

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 request mapping di controller, menambah query di mapper dan menambah method di interface service dan class yang implement untuk memanggil fungsi di mapper tadi. Begitu selanjutnya.

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: <http://localhost:8080/mahasiswa/ubah/141021354001>

Fitur Lain yang Mendukung Aplikasi

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

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	<input type="text"/>	This is a required field
Tempat Lahir	<input type="text"/>	This is a required field
Tanggal Lahir	<input type="text" value="ex:1994-05-05"/>	This is a required field
Jenis Kelamin	<input type="text" value="Laki-Laki"/>	
Agama	<input type="text" value="Protestan"/>	
Golongan Darah	<input type="text" value="AB-"/>	
Tahun Masuk	<input type="text"/>	This is a required field
Jalur Masuk	<input type="text" value="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 execution 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

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Ti...
1	21:02:54.268	Thread Group 1...	HTTP Request	217	✓	52251	161	165	55
2	21:02:54.370	Thread Group 1...	HTTP Request	123	✓	52251	161	76	2
3	21:02:54.473	Thread Group 1...	HTTP Request	91	✓	52251	161	65	2
4	21:02:54.613	Thread Group 1...	HTTP Request	86	✓	52251	161	57	6
5	21:02:54.703	Thread Group 1...	HTTP Request	99	✓	52251	161	54	2
6	21:02:54.784	Thread Group 1...	HTTP Request	75	✓	52251	161	49	2
7	21:02:54.879	Thread Group 1...	HTTP Request	73	✓	52251	161	48	2
8	21:02:54.992	Thread Group 1...	HTTP Request	103	✓	52251	161	77	1
9	21:02:55.102	Thread Group 1...	HTTP Request	76	✓	52251	161	52	2
10	21:02:55.180	Thread Group 1...	HTTP Request	78	✓	52251	161	44	3
11	21:02:55.290	Thread Group 1...	HTTP Request	68	✓	52251	161	41	2
12	21:02:55.394	Thread Group 1...	HTTP Request	68	✓	52251	161	42	1
13	21:02:55.494	Thread Group 1...	HTTP Request	87	✓	52251	161	51	1
14	21:02:55.577	Thread Group 1...	HTTP Request	76	✓	52251	161	51	2
15	21:02:55.678	Thread Group 1...	HTTP Request	83	✓	52251	161	71	2
16	21:02:55.794	Thread Group 1...	HTTP Request	135	✓	52251	161	93	2
17	21:02:55.894	Thread Group 1...	HTTP Request	98	✓	52251	161	54	2
18	21:02:55.996	Thread Group 1...	HTTP Request	106	✓	52251	161	72	2
19	21:02:56.094	Thread Group 1...	HTTP Request	98	✓	52251	161	52	2
20	21:02:56.198	Thread Group 1...	HTTP Request	135	✓	52251	161	67	2
21	21:02:56.308	Thread Group 1...	HTTP Request	139	✓	52251	161	80	2
22	21:02:56.393	Thread Group 1...	HTTP Request	96	✓	52251	161	69	2
23	21:02:56.493	Thread Group 1...	HTTP Request	69	✓	52251	161	44	2
24	21:02:56.593	Thread Group 1...	HTTP Request	78	✓	52251	161	42	1
25	21:02:56.708	Thread Group 1...	HTTP Request	79	✓	52251	161	47	2
26	21:02:56.797	Thread Group 1...	HTTP Request	108	✓	52251	161	56	2

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 59 Average 91 Deviation 33

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 synchronous 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

View Results in Table

Name:View Results in Table

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:

ErrorsSuccesses

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Ti...
1	21:04:42.335	Thread Group 1...	HTTP Request	358	<div></div>	52251	161	316	24
2	21:04:42.345	Thread Group 1...	HTTP Request	416	<div></div>	52251	161	351	16
3	21:04:42.569	Thread Group 1...	HTTP Request	450	<div></div>	52251	161	391	2
4	21:04:42.544	Thread Group 1...	HTTP Request	499	<div></div>	52251	161	316	1
5	21:04:42.586	Thread Group 1...	HTTP Request	569	<div></div>	52251	161	527	1
6	21:04:42.731	Thread Group 1...	HTTP Request	430	<div></div>	52251	161	392	5
7	21:04:42.712	Thread Group 1...	HTTP Request	787	<div></div>	52251	161	495	1
8	21:04:42.436	Thread Group 1...	HTTP Request	1299	<div></div>	52251	161	745	2
9	21:04:42.735	Thread Group 1...	HTTP Request	1007	<div></div>	52251	161	183	2
10	21:04:42.907	Thread Group 1...	HTTP Request	835	<div></div>	52251	161	434	17
11	21:04:42.944	Thread Group 1...	HTTP Request	898	<div></div>	52251	161	304	2
12	21:04:42.934	Thread Group 1...	HTTP Request	913	<div></div>	52251	161	311	2
13	21:04:42.617	Thread Group 1...	HTTP Request	1235	<div></div>	52251	161	534	1
14	21:04:42.623	Thread Group 1...	HTTP Request	1265	<div></div>	52251	161	800	1
15	21:04:43.043	Thread Group 1...	HTTP Request	909	<div></div>	52251	161	595	14
16	21:04:42.964	Thread Group 1...	HTTP Request	1035	<div></div>	52251	161	628	2
17	21:04:42.731	Thread Group 1...	HTTP Request	1283	<div></div>	52251	161	815	2
18	21:04:42.449	Thread Group 1...	HTTP Request	1572	<div></div>	52251	161	1378	2
19	21:04:43.688	Thread Group 1...	HTTP Request	362	<div></div>	52251	161	212	2
20	21:04:42.379	Thread Group 1...	HTTP Request	1689	<div></div>	52251	161	1479	181
21	21:04:43.121	Thread Group 1...	HTTP Request	977	<div></div>	52251	161	719	3
22	21:04:43.105	Thread Group 1...	HTTP Request	1034	<div></div>	52251	161	773	2
23	21:04:43.012	Thread Group 1...	HTTP Request	1125	<div></div>	52251	161	526	1
24	21:04:42.734	Thread Group 1...	HTTP Request	1422	<div></div>	52251	161	1311	2
25	21:04:42.322	Thread Group 1...	HTTP Request	1880	<div></div>	52251	161	537	35
26	21:04:43.918	Thread Group 1...	HTTP Request	381	<div></div>	52251	161	336	1

Scroll automatically?

Child samples?

No of Samples 500

Latest Sample 12610

Average 8522

Deviation 4367

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 synchronous 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

View Results in Table

Name:

View Results in Table

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:

☐ Errors

☐ Successes

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Ti...
1	21:06:31.665	Thread Group 1...	HTTP Request	341		52251	161	269	2
2	21:06:31.698	Thread Group 1...	HTTP Request	347		52251	161	313	2
3	21:06:31.697	Thread Group 1...	HTTP Request	362		52251	161	263	1
4	21:06:31.743	Thread Group 1...	HTTP Request	362		52251	161	230	2
5	21:06:31.428	Thread Group 1...	HTTP Request	703		52251	161	674	3
6	21:06:31.704	Thread Group 1...	HTTP Request	460		52251	161	306	2
7	21:06:31.503	Thread Group 1...	HTTP Request	697		52251	161	657	1
8	21:06:31.428	Thread Group 1...	HTTP Request	781		52251	161	710	6
9	21:06:31.574	Thread Group 1...	HTTP Request	643		52251	161	601	1
10	21:06:31.619	Thread Group 1...	HTTP Request	644		52251	161	598	1
11	21:06:31.460	Thread Group 1...	HTTP Request	847		52251	161	642	2
12	21:06:31.650	Thread Group 1...	HTTP Request	698		52251	161	645	1
13	21:06:31.776	Thread Group 1...	HTTP Request	578		52251	161	523	1
14	21:06:31.710	Thread Group 1...	HTTP Request	663		52251	161	644	2
15	21:06:31.701	Thread Group 1...	HTTP Request	737		52251	161	483	2
16	21:06:31.420	Thread Group 1...	HTTP Request	1034		52251	161	955	13
17	21:06:31.775	Thread Group 1...	HTTP Request	909		52251	161	862	1
18	21:06:31.622	Thread Group 1...	HTTP Request	1089		52251	161	1011	1
19	21:06:31.697	Thread Group 1...	HTTP Request	1082		52251	161	1040	1
20	21:06:31.717	Thread Group 1...	HTTP Request	1127		52251	161	1111	1
21	21:06:31.480	Thread Group 1...	HTTP Request	1408		52251	161	691	26
22	21:06:31.729	Thread Group 1...	HTTP Request	1291		52251	161	951	1
23	21:06:31.681	Thread Group 1...	HTTP Request	1378		52251	161	944	1
24	21:06:31.754	Thread Group 1...	HTTP Request	1326		52251	161	1158	1064
25	21:06:31.728	Thread Group 1...	HTTP Request	1463		52251	161	1126	2
26	21:06:31.707	Thread Group 1...	HTTP Request	1651		52251	161	1004	2

