

WRITE UP TUGAS 1 APAP

Oleh Muhammad Akbar Setiadi (1506737810)

FITUR 1

1. Pada package **com.akbar.tugas1.model**, saya membuat class **PendudukModel** dengan code di bawah ini

```
package com.akbar.tugas1.model;

import javax.validation.constraints.NotNull;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class PendudukModel {
    String id;
    @NotNull
    private String nik;
    @NotNull
    private String nama;
    @NotNull
    private String tempatLahir;
    @NotNull
    private String tglLahir;
    private String alamat;
    private String rt;
    private String rw;
    private String kelurahan;
    private String kecamatan;
    private String kota;
    @NotNull
    private String golonganDarah;
    @NotNull
    private String agama;
    @NotNull
    private String statusPerkawinan;
    @NotNull
    private String pekerjaan;
    @NotNull
    private String isWNI;
    @NotNull
    private String isWafat;
    @NotNull
    private String jenisKelamin;
    @NotNull
    private String idKeluarga;
    @NotNull
    private String statusDlmKeluarga;
```

```
}
```

Variabel yang memiliki anotasi `@NotNull` adalah variabel yang harus diisi pada fitur tambah penduduk. Semua variabel tersebut adalah semua atribut yang ada pada tabel penduduk ditambah dengan variabel yang diperlukan ketika melakukan query `SELECT` agar sesuai dengan fitur yang diinginkan.

2. Kemudian pada folder **com.akbar.tugas1.dao**, saya membuat interface **PendudukMapper**, yang berisi *converting* query `SELECT` ke class **PendudukModel** yang sudah dibuat di atas dengan method di bawah ini

```
PendudukMapper.java  PendudukModel.java
1 package com.akbar.tugas1.dao;
2
3* import java.util.List;
4
5 |
6 @Mapper
7 public interface PendudukMapper {
8     @Select("SELECT p.nik,p.nama, p.jenis_kelamin from penduduk p, keluarga k where p.id_keluarga = k.id and k.id_kelurahan = #{id}")
9     List<PendudukModel> selectPendudukByIdKelurahan (@Param("id") String idKelurahan);
10
11     @Select ("select p.nik,p.nama,p.tempat_lahir,p.tanggal_lahir,k.alamat,k.rt,k.rw,k.nama_kelurahan,
12             + "kc.nama_kecamatan, ko.nama_kota, p.golongan_darah, p.agama, p.status_perkawinan, p.pekerjaan, p.is_wni, p.is_wafat "
13             + "from penduduk p, keluarga k, kelurahan kl,kecamatan kc,kota ko where p.id_keluarga = k.id and "
14             + "k.id_kelurahan = kl.id and kl.id_kecamatan = kc.id and kc.id_kota = ko.id and p.nik = #{nik}")
15     @Results (value = {
16         @Result (property = "nik", column = "nik"),
17         @Result (property = "nama", column = "nama"),
18         @Result (property = "tempatLahir", column = "tempat_lahir"),
19         @Result (property = "tgllahir", column = "tanggal_lahir"),
20         @Result (property = "alamat", column = "alamat"),
21         @Result (property = "rt", column = "rt"),
22         @Result (property = "rw", column = "rw"),
23         @Result (property = "kelurahan", column = "nama_kelurahan"),
24         @Result (property = "kecamatan", column = "nama_kecamatan"),
25         @Result (property = "kota", column = "nama_kota"),
26         @Result (property = "golonganDarah", column = "golongan_darah"),
27         @Result (property = "agama", column = "agama"),
28         @Result (property = "statusPerkawinan", column = "status_perkawinan"),
29         @Result (property = "pekerjaan", column = "pekerjaan"),
30         @Result (property = "isWNI", column = "is_wni"),
31         @Result (property = "isWafat", column = "is_wafat")
32     })
33     PendudukModel selectPenduduk (@Param ("nik") String nik);
34 }
```

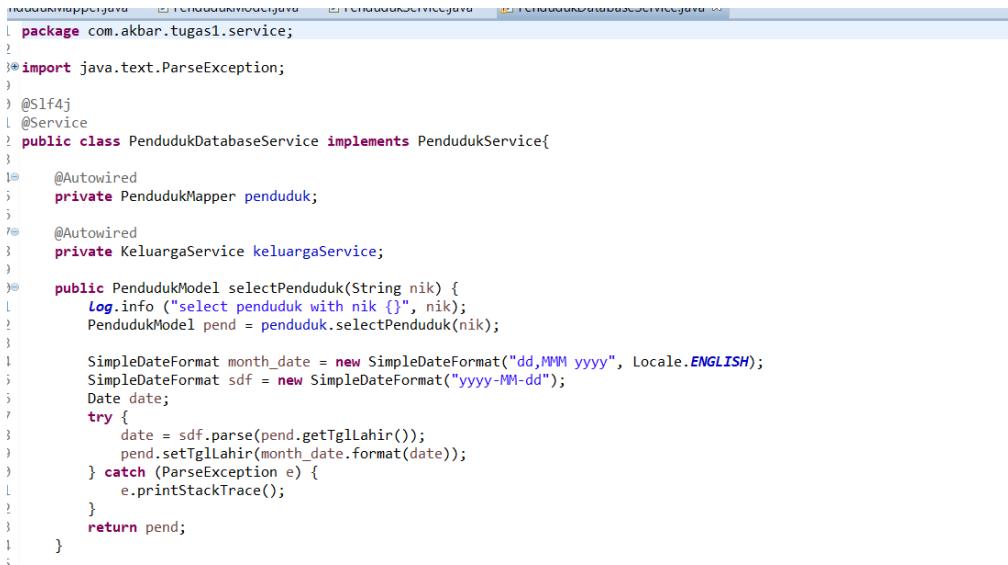
3. Pada package **com.akbar.tugas1.service**, saya menambahkan interface **PendudukService** dengan code di bawah ini.

```
package com.akbar.tugas1.service;

import java.util.List;
import com.akbar.tugas1.model.PendudukModel;

public interface PendudukService {
    PendudukModel selectPenduduk(String nik);
}
```

4. Pada package yang sama, dibuat class **PendudukDatabaseService** untuk melakukan implementasi method yang ada di dalam **PendudukService**. Di sini juga yang akan dilakukan implementasi *business logic* terhadap keseluruhan projek.



```
package com.akbar.tugas1.service;
import java.text.ParseException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class PendudukDatabaseService implements PendudukService{
    @Autowired
    private PendudukMapper penduduk;
    @Autowired
    private KeluargaService keluargaService;
    public PendudukModel selectPenduduk(String nik) {
        log.info ("select penduduk with nik {}", nik);
        PendudukModel pend = penduduk.selectPenduduk(nik);
        SimpleDateFormat month_date = new SimpleDateFormat("dd,MMM yyyy", Locale.ENGLISH);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        Date date;
        try {
            date = sdf.parse(pend.getTglLahir());
            pend.setTglLahir(month_date.format(date));
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return pend;
    }
}
```

Di dalam method tersebut, saya melakukan konversi tanggal lahir dari database (yyyy-mm-dd) menjadi konversi yang diinginkan (dd,mm yyyy) dalam bahasa Inggris.

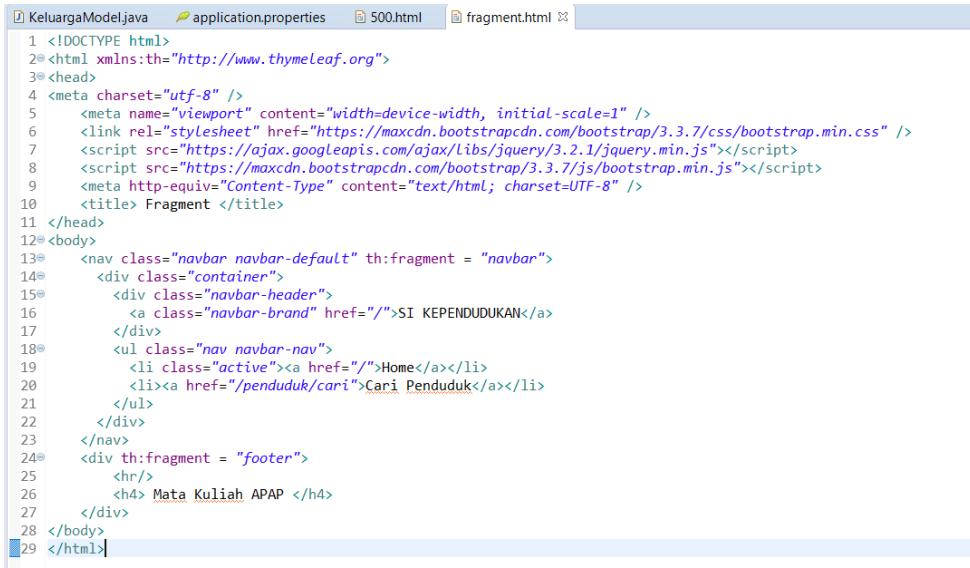
5. Kemudian pada package **com.akbar.tugas1.controller**, saya membuat class **PendudukController** dengan code di bawah ini



```
package com.akbar.tugas1.controller;
import org.springframework.beans.factory.annotation.Autowired;
@Controller
public class PendudukController {
    @Autowired
    PendudukService pendudukDAO;
    @Autowired
    KeluargaService keluargaDAO;
    @Autowired
    KelurahanService kelurahanDAO;
    @RequestMapping("/")
    public String home()
    {
        return "form-pendudukkeluarga";
    }
    @RequestMapping("/penduduk")
    public String findPenduduk(Model model,@ModelAttribute("sukses_ubah_wafat") String statusWafat,@RequestParam(value = "nik") String nik)
    {
        if(statusWafat.equals(""))
        {
            PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
            model.addAttribute("penduduk",penduduk);
            return "get-penduduk";
        } else
        {
            model.addAttribute("nik", nik);
            return "success-updatewafat";
        }
    }
}
```

Dapat dilihat di atas, terdapat tiga file html yang nantinya akan dibuat, yaitu **form-pendudukkeluarga** yang merupakan halaman utama dari website ini serta **get-penduduk** untuk mencari penduduk dengan nik yang diinginkan (form input pada form-pendudukkeluarga) dan juga **success-updatewafat** adalah halaman ketika tombol Update Kematian diklik (akan dijelaskan pada fitur 7).

6. Setelah itu, masuk ke dalam package **src/main/resources**, saya menambahkan folder **templates** dan menambahkan folder **fragments** di dalam folder templates dan membuat file **fragment.html** di dalam folder fragments



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="utf-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1" />
6   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
7   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
8   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
9   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
10  <title> Fragment </title>
11 </head>
12 <body>
13  <nav class="navbar navbar-default" th:fragment = "navbar">
14    <div class="container">
15      <div class="navbar-header">
16        <a class="navbar-brand" href="/">SI KEPENDUDUKAN</a>
17      </div>
18      <ul class="nav navbar-nav">
19        <li class="active"><a href="/">Home</a></li>
20        <li><a href="/penduduk/cari">Cari Penduduk</a></li>
21      </ul>
22    </div>
23  </nav>
24  <div th:fragment = "footer">
25    <hr/>
26    <h4> Mata Kuliah APAP </h4>
27  </div>
28 </body>
29 </html>
```

7. Setelah itu, di dalam folder **templates** saya menambahkan **form-pendudukkeluarga.html** di dalam folder templates.



```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<title>Home</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
  <nav th:replace = "fragments/navbar :: navbar"></nav>
  <div class="container">
    <h1>Sistem Informasi Kependudukan Online Provinsi DKI Jakarta</h1>
    <p>Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta</p>
    <h3>Lihat Data Penduduk Berdasarkan NIK</h3>
    <form action="/penduduk" method="get">
      <div>
        <label for="nik">Masukkan Nomor Induk Kependudukan</label><br /><input type="text" name="nik" />
      </div><br />
      <div>
        <button class="btn btn-info" type="submit">Lihat</button>
      </div>
    </form>
    <h3>Lihat Data Keluarga Berdasarkan NKK</h3>
    <form action="/keluarga" method="get">
      <div>
        <label for="nkk">Masukkan Nomor Kartu Keluarga</label><br /><input type="text" name="nkk" />
      </div><br />
      <div>
        <button type="submit" class="btn btn-info">Lihat</button>
      </div>
    </form>
  </div>
</body>
</html>
```

Kemudian lanjut di dalam folder templates, saya juga menambahkan file **get-penduduk.html** sebagai berikut

Muhammad Akbar Setiadi
1506737810
APAP B

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3     <head>
4         <meta name="viewport" content="width=device-width, initial-scale=1" />
5         <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
6         <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
7         <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
8         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
9         <title>View Penduduk by NIK</title>
10    </head>
11    <body>
12        <nav th:replace = "fragments/fragment :: navbar"></nav>
13        <div class="container">
14            <h1 class="page-header"><b>Lihat Data Penduduk : <span th:text="${penduduk.nik}"></span></b></h1>
15            <table class="table">
16                <tbody>
17                    <tr>
18                        <td>NIK</td>
19                        <td th:text="${penduduk.nik}"></td>
20                    </tr>
21                    <tr>
22                        <td>Nama</td>
23                        <td th:text="${penduduk.nama}"></td>
24                    </tr>
25                    <tr>
26                        <td>Alamat</td>
27                        <td th:text="${penduduk.alamat}"></td>
28                    </tr>
29                    <tr>
30                        <td>Tempat/Tanggal Lahir</td>
31                        <td th:text="${penduduk.tempatLahir} + ' ' + ${penduduk.tglLahir}"></td>
32                    </tr>
33                    <tr>
34                        <td>RT/Ruk</td>
35                        <td th:text="${penduduk.rt} + '/' + ${penduduk.rw}"></td>
36                    </tr>
37                    <tr>
38                        <td>Kelurahan/Desa</td>
39                        <td th:text="${penduduk.kelurahan}"></td>
40                    </tr>
41                    <tr>
42                        <td>Kecamatan</td>
43                        <td th:text="${penduduk.kecamatan}"></td>
44                    </tr>
45                    <tr>
46                        <td>Kota</td>
47                        <td th:text="${penduduk.kota}"></td>
48                    </tr>
49                    <tr>
50                        <td>Golongan Darah</td>
51                        <td th:text="${penduduk.golonganDarah}"></td>
52                    </tr>
53                    <tr>
54                        <td>Agama</td>
55                        <td th:text="${penduduk.agama}"></td>
56                    </tr>
57                    <tr>
58                        <td>Status Perkainanc</td>
59                        <td th:text="${penduduk.statusPerkawinan}"></td>
60                    </tr>
61                    <tr>
62                        <td>Pekerjaan</td>
63                        <td th:text="${penduduk.pekerjaan}"></td>
64                    </tr>
65                    <tr>
66                        <td>Kewarganegaraan</td>
67                        <td th:text="${penduduk.isWNI} == '1' ? 'WNA' : 'Wafat'"></td>
68                    </tr>
69                    <tr>
70                        <td>Status Kematian</td>
71                        <td th:text="${penduduk.isWafat} == '1' ? 'Wafat' : 'Hidup'"></td>
72                    </tr>
73                </tbody>
74            </table>
75            <form action="/penduduk/mati" method="post">
76                <button type="submit" class="btn btn-warning" th:field="${nik}" name="nik" th:value="${penduduk.nik}">UPDATE KEMATIAN</button>
77            </form>
78        </div>
79    </body>
80 </html>
```

Code :

