

Khrisna Widhi D

155150201111275

## 1. @Controller

Controller 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. Pada Controller pertama kali data data yang dikirim dari form atau browser untuk diolah dan dikirim kelevel berikutnya pada spring MVC.

@Controller

```
01  @Controller
02  public class JSONController {
03
04  @RequestMapping(value = "/kfc/brands/{name}", method = RequestMethod
05  public @ResponseBody
06  Shop getShopInJSON(@PathVariable String name) {
07  contactsDAO.test();
08  validator.printTestDao();
09
10  Shop shop = new Shop();
11  shop.setName(name);
12  shop.setStaffName(new String[]{"mkyong1", "mkyong2"});
13  return shop;
14  }
15  }
```

## 2. @Services

Annotasi @service 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 bukan pada controller ataupun DAO.

```
01  @Service
02  public class CustomerServiceImpl {
03      private CustomerDao customerDao;
04
05      @Override
06      @Transactional
07      public Customer getCustomerById(String id) {
08          Customer customer = null;
09          try {
10              customer = customerDao.getCustomerById(id);
11          } catch (Exception e) {
12              e.printStackTrace();
13          }
14          return customer;
15      }
16
17      @Autowired
18      @Qualifier("customerDao")
19      public void setCustomerDao(CustomerDao customerDao) {
20          this.customerDao = customerDao;
21      }
22  }
```

### 3. @Repository

Repository adalah anotasi yang digunakan untuk menunjukkan sebuah class akan dijadikan DAO, pada class ini proses query logic seharusnya dibuat.

```
1 | @Repository
2 | public class Pegawai {
3 |
4 | public void InsertData(){
5 |     System.out.println(""Here Create your Query Logic"");
6 | }
7 | }
```

### 4. @Autowire

Sebenarnya ini agak sudah kalau diartikan dalam bahasa indonesia, tapi biasanya anotasi ini saya gunakan untuk proses dependency injection, untuk menyederhanakan pengguna setter method untuk memanggil sebuah class pada dependency injection.

```
1 | @Autowired
2 | private Pegawai contactsDAO;
3 |
4 | @Autowired
5 | private TestDao validator;
```

### 5. @Qualifier

Adalah anotasi yang digunakan bersama dengan AutorWire yang digunakan sebagai discriminator.

```
1 | @Autowired
2 | @Qualifier(""customerDao"")
3 | public void setCustomerDao(CustomerDao customerDao) {
4 |     this.customerDao = customerDao;
5 | }
```

### 6. @Component

Adalah Annotation umum yang digunakan untuk manajemen component2 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.

```
1  @Component
2  public class COnnectionUtil {
3
4  public void getConnction(){
5  System.out.println(""Here global method used to make a connection to
6  }
7  }
```

---