TUGAS INDIVIDU 1 ARSITEKTUR DAN PEMROGRAMAN APLIKASI ENTERPRISE



Disusun oleh:

ANDESTA PUTRA NPM. 1706106633

PROGRAM SARJANA 1 EKSTENSI SISTEM INFORMASI FAKULTAS ILMU KOMPUTER UNIVERSITAS INDONESIA 2018

1. Menampilkan Data Mahasiswa

Untuk membuat data mahasiswa kita perlu membuat sebuah method pada studentcontroller dengan path ("/"). Berikut ini merupakan method tersebut dalam kasus ini nama method tersebut adalah index.

```
@RequestMapping("/")
public String index (Model model)
{
    model.addAttribute ("title", "Home");
    return "index";
}
```

Pada script diatas kita akan membuat model atribut title dengan tulisan home yang nantinya akan digunakan sebagai title pada view index. Untuk view index sendiri akan kita buat form dengan method get yang berisi inputan npm dari mahasiswa dan action dari form tersebut akan menuju ke method viewMahasiswa dengan path ("/mahasiswa"). Isi dari method tersebut adalah sebagai berikut.

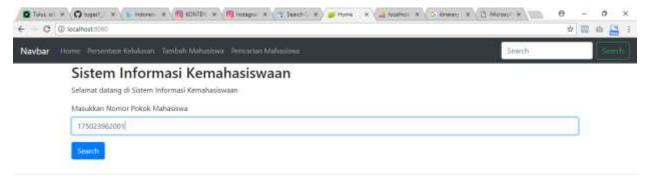
```
@RequestMapping(value = "/mahasiswa", method = RequestMethod.GET)
public String viewMahasiswa (Model model, @RequestParam(value = "npm", required =
false) String npm)
{
    StudentModel mahasiswa = studentDAO.selectMahasiswa (npm);

    if (mahasiswa != null) {
        model.addAttribute ("title", "Tampil Mahasiswa");
        model.addAttribute ("mahasiswa", mahasiswa);
        return "view";
    } else {
        model.addAttribute ("title", "Gagal Tampil Mahasiswa");
        model.addAttribute ("npm", npm);
        return "not-found";
    }
}
```

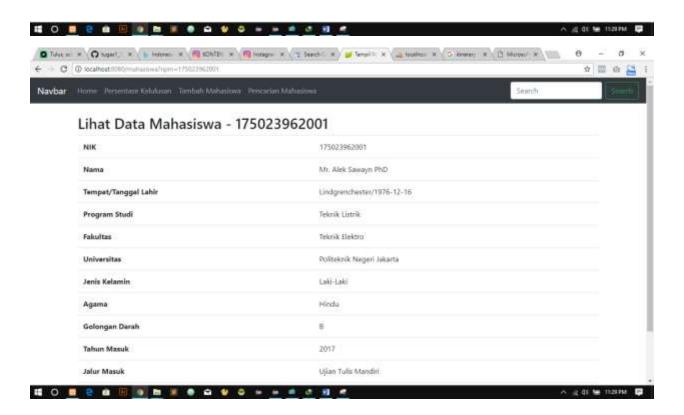
Pada method tersebut parameter npm akan digunakan sebagai parameter dari function selectMahasiswa. Didalam function tersebut juga terdapat studentMapper selectMahasiswa dengan script sebagai berikut.

```
@Select("select m.*, p.nama_prodi, f.nama_fakultas, u.nama_univ from mahasiswa m,
program_studi p, fakultas f, universitas u where m.id_prodi = p.id AND p.id_fakultas
= f.id AND f.id_univ = u.id AND npm = #{npm}")
    StudentModel selectMahasiswa (@Param("npm") String npm);
```

Pada mapper selectMahasiswa kita akan memilih mahasiswa berdasarkan npm yang telah diinputkan beserta dengan keterangan tambahan lainnya yaitu nama prodi, fakultas dan universitas. Untuk tampilan dari program adalah sebagai berikut.



Mata Kuliah APAP



2. Menambahkan Data Mahasiswa

Untuk menambahkan data mahasiswa kita perlu membuat method tambahMahasiswa pada path("/mahasiswa/tambah"). Isi dari method terebut adalah untuk menampilkan view form tambah mahasiswa. Berikut merupakan script dari method tersebut.

```
@RequestMapping(value = "/mahasiswa/tambah")
public String tambahMahasiswa (Model model)
{
        List<ProdiModel> prodi = prodiDAO.selectAllProdi();
        model.addAttribute ("prodi", prodi);
        model.addAttribute ("title", "Tambah Mahasiswa");
        return "form-add";
}
```

