### A. Proses saya dalam mengembangkan tugas ini

Pada pengerkaan 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 *method*nya 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). Untun *controller* nya, terdapat 2 controller yaitu PendudukController dan KeluargaController.

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

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);
}
```

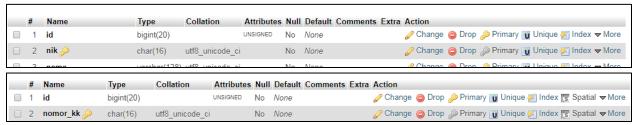
Pada PendudukController dibuat seperti berikut :

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.

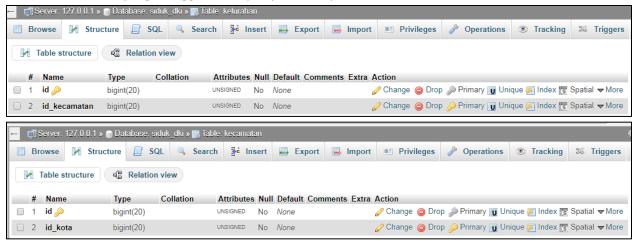
Kelas: B

# 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)*.



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.

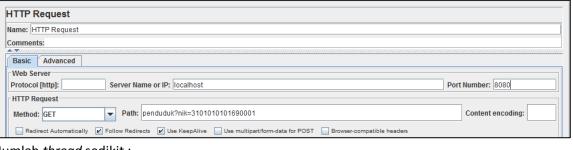




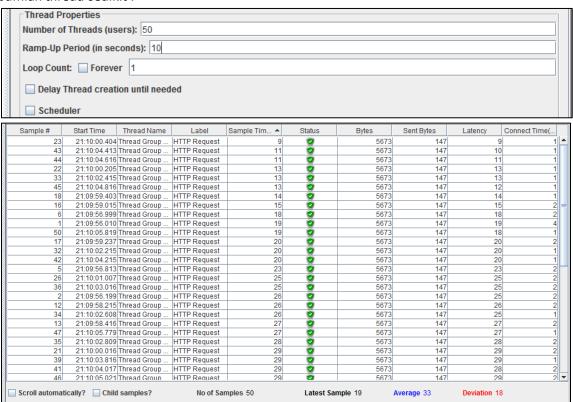
## C. Stress testing

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



#### Jumlah thread sedikit:



Jumlah thread banyak:

Thread Propertie	s								
Number of Thread	Is (users): 500								
Ramp-Up Period (	in seconds): 10	)							
Loop Count: F	orever 1								
Delay Thread	creation until ne	eeded							
Scheduler									
			1						
Sample #	Start Time	Thread Name	Label	Sample Tim	Status	Bytes	Sent Bytes	Latency	Connect Time(
327		Thread Group	HTTP Request	7	•	5673	147	7	
329		Thread Group	HTTP Request	7	<b>©</b>	5673	147	7	0
330		Thread Group	HTTP Request	7	<b>©</b>	5673	147	7	0
398		Thread Group	HTTP Request	7	9	5673	147		0
399		Thread Group	HTTP Request	7	<u> </u>	5673	147		0
419		Thread Group	HTTP Request	7	<u> </u>	5673	147 147		1
439		Thread Group	HTTP Request	7	<u> </u>	5673		7	1
465		Thread Group		7	0	5673 5673	147 147	7	1
466 468		Thread Group Thread Group	HTTP Request HTTP Request	7		5673	147	7	1
470		Thread Group	HTTP Request	7	9	5673	147	7	
470		Thread Group	HTTP Request	7	ĕ	5673	147	7	1
475		Thread Group		7	ĕ	5673	147	7	1
478		Thread Group	HTTP Request	7	ŏ	5673	147	7	1
496		Thread Group	HTTP Request	7	ŏ	5673	147	7	1
498		Thread Group	HTTP Request	7		5673	147	7	1
499		Thread Group	HTTP Request	7	- 5	5673	147	7	1
73		Thread Group		8	- Š	5673	147	. 8	
164		Thread Group	HTTP Request	8	ŏ	5673	147	8	
176		Thread Group	HTTP Request	8	Ö	5673	147	8	0
178		Thread Group	HTTP Request	8	Ö	5673	147	8	
180	21:13:30.402	Thread Group	HTTP Request	8	Ö	5673	147	8	0
181	21:13:30.418	Thread Group	HTTP Request	8	0	5673	147	8	0
182	21:13:30.429	Thread Group	HTTP Request	8	0	5673	147	8	0
183	21:13:30.466	Thread Group	HTTP Request	8	0	5673	147	8	1
184		Thread Group	HTTP Request	8	0	5673	147	8	
