**Struktur Project Template Baru**

* Untuk sampai dapat menampilkan sebuah halaman, stack yang dipakai hampir sama dengan template yang lama.
* Stack Controller-Services-Mapper Java-Mapper XML masih dipakai pada project template yang baru ini.
* Yang perlu diperhatikan diantaranya:
  + View jsp untuk setiap modul totalnya 2 (dua) file
    - File 1 untuk halaman list modul tersebut.
    - File 2 dipakai bersamaan pada method view detail, create/request, edit, dan delete/terminate.

***\*contoh:*** *user.jsp (file 1) dan user-action.jsp (file 2)*

* + Pada method yang difungsikan untuk menghandle request, tidak lagi memakai anotasi @RequestMapping, akan tetapi menyesuaikan jenis method formnya, jika method GET maka menggunakan anotasi @GetMapping, jika method POST maka menggunakan anotasi @PostMapping disertai url pathnya.

***\*contoh:*** *@GetMapping(“{id}/detail”) atau @PostMapping(“edit”)*

* + Pada method yang membutuhkan otorisasi, ditambahkan anotasi @PreAuthorize. Singkatnya anotasi ini akan menggantikan logika if-else yang biasa dipakai developer untuk melakukan cross check terhadap logged user roles apakah memiliki otoritas pada sebuah method.

***\*contoh proses otorisasi lama (logika checking allowed action di setiap Controller):***

|  |
| --- |
| if(!isActionAllowed((String) model.get("roleRequest"))) {  if(getLOGIN().get("roles").equals("NO\_ROLE")) {  model.clear();  return "redirect:/login";  } else {  model.clear();  return "redirect:/403";  }  } |

***\*Note:*** *cara ini kurang baik untuk dilakukan, karena sebisa mungkin hindari menempatkan blok logika pada bagian Controller suatu modul, usahakan semua logika ada di bagian Services.*

***\*contoh proses otorisasi baru (penerapan Method Level Authorization):***

@PreAuthorize(“has Authority(‘CREATE USER’)”)

* + Pada bagian Services dibagi menjadi 2 file (interface dan implementation)
  + Pada bagian interface service akan mengextend ke class interface BaseModelService, disini berisi default method yang biasa digunakan berulang-ulang seperti findAll(), findById(), insert(), update(), dan disable() atau delete().
  + Maka class service implementation secara default akan melakukan auto generate method-method default yang ada pada class BaseModelService.
  + Pada bagian interface Mapper akan meng-extend ke interface BaseModelMapper, dimana di dalam class interface tersebut sudah ada method-method default yang nantinya akan digunakan di class service implementation.
  + Pada bagian mapper xml pendeklarasian resultMap sama seperti project template yang lama dan ada file base core dimana di dalamnya terdapat query sql created\_by, created\_date, modified\_by, modified\_date, dan is\_enabled yang bisa di include ke beberapa query select yang memerlukan informasi tersebut.
  + Pada package mapper xml menggunakan bantuan library Liquibase untuk inisiasi pembuatan seluruh data tabel, ada pada package src/main/resources/db, …/db.public, dan …/db.csv. (penjelasan mengenai Liquibase akan dijelaskan lebih lanjut pada sesi Liquibase)

***-end-***