

# 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

---

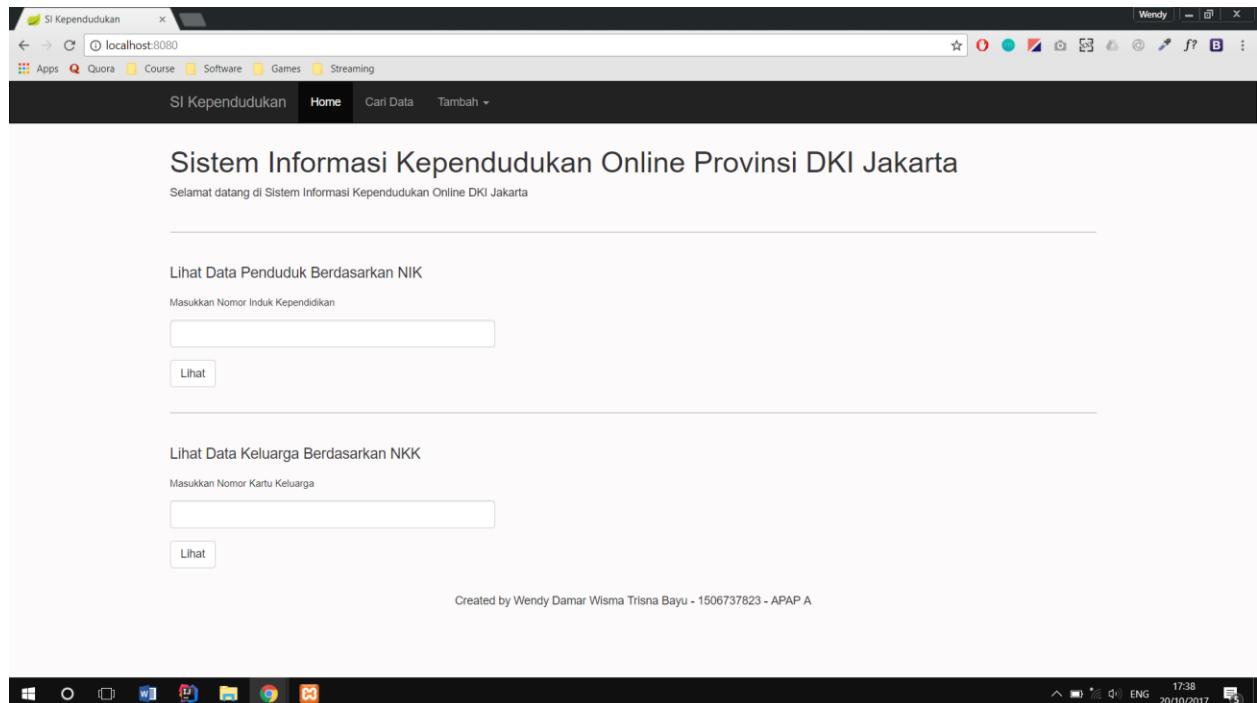
### 1. Tampilkan Data Penduduk Berdasarkan NIK

**Initial Page:** /

**Form Request:** GET, /penduduk

**Response Page:** /penduduk?nik={NIK}

Pada fitur ini saya menggunakan form get dimana form ini akan memassing 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.

Lihat Data Penduduk - 3101010101800001

Update

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

Nonaktifkan

Lihat Keluarga

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.

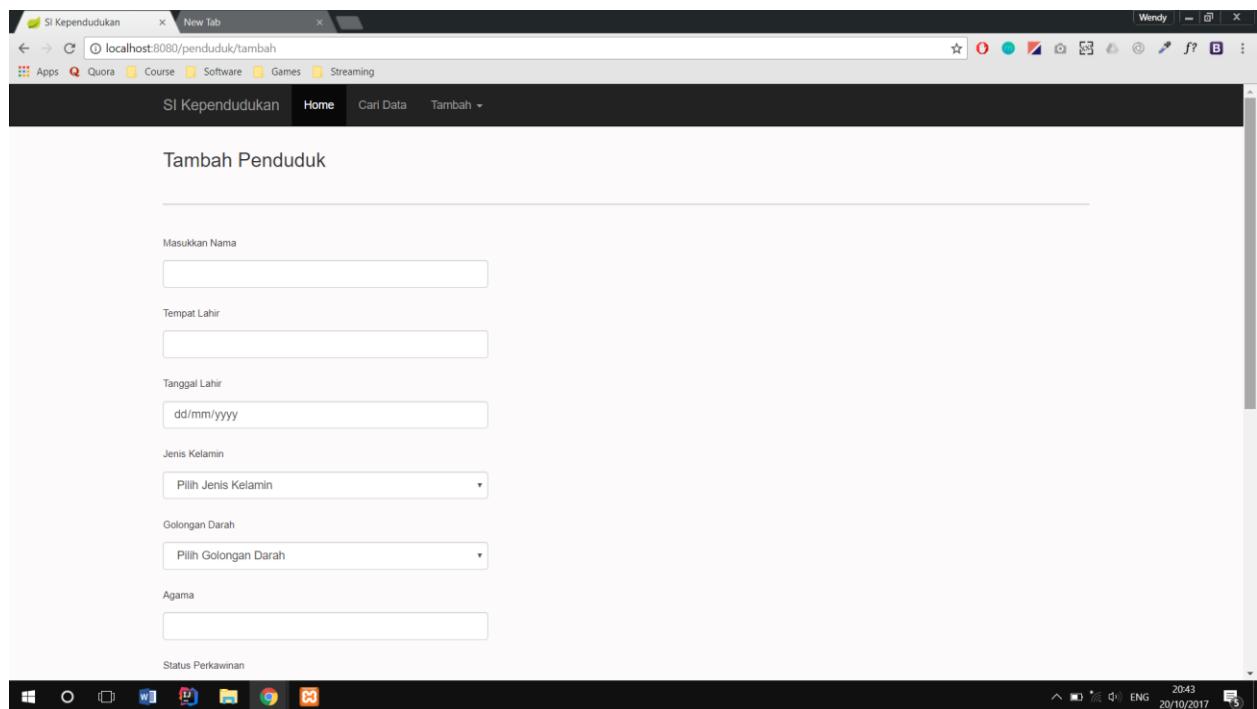
### 3. Menambahkan Penduduk Baru Sebagai Anggota Keluarga

**Initial Page:** /penduduk/tambah

**Form Request:** POST, /penduduk/tambah

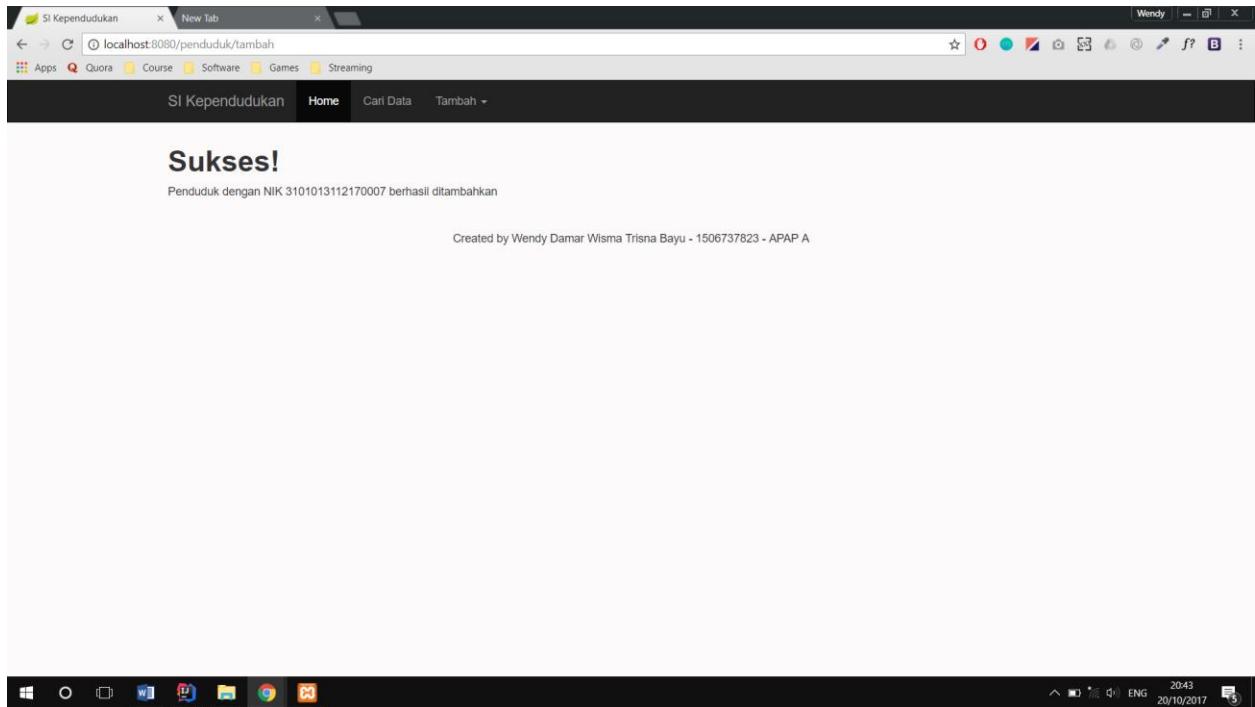
**Response Page:** /penduduk/tambah

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.

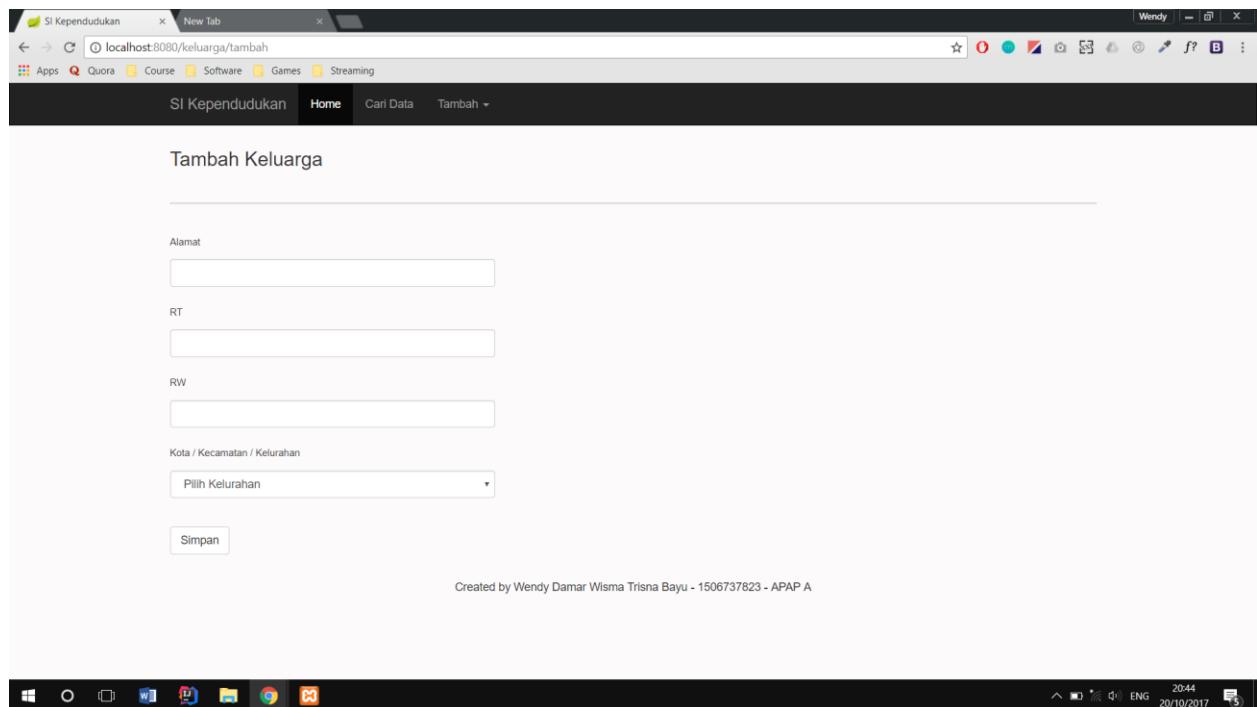
#### 4. Menambahkan Keluarga Baru

**Initial Page:** /keluarga/tambah

**Form Request:** POST, /keluarga/tambah

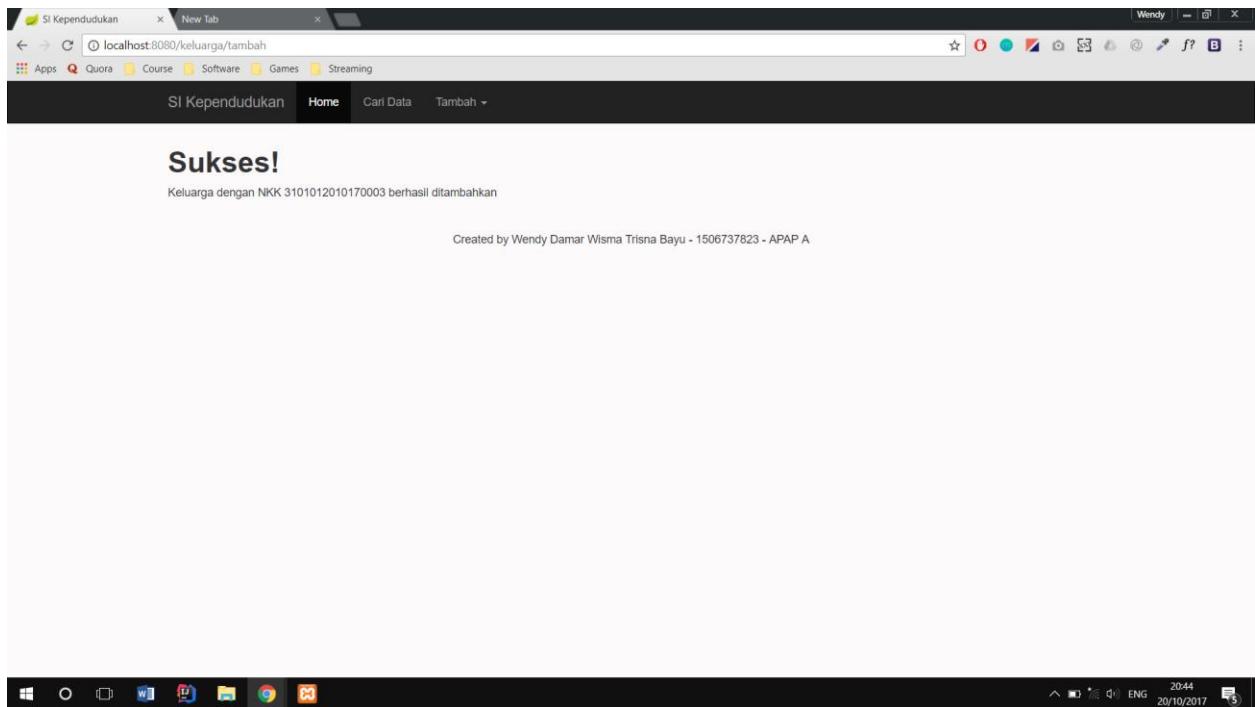
**Response Page:** /keluarga/tambah

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.

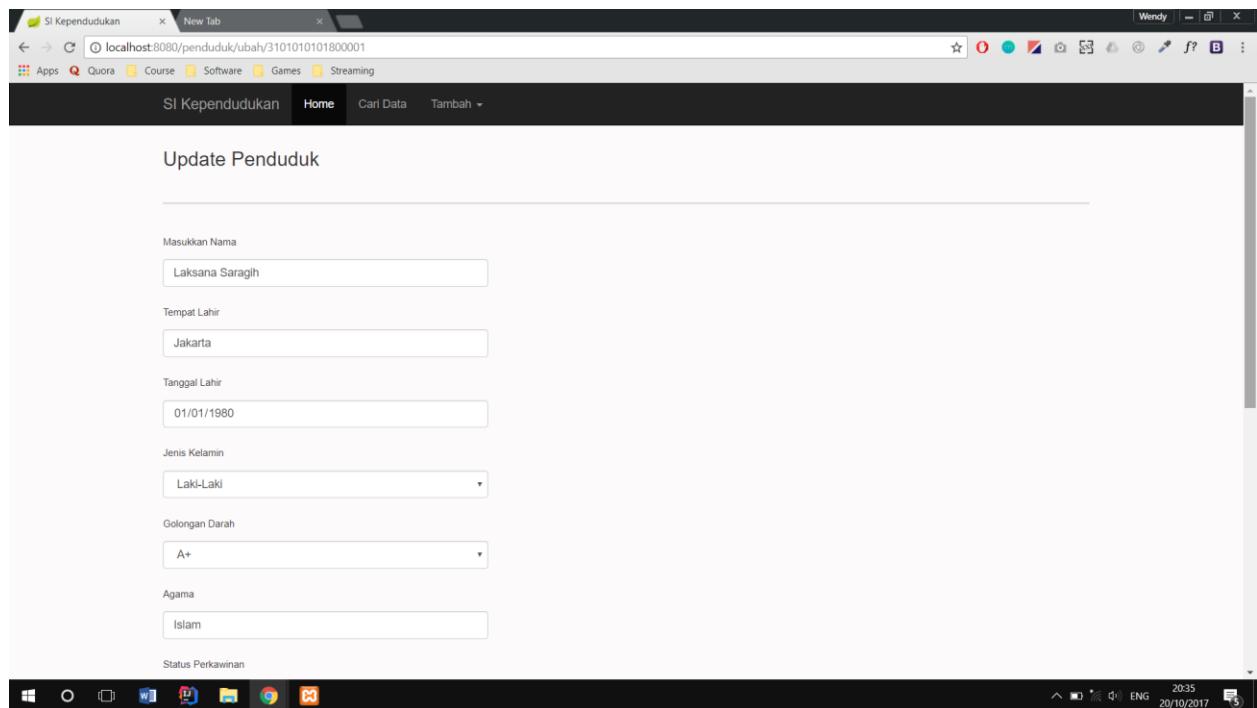
## 5. Mengubah Data Penduduk

**Initial Page:** /penduduk/ubah/{NIK}

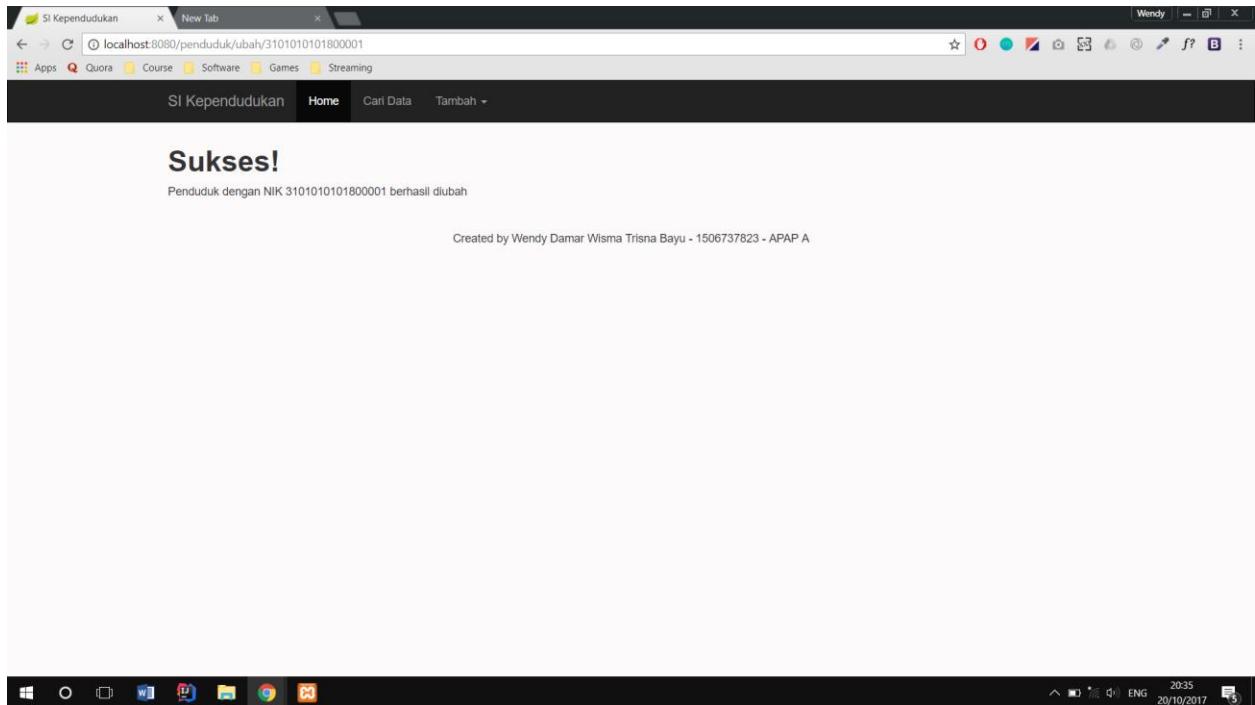
**Form Request:** POST, /penduduk/ubah/{NIK}

