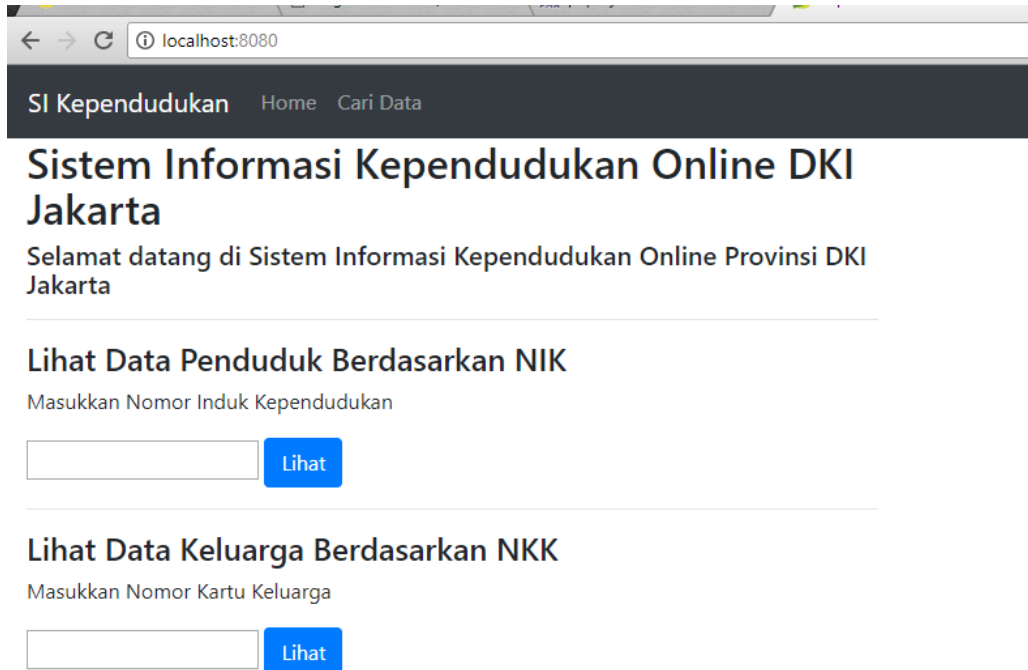
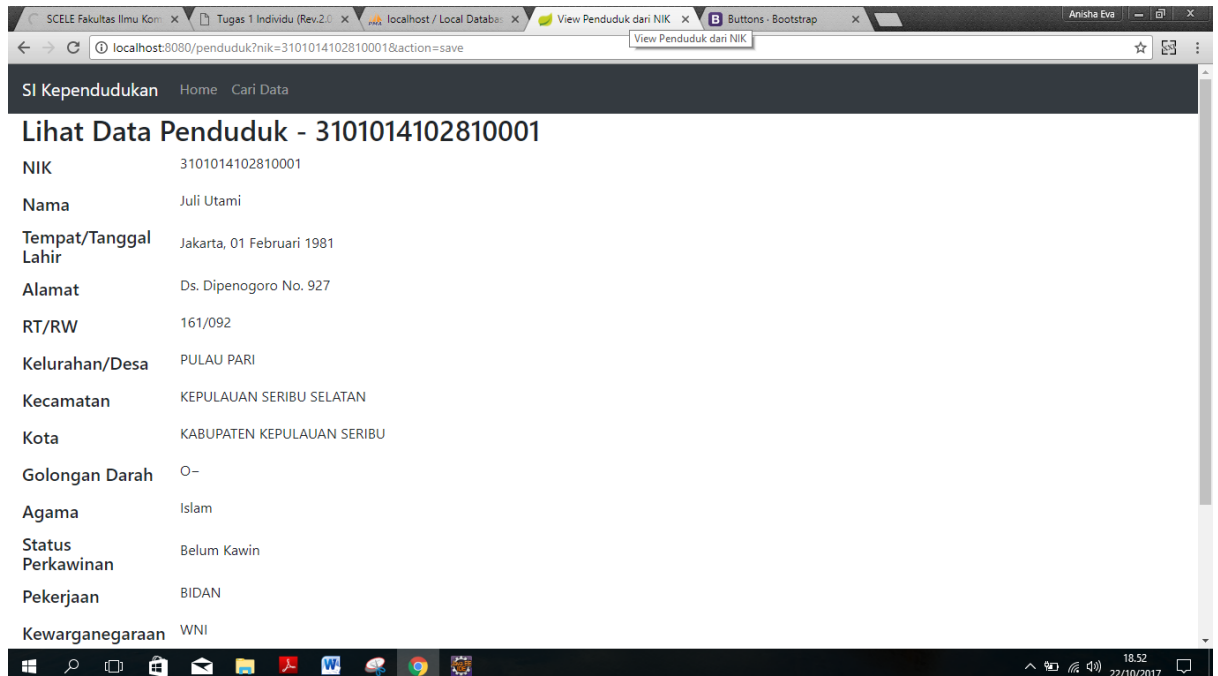


Berikut ini tampilan index.html pada program saya.



