

Write Up

Proses Pengembangan Tugas 1

Pertama dan yang paling utama saya membaca soal terlebih dahulu mulai dari tujuannya apa dan seberapa kompleks program yang akan dibuat. Ketika sudah selesai, saya memulai dari membuat model yang ada dan disesuaikan dengan kebutuhan dari program. Setelah itu saya mulai membuat frontend berdasarkan apa yang diminta pada soal dan menambahkan sesuai dengan apa yang akan disimpan dan diminta oleh program. Lalu saya membuat dummy data pada front end sekaligus pengimplementasian dependencies yang ada seperti bootstrap, jquery, dan juga datatables.

Ketika front end sudah selesai, saya mulai membentuk controller dimana flow program mulai berjalan dari awal hingga akhir tanpa ada fungsi yang terimplementasi. Selah itu saya baru mencoba bagaimana mengintegrasikan database pada program. Ketika sudah oke, saya mulai mengimplementasikan program satu per satu.

Note: Setiap DAO dikelompokkan berdasarkan akses menuju skemanya. Begitu pula dengan Service nya Controller juga dikelompokkan berdasarkan fungsinya. Dan nama database yang diset ialah **data_penduduk**

List Fitur

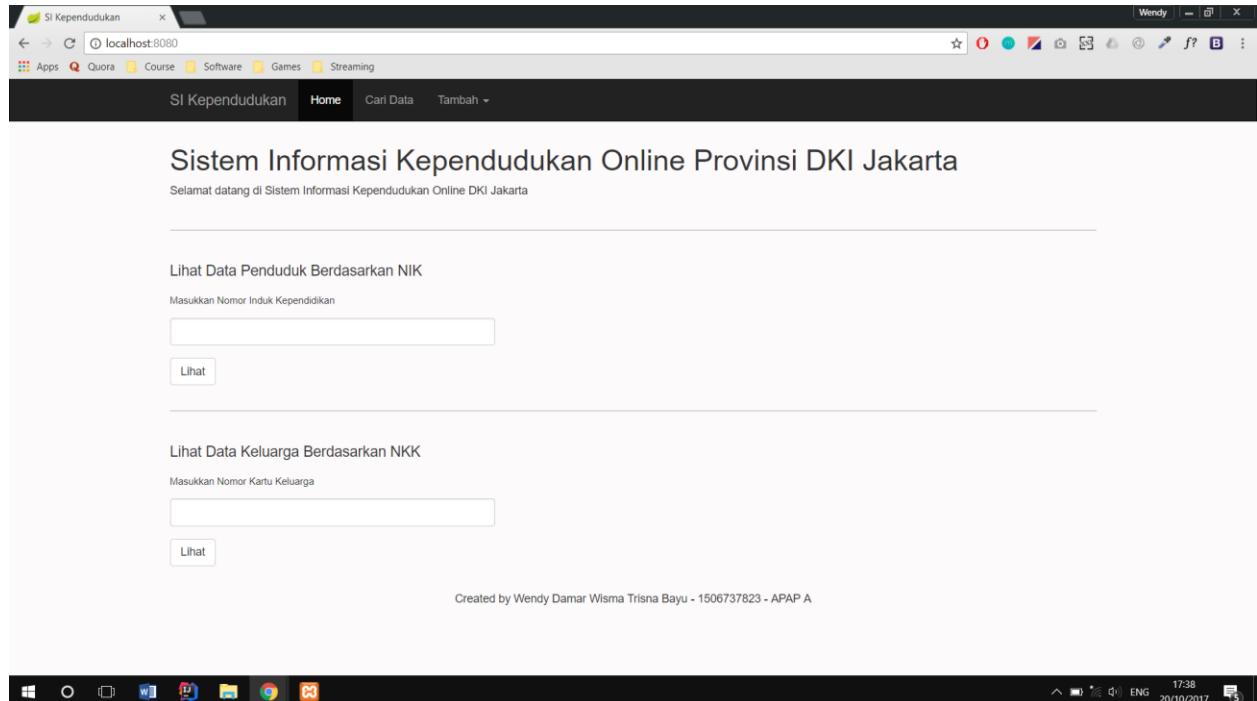
1. Tampilkan Data Penduduk Berdasarkan NIK

Initial Page: /

Form Request: GET, /penduduk

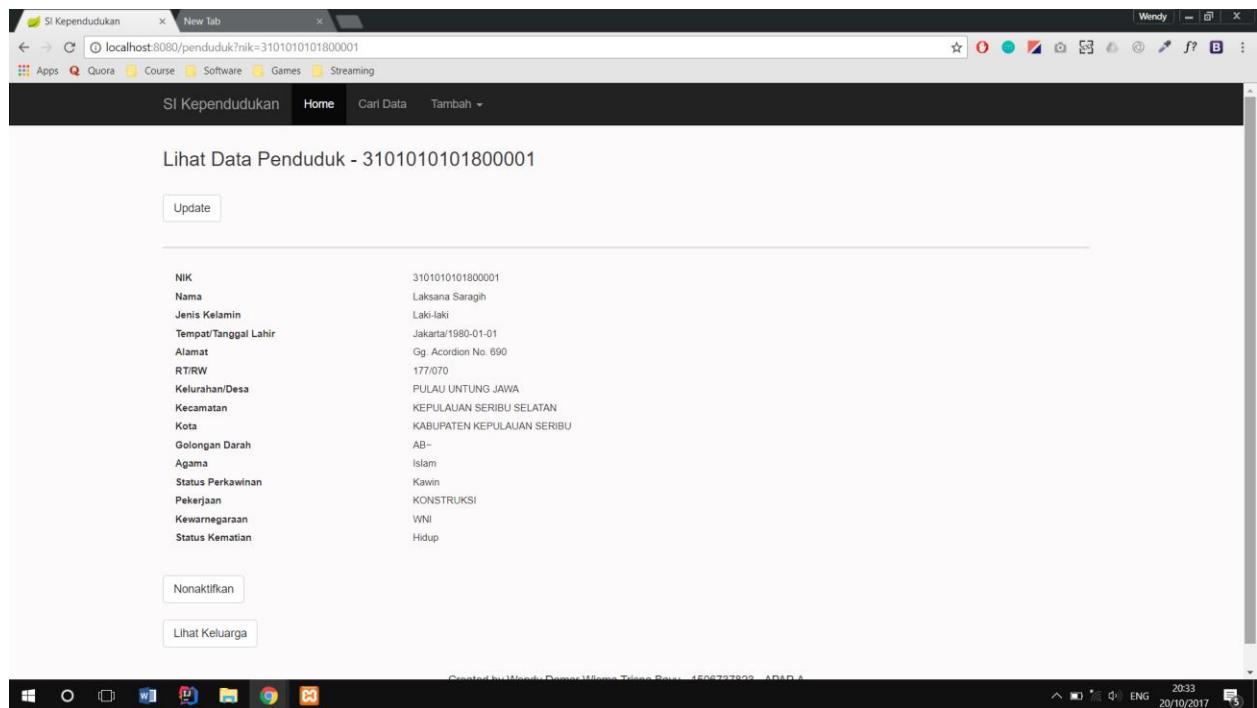
Response Page: /penduduk?nik={NIK}

Pada fitur ini saya menggunakan form get dimana form ini akan mempassing nik menuju controller dan pada controller ia akan menjalankan service. Form ini terdapat pada halaman utama ("/") dimana ia hanya perlu memasukkan input NIK pada kolom lalu tekan lihat. Setelah itu form akan dipassing menuju mapping ("/penduduk") dimana ia memiliki parameter nik sesuai yang telah dimasukkan sebelumnya.



Pada Controller ia akan mengecek apakah nik yang dipassing itu kosong atau tidak. Jika kosong maka Controller akan memberikan halaman error dan jika tidak maka program akan mengambil data penduduknya.

Setelah itu dicek kembali apakah nik tersebut tersedia pada database? Jika tidak Controller akan memberikan halaman error dan jika ada maka aprogram akan lanjut menginisialisasi keluarga, kelurahan, kecamatan, dan kota yang bersangkutan.



Lalu terdapat pengecekan perubahan data jika bernilai Boolean. Contoh jenis kelamin, is_wni, dan juga is_wafat untuk disesuaikan dengan output. Dan pada Controller akan merefer balik menuju halaman view penduduk.

2. Tampilkan Data Keluarga Beserta Daftar Anggotanya Berdasarkan Nomor KK

Initial Page: /

Form Request: GET, /keluarga

Response Page: /keluarga?nkk={NKK}

Pada fitur ini saya menggunakan form get dimana form ini akan mempassing nik menuju controller dan pada controller ia akan menjalankan service. Form ini terdapat pada halaman utama (“/”) dimana ia hanya perlu memasukkan input NKK pada kolom lalu tekan lihat. Setelah itu form akan dipassing menuju mapping (“/keluarga”) dimana ia memiliki parameter nkk sesuai yang telah dimasukkan sebelumnya.

Pada Controller ia akan mengecek apakah nkk yang dipassing itu kosong atau tidak. Jika kosong maka Controller akan memberikan halaman error dan jika tidak maka program akan mengambil data keluarganya.

Lihat Data Keluarga - 3101012708140001

No	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status dalam keluarga	Kewarganegaraan
1	Laksana Saragih	310101010180001	Laki-laki	Jakarta	1980-01-01	Islam	KONSTRUKSI	Kawin	Kepala Keluarga	WNI
2	Vanesa Mulyani	3101014309150002	Perempuan	Jakarta	2015-09-03	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI
3	Tina Astuti	3101016906860001	Perempuan	Jakarta	1986-06-29	Islam	PENATA RAMBUT	Kawin	Istri	WNI

Created by Wendy Damar Wisma Trisna Bayu - 1506737823 - APAP A

Setelah itu dicek kembali apakah nkk tersebut tersedia pada database? Jika tidak Controller akan memberikan halaman error dan jika ada maka program akan lanjut menginisialisasi list penduduk, kelurahan, kecamatan, dan kota yang bersangkutan.

Lalu terdapat pengecekan perubahan data jika bernilai Boolean. Contoh jenis kelamin, is_wni, dan juga is_wafat untuk disesuaikan dengan output. Dan pada Controller akan merefer balik menuju halaman view keluarga.

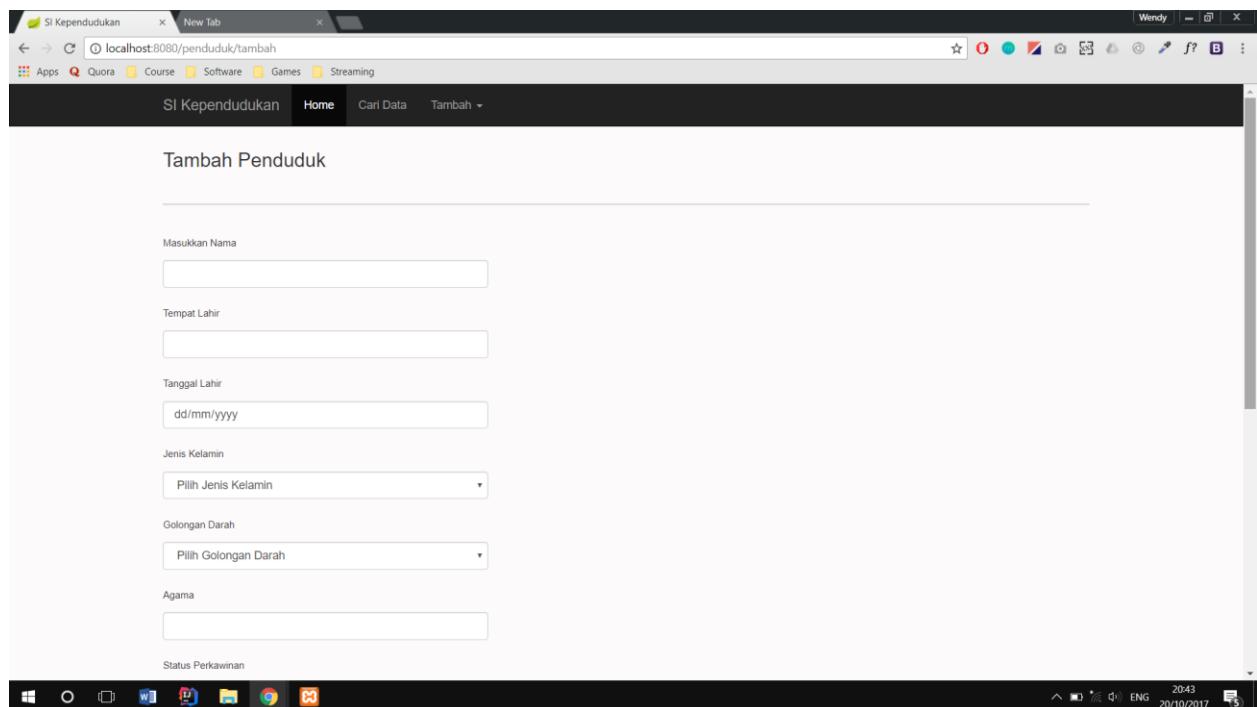
3. Menambahkan Penduduk Baru Sebagai Anggota Keluarga

Initial Page: /penduduk/tambah

Form Request: POST, /penduduk/tambah

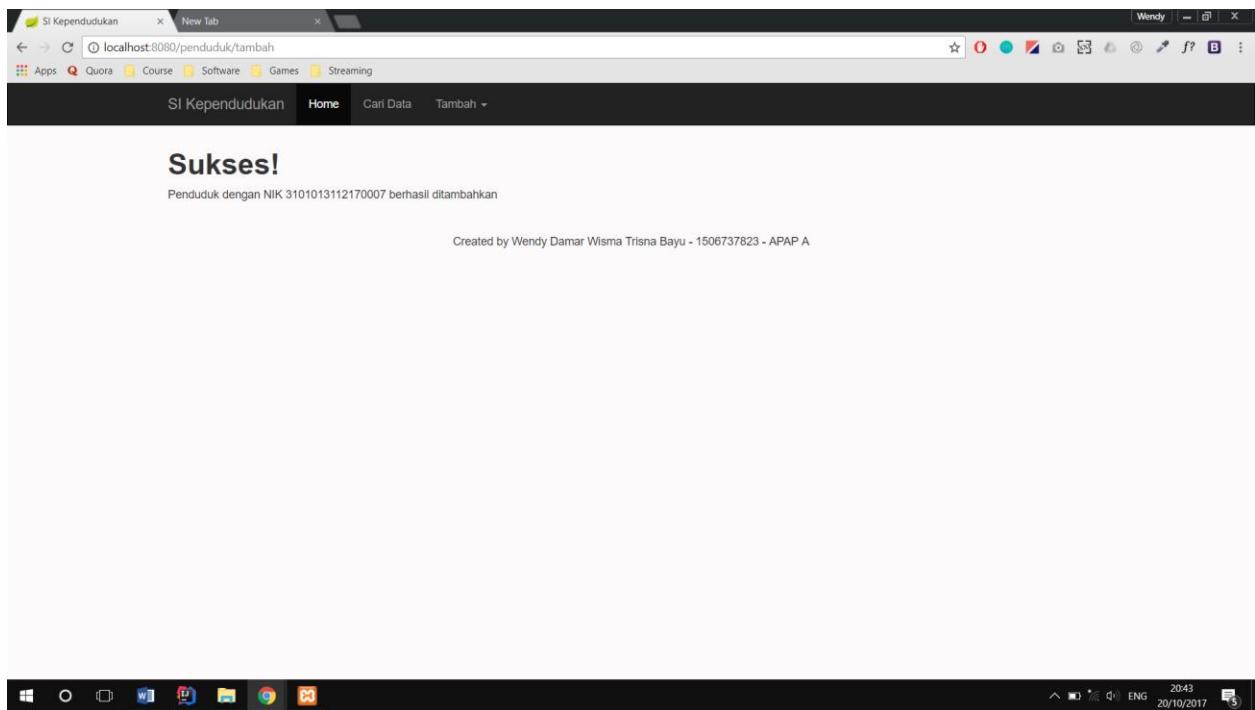
Response Page: /penduduk/tambah

Pada fungsi ini kita akan menuju halaman “/penduduk/tambah” yang telah disediakan pada navigation bar. Lalu jika kita telah mengklik tersebut maka kita akan menuju Controller dimana kita akan menginisialisasi Penduduk terlebih dahulu supaya dapat didefinisikan pada view karena kita akan menggunakan passing variable menggunakan object. Setelah didefinisikan dan di addAttribute, maka Controller akan mengirim kita pada form registrasi penduduk.



Pada form telah disinkronisasi dengan objek Penduduk dan format yang digunakan ialah post dengan action menuju “penduduk/tambah/submit”. Setelah semua data dimasukkan, maka halaman akan mengirim pada Controller.

Pada Controller pertama-tama kita kita menginisialisasi objek penduduk terlebih dahulu. Setelah itu kita cek menggunakan @Valid apakah sudah sesuai dengan syarat dari modelnya tersebut. Jika sudah maka kita akan menginisialisasi Keluarga berdasarkan input data tersebut. Jika tidak ada maka Controller akan mengirimkan halaman error, dan jika tidak maka program akan lanjut menginisialisasi keluarga dan kecamatan untuk mendapatkan kode kecamatan demi meng generate NIK.



Lalu pada mengenerate NIK kita akan mengambil dari kode kecamatan yang telah di substring sebanyak 6 huruf. Lalu mengambil dan mengolah tanggal lahir untuk dimasukkan pada nik. Namun sebelum itu dilakukan string split terlebih dahulu karena format yang diberikan date sedikit berbeda dengan yang diinginkan. Setelah itu diberi pengecekan juga apakah penduduk merupakan laki-laki atau perempuan. Jika perempuan maka pada tanggal ditambah 40. Setelah itu kita mengecek pada database apakah sudah ada nik yang menyerupai dengan nik yang sudah digenerate (sebanyak 12 digit). Jika tidak ada maka kita akan mengenerate 0001 pada 4 digit terakhir, jika tidak maka kita akan menggenerate 4 digit akhir yang terbesar namun ditambah satu. Setelah itu kita menambahkan penduduk dengan

menggunakan addPenduduk dan jika telah berhasil maka kita akan menampilkan halaman konfirmasi bahwa proses tambah penduduk sudah berhasil.

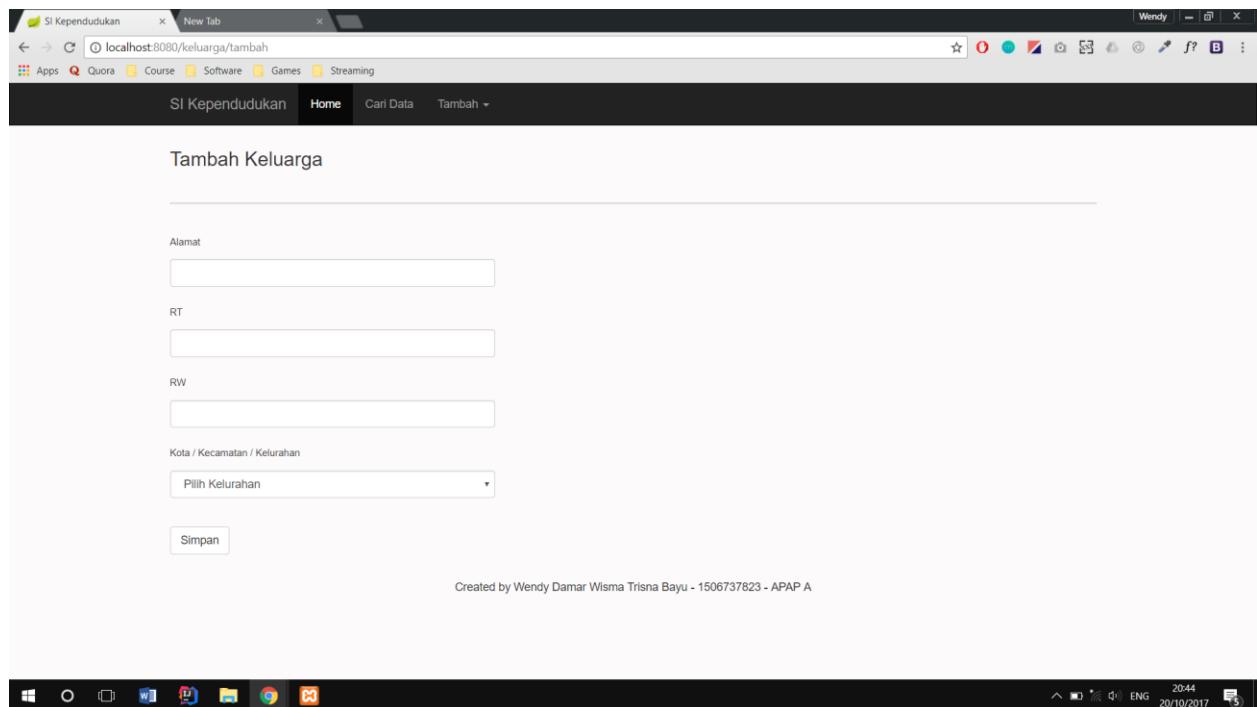
4. Menambahkan Keluarga Baru

Initial Page: /keluarga/tambah

Form Request: POST, /keluarga/tambah

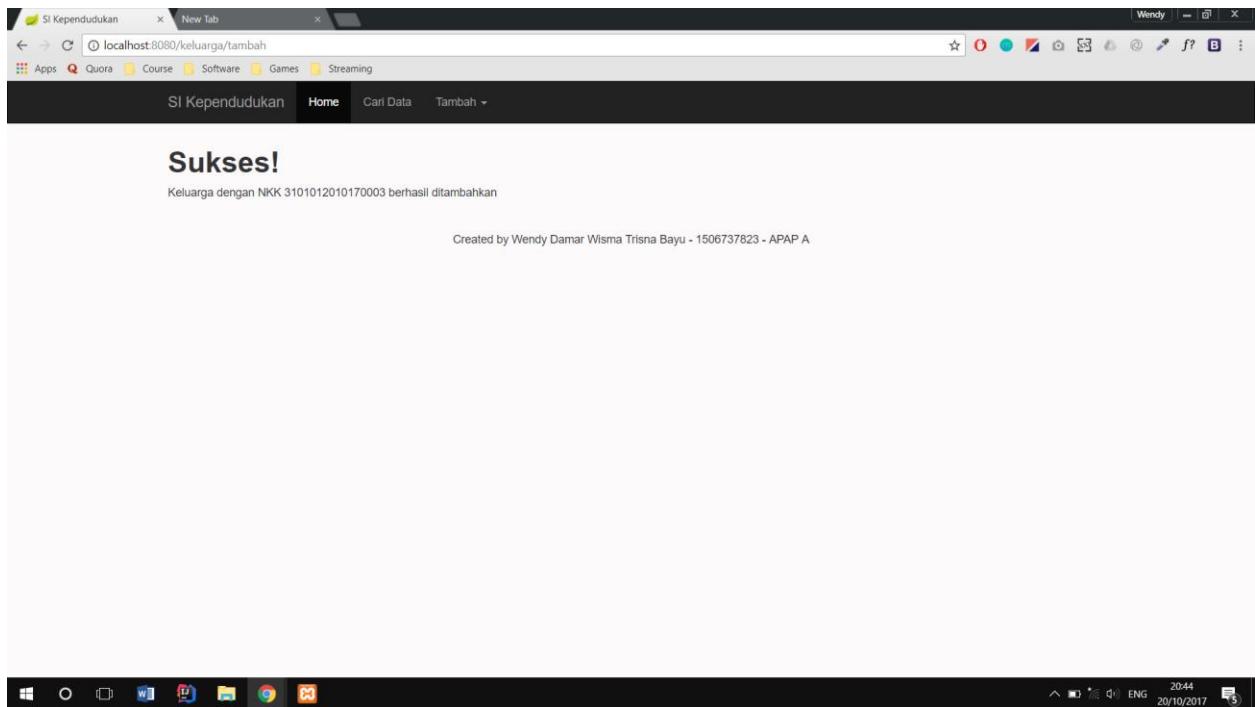
Response Page: /keluarga/tambah

Pada fungsi ini kita akan menuju halaman “/keluarga/tambah” yang telah disediakan pada navigation bar. Lalu jika kita telah mengklik tersebut maka kita akan menuju Controller dimana kita akan menginisialisasi Keluarga terlebih dahulu supaya dapat didefinisikan pada view karena kita akan menggunakan passing variable menggunakan object. Setelah didefinisikan dan di addAttribute, kita juga mendefinisikan list keluarahan yang ada demi untuk mengisi form list kelurahan, kota, dan kecamatan. Dan setelah itu, Controller akan mengirim kita pada form registrasi keluarga.



