

# Object Oriented Programming

# Agenda

1

Intro

2

Java Architecture

3

Java Building Blocks

4

Accessing Class Members

---

Intro

---

# Intro

- Konsep OOP (Object Oriented Programming) cukup populer karena dapat mengurangi kompleksitas dari sebuah program.
- Java merupakan salah satu bahasa pemrograman yang menggunakan konsep ini.
- Beberapa fitur yang ada di Java yang cukup populer adalah:
  - Sederhana: Saat sebuah objek class dibuat, kita perlu mengalokasikan memori. Lalu, ketika objek tidak lagi digunakan, alokasi memori yang digunakan objek tadi akan dilepaskan. Di Java, alokasi memori dan deallocation terjadi secara otomatis. Proses deallocation memori secara otomatis inilah yang disebut dengan garbage collection.

# Intro

- **Berorientasi objek:** Di Java, program harus dikemas di dalam sebuah class. Kemudian, Java menggunakan objek untuk merefer entity dunia nyata seperti mahasiswa, mobil, dll.
- **Platform yang mandiri:** Ketika kita meng-compile sebuah program yang free-error, compiler di Java akan mengkonversi program ke sebuah platform kode yang mandiri dan disebut dengan bytecode. Kemudian, Java Virtual Machine (JVM) menerjemahkan bycode tersebut ke bahasa mesin yang bisa dijalankan di mesin. Hal ini merupakan salah satu keunggulan Java karena setiap komputer memiliki JVM.

# Intro

- **Portable:** Portability mengacu kepada kemampuan program untuk dijalankan di berbagai platform tanpa merubah source code dari program. Java. Sebuah program yang dibuat di sebuah komputer dapat berjalan di komputer lain meski memiliki OS yang berbeda. Java mengizinkan cross-platform semacam ini.
- **Terdistribusi:** Java dapat digunakan untuk membuat aplikasi yang bisa di distribusikan di berbagai mesin di jaringan seperti di internet.
- **Secure:** Java memiliki fitur built-in security yang memastikan bahwa program Java memberikan akses hanya kepada yang berhak.

# Intro

- **Robust:** Java memiliki beragam fitur seperti memory management dan exception handling yang membuatnya menjadi sebuah bahasa pemrograman yang kuat/ robust.
- **Multithreaded:** Java juga menyediakan konsep multithreading yang mengizinkan sebuah program untuk dieksekusi secara simultan/ bersamaan dengan memanfaatkan threads. Threads merupakan unit terkecil dari eksekusi program di Java.

# Java Architecture

---



# Java Architecture

Java Architecture menjelaskan tentang komponen yang terkait pembuatan dan eksekusi code program Java, Komponen ini antara lain:

- **Source file:** Aplikasi Java terdiri dari source code yang mengizinkan sebuah aplikasi memiliki fungsi-fungsi yang dibutuhkan. Source code tersebut disimpan dengan ekstensi .java.
- **Class file:** Ketika kita membuat file .java, selanjutnya file tersebut akan di compile untuk membuat sebuah file .class. File class ini berisikan bytecode yang dibuat oleh compiler.
- **JVM:** JVM merupakan sebuah aplikasi yang bertanggung jawab untuk mengeksekusi program Java disebuah komputer. Komponen utama dari JVM ini adalah Class loader, Execution engine dan Just-In Time (JIT) compiler, dan bytecode verifier.

# Java Architecture

- **API:** Java API merupakan kumpulan komponen software seperti class-class, interface-interface, dan method-method yang menyediakan beragam capability seperti GUI, Date and Time, dan Calendar.