The screenshot shows the index page of the 'SI Kependudukan Online DKI Jakarta' application. The page has a dark blue header with the title 'SI Kependudukan' and navigation links 'Home' and 'Cari Data'. Below the header, the main title 'Sistem Informasi Kependudukan Online DKI Jakarta' is displayed in a large, bold font. Underneath, a subtitle reads 'Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta'. The page is divided into two main sections. The first section is titled 'Lihat Data Penduduk Berdasarkan NIK' and includes a prompt 'Masukkan Nomor Induk Kependudukan'. It features a text input field and a blue 'Lihat' button. The second section is titled 'Lihat Data Keluarga Berdasarkan NKK' and includes a prompt 'Masukkan Nomor Kartu Keluarga'. It also features a text input field and a blue 'Lihat' button.

1. Tampilan Data Penduduk Berdasarkan NIK
  - Ketika pencarian sudah ditemukan



The screenshot shows the 'Lihat Data Penduduk' page after a successful search. The page title is 'Lihat Data Penduduk - 3101014102810001'. The data is presented in a table with two columns: labels and values. The data is as follows:

NIK	3101014102810001
Nama	Juli Utami
Tempat/Tanggal Lahir	Jakarta, 01 Februari 1981
Alamat	Ds. Dipenogoro No. 927
RT/RW	161/092
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	O-
Agama	Islam
Status Perkawinan	Belum Kawin
Pekerjaan	BIDAN
Kewarganegaraan	WNI

Kewarganegaraan WNI

Jenis Kelamin Wanita

Status Kematian Wafat

Nonaktifkan

Perbaharui Data

- Pada PendudukController

ment.html index.html cari-penduduk.html PendudukController.java vi

```

@RequestMapping("/penduduk")
public String view (Model model,
    @RequestParam(value = "nik", required = false) String nik,
    @RequestParam(value = "tanggal", required = false) String tanggal,
    @RequestParam(value = "bulan", required = false) String bulan,
    @RequestParam(value = "tahun", required = false) String tahun)
{
    PendudukModel penduduk = pendudukDAO.selectPenduduk (nik);
    if (penduduk != null) {
        KeluargaModel keluarga = keluargaDAO.selectKeluarga(nik);
        KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(nik);
        KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(nik);
        KotaModel kota = kotaDAO.selectKota(nik);

        String[] tglLahir = penduduk.getTanggal_lahir().split("-");
        String tahun1 = tglLahir[0];
        String bulan1 = tglLahir[1];
        String tanggal1 = tglLahir[2];
        int bulanFix = Integer.parseInt(bulan1);

        if (bulanFix == 1) {
            bulan1 = "Januari";
        } else if (bulanFix == 2) {
            bulan1 = "Februari";
        } else if (bulanFix == 3) {
            bulan1 = "Maret";
        } else if (bulanFix == 4) {
            bulan1 = "April";
        }
    }
}

```

```

ml    index.html    cari-penduduk.html    PendudukController.java
    } else if (bulanFix == 5) {
        bulan1 = "Mei";
    } else if (bulanFix == 6) {
        bulan1 = "Juni";
    } else if (bulanFix == 7) {
        bulan1 = "Juli";
    } else if (bulanFix == 8) {
        bulan1 = "Agustus";
    } else if (bulanFix == 9) {
        bulan1 = "September";
    } else if (bulanFix == 10) {
        bulan1 = "Oktober";
    } else if (bulanFix == 11) {
        bulan1 = "November";
    } else {
        bulan1 = "Desember";
    }

    model.addAttribute("penduduk", penduduk);
    model.addAttribute("keluarga", keluarga);
    model.addAttribute("kelurahan", kelurahan);
    model.addAttribute("kecamatan", kecamatan);
    model.addAttribute("kota", kota);
    model.addAttribute("tanggal", tanggal1);
    model.addAttribute("bulan", bulan1);
    model.addAttribute("tahun", tahun1);

    return "view-penduduk";

```

- Cara saya membuat fitur tersebut adalah dengan menyocokkan nik yang dimasukkan dengan nik yang dimiliki penduduk yang terdaftar. Kemudian saya akan menyocokkan id keluarga penduduk tersebut ke tabel keluarga untuk memperoleh alamat penduduk. Pada tabel keluarga terdapat id kelurahan yang berfungsi untuk menghubungkan dengan kecamatan, kemudian di tabel kecamatan kita dapat memperoleh informasi kota. Untuk lebih lengkapnya dapat dilihat pada penduduk mapper, service, dan service database.
2. Tampilan Data Keluarga Berdasarkan Nomor KK
- Ketika pencarian sudah ditemukan

