

























## Tugas 1

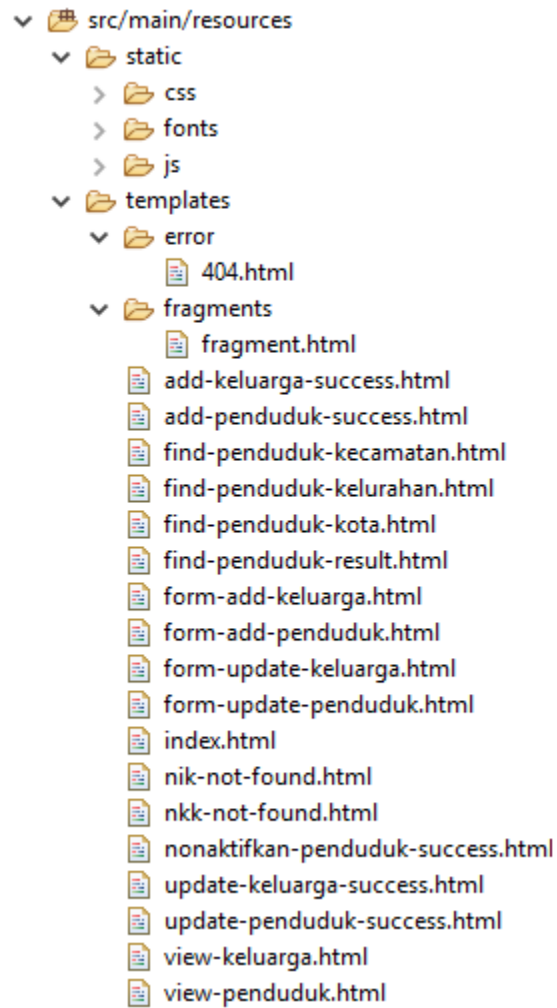
### Sistem Kependudukan Online Provinsi DKI Jakarta

#### A. **WRITE UP TUGAS 1**

Pada pengembangan tugas ini, saya membuat sebuah *project* dengan menggunakan konsep MVC dimana terdapat kelas *controller*, *mapper*, *service* dan kelas *model*. Pada tugas ini, saya membuat dua buah *controller*, yaitu *PendudukController* dan *KeluargaController*. *PendudukController* dibuat untuk meng-*handle* fitur-fitur yang berkaitan dengan data-data penduduk seperti fitur 1, 3, 5, 7 dan 8. Sedangkan *KeluargaController* dibuat untuk meng-*handle* fitur-fitur yang berkaitan dengan data-data keluarga seperti fitur 2, 4 dan 6. Pada *project* ini, saya juga membuat tiga buah kelas *mapper*, yaitu *PendudukMapper*, *KeluargaMapper* dan *WilayahMapper*. Kelas *PendudukMapper* dibuat untuk mengakses dan memanipulasi data-data yang terdapat dalam tabel *Penduduk* di dalam database. Kelas *KeluargaMapper* dibuat untuk megakses dan memanipulasi data-data dalam tabel *Keluarga* dalam database. Kelas *WilayahMapper* dibuat untuk mengakses dan memanipulasi data-data yang terdapat dalam tabel *kelurahan*, *kecamatan* dan *kota* yang ada di database. Sama halnya seperti kelas *mapper*, kelas *service* saya buat menggunakan konsep yang sama dimana terdapat tiga buah *interface* dan tiga buah kelas *service* dimana masing-masing satu untuk keluarga, penduduk dan wilayah. Untuk kelas *model*, saya membuat setiap kelas untuk masing-masing tabel yang ada di database siduk untuk merepresentasikan masing-masing dari tabel tersebut. Untuk lebih jelas mengenai struktur tersebut, dapat dilihat pada gambar berikut.