# Java Building Blocks

---

# Java Building Blocks

- Program Java terdiri dari building blocks seperti classes, data types, class members, dan packages.
- CLASS
- Sebuah class merupakan karakteristik dan sifat dari sebuah objek. Sebuah class memiliki variabel-variabel dan fungsi dengan karakteristik yang sama. Kita bisa membuat sebuah class dengan cara berikut:

```
class <nama_class>{  
    // deklarasi member dari variabel dan method  
}
```

- Contoh:

```
class bunga{  
    String nama;  
}
```

# Java Building Blocks

- Terdapat beberapa aturan yang harus kita ikuti dalam membuat nama class di Java, dimana jika aturan ini tidak diikuti akan memunculkan error. Aturan tersebut antara lain:
  - Nama class tidak boleh menggunakan spasi dan simbol seperti ?, !, #, @, %, &, {}, [], :, ;, ", dan /.
  - Nama class harus unik.
  - Nama class harus diawali dengan huruf atau underscore, atau simbol dolar (\$).
  - Nama class tidak boleh sebuah keyword.

# Java Building Blocks

## Java Keyword List

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

# Java Building Blocks

- Cara terbaik membuat nama sebuah class Java adalah menggunakan kata benda, huruf awal sebaiknya huruf kapital, dan jika nama class terdiri dari beberapa kata, maka setiap awalan kata dijadikan kapital.

# Java Building Blocks

- **DATA TYPES**
- Ketika kita membuat sebuah aplikasi, kita akan menyimpan dan memanipulasi beragam data. Beragam data ini memiliki tipe yang berbeda-beda.
- Terdapat delapan primitive data type dan dikelompokkan sebagai berikut:
  - **Integer data type:** menyimpan nilai integer dan dibagi menjadi empat yakni byte, short, int, dan long.
  - **Floating point type:** Dapat menyimpan angka desimal dan dibagi menjadi dua jenis, yakni float dan double.
  - **Boolean type:** menyimpan nilai TRUE/ FALSE.
  - **Character type:** Dapat menyimpan karakter tunggal seperti simbol, huruf, dan angka.



# Java Building Blocks

- DATA TYPES

Data Type	Size	Range of values that can be stored	Default value
byte	1 byte	-128 to 127	0
short	2 bytes	-32768 to 32767	0
int	4 bytes	-2,147,483,648 to 2,147,483,647	0
long	8 bytes	-9,223,372,036,854,775,808 to 9223372036854750000	0
float	4 bytes	3.4e-038 to 3.4e+038	0.0f
double	8 bytes	1.7e-308 to 1.7e+038	0.0d
boolean	1 bit	true or false	false
char	2 bytes		\u0000

# Java Building Blocks

- **PACKAGE**
- Package merupakan kumpulan class-class.
- Package menyediakan space yang digunakan untuk mengatur class-class yang saling terkait satu sama lain.
- Sebuah class hanya boleh berada di satu package.

# Accessing Class Members

---

# Accessing Class Members

- Class member merupakan karakteristik dan sifat dari sebuah objek class.
- Kita bisa menyembunyikan atau melindungi beberapa class members agar tidak bisa diakses oleh class lain.
- Proses menyembunyikan class members ini disebut juga dengan enkapsulasi.
- Untuk meningkatkan security di sebuah Java class, kita dapat menggunakan access specifier.
- Access specifier mengontrol akses ke class members dan terdiri dari:
  - **Private:** merupakan akses yang membuat class members hanya

# Accessing Class Members

- **Access specifier mengontrol akses ke class members dan terdiri dari:**
  - **Private:** merupakan akses yang membuat class members hanya terlihat oleh class yang sama.
  - **Protected:** class members yang memiliki akses protected hanya bisa di akses oleh class di dalam package yang sama atau subclass di luar package.
  - **Public:** class members dengan akses ini bisa di akses oleh class lain di dalam maupun luar package yang sama.

# Accessing Class Members

- Sebuah class bisa mengakses members dari class lain dengan menggunakan objek.
- Cara untuk membuat objek class adalah:

```
Nama_Class <nama_objek> = new Nama_Class();
```

**Contoh:**

```
Bunga obj = new Bunga();
```

# Accessing Class Members

Thank You!

---

*I Am Who I'm Believe I Am*