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>View Penduduk by NIK</title>
    </head>
    <body>
        <nav th:replace = "fragments/fragment :: navbar"></nav>
        <div class="container">
            <h1 class="page-header"><b>Lihat Data Penduduk : <span
            th:text="${penduduk.nik}"></span></b></h1>
            <table class="table">
                <tbody>
                    <tr>
                        <td>NIK</td>
                        <td th:text="${penduduk.nik}"></td>
                    </tr>
                    <tr>
```

```
        <td>Nama</td>
        <td th:text="${penduduk.nama}"></td>
    </tr>
    <tr>
        <td>Alamat</td>
        <td th:text="${penduduk.alamat}"></td>
    </tr>
    <tr>
        <td>Tempat/Tanggal Lahir</td>
        <td th:text="${penduduk.tempatLahir} + ', ' +
${penduduk.tglLahir}"></td>
    </tr>
    <tr>
        <td>RT/RW</td>
        <td th:text="${penduduk.rt} + '/' + ${penduduk.rw}"></td>
    </tr>
    <tr>
        <td>Kelurahan/Desa</td>
        <td th:text="${penduduk.kelurahan}"></td>
    </tr>
    <tr>
        <td>Kecamatan</td>
        <td th:text="${penduduk.kecamatan}"></td>
    </tr>
    <tr>
        <td>Kota</td>
        <td th:text="${penduduk.kota}"></td>
    </tr>
    <tr>
        <td>Golongan Darah</td>
        <td th:text="${penduduk.golonganDarah}"></td>
    </tr>
    <tr>
        <td>Agama</td>
        <td th:text="${penduduk.agama}"></td>
    </tr>
    <tr>
        <td>Status Perkawinan</td>
        <td th:text="${penduduk.statusPerkawinan}"></td>
    </tr>
    <tr>
        <td>Pekerjaan</td>
        <td th:text="${penduduk.pekerjaan}"></td>
    </tr>
    <tr>
        <td>Kewarganegaraan</td>
        <td th:text="${penduduk.isWNI} == '1' ? 'WNI' : 'WNA'"></td>
    </tr>
    <tr>
        <td>Status Kematian</td>
        <td th:text="${penduduk.isWafat} == '1' ? 'Wafat' :
'Hidup'"></td>
    </tr>
</tbody>
</table>
<form action="/penduduk/mati" method="post">
    <button type="submit" class="btn btn-warning">
        UPDATE KEMATIAN
    </button>
</form>
</div>
</body>
</html>
```

Muhammad Akbar Setiadi
1506737810
APAP B

Di dalam file-file di atas, saya menggunakan framework bootstrap untuk memperbaik penampilan.

8. Menjalankan program di localhost:8080/



Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Lihat Data Penduduk Berdasarkan NIK

Masukkan Nomor Induk Kependudukan

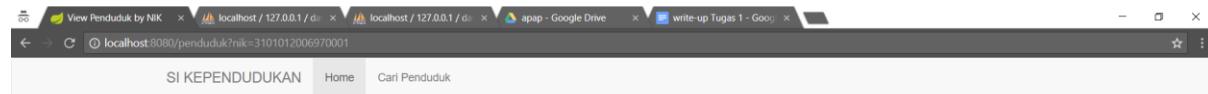
Lihat

Lihat Data Keluarga Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

Lihat

9. Mencoba memasukan nik pada input nik di atas, dan akan merefer ke halaman <http://localhost:8080/penduduk?nik=3101012006970001>



Lihat Data Penduduk : 3101012006970001

| | |
|----------------------|----------------------------|
| NIK | 3101012006970001 |
| Nama | akbar |
| Alamat | Jln. Cikapayang No. 594 |
| Tempat/Tanggal Lahir | pekalongan, 20 Jun 1997 |
| RT/RW | 149/090 |
| Kelurahan/Desa | PULAU PARI |
| Kecamatan | KEPULAUAN SERIBU SELATAN |
| Kota | KABUPATEN KEPULAUAN SERIBU |
| Golongan Darah | A- |
| Agama | islam |
| Status Perkawinan | Kawin |
| Pekerjaan | kar |
| Kewarganegaraan | WNI |
| Status Kematian | Hidup |

UPDATE KEMATIAN



FITUR 2

1. Di dalam package **com.akbar.tugas1.model**, saya menambahkan class **KeluargaModel** sebagai berikut

```
1 package com.akbar.tugas1.model;
2
3 import java.util.List;
4
5 @Data
6 @AllArgsConstructor
7 @NoArgsConstructor
8 public class KeluargaModel {
9     String id;
10    String nkk;
11    @NotNull
12    String alamat;
13    @NotNull
14    String rt;
15    @NotNull
16    String rw;
17    @NotNull
18    String idkelurahan;
19    String kelurahan;
20    String kecamatan;
21    String kota;
22    @NotNull
23    String isTdkBerlaku;
24    List<PendudukModel> anggotaKeluarga;
25
26 }
```

2. Di dalam package **com.akbar.tugas1.dao**, saya menambahkan interface **KeluargaMapper**.

```
1 package com.akbar.tugas1.dao;
2
3 import java.util.List;
4
5
6 @Mapper
7 public interface KeluargaMapper {
8     @Select ("select k.id,k.nomor_kk, k.alamat, k.rt,k.rw,kl.nama_kelurahan,kc.nama_kecamatan,ko.nama_kota "
9             + "from keluarga k, kelurahan kl, kecamatan kc, kota ko where k.id_kelurahan = kl.id and kl.id_kecamatan = kc.id "
10            + "and kc.id_kota = ko.id and k.nomor_kk = #{nkk} ")
11     @Results (value = 1)
12     @Result (property = "id", column = "id"),
13     @Result (property = "nkk", column = "nomor_kk"),
14     @Result (property = "alamat", column = "alamat"),
15     @Result (property = "rt", column = "rt"),
16     @Result (property = "rw", column = "rw"),
17     @Result (property = "kelurahan", column = "nama_kelurahan"),
18     @Result (property = "kecamatan", column = "nama_kecamatan"),
19     @Result (property = "kota", column = "nama_kota"),
20     @Result (property = "anggotaKeluarga", column = "id", javaType = List.class,
21             manyToMany (select = "selectPenduduk"))
22 }
23 KeluargaModel selectKeluarga(@Param ("nkk") String nkk);
24 }
```

Di code atas tersebut juga dilakukan select terhadap penduduk (list) dengan id keluarga sama dengan keluarga dengan parameter nkk

3. Di dalam package **com.akbar.tugas1.service**, saya menambahkan interface **KeluargaService**

```
1 package com.akbar.tugas1.service;
2
3 import com.akbar.tugas1.model.KeluargaModel;
4
5 public interface KeluargaService {
6
7     KeluargaModel selectKeluarga(String nomorKK);
8     void insertKeluarga(KeluargaModel keluarga);
9     void updateKeluarga(KeluargaModel keluarga, String nkk);
10    String getKodeKecamatan (String kecamatan);
11    String selectKelurahanByName (String kelurahan);
12    String getKodeKelurahan(String idKelurahan);
13    KeluargaModel selectKeluargaTable (String nomorKK);
14    void updateTidakBerlaku(String id);
15 }
16 }
```

4. Di dalam package yang sama, juga menambahkan class **KeluargaDatabaseService** dengan mengimplementasi di dalam **KeluargaService** dan **KeluargaMapper**

```
package com.akbar.tugas1.service;
import java.text.SimpleDateFormat;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
public class KeluargaDatabaseService implements KeluargaService {
    @Autowired
    private KeluargaMapper keluarga;
    @Autowired
    private PendudukService pendudukService;
    @Override
    public KeluargaModel selectKeluarga(String nomorKK) {
        log.info("select keluarga with nkk {}", nomorKK);
        KeluargaModel kel = keluarga.selectKeluarga(nomorKK);
        for(int i = 0; i < kel.getAnggotaKeluarga().size();i++) {
            SimpleDateFormat month_date = new SimpleDateFormat("dd,MMMMMM yyyy", Locale.ENGLISH);
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            Date date;
            try {
                date = sdf.parse(kel.getAnggotaKeluarga().get(i).getTglLahir());
                kel.getAnggotaKeluarga().get(i).setTglLahir((month_date.format(date)));
            } catch (ParseException e) {
                e.printStackTrace();
            }
            if(kel.getAnggotaKeluarga().get(i).getJenisKelamin().equals("1")) {
                kel.getAnggotaKeluarga().get(i).setJenisKelamin("Perempuan");
            } else {
                kel.getAnggotaKeluarga().get(i).setJenisKelamin("Laki-laki");
            }
            if(kel.getAnggotaKeluarga().get(i).getIsWNI().equals("1")) {
                kel.getAnggotaKeluarga().get(i).setIsWNI("WNI");
            } else {

```

Berikut di bawah adalah method untuk memilih keluarga dengan parameter nkk di database.

```
@Override
public KeluargaModel selectKeluarga(String nomorKK) {
    log.info("select keluarga with nkk {}", nomorKK);
    KeluargaModel kel = keluarga.selectKeluarga(nomorKK);
    for(int i = 0; i < kel.getAnggotaKeluarga().size();i++) {
        SimpleDateFormat month_date = new SimpleDateFormat("dd,MMMMMM yyyy", Locale.ENGLISH);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        Date date;
        try {
            date = sdf.parse(kel.getAnggotaKeluarga().get(i).getTglLahir());
            kel.getAnggotaKeluarga().get(i).setTglLahir((month_date.format(date)));
        } catch (ParseException e) {
            e.printStackTrace();
        }
        if(kel.getAnggotaKeluarga().get(i).getJenisKelamin().equals("1")) {
            kel.getAnggotaKeluarga().get(i).setJenisKelamin("Perempuan");
        } else {
            kel.getAnggotaKeluarga().get(i).setJenisKelamin("Laki-laki");
        }
        if(kel.getAnggotaKeluarga().get(i).getIsWNI().equals("1")) {
            kel.getAnggotaKeluarga().get(i).setIsWNI("WNI");
        } else {
            kel.getAnggotaKeluarga().get(i).setIsWNI("WNA");
        }
    }
    return kel;
}
}
```

Saya mengubah format tanggal lahir menjadi dd, mm yyyy seperti fitur 1. Kemudian juga mengubah format Jenis Kelamin dan Kewarganegaraan.

5. Lanjut di dalam controller, saya menambahkan method findKeluarga dengan implementasi code seperti di bawah ini

```
@RequestMapping("/keluarga")
public String findKeluarga(Model model, @RequestParam(value = "nkk") String nkk)
{
    KeluargaModel keluarga = keluargaDAO.selectKeluarga(nkk);
    model.addAttribute("keluarga", keluarga);
    return "get-keluarga";
}
```

6. Dan yang terakhir adalah membuat file **get-keluarga.html** di dalam folder **templates** di **src/main/resources** seperti di bawah ini

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
        <script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>View Keluarga by NKK</title>
</head>
<body>
<nav th:replace = "fragments/fragment :: navbar"></nav>
<div class="container">
    <h1 class="page-header"><b>Lihat Data Keluarga : <span>
th:text="${keluarga.nkk}"</span></b></h1>
    <table class="table">
        <tbody>
            <tr>
                <td>Nomor KK</td>
                <td th:text="${keluarga.nkk}"></td>
            </tr>
            <tr>
                <td>Alamat</td>
                <td th:text="${keluarga.alamat}"></td>
            </tr>
            <tr>
                <td>RT</td>
                <td th:text="${keluarga.rt}"></td>
            </tr>
            <tr>
                <td>RW</td>
                <td th:text="${keluarga.rw}"></td>
            </tr>
            <tr>
                <td>Kelurahan</td>
                <td th:text="${keluarga.kelurahan}"></td>
            </tr>
            <tr>
                <td>Kecamatan</td>
                <td th:text="${keluarga.kecamatan}"></td>
            </tr>
            <tr>
                <td>Kota</td>
                <td th:text="${keluarga.kota}"></td>
            </tr>
        </tbody>
    </table>
    <p class="lead">Anggota Keluarga</p>
    <table class="table">
        <thead class="thead-light">
            <tr>
                <th scope="col">No.</th>
                <th scope="col">Nama Lengkap</th>
                <th scope="col">NIK</th>
                <th scope="col">Jenis Kelamin</th>
                <th scope="col">Tempat Lahir</th>
                <th scope="col">Tanggal Lahir</th>
                <th scope="col">Agama</th>
                <th scope="col">Pekerjaan</th>
                <th scope="col">Status Perkawinan</th>
                <th scope="col">Status dalam Keluarga</th>
                <th scope="col">Kewarganegaraan</th>
            </tr>
        </thead>
        <tbody>
            <tr th:each="anggota,iterationStatus:
${keluarga.anggotaKeluarga}">
                <td th:text="${iterationStatus.index + 1}"></td>
                <td th:text="${anggota.nama}">Nama</td>
            </tr>
        </tbody>
    </table>
</div>

```

```
<td th:text="${anggota.nik}">NIK</td>
<td th:text="${anggota.jenisKelamin}">Jenis Kelamin</td>
<td th:text="${anggota.tempatLahir}">tempat lahir</td>
<td th:text="${anggota.tglLahir}">tgl lahir</td>
<td th:text="${anggota.agama}">agama</td>
<td th:text="${anggota.pekerjaan}">pekerjaan</td>
<td th:text="${anggota.pekerjaan}">Mark</td>
<td th:text="${anggota.statusDlmKeluarga}">Mark</td>
<td th:text="${anggota.isWNI}">Mark</td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>
```

7. Mencoba menjalankan kembali di localhost:8080 dan memasukan nkk di form di bawah ini

SI KEPENDUDUKAN Home Cari Penduduk

Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Lihat Data Penduduk Berdasarkan NIK

Masukkan Nomor Induk Kependudukan

Lihat

Lihat Data Keluarga Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

3101010101050001

Lihat

8. Klik tombol lihat, dan berikut hasilnya



SI KEPENDUDUKAN Home Cari Penduduk

Lihat Data Keluarga : 3101010101050001

| | |
|-----------|----------------------------|
| Nomor KK | 3101010101050001 |
| Alamat | Jln. Cikapayang No. 594 |
| RT | 149 |
| RW | 090 |
| Kelurahan | PULAU PARI |
| Kecamatan | KEPULAUAN SERIBU SELATAN |
| Kota | KABUPATEN KEPULAUAN SERIBU |

| Anggota Keluarga | | | | | | | | | | |
|------------------|-------------------------|------------------|---------------|--------------|------------------|-------|---------------------|---------------------|-----------------------|-----------------|
| No. | Nama Lengkap | NIK | Jenis Kelamin | Tempat Lahir | Tanggal Lahir | Agama | Pekerjaan | Status Perkawinan | Status dalam Keluarga | Kewarganegaraan |
| 1 | Taswir Rajata | 3101010303890001 | Laki-laki | Jakarta | 03.March 1989 | Islam | KARYAWAN HONORER | KARYAWAN HONORER | Kepala Keluarga | WNI |
| 2 | Mulyono Hutagalung S.Gz | 3101011408150001 | Laki-laki | Jakarta | 14.August 2015 | Islam | BELUM/TIDAK BEKERJA | BELUM/TIDAK BEKERJA | Anak | WNI |
| 3 | Yuliana Wahyuni | 3101016312160001 | Perempuan | Jakarta | 23.December 2016 | Islam | BELUM/TIDAK BEKERJA | BELUM/TIDAK BEKERJA | Anak | WNI |
| 4 | Ifa Novitasari | 3101014712830001 | Perempuan | Jakarta | 07.December 1983 | Islam | PETANI/PEKEBUN | PETANI/PEKEBUN | Istri | WNI |
| 5 | akbar | 3101012006970001 | Laki-laki | pekalongan | 20.June 1997 | islam | kar | kar | Anak | WNI |
| 6 | akbars | 3101012006970002 | Laki-laki | asda | 20.June 1997 | islam | isald | isald | Anak | WNI |

FITUR 3 : Menambahkan Penduduk Baru Sebagai Anggota Keluarga

- Di dalam interface **PendudukMapper**, dilakukan penambahan implementasi insert query dengan parameter semua atribut yang ada di dalam tabel penduduk dengan dibungkus di dalam class **PendudukModel**

```
i3
i4@Insert ("INSERT INTO `penduduk`(`nik`, `nama`, `tempat_lahir`, `tanggal_lahir`, `jenis_kelamin`, "
i5+ " `is_wni`, `id_keluarga`, `agama`, `pekerjaan`, `status_perkawinan`, `status_dalam_keluarga`, "
i6+ " `golongan_darah`, `is_wafat`) VALUES (#{nik},#{nama},#{tempatLahir},#{tgllahir},#{jeniskelamin},#{isWNI},"
i7+ " #{idKeluarga},#{agama},#{pekerjaan},#{statusPerkawinan},#{statusDlmKeluarga},#{golonganDarah},#{isWafat})")
i8 void insertPenduduk(PendudukModel penduduk);
.o
```

Kemudian juga dilakukan penambahan implementasi untuk mencari nik dengan kode kecamatan dan kode lahir yang sama serta untuk mencari kode kelurahan

```
@Select ("select distinct k1.kode_kelurahan from kelurahan k1, keluarga k where k1.id = k.id_kelurahan and k.id = #{id}")
String getKodeLurah(@Param("id") String id);

@Select ("select nik from penduduk where nik like #{nik} order by nik desc limit 1")
String nikLastSame(@Param("nik") String nik);
```

- Di dalam interface **PendudukService**, dilakukan penambahan method seperti di atas

```
PendudukController.java  PendudukMapper.java  PendudukModel.java  PendudukServ
1 package com.akbar.tugas1.service;
2
3 import java.util.List;
4
5 public interface PendudukService {
6     PendudukModel selectPenduduk(String nik);
7     void insertPenduduk(PendudukModel penduduk);
8     String nikLastSame(String nik);
9     String getKodeLurah(String id);
10    String getKodeLurah(String id);
11 }
```

- Di dalam class **PendudukDatabaseService**, dilakukan penambahan 3 method untuk implementasi method di **PendudukService**.