- ▼  APAPTugas1 [boot] [devtools]
  - >  Spring Elements
  - ▼  src/main/java
    - >  com.example
    - ▼  com.example.controller
      - >  KeluargaController.java
      - >  PendudukController.java
    - ▼  com.example.dao
      - >  KeluargaMapper.java
      - >  PendudukMapper.java
      - >  WilayahMapper.java
    - ▼  com.example.model
      - >  KecamatanModel.java
      - >  KeluargaModel.java
      - >  KelurahanModel.java
      - >  KotaModel.java
      - >  PendudukModel.java
    - ▼  com.example.service
      - >  KeluargaService.java
      - >  KeluargaServiceDatabase.java
      - >  PendudukService.java
      - >  PendudukServiceDatabase.java
      - >  WilayahService.java
      - >  WilayahServiceDatabase.java

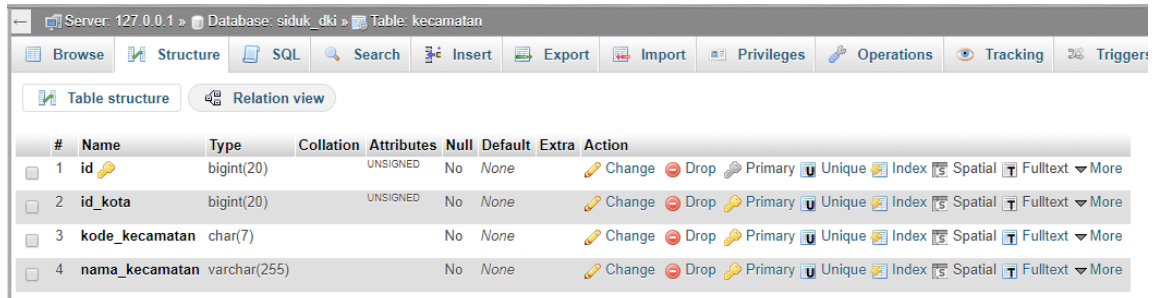
Sedangkan untuk halaman view, terdapat delapan belas halaman dengan gambaran struktur sebagai berikut.



## B. OPTIMASI DATABASE DAN STRESS TESTING

Untuk mempercepat eksekusi hasil *query*, saya meng-set atribut id sebagai *primary key* pada masing-masing tabel:

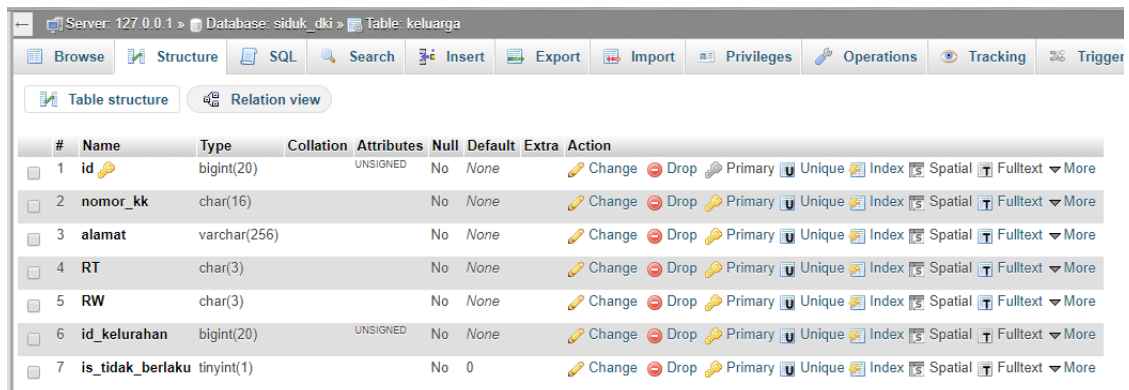
### 1. Tabel Kecamatan



The screenshot shows the MySQL Table Structure for the 'kecamatan' table. The table has four columns: 'id' (bigint(20), UNSIGNED, No Null, Primary Key), 'id\_kota' (bigint(20), UNSIGNED, No Null, Primary Key), 'kode\_kecamatan' (char(7), No Null, Primary Key), and 'nama\_kecamatan' (varchar(255), No Null, Primary Key). Each column has a 'Change' icon and a 'Drop' icon in the 'Action' column.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	id_kota	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	kode_kecamatan	char(7)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	nama_kecamatan	varchar(255)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