Course: [SI.Reg] Arsitekt: X Tugas 1 Individu (Rev.2.0) localhost / Local Database: X View Keluarga dari NKK Buttons - Bootstrap Anisha Eva

localhost:8080/keluarga?nkk=3171010806150001&action=save

SI Kependudukan Home Cari Data

### Lihat Data Keluarga - 3171010806150001

NKK 3171010806150001

Alamat Gg. Ahmad Dahlan No. 518

RT/RW 142/002

Kelurahan/Desa JAGAKARSA

Kecamatan JAGAKARSA

Kota KOTA JAKARTA SELATAN

No.	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status dalam Keluarga	Kewarganegaraan	Status KK
1	Padmi Sakura Farida	3171014305150001	Wanita	Jakarta	2015-05-03	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI	Berlaku
2	Salsabila Olivia Mardiyah S.Ked	3171015812140001	Wanita	Jakarta	2014-12-18	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI	Berlaku
3	Artanto Kusumo S.Sos	3171011509160002	Pria	Jakarta	2016-09-15	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI	Berlaku
	Jamalia				1993-03		TENTARA				

Windows Taskbar: 19:30 22/10/2017

- Pada KeluargaController

```
ml view-keluarga.html view-penduduk.html KeluargaController.java
@RequestMapping("/keluarga")
public String view (Model model,
    @RequestParam(value = "nkk", required = false) String nkk)
{
    KeluargaModel keluarga = keluargaDAO.selectKeluargaByNKK(nkk);
    if (keluarga != null) {
        KelurahanModel kelurahan = kelurahanDAO.selectKelurahanByNKK(nkk);
        KecamatanModel kecamatan = kecamatanDAO.selectKecamatanByNKK(nkk);
        KotaModel kota = kotaDAO.selectKotaByNKK(nkk);

        model.addAttribute("keluarga", keluarga);
        model.addAttribute("kelurahan", kelurahan);
        model.addAttribute("kecamatan", kecamatan);
        model.addAttribute("kota", kota);

        return "view-keluarga";
    } else {
        model.addAttribute("nkk", nkk);
        return "not-found";
    }
}
```

- Cara saya membuat fitur ini adalah dengan menyocokkan nomor kk yang dimasukkan dengan data yang terdaftar. Kemudian saya mencari kelurahan, kecamatan, dan kota yang sesuai dengan nomor kk tersebut dengan cara menyocokkan id kelurahan pada tabel keluarga dengan id pada tabel kelurahan, kemudian id kecamatan pada tabel kelurahan dengan id pada tabel kecamatan, dan id kota pada tabel kecamatan dengan id pada tabel kota. Untuk lebih lengkapnya dapat dilihat pada penduduk mapper, service, dan service database.

### 3. Menambahkan Penduduk

- Form untuk menambahkan penduduk

Course: [SI.Reg] Arsitektur x Tugas 1 Individu (Rev.2.0) x phpMyAdmin x Tambah Penduduk x Buttons - Bootstrap x Anisha Eva

localhost:8080/penduduk/tambah

SI Kependudukan Home Cari Data

## Tambah Penduduk

Submit

Nama  
hehe

Tempat Lahir  
depok

Tanggal Lahir  
07/04/2017

Jenis Kelamin  
Perempuan

Kewarganegaraan  
WNI

ID Keluarga  
2

Agama  
Islam

Pekerjaan  
mahasiswa

Status Perkawinan  
Belum Kawin

Status Dalam Keluarga  
anak

Golongan Darah  
A

Windows taskbar: 20:17 22/10/2017

- Ketika penduduk berhasil ditambahkan

Course: [SI.Reg] Arsitektur x Tugas 1 Individu (Rev.2.0) x phpMyAdmin x Tambah Penduduk x Buttons - Bootstrap x Anisha Eva

localhost:8080/penduduk/tambah/submit

SI Kependudukan Home Cari Data

## Sukses!

Penduduk dengan NIK 3101015005170001 berhasil ditambahkan

Windows taskbar: 20:42 22/10/2017

- Tampilan pada PendudukController

```

dex.html PendudukController.java view-penduduk.html form-tambah-penduduk.h... success-tambah-pe
7 @RequestMapping("/penduduk/tambah/submit")
8 public String tambahPendudukHasil (
9     @RequestParam(value = "nama", required = false) String nama,
10    @RequestParam(value = "tempat_lahir", required = false) String tempat_lahir,
11    @RequestParam(value = "tanggal_lahir", required = false) String tanggal_lahir,
12    @RequestParam(value = "jenis_kelamin", required = false) int jenis_kelamin,
13    @RequestParam(value = "is_wni", required = false) int is_wni,
14    @RequestParam(value = "id_keluarga", required = false) int id_keluarga,
15    @RequestParam(value = "agama", required = false) String agama,
16    @RequestParam(value = "pekerjaan", required = false) String pekerjaan,
17    @RequestParam(value = "status_perkawinan", required = false) String status_perkawinan,
18    @RequestParam(value = "status_dalam_keluarga", required = false) String status_dalam_keluarga,
19    @RequestParam(value = "golongan_darah", required = false) String golongan_darah,
20    Model model)
21 {
22     String kodeKecamatan = pendudukDAO.kodeKecamatan(id_keluarga);
23     String nikDepan = kodeKecamatan.substring(0,6);
24
25     String[] tanggalLahir = tanggal_lahir.split("-");
26     String bulan = tanggalLahir[2];
27     String tanggal = tanggalLahir[1];
28     String tahun = tanggalLahir[0].substring(2, 4);
29
30     if (jenis_kelamin == 1) {
31         int tgl = Integer.parseInt(tanggal);
32         tgl += 40;
33         tanggal = "" + tgl;
34     }
35
36     String tglFix = "" + tanggal + bulan + tahun;
37     nikDepan += tglFix;
38
39     int hitungNIK = pendudukDAO.hitungNIK(nikDepan)+1;
40
41     String urutanNIK = "" + hitungNIK;
42
43     if(hitungNIK/10 == 0) {
44         nikDepan += "000" + urutanNIK;
45     } else if(hitungNIK/100 == 0) {
46         nikDepan += "00" + urutanNIK;
47     } else if (hitungNIK/1000 == 0) {
48         nikDepan += "0" + urutanNIK;
49     }
50
51     PendudukModel penduduk = new PendudukModel (0, nikDepan, nama, tempat_lahir, tanggal_lahir,
52         jenis_kelamin, is_wni, id_keluarga, agama, pekerjaan, status_perkawinan, status_dalam_keluarga,
53         golongan_darah, 0);
54     pendudukDAO.tambahPenduduk (penduduk);
55     model.addAttribute("penduduk", penduduk);
56     return "success-tambah-penduduk";
57 }

```

- Cara adalah dengan menyimpan semua data yang dimasukkan kemudian memisahkan tanggal lahir yang dimasukkan karena akan digunakan untuk menentukan NIK penduduk. Kemudian saya akan menghitung 4 digit terakhir dengan if clause dengan kondisi pembagian. Untuk lebih lengkapnya dapat dilihat pada penduduk mapper, service, dan service database.

#### 4. Menambahkan Keluarga Baru

- Form menambahkan keluarga

Course: [SI.Reg] Arsitekt: x Tugas 1 Individu (Rev.2.0) x localhost / Local Databa: x Tambah Keluarga x Buttons - Bootstrap x Anisha Eva

← → ↻ localhost:8080/keluarga/tambah

SI Kependudukan Home Cari Data

## Tambah Keluarga

Alamat

jl jalan

RT

07

RW

15

Kelurahan/Desa

Pulau Tidung

Kecamatan

Kepulauan Seribu Selatan

Kota

Kabupaten Kepulauan Seri

Submit

Windows taskbar: 20:55 22/10/2017

- Tampilan jika keluarga berhasil ditambahkan

Course: [SI.Reg] Arsitekt: x Tugas 1 Individu (Rev.2.0) x localhost / Local Databa: x Tambah Keluarga x Buttons - Bootstrap x Anisha Eva

← → ↻ localhost:8080/keluarga/tambah/submit

SI Kependudukan Home Cari Data

## Sukses!

Keluarga dengan NKK 3101012258170001 berhasil ditambahkan

Windows taskbar: 20:59 22/10/2017

- Tampilan pada KeluargaController

ludukCon... view-pendudu... form-tambah-... form-tambah-... success-tam..

```
@RequestMapping("keluarga/tambah")
public String tambahKeluarga() {
    return "form-tambah-keluarga";
}

@RequestMapping("/keluarga/tambah/submit")
public String tambahKeluarga (
    @RequestParam(value = "alamat", required = false) String alamat,
    @RequestParam(value = "rt", required = false) String rt,
    @RequestParam(value = "rw", required = false) String rw,
    @RequestParam(value = "kelurahan", required = false) String kelurahan,
    @RequestParam(value = "kecamatan", required = false) String kecamatan,
    Model model)
{
    String nkk = "";
    String kodeKelurahan = keluargaDAO.kodeKelurahan(kelurahan);
    String kodeKecamatan = keluargaDAO.kodeKecamatan(kecamatan);
    String nkkDepan = kodeKecamatan.substring(0,6);

    DateFormat dateFormat = new SimpleDateFormat("dd-mm-yyyy");
    Date hariIni = new Date();
    String nkkSelanjutnya = dateFormat.format(hariIni);

    String[] splitHari = nkkSelanjutnya.split("-");
    String bulan = splitHari[1];
    String tanggal = splitHari[0];
    String tahun = splitHari[2].substring(2, 4);

    nkk = "" + nkkDepan + tanggal + bulan + tahun;

    int hitungNKK = keluargaDAO.hitungNKK(nkk)+1;

    String urutanNKK = "" + hitungNKK;
```

view-pendudu... form-tambah-... form-tambah-... success-tam... success-tam...

```
String urutanNKK = "" + hitungNKK;
if(hitungNKK/10 == 0) {
    nkk += "000" + urutanNKK;
} else if(hitungNKK/100 == 0) {
    nkk += "00" + urutanNKK;
} else if (hitungNKK/1000 == 0) {
    nkk += "0" + urutanNKK;
}

KeluargaModel keluarga = new KeluargaModel (0, nkk, alamat, rt, rw,
    kodeKelurahan, 0, null);
keluargaDAO.tambahKeluarga (keluarga);
model.addAttribute("keluarga", keluarga);
return "success-tambah-keluarga";
}
```

- Cara adalah dengan menyimpan semua data yang dimasukkan kemudian memisahkan tanggal input yang dimasukkan karena akan digunakan untuk menentukan NKK keluarga. Kemudian saya akan menghitung 4 digit terakhir dengan if clause dengan kondisi pembagian. Saya akan mengambil kode kecamatan untuk 4 digit pertama NKK karena kode kecamatan sudah mencakup kelurahan, kecamatan, dan kota. Untuk lebih lengkapnya dapat dilihat pada penduduk mapper, service, dan service database.



