

Nama : Andre Ramadhani
NPM : 1506689484
Kelas : APAP – A

Write-Up Tugas 1

Dalam mengembangkan tugas ini, berbagai konsep dalam Spring seperti MVC dan *layering* digunakan di sini. Selain itu, konsep *presentation* seperti *Thymeleaf* juga digunakan untuk membuat tampilan website yang dinamis.

Dalam tugas ini, saya diminta untuk membuat fitur-fitur utama berikut:

1. Menampilkan data penduduk berdasarkan NIK
2. Menampilkan data keluarga berdasarkan NKK
3. Menambah penduduk baru sebagai anggota keluarga
4. Menambah keluarga baru
5. Mengubah data penduduk
6. Mengubah data keluarga
7. Mengubah status kematian penduduk
8. Menampilkan data penduduk berdasarkan kota/kabupaten, kecamatan, dan kelurahan tertentu

Untuk mengembangkan fitur-fitur tersebut, akan dijelaskan pada bagian di bawah ini.

Fitur 1 : Menampilkan data penduduk berdasarkan NIK

Untuk mengembangkan fitur ini, dibutuhkan dua page HTML, yaitu `index.html` dan `penduduk.html` yang menampilkan halaman utama dan data penduduk yang sedang ditampilkan.

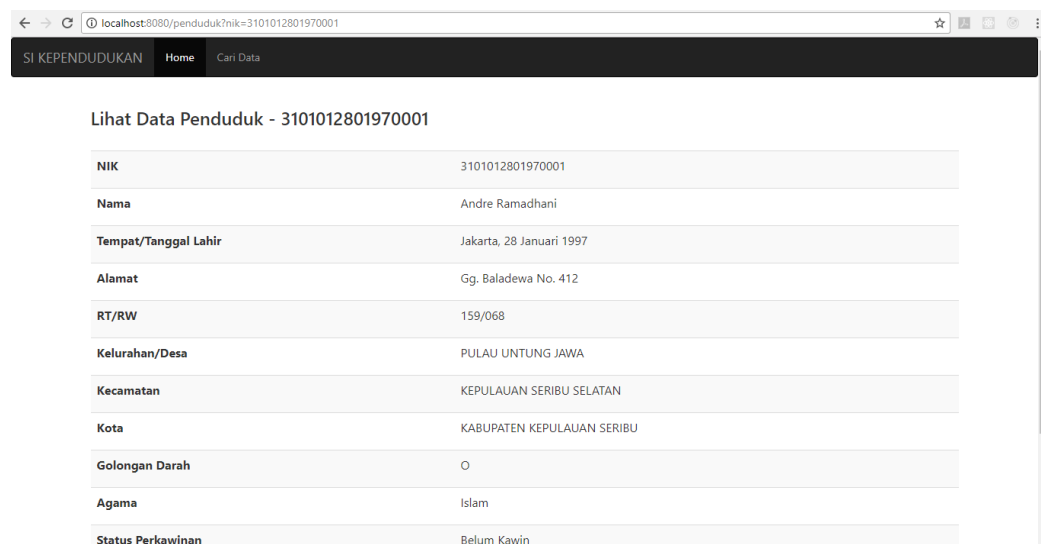
The screenshot displays the web application interface for 'SI KEPENDUDUKAN'. At the top, there is a dark navigation bar with the text 'SI KEPENDUDUKAN' and two links: 'Home' and 'Cari Data'. Below the navigation bar, the main heading reads 'Sistem Informasi Kependudukan Online Provinsi DKI Jakarta', followed by a subtitle 'Selamat datang di Sistem Informasi Online Provinsi DKI Jakarta'. The interface is divided into two main sections. The first section is titled 'Lihat Data Penduduk Berdasarkan NIK' and contains a label 'Masukkan Nomor Induk Kependudukan' above a text input field. Below the input field is a blue button labeled 'Lihat'. The second section is titled 'Lihat Data Keluarga Berdasarkan NKK' and contains a label 'Masukkan Nomor Kartu Keluarga' above another text input field. Below this input field is also a blue button labeled 'Lihat'.

Gambar 1: Tampilan `index.html`

Untuk menyelesaikan fitur ini, dibutuhkan berbagai method untuk melakukan kueri dari *database*. Method-method ini diimplementasikan dalam kelas PendudukMapper, PendudukService, dan PendudukServiceDatabase. Method-method tersebut adalah :

- **getPenduduk(String nik).** Method ini digunakan untuk mendapatkan data penduduk berdasarkan NIK.
- **getKeluarga(int id).** Method ini digunakan untuk mendapatkan data keluarga berdasarkan ID. Untuk kasus ini, method ini digunakan untuk mendapatkan data yang berhubungan dengan penduduk, seperti alamat, RT/RW, dan sebagainya.
- **getKelurahan(int id).** Method ini digunakan untuk mendapatkan data kelurahan berdasarkan ID. Untuk kasus ini, method ini digunakan untuk mendapatkan data yang berhubungan dengan penduduk, seperti nama kelurahan.
- **getKecamatan(int id).** Method ini digunakan untuk mendapatkan data kecamatan berdasarkan ID. Untuk kasus ini, method ini digunakan untuk mendapatkan data yang berhubungan dengan penduduk, seperti nama kecamatan.
- **getKota(int id).** Method ini digunakan untuk mendapatkan data kota berdasarkan ID. Untuk kasus ini, method ini digunakan untuk mendapatkan data yang berhubungan dengan penduduk, seperti nama kota.