Pada form telah disinkronisasi dengan objek Penduduk dan format yang digunakan ialah post dengan action menuju “keluarga/tambah/submit”. Setelah semua data dimasukkan, maka halaman akan mengirim pada Controller.

Pada Controller pertama-tama kita kita menginisialisasi objek keluarga terlebih dahulu. Setelah itu kita cek menggunakan @Valid apakah sudah sesuai dengan syarat dari modelnya tersebut. Jika sudah maka kita akan menginisialisasi kelurahan untuk mendapatkan kode kecamatan demi meng generate NKK.



Lalu pada mengenerate NKK kita akan mengambil dari kode kecamatan yang telah di substring sebanyak 6 huruf. Lalu mengambil dan mengolah tanggal saat ini (pengisian) untuk dimasukkan pada NKK. Namun sebelum itu dilakukan string split terlebih dahulu karena format yang diberikan date sedikit berbeda dengan yang diinginkan. Setelah itu kita mengecek pada database apakah sudah ada nkk yang menyerupai dengan nkk yang sudah digenerate (sebanyak 12 digit). Jika tidak ada maka kita akan mengenerate 0001 pada 4 digit terakhir, jika tidak maka kita akan menggenerate 4 digit akhir yang terbesar namun ditambah satu. Setelah itu kita menambahkan keluarga dengan menggunakan addKeluarga dan jika telah berhasil maka kita akan menampilkan halaman konfirmasi bahwa proses tambah keluarga sudah berhasil.

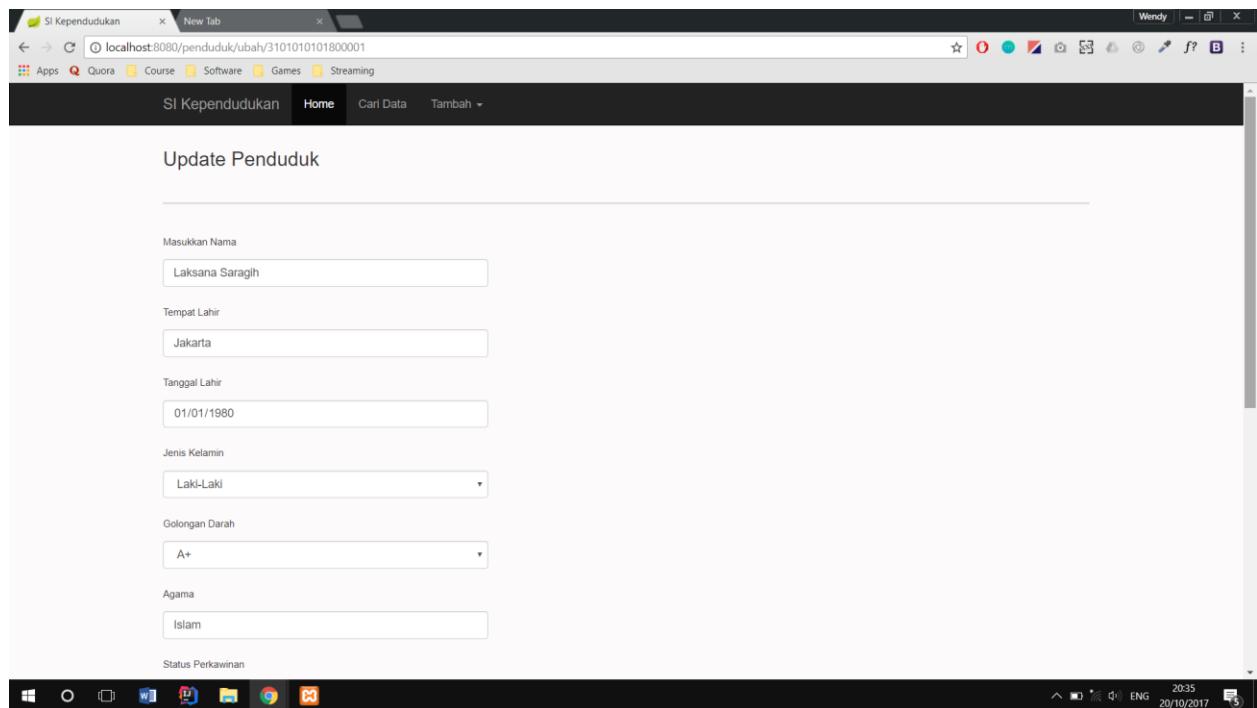
5. Mengubah Data Penduduk

Initial Page: /penduduk/ubah/{NIK}

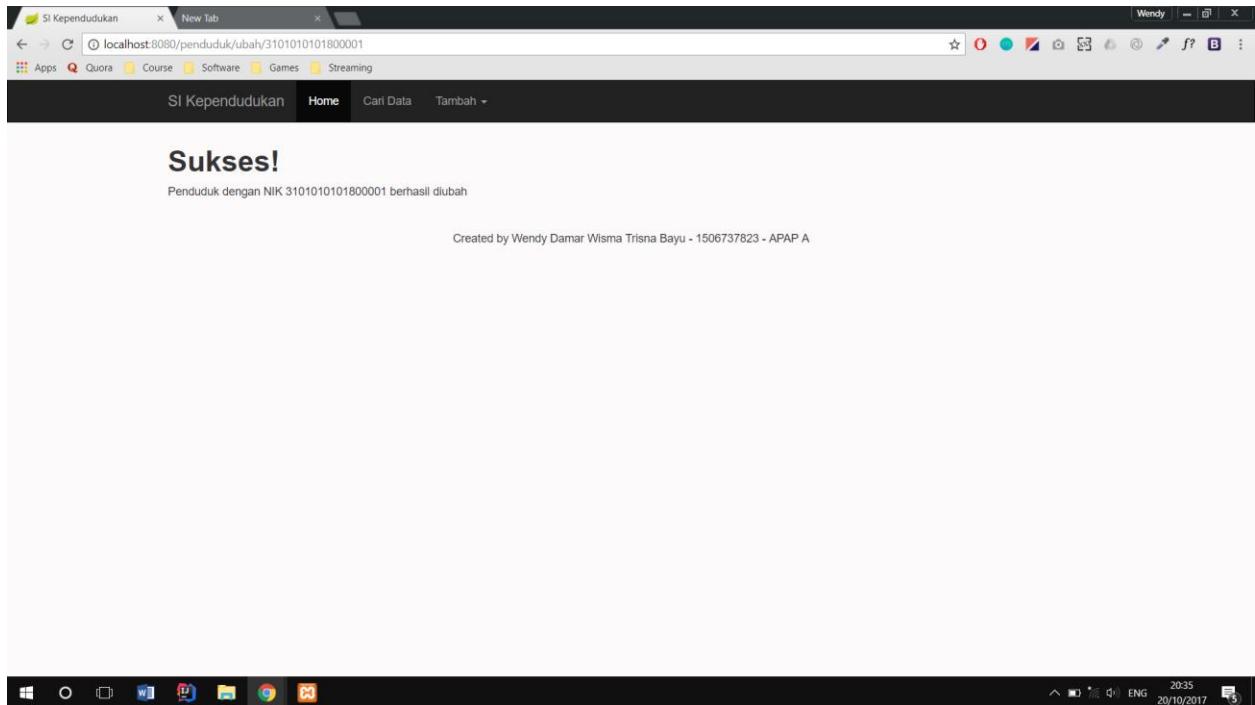
Form Request: POST, /penduduk/ubah/{NIK}

Response Page: /penduduk/ubah/{NIK}

Pada web ini kita dapat mengubah data penduduk pada setiap halaman lihat penduduk dimana dia akan mengeluarkan halaman update (Halaman menuju “penduduk/ubah/{nik}”). Sebelum masuk pada halaman ubah, kita sebelumnya mencari terlebih dahulu apakah nik yang ingin diubah telah terisi atau tidak. Setelah itu kita akan melakukan validasi apakah penduduk telah tersedia dengan nik yang telah diberikan. Jika tidak ada maka akan diberikan halaman bahwa nik tidak ditemukan. Dan jika ada ia akan memberikan halaman update yang isinya kurang lebih seperti registrasi namun bedanya ia actionnya menuju “penduduk/ubah/submit”, menampilkan hasil data sebelumnya pada form, dan mengirimkan nik lama untuk pengecekan.



Lalu jika semua data telah tersubmit, maka akan masuk pada Controller dan dia akan mengecek apakah format yang sudah diinput sesuai dengan format pada model. Setelah itu melakukan validasi apakah ada data yang dapat merubah NIK. Jika tidak ada maka ia akan menginput NIK berdasarkan nik lama. Dan jika tidak ia akan meng generate yang baru sesuai dengan ketentuannya. Dan jika sudah maka iakan meng update dan menampilkan halaman apabila sukses.



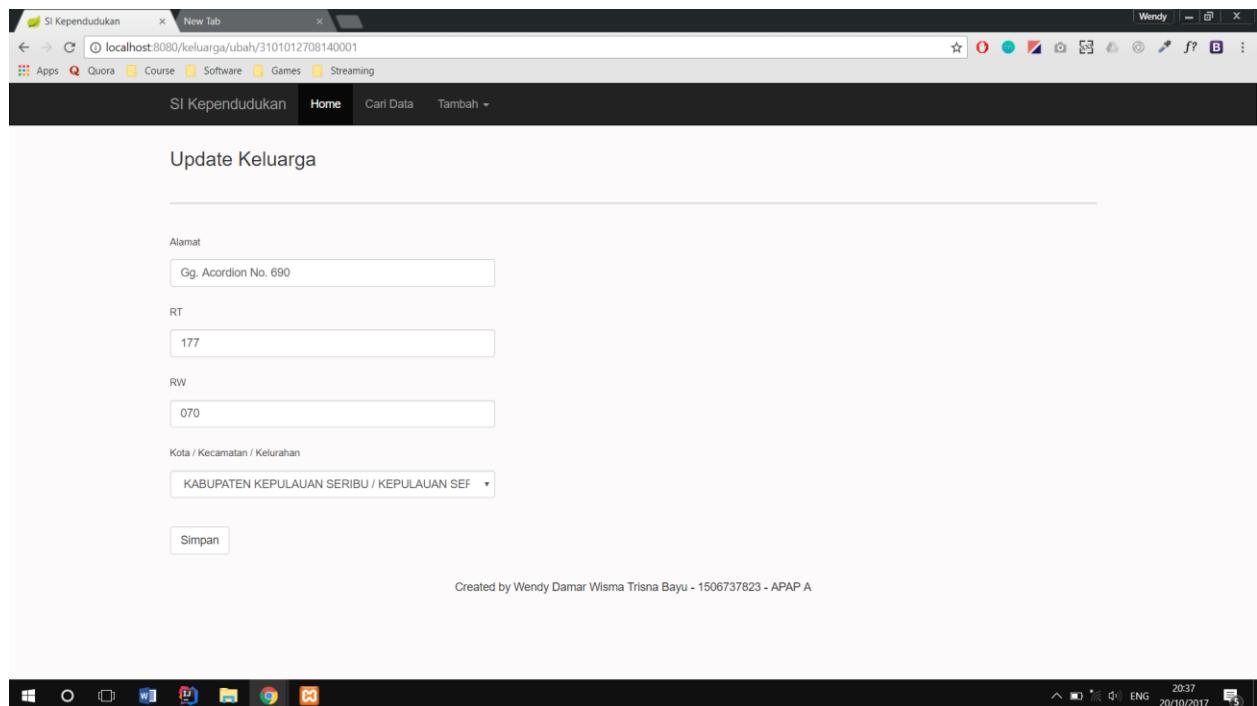
6. Mengubah Data Keluarga

Initial Page: /keluarga/ubah/{NKK}

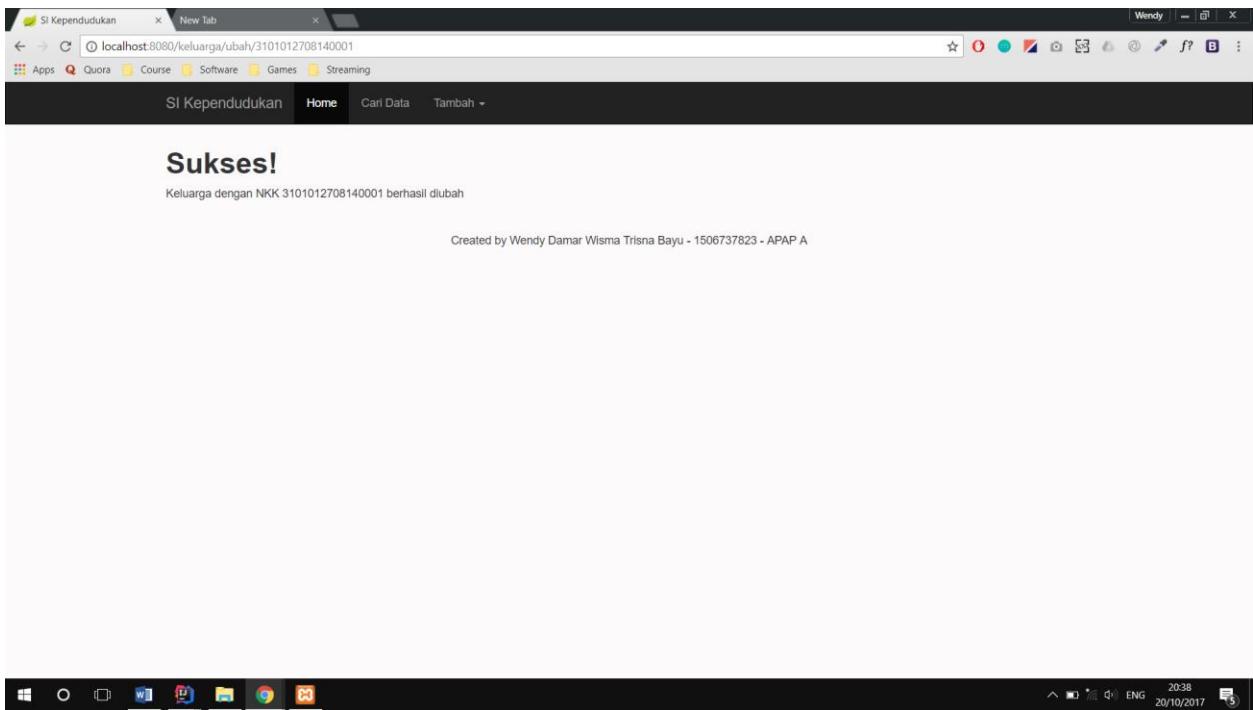
Form Request: POST, /keluarga/ubah/{NKK}

Response Page: /keluarga/ubah/{NKK}

Pada web ini kita dapat mengubah data keluarga pada setiap halaman lihat keluarga dimana dia akan mengeluarkan halaman update (Halaman menuju "keluarga/ubah/{nkk}"). Sebelum masuk pada halaman ubah, kita sebelumnya mencari terlebih dahulu apakah nkk yang ingin diubah telah terisi atau tidak. Setelah itu kita akan melakukan validasi apakah keluarga telah tersedia dengan nkk yang telah diberikan. Jika tidak ada maka akan diberikan halaman bahwa nkk tidak ditemukan. Dan jika ada ia akan memberikan halaman update yang isinya kurang lebih seperti registrasi namun bedanya ia actionnya menuju "keluarga/ubah/submit", menampilkan hasil data sebelumnya pada form, dan mengirimkan nik lama untuk pengecekan.



Lalu jika semua data telah tersubmit, maka akan masuk pada Controller dan dia akan mengecek apakah format yang sudah diinput sesuai dengan format pada model. Setelah itu melakukan validasi apakah ada data yang dapat merubah NKK. Jika tidak ada maka ia akan menginput NKK berdasarkan nkk lama. Dan jika tidak ia akan meng generate yang baru sesuai dengan ketentuannya. Dan jika sudah maka iakan meng update keluarga sekaligus anggotanya (menggunakan loop) dan menampilkan halaman apabila sukses.

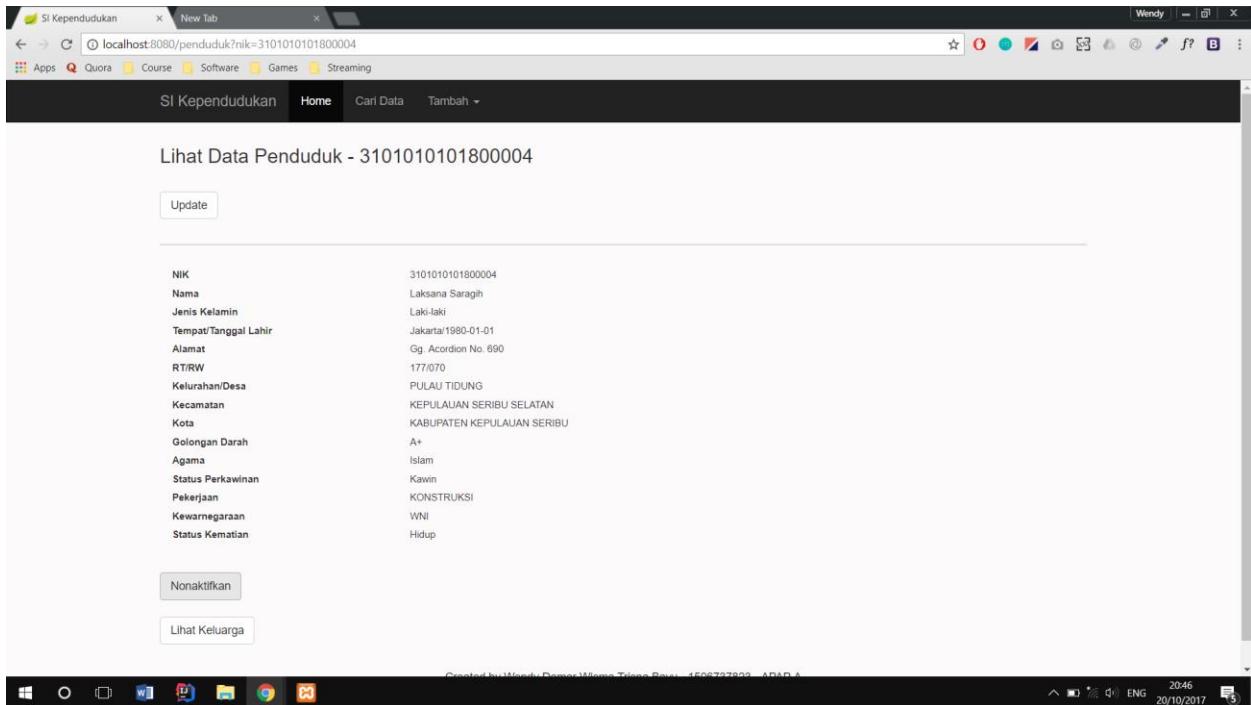


7. Mengubah Status Kematian Penduduk

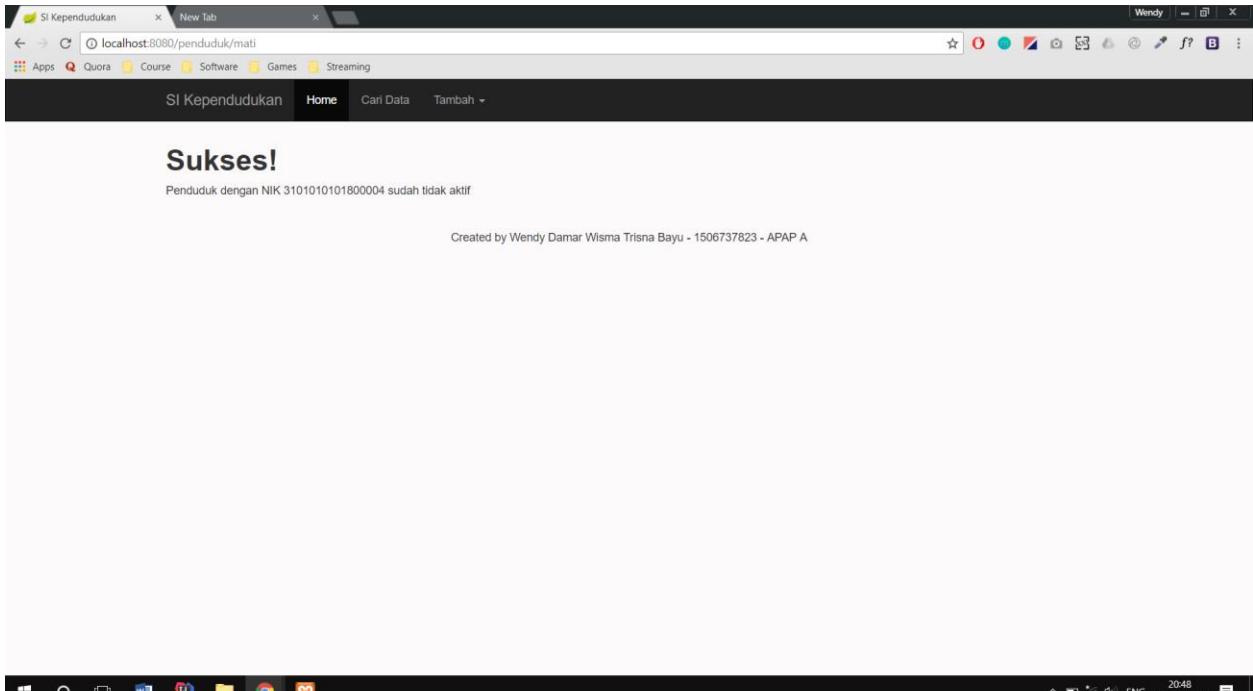
Initial Page: /penduduk?nik={NIK}

Form Request: POST, /penduduk/mati

Response Page: /penduduk?nik-{NIK}



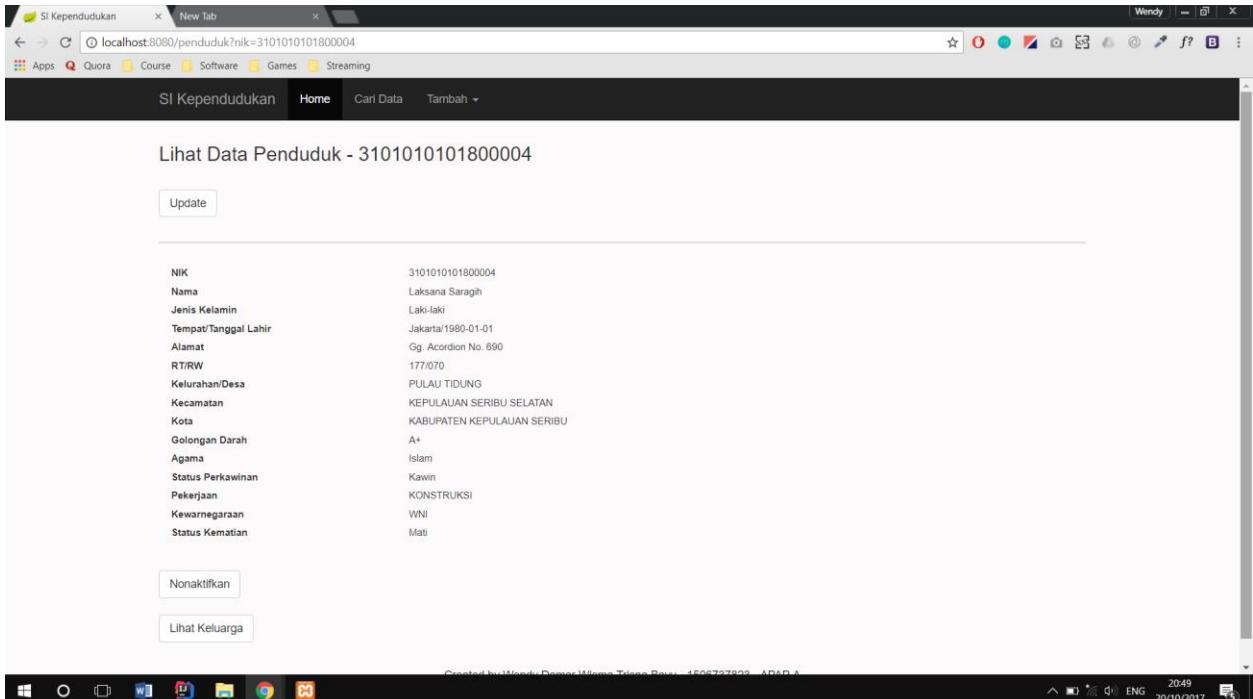
Hal ini dapat dilakukan dengan mengklik button pada lihat penduduk dimana ia merupakan sebuah form post yang akan meredirect menuju “penduduk/mati” dan ia akan melakukan pengecekan apakah nik telah tersedia. Jika ada ia akan mengupdate status is_wafat menjadi 1 sesuai dengan nik tersebut. Setelah itu kita akan melakukan pengecekan terhadap list keluarga mereka apakah KK masih valid atau tidak dengan cara mencari keluarga mereka berdasarkan id_keluarga nik tersebut. Lalu kita get semua data apabila semua sudah wafat maka kk sudah tidak berlaku dan akan diubah. Namun jika tidak ia tidak akan merubah status kk. Dan jika sudah selesai ia akan mengeluarkan halaman konfirmasi apabila berhasil. Lalu akan diredirect menuju halaman di ajika sudah melewati 5 detik



Sukses!

