

A. Proses saya dalam mengembangkan tugas ini

Pada pengerjaan tugas ini, saya memulai dengan membuat file-file untuk *model*, *service*, dan *mapper(dao)* untuk semua tabel yang ada pada *database* (semua file masih kosong kecuali *model*, karena model dibuat berdasarkan tabel-tabel yang ada pada *database*), lalu saya membuat *controller* untuk penduduk. Kemudian selanjutnya saya mengerjakannya per fitur, mulai dari fitur pertama hingga fitur ke delapan.

Setiap pengerjaan fitur dimulai dari membuat *query* yang dibutuhkan untuk fitur tersebut, lalu dilanjutkan dibuat *methodnya* di *interface service* yang diimplementasikan pada *class servicedatabase*. Selanjutnya dibuat file html yang dibutuhkan di folder *templates*. Lalu pada *controller* dibuat *dao* yang berkaitan dan dipanggil *methodnya*, lalu tambahkan atribut yang dibutuhkan di *tempates* melalui *model*.

Pada tugas ini terdapat 5 *model* yaitu PendudukModel, KeluargaModel, KelurahanModel, KecamatanModel, dan KotaModel. Untuk *service* dibuat untuk semua *model*. Lalu *mapper* juga ada 5, yaitu untuk mengakses setiap tabel terdapat mappernya masing-masing (PendudukMapper, KeluargaMapper, KelurahanMapper, KecamatanMapper, KotaMapper). Untuk *controller* nya, terdapat 2 *controller* yaitu PendudukController dan KeluargaController.

Contoh untuk mengembangkan fitur 1, pada PendudukMapper dibuat *query* untuk mencari penduduk dengan nik tertentu.

```
//Untuk mencari data seorang penduduk
@Select("select * from penduduk p, kota ko, kecamatan kec, kelurahan kel, keluarga k " +
        "where ko.id = kec.id_kota and kec.id = kel.id_kecamatan and " +
        "kel.id = k.id_kelurahan and p.id_keluarga = k.id and p.nik = #{nik}")
PendudukModel selectPenduduk(@Param("nik") String nik);
```

Lalu pada *interface* PendudukService dibuat *method* :

```
public interface PendudukService {
    PendudukModel selectPenduduk(String nik);
}
```

Dan diimplementasikan di *class* PendudukServiceDatabase yang mengakses mapper dengan query yang sudah dibuat :

```
@Autowired
private PendudukMapper pendudukMapper;

@Override
public PendudukModel selectPenduduk(String nik) {
    log.info("select penduduk with nik {}", nik);
    return pendudukMapper.selectPenduduk(nik);
}
```

Pada PendudukController dibuat seperti berikut :

```
@RequestMapping("/penduduk")
public String viewPenduduk(Model model,
    @RequestParam(value = "nik", required = false) String nik)
{
    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
    if(penduduk != null) {
        model.addAttribute(s: "penduduk", penduduk);
        return "view-penduduk";
    } else {
        model.addAttribute(s: "nik", nik);
        return "not-found-penduduk";
    }
}
```

Lalu membuat *template-template* yang dibutuhkan. Terkadang dalam pengerjaan saya membuat *template* dahulu baru melengkapi *controller*-nya. Untuk pengerjaan fitur-fitur lain step-step yang saya lakukan kurang lebih sama dengan step-step pada pengerjaan fitur 1 ini.

B. Optimasi Database

Untuk optimasi database yang saya lakukan adalah menambahkan *primary key* pada tiap-tiap tabel melalui phpmyadmin. Pada tabel penduduk dan keluarga saya menjadikan nik dan nkk sebagai *primary key*, hal ini sangat mempercepat proses pencarian penduduk untuk fitur pertama (*view-penduduk*) serta mempercepat proses pencarian keluarga untuk fitur kedua (*view keluarga*).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None			Change Drop Primary Unique Index More
2	nik	char(16)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index More

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None			Change Drop Primary Unique Index Spatial More
2	nomor_kk	char(16)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial More

Lalu selain itu, untuk table kelurahan, kecamatan, dan kota, saya menjadikan kolom id masing-masing tabel sebagai *primary key*. Hal ini bisa mempercepat jalanny fitur 8 (lihat penduduk yang ada di suatu kelurahan) yang menggunakan query dari banyak tabel.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None			Change Drop Primary Unique Index Spatial More
2	id_kecamatan	bigint(20)		UNSIGNED	No	None			Change Drop Primary Unique Index Spatial More

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None			Change Drop Primary Unique Index Spatial More
2	id_kota	bigint(20)		UNSIGNED	No	None			Change Drop Primary Unique Index Spatial More