Setelah method-method tersebut diselesaikan, method pada controller dibuat untuk mengambil parameter berupa NIK dan menampilkan data penduduk berdasarkan NIK. Kemudian, method tersebut mengembalikan nama dari page HTML yang dituju, yaitu penduduk.html. Hasilnya dapat dilihat sebagai berikut.



NIK	3101012801970001
Nama	Andre Ramadhani
Tempat/Tanggal Lahir	Jakarta, 28 Januari 1997
Alamat	Gg. Baladewa No. 412
RT/RW	159/068
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	O
Agama	Islam
Status Perkawinan	Belum Kawin

Gambar 2: Tampilan penduduk.html dengan NIK yang sudah ada.

Fitur 2 : Menampilkan Data Keluarga

Untuk mengembangkan fitur ini, dibutuhkan dua page HTML, yaitu `index.html` dan `keluarga.html` yang menampilkan halaman utama dan data keluarga yang sedang ditampilkan.

Untuk menyelesaikan fitur ini, dibutuhkan berbagai method untuk melakukan kueri dari *database*. Method-method ini diimplementasikan dalam kelas `PendudukMapper`, `PendudukService`, dan `PendudukServiceDatabase`. Method-method tersebut adalah :

- **`getKelurahan(int id)`**. Method ini digunakan untuk mendapatkan data kelurahan berdasarkan ID. Untuk kasus ini, method ini digunakan untuk mendapatkan data yang berhubungan dengan keluarga, seperti nama kelurahan.
- **`getKecamatan(int id)`**. Method ini digunakan untuk mendapatkan data kecamatan berdasarkan ID. Untuk kasus ini, method ini digunakan untuk mendapatkan data yang berhubungan dengan keluarga, seperti nama kecamatan.
- **`getKota(int id)`**. Method ini digunakan untuk mendapatkan data kota berdasarkan ID. Untuk kasus ini, method ini digunakan untuk mendapatkan data yang berhubungan dengan keluarga, seperti nama kota.
- **`getKeluargaFromNKK(String nkk)`**. Method ini digunakan untuk mengambil data penduduk berdasarkan NKK.
- **`getAnggotaKeluarga(int id)`**. Method ini untuk mendapatkan anggota keluarga berdasarkan ID keluarga.

Setelah method-method tersebut diselesaikan, method pada controller dibuat untuk mengambil parameter berupa NKK dan menampilkan data keluarga berdasarkan NIK. Kemudian, method tersebut mengembalikan nama dari page HTML yang dituju, yaitu `keluarga.html`. Hasilnya dapat dilihat sebagai berikut.

SI KEPENDUDUKAN

Home

Cari Data

Lihat Data Keluarga - 3101022210170001

NKK	3101022210170001
Alamat	Jln. Ahmad Zaini No. 20
RT/RW	011/077
Kelurahan/Desa	PULAU PANGGANG
Kecamatan	KEPULAUAN SERIBU UTARA
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	Kamila Yulianti	3101016307830002	Wanita	Jakarta	1983-07-23	Katholik	TRANSPORTASI	Kawin	Istri	WNI
2	Jati Megantara	3101012005850002	Pria	Jakarta	1985-05-20	Katholik	PERANCANG BUSANA	Kawin	Kepala Keluarga	WNI

Gambar 3 : Tampilan keluarga.html dengan NKK tertentu

Fitur 3 : Menambahkan Penduduk Baru Sebagai Anggota Keluarga

Untuk mengembangkan fitur ini, dibutuhkan satu page HTML, yaitu tambah-penduduk.html yang menampilkan form untuk membuat keluarga baru.

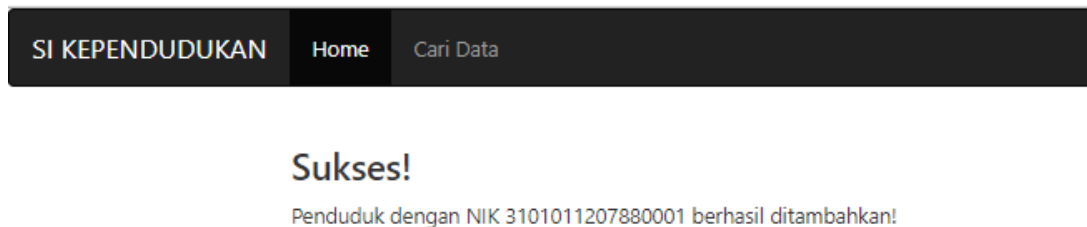
SI KEPENDUDUKAN																			
Home		Cari Data																	
Tambah Penduduk																			
Nama																			
<input type="text"/>																			
Tempat Lahir																			
<input type="text"/>																			
Tanggal Lahir																			
<input type="text" value="yyyy-mm-dd"/>																			
Jenis Kelamin																			
<input type="text" value="Laki-laki"/>																			
Golongan Darah																			
<input type="text" value="A"/>																			
Agama																			
<input type="text"/>																			
Status Perkawinan																			
<input type="text" value="Kawin"/>																			
Pekerjaan																			
<input type="text"/>																			
Kewarganegaraan																			
<input type="text"/>																			

Gambar 4 : Tampilan tambah-penduduk.html

Untuk membuat fitur ini, method yang dikembangkan adalah :

- **addPenduduk(PendudukModel penduduk).** Method ini digunakan untuk memasukkan data dari PendudukModel yang dibuat di controller ke dalam database.