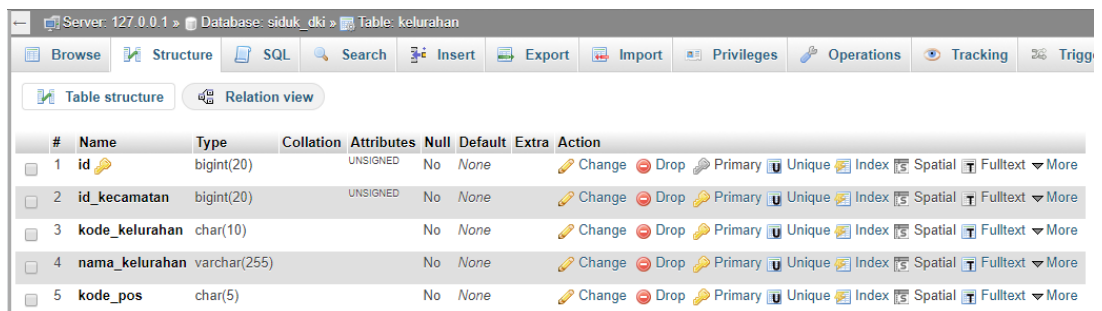
### 2. Tabel Keluarga



The screenshot shows the MySQL Table Structure for the 'keluarga' table. The table has seven columns: 'id' (bigint(20), UNSIGNED, No Null, Primary Key), 'nomor\_kk' (char(16), No Null, Primary Key), 'alamat' (varchar(256), No Null, Primary Key), 'RT' (char(3), No Null, Primary Key), 'RW' (char(3), No Null, Primary Key), 'id\_kelurahan' (bigint(20), UNSIGNED, No Null, Primary Key), and 'is\_tidak\_berlaku' (tinyint(1), No Null, Primary Key). Each column has a 'Change' icon and a 'Drop' icon in the 'Action' column.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	nomor_kk	char(16)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	alamat	varchar(256)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	RT	char(3)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
5	RW	char(3)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
6	id_kelurahan	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext More
7	is_tidak_berlaku	tinyint(1)			No	0		Change Drop Primary Unique Index Spatial Fulltext More

### 3. Tabel Kelurahan



The screenshot shows the MySQL Table Structure for the 'kelurahan' table. The table has five columns: 'id' (bigint(20), UNSIGNED, No Null, Primary Key), 'id\_kecamatan' (bigint(20), UNSIGNED, No Null, Primary Key), 'kode\_kelurahan' (char(10), No Null, Primary Key), 'nama\_kelurahan' (varchar(255), No Null, Primary Key), and 'kode\_pos' (char(5), No Null, Primary Key). Each column has a 'Change' icon and a 'Drop' icon in the 'Action' column.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	id_kecamatan	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	kode_kelurahan	char(10)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	nama_kelurahan	varchar(255)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
5	kode_pos	char(5)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

#### 4. Tabel Kota

Server: 127.0.0.1 » Database: siduk\_dki » Table: kota

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fu
2	kode_kota	char(4)			No	None		Change Drop Primary Unique Index Spatial Fu
3	nama_kota	varchar(255)			No	None		Change Drop Primary Unique Index Spatial Fu

#### 5. Tabel Penduduk

Server: 127.0.0.1 » Database: siduk\_dki » Table: penduduk

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	nik	char(16)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	nama	varchar(128)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	tempat_lahir	varchar(128)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
5	tanggal_lahir	date			No	None		Change Drop Primary Unique Index Spatial Fulltext More

