

Fitur 1 : Cari Penduduk

Jadi untuk fitur ini kodingan yang saya gunakan adalah

Controller

```
@RequestMapping("/penduduk")
```

```
public String findPenduduk (@RequestParam(value = "nik", required=true) String nik, Model model)
{
```

Pada tahap ini melakukan pembuatan objek baru dengan menggunakan selectPenduduk pada Mapper, yang parameternya adalah nik. Jadi nantinya akan mengembalikan Objek PendudukModel

```
PendudukModel a = sidukDAO.selectPenduduk(nik);
```

Selanjutnya karena yang akan dicetak bulan lahirnya dalam bentuk nama, maka akan dibuat condition untuk menyesuaikan bula lahir dengan namanya yaitu dengan cara di split. Setelah itu akan disatukan lagi, tetapi dibalik dengan format dd-mm-yyyy

```
String bulan = "";
```

```
String[] ttl = a.tanggal_lahir.split("-");
```

```
if(ttl[1].equals("01")) {
```

```
    bulan = "Januari";
```

```
}else if(ttl[1].equals("02")) {
```

```
    bulan = "Februari";
```

```
}else if(ttl[1].equals("03")) {
```

```
    bulan = "Maret";
```

```
}else if(ttl[1].equals("04")) {
```

```
    bulan = "April";
```

```
}else if(ttl[1].equals("05")) {
```

```
    bulan = "Mei";
```

```
}else if(ttl[1].equals("06")) {
```

```
    bulan = "Juni";
```

```
}else if(ttl[1].equals("07")) {
```

```
    bulan = "Juli";
```

```

}else if(ttl[1].equals("08")) {
    bulan = "Agustus";
}else if(ttl[1].equals("09")) {
    bulan = "September";
}else if(ttl[1].equals("10")) {
    bulan = "Oktober";
}else if(ttl[1].equals("11")) {
    bulan = "November";
}else if(ttl[1].equals("12")) {
    bulan = "Desember";
}

a.tanggal_lahir = ttl[2] + " " + bulan + " " + ttl[0];

```

Setelah itu tinggal diadd lalu return page penduduk

```

model.addAttribute("penduduk", a);
return "penduduk";
}

```

Mapper

Jadi pada model ada beberapa variabel baru salah satunya adalah keluarga. Keluarga ini berfungsi untuk mengambil model keluarga, sehingga penduduk dapat dipastikan anggota dari keluarga yang mana dan dapat mengambil properti dari keluarga seperti alamat. Karena nanti pada view akan diperlukan alamat, kelurahan, kecamatan dan kota. Method selectPenduduk ini menyambung dengan yang ada di Mapper lain yaitu selectKeluarga, selectKelurahan, selectKecamatan dan selectKota yang bisa dilihat di bawah

```

@Select("select * from penduduk where nik = #{nik}")
@Results(value =
{
    @Result(property="id_keluarga", column="id_keluarga"),
    @Result(property="keluarga", column="id_keluarga",
        javaType=KeluargaModel.class,
        many=@Many(select="selectKeluarga"))
}
)

```

```
}}
```

```
PendudukModel selectPenduduk (@Param("nik") String nik);
```

```
@Select("select * from kecamatan where id = #{id}")
```

```
    @Results(value =
```

```
        {@Result(property="kota", column="id_kota",
```

```
            javaType=KecamatanModel.class,
```

```
            many=@Many(select="selectKota"))
```

```
    })
```

```
    KecamatanModel selectKecamatan (@Param("id") int id);
```

```
@Select("select * from keluarga where id = #{id}")
```

```
    @Results(value =
```

```
        {@Result(property="id_kelurahan", column="id_kelurahan"),
```

```
        @Result(property="lurah", column="id_kelurahan",
```

```
            javaType=KelurahanModel.class,
```

```
            many=@Many(select="selectKelurahan"))
```

```
    })
```

```
    KeluargaModel selectKeluarga (@Param("id") int id);
```

```
@Select("select * from kelurahan where id = #{id}")
```

```
    @Results(value =
```

```
        {@Result(property="camat", column="id_kecamatan",
```

```
            javaType=KecamatanModel.class,
```

```
            many=@Many(select="selectKecamatan"))
```

```
    })
```

```
    KelurahanModel selectKelurahan (@Param("id") int id);
```

```
@Select("select * from kota where id = #{id}")
```

```
KotaModel selectKota (@Param("id") int id);
```

Untuk tahapan dalam bentuk gambar bisa dilihat di screenshot dibawah ini

The first screenshot shows the 'SI Kependudukan' homepage. It has a navigation bar with links: Home, Cari Data, Tambah Penduduk, Tambah Keluarga, and Panduan. Below the navigation bar is a search bar with a 'Search' button. The main content area has a large heading 'Selamat Datang di Situs Kependudukan DKI!'. Below this are two sections: 'Cari Penduduk Berdasarkan NIK' and 'Cari Keluarga Berdasarkan NKK'. Each section has a text input field for the respective ID and a 'Cari' button.

The second screenshot shows the 'Penduduk' page. It has the same navigation bar and search bar. Below the search bar is a table with the following data:

NIK	3101010101800001
Nama	Laksana Saragih
Tempat/Tanggal Lahir	Jakarta, 01 Januari 1980
Alamat	Gg. Acordion No. 690
RT/RW	177/070
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	AB-
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	KONSTRUKSI
Kewarganegaraan	WNI
Status Kematian	Hidup

Below the table are two buttons: 'Update Data' and 'Nonaktifkan'.

Fitur 2: Cari Keluarga

Jadi untuk fitur ini kodingan yang saya gunakan adalah

Controller

```
@RequestMapping("/keluarga")
```

```
public String findKeluarga (@RequestParam(value = "nkk", required=true) String nkk, Model model)
{
```

Untuk fitur ini, pertama akan mencari keluarga dulu berdasarkan nomor kk di database, setelah itu akan di add ke attribute untuk digunakan

```
KeluargaModel a = sidukDAO.cariKeluarga(nkk);

model.addAttribute("keluarga", a);

return "keluarga";
}
```

Mapper

Jadi untuk fitur ini, modelnya ditambahkan atribut yaitu List yang berisi anggota keluarga. Karena nantinya anggota keluarga akan dicetak di halaman keluarga. Mencari anggota dilakukan dengan method cariAnggota dan ada pula atribut baru yaitu lurah, bentuknya KelurahanModel. Karena nanti akan dibutuhkan maka dari itu disimpan. Method untuk selectKelurahan sudah ada di fitur sebelumnya.

```
@Select("select * from keluarga where nomor_kk = #{nomor_kk}")
@Results(value =
{
    @Result(property="id_kelurahan", column="id_kelurahan"),
    @Result(property="id", column="id"),
    @Result(property="lurah", column="id_kelurahan",
        javaType=KelurahanModel.class,
        many=@Many(select="selectKelurahan")),
    @Result(property="anggota", column="id",
        javaType=List.class,
        many=@Many(select="cariAnggota"))
})
KeluargaModel cariKeluarga (@Param("nomor_kk") String nomor_kk);
```

```
@Select("select * from penduduk where id_keluarga = #{id}")
List<PendudukModel> cariAnggota (@Param("id") int id);
```

HTML

Main point dari html pada fitur ini adalah tabel anggota keluarga. Karena anggota keluarga harus didapatkan dari kelas lain. Maka dari itu di tabel akan dilakukan iterasi pada List anggota keluarga yang sebelumnya sudah terisi, lalu akan dicetak dan dibentuk jadi dataTables

<div>

<table id="myTable" class="display">

<thead>

<tr>

<th>No</th>

<th>Nama Lengkap</th>

<th>NKK</th>

<th>Jenis Kelamin</th>

<th>Tempat Lahir</th>

<th>Tanggal Lahir</th>

<th>Agama</th>

<th>Pekerjaan</th>

<th>Status Perkawinan</th>

<th>Status dalam Keluarga</th>

<th>Kewarganegaraan</th>

</tr>

</thead>

<tbody>

<tr th:each="anggota,iterationStatus: \${keluarga.anggota}">

<th th:text="\${iterationStatus.count}">No. 1</th>

<th th:text="\${anggota.nama}">Student NPM</th>

<th th:text="\${anggota.nik}">Student Name</th>

<th th:text="\${anggota.jenis_kelamin}==0? 'Laki-laki' : 'Perempuan'">Student NPM</th>

