WRITE-UP

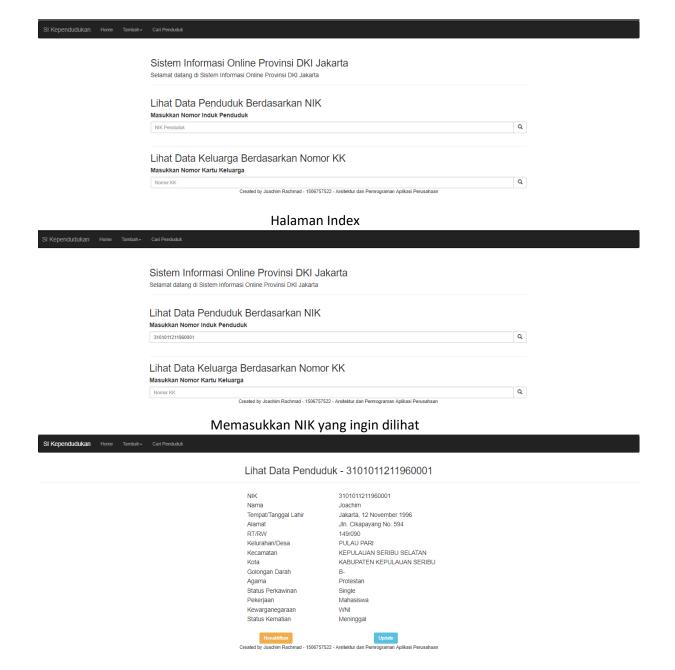
1. Tampilkan Data Penduduk Berdasarkan NIK

Initial Page: /

Form Request: GET, /penduduk

Response Page: /penduduk?nik={NIK}

Pada fitur ini, saya menggunakan form yang akan memberikan NIK yang dimasukkan dan diolah di controller. Pada controller, NIK akan di cek melalui service yang akan menjalankan query pada PendudukMapper. Action yang dimiliki oleh form ini akan melakukan redirect ke halaman /penduduk dengan parameter yang sudah diinput sebelumnya. Parameter yang diterima akan divalidasi terlebih dahulu. Apabila NIK tidak ada pada database, maka akan mengembalikan error. Pada controller, dilakukan pengecekan terhadap beberapa nilai boolean, dilakukan set untuk variabel yang memiliki nilai boolean



View dari NIK yang ingin dilihat



Error jika NIK tidak ditemukan atau tidak ada

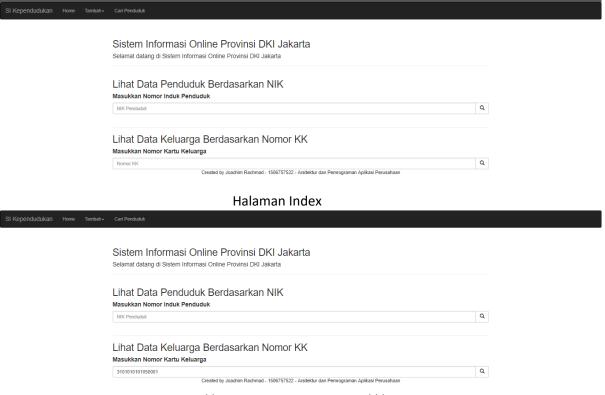
2. Tampilkan Data Keluarga Beserta Daftar Anggotanya Berdasarkan Nomor KK