**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.



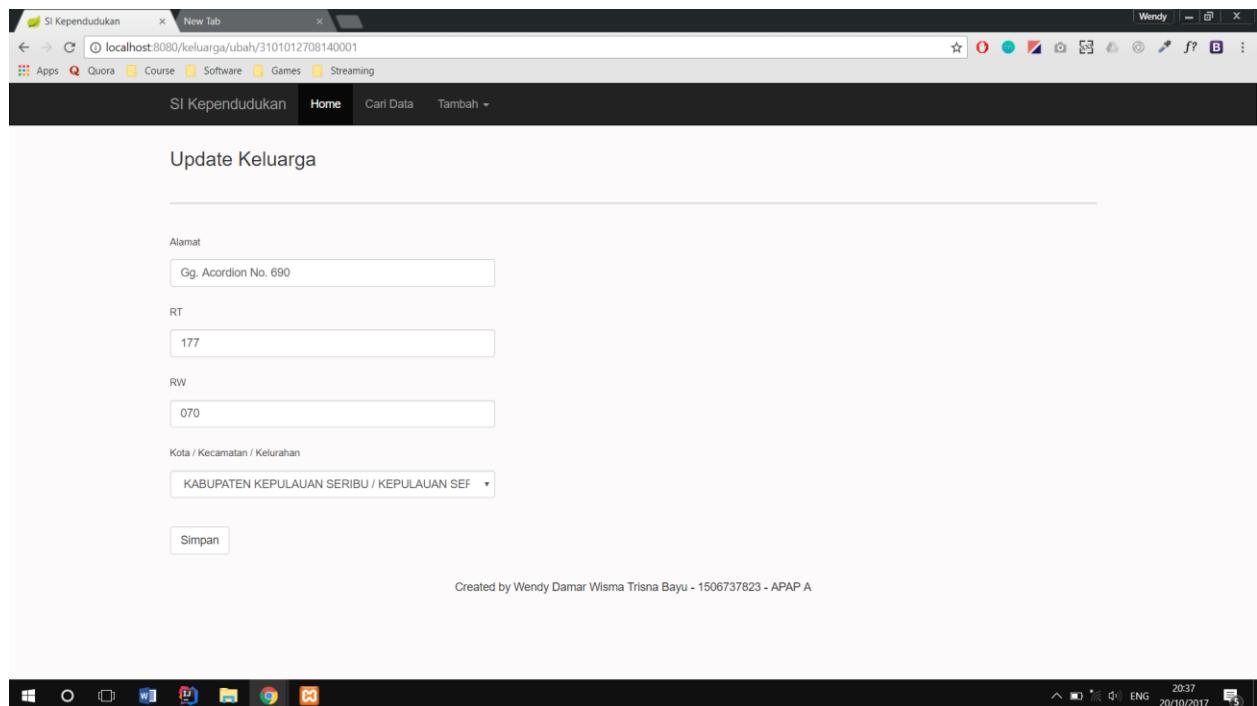
## 6. Mengubah Data Keluarga

**Initial Page:** /keluarga/ubah/{NKK}

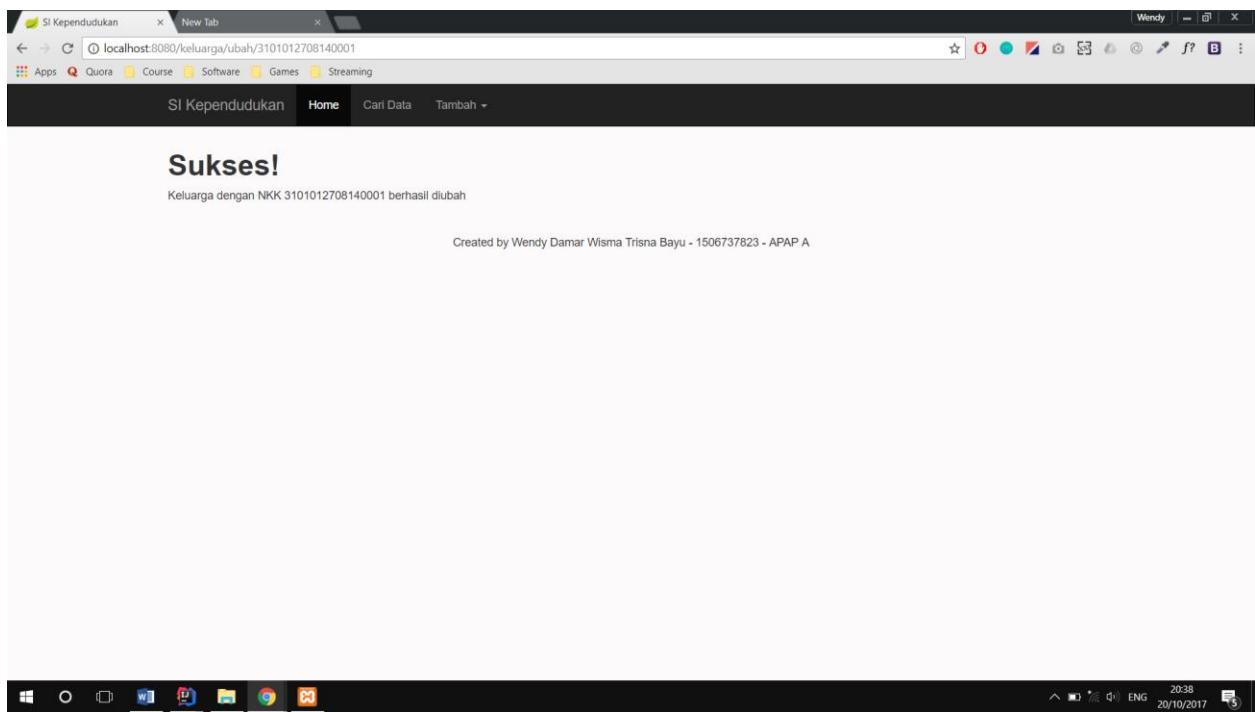
**Form Request:** POST, /keluarga/ubah/{NKK}

**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.

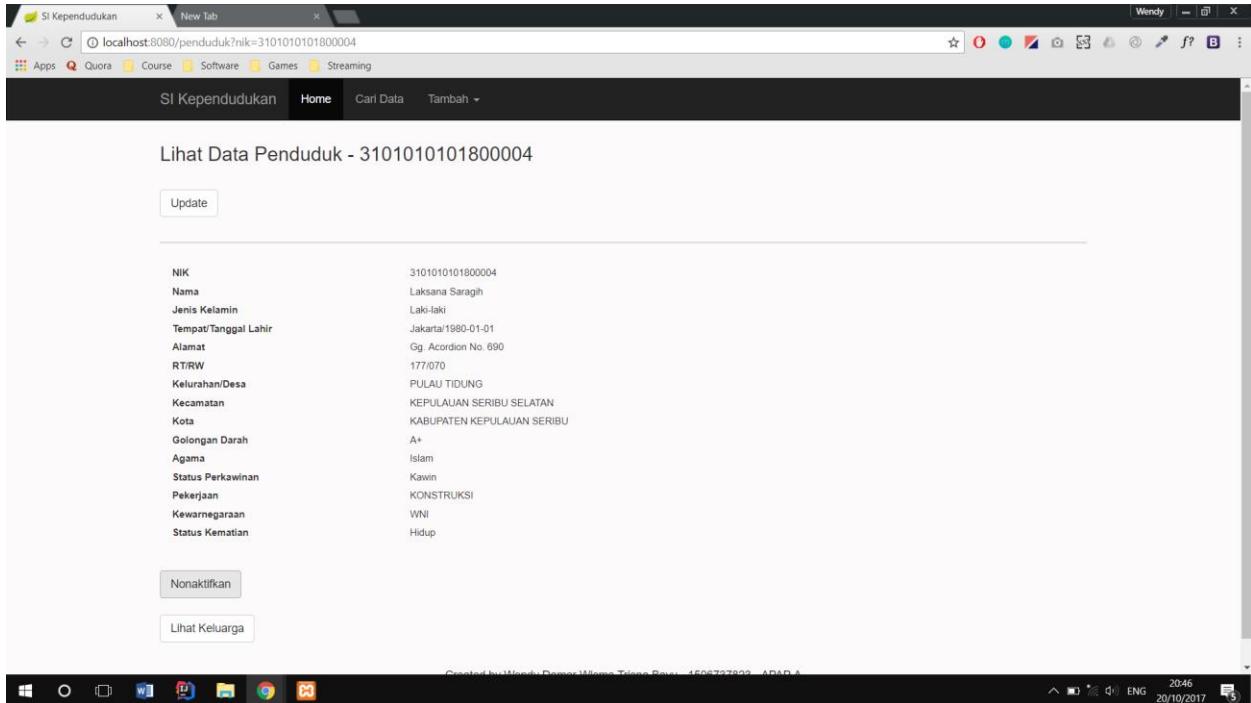


## 7. Mengubah Status Kematian Penduduk

**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

The screenshot shows a web browser window titled "SI Kependudukan". The URL in the address bar is "localhost:8080/penduduk/mati". The page displays a success message: "Sukses!" followed by the text "Penduduk dengan NIK 3101010101800004 sudah tidak aktif". Below this, it says "Created by Wendy Damar Wisma Trisna Bayu - 1506737823 - APAP A". The browser's toolbar and taskbar are visible at the top and bottom respectively.

**Sukses!**  
Penduduk dengan NIK 3101010101800004 sudah tidak aktif  
Created by Wendy Damar Wisma Trisna Bayu - 1506737823 - APAP A

The screenshot shows a web browser window titled "SI Kependudukan". The URL in the address bar is "localhost:8080/penduduk?nik=3101010101800004". The page displays a header "Lihat Data Penduduk - 3101010101800004" and a "Update" button. Below this, there is a table of user information:

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

Below the table are two buttons: "Nonaktifkan" and "Lihat Keluarga". The browser's toolbar and taskbar are visible at the top and bottom respectively.

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

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

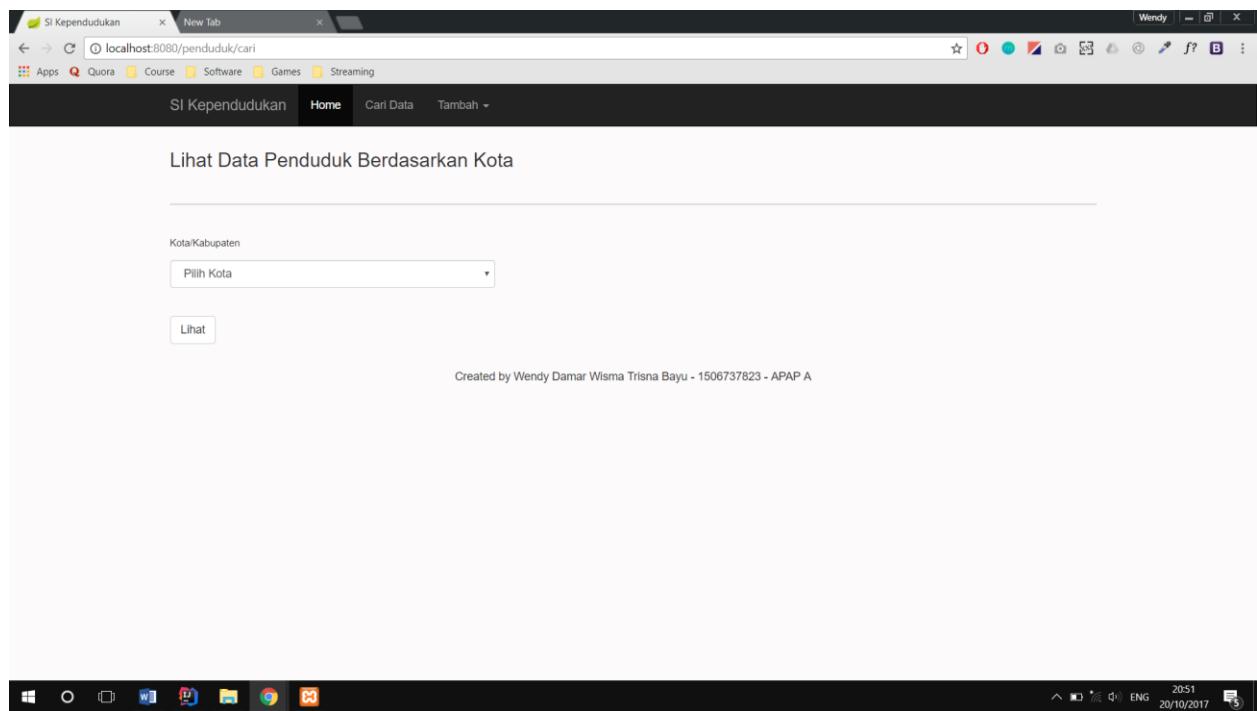
**Initial Page:** /penduduk/cari

**Request Method:** GET,

/penduduk/cari?kt={ID\_KOTA}&kc={ID\_KECAMATAN}&kl={ID\_KELURAHAN}

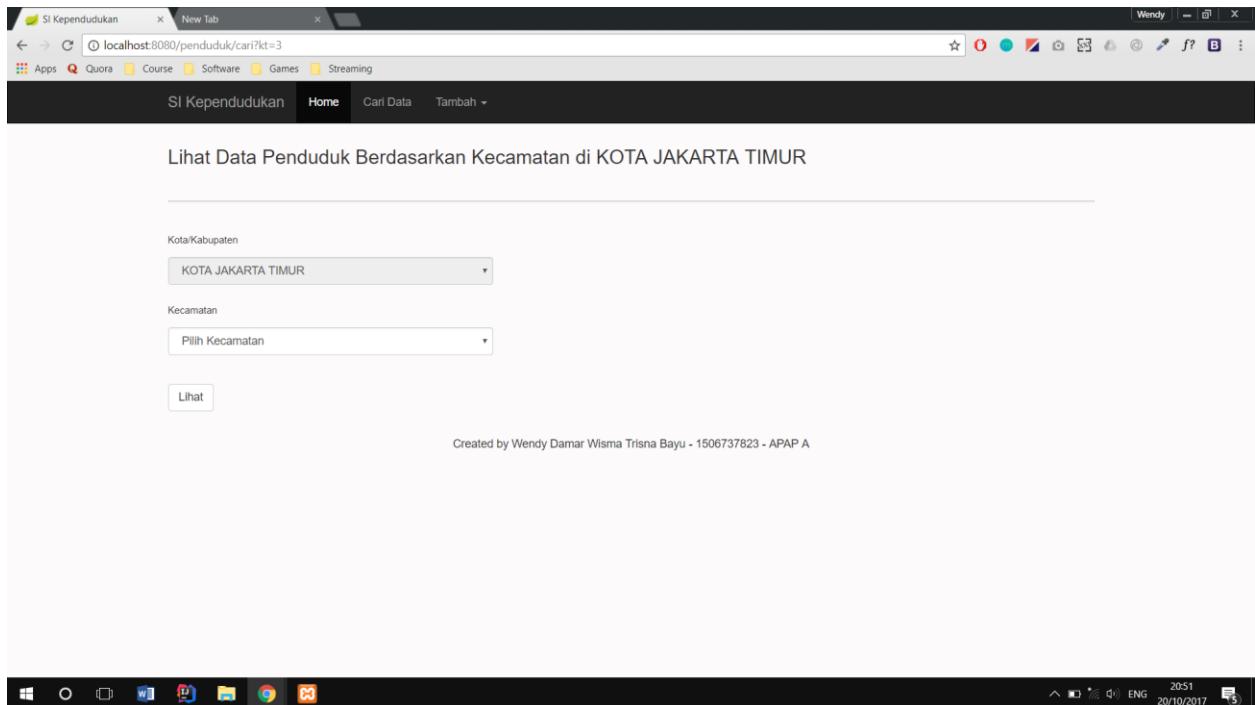
**Response Page:** /penduduk/cari?kt={ID\_KOTA}&kc={ID\_KECAMATAN}&kl={ID\_KELURAHAN}

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.