Server: 127.0.0.1 » Database: siduk_dki » Table: kota

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None			Change Drop Primary Unique Index Spatial Fulltext More
2	kode_kota	char(4)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext More

C. Stress testing

Untuk *stress testing* digunakan apache jmeter. Berikut uji coba jmeter untuk 3 fitur pada tugas ini yaitu untuk fitur 1 (menampilkan data penduduk berdasarkan nik), fitur 2 (menampilkan data keluarga berdasarkan nkk) dan fitur 8 (menampilkan *list* penduduk di suatu kelurahan, kecamatan, kota tertentu).

a. Untuk Fitur 1 yaitu menampilkan data-data penduduk berdasarkan nik

HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8080

HTTP Request

Method: GET Path: penduduk?nik=3101010101690001 Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

Jumlah *thread* sedikit :

Thread Properties

Number of Threads (users): 50

Ramp-Up Period (in seconds): 10

Loop Count: ☐ Forever 1

☐ Delay Thread creation until needed

☐ Scheduler

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
23	21:10:00.404	Thread Group ...	HTTP Request	9	✓	5673	147	9	1
43	21:10:04.413	Thread Group ...	HTTP Request	11	✓	5673	147	10	1
44	21:10:04.616	Thread Group ...	HTTP Request	11	✓	5673	147	11	1
22	21:10:00.205	Thread Group ...	HTTP Request	13	✓	5673	147	13	1
33	21:10:02.415	Thread Group ...	HTTP Request	13	✓	5673	147	13	1
45	21:10:04.816	Thread Group ...	HTTP Request	13	✓	5673	147	12	1
18	21:09:59.403	Thread Group ...	HTTP Request	14	✓	5673	147	14	1
16	21:09:59.015	Thread Group ...	HTTP Request	15	✓	5673	147	15	2
6	21:09:56.999	Thread Group ...	HTTP Request	18	✓	5673	147	18	2
1	21:09:56.010	Thread Group ...	HTTP Request	19	✓	5673	147	19	4
50	21:10:05.819	Thread Group ...	HTTP Request	19	✓	5673	147	18	1
17	21:09:59.237	Thread Group ...	HTTP Request	20	✓	5673	147	20	2
32	21:10:02.215	Thread Group ...	HTTP Request	20	✓	5673	147	20	1
42	21:10:04.215	Thread Group ...	HTTP Request	20	✓	5673	147	20	1
5	21:09:56.813	Thread Group ...	HTTP Request	23	✓	5673	147	23	2
26	21:10:01.007	Thread Group ...	HTTP Request	25	✓	5673	147	25	2
36	21:10:03.016	Thread Group ...	HTTP Request	25	✓	5673	147	25	2
2	21:09:56.199	Thread Group ...	HTTP Request	26	✓	5673	147	25	2
12	21:09:58.215	Thread Group ...	HTTP Request	26	✓	5673	147	26	2
34	21:10:02.608	Thread Group ...	HTTP Request	26	✓	5673	147	25	1
13	21:09:58.416	Thread Group ...	HTTP Request	27	✓	5673	147	27	2
47	21:10:05.779	Thread Group ...	HTTP Request	27	✓	5673	147	27	1
35	21:10:02.809	Thread Group ...	HTTP Request	28	✓	5673	147	28	2
21	21:10:00.016	Thread Group ...	HTTP Request	29	✓	5673	147	29	2
39	21:10:03.816	Thread Group ...	HTTP Request	29	✓	5673	147	29	1
41	21:10:04.017	Thread Group ...	HTTP Request	29	✓	5673	147	28	2
46	21:10:05.021	Thread Group ...	HTTP Request	29	✓	5673	147	29	2

☐ Scroll automatically? ☐ Child samples? No of Samples 50 Latest Sample 19 Average 33 Deviation 18

Jumlah *thread* banyak :