<th th:text="\${anggota.tempat_lahir}">Student Name</th>

<th th:text="\${anggota.tanggal_lahir}">Student Name</th>

<th th:text="\${anggota.agama}">Student Name</th>

```

<th th:text="{ anggota.pekerjaan}">Student Name</th>
<th th:text="{ anggota.status_perkawinan}">Student Name</th>
<th th:text="{ anggota.status_dalam_keluarga}">Student Name</th>
<th th:text="{ anggota.is_wni == 1}? 'WNI' : 'WNA'"></th>

```

```

</tr>

```

```

</tbody>

```

```

</table>

```

```

<hr/>

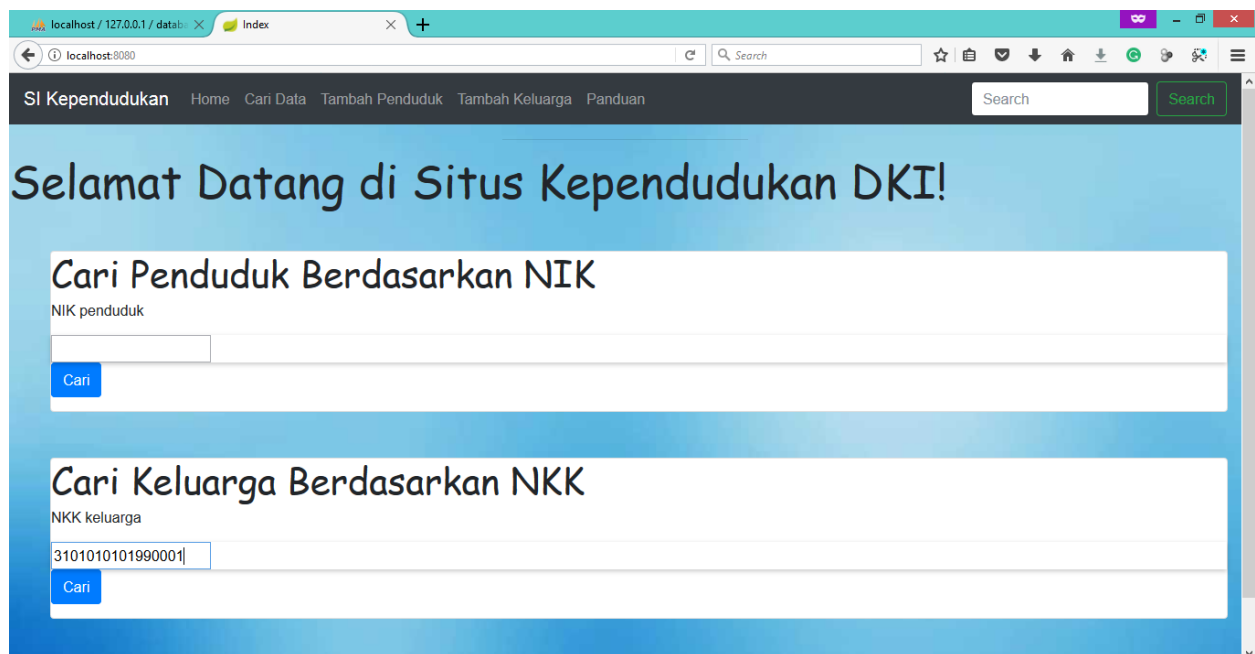
```

```

</div>

```

Step fitur ini bisa dilihat di screenshot dibawah ini



No	Nama Lengkap	NKK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status dalam Keluarga	Kewarganegaraan
1	Mutia Intan Wahyuni	3101015608150001	Perempuan	Solok	2015-08-16	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI
2	Ani Shania Permata S.E.I	3101015602950001	Perempuan	Jakarta	1995-02-16	Islam	KARYAWAN HONORER	Kawin	Istri	WNI
3	Banawa Pradana	3101012104800002	Laki-laki	Mojokerto	1980-04-21	Islam	KONSTRUKSI	Kawin	Kepala Keluarga	WNI

Fitur 3 : Tambah Penduduk

Untuk fitur ini, kodingan yang saya gunakan adalah

Controller

Pada fitur 3, mapping di kontrollernya dibagi 2, GetMapping dan PostMapping. Tujuannya agar mappingnya bisa sama tetapi hasil halamannya beda. Jadi pertama di GetMapping ini kita buat dulu PendudukModel yang kosong untuk digunakan di form.

@GetMapping("/penduduk/tambah")

```
public String tambahPenduduk(Model model)
{
    model.addAttribute("penduduk", new PendudukModel());
    return "tambahpenduduk";
}
```

Pada PostMapping, kita akan melakukan segala fungsi setelah form di post. Pertama untuk validasi keluarga, dicek dulu dengan method selectKeluarga yang sudah ada sebelumnya. Jjika null maka akan dikembalikan ke halaman error yang akan dibahas selanjutnya. Setelah itu jika keluarganya ada, akan diproses untuk generate nik.

@PostMapping(value= "/penduduk/tambah")

```
public String tambahSubmit (@ModelAttribute PendudukModel penduduk, Model model)
```



```
{
```

```
KeluargaModel keluarga = sidukDAO.selectKeluarga(penduduk.id_keluarga);
```

```
if(keluarga!=null) {
```

```
    penduduk.is_wafat = 0;
```

```
    String kecamatan = keluarga.lurah.camat.kode_kecamatan.substring(0, 6);
```

```
    String[] lahir = penduduk.tanggal_lahir.split("-");
```

```
    String bulan = lahir[1];
```

```
    String tahun = lahir[0].substring(2, 4);
```

```
    String tanggal = lahir[2];
```

```
    int index= 0;
```

Seperti yang diketahui, jika jenis kelamin perempuan maka tanggal lahir ditambah 40. NIK awalnya dibagi 3 bagian: awal, tengah, akhir. Awal untuk kode propinsi, kota dan kecamatan, tengah untuk tanggal lahir dan akhir untuk index.

```
    if(penduduk.jenis_kelamin==1) {
```

```
        int masuk = Integer.parseInt(tanggal);
```

```
        masuk+=40;
```

```
        tanggal = masuk+"";
```

```
    }
```

Setelah itu akan dicek apakah nik serupa sudah ada dengan cekPenduduk. Hal ini untuk menentukan index akhir dari nik. Jika sudah ada maka akan ditambah 1 dari index terbesar. Jika belum maka mulai dari nol.

```
    String tengah = tanggal + bulan + tahun;
```

```
penduduk.nikcheck = kecamatan + tengah;
```

```
PendudukModel a = sidukDAO.cekPenduduk(penduduk.nikcheck);
```

```
if(a!=null) {  
    index += Integer.parseInt(a.nik.substring(12, 16));  
    index++;  
}else {  
    index = 0;  
}
```

Setelah itu set NIK dan gunakan method addPenduduk. AddAttribute model, setelah itu akan muncul tampilan

```
String akhir = String.format("%04d", index);  
penduduk.nik = kecamatan + tengah + akhir;  
sidukDAO.addPenduduk(penduduk);  
model.addAttribute("penduduk", penduduk);  
  
}  
  
return "success";  
  
}
```

Mapper

Ini adalah mapper untuk insert. Jadi akan diinsert ke database sesuai fieldnya

```
@Insert("INSERT INTO penduduk (nama, tempat_lahir, tanggal_lahir, "  
        + "jenis_kelamin, is_wni, is_wafat, golongan_darah, status_perkawinan, "
```

```

+ "status_dalam_keluarga, agama, pekerjaan, id_keluarga, nik) VALUES "
+ "({nama}, #{tempat_lahir}, #{tanggal_lahir}, #{jenis_kelamin}, #{is_wni},
#{is_wafat}, #{golongan_darah}, "
+ "#{status_perkawinan}, #{status_dalam_keluarga}, #{agama}, "
+ "#{pekerjaan}, #{id_keluarga}, #{nik}))"

void addPenduduk (PendudukModel penduduk);

```

Ini method untuk mengecek nik. Jadi kita mencari yang 12 digit depannya sama, karena 4 digit belakangnya index. Lalu kita order sesuai id dari terbesar sampai terkecil, lalu limit 1 untuk dapat yang terakhir masuk

```

@Select("select * from penduduk where nik like (concat("#{nikcheck}','')) ORDER BY id DESC
LIMIT 1")