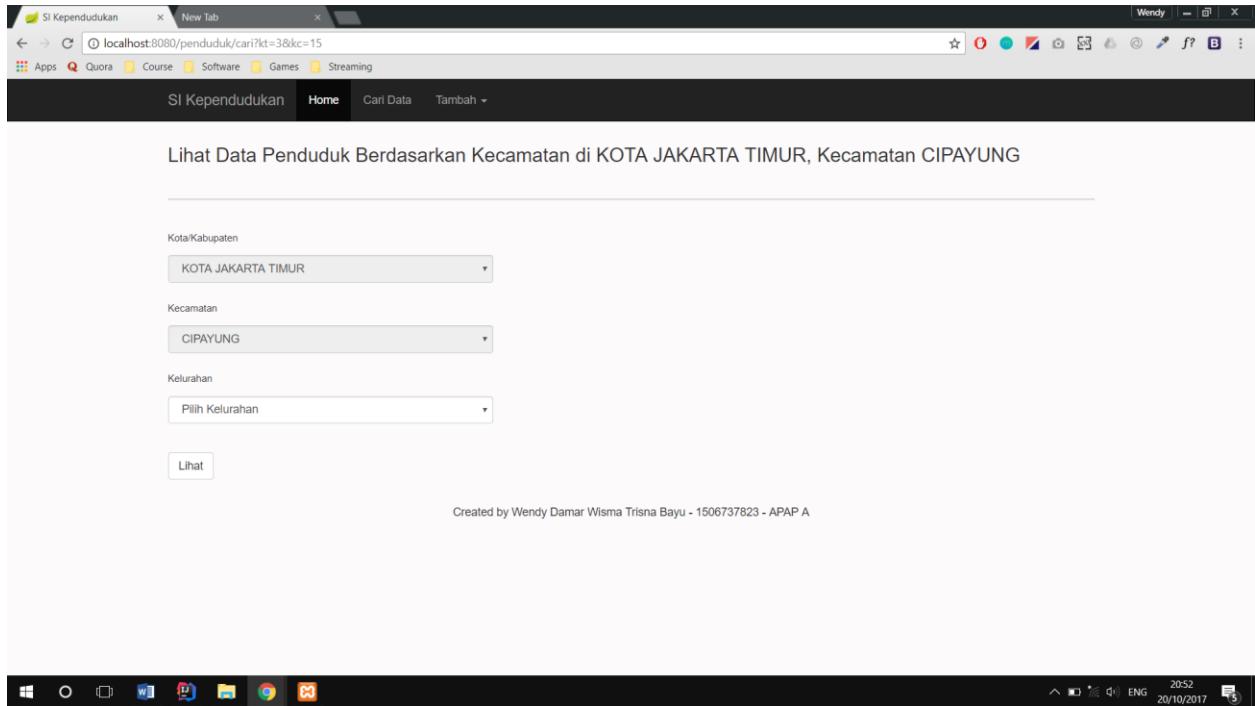


Jika sudah dipilih maka program akan mengirimkan parameter kota dan masuk ke Controller “penduduk/cari” lagi namun dengan kondisi sudah memiliki kota. Pada saat itu kita akan validasi apakah

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	Nama	Tanggal Lahir
3172032009170003	Purwa Irawan	2017-09-20

The "Penduduk Dengan Usia Tertua" table shows:

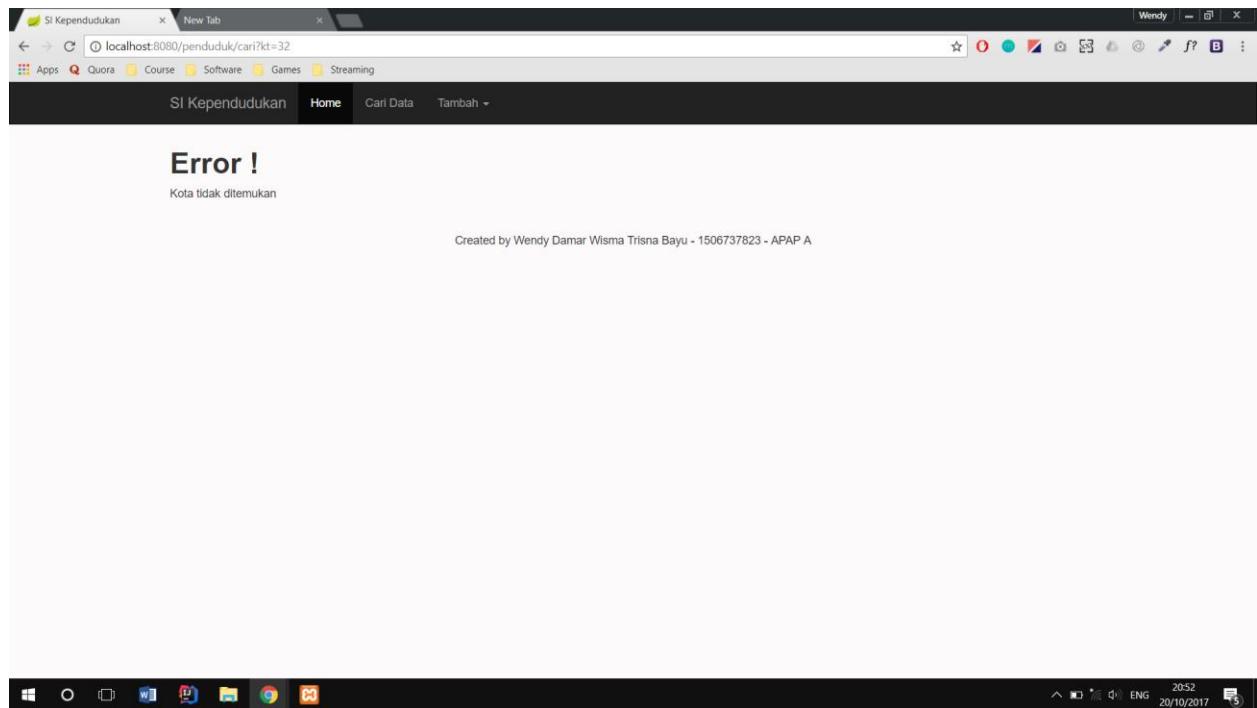
NIK	Nama	Tanggal Lahir
3172032809620001	Kenari Saputra	1952-09-28

Below these tables is a data grid with columns: NIK, Nama, and Jenis Kelamin. The data in the grid is as follows:

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**

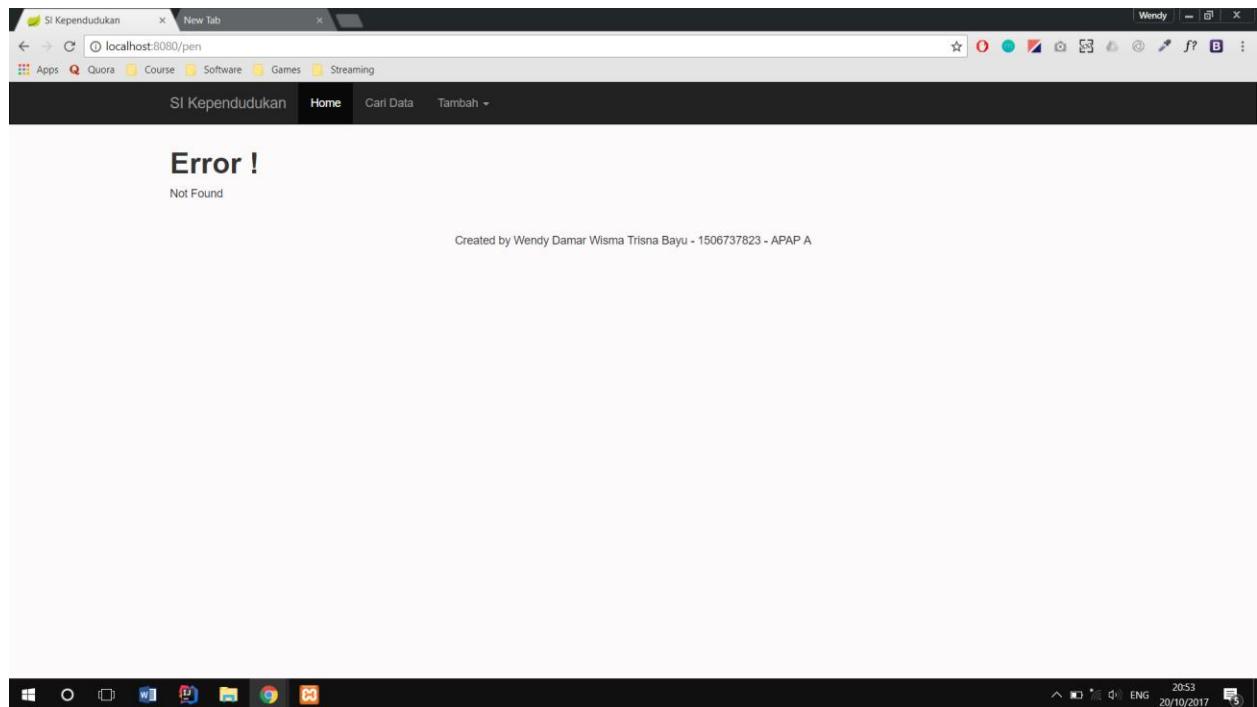
NIK	3172032809520001
Nama	Kenari Saputra
Tanggal Lahir	1952-09-28

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.

The screenshot shows a web browser window titled "SI Kependudukan". The address bar displays "localhost:8080/penduduk/cari?kt=3&kc=15&kd=85". The page content includes a heading "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". Both tables have columns for NIK, Name, and Date of Birth. A search bar is present at the bottom of the page, along with a table of all population data.

Penduduk Dengan Usia Termuda		
NIK	Nama	Tanggal Lahir
3172032009170003	Purwa Irawan	2017-09-20

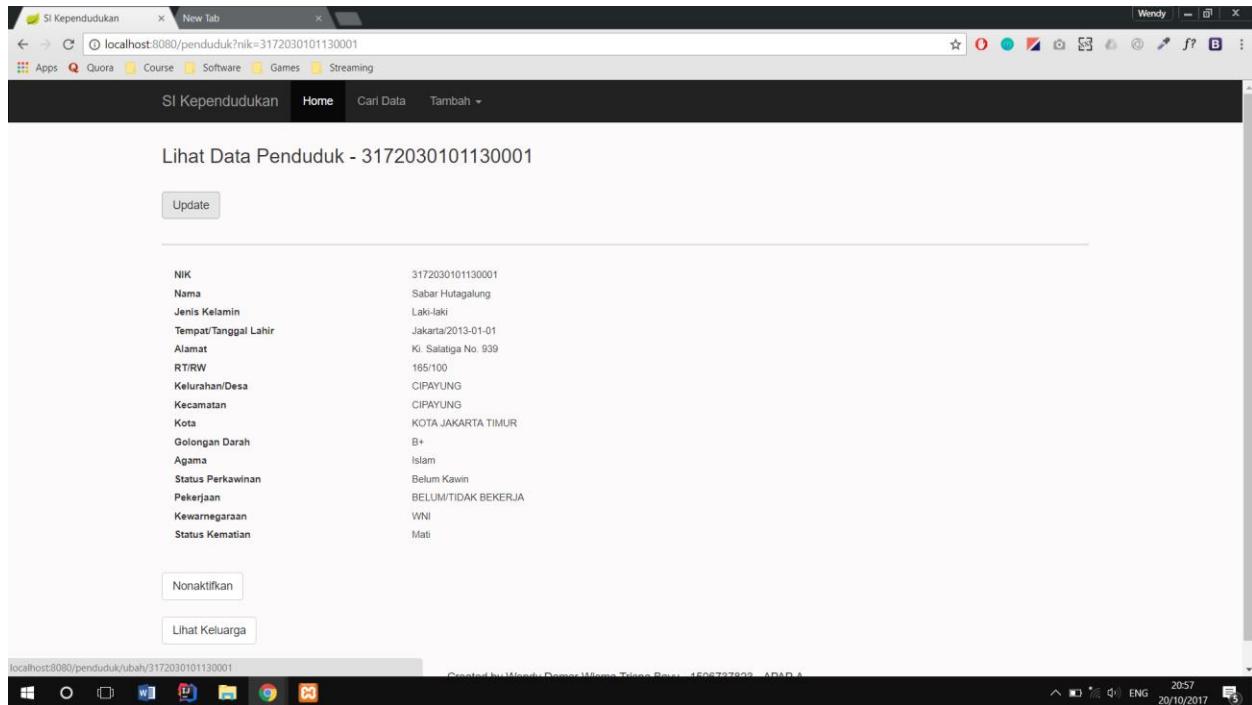
Penduduk Dengan Usia Tertua		
NIK	Nama	Tanggal Lahir
3172032809520001	Kenari Saputra	1952-09-28

Show	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

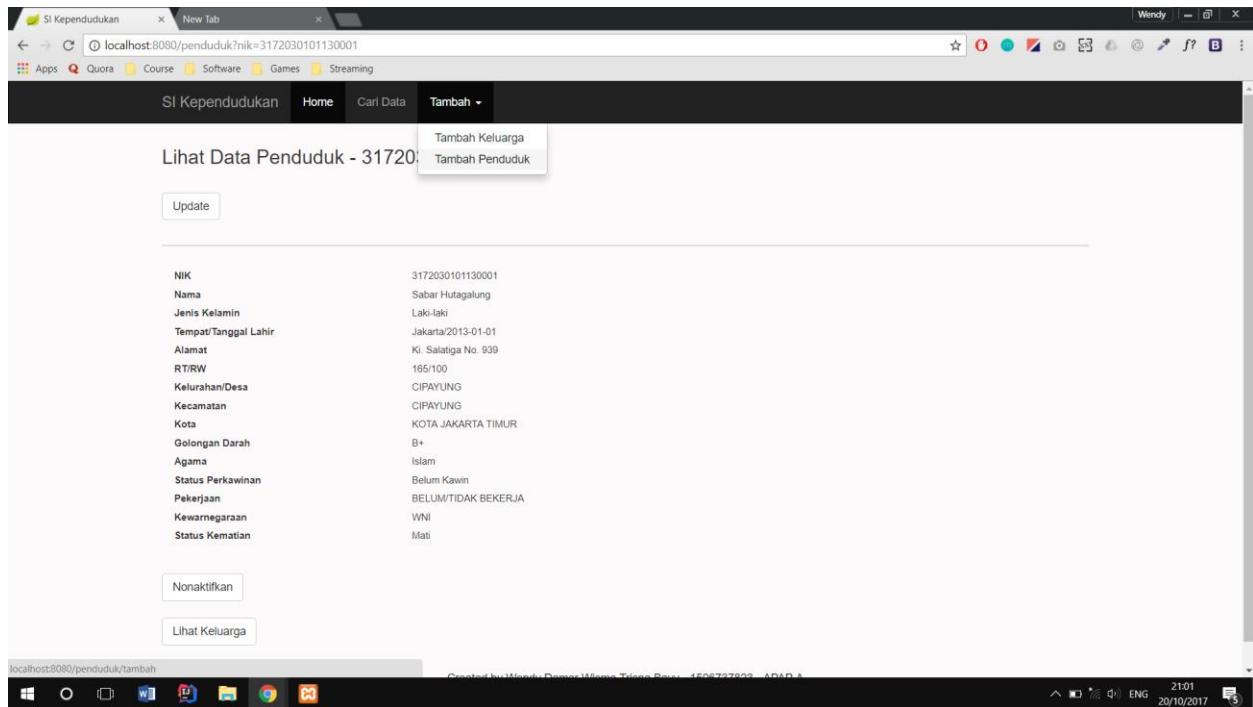
### c. Membuat HyperLink Update

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

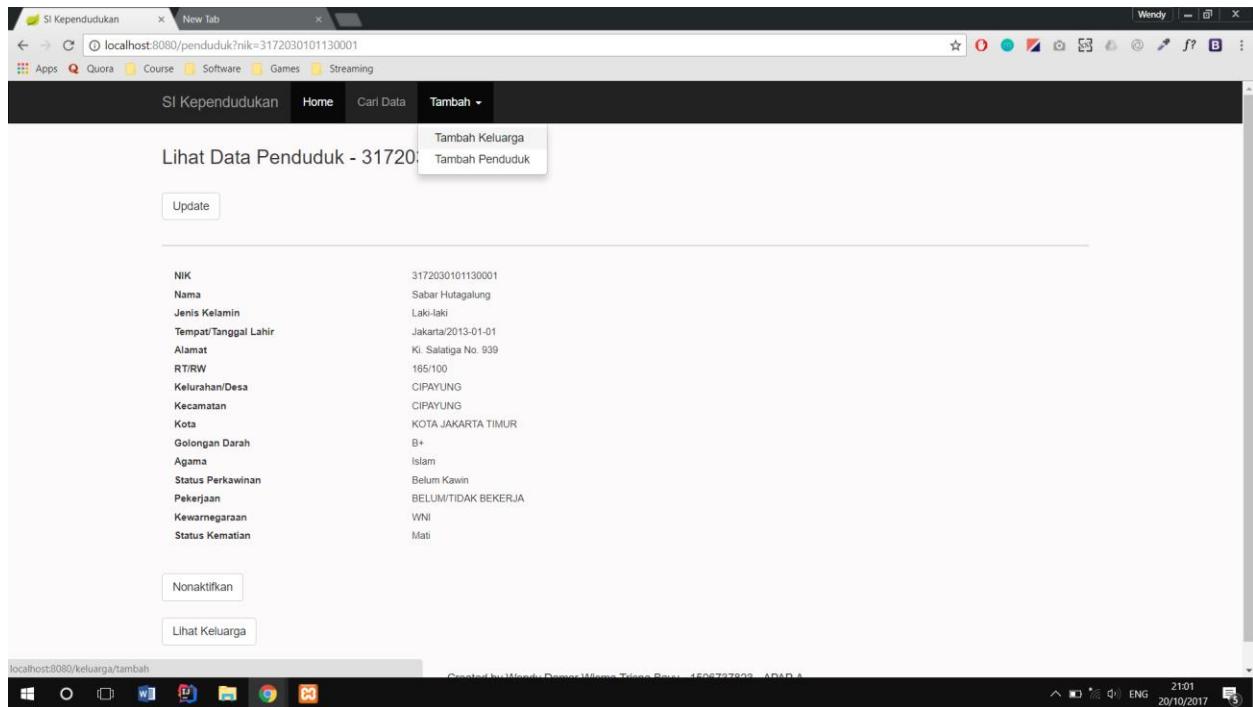
Fitur ini ialah menambahkan link menuju halaman tambah penduduk pada navigation bar. Fitur ini menggunakan href seperti biasa menuju /penduduk/tambah