Initial Page: /

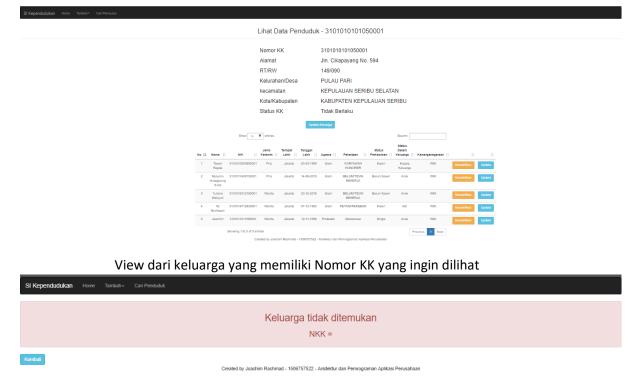
Form Request: GET, /keluarga

Response Page: /keluarga?nkk={NKK}

Pada fitur ini, saya menggunakan form yang akan memberikan Nomor KK yang dimasukkan dan diolah di controller. Pada controller, Nomor KK akan di cek melalui service yang akan menjalankan query pada Keluarga Mapper. Action yang dimiliki oleh form ini akan melakukan redirect ke halaman /keluarga dengan parameter yang sudah diinput sebelumnya. Parameter yang diterima akan divalidasi terlebih dahulu. Apabila Nomor KK tidak ada pada database, maka akan mengembalikan error. Pada controller, dilakukan pengecekan terhadap beberapa nilai boolean, dilakukan set untuk variabel yang memiliki nilai boolean. Pada view fitur ini dilakukan looping untuk mengembalikan list dari anggota yang memiliki Nomor KK seperti input. Pengambilan anggota keluarga dilakukan pada KeluargaMapper yang akan memanggil method selectAnggota_keluarga dengan parameter Nomor KK sesuai input. Method ini akan mengembalikan list dari PendudukModel.



Memasukkan Nomor KK yang ingin dilihat



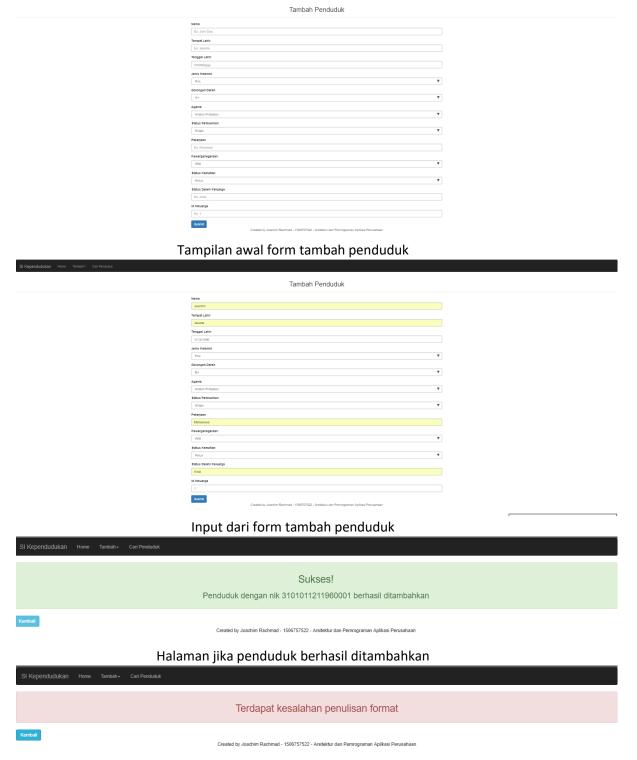
Error jika Nomor KK tidak ditemukan atau tidak ada

3. Menambahkan Penduduk Baru Sebagai Anggota Keluarga

Initial Page: /penduduk/tambah

Form Request: POST, /penduduk/tambah Response Page: /penduduk/tambah

Fungsi ini terdapat pada halaman "/penduduk/tambah". Apabila halaman diakses, maka controller akan mengembalikan view dengan form untuk menambahkan penduduk. Form ini akan mengambalikan variabel dengan nama yang sesuai dengan PendudukModel dan diterima sebagai object pada controller. Pada controller akan dilakukan pengecekan validasi terlebih dahulu, bila terdapat kesalahan format penulisan, maka akan mengembalikan error. Jika tidak, maka akan dilakukan pembuatan NIK. Pembuatan NIK dilakukan dengan melakukan pemotongan 6 kode awal dari kode kecamatan. Tanggal lahir juga diolah dengan split karena format dari nilai yang disimpan berbeda dengan format yang digunakan untuk NIK. Lalu dilakukan pengecekan terhadap jenis kelamin, apabila jenis kelamin adalah perempuan, maka tanggal dari tanggal lahir akan ditambahkan 40. Lalu lakukan pencarian terhadap penduduk yang memiliki tanggal lahir dan domisili yang sama dengan penduduk baru tersebut. Apabila ada penduduk lain yang memiliki tanggal lahir dan domisili yang sama, maka NIK akan ditambahkan dengan "0001" dan di parse menjadi tipe Long. Setelah dilakukan parsing, tambahkan dengan size dari list penduduk yang memiliki tanggal lahir dan domisili yang sama. Setelah berhasil generate NIK, dilakukan set NIK penduduk dan tambahkan penduduk ke database dengan method addPenduduk.