```

```

PendudukModel cekPenduduk (@Param("nikcheck") String nikcheck);

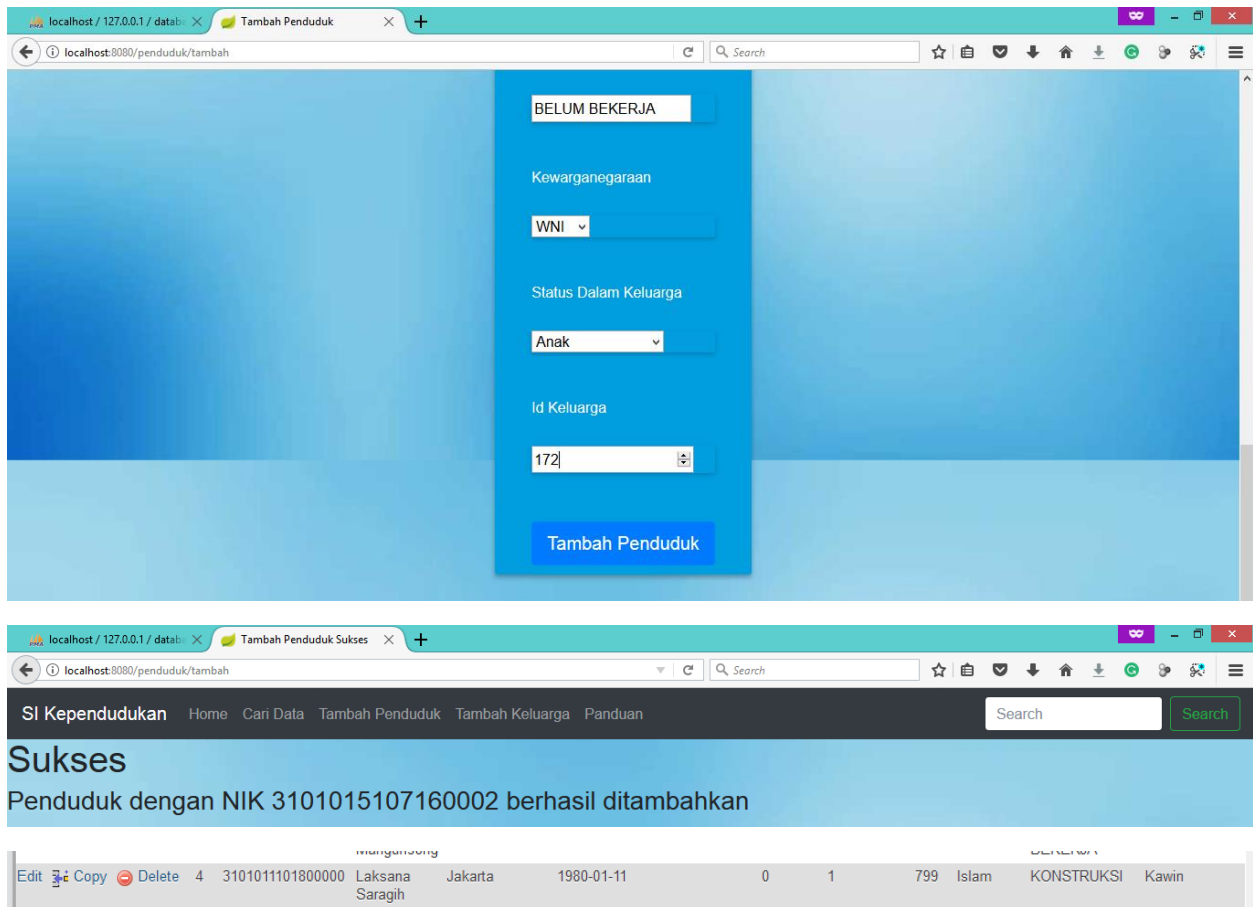
```

HTML

Ada di penduduk.html. Pada page ini, untuk tanggal divalidasi dengan input menggunakan kalender. Lalu number untuk id keluarga dan menggunakan dropdown untuk yang merupakan pilihan seperti golongan darah dan jenis kelamin. Semua disesuaikan dengan isian resmi KK yang berlaku di DKI. Lalu untuk field kosong akan di arahkan ke halaman error

Untuk step dapat dilihat dibawah ini

The screenshot shows a web application for managing residents. The interface includes a sidebar with navigation options and a main form for adding a new resident. The form fields are labeled 'Masukkan nama', 'Jenis Kelamin', 'Tempat Lahir', and 'Tanggal Lahir'. The 'Jenis Kelamin' field is a dropdown menu currently showing 'Laki-laki'. The 'Tempat Lahir' and 'Tanggal Lahir' fields are text inputs. A 'Search' button is located in the top right corner of the page.



Fitur 4 : Tambah Keluarga

Kodingan yang saya gunakan ialah

Controller

Fitur ini mirip dengan fitur 3, ada 2 mapping, dengan get dan post. Untuk GetMapping akan menggunakan list kelurahan untuk dropdown. Jadi kelurahan dipilih sesuai database, otomatis akan tersambung ke kecamatan dan kota. Hal ini dalam upaya validasi agar kelurahan kecamatan dan kota sesuai database. Setelah itu di addAttribute

@GetMapping("/keluarga/tambah")

```
public String tambahKeluarga(Model model)
{
    List<KelurahanModel> lurah = sidukDAO.selectAllKelurahan();
    model.addAttribute("kelurahan", lurah);
    model.addAttribute("keluarga", new KeluargaModel());
    return "tambahkeluarga";
}
```

```
}
```

```
@PostMapping(value= "/keluarga/tambah")
```

```
public String tambahKeluargaSubmit (@ModelAttribute KeluargaModel keluarga, Model model)
```

```
{
```

Karena bagian tengah nomor KK menggunakan tanggal dibuat, maka akan diambil tanggal sekarang menggunakan LocalDate. Setelah itu mirip dengan penduduk, akan di cek apakah sudah ada nomor kk serupa. Jika sudah ada, maka akan diambil index terbesar lalu ditambahkan dengan 1. Dilakukan insert ke database lalu model.addAttribute

```
DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd-MM-yyyy");
```

```
LocalDate ld = LocalDate.now();
```

```
String[] today = dtf.format(ld).split("-");
```

```
KelurahanModel lrh = sidukDAO.selectKelurahan(keluarga.id_kelurahan);
```

```
String awal = lrh.camat.kode_kecamatan.substring(0, 6);
```

```
String tengah = today[0] + today[1] + today[2].substring(2, 4);
```

```
keluarga.is_tidak_berlaku = 0;
```

```
keluarga.nkkcheck = awal + tengah;
```

```
int b1= Integer.parseInt(keluarga.RT);
```

```
int b= Integer.parseInt(keluarga.RW);
```

```
keluarga.RT = String.format("%03d", b1);
```

```
keluarga.RW = String.format("%03d", b);
```

```
KeluargaModel a = sidukDAO.cekNKK(keluarga.nkkcheck);
```

```
int index = 0;
```

```

if(a!=null) {
    index += Integer.parseInt(a.nomor_kk.substring(12, 16));
    index++;
}else {
    index = 0;
}

String akhir = String.format("%04d", index);
keluarga.nomor_kk = awal + tengah + akhir;

sidukDAO.addKeluarga(keluarga);
model.addAttribute("keluarga",keluarga);

return "success-keluarga";

}

```

Mapper

Jadi untuk mappernya ada addKeluarga untuk insert, cekNKK untuk mencari NKK serupa yang mirip seperti menyamakan NIK di penduduk dan select semua klurahan untuk populate dropdown

```

@Insert("INSERT INTO keluarga (nomor_kk, alamat, RT, "
        + "RW, id_kelurahan, is_tidak_berlaku) VALUES "
        + "({nomor_kk}, {alamat}, {RT}, {RW}, {id_kelurahan}, "
        + "{is_tidak_berlaku})")

```

```

void addKeluarga (KeluargaModel keluarga);

```

```

@Select("select * from keluarga where nomor_kk like (concat({nkkcheck}, '%')) ORDER BY id
DESC LIMIT 1")

```

```

KeluargaModel cekNKK (@Param("nkkcheck") String nkkcheck);

```

```

@Select("select * from kelurahan where id_kecamatan = #{id_kecamatan}")

```

```
List<KelurahanModel> selectAllKelurahanID (@Param("id_kecamatan") int id_kecamatan);
```

Step bisa dilihat disini

The image shows two screenshots of a web application. The top screenshot displays the 'Ubah Keluarga' form with the following fields: 'Alamat' (Address) with the value 'Ki. Katamso No. 429', 'RT' (RT) with the value '125', 'RW' (RW) with the value '094', and 'Kelurahan' (Sub-district) with a dropdown menu showing 'PULAU PARI'. The bottom screenshot shows a success message 'Sukses!' (Success!) and 'Keluarga dengan NKK 3171012210170000 berhasil ditambahkan' (Family with NKK 3171012210170000 successfully added). Below the message is a table with the following data:

	Edit	Copy	Delete	2	3171020101990002	Ki. Katamso No. 429	125	094	15	0
					3171020101990002	Ki. Katamso No. 429	125	094	15	0

Fitur 5 : Ubah Penduduk

Di fitur ini kodingan yang saya gunakan adalah

Controller

```
@GetMapping("/penduduk/ubah/{NIK}")
```

```
public String ubahPenduduk( Model model, @PathVariable(value = "NIK") String NIK)
```

```
{
```

Jadi untuk di method getmappingnya, kita select dulu penduduk yang mau diubah karena field dari form harus teriisi dengan value yang ada sekarang. Setelah itu dicek apakat statusnya wafat. Jika sudah wafat maka tidak boleh lagi mengganti data.

```
PendudukModel a = sidukDAO.selectPenduduk(NIK);
```

```

        if(a.is_wafat==0) {
            model.addAttribute("penduduk", a);
            model.addAttribute("nik", NIK);
            return "ubahpenduduk";
        }else {
            model.addAttribute("nik", NIK);
            return "mati";
        }
    }
}

```

```

@PostMapping(value= "/penduduk/ubah/{NIK}")

public String ubahPendudukSubmit (@ModelAttribute PendudukModel penduduk, Model model,
@PathVariable(value = "NIK") String NIK)
{

```

Mirip seperti tambah penduduk, kita belah nik jadi tiga bagian dan menyimpan nik awalnya, dengan asumsi bahwa perubahan yang dilakukan akan mengubah NIK

```

PendudukModel p = sidukDAO.selectPenduduk(NIK);

String niklama = p.nik;

String[] lahir = penduduk.tanggal_lahir.split("-");

String awal = penduduk.nik.substring(0,6);

String tengah = lahir[2] + lahir[1] + lahir[0].substring(2, 4);

String akhir = penduduk.nik.substring(11, 16);

int tl = 0;

```

Jika user mengganti keluarga, tentunya kita harus cek ulang apakah ada. Maka dari itu akan dilakukan pengecekan lagi seperti fitur 3. Jika keluarga tidak ada maka error, jika ada maka akan memproses perubahan

```

KeluargaModel keluarga = sidukDAO.selectKeluarga(penduduk.id_keluarga);

```



```
if(keluarga!=null) {
```

Diketahui bahwa ada 3 hal yang merubah nik. Tanggal lahir, jenis kelamin dan keluarga karena bisa jadi ada perubahan domisili. Maka dicek apakah yang baru dengan yang lama sama. Jika tidak maka akan dilakukan generate nik lagi seperti di fitur 3, hanya lebih sederhana

```
if(p.tanggal_lahir!=penduduk.tanggal_lahir) {
```

```
String bulan = lahir[1];
```

```
String tahun = lahir[0].substring(2, 4);
```

```
String tanggal = lahir[2];
```

```
tengah = tanggal + bulan + tahun;
```

```
tl= Integer.parseInt(tanggal);
```

```
}
```

```
if(p.id_keluarga!=penduduk.id_keluarga) {
```

```
    awal = keluarga.lurah.camat.kode_kecamatan.substring(0, 6);
```

```
}
```

```
if(p.jenis_kelamin!=penduduk.jenis_kelamin) {
```

```
    if(penduduk.jenis_kelamin==1) {
```

```
        tl = Integer.parseInt(lahir[2]);
```

```
        tl+=40;
```

```
        tengah = tl + "" + lahir[1] + lahir[0];
```

```
    }
```

```
}
```

```
String nikcheck = awal+tengah;
```

```

        if(nikcheck!=p.nik.substring(0, 12)) {
            PendudukModel p1 = sidukDAO.cekPenduduk(nikcheck);
            int index =0;

            if(p1!=null) {
                index += Integer.parseInt(p1.nik.substring(12, 16));
                index++;
            }else {
                index = 0;
            }

            akhir = String.format("%04d", index);
            penduduk.nik = awal + tengah + akhir;
        }else {
            penduduk.nik = niklama;
        }

        sidukDAO.updatePenduduk(penduduk);
        model.addAttribute("nik", NIK);
        model.addAttribute("baru", penduduk);
        model.addAttribute("lama", niklama);

    }

    return "ubahpenduduk-sukses";

}

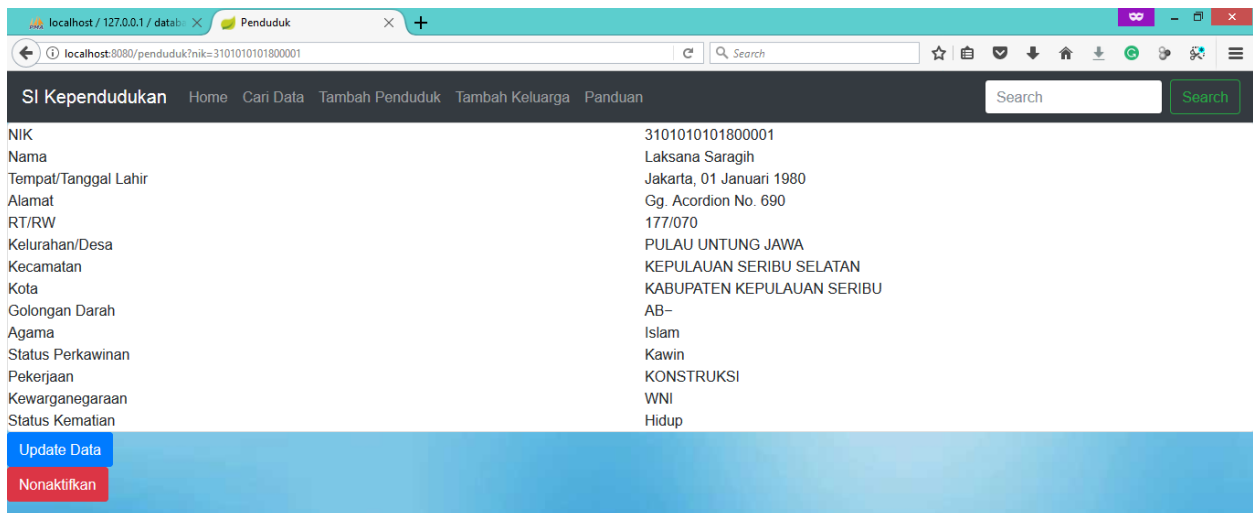
```

Mapper

```
@Update("UPDATE penduduk SET nik= #{nik}, nama = #{nama}, tempat_lahir = #{tempat_lahir}, "
      + "tanggal_lahir = #{tanggal_lahir}, jenis_kelamin = #{jenis_kelamin}, "
      + "is_wni= #{is_wni}, is_wafat= #{is_wafat}, golongan_darah= #{golongan_darah}, "
      + "status_perkawinan= #{status_perkawinan}, status_dalam_keluarga= "
      + "status_dalam_keluarga}, "
      + "agama= #{agama}, pekerjaan= #{pekerjaan}, id_keluarga= #{id_keluarga} "
      + "where id=#{id}")

void updatePenduduk (PendudukModel penduduk);
```

Step dapat dilihat dibawah ini. Yang dicetak NIK lama tetapi karena tanggal berubah, bisa dilihat di database bahwa NIK berubah



NIK	3101010101800001
Nama	Laksana Saragih
Tempat/Tanggal Lahir	Jakarta, 01 Januari 1980
Alamat	Gg. Acordion No. 690
RT/RW	177/070
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	AB-
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	KONSTRUKSI
Kewarganegaraan	WNI
Status Kematian	Hidup

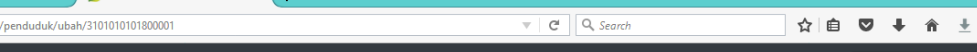
Update Data

Nonaktifkan

The screenshot shows a web browser window with the address bar displaying 'localhost / 127.0.0.1 / datab...'. The browser has multiple tabs, including 'Ubah Penduduk'. The main content area is a form titled 'Ubah Penduduk' (Update Resident) with a blue background. The form contains the following fields:

- Jenis Kelamin** (Gender): A dropdown menu with 'Laki-laki' (Male) selected.
- Tempat Lahir** (Place of Birth): A text input field containing 'Jakarta'.
- Tanggal Lahir** (Date of Birth): A text input field containing '1980-01-01'.

A calendar widget is open below the date input field, showing the month of January 1980. The calendar has a header 'January 1980' and a grid of days. The date '1' is highlighted in the first row, under the 'Su' (Sunday) column.



localhost:3080/penduduk/ubah/3101010101800001

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan

Sukses!