### 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.

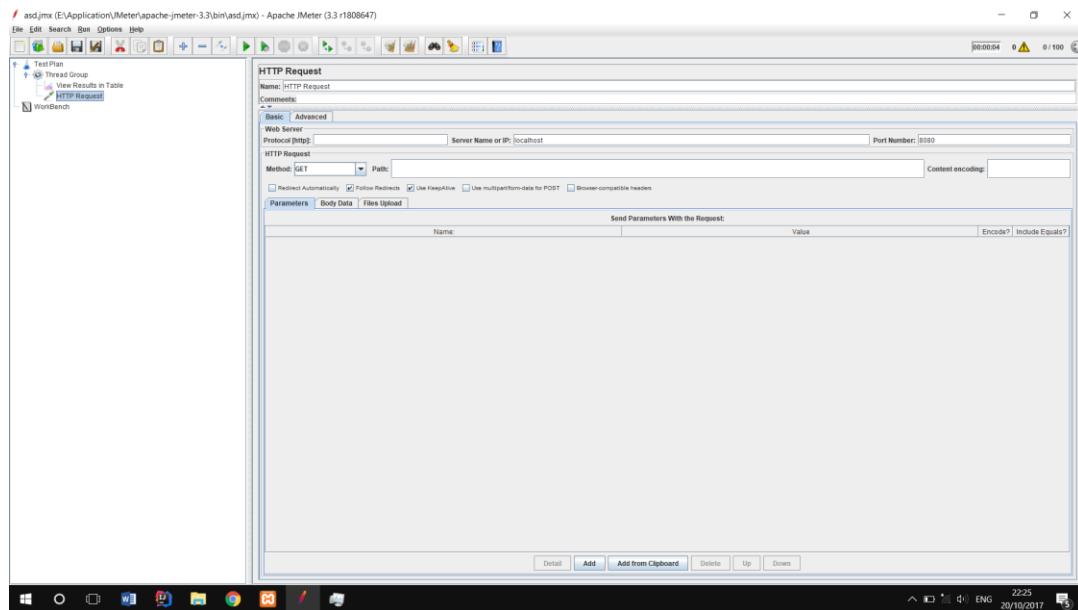
# Stress Testing

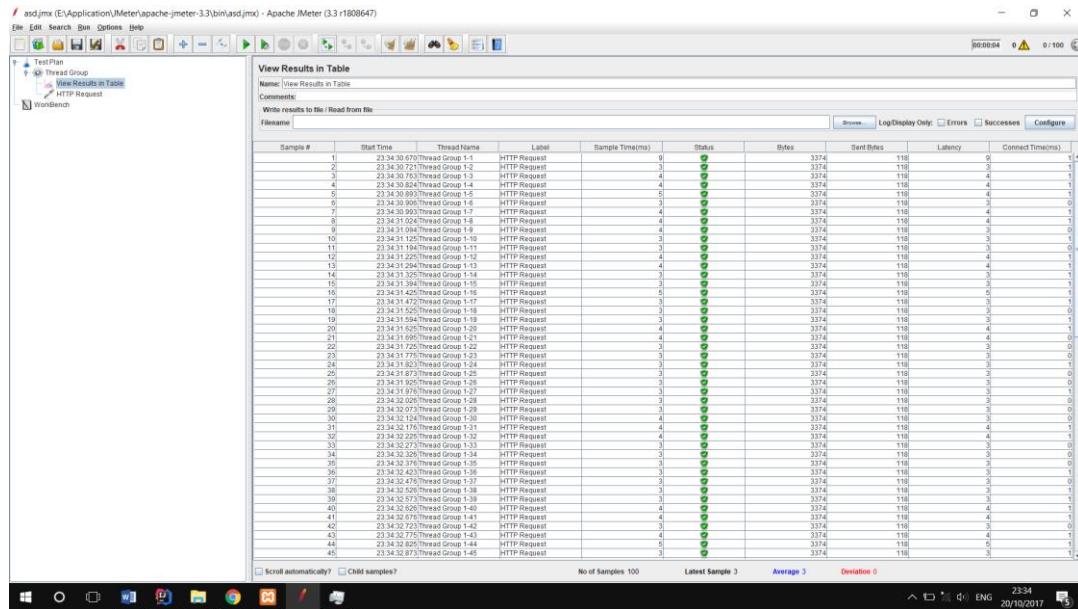
**Ramp Up:** 5 s

**Thread:** 100

1. Akses halaman utama

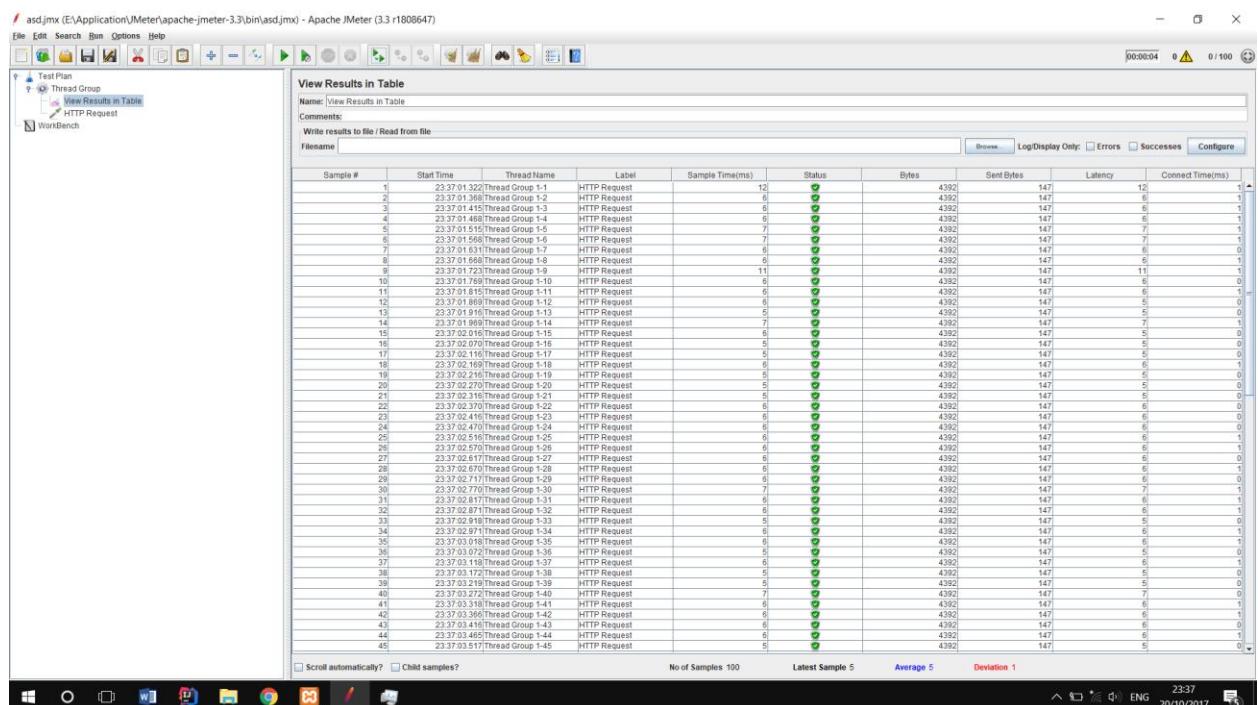
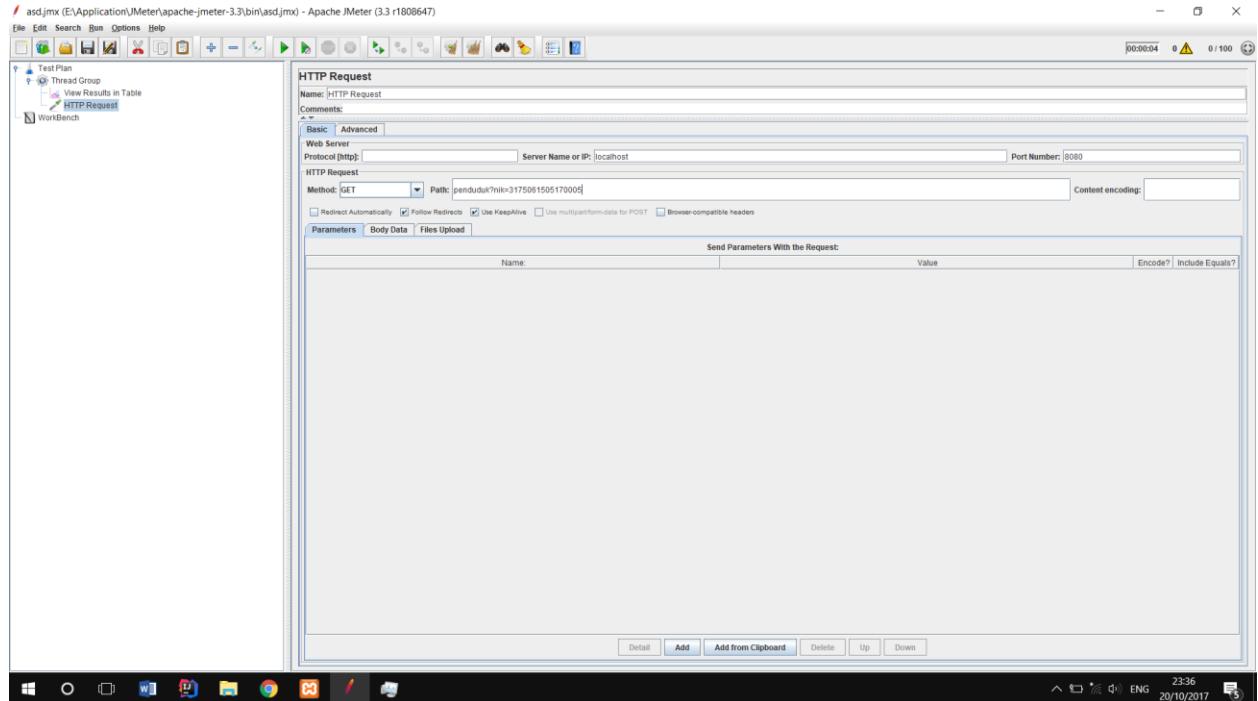
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html dan css pada web.





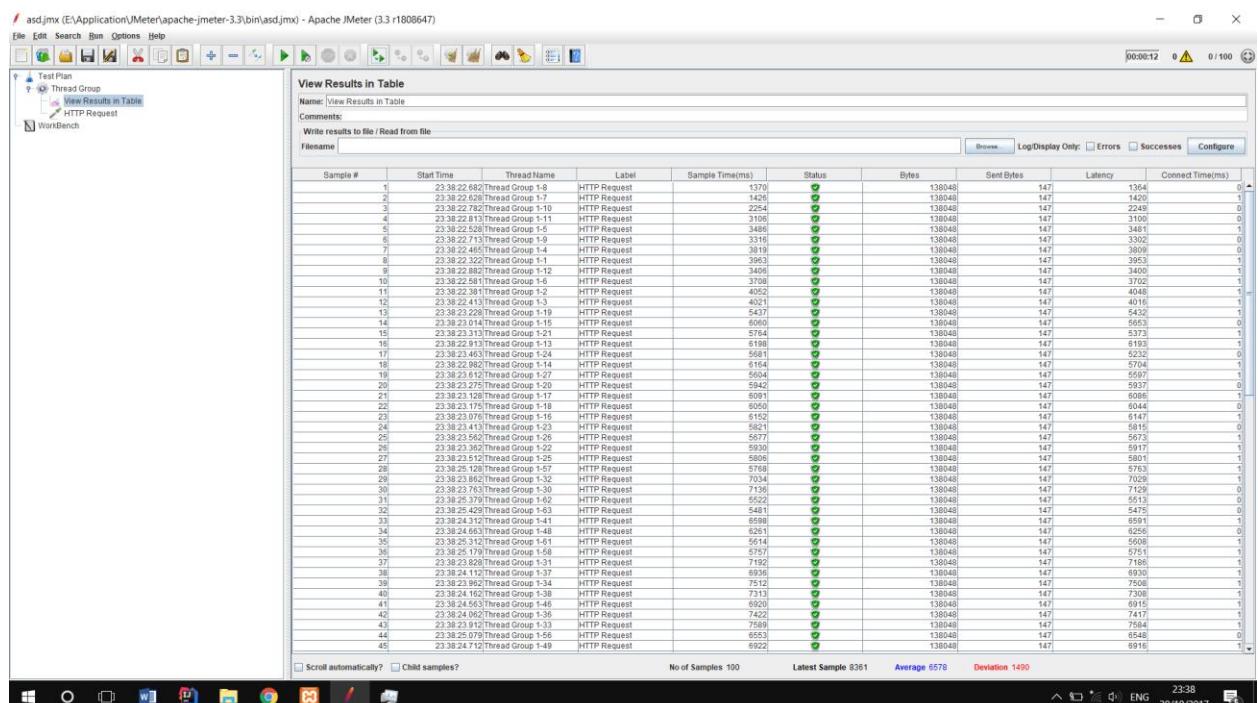
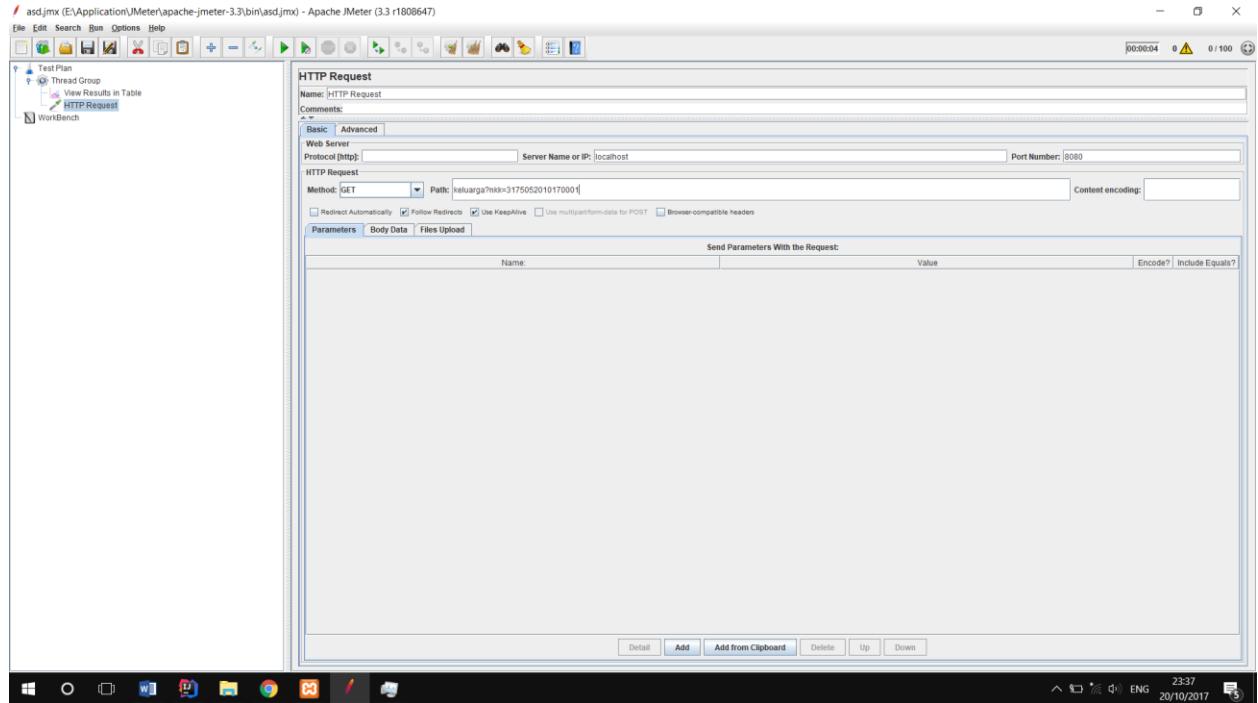
## 2. Akses lihat penduduk

