

Glosarium dicoding

Abstract Class

Sebuah *class* yang ditandai dengan keyword “abstract”, yang mempunyai beberapa properti dan atau fungsi yang belum terdapat implementasinya. Class ini harus diimplementasi terlebih dahulu oleh *class* lainnya agar bisa diinisiasi.

Android

Sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.

Attribute

Merepresentasikan karakteristik dari suatu objek atau properti dari sebuah objek.

Anonymous Class

Sub-class dari hasil implementasi sebuah *interface* atau *abstract class* pada suatu *variable*, maupun pada *function argument*. Class ini berupa objek dari *interface/abstract class* tersebut dan tidak memiliki nama tersendiri seperti *class* yang lain

Anonymous Function

Fungsi yang dideklarasikan tanpa nama.

Argument

Sebuah nilai yang akan disematkan atau diteruskan ke dalam parameter sebuah fungsi.

Asynchronous

Proses yang berjalan tidak berurutan, bersamaan/berbarengan dalam suatu waktu.

Ascending

Proses urutan dari nilai terendah ke nilai tertinggi.

B

Bug

Segala bentuk kecacatan dalam suatu aplikasi.

Branch

Cabang atau kondisi yang sudah ditentukan sebelumnya.

Blok Kode/Body

Area dalam kode yang didalamnya terdapat beberapa *statement* atau instruksi kode.

Blueprint

Rancangan yang mendefinisikan variabel dan fungsi-fungsi yang ada pada sebuah objek.

Behaviour

Serangkaian tindakan oleh sebuah entitas buatan yang berhubungan dengan dirinya sendiri.

Boilerplate Code

Penulisan baris kode berulang di banyak tempat tanpa adanya perubahan.

Bottleneck

Sebuah kondisi yang akan menyebabkan sebuah proses terhambat atau menurunkan performa.

Backward Compatibility

Dukungan sebuah sistem, *product*, atau teknologi yang dapat dijalankan dengan baik pada sistem versi lama.

C**Callback**

Fungsi yang akan diteruskan sebagai argumen ke kode lain yang diharapkan untuk dieksekusi sebagai argumen pada waktu yang tepat.

Camel Case

Teknik atau ketetapan penulisan dalam bahasa kotlin, di mana setiap awal dari sebuah kata menggunakan huruf kapital dan ditulis tanpa menggunakan spasi (antar kata). Teknik ini digunakan untuk penamaan sebuah fungsi, *class*, dan juga *property*.

Casting

Proses konversi tipe data dari sebuah variabel menjadi tipe data lain yang memiliki kaitan atau hubungan dengan tipe data tersebut. Hubungan tipe data ini terbatas hanya *subclass* (anak) atau *parent class* (induk) dari tipe data tersebut.

Class

Class merupakan *blueprint* atau definisi struktur dari sebuah objek yang belum diinisiasi. Di dalamnya terdapat kumpulan properti dan fungsi yang mencerminkan kegunaan dari *class*.

Clean Code

Aturan dalam penulisan kode saat pengembangan aplikasi yang digagas oleh Robert C. Martin, di mana menerapkan beberapa teknik tertentu agar kode yang dituliskan bisa dengan mudah dibaca, ditulis, maupun diperbaiki.

Collection

Objek yang dapat digunakan untuk menyimpan sekumpulan objek.

Control Flow

Cara kita mengontrol alur dari sebuah program berdasarkan kondisi saat program tersebut berjalan.

Constant (Konstanta)

Tidak berubah-ubah nilainya.

Constructor

Method yang secara default sudah terbentuk ketika kelas dibuat. Ketika suatu kelas dibuat (instansiasi) maka konstruktor akan terpanggil juga.

Container

Jenis komponen yang menampung komponen lainnya.

Coroutines

Merupakan fitur dari bahasa Kotlin untuk melakukan suatu proses secara asynchronous tanpa mengemblok proses lainnya dengan menggunakan mekanisme *structured concurrency*. Berbeda dengan

Thread, ia sangat ringan sehingga ribuan Coroutine bisa dibuat dengan cepat.

