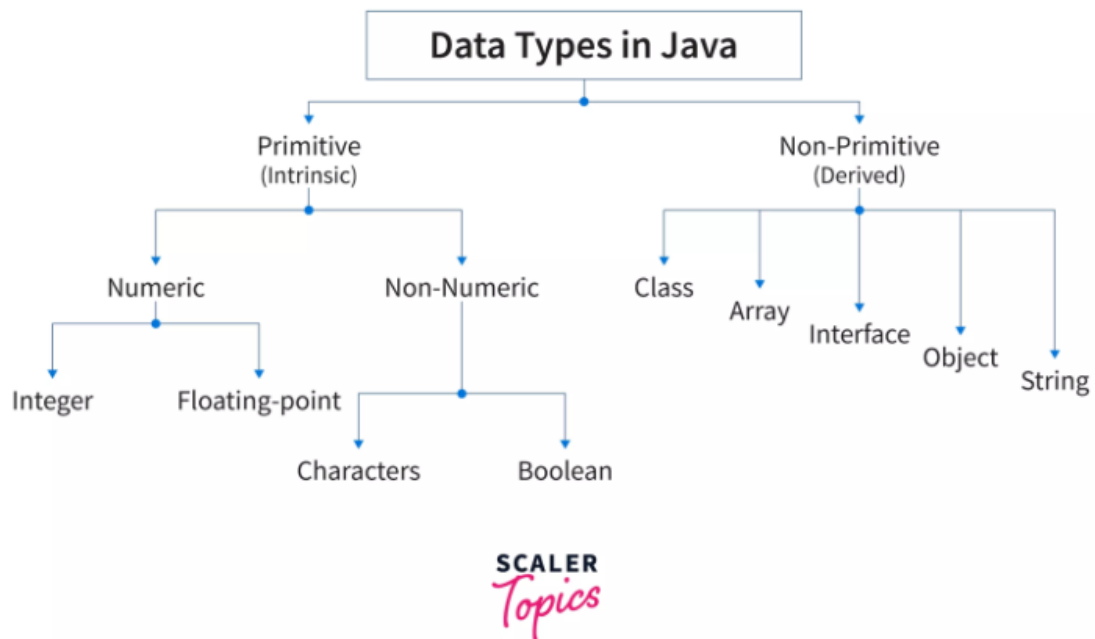
2.2 Struktur Data Java

2.2.1 Tipe Data Java

Setiap bahasa pemrograman memiliki beberapa tipe data (*Data Type*) untuk menggolongkan data yang nantinya akan diproses oleh bahasa pemrograman tersebut. Terdapat dua jenis tipe data yang terdapat pada keyword java (tertanam pada java compiler). Tipe data tersebut adalah tipe data primitif dan tipe data non-primitif.

- **Tipe Data Primitif** : Tipe data primitif merupakan tipe data yang sudah didefinisikan oleh Java. Tipe data primitif terdiri dari :
 - **Integer** : Tipe data ini digunakan untuk menyimpan bilangan bulat. Tipe data ini terdiri dari :
 - **byte** : Tipe data ini digunakan untuk menyimpan bilangan bulat dengan ukuran 8 bit.
 - **short** : Tipe data ini digunakan untuk menyimpan bilangan bulat dengan ukuran 16 bit.
 - **int** : Tipe data ini digunakan untuk menyimpan bilangan bulat dengan ukuran 32 bit.
 - **long** : Tipe data ini digunakan untuk menyimpan bilangan bulat dengan ukuran 64 bit.
 - **Floating Point** : Tipe data ini digunakan untuk menyimpan bilangan pecahan. Tipe data ini terdiri dari :
 - **float** : Tipe data ini digunakan untuk menyimpan bilangan pecahan dengan ukuran 32 bit.
 - **double** : Tipe data ini digunakan untuk menyimpan bilangan pecahan dengan ukuran 64 bit.
 - **Boolean** : Tipe data ini digunakan untuk menyimpan nilai kebenaran. Tipe data ini terdiri dari :
 - **boolean** : Tipe data ini digunakan untuk menyimpan nilai kebenaran dengan ukuran 1 bit.
 - **Character** : Tipe data ini digunakan untuk menyimpan karakter. Tipe data ini terdiri dari :
 - **char** : Tipe data ini digunakan untuk menyimpan karakter dengan ukuran 16 bit.
- **Tipe Data Non-Primitif** : Tipe data non primitif merupakan tipe data yang dibuat oleh pengguna. Tipe data non primitif terdiri dari :
 - **String** : Tipe data ini digunakan untuk menyimpan karakter dengan ukuran yang tidak terbatas.
 - **Object** : Tipe data ini digunakan untuk menyimpan objek.
 - **Interface** : Tipe data ini digunakan untuk menyimpan interface.