- Method mencari kode kelurahan berdasarkan parameter id_keluarga

```
public String getKodeLurah(String id) {
    return penduduk.getKodeLurah(id);
}
```

- Method mencari nik yang sama

```
    public String nikLastSame(String nik) {
        return penduduk.nikLastSame(nik);
    }
```

c. Method menambah penduduk

```
43@  @Override
44  public void insertPenduduk(PendudukModel pend) {
45      String kodeKecamatan = getKodeLurah(pend.getIdKeluarga()).substring(0,6);
46      String tglLahir = pend.getTglLahir();
47      String kodeLahir = null;
48      if (pend.getJenisKelamin().equalsIgnoreCase("1")) {
49          int tambah = Integer.parseInt(tglLahir.substring(8,10)) + 40;
50          kodeLahir = tambah + tglLahir.substring(5,7) + tglLahir.substring(2,4);
51      } else {
52          kodeLahir = tglLahir.substring(8,10) + tglLahir.substring(5,7) + tglLahir.substring(2,4);
53      }
54      String nikLast = nikLastSame(kodeKecamatan + kodeLahir + "%");
55      if(nikLast != null) {
56          nikLast = Long.toString(Long.parseLong(nikLast) + 1);
57      } else {
58          nikLast = kodeKecamatan + kodeLahir + "0001";
59      }
60      pend.setNik(nikLast);
61      penduduk.insertPenduduk(pend);
62      Log.info("insert penduduk with nik {}", pend.getNik());
63  }
```

Di method atas dilakukan generate NIK sesuai dengan soal, dengan mengambil terlebih dahulu kodeKecamatan didapat dari method getKodeLurah dengan substring sampai karakter ke 6. Terdapat dua *decision*, apakah penduduk tersebut laki atau perempuan. Jika perempuan, ditambah 40 pada kodeLahir.

Kemudian dilakukan cek apakah terdapat nik yang sama (sampai kode kecamatan + kode lahir). Jika iya, dilakukan penambahan 1 pada nik.

4. Lanjut ke dalam class **PendudukController.java**, menambah dua method dengan anotasi RequestMapping untuk masuk ke halaman form untuk menambah penduduk dan PostMapping untuk dilakukan penambahan penduduk dari method yang sudah dibuat di atas.

```
59
60@  @RequestMapping("/penduduk/tambah")
61  public String tambahPenduduk(Model model)
62  {
63      PendudukModel penduduk = new PendudukModel();
64      model.addAttribute("penduduk",penduduk);
65      return "form-addpenduduk";
66  }
67
68@  @PostMapping("/penduduk/tambah")
69  public String submitTambahPenduduk(Model model, @ModelAttribute PendudukModel penduduk)
70  {
71      pendudukDAO.insertPenduduk(penduduk);
72      model.addAttribute("penduduk",penduduk);
73      return "success-addpenduduk";
74  }
75
```

5. Setelah itu, dilakukan penambahan file **form-addpenduduk.html** yang berisi form atribut pada tabel penduduk dengan code di bawah

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">

<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Tambah Penduduk</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header"><b>Tambah Penduduk</b></h1>

        <form action="/penduduk/tambah" method="post" th:object="${penduduk}" >
            <div>
                <label for="nama">Nama</label><br/> <input type="text" name="nama" th:field="*{nama}" required="required"/>
            </div>
            <div>
                <label for="tempatLahir">Tempat Lahir</label><br/> <input type="text" name="tempatLahir" th:field="*{tempatLahir}" required="required" />
            </div>
            <div>
                <label for="tglLahir">Tanggal Lahir Format (YYYY-MM-DD)</label><br/> <input type="text" name="tglLahir" th:field="*{tglLahir}" required="required" />
            </div>
            <div>
                <label for="jenisKelamin">Jenis Kelamin</label><br/>
                <select name="jenisKelamin" th:field="*{jenisKelamin}">
                    <option value="0">Pria</option>
                    <option value="1">Wanita</option>
                </select>
            </div>
            <div>
                <label for="isWNI">Kewarganegaraan</label><br/>
                <select name="isWNI" th:field="*{isWNI}">
                    <option value="1">WNI</option>
                    <option value="0">WNA</option>
                </select>
            </div>
            <div>
                <label for="idKeluarga">ID Keluarga</label><br/> <input type="text" name="idKeluarga" th:field="*{idKeluarga}" required="required"/>
            </div>
            <div>
                <label for="agama">Agama</label><br/> <input type="text" name="agama" th:field="*{agama}" required="required" />
            </div>
            <div>
                <label for="pekerjaan">Pekerjaan</label><br/> <input type="text" name="pekerjaan" th:field="*{pekerjaan}" required="required" />
            </div>
            <div>
                <label for="statusPerkawinan">Status Perkawinan</label><br/>
                <select name="statusPerkawinan" th:field="*{statusPerkawinan}">
                    <option value="Kawin">Kawin</option>
                    <option value="Belum Kawin">Belum Kawin</option>
                </select>
            </div>
            <div>
                <label for="statusDlmKeluarga">Status Dalam Keluarga</label><br/>
                <select name="statusDlmKeluarga" th:field="*{statusDlmKeluarga}">
                    <option value="Anak">Anak</option>
                    <option value="Kepala Keluarga">Kepala Keluarga</option>
                    <option value="Istri">Istri</option>
                </select>
            </div>
        </form>
    </div>
</body>
```

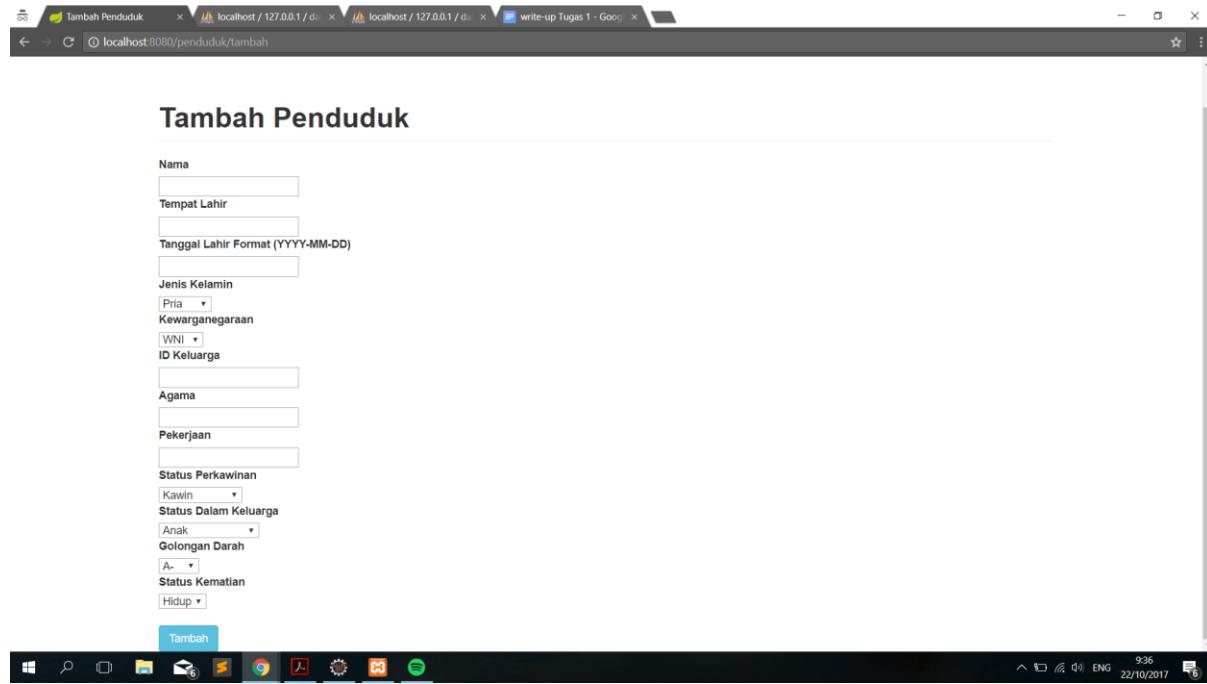
```
        <option value="Pembantu">Pembantu</option>
        <option value="Famili Lain">Famili Lain</option>
    </select>
</div>
<div>
    <label for="golonganDarah">Golongan Darah</label><br/>
    <select name="golonganDarah" th:field="*{golonganDarah}" >
        <option value="A- ">A-</option>
        <option value="A+ ">A+</option>
        <option value="B- ">B-</option>
        <option value="B+ ">B+</option>
        <option value="AB- ">AB-</option>
        <option value="AB+ ">AB+</option>
        <option value="O- ">O-</option>
        <option value="O+ ">O+</option>
    </select>
</div>
<div>
    <label for="isWafat">Status Kematian</label><br/>
    <select name="isWafat" th:field="*{isWafat}" >
        <option value="0">Hidup</option>
        <option value="1">Wafat</option>
    </select>
</div><br/>
<div>
    <button class="btn btn-info" type="submit" name="action" value="save">Tambah</button>
</div>
</form>
</div>
</body>
</html>
```

6. Dan juga menambah file **success-addpenduduk.html** jika sudah berhasil ditambahkan

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Tambah Penduduk</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Sukses</h1>
        <h3 th:text="'Penduduk dengan NIK ' + ${penduduk.nik} + ' berhasil ditambahkan'">Berhasil!</h3>
    </div>
</body>
</html>
```

7. Kemudian menjalankan website di localhost:8080/penduduk/tambah



Tambah Penduduk

Nama

Tempat Lahir

Tanggal Lahir Format (YYYY-MM-DD)

Jenis Kelamin

Kewarganegaraan

ID Keluarga

Agama

Pekerjaan

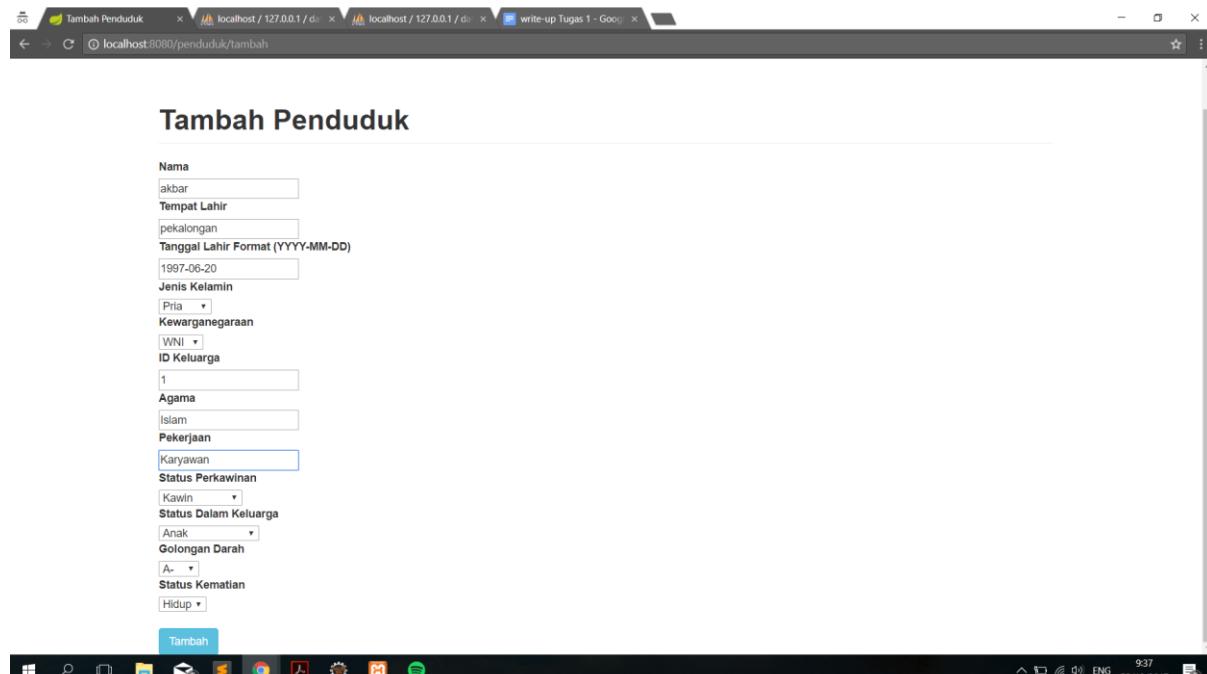
Status Perkawinan

Status Dalam Keluarga

Golongan Darah

Status Kematian

Mengisi semua form di atas



Tambah Penduduk

Nama

Tempat Lahir

Tanggal Lahir Format (YYYY-MM-DD)

Jenis Kelamin

Kewarganegaraan

ID Keluarga

Agama

Pekerjaan

Status Perkawinan

Status Dalam Keluarga

Golongan Darah

Status Kematian

Klik tombol Tambah dan masuk ke halaman sukses



Sukses

Penduduk dengan NIK 3101012006970003 berhasil ditambahkan



FITUR 4 : Menambahkan Keluarga Baru

1. Menambah method di dalam interface **KeluargaMapper** untuk implementasi dalam *INSERT* Query pada tabel keluarga, mencari kode kecamatan berdasarkan parameter nama kecamatan serta mencari nkk yang sama.

```
:8
:9@Insert ("INSERT INTO `keluarga`(`nomor_kk`, `alamat`, `RT`, `RW`, `id_kelurahan`, `is_tidak_berlaku`) VALUES "
:0
:1    + "({nkk},#{alamat},#{rt},#{rw},#{idKelurahan},#{isTdkBerlaku})"
:2
:3@Select ("SELECT kode_kecamatan FROM kecamatan WHERE nama_kecamatan = #{kecamatan}")
:4 String getCodeKecamatan(@Param ("kecamatan") String kecamatan);
:5
:6@Select ("SELECT nomor_kk from keluarga where nomor_kk like #{nkk} order by id desc limit 1")
:7 String nkkLast(@Param ("nkk") String nkk);
:8
```