☐ Scroll automatically?

☐ Child samples?

No of Samples 1000

Latest Sample 24203

Average 14100

Deviation 7710

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 synchronous 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

View Results in Table

Name:View Results in Table

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:

ErrorsSuccesses

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Ti...
1	21:09:25.478	Thread Group 1...	HTTP Request	335	✓	52251	161	129	5
2	21:09:25.497	Thread Group 1...	HTTP Request	431	✓	52251	161	71	1
3	21:09:25.597	Thread Group 1...	HTTP Request	885	✓	52251	161	533	1
4	21:09:25.647	Thread Group 1...	HTTP Request	1537	✓	52251	161	1483	1
5	21:09:25.589	Thread Group 1...	HTTP Request	1752	✓	52251	161	1549	2
6	21:09:25.651	Thread Group 1...	HTTP Request	4407	✓	52251	161	216	1
7	21:09:25.583	Thread Group 1...	HTTP Request	4519	✓	52251	161	335	5
8	21:09:25.578	Thread Group 1...	HTTP Request	4699	✓	52251	161	1326	4
9	21:09:25.713	Thread Group 1...	HTTP Request	4649	✓	52251	161	1212	1
10	21:09:25.779	Thread Group 1...	HTTP Request	5022	✓	52251	161	3578	1
11	21:09:25.574	Thread Group 1...	HTTP Request	5268	✓	52251	161	3548	1
12	21:09:25.715	Thread Group 1...	HTTP Request	5221	✓	52251	161	3568	1
13	21:09:25.793	Thread Group 1...	HTTP Request	5145	✓	52251	161	1474	2
14	21:09:25.710	Thread Group 1...	HTTP Request	5240	✓	52251	161	1450	13
15	21:09:25.580	Thread Group 1...	HTTP Request	5371	✓	52251	161	421	1
16	21:09:25.721	Thread Group 1...	HTTP Request	5280	✓	52251	161	5255	1
17	21:09:25.573	Thread Group 1...	HTTP Request	6035	✓	52251	161	238	1
18	21:09:25.608	Thread Group 1...	HTTP Request	6002	✓	52251	161	1404	1
19	21:09:25.719	Thread Group 1...	HTTP Request	5892	✓	52251	161	1296	7
20	21:09:25.620	Thread Group 1...	HTTP Request	5991	✓	52251	161	404	1
21	21:09:25.485	Thread Group 1...	HTTP Request	6127	✓	52251	161	981	2
22	21:09:25.580	Thread Group 1...	HTTP Request	6033	✓	52251	161	268	1
23	21:09:25.582	Thread Group 1...	HTTP Request	6032	✓	52251	161	301	2
24	21:09:25.504	Thread Group 1...	HTTP Request	6112	✓	52251	161	606	1
25	21:09:25.647	Thread Group 1...	HTTP Request	5970	✓	52251	161	464	1
26	21:09:25.489	Thread Group 1...	HTTP Request	6131	✓	52251	161	297	1

Scroll automatically?

Child samples?

No of Samples 5000

Latest Sample 225683

Average 95662

Deviation 67800

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 synchronous 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.