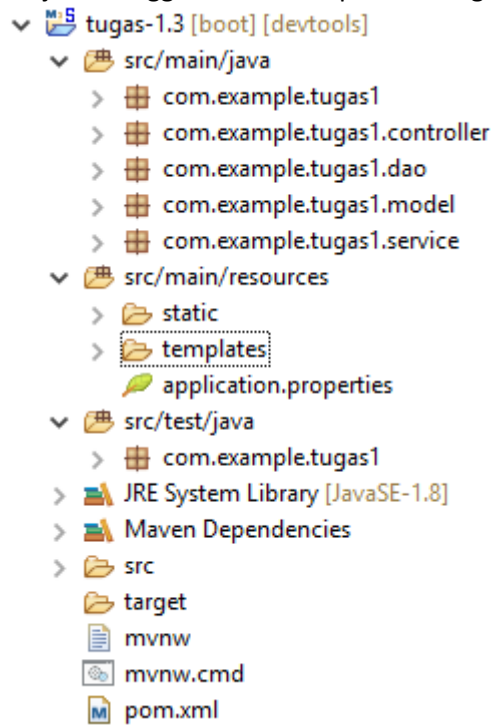


1. Struktur

a. Project

Project menggunakan konsep MVC dengan beberapa package



b. Package

Terdapat 5 package yang ada dalam project ini:

- Package `com.example.tugas1`
Berisikan `Application.java` yang berfungsi sebagai main class dari project
- Package `com.example.tugas1.controller`
Berisikan controller yang berfungsi untuk menerima url yang ada, lalu menambahkan informasi yang dibutuhkan ke dalam Model dan menampilkan halaman yang sesuai. Selain itu controller juga melakukan pemanggilan terhadap method-method yang ada pada service. Dalam package ini hanya terdapat satu Controller yaitu `SidukController`.
- Package `com.example.tugas1.dao`
Berisikan interface Mapper yang berfungsi untuk menghubungkan project dengan *database* yang ada. Dalam project ini terdapat 6 Mapper yaitu
 1. `PendudukMapper`
Merupakan Mapper yang berfungsi untuk mengambil informasi mengenai penduduk dari *database*
 2. `KeluargaMapper`
Merupakan Mapper yang berfungsi untuk mengambil informasi mengenai keluarga dari *database*
 3. `KelurahanMapper`
Merupakan Mapper yang berfungsi untuk mengambil informasi mengenai kelurahan dari *database*

4. KecamatanMapper

Merupakan Mapper yang berfungsi untuk mengambil informasi mengenai kecamatan dari *database*

5. KotaMapper

Merupakan Mapper yang berfungsi untuk mengambil informasi mengenai kota dari *database*

6. LokasiMapper

Merupakan Mapper yang berfungsi untuk mengambil informasi mengenai alamat dari *database*

- Package com.example.tugas1.model

Berisikan model-model yang berfungsi untuk menampung informasi dari *database*. Pada package ini, terdapat 6 model yaitu:

1. PendudukModel

Merupakan model yang berfungsi untuk menampung informasi dari *database* dari relasi penduduk beserta informasi yang perlu ditampilkan mengenai penduduk

```
public class PendudukModel {  
    private int id;  
    private String nik;  
    private String nama;  
    private String tempat_lahir;  
    private String tanggal_lahir;  
    private int is_wni;  
    private int id_keluarga;  
    private String agama;  
    private String pekerjaan;  
    private String status_perkawinan;  
    private String status_dalam_keluarga;  
    private String golongan_darah;  
    private int is_wafat;  
    private int jenis_kelamin;  
    private String rt;  
    private String rw;  
    private String nama_kelurahan;  
    private String nama_kecamatan;  
    private String nama_kota;  
    private String nkk;  
    private String alamat;  
}
```

2. KeluargaModel

Merupakan model yang berfungsi untuk menampung informasi dari *database* dari relasi keluarga beserta informasi yang perlu ditampilkan mengenai keluarga

```
public class KeluargaModel {  
    private int id;  
    private String nomor_kk;  
    private String alamat;  
    private String rt;  
    private String rw;  
    private int id_kelurahan;  
    private int id_kecamatan;  
    private int id_kota;  
    private boolean is_tidakberlaku;  
    private String nama_kelurahan;  
    private String nama_kecamatan;  
    private String nama_kota;  
    private List<PendudukModel> anggota;  
}
```

3. KelurahanModel

Merupakan model yang berfungsi untuk menampung informasi dari *database* dari relasi kelurahan beserta informasi yang perlu ditampilkan mengenai kelurahan

```
public class KelurahanModel {  
    private int id;  
    private int id_kecamatan;  
    private String kode_kelurahan;  
    private String nama_kelurahan;  
    private int kode_pos;  
}
```