Penduduk dengan NIK 3101010101800004 sudah tidak aktif

Created by Wendy Damar Wisma Trisna Bayu - 1506737823 - APAP A



Lihat Data Penduduk - 3101010101800004

Update

NIK	3101010101800004
Nama	Laksana Saragih
Jenis Kelamin	Laki-laki
Tempat/Tanggal Lahir	Jakarta/1980-01-01
Alamat	Gg. Accordion No. 690
RTRW	177/070
Kelurahan/Desa	PULAU TIDUNG
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	A+
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	KONSTRUKSI
Kewarganegaraan	WNI
Status Kematian	Mati

Nonaktifkan

Lihat Keluarga

Created by Wendy Damar Wisma Trisna Bayu - 1506737823 - APAP A

20:49 20/10/2017

8. Tampilkan Data Penduduk Berdasarkan Kota/Kabupaten, Kecamatan, dan Kelurahan Tertentu

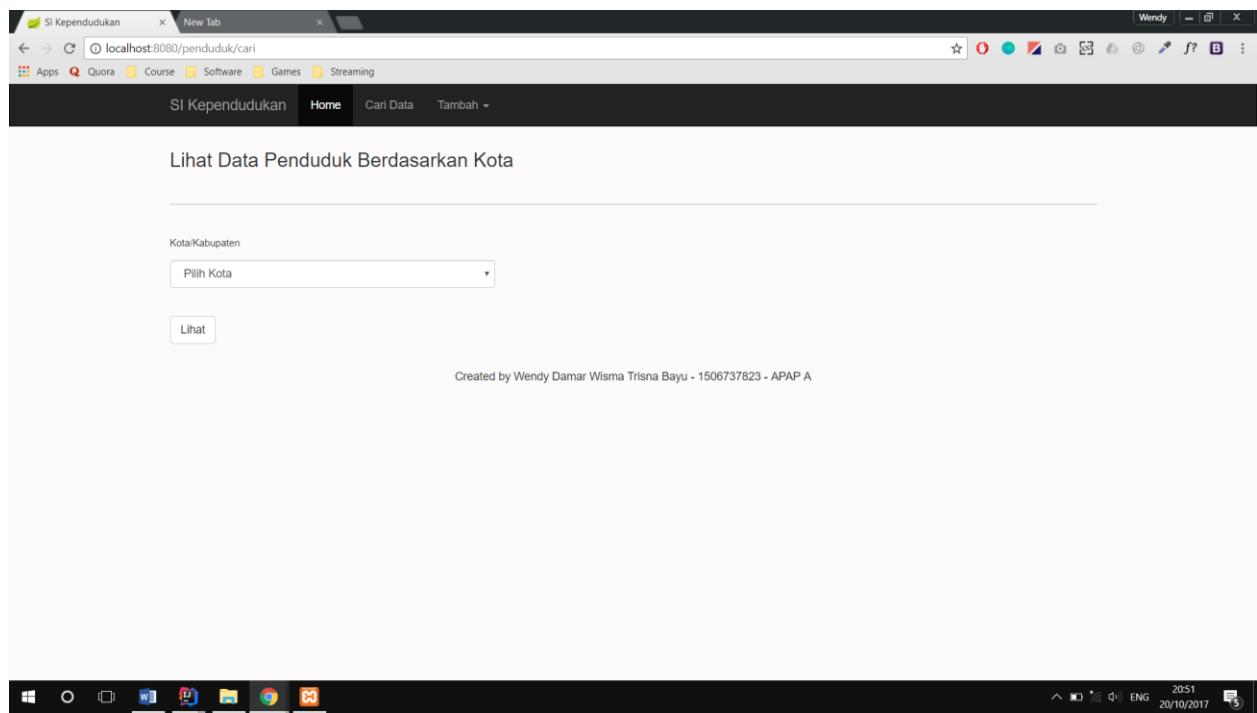
Initial Page: /penduduk/cari

Request Method: GET,

/penduduk/cari?kt={ID_KOTA}&kc={ID_KECAMATAN}&kl={ID_KELURAHAN}

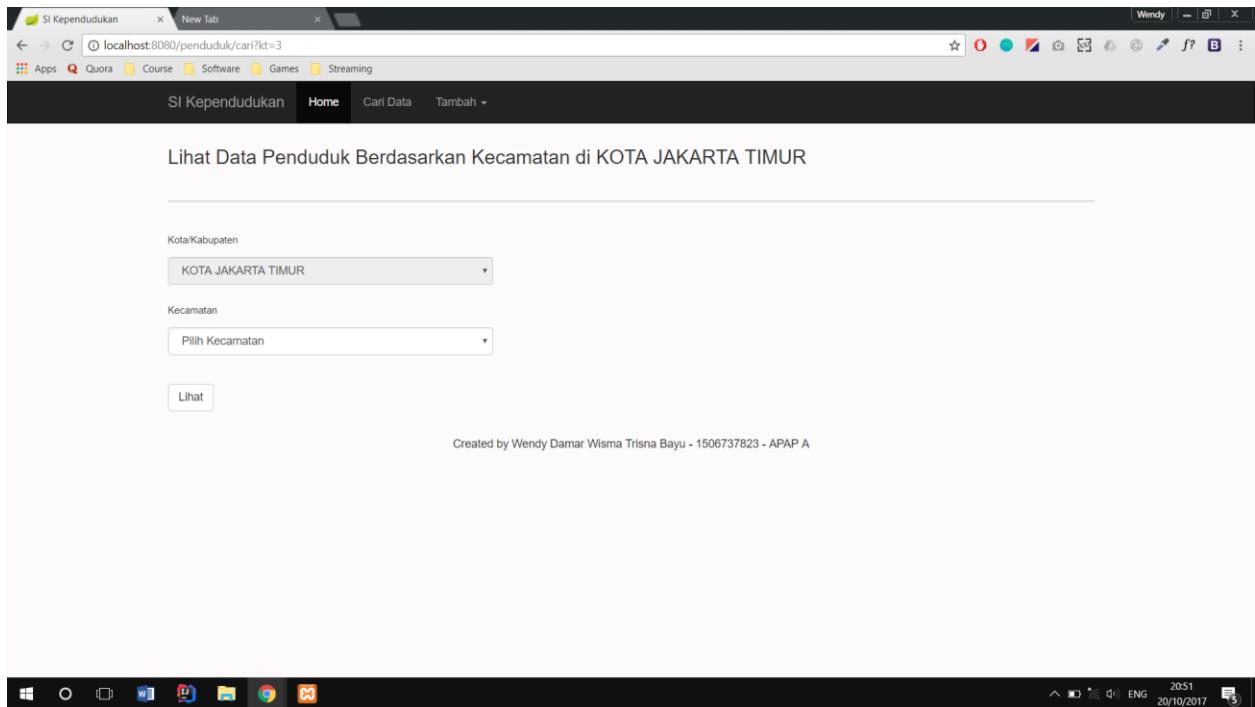
Response Page: /penduduk/cari?kt={ID_KOTA}&kc={ID_KECAMATAN}&kl={ID_KELURAHAN}

Fitur ini dapat diakses melalui navigation bar dimana jika di klik maka akan diantarkan menuju halaman pencarian penduduk berdasarkan lokasi. Flow yang dilakukan pertama ialah ketika menuju halaman “penduduk/cari”. Dimana hal ini akan diambil oleh Controller yang mana ia akan melakukan pengecekan apakah terdapat parameter kecamatan, kelurahan, ataupun kota. Karena awalnya kosong maka ia akan mengselect semua kota untuk ditampilkan pada form get pemilihan kota. Lalu Controller akan menampilkan halaman tersebut.

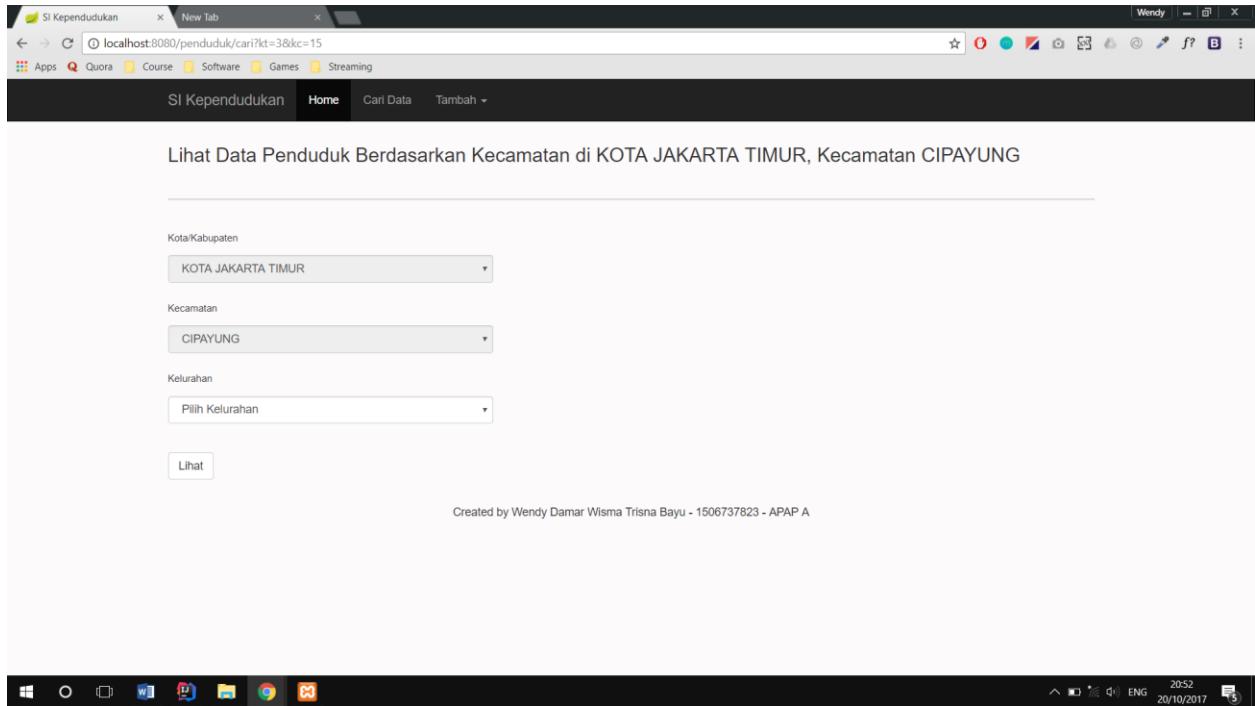


Jika sudah dipilih maka program akan mengirimkan parameter kota dan masuk ke Controller “penduduk/cari” lagi namun dengan kondisi sudah memiliki kota. Pada saat itu kita akan validasi apakah

tersedia kota tersebut pada database, jika tersedia maka akan lanjut meselect semua kecamatan yang terdapat pada kota tersebut sesuai dengan input dan menampilkan pada halaman pemilihan kecamatan. Jika tidak ia akan mengembalikan menuju halaman jika kota tersebut tidak ditemukan.



Jika sudah dipilih maka program akan mengirimkan parameter kota dan kecamatan masuk ke Controller "penduduk/cari" lagi namun dengan kondisi sudah memiliki kota dan kecamatan. Pada saat itu kita akan validasi apakah tersedia kecamatan tersebut pada database, jika tersedia maka akan lanjut meselect semua kelurahan yang terdapat pada kecamatan tersebut sesuai dengan input dan menampilkan pada halaman pemilihan kelurahan. Jika tidak ia akan mengembalikan menuju halaman jika kecamatan tersebut tidak ditemukan.



Jika sudah dipilih maka program akan mengirimkan parameter kota, kelurahan, dan kecamatan masuk ke Controller “penduduk/cari” lagi namun dengan kondisi sudah memiliki kota, kelurahan dan kecamatan. Pada saat itu kita akan validasi apakah tersedia kelurahan tersebut pada database, jika tidak ia akan mengembalikan menuju halaman jika kelurahan tersebut tidak ditemukan. Jika tersedia maka akan lanjut meselect semua penduduk yang terdapat pada kelurahan tersebut sesuai dengan input dan mengeset agar sesuai dengan input yang diinginkan dan kita mencari penduduk tertua dan termuda dengan menggunakan looping dan menampilkannya pada halaman hasil pencarian.

The screenshot shows a web browser window titled "SI Kependudukan". The URL in the address bar is "localhost:8080/penduduk/cari?kt=3&kc=15&id=85". The page displays population data for Cipayung, Jakarta Timur. At the top, it says "Lihat Data Penduduk Berdasarkan Kecamatan di KOTA JAKARTA TIMUR, Kecamatan CIPAYUNG, Kelurahan CIPAYUNG". Below this, there are two tables: "Penduduk Dengan Usia Termuda" and "Penduduk Dengan Usia Tertua". The "Penduduk Dengan Usia Termuda" table shows:

NIK	3172032009170003
Nama	Purwa Irawan
Tanggal Lahir	2017-09-20

The "Penduduk Dengan Usia Tertua" table shows:

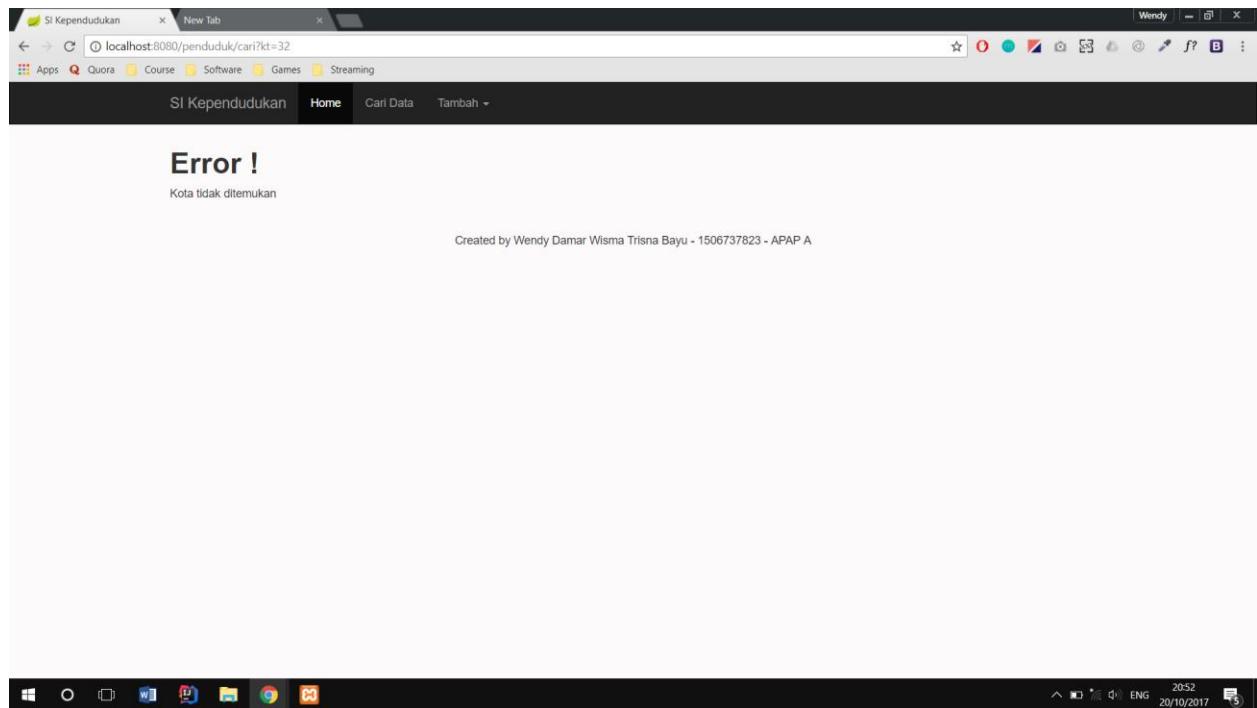
NIK	3172032809620001
Nama	Kenari Saputra
Tanggal Lahir	1952-09-28

Below these tables is a data grid with columns: NIK, Nama, and Jenis Kelamin. The data grid contains the following rows:

NIK	Nama	Jenis Kelamin
3172030101130001	Sabar Hutagalung	Laki-laki
3172030102140001	Imam Uwais M.Farm	Laki-laki
3172030102800001	Taswir Aswani Firmansyah S.T.	Laki-laki
3172030103980001	Muni Budiyanto	Laki-laki
3172030104160003	Kariman Prabowo	Laki-laki
3172030104550001	Jumadi Nababan	Laki-laki

9. Validasi Input Untuk Semua Form POST

Pada hal ini semua input telah dicek dengan menggunakan @Valid + BindingResult agar sesuai dengan aturan pada model, menggunakan pengecekan isNull pada @PathVariable, menggunakan pengecekan isNull pada @RequestParam, dan pengecekan manual pada Database. Dan jika terdapat error diluar itu maka akan dihandle dengan halaman universal Error dimana dia akan mengambil data pada \${error}



10. Menampilkan Penduduk Paling Muda dan Paling Tua di Suatu Kelurahan

Kita mencari penduduk tertua dan termuda dengan menggunakan looping pada setiap list yang telah di hasilkan pada “penduduk/cari” dan menampilkannya pada halaman hasil pencarian.

The screenshot shows a web browser window titled "SI Kependudukan". The URL in the address bar is "localhost:8080/penduduk/cari?kt=3&kc=15&id=85". The page displays search results for the youngest and oldest residents of Cipayung, Jakarta Timur. It includes two tables and a list of names.

Penduduk Dengan Usia Termuda

NIK	3172032009170003
Nama	Purwa Irawan
Tanggal Lahir	2017-09-20

Penduduk Dengan Usia Tertua

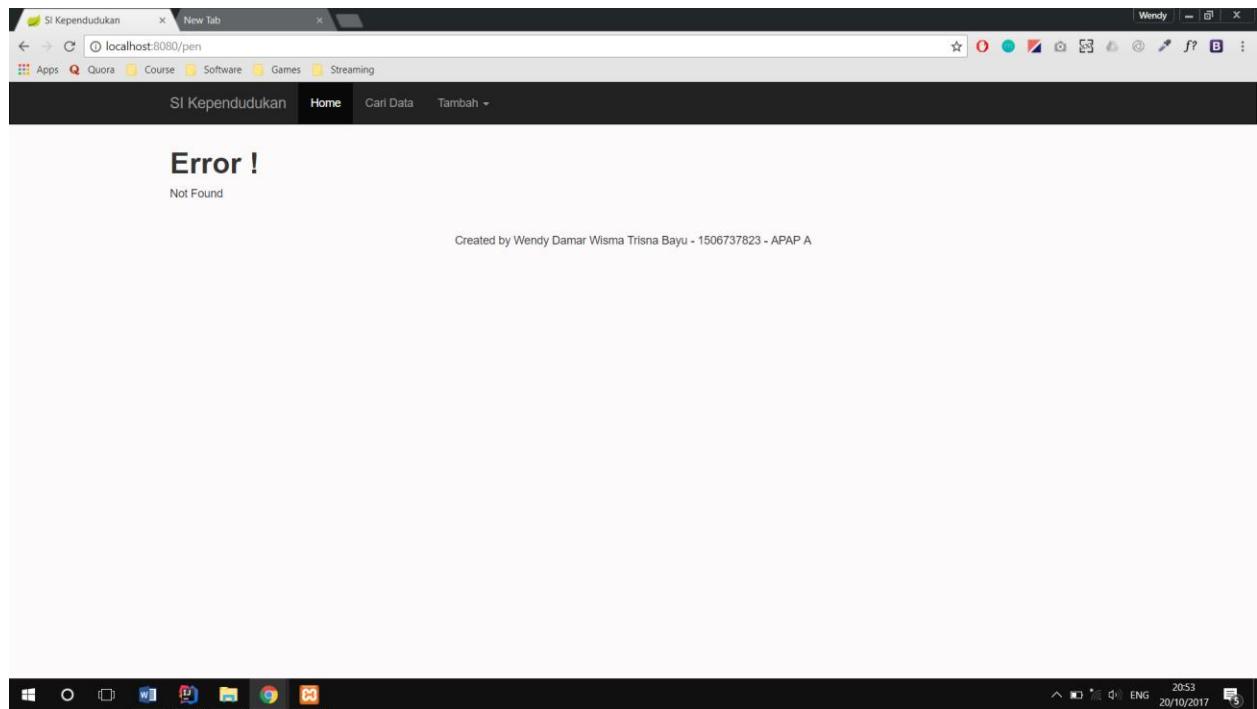
NIK	3172032809520001
Nama	Kenari Saputra
Tanggal Lahir	1952-09-28

Below these tables is a table listing names:

NIK	Nama	Jenis Kelamin
3172030101130001	Sabar Hutagalung	Laki-laki
3172030102140001	Imam Uwais M.Farm	Laki-laki
3172030102800001	Taswir Aswani Firmansyah S.T.	Laki-laki
3172030103980001	Muni Budiyanto	Laki-laki
3172030104160003	Kariman Prabowo	Laki-laki
3172030104550001	Jumadi Nababan	Laki-laki

11. Menambahkan Error Page

Saya meletakkan halaman error pada root dari templates dengan nama “error.html” dimana ia akan langsung mengambil semua halaman error dan menggantikan seluruh whitelabel dengan halaman tersebut dan pada halaman tersebut telah dilampirkan message error sesuai yang telah dilampirkan pada Controller Error native pada Spring (menggunakan \${error}) + Jika page not found



12. Fitur-Fitur Lain yang Mendukung Aplikasi

a. Membuat HyperLink menuju Keluarga Penduduk

Fitur ini hadir pada halaman lihat penduduk dimana akan terdapat form get yang jika diklik akan merefer langsung pada keluarga dari penduduk tersebut. Metode ini menggunakan input hidden dimana valuenya diambil dari nkk dari penduduk tersebut.

The screenshot shows a web browser window titled "SI Kependudukan". The URL in the address bar is "localhost:8080/penduduk?nik=3172030101130001". The page content is a user profile for a person with NIK 3172030101130001. The profile includes fields such as Name (Sabar Hutagalung), Gender (Laki-laki), Birthplace/Date (Jakarta/2013-01-01), Address (Ki. Salatiga No. 939), RT/RW (165/100), Village/Desa (CIPAYUNG), Sub-District (CIPAYUNG), City (KOTA JAKARTA TIMUR), Blood Type (B+), Religion (Islam), Marital Status (Belum Kawin), Employment Status (BELUM/TIDAK BEKERJA), Nationality (WNI), and Death Status (Mati). Below the profile, there are two buttons: "Nonaktifkan" and "Lihat Keluarga". At the bottom of the page, a footer note reads "Created by Wendy Damar Wisma Trisna Bayu - 1506737823 - APAP A". The browser interface shows various tabs and icons at the top and bottom.

b. Membuat HyperLink NIK

Fitur ini hadir pada setiap halaman jika terdapat NIK. Dimana NIK akan menjadi sebuah link dimana jika di klik ia akan menuju halaman penduduk tersebut. Metode ini menggunakan href pada setiap link dengan mengappend NIK dari penduduk tersebut.

Lihat Data Penduduk Berdasarkan Kecamatan di KOTA JAKARTA TIMUR, Kecamatan CIPAYUNG, Kelurahan CIPAYUNG

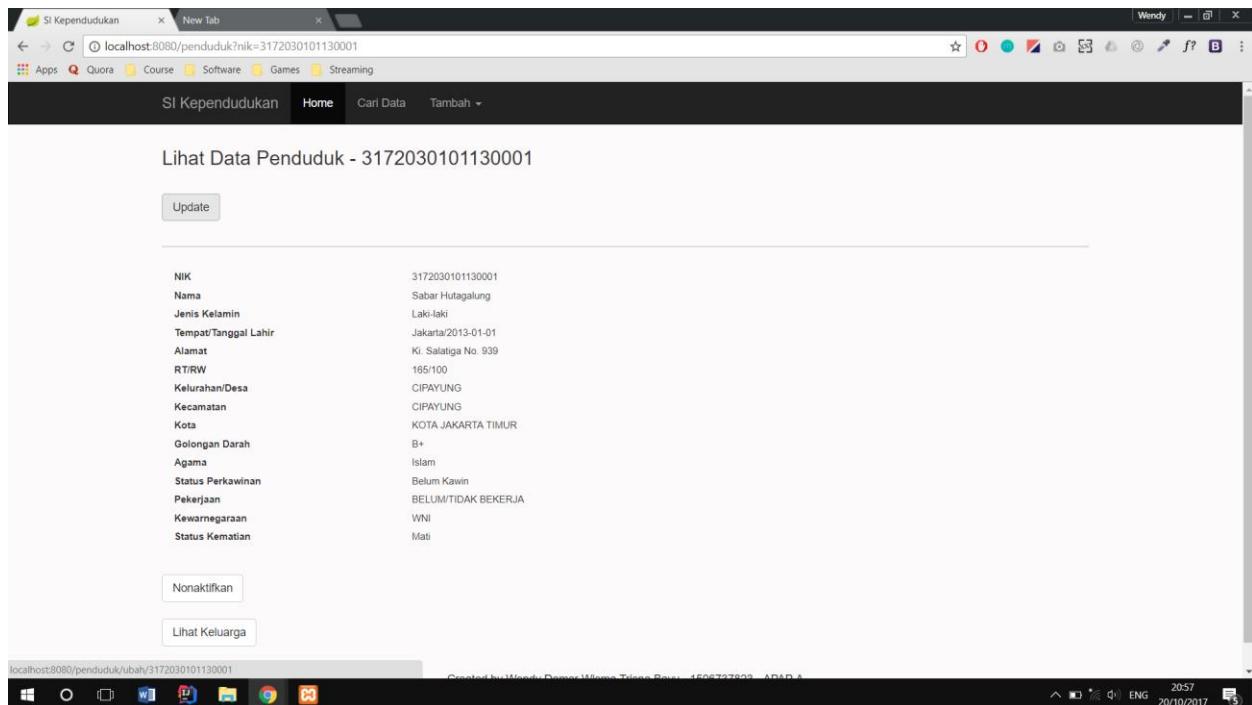
Penduduk Dengan Usia Termuda		Penduduk Dengan Usia Tertua	
NIK	Nama	NIK	Nama
3172032009170003	Purwa Irawan	3172032809520001	Kenari Saputra
Tanggal Lahir	2017-09-20	Tanggal Lahir	1952-09-28

Show 10 entries Search:

NIK	Nama	Jenis Kelamin
3172030101130001	Sabar Hutagalung	Laki-laki
3172030102140001	Imam Uwais M.Farm	Laki-laki
3172030102800001	Taswir Aswani Firmansyah S.T.	Laki-laki
3172030103980001	Muni Budiyanto	Laki-laki
3172030104160003	Kariman Prabowo	Laki-laki
3172030104550001	Jumadi Nababan	Laki-laki

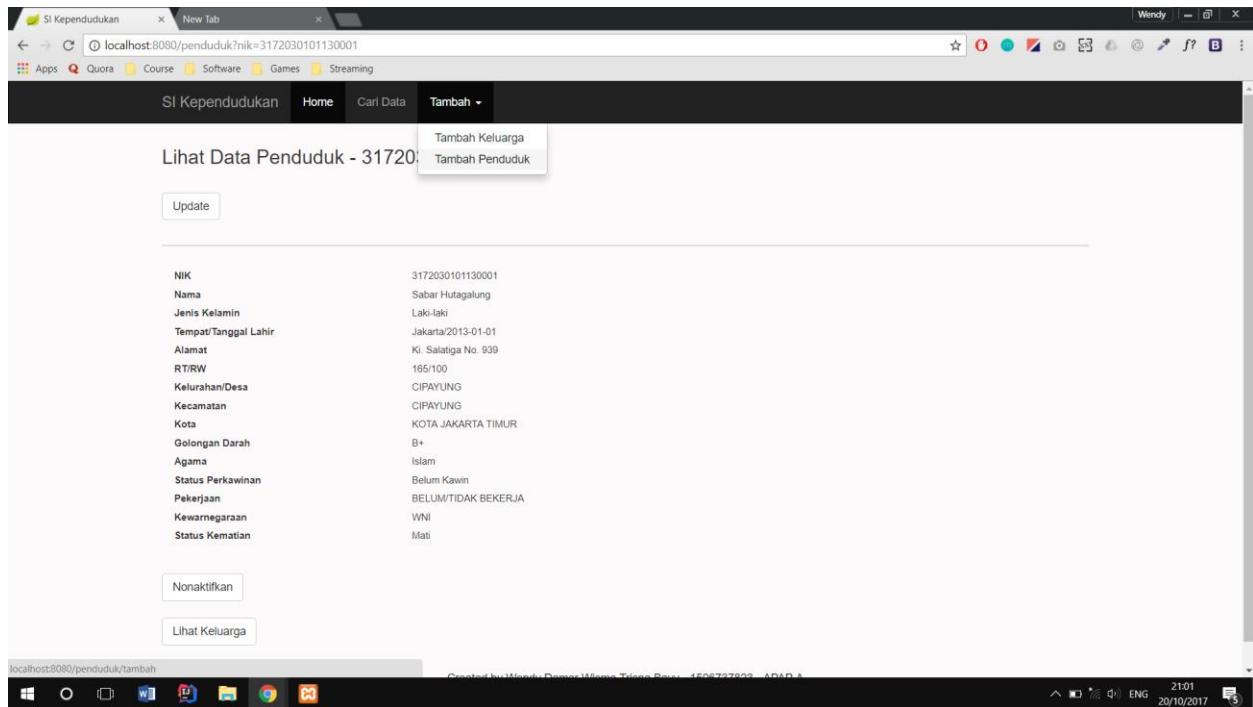
c. Membuat HyperLink Update

Fitur ini hadir pada halaman keluarga dan penduduk dimana mereka dapat menuju halaman ubah keluarga atau ubah penduduk yang bersangkutan hanya mengklik button pada halaman lihat keluarga/lihat penduduk. Fitur ini menggunakan href dengan mengappend nik/nkk terkait



d. Membuat Tombol navigasi menuju Tambah Penduduk

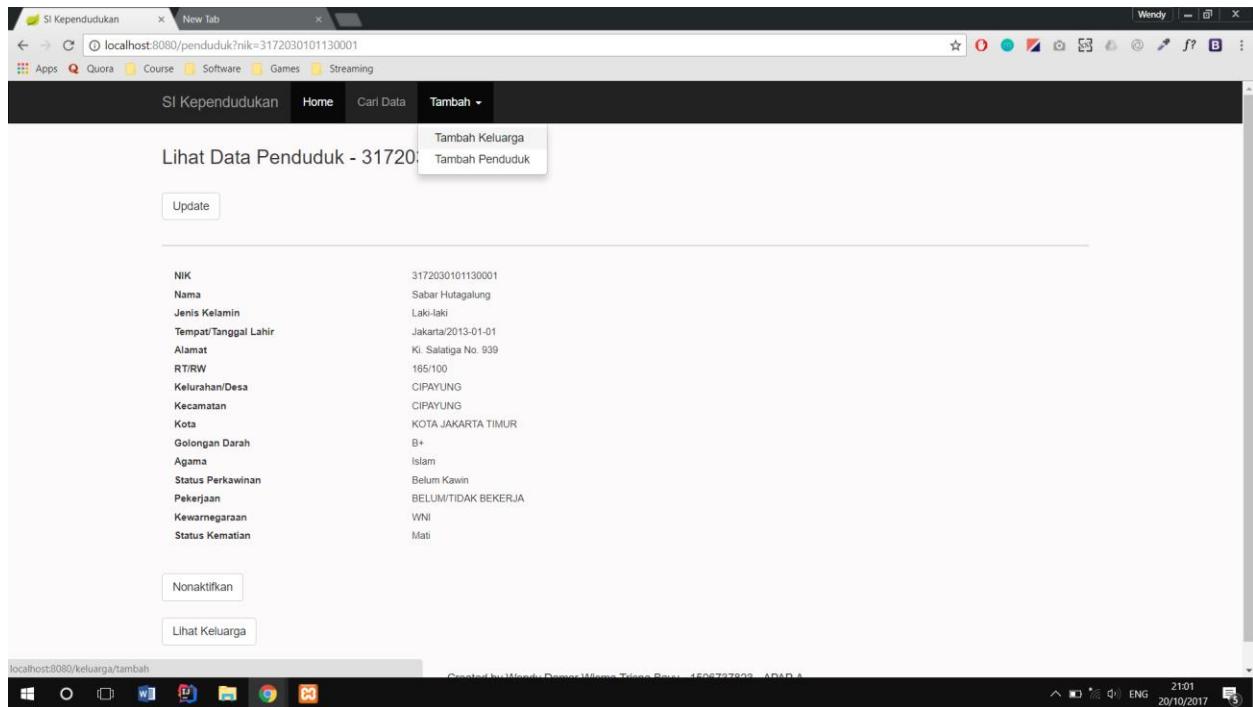
Fitur ini ialah menambahkan link menuju halaman tambah penduduk pada navigation bar. Fitur ini menggunakan href seperti biasa menuju /penduduk/tambah



e. Membuat Tombol navigasi menuju Tambah Keluarga

Fitur ini ialah menambahkan link menuju halaman tambah keluarga pada navigation bar.

Fitur ini menggunakan href seperti biasa menuju /keluarga/tambah



Optimasi Database

Berikut hal-hal yang saya lakukan demi mempercepat dan mengefisienkan waktu untuk database:

1. Menambahkan Primary Key dan Index pada setiap Id
2. Menambahkan Index pada NKK dan NIK
3. Menambahkan Serial pada ID Keluarga dan Penduduk

Penambahan Primary key dan index digunakan apabila pencarian menggunakan Id sering dilakukan dan memiliki jumlah data yang sangat besar sehingga dapat mengefisienkan waktu. Begitu pula dengan penambahan index pada NKK dan NIK.

Lalu penambahan serial dilakukan supaya dalam proses insert kita tak perlu lagi mencari jumlah size dari database untuk mencari ID untuk data yang akan dimasukkan. Maka dari itu hal ini dapat mempercepat kinerja dari program.

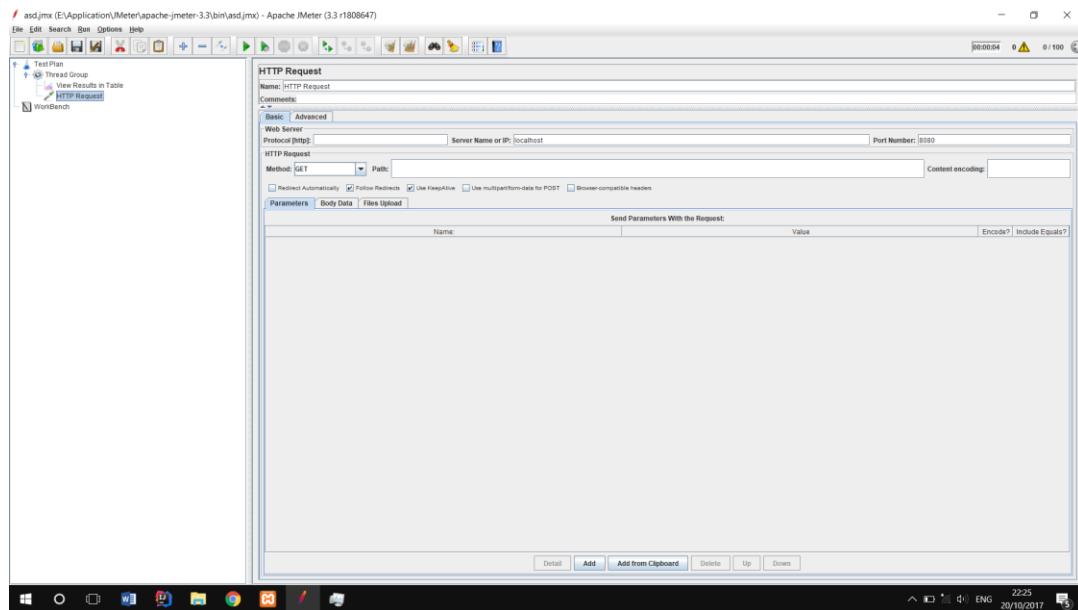
Stress Testing

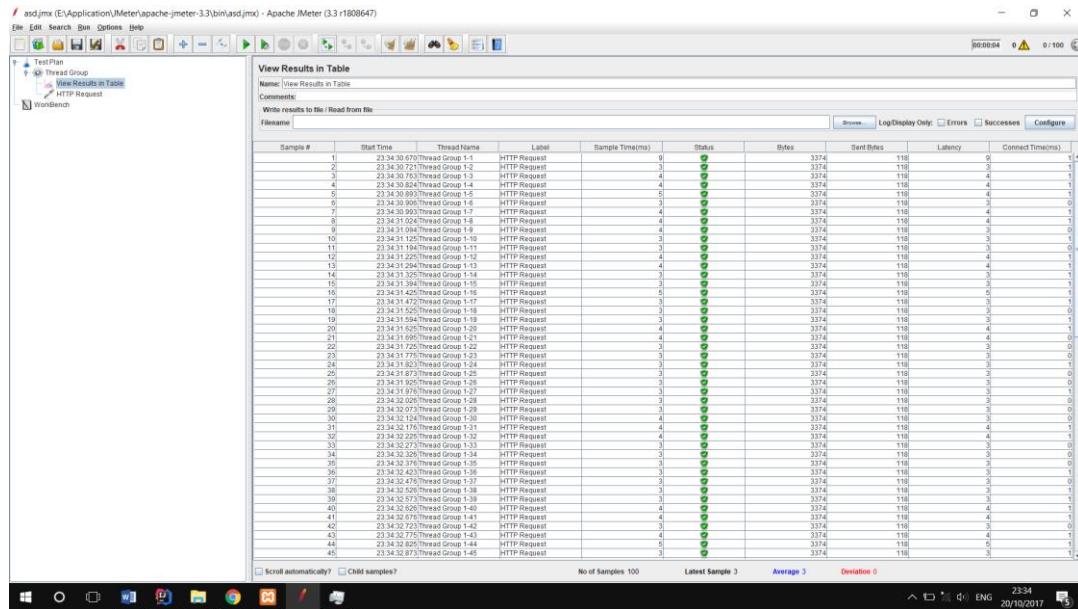
Ramp Up: 5 s

Thread: 100

1. Akses halaman utama

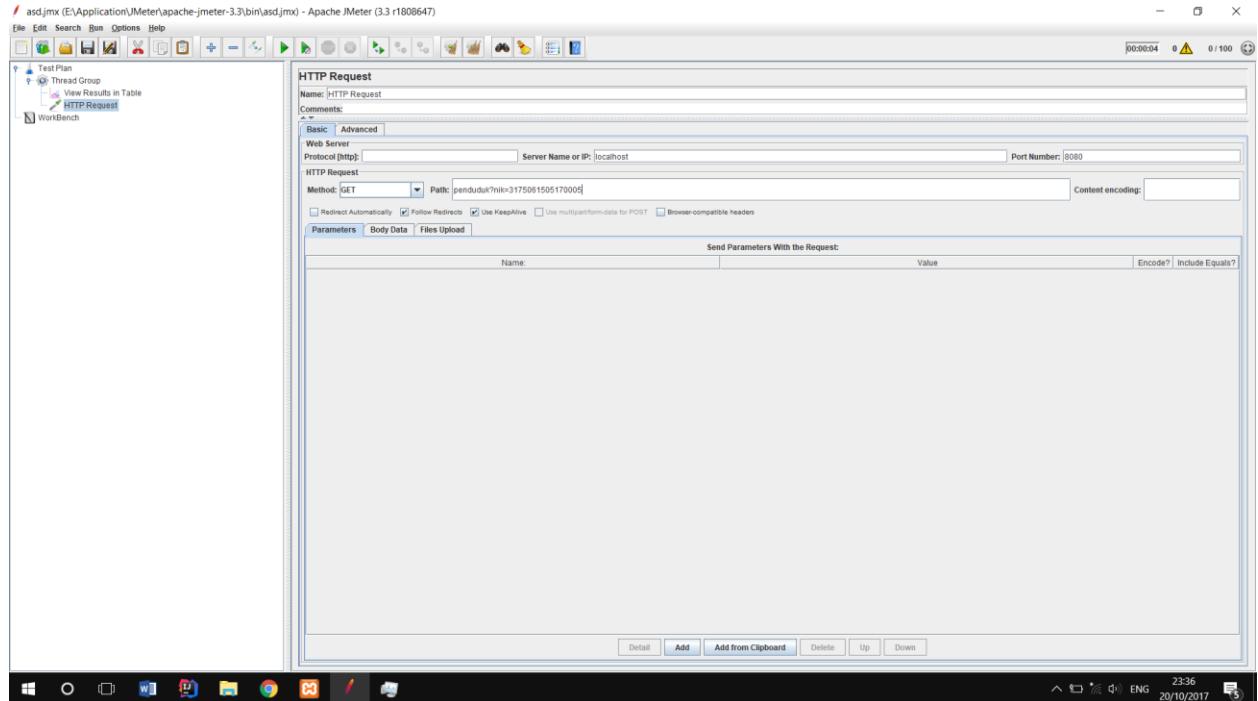
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html dan css pada web.





2. Akses lihat penduduk

Pada halaman ini juga dapat berjalan dengan cepat dan dikarenakan efisiensi query dan juga penggunaan primary key dan index pada id penduduk

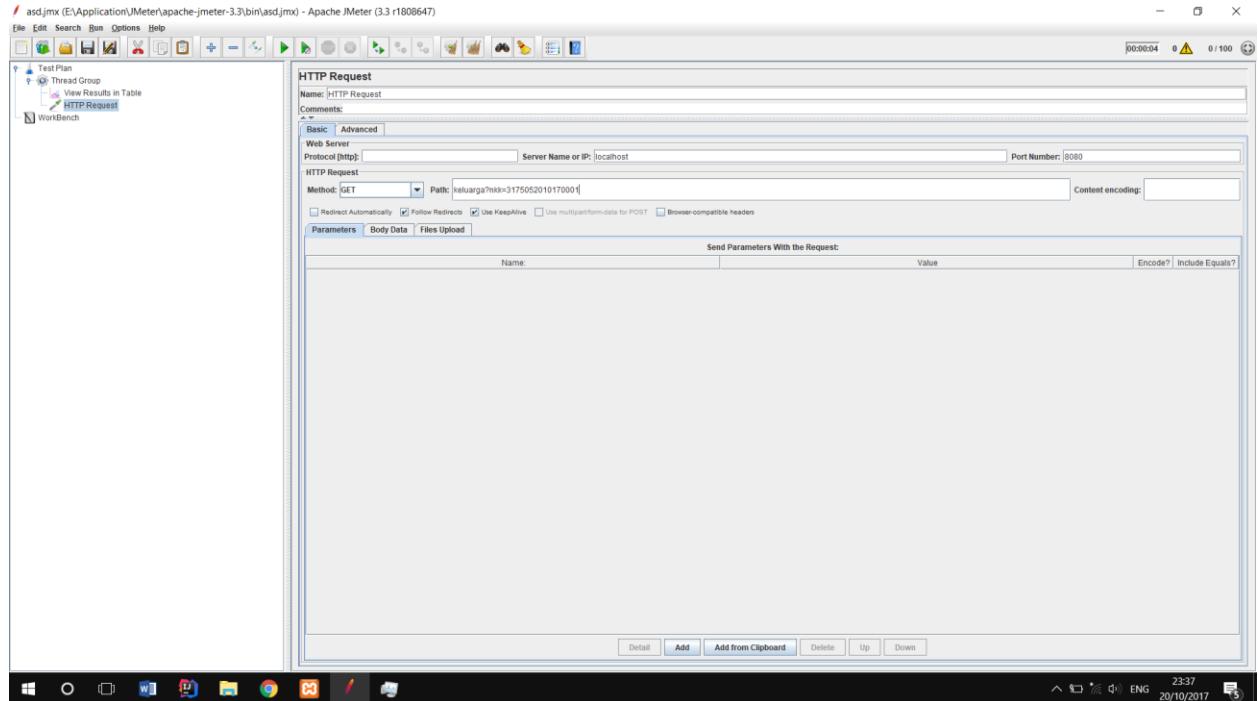


The screenshot shows the Apache JMeter interface with the following details:

- Test Plan:** The main window displays a single Thread Group under the Test Plan tree.
- View Results in Table Configuration:**
 - Name: View Results in Table
 - Comments: Write results to file / Read from file
 - Filename: (empty)
- Table Data:** A large table titled "View Results in Table" lists 100 rows of request details. The columns are: Sample #, Start Time, Thread Name, Label, Status, Sample Time(ms), Bytes, Sent Bytes, Latency, and Connected time(ms). Each row shows a timestamp starting from 23.37.01.322 and ending at 23.37.03.517, with various thread group labels like Thread Group 1-1 through Thread Group 1-45.
- Log/Display Only:** Options for Errors, Successes, and Configure are visible at the top of the table.
- WorkBench:** A panel on the right shows the current status: 00:00:04, 0 / 100.