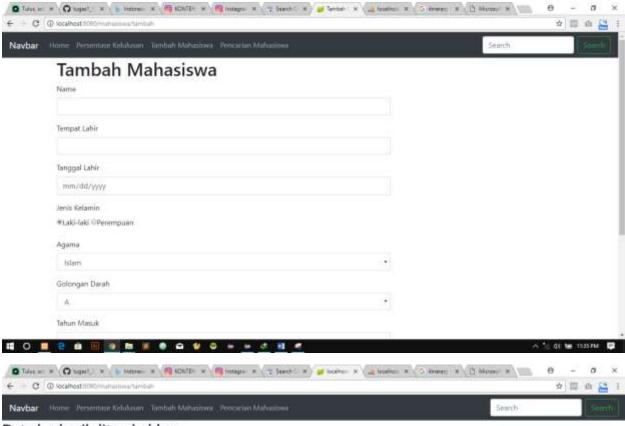
Kita juga memanggil list prodi yang nantinya akan digunakan untuk menampikan dropdown prodi.

Selanjutnya ketika form dengan method post yang telah kita buat disubmit maka akan mengarah ke path yang sama yaitu ("/mahasiswa/tambah") dengan method tambahMahasiswa. Berikut ini merupakan script dari method tersebut.

```
@RequestMapping(value = "/mahasiswa/tambah", method = RequestMethod.POST)
    public String tambahMahasiswa (Model model, @ModelAttribute StudentModel
mahasiswa)
    {
      String arrayJalur_masuk[], npm, npm2;
      int npm int;
      StudentModel kodeUnik = studentDAO.selectKodeUnik (mahasiswa.getId prodi());
      arrayJalur masuk = mahasiswa.getJalur masuk().split(",");
mahasiswa.getTahun masuk().substring(2)+kodeUnik.getKode univ()+kodeUnik.getKode prod
i()+arrayJalur_masuk[0];
      npm2 =
mahasiswa.getTahun_masuk().substring(2)+kodeUnik.getKode_univ()+kodeUnik.getKode_prod
i();
      StudentModel mahasiswa2 = studentDAO.cekMahasiswa (npm);
      if(mahasiswa2 != null){
             npm = mahasiswa2.getNpm().substring(7);
             npm int = Integer.parseInt(npm)+1;
             npm = npm2+npm_int;
      }else{
             npm = npm + "001";
      }
      mahasiswa.setJalur masuk(arrayJalur masuk[1]);
      mahasiswa.setNpm(npm);
      mahasiswa.setStatus("Aktif");
        studentDAO.tambahMahasiswa (mahasiswa);
        return "success-add";
```

}

Berikut ini merupakan tampilan dari tambah data mahasiswa.



Data berhasil ditambahkan

Mata Kuliah APAP



3. Merubah Data Mahasiswa

Mirip dengan menambah data mahasiswa, bedanya kita akan menggunakna path ("/mahasiswa/ubah/{npm}") dan method yang akan kita gunakan ialah method update mahasiswa sebagai berikut.

```
@RequestMapping("/mahasiswa/ubah/{npm}")
    public String updateMahasiswa (Model model, @PathVariable(value = "npm") String
npm)
    {
        StudentModel mahasiswa = studentDAO.selectMahasiswa (npm);
        if (mahasiswa != null) {
             List<ProdiModel> prodi = prodiDAO.selectAllProdi();
            model.addAttribute ("prodi", prodi);
            model.addAttribute ("title", "Update Mahasiswa");
            model.addAttribute ("mahasiswa", mahasiswa);
            return "form-update";
        } else {
            model.addAttribute ("title", "Update Mahasiswa");
            model.addAttribute ("npm", npm);
            return "not-found";
        }
    }
```

Pada method tersebut kita akan melakukan selectMahasiswa dengan npm sebagai parameternya, jika mahasiswa ditemukan maka akan dijadikan sebagai model atribut dimana datanya akan digunakan untuk menjadi value dari form-update yang telah dibuat. Setelah data dirubah maka akan menuju ke path yang sama yaitu ("/mahasiswa/ubah/{npm}") yang juga menggunakan method updateMahasiswa sebagai berikut.

```
@RequestMapping(value = "/mahasiswa/ubah/{npm}", method = RequestMethod.POST)
public String updateMahasiswa(Model model, @ModelAttribute StudentModel
mahasiswa, BindingResult bindingResult){

    if(bindingResult.hasErrors())
        return "errorForm";

    String arrayJalur_masuk[], npm, npm2;
    int npm_int;
    StudentModel kodeUnik = studentDAO.selectKodeUnik (mahasiswa.getId_prodi());

    arrayJalur_masuk = mahasiswa.getJalur_masuk().split(",");
    npm =
mahasiswa.getTahun_masuk().substring(2)+kodeUnik.getKode_univ()+kodeUnik.getKode_prodi()+arrayJalur_masuk[0];
    npm2 =
mahasiswa.getTahun_masuk().substring(2)+kodeUnik.getKode_univ()+kodeUnik.getKode_prodi();

    StudentModel mahasiswa2 = studentDAO.cekMahasiswa (npm);
```

Tampilan dari updateMahasiswa adalah sebagai berikut.

