

Nama : Adil Krisnadi Pradana
NPM : 1406623631
Kelas : APAP-C

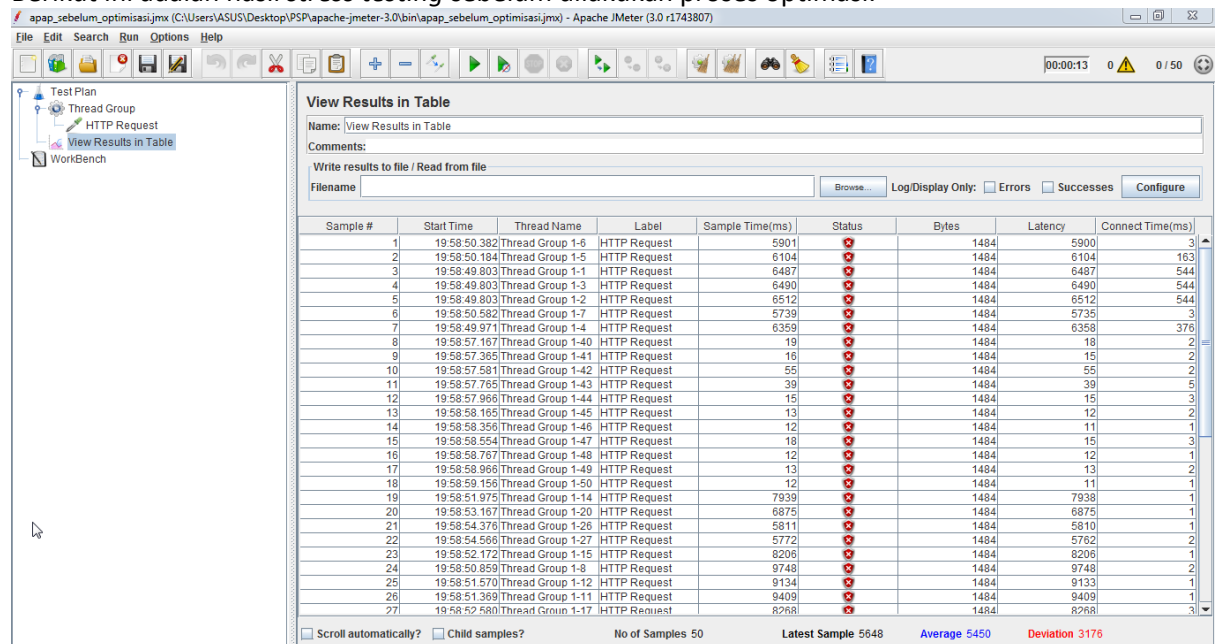
WRITE-UP TUGAS 1 APAP

1. Penerapan MVC dan *layering*

Tugas 1 APAP kali ini menerapkan konsep MVC (Model-View-Controller). *Package* yang digunakan pada tugas 1 APAP kali ini terdiri dari empat buah *package*, yakni *package* controller, dao, model, dan service. *Package* controller berisi file PageController yang berfungsi sebagai *controller* utama untuk melakukan *redirecting* ke halaman yang kita inginkan, seperti halaman add, edit, atau view. *Package* dao berisi file-file Mapper (KelurahanMapper, KecamatanMapper, KotaMapper, KeluargaMapper, PendudukMapper) yang digunakan untuk keperluan *database*. Lalu *package* model berisi file-file Model (KelurahanModel, KecamatanModel, KotaModel, KeluargaModel, PendudukModel) yang digunakan untuk menyimpan berbagai atribut yang dibutuhkan. Sementara *package* service berisi file-file Service yang digunakan untuk keperluan fungsional. Selain itu, tugas 1 APAP kali ini juga menggunakan beberapa *dependencies* yang telah banyak digunakan di tutorial-tutorial sebelumnya, diantaranya Thymeleaf, Web, Devtools, Mybatis, Lombok, dan juga MySQL.

