

## Tugas 1

## Sistem Kependudukan Online Provinsi DKI Jakarta

## Stress Testing dan Optimasi *Database*

Sebelum melakukan optimasi *database*, *stress testing* dilakukan namun selalu terjadi status gagal yang mana menandakan sistem tidak *reliable* dan tidak *responsive*. Kemudian, dilakukan optimasi *database* dengan cara penambahan *primary key* pada setiap kolom yang ada di *database*. *Primary key* untuk semua tabel adalah *id* yang mana untuk *id* keluarga dan *id* penduduk diset *autoincrement* (untuk mempercepat proses *query* sehingga tidak perlu ada logic untuk menambah *id* pada project yang dikerjakan) dan pasti unik. Kemudian, untuk mempercepat proses pencarian, nomor *kk* pada tabel keluarga dijadikan index karena akan sering diakses, juga *nik* pada tabel penduduk. Sehingga, sistem akan berjalan lebih cepat dan dapat diakses oleh ribuan orang. Hal ini dapat dilakukan dengan melakukan *stress testing* kembali. *Stress testing* dilakukan dengan *number of threads* (jumlah *thread* yang akan mengakses sistem) sebanyak 2000 (untuk mencari keluarga) dan 1000 (untuk mencari penduduk) karena sistem yang dibuat ingin diuji reliabilitasnya ketika diakses oleh ribuan orang. Kemudian *ramp-up period* di-set 10 detik yang merupakan waktu maksimal untuk membuat *thread* sampai pada jumlah yang diinginkan karena sistem ingin diuji kecepatannya. Serta *loop count* karena uji coba cukup untuk dilakukan satu kali, maka di-set 1. Berikut adalah hasilnya:

- Mencari keluarga dengan nkk yang diinginkan

## View Results in Table

Name:

Comments:

☐ Write results to file / Read from file

File name:

Log/Display Only:
☐ Errors
☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	22:38:31.757	Thread Group ...	HTTP Request	96	✓	4368	147	96	1
2	22:38:31.811	Thread Group ...	HTTP Request	167	✓	4368	147	167	1
3	22:38:31.761	Thread Group ...	HTTP Request	217	✓	4368	147	217	1
4	22:38:31.733	Thread Group ...	HTTP Request	245	✓	4368	147	245	1
5	22:38:31.754	Thread Group ...	HTTP Request	224	✓	4368	147	224	1
6	22:38:31.760	Thread Group ...	HTTP Request	227	✓	4368	147	227	1
7	22:38:31.808	Thread Group ...	HTTP Request	185	✓	4368	147	185	1
8	22:38:31.713	Thread Group ...	HTTP Request	283	✓	4368	147	283	1
9	22:38:31.721	Thread Group ...	HTTP Request	277	✓	4368	147	277	1
10	22:38:31.729	Thread Group ...	HTTP Request	418	✓	4368	147	418	1
11	22:38:31.989	Thread Group ...	HTTP Request	158	✓	4368	147	158	1
12	22:38:31.777	Thread Group ...	HTTP Request	370	✓	4368	147	370	1
13	22:38:31.850	Thread Group ...	HTTP Request	298	✓	4368	147	298	1
14	22:38:31.831	Thread Group ...	HTTP Request	319	✓	4368	147	319	1
15	22:38:31.772	Thread Group ...	HTTP Request	378	✓	4368	147	378	1
16	22:38:31.846	Thread Group ...	HTTP Request	304	✓	4368	147	304	1
17	22:38:31.828	Thread Group ...	HTTP Request	325	✓	4368	147	325	1
18	22:38:31.791	Thread Group ...	HTTP Request	362	✓	4368	147	362	1
19	22:38:31.993	Thread Group ...	HTTP Request	160	✓	4368	147	160	1
20	22:38:31.781	Thread Group ...	HTTP Request	372	✓	4368	147	372	1
21	22:38:31.805	Thread Group ...	HTTP Request	349	✓	4368	147	349	1
22	22:38:31.825	Thread Group ...	HTTP Request	329	✓	4368	147	329	1
23	22:38:31.993	Thread Group ...	HTTP Request	305	✓	4368	147	304	1
24	22:38:31.996	Thread Group ...	HTTP Request	302	✓	4368	147	301	1
25	22:38:31.978	Thread Group ...	HTTP Request	321	✓	4368	147	321	1
26	22:38:31.980	Thread Group ...	HTTP Request	319	✓	4368	147	319	1
27	22:38:32.155	Thread Group ...	HTTP Request	144	✓	4368	147	144	1

☐ Scroll automatically?
☐ Child samples?
No of Samples 3500
Latest Sample 246708
Average 44404
Deviation 53579