Pada saat data-data dari form masuk ke controller, data digunakan sebagai parameter untuk membuat objek PendudukModel. Awalnya, NIK masih bernilai null karena tidak ada input untuk NIK. Dengan data-data yang diperlukan, NIK di-generate dengan menggunakan helper method bernama **generateNIK()** (terdapat pada PendudukController.java). Setelah itu, nomor NIK yang sudah siap akan dimasukkan ke objek PendudukModel, lalu objek tersebut dimasukkan ke database. Untuk memberi tahu bahwa data berhasil dimasukkan, sistem mengeluarkan page berikut.



Gambar 5 : Tampilan sukses

Fitur 4 : Menambah Keluarga Baru

Untuk mengembangkan fitur ini, dibutuhkan satu page HTML, yaitu tambah-keluarga.html yang menampilkan form untuk membuat keluarga baru.

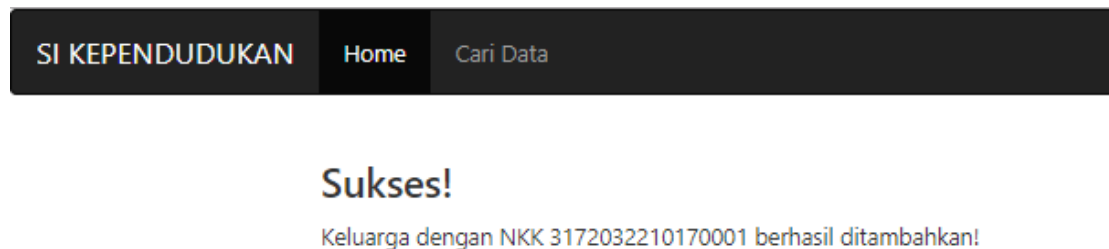
Gambar 6 : Tampilan menambah keluarga

Untuk membuat fitur ini, diperlukan method berikut :

- **addKeluarga(KeluargaModel keluarga).** Method ini digunakan untuk memasukkan data dari KeluargaModel yang dibuat di controller ke dalam database.

- **getAllKelurahan()**. Untuk mendapatkan semua data kelurahan ketika memasukkan alamat.

Pada saat data-data dari form masuk ke controller, data digunakan sebagai parameter untuk membuat objek KeluargaModel. Awalnya, NKK dan anggota masih bernilai null karena tidak ada input untuk NIK dan anggota. Dengan data-data yang diperlukan, NKK di-generate dengan menggunakan helper method bernama **generateNKK()** (terdapat pada PendudukController.java). Setelah itu, nomor NKK yang sudah siap akan dimasukkan ke objek KeluargaModel, lalu objek tersebut dimasukkan ke database. Untuk memberi tahu bahwa data berhasil dimasukkan, sistem mengeluarkan page berikut.



Gambar 7 : Tampilan sukses menambah keluarga

Fitur 5 : Mengubah Data Penduduk

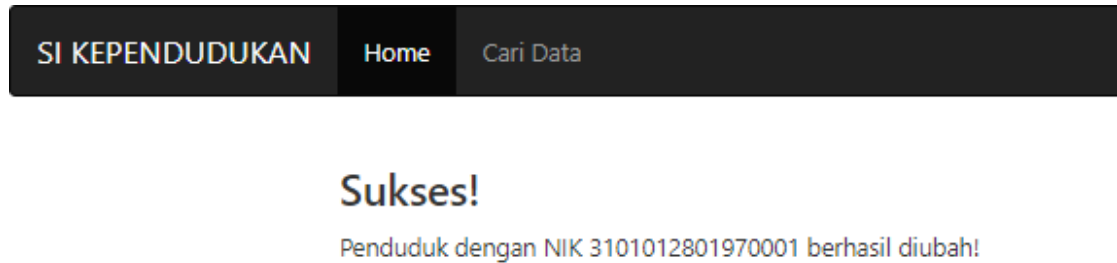
Untuk mengembangkan fitur ini, dibutuhkan satu page HTML, yaitu update-penduduk.html yang menampilkan form untuk mengubah data penduduk.

Gambar 8 : Tampilan ubah penduduk

Untuk membuat fitur ini, dibutuhkan method berikut :

- **updatePenduduk()**. Method ini digunakan untuk mengubah data penduduk dengan parameter perubahan penduduk.

Data-data yang masuk ke controller adalah data-data yang terdapat pada form. Perubahan dilimitasi dengan tidak bisa mengubah tempat dan tanggal lahir karena pada dasarnya tidak mungkin seseorang dapat berganti tanggal lahir. Jika terdapat perubahan pada data keluarga, maka secara otomatis NIK baru akan dibuat untuk penduduk tersebut. Jika perubahan sukses, maka akan ditampilkan page berikut.



Gambar 9 : Tampilan sukses ubah penduduk

Fitur 6 : Mengubah Data Keluarga

Untuk mengembangkan fitur ini, dibutuhkan satu page HTML, yaitu update-keluarga.html yang menampilkan form untuk mengubah data keluarga.