Penduduk dengan NIK 3101010101800001 berhasil diubah

mangrove					status								
Edit	Copy	Delete	4	3101011101800000	Laksana Saragih	Jakarta	1980-01-11	0	1	799	Islam	KONSTRUKSI	Kawin

Fitur 6 : Ubah Keluarga

Di fitur ini kodingan yang saya gunakan adalah

Controller

Mirip fitur 5 hanya saja objeknya keluarga. Sama dengan fitur 5, keluarga yang KK-nya sudah nonaktif akan diarahkan ke page lain. Jika masih aktif maka bisa merubah data.

```
@GetMapping("/keluarga/ubah/{NKK}")
```

```
public String ubahKeluarga( Model model, @PathVariable(value = "NKK") String NKK)
{
    KeluargaModel a = sidukDAO.cariKeluarga(NKK);
    if(a.is_tidak_berlaku==0) {
        List<KelurahanModel> lurah = sidukDAO.selectAllKelurahan();
        model.addAttribute("kelurahan", lurah);
        model.addAttribute("keluarga", a);
        model.addAttribute("nkk", NKK);
        return "ubahkeluarga";
    }else {
        return "mati";
    }
}
```

```
@PostMapping(value= "/keluarga/ubah/{NKK}")
```

```
public String ubahKeluargaSubmit (@ModelAttribute KeluargaModel keluarga, Model model,
@PathVariable(value = "NKK") String NKK)
{
```

Mirip fitur 5, akan dilakukan pembagian nomor kk jadi 3. Setelah itu akan dicek apakah kelurahan berubah. Jika berubah maka NKK akan berubah dan index pun perlu dicek.

```
KeluargaModel p = sidukDAO.cariKeluarga(NKK);
```

```
String nkklama = p.nomor_kk;
```

```
String awal = keluarga.nomor_kk.substring(0,6);
```

```
String tengah = keluarga.nomor_kk.substring(6, 12);
```

```
String akhir = keluarga.nomor_kk.substring(12, 16);
```

```
int tl = 0;
```

```
if(p.id_kelurahan!=keluarga.id_kelurahan) {
```

```
    KelurahanModel lurah = sidukDAO.selectKelurahan(keluarga.id_kelurahan);
```

```
    awal = lurah.camat.kode_kecamatan.substring(0, 6);
```

```
}
```

```
if(p.RT!=keluarga.RT || p.RW!=keluarga.RW) {
```

```
    int b1= Integer.parseInt(keluarga.RT);
```

```
int b= Integer.parseInt(keluarga.RW);
```

```
keluarga.RT = String.format("%03d", b1);
```

```
keluarga.RW = String.format("%03d", b);
```

```
}
```

```
String nkkcheck = awal+tengah;
```

```
if(nkkcheck!=p.nomor_kk.substring(0, 12)) {
```

```
    KeluargaModel p1 = sidukDAO.cekNKK(nkkcheck);
```

```
int index =0;
```

```
if(p1!=null) {
```

```
    index += Integer.parseInt(p1.nomor_kk.substring(12, 16));
```

```
    index++;
```

```
}else {
```

```

        index = 0;
    }

    akhir = String.format("%04d", index);
    keluarga.nomor_kk = awal + tengah + akhir;
} else {
    keluarga.nomor_kk = nkklama;
}

sidukDAO.updateKeluarga(keluarga);
model.addAttribute("nkk", NKK);
model.addAttribute("baru", keluarga);
model.addAttribute("lama", nkklama);

```

```

return "ubahkeluarga-sukses";

```

```

}

```

Mapper

```

@Update("UPDATE keluarga SET nomor_kk= #{nomor_kk}, alamat = #{alamat}, RT = #{RT}, "
        + "RW = #{RW}, id_kelurahan = #{id_kelurahan}, is_tidak_berlaku=
        #{is_tidak_berlaku} where id=#{id}")

```

```

void updateKeluarga (KeluargaModel keluarga);

```

Step dapat dilihat dibawah ini. Yang dicetak NKK lama tetapi karena kelurahan berubah, bisa dilihat di database bahwa NKK berubah

localhost / 127.0.0.1 / data: Ubah Keluarga

localhost:8080/keluarga/ubah/3101010101990001

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan Search Search

Alamat

Ki. Katamso No. 429

RT

125

RW

094

Kelurahan

PULAU PARI

localhost / 127.0.0.1 / data: Ubah Keluarga

localhost:8080/keluarga/ubah/3101010101990001

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan Search Search

Alamat

Ki. Katamso No. 429

RT

125

RW

094

Kelurahan

KEBAGUSAN

localhost / 127.0.0.1 / data: Ubah Keluarga Sukses

localhost:8080/keluarga/ubah/3101010101990001

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan Search Search

Sukses!

Penduduk dengan NIK 3101010101990001 berhasil diubah

<input type="checkbox"/>	Edit Copy Delete	2	3171020101990002	Ki. Katamso No. 429	125	094	15	0
--------------------------	------------------	---	------------------	---------------------	-----	-----	----	---

Fitur 8 : Cari penduduk berdasarkan kota, kecamatan, kelurahan

Di fitur ini kodingan yang saya gunakan adalah

Controller

Disini saya pakai satu mapping tetapi banyak kondisinya. Paramnya adalah kt, kc, kl

```
@RequestMapping("/penduduk/cari")
```

```
    public String cari(@RequestParam(value = "kt", required=false) Integer kt,  
                      @RequestParam(value = "kc", required=false) Integer kc,  
                      @RequestParam(value = "kl", required=false) Integer kl, Model model) {
```

Karena kt dan kc diperlukan tetapi dikhawatirkan hilang, maka kt dan kc yang sudah didapat akan disimpan. Kl tidak disimpan karena pasti tersimpan mengingat ia di urutan terakhir post

```
        String KT="";
```

```
        String KC="";
```

Jika belum ada apa-apa, maka kita akan melakukan list semua kota untuk dropdown dan dikembalikan ke page cari

```
        if(kt==null&&kc==null&&kl==null) {
```

```
            List<KotaModel> kotas = sidukDAO.selectAllKota();
```

```
            model.addAttribute("lkota", kotas);
```

```
            return "cari";
```

```
        }else if(kt!=null&&kc==null&&kl==null){
```

Setelah kota disubmit maka akan muncul dropdown kecamatan. Sementara dropdown kota akan dihilangkan lalu diganti dengan nama kota yang dipilih. Lalu nilai kt akan disimpan. Setelah itu dilakukan lagi pembuatan list kecamatan untuk populate dropdown

```
            List<KecamatanModel> kecamatans = sidukDAO.selectAllKecamatan(kt);
```

```
            KT = kt+"";
```

```
            model.addAttribute("kt", KT);
```

```
            KotaModel kota = sidukDAO.selectKota(kt);
```

```
            model.addAttribute("kecamatans", kecamatans);
```

```
model.addAttribute("lkota", kota);  
    return "cari-kota";
```

```
}else if(kt!=null&&kc!=null&&kl==null){
```

Setelah kecamatan disubmit maka akan muncul dropdown kelurahan. Sementara dropdown kota dan kecamatan akan dihilangkan lalu diganti dengan nama yang dipilih. Lalu nilai kt dan kc akan disimpan. Setelah itu dilakukan lagi pembuatan list kelurahan untuk populate dropdown

```
    KC = kc+"";  
    KT = kt+"";  
    List<KelurahanModel> kelurahans = sidukDAO.selectAllKelurahanID(kc);  
    KecamatanModel kecamatans = sidukDAO.selectKecamatan(kc);  
    KotaModel kota = sidukDAO.selectKota(kt);  
    model.addAttribute("kc", KC);  
    model.addAttribute("kt", KT);  
    model.addAttribute("kecamatans", kecamatans);  
    model.addAttribute("kelurahans", kelurahans);  
    model.addAttribute("lkota", kota);  
    return "cari-camat";
```

```
}else{
```

Setelah kelurahan disubmit maka akan dilakukan populate list penduduk yang ada di kelurahan tersebut. Akan digunakan cariPendudukKelurahan dengan parameter id kelurahan, diambil penduduk termuda dan tertuanya untuk dicetak

```
    KecamatanModel kecamatans = sidukDAO.selectKecamatan(kc);  
    KotaModel kota = sidukDAO.selectKota(kt);  
    KelurahanModel kelurahans = sidukDAO.selectKelurahan(kl);  
    List<PendudukModel> keluarga = sidukDAO.cariPendudukKelurahan(kl);  
    PendudukModel muda = keluarga.get(keluarga.size()-1);  
    PendudukModel tua = keluarga.get(0);
```

```

        model.addAttribute("keluargas", keluargas);

        model.addAttribute("tua", tua);

        model.addAttribute("muda", muda);

        model.addAttribute("kecamatanans", kecamatanans);

        model.addAttribute("kelurahans", kelurahans);

        model.addAttribute("lkota", kota);

        return "cari-lurah";

    }

}