- **Array** : Tipe data ini digunakan untuk menyimpan array.
- **Class** : Tipe data ini digunakan untuk menyimpan kelas.



Gambar 2.2 Tipe Data Pada Java

2.2.2 Ukuran Tipe Data Java

Dalam bahasa pemrograman setiap tipe data memiliki batasan ukuran memori. Hal ini agar data yang akan dikelola menjadi lebih terstruktur sehingga proses yang akan dilakukan oleh komputer akan lebih efektif dan efisien.

Terdapat beberapa ukuran tipe data dalam Java yaitu :

- Ukuran Data Tipe Data Primitif
 - **byte** : Ukuran tipe data ini adalah 8 bit.
 - **short** : Ukuran tipe data ini adalah 16 bit.
 - **int** : Ukuran tipe data ini adalah 32 bit.
 - **long** : Ukuran tipe data ini adalah 64 bit.
 - **float** : Ukuran tipe data ini adalah 32 bit.
 - **double** : Ukuran tipe data ini adalah 64 bit.
 - **boolean** : Ukuran tipe data ini adalah 1 bit.
 - **char** : Ukuran tipe data ini adalah 16 bit.

Tipe Data	Keterangan	Ukuran	Jangkauan
<i>byte</i>	Bilangan bulat	8 bit	-128 → 127
<i>short</i>	Bilangan bulat	16 bit	-32.768 → 32.767
<i>int</i>	Bilangan bulat	32 bit	-2.147.483.648 → 2.147.483.647
<i>long</i>	Bilangan bulat	64 bit	-9.223.372.036.854.775.808 → 9.223.372.036.854.775.807
<i>float</i>	Bilangan pecahan	32bit	-3.4E38 → 3.4E38
<i>double</i>	Bilangan pecahan	64bit	-1.7E308 → 1.7E308
<i>char</i>	Karakter (unicode)	16 bit	\u0000 → \uffff
<i>boolean</i>	Logika (true/false)		True/false

Gambar 2.3 Panjang Tipe Data Primitif

- Ukuran Data Tipe Data Non Primitive
 - **String** : Ukuran tipe data ini adalah tidak terbatas.
 - **Object** : Ukuran tipe data ini adalah tidak terbatas.
 - **Interface** : Ukuran tipe data ini adalah tidak terbatas.
 - **Array** : Ukuran tipe data ini adalah tidak terbatas.
 - **Class** : Ukuran tipe data ini adalah tidak terbatas.

2.3 Keyword pada Java

Java memiliki sejumlah kata yang bermakna khusus yang digolongkan sebagai kata kunci. Diantara kata kunci (*keyword*) tersebut adalah sebagai berikut :

abstract	private	case	interface
continue	this	enum	static
for	break	instanceof	void
new	double	return	class
switch	implements	transient	finally
assert	protected	catch	long
default	throw	extends	strictfp
package	byte	int	volatile
synchronized	else	short	float
boolean	import	try	native
do	public	char	super
if	throws	final	while

Gambar 2.4 Keyword pada Java

2.4 IDE Java IntelliJ IDEA

IDE atau *Integrated Development Environment* merupakan perangkat lunak yang digunakan untuk memudahkan pengembangan aplikasi. IDE memiliki beberapa fitur yang dapat memudahkan pengembangan aplikasi. Beberapa fitur tersebut adalah sebagai berikut :

- **Code Editor** : Fitur ini digunakan untuk menulis kode program.
- **Code Completion** : Fitur ini digunakan untuk menampilkan kode program yang dapat digunakan.
- **Code Generation** : Fitur ini digunakan untuk menghasilkan kode program.

- **Code Navigation** : Fitur ini digunakan untuk menavigasi kode program.
- **Code Refactoring** : Fitur ini digunakan untuk memperbaiki kode program.
- **Code Templates** : Fitur ini digunakan untuk menampilkan kode program yang dapat digunakan.
- **Code Inspection** : Fitur ini digunakan untuk memeriksa kode program.

2.4.1 Membuat Project Java di IntelliJ IDEA

Untuk membuat project Java di IntelliJ IDEA dapat dilakukan dengan langkah-langkah sebagai berikut :

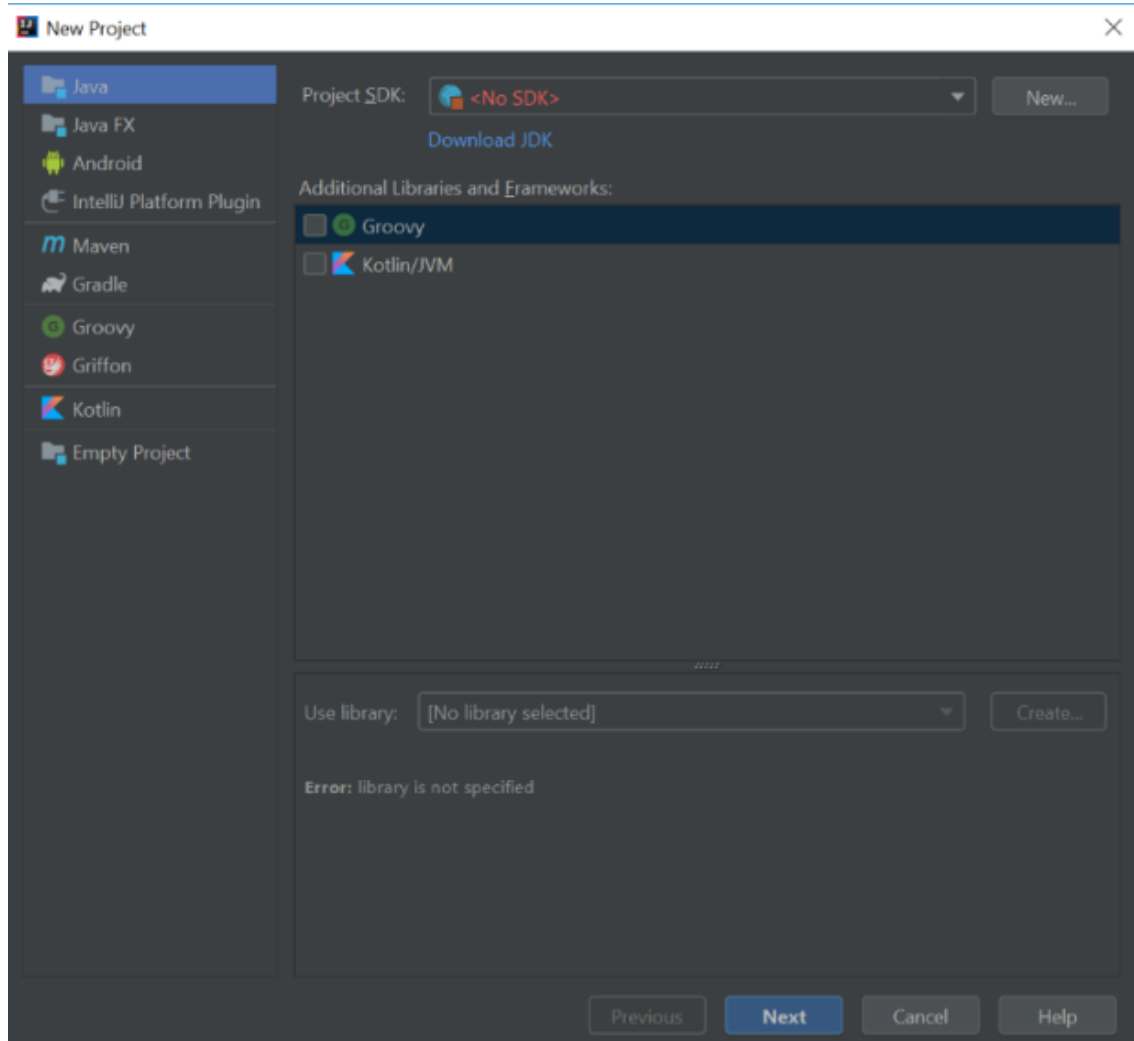
1. Buka IntelliJ IDEA.

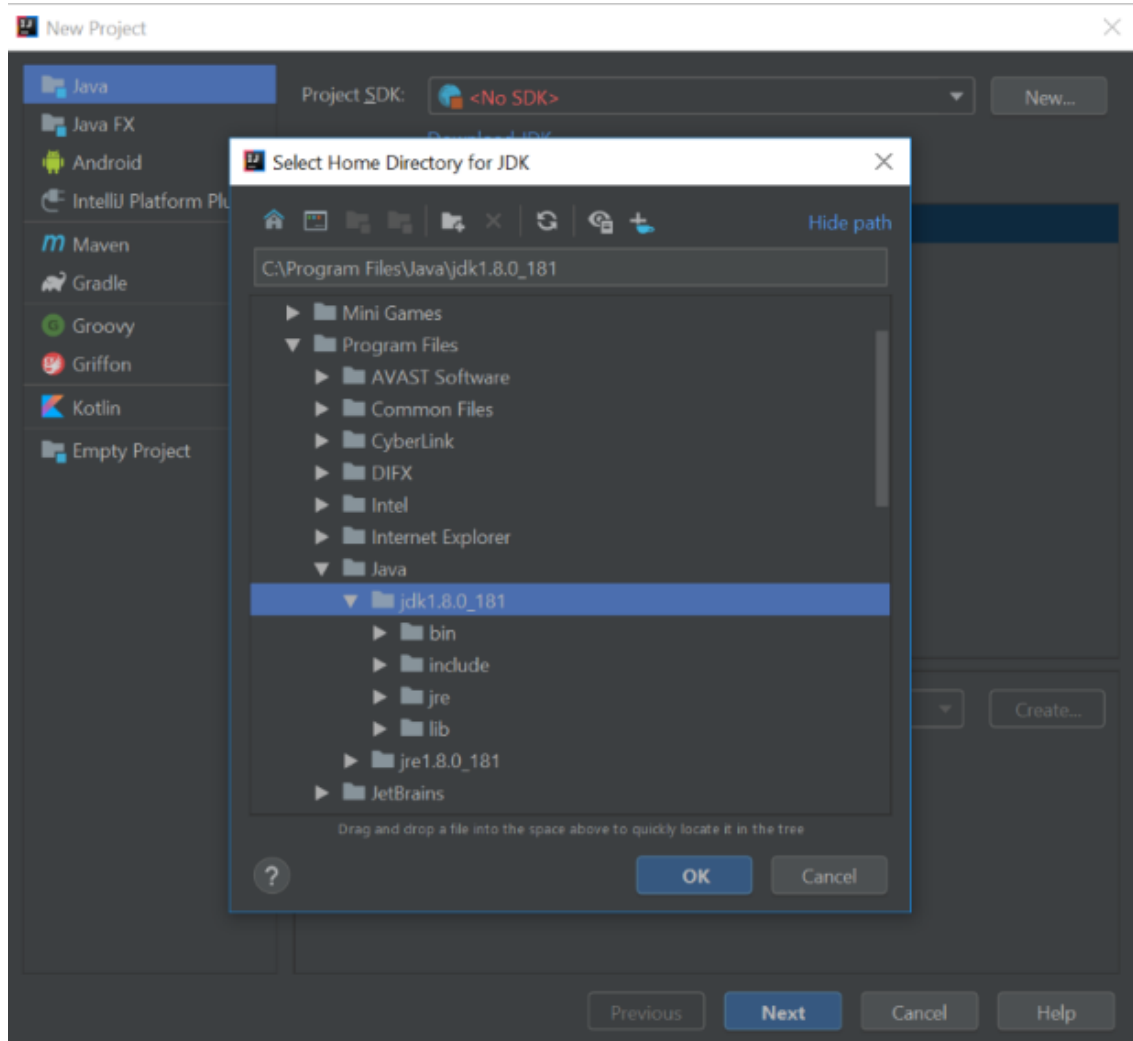