## 5. Mengubah Data Penduduk

- Form tampilan untuk mengubah data penduduk

Course: [SI.Reg] Arsitekt: x Tugas 1 Individu (Rev.2.0) x localhost / Local Database: x Update Penduduk x Buttons - Bootstrap x Anisha Eva

localhost:8080/penduduk/ubah/3101015013170001

SI Kependudukan Home Cari Data

### Update Penduduk

NIK  
3101015013170001

Nama  
coba

Tempat Lahir  
bismillah

Tanggal Lahir  
10/01/2017

Jenis Kelamin  
Laki-laki

Kewarganegaraan  
WNI

ID Keluarga  
12

Agama  
Hindu

Pekerjaan  
Pegawai

Status Perkawinan  
Kawin

Status Dalam Keluarga  
Ibu

Golongan Darah  
AB

Status Kematian  
Hidup

Submit

- Hasil setelah berhasil mengubah data penduduk

Course: [SI.Reg] Arsitekt: x Tugas 1 Individu (Rev.2.0) x localhost / Local Database: x Update Penduduk x Buttons - Bootstrap x Anisha Eva

localhost:8080/penduduk/ubah/3101015013170001

SI Kependudukan Home Cari Data

**Sukses!**

Penduduk dengan NIK 3101015013170001 berhasil diubah

### Update Penduduk

NIK  
3101015013170001

Nama  
coba

Tempat Lahir  
bismillah

Tanggal Lahir  
10/01/2017

Jenis Kelamin  
Laki-laki

Kewarganegaraan  
WNI

ID Keluarga  
12

Agama  
Hindu

Pekerjaan  
Pegawai

Status Perkawinan  
Kawin

Status Dalam Keluarga  
Ibu

Golongan Darah  
AB

Status Kematian  
Hidup

Submit

- PendudukController untuk mengubah data penduduk

```
@RequestMapping("/penduduk/ubah/{nik}")
public String updatePendudukIndex (Model model, @PathVariable(value = "nik") String nik)
{
    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
    if (penduduk != null) {
        model.addAttribute("penduduk", penduduk);
        return "form-update-penduduk";
    } else {
        return "not-found";
    }
}

@RequestMapping(value = "/penduduk/ubah/{nik}", method = RequestMethod.POST)
public String updatePenduduk (Model model, PendudukModel penduduk)
{
    boolean flagSukses = true;

    String kodeKecamatan = pendudukDAO.kodeKecamatan(penduduk.getId_keluarga());
    String nikDepan = kodeKecamatan.substring(0,6);

    String[] tanggalLahir = penduduk.getTanggal_lahir().split("-");
    String bulan = tanggalLahir[2];
    String tanggal = tanggalLahir[1];
    String tahun = tanggalLahir[0].substring(2, 4);

    if (penduduk.getJenis_kelamin() == 1) {
        int tgl = Integer.parseInt(tanggal);
        tgl += 40;
        tanggal = "" + tgl;
    }
}
```

```
PendudukController.java KeluargaController.java form-update-penduduk.html

String tglFix = "" + tanggal + bulan + tahun;
nikDepan += tglFix;

int hitungNIK = pendudukDAO.hitungNIK("%"+nikDepan+"")+1;

String urutanNIK = "" + hitungNIK;

if(hitungNIK/10 == 0) {
    nikDepan += "000" + urutanNIK;
} else if(hitungNIK/100 == 0) {
    nikDepan += "00" + urutanNIK;
} else if (hitungNIK/1000 == 0) {
    nikDepan += "0" + urutanNIK;
}

PendudukModel pendudukUpdate = new PendudukModel (penduduk.getId(), nikDepan, penduduk.getNama(), penduduk.getTempat_lahir(),
penduduk.getJenis_kelamin(), penduduk.getIs_wni(), penduduk.getId_keluarga(), penduduk.getAgama(), penduduk.getPekerjaan,
penduduk.getStatus_perkawinan(), penduduk.getStatus_dalam_keluarga(), penduduk.getGolongan_darah(), penduduk.getIs_wafat);

model.addAttribute ("penduduk", pendudukUpdate);
model.addAttribute("flagSukses", flagSukses);
model.addAttribute("nikLama", penduduk.getNik());
pendudukDAO.updatePenduduk(pendudukUpdate);

return "form-update-penduduk";
```

- Cara saya adalah dengan memberikan flag untuk membedakan RequestMapping sehingga ketika data berhasil dimasukkan akan muncul tampilan sukses. Kemudian saya akan menyimpan kembali tanggal ulang tahun yang diubah kemudian semua input akan diproses seperti tambah penduduk apabila ada perubahan. Namun untuk NIK saya hanya menampilkan saja namun tidak bisa diubah. Ketika dihalaman sukses , dibawah

kalimat sukses adalah NIK lama dan dalam keterangan dibawahnya adalah NIK yang telah diupdate. Untuk lebih lengkapnya dapat dilihat pada penduduk mapper, service, dan service database.