Crash

Keadaan dari suatu sistem (aplikasi atau sistem operasi), di mana sistem tersebut tidak berfungsi dengan normal dan terpaksa mengakhiri berjalannya sistem tersebut

Curly Braces

Salah satu jenis dari tanda kurung yang ditandai dengan simbol '{' dan '}'. Digunakan untuk mengawali dan mengakhiri dari suatu blok kode dalam pemrograman.

D

Debug

Proses mengidentifikasi atau menghapus *error* pada sebuah perangkat keras atau perangkat lunak.

Debugger

Tools untuk mencari suatu *bug*.

Dispatcher

Utilitas dalam kotlin yang menentukan jenis thread atau utas apa yang akan digunakan untuk eksekusi.

Data Classes

Kelas sederhana yang bisa berperan sebagai *data container* pada Kotlin. Di dalamnya terdapat beberapa fungsi yang sudah disediakan untuk meng-*handle* beberapa operasi data seperti equals(), toString(), hashCode(), & copy()

Descending

Urutan terbalik dimulai dari nilai yang lebih besar.

E

Encapsulation

Konsep fundamental dalam pemrograman berbasis objek yang

membungkus data dengan beberapa *method/fungsi* yang mengelola data, dan membatasi akses data secara langsung dari luar unit (*class*).

Enum

Fitur yang bisa digunakan untuk menyimpan kumpulan objek yang telah didefinisikan menjadi tipe data konstanta.

Expression

Kombinasi dari satu atau lebih nilai eksplisit, konstanta, variabel, operator dan fungsi yang diinterpretasikan dan dihitung untuk menghasilkan nilai lain.

Exit Controlled Loop

Teknik eksekusi dalam perulangan, di mana blok kode atau statement dijalankan di awal, baru kemudian mengecek kondisi setelah blok/statement tersebut selesai dijalankan. Contohnya adalah perulangan *do ... while*

Extends

Kata kunci yang digunakan untuk mengimplementasikan konsep *inheritance*.

Exception

Kesalahan yang muncul saat aplikasi dijalankan.

Eager Evaluation

Proses evaluasi nilai yang dilakukan sedini mungkin pada variable.

F

Function Types

Tipe dari sebuah *function* yang mencerminkan *data output* keluaran dari function tersebut.

Flat Case

Sebuah aturan format penulisan kode yang dalam penulisannya menggunakan huruf kecil dan tanpa menggunakan penghubung seperti *underscore*.

Filter

Proses penyaringan terhadap sebuah kumpulan objek.

Function Header

Bagian yang merupakan konstruksi dari sebuah fungsi untuk menentukan perilakunya akan seperti apa.

Function Body

Bagian dari sebuah fungsi yang terdapat kumpulan *statement* yang menjalankan aksi tertentu.

Functional Programming

Salah satu paradigma dalam pemrograman, di mana struktur dalam kode program disusun murni hanya dari function yang mendeskripsikan apa yang ingin dilakukan.

Function Literal

Istilah lain dari Anonymous Function.

Function Inside Function (Nested Function)

Istilah yang digunakan untuk *function* yang dideklarasikan di dalam *function*.

Framework

Kerangka kerja yang sudah disediakan untuk mengembangkan aplikasi. Kerangka kerja di sini sangat membantu developer dalam menuliskan sebuah kode dengan lebih terstruktur dan tersusun rapi.

G

Getter

Fungsi yang digunakan untuk membaca nilai sebuah variabel yang terenkapsulasi.

Generic

Menggambarkan karakteristik (gambaran umum) dari sebuah jenis atau kelompok tetapi tidak spesifik.

H

Higher-order Function

Fungsi yang menerima fungsi lain sebagai parameternya, atau mengembalikan fungsi lain sebagai nilai kembaliannya.

I

Implicit/Implisit

Termasuk (terkandung) di dalamnya (meskipun tidak dinyatakan secara jelas atau terang-terangan).

Instance

Perwujudan dari sebuah *class* (kelas).

Inisialisasi

Sebuah proses pemberian nilai awal setelah deklarasi sebuah variabel.

Infinite Loop

Keadaan di mana proses iterasi terus berjalan tanpa henti.

Indeks

Indikator dalam bentuk angka yang merepresentasikan posisi dari sebuah item di dalam sebuah list atau daftar (daftar karakter, array, dsb).

