

Write Up Tugas 1.

Proses Pengembangan

Pada pembuatan tugas 1 kali ini, saya mulai dengan membaca soal untuk menganalisa dan mendapatkan spesifikasi aplikasi yang diminta, setelah mengetahui apa saja fitur yang harus dikerjakan, saya mulai membuat aplikasi pada bagian *front-end*.

Dalam mengerjakan bagian *front-end*, saya sempat bertanya kepada salah satu asisten dosen, yaitu kak Jundi Ahmad Alwan, apakah saya boleh membuat aplikasi tanpa menggunakan *framework CSS Bootstrap*, alasan saya menanyakan hal ini karena saya lebih suka membuat sisi *front-end* dari sebuah website tanpa menggunakan framework CSS, karena hal tersebut menurut saya hanya menyusahkan dan membuat kerja saya semakin lama.

Setelah dikonfirmasi boleh untuk tidak menggunakan *framework CSS Bootstrap*, saya mulai membuat sisi *front-end* dengan pertama-tama memilih library untuk CSS reset agar tampilan website terus konsisten pada browser yang berbeda beda, untuk melakukan ini, saya memilih untuk menggunakan [**Sanitize.css**](#). Setelah memilih library untuk CSS reset, saya memilih font [**Inter UI**](#). Saya juga menggunakan *icon-font* yang saya pilih sendiri dan disediakan oleh [**Flaticon**](#). Setelah mendapatkan semua dependensi yang diperlukan saya mulai membuat tampilan dalam bentuk *HTML & CSS* untuk setiap halaman, tampilan ini masih berbentuk *static* dan belum menampilkan data dinamis apapun.

Setelah selesai membuat tampilan, saya lalu membuat Model untuk setiap tabel database terlebih dahulu, disini saya menyiapkan 5 model untuk dipakai, yaitu PendudukModel, KeluargaModel, KelurahanModel, KecamatanModel, dan KotaModel. Semua model ini memiliki variabel data yang sesuai dengan kolom yang ada pada tabel masing masing, dan memiliki validasi sesuai dengan requirement yang ada (NotNull, Type, Size). Setelah selesai membuat model, saya kemudian membuat Service & DAO Mapper untuk setiap tabel dan menggunakan Model yang diperlukan untuk masing masing query/request. Setelah DAO & Service Siap, saya melanjutkan membuat PageController, sebuah *controller* utama yang mengendalikan setiap request yang diberikan ke aplikasi. Setelah semua class siap, saya mulai mengerjakan aplikasi per fitur, penjelasan setiap fitur ada pada bagian dibawah ini.

Daftar Fitur Aplikasi

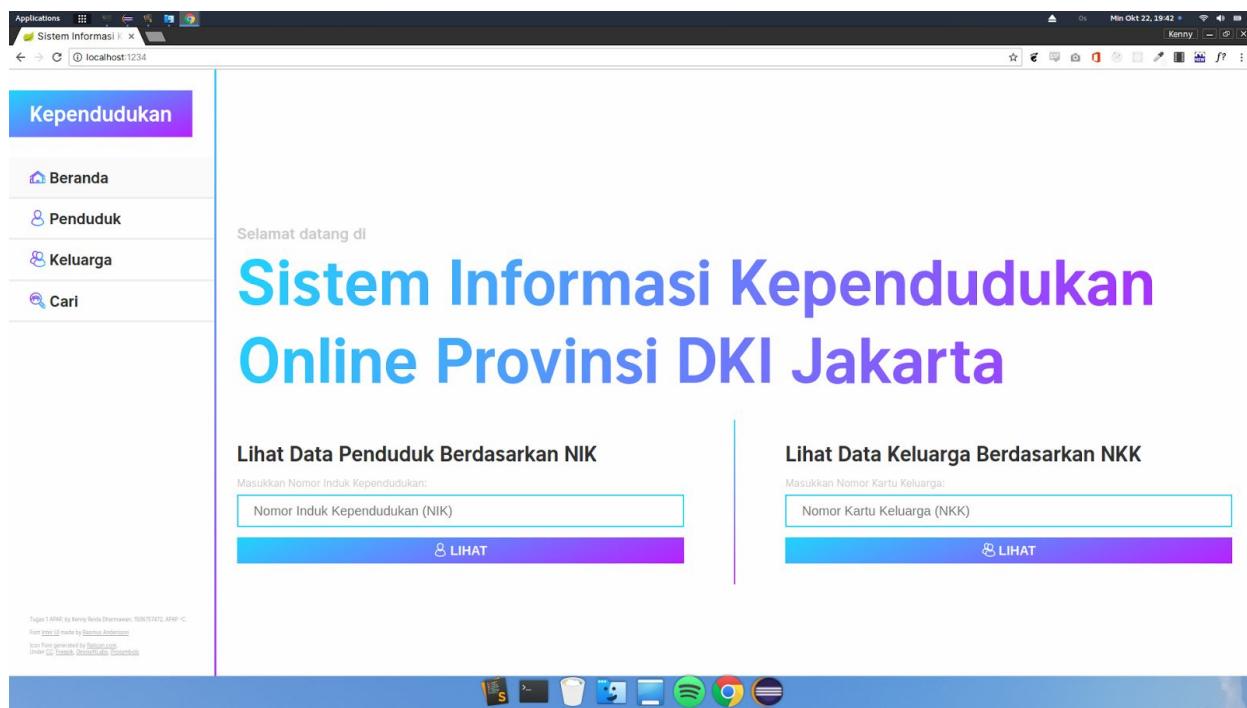
1. Tampilkan Data Penduduk Berdasarkan NIK

Untuk mengakses fitur pertama, pengguna dapat membuka halaman index ('/') [Gambar 1.1] atau halaman Penduduk ('/penduduk') [Gambar 1.2] dan memasukan NIK yang ingin dicari pada input yang ada.

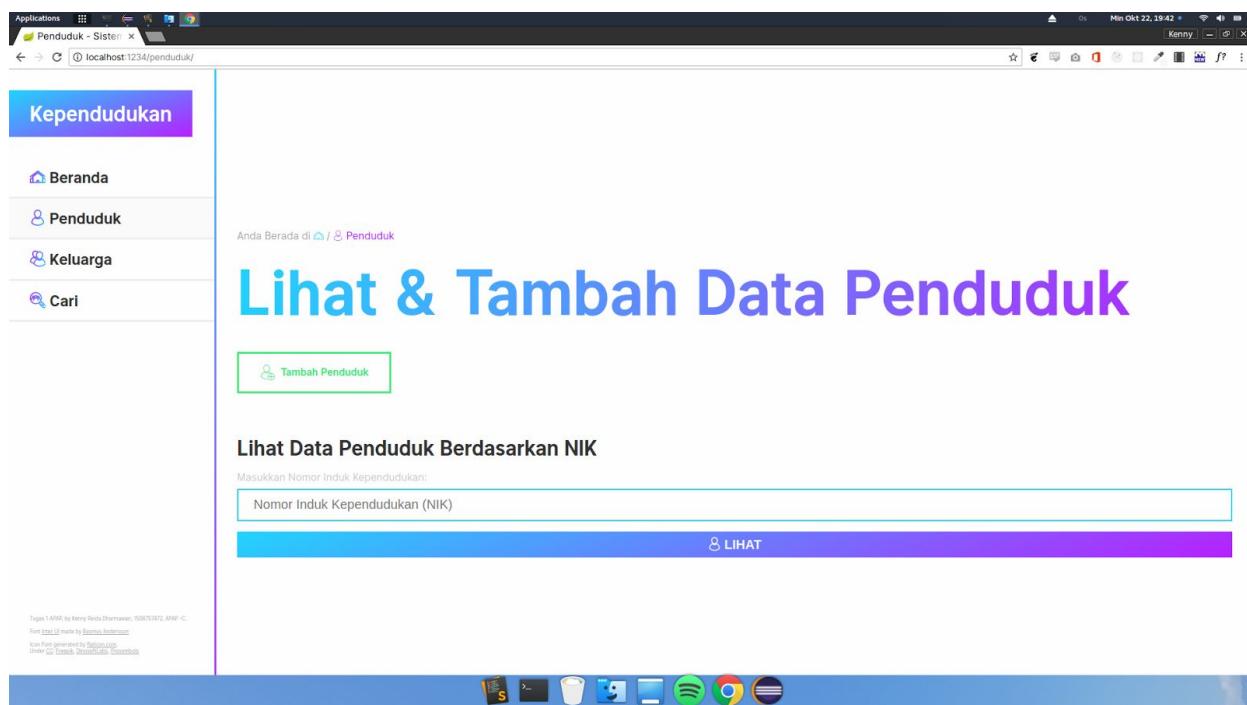
Setelah memasukkan input, pengguna lalu akan diarahkan ke Halaman Lihat Penduduk Berdasarkan NIK ('/penduduk?nik=xxxx') [Gambar 1.3, 1.4, & 1.5]. Pada halaman tersebut, pengguna dapat melihat data lengkap dari seorang penduduk dengan NIK yang telah dimasukkan, yang tentunya menampilkan data yang mudah dicerna. Untuk menampilkan data ini, saya melakukan query ke tabel pengguna untuk mendapatkan pengguna berdasarkan NIK yang diberikan, lalu saya melakukan query ke tabel keluarga untuk mendapatkan id_kelurahan penduduk tersebut, setelah mendapatkan id_kelurahan, saya melakukan query ke tabel kelurahan untuk mendapatkan id_kecamatan dan nama_kelurahan, lalu saya melakukan query ke tabel kecamatan untuk mendapatkan id_kota dan nama_kecamatan, akhirnya saya melakukan query ke tabel kota untuk mendapatkan nama_kota, semua query bersambung ini dilakukan dengan parameter id tabel masing masing yang diberikan saat query berlangsung.

Setelah berhasil melakukan query data, lalu saya menampilkan data tersebut dengan cara memasukan data yang telah diambil ke template view yang telah dibuat, proses mencari data dilakukan oleh Service, yang memanggil DAO untuk meminta data ke database. Service dipanggil oleh Controller, dimana Controller akan meminta data dari Service dan memasukkannya ke view template, data yang diminta harus sesuai berupa Model yang telah dibuat untuk mempunyai semua data yang ada di suatu tabel database yang terkait.

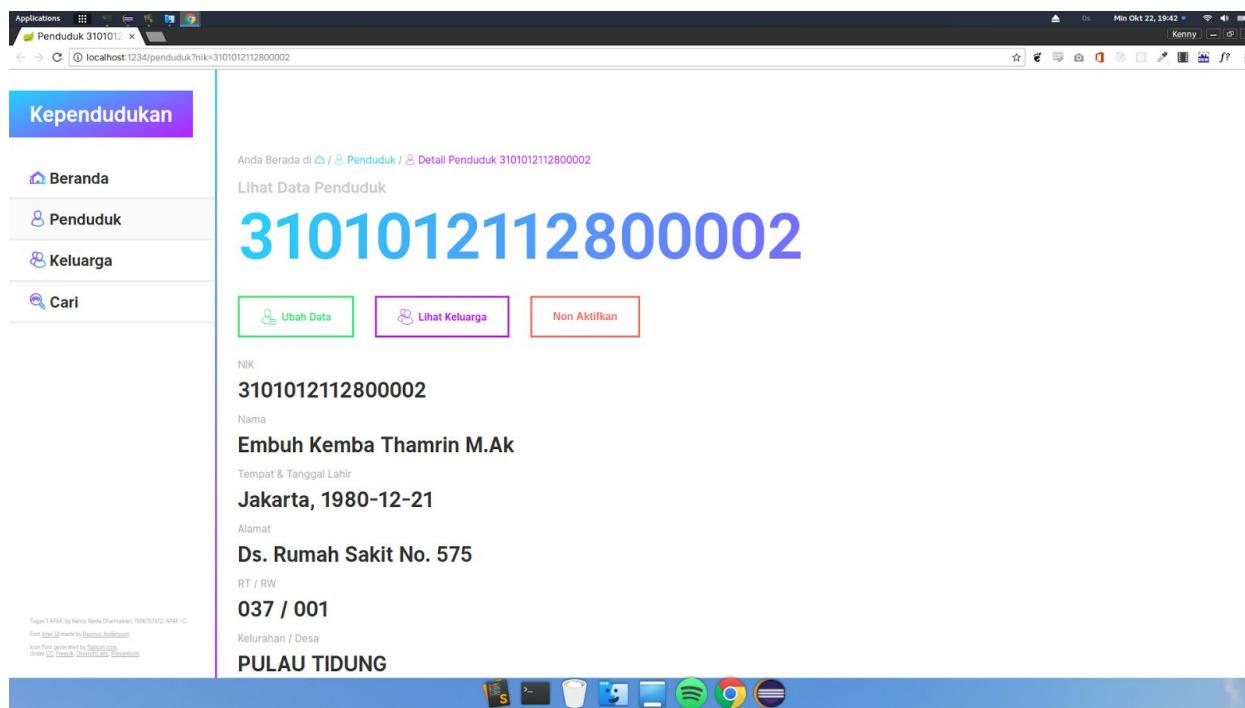
Setelah data yang dicari ada dan dilanjutkan ke view, maka thymeleaf akan memasukkan data ke template dan merender html sesuai dengan template yang kita buat untuk dikembalikan ke client yang merequest template tersebut. Tentunya dalam merender tampilan ini, saya menampilkan data yang lebih mudah dibaca, seperti ketika status data is_wafat = 0, maka saya akan menuliskan status penduduk = Hidup, dsb.



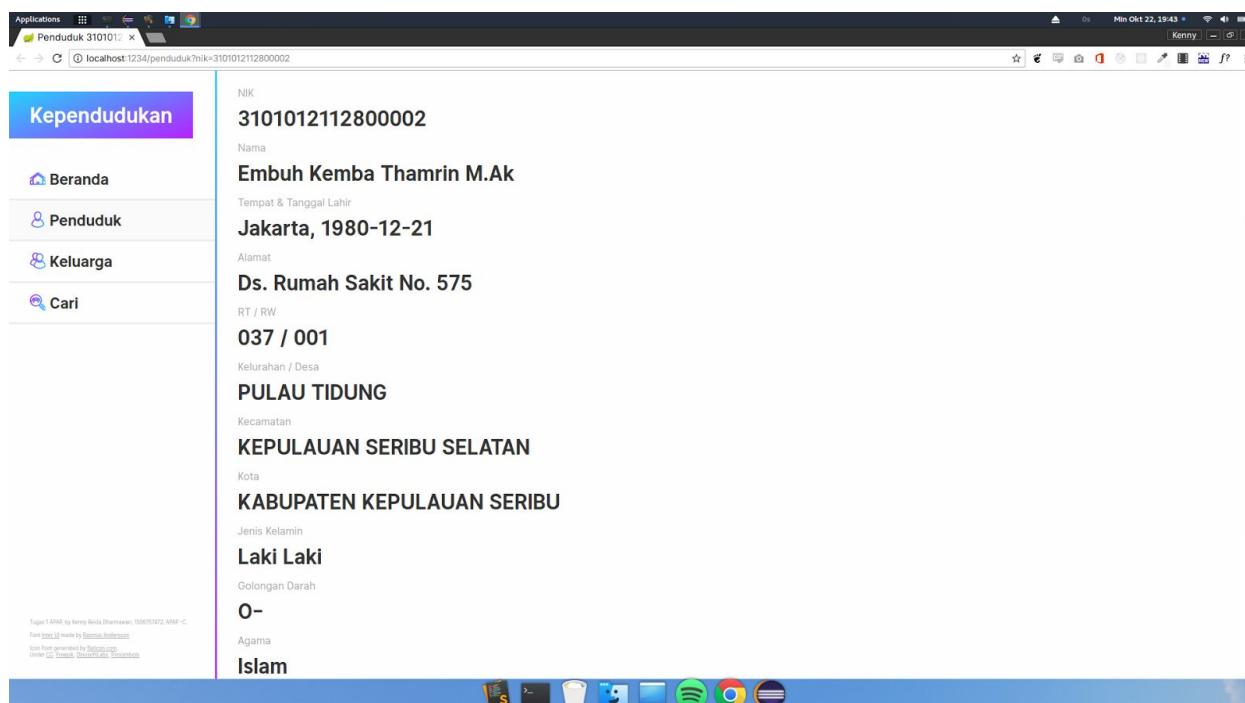
Gambar 1.1 - Halaman Index ('/')



Gambar 1.2 - Halaman Penduduk ('/penduduk/')



Gambar 1.3 - Halaman Lihat Penduduk Berdasarkan NIK ('/penduduk?nik=xxxx')



Gambar 1.4 - Halaman Lihat Penduduk Berdasarkan NIK ('/penduduk?nik=xxxx') - Lanjutan



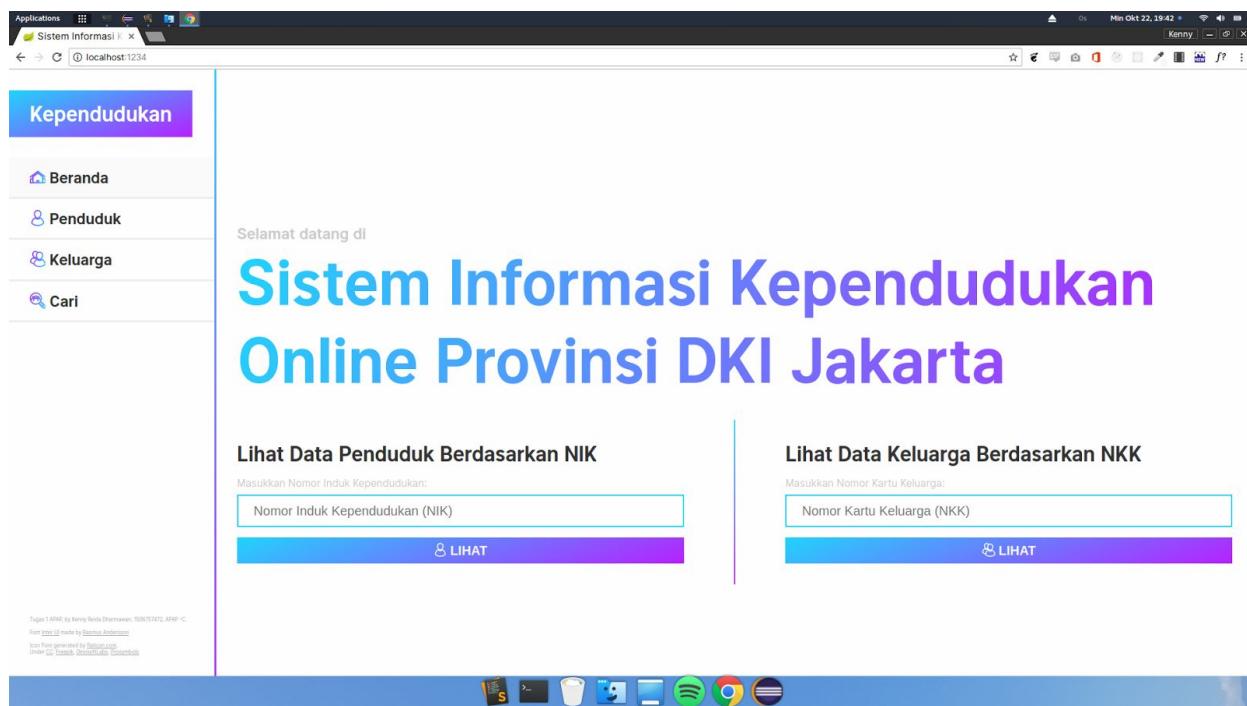
Gambar 1.5 - Halaman Lihat Penduduk Berdasarkan NIK ('/penduduk?nik=xxxx') - Lanjutan

2. Tampilkan Data Keluarga Berdasarkan NKK

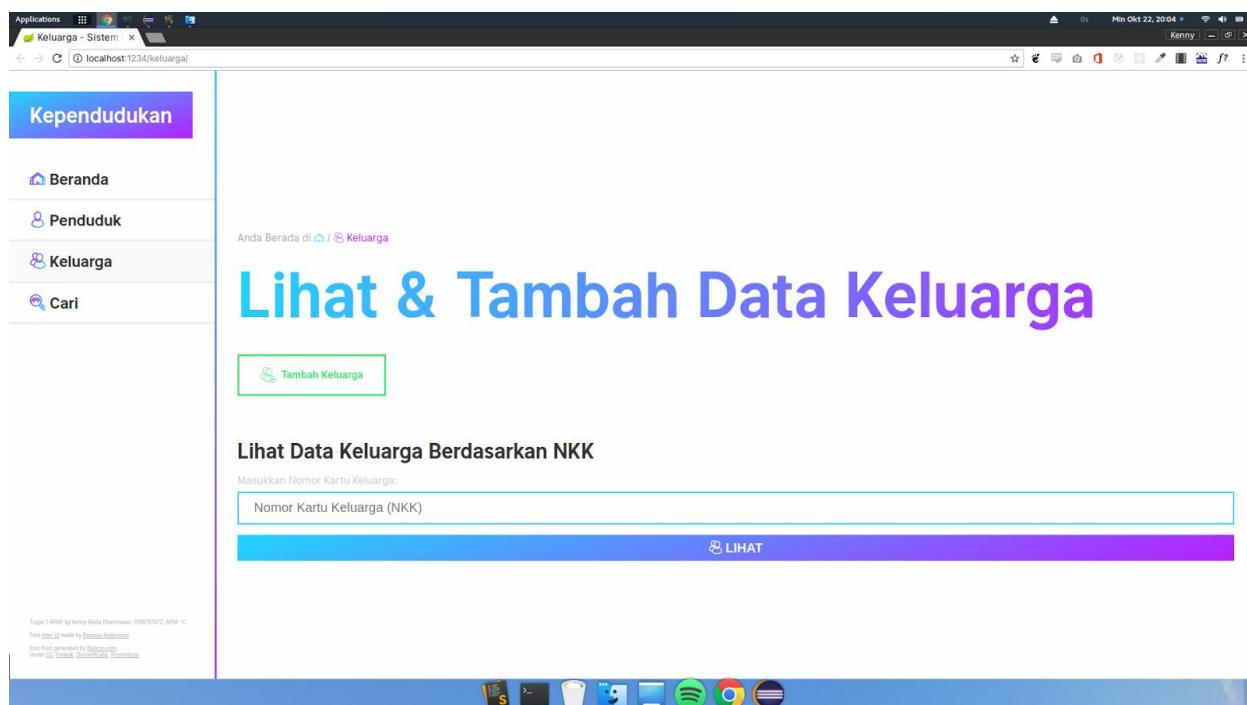
Untuk mengakses fitur kedua, pengguna dapat membuka halaman index ('/') [Gambar 2.1] atau halaman Keluarga ('/keluarga/') [Gambar 2.2] dan memasukan NKK yang ingin dicari pada input yang ada.