- Mencari penduduk dengan nik yang diinginkan

View Results in Table									
Name: View Results in Table									
Comments:									
Write results to file / Read from file									
Filename									
		Browse...		Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes		Configure			
Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	23:04:18.203	Thread Group ...	HTTP Request	34	✓	4691	147	28	2
2	23:04:18.263	Thread Group ...	HTTP Request	10	✓	4691	147	10	1
3	23:04:18.213	Thread Group ...	HTTP Request	60	✓	4691	147	60	1
4	23:04:18.240	Thread Group ...	HTTP Request	34	✓	4691	147	33	1
5	23:04:18.223	Thread Group ...	HTTP Request	59	✓	4691	147	59	1
6	23:04:18.253	Thread Group ...	HTTP Request	31	✓	4691	147	31	1
7	23:04:18.274	Thread Group ...	HTTP Request	14	✓	4691	147	14	1
8	23:04:18.243	Thread Group ...	HTTP Request	45	✓	4691	147	45	1
9	23:04:18.283	Thread Group ...	HTTP Request	11	✓	4691	147	11	1
10	23:04:18.293	Thread Group ...	HTTP Request	10	✓	4691	147	10	1
11	23:04:18.303	Thread Group ...	HTTP Request	81	✓	4691	147	81	1
12	23:04:18.334	Thread Group ...	HTTP Request	79	✓	4691	147	79	1
13	23:04:18.313	Thread Group ...	HTTP Request	106	✓	4691	147	105	1
14	23:04:18.391	Thread Group ...	HTTP Request	29	✓	4691	147	23	1
15	23:04:18.385	Thread Group ...	HTTP Request	40	✓	4691	147	40	1
16	23:04:18.387	Thread Group ...	HTTP Request	41	✓	4691	147	41	1
17	23:04:18.386	Thread Group ...	HTTP Request	45	✓	4691	147	45	1
18	23:04:18.390	Thread Group ...	HTTP Request	58	✓	4691	147	57	1
19	23:04:18.422	Thread Group ...	HTTP Request	39	✓	4691	147	39	6
20	23:04:18.412	Thread Group ...	HTTP Request	49	✓	4691	147	48	18
21	23:04:18.431	Thread Group ...	HTTP Request	36	✓	4691	147	36	1
22	23:04:18.443	Thread Group ...	HTTP Request	37	✓	4691	147	37	0
23	23:04:18.453	Thread Group ...	HTTP Request	30	✓	4691	147	30	0
24	23:04:18.463	Thread Group ...	HTTP Request	40	✓	4691	147	40	0
25	23:04:18.473	Thread Group ...	HTTP Request	36	✓	4691	147	36	0
26	23:04:18.483	Thread Group ...	HTTP Request	40	✓	4691	147	40	0
27	23:04:18.494	Thread Group ...	HTTP Request	31	✓	4691	147	31	0
<input type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples? No of Samples 1000 Latest Sample 148584 Average 30264 Deviation 41153									

Dapat terlihat bahwa sistem berhasil dijalankan dan waktu untuk mengeksekusi setiap alamat hanya 96-400 ms dan 34-106 ms yang mana berarti sistem memiliki kecepatan yang tinggi dan *reliable*.

## Penjelasan fitur

### 1. Fitur melihat data penduduk

Fitur ini dapat diakses melalui halaman localhost:8080/ yang mana merupakan halaman home dari sistem, sehingga ketika *user* membuka sistem ini, *user* akan langsung diarahkan pada halaman ini. Untuk mencari data penduduk berdasarkan nik, *user* bisa mengisi kolom input nik yang berada di bawah tulisan “Masukan Nomor Induk Kependudukan” yang akan me-*redirect* ke halaman localhost:8080/penduduk?nik={nomor nik}. Di sini saya melakukan validasi *front-end* dengan menambahkan *required* pada input kolom tersebut. Kemudian, pada *back-end* saya melakukan validasi untuk mengecek apakah nik yang dicari ada pada *database* pada *service layer* untuk penduduk dan juga diberikan kondisi pada *controller* penduduk. Sedangkan untuk fungsi fitur tersebut sudah berjalan sebagaimana yang diminta jika masukan *user* benar.

### 2. Fitur Melihat Keluarga

Untuk mencari data keluarga berdasarkan nkk, *user* bisa mengisi kolom input nkk yang berada di bawah tulisan “Masukan Nomor Kartu Keluarga” yang akan me-*redirect* ke halaman localhost:8080/keluarga?nkk={nomor nkk}. Di sini saya melakukan validasi *front-end* dengan menambahkan *required* pada input kolom tersebut. Kemudian, pada *back-end* saya melakukan validasi jika nkk yang dimasukan tidak ada di *database*, maka akan ditampilkan halaman error, yang mana halaman tersebut merupakan hasil dari proses *service layer* untuk keluarga *controller*.