## 6. Mengubah Data Keluarga

- Tampilan form untuk mengubah data keluarga

--Data awal

Course: [SI.Reg] Arsitekt... x Tugas 1 Individu (Rev.2.0) x localhost / Local Databa... x Update Keluarga x Buttons - Bootstrap x Anisha Eva

localhost:8080/keluarga/ubah/3101012234170001

SI Kependudukan Home Cari Data

### Update Keluarga

Nomor KK  
3101012234170001

Alamat  
Jl sini' situ

RT  
22

RW  
24

Kelurahan/Desa  
PULAU TIDUNG

Kecamatan  
KEPULAUAN SERIBU SEL

Kota  
KABUPATEN KEPULAUAN

Submit

Windows taskbar: 21:35 22/10/2017

--Data yang sudah diupdate

Course: [SI.Reg] Arsitekt... x Tugas 1 Individu (Rev.2.0) x localhost / Local Databa... x Update Keluarga x Buttons - Bootstrap x Anisha Eva

localhost:8080/keluarga/ubah/3101012234170001

SI Kependudukan Home Cari Data

### Update Keluarga

Nomor KK  
3101012234170001

Alamat  
Jl jalan lagi

RT  
12

RW  
08

Kelurahan/Desa  
CIGANJUR

Kecamatan  
JAGAKARSA

Kota  
KOTA JAKARTA SELATAN

Submit

Windows taskbar: 21:37 22/10/2017

- Tampilan sukses melakukan perubahan

Course: [SI.Reg] Arsitekt... X Tugas 1 Individu (Rev.2.0) X localhost / Local Database X Update Keluarga X Buttons - Bootstrap X Anisha Eva

localhost:8080/keluarga/ubah/3171012246170001

SI Kependudukan Home Cari Data

## Sukses!

Keluarga dengan NKK 3171012246170001 berhasil diubah

## Update Keluarga

Nomor KK

Alamat

RT

RW

Kelurahan/Desa

Kecamatan

Kota

Submit

21:46 22/10/2017

- KeluargaController untuk update keluarga

```

PendudukController.java KeluargaController.java form-update-penduduk.html form-update-keluarga.html
@RequestMapping("/keluarga/ubah/{nomor_kk}")
public String updateKeluarga (Model model, @PathVariable(value = "nomor_kk") String nomor_kk)
{
    KeluargaModel keluarga = keluargaDAO.selectKeluargaByNKK(nomor_kk);
    if (keluarga != null) {
        model.addAttribute("keluarga", keluarga);
        model.addAttribute("kelurahan", kelurahanDAO.selectNamaKelurahanById(keluarga.getId_kelurahan()));
        model.addAttribute("kecamatan", kecamatanDAO.selectNamaKecamatanByIdKelurahan(keluarga.getId_kelurahan()));
        model.addAttribute("kota", kotaDAO.selectNamaKotaByIdKelurahan(keluarga.getId_kelurahan()));
        return "form-update-keluarga";
    } else {
        return "not-found";
    }
}

@RequestMapping(value="/keluarga/ubah/{nomor_kk}", method = RequestMethod.POST)
public String updateKeluarga (Model model, KeluargaModel keluarga,
    @RequestParam(value = "kelurahan", required = false) String kelurahan,
    @RequestParam(value = "kecamatan", required = false) String kecamatan,
    @RequestParam(value = "kota", required = false) String kota)
{
    boolean flagSukses = true;

    String kodeKelurahan = keluargaDAO.kodeKelurahan(kelurahan);
    String kodeKecamatan = keluargaDAO.kodeKecamatan(kecamatan);
    //String kodeKota = keluargaDAO.kodeKecamatan(kota);
    String nkkSementara = kodeKecamatan.substring(0,6);
    String nkk = nkkSementara;

    DateFormat dateFormat = new SimpleDateFormat("dd-mm-yyyy");
    Date hariIni = new Date();