Setelah berhasil memasukkan NKK, maka pengguna akan diarahkan ke halaman Lihat Keluarga Berdasarkan NKK ('/keluarga?nkk=xxx') [Gambar 1.3 & 1.4], Pada proses ini, kurang lebih pada dasarnya sama dengan query ke database sesuai dengan Fitur 1, akan tetapi pada fitur ini dilakukan query ke tabel keluarga terlebih dahulu untuk mendapatkan keluarga, lalu ke tabel penduduk untuk mendapatkan semua penduduk yang terdaftar pada keluarga tersebut. Tak lupa juga melakukan query ke tabel kelurahan, kecamatan, dan kota untuk mendapatkan nama kelurahan, kecamatan, dan kota.

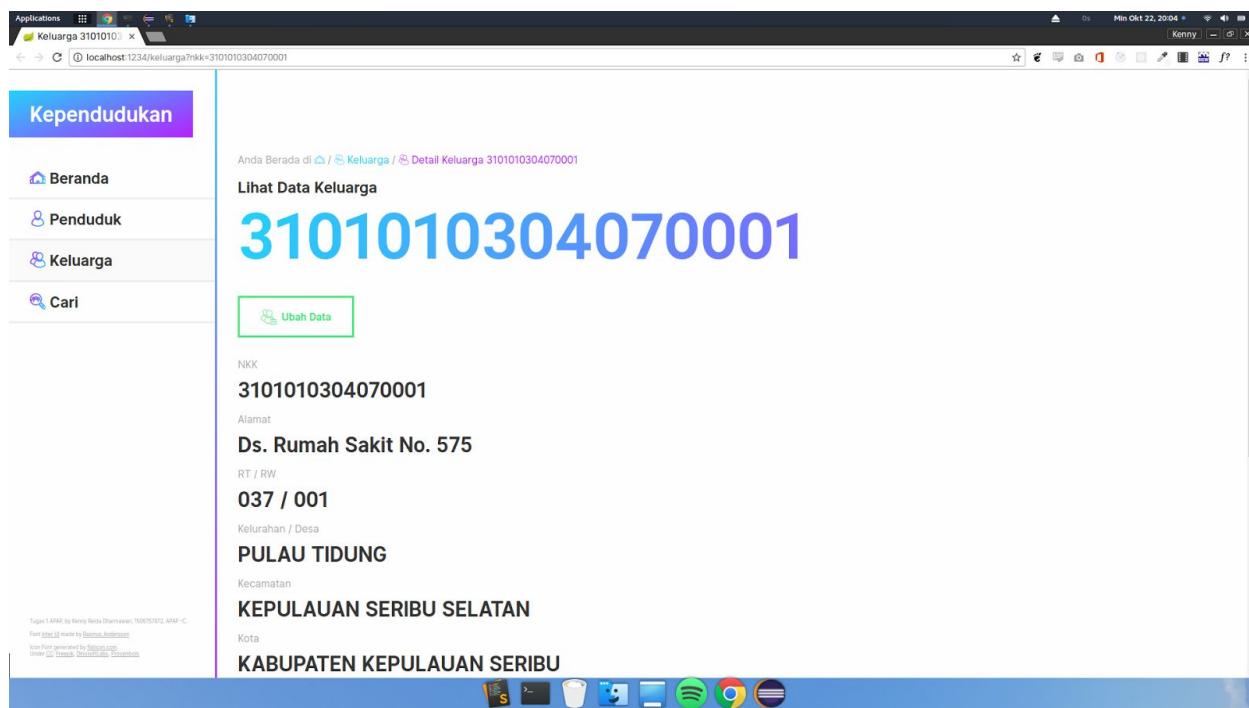
Setiap anggota keluarga ditampilkan dalam sebuah tabel, dimana pada kolom terakhir setiap baris data tersebut, terdapat link yang ketika dipencet akan mengarahkan pengguna ke halaman data penduduk tersebut.



Gambar 2.1 - Halaman Index ('/')



Gambar 2.2 - Halaman Keluarga ('/keluarga/')



Gambar 2.3 - Halaman Lihat Keluarga Berdasarkan NKK ('/keluarga?nkk=xxx')

No	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga	Kewarganegaraan	Data
1	Ida Pratiwi	3101017110820001	Wanita	Jakarta	1982-10-31	Islam	PETERNAK	Kawin	Istri	WNI	Lihat
2	Prasetya Megantara	3101010405100001	Laki Laki	Jakarta	2010-05-04	Islam	PELAJAR/MAHASISWA	Belum Kawin	Anak	WNI	Lihat
3	Embeh Kemba Thamrin M.Ak	3101012112800002	Laki Laki	Jakarta	1980-12-21	Islam	PENATA RAMBUT	Kawin	Kepala Keluarga	WNI	Lihat
4	Jaiman Wawan Nainggolan S.Pd	3101011409170002	Laki Laki	Jakarta	2017-09-14	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI	Lihat

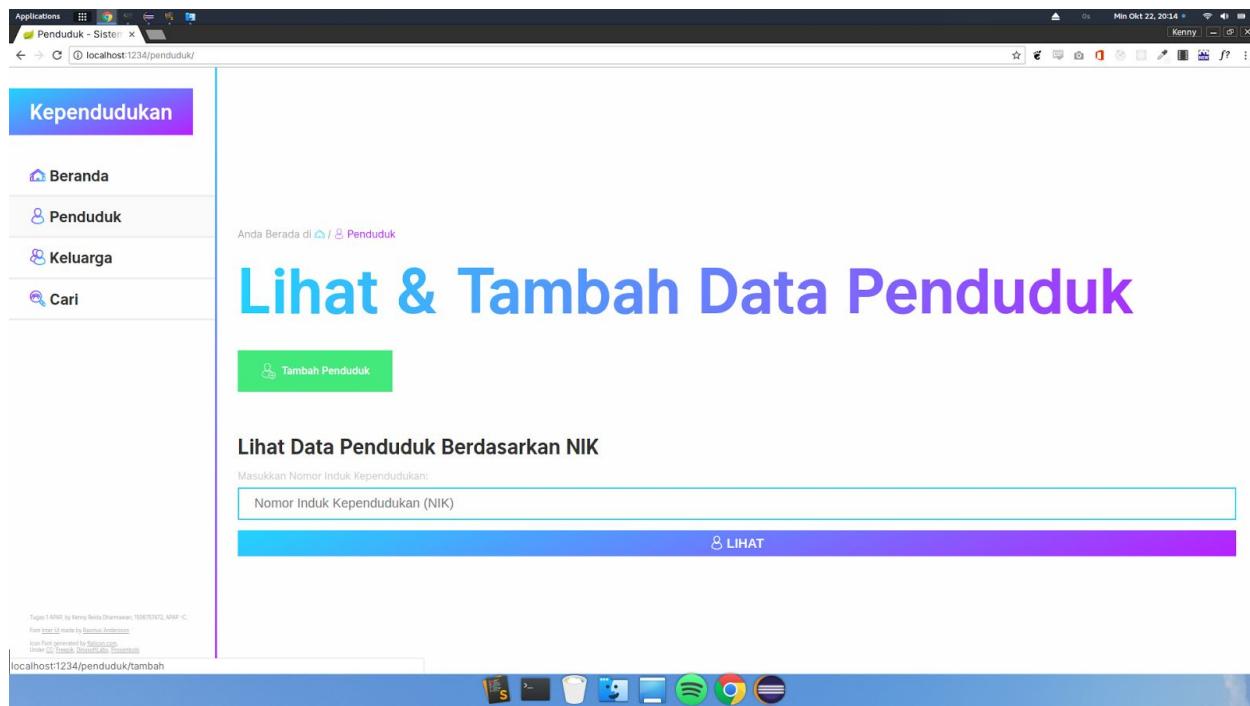
Gambar 2.4 - Halaman Lihat Keluarga Berdasarkan NKK ('/keluarga?nkk=xxx') - Lanjutan

3. Tambah Penduduk Baru Sebagai Anggota Keluarga

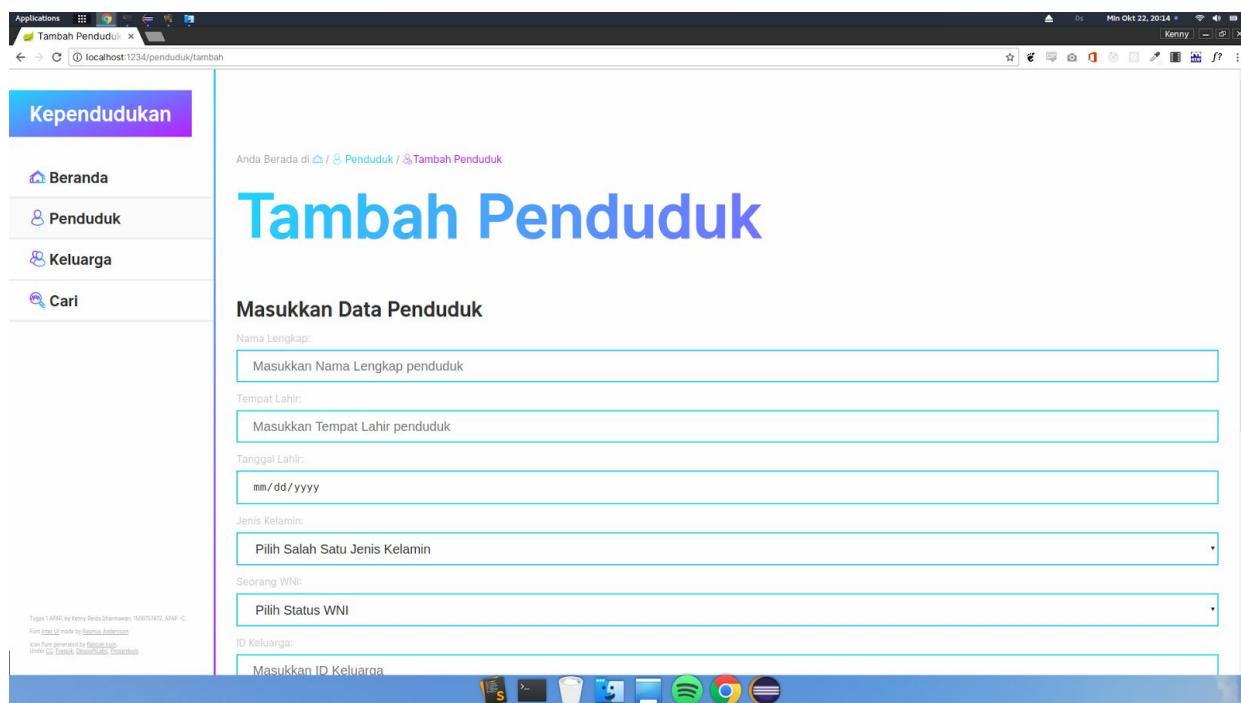
Dalam Fitur ke tiga, pengguna bisa menambahkan data penduduk baru dengan cara menavigasi ke halaman Penduduk ('/penduduk') dan menekan tombol Tambah Penduduk seperti pada [Gambar 3.1]. Setelah menekan tombol, maka pengguna akan diarahkan ke halaman Form Penambahan Penduduk ('/penduduk/tambah') [Gambar 3.2, 3.3]. Pada halaman ini, pengguna akan diminta untuk mengisi data penduduk baru, setelah semua diisi, maka pengguna dapat menekan tombol TAMBAH seperti terlihat pada [Gambar 3.3].

Setelah berhasil mengisi form dan menekan tombol TAMBAH, maka pengguna akan diarahkan ke halaman Penambahan Data Sukses ('/penduduk/tambah') jika data yang diberikan valid (Terkait validasi dapat dilihat pada Fitur 10: Validasi dibawah). Jika valid, maka pengguna akan diberikan tampilan konfirmasi bahwa data berhasil dimasukkan, dan dapat menekan tombol Lihat Data [Gambar 3.4] agar diarahkan ke halaman Lihat Penduduk Berdasarkan NIK ('/penduduk?nik=xxx') [Gambar 3.5].

Dalam membuat fitur ini, saya membuat 2 kontroller untuk link '/penduduk/tambah', satu untuk menerima request dengan method GET dan akan mengembalikan tampilan formulir data, dan satu lagi dengan method POST untuk menangkap data yang dikirimkan yang berupa data yang sesuai dengan PendudukModel. Jika data sudah sesuai dan valid, maka dari data tersebut akan dibuatkan penduduk baru dengan NIK yang digenerate sesuai dengan spesifikasi NIK yang ada. Dalam menggenerate NIK, program akan membuat prefix yang terdiri dari kode provinsi-kota-kecamatan yang diambil dari field kode_kecamatan pada kecamatan keluarga penduduk yang dituju. Setelah mendapatkan kode kecamatan, lalu prefix akan ditambahkan lagi dengan tanggal lahir penduduk, jika wanita maka hari lahir akan ditambahkan 40 hari, prefix tanggal lahir berformat dd/mm/yy, setelah mendapatkan prefix yang diharapkan, maka selanjutnya program akan melakukan query untuk menghitung penduduk dengan prefix yang sama, ketika sudah mendapatkan jumlah penduduk dengan prefix yang sama, maka prefix akan ditambahkan nomor urut yang sesuai, lalu jadilah NIK. Karena pada database id tidak di autogenerate, maka dalam pembuatan penduduk maka id akan digenerate program, dengan cara menghitung jumlah seluruh penduduk yang ada dan menambahkan satu angka lagi untuk menjadikan ID penduduk yang baru.



Gambar 3.1 - Halaman Penduduk ('/penduduk/') - Klik Tombol Tambah Penduduk



Gambar 3.2 - Form Tambah Penduduk ('/penduduk/tambah/')

Jenis Kelamin:
Pilih Salah Satu Jenis Kelamin

Seorang WNI:
Pilih Status WNI

ID Keluarga:
Masukkan ID Keluarga

Status Dalam Keluarga:
Masukkan Status Dalam Keluarga

Agama:
Pilih Salah Satu Agama

Pekerjaan:
Pilih Salah Satu Pekerjaan Dibawah Ini

Status Perkawinan:
Masukkan Status Kawin

Golongan Darah:
Masukkan Golongan Darah

TAMBAH

Gambar 3.3 - Form Tambah Penduduk ('/penduduk/tambah/') - Lanjutan

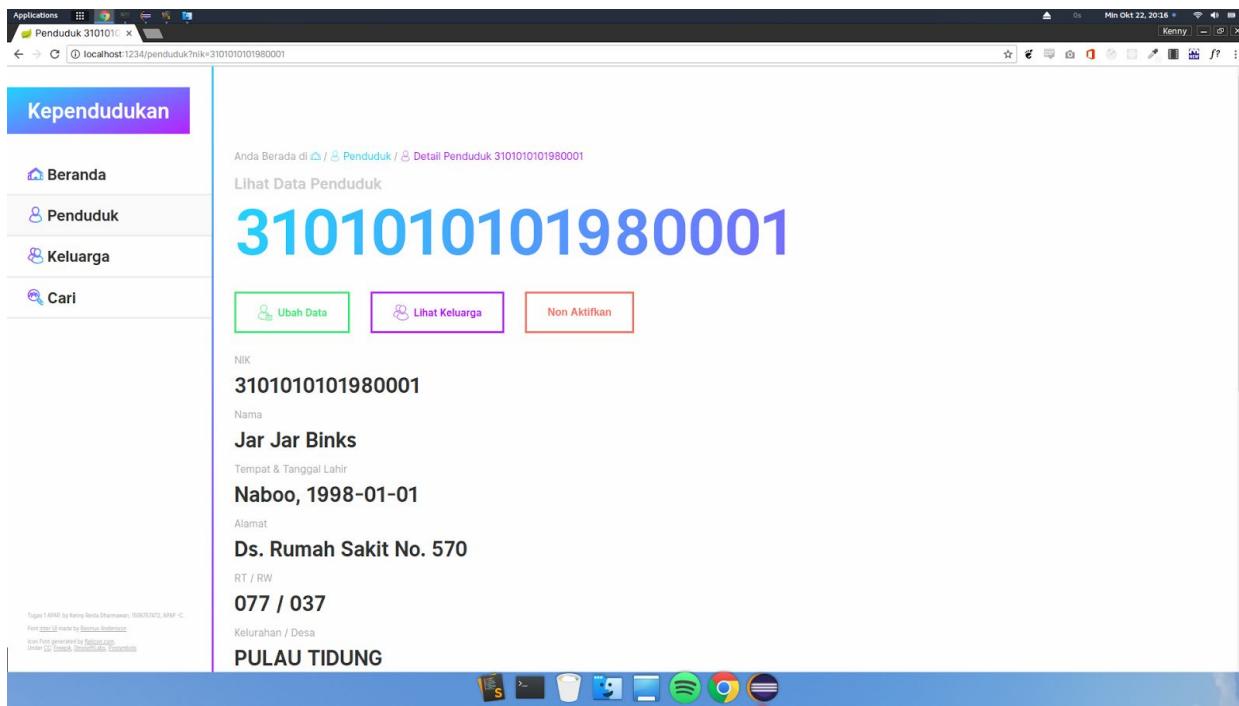
Anda Berada di [Beranda](#) / [Penduduk](#) / [Tambah Penduduk](#) / [Tambah Penduduk Berhasil](#)

Penduduk Berhasil Ditambahkan!

Dengan NIK: 3101010101980001

[Lihat Data](#)

Gambar 3.4 - Penambahan Data Sukses ('/penduduk/tambah/')



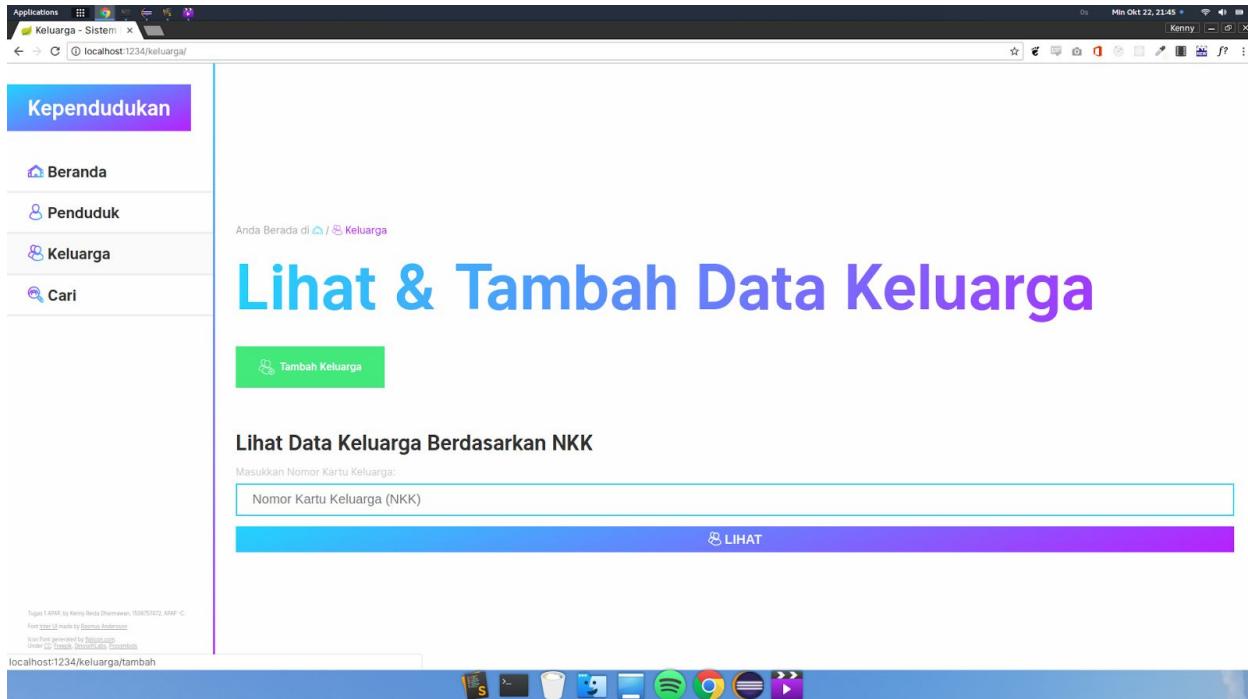
Gambar 3.5 - Lihat Data Penduduk yang Baru saja dibuat.