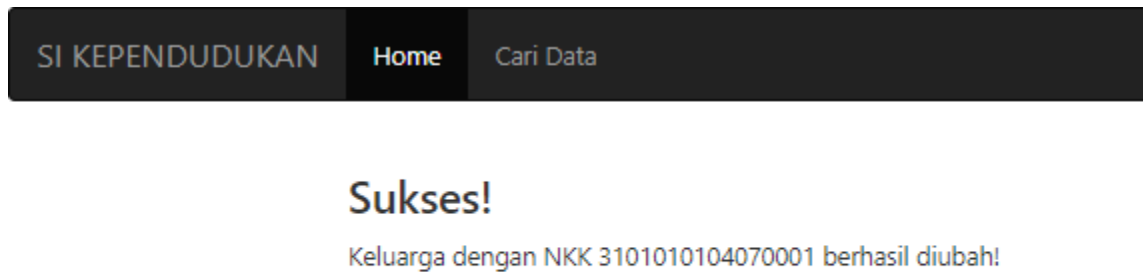
The image shows a web application interface for 'SI KEPENDUDUKAN'. At the top, there is a dark navigation bar with three items: 'SI KEPENDUDUKAN' in white, 'Home' in white, and 'Cari Data' in light blue. Below the navigation bar, the main content area has a white background. The title 'Ubah Keluarga' is centered at the top of the form. Below the title, there are four input fields: 'Alamat' with the value 'Gg. Bappenas No. 9', 'RT' with the value '013', 'RW' with the value '008', and 'Kelurahan, Kecamatan, dan Kota' with a dropdown menu showing 'PULAU PARI, KEPULAUAN SERIBU SELATAN, KABUPATEN KEPULAUAN SERIBU'. At the bottom left of the form, there is a blue button labeled 'Simpan'.

Gambar 10 : Tampilan ubah keluarga

Method yang digunakan untuk mengimplementasikan fitur ini adalah :

- **updateKeluarga()**. Method ini digunakan untuk mengubah data keluarga dengan parameter perubahan keluarga.

Data-data yang masuk ke controller adalah data-data yang terdapat pada form. Jika terdapat perubahan pada data kelurahan, maka secara otomatis NKK baru akan dibuat untuk penduduk tersebut. Jika perubahan sukses, maka akan ditampilkan page berikut.



Gambar 11 : Tampilan sukses ubah keluarga

Fitur 7 : Mengubah Status Kematian

Untuk mengembangkan fitur ini, dibutuhkan satu page HTML, yaitu penduduk.html yang menampilkan data penduduk. Pada halaman tersebut, jika status kematian seorang penduduk masih dinyatakan hidup, maka terdapat tombol untuk menonaktifkan penduduk dan mengubah status kematiannya menjadi wafat.

NIK	3101010405160001
Nama	Prasetya Pangestu
Tempat/Tanggal Lahir	Jakarta, 04 Mei 2016
Alamat	Jln. Sukabumi No. 234
RT/RW	132/017
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	O-
Agama	Islam
Status Perkawinan	Belum Kawin
Pekerjaan	BELUM/TIDAK BEKERJA
Kewarganegaraan	WNI
Status Kematian	Hidup

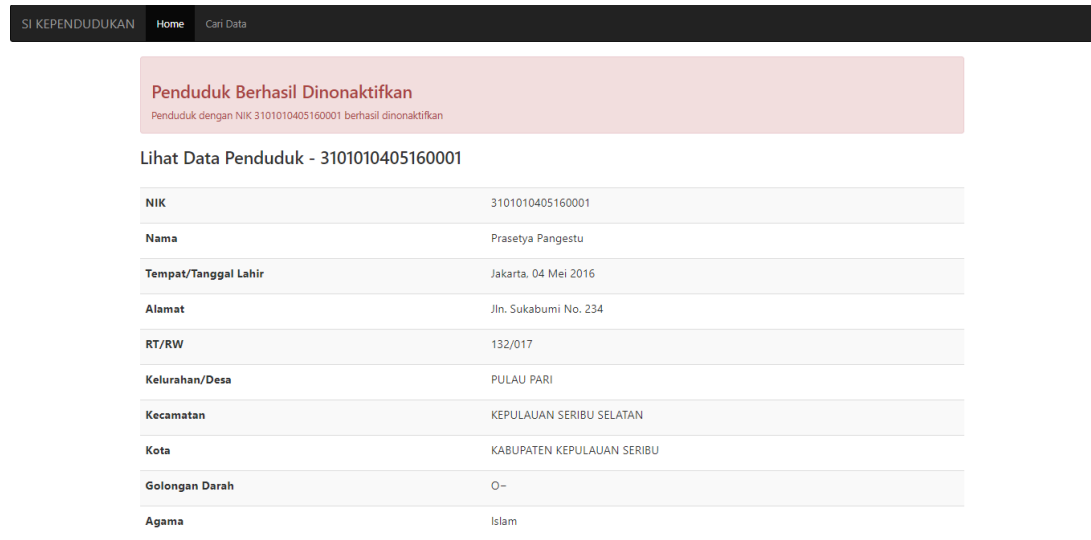
[Nonaktifkan](#)

Gambar 12 : Pada halaman penduduk, terdapat tombol nonaktif

Method yang dikembangkan untuk membuat fitur ini adalah :

- **setDeath()**. Method ini untuk mengubah status kematian seorang penduduk di database.
- **setInactive()**. Method ini untuk mengubah status berlakunya keluarga akibat semua anggota keluarganya yang telah wafat.