Untuk *stress testing*, saya melakukan uji coba pada halaman localhost:8080/keluarga?nkk=3101010101050001. Hasil yang didapatkan cukup memuaskan karena data dapat diakses dengan tidak membutuhkan waktu lama, yaitu rata-rata sekitar 14 ms dengan deviasi sebesar 13 ms. Berikut hasil *stress testing* yang saya lakukan.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

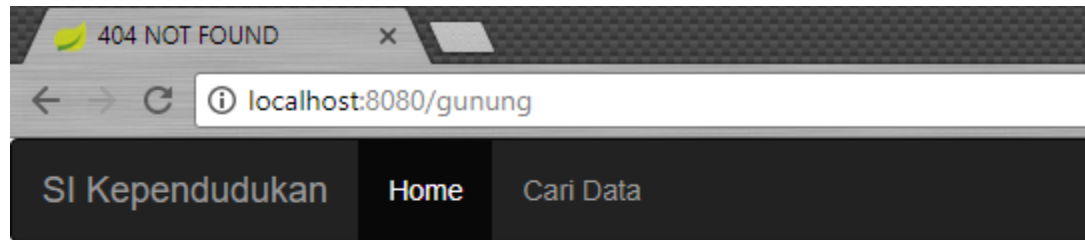
Filename
Browse...
Log/Display Only:
☐ Errors
☐ Successes
Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time...
974	20:15:08.349	Thread Group 1...	HTTP Request	9	✓	7511	234	0	3
975	20:15:08.360	Thread Group 1...	HTTP Request	11	✓	7511	234	5	1
976	20:15:08.369	Thread Group 1...	HTTP Request	10	✓	7511	234	4	1
977	20:15:08.380	Thread Group 1...	HTTP Request	18	✓	7511	234	5	1
978	20:15:08.400	Thread Group 1...	HTTP Request	18	✓	7511	234	9	4
979	20:15:08.409	Thread Group 1...	HTTP Request	17	✓	7511	234	8	2
980	20:15:08.391	Thread Group 1...	HTTP Request	35	✓	7511	234	13	2
981	20:15:08.421	Thread Group 1...	HTTP Request	15	✓	7511	234	4	1
982	20:15:08.430	Thread Group 1...	HTTP Request	17	✓	7511	234	9	2
983	20:15:08.440	Thread Group 1...	HTTP Request	13	✓	7511	234	5	2
984	20:15:08.451	Thread Group 1...	HTTP Request	14	✓	7511	234	6	2
985	20:15:08.460	Thread Group 1...	HTTP Request	15	✓	7511	234	5	2
986	20:15:08.470	Thread Group 1...	HTTP Request	10	✓	7511	234	4	2
987	20:15:08.491	Thread Group 1...	HTTP Request	11	✓	7511	234	4	2
988	20:15:08.502	Thread Group 1...	HTTP Request	13	✓	7511	234	6	2
989	20:15:08.500	Thread Group 1...	HTTP Request	39	✓	7511	234	9	3
990	20:15:08.521	Thread Group 1...	HTTP Request	18	✓	7511	234	13	9
991	20:15:08.528	Thread Group 1...	HTTP Request	18	✓	7511	234	11	7
992	20:15:08.542	Thread Group 1...	HTTP Request	10	✓	7511	234	3	1
993	20:15:08.553	Thread Group 1...	HTTP Request	12	✓	7511	234	6	2
994	20:15:08.561	Thread Group 1...	HTTP Request	16	✓	7511	234	9	3
995	20:15:08.571	Thread Group 1...	HTTP Request	14	✓	7511	234	4	3
996	20:15:08.579	Thread Group 1...	HTTP Request	18	✓	7511	234	11	9
997	20:15:08.588	Thread Group 1...	HTTP Request	15	✓	7511	234	4	2
998	20:15:08.594	Thread Group 1...	HTTP Request	18	✓	7511	234	9	2
999	20:15:08.609	Thread Group 1...	HTTP Request	17	✓	7511	234	7	3
1000	20:15:08.628	Thread Group 1...	HTTP Request	13	✓	7511	234	7	1