```

Mapper

```
@Select("select * from kelurahan where id_kecamatan = #{id_kecamatan}")
```

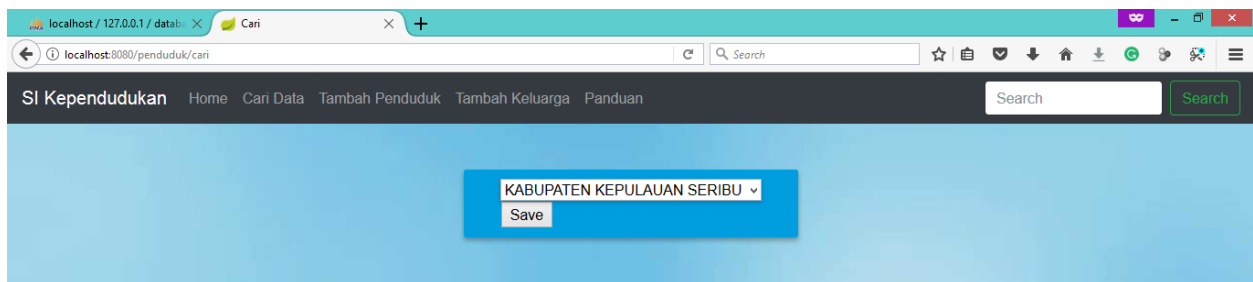
```
List<KelurahanModel> selectAllKelurahanID (@Param("id_kecamatan") int id_kecamatan);
```

Jadi yang dibawah ini cukup penting karena akan dipilih penduduk sesuai kelurahannya, dan diurut sesuai tanggal lahir. Hal ini bertujuan agar mudah mendapat penduduk termuda dan tertua

```
@Select("select p.nama, p.nik, p.jenis_kelamin, p.tanggal_lahir, p.tempat_lahir from penduduk p,
keluarga k where p.id_keluarga = k.id and id_kelurahan = #{id_kelurahan} ORDER BY p.tanggal_lahir ASC")
```

```
List<PendudukModel> cariPendudukKelurahan(@Param("id_kelurahan") int id);
```

Berikut ini adalah stepnya



localhost / 127.0.0.1 / datab... Cari Kota

localhost:8080/penduduk/cari?kt=1&action=save

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan

Search Search

List Penduduk KABUPATEN KEPULAUAN SERIBU

Nama Kota
KABUPATEN KEPULAUAN SERIBU

KEPULAUAN SERIBU SELATAN

Save

localhost / 127.0.0.1 / datab... Cari Kecamatan

localhost:8080/penduduk/cari?kt=1&kc=1

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan

Search Search

List Penduduk KABUPATEN KEPULAUAN SERIBU, KEPULAUAN SERIBU SELATAN

Nama Kota
KABUPATEN KEPULAUAN SERIBU

Nama Kecamatan
KEPULAUAN SERIBU SELATAN

PULAU TIDUNG

Save

localhost / 127.0.0.1 / datab... Cari Lurah

localhost:8080/penduduk/cari?kt=1&kc=1&kl=3

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan

Search Search

List Penduduk KABUPATEN KEPULAUAN SERIBU, KEPULAUAN SERIBU SELATAN, PULAU UNTUNG JAWA

Penduduk Termuda

NIK: 3101015909170001

Nama lengkap: Wirda Uyainah S.Pd

TTL: Jakarta, 2017-09-19

Penduduk Tertua

NIK: 3101015810520002

Nama lengkap: Ciaobella Yolanda

TTL: Jakarta, 1952-10-18

Show 10 entries Search:

NIK	Nama Lengkap	Jenis Kelamin
3101010101690001	Ridwan Hasim Marpaung	Laki-laki
3101010102930001	Erik Ganep Budiman	Laki-laki
3101010102940001	Umay Ozy Prasasta	Laki-laki
3101010103160001	Marsito Samosir	Laki-laki

Fitur 7 :Ubah status kematian

Di fitur ini kodingan yang saya gunakan adalah

Controller

Fitur ini belum berjalan sempurna karena belum bisa redirect dan masih ada {NIK} sebagai pathvariablenya. Methodnya juga belum post, hanya mengganti status is_wafat dan mengecek apakah dia anggota satu-satunya dari keluarga tersebut yang masih aktif.

```
@RequestMapping("/penduduk/mati/{NIK}")
```

```
public String ubahPendudukNonaktif(@PathVariable(value = "NIK") String NIK, Model model)
{
    PendudukModel a = sidukDAO.selectPenduduk(NIK);

    List<PendudukModel> b = sidukDAO.cariAnggotaHidup(a.id_keluarga);
    if(b.size()==1) {
        a.keluarga.is_tidak_berlaku = 1;
    }
    a.is_wafat = 1;

    sidukDAO.updateKeluarga(a.keluarga);
    sidukDAO.updatePenduduk(a);
    model.addAttribute("penduduk", a);
    model.addAttribute("keluarga", a.keluarga);
    model.addAttribute("nik", a.nik);
    return "mati";
}
```

Mapper

Yang paling penting adalah mencari list anggota keluarga yang masih hidup, agar objek keluarga juga dinonaktifkan bila penduduk yang diubah statusnya adalah anggota terakhir yang hidup.

```
@Select("select * from penduduk where id_keluarga = #{id} and is_wafat=0")
```

```
List<PendudukModel> cariAnggotaHidup (@Param("id") int id);
```

Berikut adalah stepnya

localhost / 127.0.0.1 / data: Penduduk

localhost:8080/penduduk?nik=3101010101800001

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga Panduan Search Search

NIK	3101010101800001
Nama	Laksana Saragih
Tempat/Tanggal Lahir	Jakarta, 01 Januari 1980
Alamat	Gg. Acordion No. 690
RT/RW	177/070
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	AB-
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	KONSTRUKSI
Kewarganegaraan	WNI
Status Kematian	Hidup

Update Data

Nonaktifkan

Pilih nonaktifkan

Optimasi dan Stress Testing

Untuk optimasi, saya mengurangi penggunaan nested query yang menggunakan many. Saya juga mengurangi penggunaan List atau join yang tidak diperlukan. Untuk stress testing saya melakukan untuk number of threads 100 dan ramp-up period 10, hasilnya tidak bagus, masih lebih dari harapan

Fitur 1

Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - View Results in Table
 - WorkBench

View Results in Table

Name: View Results in Table

Comments: cari penduduk

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency
1	22:20:15.094	Thread Grou...	HTTP Request	41378	✓	4568	146	413
2	22:20:13.838	Thread Grou...	HTTP Request	42641	✓	4568	146	426
3	22:20:16.548	Thread Grou...	HTTP Request	39935	✓	4568	146	399
4	22:20:16.052	Thread Grou...	HTTP Request	40434	✓	4568	146	404
5	22:20:13.837	Thread Grou...	HTTP Request	42652	✓	4568	146	426
6	22:20:18.128	Thread Grou...	HTTP Request	60670	✓	4568	146	606
7	22:20:17.529	Thread Grou...	HTTP Request	61275	✓	4568	146	612
8	22:20:19.337	Thread Grou...	HTTP Request	59477	✓	4568	146	594
9	22:20:20.733	Thread Grou...	HTTP Request	58100	✓	4568	146	581
10	22:20:18.228	Thread Grou...	HTTP Request	60608	✓	4568	146	606
11	22:20:18.629	Thread Grou...	HTTP Request	60233	✓	4568	146	602
12	22:20:19.236	Thread Grou...	HTTP Request	59657	✓	4568	146	596
13	22:20:22.422	Thread Grou...	HTTP Request	56484	✓	4568	146	564
14	22:20:22.820	Thread Grou...	HTTP Request	56111	✓	4568	146	561
15	22:20:16.348	Thread Grou...	HTTP Request	62624	✓	4568	146	626
16	22:20:19.631	Thread Grou...	HTTP Request	59344	✓	4568	146	593
17	22:20:13.874	Thread Grou...	HTTP Request	65130	✓	4568	146	651
18	22:20:18.430	Thread Grou...	HTTP Request	60598	✓	4568	146	605
19	22:20:21.223	Thread Grou...	HTTP Request	57844	✓	4568	146	578
20	22:20:20.330	Thread Grou...	HTTP Request	58745	✓	4568	146	587