3. Akses lihat keluarga

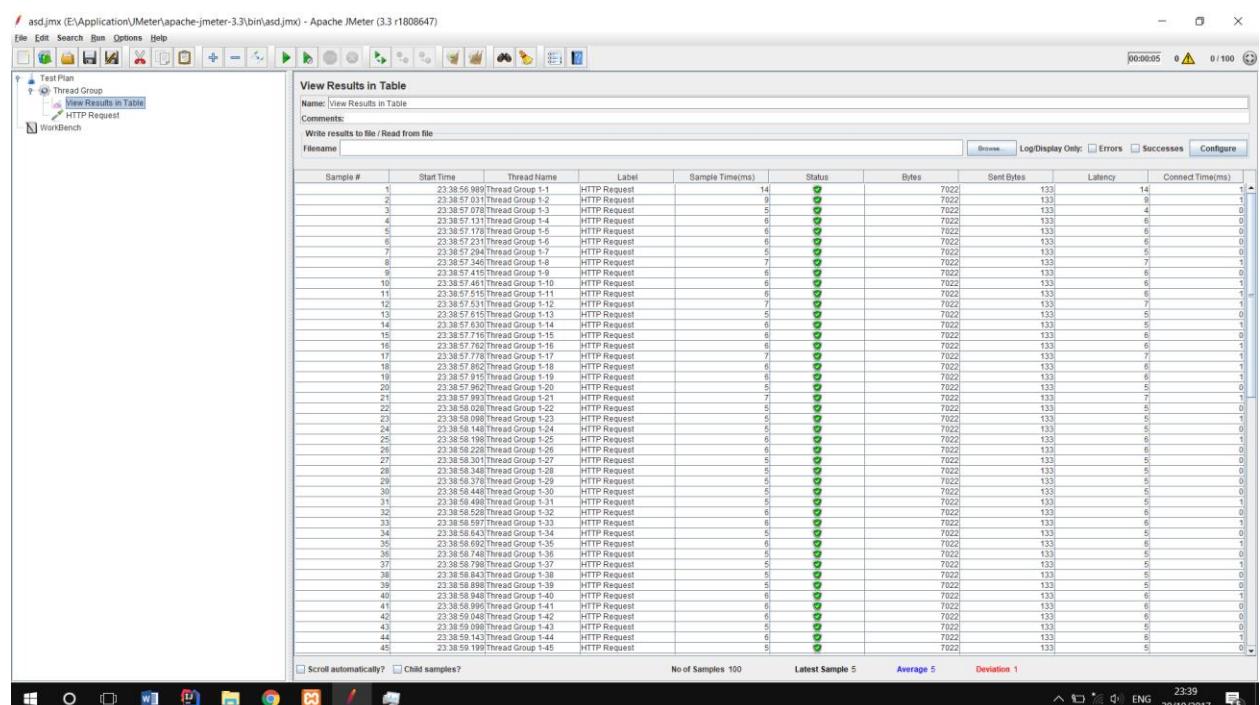
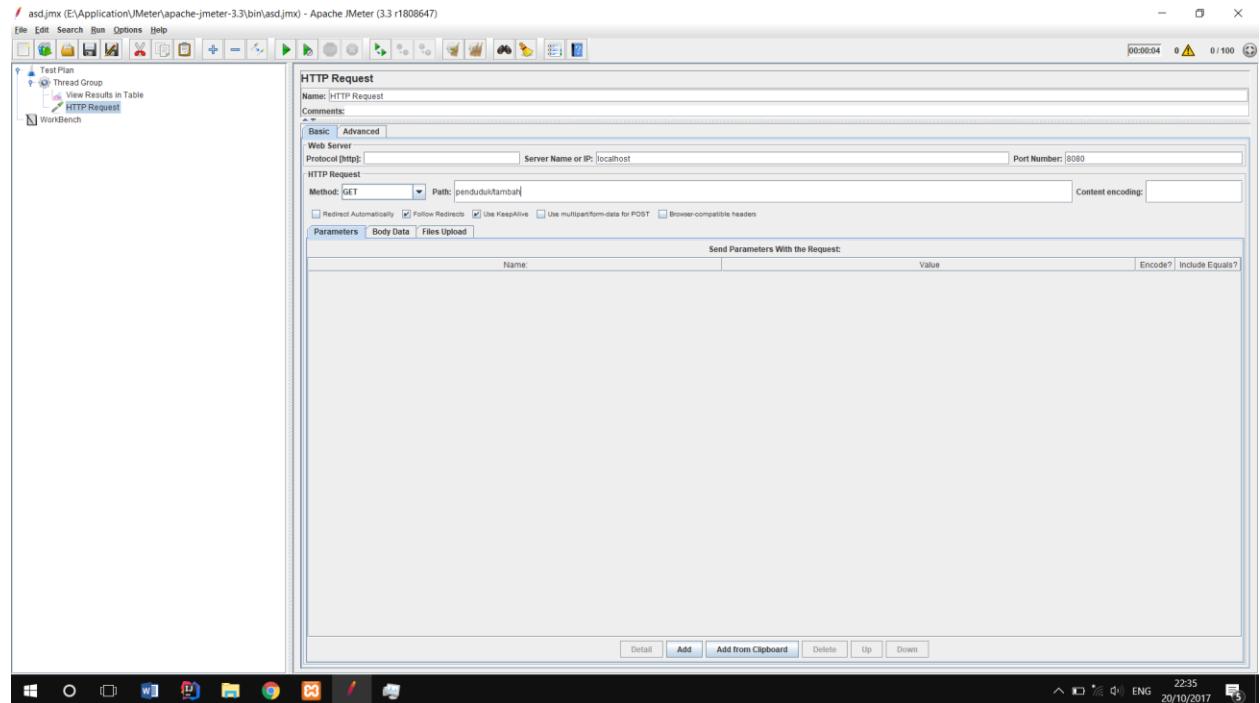
Pada halaman ini juga dapat berjalan dengan cepat man dikarenakan efisiensi query dan juga penggunaan primary key dan index pada id keluarga



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connected Time(ms)
1	23:38:22.682	Thread Group 1-8	HTTP Request	1370	✓	138048	147	1364	0
2	23:38:22.628	Thread Group 1-7	HTTP Request	1426	✓	138048	147	1420	1
3	23:38:22.782	Thread Group 1-10	HTTP Request	2254	✓	138048	147	2249	0
4	23:38:22.782	Thread Group 1-11	HTTP Request	2106	✓	138048	147	2106	0
5	23:38:22.528	Thread Group 1-5	HTTP Request	3486	✓	138048	147	3481	1
6	23:38:22.713	Thread Group 1-9	HTTP Request	3316	✓	138048	147	3302	0
7	23:38:22.465	Thread Group 1-4	HTTP Request	3819	✓	138048	147	3809	0
8	23:38:22.528	Thread Group 1-6	HTTP Request	3983	✓	138048	147	3983	0
9	23:38:22.882	Thread Group 1-12	HTTP Request	3406	✓	138048	147	3400	1
10	23:38:22.581	Thread Group 1-6	HTTP Request	3708	✓	138048	147	3702	1
11	23:38:22.381	Thread Group 1-2	HTTP Request	4052	✓	138048	147	4048	1
12	23:38:22.381	Thread Group 1-3	HTTP Request	4021	✓	138048	147	4016	1
13	23:38:23.228	Thread Group 1-18	HTTP Request	5437	✓	138048	147	5432	1
14	23:38:23.014	Thread Group 1-15	HTTP Request	6050	✓	138048	147	5653	0
15	23:38:23.315	Thread Group 1-21	HTTP Request	5754	✓	138048	147	5373	1
16	23:38:23.315	Thread Group 1-13	HTTP Request	5188	✓	138048	147	5173	1
17	23:38:23.463	Thread Group 1-24	HTTP Request	5661	✓	138048	147	5232	0
18	23:38:22.982	Thread Group 1-14	HTTP Request	6164	✓	138048	147	5704	1
19	23:38:23.812	Thread Group 1-19	HTTP Request	5604	✓	138048	147	5607	0
20	23:38:23.228	Thread Group 1-20	HTTP Request	5942	✓	138048	147	5937	0
21	23:38:23.128	Thread Group 1-17	HTTP Request	6091	✓	138048	147	6086	1
22	23:38:23.175	Thread Group 1-18	HTTP Request	6050	✓	138048	147	6044	0
23	23:38:23.175	Thread Group 1-16	HTTP Request	6152	✓	138048	147	6147	0
24	23:38:23.415	Thread Group 1-23	HTTP Request	5421	✓	138048	147	5416	0
25	23:38:23.562	Thread Group 1-28	HTTP Request	5677	✓	138048	147	5673	1
26	23:38:23.362	Thread Group 1-22	HTTP Request	5930	✓	138048	147	5917	1
27	23:38:23.252	Thread Group 1-25	HTTP Request	5805	✓	138048	147	5801	1
28	23:38:23.126	Thread Group 1-67	HTTP Request	5768	✓	138048	147	5763	1
29	23:38:23.862	Thread Group 1-32	HTTP Request	7034	✓	138048	147	7029	1
30	23:38:23.763	Thread Group 1-30	HTTP Request	7136	✓	138048	147	7129	0
31	23:38:23.763	Thread Group 1-31	HTTP Request	5582	✓	138048	147	5573	0
32	23:38:25.429	Thread Group 1-63	HTTP Request	5481	✓	138048	147	5475	0
33	23:38:24.312	Thread Group 1-41	HTTP Request	6598	✓	138048	147	6591	1
34	23:38:24.663	Thread Group 1-48	HTTP Request	6261	✓	138048	147	6256	0
35	23:38:24.663	Thread Group 1-51	HTTP Request	5514	✓	138048	147	5506	1
36	23:38:25.179	Thread Group 1-68	HTTP Request	5757	✓	138048	147	5751	1
37	23:38:23.828	Thread Group 1-31	HTTP Request	7192	✓	138048	147	7186	1
38	23:38:24.112	Thread Group 1-37	HTTP Request	6936	✓	138048	147	6930	1
39	23:38:23.828	Thread Group 1-32	HTTP Request	7512	✓	138048	147	7506	1
40	23:38:24.162	Thread Group 1-38	HTTP Request	7313	✓	138048	147	7308	1
41	23:38:24.563	Thread Group 1-46	HTTP Request	6920	✓	138048	147	6915	1
42	23:38:24.052	Thread Group 1-33	HTTP Request	7422	✓	138048	147	7417	1
43	23:38:24.052	Thread Group 1-35	HTTP Request	7587	✓	138048	147	7584	1
44	23:38:25.079	Thread Group 1-56	HTTP Request	6553	✓	138048	147	6548	0
45	23:38:24.712	Thread Group 1-49	HTTP Request	6922	✓	138048	147	6916	1

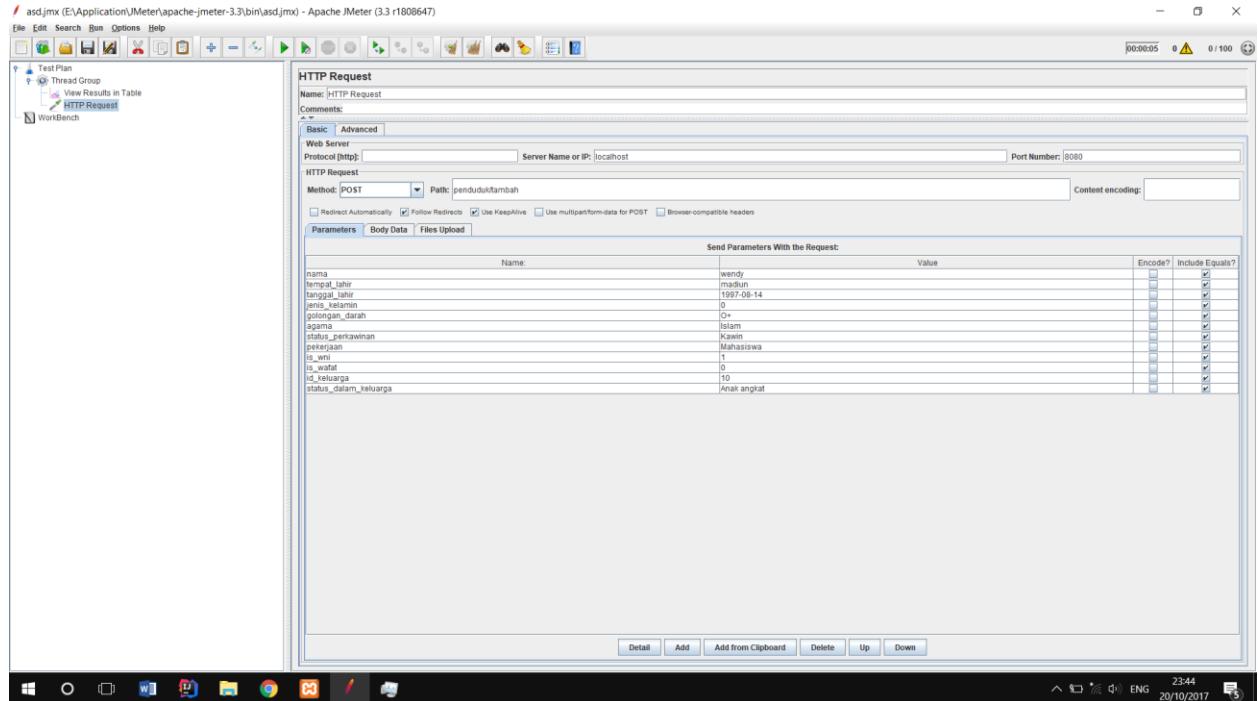
4. Akses tambah penduduk

Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html dan css pada web.



5. Akses tambah penduduk submit

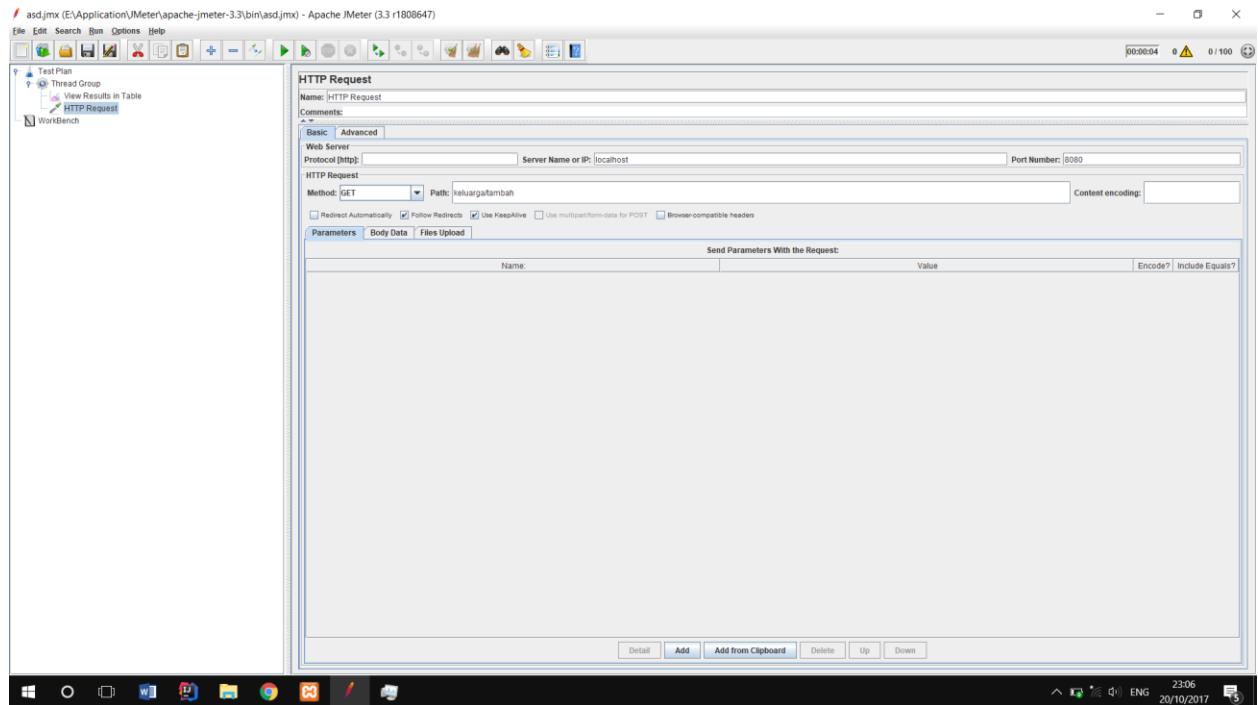
Halaman dapat berjalan dengan optimal hal ini dikarenakan minimnya penggunaan query dan penggunaan primary key, serial, dan index.

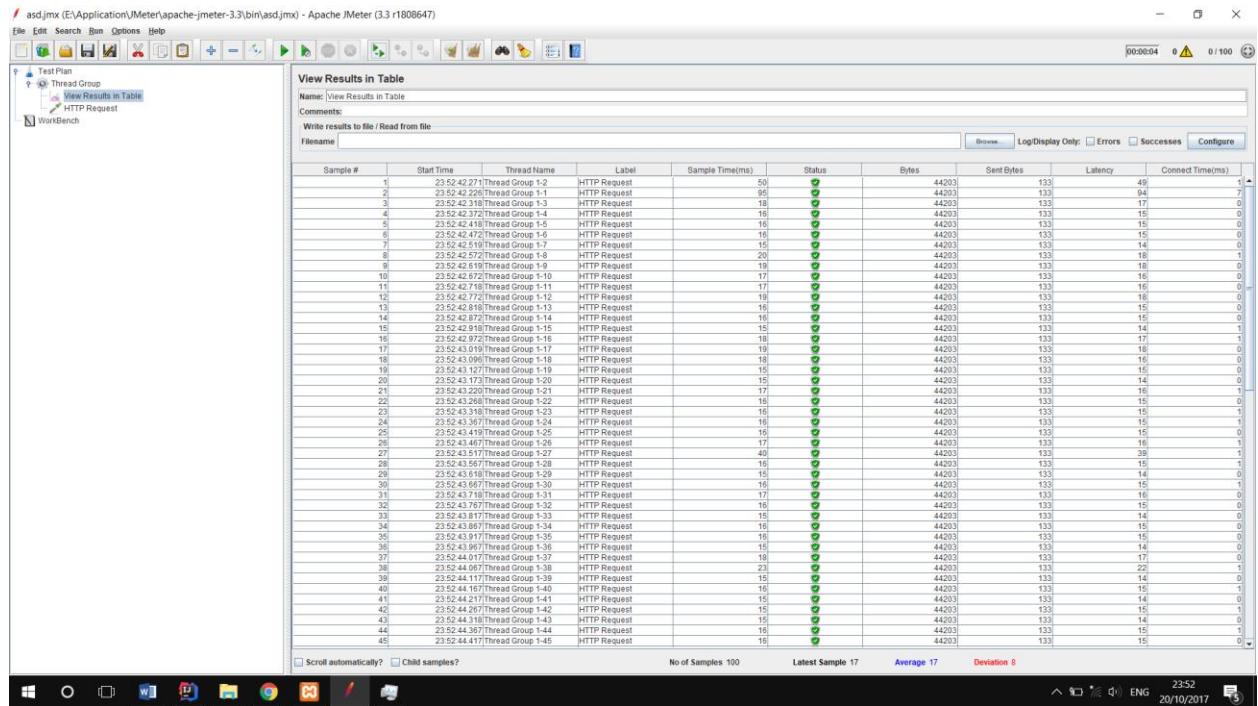


Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connected time(ms)
1	23:44:59.321	Thread Group 1-3	HTTP Request	93	✓	2699	416	93	1
2	23:44:59.268	Thread Group 1-2	HTTP Request	185	✓	2699	416	185	1
3	23:44:59.368	Thread Group 1-4	HTTP Request	102	✓	2699	416	102	1
4	23:44:59.288	Thread Group 1-1	HTTP Request	268	✓	2699	416	268	1
5	23:44:59.421	Thread Group 1-5	HTTP Request	65	✓	2699	416	65	1
6	23:44:59.456	Thread Group 1-6	HTTP Request	70	✓	2699	416	70	1
7	23:44:59.523	Thread Group 1-7	HTTP Request	20	✓	2699	416	20	0
8	23:44:59.523	Thread Group 1-8	HTTP Request	23	✓	2699	416	23	0
9	23:44:59.522	Thread Group 1-9	HTTP Request	29	✓	2699	416	29	0
10	23:44:59.669	Thread Group 1-10	HTTP Request	22	✓	2699	416	22	0
11	23:44:59.722	Thread Group 1-11	HTTP Request	32	✓	2699	416	32	0
12	23:44:59.722	Thread Group 1-12	HTTP Request	24	✓	2699	416	24	1
13	23:44:59.822	Thread Group 1-13	HTTP Request	31	✓	2699	416	31	1
14	23:44:59.869	Thread Group 1-14	HTTP Request	34	✓	2699	416	34	1
15	23:44:59.922	Thread Group 1-15	HTTP Request	31	✓	2699	416	31	1
16	23:45:00.022	Thread Group 1-6	HTTP Request	34	✓	2699	416	34	0
17	23:45:00.022	Thread Group 1-17	HTTP Request	236	✓	2699	416	236	1
18	23:45:00.122	Thread Group 1-18	HTTP Request	268	✓	2699	416	268	1
19	23:45:00.070	Thread Group 1-9	HTTP Request	320	✓	2699	416	320	0
20	23:45:00.122	Thread Group 1-22	HTTP Request	274	✓	2699	416	274	1
21	23:45:00.169	Thread Group 1-20	HTTP Request	359	✓	2699	416	359	1
22	23:45:00.223	Thread Group 1-21	HTTP Request	454	✓	2699	416	454	1
23	23:45:00.223	Thread Group 1-23	HTTP Request	500	✓	2699	416	500	1
24	23:45:00.305	Thread Group 1-24	HTTP Request	610	✓	2699	416	610	1
25	23:45:00.355	Thread Group 1-24	HTTP Request	622	✓	2699	416	622	1
26	23:45:00.555	Thread Group 1-28	HTTP Request	502	✓	2699	416	502	0
27	23:45:00.595	Thread Group 1-29	HTTP Request	575	✓	2699	416	575	0
28	23:45:00.454	Thread Group 1-26	HTTP Request	727	✓	2699	416	727	1
29	23:45:00.505	Thread Group 1-27	HTTP Request	830	✓	2699	416	830	1
30	23:45:00.954	Thread Group 1-36	HTTP Request	543	✓	2699	416	543	0
31	23:45:01.001	Thread Group 1-37	HTTP Request	662	✓	2699	416	662	0
32	23:45:00.904	Thread Group 1-35	HTTP Request	596	✓	2699	416	596	0
33	23:45:00.655	Thread Group 1-30	HTTP Request	1208	✓	2699	416	1208	0
34	23:45:01.105	Thread Group 1-38	HTTP Request	786	✓	2699	416	786	1
35	23:45:01.055	Thread Group 1-41	HTTP Request	737	✓	2699	416	737	1
36	23:45:01.055	Thread Group 1-38	HTTP Request	906	✓	2699	416	906	1
37	23:45:01.006	Thread Group 1-37	HTTP Request	954	✓	2699	416	954	0
38	23:45:00.707	Thread Group 1-31	HTTP Request	1253	✓	2699	416	1253	0
39	23:45:01.032	Thread Group 1-32	HTTP Request	1260	✓	2699	416	1260	1
40	23:45:01.809	Thread Group 1-53	HTTP Request	156	✓	2699	416	156	1
41	23:45:00.856	Thread Group 1-34	HTTP Request	1105	✓	2699	416	1105	0
42	23:45:01.055	Thread Group 1-48	HTTP Request	955	✓	2699	416	955	0
43	23:45:01.557	Thread Group 1-48	HTTP Request	570	✓	2699	416	570	0
44	23:45:01.657	Thread Group 1-50	HTTP Request	470	✓	2699	416	470	0
45	23:45:01.356	Thread Group 1-44	HTTP Request	772	✓	2699	416	772	1