4. Tambah Keluarga Baru

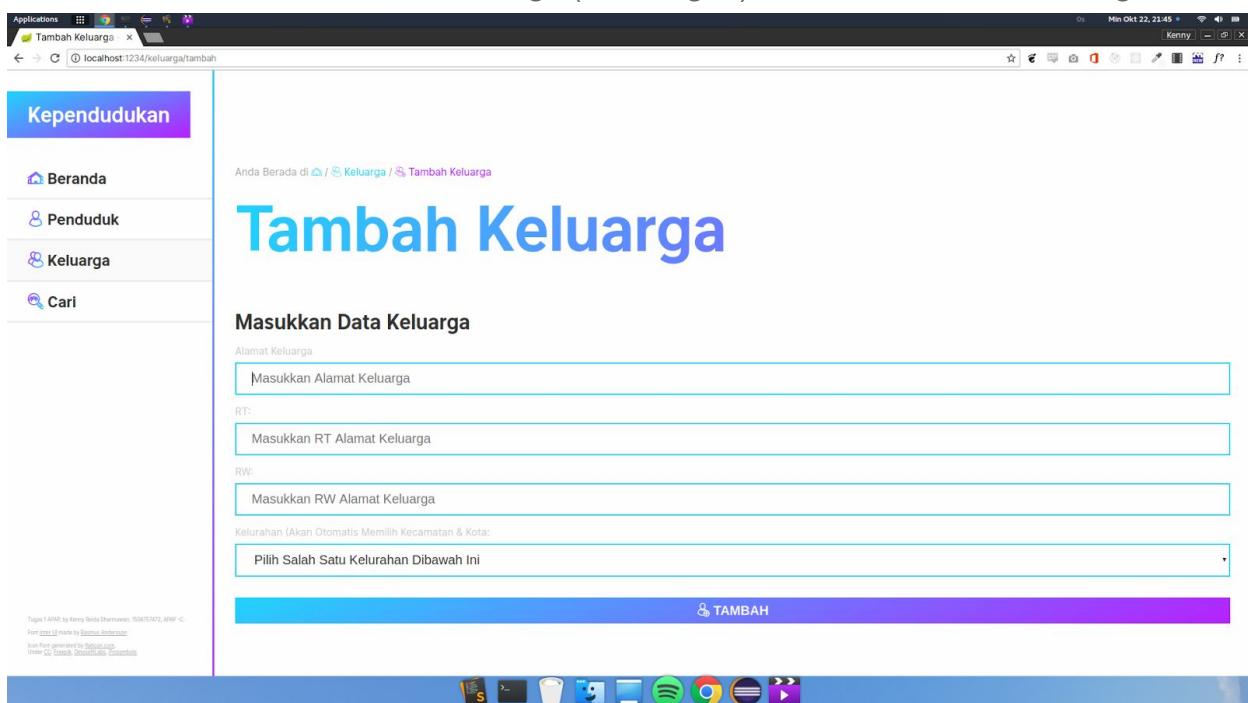
Dalam Fitur ke empat, sama dengan fitur ketiga, pengguna bisa menuju ke halaman Keluarga ('/keluarga/') dan menekan tombol Tambah Keluarga [Gambar 4.1], setelah itu pengguna akan diarahkan ke halaman form Tambah Keluarga ('/keluarga/tambah') [Gambar 4.2], setelah mengisi form dengan benar maka pengguna akan diarahkan ke halaman Tambah Keluarga Sukses ('/keluarga/tambah/') [Gambar 4.3] dimana pada halaman tersebut pengguna bisa melihat NKK dari keluarga yang baru dibuat dan dapat menekan tombol Lihat Data untuk diarahkan ke halaman Lihat Keluarga Berdasarkan NKK ('/keluarga?nkk=xxxx') [Gambar 4.4]

Dalam membuat Keluarga baru, setelah mendapatkan data baru, maka program akan menggenerate NKK sesuai dengan spesifikasi, pertama dengan mendapatkan prefix Provinsi-Kota-Kecamatan dari Kecamatan yang dipilih (diambil dari Kelurahan yang dipilih). Setelah mendapat prefix tersebut, program akan mencatat kapan proses pembuatan terjadi dan diambil hari-bulan-tahun dalam format DDMMYY dan digabungkan dengan prefix lokasi untuk dijadikan prefix nkk, dimana setelah mendapatkan prefixnya, program akan menghitung nkk yang sudah ada dengan prefix yang diminta untuk mengetahui nomor urut nkk. Setelah mendapatkan nomor

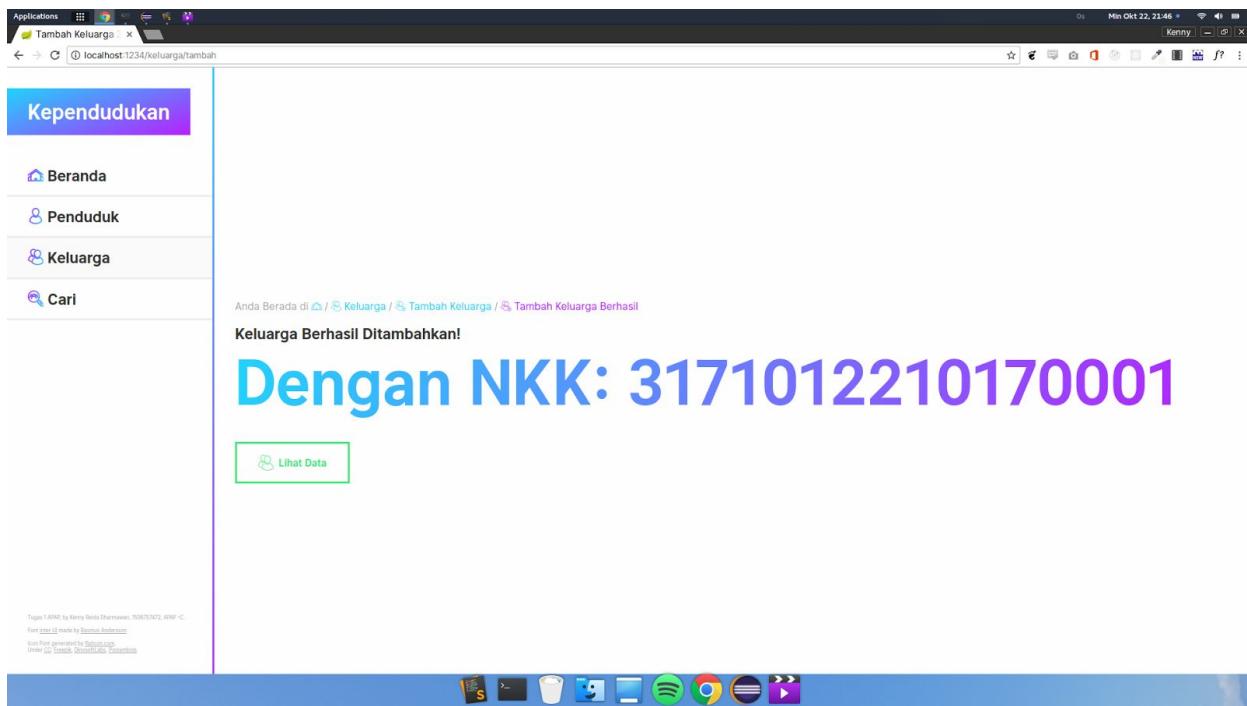
urut dan menambahkannya dengan prefix, maka jadilah NKK untuk keluarga yang baru dibuat, setelah itu data yang ada dan nkk nya dimasukkan ke database pada tabel keluarga.



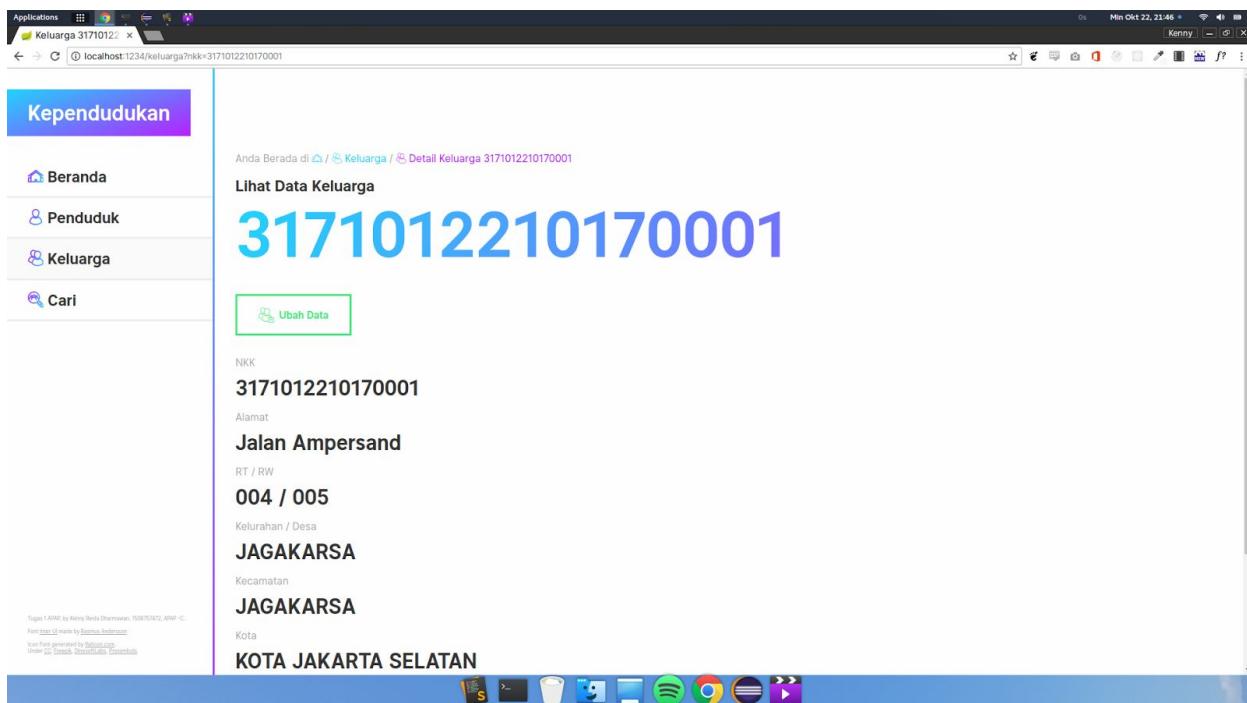
Gambar 4.1 - Halaman Keluarga ('/keluarga/') - Tombol Tambah Keluarga



Gambar 4.2 - Form Tambah Keluarga ('/keluarga/tambah/')



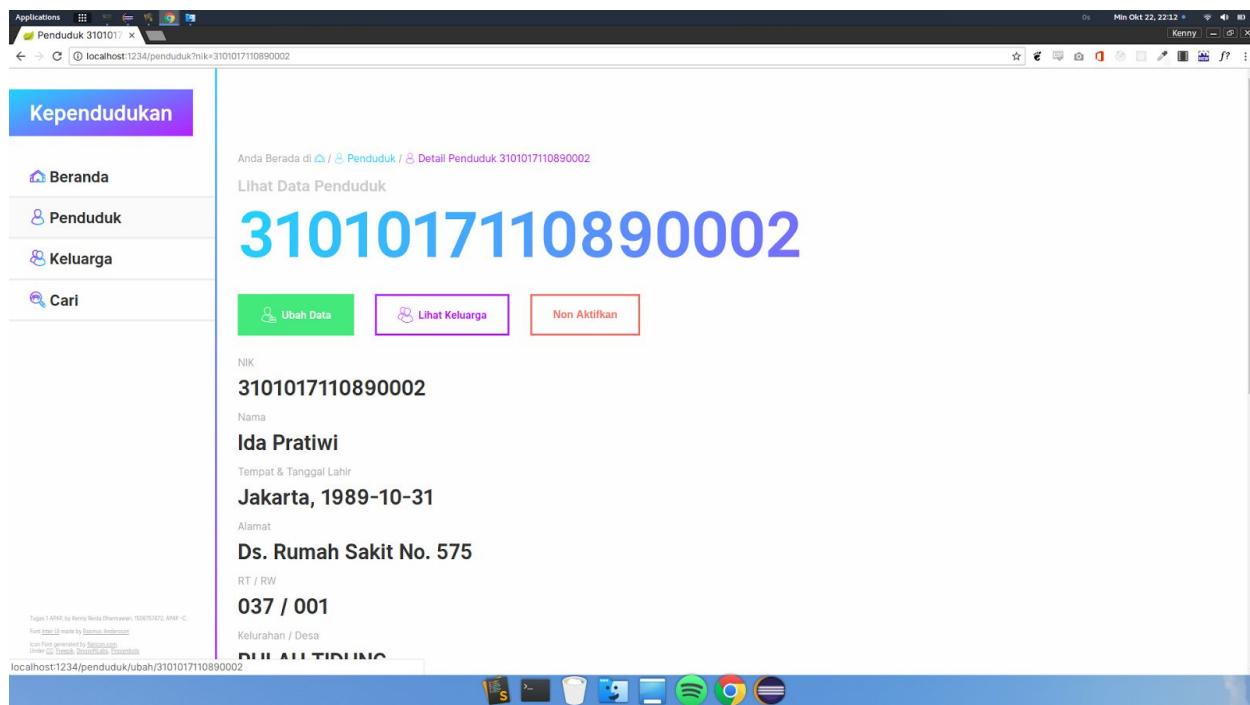
Gambar 4.3 - Sukses Tambah Keluarga ('/keluarga/tambah/')



Gambar 4.4 - Lihat Keluarga Berdasarkan NKK ('/keluarga?nkk=xxxx')

5. Ubah Data Penduduk

Dalam Fitur ke lima, pengguna bisa merubah data, dalam merubah data penduduk bisa menekan tombol Ubah Data [Gambar 5.1], setelah menekan tombol pengguna akan diarahkan ke form Ubah Data ('/penduduk/ubah/nikxxxx') [Gambar 5.2], disini ketika pengguna mengubah data dan menekan tombol simpan maka pengguna akan diarahkan ke halaman Sukses Ubah Data [Gambar 5.3] dengan asumsi data valid, jika ternyata data yang dirubah berhubungan dengan NIK, seperti tanggal lahir [Gambar 5.4], maka ketika tombol simpan dipencet, maka program akan membuatkan NIK baru dengan prosedur sama seperti prosedur pembuatan NIK saat pembuatan penduduk baru di tahap 3, dan pengguna akan diarahkan ke halaman Sukses Ubah Data dan ditampilkan NIK baru [Gambar 5.5] dengan link ubah data masih NIK lama dan breadcrumbs masih NIK lama juga.



Gambar 5.1 - Penduduk ('/penduduk?nik=xxxx') - Tombol Ubah Data

Kependudukan

Beranda

Penduduk

Keluarga

Cari

Ubah Data Penduduk

NIK: 3101017110890002

Nama Lengkap: Ida Pratiwi Martini

Tempat Lahir: Jakarta

Tanggal Lahir: 10/31/1989

Jenis Kelamin: Wanita

Seorang WNI: Ya

ID Keluarga: 67

Gambar 5.2 - Form Ubah Data Penduduk ('/penduduk/ubah/nikxxxx/') - Ubah Nama

Kependudukan

Beranda

Penduduk

Keluarga

Cari

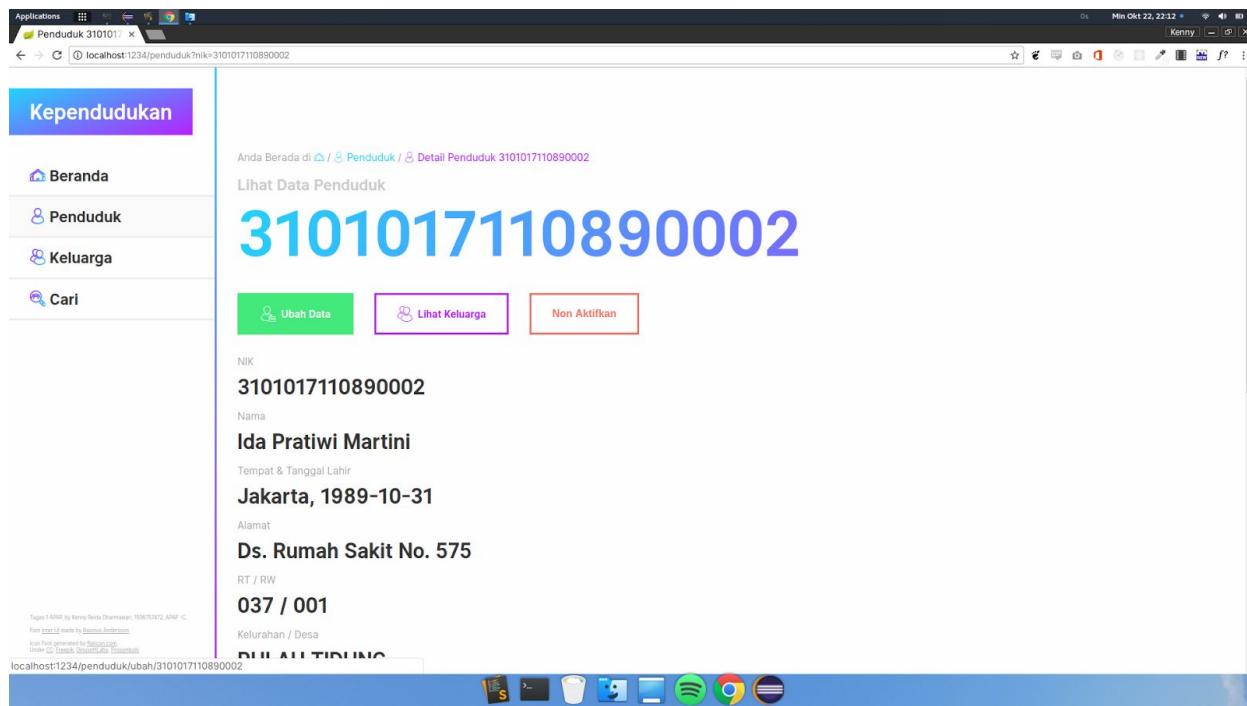
Ubah Data Penduduk

Data Penduduk NIK: 3101017110890002 Berhasil Diubah!

