

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.

Tugas ke-1 x Bugas x Interaksi x KONTEN x Instagram x Search x Home x localhost x library x Microsoft x

localhost:8080

Navbar Home Persentase Kelulusan Tambah Mahasiswa Pencarian Mahasiswa Search Search

Sistem Informasi Kemahasiswaan

Selamat datang di Sistem Informasi Kemahasiswaan

Masukkan Nomor Pokok Mahasiswa

Search

Mata Kuliah APAP

Tugas ke-1 x Bugas x Interaksi x KONTEN x Instagram x Search x Tampil x localhost x library x Microsoft x

localhost:8080/mahasiswa?npm=175023962001

Navbar Home Persentase Kelulusan Tambah Mahasiswa Pencarian Mahasiswa Search Search

Lihat Data Mahasiswa - 175023962001

NIK	175023962001
Nama	Mr. Alek Sawayn PhD
Tempat/Tanggal Lahir	Lindgrenchester/1976-12-16
Program Studi	Teknik Listrik
Fakultas	Teknik Elektro
Universitas	Politeknik Negeri Jakarta
Jenis Kelamin	Laki-Laki
Agama	Hindu
Golongan Darah	B
Tahun Masuk	2017
Jalur Masuk	Ujian Tulis Mandiri

2. Menambahkan Data Mahasiswa

Untuk menambahkan data mahasiswa kita perlu membuat method tambahMahasiswa pada path("/mahasiswa/tambah"). Isi dari method tersebut 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 menampilkan 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(",");
    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);

    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.

The screenshot displays a web application interface for adding a new student. The page title is "Tambah Mahasiswa". The form contains the following fields:

- Name:
- Tempat Lahir:
- Tanggal Lahir:
- Jenis Kelamin: ☐ Laki-laki ☐ Perempuan
- Agama:
- Golongan Darah:
- Tahun Masuk:

Below the form, a confirmation message is displayed: "Data berhasil ditambahkan". Underneath this message, the text "Mata Kuliah APAP" is visible. The browser's address bar shows the URL "localhost:8080/mahasiswa/tambah".

3. Merubah Data Mahasiswa

Mirip dengan menambah data mahasiswa, bedanya kita akan menggunakan 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);
```

```

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.updateMahasiswa (mahasiswa);

model.addAttribute ("title", "Update Mahasiswa");
return "success-update";
}

```

Tampilan dari updateMahasiswa adalah sebagai berikut.

The screenshot displays a web browser window with the URL `localhost:8080/mahasiswa/ubah/162911153001`. The page features a dark navigation bar with the title 'Tambah Mahasiswa' and a search bar. The main content area contains a form with the following fields:

- Name:** Abbie Bins
- Tempat Lahir:** West Rickleville
- Tanggal Lahir:** 03/07/1983
- Jenis Kelamin:** Laki-laki (selected)
- Agama:** Islam (selected)
- Golongan Darah:** A (selected)
- Tahun Masuk:** (empty)

A 'Low Disk Space' warning is visible in the bottom right corner, stating: 'We need more space to install important updates. Select Fix issues to get started.'