2. Stress Testing

Berikut ini adalah hasil *stress testing* sebelum dilakukan proses optimasi:



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency	Connect Time(ms)
1	19:58:50.382	Thread Group 1-6	HTTP Request	5901	Success	1484	5900	3
2	19:58:50.184	Thread Group 1-5	HTTP Request	6104	Success	1484	6104	163
3	19:58:49.803	Thread Group 1-1	HTTP Request	6487	Success	1484	6487	544
4	19:58:49.803	Thread Group 1-3	HTTP Request	6490	Success	1484	6490	544
5	19:58:49.803	Thread Group 1-2	HTTP Request	6512	Success	1484	6512	544
6	19:58:50.582	Thread Group 1-7	HTTP Request	5739	Success	1484	5735	3
7	19:58:49.971	Thread Group 1-4	HTTP Request	6359	Success	1484	6358	376
8	19:58:57.167	Thread Group 1-40	HTTP Request	19	Success	1484	18	2
9	19:58:57.365	Thread Group 1-41	HTTP Request	16	Success	1484	15	2
10	19:58:57.581	Thread Group 1-42	HTTP Request	55	Success	1484	55	2
11	19:58:57.785	Thread Group 1-43	HTTP Request	39	Success	1484	39	5
12	19:58:57.966	Thread Group 1-44	HTTP Request	15	Success	1484	15	3
13	19:58:58.165	Thread Group 1-45	HTTP Request	13	Success	1484	12	2
14	19:58:58.356	Thread Group 1-46	HTTP Request	12	Success	1484	11	1
15	19:58:58.554	Thread Group 1-47	HTTP Request	18	Success	1484	15	3
16	19:58:58.767	Thread Group 1-48	HTTP Request	12	Success	1484	12	1
17	19:58:58.966	Thread Group 1-49	HTTP Request	13	Success	1484	13	2
18	19:58:59.156	Thread Group 1-50	HTTP Request	12	Success	1484	11	1
19	19:58:51.975	Thread Group 1-14	HTTP Request	7939	Success	1484	7938	1
20	19:58:53.167	Thread Group 1-20	HTTP Request	6875	Success	1484	6875	1
21	19:58:54.376	Thread Group 1-26	HTTP Request	5811	Success	1484	5810	1
22	19:58:54.566	Thread Group 1-27	HTTP Request	5772	Success	1484	5762	2
23	19:58:52.172	Thread Group 1-15	HTTP Request	8206	Success	1484	8206	1
24	19:58:50.859	Thread Group 1-8	HTTP Request	9748	Success	1484	9748	2
25	19:58:51.570	Thread Group 1-12	HTTP Request	9134	Success	1484	9133	1
26	19:58:51.359	Thread Group 1-11	HTTP Request	9409	Success	1484	9409	1
27	19:58:42.580	Thread Group 1-17	HTTP Request	8268	Success	1484	8268	3

Berikut ini adalah hasil *stress testing* setelah dilakukan proses optimasi:

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency	Connect Time(ms)
1	20:18:20.940	Thread Group 1-2	HTTP Request	505	Success	1484	504	362
2	20:18:20.940	Thread Group 1-1	HTTP Request	505	Success	1484	504	374
3	20:18:21.288	Thread Group 1-4	HTTP Request	157	Success	1484	157	28
4	20:18:21.077	Thread Group 1-3	HTTP Request	378	Success	1484	378	224
5	20:18:21.495	Thread Group 1-5	HTTP Request	56	Success	1484	56	3
6	20:18:21.678	Thread Group 1-6	HTTP Request	29	Success	1484	29	3
7	20:18:21.883	Thread Group 1-7	HTTP Request	18	Success	1484	18	2
8	20:18:22.082	Thread Group 1-8	HTTP Request	16	Success	1484	16	2
9	20:18:22.282	Thread Group 1-9	HTTP Request	16	Success	1484	16	2
10	20:18:22.482	Thread Group 1-10	HTTP Request	16	Success	1484	16	2
11	20:18:22.679	Thread Group 1-11	HTTP Request	15	Success	1484	15	2
12	20:18:22.881	Thread Group 1-12	HTTP Request	16	Success	1484	16	2
13	20:18:23.081	Thread Group 1-13	HTTP Request	18	Success	1484	18	1
14	20:18:23.280	Thread Group 1-14	HTTP Request	25	Success	1484	25	3
15	20:18:23.483	Thread Group 1-15	HTTP Request	63	Success	1484	63	39
16	20:18:23.686	Thread Group 1-16	HTTP Request	51	Success	1484	51	2
17	20:18:23.882	Thread Group 1-17	HTTP Request	24	Success	1484	24	2
18	20:18:24.081	Thread Group 1-18	HTTP Request	19	Success	1484	19	2
19	20:18:24.278	Thread Group 1-19	HTTP Request	17	Success	1484	17	4
20	20:18:24.479	Thread Group 1-20	HTTP Request	18	Success	1484	18	4
21	20:18:24.681	Thread Group 1-21	HTTP Request	17	Success	1484	17	3
22	20:18:24.879	Thread Group 1-22	HTTP Request	16	Success	1484	16	2
23	20:18:25.080	Thread Group 1-23	HTTP Request	13	Success	1484	13	2
24	20:18:25.279	Thread Group 1-24	HTTP Request	12	Success	1484	12	1
25	20:18:25.480	Thread Group 1-25	HTTP Request	14	Success	1484	14	2
26	20:18:25.680	Thread Group 1-26	HTTP Request	14	Success	1484	14	1
27	20:18:25.879	Thread Group 1-27	HTTP Request	12	Success	1484	12	3