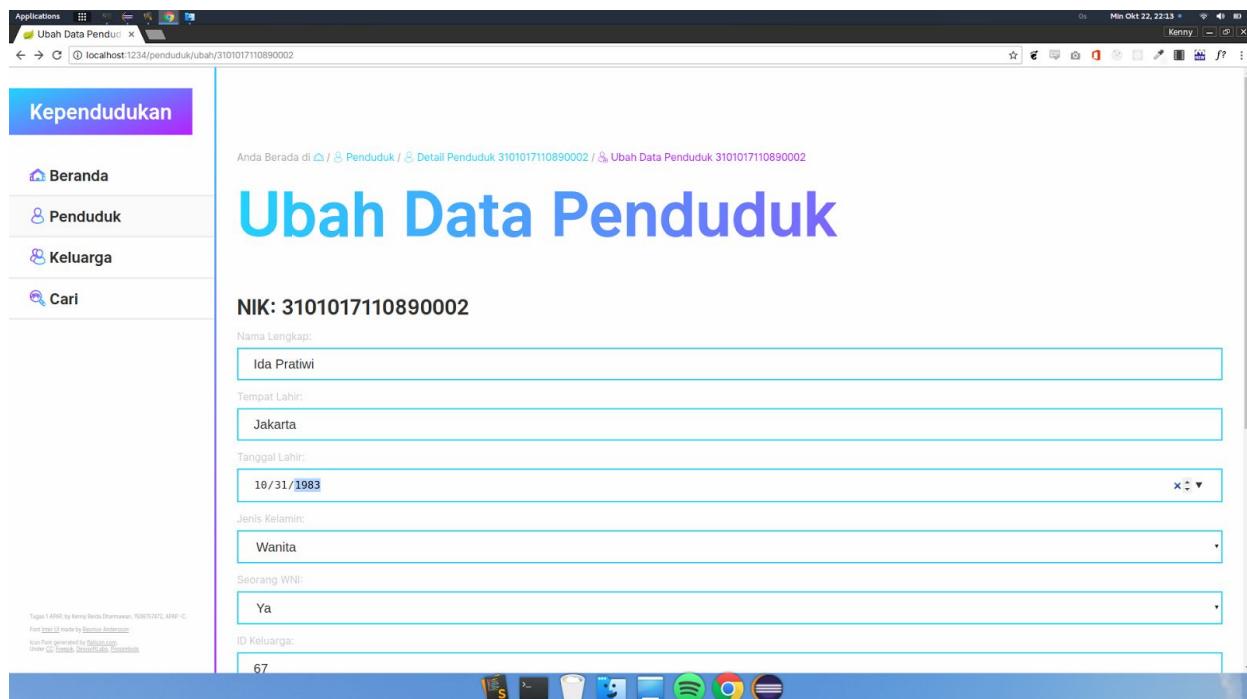
Dengan NIK Sekarang:
3101017110890002

Lihat Data Baru

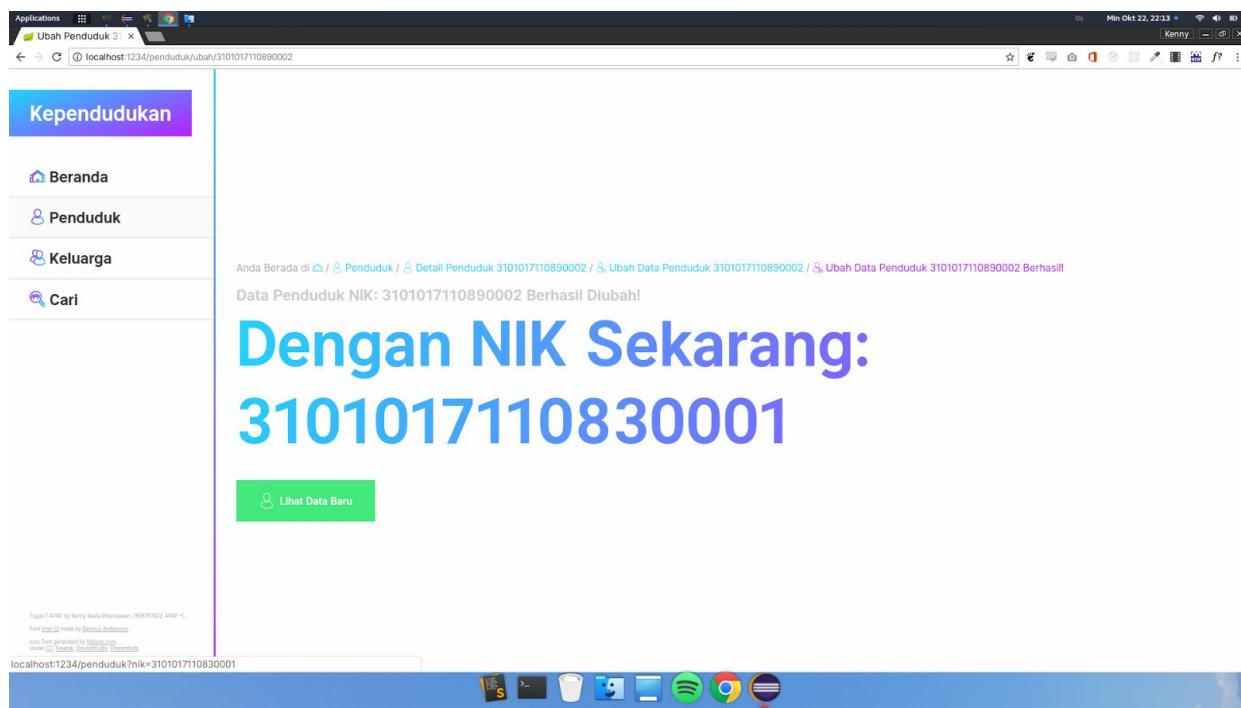
Gambar 5.3 - Sukses Ubah Data Penduduk ('/penduduk/ubah/nikxxxx/') - NIK Sama



Gambar 5.3 - Penduduk ('/penduduk?nik=xxxx') - Data Telah Terubah



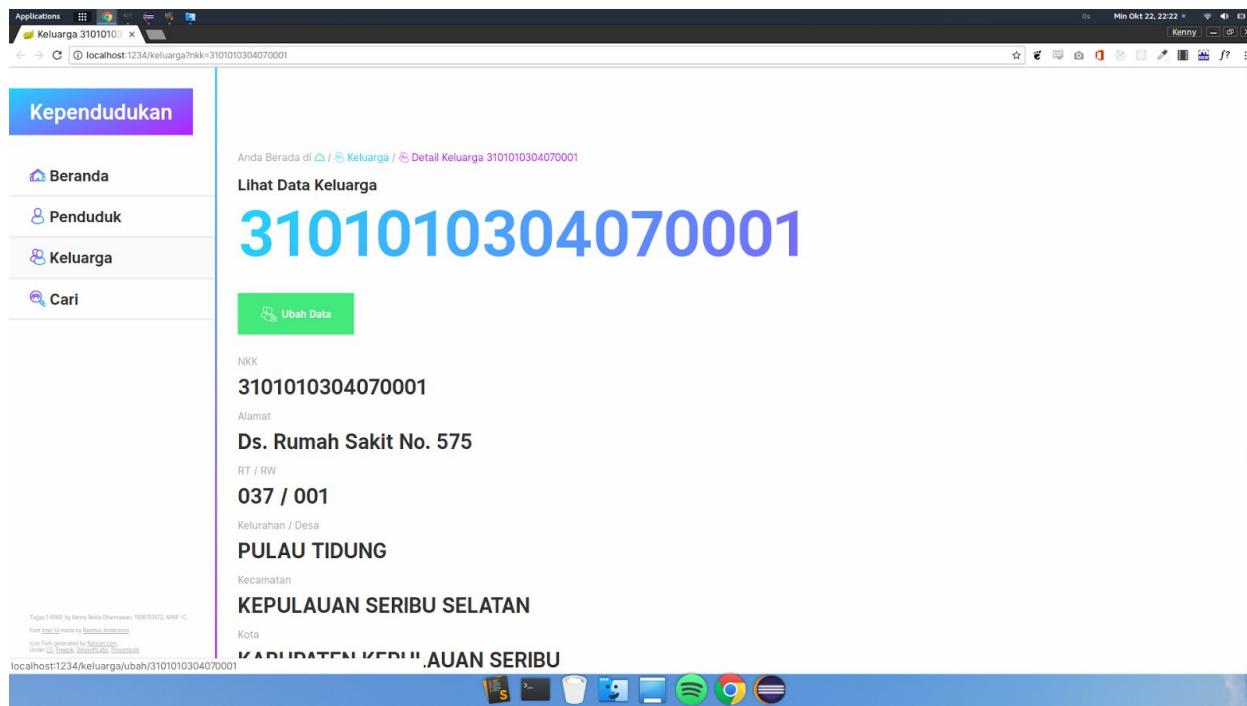
Gambar 5.4 - Form Ubah Data Penduduk ('/penduduk/ubah/nikxxxx') - Ubah Tanggal Lahir



Gambar 5.5 - Sukses Ubah Data Penduduk ('/penduduk/ubah/nikxxxx/') - NIK Baru

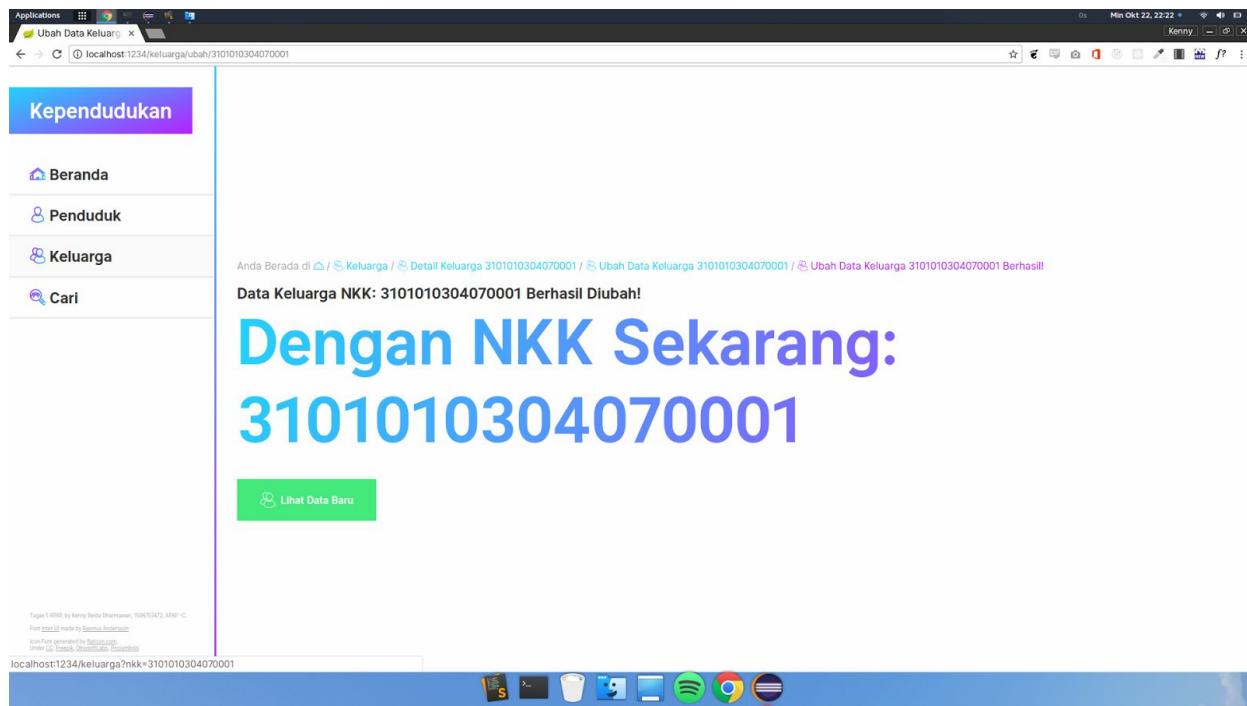
6. Ubah Data Keluarga

Sama seperti fitur 5, pengguna bisa juga merubah data untuk keluarga, dalam merubah data keluarga, pengguna bisa menekan tombol Ubah Data [Gambar 6.1], lalu akan diarahkan ke form ubah data, jika data yang dirubah dan disimpan tidak berhubungan dengan NKK [Gambar 6.2, 6.3, 6.4], maka data akan disimpan langsung tanpa ada perubahan NKK, ketika data yang dirubah terlibat dengan perubahan NKK, maka NKK baru akan dibuat dengan proses seperti pada fitur 4 yaitu fitur pembuatan keluarga baru, akan tetapi, ketika terjadi penggantian NKK, yang berpengaruh hanyalah field kelurahan, maka dari itu ketika NKK dirubah, maka NIK dari setiap anggota keluarga juga harus dirubah, sehingga pada proses ini program juga membuat NIK baru untuk setiap anggota keluarga sesuai prosedur pada tahap 3 yaitu pembuatan penduduk baru. Ketika NKK baru dan NIK baru sudah dibuat, maka data keluarga dan penduduk diupdate dan pengguna akan diberikan halaman Ubah Sukses dengan menampilkan NKK baru dan NKK lama, ketika dilihat data keluarga yang baru, maka terlihat NKK dan semua NIK setiap anggota keluarga juga diperbaharui [Gambar 6.5, 6.6, 6.7. 6.8].

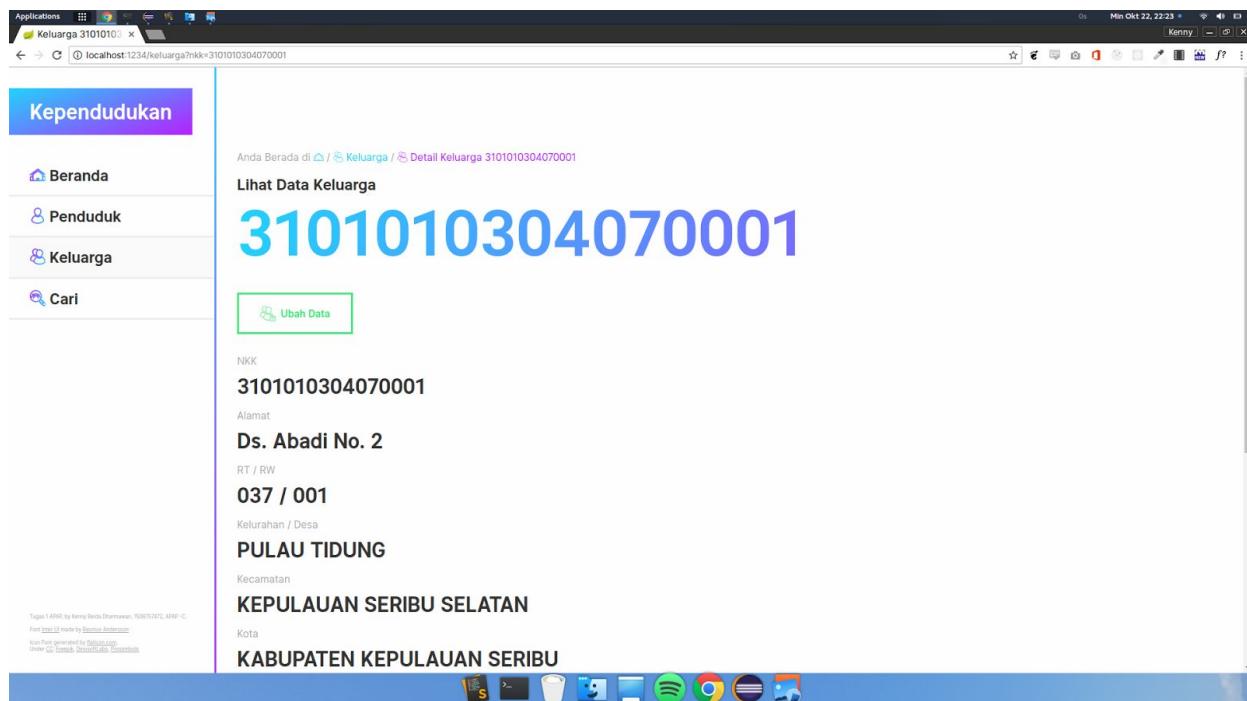


Gambar 6.1 - Keluarga ('/keluarga?nkk=xxxx') - Tombol Tambah Keluarga

Gambar 6.2 - Form Ubah Data Keluarga ('/keluarga/ubah/nkkxxxx') - Ubah Alamat



Gambar 6.3 - Sukses Ubah Data Keluarga ('/keluarga/ubah/nkkxxxx') - NIK Sama



Gambar 6.4 - Keluarga ('/keluarga?nkk=xxxx') - Ubah Alamat Sukses

Kependudukan

Beranda

Penduduk

Keluarga

Cari

NKK: 3101010304070001

Alamat Keluarga

Ds. Abadi No. 2

RT:

037

RW:

001

Kelurahan (Akan Otomatis Memilih Kecamatan & Kota):

TANJUNG BARAT

TAMBAH

Gambar 6.5 - Form Ubah Data Keluarga ('/keluarga/ubah/nkkxxxx/') - Ubah Kelurahan

Kependudukan

Beranda

Penduduk

Keluarga

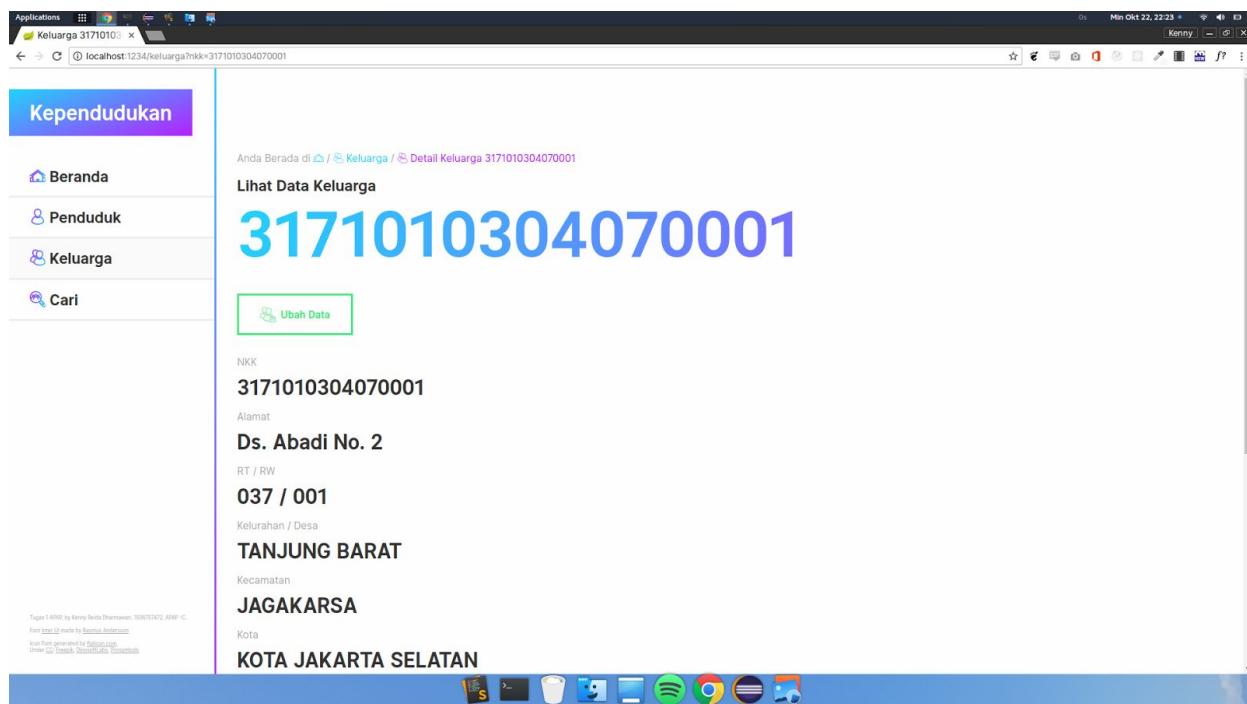
Cari

Data Keluarga NKK: 3101010304070001 Berhasil Diubah!

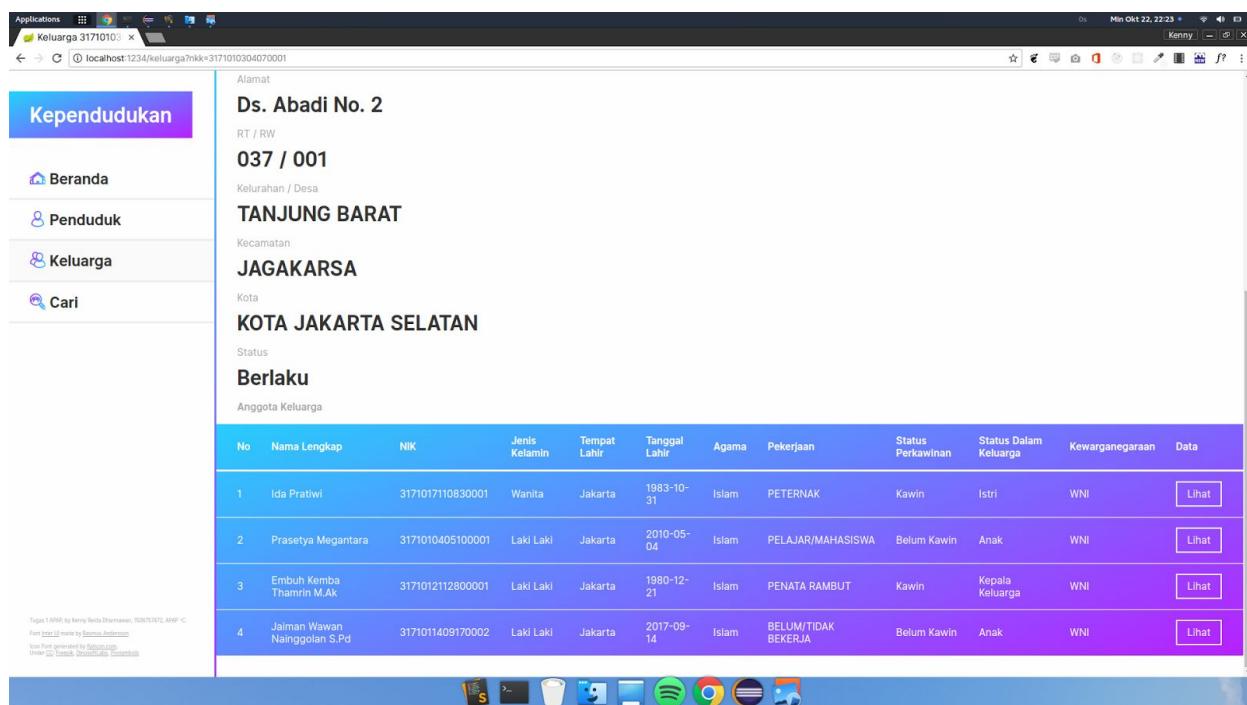
**Dengan NKK Sekarang:
3171010304070001**