```

```

PendudukController.java KeluargaController.java form-update-penduduk.html form-update-keluarga.html
DateFormat dateFormat = new SimpleDateFormat("dd-mm-yyyy");
Date hariIni = new Date();
String nkkSelanjutnya = dateFormat.format(hariIni);

String[] splitHari = nkkSelanjutnya.split("-");
String bulan = splitHari[1];
String tanggal = splitHari[0];
String tahun = splitHari[2].substring(2, 4);

nkkSementara += "" + tanggal + bulan + tahun;

int hitungNKK = keluargaDAO.hitungNKK(""+nkkSementara+"")+1;

String urutanNKK = "" + hitungNKK;
if(hitungNKK/10 == 0) {
    nkkSementara += "000" + urutanNKK;
} else if(hitungNKK/100 == 0) {
    nkkSementara += "00" + urutanNKK;
} else if (hitungNKK/1000 == 0) {
    nkkSementara += "0" + urutanNKK;
}

keluarga.setNomor_kk(nkkSementara);
keluarga.setId_kelurahan(kodeKelurahan);

List<PendudukModel> anggotaKeluarga = keluargaDAO.selectAnggotaKeluargaByIdKeluarga(keluarga.getId());

for(int i = 0; i < anggotaKeluarga.size(); i++) {
    PendudukModel tempPenduduk = anggotaKeluarga.get(i);
    String nikBaru = nkk + tempPenduduk.getNik().substring(6,12);
    int hitungNIK = pendudukDAO.hitungNIK(nikBaru) + 1;

    String urutanNIK = "" + hitungNIK;
    if(hitungNIK/10 == 0) {
        nkkSementara += "000" + urutanNIK;
    } else if(hitungNIK/100 == 0) {
        nkkSementara += "00" + urutanNIK;
    } else if (hitungNIK/1000 == 0) {
        nkkSementara += "0" + urutanNIK;
    }

    tempPenduduk.setNik(nikBaru);
    pendudukDAO.updatePenduduk(tempPenduduk);
}

KeluargaModel keluargaUpdate = new KeluargaModel (keluarga.getId(), keluarga.getNomor_kk(), keluarga.getAlamat(), keluar
    kodeKelurahan, 0, null);

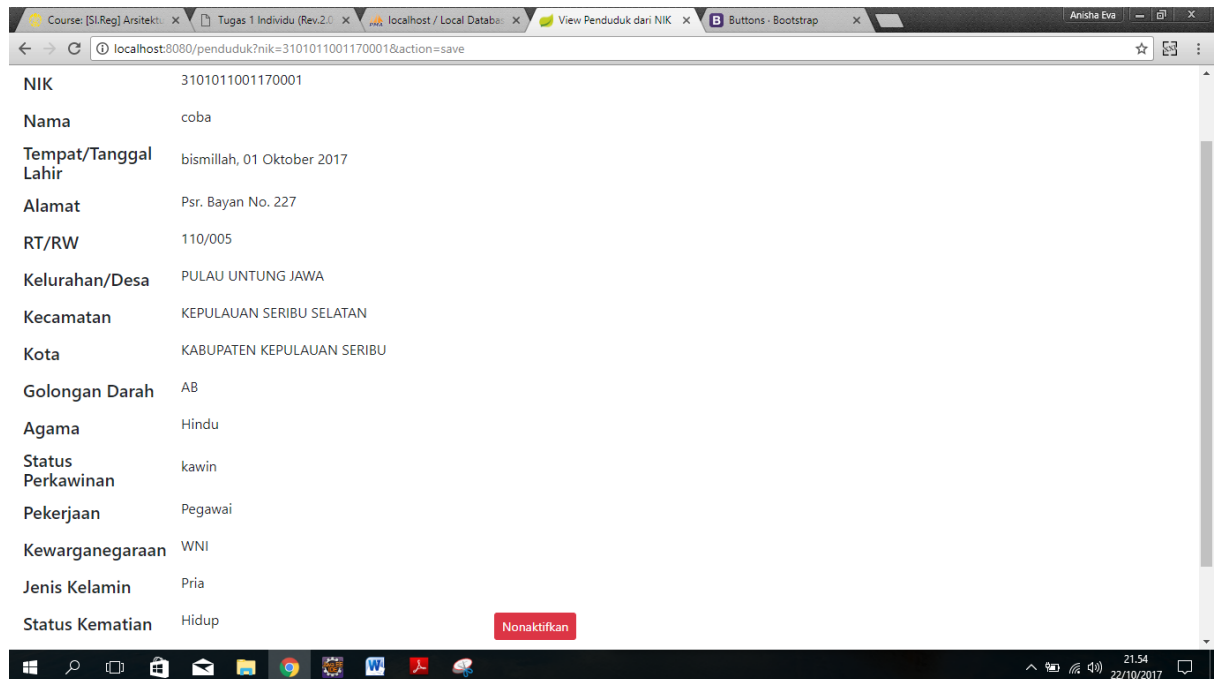
model.addAttribute("keluarga", keluargaUpdate);
model.addAttribute("flagSukses", flagSukses);
model.addAttribute("nomorKkLama", keluarga.getNomor_kk());
keluargaDAO.updateKeluarga(keluargaUpdate);
return "form-update-keluarga";
}

```

- Cara saya adalah dengan memberikan flag untuk membedakan RequestMapping sehingga ketika data berhasil dimasukkan akan muncul tampilan sukses. Kemudian saya akan menyimpan kembali tanggal data dimasukkan yang diubah kemudian semua input akan diproses seperti tambah keluarga apabila ada perubahan. Namun untuk NKK saya hanya menampilkan saja namun tidak bisa diubah. Ketika dihalaman sukses, dibawah kalimat sukses adalah NKK lama dan dalam keterangan dibawahnya adalah NKK yang telah diupdate. Untuk lebih lengkapnya dapat dilihat pada penduduk mapper, service, dan service database.