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



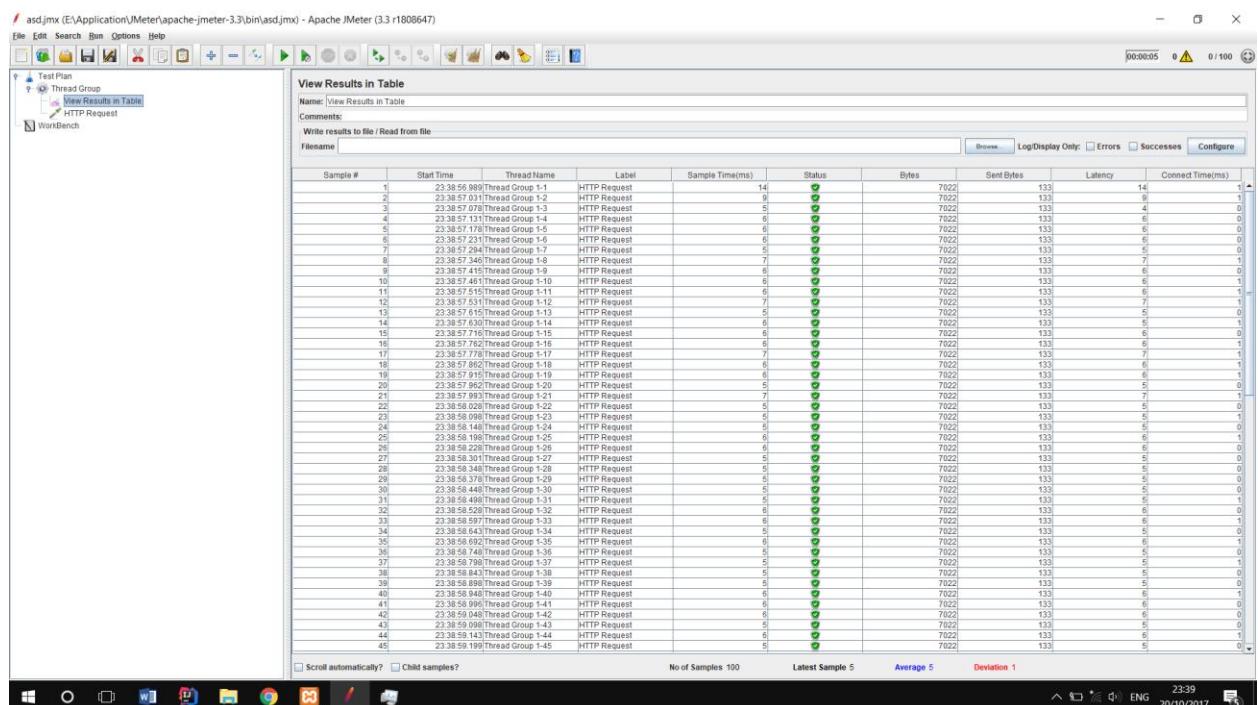
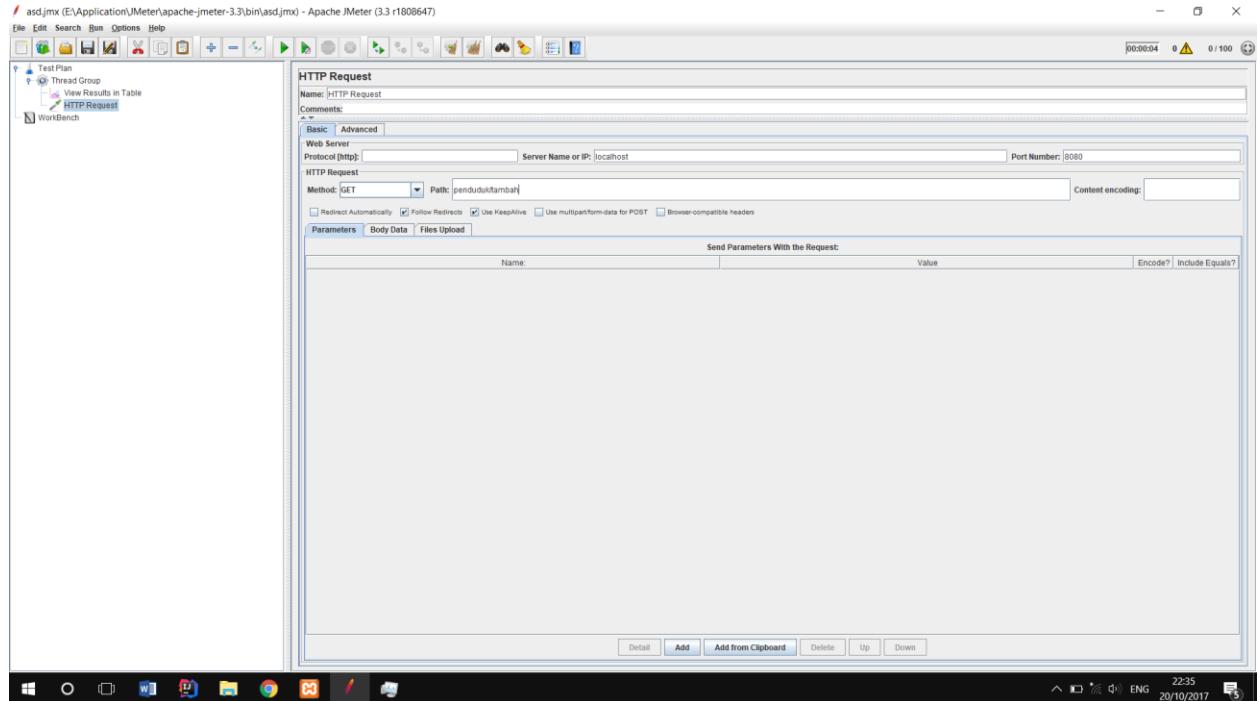
### 3. Akses lihat keluarga

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



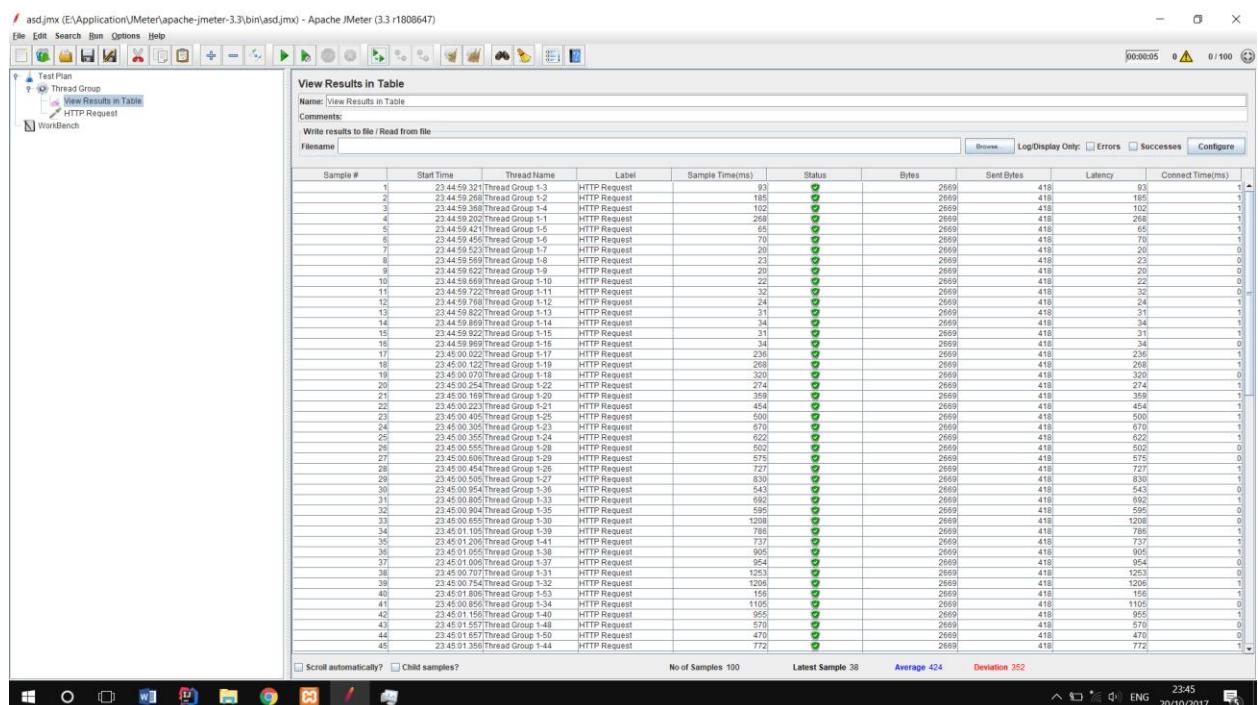
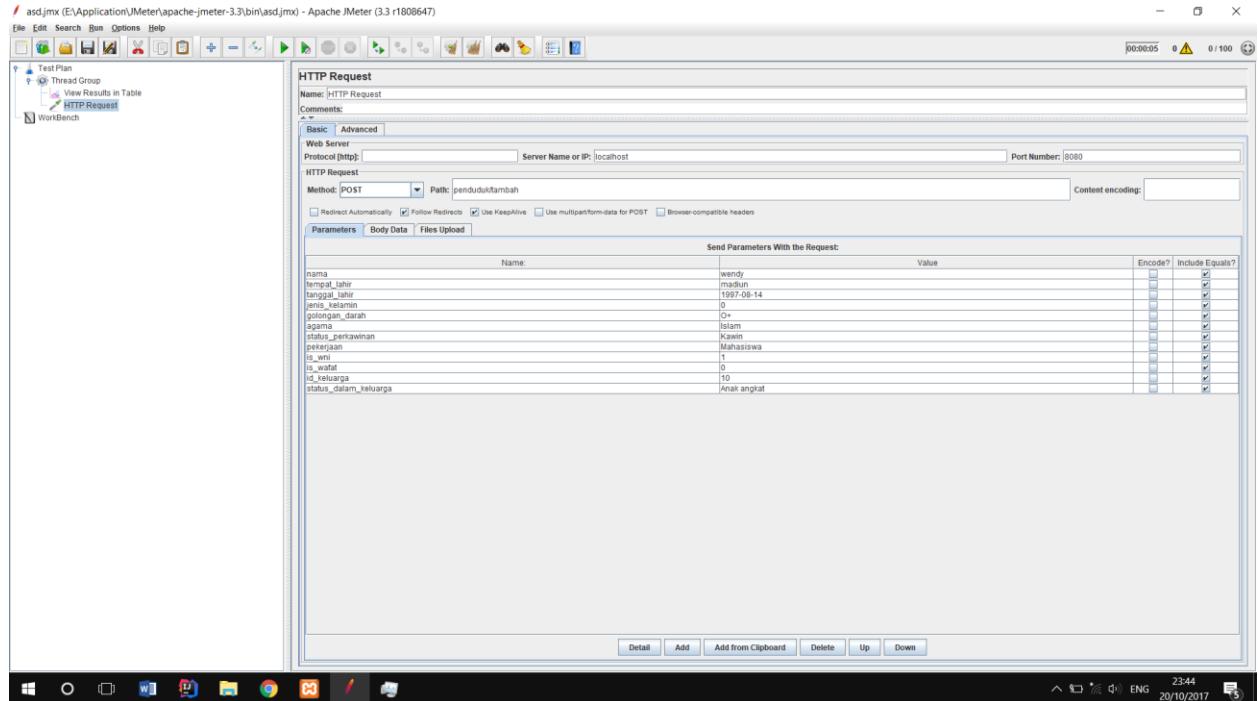
#### 4. Akses tambah penduduk

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



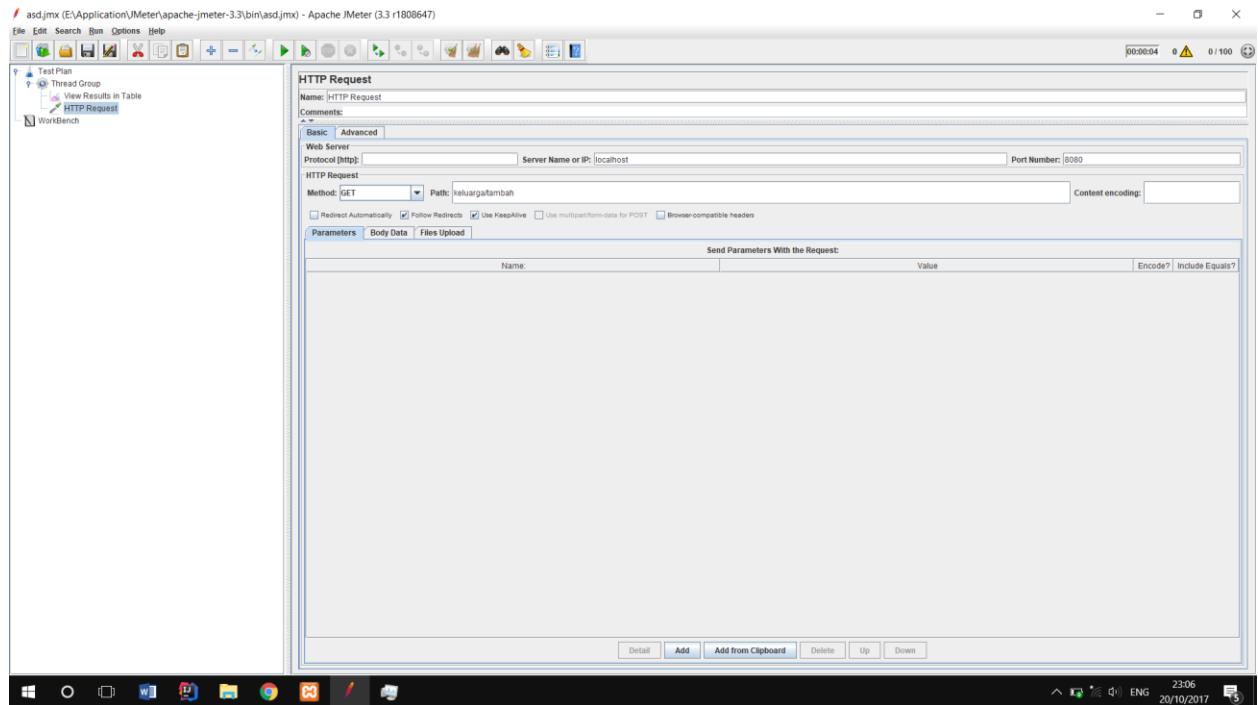
## 5. Akses tambah penduduk submit

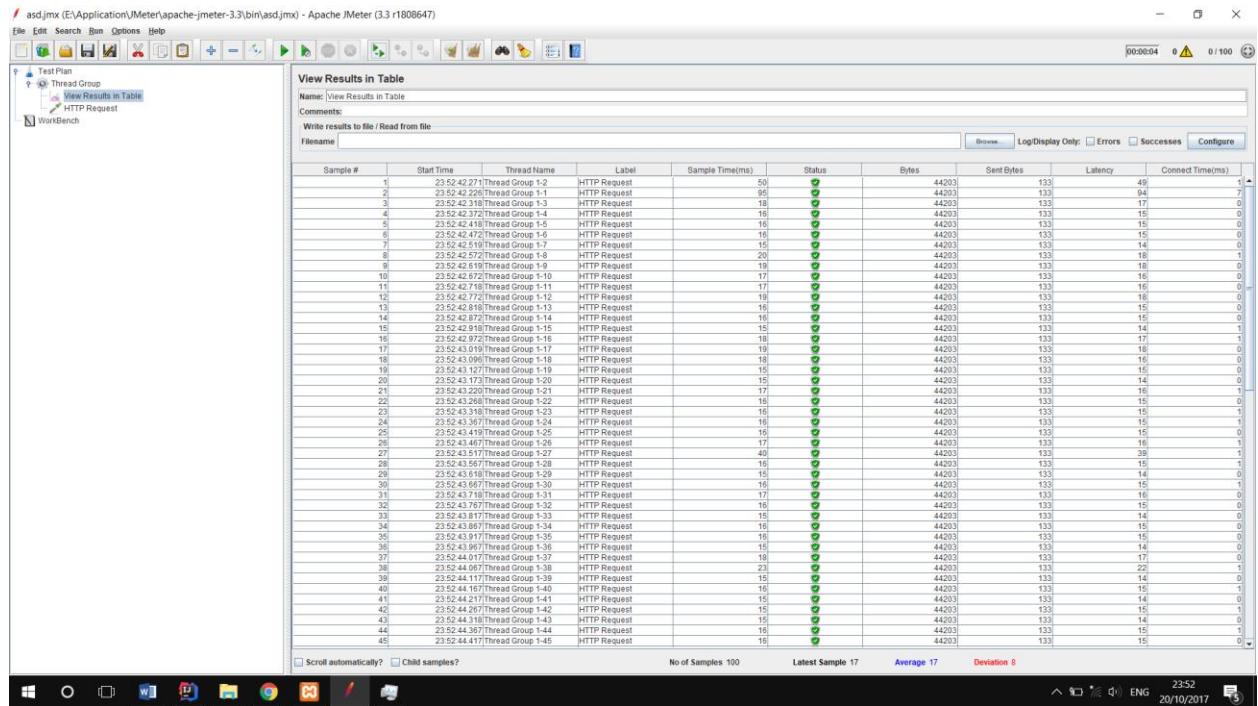
Halaman dapat berjalan dengan optimal hal ini dikarenakan minimnya penggunaan query dan penggunaan primary key, serial, dan index.



## 6. Akses tambah keluarga

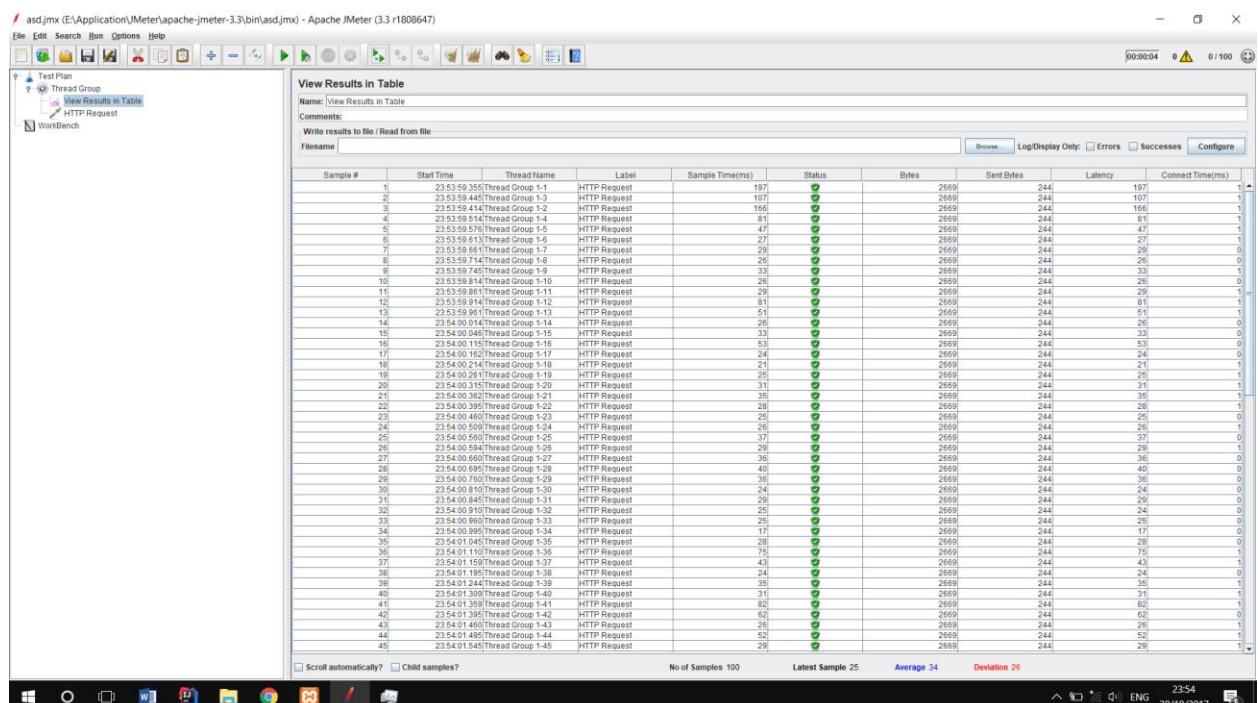
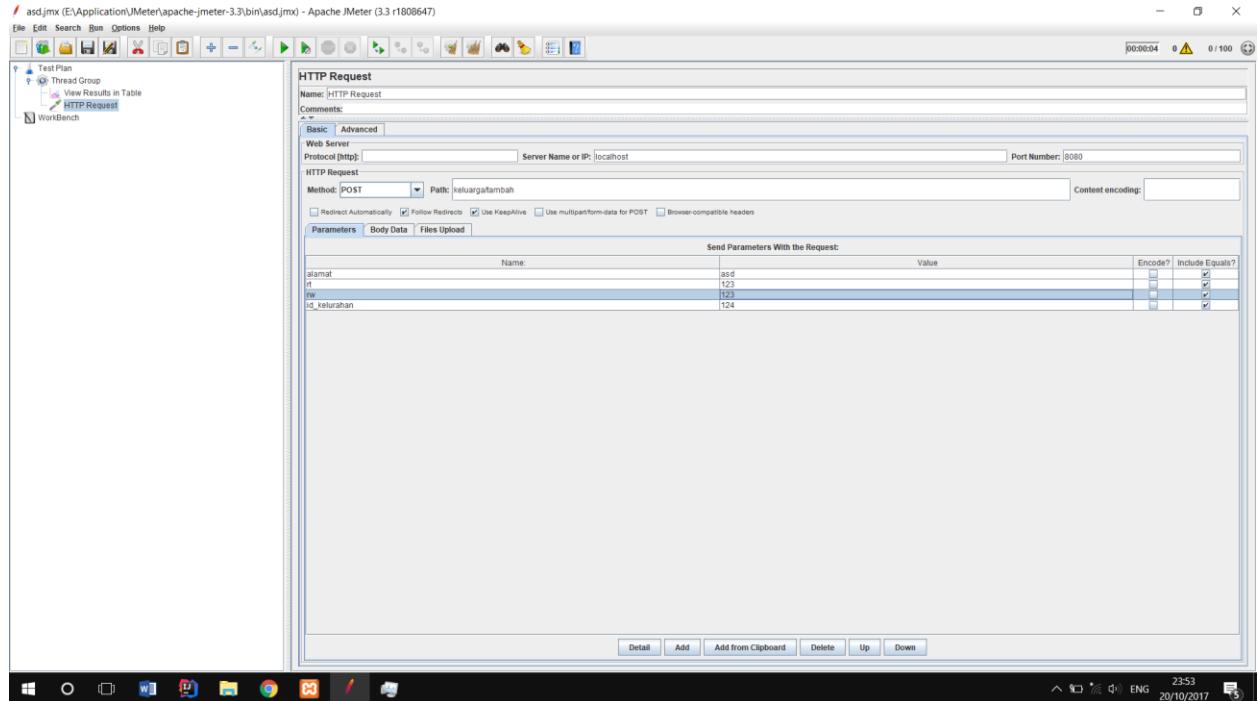
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html, css dan hasil looping mengenai kelurahan kecamatan dan kota yang akan dituju pada web. Hal ini cukup cepat karena penggunaan primary key dan index pada setiap id.