Pada saat tombol diklik, maka sistem akan mengubah status kematiannya menjadi wafat. Setelah itu, sistem akan mengecek anggota keluarga yang terhubung dengan penduduk yang dinonaktifkan dan menghitung berapa yang masih hidup. Jika sudah mati semua, maka sistem akan menjalankan setInactive(). Tampilan sukses ini agak berbeda dari yang lain, yakni sebagai berikut.



Gambar 13 : Tampilan pesan sukses menonaktifkan penduduk

Fitur 8 : Melihat Data Penduduk Berdasarkan Kota/Kabupaten, Kecamatan, dan Kelurahan Tertentu

Fitur ini membutuhkan 4 page HTML, yaitu cari penduduk untuk masing-masing kota/kabupaten, kecamatan, dan kelurahan, serta data penduduk.

Untuk mengimplementasikan fitur ini, dibutuhkan method berikut :

- **getAllKota()**. Untuk mendapatkan semua kota yang ada di database
- **getKecamatanFromKota()**. Mendapatkan semua kecamatan berdasarkan kota yang dipilih
- **getKelurahanFromKecamatan()**. Mendapatkan semua kelurahan berdasarkan kecamatan yang dipilih
- **getPendudukInKelurahan()**. Mendapatkan semua penduduk dari suatu kelurahan.

Pada awalnya, sistem akan menampilkan page dimana kita harus memilih kota terlebih dahulu. Setelah kota dipilih, maka kita menentukan kecamatan, lalu menentukan kelurahan. Setelah itu, semua penduduk diambil dengan menggunakan kueri dari database dan ditampilkan dengan bantuan plugin JQuery berupa DataTables. Penggambarannya adalah sebagai berikut.

Lihat Penduduk Berdasarkan Kota, Kecamatan dan Kelurahan

Pilih Kota

Pilih kota...

Lihat

Lihat Penduduk Berdasarkan Kota, Kecamatan dan Kelurahan

Kota

KOTA JAKARTA BARAT

Kecamatan

KEBON JERUK

Lihat

Lihat Penduduk Berdasarkan Kota, Kecamatan dan Kelurahan

Kota

KOTA JAKARTA BARAT

Kecamatan

KEBON JERUK

Kelurahan

Pilih kelurahan...

Lihat

Data Penduduk

Kota	KOTA JAKARTA BARAT
Kecamatan	KEBON JERUK
Kelurahan	KEBON JERUK

Show 10 entries

Search:

No.	NIK	Nama	Jenis Kelamin
1	3174026108170002	Gawati Iriana Anggraini S.I.P	Perempuan
2	3174026312160002	Gawati Padmasari M.Kom.	Perempuan
3	3174025904160004	Dian Nurdyanti	Perempuan
4	3174024904170001	Cinthia Pudjiastuti M.M.	Perempuan
5	3174026107160001	Calista Nurdyanti	Perempuan
6	3174026503820001	Ira Puspita S.H.	Perempuan
7	3174024912130001	Queen Kuswandari	Perempuan
8	3174026504860001	Halima Mardhiyah	Perempuan
9	3174026504960001	Cornelia Agustina	Perempuan
10	3174026504990001	Ifa Latika Hartati M.Pd	Perempuan

Showing 1 to 10 of 1,095 entries

Gambar 14 : Alur pencarian penduduk

Fitur Tambahan : Menampilkan Usia Termuda dan Tertua di Suatu Kelurahan

Fitur ini diletakkan pada page yang sama dengan implementasi fitur 8. Untuk implementasi fitur ini, method yang dibuat adalah :

- **getYoungest()**. Untuk mengambil data penduduk termuda
- **getOldest()**. Untuk mengambil data penduduk tertua

Implementasi fitur ini cukup mudah. Yang perlu dilakukan adalah melakukan kueri dengan tiap method tersebut lalu memunculkannya di page. Fitur ini bermanfaat untuk melihat data penduduk tertua dan termuda dalam suatu kelurahan.

The screenshot shows a web application interface for population data. At the top, there's a navigation bar with 'SI KEPENDUDUKAN', 'Home', and 'Cari Data'. Below this, the 'Data Penduduk' section is displayed. It includes a table with the following data:

Kota	KOTA JAKARTA TIMUR
Kecamatan	KRAMAT JATI
Kelurahan	CILILITAN

Below the table, there are two sections: 'Penduduk dengan Usia Termuda' and 'Penduduk dengan Usia Tertua'. Each section displays a table with the following data:

NIK	Nama	Tanggal Lahir
3172054609170001	Zelaya Mila Pertiwi	2017-09-06

And for the oldest resident:

NIK	Nama	Tanggal Lahir
3172055711530001	Wani Padmasari	1953-11-17

At the bottom, there's a search bar and a table showing a list of residents. The table has columns: No., NIK, Nama, and Jenis Kelamin. The data is as follows:

No.	NIK	Nama	Jenis Kelamin
1	3172056112880001	Yance Uli Suryatmi M.Kom.	Perempuan
2	3172056112950001	Titi Yuniar	Perempuan
3	3172056203890001	Titi Rahayu	Perempuan
4	3172056302040001	Aissa Mulandari	Perempuan

Sayangnya, implementasi fitur ini sangat kotor karena melakukan kueri pada data penduduk dua kali, sehingga akan sangat memakan waktu ketika mengaksesnya.