2. Kemudian di dalam interface **KeluargaService**, juga dilakukan penambahan method yang sama seperti di atas

```
void insertKeluarga(KeluargaModel keluarga);
String getCodeKecamatan (String kecamatan);
String nkkLast(String nkk);
```

3. Di dalam class **KeluargaDatabaseService** dilakukan implementasi dari ketiga method tersebut

- a. Method getCodeKecamatan

Untuk mencari kode kecamatan berdasarkan namanya (parameter)

```
@Override
public String getCodeKecamatan(String kecamatan) {
    log.info("get kode kecamatan with nama kecamatan = {}", kecamatan);
    return keluarga.getCodeKecamatan(kecamatan);
}
```

- b. Method nkkLast

Untuk mencari nkk terakhir yang sama

```
    @Override
    public String nkkLast(String nkk) {
        // TODO Auto-generated method stub
        return keluarga.nkkLast(nkk);
    }
```

c. Method insertKeluarga

```
65
66@Override
67public void insertKeluarga(KeluargaModel kel) {
68    String kodeKecamatan = getKodeKecamatan(kel.getKecamatan()).substring(0,6);
69
70    Date date = Calendar.getInstance().getTime();
71    DateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
72    String today = formatter.format(date);
73    String kodeWaktu = today.substring(0,2) + today.substring(3,5) + today.substring(8,10);
74    String kode = nkkLast(kodeKecamatan + kodeWaktu + '%');
75    if (kode != null) {
76        kode = Long.toString(Long.parseLong(kode) + 1);
77    } else {
78        kode = kodeKecamatan + kodeWaktu + "0001";
79    }
80    String idKel = keluarga.selectKelurahanByName(kel.getKelurahan());
81    kel.setIdKelurahan(idKel);
82    kel.setNkk(kode);
83    System.out.println(kel.getNkk());
84    kel.setIsTdkBerlaku("0");
85    keluarga.insertKeluarga(kel);
86}
87
```

Di method ini dilakukan generate nkk. Yaitu dengan mencari kodeKecamatan sampai pada karakter ke 6. Kemudian menambahkannya dengan waktu (kodeWaktu) kapan nkk dibuat. Dan dicari apakah ada nkk yang sama (sampai 12 karakter), jika ada, maka nkk terakhir ditambah 1, jika tidak, kodekecamatan+kodewaktu + '0001'

4. Di dalam class **PendudukController**, dilakukan penambahan 2 method, yaitu menambah dua method dengan anotasi RequestMapping untuk masuk ke halaman form untuk menambah keluarga dan PostMapping untuk dilakukan penambahan keluarga dari method yang sudah dibuat di atas.

```
5
6@RequestMapping("/keluarga/tambah")
7public String tambahKeluarga(Model model) {
8    KeluargaModel keluarga = new KeluargaModel();
9    model.addAttribute("keluarga", keluarga);
10   return "form-addkeluarga";
11}
12
13@PostMapping("/keluarga/tambah")
14public String submitTambahKeluarga(Model model, @ModelAttribute KeluargaModel keluarga)
15{
16    keluargaDAO.insertKeluarga(keluarga);
17    model.addAttribute("keluarga", keluarga);
18    return "success-addkeluarga";
19}
```

5. Lanjut ke folder **templates**, menambahkan dua file html, yaitu form-addkeluarga.html dengan code seperti di bawah

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Tambah Keluarga</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Tambah Keluarga</h1>

        <form action="/keluarga/tambah" method="post" th:object="${keluarga}" >
            <div>
                <label for="alamat">Alamat</label><br/> <input type="text" name="alamat" th:field="*{alamat}" required="required"/>
            </div>
            <div>
                <label for="rt">RT</label> <br/><input type="text" name="rt" th:field="*{rt}" required="required" />
            </div>
            <div>
                <label for="rw">RW</label> <br/><input type="text" name="rw" th:field="*{rw}" required="required" />
            </div>
            <div>
                <label for="kelurahan">Kelurahan/Desa</label><br/> <input type="text" name="kelurahan" th:field="*{kelurahan}" required="required"/>
            </div>
            <div>
                <label for="kecamatan">Kecamatan</label><br/> <input type="text" name="kecamatan" th:field="*{kecamatan}" required="required" />
            </div>
            <div>
                <label for="kota">Kota</label><br/> <input type="text" name="kota" th:field="*{kota}" required="required"/>
            </div><br/>
            <div>
                <button type="submit" class="btn btn-info" name="action" value="save">Tambah</button>
            </div>
        </form>
    </div>
</body>
</html>
```

serta success-addkeluarga jika keluarga berhasil ditambahkan

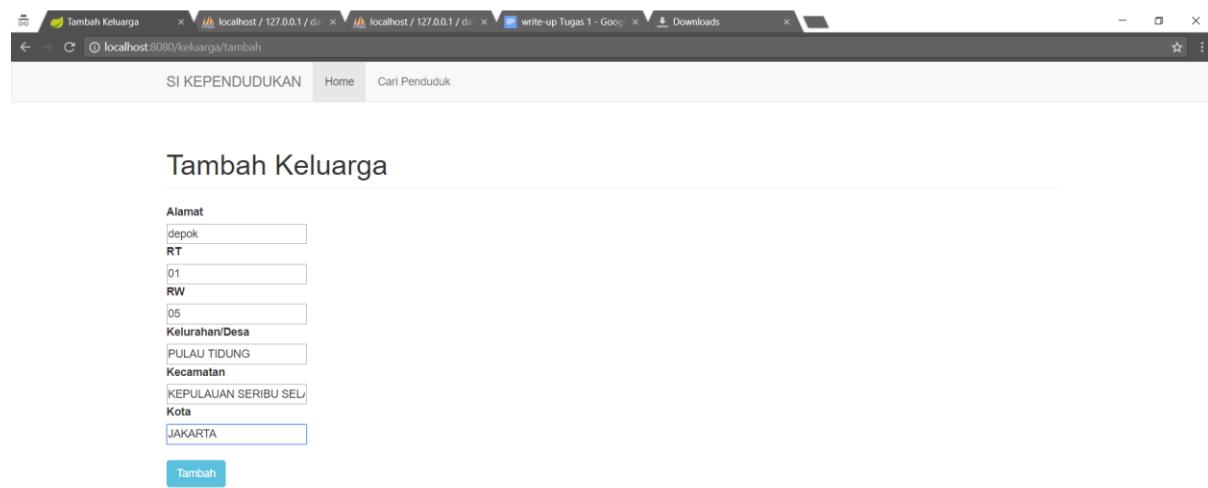
```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Tambah Keluarga</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Sukses</h1>
        <h3 th:text=""'Keluarga dengan NKK ' + ${keluarga.nkk} + ' berhasil
ditambahkan'">Berhasil!</h3>
    </div>
</body>

</html>
```

6. Kemudian menjalankan website di localhost:8080/keluarga/tambah



Klik tombol tambah dan masuk ke halam di bawah ini



Sukses

Keluarga dengan NKK 3101012210170001 berhasil ditambahkan



FITUR 5 : Mengubah Data Penduduk

- Menambahkan method implementasi UPDATE query di dalam **PendudukMapper**

```
5
6     @Update ("UPDATE `penduduk` SET `nik`=#{penduduk.nik},`nama`=#{penduduk.nama},`tempat_lahir`=#{penduduk.tempatLahir},"
7         + "`tanggal_lahir`=#{penduduk.tglLahir},"
8         + "`jenis_kelamin`=#{penduduk.jenisKelamin},`is_wni`=#{penduduk.isWNI},`id_keluarga`=#{penduduk.idKeluarga},`agama`=#{penduduk.agama},"
9         + "`pekerjaan`=#{penduduk.pekerjaan},`status_perkawinan`=#{penduduk.statusPerkawinan},"
0         + "`status_dalam_keluarga`=#{penduduk.statusDlmKeluarga},`golongan_darah`=#{penduduk.golonganDarah},`is_wafat`=#{penduduk.isWafat}"
1         + "WHERE nik = #{nomor_ik}")
2     void updatePenduduk (@Param("penduduk") PendudukModel penduduk, @Param("nomor_ik") String nomor_ik);
3
```

- Menambahkan method di interface **PendudukService** seperti method di atas dengan parameter class PendudukModel dan nik

```
void updatePenduduk (PendudukModel penduduk, String nomor_ik);
```

- Menambahkan method di class **PendudukDatabaseService** untuk implementasi method di atas

```
@Override
public void updatePenduduk(PendudukModel pend, String nomor_ik) {
    String kodeKecamatan = getKodeLurah(pend.getIdKeluarga()).substring(0,6);
    String tglLahir = pend.getTglLahir();
    String kodeLahir = null;
    if (pend.getJenisKelamin().equalsIgnoreCase("1")) {
        int tambah = Integer.parseInt(pend.getTglLahir().substring(8,10)) + 40;
        kodeLahir = tambah + tglLahir.substring(5,7) + tglLahir.substring(2,4);
    } else {
        kodeLahir = tglLahir.substring(8,10) + tglLahir.substring(5,7) + tglLahir.substring(2,4);
    }
    String nikLast = nikLastSame(kodeKecamatan + kodeLahir + "%");
    if(nikLast != null && !((kodeKecamatan+kodeLahir).equals(nomor_ik.substring(0,12)))) {
        nikLast = Long.toString(Long.parseLong(nikLast) + 1);
    } else {
        nikLast = kodeKecamatan + kodeLahir + "0001";
    }
    pend.setNik(nikLast);
    penduduk.updatePenduduk(pend,nomor_ik);
    Log.info("update penduduk with nik {}", nomor_ik);
}
```

Kurang lebih sama seperti insertPenduduk, dimana ada generate nik dan dilakukan updatePenduduk berdasarkan parameter nik yang diberikan.

- Menambahkan method pada class **PendudukController** seperti di bawah ini

```
    @RequestMapping("/penduduk/ubah/{nomor_ik}")
    public String updatePenduduk(Model model,
        @PathVariable(value = "nomor_ik") String nik) {
        PendudukModel penduduk = pendudukDAO.selectPendudukTable(nik);
        model.addAttribute("penduduk", penduduk);
        return "form-updatependuduk";
    }

    @PostMapping("/penduduk/ubah/{nomor_ik}")
    public String submitUpdatePenduduk(Model model, @ModelAttribute PendudukModel penduduk, @PathVariable(value="nomor_ik") String nik) {
        pendudukDAO.updatePenduduk(penduduk, nik);
        penduduk.setNik(nik);
        model.addAttribute("penduduk", penduduk);
        return "success-updatependuduk";
    }
}
```

Ada dua method yaitu dengan anotation `@RequestMapping` untuk menampilkan halaman form untuk mengubah suatu tupel di tabel penduduk dan dengan anotation `@PostMapping` untuk menampilkan halaman jika sukses diubah.

5. Membuat file .html di dalam folder **templates**, yaitu `form-updatependuduk` seperti di bawah

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Ubah Data Penduduk</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Ubah Penduduk</h1>

        <form th:action="${penduduk.nik}" method="post" th:object="${penduduk}" >
            <div>
                <label for="nama">Nama</label><br/> <input type="text"
name="nama" value="${penduduk.nama}" th:field="*{nama}" />
            </div>
            <div>
                <label for="tempatLahir">Tempat Lahir</label><br/> <input
type="text" name="tempatLahir" value="${penduduk.tempatLahir}" th:field="*{tempatLahir}" />
            </div>
            <div>
                <label for="tglLahir">Tanggal Lahir</label><br/> <input
type="text" name="tglLahir" value="${penduduk.tglLahir}" th:field="*{tglLahir}" />
            </div>
            <div>
                <label for="jenisKelamin">Jenis Kelamin</label><br/>
                <select name="jenisKelamin"
th:value="${penduduk.jenisKelamin}" th:field="*{jenisKelamin}">
                    <option value="0">Pria</option>
                    <option value="1">Wanita</option>
                </select>
            </div>
            <label for="isWNI">Kewarganegaraan</label><br/>
            <select name="isWNI" th:value="${penduduk.isWNI}">
```

```
th:field="*{isWNI}">
    <option value="1">WNI</option>
    <option value="0">WNA</option>
</select>
<div> <label for="idKeluarga">ID Keluarga</label><br/> <input
type="text" name="idKeluarga" value="${penduduk.idKeluarga}" th:field="*{idKeluarga}" />
</div>
<div> <label for="agama">Agama</label><br/> <input type="text"
name="agama" value="${penduduk.agama}" th:field="*{agama}" />
</div>
<div> <label for="pekerjaan">Pekerjaan</label><br/> <input
type="text" name="pekerjaan" value="${penduduk.pekerjaan}" th:field="*{pekerjaan}" />
</div>
<div> <label for="statusPerkawinan">Status Perkawinan</label><br/>
<select name="statusPerkawinan"
th:value="${penduduk.statusPerkawinan}" th:field="*{statusPerkawinan}" >
    <option value="Kawin">Kawin</option>
    <option value="Belum Kawin">Belum Kawin</option>
</select>
</div>
<div> <label for="statusDlmKeluarga">Status Dalam
Keluarga</label><br/>
<select name="statusDlmKeluarga"
th:value="${penduduk.statusDlmKeluarga}" th:field="*{statusDlmKeluarga}" >
    <option value="Anak">Anak</option>
    <option value="Kepala Keluarga">Kepala
Keluarga</option>
    <option value="Istri">Istri</option>
    <option value="Pembantu">Pembantu</option>
    <option value="Famili Lain">Famili Lain</option>
</select>
</div>
<div> <label for="golonganDarah">GolonganDarah</label><br/>
<select name="golonganDarah"
th:value="${penduduk.golonganDarah}" th:field="*{golonganDarah}" >
    <option value="A- ">A-</option>
    <option value="A+ ">A+</option>
    <option value="B- ">B-</option>
    <option value="B+ ">B+</option>
    <option value="AB- ">AB-</option>
    <option value="AB+ ">AB+</option>
    <option value="O- ">O-</option>
    <option value="O+ ">O+</option>
</select>
</div>
<div> <label for="isWafat">Status Kematian</label><br/>
<input name="isWafat" type="hidden"
th:value="${penduduk.isWafat}" th:field="*{isWafat}" />
<select name="isWafat" disabled="disabled"
th:value="${penduduk.isWafat}" th:field="*{isWafat}" >
    <option value="0">Hidup</option>
    <option value="1">Wafat</option>
</select>
</div><br/>
<div> <button class="btn btn-info" type="submit" name="action"
value="save">Ubah</button>
</div>
```

```
        </form>
    </div>
</body>

</html>
```

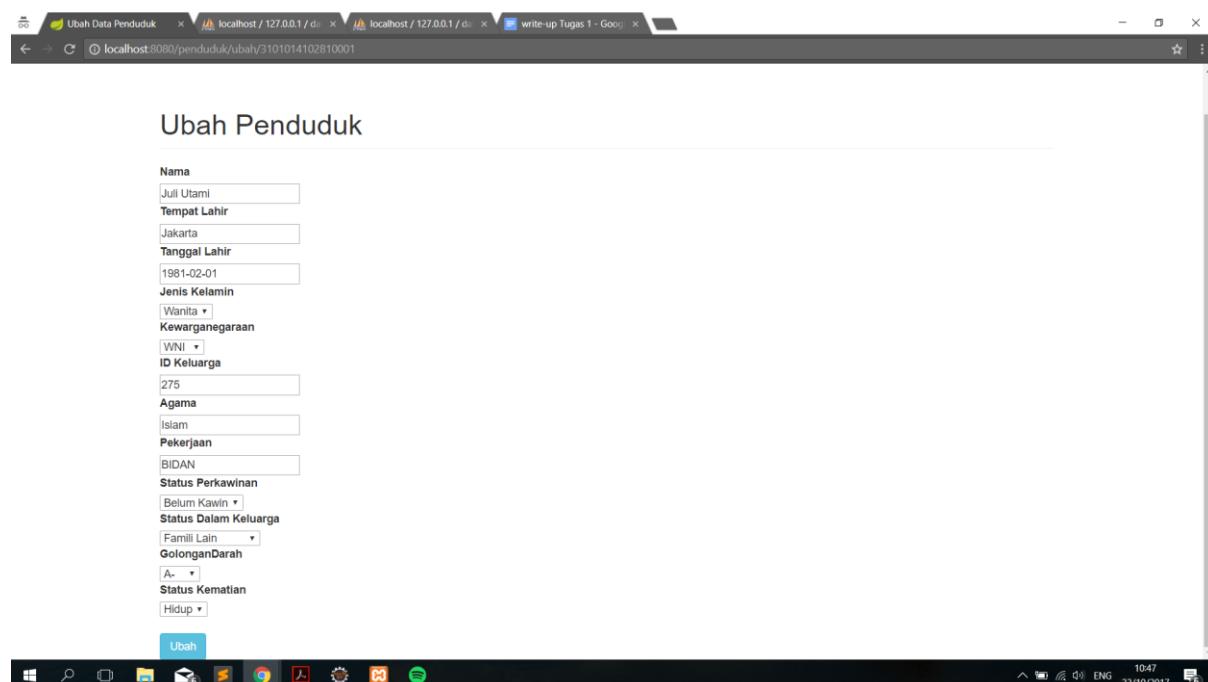
Dan juga pada success-updatependuduk.html

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Ubah Data Penduduk</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Sukses Update</h1>
        <h3 th:text=""'Penduduk dengan NIK ' + ${penduduk.nik} + ' berhasil
diubah '">Berhasil!</h3>
    </div>
</body>

</html>
```

6. Menjalankan website di localhost:8080/penduduk/ubah/{nik}
Misalkan mengubah nik 3101014102810001, diakses di
<http://localhost:8080/penduduk/ubah/3101014102810001>



Muhammad Akbar Setiadi
1506737810
APAP B

Mencoba mengganti nama dan tanggal lahir
Sebelumnya

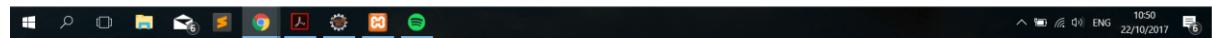
| id | nik | nama | tempat_lahir | tanggal_lahir | jenis_kelamin | is_wni | id_keluarga | agama | pekerjaan | status_perkawinan |
|--------|------------------|------------|--------------|---------------|---------------|--------|-------------|-------|-----------|-------------------|
| 281449 | 3101014102810001 | Juli Utami | Jakarta | 1981-02-01 | 1 | 1 | 275 | Islam | BIDAN | Belum Kawin |

Setelah dilakukan pengubahan nama dan tanggal lahir (nik berubah)



Sukses Update

Penduduk dengan NIK 3101014102810001 berhasil diubah



Di halaman tersebut, NIK memakai yang lama
Dan di database, nama dan tanggal lahir serta nik berubah

| id | nik | nama | tempat_lahir | tanggal_lahir | jenis_kelamin | is_wni | id_keluarga | agama | pekerjaan | status_perkawinan |
|--------|------------------|---------------|--------------|---------------|---------------|--------|-------------|-------|-----------|-------------------|
| 281449 | 3101016002810001 | Agustus Utami | Jakarta | 1981-02-20 | 1 | 1 | 275 | Islam | BIDAN | Belum Kawin |

FITUR 6 : Mengubah Data Keluarga

- Menambahkan method implementasi UPDATE query di dalam **KeluargaMapper**

```
@Update ("UPDATE `keluarga` SET `nomor_kk`=#{keluarga.nkk},`alamat` = #{keluarga.alamat},"
+ " `RT`=#{keluarga.rt},`RW`=#{keluarga.rw}, `id_kelurahan` = #{keluarga.idKelurahan},`is_tidak_berlaku` = #{keluarga.isTdkBerlaku} "
+ "WHERE `nomor_kk`=#{nomor_kk}")
void updateKeluarga(@Param("keluarga") KeluargaModel keluarga, @Param("nomor_kk") String nkk);
```

- Menambahkan method di interface **KeluargaService** seperti method di atas dengan parameter class KeluargaModel dan nkk

```
void updateKeluarga(KeluargaModel keluarga, String nkk);
```

- Selain itu juga menambahkan method selectPenduduk di interface **KeluargaMapper** untuk mengambil semua penduduk dengan id_keluarga yang sama dengan id keluarga tersebut

```
@Select ("SELECT p.nik, p.nama, p.tempat_lahir, p.tanggal_lahir, p.jenis_kelamin, p.is_wni, p.id_keluarga, p.agama, " + "p.pekerjaan, p.status_perkawinan, p.status_dalam_keluarga, p.golongan_darah,p.is_wafat " + "from penduduk p, keluarga k where p.id_keluarga = k.id and k.id = #{id}")  
@Results (value = {  
    @Result (property = "nik" , column = "nik" ),  
    @Result (property = "nama" , column = "nama" ),  
    @Result (property = "tempat_lahir" , column = "tempat_lahir" ),  
    @Result (property = "tgllahir" , column = "tanggal_lahir" ),  
    @Result (property = "jenisKelamin" , column = "jenis_kelamin" ),  
    @Result (property = "isWNI" , column = "is_wni" ),  
    @Result (property = "idKeluarga" , column = "id_keluarga" ),  
    @Result (property = "agama" , column = "agama" ),  
    @Result (property = "statusPerkawinan" , column = "status_perkawinan" ),  
    @Result (property = "pekerjaan" , column = "pekerjaan" ),  
    @Result (property = "golonganDarah" , column = "golongan_darah" ),  
    @Result (property = "statusDlmKeluarga" , column = "status_dalam_keluarga" ),  
    @Result (property = "isWafat" , column = "is_wafat" )  
})  
List<PendudukModel> selectPenduduks (@Param ("id") String id);
```

4. Menambahkan method di class **KeluargaDatabaseService** untuk implementasi method di atas

```
6@  @Override  
7  public void updateKeluarga(KeluargaModel kel, String nkk) {  
8      String kodekelurahan = getKodeKelurahan(kel.getIdKelurahan()).substring(0,6);  
9      String nomorKK = null;  
0      boolean isChanged = false;  
1      List<PendudukModel> pendudukUpdate = null;  
2      if(!kodekelurahan.equals(nkk.substring(0,6))) {  
3          nomorKK = kodekelurahan + nkk.substring(6,12);  
4          nomorKK = keluarga.nkkLast(nomorKK+"%");  
5          if(nomorKK != null) {  
6              nomorKK = Long.toString(Long.parseLong (nomorKK) + 1);  
7          } else {  
8              nomorKK = kodekelurahan + nkk.substring(6,12) + "0001";  
9          }  
0          isChanged = true;  
1          pendudukUpdate = keluarga.selectPenduduks(kel.getId());  
2      } else {  
3          nomorKK = nkk;  
4      }  
5      kel.setNkk(nomorKK);  
6      keluarga.updateKeluarga(kel, nkk);  
7      if(isChanged) {  
8          for(int i = 0; i < pendudukUpdate.size(); i++) {  
9              pendudukService.updatePenduduk(pendudukUpdate.get(i),pendudukUpdate.get(i).getNik());  
0          }  
1      }  
2      log.info("update keluarga with = {}", kel.getNkk());  
3  }
```

Kurang lebih sama seperti insertKeluarga, dimana ada generate nkk dan dilakukan updateKeluarga berdasarkan parameter nkk yang diberikan. Kemudian jika terjadi perubahan pada nkk, tentu **penduduk dengan id keluarga tersebut juga berubah niknya**

5. Menambahkan method pada class **PendudukController** seperti di bawah ini

```
7  
8@  @RequestMapping("/keluarga/ubah/{nkk}")  
9  public String updateKeluarga(Model model,  
0      @PathVariable(value = "nkk") String nkk)  
1  {  
2      KeluargaModel keluarga = keluargaDAO.selectKeluargaTable(nkk);  
3      model.addAttribute("keluarga",keluarga);  
4      return "form-updatekeluarga";  
5  }  
6  
7@  @PostMapping("/keluarga/ubah/{nkk}")  
8  public String submitUpdateKeluarga(Model model, @ModelAttribute KeluargaModel keluarga,  
9      @PathVariable(value="nkk") String nkk)  
0  {  
1      keluargaDAO.updateKeluarga(keluarga, nkk);  
2      keluarga.setNkk(nkk);  
3      model.addAttribute("keluarga",keluarga);  
4      return "success-updatekeluarga";  
5  }
```

Ada dua method yaitu updateKeluarga dengan anotation @RequestMapping untuk menampilkan halaman form untuk mengubah suatu tupel di tabel keluarga dan submitUpdateKeluarga dengan anotation @PostMapping untuk menampilkan halaman jika sukses diubah.

6. Membuat file .html di dalam folder **templates**, yaitu form-updatekeluarga seperti di bawah

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<title>Ubah Data Keluarga</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Ubah Keluarga</h1>

        <form th:action="${keluarga.nkk}" method="post" th:object="${keluarga}" >
            <div>
                <label for="alamat">Alamat</label><br/> <input type="text" name="alamat" value="${keluarga.alamat}" th:field="*{alamat}" />
            </div>
            <div>
                <label for="rt">RT</label><br/> <input type="text" name="rt" value="${keluarga.rt}" th:field="*{rt}" />
            </div>
            <div>
                <label for="rw">RW</label><br/> <input type="text" name="rw" value="${keluarga.rw}" th:field="*{rw}" />
            </div>
            <div>
                <label for="idKelurahan">ID Kelurahan</label><br/> <input type="text" name="idKelurahan" value="${keluarga.idKelurahan}" th:field="*{idKelurahan}" />
            </div>
            <div>
                <label for="isTdkBerlaku">Status Berlaku</label><br/>
                <input name="isTdkBerlaku" type="hidden" th:value="${keluarga.isTdkBerlaku}" th:field="*{isTdkBerlaku}" />
                <select name="isTdkBerlaku" disabled="disabled" th:value="${keluarga.isTdkBerlaku}" th:field="*{isTdkBerlaku}">
                    <option value="0">Tidak Berlaku</option>
                    <option value="1">Berkelaku</option>
                </select>
            </div><br/>
            <div>
                <button class="btn btn-info" type="submit" name="action" value="save">Ubah</button>
            </div>
        </form>
    </div>
</body>
</html>
```

Dan juga pada success-updatekeluarga.html

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
```

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Ubah Data Keluarga</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Sukses Update</h1>
        <h3 th:text="'Keluarga dengan NKK ' + ${keluarga.nkk} + ' berhasil
diubah'">Berhasil!</h3>
    </div>
</body>

</html>
```

7. Menjalankan website di localhost:8080/keluarga/ubah/{nik}
Misalkan mengubah nkk 3101010101050001, diakses di
<http://localhost:8080/keluarga/ubah/3101010101050001>



Ubah Keluarga

Alamat

RT

RW

ID Kelurahan

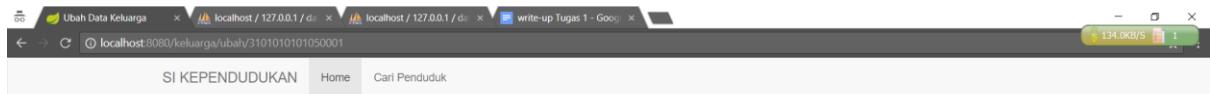
Status Berlaku

[Ubah](#)

Mencoba mengganti alamat dan id kelurahan
Sebelumnya

| # | id | nomor_kk | alamat | RT | RW | id_kelurahan | is_tidak_berlaku |
|----|----|------------------|------------------------|-----|-----|--------------|------------------|
| te | 1 | 3101010101050001 | Jln. Cikampang No. 594 | 149 | 090 | 2 | 0 |

Setelah dilakukan pengubahan alamat dan id kelurahan (nkk berubah)



Sukses Update

Keluarga dengan NKK 3101010101050001 berhasil diubah

A screenshot of a Windows desktop environment. In the center, there is a table from a database. The columns are labeled "id", "nomor_kk", "alamat", "RT", "RW", "id_kelurahan", and "is_tidak_berlaku". One row of data is visible: id=1, nomor_kk=3171030101050001, alamat=Jln. Kacimpang No. 594, RT=149, RW=090, id_kelurahan=20, is_tidak_berlaku=0.

| | id | nomor_kk | alamat | RT | RW | id_kelurahan | is_tidak_berlaku |
|---|------------------|------------------------|---------------|-----------|-----------|---------------------|-------------------------|
| 1 | 3171030101050001 | Jln. Kacimpang No. 594 | | 149 | 090 | 20 | 0 |

FITUR 7 : Mengubah Status Kematian Penduduk

- Menambah method updateTidakBerlaku di interface **KeluargaMapper** dan method updateWafat di interface **PendudukMapper**

```
    @Update("UPDATE `keluarga` SET `is_tidak_berlaku` = 1 WHERE `id` = #{id})  
    void updateTidakBerlaku(@Param("id") String id);
```

```
    @Update("UPDATE `penduduk` SET `is_wafat` = '1' WHERE `nik` = #{nik}")  
    void updateWafat(@Param("nik") String nik);
```

- Menambah method yang sama di interface **KeluargaService** untuk updateTidakBerlaku

```
    void updateTidakBerlaku(String id);
```

dan di interface **PendudukService** untuk updateWafat

```
    void updateWafat(String nik);
```

- Mengimplementasikan method updateTidakBerlaku di atas dengan *business logic* di **KeluargaDatabaseService**

```
    @Override
    public void updateTidakBerlaku(String id) {
        List<PendudukModel> anggotaKeluarga = keluarga.selectPenduduk(id);
        boolean semuaWafat = true;
        for(int i = 0; i < anggotaKeluarga.size(); i++) {
            if(anggotaKeluarga.get(i).getIsWafat().equals("0")) {
                semuaWafat = false;
            }
        }
        if(semuawafat) {
            keluarga.updateTidakBerlaku(id);
            log.info("update nkk menjadi tidak berlaku dengan id keluarga = {}", id);
        }
        log.info("nkk tetap berlaku dengan id keluarga = {}", id);
    }
}
```

Pertama, diambil semua penduduk dengan atribut id keluarga yang sama dengan id keluarga tersebut. Setelah itu dilakukan pengecekan apakah semua anggota keluarga tersebut sudah wafat atau belum. Jika sudah, maka nkk tersebut sudah tidak aktif.

4. Mengimplementasikan method updateWafat di **PendudukDatabaseService**

```
    @Override
    public void updateWafat(String nik) {
        penduduk.updateWafat(nik);
        PendudukModel orang = penduduk.selectPendudukTable(nik);
        keluargaService.updateTidakBerlaku(orang.getIdKeluarga());
        log.info("penduduk wafat dengan nik = {}", nik);
    }
}
```

Di method ini memanggil method updateTidakBerlaku yang sudah saya buat di atas.