Halaman error jika input tidak lolos validasi

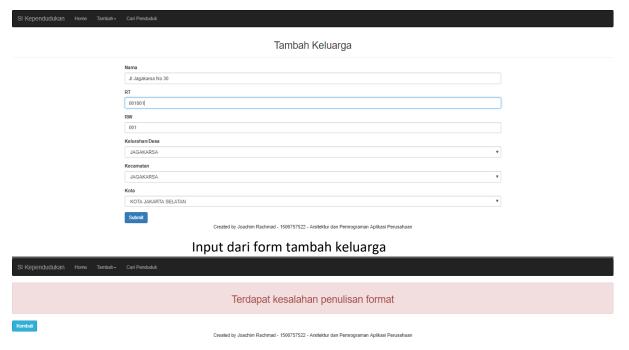
4. Menambahkan Keluarga Baru Initial Page: /keluarga/tambah

Form Request: POST, /keluarga/tambah Response Page: /keluarga/tambah

Fungsi ini terdapat pada halaman "/keluarga/tambah". Apabila halaman diakses, maka controller akan mengembalikan view dengan form untuk menambahkan keluarga. Form ini akan mengambalikan variabel dengan nama yang sesuai dengan KeluargaModel dan diterima sebagai object pada controller. Pada controller akan dilakukan pengecekan validasi terlebih dahulu, bila terdapat kesalahan format penulisan, maka akan mengembalikan error. Jika tidak, maka akan dilakukan pembuatan Nomor KK. Pembuatan Nomor KK dilakukan dengan melakukan pemotongan 6 kode awal dari kode kelurahan dan tanggal pembuatan KK yang disimpan dengan localdate. Lalu lakukan pencarian terhadap keluarga yang memiliki tanggal pembuatan dan domisili yang sama dengan penduduk baru tersebut. Apabila ada keluarga lain yang memiliki tanggal pembuatan dan domisili, maka Nomor KK akan ditambahkan dengan "0001" dan di parse menjadi tipe Long. Setelah dilakukan parsing, tambahkan dengan size dari list keluarga yang memiliki tanggal pembuatan dan domisili. Setelah berhasil generate Nomor KK, dilakukan set Nomor KK dan tambahkan keluarga ke database dengan method addKeluarga.

SI Kependudukan Home Tambah+	Cari Penduduk
	Tambah Keluarga
lamban Keluarga	
	Nama
	Jalan
	RT
	Ex 001
	RW
	Ec 001
	FULAU TIDUNG v
	Kecamatan
	NEE-PULAUAN SERIBU SELATAN ▼
	Kota
	KABUPATEN KEPULAUAN SERIBU ▼
	Submit
	Created by Joachim Rachmad - 1506757522 - Aratlektur dan Penrograman Aplikasi Perusahaan
	Tampilan awal form tambah keluarga
SI Kependudukan Home Tambah+	Cari Penduduk
	Tambah Keluarga
	Nama
	JI Jagakarsa No 30
	RT 001
	RW
	001
	Kelurahan/Desa
	Jagaxarsa ▼
	Kecamatan
	JAGAKARSA ¥
	Kota
	KOTA JAKARTA SELATAN Y
	Submit
	Created by Joachim Rachmad - 1506757522 - Arsitektur dan Penrrograman Aplikasi Perusahaan
Input dari form tambah keluarga	
SI Kependudukan Home Tambah+	Cari Penduduk
	Cultocal
	Sukses!
	Keluarga dengan nkk 3171012210170003 berhasil ditambahkan
Kembali	