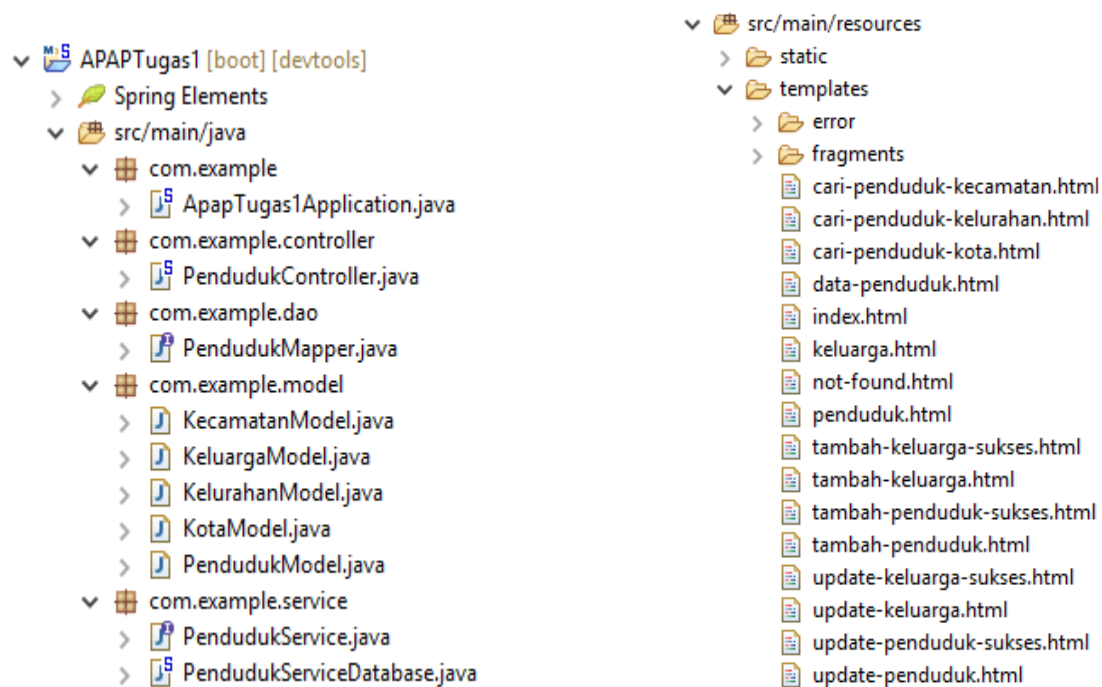
Optimasi Database

Dalam tugas ini, optimasi database yang dilakukan adalah sebagai berikut:

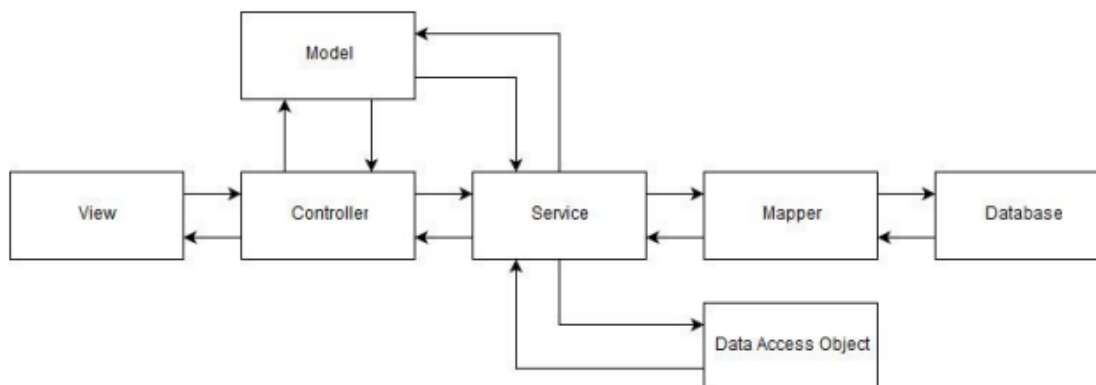
- Melakukan primary indexing dengan memberikan primary key pada atribut "id" di setiap tabel. Selain itu, karena ukuran tabel penduduk dan keluarga yang dapat bertambah, maka id pada kedua tabel tersebut diberikan AUTO_INCREMENT.
- Memberikan secondary indexing pada setiap kode unik (NIK, NKK, kode kelurahan, dan sebagainya).

Konsep MVC dan layering

Dalam mengembangkan tugas ini, struktur folder dalam project ini adalah sebagai berikut.



Secara garis besar, kaitan antar komponen yang terdapat pada framework ini adalah sebagai berikut.



Konsep MVC tersebut diimplementasikan dengan menggunakan framework Spring Boot. Jika ditinjau dari struktur foldernya, maka representasi dari komponen-komponen ini adalah sebagai berikut.