Sedangkan untuk fungsi fitur tersebut sudah berjalan sebagaimana yang diminta jika masukan *user* benar.

### 3. Fitur Menambah Penduduk

Untuk menambahkan data penduduk, saya melakukan pengembangan sistem dengan menambahkan navbar “Tambah Data” yang jika diklik akan memunculkan *dropdown*. Untuk menambah data penduduk, *user* dapat mengklik “Data Penduduk” sehingga *user* tidak perlu mengakses secara manual melalui halaman localhost:8080/penduduk/tambah. Untuk *form* yang muncul setelah *user* mengklik navbar tersebut, saya melakukan validasi *front-end* dengan menambahkan *required* pada setiap kolom yang ada. Kemudian, untuk *back-end* saya menambahkan pengecekan id keluarga pada *service layer* dan memberikan kondisi pada *controller* jika id keluarga tidak ada maka akan menampilkan halaman *error*. Sedangkan untuk fungsi fitur tersebut sudah berjalan sebagaimana yang diminta jika masukan *user* benar.

### 4. Fitur Menambah Keluarga

Untuk mengakses fitur tersebut, saya melakukan pengembangan yang sama dengan fitur 3 sehingga *user* hanya perlu mengklik “Data Keluarga” pada *dropdown* navbar “Tambah Data”. Kemudian, untuk validasi *front-end* saya menambahkan *required* untuk setiap kolom, dan validasi *back-end* dengan melakukan pengecekan apakah kelurahan yang dimasukkan berada pada kecamatan yang dimasukkan serta kota yang dimasukkan pada *service layer*, dan memberikan kondisi pada *controller* sehingga jika kelurahan/kecamatan/kota tidak berada pada satu domisili akan dihasilkan halaman *error*. Validasi juga dilakukan untuk input setiap kelurahan/kecamatan/kota apakah ada di *database* atau tidak, jika tidak ada maka akan dihasilkan halaman *error*. Sedangkan untuk fungsi fitur tersebut sudah berjalan sebagaimana yang diminta jika masukan *user* benar.

### 5. Fitur Mengubah Data Penduduk

Untuk fitur ini, *user* dapat mengakses halaman localhost:8080/penduduk/ubah/{nomor nik}, dengan nomor nik yang diinginkan. Kemudian untuk halaman *form* yang muncul, saya juga melakukan validasi *front-end* dengan *required* pada setiap kolom, dan juga *back-end* untuk mengecek apakah id keluarga yang diganti ada atau tidak di *database* pada *service layer* dan memberikan kondisi pada *controller* sehingga ketika id keluarga yang dimasukkan tidak ada di *database*, akan menghasilkan halaman *error*. Sedangkan untuk fungsi fitur tersebut sudah berjalan sebagaimana yang diminta jika masukan *user* benar.

### 6. Fitur Mengubah Keluarga

Untuk mengubah keluarga *user* dapat mengakses halaman localhost:8080/keluarga/ubah/(nomor kk) dengan nomor kk yang diinginkan. Kemudian, untuk *form* yang muncul saya juga melakukan validasi *front-end required* serta *back-end* untuk mengecek kelurahan kecamatan kota apakah berada pada domisili yang sama serta apakah ada di *database* pada *service layer* dan memberikan kondisi pada *controller* sehingga akan ditampilkan *error* jika data yang diubah tidak memenuhi validasi. Sedangkan untuk fungsi fitur tersebut sudah berjalan sebagaimana yang diminta jika masukan *user* benar.

## 7. Fitur Menonaktifkan Penduduk

Fitur menonaktifkan penduduk dilakukan *handle* dengan men-*disable button* nonaktif pada setiap halaman data penduduk jika penduduk tersebut sudah wafat. Jika penduduk hidup dan ingin di-*update* ketika sudah wafat, maka *user* hanya perlu mengklik *button* nonaktifkan pada halaman data penduduk yang diinginkan.

## 8. Fitur Melihat Data Penduduk Berdasarkan Domisili

Untuk fitur ini *user* dapat mengakses halaman localhost:8080/penduduk/cari yang mana akan menampilkan *dropdown list of* kota, yang ketika *user* mengklik salah satu kota tersebut, akan me-*redirect* ke halaman untuk memilih kecamatan yang berada pada kota tersebut, yang kemudian jika sudah dipilih akan me-*redirect* ke halaman kelurahan yang berada pada kecamatan tersebut, yang ketika dipilih akan menampilkan data penduduk yang berada pada domisili tersebut menggunakan datatables.

### Tambahan:

Sistem dikembangkan dengan *bootstrap* yang sudah dimasukkan ke folder *static* pada *project*, serta terdapat folder *fragment* dengan file *fragments* yang berfungsi untuk membuat *fragment* header dan footer yang akan dimasukkan pada setiap halaman *view* sebagai berikut:

