TUGAS 1 AAE

Spring Framework Annotations

1. @Required

   Anotasi ini digunakan pada bean setter methods. @Required menunjukan bahwa *bean* yang terpengaruh harus diisi pada waktu konfigurasi dengan properti yang diperlukan.

2. @Autowired

   Anotasi ini digunakan pada fields, setter methods, dan constructor. @Autowired berfungsi untuk mereferensikan suatu bean.

```
@Component
public class Customer {
    private Person person;
    @Autowired
    public Customer (Person person) {
      this.person=person;
    }
}
```

3. @Qualifier

   Anotasi ini digunakan bersama @Autowired untuk mengontrol dependency injection process.

```
@Autowired
@Qualifier(&amp;quot;customerDao&amp;quot;)
public void setCustomerDao(CustomerDao customerDao) {
this.customerDao = customerDao;
}
```

4. @Configuration

   Anotasi ini digunakan pada kelas yang mendefinisikan beans. Java class yang memiliki anotasi @Configuration akan memiliki method untuk instansiasi dan mengkonfigurasi dependensi.

```
@Configuration
public class DataConfig{
  @Bean
  public DataSource source(){
    DataSource source = new OracleDataSource();
    source.setURL();
    source.setUser();
    return source;
  }
  @Bean
  public PlatformTransactionManager manager(){
    PlatformTransactionManager manager = new BasicDataSourceTransactionManager();
    manager.setDataSource(source());
    return manager;
  }
}
```

5. @ComponentScan

Anotasi ini digunakan untuk menentukan base package menggunakan atribut basePackageClasses atau basePackage untuk dipindai.

```java
package guru.springframework.blog.componentscan.example.demopackageA;

import org.springframework.stereotype.Component;

@Component("demoBeanA")
public class DemoBeanA {
}
```

6. @Bean

Anotasi ini digunakan pada level method. Anotasi ini bekerja dengan @Configuration untuk membuat Spring Beans. Method yang memiliki anotasi @Bean bekerja sebagai Bean ID dan akan mengembalikkan nilai Bean yang asli.

```java
@Configuration
public class AppConfig{
  @Bean
  public Person person(){
    return new Person(address());
  }
  @Bean
  public Address address(){
    return new Address();
  }
}
```

7. @Component

Anotasi ini adalah anotasi umum yang digunakan untuk memanagement komponen – komponen spring. Penggunaannya sebenarnya hampir sama dengan @Repository, @Service dan @Controller tapi dengan fungsi yang lebih umum, artinya ketika membuat sebuah class yang bukan ditujukan sebagai controller ataupun service maka @Component lebih pas untuk digunakan.

```java
@Component
public class COnnectionUtil {

public void getConnction(){
System.out.println(&amp;quot;Here global method used to make a connection to
}
}
```

8. @Controller

Anotasi ini adalah anotasi yang digunakan untuk menunjukkan bahwa sebuah class adalah controller, yang terdiri dari method – method atau fungsi – fungsi yang akan melayani dan berhubungan langsung dengan http request dan request mapping.

```
@Controller
public class JSONController {

@RequestMapping(value = &quot;/kfc/brands/{name}&quot;, method = RequestMeth
public @ResponseBody
Shop getShopInJSON(@PathVariable String name) {
contactsDAO.test();
validator.printTestDao();

Shop shop = new Shop();
shop.setName(name);
shop.setStaffName(new String[]{&quot;mkyong1&quot;, &quot;mkyong2&quot;});
return shop;
}
}
```

9. @Service

Anotasi ini digunakan untuk menunjukkan sebuah class yang akan digunakan sebagai pusat service dari aplikasi yang akan dibangun. Pusat service yang dimaksud adalah semua proses yang berhubungan dengan business logic akan dilakukan pada class ini.

```
@Service
public class CustomerServiceImpl  {
    private CustomerDao customerDao;

    @Override
    @Transactional
    public Customer getCustomerById(String id) {
    Customer customer = null;
    try {
        customer = customerDao.getCustomerById(id);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return customer;
    }

    @Autowired
    @Qualifier(&quot;customerDao&quot;)
    public void setCustomerDao(CustomerDao customerDao) {
    this.customerDao = customerDao;
    }
}
```

10. @Repository

Repository adalah anotasi yang digunakan untuk menunjukkan sebuah class akan dijadian DAO (Data Access Object), pada class ini proses query logic seharusnya dibuat.

```
@Repository
public class Pegawai {

public void InsertData(){
System.out.println(&amp;quot;Here Create your Query Logic&amp;quot;);
}
}
```