Thread Properties

Number of Threads (users): 500

Ramp-Up Period (in seconds): 10

Loop Count: ☐ Forever 1

☐ Delay Thread creation until needed

☐ Scheduler

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
327	21:13:33.453	Thread Group ...	HTTP Request	7	✓	5673	147	7	0
329	21:13:33.475	Thread Group ...	HTTP Request	7	✓	5673	147	7	0
330	21:13:33.495	Thread Group ...	HTTP Request	7	✓	5673	147	7	0
398	21:13:34.904	Thread Group ...	HTTP Request	7	✓	5673	147	7	0
399	21:13:34.928	Thread Group ...	HTTP Request	7	✓	5673	147	7	0
419	21:13:35.328	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
439	21:13:35.728	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
465	21:13:36.239	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
466	21:13:36.260	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
468	21:13:36.300	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
470	21:13:36.343	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
472	21:13:36.382	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
475	21:13:36.447	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
478	21:13:36.503	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
496	21:13:36.860	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
498	21:13:36.902	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
499	21:13:36.927	Thread Group ...	HTTP Request	7	✓	5673	147	7	1
73	21:13:28.220	Thread Group ...	HTTP Request	8	✓	5673	147	8	1
164	21:13:30.075	Thread Group ...	HTTP Request	8	✓	5673	147	8	0
176	21:13:30.315	Thread Group ...	HTTP Request	8	✓	5673	147	8	0
178	21:13:30.336	Thread Group ...	HTTP Request	8	✓	5673	147	8	1
180	21:13:30.402	Thread Group ...	HTTP Request	8	✓	5673	147	8	0
181	21:13:30.418	Thread Group ...	HTTP Request	8	✓	5673	147	8	0
182	21:13:30.429	Thread Group ...	HTTP Request	8	✓	5673	147	8	0
183	21:13:30.466	Thread Group ...	HTTP Request	8	✓	5673	147	8	1
184	21:13:30.484	Thread Group ...	HTTP Request	8	✓	5673	147	8	1
191	21:13:30.626	Thread Group ...	HTTP Request	8	✓	5673	147	8	0

☐ Scroll automatically? ☐ Child samples? No of Samples 500 Latest Sample 9 Average 37 Deviation 55

b. Untuk fitur 2 yaitu menampilkan data keluarga berdasarkan nkk

Basic	Advanced
Web Server	
Protocol [http]:	Server Name or IP: localhost Port Number: 8080
HTTP Request	
Method: GET	Path: keluarga?nkk=3171092110170001 Content encoding:
<input type="checkbox"/> Redirect Automatically <input checked="" type="checkbox"/> Follow Redirects <input checked="" type="checkbox"/> Use Keep-Alive <input type="checkbox"/> Use multipart/form-data for POST <input type="checkbox"/> Browser-compatible headers	

Jumlah *thread* sedikit :

Thread Properties										
Number of Threads (users):		50								
Ramp-Up Period (in seconds):		10								
Loop Count:		<input type="checkbox"/> Forever 1								

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
2	21:34:47.653	Thread Group ...	HTTP Request	434	✓	7886	147	434	1
6	21:34:47.854	Thread Group ...	HTTP Request	487	✓	7886	147	487	1
4	21:34:47.453	Thread Group ...	HTTP Request	639	✓	7886	147	639	1
17	21:34:50.656	Thread Group ...	HTTP Request	656	✓	7886	147	656	1
8	21:34:48.453	Thread Group ...	HTTP Request	728	✓	7886	147	728	1
46	21:34:56.060	Thread Group ...	HTTP Request	736	✓	7886	147	736	0
1	21:34:47.253	Thread Group ...	HTTP Request	833	✓	7886	147	833	1
16	21:34:50.456	Thread Group ...	HTTP Request	854	✓	7886	147	854	1
12	21:34:49.455	Thread Group ...	HTTP Request	874	✓	7886	147	874	2
9	21:34:48.253	Thread Group ...	HTTP Request	929	✓	7886	147	929	1
44	21:34:55.859	Thread Group ...	HTTP Request	934	✓	7886	147	934	0
21	21:34:51.656	Thread Group ...	HTTP Request	968	✓	7886	147	968	1
5	21:34:47.054	Thread Group ...	HTTP Request	1050	✓	7886	147	1050	1
18	21:34:50.255	Thread Group ...	HTTP Request	1058	✓	7886	147	1057	1
14	21:34:49.256	Thread Group ...	HTTP Request	1078	✓	7886	147	1078	1
40	21:34:55.659	Thread Group ...	HTTP Request	1122	✓	7886	147	1122	0
7	21:34:48.053	Thread Group ...	HTTP Request	1125	✓	7886	147	1125	1
25	21:34:51.456	Thread Group ...	HTTP Request	1181	✓	7886	147	1181	1
49	21:34:56.659	Thread Group ...	HTTP Request	1184	✓	7886	147	1184	1
36	21:34:54.458	Thread Group ...	HTTP Request	1213	✓	7886	147	1213	1
3	21:34:46.853	Thread Group ...	HTTP Request	1238	✓	7886	147	1238	2
20	21:34:50.055	Thread Group ...	HTTP Request	1269	✓	7886	147	1269	1
11	21:34:49.053	Thread Group ...	HTTP Request	1274	✓	7886	147	1274	1
27	21:34:52.056	Thread Group ...	HTTP Request	1304	✓	7886	147	1304	1
42	21:34:55.459	Thread Group ...	HTTP Request	1333	✓	7886	147	1333	0
23	21:34:51.256	Thread Group ...	HTTP Request	1369	✓	7886	147	1369	1
48	21:34:56.459	Thread Group ...	HTTP Request	1384	✓	7886	147	1384	1

