

Tampilan Homepage dan Penjelasan Umum



Aplikasi dikembangkan dengan Java Spring Boot, MySQL, Bootstrap4 dan library js pendukung tampilan lainnya. Aplikasi ini menggunakan MVC Layering seperti yang umum digunakan pada program yang dikembangkan menggunakan Java Spring Boot.

Aplikasi ini memiliki layer Service, Controller, Mapper dan Model. Terdapat 5 Model yaitu PendudukModel, KeluargaModel, KelurahanModel, KecamatanModel, dan KotaModel. Controller sendiri berfungsi untuk melakukan handling request kemudian biasanya memanggil Service untuk melakukan manipulasi data pada model. Query pada database terdapat pada Mapper layer.

Fitur #1 : Tampilkan Penduduk Berdasarkan NIK

- 1) Membuat method pada controller untuk handling request user
- 2) Menggunakan method selectPenduduk(nik) untuk meretrieve data
- 3) Data tersebut disimpan pada variable bertipe data PendudukModel kemudian ditambahkan atributnya pada template html yang dikembalikan oleh method pada controller yang dibuat

Fitur #2 : Tampilkan Keluarga Berdasarkan NKK

- 1) Membuat method pada controller untuk handling request user
- 2) Menggunakan method selectKeluarga(nomer_kk) untuk meretrieve data
- 3) Data tersebut disimpan pada variable bertipe data KeluargaModel kemudian ditambahkan atributnya pada template html yang dikembalikan oleh method pada controller yang dibuat

Fitur #3 : Menambahkan Penduduk Baru

- 1) Membuat method pada controller untuk handling request user. Terdapat dua method untuk fitur ini. Method pertama berfungsi untuk menampilkan halaman form menambah penduduk sementara method kedua berfungsi untuk handling data yang dikirimkan melalui formulir.
- 2) Menerima data yang tersimpan pada objek yang dikirimkan user dengan tipe data pendudukModel
- 3) Menggenerate NIK dengan mengambil kode kecamatan + tanggal lahir yang disesuaikan formatnya dan counter 4 digit di belakang
- 4) Menggunakan method addPenduduk(penduduk) untuk memasukan data
- 5) Menambahkan attribute nik untuk ditampilkan pada halaman sukses

Fitur #4 : Menambahkan Keluarga Baru

- 1) Membuat method pada controller untuk handling request user. Terdapat dua method untuk fitur ini. Method pertama berfungsi untuk menampilkan halaman form menambah keluarga sementara method kedua berfungsi untuk handling data yang dikirimkan melalui formulir.
- 2) Menerima data yang tersimpan pada objek yang dikirimkan user dengan tipe data keluargaModel
- 3) Menggenerate NIK dengan mengambil kode kecamatan + tanggal hari ini yang sudah disesuaikan formatnya dan counter 4 digit di belakang
- 4) Menggunakan method addKeluarga(keluarga) untuk memasukan data
- 5) Menambahkan attribute nkk untuk ditampilkan pada halaman sukses

Fitur #5 : Mengubah Data Penduduk

- 1) Membuat method pada controller untuk handling request user. Terdapat method untuk fitur ini. Method pertama berfungsi untuk menampilkan halaman form menambah penduduk sementara method kedua berfungsi untuk handling data yang dikirimkan melalui formulir.
- 2) Menggunakan method updatePenduduk(penduduk) untuk melakukan update data penduduk dengan data yang dikirimkan oleh user melalui form
- 3) Menampilkan halaman sukses

Fitur #6 : Mengubah Data Keluarga

- 1) Membuat method pada controller untuk handling request user. Terdapat dua method untuk fitur ini. Method pertama berfungsi untuk menampilkan halaman form menambah penduduk sementara method kedua berfungsi untuk handling data yang dikirimkan melalui formulir.

- 2) Menggunakan method `updateKeluarga(keluarga)` untuk melakukan update data penduduk dengan data yang dikirimkan oleh user melalui form
- 3) Menampilkan halaman sukses

Fitur #7 : Mengubah Status Kematian

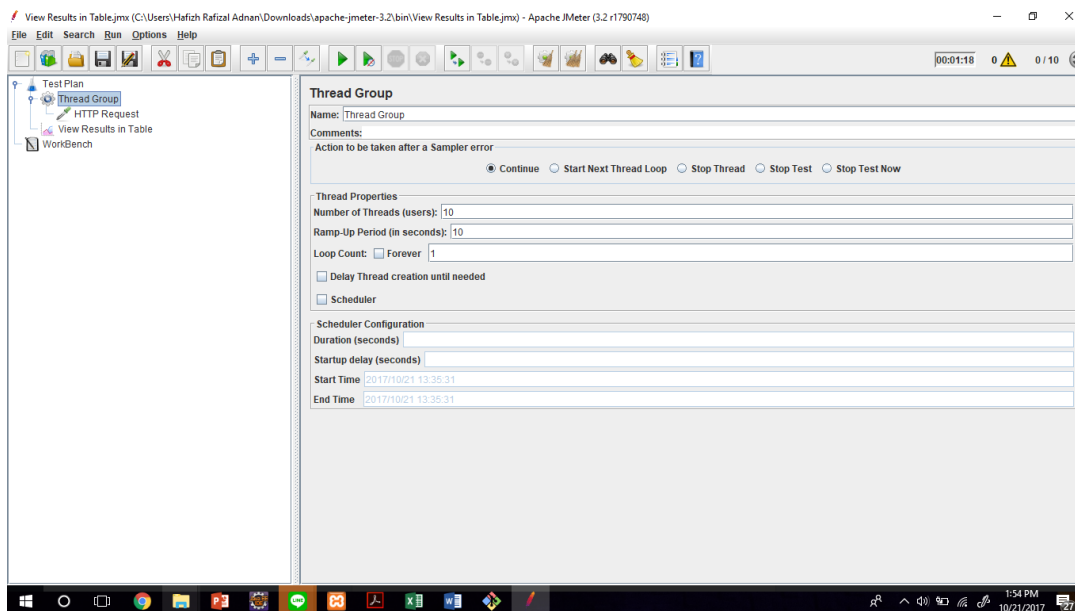
- 1) Membuat method pada controller untuk handling request user
- 2) MENGGUnakan method `deactivatePenduduk(nik)` untuk mematikan penduduk dan mengeksekusi method `deactivateKeluarga(nkk)` jika semua anggota di keluarga telah meninggal
- 3) Menampilkan halaman sukses

Fitur #8 : Cari Penduduk

- 1) Membuat method pada controller untuk handling request user
- 2) Terdapat 3 tahap dimana pertama melakukan retrieve data kota, kemudian data kecamatan berdasarkan id kota, kemudian kelurahan berdasarkan id_kecamatan dan penduduk berdasarkan id_kelurahan
- 3) Dibuat juga query menentukan **penduduk termuda dan tertua**
- 4) Data tersebut disimpan pada variable bertipe data `PendudukModel` kemudian ditambahkan atributnya pada template html yang dikembalikan oleh method pada controller yang dibuat

Stress Testing SIDUK DKI

Sample Fitur : Cari Penduduk Berdasarkan NIK



Hasil sebelum optimasi

View Results in Table.jmx (C:\Users\Hafizh Rafizal Adnan\Downloads\apache-jmeter-3.2\bin\View Results in Table.jmx) - 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:
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	13:49:52.536	Thread Group 1_	HTTP Request	71899	✓	3345	147	71899	2
2	13:49:55.540	Thread Group 1_	HTTP Request	68897	✓	3345	147	68897	1
3	13:49:53.538	Thread Group 1_	HTTP Request	72960	✓	3345	147	72959	1
4	13:49:54.538	Thread Group 1_	HTTP Request	71960	✓	3345	147	71959	1
5	13:49:56.538	Thread Group 1_	HTTP Request	71730	✓	3345	147	71730	0
6	13:49:57.539	Thread Group 1_	HTTP Request	70730	✓	3345	147	70730	1
7	13:49:59.539	Thread Group 1_	HTTP Request	70311	✓	3345	147	70311	1
8	13:50:00.538	Thread Group 1_	HTTP Request	70335	✓	3345	147	70335	0
9	13:50:01.539	Thread Group 1_	HTTP Request	69334	✓	3345	147	69334	1
10	13:49:58.548	Thread Group 1_	HTTP Request	72326	✓	3345	147	72326	1

☐ Scroll automatically? ☐ Child samples? No of Samples 10 Latest Sample 72326 Average 71048 Deviation 1265

1:51 PM 10/21/2017

Hasil setelah optimasi

View Results in Table.jmx (C:\Users\Hafizh Rafizal Adnan\Downloads\apache-jmeter-3.2\bin\View Results in Table.jmx) - 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:
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	13:59:27.934	Thread Group 1_	HTTP Request	937	✓	3345	147	937	2
2	13:59:28.938	Thread Group 1_	HTTP Request	875	✓	3345	147	875	1
3	13:59:29.938	Thread Group 1_	HTTP Request	1279	✓	3345	147	1279	0
4	13:59:30.937	Thread Group 1_	HTTP Request	1309	✓	3345	147	1309	1
5	13:59:31.937	Thread Group 1_	HTTP Request	1250	✓	3345	147	1250	1
6	13:59:32.937	Thread Group 1_	HTTP Request	1192	✓	3345	147	1192	1
7	13:59:33.938	Thread Group 1_	HTTP Request	1043	✓	3345	147	1043	1
8	13:59:34.938	Thread Group 1_	HTTP Request	929	✓	3345	147	929	1
9	13:59:35.939	Thread Group 1_	HTTP Request	1013	✓	3345	147	1013	0
10	13:59:36.939	Thread Group 1_	HTTP Request	954	✓	3345	147	954	1

☐ Scroll automatically? ☐ Child samples? No of Samples 10 Latest Sample 954 Average 1078 Deviation 155

1:59 PM 10/21/2017

Hasil optimasi. Adapun yang dilakukan adalah **penambahan primary key pada database dan indexing pada kolom yang sering menjadi referensi pada aplikasi**. Dari hasil testing tersebut dapat disimpulkan bahwa optimasi berhasil mempercepat proses hamper 50x sehingga saat ini aplikasi SIDUK DKI menjadi lebih cepat dan reliable. Testing sengaja saya buat dengan thread user 10 karena saat saya melakukan stress testing sebelum optimasi dengan thread user yang banyak aplikasi JMeter berhenti secara tiba-tiba.