- **View** direpresentasikan dengan folder templates yang berisi file HTML. Digunakan untuk menampilkan antarmuka sistem.
- **Controller** direpresentasikan dengan package `com.example.controller` yang berisi `PendudukController.java`.
- **Model** direpresentasikan dengan package `com.example.model` yang berisi model-model yang digunakan sebagai representasi dari Mapper.
- **Service** direpresentasikan dengan package `com.example.service` yang berisi interface yang menghubungkan Controller dan Mapper.

- **Mapper dan Data Access Object** direpresentasikan dengan package `com.example.dao` yang berisi interface Mapper dan kelas yang berisi perintah untuk melakukan kueri SQL.
- **Database** merupakan infrastruktur database yang dipakai, yaitu MySQL.

Jika ditinjau dari *layering*, maka representasi dari struktur folder ini adalah sebagai berikut.

- **Presentation layer** direpresentasikan dengan folder templates yang menampilkan antarmuka sistem.
- **Domain logic** direpresentasikan oleh semua package Java yang ada. Layer ini memisahkan antara *presentation* dan *data source*.
- **Data source** yaitu database yang digunakan.

Stress Testing

Dalam melakukan eksperimen *stress testing* ini, saya menggunakan Apache JMeter. Testing dilakukan untuk mengetahui performa website saat diakses banyak orang. Eksperimen ini dilakukan dengan *number of threads (users)* = 100, *ramp-up period* 10 detik, dan *loop count* = 1. Dalam eksperimen ini, fitur yang dilibatkan adalah fitur 1, 2, dan 8. Hal ini dikarenakan fitur-fitur ini berfokus pada kueri data dari tiap modelnya.

Hasilnya adalah sebagai berikut:

- Fitur 1 (Melihat Data Penduduk)

View Results in Table

Name: View Results in Table

Comments:

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	21:07:04.283	Thread Group ...	HTTP Request	7509	✓	2392	147	7509	1
2	21:07:03.690	Thread Group ...	HTTP Request	8110	✓	2392	147	8110	2
3	21:07:04.674	Thread Group ...	HTTP Request	7316	✓	2392	147	7316	1
4	21:07:04.189	Thread Group ...	HTTP Request	7804	✓	2392	147	7803	1
5	21:07:03.986	Thread Group ...	HTTP Request	8012	✓	2392	147	8012	1
6	21:07:03.581	Thread Group ...	HTTP Request	8421	✓	2392	147	8420	2
7	21:07:03.893	Thread Group ...	HTTP Request	8110	✓	2392	147	8109	1
8	21:07:04.581	Thread Group ...	HTTP Request	7425	✓	2392	147	7425	1
9	21:07:04.080	Thread Group ...	HTTP Request	7927	✓	2392	147	7927	1
10	21:07:04.377	Thread Group ...	HTTP Request	7634	✓	2392	147	7634	1
11	21:07:03.783	Thread Group ...	HTTP Request	8228	✓	2392	147	8228	2
12	21:07:04.486	Thread Group ...	HTTP Request	7527	✓	2392	147	7527	1
13	21:07:04.881	Thread Group ...	HTTP Request	13516	✓	2392	147	13516	1
14	21:07:04.774	Thread Group ...	HTTP Request	13623	✓	2392	147	13623	1
15	21:07:06.274	Thread Group ...	HTTP Request	17981	✓	2392	147	17981	1
16	21:07:10.375	Thread Group ...	HTTP Request	13887	✓	2392	147	13887	0
17	21:07:05.779	Thread Group ...	HTTP Request	18503	✓	2392	147	18503	1
18	21:07:04.975	Thread Group ...	HTTP Request	19317	✓	2392	147	19317	1
19	21:07:08.317	Thread Group ...	HTTP Request	15977	✓	2392	147	15977	0
20	21:07:07.874	Thread Group ...	HTTP Request	16424	✓	2392	147	16423	1
21	21:07:07.996	Thread Group ...	HTTP Request	16321	✓	2392	147	16321	1
22	21:07:08.990	Thread Group ...	HTTP Request	15327	✓	2392	147	15327	2
23	21:07:10.291	Thread Group ...	HTTP Request	14030	✓	2392	147	14030	1
24	21:07:08.480	Thread Group ...	HTTP Request	22323	✓	2392	147	22323	2
25	21:07:08.093	Thread Group ...	HTTP Request	22710	✓	2392	147	22710	1
26	21:07:06.891	Thread Group ...	HTTP Request	23919	✓	2392	147	23919	1
27	21:07:10.181	Thread Group ...	HTTP Request	20635	✓	2392	147	20635	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 18784 Average 19430 Deviation 5116

- Fitur 2 (Melihat Data Keluarga)