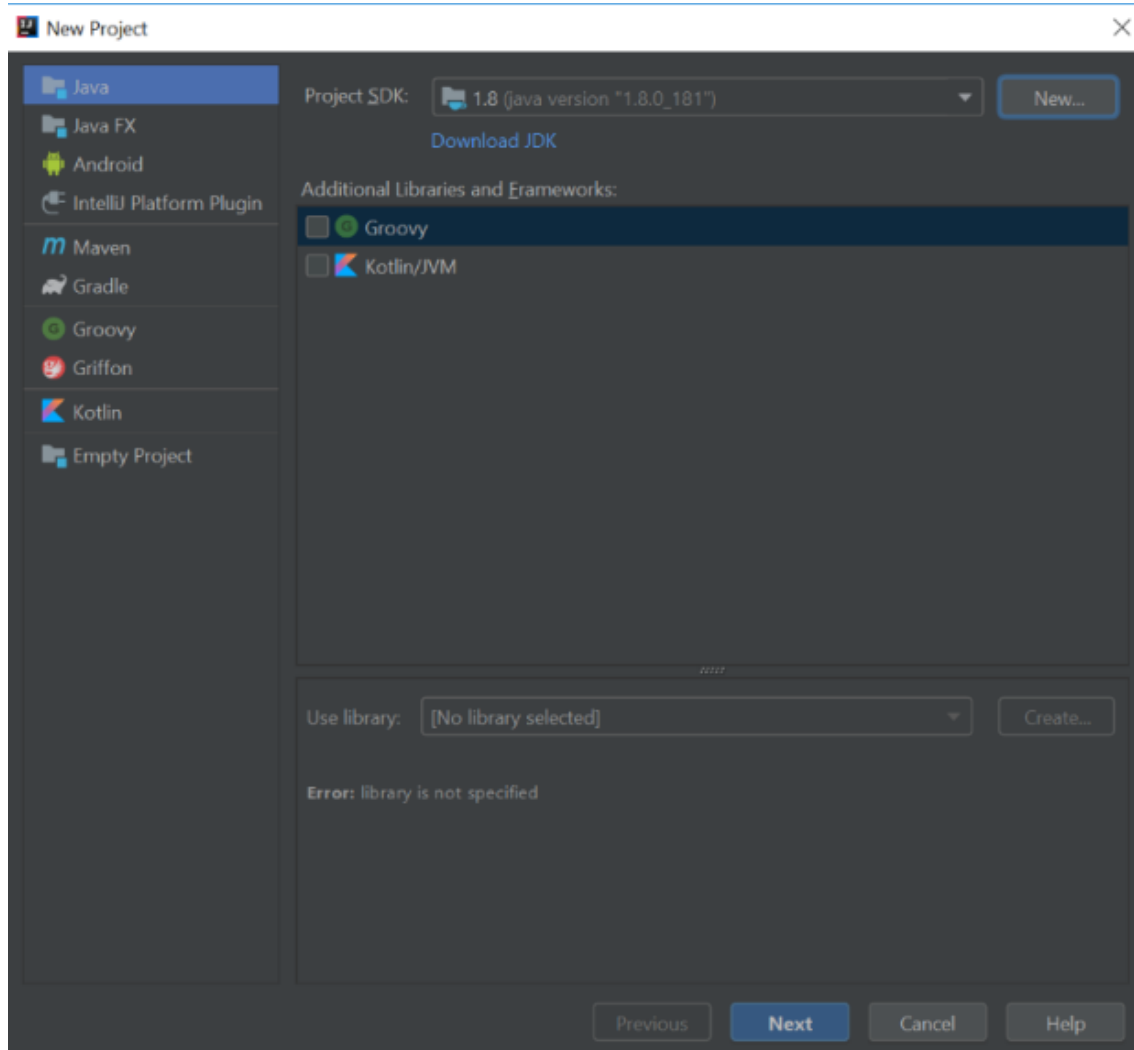
2. Pilih menu **File > New > Project**.