Lihat Data Baru

Gambar 6.6 - Form Ubah Data Keluarga ('/keluarga/ubah/nkkxxxx/') - NIK Baru



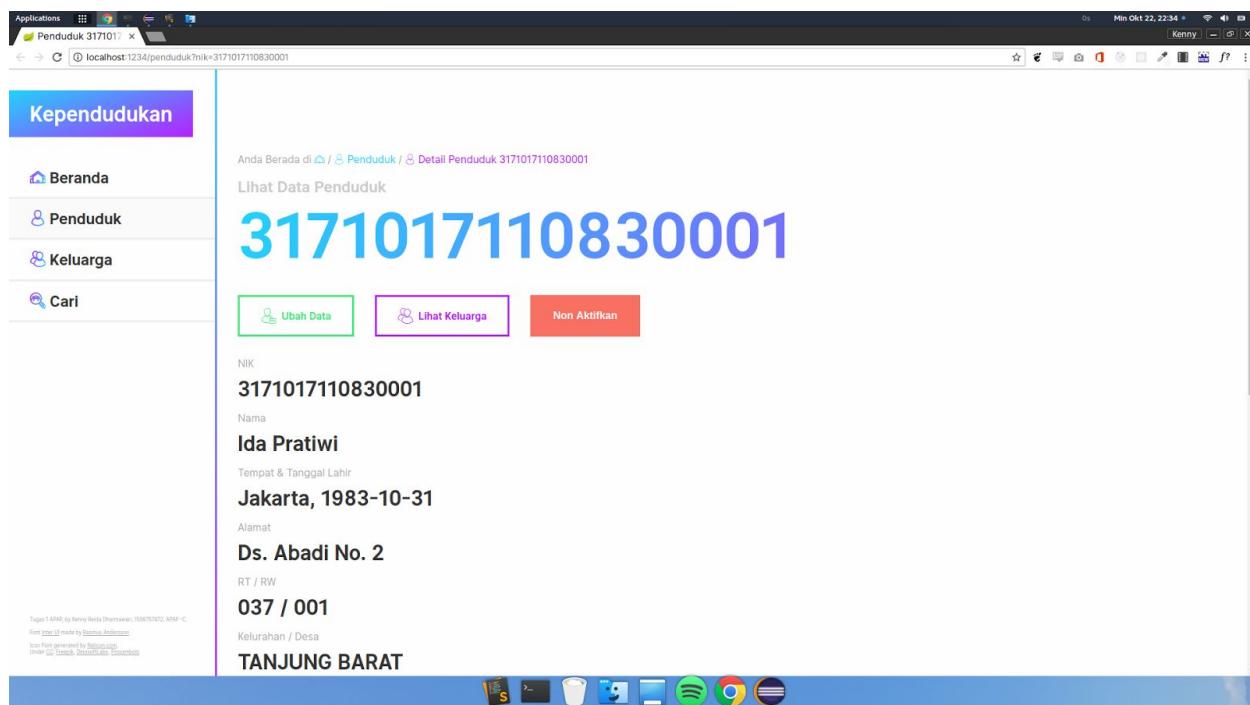
Gambar 6.7 - Keluarga ('/keluarga?nkk=xxxx') - Ubah Kelurahan Sukses & NKK Baru



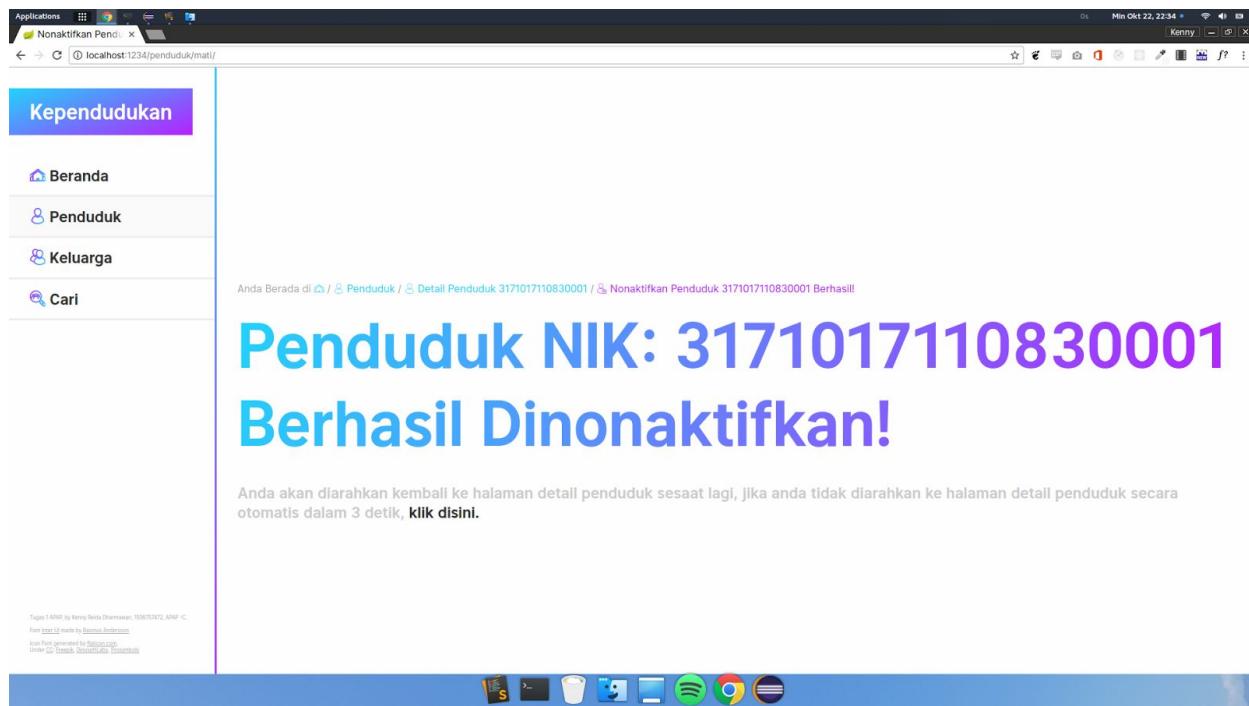
Gambar 6.8 - Keluarga ('/keluarga?nkk=xxxx') - Ubah Kelurahan juga Mengubah NIK Setiap Anggota Keluarga

7. Mengubah Status Kematian Penduduk

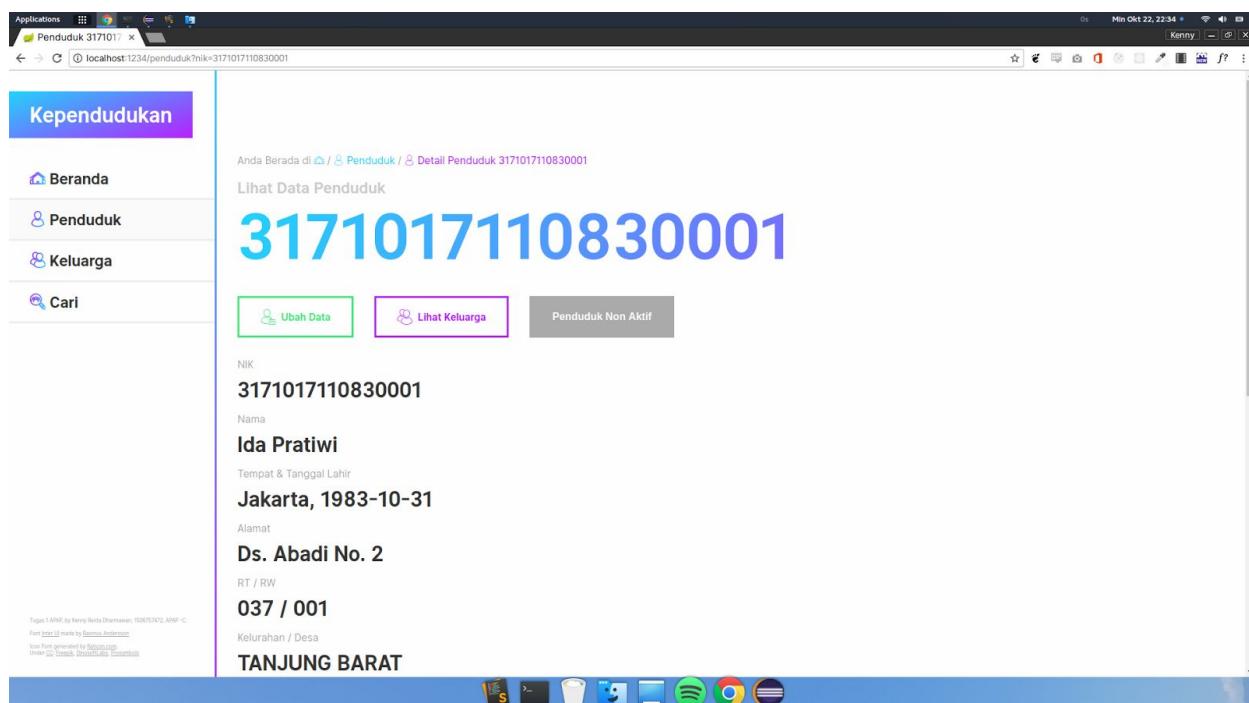
Pada fitur ke 7, pengguna bisa mengganti status kematian penduduk dengan cara menekan tombol Non-Aktifkan pada halaman Lihat Penduduk Berdasarkan NIK [Gambar 7.1], setelah menekan tombol non-aktifkan maka pengguna akan diarahkan ke halaman konfirmasi bahwa pengguna telah di non-aktifkan, dan akan di arahkan ke halaman Lihat Penduduk Berdasarkan NIK secara otomatis setelah 3 detik atau jika pengguna menekan tombol Klik Disini [Gambar 7.2, 7.3], jika semua anggota keluarga sudah wafat maka status keluarga juga dijadikan tidak aktif secara otomatis [Gambar 7.4]

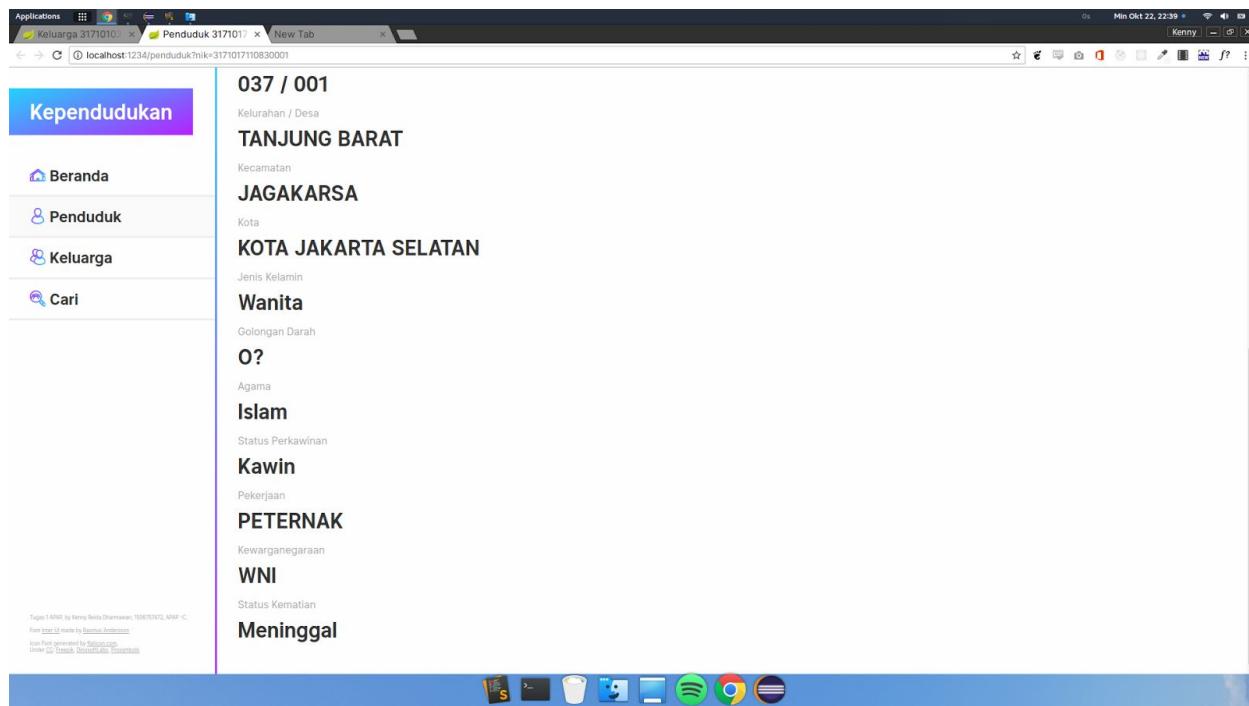


Gambar 7.1 - Lihat Penduduk Berdasarkan NIK ('/penduduk?nik=xxxx') - Klik Tombol Non Aktifkan

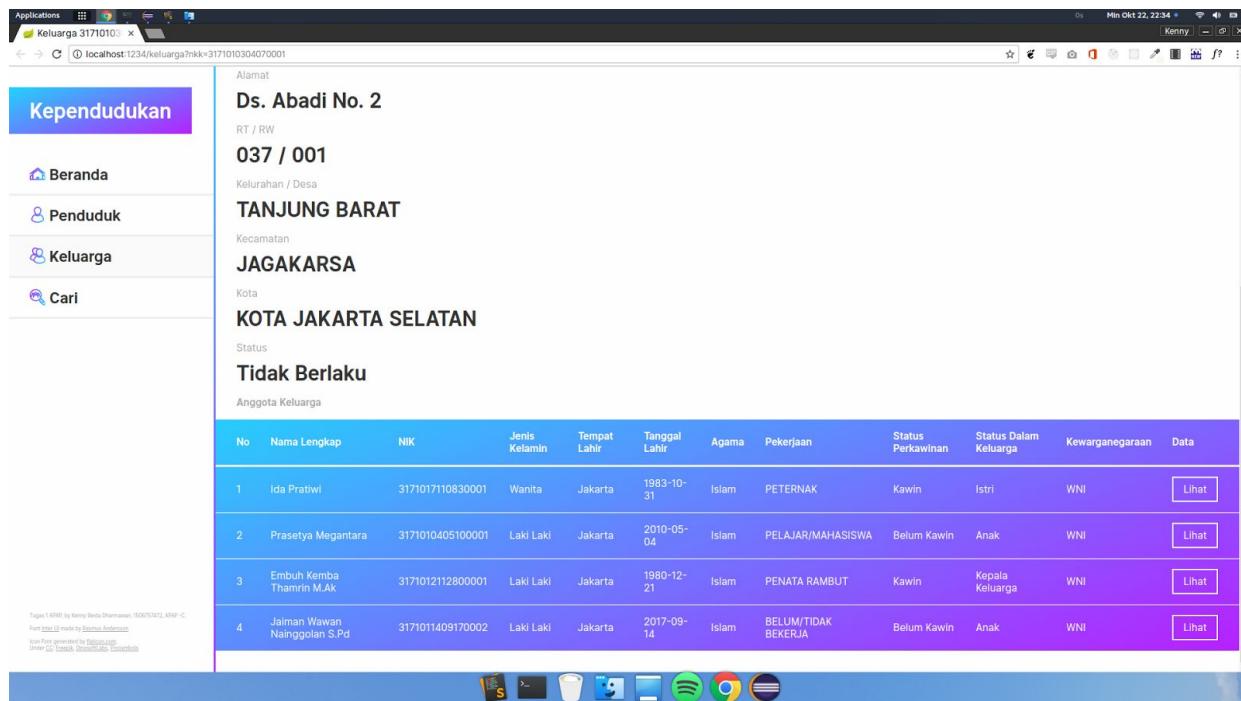


Gambar 7.2 - Sukses Set Penduduk Menjadi Wafat ('/penduduk/mati/')





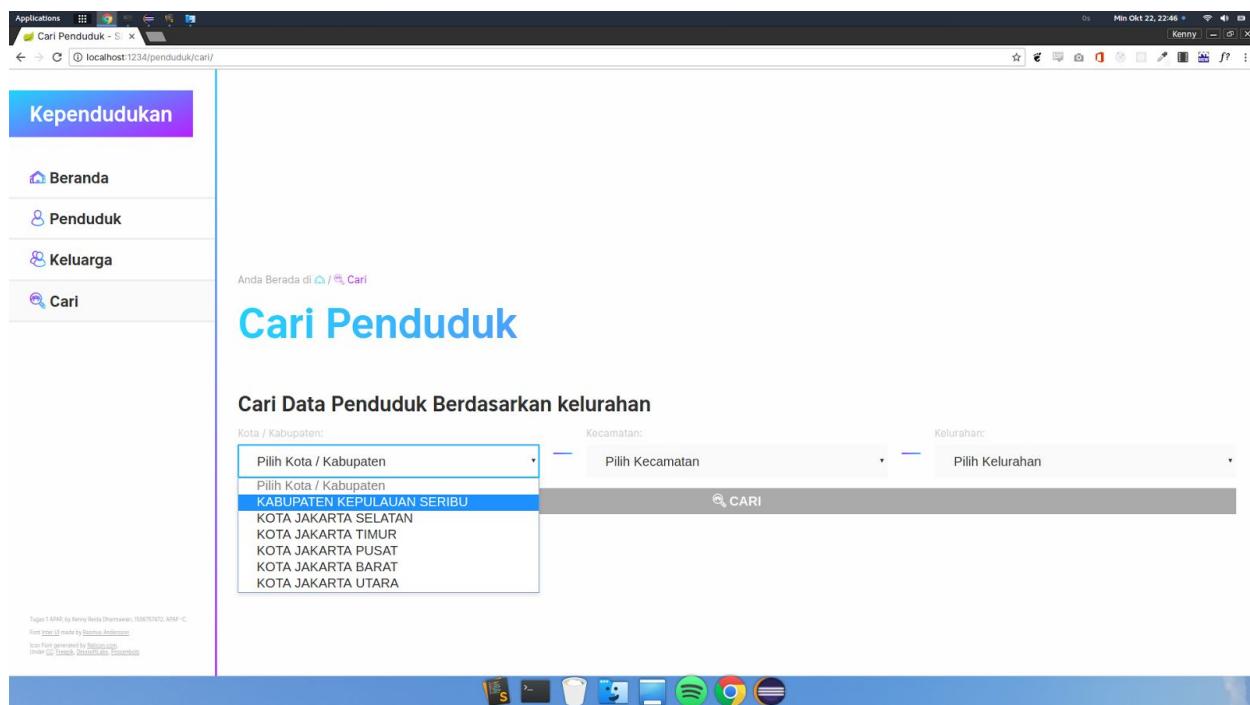
Gambar 7.3 A & B - Lihat Penduduk Berdasarkan NIK ('/penduduk?nik=xxxx') - Penduduk Non-Aktif, tombol terdisabled, status Meninggal



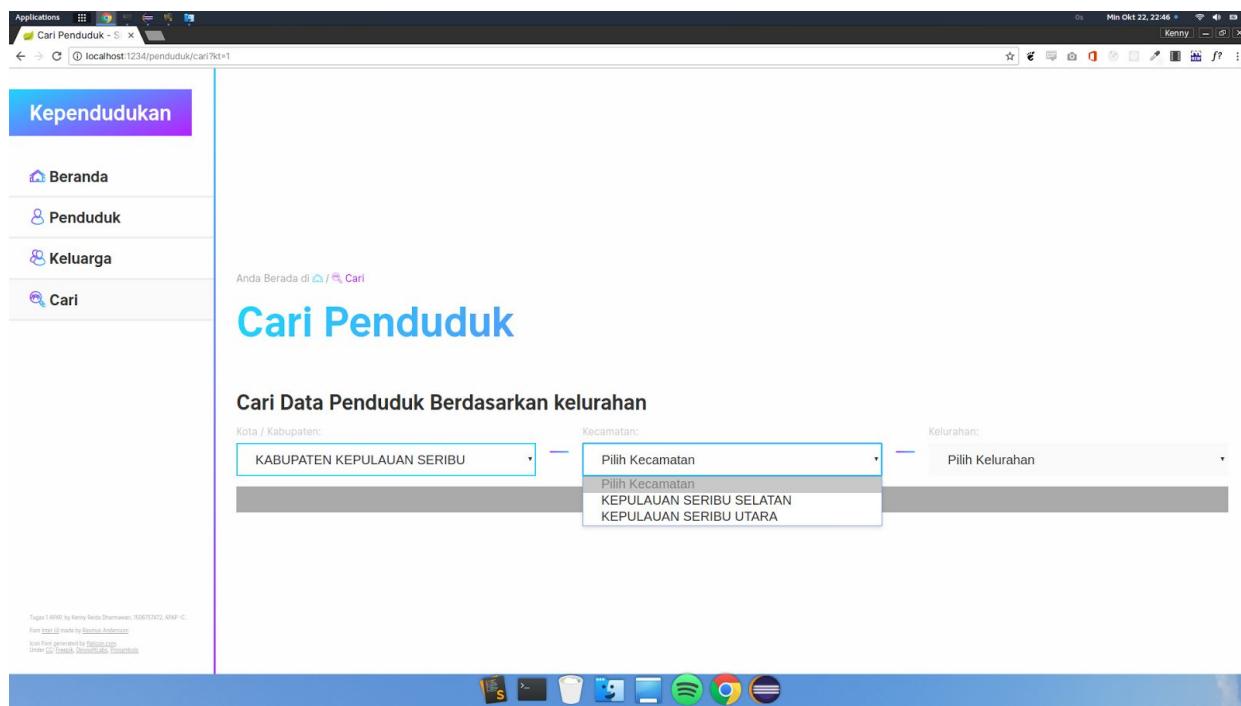
Gambar 7.4 - Lihat Keluarga Berdasarkan NKK ('/keluarga?nkk=xxxx') - Status Keluarga Tidak Berlaku Ketika semua Anggota telah Meninggal