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	21:10:05.241	Thread Group ...	HTTP Request	12427	✓	3626	147	12427	0
2	21:10:04.835	Thread Group ...	HTTP Request	12835	✓	3626	147	12835	1
3	21:10:04.636	Thread Group ...	HTTP Request	13039	✓	3626	147	13038	2
4	21:10:05.037	Thread Group ...	HTTP Request	12644	✓	3626	147	12644	1
5	21:10:04.929	Thread Group ...	HTTP Request	12757	✓	3626	147	12757	1
6	21:10:05.560	Thread Group ...	HTTP Request	12132	✓	3626	147	12132	1
7	21:10:05.461	Thread Group ...	HTTP Request	12231	✓	3626	147	12231	0
8	21:10:05.350	Thread Group ...	HTTP Request	12348	✓	3626	147	12348	1
9	21:10:05.131	Thread Group ...	HTTP Request	12582	✓	3626	147	12582	1
10	21:10:04.726	Thread Group ...	HTTP Request	12987	✓	3626	147	12987	1
11	21:10:05.632	Thread Group ...	HTTP Request	12091	✓	3626	147	12091	0
12	21:10:05.740	Thread Group ...	HTTP Request	14153	✓	3626	147	14153	1
13	21:10:09.343	Thread Group ...	HTTP Request	17276	✓	3626	147	17276	1
14	21:10:06.826	Thread Group ...	HTTP Request	19849	✓	3626	147	19849	1
15	21:10:08.645	Thread Group ...	HTTP Request	18052	✓	3626	147	18052	0
16	21:10:08.841	Thread Group ...	HTTP Request	17869	✓	3626	147	17868	1
17	21:10:06.247	Thread Group ...	HTTP Request	20463	✓	3626	147	20462	1
18	21:10:08.345	Thread Group ...	HTTP Request	18370	✓	3626	147	18370	0
19	21:10:06.141	Thread Group ...	HTTP Request	20584	✓	3626	147	20584	1
20	21:10:06.540	Thread Group ...	HTTP Request	20186	✓	3626	147	20185	1
21	21:10:09.145	Thread Group ...	HTTP Request	17591	✓	3626	147	17590	1
22	21:10:07.426	Thread Group ...	HTTP Request	19310	✓	3626	147	19309	1
23	21:10:09.243	Thread Group ...	HTTP Request	17516	✓	3626	147	17516	1
24	21:10:07.840	Thread Group ...	HTTP Request	18934	✓	3626	147	18934	1
25	21:10:09.744	Thread Group ...	HTTP Request	17033	✓	3626	147	17032	1
26	21:10:05.925	Thread Group ...	HTTP Request	20855	✓	3626	147	20855	1
27	21:10:08.745	Thread Group ...	HTTP Request	18075	✓	3626	147	18075	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 17184 Average 18015 Deviation 2316

- Fitur 8 (Melihat Data Penduduk Berdasarkan Kota, Kecamatan, dan Kelurahan)

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	21:12:08.184	Thread Group ...	HTTP Request	28185	✓	94470	149	28180	1
2	21:12:09.388	Thread Group ...	HTTP Request	26986	✓	94470	149	26769	1
3	21:12:09.849	Thread Group ...	HTTP Request	26529	✓	94470	149	26524	1
4	21:12:09.265	Thread Group ...	HTTP Request	27126	✓	94470	149	27119	1
5	21:12:08.665	Thread Group ...	HTTP Request	27727	✓	94470	149	27721	0
6	21:12:08.766	Thread Group ...	HTTP Request	28003	✓	94470	149	27953	1
7	21:12:09.465	Thread Group ...	HTTP Request	27798	✓	94470	149	27793	1
8	21:12:07.462	Thread Group ...	HTTP Request	29958	✓	94470	149	29954	2
9	21:12:07.669	Thread Group ...	HTTP Request	29757	✓	94470	149	29562	1
10	21:12:08.278	Thread Group ...	HTTP Request	29165	✓	94470	149	29160	2
11	21:12:07.779	Thread Group ...	HTTP Request	29693	✓	94470	149	29686	1
12	21:12:07.575	Thread Group ...	HTTP Request	30758	✓	94470	149	30754	2
13	21:12:07.872	Thread Group ...	HTTP Request	30462	✓	94470	149	30458	2
14	21:12:08.075	Thread Group ...	HTTP Request	30261	✓	94470	149	30255	1
15	21:12:08.566	Thread Group ...	HTTP Request	29843	✓	94470	149	28948	1
16	21:12:09.665	Thread Group ...	HTTP Request	28990	✓	94470	149	28985	1
17	21:12:09.065	Thread Group ...	HTTP Request	29637	✓	94470	149	29632	1
18	21:12:09.966	Thread Group ...	HTTP Request	28748	✓	94470	149	28743	1
19	21:12:09.550	Thread Group ...	HTTP Request	29180	✓	94470	149	29176	1
20	21:12:08.965	Thread Group ...	HTTP Request	29768	✓	94470	149	29763	1
21	21:12:09.766	Thread Group ...	HTTP Request	28972	✓	94470	149	28965	1
22	21:12:08.372	Thread Group ...	HTTP Request	30369	✓	94470	149	30363	2
23	21:12:08.851	Thread Group ...	HTTP Request	29916	✓	94470	149	29910	1
24	21:12:08.466	Thread Group ...	HTTP Request	30301	✓	94470	149	30296	1
25	21:12:09.165	Thread Group ...	HTTP Request	29620	✓	94470	149	29614	1
26	21:12:07.981	Thread Group ...	HTTP Request	31089	✓	94470	149	30807	1
27	21:12:11.966	Thread Group ...	HTTP Request	28207	✓	94470	149	28202	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 30557 Average 30415 Deviation 1609

Berdasarkan eksperimen, dapat disimpulkan bahwa sistem ini masih lambat dalam memproses data-data seperti penduduk, keluarga, kelurahan, dan sebagainya. Terlihat bahwa rata-rata waktu yang digunakan untuk mengakses sistem lebih besar dari 10 detik. Meskipun lambat, sistem ini cukup *reliable* karena dapat menyediakan data dan fitur dengan tepat.