4. KecamatanModel

Merupakan model yang berfungsi untuk menampung informasi dari *database* dari relasi kecamatan beserta informasi yang perlu ditampilkan mengenai kecamatan

```
public class KecamatanModel {  
    private int id;  
    private String kode_kecamatan;  
    private int id_kota;  
    private String nama_kecamatan;  
    private List<KelurahanModel> kelurahan_list;  
}
```

5. KotaModel

Merupakan model yang berfungsi untuk menampung informasi dari *database* dari relasi kota beserta informasi yang perlu ditampilkan mengenai kota

```
public class KotaModel {  
    private int id;  
    private int kode_kota;  
    private String nama_kota;  
    private List<KecamatanModel> kecamatan_list;  
}
```

6. AlamatModel

Merupakan model yang berfungsi untuk menampung informasi dari *database* yang merupakan gabungan dari relasi kelurahan, kecamatan serta kota

```
public class AlamatModel {  
    private String kode_kelurahan;  
    private String kode_kecamatan;  
    private String kode_kota;  
    private int id_kelurahan;  
    private int id_kecamatan;  
    private int id_kota;  
    private String nama_kelurahan;  
    private String nama_kecamatan;  
    private String nama_kota;  
}
```

- Package com.example.tugas1.service

Pada *package* ini terdapat 6 *interface* dengan masing-masing 1 *class* yang mengimplementasikannya

1. *Class* PendudukServiceDatabase yang mengimplementasi *interface* PendudukService
Berfungsi sebagai perhitungan utama mengenai informasi penduduk yang akan ditampilkan
2. *Class* KeluargaServiceDatabase yang mengimplementasi *interface* KeluargaService
Berfungsi sebagai perhitungan utama mengenai informasi keluarga yang akan ditampilkan
3. *Class* KelurahanServiceDatabase yang mengimplementasi *interface* KelurahanService
Berfungsi sebagai perhitungan utama mengenai informasi kelurahan yang akan ditampilkan
4. *Class* KecamatanServiceDatabase yang mengimplementasi *interface* KecamatanService
Berfungsi sebagai perhitungan utama mengenai informasi kecamatan yang akan ditampilkan
5. *Class* KotaServiceDatabase yang mengimplementasi *interface* KotaService
Berfungsi sebagai perhitungan utama mengenai informasi kota yang akan ditampilkan
6. *Class* LokasiServiceDatabase yang mengimplementasi *interface* LokasiService
Berfungsi sebagai perhitungan utama mengenai informasi lokasi/alamat yang akan ditampilkan

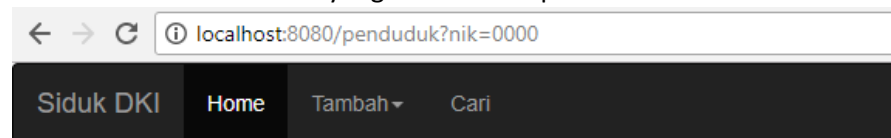
c. Implementasi MVC

- Model
Merupakan kumpulan kelas yang menggambarkan struktur data dari database. Dimana pada project ini, yang termasuk ke dalam Model adalah package Model
- View
Merupakan kumpulan kelas yang merepresentasikan *user interface*. Dimana pada project ini, yang termasuk ke dalam View adalah templates yang ada di folder static
- Controller
Merupakan kumpulan kelas yang mengendalikan alur program secara keseluruhan. Dimana pada project ini, yang termasuk ke dalam Controller adalah package Controller, Service dan DAO

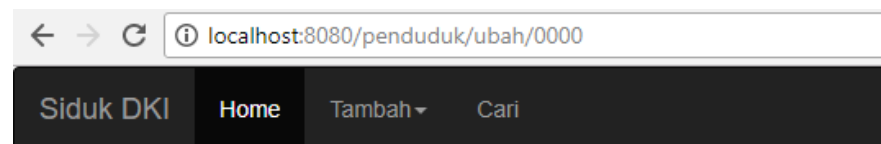
2. Fitur lain

a. Fitur error bila nik tidak ditemukan

Fitur ini akan muncul ketika nik yang dimasukkan pada fitur 1 dan fitur 5 tidak valid



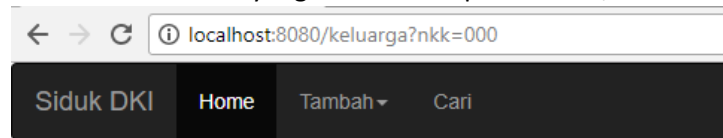
Gambar 2.1 Halaman Nik tidak ditemukan pada fitur 1