8. Tampilkan Penduduk Berdasarkan Kota-Kecamatan-Kelurahan Tertentu

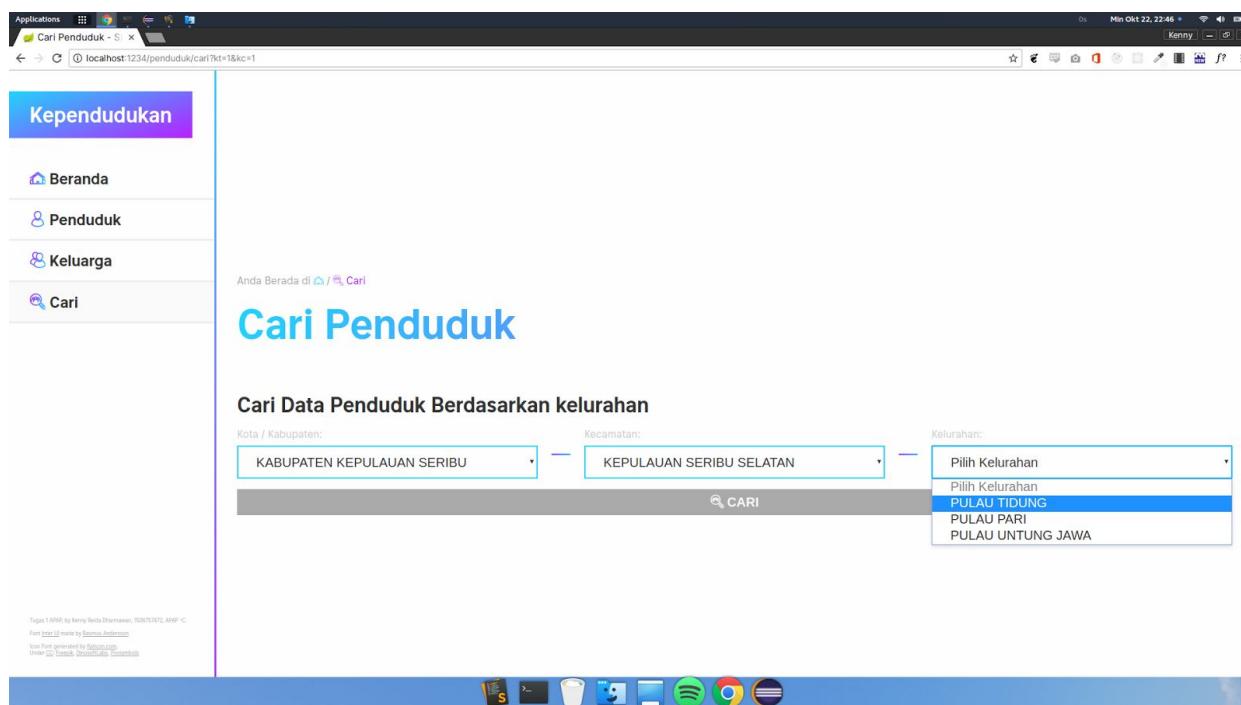
Pada fitur ke 8, pengguna dapat membuka halaman Cari ('/cari') [Gambar 8.1] dan memilih kota, setelah kota dipilih maka pengguna akan otomatis diarahkan ke Halaman Cari Kecamatan ('/cari?kt=idKota') [Gambar 8.2], dimana pengguna bisa memilih kecamatan dan otomatis diarahkan ke halaman Cari Kelurahan ('/cari?kt=idKota&kc=idKec') [Gambar 8.3], lalu setelah memilih kelurahan maka tombol cari dapat ditekan [Gambar 8.4], dimana setelah ditekan maka pengguna akan diarahkan ke halaman hasil pencarian ('/cari?kt=idKota&kc=idKec&kl=idKel') [Gambar 8.5]. Disini saya melakukan query yang mencari setiap keluarga yang ada pada kelurahan yang dipilih dan juga mendapatkan semua anggota keluarga dari daftar keluarga tersebut dan menampilkannya sebagai tabel yang kemudian diberi pagination oleh library **DataTables**.



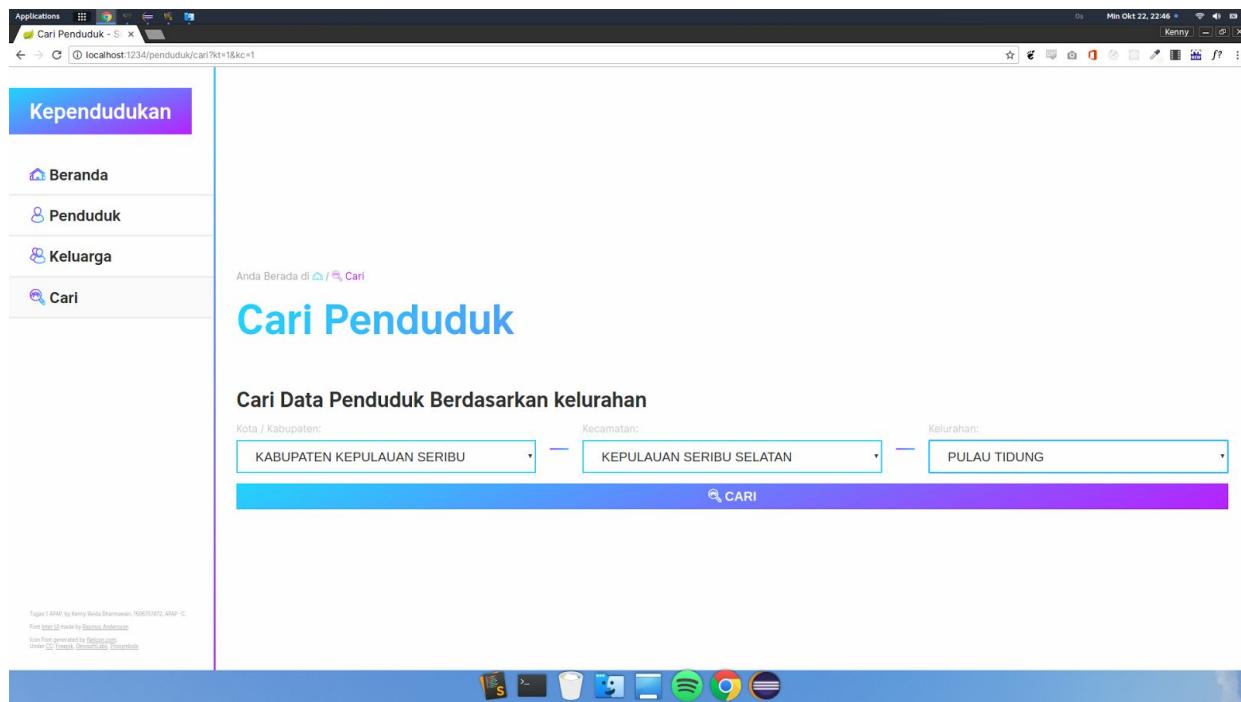
Gambar 8.1 - Cari ('/cari')



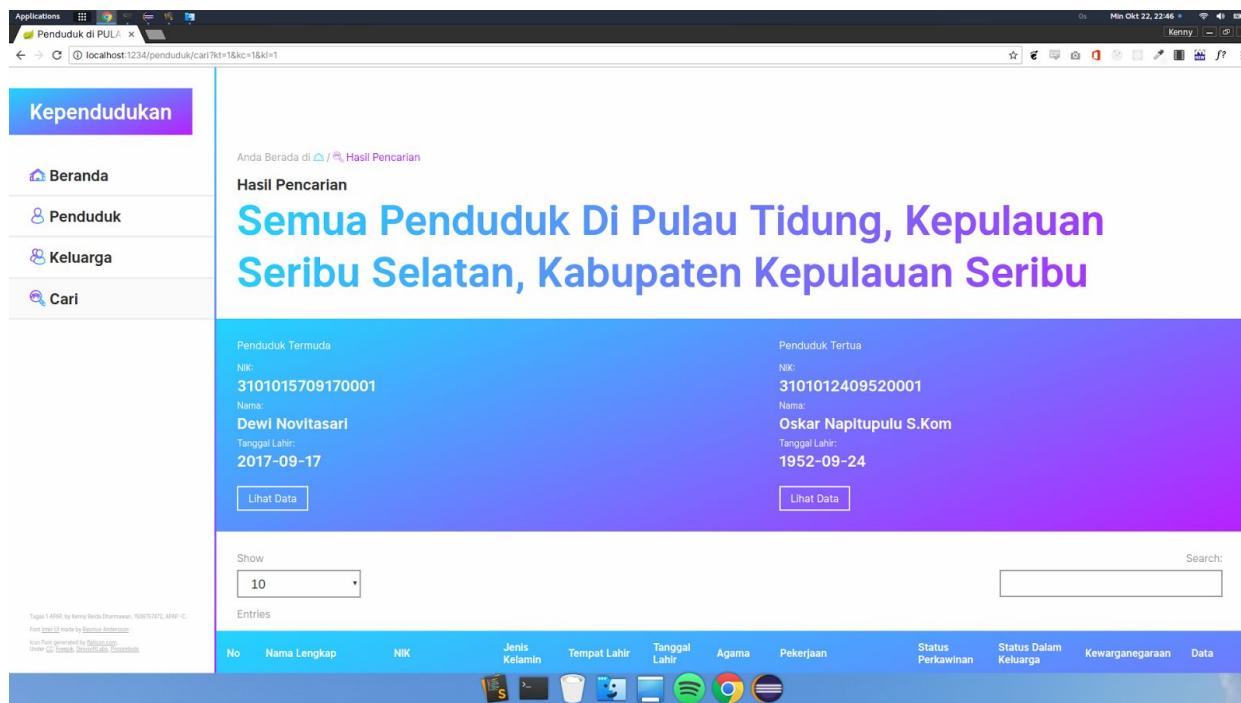
Gambar 8.2 - Cari & Pilih Kecamatan



Gambar 8.3 - Cari & Pilih Kelurahan



Gambar 8.4 - Tombol Cari bisa di klik



No	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga	Kewarganegaraan	Data
1	Darjan Prabowo	3101011508890001	Laki Laki	Jakarta	1988-08-15	Katholik	PENELITI	Kawin	Kepala Keluarga	WNI	<button>Lihat</button>
2	Vicky Puspasari	3101014704860001	Wanita	Madiun	1986-04-07	Katholik	PETERNAK	Kawin	Istri	WNI	<button>Lihat</button>
3	Zalindra Laksita S.Psi	3101016002140001	Wanita	Jakarta	2014-02-20	Katholik	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI	<button>Lihat</button>
4	Paulin Cornelia Usamah	3101015009690001	Wanita	Padang Sidempuan	1969-09-10	Islam	ANGGOTA MAHKAMAH KONSTITUSI	Belum Kawin	Famili Lain	WNI	<button>Lihat</button>
5	Unjani Latika Hariyah S.Ked	3101015802160001	Wanita	Lubuklinggau	2016-02-18	Katholik	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI	<button>Lihat</button>
6	Kairav Reksa Hidayanto M.Kom.	3101010302150003	Laki Laki	Jakarta	2015-02-03	Katholik	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI	<button>Lihat</button>
7	Lega Iswahyudi S.I.P	3101010506560001	Laki Laki	Sawah Lunto	1956-06-05	Islam	PEMBANTU RUMAH TANGGA	Cerai Hidup	Pembantu	WNI	<button>Lihat</button>
8	Heru Ardianto	3101011402860001	Laki Laki	Jakarta	1986-02-14	Islam	KONSTRUKSI	Kawin	Kepala Keluarga	WNI	<button>Lihat</button>
9	Putri Lestari	3101016412820001	Wanita	Jakarta	1982-12-24	Islam	PEMBANTU RUMAH TANGGA	Belum Kawin	Pembantu	WNI	<button>Lihat</button>
10	Uchita Mayasari	3101014310850001	Wanita	Jakarta	1985-10-03	Islam	BIDAN	Kawin	Istri	WNI	<button>Lihat</button>

Showing 1 to 10 of 1,151 entries

Previous 1 2 3 4 5 ... 116 Next

Gambar 8.5 A & B - Hasil Pencarian

9. Validasi Data Untuk Semua Form POST

Pada fitur ini, saya telah melakukan validasi pada sisi backend dibagian Model dan Controller, yang dimana ketika terjadi data yang tidak valid, maka user akan dikembalikan ke halaman isi data dengan notifikasi field apa saja yang error seperti gambar dibawah ini [Gambar 9.1]. Saya juga telah memberikan halaman ketika data penduduk, keluarga, atau hasil pencarian tidak ada [Gambar 9.2]

Applications Min Okt 22, 22:52 Kenny

Tambah Penduduk x localhost:1234/penduduk/tambah

Kependudukan

- [Beranda](#)
- [Penduduk](#)
- [Keluarga](#)
- [Cari](#)

Anda Berada di [Penduduk](#) / [Tambah Penduduk](#)

Tambah Penduduk

Masukkan Data Penduduk

Nama Lengkap:

Error: size must be between 1 and 128

Tempat Lahir:

Error: size must be between 1 and 128

Tanggal Lahir:

Error: size must be between 1 and 128

Jenis Kelamin:

Seorang WNI:

ID Keluarga:

Tugas 1 APAP by Kenny Reida Dharmawan, TOSN07002, APAP-C.
Form generated by [Formularis](#).
Icon Font generated by [Fontello](#).
Icons by [Iconshock](#).

Applications Min Okt 22, 22:52 Kenny

Ubah Data Penduduk x localhost:1234/penduduk/ubah/3171034306930001

Kependudukan

- [Beranda](#)
- [Penduduk](#)
- [Keluarga](#)
- [Cari](#)

Anda Berada di [Penduduk](#) / [Detail Penduduk 3171034306930001](#) / [Ubah Data Penduduk 3171034306930001](#)

Ubah Data Penduduk

NIK: 3171034306930001

Nama Lengkap:

Error: size must be between 1 and 128

Tempat Lahir:

Tanggal Lahir:

Jenis Kelamin:

Seorang WNI:

ID Keluarga:

Tugas 1 APAP by Kenny Reida Dharmawan, TOSN07002, APAP-C.
Form generated by [Formularis](#).
Icon Font generated by [Fontello](#).
Icons by [Iconshock](#).

Kependudukan

Beranda

Penduduk

Keluarga

Cari

Anda Berada di / Keluarga / Tambah Keluarga

Tambah Keluarga

Masukkan Data Keluarga

Alamat Keluarga

Masukkan Alamat Keluarga

Error: size must be between 1 and 256

RT:

Masukkan RT Alamat Keluarga

Error: size must be between 3 and 3

RW:

Masukkan RW Alamat Keluarga

Error: size must be between 3 and 3

Kelurahan (Akan Otomatis Memilih Kecamatan & Kota):

PULAU TIDUNG

Error: may not be null

TAMBAH

Kependudukan

Beranda

Penduduk

Keluarga

Cari

Anda Berada di / Keluarga / Detail Keluarga 3101010203000001 / Ubah Data Keluarga 3101010203000001

Ubah Data Keluarga

NKK: 3101010203000001

Alamat Keluarga

Masukkan Alamat Keluarga

Error: size must be between 1 and 256

RT:

097

RW:

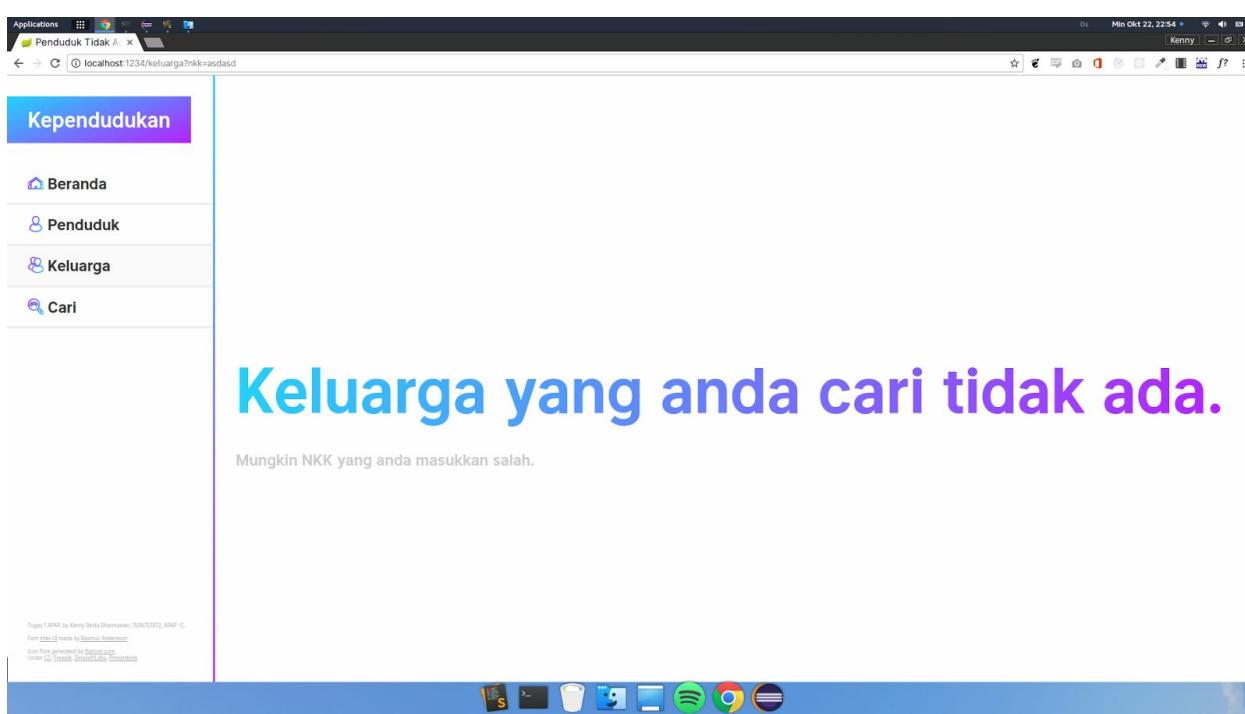
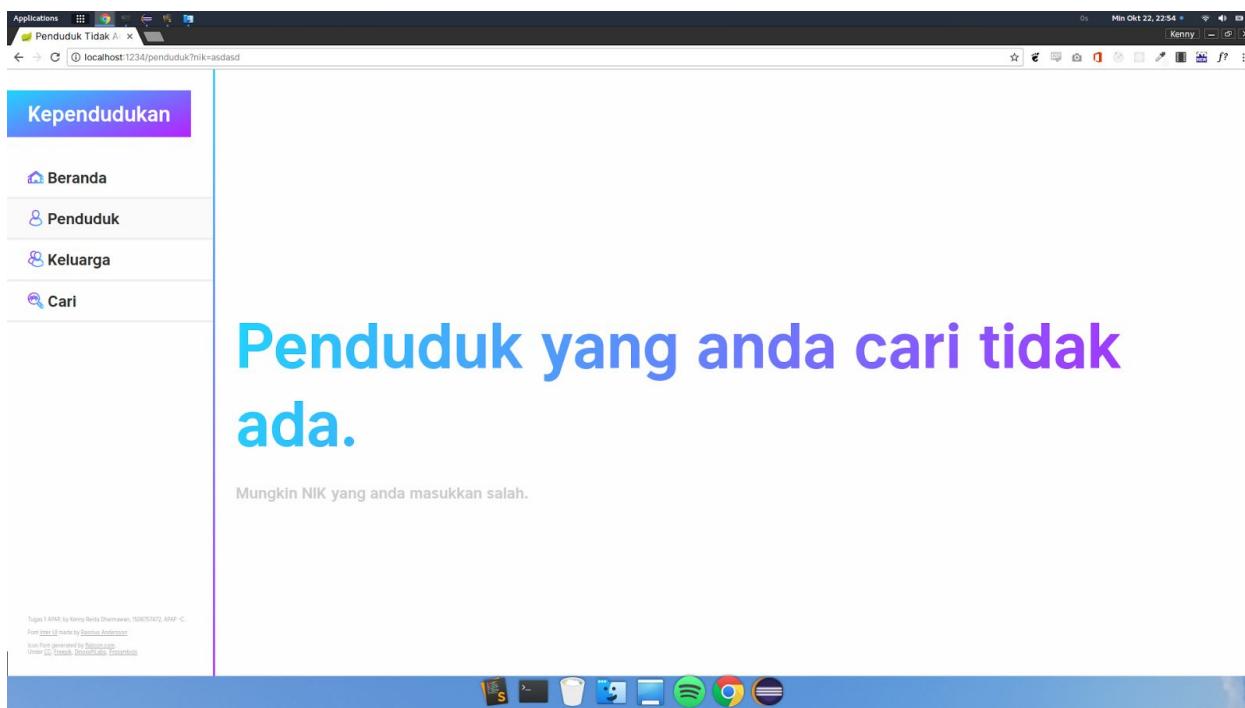
043

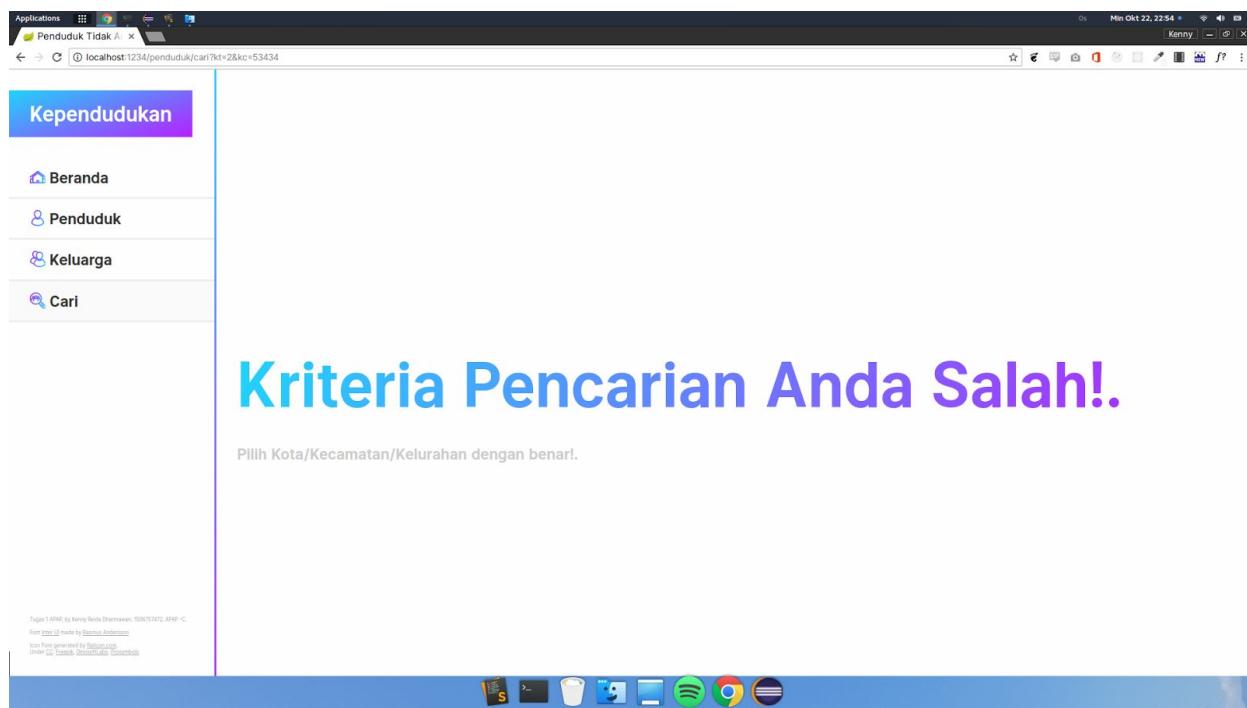
Kelurahan (Akan Otomatis Memilih Kecamatan & Kota):

PULAU UNTUNG JAWA

TAMBAH

Gambar 9.1 A, B, C, & D - Validasi pada semua Form Post





Gambar 9.2 A, B, & C - Halaman Ketika Parameter NIK/NKK/Pencarian Salah.