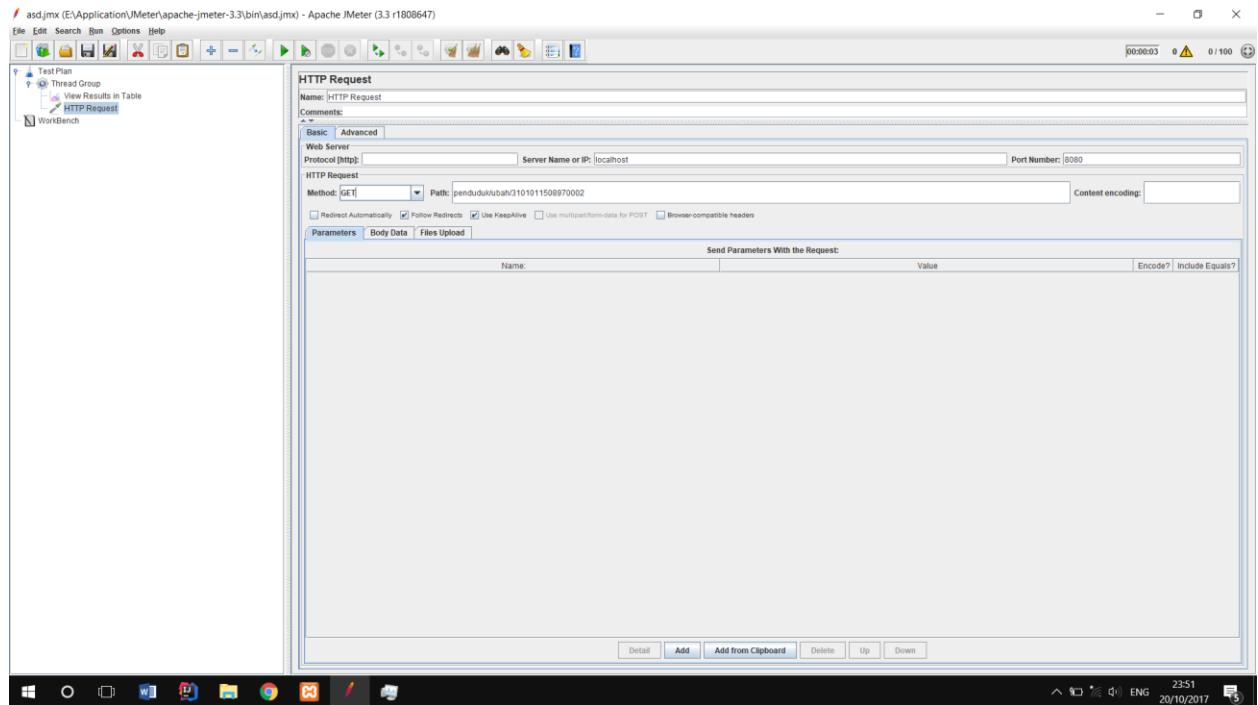
## 7. Akses tambah keluarga submit

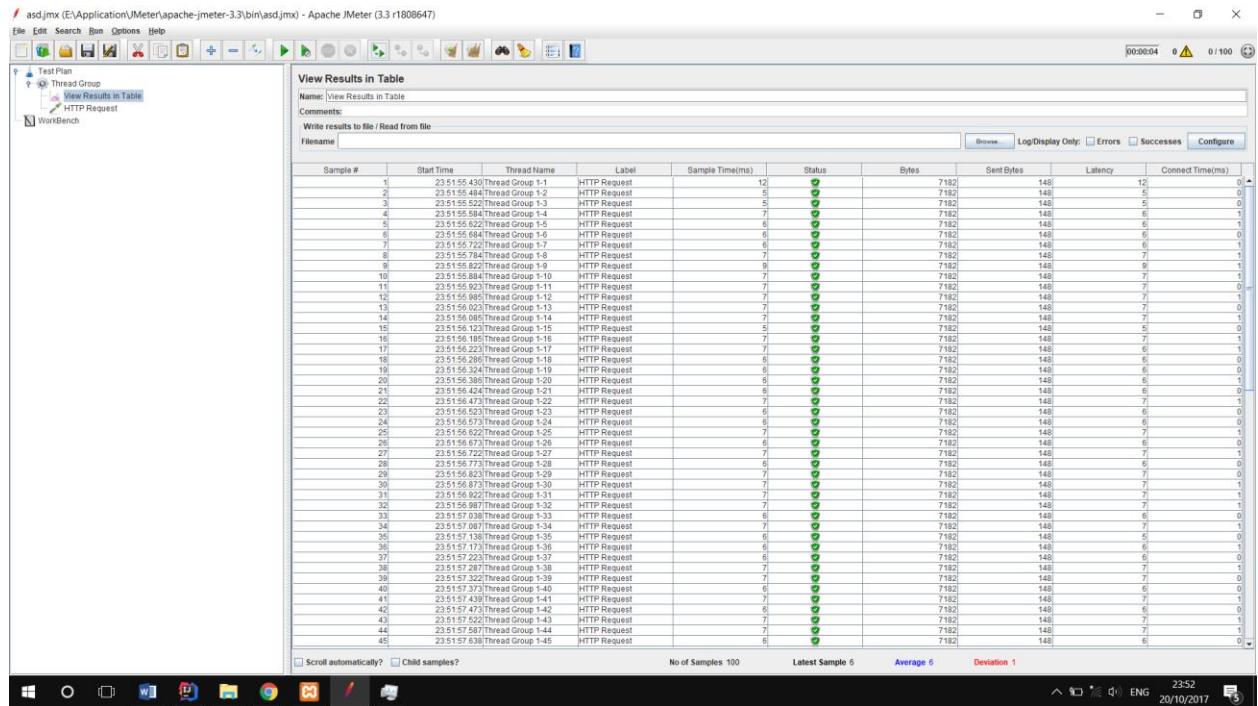
Halaman dapat berjalan dengan optimal hal ini dikarenakan minimnya penggunaan query dan penggunaan primary key, serial, dan index.



## 8. Akses update penduduk

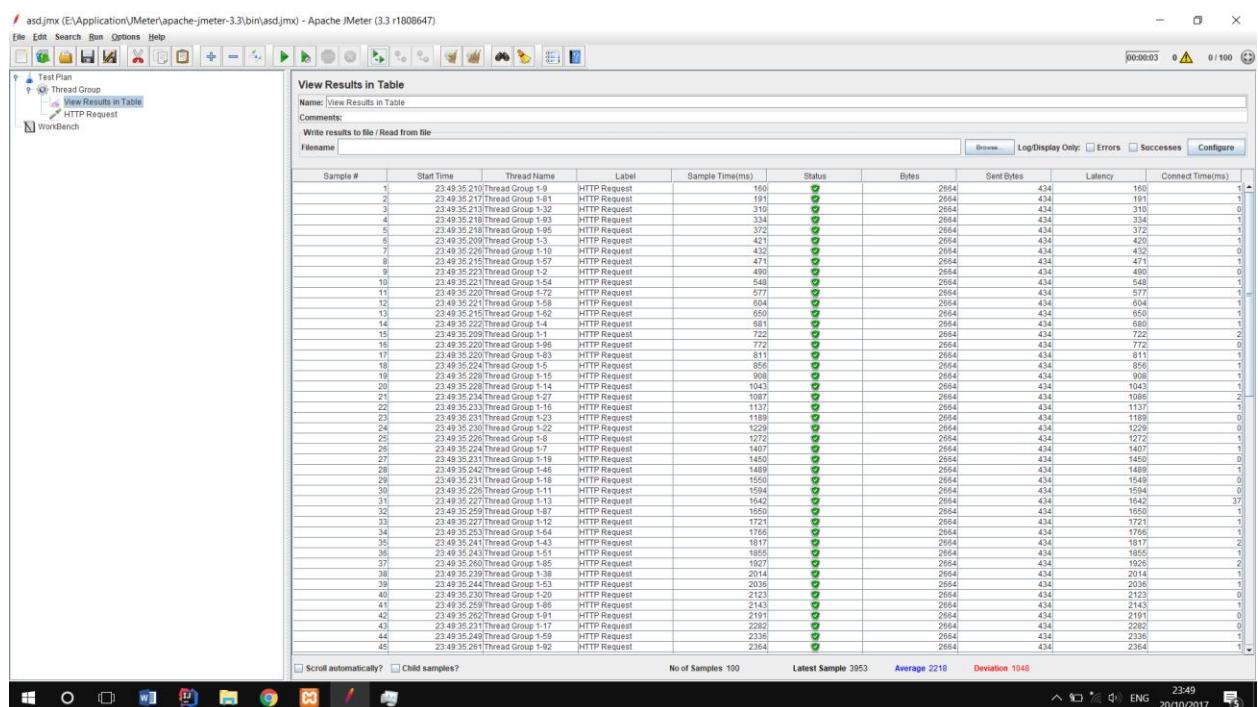
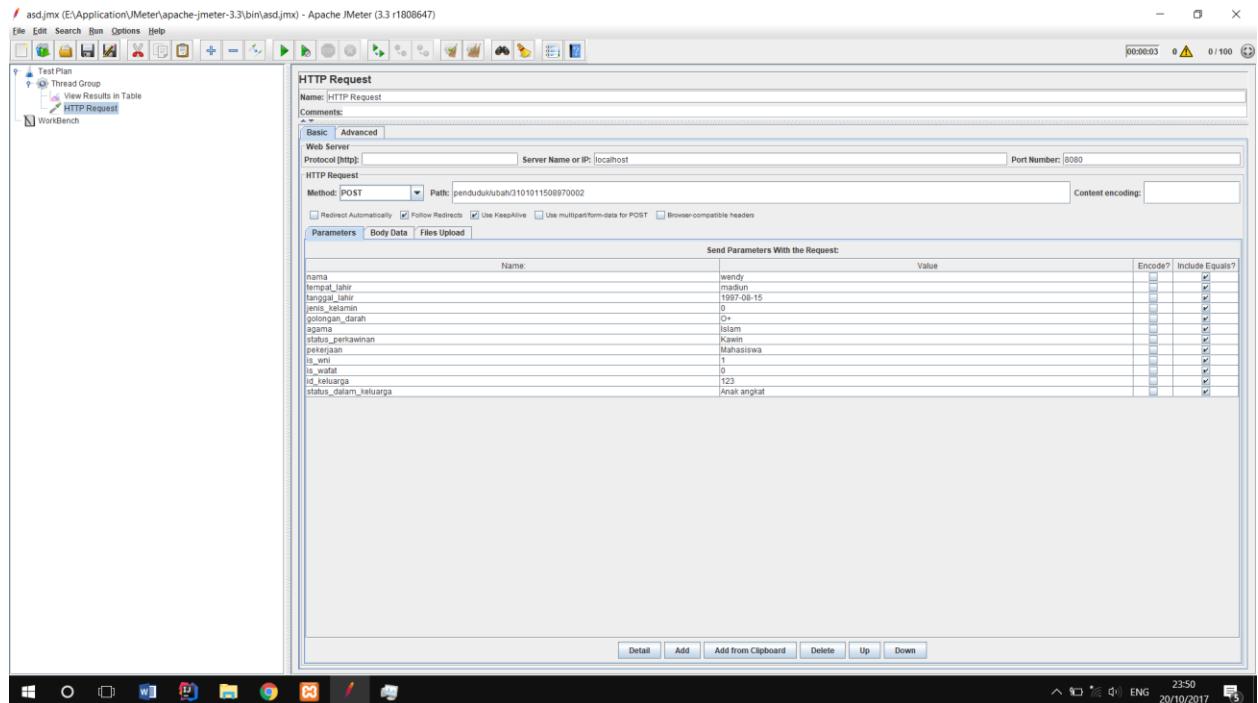
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select penduduk pada web. Dan hal ini didukung dengan penggunaan index pada nik.





## 9. Akses update penduduk submit

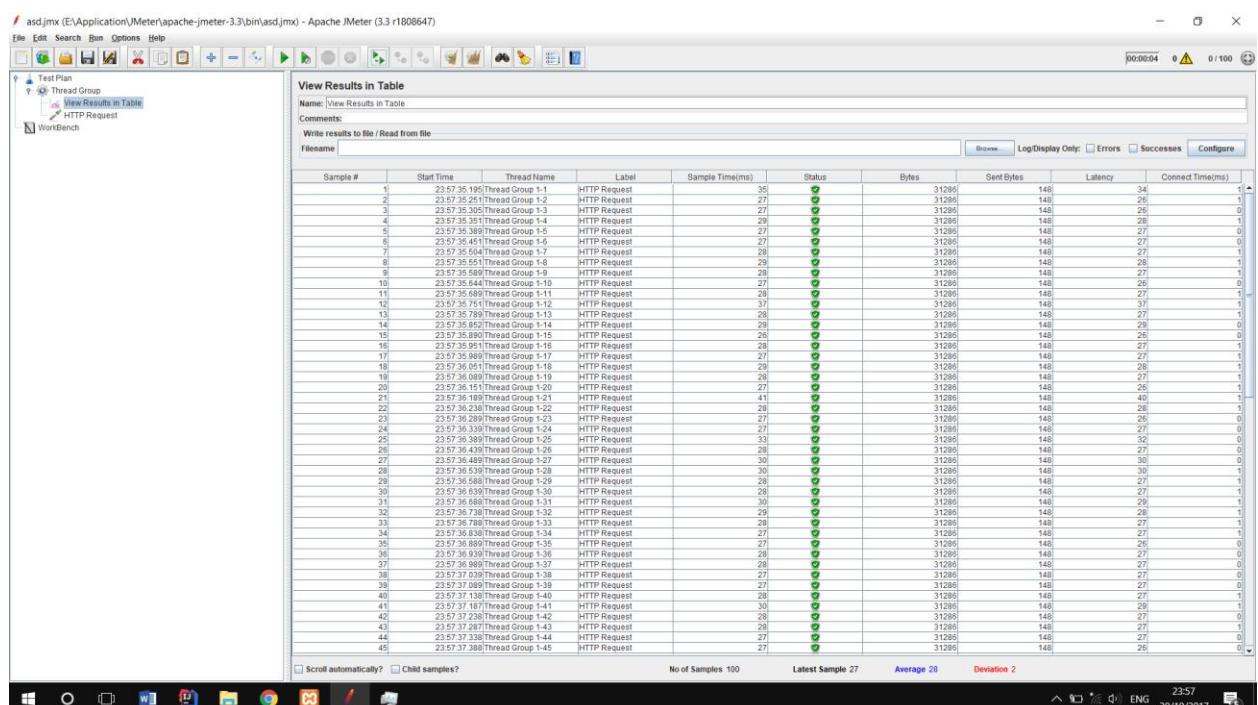
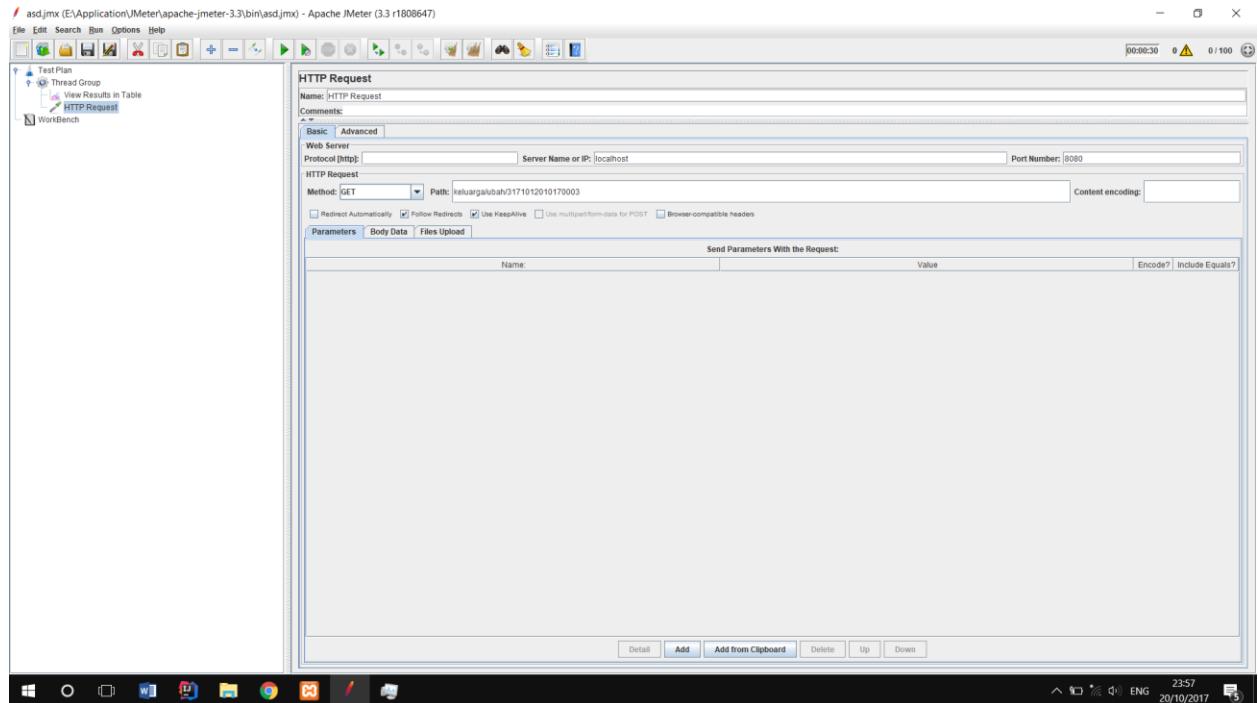
Halaman dapat berjalan dengan optimal hal ini dikarenakan minimnya penggunaan query dan penggunaan primary key dan index.



