# Penjelasan Tugas 1

Pada tugas 1 ini yang pertama saya lakukan adalah merancang aplikasi yaitu mencari template bootstrap yang sesuai, membuat spring boot project nya. Kemudian saya mengimplementasikan template tadi dan membuat fragment untuk header menu dan head nya sehingga dapat di extend di view lain yang menggunakan. Kemudian menyiapkan semua package, menambahkan Model berdasarkan entity yang ada di database. Setelah itu membuat controller yang akan menerima request. Langkah selanjutnya menambahkan class mapper untuk masing-masing model. Kemudian menambah interface service dan class yang mengimplement.

Untuk setiap fitur, pertama tambahkan requeest mapping di controller, menambah query di mapper dan menambah method di interface service dan class yang implement untuk memanggil fungsi di mapper tadi. Begitu selanjutanya.

Setelah fungsionalitas selesai, saya melakukan optimasi query dengan menambah indexing untuk column yang digunakan sebagai where clause di query.

Kemudian melakukan stress test dengan beberapa test plan yang berbeda

#### Fitur Mengubah Data Mahasiswa

Fitur ini dapat diakses melalui menu cari mahasiswa, pada table yang berisi data mahasiswa terdapat action ubah yang akan redirect ke fitur update. Atau jika ingin hit secara langsung dari url maka contoh url nya adalah: <a href="http://localhost:8080/mahasiswa/ubah/141021354001">http://localhost:8080/mahasiswa/ubah/141021354001</a>

#### Fitur Lain yang Mendukung Aplikasi

Fitur lain yang mendukung aplikasi adalah menambah validasi form di view yaitu menggunakan jquery form validator (<a href="http://www.formvalidator.net">http://www.formvalidator.net</a>)

Form yang di validasi adalah FORM ADD dan FORM UPDATE, diakses melalui menu TAMBAH MAHASISWA

Contoh hasil validasinya adalah sebagai berikut:

### Tambah Mahasiswa

Nama Lengkap		This is a required field
Tempat Lahir		This is a required field
Tanggal Lahir	ex:1994-05-05	This is a required field
Jenis Kelamin	Laki-Laki v	
Agama	Protestan •	
Golongan Darah	AB- ▼	
Tahun Masuk		This is a required field
Jalur Masuk	Undangan Reguler/SNMPTN •	

## **Optimasi Database**

Optimasi Database yang saya lakukan adalah dengan menambah index untuk column yang digunakan sebagai where clause di table mahasiswa. Column yang ditambah indexnya adalah:

id\_prodi, tahun\_masuk, status. Index akan mempercepat proses search ketika terdapat where clause pada query select.

```
-- Indexes for table `mahasiswa`

--

ALTER TABLE `mahasiswa`

ADD PRIMARY KEY (`id`),

ADD KEY `idx_npm` (`npm`),

ADD KEY `idx_idProdi` (`id_prodi`),

ADD KEY `idx_tahunMasuk` (`tahun_masuk`),

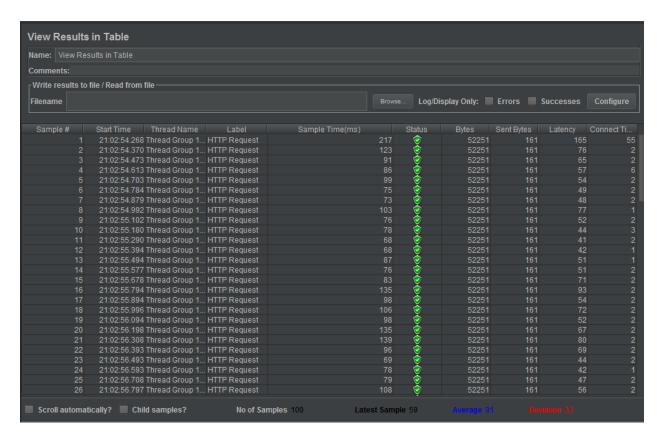
ADD KEY `idx_status` (`status`);
```

Optimasi yang lain adalah pada query select dari beberapa table menggunakan JOIN daripada menggunakan NESTED QUERY yang execustion time nya lebih lama daripada JOIN.

### **Stress Testing**

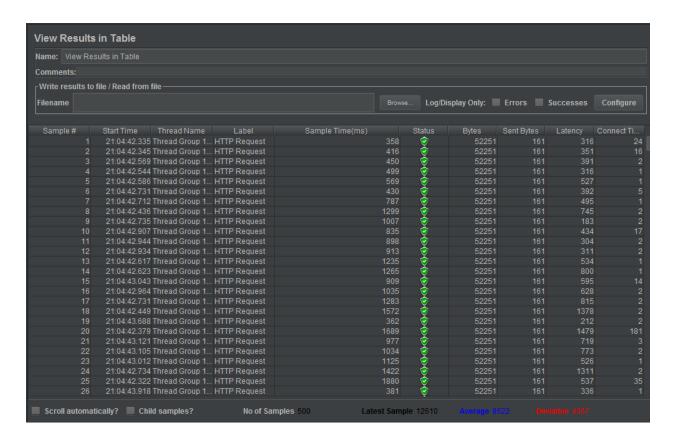
Stress testing menggunakan jMeter dengan beberapa test plan. Namun yang menjadi sasaran utama stress test ini adalah untuk fitur search all mahasiswa berdasarkan prodi.