10. Penduduk Paling Tua & Paling Muda

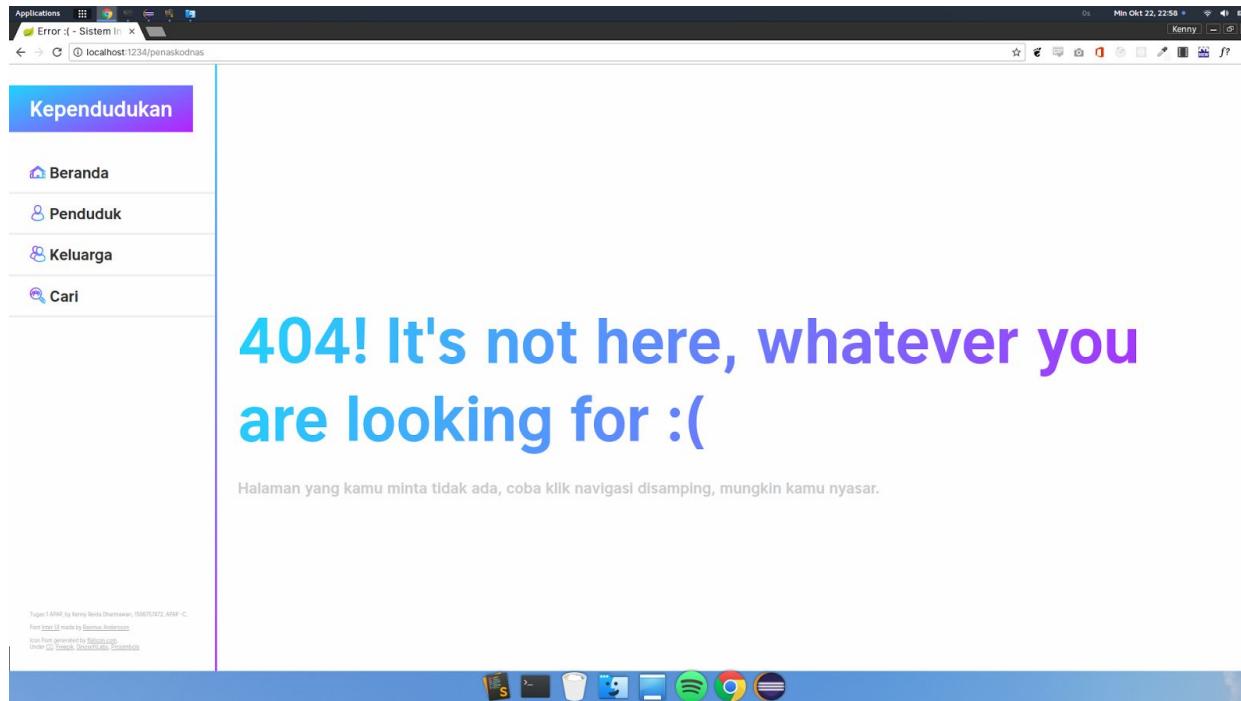
Pada fitur ini, saya juga telah menampilkan penduduk yang paling tua dan paling muda dalam suatu pencarian, dapat dilihat pada [Gambar 10] Dibawah Ini.

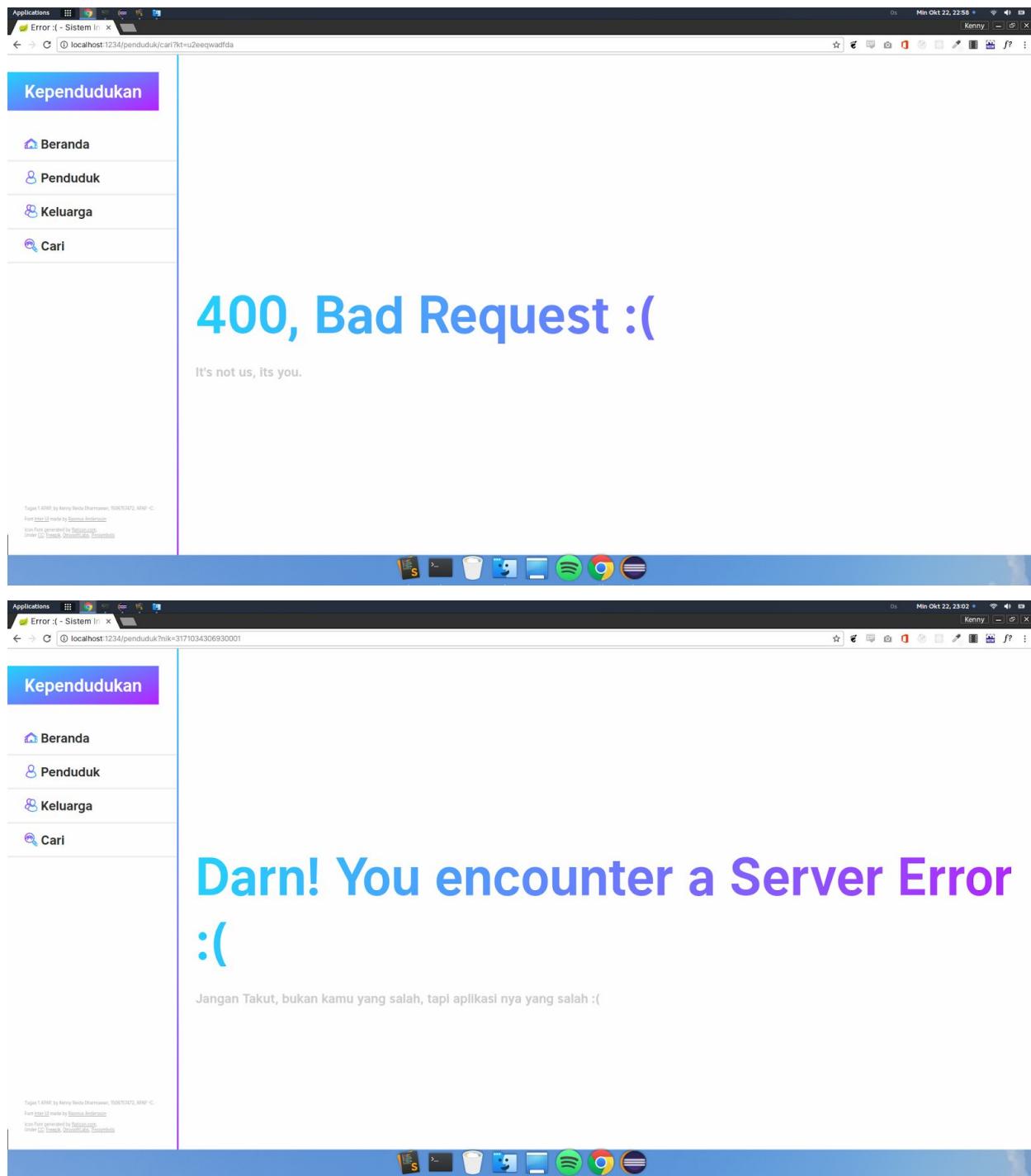


Gambar 10 - Penduduk Paling Muda & Paling Tua

11. Halaman Error

Pada fitur ini, saya juga telah menampilkan halaman khusus ketika terjadi error, seperti pada [Gambar 11] dibawah ini.





Gambar 11 - Custom 404, 400, dan 5xx Error Pages.

12. Fitur Tambahan

Pada Aplikasi ini, saya menambahkan navigasi khusus yang berguna untuk menavigasi antar halaman, saya menambahkan shortcut tombol tambah pada setiap halaman penduduk dan keluarga, tombol ubah data pada setiap detail data penduduk dan keluarga, breadcrumbs pada setiap halaman untuk membantu navigasi, dapat dilihat pada gambar gambar fitur-fitur sebelumnya.

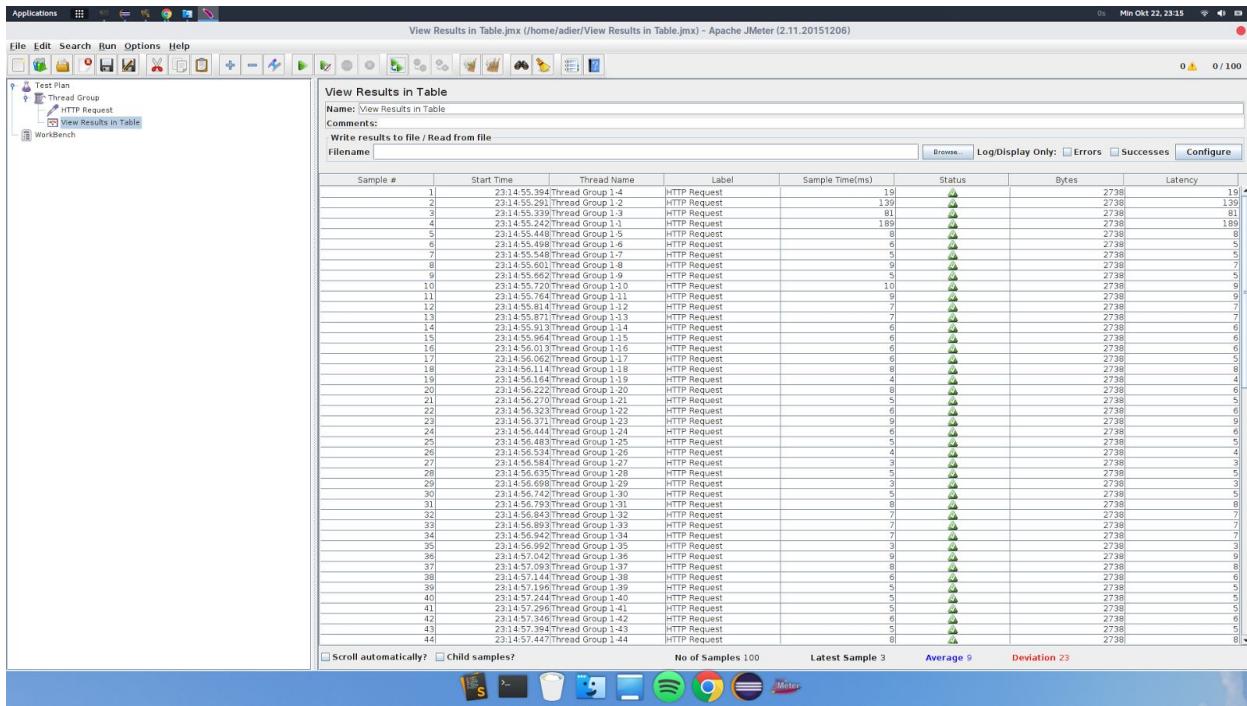
Optimasi Database

Untuk membuat database optimal dan dapat digunakan dengan cepat, saya menggunakan teknik indexing untuk beberapa kolom yang biasa digunakan untuk query, saya menambahkan primary key untuk setiap kolom *id* pada masing masing tabel, pada tabel penduduk saya menambahkan indexing pada kolom *nik* & *id_keluarga* karena kedua kolom tersebut sering digunakan untuk mencari data penduduk, pada tabel keluarga saya menambahkan indexing pada kolom *nomor_kk* & *id_kelurahan* karena sering digunakan untuk mencari data keluarga, pada kelurahan, kecamatan, dan kota saya tidak menambahkan indexing khusus karena kolom *id* di setiap table sudah saya atur sebagai primary key dan otomatis memiliki *indexing* tersendiri.

Stress Testing

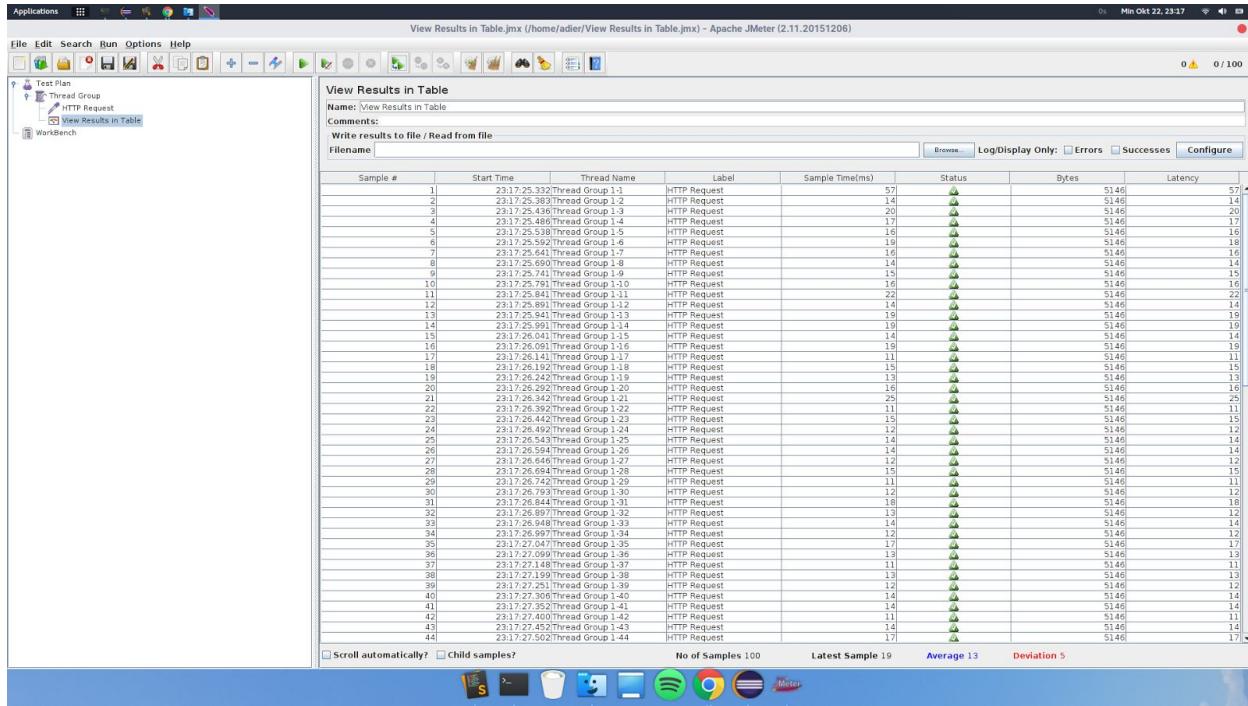
Ramp Up: 5s - Thread: 100

1. Akses Index



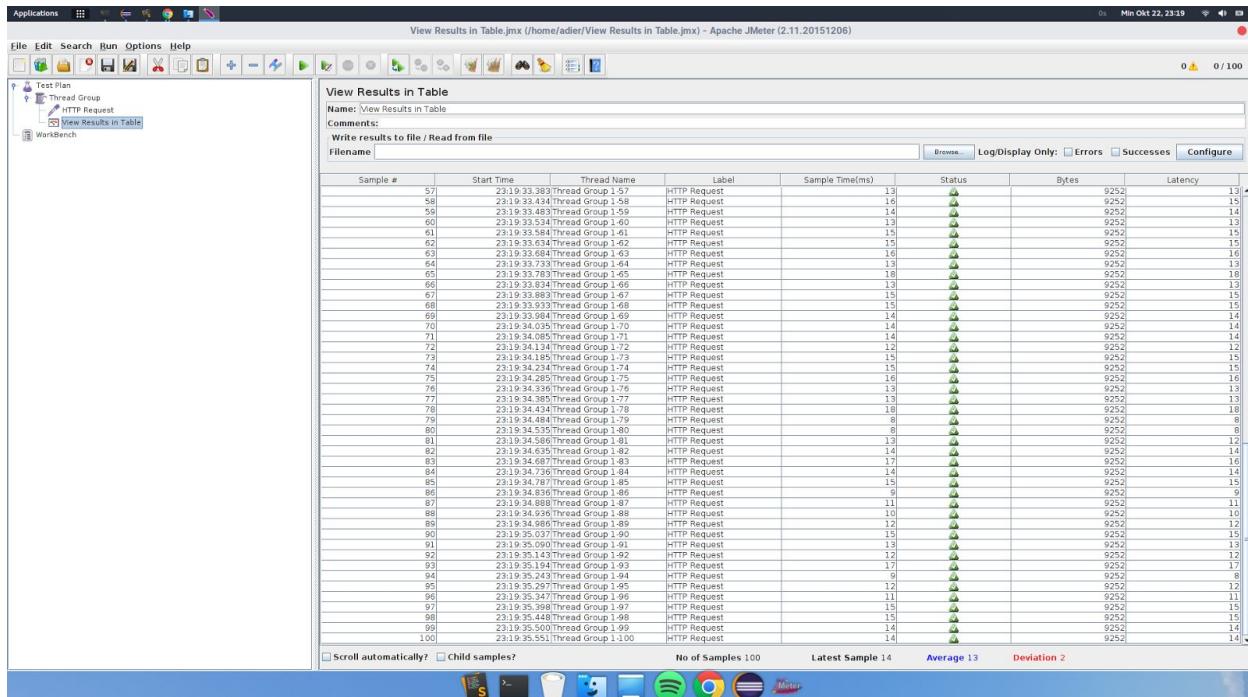
Terlihat Akses Index sangatlah cepat, dengan 100 thread dapat mengakses dalam waktu rata rata 9 Milisecond.

2. Akses Lihat Penduduk Berdasarkan NIK



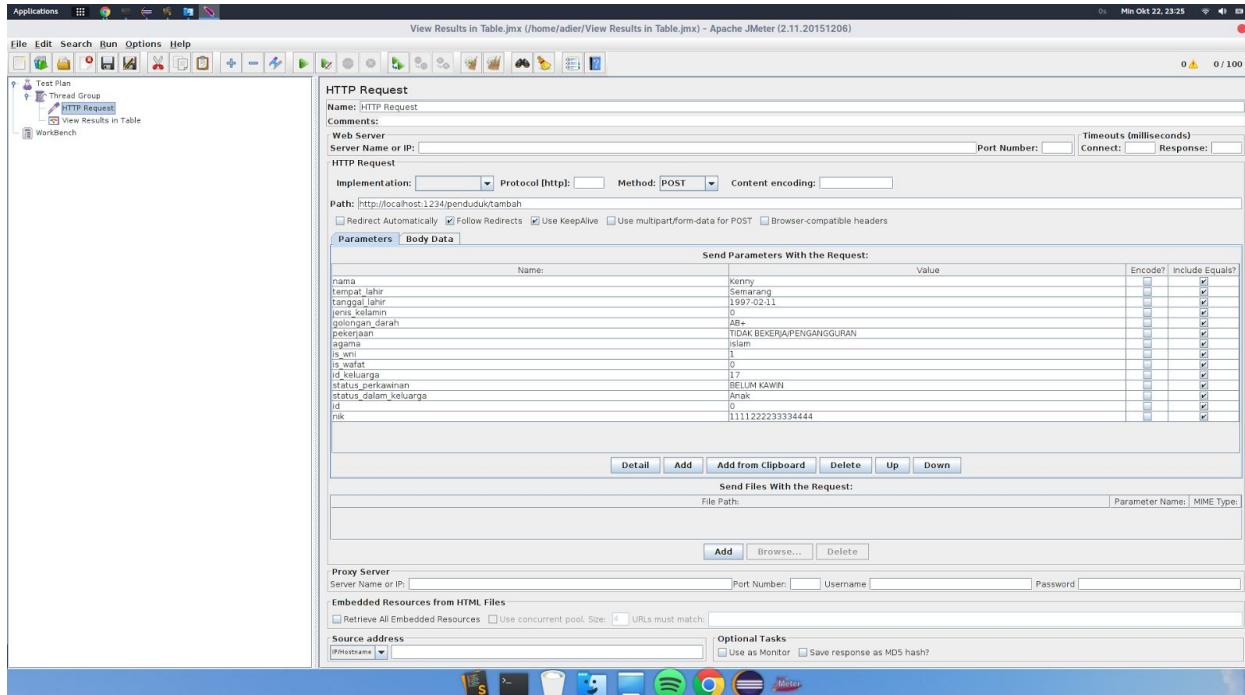
Terlihat akses melihat penduduk berdasarkan NIK juga sangatlah cepat, dengan 100 thread mengakses dapat diakses dalam waktu rata rata 13 Milisecond.