191	21:13:30 626	Thread Group	HTTP Request	8	Ø	5673	147	8	0
Scroll automat	ically? 🔲 Chile	d samples?	No of Sar	nples 500	Latest Sa	mple 9	Average 37	Deviation 55	

b. Untuk fitur 2 yaitu menampilkan data keluarga berdasarkan nkk



## Jumlah thread sedikit:



34	7886 7886 7886 7886 7886 7886 7886 7886	147 147 147 147 147 147 147 147 147	434 487 639 656 728 735 833 854	1 1 1 1 1 0
39	7886 7886 7886 7886 7886 7886 7886 7886	147 147 147 147 147 147 147	639 656 728 735 833 854	1 1 1 0
39	7886 7886 7886 7886 7886 7886 7886	147 147 147 147 147 147	656 728 735 833 854	1 1 0
28	7886 7886 7886 7886 7886 7886	147 147 147 147 147	728 735 833 854	1 0 1
36	7886 7886 7886 7886 7886	147 147 147 147	735 833 854	0
33 <b>2</b> 54 <b>2</b> 74 <b>2</b>	7886 7886 7886 7886	147 147 147	833 854	1
54 <b>②</b> 74 <b>②</b> 29 <b>②</b>	7886 7886 7886	147 147	854	
74 <b>②</b> 29 <b>②</b>	7886 7886	147		
29	7886			1
29 🔮			874	2
34	7006	147	929	1
	/880	147	934	0
68	7886	147	968	1
50	7886	147	1050	
58	7886	147	1057	1
78	7886	147	1078	1
22	7886	147	1122	0
25	7886	147	1125	1
31	7886	147	1181	1
84	7886	147	1184	1
13	7886	147	1213	1
38	7886	147	1238	2
69	7886	147	1269	1
74	7886	147	1274	1
04	7886	147	1304	1
	7886	147	1333	0
33	7886	147	1369	
33 <b>②</b> 59 <b>②</b>	7886	147	1384	1
		69 <b>7886</b> 84 <b>7886</b>	69 7886 147 R4 7886 147	69 7886 147 1369 R4 7886 147 1384

Jumlah thread banyak:

Thread Properties										
Number of Threads (users): 500										
Ramp-Up Period (in seconds): 10										
Loop Count: Fo	Loop Count: Forever 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	<b>3</b>	7886	147	1461	1 🔺	
14	21:39:02.503	Thread Group	HTTP Request	1509	Ø	7886	147	1509	1 =	
18	21:39:02.495	Thread Group	HTTP Request	1530	0	7886	147	1530	1	
12	21:39:02.456	Thread Group	HTTP Request	1544	<b>②</b>	7886	147	1543	0	
7	21:39:02.415	Thread Group	HTTP Request	1553	<b>②</b>	7886	147	1553	1	
2	21:39:02.379	Thread Group	HTTP Request	1568	<b>②</b>	7886	147	1568	1	
6		Thread Group	HTTP Request	1591	<b>©</b>	7886	147	1591	1	
8		Thread Group	HTTP Request	1603	<b>©</b>	7886	147	1603	0	
5		Thread Group	HTTP Request	1624	<b>©</b>	7886	147	1624	1	
4		Thread Group	HTTP Request	1656	<b>©</b>	7886	147	1656		
9		Thread Group	HTTP Request	1685	9	7886	147	1685	1	
1		Thread Group	HTTP Request	1707	<b>©</b>	7886	147	1707	1	
3		Thread Group	HTTP Request	1715	9	7886	147	1715	1	
24		Thread Group	HTTP Request	1772	<b>9</b>	7886	147	1772	1	
10		Thread Group	HTTP Request	1789	<b>②</b>	7886	147	1789	1	