Fitur 2

Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

Test Plan
Thread Group
HTTP Request
View Results in Table
WorkBench

View Results in Table
Name: View Results in Table
Comments: cari keluarga
Write results to file / Read from file
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	22:23:51.881	Thread Group ...	HTTP Request	9800	✓	5585	146	9800	2
2	22:23:52.102	Thread Group ...	HTTP Request	9584	✓	5585	146	9583	2
3	22:23:52.183	Thread Group ...	HTTP Request	9517	✓	5585	146	9517	1
4	22:23:52.683	Thread Group ...	HTTP Request	11264	✓	5585	146	11263	2
5	22:23:52.383	Thread Group ...	HTTP Request	11590	✓	5585	146	11590	1
6	22:23:52.283	Thread Group ...	HTTP Request	18223	✓	5585	146	18223	1
7	22:23:52.583	Thread Group ...	HTTP Request	17928	✓	5585	146	17927	1
8	22:23:51.990	Thread Group ...	HTTP Request	18547	✓	5585	146	18547	3
9	22:23:52.486	Thread Group ...	HTTP Request	24177	✓	5585	146	24177	1
10	22:23:52.991	Thread Group ...	HTTP Request	26839	✓	5585	146	26839	1
11	22:23:53.994	Thread Group ...	HTTP Request	37385	✓	5585	146	37384	1
12	22:23:54.596	Thread Group ...	HTTP Request	41031	✓	5585	146	41030	3
13	22:23:53.623	Thread Group ...	HTTP Request	42082	✓	5585	146	42081	2
14	22:23:53.292	Thread Group ...	HTTP Request	42453	✓	5585	146	42452	1
15	22:23:52.791	Thread Group ...	HTTP Request	43026	✓	5585	146	43026	1
16	22:23:52.926	Thread Group ...	HTTP Request	42954	✓	5585	146	42954	1
17	22:23:55.101	Thread Group ...	HTTP Request	40986	✓	5585	146	40986	2
18	22:23:54.496	Thread Group ...	HTTP Request	50566	✓	5585	146	50566	1
19	22:23:55.904	Thread Group ...	HTTP Request	49282	✓	5585	146	49282	1
20	22:23:53.191	Thread Group ...	HTTP Request	52201	✓	5585	146	52201	1
21	22:23:55.201	Thread Group ...	HTTP Request	50300	✓	5585	146	50299	1
22	22:23:53.692	Thread Group ...	HTTP Request	52520	✓	5585	146	52520	2
23	22:23:54.196	Thread Group ...	HTTP Request	52108	✓	5585	146	52107	1
24	22:23:53.091	Thread Group ...	HTTP Request	57351	✓	5585	146	57350	1
25	22:23:53.393	Thread Group ...	HTTP Request	57228	✓	5585	146	57228	1
26	22:23:55.832	Thread Group ...	HTTP Request	57844	✓	5585	146	57844	1
27	22:23:57.810	Thread Group ...	HTTP Request	56920	✓	5585	146	56920	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 64743 Average 54380 Deviation 14303

Fitur 5

Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

Test Plan
Thread Group
HTTP Request
View Results in Table
WorkBench

View Results in Table
Name: View Results in Table
Comments: subah pendudu
Write results to file / Read from file
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	22:28:23.739	Thread Group ...	HTTP Request	23168	✓	8935	147	23168	1
2	22:28:24.141	Thread Group ...	HTTP Request	22769	✓	8935	147	22768	1
3	22:28:23.638	Thread Group ...	HTTP Request	23283	✓	8935	147	23283	1
4	22:28:23.538	Thread Group ...	HTTP Request	23383	✓	8935	147	23383	1
5	22:28:24.041	Thread Group ...	HTTP Request	26202	✓	8935	147	26201	1
6	22:28:23.436	Thread Group ...	HTTP Request	26885	✓	8935	147	26885	3
7	22:28:24.242	Thread Group ...	HTTP Request	28092	✓	8935	147	28092	1
8	22:28:23.839	Thread Group ...	HTTP Request	28571	✓	8935	147	28571	1
9	22:28:23.939	Thread Group ...	HTTP Request	28510	✓	8935	147	28510	1
10	22:28:26.545	Thread Group ...	HTTP Request	36129	✓	8935	147	36129	2
11	22:28:25.946	Thread Group ...	HTTP Request	36732	✓	8935	147	36732	1
12	22:28:25.445	Thread Group ...	HTTP Request	67447	✓	8935	147	67446	1
13	22:28:24.642	Thread Group ...	HTTP Request	68324	✓	8935	147	68324	1
14	22:28:26.045	Thread Group ...	HTTP Request	69933	✓	8935	147	69931	1
15	22:28:25.644	Thread Group ...	HTTP Request	67358	✓	8935	147	67358	2
16	22:28:25.744	Thread Group ...	HTTP Request	67260	✓	8935	147	67260	1
17	22:28:24.945	Thread Group ...	HTTP Request	68086	✓	8935	147	68086	1
18	22:28:26.746	Thread Group ...	HTTP Request	66317	✓	8935	147	66317	1
19	22:28:24.441	Thread Group ...	HTTP Request	68670	✓	8935	147	68670	2
20	22:28:24.541	Thread Group ...	HTTP Request	68573	✓	8935	147	68573	2
21	22:28:25.043	Thread Group ...	HTTP Request	68104	✓	8935	147	68104	2
22	22:28:25.543	Thread Group ...	HTTP Request	67626	✓	8935	147	67626	1
23	22:28:26.645	Thread Group ...	HTTP Request	66541	✓	8935	147	66541	2
24	22:28:25.144	Thread Group ...	HTTP Request	68050	✓	8935	147	68050	2
25	22:28:25.844	Thread Group ...	HTTP Request	67362	✓	8935	147	67362	1
26	22:28:24.742	Thread Group ...	HTTP Request	68481	✓	8935	147	68481	1
27	22:28:26.445	Thread Group ...	HTTP Request	66778	✓	8935	147	66778	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 75515 Average 68350 Deviation 14762

Fitur 6

Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

Test Plan
Thread Group
HTTP Request
View Results in Table
WorkBench

View Results in Table
Name: View Results in Table
Comments:subah keluarga
Write results to file / Read from file
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	22:26:35.765	Thread Group ...	HTTP Request	8442	✓	23125	147	8441	1
2	22:26:36.070	Thread Group ...	HTTP Request	9870	✓	23125	147	9869	1
3	22:26:35.665	Thread Group ...	HTTP Request	10429	✓	23125	147	10427	1
4	22:26:35.468	Thread Group ...	HTTP Request	11922	✓	23125	147	11921	1
5	22:26:35.565	Thread Group ...	HTTP Request	11836	✓	23125	147	11834	2
6	22:26:35.864	Thread Group ...	HTTP Request	17236	✓	23125	147	17226	1
7	22:26:35.370	Thread Group ...	HTTP Request	17764	✓	23125	147	17762	28
8	22:26:35.967	Thread Group ...	HTTP Request	17215	✓	23125	147	17214	1
9	22:26:36.666	Thread Group ...	HTTP Request	18331	✓	23125	147	18317	1
10	22:26:36.864	Thread Group ...	HTTP Request	18544	✓	23125	147	18541	1
11	22:26:36.464	Thread Group ...	HTTP Request	20178	✓	23125	147	20176	1
12	22:26:37.565	Thread Group ...	HTTP Request	19103	✓	23125	147	19101	1
13	22:26:36.566	Thread Group ...	HTTP Request	21334	✓	23125	147	21332	2
14	22:26:37.276	Thread Group ...	HTTP Request	20632	✓	23125	147	20630	1
15	22:26:36.968	Thread Group ...	HTTP Request	20963	✓	23125	147	20962	1
16	22:26:36.268	Thread Group ...	HTTP Request	37063	✓	23125	147	37061	1
17	22:26:37.965	Thread Group ...	HTTP Request	35662	✓	23125	147	35660	1
18	22:26:37.463	Thread Group ...	HTTP Request	36171	✓	23125	147	36166	1
19	22:26:37.165	Thread Group ...	HTTP Request	36504	✓	23125	147	36500	1
20	22:26:39.065	Thread Group ...	HTTP Request	40753	✓	23125	147	40751	2
21	22:26:37.065	Thread Group ...	HTTP Request	42775	✓	23125	147	42773	2
22	22:26:43.519	Thread Group ...	HTTP Request	36321	✓	23125	147	36320	2
23	22:26:37.665	Thread Group ...	HTTP Request	42685	✓	23125	147	42683	2
24	22:26:38.065	Thread Group ...	HTTP Request	42298	✓	23125	147	42296	1
25	22:26:40.522	Thread Group ...	HTTP Request	39892	✓	23125	147	39886	1
26	22:26:38.865	Thread Group ...	HTTP Request	41739	✓	23125	147	41738	1
27	22:26:38.168	Thread Group ...	HTTP Request	44446	✓	23125	147	44441	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 44362 Average 39485 Deviation 10271