Identifier

Sebuah nama yang diberikan ke sebuah elemen/variable di dalam sebuah program untuk menandakannya.

Input dan Output (I/O)

Proses penulisan dan pembacaan data.

Inter Process Communication (IPC)

Mekanisme pertukaran data antara satu proses dengan proses lainnya, baik itu proses yang berada di dalam komputer yang sama, atau komputer jarak jauh yang terhubung melalui jaringan.

Intersect

Fungsi yang dapat digunakan untuk menampilkan irisan dari dua Sets berbeda.

Immutable

Objek yang nilainya tidak bisa diubah saat *runtime*.

Inner Function

Istilah lain dari Function Inside Function.

Inner Class

Kelas yang dideklarasikan di dalam sebuah kelas. Biasa disebut dengan Nested Class. Fungsinya yaitu untuk mendukung proses yang ada di kelas utamanya. Dengan menambahkan keyword ini, class tersebut dapat mengakses member dari kelas yang ada di luarnya.

Interoperabilitas

Proses di mana suatu aplikasi bisa berinteraksi dengan aplikasi lainnya melalui suatu protokol yang disetujui bersama lewat bermacam-macam jalur komunikasi.

J

JVM

Aplikasi khusus yang harus diinstall agar sebuah komputer bisa menjalankan program yang ditulis dalam bahasa Java.

JSON (JavaScript Object Notation)

Format data yang menyajikan pasangan *key* dan *value* yang digunakan dalam pertukaran data antar program aplikasi dan juga bisa digunakan sebagai tempat penyimpanan dari suatu data.

K

Konsol

Tool yang dapat digunakan untuk menjalankan perintah atau melihat pesan *log*.

Keyword

Dalam pemrograman, *keyword* adalah susunan huruf atau frasa yang sudah disediakan oleh bahasa program tertentu dan mempunyai arti spesial pada *compiler*.

Kompiler

Perangkat lunak yang digunakan untuk menerjemahkan bahasa pemrograman menjadi program *system*.

Kotlin Standard Library

API atau fitur yang secara default sudah disediakan oleh Kotlin.

L

Lazy Evaluation

Proses evaluasi pemberian nilai saat benar-benar dibutuhkan.

Library Function

Merupakan sekumpulan fungsi bawaan **Kotlin Standart Library** yang dapat digunakan untuk menerapkan suatu komponen dengan cara yang sederhana dengan memanfaatkan Lambda Expression.

Life Cycle

Alur hidup dari sebuah perangkat lunak.

M

Method

Bagian dari *class* yang menjalankan suatu aksi tertentu.

Member Function

Fungsi yang terdapat pada sebuah *class*.

Modifier

Merupakan bagian spesifik dari sebuah bahasa pemrograman yang ditujukan untuk mengatur bagaimana hak akses dari sebuah kelas, fungsi, properti dan variabel. Bisa juga disebut sebagai *access modifier*.

Memory Address

Bagian yang digunakan untuk menampung alamat data atau instruksi pada memori utama yang akan diambil atau yang akan diletakkan.

Mutable

Tipe variable yang bisa diubah nilainya.

Mapping

Proses perubahan dari satu jenis model data ke jenis model lainnya.

N

Native

Asli atau murni.

Named Argument

Fitur yang digunakan untuk melampirkan argumen dengan memanfaatkan nama parameter.

Nullable

Kondisi dari sebuah variabel yang dapat menampung nilai *null*.

Null

Tidak memiliki nilai/value.

NullPointerException

Kesalahan yang muncul saat program mencoba menggunakan referensi objek yang memiliki nilai null.

O

Objek / Object

Merupakan instance atau wujud nyata dari suatu kelas.

Ordinal

Properti yang digunakan untuk mendapatkan posisi dari konstanta yang berada di dalam kelas enum.

Open

Merupakan keyword pada Class/Method yang dapat di-extend dan di-override oleh turunannya. Merupakan kebalikan dari Final.

Operator

Fungsi tertentu yang digunakan untuk menjalankan sebuah proses untuk menentukan hasil.

Object-Oriented Programming

Sebuah paradigma dalam menyusun program di mana menggunakan pendekatan objek di dunia nyata. Setiap objek memiliki data dan prosedur yang bisa dilakukan.