6. Akses tambah keluarga

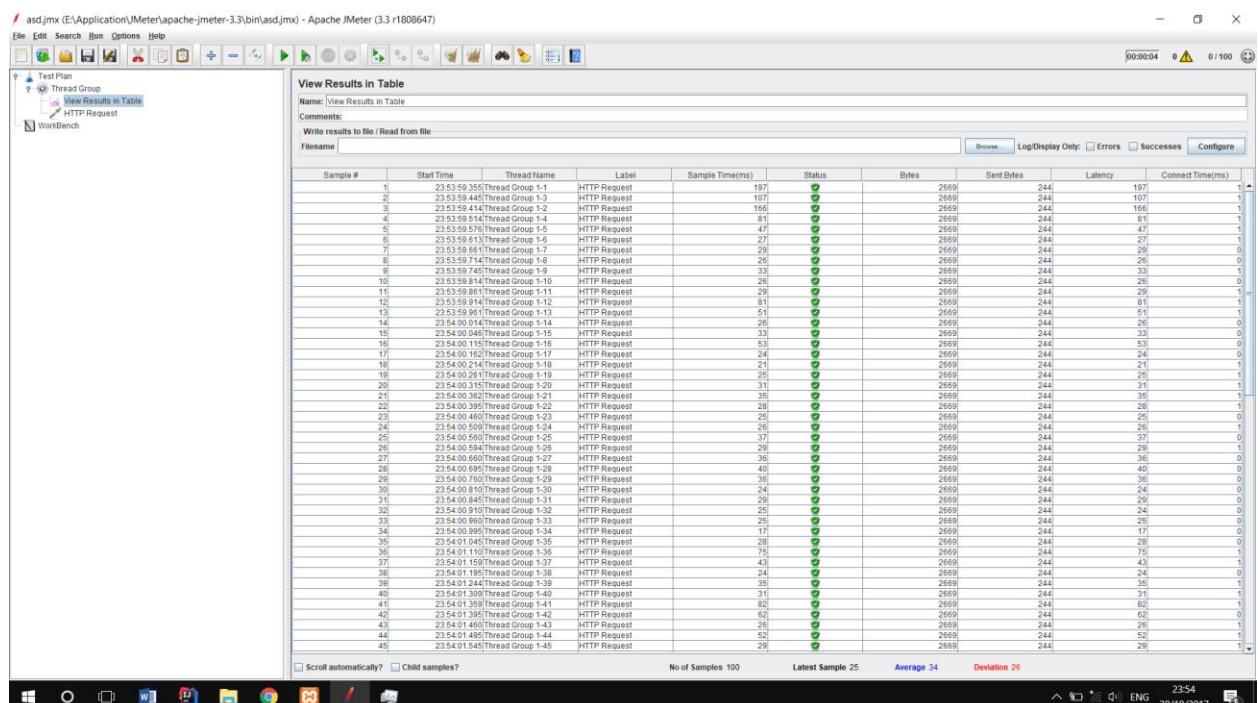
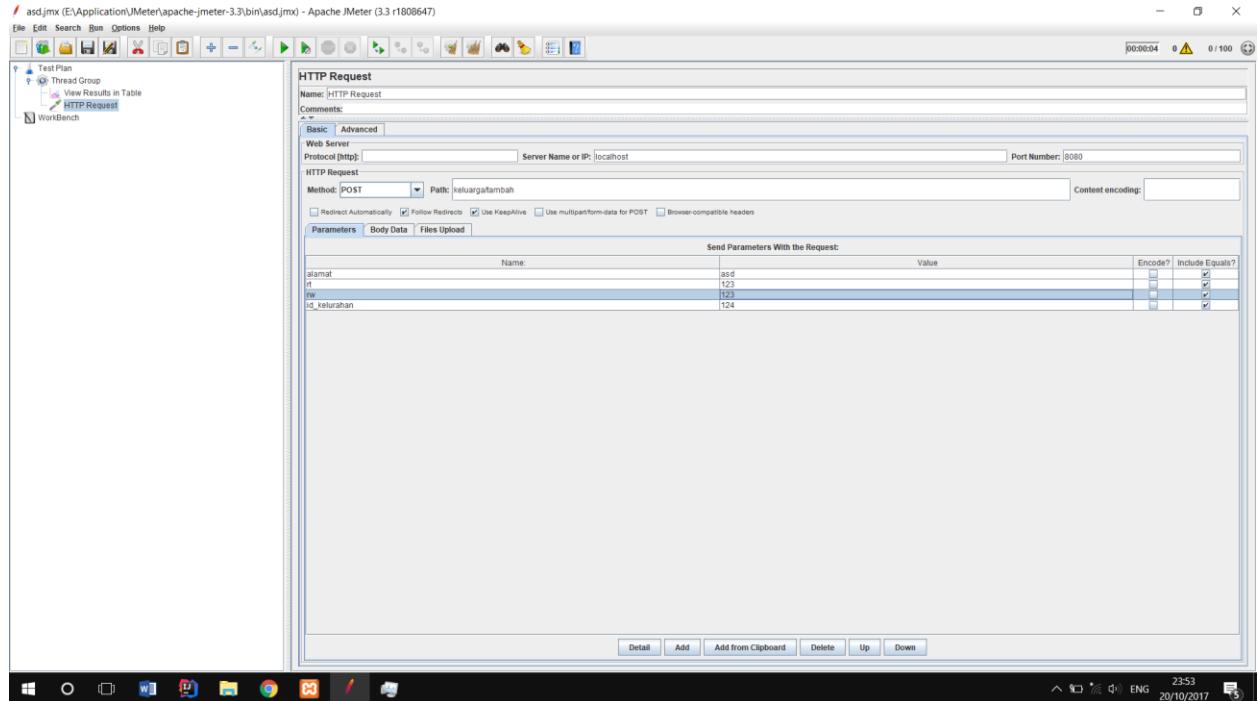
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html, css dan hasil looping mengenai kelurahan kecamatan dan kota yang akan dituju pada web. Hal ini cukup cepat karena penggunaan primary key dan index pada setiap id.





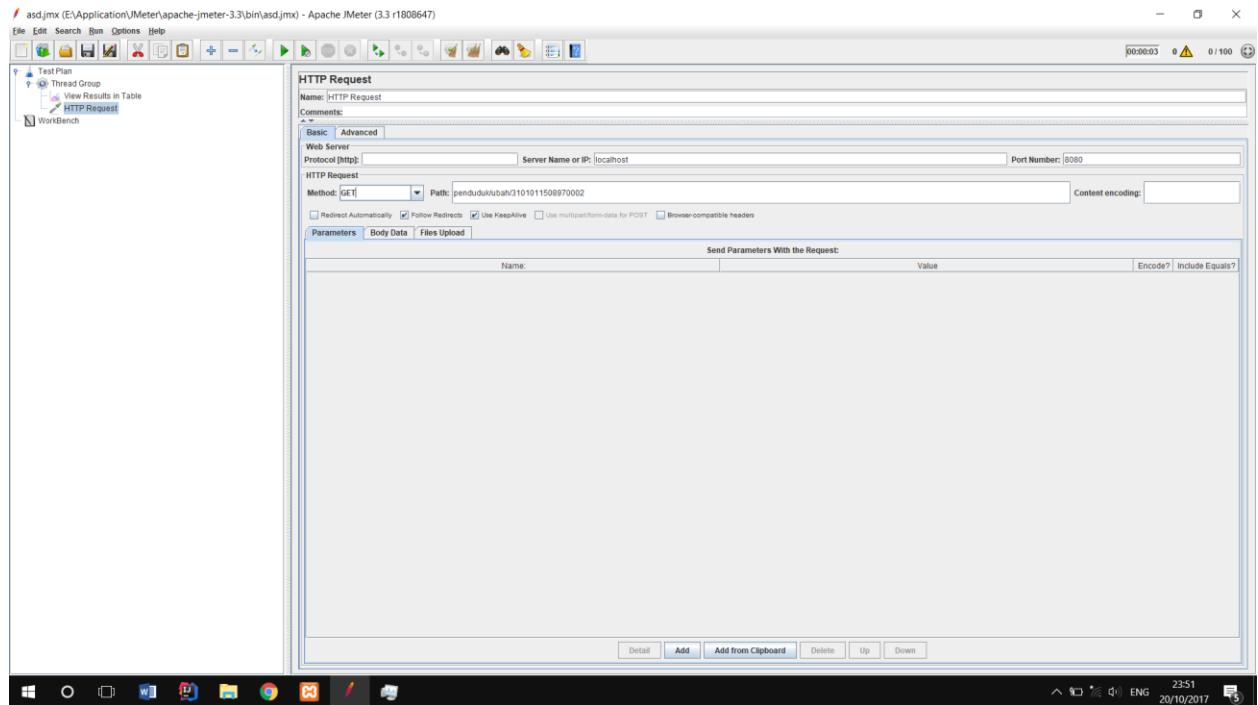
7. Akses tambah keluarga submit

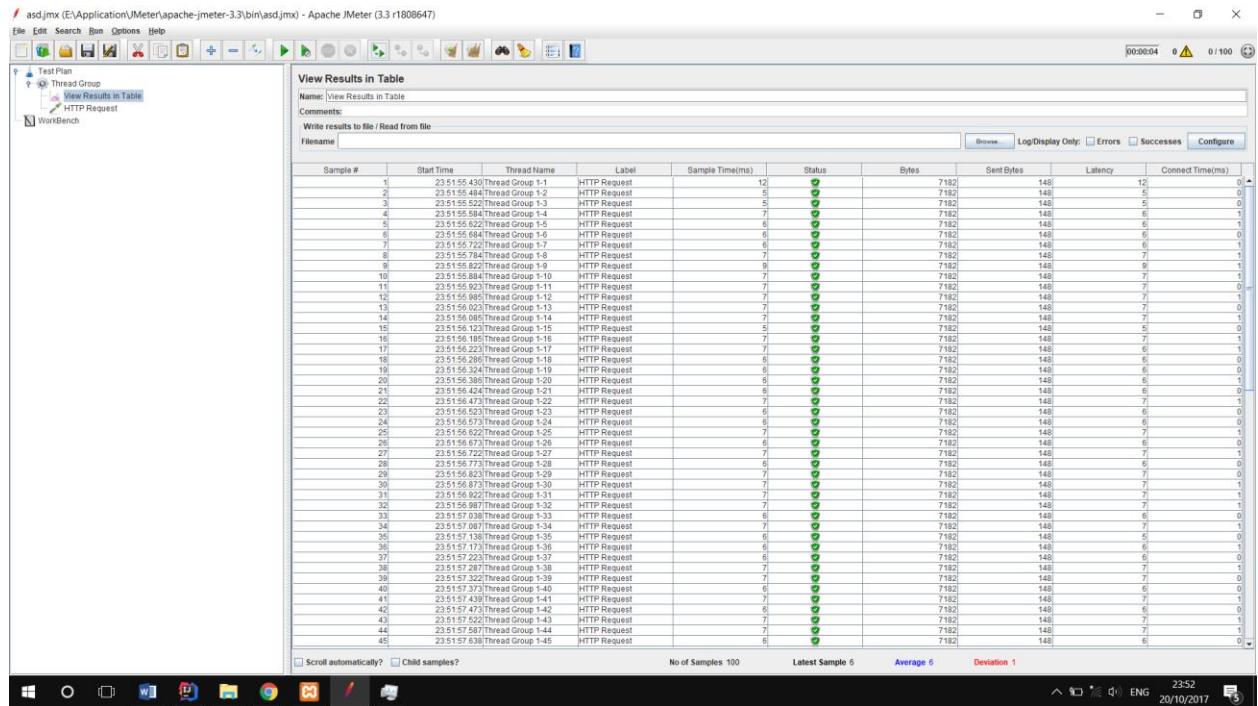
Halaman dapat berjalan dengan optimal hal ini dikarenakan minimnya penggunaan query dan penggunaan primary key, serial, dan index.



8. Akses update penduduk

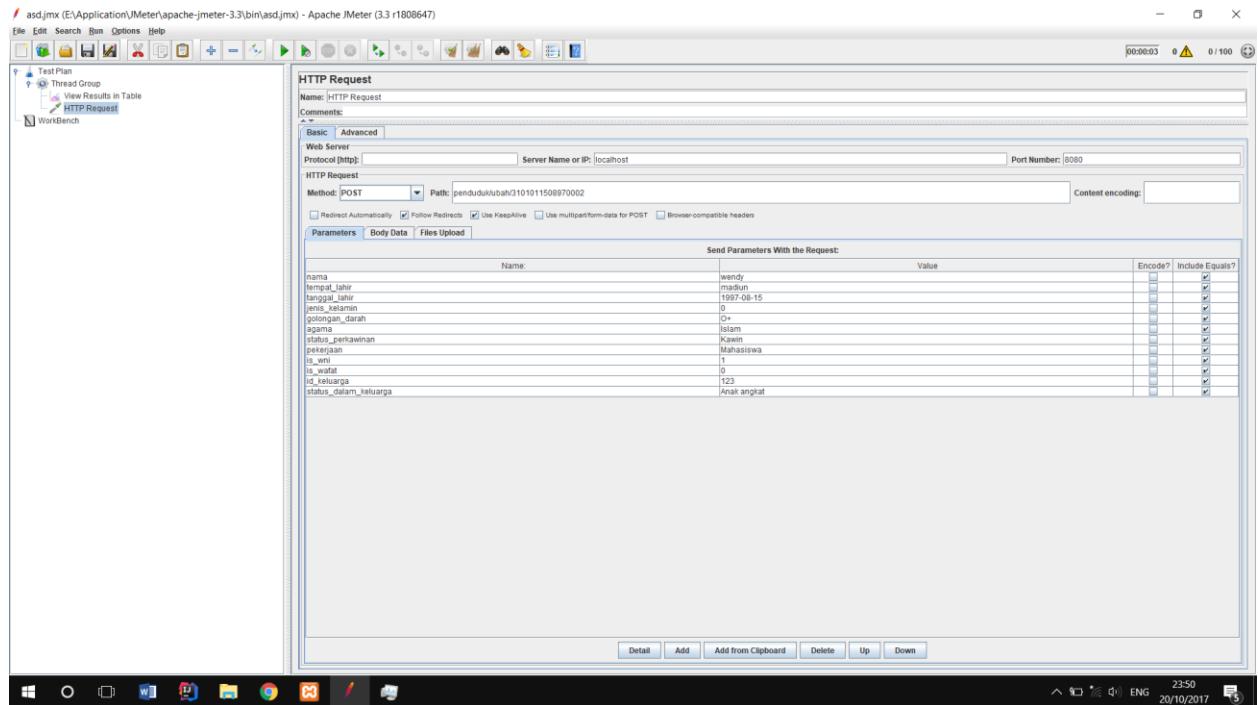
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select penduduk pada web. Dan hal ini didukung dengan penggunaan index pada nik.





9. Akses update penduduk submit

Halaman dapat berjalan dengan optimal hal ini dikarenakan minimnya penggunaan query dan penggunaan primary key dan index.



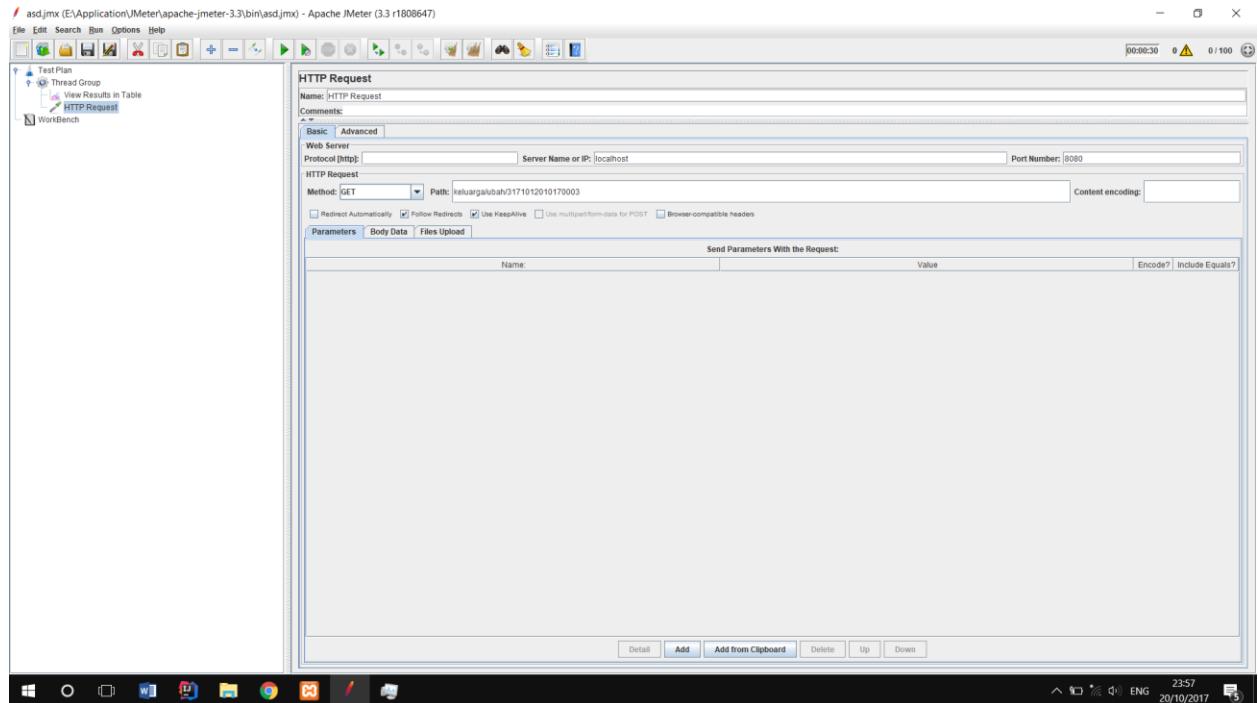
The screenshot shows the JMeter 'View Results in Table' listener. It displays the results of 45 HTTP requests, showing the following columns:

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connected Time(ms)
1	23:49:35.210	Thread Group 1-9	HTTP Request	160	✓	2664	434	160	1
2	23:49:35.217	Thread Group 1-81	HTTP Request	191	✓	2664	434	191	1
3	23:49:35.213	Thread Group 1-32	HTTP Request	310	✓	2664	434	310	0
4	23:49:35.218	Thread Group 1-93	HTTP Request	334	✓	2664	434	334	1
5	23:49:35.220	Thread Group 1-59	HTTP Request	372	✓	2664	434	372	1
6	23:49:35.209	Thread Group 1-3	HTTP Request	421	✓	2664	434	420	1
7	23:49:35.226	Thread Group 1-54	HTTP Request	432	✓	2664	434	432	0
8	23:49:35.215	Thread Group 1-57	HTTP Request	471	✓	2664	434	471	1
9	23:49:35.224	Thread Group 1-72	HTTP Request	486	✓	2664	434	486	0
10	23:49:35.221	Thread Group 1-54	HTTP Request	548	✓	2664	434	548	1
11	23:49:35.220	Thread Group 1-72	HTTP Request	577	✓	2664	434	577	1
12	23:49:35.221	Thread Group 1-68	HTTP Request	604	✓	2664	434	604	1
13	23:49:35.224	Thread Group 1-52	HTTP Request	650	✓	2664	434	650	1
14	23:49:35.222	Thread Group 1-4	HTTP Request	681	✓	2664	434	680	1
15	23:49:35.209	Thread Group 1-1	HTTP Request	722	✓	2664	434	722	2
16	23:49:35.220	Thread Group 1-89	HTTP Request	772	✓	2664	434	772	0
17	23:49:35.220	Thread Group 1-53	HTTP Request	811	✓	2664	434	811	1
18	23:49:35.224	Thread Group 1-6	HTTP Request	856	✓	2664	434	856	1
19	23:49:35.226	Thread Group 1-16	HTTP Request	908	✓	2664	434	908	1
20	23:49:35.224	Thread Group 1-73	HTTP Request	1043	✓	2664	434	1043	1
21	23:49:35.224	Thread Group 1-27	HTTP Request	1087	✓	2664	434	1086	2
22	23:49:35.223	Thread Group 1-16	HTTP Request	1137	✓	2664	434	1137	1
23	23:49:35.231	Thread Group 1-23	HTTP Request	1189	✓	2664	434	1189	0
24	23:49:35.224	Thread Group 1-89	HTTP Request	1226	✓	2664	434	1226	0
25	23:49:35.220	Thread Group 1-8	HTTP Request	1272	✓	2664	434	1272	1
26	23:49:35.224	Thread Group 1-7	HTTP Request	1407	✓	2664	434	1407	1
27	23:49:35.231	Thread Group 1-18	HTTP Request	1450	✓	2664	434	1450	0
28	23:49:35.224	Thread Group 1-48	HTTP Request	1489	✓	2664	434	1489	1
29	23:49:35.211	Thread Group 1-18	HTTP Request	1550	✓	2664	434	1549	0
30	23:49:35.226	Thread Group 1-11	HTTP Request	1594	✓	2664	434	1594	0
31	23:49:35.227	Thread Group 1-13	HTTP Request	1642	✓	2664	434	1642	37
32	23:49:35.224	Thread Group 1-27	HTTP Request	1655	✓	2664	434	1655	1
33	23:49:35.227	Thread Group 1-12	HTTP Request	1721	✓	2664	434	1721	1
34	23:49:35.253	Thread Group 1-64	HTTP Request	1766	✓	2664	434	1766	1
35	23:49:35.241	Thread Group 1-43	HTTP Request	1817	✓	2664	434	1817	2
36	23:49:35.224	Thread Group 1-89	HTTP Request	1857	✓	2664	434	1855	1
37	23:49:35.260	Thread Group 1-85	HTTP Request	1927	✓	2664	434	1926	2
38	23:49:35.239	Thread Group 1-38	HTTP Request	2014	✓	2664	434	2014	1
39	23:49:35.224	Thread Group 1-52	HTTP Request	2038	✓	2664	434	2038	1
40	23:49:35.230	Thread Group 1-20	HTTP Request	2123	✓	2664	434	2123	0
41	23:49:35.259	Thread Group 1-86	HTTP Request	2143	✓	2664	434	2143	1
42	23:49:35.262	Thread Group 1-91	HTTP Request	2191	✓	2664	434	2191	0
43	23:49:35.224	Thread Group 1-17	HTTP Request	2262	✓	2664	434	2262	0
44	23:49:35.240	Thread Group 1-59	HTTP Request	2336	✓	2664	434	2336	1
45	23:49:35.261	Thread Group 1-92	HTTP Request	2364	✓	2664	434	2364	1