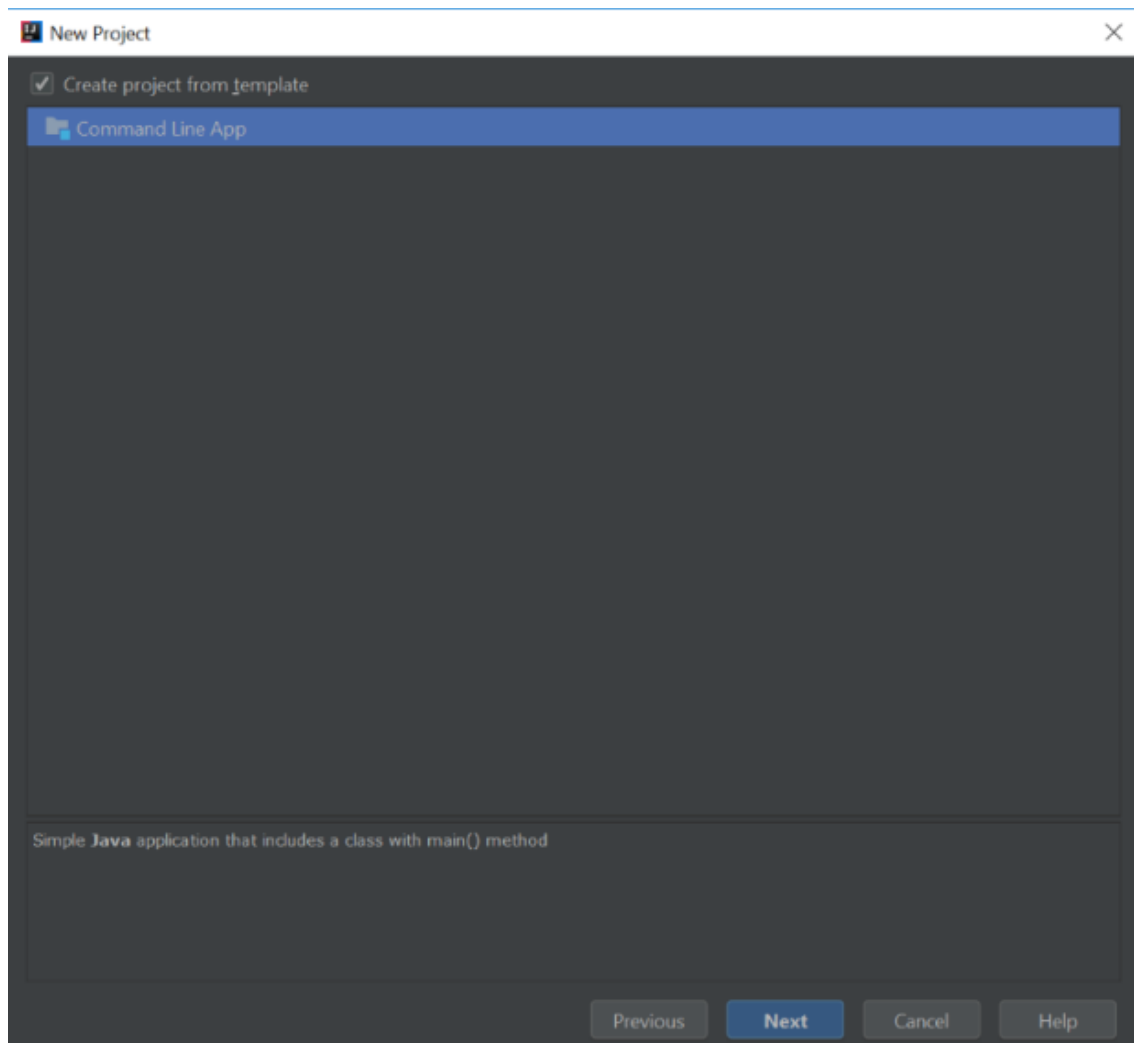


3. Pilih **Java** pada bagian **Project SDK**.

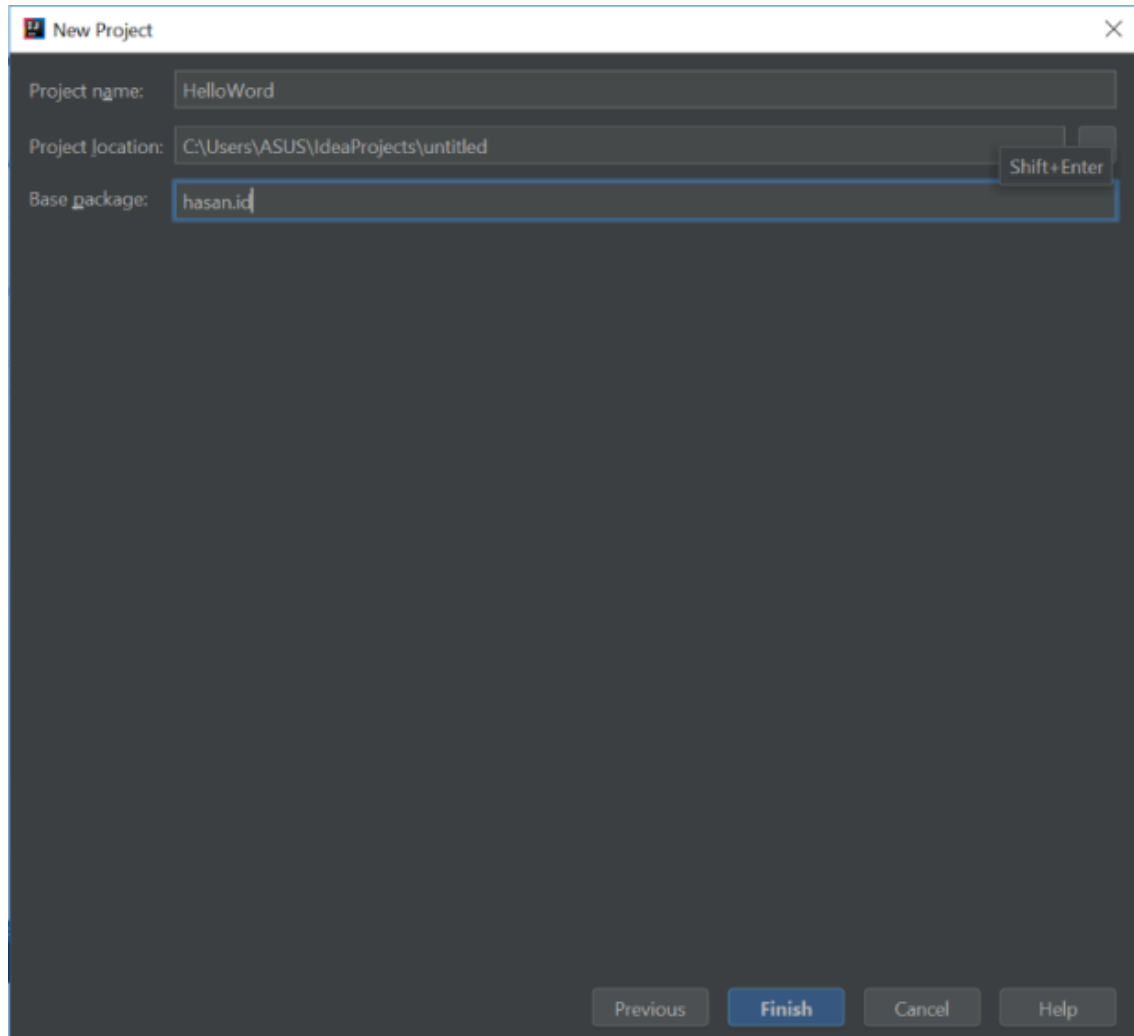


4. Pilih **Java** pada bagian **Project language level**.

5. Pilih **Next**.



6. Masukkan nama project pada bagian **Project name**.



7. Pilih **Finish**.