☐ Scroll automatically? ☐ Child samples? No of Samples 50 Latest Sample 1585 Average 1351 Deviation 458

Jumlah *thread* banyak :

Thread Properties									
Number of Threads (users):		500							
Ramp-Up Period (in seconds):		10							
Loop Count:		<input type="checkbox"/> Forever <input checked="" type="checkbox"/> 1							

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
11	21:39:02.528	Thread Group ...	HTTP Request	1461	✓	7886	147	1461	1
14	21:39:02.503	Thread Group ...	HTTP Request	1509	✓	7886	147	1509	1
18	21:39:02.495	Thread Group ...	HTTP Request	1530	✓	7886	147	1530	1
12	21:39:02.456	Thread Group ...	HTTP Request	1544	✓	7886	147	1543	0
7	21:39:02.415	Thread Group ...	HTTP Request	1553	✓	7886	147	1553	1
2	21:39:02.379	Thread Group ...	HTTP Request	1568	✓	7886	147	1568	1
6	21:39:02.368	Thread Group ...	HTTP Request	1591	✓	7886	147	1591	1
8	21:39:02.376	Thread Group ...	HTTP Request	1603	✓	7886	147	1603	0
5	21:39:02.335	Thread Group ...	HTTP Request	1624	✓	7886	147	1624	1
4	21:39:02.295	Thread Group ...	HTTP Request	1656	✓	7886	147	1656	1
9	21:39:02.295	Thread Group ...	HTTP Request	1685	✓	7886	147	1685	1
1	21:39:02.236	Thread Group ...	HTTP Request	1707	✓	7886	147	1707	1
3	21:39:02.232	Thread Group ...	HTTP Request	1715	✓	7886	147	1715	1
24	21:39:02.288	Thread Group ...	HTTP Request	1772	✓	7886	147	1772	1
10	21:39:02.191	Thread Group ...	HTTP Request	1789	✓	7886	147	1789	1
15	21:39:02.211	Thread Group ...	HTTP Request	1808	✓	7886	147	1807	1
21	21:39:02.151	Thread Group ...	HTTP Request	1895	✓	7886	147	1895	1
19	21:39:02.131	Thread Group ...	HTTP Request	1899	✓	7886	147	1898	1
16	21:39:02.113	Thread Group ...	HTTP Request	1910	✓	7886	147	1910	1
17	21:39:02.111	Thread Group ...	HTTP Request	1913	✓	7886	147	1913	1
13	21:39:02.052	Thread Group ...	HTTP Request	1948	✓	7886	147	1948	0
20	21:39:02.071	Thread Group ...	HTTP Request	1972	✓	7886	147	1972	1
22	21:39:02.033	Thread Group ...	HTTP Request	2013	✓	7886	147	2013	13
23	21:39:02.031	Thread Group ...	HTTP Request	2024	✓	7886	147	2024	16
30	21:39:03.923	Thread Group ...	HTTP Request	4018	✓	7886	147	4018	1
62	21:39:04.022	Thread Group ...	HTTP Request	4079	✓	7886	147	4079	0
27	21:39:03.811	Thread Group ...	HTTP Request	4119	✓	7886	147	4119	1

☐ Scroll automatically? ☐ Child samples? No of Samples 500 Latest Sample 28593 Average 16656 Deviation 8293

c. Untuk fitur 8 yaitu menampilkan data penduduk berdasarkan kelurahan, kecamatan, kota

Web Server		
Protocol [http]:	Server Name or IP: localhost	Port Number: 8080
HTTP Request		
Method: GET	Path: penduduk/cari?kt=2&kc=11	Content encoding:
<input type="checkbox"/> Redirect Automatically	<input checked="" type="checkbox"/> Follow Redirects	<input checked="" type="checkbox"/> Use KeepAlive
<input type="checkbox"/> Use multipart/form-data for POST	<input type="checkbox"/> Browser-compatible headers	

Untuk *thread* sedikit :

Thread Properties									
Number of Threads (users):		50							
Ramp-Up Period (in seconds):		10							
Loop Count:		<input type="checkbox"/> Forever <input checked="" type="checkbox"/> 1							