Halaman jika keluarga berhasil ditambahkan



Halaman error jika input tidak lolos validasi

5. Mengubah Data Penduduk

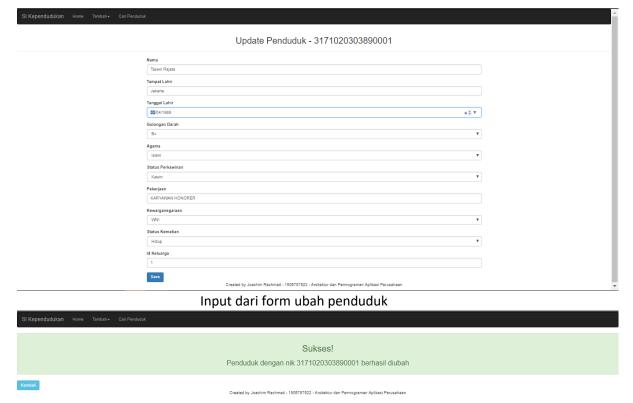
Initial Page: /penduduk/ubah/{NIK}

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

Fitur ini terdapat pada halaman "penduduk/ubah/{NIK}". NIK akan di cek kembali apakah ada di database atau tidak, jika tidak ada maka akan mengembalikan halaman error. Jika NIK ada, maka di redirect ke halaman form update penduduk. Pada fitur ini dilakukan pengecekan untuk tanggal lahir. Jika tanggal lahir diubah, maka akan dilakukan perubahan terhadap NIK. Begitu juga dengan alamat, apabila terdapat perubahan maka akan dilakukan perubahan terhadap NIK. Setelah melakukan perubahan NIK, NIK di set ke dalam object dan dimasukkan ke dalam database.



Tampilan awal form ubah penduduk



Halaman jika berhasil update penduduk

6. Mengubah Data Keluarga

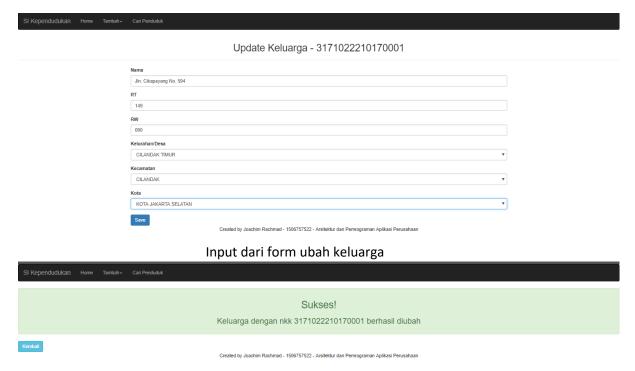
Initial Page: /keluarga/ubah/{NKK}

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

Fitur ini terdapat pada halaman "keluarga/ubah/{NKK}". Nomor KK akan di cek kembali apakah ada di database atau tidak, jika tidak ada maka akan mengembalikan halaman error. Jika Nomor KK ada, maka di redirect ke halaman form update keluarga. Pada fitur ini dilakukan pengecekan untuk domisili. Jika domisili diubah, maka akan dilakukan perubahan terhadap NKK. Begitu juga dengan tanggal pembuatan, apabila terdapat perubahan maka akan dilakukan perubahan terhadap tanggal pembuatan. Setelah melakukan perubahan Nomor KK, Nomor KK di set ke dalam object dan dimasukkan ke dalam database. Pada fitur ini dilakukan juga pengecekan terhadap NIK dari anggota keluarga. Apabila domisili dari KK berbeda dengan awalnya, maka NIK anggota keluarga harus diubah juga.



Tampilan awal form ubah keluarga



Halaman jika berhasil update keluarga

7. Mengubah Status Kematian Penduduk

Initial Page: /penduduk/{NIK}

Form Request: POST, /penduduk/mati Response Page: /penduduk/{NIK}

Fitur ini dapat dilakukan dengan melakukan klik button yang ada pada view penduduk. Ketika klik button tersebut, maka halaman akan ke redirect ke "/penduduk/mati". Pada controller, dilakukan pencarian penduduk dengan NIK yang di passing dari html ke controller. Setelah mendapatkan objek penduduk tersebut, set ulang is wafat. Lalu lakukan pencarian keluarga dengan id keluarga pada penduduk tersebut. Lakukan looping terhadap semua anggota keluarga tersebut, bila semua anggota keluarga telah wafat, maka harus set is_tidak_berlaku pada Nomor KK dengan id keluarga penduduk tersebut. Setelah itu lakukan update penduduk.