15		Thread Group	HTTP Request	1808	<b>9</b>	7886	147	1807	1	
21			HTTP Request	1895	<b>©</b>	7886	147	1895	1	
19		Thread Group	HTTP Request	1899	<b>Ø</b>	7886	147	1898		
16		Thread Group	HTTP Request	1910	Ø	7886	147	1910		
17		Thread Group	HTTP Request	1913	9	7886	147	1913		
13		Thread Group	HTTP Request	1948	9	7886	147	1948		
20		Thread Group	HTTP Request	1972	9	7886	147	1972	1	
22		Thread Group	HTTP Request	2013	9	7886	147	2013		
23		Thread Group	HTTP Request	2024	9	7886	147	2024		
30		Thread Group	HTTP Request	4018	9	7886	147	4018		
62		Thread Group	HTTP Request	4079	9	7886	147	4079		
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 Po	ort Number: 8080
HTTP Request	
Method: GET    ■ Path: penduduk/cari?kt=2&kc=11	Content encoding:
Redirect Automatically 🗹 Follow Redirects 🗹 Use KeepAlive 🗌 Use multipart/form-data for POST 🔲 Browser-compatible headers	

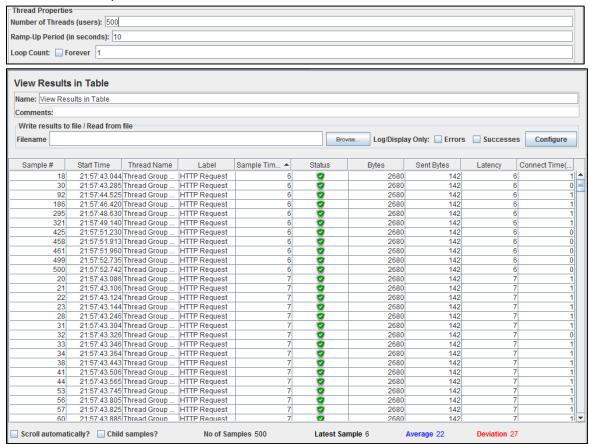
## Untuk thread sedikit:

Thread Properties
Number of Threads (users): 50
Ramp-Up Period (in seconds): 10
Loop Count: Forever 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	<b>②</b>	2680	142	6	0 4
46	21:56:01.366	Thread Group	HTTP Request	6	0	2680	142	6	0
49	21:56:01.967	Thread Group	HTTP Request	6	<b>②</b>	2680	142	6	0
8	21:55:53.761	Thread Group	HTTP Request	7	<b>②</b>	2680	142	7	1
9	21:55:53.960	Thread Group	HTTP Request	7	<b>②</b>	2680	142	7	1
29	21:55:57.965	Thread Group	HTTP Request	7	<b>②</b>	2680	142	7	0
36		Thread Group		7	<b>②</b>	2680	142	7	0
45	21:56:01.166	Thread Group	HTTP Request	7	<b>②</b>	2680	142	7	0 =
47		Thread Group		7	<b>©</b>	2680	142	7	0
5		Thread Group		8	<b>©</b>	2680	142	8	1
24		Thread Group		8	<b>©</b>	2680	142	8	0
28		Thread Group		8	<b>©</b>	2680	142	8	0
32		Thread Group		8	<b>©</b>	2680	142	8	
37		Thread Group		8	<b>9</b>	2680	142	8	
43		Thread Group		8	<b>9</b>	2680	142	8	
16		Thread Group		9	<b>3</b>	2680	142	9	0
22		Thread Group		9	<b>9</b>	2680	142	8	0
30		Thread Group		9	<b>9</b>	2680	142	9	0
35		Thread Group		9	<b>9</b>	2680	142	8	
40		Thread Group		9	<b>3</b>	2680	142	9	1
41		Thread Group		9	<b>9</b>	2680	142	9	1
48		Thread Group		9	<b>©</b>	2680	142	8	-
13			HTTP Request	10		2680	142	10	
19		Thread Group		10		2680	142	10	
23		Thread Group		10		2680	142	9	
31			HTTP Request	10		2680	142	9	
42	21:56:00 738	Thread Group	HTTP Request	10	•	2680	142	10	1 1
Scroll automati	cally? 🔲 Child	d samples?	No of Sar	nples 50	Latest Sai	mple 15	Average 15	Deviation 11	

Kelas : B

### Untuk thread banyak:



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.