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
38	21:55:59.766	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
46	21:56:01.366	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
49	21:56:01.967	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
8	21:55:53.761	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
9	21:55:53.960	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
29	21:55:57.965	Thread Group ...	HTTP Request	7	✓	2680	142	7	0
36	21:55:59.365	Thread Group ...	HTTP Request	7	✓	2680	142	7	0
45	21:56:01.166	Thread Group ...	HTTP Request	7	✓	2680	142	7	0
47	21:56:01.567	Thread Group ...	HTTP Request	7	✓	2680	142	7	0
5	21:55:53.160	Thread Group ...	HTTP Request	8	✓	2680	142	8	1
24	21:55:56.964	Thread Group ...	HTTP Request	8	✓	2680	142	8	0
28	21:55:57.764	Thread Group ...	HTTP Request	8	✓	2680	142	8	0
32	21:55:58.565	Thread Group ...	HTTP Request	8	✓	2680	142	8	0
37	21:55:59.565	Thread Group ...	HTTP Request	8	✓	2680	142	8	0
43	21:56:00.766	Thread Group ...	HTTP Request	8	✓	2680	142	8	0
16	21:55:55.363	Thread Group ...	HTTP Request	9	✓	2680	142	9	0
22	21:55:56.564	Thread Group ...	HTTP Request	9	✓	2680	142	8	0
30	21:55:58.165	Thread Group ...	HTTP Request	9	✓	2680	142	9	0
35	21:55:59.165	Thread Group ...	HTTP Request	9	✓	2680	142	8	0
40	21:56:00.738	Thread Group ...	HTTP Request	9	✓	2680	142	9	1
41	21:56:00.738	Thread Group ...	HTTP Request	9	✓	2680	142	9	1
48	21:56:01.767	Thread Group ...	HTTP Request	9	✓	2680	142	8	1
13	21:55:54.762	Thread Group ...	HTTP Request	10	✓	2680	142	10	1
19	21:55:55.962	Thread Group ...	HTTP Request	10	✓	2680	142	10	1
23	21:55:56.763	Thread Group ...	HTTP Request	10	✓	2680	142	9	1
31	21:55:58.365	Thread Group ...	HTTP Request	10	✓	2680	142	9	0
42	21:56:00.738	Thread Group ...	HTTP Request	10	✓	2680	142	10	1

☐ Scroll automatically? ☐ Child samples? No of Samples 50 Latest Sample 15 Average 15 Deviation 11

Untuk *thread* banyak :

Thread Properties

Number of Threads (users): 500

Ramp-Up Period (in seconds): 10

Loop Count: ☐ Forever 1

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename

Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
18	21:57:43.044	Thread Group ...	HTTP Request	6	✓	2680	142	6	1
30	21:57:43.285	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
92	21:57:44.525	Thread Group ...	HTTP Request	6	✓	2680	142	6	1
186	21:57:46.420	Thread Group ...	HTTP Request	6	✓	2680	142	6	1
295	21:57:48.630	Thread Group ...	HTTP Request	6	✓	2680	142	6	1
321	21:57:49.140	Thread Group ...	HTTP Request	6	✓	2680	142	6	1
425	21:57:51.230	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
458	21:57:51.913	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
461	21:57:51.960	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
499	21:57:52.735	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
500	21:57:52.742	Thread Group ...	HTTP Request	6	✓	2680	142	6	0
20	21:57:43.086	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
21	21:57:43.106	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
22	21:57:43.124	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
23	21:57:43.144	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
28	21:57:43.246	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
31	21:57:43.304	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
32	21:57:43.326	Thread Group ...	HTTP Request	7	✓	2680	142	7	0
33	21:57:43.346	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
34	21:57:43.364	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
38	21:57:43.443	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
41	21:57:43.506	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
44	21:57:43.565	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
53	21:57:43.745	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
56	21:57:43.805	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
57	21:57:43.825	Thread Group ...	HTTP Request	7	✓	2680	142	7	1
60	21:57:43.885	Thread Group ...	HTTP Request	7	✓	2680	142	7	1

☐ Scroll automatically? ☐ Child samples? No of Samples 500 Latest Sample 6 Average 22 Deviation 27

Jika dilihat dari hasil eksperimen menggunakan jmeter tersebut, kecepatannya sudah cukup cepat untuk fitur 1 dan 8, yaitu tidak sampai 1 detik. Dan untuk fitur 2 mungkin masih kurang cepat ketika yang mengakses dalam jumlah besar.