☐ Scroll automatically?
☐ Child samples?
No of Samples 1000
Latest Sample 13
Average 14
Deviation 13

### C. FITUR-FITUR LAIN

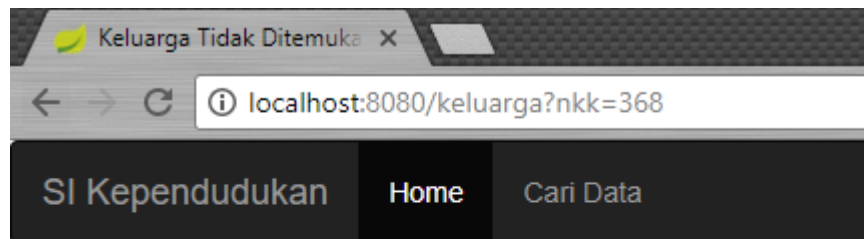
Fitur lain yang saya kembangkan adalah saya membuat halaman *error* dimana merupakan halaman *redirect* jika *user* mengakses halaman yang tidak ada.



# ERROR!

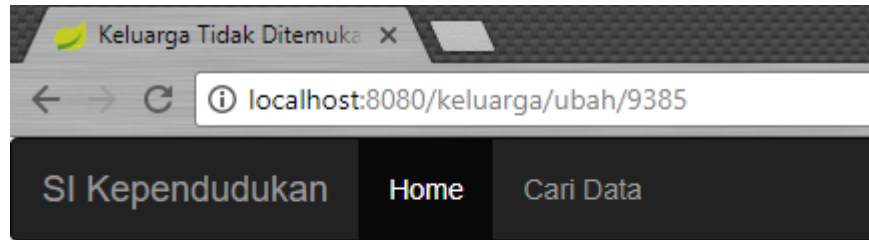
Halaman yang dicari tidak ditemukan

Selain itu, saya juga meng-*handle* jika *user* memasukkan nilai NKK yang salah pada fitur 2 dan 6 serta jika NIK salah pada fitur 1 dan 5.



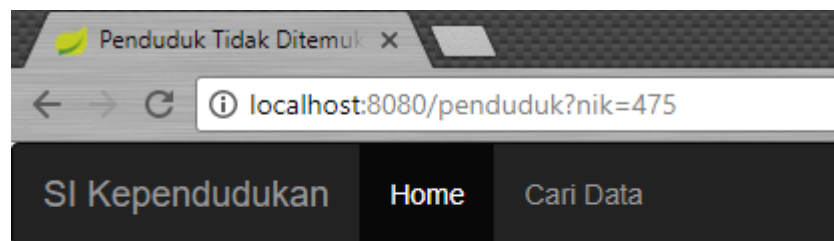
# KELUARGA TIDAK DITEMUKAN

Keluarga dengan NKK 368 tidak ditemukan



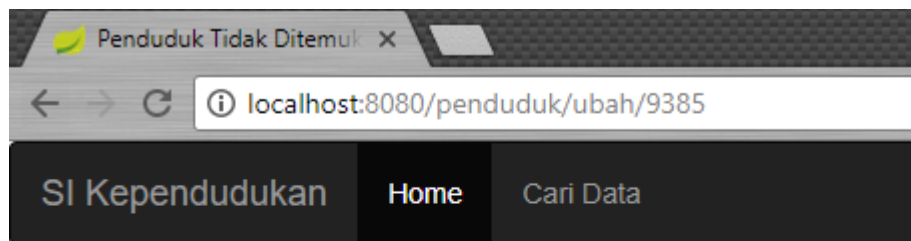
## KELUARGA TIDAK DITEMUKAN

Keluarga dengan NKK 9385 tidak ditemukan



## PENDUDUK TIDAK DITEMUKAN

Penduduk dengan NIK 475 tidak ditemukan



## PENDUDUK TIDAK DITEMUKAN

Penduduk dengan NIK 9385 tidak ditemukan