

Fitur 1

NIK

Lihat

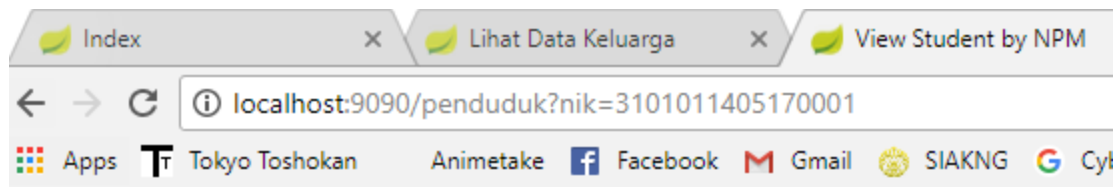
NKK

Lihat

Fitur ini dimulai dari halaman index. Seperti gambar di soal, akan muncul 2 form pada halaman index yaitu "NIK" dan "NKK". Fitur ini adalah yang "NIK"

Halaman index memiliki alamat "/"

Fitur ini menampilkan data dengan alamat " <http://localhost:9090/penduduk?nik={nik}> "



NIK = 3101011405170001

Nama = Heru Haryanto

Tanggal Lahir = 2017-05-14

Alamat = Ds. Adisumarmo No. 43

RT/RW = 79/25

Kelurahan/Desa = PULAU TIDUNG

Kota = KABUPATEN KEPULAUAN SERIBU

Golongan Darah = O-

Agama = Islam

Status Perkawinan = Belum Kawin

Pekerjaan = BELUM/TIDAK BEKERJA

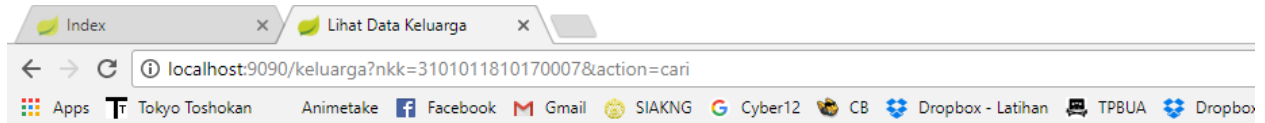
Kewarganegaraan = WNI

Status Kematian = Hidup

Fitur ini selesai dengan mengambil data dari beberapa table seperti penduduk, keluarga, kelurahan, kecamatan, dan kota.

Fitur 2

Fitur ini dimulai dari halaman index, yaitu form kedua setelah form fitur pertama. Fitur ini akan menampilkan data tentang keluarga di atas kiri dan menampilkan penduduk dari keluarga tersebut dalam bentuk datables. Fitur ini mirip fitur sebelumnya, yaitu mengambil dari seluruh tabel yang ada. Perbedaan dari fitur sebelumnya adalah fitur ini mengambil penduduk dalam bentuk "List" agar dapat menampilkan penduduk lebih dari 1. Berikut adalah contoh tampilan fitur ini.



Data Penduduk

NKK = 63486

Alamat = ZXC

RT/RW = 2/3

Kelurahan/Desa = PULAU TIDUNG

Kecamatan = KEPULAUAN SERIBU SELATAN

Kota = KABUPATEN KEPULAUAN SERIBU

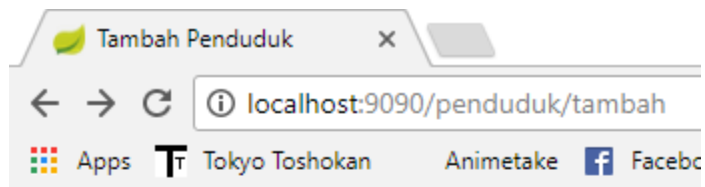
Show entries

No.	Nama Lengkap	
1	ASDFGH	3101015990130014
2	qwe	3101015996050012

Showing 1 to 2 of 2 entries

Fitur 3

Berikut adalah contoh tampilan dari fitur 3.



Tambah Penduduk

Nama

Tempat_lahir

Tanggal Lahir

Golongan Darah

Agama

Status Perkawinan

Pekerjaan

Kewarganegaraan

Status Kematian

Id Keluarga

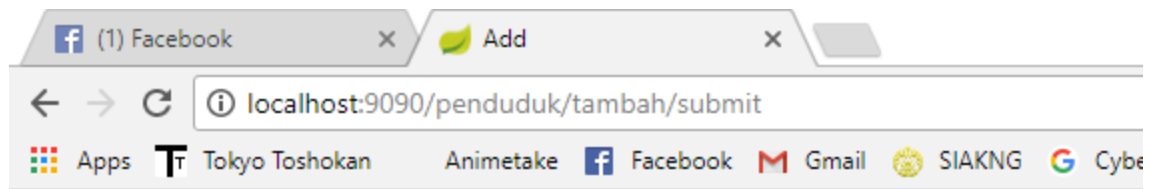
Status Dalam Keluarga

Jenis Kelamin

Fitur ini dapat menambahkan penduduk baru. Fitur ini akan menjalankan query insert dan akan dialihkan ke halaman add-penduduk-success.html jika berhasil menambahkan. Dalam fitur ini belum ada fitur validasi.

Fitur ini juga akan menjalankan fungsi “construct nik” pada controller untuk membuat suatu NIK. Fungsi ini akan mencari NIK min dan NIK max untuk membuat NIK yang paling baru.

Berikut ini adalah tampilan jika berhasil menambah penduduk.

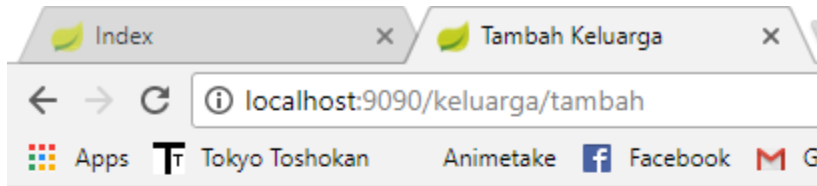


Sukses!

Penduduk dengan NIK = 3101015990130015 berhasil ditambahkan

Fitur 4

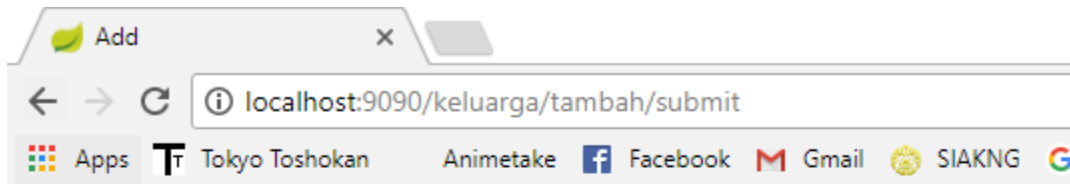
Berikut adalah contoh tampilan dari fitur 4.



Tambah Keluarga

alamat
RT
RW
Kelurahan/Desa

Seperti fitur sebelumnya, fitur ini akan melakukan query insert. Tetapi ada sedikit perbedaan dimana fitur ini akan membuat nkk baru dengan fungsi “constructnkk”. Dalam fitur ini tidak ditampilkan kota/kecamatan karena dalam keluarga hanya disimpan data id_kelurahan saja. Berikut ini adalah tampilannya jika berhasil menambahkan keluarga baru.

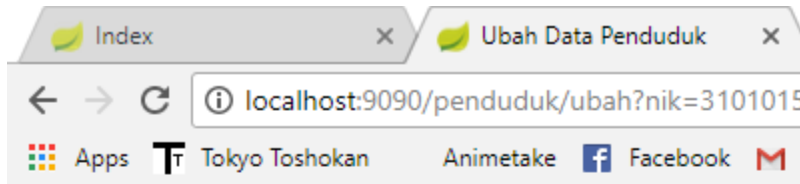


Sukses!

Keluarga dengan NKK 3101012210170001 berhasil ditambahkan

Fitur 5

Berikut ini adalah contoh tampilan dari fitur 5.



Update Penduduk

Nama	<input type="text" value="ASDFGH"/>
Tempat_lahir	<input type="text" value="ASDFG"/>
Tanggal Lahir	<input type="text" value="1990-05-13"/>
Golongan Darah	<input type="text" value="AB"/>
Agama	<input type="text" value="babilonia"/>
Status Perkawinan	<input type="text" value="Belum Kawin"/>
Pekerjaan	<input type="text" value="Nganggur"/>
Kewarganegaraan	<input type="text" value="indonesia"/>
Status Kematian	<input type="text" value="Wafat"/>
Id Keluarga	<input type="text" value="63486"/>
Status Dalam Keluarga	<input type="text" value="budak"/>
<input type="button" value="Simpan"/>	

Fitur ini akan mengambil data dari database tentang nik yang ingin diubah. Fitur ini dapat diakses dengan format alamat “localhost:9090/penduduk/ubah?nik={nik}”

Fitur ini mengambil data dari table penduduk untuk ditampilkan.

Jika tanggal lahir diubah, maka nik akan diupdate (bukan dibuat baru) sehingga tanggal pembuatan masih sama. (asumsi NIK menggunakan **tanggal pembuatan, bukan tanggal perubahan**).

Fitur ini akan menjalankan query update jika menekan tombol simpan dan dialihkan ke halaman sukses update (update-penduduk-success.html). Berikut adalah contoh tampilan suksesnya

Sukses!

Penduduk dengan NIK = 3101015990130014 berhasil diubah

Fitur 6

Berikut adalah contoh tampilan dari fitur ubah keluarga.

Ubah Data Keluarga

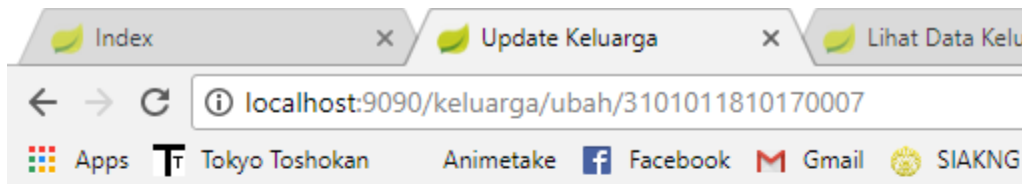
Alamat

RT

RW

Kelurahan/Desa

Fitur ini dapat diakses dengan alamat “localhost:9090/keluarga/ubah”. Fitur ini akan mengambil data dari table keluarga dan melakukan query update untuk mengubah data. NKK tidak akan dibuat baru jika kelurahan tidak diubah oleh pengguna. Jika kelurahan diubah, maka NKK juga berubah dan penduduk yang ada di keluarga tersebut juga berubah NIKnya. Berikut ini adalah contoh tampilan sukses jika keluarga berhasil diubah.

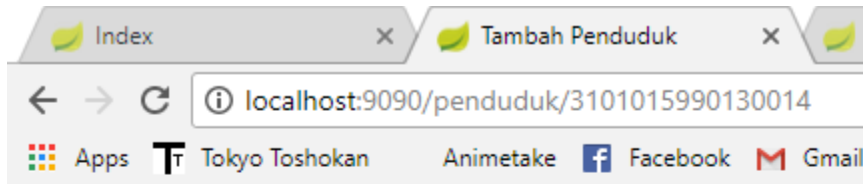


Sukses!

Keluarga dengan NKK = 3101011810170007 berhasil diubah

Fitur 7

Berikut adalah contoh tampilan dari fitur ini.

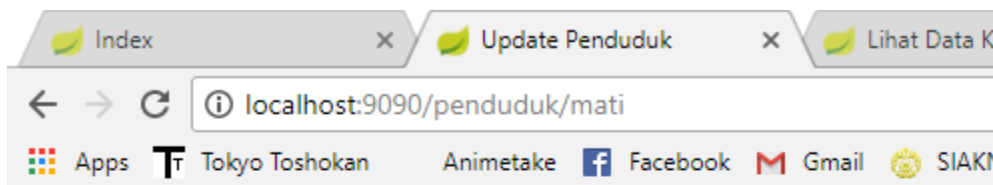


Ubah Status Kematian

NIK = 3101015990130014

Status Kematian

Didalam fitur ini, hanya terdapat field NIK yang tidak dapat diubah. Selain itu terdapat juga status kematian yang dapat diubah. Jika ditekan tombol simpan akan keluar halaman sebagai berikut:



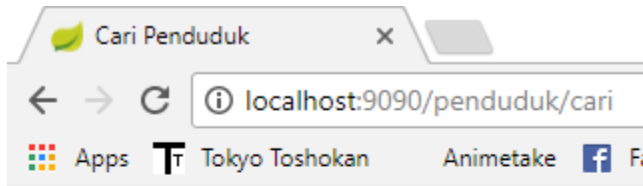
Sukses!