Override

Kata kunci yang digunakan untuk mengambil alih fungsi yang diwariskan.

P

Proses Iterasi

sifat tertentu dari algoritma atau program komputer dimana suatu urutan atau lebih dari langkah algoritmik dilakukan di *loop* program.

Package

Cara yang digunakan untuk mengelompokkan kumpulan kelas.

Parameter

Nilai inputan berupa variabel pada saat fungsi itu didefinisikan.

Parenthesis

Tanda kurung yang berada pada sebuah *function* atau *constructor*.

Properties

Atribut yang menjadi anggota dari suatu kelas. Properti digunakan untuk menyimpan data yang relevan dengan kelas. Seperti yang sudah disebutkan sebelumnya seperti tinggi, berat, dan umur adalah properti dari suatu kelas.

Plugins

Kode dengan fungsi tertentu yang memungkinkan aplikasi atau program untuk menjalankan fitur tambahan di aplikasi atau program tersebut.

R

Reference

Suatu value yang mempunyai akses secara tidak langsung ke data atau objek lain.

Range

Merupakan salah satu tipe data yang unik dimana kita dapat menentukan nilai awal dan nilai akhir.

Return Type

Tipe kembalian dari sebuah function.

Real-time

Kondisi pengoperasian dari suatu sistem perangkat keras dan perangkat lunak yang dibatasi oleh rentang waktu dan memiliki tenggat waktu (deadline) yang jelas, relatif terhadap waktu suatu peristiwa.

Reactive

Model algoritma pemrograman yang merepresentasikan data sebagai aliran yang dapat diproses.

Read-Only

Adalah kondisi di mana kita hanya dapat menjalankan proses membaca namun tidak dapat menjalankan proses menulis.

Receiver Type

Teknik dalam deklarasi suatu fungsi di mana fungsi tersebut akan digunakan hanya pada tipe data/objek tertentu, sesuai dengan tipe data/objek yang dideklarasikan saat mendefinisikan fungsi tersebut.

Recursive Function

Fungsi yang memanggil dirinya sendiri.

S**Sealed Class**

Class yang merepresentasikan pembatasan hierarki pada Class. Mirip dengan Enum Class, bedanya Enum hanya memiliki satu objek per Class, sedangkan Sealed Class dapat memiliki beberapa objek dari class yang sama.

Setter

Fungsi yang digunakan untuk mengubah nilai sebuah variabel yang terenkapsulasi.

Synthetic method

Method yang dihasilkan oleh compiler dan tidak terlihat di dalam kode sumber.

Statically Typed

Semua variabel mempunyai tipe yang tetap dan harus dideklarasikan sejak awal.

Statement

Sebuah instruksi dalam bentuk kode yang digunakan oleh komputer untuk menjalankan aksi tertentu.

Suspend Function

Fungsi regular yang mempunyai modifier suspend pada definisi fungsinya, yang mengindikasikan bahwa fungsi tersebut bisa memblokir eksekusi dari coroutine.

Single Value

Nilai tunggal atau suatu nilai yang tidak dapat dibagi-bagi

Server

Merupakan jaringan komputer yang berperan penting dalam menjalankan koneksi internet.

Server-side

Suatu program yang berjalan di server yang menangani pembuatan konten halaman website. Server-side merupakan sistem yang berjalan di server.

T

Type Inference

Kemampuan dari suatu *compiler* untuk mendeteksi tipe dari sebuah variabel secara otomatis.

Thread

Sekumpulan instruksi yang dapat dijalankan pada CPU.

Tail recursion

Sebuah teknik dalam pemanggilan rekursif di mana pemanggilan tersebut (recursive call) diletakkan pada akhir dari sebuah *function*.

U

Union

Sekumpulan data yang anggotanya tersimpan di dalam memori yang sama dan saling mempengaruhi satu sama lain.

V

Variable

Dalam konteks pemrograman, variable adalah tempat yang digunakan untuk menyimpan suatu data.

Visibility Modifier

Keyword yang mengatur aksesibilitas dari suatu *class*, objek, *interface*, fungsi, properti. Contohnya *public*, *private*, *protected*, dan *internal*.