5. Menambahkan method di **PendudukController**

```
    @RequestMapping("/penduduk")
    public String findPenduduk(Model model,@ModelAttribute("sukses_ubah_wafat") String statusWafat,@RequestParam(value = "nik") String nik)
    {
        if(statusWafat.equals("")) {
            PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
            model.addAttribute("penduduk",penduduk);
            return "get-penduduk";
        } else
        {
            model.addAttribute("nik", nik);
            return "success-updatewafat";
        }
    }

    @PostMapping("/penduduk/mati")
    public RedirectView submitUpdatePendudukMati(Model model, @RequestParam("nik") String nik, RedirectAttributes redirectAtt) {
        pendudukDAO.updateWafat(nik);
        redirectAtt.addFlashAttribute("sukses_ubah_wafat", "1");
        redirectAtt.addAttribute("nik",nik);
        return new RedirectView("/penduduk");
    }
}
```

Method findPenduduk sebenarnya sudah dibuat di fitur pertama karena berhubungan, dimana di halaman get-penduduk terdapat tombol Update Kematian. Jika diklik maka akan menuju method submitUpdatePendudukMati. Di method ini, dipanggil method updateWafat yang sudah dibuat di atas. Kemudian menambahkan atribut nik serta flash atribut sukses_ubah_wafat dan me *redirect* ke method findPenduduk.

Di method findPenduduk menangkap flash atribut sukses_ubah_wafat, sehingga menampilkan halaman success-updatewafat

6. Membuat halaman success-updatewafat di folder **templates**.

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

```
<script>
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Update Kematian</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container">
        <h1 class="page-header">Sukses!</h1>
        <h3 th:text="'Penduduk dengan NIK ' + ${nik} + ' sudah tidak
aktif'">Berhasil!</h3>
    </div>
</body>

</html>
```

7. Menjalankan website di halaman utama dan memasukan nik, contoh nik nya adalah
3101016002810001



Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Lihat Data Penduduk Berdasarkan NIK

Masukkan Nomor Induk Kependudukan

3101016002810001

Lihat

Lihat Data Keluarga Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

Lihat



Klik Lihat dan menuju halaman di bawah ini

Muhammad Akbar Setiadi
1506737810
APAP B

Lihat Data Penduduk : 3101016002810001

| | |
|----------------------|----------------------------|
| NIK | 3101016002810001 |
| Nama | Agustus Utami |
| Alamat | Ds. Dipenogoro No. 927 |
| Tempat/Tanggal Lahir | Jakarta, 20.Feb 1981 |
| RT/RW | 161/092 |
| Kelurahan/Desa | PULAU PARI |
| Kecamatan | KEPULAUAN SERIBU SELATAN |
| Kota | KABUPATEN KEPULAUAN SERIBU |
| Golongan Darah | A- |
| Agama | Islam |
| Status Perkawinan | Belum Kawin |
| Pekerjaan | BIDAN |
| Kewarganegaraan | WNI |
| Status Kematian | Hidup |

UPDATE KEMATIAN

Ketika mengklik tombol Update Kematian maka akan direct ke halaman di bawah ini

Sukses!

Penduduk dengan NIK 3101016002810001 sudah tidak aktif

Jika direfresh, maka akan menuju halaman di bawah ini

Lihat Data Penduduk : 3101016002810001

| | |
|----------------------|----------------------------|
| NIK | 3101016002810001 |
| Nama | Agustus Utami |
| Alamat | Ds. Dipenogoro No. 927 |
| Tempat/Tanggal Lahir | Jakarta, 20.Feb 1981 |
| RT/RW | 161/092 |
| Kelurahan/Desa | PULAU PARI |
| Kecamatan | KEPULAUAN SERIBU SELATAN |
| Kota | KABUPATEN KEPULAUAN SERIBU |
| Golongan Darah | A- |
| Agama | Islam |
| Status Perkawinan | Belum Kawin |
| Pekerjaan | BIDAN |
| Kewarganegaraan | WNI |
| Status Kematian | Wafat |

UPDATE KEMATIAN

Status Kematian penduduk tersebut menjadi wafat

FITUR 8 : Tampilkan Data Penduduk Berdasarkan Kota/Kabupaten, Kecamatan, dan Kelurahan Tertentu

- Untuk mengembangkan fitur ini, diperlukan model baru yang bernama **KelurahanModel** yang merepresentasikan suatu tempat di dalam package **com.akbar.tugas1.model**

```
1 package com.akbar.tugas1.model;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 @Data
8 @AllArgsConstructor;
9 @NoArgsConstructor;
10 public class KelurahanModel {
11     String id;
12     String idKota;
13     String idKecamatan;
14     String namaKelurahan;
15     String namaKota;
16     String namaKecamatan;
17     String idKelurahan;
18 }
19
```

- Membuat interface **KelurahanMapper** di package **Mapper** dengan mengimplementasikan beberapa method yang dibutuhkan.

```
KelurahanMapper.java ① KelurahanModel.java
1  @Mapper
2  public interface KelurahanMapper {
3      @Select("Select id,nama_kota from kota")
4      @Results (value = {
5          @Result (property = "idKota" , column = "id" ),
6          @Result (property = "namakota" , column = "nama_kota" )
7      })
8      List<KelurahanModel> selectAllKota();
9
10     @Select("select nama_kota from kota where id = #{id}")
11     String selectKotaById(@Param("id") String idKota);
12
13     @Select("select nama_kecamatan from kecamatan where id = #{id}")
14     String selectKecamatanById(@Param("id") String idKecamatan);
15
16     @Select("select nama_kelurahan from kelurahan where id = #{id}")
17     String selectKelurahanById(@Param("id") String idKelurahan);
18
19     @Select("SELECT k.id, k.nama_kecamatan from kecamatan k,kota ko where k.id_kota = ko.id and ko.id = #{id}")
20     @Results(value = {
21         @Result (property = "idKecamatan" , column = "id"),
22         @Result (property = "namakecamatan" , column = "nama_kecamatan")
23     })
24     List<KelurahanModel> selectKecamatanByIdKota(@Param("id") String idKota);
25
26     @Select("SELECT kel.id,kel.nama_kelurahan from kelurahan kel, kecamatan kec "
27             + "where kel.id_kecamatan = kec.id and kec.id = #{id}")
28     @Results(value = {
29         @Result(property = "id" , column = "id"),
30         @Result(property = "namakelurahan" , column = "nama_kelurahan"),
31     })
32     List<KelurahanModel> selectKelurahanByIdKecamatan(@Param("id") String idKecamatan);
33
34 }
```

Method - method tersebut adalah

1. `selectAllKota()` : memilih semua kota (id dan nama) yang ada di tabel kota
 2. `selectKotaById()` : return nama kota dengan parameter id kota yang diberikan
 3. `selectKecamatanById()` : return nama kecamatan dengan parameter id kecamatan yang diberikan
 4. `selectKelurahanById()` : return nama kelurahan dengan parameter id kelurahan yang diberikan
 5. `selectKecamatanByIdKota()` : memilih semua kecamatan berdasarkan id kota yang ada di parameter
 6. `selectKelurahanByIdKecamatan()` : memilih semua kelurahan berdasarkan id kecamatan yang ada di parameter
3. Selain itu, di dalam interface **PendudukMapper** ditambahkan method di bawah ini

```
@Mapper
public interface PendudukMapper {
    @Select("SELECT p.nik,p.nama, p.jenis_kelamin from penduduk p, keluarga k where p.id_keluarga = k.id and k.id_kelurahan = #{id}")
    @Results (value = {
        @Result (property = "nik" , column = "nik" ),
        @Result (property = "nama" , column = "nama" ),
        @Result (property = "tgllahir" , column = "tanggal_lahir" ),
        @Result (property = "jenisKelamin" , column = "jenis_kelamin" )
    })
    List<PendudukModel> selectPendudukByIdKelurahan (@Param("id") String idKelurahan);
```

Serta implementasinya di class **PendudukDatabaseService**

```
108
109     @Override
110     public List<PendudukModel> selectPendudukByIdKelurahan(String id) {
111         Log.info("select semua penduduk dengan id kelurahan = {}", id);
112         return penduduk.selectPendudukByIdKelurahan(id);
113     }
114 }
```

4. Membuat interface **KelurahanService** yang ada di package **com.akbar.tugas1.service** dan membuat method-method di atas

```

1 package com.akbar.tugas1.service;
2
3 import java.util.List;
4
5 public interface KelurahanService {
6     List<KelurahanModel> selectAllKota();
7     List<KelurahanModel> selectKecamatanByIdKota(@Param("id") String idKota);
8     List<KelurahanModel> selectKelurahanByIdKecamatan(@Param("id") String idKecamatan);
9     String selectKotaById(@Param("id") String idKota);
10    String selectKecamatanById(@Param("id") String idKecamatan);
11    String selectKelurahanById(@Param("id") String idKelurahan);
12 }
13
14 }
```

5. Membuat class **KelurahanDatabaseService** di package **com.akbar.tugas1.service** dan mengimplementasikan semua method yang ada di interface **KelurahanService** dan interface **KelurahanMapper**

```

package com.akbar.tugas1.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.akbar.tugas1.dao.KelurahanMapper;
import com.akbar.tugas1.model.KelurahanModel;

import lombok.extern.slf4j.Slf4j;

@Slf4j
@Service
public class KelurahanDatabaseService implements KelurahanService{
    @Autowired
    private KelurahanMapper kelurahan;

    @Override
    public List<KelurahanModel> selectAllKota() {
        log.info("select semua kota");
        return kelurahan.selectAllKota();
    }

    @Override
    public List<KelurahanModel> selectKecamatanByIdKota(String idKota) {
        log.info("select semua kecamatan dengan id kota = {}", idKota);
        return kelurahan.selectKecamatanByIdKota(idKota);
    }

    @Override
    public List<KelurahanModel> selectKelurahanByIdKecamatan(String idKecamatan) {
        log.info("select kelurahan dengan id kecamatan = {}", idKecamatan);
        return kelurahan.selectKelurahanByIdKecamatan(idKecamatan);
    }

    @Override
    public String selectKotaById(String idKota) {
        log.info("select nama kota dengan id = {}", idKota);
        return kelurahan.selectKotaById(idKota);
    }

    @Override
    public String selectKecamatanById(String idKecamatan) {
        log.info("select kecamatan dengan id = {}", idKecamatan);
        return kelurahan.selectKecamatanById(idKecamatan);
    }
}
```

```
@Override  
public String selectKelurahanById(String idKelurahan) {  
    log.info("Select kelurahan dengan id = {}", idKelurahan);  
    return kelurahan.selectKelurahanById(idKelurahan);  
}  
}
```

6. Menambahkan method viewPenduduk di **PendudukController** dengan anotasi `@GetMapping` sebagai berikut

```
@GetMapping("/penduduk/cari")  
public String viewPenduduk(Model model,@RequestParam(value="kt",required=false) String idKota,  
    @RequestParam(value = "kc",required =false) String idKecamatan,  
    @RequestParam(value="kl",required=false) String idKelurahan)  
{  
    if(idKota == null) {  
        model.addAttribute("listKota",kelurahanDAO.selectAllKota());  
        return "form-viewpenduduk";  
    } else if(idKecamatan == null) {  
        model.addAttribute("listKecamatan",kelurahanDAO.selectKecamatanByIdKota(idKota));  
        model.addAttribute("namaKota",kelurahanDAO.selectKotaById(idKota));  
        model.addAttribute("idKota",idKota);  
        return "form-viewpenduduk";  
    } else if(idKelurahan == null) {  
        model.addAttribute("listKelurahan",kelurahanDAO.selectKelurahanByIdKecamatan(idKecamatan));  
        model.addAttribute("namaKota",kelurahanDAO.selectKotaById(idKota));  
        model.addAttribute("namaKecamatan",kelurahanDAO.selectKecamatanById(idKecamatan));  
        model.addAttribute("idKota",idKota);  
        model.addAttribute("idKecamatan",idKecamatan);  
        return "form-viewpenduduk";  
    } else {  
        model.addAttribute("listPenduduk", pendudukDAO.selectPendudukByIdKelurahan(idKelurahan));  
        model.addAttribute("namaKota",kelurahanDAO.selectKotaById(idKota));  
        model.addAttribute("namaKecamatan",kelurahanDAO.selectKecamatanById(idKecamatan));  
        model.addAttribute("namaKelurahan", kelurahanDAO.selectKelurahanById(idKelurahan));  
        model.addAttribute("idKota",idKota);  
        model.addAttribute("idKecamatan",idKecamatan);  
        model.addAttribute("idKelurahan",idKelurahan);  
        return "view-penduduk";  
    }  
}
```

Sesuai dengan penjelasan soal, terdapat 3 parameter dengan required=false, yaitu kt merujuk id kota, kc merujuk id kecamatan dan kl merujuk id kelurahan.

Kemudian untuk menambahkan atribut pada model (model.addAttribute), dibutuhkan beberapa *desicion* seperti di atas. Karena tiap parameter yang diberikan, mempunyai tampilan yang berbeda pula tergantung parameternya.

1. Jika menuju ke localhost:8080/cari, masuk ke if pertama
2. Jika menuju ke localhost:8080/cari?kt={idKota} , masuk ke if ke dua
3. Jika menuju ke localhost:8080/cari?kt={idKota}&kc={idKecamatan}, masuk ke if ketiga
4. Jika menuju ke localhost:8080/cari?kt={idKota}&kc={idKecamatan}&kl={idKelurahan} , masuk ke if keempat

7. Membuat file html dengan nama form-viewpenduduk dengan code di bawah

```
<!DOCTYPE HTML>  
<html xmlns="http://www.w3.org/1999/xhtml"  
      xmlns:th="http://www.thymeleaf.org">  
<head>  
<meta name="viewport" content="width=device-width, initial-scale=1" />  
<link rel="stylesheet"  
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  
<script  
      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
<title>View Penduduk In Area</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <nav th:replace = "fragments/fragment :: navbar"></nav>
    <div class="container" th:if="${namaKota == null}">
        <h3>Lihat Data Penduduk Berdasarkan Kota</h3>
        <form action="/penduduk/cari" method="get">
            <select name="kt">
                <option th:each="kota: ${listKota}" th:value="${kota.idKota}" th:text="${kota.namaKota}">Tidak ditemukan</option>
            </select>
            <div><br/>
                <button class="btn btn-info" type="submit">Lihat</button>
            </div>
        </form>
    </div>

    <div class="container" th:if="${namaKota != null and namaKecamatan == null}">
        <h3>Lihat Data Penduduk Berdasarkan Kecamatan di <span th:text="${namaKota}" th:remove="tag"></span></h3>
        <form action="/penduduk/cari" method="get">
            <input type="hidden" name="kt" th:value="${idKota}" />
            <select disabled="disabled" name="kt" th:value="${idKota}">
                <option th:value="${idKota}" th:text="${namaKota}" selected="selected"></option>
            </select>
            <select name="kc">
                <option th:each="kecamatan: ${listKecamatan}" th:value="${kecamatan.idKecamatan}" th:text="${kecamatan.namaKecamatan}">Tidak ditemukan</option>
            </select>
            <div><br/>
                <button class="btn btn-info" type="submit">Lihat</button>
            </div>
        </form>
    </div>

    <div class="container" th:if="${namaKecamatan != null and namaKelurahan == null}">
        <h3>Lihat Data Penduduk Berdasarkan Kelurahan di <span th:text="${namaKota}" th:remove="tag"></span>, Kecamatan <span th:text="${namaKecamatan}" th:remove="tag"></span></h3>
        <form action="/penduduk/cari" method="get">
            <input type="hidden" name="kt" th:value="${idKota}" />
            <input type="hidden" name="kc" th:value="${idKecamatan}" />
            <select disabled="disabled" name="kt" th:value="${idKota}">
                <option th:value="${idKota}" th:text="${namaKota}" selected="selected"></option>
            </select>
            <select disabled="disabled" name="kc" th:value="${idKecamatan}">
                <option th:value="${idKecamatan}" th:text="${namaKecamatan}" selected="selected"></option>
            </select>
            <select name="kl">
                <option th:each="kelurahan: ${listKelurahan}" th:value="${kelurahan.id}" th:text="${kelurahan.namaKelurahan}">Tidak ditemukan</option>
            </select>
            <div><br/>
                <button class="btn btn-info" type="submit">Lihat</button>
            </div>
        </form>
    </div>
</body>
```

```

<div class="container" th:if="${namaKelurahan != null}">
    <h3>Lihat Data Penduduk Berdasarkan Kelurahan di <span
th:text="${namaKota}" th:remove="tag"></span>,
    Kecamatan <span th:text="${namaKecamatan}" th:remove="tag"></span>,
    Kelurahan <span th:text="${namaKelurahan}" th:remove="tag"></span></h3>
    </div>
</body>
</html>