## 7. Mengubah Status Kematian Penduduk

- Tampilan untuk mengubah status kematian penduduk



Course: [SI.Reg] Arsitektur x Tugas 1 Individu (Rev.2.0) x localhost / Local Database x View Penduduk dari NIK x Buttons - Bootstrap x Anisha Eva

localhost:8080/penduduk?nik=3101011001170001&action=save

NIK	3101011001170001
Nama	coba
Tempat/Tanggal Lahir	bismillah, 01 Oktober 2017
Alamat	Psr. Bayan No. 227
RT/RW	110/005
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	AB
Agama	Hindu
Status Perkawinan	kawin
Pekerjaan	Pegawai
Kewarganegaraan	WNI
Jenis Kelamin	Pria
Status Kematian	Hidup

Nonaktifkan

21:54 22/10/2017

- Hasil jika telah berhasil mengubah status



- PendudukController untuk mengubah status kematian

```
@RequestMapping(value="/penduduk", method = RequestMethod.POST)
public String statusKematianPenduduk(@RequestParam(value = "nik", required = false) String nik,
    @RequestParam(value = "id_keluarga", required = false) String id_keluarga,
    Model model) {
    boolean statusKematian = false;
    pendudukDAO.statusKematianPenduduk(nik);

    int statusAnggotaKeluarga = pendudukDAO.statusKeluarga(id_keluarga);
    if (statusAnggotaKeluarga < 1) {
        keluargaDAO.statusKK(id_keluarga);
    }

    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
    KeluargaModel keluarga = keluargaDAO.selectKeluarga(nik);
    KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(nik);
    KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(nik);
    KotaModel kota = kotaDAO.selectKota(nik);

    model.addAttribute("penduduk", penduduk);
    model.addAttribute("keluarga", keluarga);
    model.addAttribute("kelurahan", kelurahan);
    model.addAttribute("kecamatan", kecamatan);
    model.addAttribute("kota", kota);

    model.addAttribute("statusKematian", true);
    model.addAttribute("nik", nik);

    return "success-status-kematian";
}
```

- Fitur ini dapat diakses pada halaman view penduduk dengan nik yang telah dimasukkan. Kemudian saya membuat flag untuk membedakan RequestParameter. Kemudian saya mengecek terlebih dahulu apakah seluruh keluarga telah meninggal atau tidak, jika sudah maka status KK akan diubah juga. Baru kemudian saya menjalankan fungsi untuk menonaktifkan status kematian penduduk. Untuk lebih lengkapnya dapat dilihat pada penduduk mapper, service, dan service database.

## 8. Tampilan Data Penduduk Berdasarkan Kota, Kecamatan, dan Kelurahan

- PendudukController untuk menampilkan fitur ini

```
3     }
4
5     @RequestMapping("/penduduk/cari-kelurahan")
6     public String pendudukSatuKelurahan(@RequestParam(value = "kota", required = false) String kota,
7         @RequestParam(value = "kec", required = false) String kec,
8         @RequestParam(value = "kel", required = false) String kel,
9         Model model) {
10
11         boolean sudah = false;
12         if (kel != null) {
13             model.addAttribute("nama_kota", kotaDAO.namaKotaById(Integer.parseInt(kota)));
14             model.addAttribute("nama_kecamatan", kecamatanDAO.namaKecamatanById(Integer.parseInt(kec));
15             model.addAttribute("nama_kelurahan", kelurahanDAO.namaKelurahanById(Integer.parseInt(kel));
16             model.addAttribute("kl", kl);
17
18             model.addAttribute("sudah", true);
19             List<PendudukModel> daftarPenduduk = pendudukDAO.daftarPendudukSatuKelurahan(Integer.parseInt(kel));
20             model.addAttribute("daftarPenduduk", daftarPenduduk);
21         }
22     }
```

- Fitur ini belum selesai. Ide saya adalah dengan menyimpan nama dari kelurahan, kota, dan kecamatan yang telah dipilih kemudian saya akan menyocokkannya dengan idnya sehingga dapat memperoleh id dan menyocokkannya dengan tabel lain serta mendapatkan daftar keluarga dengan alamat yang sesuai.

#### BONUS

9. Pada bonus ini, saya melakukan validasi jika input kosong dan jika tanggal tidak sesuai saya menggunakan DateFormatter sehingga semua data yang dimasukkan akan berubah sesuai dengan DateFormatter yang telah saya tetapkan.
12. - Optimasi Database saya lakukan dengan membuat primary key di setiap table, seperti nik sebagai primary key di tabel penduduk, id sebagai primary key di tabel keluarga, id sebagai primary key di tabel kelurahan, id sebagai primary key di tabel kecamatan, dan id sebagai primary key di tabel kota. Saya juga membuat index untuk data yang sering digunakan terkait dengan tabel lain, seperti id keluarga di tabel penduduk untuk hubungan dengan tabel keluarga, id kelurahan di tabel keluarga untuk hubungan di tabel kelurahan dan sejenisnya.