Tampilan sebelum dilakukan nonaktif penduduk



Tampilan sesudah dilakukan nonaktif penduduk

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

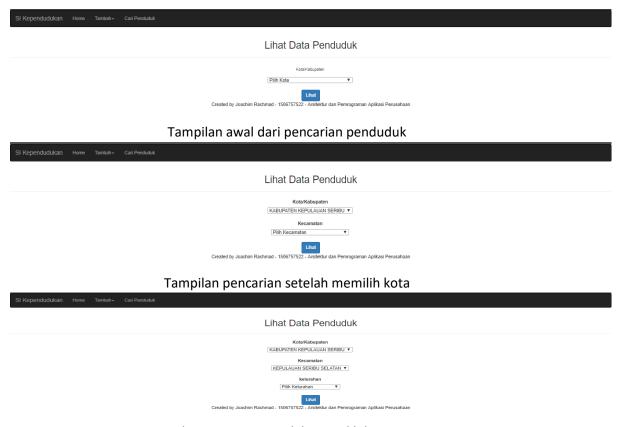
Tertentu

Initial Page: /penduduk/cari Request Method: GET,

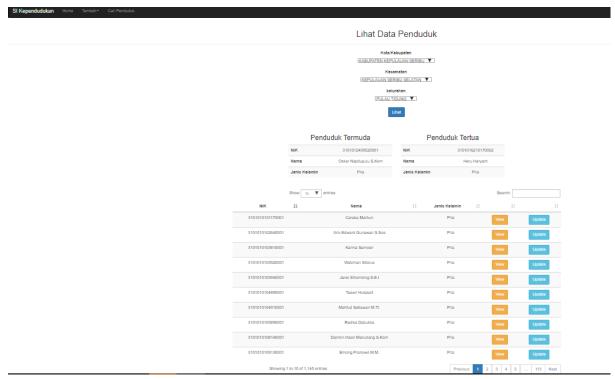
/penduduk/cari?kt={ID KOTA}&kc={ID KECAMATAN}&kl={ID KELURAHAN}

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

Fitur ini terdapat pada halaman "/penduduk/cari". Pada fitur ini dilakukan pencarian secara bertahap mulai dari kota, kecamatan, dan kelurahan. Id dari kota, kecamatan, dan kelurahan yang dipilih dijadikan sebagai parameter untuk menjalankan query yang sudah dibuat pada masing-masing mapper



Tampilan pencarian setelah memilih kecamatan



Tampilan dari pencarian penduduk pada suatu kelurahan tertentu

9. Validasi Input Untuk Semua Form POST

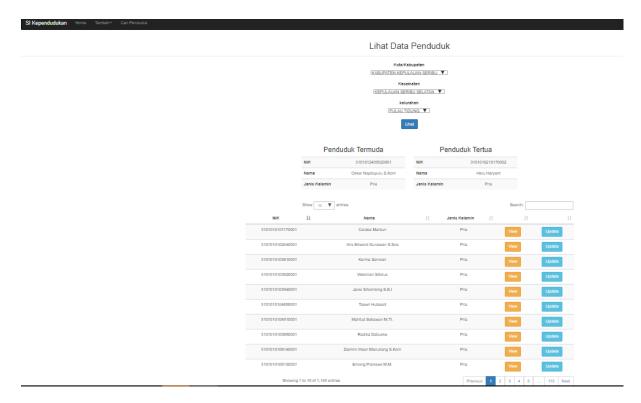
Fitur ini dibuat dengan melakukan pengecekan terhadap form yang diberikan. Forn dibuat agar tidak null dan memiliki panjang karakter sesuai database serta ukuran yang menyesuaikan dengan model



Tampilan jika ada kesalahan dalam penulisan format form

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

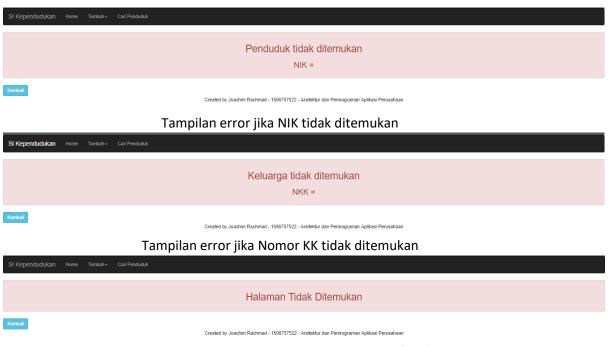
Fitur ini dibuat dengan melakukan compare dari 2 buah object penduduk dengan membandingkan tanggal lahir.



Tampilan dari penduduk paling muda dan penduduk paling tua

11. Menambahkan Error Page

Pada fitur ini, dibuat error page 404 untuk mengatasi halaman yang tidak tersedia. Dibuat juga error page untuk menandakan pencarian NIK dan NKK terdapat pada database atau tidak



Tampilan error jika halaman tidak ditemukan (404)

12. Fitur-Fitur Lain yang Mendukung Aplikasi

Fitur lain yang mendukung aplikasi adalah dengan menghubungankan fitur-fitur yang telah dibuat untuk mempermudah pemakaian