3. Akses Lihat Keluarga Berdasarkan NKK



Terlihat Akses melihat penduduk berdasarkan NKK juga sama cepatnya dengan fitur kedua cepat, dengan 100 thread mengakses dapat diakses dalam waktu rata rata 13 Milisecond.

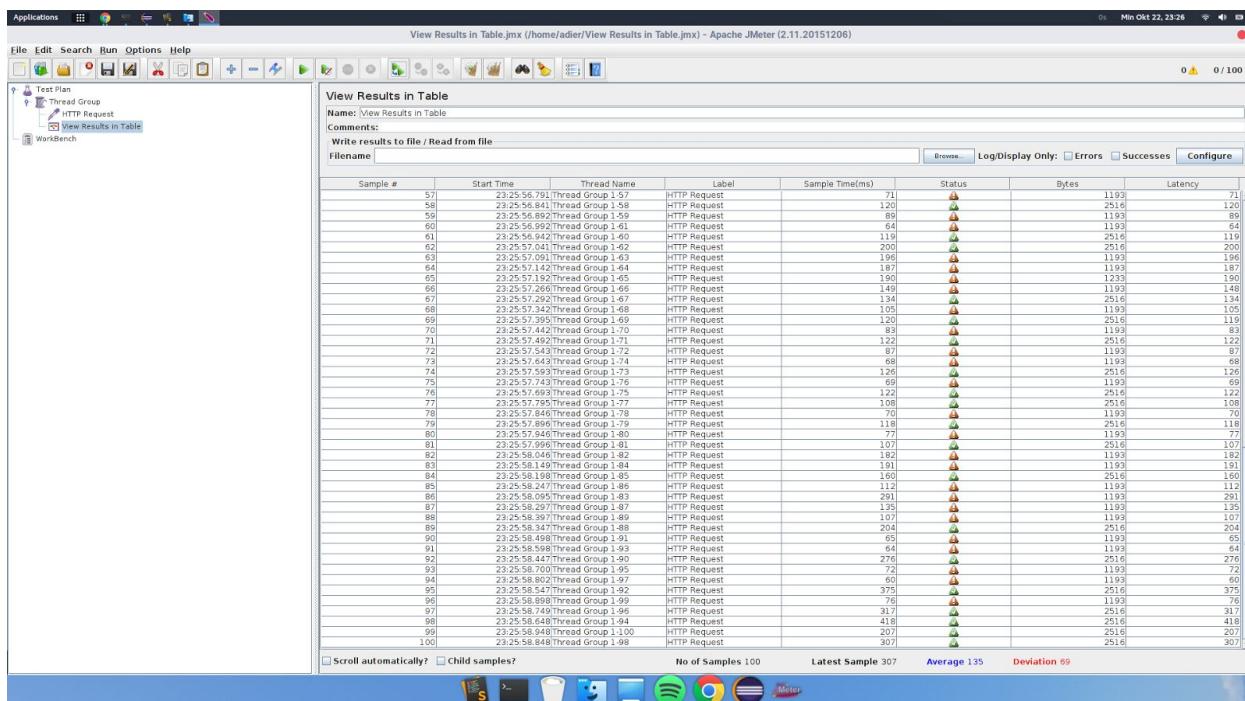
4. Akses Tambah Penduduk



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

- Test Plan:** Contains a Thread Group and an HTTP Request sampler.
- HTTP Request Sampler Configuration:**
 - Name: HTTP Request
 - Protocol: http://localhost:1234/penduduk/tambah
 - Method: POST
 - Content encoding: (empty)
 - Parameters (Table):

Name	Value	Encode?	Include Equals?
nama	Kenny		
tempat_lahir	Semarang		
tanggal_lahir	1997-02-11		
jenis_kelamin	0		
golongan_darah	A+		
pasangan	TIDAK BEKERJA/PENGANGGURAN		
agama	islam		
is_wanita	1		
is_wafat	0		
id_dalam_keluarga	17		
status_perkawinan	BELUM KAWIN		
status_dalam_keluarga	Anak		
id	0		
nik	1111222233344444		
 - Send Parameters With the Request:
 - Send Files With the Request:
- Proxy Server:** Server Name or IP: (empty), Port Number: (empty), Username: (empty), Password: (empty).
- Embedded Resources from HTML Files:** Retrieve All Embedded Resources, Use concurrent pool. Size: 4, URLs must match: (empty).
- Source address:** Use as Monitor, Save response as MD5 hash.



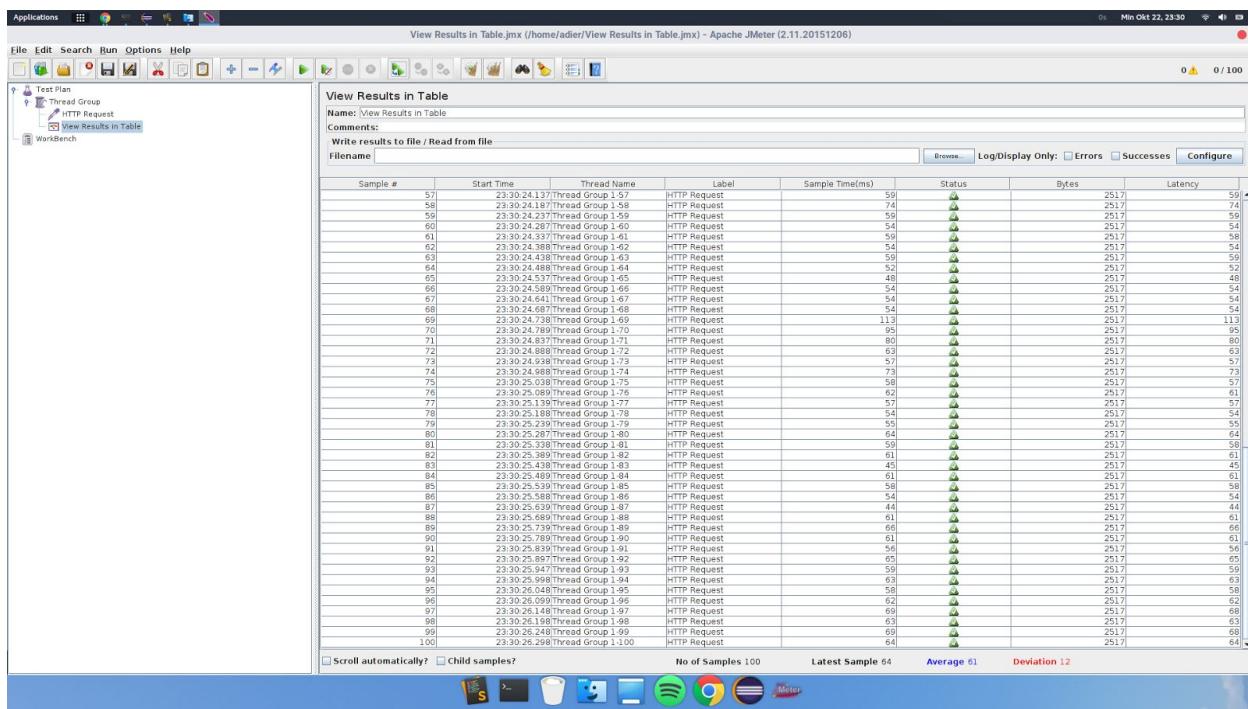
The screenshot shows the Apache JMeter interface with the following details:

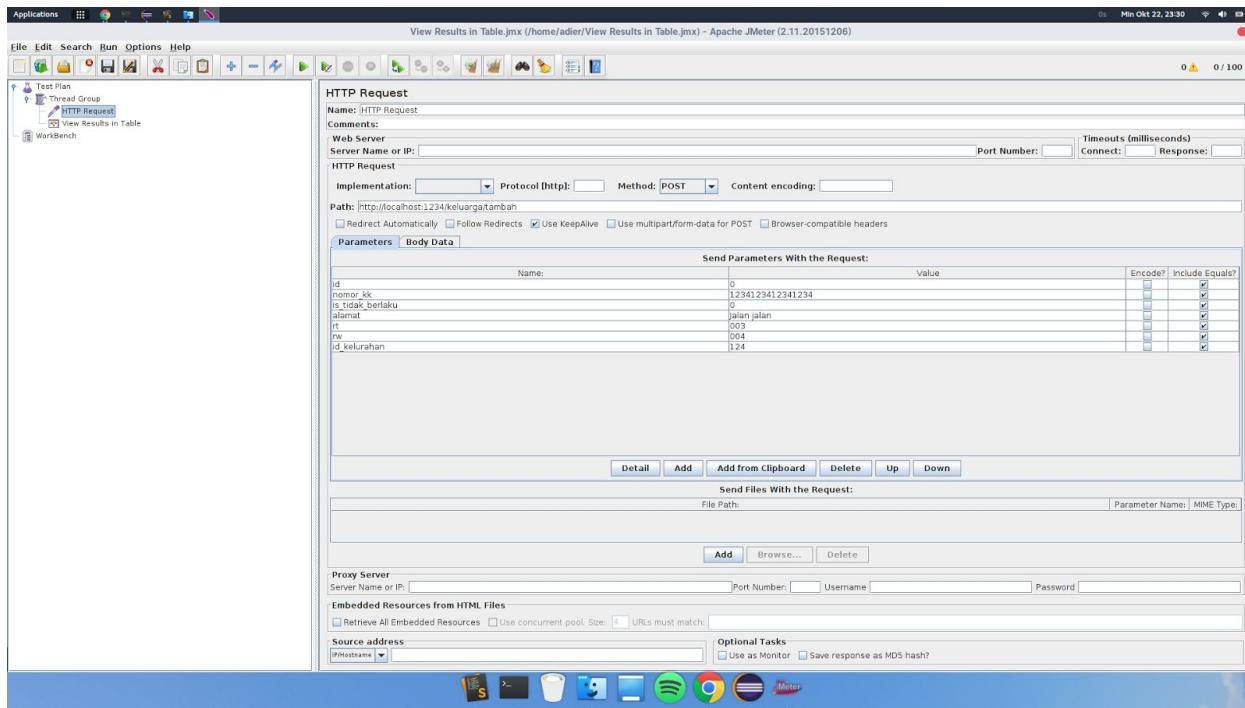
- Test Plan:** Contains a Thread Group and a View Results in Table listener.
- View Results in Table Listener Configuration:**
 - Name: View Results in Table
 - Comments: Write results to file / Read from file
 - File name: (empty)
 - Log/Display Only: Errors, Successes, Configure
- Results Data Table:**

Sample #	Start Time	Thread ID	HTTP Request	Label	Sample Time(ms)	Status	Bytes	Latency
57	23:25:55.791	Thread Group 1-57	HTTP Request		71	1193	2516	71
58	23:25:56.841	Thread Group 1-58	HTTP Request		120	1193	2516	120
59	23:25:56.892	Thread Group 1-59	HTTP Request		89	1193	2516	89
60	23:25:56.992	Thread Group 1-61	HTTP Request		64	1193	2516	64
61	23:25:57.041	Thread Group 1-65	HTTP Request		115	1193	2516	119
62	23:25:57.041	Thread Group 1-62	HTTP Request		200	1193	2516	200
63	23:25:57.091	Thread Group 1-63	HTTP Request		196	1193	2516	196
64	23:25:57.142	Thread Group 1-64	HTTP Request		187	1193	2516	187
65	23:25:57.142	Thread Group 1-65	HTTP Request		195	1193	2516	195
66	23:25:57.269	Thread Group 1-66	HTTP Request		149	1193	2516	148
67	23:25:57.292	Thread Group 1-67	HTTP Request		134	1193	2516	134
68	23:25:57.342	Thread Group 1-68	HTTP Request		105	1193	2516	105
69	23:25:57.342	Thread Group 1-69	HTTP Request		120	1193	2516	120
70	23:25:57.442	Thread Group 1-70	HTTP Request		83	1193	2516	83
71	23:25:57.492	Thread Group 1-71	HTTP Request		122	1193	2516	122
72	23:25:57.492	Thread Group 1-72	HTTP Request		87	1193	2516	87
73	23:25:57.493	Thread Group 1-74	HTTP Request		66	1193	2516	68
74	23:25:57.749	Thread Group 1-73	HTTP Request		126	1193	2516	126
75	23:25:57.749	Thread Group 1-76	HTTP Request		69	1193	2516	69
76	23:25:57.749	Thread Group 1-77	HTTP Request		122	1193	2516	122
77	23:25:57.759	Thread Group 1-77	HTTP Request		108	1193	2516	108
78	23:25:57.846	Thread Group 1-78	HTTP Request		70	1193	2516	70
79	23:25:57.896	Thread Group 1-79	HTTP Request		118	1193	2516	118
80	23:25:58.046	Thread Group 1-80	HTTP Request		77	1193	2516	77
81	23:25:58.996	Thread Group 1-81	HTTP Request		107	1193	2516	107
82	23:25:58.046	Thread Group 1-82	HTTP Request		182	1193	2516	182
83	23:25:58.149	Thread Group 1-84	HTTP Request		191	1193	2516	191
84	23:25:58.149	Thread Group 1-85	HTTP Request		146	1193	2516	146
85	23:25:58.247	Thread Group 1-86	HTTP Request		112	1193	2516	112
86	23:25:58.095	Thread Group 1-83	HTTP Request		291	1193	2516	291
87	23:25:58.297	Thread Group 1-87	HTTP Request		135	1193	2516	135
88	23:25:58.347	Thread Group 1-89	HTTP Request		103	1193	2516	103
89	23:25:58.347	Thread Group 1-88	HTTP Request		204	1193	2516	204
90	23:25:58.498	Thread Group 1-91	HTTP Request		65	1193	2516	65
91	23:25:58.498	Thread Group 1-92	HTTP Request		64	1193	2516	64
92	23:25:58.427	Thread Group 1-93	HTTP Request		276	1193	2516	276
93	23:25:58.700	Thread Group 1-95	HTTP Request		72	1193	2516	72
94	23:25:58.802	Thread Group 1-97	HTTP Request		60	1193	2516	60
95	23:25:58.802	Thread Group 1-98	HTTP Request		97	1193	2516	97
96	23:25:58.898	Thread Group 1-99	HTTP Request		76	1193	2516	76
97	23:25:58.749	Thread Group 1-96	HTTP Request		317	1193	2516	317
98	23:25:58.648	Thread Group 1-94	HTTP Request		418	1193	2516	418
99	23:25:58.948	Thread Group 1-100	HTTP Request		207	1193	2516	207
100	23:25:58.848	Thread Group 1-98	HTTP Request		307	1193	2516	307

Terlihat Akses tambah penduduk terhitung cepat dengan waktu rata rata 135 Milisecond, ada beberapa yang error karena saat proses pembuatan ada yang berjalan bersamaan sehingga ada data yang bertabrakan

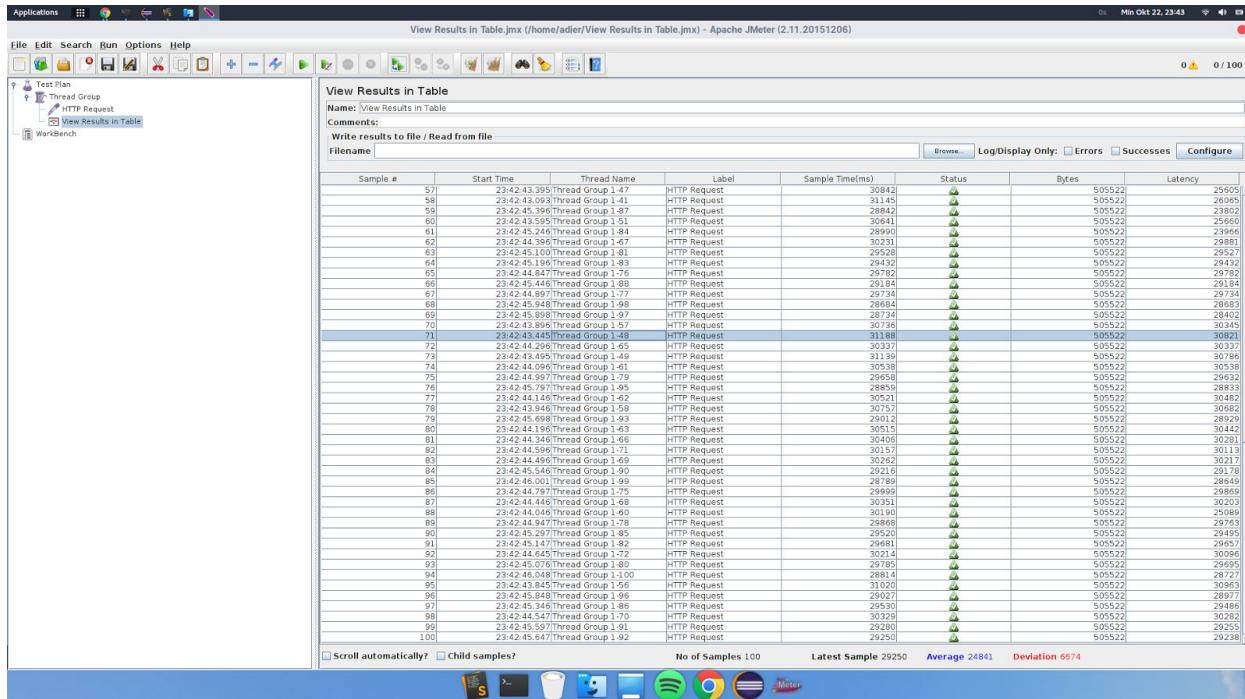
5. Akses Tambah Keluarga

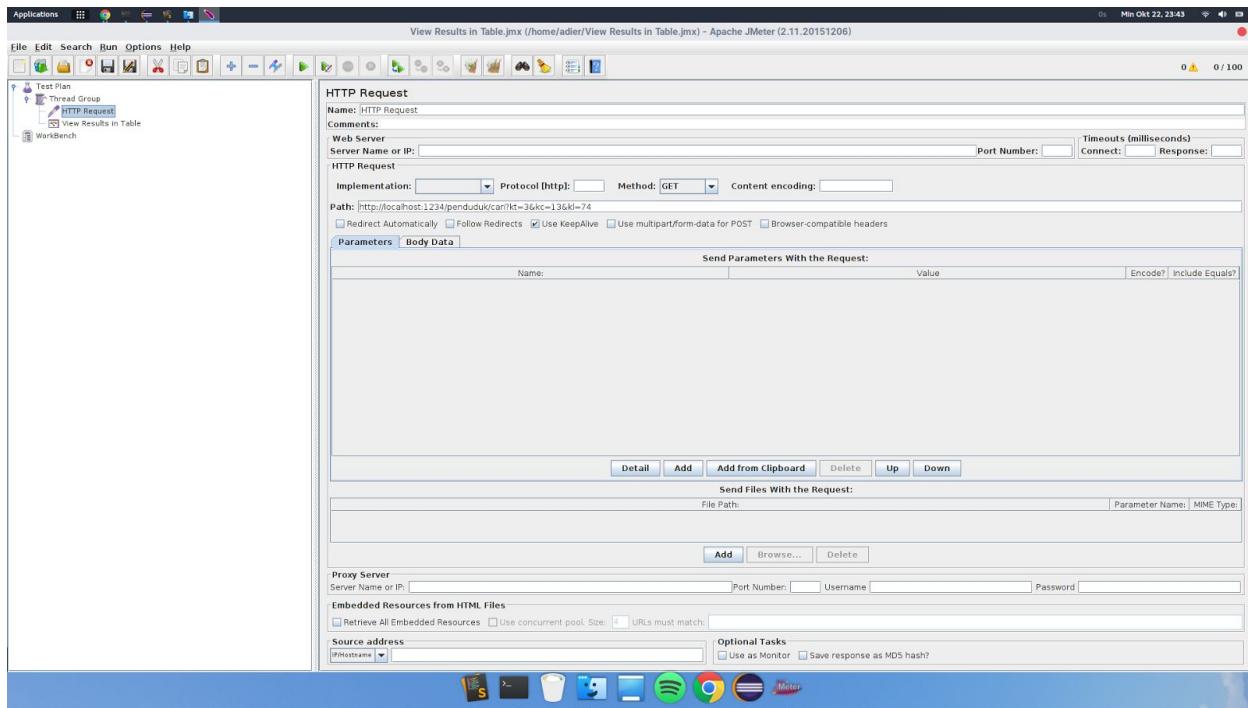




Terlihat Akses tambah keluarga terhitung cepat dengan waktu rata rata 61 Milisecond,

6. Akses Cari Penduduk





Terlihat Akses ubah keluarga terhitung standar dengan rata rata waktu 24 detik.