Stress testing sendiri dilakukan pada pencarian daftar semua penduduk yang terdapat di sebuah kelurahan pada sebuah kecamatan dalam kota tertentu. Sebelum melakukan optimasi pada *database*, perlu waktu yang lumayan lama untuk dapat melihat keseluruhan daftar penduduk yang ditampilkan. Hal tersebut dapat dilihat dari besarnya *sample time* dan *latency* pada hasil *stress testing* sebelum optimasi. Barulah setelah optimasi dilakukan, seluruh daftar penduduk dapat ditampilkan dalam waktu singkat di mana *sample time* dan *latency* yang dihasilkan juga jauh berkurang drastis.

3. Optimasi *Database*

Berikut ini adalah langkah-langkah dari optimasi *database* yang dilakukan:

- Pada tabel Kelurahan, ditambahkan *primary key* pada “id” dan juga *index* pada “id”, “id_kecamatan”, dan “kode_kelurahan”
- Pada tabel Kecamatan, ditambahkan *primary key* pada “id” dan juga *index* pada “id”, “id_kota”, dan “kode_kecamatan”
- Pada tabel Kota, ditambahkan *primary key* pada “id” dan juga *index* pada “id” dan “kode_kota”
- Pada tabel Keluarga, ditambahkan *primary key* pada “id” dan juga *index* pada “id” dan “nomor_kk”
- Pada tabel Penduduk, ditambahkan *primary key* pada “id” dan juga *index* pada “id” dan “nik”

4. Fitur Tambahan

Beberapa fitur tambahan yang dikerjakan adalah adanya *Not Found Handler*, di mana dengan menggunakan file “404.html”, dapat melakukan *handle* secara otomatis terhadap *error* seperti saat melakukan *mapping* ke halaman yang tidak ada



Halaman Tidak Ditemukan

Selain itu, fitur lain yang telah diimplementasikan adalah penambahan validasi *required* di semua *form* POST (seperti untuk add dan view), juga telah otomatis melakukan validasi pada fitur Cari, di mana *user* harus memasukkan data secara berurutan mulai dari Kota, Kecamatan, lalu Kelurahan, menggunakan pilihan *dropdown* sebelum mulai mencari daftar penduduk.

localhost:8080/penduduk/tambah

Navbar Home Penduduk Keluarga Cari Data Search Search

Tambah Penduduk

Masukkan Data Penduduk

Nama Lengkap:

Masukkan Nama Lengk

Please fill out this field.

Tempat Lahir:

Masukkan Tempat Lahir

localhost:8080/penduduk/cari

Navbar Home Penduduk Keluarga Cari Data Search Search

Cari Penduduk

Kota / Kabupaten:

Pilih Kota / Kabupaten

Kecamatan:

Pilih Kecamatan

Kelurahan:

Pilih Kelurahan

Cari