Penduduk dengan NIK = 3101015990130014 berhasil diubah

Jika mengubah status kematian menjadi mati, didalam controller akan memeriksa apakah seluruh penduduk didalam keluarga penduduk ini sudah mati. Jika iya akan merubah status nkk menjadi tidak berlaku.

Fitur 8

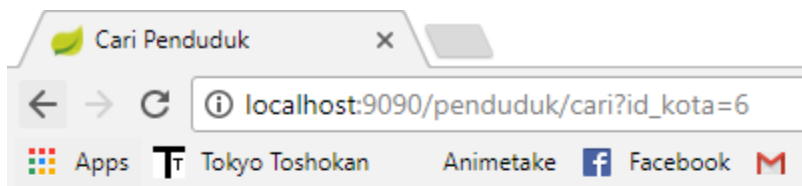
Berikut adalah contoh tampilan awal fitur ini.



Search Penduduk

Kota

Jika kota telah dipilih, akan menampilkan tampilan seperti ini.

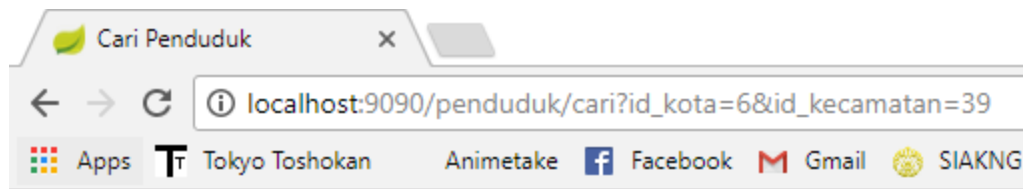


Search Penduduk

Kota

Kecamatan

Kota akan didisable dan tidak akan dimasukkan valuenya kedalam controller. Oleh karena itu pada halaman ini ada input type hidden untuk mengisi id kota.



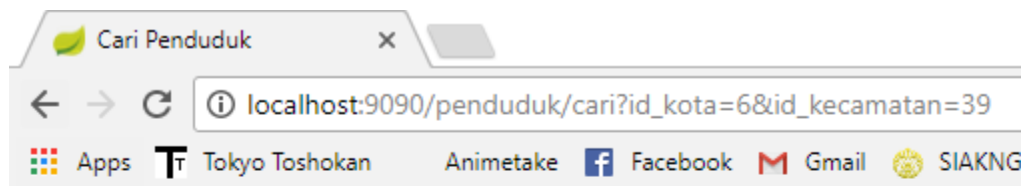
Search Penduduk

Kota

Kecamatan

Kelurahan/Desa

Sama seperti sebelumnya, kecamatan akan menjadi *disabled* dan akan dibuat input type hidden tentang kecamatan yang berisi value `id_kecamatan`.



Search Penduduk

Kota

Kecamatan

Kelurahan/Desa

Jika kelurahan dipilih, tidak langsung dialihkan ke controller, tetapi harus menekan tombol lihat untuk ditampilkan menuju halaman yang berisikan datables. Berikut adalah contoh tampilan datables fitur ini.

Lihat Data Keluarga

localhost:9090/penduduk/cari?id_kota=6&id_kecamatan=39&id_kelurahan=241&action=Lihat

Apps Tokyo Toshokan Animatek Facebook Gmail SIAKNG Cyber12 CB Dropbox - Latihan TPBUA Dropbox - Public WindowBuilder Pro E JawzTranslations

Daftar Penduduk

Show 10 entries Search:

NIK	Nama	Jenis Kelamin
3175010102890001	Maras Kurniawan	Perempuan
3175010103150001	Tri Ajimat Mangunsong	Perempuan
3175010103860001	Mumpuni Kurniawan	Perempuan
3175010104150002	Bagas Siregar	Perempuan
3175010104690001	Martana Wibowo	Perempuan
3175010105160002	Darmanto Thamrin	Perempuan
3175010106140001	Elon Maryadi	Perempuan

Query dari fitur ini menggunakan select biasa, dengan tambahan IN untuk memilih beberapa NIK yang ada. Fitur ini hanya menggunakan 2 halaman yaitu cari-penduduk.html dan cari-penduduk-success.html.

Fungsi dalam *controller* yang digunakan dalam fitur ini hanya 1 saja, cukup dengan menerima parameter dan melakukan if. Jika value masih null, maka akan di add model dengan attribute null dan akan divalidasi di html nya.

Optimasi Database

Optimasi dilakukan dengan memberikan index kepada id dari setiap table, nik, dan nkk. Selain itu, id pada tiap table juga diubah menjadi primary key dan disetting menjadi auto increment. Berikut ini adalah hasil uji coba dengan menggunakan jmeter.

HTTP Request Configuration:

- Name: HTTP Request
- Comments:
- Basic Tab:
 - Web Server:
 - Protocol [http]:
 - Server Name or IP: localhost
 - Port Number:
 - HTTP Request:
 - Method: GET
 - Path: http://localhost:9090/penduduk?nik=3101011405170001
 - Content encoding:
 - ☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers
- Parameters Tab:
 - Send Parameters With the Request:
 - Table with columns: Name, Value, Encode?, Include Equals?

View Results in Table:

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☐ Successes ☒ Configuration

Sample #	Start Time	Thread Name	Label	Sample Time	Status	Bytes	Sent Bytes	Latency	Connect Time
1	14:45:18.414	Thread Group ...	HTTP Request	6072	✓	802	147	6071	202
2	14:45:18.452	Thread Group ...	HTTP Request	6046	✓	802	147	6046	414
3	14:45:18.338	Thread Group ...	HTTP Request	6167	✓	802	147	6167	519
4	14:45:18.335	Thread Group ...	HTTP Request	6194	✓	802	147	6194	461
5	14:45:18.443	Thread Group ...	HTTP Request	6093	✓	802	147	6093	340
6	14:45:18.347	Thread Group ...	HTTP Request	6240	✓	802	147	6240	501
7	14:45:18.407	Thread Group ...	HTTP Request	6439	✓	802	147	6439	2155
8	14:45:18.739	Thread Group ...	HTTP Request	6118	✓	802	147	6118	1843
9	14:45:18.758	Thread Group ...	HTTP Request	6126	✓	802	147	6126	1817
10	14:45:18.351	Thread Group ...	HTTP Request	6552	✓	802	147	6552	2194
11	14:45:18.456	Thread Group ...	HTTP Request	6495	✓	802	147	6495	406
12	14:45:17.996	Thread Group ...	HTTP Request	7160	✓	802	147	7160	2578
13	14:45:18.758	Thread Group ...	HTTP Request	6408	✓	802	147	6408	1817
14	14:45:18.740	Thread Group ...	HTTP Request	6452	✓	802	147	6452	1841
15	14:45:18.841	Thread Group ...	HTTP Request	6390	✓	802	147	6390	1707
16	14:45:18.042	Thread Group ...	HTTP Request	7198	✓	802	147	7198	2500
17	14:45:18.459	Thread Group ...	HTTP Request	6809	✓	802	147	6807	2114
18	14:45:18.044	Thread Group ...	HTTP Request	7299	✓	802	147	7299	2494
19	14:45:18.720	Thread Group ...	HTTP Request	6635	✓	802	147	6635	1866
20	14:45:18.037	Thread Group ...	HTTP Request	7342	✓	802	147	7342	2511
21	14:45:18.217	Thread Group ...	HTTP Request	7176	✓	802	147	7176	2323
22	14:45:18.764	Thread Group ...	HTTP Request	6652	✓	802	147	6652	1810
23	14:45:18.763	Thread Group ...	HTTP Request	6660	✓	802	147	6660	1811
24	14:45:18.455	Thread Group ...	HTTP Request	6987	✓	802	147	6987	2105
25	14:45:18.040	Thread Group ...	HTTP Request	7417	✓	802	147	7417	2503
26	14:45:18.026	Thread Group ...	HTTP Request	7454	✓	802	147	7454	2530
27	14:45:18.346	Thread Group ...	HTTP Request	7153	✓	802	147	7152	2228
28	14:45:18.027	Thread Group ...	HTTP Request	7497	✓	802	147	7497	2527
29	14:45:18.441	Thread Group ...	HTTP Request	7112	✓	802	147	7112	2115

☐ Scroll automatically? ☐ Child samples? No of Samples 1000 Latest Sample 23499 Average 15260 Deviation 4723

Tugas1.jmx (C:\Eclipse Workspace\Tugas1\APAP\Tugas1.jmx) - Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

00:00:24 0 0 / 1000

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(
972	14:45:18.181	Thread Group ...	HTTP Request	23250	✓	802	147	23250	4845
973	14:45:18.424	Thread Group ...	HTTP Request	23011	✓	802	147	23011	4602
974	14:45:18.423	Thread Group ...	HTTP Request	23018	✓	802	147	23018	4604
975	14:45:18.273	Thread Group ...	HTTP Request	23185	✓	802	147	23185	3778
976	14:45:18.256	Thread Group ...	HTTP Request	23199	✓	802	147	23199	3797
977	14:45:18.403	Thread Group ...	HTTP Request	23074	✓	802	147	23074	3691
978	14:45:18.416	Thread Group ...	HTTP Request	23063	✓	802	147	23063	4609
979	14:45:18.400	Thread Group ...	HTTP Request	23089	✓	802	147	23089	4625
980	14:45:18.423	Thread Group ...	HTTP Request	23081	✓	802	147	23081	4603
981	14:45:18.265	Thread Group ...	HTTP Request	23243	✓	802	147	23243	3842
982	14:45:18.400	Thread Group ...	HTTP Request	23124	✓	802	147	23124	4625
983	14:45:18.395	Thread Group ...	HTTP Request	23132	✓	802	147	23132	3636
984	14:45:18.422	Thread Group ...	HTTP Request	23115	✓	802	147	23115	3683
985	14:45:18.296	Thread Group ...	HTTP Request	23260	✓	802	147	23260	3757
986	14:45:18.421	Thread Group ...	HTTP Request	23140	✓	802	147	23140	3649
987	14:45:18.282	Thread Group ...	HTTP Request	23297	✓	802	147	23297	3787
988	14:45:18.303	Thread Group ...	HTTP Request	23300	✓	802	147	23300	4722
989	14:45:18.298	Thread Group ...	HTTP Request	23308	✓	802	147	23308	3804
990	14:45:18.402	Thread Group ...	HTTP Request	23211	✓	802	147	23211	4624
991	14:45:18.422	Thread Group ...	HTTP Request	23196	✓	802	147	23196	3717
992	14:45:18.421	Thread Group ...	HTTP Request	23201	✓	802	147	23201	3643
993	14:45:18.420	Thread Group ...	HTTP Request	23209	✓	802	147	23209	3609
994	14:45:18.409	Thread Group ...	HTTP Request	23227	✓	802	147	23227	3691
995	14:45:18.408	Thread Group ...	HTTP Request	23232	✓	802	147	23232	4618
996	14:45:18.352	Thread Group ...	HTTP Request	23294	✓	802	147	23294	3796
997	14:45:18.303	Thread Group ...	HTTP Request	23350	✓	802	147	23350	4723
998	14:45:18.298	Thread Group ...	HTTP Request	23358	✓	802	147	23358	3736
999	14:45:18.291	Thread Group ...	HTTP Request	23372	✓	802	147	23372	4734
1000	14:45:18.176	Thread Group ...	HTTP Request	23499	✓	802	147	23499	3920

☐ Scroll automatically? ☐ Child samples? No of Samples 1000 Latest Sample 23499 Average 15260 Deviation 4723

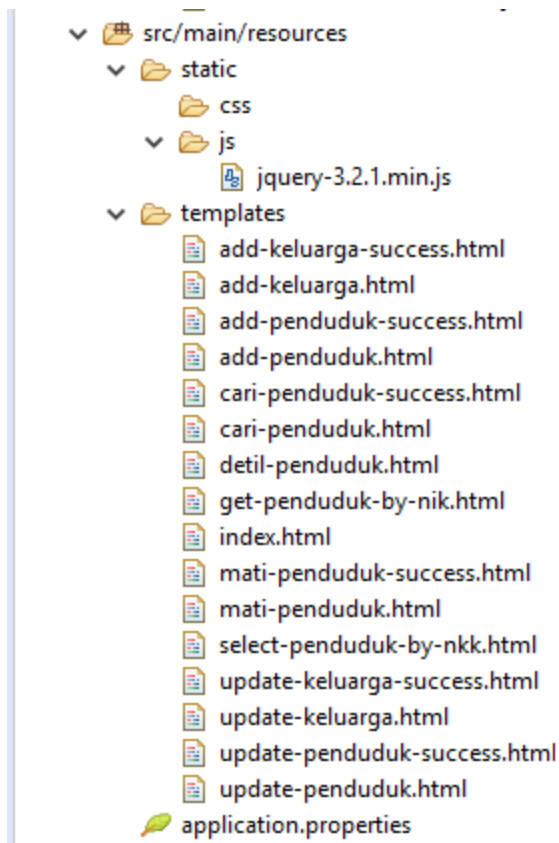
Gambar yang pertama merupakan waktu awal diakses. Waktu respon awal sebesar 6 detik sedangkan ketika sudah menumpuk sekitar 26 detik. Dari hasil ini, dapat dilihat bahwa response time masih dapat dianggap normal meski diakses menumpuk oleh 1000 orang. Hal ini karena sudah diberikan index pada bagian table yang tadi sudah disebutkan sebelumnya.

Menurut eksperimen yang penulis lakukan, sistem sudah cukup *reliable* untuk skala ribuan orang, meski jika banyak pengguna sekaligus sangat memberatkan sistem, tetapi sistem masih dapat memberikan respon dengan baik.

Struktur Project

Berikut ini adalah gambar struktur dari proyek yang penulis buat.

- ▼ Tugas1APAP [boot] [devtools]
 - ▼ src/main/java
 - ▼ com.example
 - > Tugas1ApapApplication.java
 - ▼ com.example.controller
 - > PendudukController.java
 - ▼ com.example.dao
 - > PendudukMapper.java
 - ▼ com.example.model
 - > KecamatanModel.java
 - > KeluargaModel.java
 - > KelurahanModel.java
 - > KotaModel.java
 - > PendudukModel.java
 - ▼ com.example.service
 - > KecamatanService.java
 - > KecamatanServiceDatabase.java
 - > KeluargaService.java
 - > KeluargaServiceDatabase.java
 - > KelurahanService.java
 - > KelurahanServiceDatabase.java
 - > KotaService.java
 - > KotaServiceDatabase.java
 - > PendudukService.java
 - > PendudukServiceDatabase.java



Dari gambar diatas, dapat dilihat bahwa setiap table yang ada dibuatkan service. Untuk mappernya hanya 1 saja (disatukan semua). Untuk Controller hanya ada 1 sebagai kendali utama. Untuk model tentunya terdapat 5 buah model sesuai dengan jumlah tabelnya. Kelima model tersebut tentunya merepresentasikan 5 tabel yang ada.

Untuk file html, setiap halaman sukses memiliki halaman tersendiri. Proyek yang dibuat belum menggunakan css, tetapi sudah menggunakan databales untuk seluruh bagian yang dapat menggunakan databales, seperti pada fitur cari dan fitur lihat penduduk berdasarkan nkk.

Alur kerja fitur ini dimulai dari halaman html dan dilanjutkan menuju controller. Didalam controller jika membutuhkan data dari database akan memanggil service. Service yang ada sesuai dengan jumlah table yaitu 5 buah. Jika ingin mengambil data dari table penduduk maka harus memanggil pendudukservice, jika table keluarga maka memanggil keluargaservice, dan seterusnya. Setelah itu masing-masing service akan memanggil pendudukMapper untuk menjalankan query jika ingin mengambil data.