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

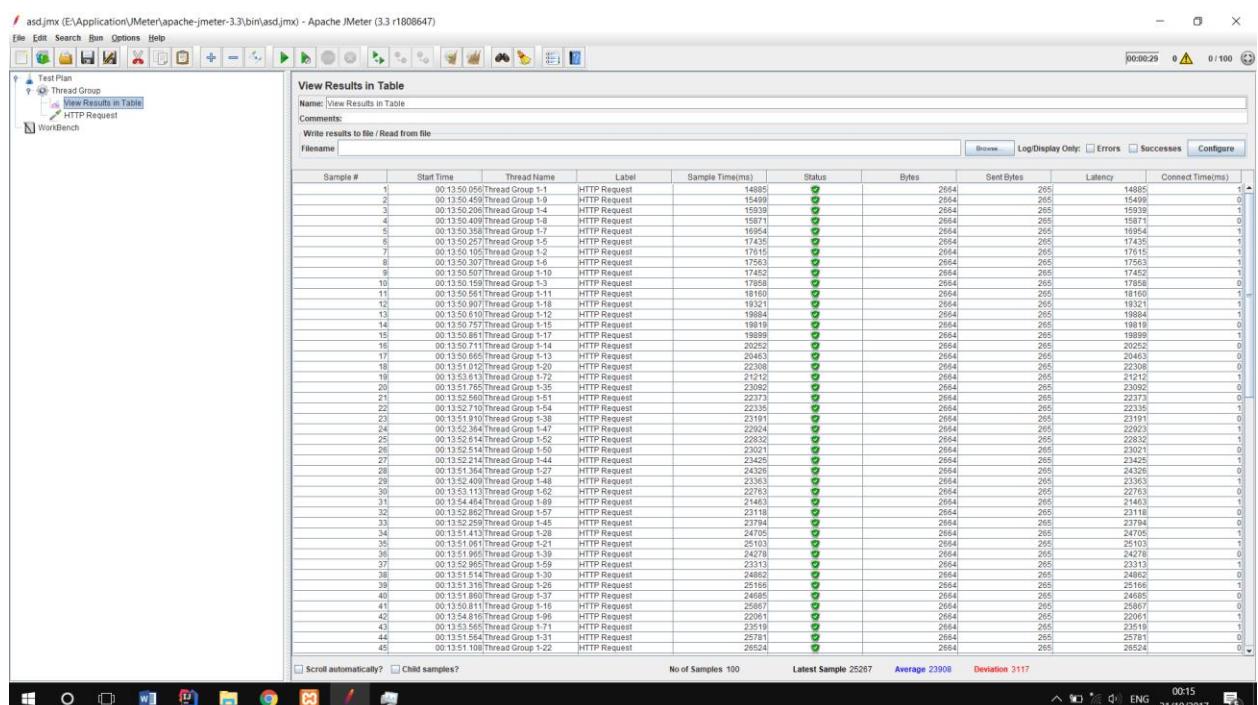
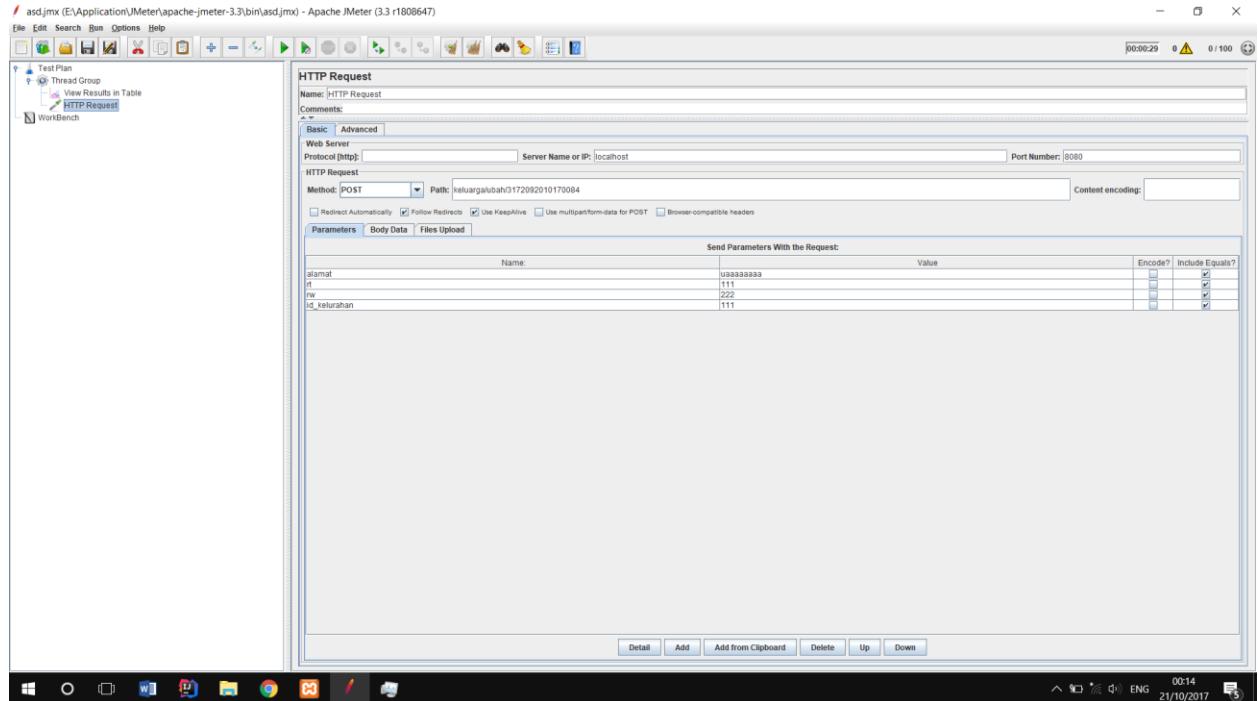
## 10. Akses update keluarga

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.



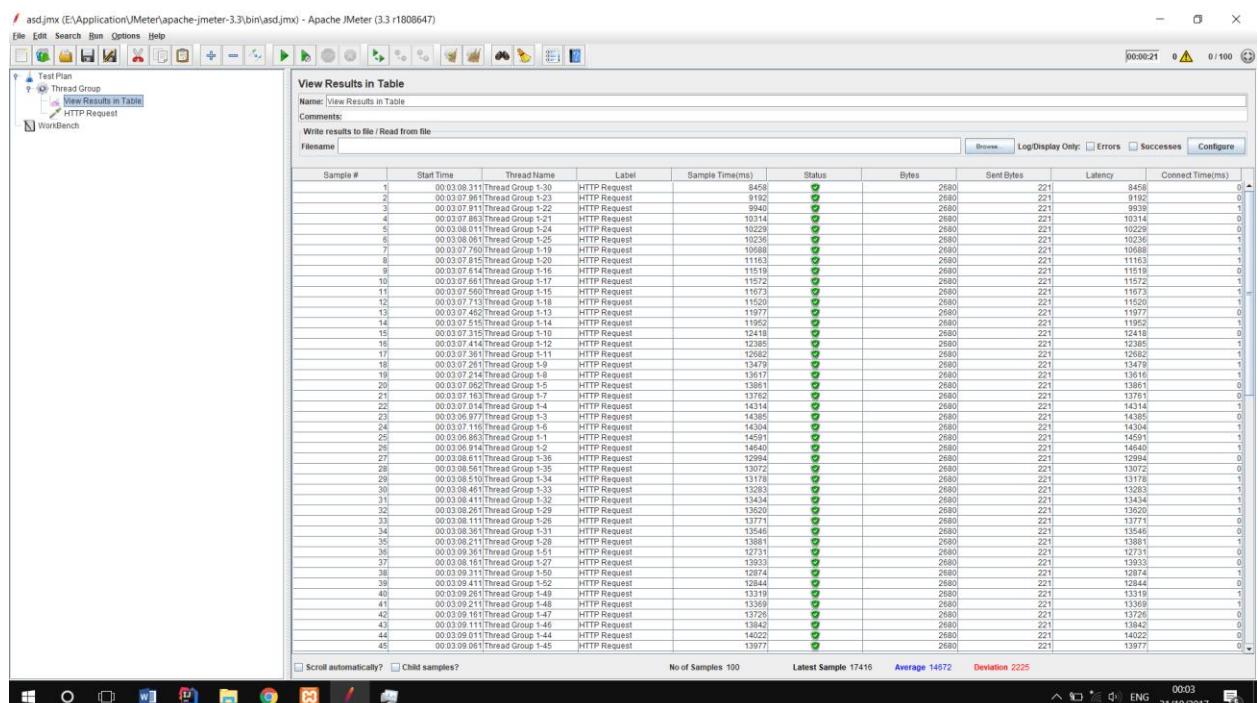
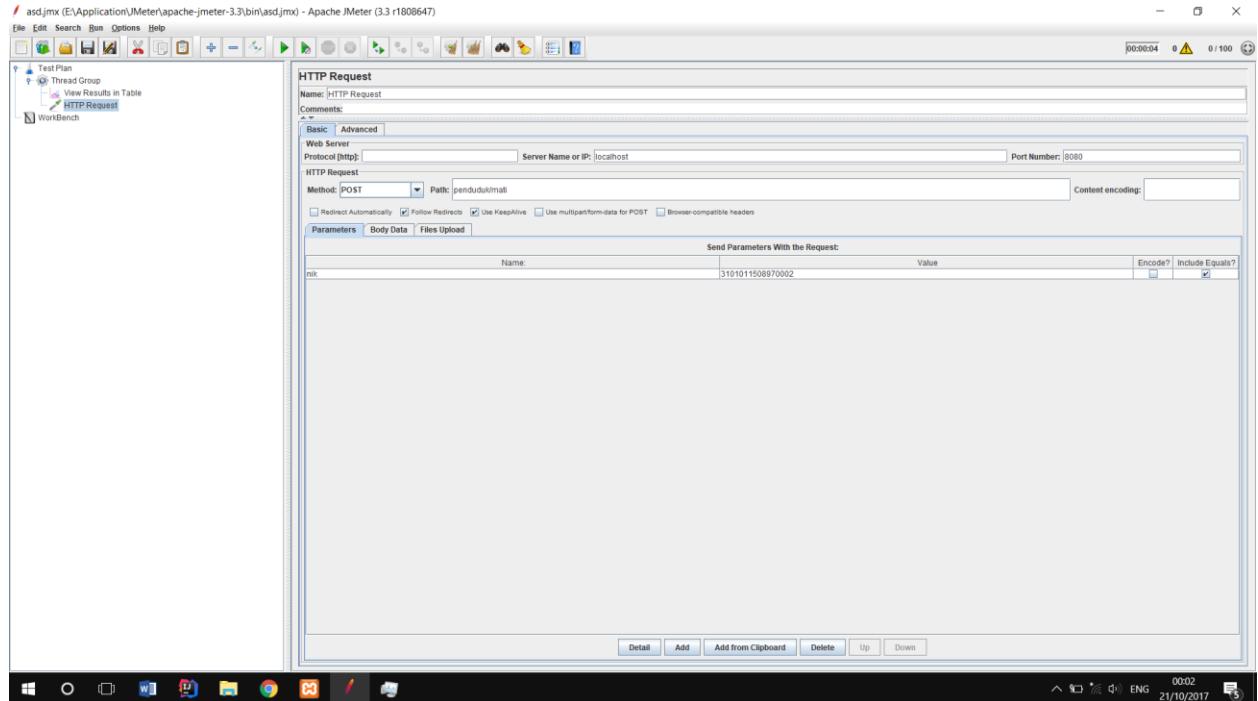
## 11. Akses update keluarga submit

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.



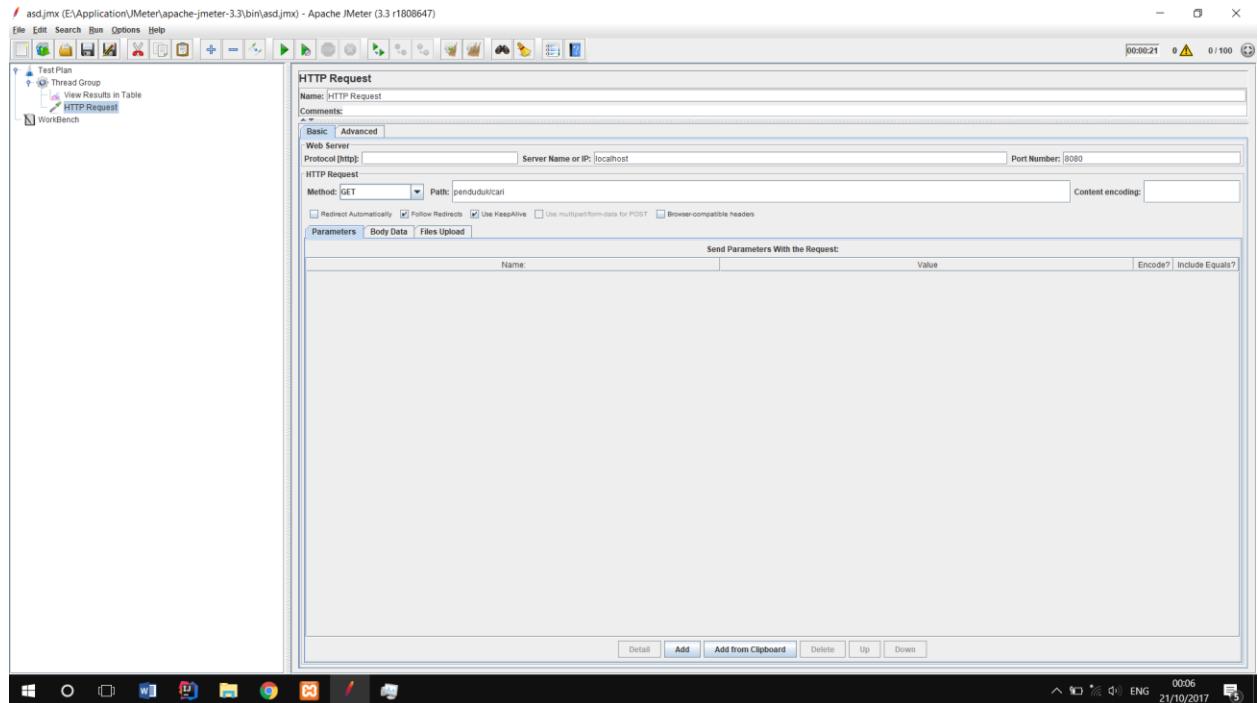
## 12. Akses nonaktifkan penduduk

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

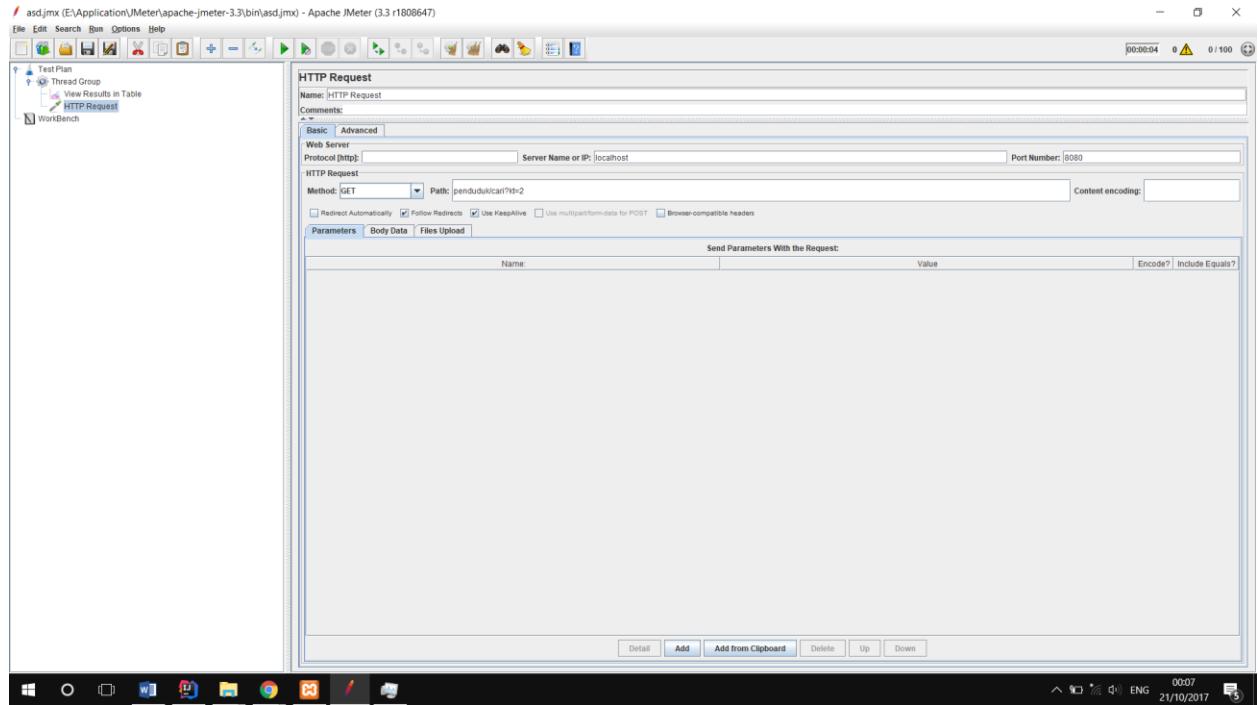
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select semua kota pada web. Dan hal ini didukung dengan penggunaan index pada kota.