Notes: Ramp Up diubah jadi 0.1 second supaya tidak terjadi error karena tidak ditemukan

10. Akses update keluarga

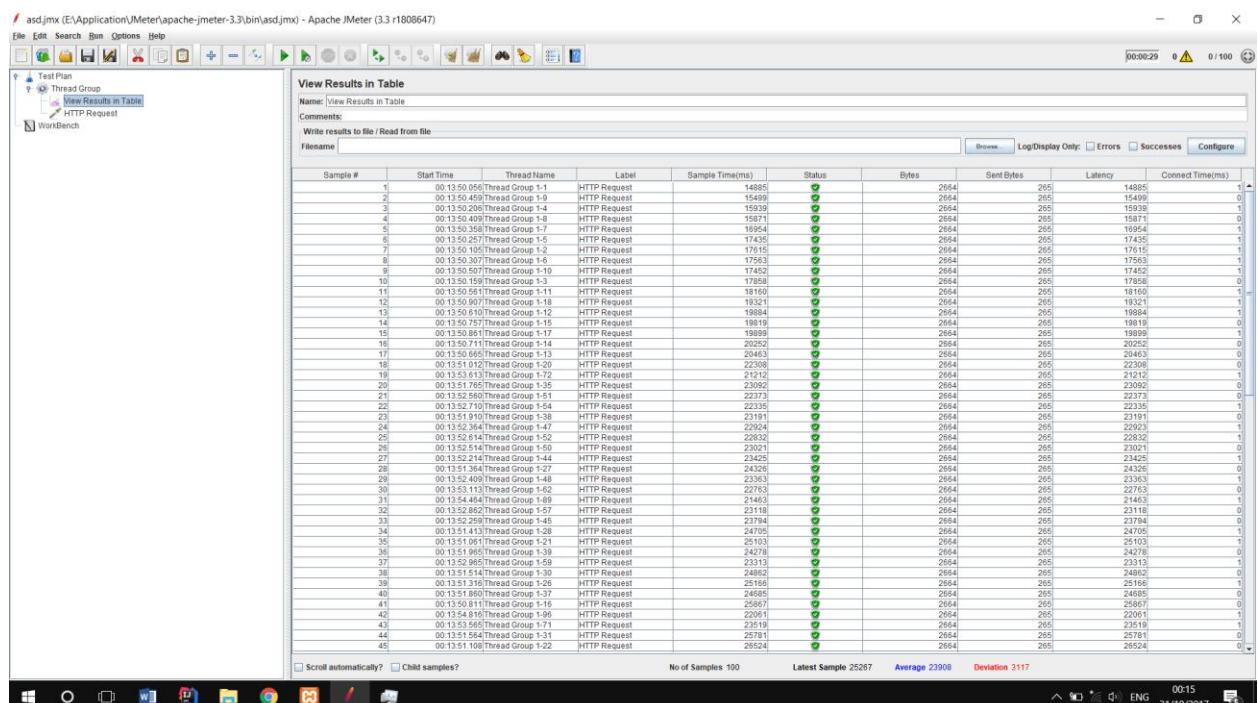
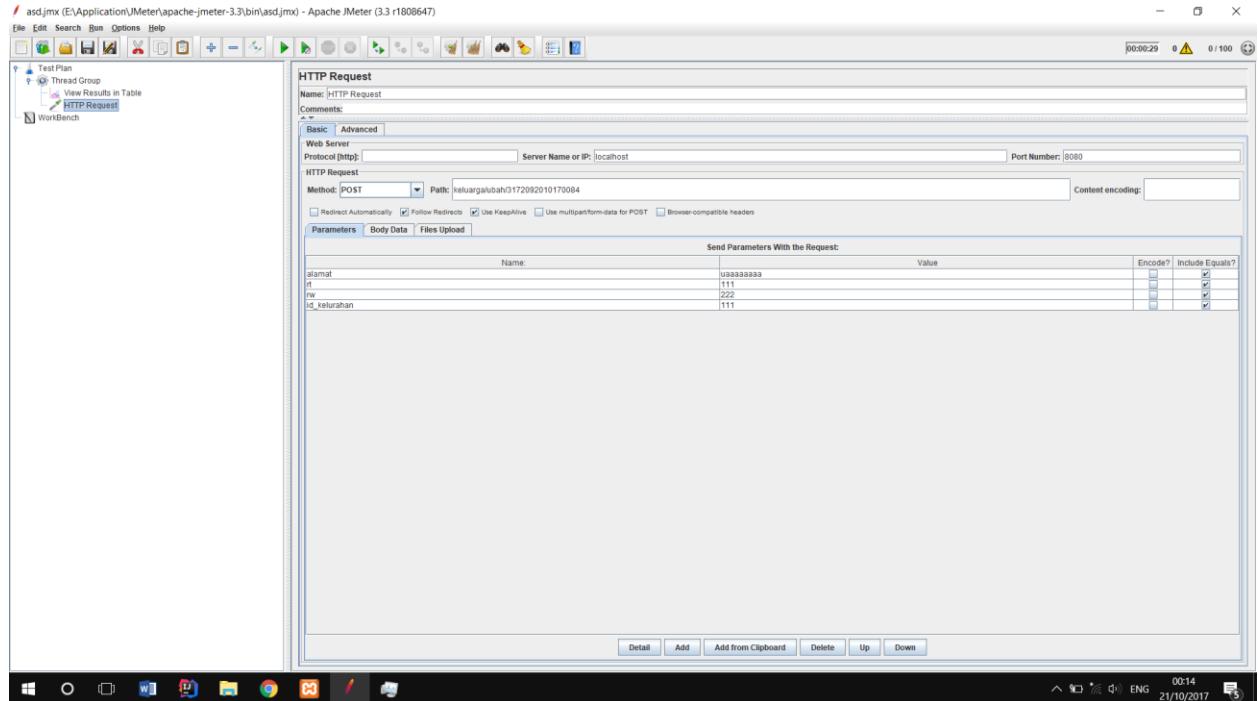
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select keluarga pada web. Dan hal ini didukung dengan penggunaan index pada nkk.



View Results in Table									
Name: View Results in Table									
Comments:									
Write results to file / Read from file									
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connected Time(ms)
1	23:57:35.195	Thread Group 1-1	HTTP Request	36	✓	31299	148	34	1
2	23:57:35.251	Thread Group 1-2	HTTP Request	27	✓	31299	148	26	1
3	23:57:35.305	Thread Group 1-3	HTTP Request	27	✓	31299	148	25	0
4	23:57:35.361	Thread Group 1-4	HTTP Request	29	✓	31299	148	28	1
5	23:57:35.389	Thread Group 1-5	HTTP Request	27	✓	31299	148	27	0
6	23:57:35.451	Thread Group 1-6	HTTP Request	27	✓	31299	148	27	0
7	23:57:35.504	Thread Group 1-7	HTTP Request	28	✓	31299	148	27	1
8	23:57:35.531	Thread Group 1-8	HTTP Request	29	✓	31299	148	28	1
9	23:57:35.589	Thread Group 1-9	HTTP Request	28	✓	31299	148	27	1
10	23:57:35.644	Thread Group 1-10	HTTP Request	27	✓	31299	148	26	0
11	23:57:35.689	Thread Group 1-11	HTTP Request	28	✓	31299	148	27	1
12	23:57:35.736	Thread Group 1-12	HTTP Request	37	✓	31299	148	37	1
13	23:57:35.789	Thread Group 1-13	HTTP Request	28	✓	31299	148	27	1
14	23:57:35.852	Thread Group 1-14	HTTP Request	29	✓	31299	148	29	0
15	23:57:35.890	Thread Group 1-15	HTTP Request	25	✓	31299	148	25	0
16	23:57:35.936	Thread Group 1-16	HTTP Request	28	✓	31299	148	27	1
17	23:57:35.969	Thread Group 1-17	HTTP Request	27	✓	31299	148	27	1
18	23:57:36.051	Thread Group 1-18	HTTP Request	29	✓	31299	148	28	1
19	23:57:36.101	Thread Group 1-19	HTTP Request	28	✓	31299	148	27	1
20	23:57:36.151	Thread Group 1-20	HTTP Request	27	✓	31299	148	26	1
21	23:57:36.189	Thread Group 1-21	HTTP Request	41	✓	31299	148	40	1
22	23:57:36.238	Thread Group 1-22	HTTP Request	28	✓	31299	148	28	1
23	23:57:36.285	Thread Group 1-23	HTTP Request	27	✓	31299	148	26	0
24	23:57:36.339	Thread Group 1-24	HTTP Request	27	✓	31299	148	27	0
25	23:57:36.389	Thread Group 1-25	HTTP Request	33	✓	31299	148	32	0
26	23:57:36.429	Thread Group 1-26	HTTP Request	28	✓	31299	148	27	0
27	23:57:36.476	Thread Group 1-27	HTTP Request	30	✓	31299	148	30	0
28	23:57:36.539	Thread Group 1-28	HTTP Request	30	✓	31299	148	30	1
29	23:57:36.588	Thread Group 1-29	HTTP Request	28	✓	31299	148	27	1
30	23:57:36.639	Thread Group 1-30	HTTP Request	28	✓	31299	148	27	1
31	23:57:36.689	Thread Group 1-31	HTTP Request	29	✓	31299	148	28	1
32	23:57:36.738	Thread Group 1-32	HTTP Request	29	✓	31299	148	28	1
33	23:57:36.788	Thread Group 1-33	HTTP Request	28	✓	31299	148	27	1
34	23:57:36.838	Thread Group 1-34	HTTP Request	27	✓	31299	148	27	1
35	23:57:36.885	Thread Group 1-35	HTTP Request	27	✓	31299	148	26	0
36	23:57:36.939	Thread Group 1-36	HTTP Request	28	✓	31299	148	27	0
37	23:57:36.989	Thread Group 1-37	HTTP Request	28	✓	31299	148	27	0
38	23:57:37.039	Thread Group 1-38	HTTP Request	27	✓	31299	148	27	0
39	23:57:37.089	Thread Group 1-39	HTTP Request	27	✓	31299	148	27	0
40	23:57:37.138	Thread Group 1-40	HTTP Request	28	✓	31299	148	27	1
41	23:57:37.187	Thread Group 1-41	HTTP Request	30	✓	31299	148	29	1
42	23:57:37.238	Thread Group 1-42	HTTP Request	28	✓	31299	148	27	0
43	23:57:37.288	Thread Group 1-43	HTTP Request	28	✓	31299	148	27	1
44	23:57:37.338	Thread Group 1-44	HTTP Request	27	✓	31299	148	27	0
45	23:57:37.388	Thread Group 1-45	HTTP Request	27	✓	31299	148	26	0

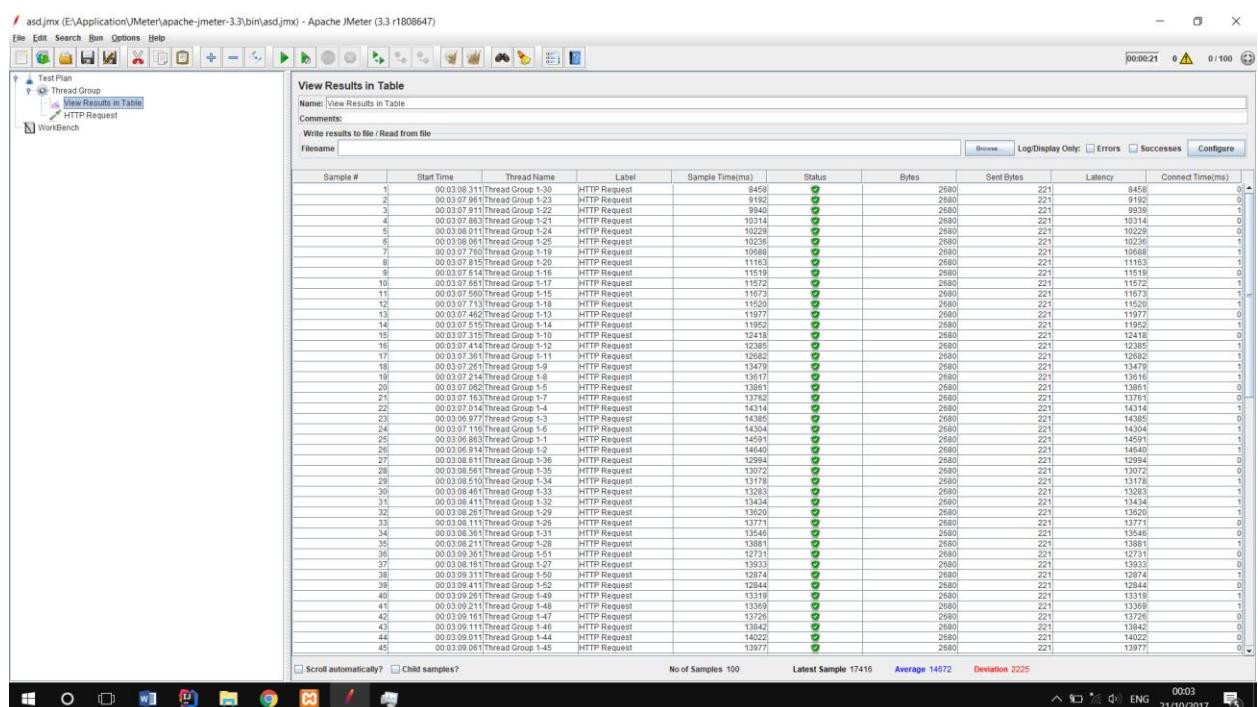
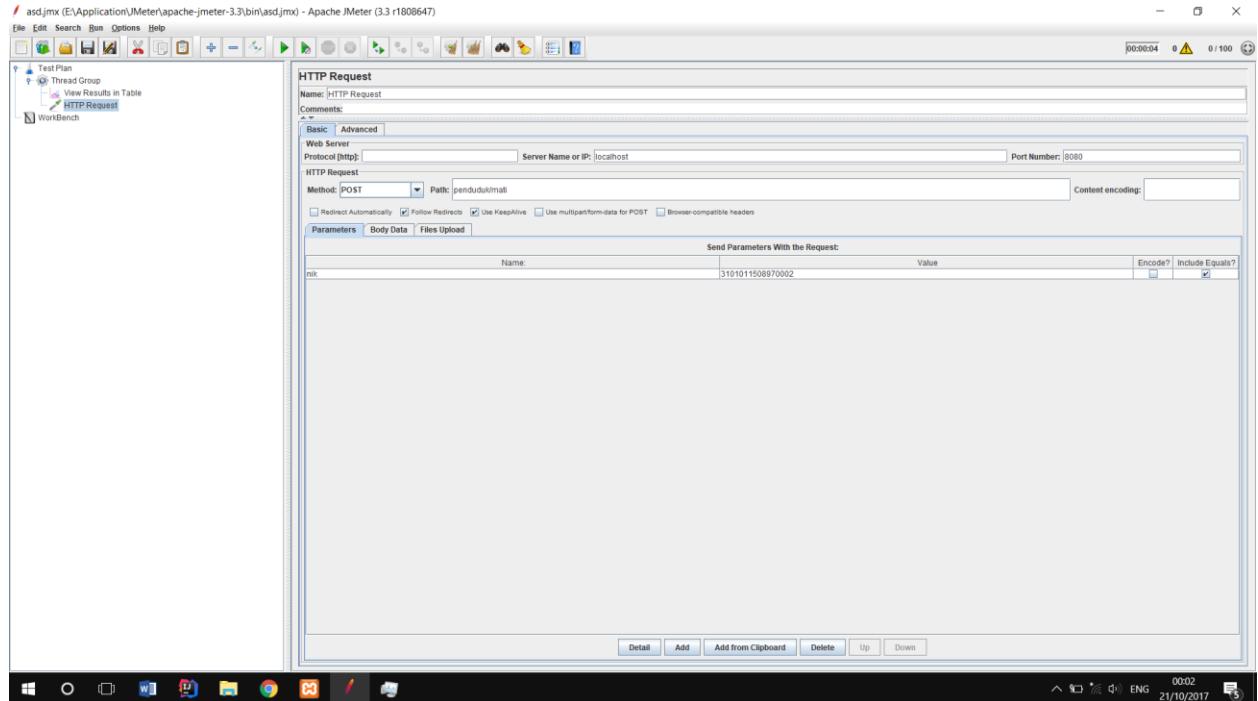
11. Akses update keluarga submit

Halaman berjalan cukup lambat namun cukup optimal jika dilihat dari seberapa banyak fungsi yang digunakan. Walaupun telah dibantu oleh optimisasi database. Hal ini dikarenakan banyaknya fungsi yang digunakan dan banyaknya query.



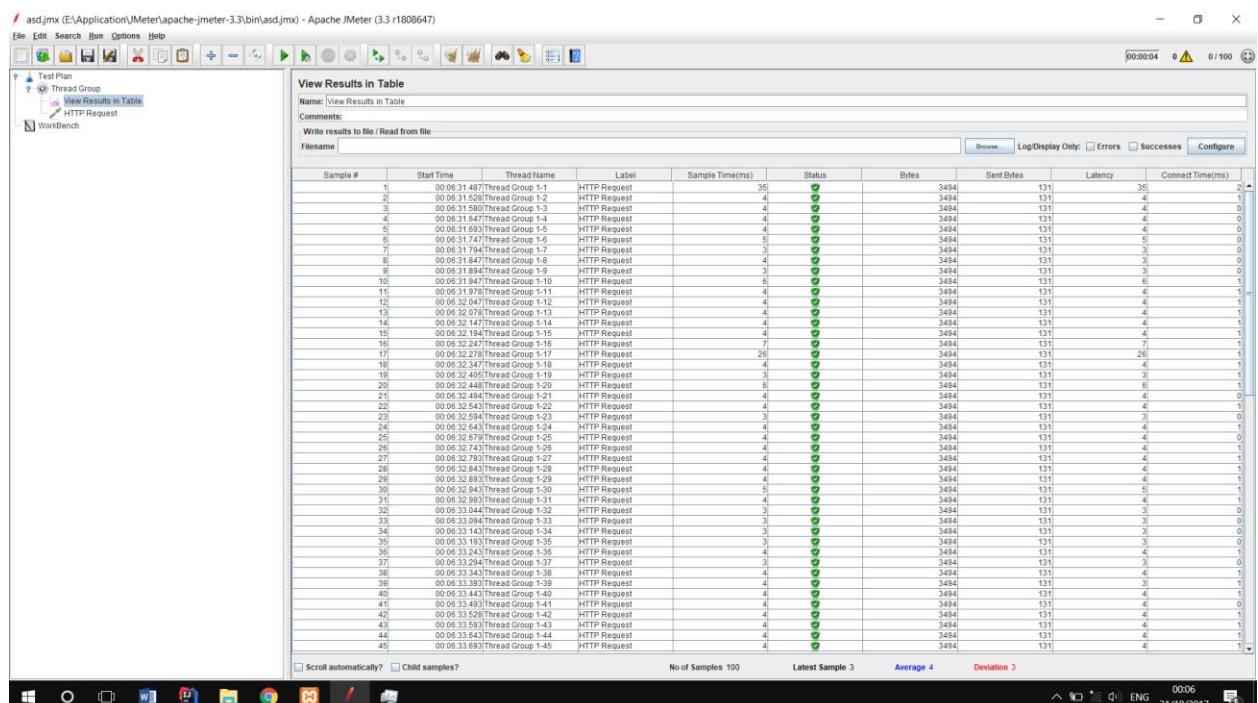
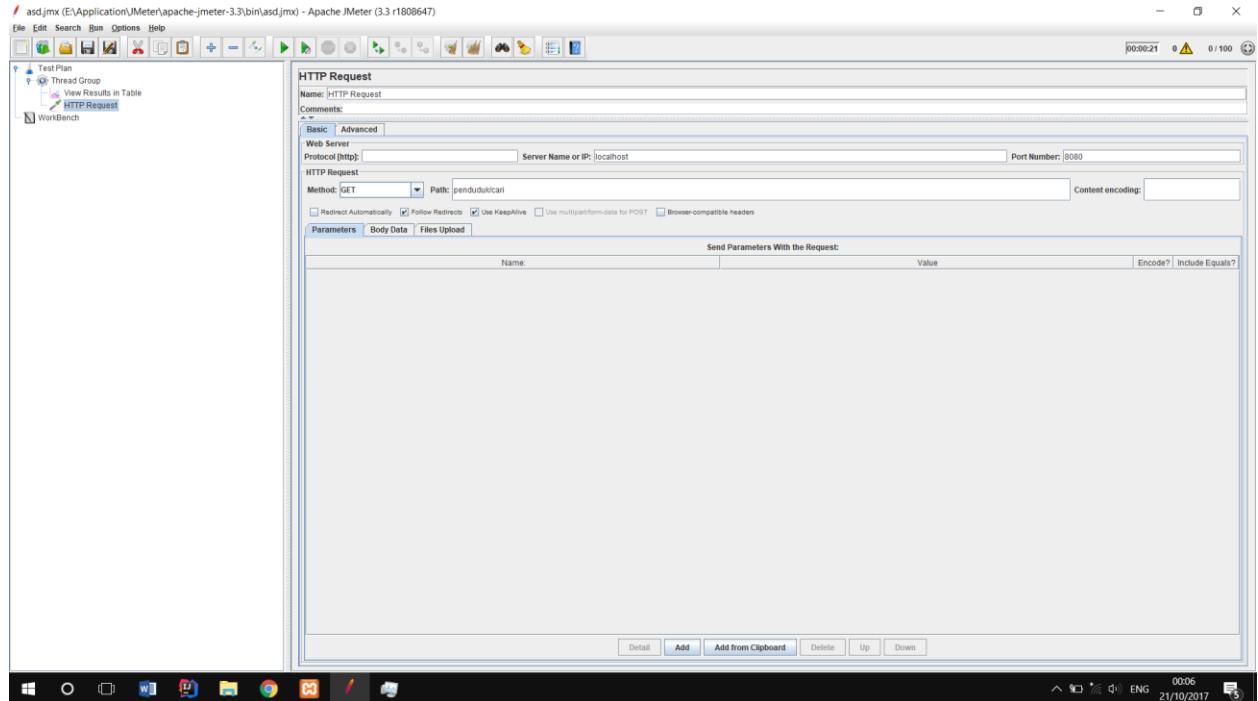
12. Akses nonaktifkan penduduk

Program berjalan kurang optimal dikarenakan banyaknya fungsi yang berkaitan mengenai cek keluarga dan penduduk sehingga terlalu banyak query dan operasi yang digunakan.