Fitur 7

Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

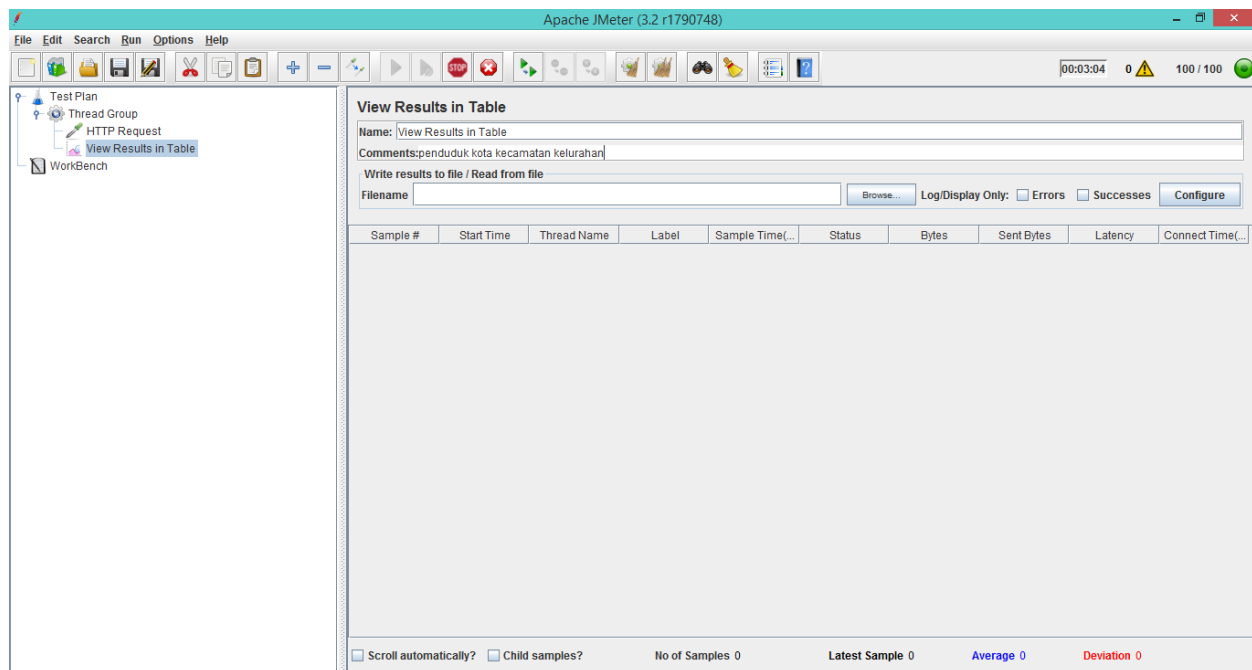
Test Plan
Thread Group
HTTP Request
View Results in Table
WorkBench

View Results in Table
Name: View Results in Table
Comments:penduduk mali
Write results to file / Read from file
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	22:31:22.960	Thread Group ...	HTTP Request	58399	✓	2941	147	58399	2
2	22:31:22.261	Thread Group ...	HTTP Request	82102	✓	2941	147	82102	1
3	22:31:22.561	Thread Group ...	HTTP Request	82336	✓	2941	147	82336	1
4	22:31:25.158	Thread Group ...	HTTP Request	81396	✓	2941	147	81396	1
5	22:31:22.361	Thread Group ...	HTTP Request	87739	✓	2941	147	87739	1
6	22:31:23.259	Thread Group ...	HTTP Request	86848	✓	2941	147	86848	1
7	22:31:24.561	Thread Group ...	HTTP Request	85549	✓	2941	147	85549	1
8	22:31:23.759	Thread Group ...	HTTP Request	86610	✓	2941	147	86610	1
9	22:31:23.859	Thread Group ...	HTTP Request	86568	✓	2941	147	86567	1
10	22:31:23.660	Thread Group ...	HTTP Request	86767	✓	2941	147	86766	1
11	22:31:24.460	Thread Group ...	HTTP Request	86466	✓	2941	147	86466	4
12	22:31:23.459	Thread Group ...	HTTP Request	87962	✓	2941	147	87961	1
13	22:31:22.662	Thread Group ...	HTTP Request	89656	✓	2941	147	89656	1
14	22:31:22.761	Thread Group ...	HTTP Request	90419	✓	2941	147	90419	2
15	22:31:23.359	Thread Group ...	HTTP Request	90156	✓	2941	147	90156	1
16	22:31:22.862	Thread Group ...	HTTP Request	91814	✓	2941	147	91814	2
17	22:31:22.161	Thread Group ...	HTTP Request	93597	✓	2941	147	93597	1
18	22:31:22.461	Thread Group ...	HTTP Request	94317	✓	2941	147	94317	1
19	22:31:24.354	Thread Group ...	HTTP Request	92811	✓	2941	147	92810	1
20	22:31:23.060	Thread Group ...	HTTP Request	94105	✓	2941	147	94104	1
21	22:31:26.857	Thread Group ...	HTTP Request	91539	✓	2941	147	91539	1
22	22:31:22.064	Thread Group ...	HTTP Request	96523	✓	2941	147	96523	2
23	22:31:30.654	Thread Group ...	HTTP Request	89391	✓	2941	147	89390	1
24	22:31:27.456	Thread Group ...	HTTP Request	92967	✓	2941	147	92967	1
25	22:31:24.959	Thread Group ...	HTTP Request	96177	✓	2941	147	96177	1
26	22:31:25.359	Thread Group ...	HTTP Request	97177	✓	2941	147	97177	1
27	22:31:31.356	Thread Group ...	HTTP Request	91300	✓	2941	147	91300	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 113486 Average 102929 Deviation 10711

Fitur 8



Bonus

1. Validasi

Untuk validasi tanggal, saya menggunakan input calendar pop up jadi inputnya pasti sesuai format. Lalu untuk yang integer formnya saya gunakan number. Lalu untuk yang optional seperti agama, jenis kelamin, kewarganegaraan dan semacamnya saya menggunakan dropdown yang isinya disesuaikan dengan standar isian kartu keluarga DKI Jakarta. Sisanya saya mengarahkan ke halaman error, misalkan untuk field kosong

2. Error Page

Dibuat 2 error page, untuk 404 dan 500. Karena keduanya merupakan error yang paling umum

3. Penduduk Termuda dan Tertua

Dapat dilihat di fitur 8. Penduduk termuda dan tertua ada di atas sebelum tabel

NIK	Nama Lengkap	Jenis Kelamin
3101010101690001	Ridwan Hasim Marpaung	Laki-laki
3101010102930001	Erik Ganep Budiman	Laki-laki
3101010102940001	Umay Ozy Prasasta	Laki-laki
3101010103160001	Marsito Samosir	Laki-laki

4. Fitur yang mendukung aplikasi

Saya tidak menambahkan banyak, hanya sedikit didesain dan navbar untuk memudahkan perjalanan fungsi. Adapun panduan hanya berisi teks untuk membantu menjalankan fungsi-fungsi yang ada pada halaman ini.

Implementasi MVC

Jadi struktur project pada main javanya terdapat 5 package, yaitu `com.example.tugas`, `com.example.tugas.model`, `com.example.tugas.dao`, `com.example.tugas.service`, `com.example.tugas.controller`. Untuk `com.example.tugas`, isinya adalah main yang menjalankan Springnya. Lalu pada `com.example.tugas.model`, terdapat 5 model yang merepresentasikan 5 objek penting yaitu penduduk, keluarga, kecamatan, kelurahan, kota. Sama halnya dengan `com.example.tugas.model`, `com.example.tugas.dao`, juga terdapat 5 mapper untuk masing-masing model. Controller terdapat 1 sebagai controller utama, sama halnya seperti servicenya, dijadikan 1 menjadi `SidukService`. Controller dan Service dijadikan satu agar semuanya terpusat dan mudah dihubungkan. Untuk di resource, di template terdapat folder error yang berisi halaman-halaman error, folder fragment untuk header dan footer, lalu halaman-halaman tampilan dari SIDUK. Lalu di static terdapat js dan css yang berfungsi untuk media memperindah tampilan.