1. Test plan dengan Number of Thread 100 Url yang di hit: localhost:8080/mahasiswa/cari3?univ=3&fakultas=16&prodi=29



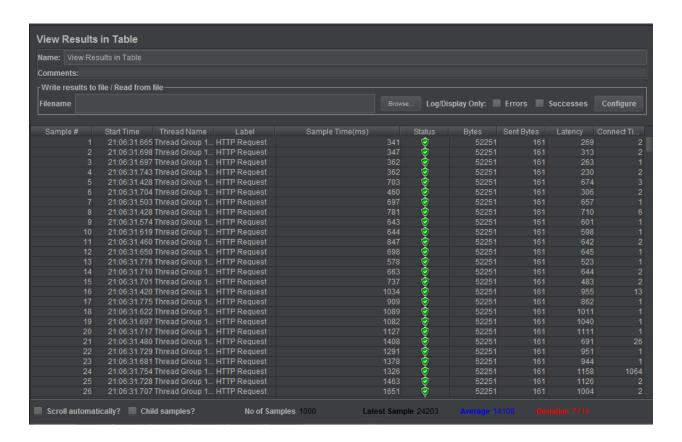
Sample time dari thread yang terakhir mengakses adalah 59 ms atau sekitar 0.059 second. Dengan average sample time 91 ms (0.09 second). Hasil ini menunjukkan bahwa dengan 100 akses secara syncronous response time aplikasi masih sangat cepat

2. Test plan dengan Number of Thread 500 Url yang di hit: localhost:8080/mahasiswa/cari3?univ=3&fakultas=16&prodi=29



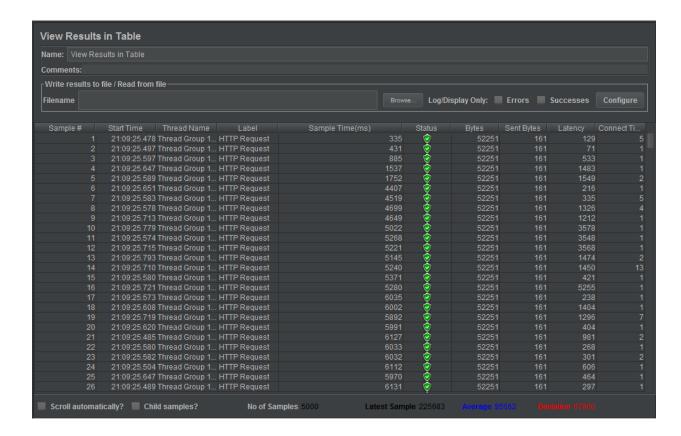
Sample time dari thread yang terakhir mengakses adalah 59 ms atau sekitar 12.61 second. Dengan average sample time 8522 ms (8.5 second). Hasil ini menunjukkan bahwa dengan 500 akses secara syncronous response time aplikasi masih sangat cepat

3. Test plan dengan Number of Thread 1000 Url yang di hit: localhost:8080/mahasiswa/cari3?univ=3&fakultas=16&prodi=29



Sample time dari thread yang terakhir mengakses adalah 59 ms atau sekitar 24 second. Dengan average sample time 8522 ms (14.1 second). Hasil ini menunjukkan bahwa dengan 1000 akses secara syncronous response time aplikasi masih sangat cepat

4. Test plan dengan Number of Thread 5000 Url yang di hit: localhost:8080/mahasiswa/cari3?univ=3&fakultas=16&prodi=29



Sample time dari thread yang terakhir mengakses adalah 225 second atau sekitar 3 menit. Dengan average sample time 95656 ms (95 second). Hasil ini menunjukkan bahwa dengan 5000 akses secara syncronous response time aplikasi masih sangat cepat

Dari hasil Stress test tersebut dapat dilihat bahwa waktu paling lama yang digunakan server untuk fully serve the request adalah sekitar 3 menit ketika akses user ke 5000. Menurut saya hasil ini masih cukup baik, yang berarti bahwa aplikasi masih tergolong reliable. Selain itu karena aplikasi juga tidak melakukan proses pengolahan yang kompleks dan penggunaan resource yang banyak. Jadi angka 3 menit masih sangat normal.