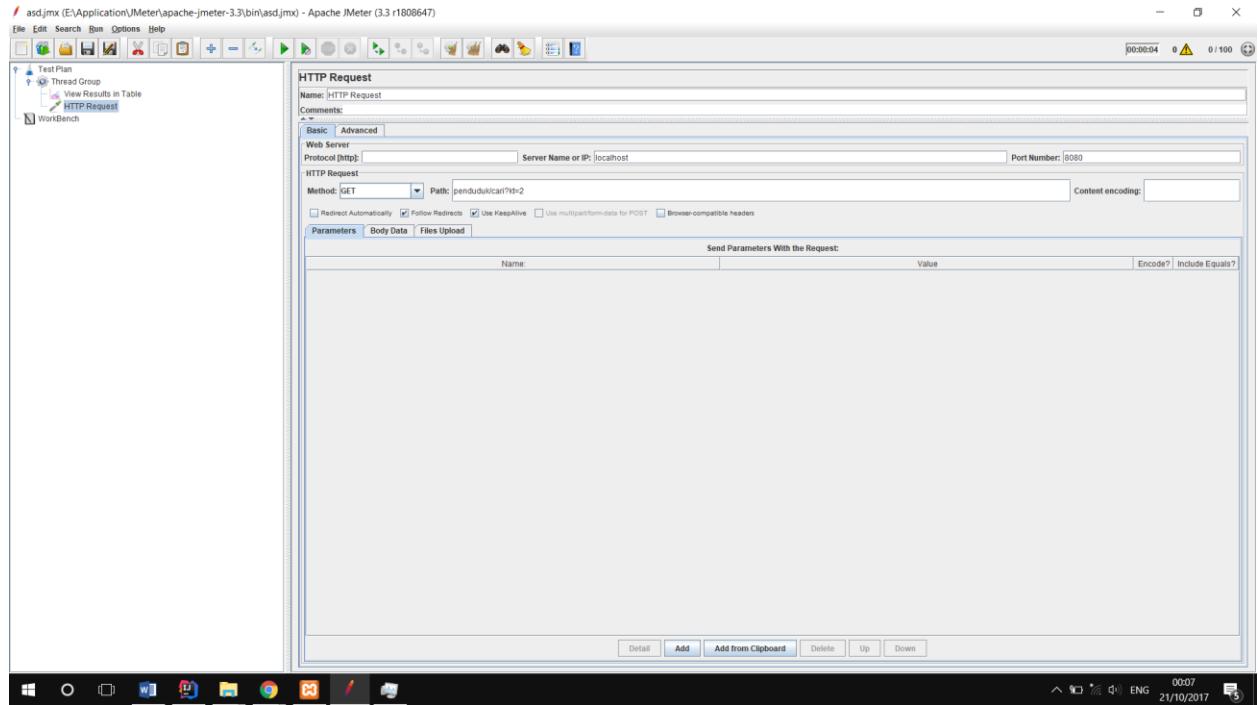
13. Akses cari penduduk kota

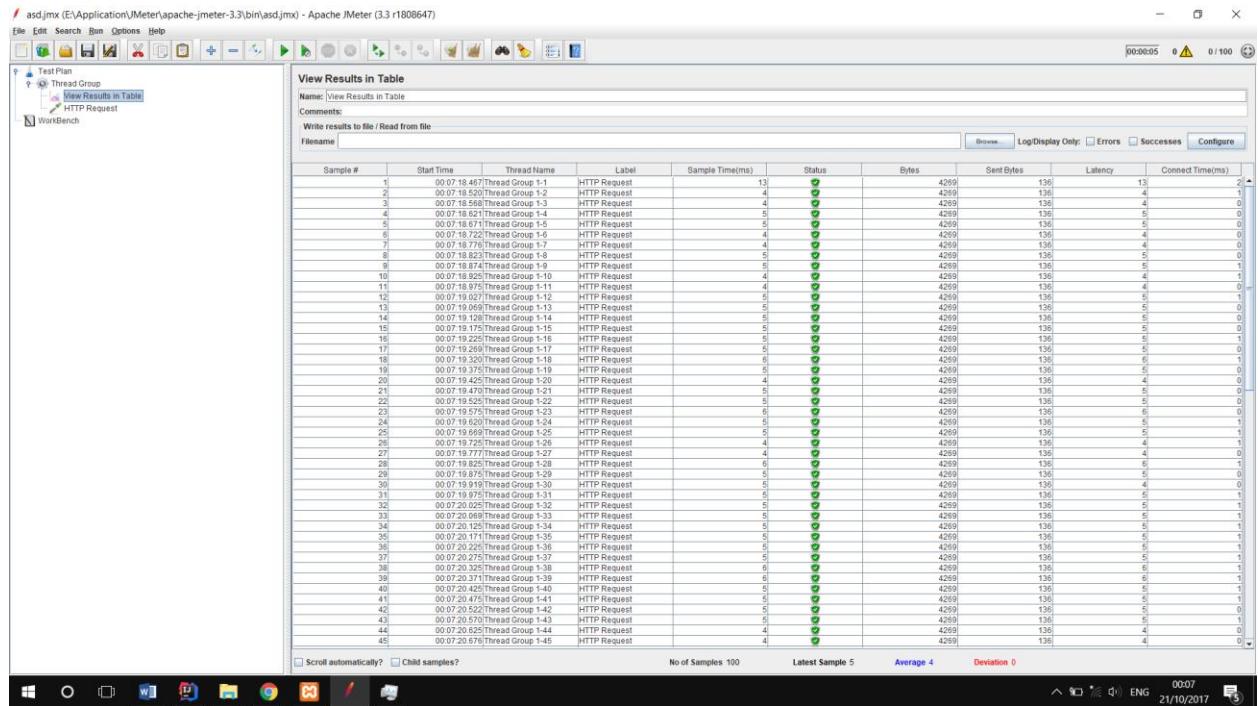
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select semua kota pada web. Dan hal ini didukung dengan penggunaan index pada kota.



14. Akses cari penduduk kecamatan

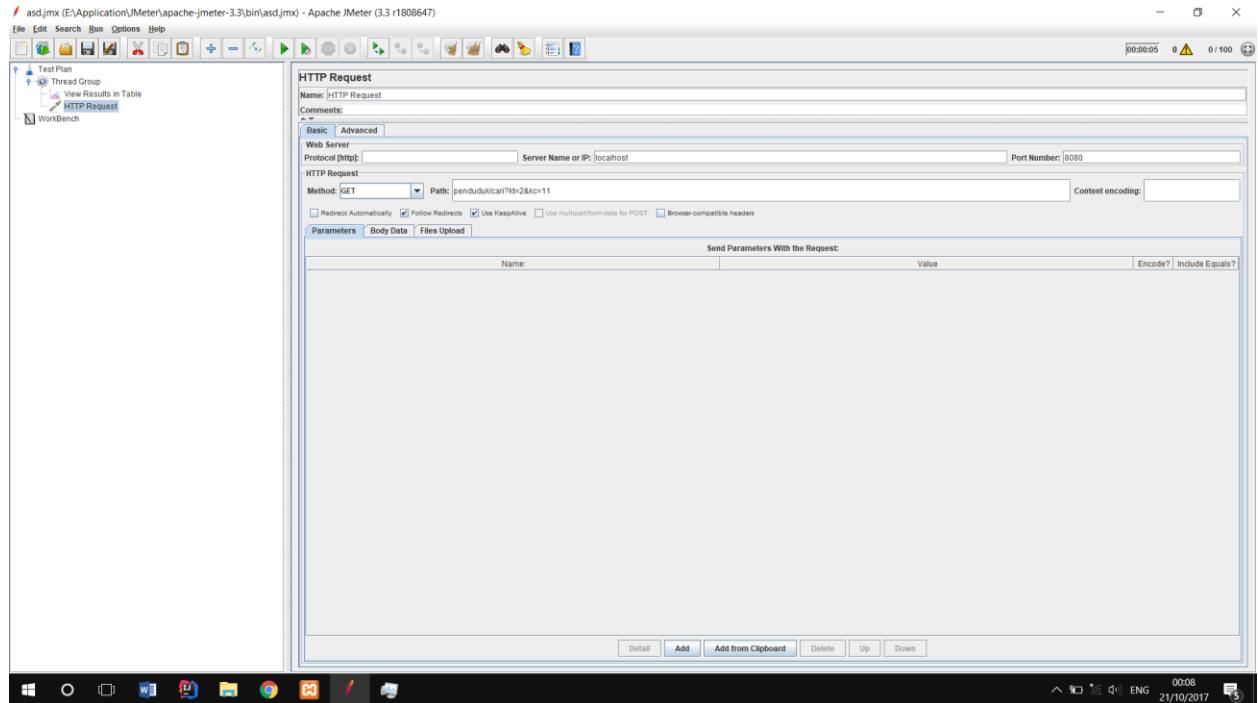
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select semua kota pada kecamatan. Dan hal ini didukung dengan penggunaan index pada kecamatan.

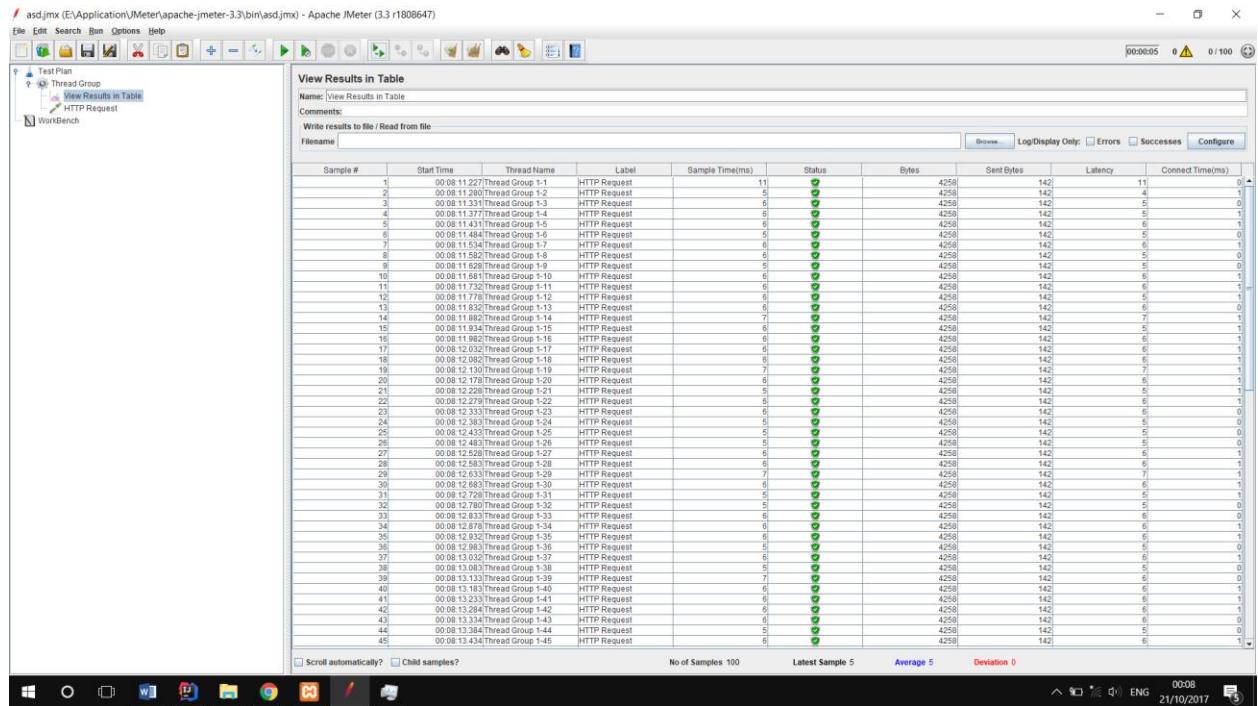




15. Akses cari penduduk kelurahan

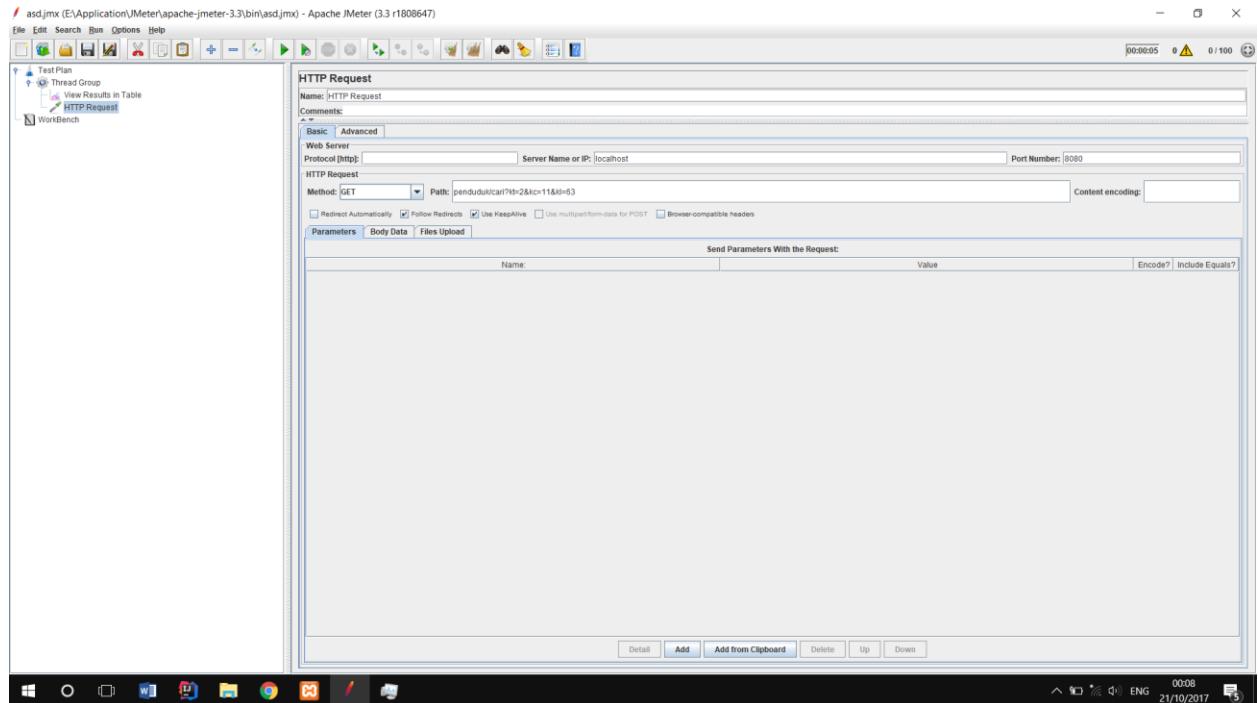
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select semua kelurahan pada web. Dan hal ini didukung dengan penggunaan index pada kelurahan.





16. Akses hasil penduduk

Halaman dapat berjalan kurang optimal dikarenakan banyaknya data dan penggunaan loop yang besar demi hanya mendapatkan penduduk tertua dan termuda.



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connected Time(ms)
1	00:09:03.731	Thread Group 1-12	HTTP Request	19124	✓	194141	148	19116	0
2	00:09:03.683	Thread Group 1-12	HTTP Request	18173	✓	194141	148	18166	1
3	00:09:03.834	Thread Group 1-10	HTTP Request	18224	✓	194141	148	18217	0
4	00:09:03.645	Thread Group 1-4	HTTP Request	18242	✓	194141	148	18358	1
5	00:09:03.580	Thread Group 1-9	HTTP Request	19367	✓	194141	148	19303	1
6	00:09:03.430	Thread Group 1-6	HTTP Request	20251	✓	194141	148	20252	0
7	00:09:03.290	Thread Group 1-3	HTTP Request	20415	✓	194141	148	20408	1
8	00:09:03.220	Thread Group 1-2	HTTP Request	20500	✓	194141	148	21127	1
9	00:09:03.230	Thread Group 1-2	HTTP Request	21686	✓	194141	148	21674	1
10	00:09:03.382	Thread Group 1-5	HTTP Request	21735	✓	194141	148	21729	1
11	00:09:03.934	Thread Group 1-11	HTTP Request	22494	✓	194141	148	22462	0
12	00:09:03.880	Thread Group 1-8	HTTP Request	22545	✓	194141	148	23557	1
13	00:09:03.180	Thread Group 1-1	HTTP Request	27726	✓	194141	148	27719	1
14	00:09:03.881	Thread Group 1-18	HTTP Request	27025	✓	194141	148	27015	0
15	00:09:03.984	Thread Group 1-17	HTTP Request	28022	✓	194141	148	28014	0
16	00:09:03.985	Thread Group 1-20	HTTP Request	28151	✓	194141	148	32148	0
17	00:09:04.785	Thread Group 1-33	HTTP Request	31516	✓	194141	148	31510	1
18	00:09:03.484	Thread Group 1-7	HTTP Request	32851	✓	194141	148	32845	0
19	00:09:05.413	Thread Group 1-46	HTTP Request	30906	✓	194141	148	30898	0
20	00:09:05.414	Thread Group 1-22	HTTP Request	32300	✓	194141	148	32270	1
21	00:09:05.682	Thread Group 1-51	HTTP Request	30791	✓	194141	148	30782	1
22	00:09:05.337	Thread Group 1-44	HTTP Request	31182	✓	194141	148	31175	0
23	00:09:05.437	Thread Group 1-32	HTTP Request	31711	✓	194141	148	31784	0
24	00:09:04.435	Thread Group 1-29	HTTP Request	32103	✓	194141	148	32097	0
25	00:09:05.222	Thread Group 1-42	HTTP Request	31309	✓	194141	148	31303	1
26	00:09:05.062	Thread Group 1-38	HTTP Request	31443	✓	194141	148	31457	1
27	00:09:05.063	Thread Group 1-39	HTTP Request	32144	✓	194141	148	32768	0
28	00:09:05.833	Thread Group 1-54	HTTP Request	31725	✓	194141	148	30719	0
29	00:09:04.987	Thread Group 1-37	HTTP Request	31576	✓	194141	148	31571	0
30	00:09:05.633	Thread Group 1-60	HTTP Request	30938	✓	194141	148	30932	0
31	00:09:05.634	Thread Group 1-61	HTTP Request	31931	✓	194141	148	32323	1
32	00:09:04.887	Thread Group 1-31	HTTP Request	32074	✓	194141	148	32087	0
33	00:09:04.296	Thread Group 1-23	HTTP Request	33404	✓	194141	148	33393	1
34	00:09:04.086	Thread Group 1-19	HTTP Request	33550	✓	194141	148	33537	0
35	00:09:04.917	Thread Group 1-27	HTTP Request	33590	✓	194141	148	33520	1
36	00:09:05.917	Thread Group 1-66	HTTP Request	33757	✓	194141	148	33750	0
37	00:09:04.335	Thread Group 1-24	HTTP Request	35360	✓	194141	148	36353	1
38	00:09:04.535	Thread Group 1-28	HTTP Request	35168	✓	194141	148	36158	1
39	00:09:05.423	Thread Group 1-47	HTTP Request	35250	✓	194141	148	36250	0
40	00:09:05.533	Thread Group 1-48	HTTP Request	34193	✓	194141	148	34191	0
41	00:09:05.187	Thread Group 1-41	HTTP Request	34550	✓	194141	148	34540	0
42	00:09:04.293	Thread Group 1-13	HTTP Request	34703	✓	194141	148	34701	0
43	00:09:05.263	Thread Group 1-43	HTTP Request	34714	✓	194141	148	34758	1
44	00:09:05.264	Thread Group 1-49	HTTP Request	34487	✓	194141	148	34458	1
45	00:09:07.187	Thread Group 1-81	HTTP Request	34219	✓	194141	148	34212	0

Penjelasan MVC

Pada program kali ini saya menyiapkan program dengan model MVC. Dimana terdapat 3 bagian penting yaitu Model, View, dan Controller. Model merupakan bagian dimana setiap objek yang ada untuk program akan dibuat. Dimana dia akan menampung data dari database yang tersambung dengan program. Model pun dibuat berdasarkan banyaknya skema table yang terdapat pada database. Dan pada setiap model diberikan suatu atribut yang mewakili model tersebut dan diberikan constructor dan getter setter. Dan pada setiap atribut diberikan batas seperti apakah boleh null atau sizenya berapa agar dapat digunakan pada Controller untuk melakukan pengecekan kevalidan suatu model.

Lalu pada View terdapat 2 template utama yaitu static dan juga templates. Static berisi kumpulan css dan js yang akan digunakan pada program tersebut. Sementara pada view terdapat 2 bagian utama yaitu includes dan juga view, dimana includes berisi fragmen yang akan digunakan pada program yang sering muncul contoh header, navbar, dan juga footer supaya proses coding menjadi lebih rapih dan enak dilihat. Lalu View berisi berbagai halaman yang digunakan pada program, mulai dari hasil pencarian penduduk, form update, dan lain-lain. Selain itu juga terdapat error.html dimana ia akan mengambil semua error pada program dan akan mengirimkan halaman ini sebagai feedbacknya.

Dan yang terakhir ialah Controller. Pada program ini Controller dibagi berdasarkan berbagai hal, pertama dibagiberdasarkan fungsinya dan kedua berdasarkan halaman atau model yang dituju. Pada layer pertama dibagi berdasarkan fungsinya dimana terdapat 4 fungsi utama yaitu Controller halaman, DAO (Data Access Object), dan Service.

Controller halaman merupakan suatu Controller dimana ia akan menerima respon dari html ingin kemana ia berada dan memproses hal tersebut dan juga memberikan arah ia ingin kemana setelah memproses hal tersebut. Dari hal ini terdapat 3 bagian Controller halaman yaitu IndexController, PendudukController, dan juga KeluargaController. IndexController digunakan sebagai Controller halaman pada halaman utama dan konfirmasi setiap hasil rekues program. Lalu pada PendudukController berfungsi sebagai Controller halaman yang memiliki keterlibatan pada fungsi pada penduduk (dalam kasus ini cari, lihat, update, tambah dan nonaktif). Sementara pada KeluargaController berisi keterlibatan fungsi pada keluarga (dalam kasus ini lihat, tambah, dan update).

Lalu pada bagian DAO, berisi berbagai query table pada tiap skema table. Hal ini dibagi berdasarkan skema pada table tersebut. Jika ia ingin mengakses data Penduduk maka ia akan masuk pada PendudukDAO, dan jika ia ingin mengambil data kelurahan maka ia akan mengambil pada KelurahanDAO.

Setelah itu terdapat Service dimana ia merupakan page tambahan dimana ia menjadi penyambung antara Controller dan DAO. Hal ini dibutuhkan agar meningkatkan decoupling dan menurunkan cohesion karena penggunaan Interface. Dan juga mempermudah kita apabila kita ingin merubah Query kita cukup membuat DAO dan implement service yang baru.

Dan yang terakhir terdapat application.properties dimana ia akan menyambungkan kita pada database terkait.