

ACTIVITY PERTEMUAN 5

NAMA : INDAH DWI APRILIANI

NPM : 50421651

KELAS : 4IA14

MATERI :

MATA PRAKTIKUM : REKAYASA PERANGKAT LUNAK 2

1. Dependency Injection (DI) adalah konsep penting dalam Spring Framework. Jelaskan apa itu Dependency Injection dan mengapa DI sangat penting dalam pengembangan aplikasi berbasis Spring Boot.

Jawab :

Dependency Injection (DI) adalah sebuah konsep desain perangkat lunak yang bertujuan untuk mengurangi ketergantungan langsung antar komponen dalam aplikasi. Dengan menggunakan DI, sebuah objek tidak perlu membuat atau mengelola ketergantungannya sendiri, tetapi ketergantungan tersebut disuntikkan (injected) oleh pihak luar, seperti framework atau kontainer.

Pada Spring Framework, DI memungkinkan objek untuk mendapatkan ketergantungannya dari Spring IoC (Inversion of Control) container. Kontainer ini bertugas untuk mengelola siklus hidup objek dan menyuntikkan dependensi ke dalam objek yang memerlukannya.

- Cara Kerja Dependency Injection dalam Spring
Spring menyediakan dua cara untuk menyuntikkan dependensi :
 - a. Constructor Injection : Dependensi disuntikkan melalui konstruktor objek.
 - b. Setter Injection : Dependensi disuntikkan melalui setter method.
 - c. Field Injection : Dependensi disuntikkan langsung ke field objek.
- Mengapa Dependency Injection Penting dalam Pengembangan Aplikasi Spring Boot
 - a. Meningkatkan Loose Coupling : DI memungkinkan pengurangan ketergantungan antar komponen. Komponen-

komponen dalam aplikasi menjadi lebih mandiri dan fleksibel karena tidak saling bergantung langsung. Ini memudahkan pengujian dan pengelolaan kode.

- b. Mempermudah Pengujian (Testing) : Dengan DI, kita bisa menyuntikkan mock atau objek tiruan (mock objects) pada saat pengujian unit. Hal ini memudahkan pengujian komponen secara terisolasi tanpa harus menginisialisasi dependensi secara manual.
- c. Meningkatkan Pemeliharaan Kode : Dengan DI, kita dapat lebih mudah mengubah implementasi dari sebuah dependensi tanpa harus mengubah kode yang menggunakannya. Ini membuat pemeliharaan kode lebih mudah dan fleksibel.
- d. Mendukung Inversi Kontrol (IoC) : DI adalah bagian dari prinsip Inversi Kontrol, di mana Spring mengontrol penciptaan dan manajemen objek-objek dalam aplikasi, sehingga aplikasi dapat lebih modular dan lebih terstruktur dengan baik.
- e. Konsistensi dan Reusabilitas : Dengan DI, kita dapat lebih konsisten dalam cara pengelolaan objek dan dependensinya. Selain itu, kita dapat menggunakan objek yang sama di berbagai bagian aplikasi, meningkatkan reusabilitas.