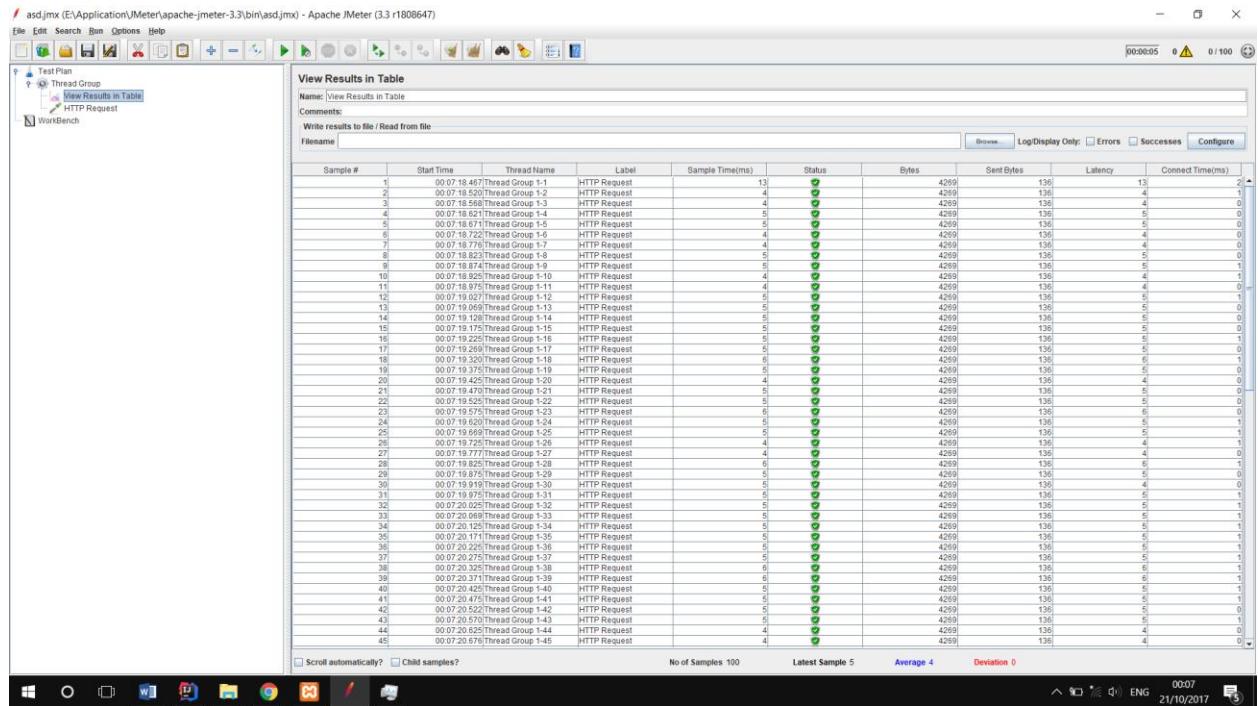


Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connected Time(ms)
1	00:06:31.487	Thread Group 1-1	HTTP Request	36	✓	3494	131	36	2
2	00:06:31.528	Thread Group 1-2	HTTP Request	4	✓	3494	131	4	1
3	00:06:31.580	Thread Group 1-3	HTTP Request	4	✓	3494	131	4	0
4	00:06:31.611	Thread Group 1-4	HTTP Request	4	✓	3494	131	4	0
5	00:06:31.653	Thread Group 1-5	HTTP Request	4	✓	3494	131	4	0
6	00:06:31.747	Thread Group 1-6	HTTP Request	5	✓	3494	131	5	0
7	00:06:31.794	Thread Group 1-7	HTTP Request	3	✓	3494	131	3	0
8	00:06:31.835	Thread Group 1-8	HTTP Request	4	✓	3494	131	4	0
9	00:06:31.894	Thread Group 1-9	HTTP Request	3	✓	3494	131	3	0
10	00:06:31.947	Thread Group 1-10	HTTP Request	6	✓	3494	131	6	1
11	00:06:31.978	Thread Group 1-11	HTTP Request	4	✓	3494	131	4	1
12	00:06:32.019	Thread Group 1-12	HTTP Request	4	✓	3494	131	4	0
13	00:06:32.078	Thread Group 1-13	HTTP Request	4	✓	3494	131	4	1
14	00:06:32.147	Thread Group 1-14	HTTP Request	4	✓	3494	131	4	1
15	00:06:32.194	Thread Group 1-15	HTTP Request	4	✓	3494	131	4	1
16	00:06:32.235	Thread Group 1-16	HTTP Request	7	✓	3494	131	7	1
17	00:06:32.278	Thread Group 1-17	HTTP Request	26	✓	3494	131	26	1
18	00:06:32.347	Thread Group 1-18	HTTP Request	4	✓	3494	131	4	1
19	00:06:32.405	Thread Group 1-19	HTTP Request	3	✓	3494	131	3	0
20	00:06:32.446	Thread Group 1-20	HTTP Request	5	✓	3494	131	5	1
21	00:06:32.494	Thread Group 1-21	HTTP Request	4	✓	3494	131	4	0
22	00:06:32.543	Thread Group 1-22	HTTP Request	4	✓	3494	131	4	1
23	00:06:32.594	Thread Group 1-23	HTTP Request	3	✓	3494	131	3	0
24	00:06:32.643	Thread Group 1-24	HTTP Request	4	✓	3494	131	4	1
25	00:06:32.679	Thread Group 1-25	HTTP Request	4	✓	3494	131	4	0
26	00:06:32.743	Thread Group 1-26	HTTP Request	4	✓	3494	131	4	1
27	00:06:32.784	Thread Group 1-27	HTTP Request	4	✓	3494	131	4	1
28	00:06:32.843	Thread Group 1-28	HTTP Request	4	✓	3494	131	4	1
29	00:06:32.893	Thread Group 1-29	HTTP Request	4	✓	3494	131	4	1
30	00:06:32.943	Thread Group 1-30	HTTP Request	5	✓	3494	131	5	1
31	00:06:33.000	Thread Group 1-31	HTTP Request	4	✓	3494	131	4	0
32	00:06:33.044	Thread Group 1-32	HTTP Request	3	✓	3494	131	3	0
33	00:06:33.094	Thread Group 1-33	HTTP Request	3	✓	3494	131	3	0
34	00:06:33.143	Thread Group 1-34	HTTP Request	3	✓	3494	131	3	0
35	00:06:33.193	Thread Group 1-35	HTTP Request	3	✓	3494	131	3	0
36	00:06:33.243	Thread Group 1-36	HTTP Request	4	✓	3494	131	4	1
37	00:06:33.294	Thread Group 1-37	HTTP Request	3	✓	3494	131	3	0
38	00:06:33.345	Thread Group 1-38	HTTP Request	4	✓	3494	131	4	1
39	00:06:33.395	Thread Group 1-39	HTTP Request	4	✓	3494	131	4	1
40	00:06:33.443	Thread Group 1-40	HTTP Request	4	✓	3494	131	4	1
41	00:06:33.493	Thread Group 1-41	HTTP Request	4	✓	3494	131	4	0
42	00:06:33.548	Thread Group 1-42	HTTP Request	4	✓	3494	131	4	1
43	00:06:33.598	Thread Group 1-43	HTTP Request	4	✓	3494	131	4	1
44	00:06:33.643	Thread Group 1-44	HTTP Request	4	✓	3494	131	4	1
45	00:06:33.693	Thread Group 1-45	HTTP Request	4	✓	3494	131	4	1

#### 14. Akses cari penduduk kecamatan

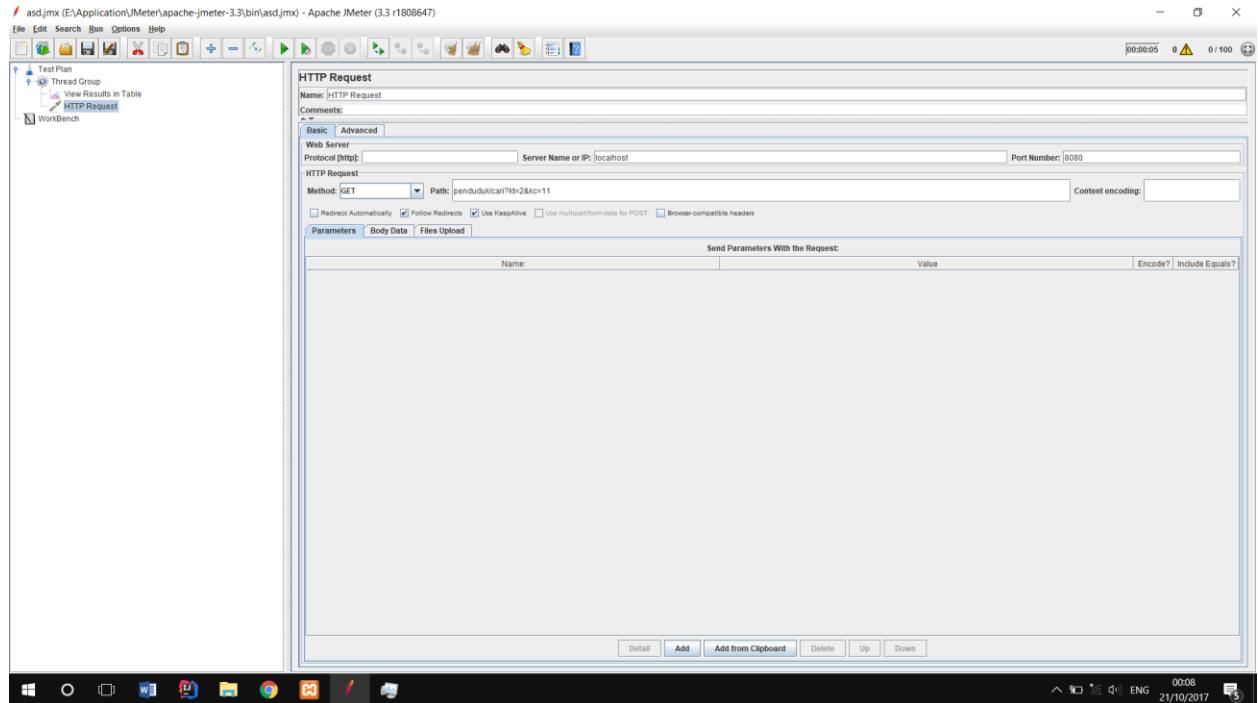
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select semua kota pada kecamatan. Dan hal ini didukung dengan penggunaan index pada kecamatan.

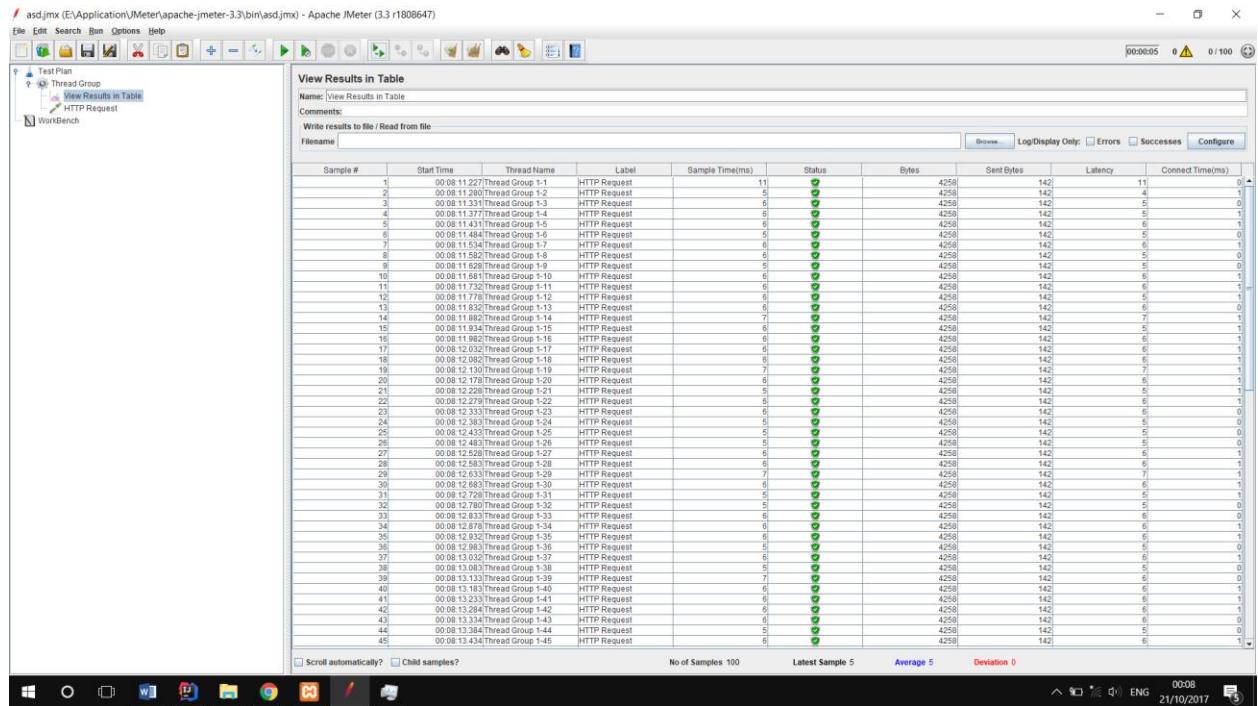




### 15. Akses cari penduduk kelurahan

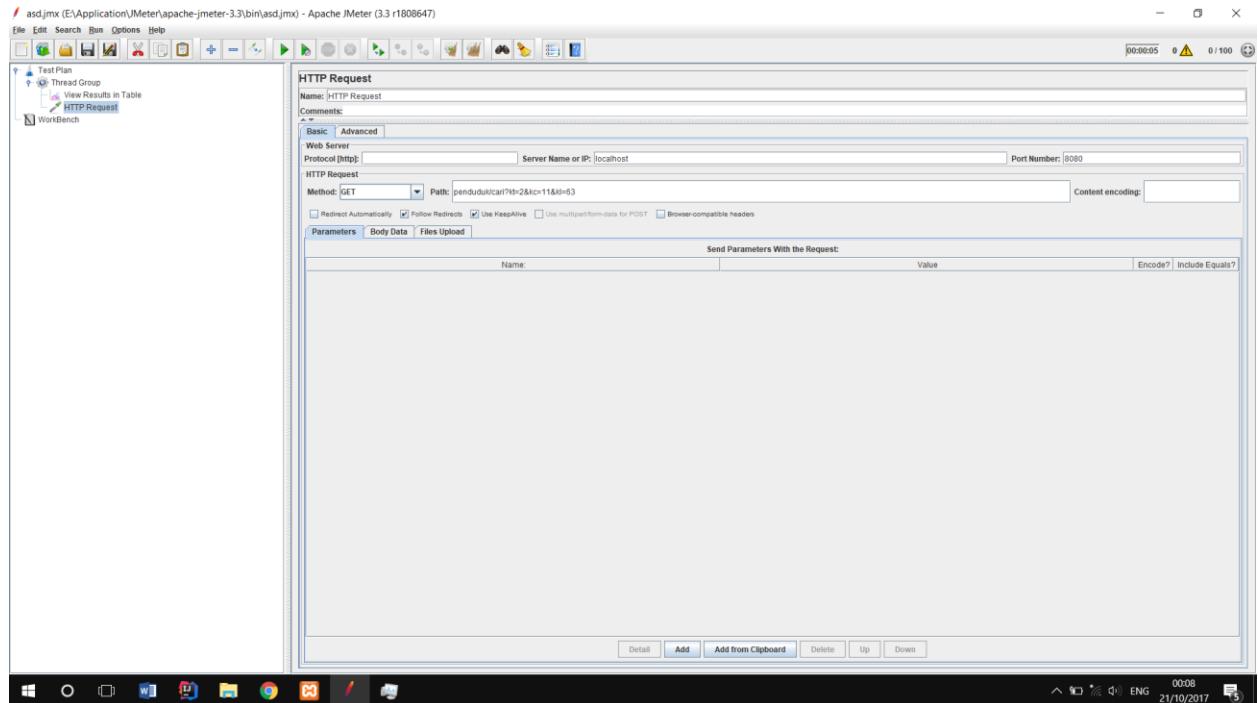
Halaman dapat berjalan dengan cukup cepat dimana web hanya menampilkan html,css, dan select semua kelurahan pada web. Dan hal ini didukung dengan penggunaan index pada kelurahan.





## 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	00:09:03:642	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	00:09:03:230	Thread Group 1-2	HTTP Request	20502	✓	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 1-6	HTTP Request	22494	✓	194141	148	22462	0
12	00:09:03:230	Thread Group 1-8	HTTP Request	22545	✓	194141	148	23571	1
13	00:09:03:160	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	00:09:03:230	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	00:09:05:433	Thread Group 1-46	HTTP Request	30906	✓	194141	148	30898	0
20	00:09:05:230	Thread Group 1-22	HTTP Request	32309	✓	194141	148	32327	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	00:09:05:230	Thread Group 1-32	HTTP Request	31711	✓	194141	148	31784	0
24	00:09:04:435	Thread Group 1-26	HTTP Request	32103	✓	194141	148	32097	0
25	00:09:05:232	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	00:09:05:230	Thread Group 1-25	HTTP Request	32144	✓	194141	148	32168	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	00:09:05:230	Thread Group 1-21	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	00:09:05:230	Thread Group 1-7	HTTP Request	34590	✓	194141	148	35503	1
36	00:09:05:937	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	00:09:05:230	Thread Group 1-41	HTTP Request	34520	✓	194141	148	34520	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	00:09:04:296	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	00:09:05:262	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