```

Terdapat 3 div, bergantung pada else if yang ada di controller di atas, karena seperti yang sudah dijelaskan, ketiganya memiliki tampilan yang berbeda

8. Membuat file view-penduduk untuk menampilkan semua penduduk di suatu kelurahan

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <link rel="stylesheet" type="text/css"
        href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.css" />
        <script type="text/javascript" charset="utf8"
        src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.js"></script>
        <title>View Penduduk by Area</title>
    </head>
    <body>
        <nav th:replace = "fragments/fragment :: navbar"></nav>
        <div class="container">
            <h3>Lihat Data Penduduk Berdasarkan Kelurahan di <span
th:text="${namaKota}" th:remove="tag"></span>,
            Kecamatan <span th:text="${namaKecamatan}" th:remove="tag"></span>,
            Kelurahan <span th:text="${namaKelurahan}" th:remove="tag"></span></h3>
            <br/><br/><br/>

            <h3>Penduduk Dengan Usia Termuda</h3>
            <table class="table">
                <tbody>
                    <tr>
                        <td>NIK</td>
                        <td th:text="${pendudukMuda.nik}">NIK</td>
                    </tr>
                    <tr>
                        <td>Nama</td>
                        <td th:text="${pendudukMuda.nama}">Nama</td>
                    </tr>
                    <tr>
                        <td>Tanggal Lahir</td>
                        <td th:text="${pendudukMuda.tglLahir}">Tanggal
Lahir</td>
                    </tr>
                </tbody>
            </table>
            <h3>Penduduk Dengan Usia Tertua</h3>
            <table class="table">
                <tbody>

```

```
<tr>
    <td>NIK</td>
    <td th:text="${pendudukTua.nik}">NIK</td>
</tr>
<tr>
    <td>Nama</td>
    <td th:text="${pendudukTua.nama}">Nama</td>
</tr>
<tr>
    <td>Tanggal Lahir</td>
    <td th:text="${pendudukTua.tglLahir}">Tanggal
Lahir</td>
</tr>
</tbody>
</table>
<br/><br/>
<table id="table_id" class="display">
    <thead>
        <tr>
            <th>No.</th>
            <th>NIK</th>
            <th>Nama</th>
            <th>Jenis Kelamin</th>
        </tr>
    </thead>
    <tbody>
        <tr th:each="penduduk,iterationStatus: ${listPenduduk}" >
            <td th:text="${iterationStatus.index +
1}"></td>
            <td th:text="${penduduk.nik}">NIK</td>
            <td th:text="${penduduk.nama}">Nama</td>
            <td th:text="${penduduk.jenisKelamin}">Jenis
Kelamin</td>
        </tr>
    </tbody>
</table>
</div>
<script>
$(document).ready( function () {
    $('#table_id').DataTable();
} );
</script>
</body>
</html>
```

9. Mencoba menjalankan di <http://localhost:8080/penduduk/cari>
Semisal saya mencoba untuk id kota = 1 dan id kecamatan = 1



Lihat Data Penduduk Berdasarkan Kelurahan di KABUPATEN KEPULAUAN SERIBU, Kecamatan KEPULAUAN SERIBU SELATAN

Kota/Kabupaten

KABUPATEN KEPULAUAN SERIBU ▾

Kecamatan

KEPULAUAN SERIBU SELATAN ▾

Kelurahan

PULAU TIDUNG ▾

Lihat



Kemudian klik tombol Lihat. Maka akan merefer ke
<http://localhost:8080/penduduk/cari?kt=1&kc=1&kl=1>



Lihat Data Penduduk Berdasarkan Kelurahan di KABUPATEN KEPULAUAN SERIBU, Kecamatan KEPULAUAN SERIBU SELATAN, Kelurahan PULAU TIDUNG

Penduduk Dengan Usia Termuda

NIK 3101015709170001

Nama Dewi Novitasari

Tanggal Lahir 2017-09-17

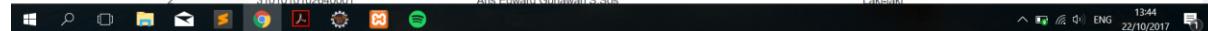
Penduduk Dengan Usia Tertua

NIK 3101012409520001

Nama Oskar Napitupulu S.Kom

Tanggal Lahir 1952-09-24

| Show 10 entries | | | |
|-----------------|------------------|---------------------------|---------------|
| No. | NIK | Nama | Jenis Kelamin |
| 1 | 3101011405170001 | Heru Haryanto | Laki-Laki |
| 2 | 3101010102640001 | Aris Edward Gunawan S.Sos | Laki-Laki |



Tanggal Lahir 2017-09-17

Penduduk Dengan Usia Tertua

| No. | NIK | Nama | Jenis Kelamin |
|-----|------------------|---------------------------------|---------------|
| 1 | 3101011405170001 | Heru Haryanto | Laki-laki |
| 2 | 3101010102640001 | Aris Edward Cunawan S.Sos | Laki-laki |
| 3 | 3101011312160002 | Pangeran Chandra Prasetyo S.Kom | Laki-laki |
| 4 | 3101010103810001 | Karma Samosir | Laki-laki |
| 5 | 3101010103920001 | Wakiman Sitorus | Laki-laki |
| 6 | 3101010103940001 | Jarwi Sihombing S.E.I | Laki-laki |
| 7 | 3101010104890001 | Taswir Hutasoit | Laki-laki |
| 8 | 3101010104910001 | Mahfud Setiawan M.TI. | Laki-laki |
| 9 | 3101010105890001 | Radika Dabukke | Laki-laki |
| 10 | 3101011502150002 | Estiono Sihombing | Laki-laki |

Show 10 entries Search: _____

Showing 1 to 10 of 1,143 entries Previous 1 2 3 4 5 ... 115 Next

13:44 22/10/2017

BONUS : Menambah Error Page

1. Terdapat dua error page yang saya buat, yaitu jika terjadi 404 error dan 500 error
2. Dengan membuat folder **error** di dalam folder **templates** dan membuat kedua file html tersebut

404 ERROR

SI KEPENDUDUKAN Home Cari Penduduk

404 Error

Halaman Tidak Ditemukan.

Kembali Ke Halaman Utama

13:46 22/10/2017

500 ERROR

Muhammad Akbar Setiadi
1506737810
APAP B



ERROR!

Nomor atau Kata Kunci Tidak Ada di Database

[Kembali Ke Halaman Utama](#)



Hanya untuk validasi jika kosong, dengan menambahkan tag `required="required"`
Contoh tampilannya seperti berikut



Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Lihat Data Penduduk Berdasarkan NIK

Masukkan Nomor Induk Kependudukan

 Please fill out this field.

Lihat Data Keluarga Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

[Lihat](#)



BONUS : Menampilkan Penduduk Paling Muda dan Paling Tua di Suatu Kelurahan

1. Membuat method `orangTertua` dan `orangTermuda` di class **PendudukDatabaseService** dan interface **PendudukService** dengan parameter list sekumpulan penduduk di suatu kelurahan

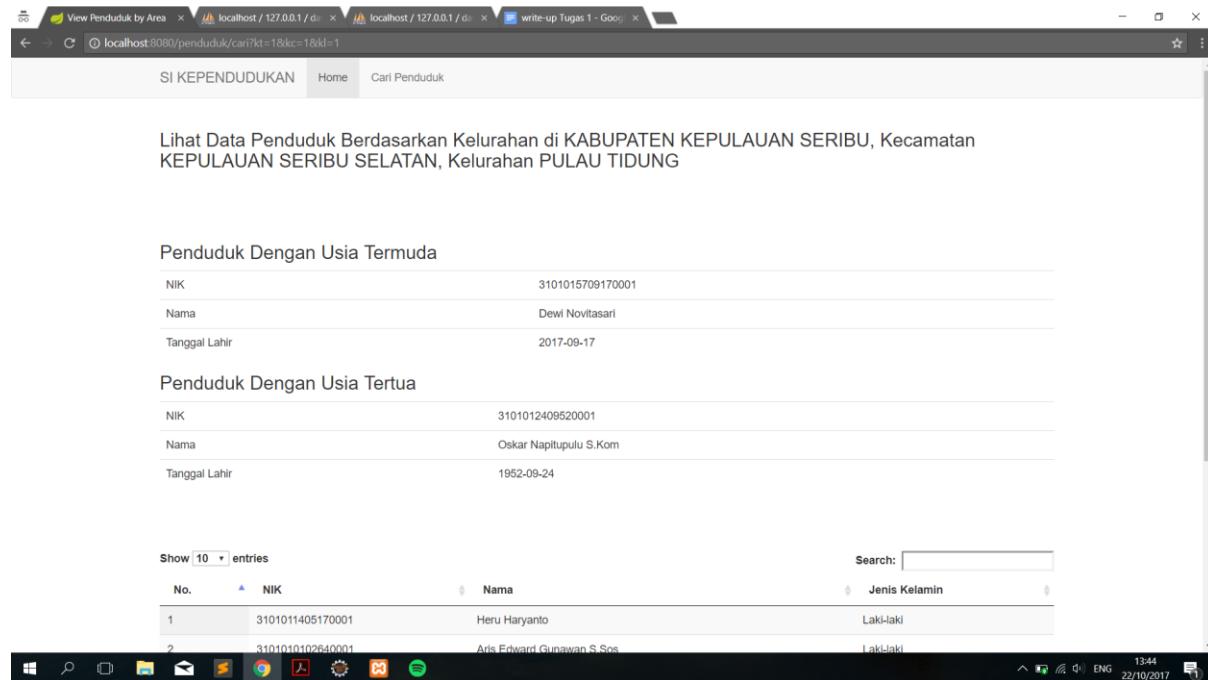
```
@Override
public PendudukModel orangTertua(List<PendudukModel> listPenduduk) {
    PendudukModel tua = listPenduduk.get(0);
    for(int i = 0; i <listPenduduk.size();i++)
    {
        if(listPenduduk.get(i).getTglLahir().compareTo(tua.getTglLahir()) < 0)
        {
            tua = listPenduduk.get(i);
        }
    }
    return tua;
}