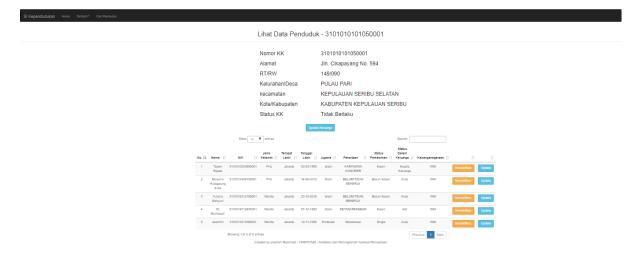
Membuat fitur yang dapat mengubah data dari penduduk dari view penduduk

Single Mahasiswa

WNI Meninggal

Status Perkawinan

Pekerjaan Kewarganegaraan Status Kematian



Membuat fitur yang dapat mengubah data keluarga, melihat anggota keluarga dan mengubah data dari anggota keluarga dari view keluarga



Membuat fitur tambah penduduk dan tambah keluarga pada navigasi

OPTIMASI DATABASE

Optimasi dilakukan dengan menambahkan beberapa atribut seperti:

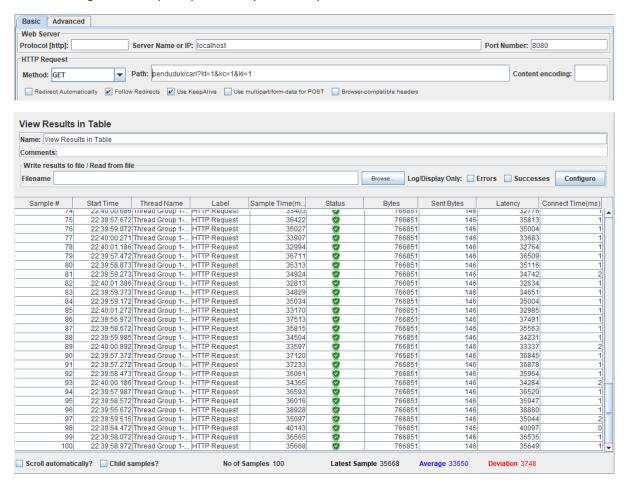
- 1. PK pada kolom id pada tabel penduduk, keluarga, kota, kecamatan, kelurahan
- 2. Index pada kolom id pada tabel penduduk, keluarga, kota, kecamatan,dan kelurahan, nik pada tabel penduduk, nomor_kk dan id_kelurahan pada tabel keluarga dan id_kecamatan pada tabel kelurahan

Penggunaan PK dan index dapat meningkatkan efesiensi karena jumlah data yang besar sehingga dapat meningkatkan efisiensi dari pencarian

STRESS TESTING

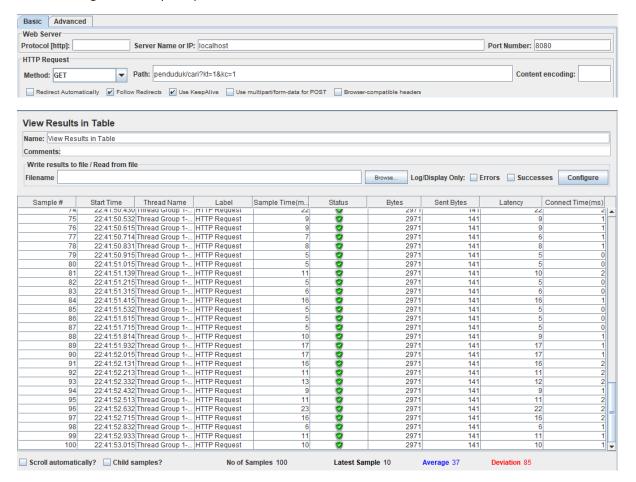
Stress testing dilakukan pada halaman yang diperkirakan memakan waktu paling lama, sehingga akan dilakukan uji coba pada halam tersebut.

Stress testing dilakukan pada pencarian penduduk pada suatu kelurahan



Hasil dari testing didapatkan bahwa untuk menjalankan request HTTP ini, untuk rata" waktu nya membutuhkan 33650 ms yang merupakan waktu yang lambat. Untuk load pengguna yang banyak, halaman ini tidak reliable.

Stress testing dilakukan pada pencarian kelurahan dari suatu kecamatan



Hasil dari testing didapatkan bahwa untuk menjalankan request HTTP ini, untuk rata" waktu nya membutuhkan 37 ms yang merupakan waktu yang cukup singkat. Untuk load pengguna yang banyak, halaman ini reliable.