@Override
public PendudukModel orangTermuda(List<PendudukModel> listPenduduk) {
    PendudukModel muda = listPenduduk.get(0);
    for(int i = 0; i <listPenduduk.size();i++)
    {
        if(muda.getTglLahir().compareTo(listPenduduk.get(i).getTglLahir()) < 0)
        {
            muda = listPenduduk.get(i);
        }
    }
    return muda;
}
```

2. Kemudian menambahkan model atribut di dalam method viewPenduduk untuk menyimpan atribut penduduk tertua dan termuda

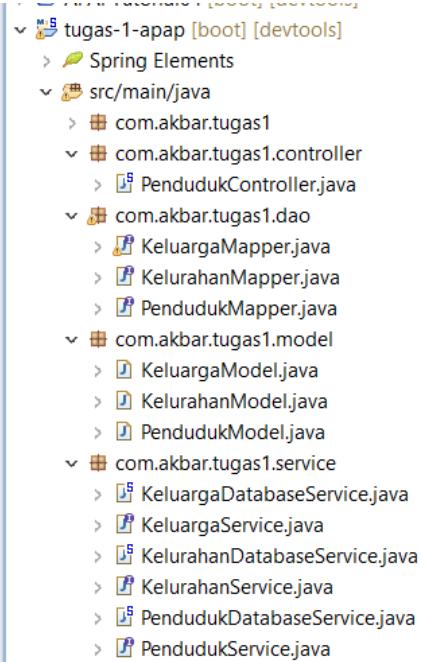
```
130
131 @GetMapping("/penduduk/cari")
132 public String viewPenduduk(Model model,@RequestParam(value="kt",required=false) String idKota,
133 @RequestParam(value = "kc",required =false) String idKecamatan,
134 @RequestParam(value="kl",required=false) String idKelurahan)
135 {
136     if(idKota == null) {
137         model.addAttribute("listKota",kelurahanDAO.selectAllKota());
138         return "form-viewpenduduk";
139     } else if(idKecamatan == null) {
140         model.addAttribute("listKecamatan",kelurahanDAO.selectKecamatanByIdKota(idKota));
141         model.addAttribute("namaKota",kelurahanDAO.selectKotaById(idKota));
142         model.addAttribute("idKota",idKota);
143         return "form-viewpenduduk";
144     } else if(idKelurahan == null) {
145         model.addAttribute("listKelurahan",kelurahanDAO.selectKelurahanByIdKecamatan(idKecamatan));
146         model.addAttribute("namaKota",kelurahanDAO.selectKotaById(idKota));
147         model.addAttribute("namaKecamatan",kelurahanDAO.selectKecamatanById(idKecamatan));
148         model.addAttribute("idKota",idKota);
149         model.addAttribute("idKecamatan",idKecamatan);
150         return "form-viewpenduduk";
151     } else {
152         List<PendudukModel> list = pendudukDAO.selectPendudukByIdKelurahan(idKelurahan);
153         model.addAttribute("listPenduduk", list);
154         model.addAttribute("pendudukMuda", pendudukDAO.orangTermuda(list));
155         model.addAttribute("pendudukTua", pendudukDAO.orangTertua(list));
156         model.addAttribute("namaKota",kelurahanDAO.selectKotaById(idKota));
157         model.addAttribute("namaKecamatan",kelurahanDAO.selectKecamatanById(idKecamatan));
158         model.addAttribute("namaKelurahan", kelurahanDAO.selectKelurahanById(idKelurahan));
159         model.addAttribute("idKota",idKota);
160         model.addAttribute("idKecamatan",idKecamatan);
161         model.addAttribute("idKelurahan",idKelurahan);
162         return "view-penduduk";
163     }
164 }
```

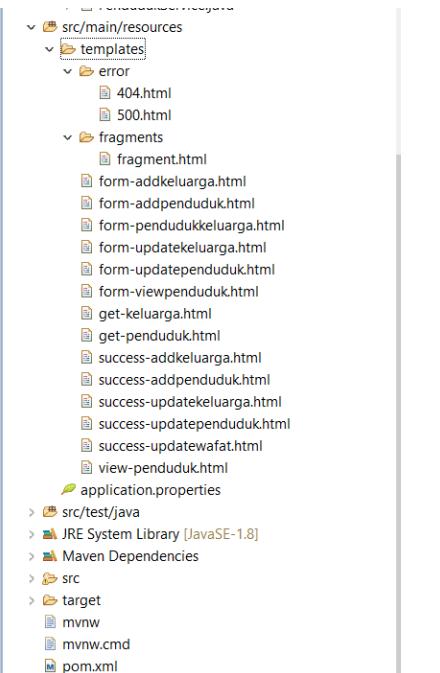
3. Menambahkan 2 tabel di view-penduduk.html untuk menampilkan penduduk tertua dan termuda seperti yang ada di atas
4. Tampilannya menjadi seperti berikut di bawah ini



Penjelasan Struktur Project, Package dan Implementasi MVC Optimasi Database

Berikut merupakan struktur project yang sudah dibuat





Pada package **com.akbar.tugas1.controller** yang merupakan Layer Controller di project ini, terdapat 1 class yang bernama PageController.java yang merepresentasikan controller untuk menghubungkan semua service dengan semua view yang sesuai.

Kemudian ada package **com.akbar.tugas1.dao** yang memiliki 3 interface untuk megimplementasikan query-query yang dibutuhkan untuk database ke dalam method di Java Spring. Ketiga interface tersebut tentu berbeda bergantung pada representasi model masing-masing.

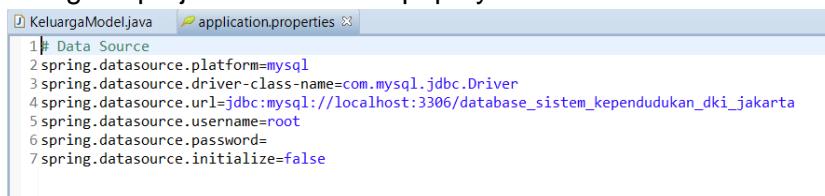
Lanjut ke dalam **com.akbar.tugas1.model** yang merupakan layer Model di project ini. Terdapat 3 class model, untuk merepresentasikan data apa saja yang diperlukan di dalam project ini. Tiap class model merepresentasikan hal yang berbeda, PendudukModel merepresentasikan tabel penduduk di database serta kaitannya dengan tabel lain (seperti alamat penduduk didapat dari tabel keluarga), KeluargaModel merepresentasikan tabel keluarga di database serta kaitannya dengan tabel lain (seperti nama kecamatan tempat tinggal keluarga tersebut didapat dari tabel kecamatan), serta KelurahanModel merepresentasikan suatu tempat yang memiliki identitas nama kelurahan dan idnya, nama kecamatan dan id nya, dan nama kota beserta id nya.

Pada **com.akbar.tugas1.service**, tempat dimana semua *business logic* yang ada di dalam project ini, seperti melakukan generate NIK, generate NKK, mengubah format tanggal, mengubah format status kematian, format jenis kelamin, dan lain lain. terdapat 6 class, dimana 3 nya adalah interface (abstrak) yang diimplementasikan semua methodnya oleh 3 concrete class yang lain (PendudukDAOService, KelurahanDAOService, KeluargaDAOService).

Kemudian pada **src/main/resources** terdapat 1 folder, yaitu templates, dimana di dalam folder templates terdapat 2 folder lagi, yaitu error dan fragments. Folder templates berisi semua View resources (file html) yang dibutuhkan untuk menampilkan halaman dengan struktur yang diinginkan. Tiap view megimplementasikan **framework Bootstrap** dan **JQuery**, serta 1 view yaitu "view-penduduk" mengimplementasikan **framework DataTables**. Di dalam folder error berisi semua view yang akan ditampilkan jika terjadi error, tergantung jenis error yang terjadi (dibedakan berdasarkan nama). Dan yang terakhir yaitu

fragments yang berisi tag apa saja yang sekiranya diimplementasikan oleh banyak view, untuk *efficiency code*.

Di dalam **src/main/resources** juga terdapat application.properties untuk menghubungkan project ke database phpmyadmin.



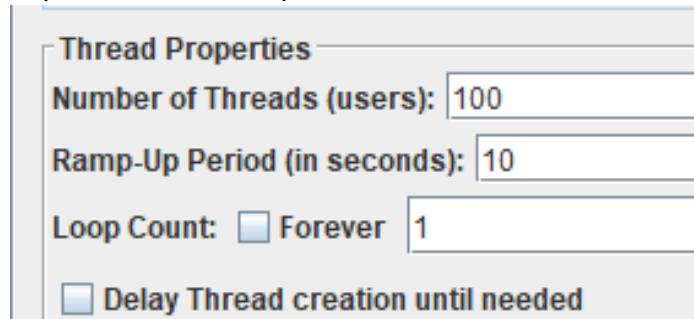
```
1# Data Source
2spring.datasource.platform=mysql
3spring.datasource.driver-class-name=com.mysql.jdbc.Driver
4spring.datasource.url=jdbc:mysql://localhost:3306/database_sistem_kependudukan_dki_jakarta
5spring.datasource.username=root
6spring.datasource.password=
7spring.datasource.initialize=false
```

Di luar itu, terdapat file pom.xml yang berisi dependensi-dependensi yang dibutuhkan oleh project ini.

Stress Testing

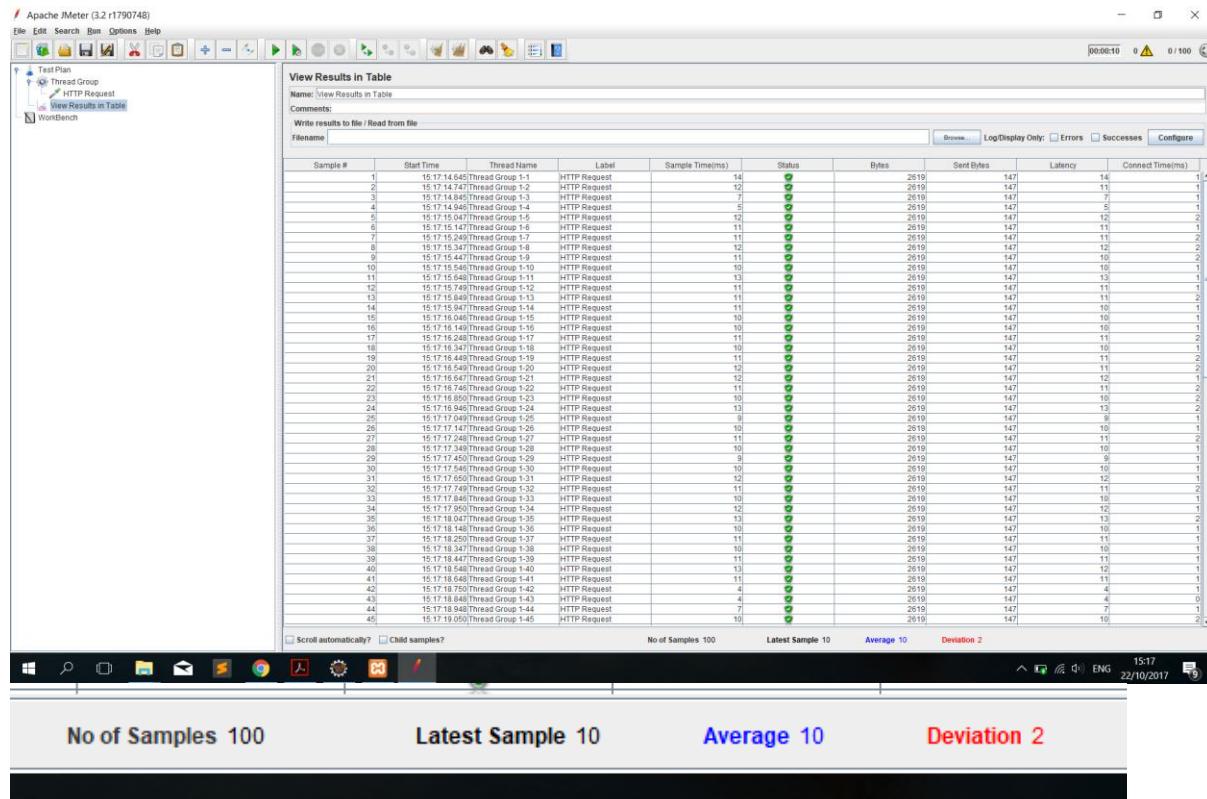
Dengan menggunakan Apache JMeter, saya mencoba menganalisis stress testing terhadap projek yang sudah dibuat.

Pertama untuk menguji fitur 1. Dimana Number of Threads saya set menjadi 100, Ramp-Up Period saya set menjadi 10 dan Loop Count 1 dan melakukan HTTP Request GET ke <http://localhost:8080/penduduk?nik=3101010103140001>



Ketika dilakukan pengujian, dari thread yang sedikit sampai banyak mempunyai sample time yang relatif konstan dengan rata - rata kurang lebih 10 mili detik.

Muhammad Akbar Setiadi
1506737810
APAP B



Kemudian saya mencoba menguji kembali untuk fitur ke 2, dengan HTTP Request Method GET ke <http://localhost:8080/keluarga/nkk?3101010104070001>

Didapat hasil sebagai berikut

No of Samples 100 Latest Sample 1727 Average 1477 Deviation 505

Rata-rata mencapai 1 detik dikarenakan ketika mencari suatu tuple di dalam tabel keluarga, juga dilakukan pencarian anggota keluarga di tabel penduduk dengan mencocokan berdasarkan id keluarga

Kemudian saya mencoba menguji kembali untuk fitur ke 3, dengan HTTP Request Method POST ke <http://localhost:8080/penduduk/tambah>

Didapat hasil sebagai berikut

No of Samples 100 Latest Sample 6 Average 8 Deviation 4

Masih terhitung cepat

Kemudian saya mencoba menguji kembali untuk fitur ke 4, dengan HTTP Request Method POST ke <http://localhost:8080/keluarga/tambah>

Didapat hasil sebagai berikut

Muhammad Akbar Setiadi
1506737810
APAP B

| | | | |
|-------------------|------------------|-----------|-------------|
| No of Samples 100 | Latest Sample 10 | Average 7 | Deviation 3 |
| | | | |

Masih terhitung cepat

Kemudian saya mencoba menguji kembali untuk fitur ke 5, dengan HTTP Request Method POST ke <http://localhost:8080/penduduk/ubah/3101010101170001>

Didapat hasil sebagai berikut

| | | | |
|-------------------|-----------------|-----------|-------------|
| No of Samples 100 | Latest Sample 8 | Average 8 | Deviation 2 |
| | | | |

Masih terhitung cepat

Kemudian saya mencoba menguji kembali untuk fitur ke 6, dengan HTTP Request Method POST ke <http://localhost:8080/keluarga/ubah/3101010104070001>

Didapat hasil sebagai berikut

| | | | |
|-------------------|-----------------|-----------|-------------|
| No of Samples 100 | Latest Sample 4 | Average 9 | Deviation 3 |
| | | | |

Masih terhitung cepat

Kemudian saya mencoba menguji kembali untuk fitur ke 7, dengan HTTP Request Method POST ke <http://localhost:8080/penduduk/mati>

Didapat hasil sebagai berikut

| | | | |
|-------------------|-----------------|-----------|-------------|
| No of Samples 100 | Latest Sample 5 | Average 4 | Deviation 1 |
| | | | |

Masih terhitung cepat

Kemudian terakhir, saya mencoba menguji kembali untuk fitur ke 8, dengan HTTP Request Method GET ke <http://localhost:8080/penduduk/ubah/3101010101170001>

Ini adalah fitur terberat karena harus melakukan SELECT query terhadap penduduk yang relatif banyak

Didapat hasil sebagai berikut

| | | | |
|-------------------|---------------------|---------------|----------------|
| No of Samples 100 | Latest Sample 20926 | Average 18111 | Deviation 5988 |
| | | | |

Dengan waktu di atas menurut saya cukup lama namun masih tetap reliable untuk diaplikasikan.

OPTIMASI DATABASE

Saya sudah melakukan berbagai upaya dalam mengoptimalkan waktu eksekusi website project ini. Di antaranya adalah dengan melakukan indexing kepada semua atribut yang

pasti unik di semua tabel yang ada di database. Kemudian menambah primary key ke setiap atribut id di setiap tabel yang ada.

Untuk implementasi query di dalam method juga sebisa mungkin saya minimalkan, seperti SELECT query yang hanya mengambil atribut yang perlu. Sehingga ada kasus dimana saya sampai membuat lebih dari 1 method untuk jenis SELECT yang sama, karena atribut yang diambil berbeda.

```
@Select ("select p.nik,p.nama,p.tempat_lahir,p.tanggal_lahir,k.alamat,k.rt,k.rw,k.nama_kelurahan,"  
+ "kc.nama_kecamatan, ko.nama_kota, p.golongan_darah, p.agama, p.status_perkawinan, p.pekerjaan, p.is_wni, p.is_wafat "  
+ "from penduduk p, keluarga k, kelurahan kl,kecamatan kc,kota ko where p.id_keluarga = k.id and "  
+ "k.id_kelurahan = kl.id and kl.id_kecamatan = kc.id and kc.id_kota = ko.id and p.nik = #{nik}")  
@Results (value = {  
    @Result (property = "nik" , column = "nik" ),  
    @Result (property = "nama" , column = "nama" ),  
    @Result (property = "tempat_lahir" , column = "tempat_lahir" ),  
    @Result (property = "tgllahir" , column = "tanggal_lahir" ),  
    @Result (property = "alamat" , column = "alamat" ),  
    @Result (property = "rt" , column = "rt" ),  
    @Result (property = "rw" , column = "rw" ),  
    @Result (property = "kelurahan" , column = "nama_kelurahan" ),  
    @Result (property = "kecamatan" , column = "nama_kecamatan" ),  
    @Result (property = "kota" , column = "nama_kota" ),  
    @Result (property = "golonganDarah" , column = "golongan_darah" ),  
    @Result (property = "agama" , column = "agama" ),  
    @Result (property = "statusPerkawinan" , column = "status_perkawinan" ),  
    @Result (property = "pekerjaan" , column = "pekerjaan" ),  
    @Result (property = "isWNI" , column = "is_wni" ),  
    @Result (property = "isWafat" , column = "is_wafat" )  
})  
PendudukModel selectPenduduk (@Param ("nik") String nik);
```

dengan

```
PendudukModel selectPenduduk (@Param ("nik") String nik);  
  
@Select ("SELECT `nik`, `nama`, `tempat_lahir`, `tanggal_lahir`, `jenis_kelamin`, `is_wni`, `id_keluarga`, `agama`, "  
+ " `pekerjaan`, `status_perkawinan`, `status_dalam_keluarga`, `golongan_darah`, `is_wafat` FROM `penduduk` WHERE nik = #{nik}")  
@Results (value = {  
    @Result (property = "nik" , column = "nik" ),  
    @Result (property = "nama" , column = "nama" ),  
    @Result (property = "tempat_lahir" , column = "tempat_lahir" ),  
    @Result (property = "tgllahir" , column = "tanggal_lahir" ),  
    @Result (property = "jenisKelamin" , column = "jenis_kelamin" ),  
    @Result (property = "isWNI" , column = "is_wni" ),  
    @Result (property = "idKeluarga" , column = "id_keluarga" ),  
    @Result (property = "agama" , column = "agama" ),  
    @Result (property = "statusPerkawinan" , column = "status_perkawinan" ),  
    @Result (property = "pekerjaan" , column = "pekerjaan" ),  
    @Result (property = "golonganDarah" , column = "golongan_darah" ),  
    @Result (property = "statusDlmKeluarga" , column = "status_dalam_keluarga" ),  
    @Result (property = "isWafat" , column = "is_wafat" )  
})  
PendudukModel selectPendudukTable (@Param ("nik") String nik);
```