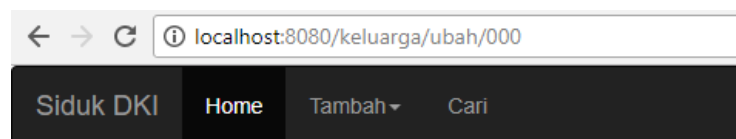
Gambar 2.2 Halaman Nik tidak ditemukan pada fitur 5

- b. Fitur error bila nkk tidak ditemukan

Fitur ini akan muncul ketika nkk yang dimasukkan pada fitur 2, dan 6 tidak valid



Gambar 2.3 Halaman Nkk tidak ditemukan pada fitur 2

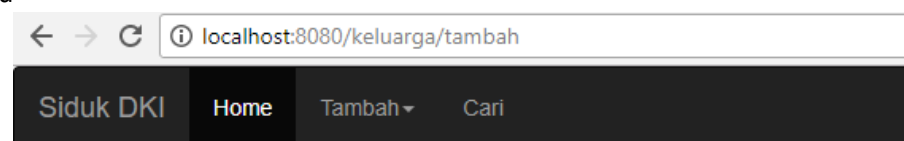


Not found

Nomor kartu keluarga 000 tidak ditemukan

Gambar 2.4 Halaman Nkk tidak ditemukan pada fitur 6

- c. Fitur error bila kombinasi nama kelurahan, nama kecamatan dan nama kota tidak valid



Not found

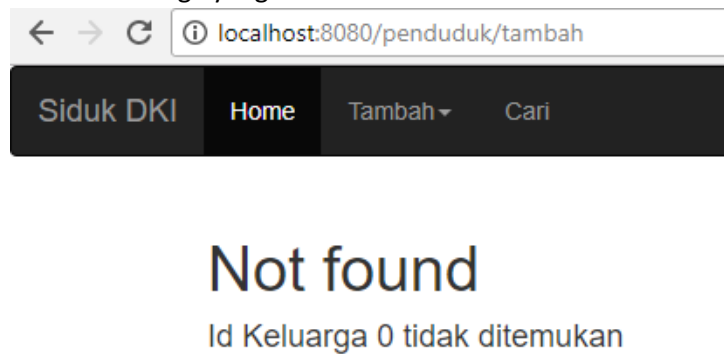
Kombinasi kelurahan, kecamatan, kota tidak ditemukan

Gambar 2.5 Halaman kobinasi kelurahan, kecamatan, kota tidak ditemukan pada fitur 4

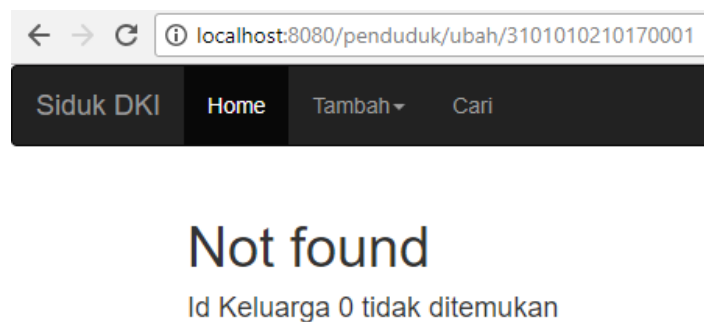


Gambar 2.6 Halaman kobinasi kelurahan, kecamatan, kota tidak ditemukan pada fitur 6

- d. Fitur error ketika id keluarga yang dimasukkan tidak valid



Gambar 2.7 Halaman kobinasi kelurahan, kecamatan, kota tidak ditemukan pada fitur 3



Gambar 2.8 Halaman kobinasi kelurahan, kecamatan, kota tidak ditemukan pada fitur 5

e. Tombol ubah

Fitur ini akan mengalihkan halaman ke form mengubah data penduduk maupun keluarga

Lihat Data Penduduk	
NIK	3101011405170001
Nama	Heru Haryanto
Tempat/Tanggal lahir	Jakarta/2017-05-14
Alamat	Ds. Adisumarmo No. 43
RT/RW	079/025
Kelurahan/Desa	PULAU TIDUNG
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota/Kabupaten	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	O-
Agama	Islam
Status Perkawinan	Belum Kawin
Pekerjaan	BELUM/TIDAK BEKERJA
Kewarganegaraan	WNI
Status Kematian	Hidup

Ubah

Gambar 2.9 Tombol ubah pada halaman view-penduduk

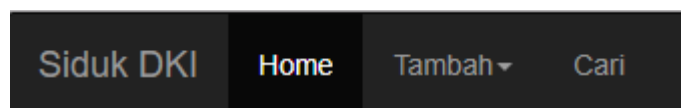
Lihat Data Keluarga - 3101012110170001

NKK	3101012110170001
Alamat	Jln. Cikapayang No. 594
RT/RW	149/090
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SEL
Kota	KABUPATEN KEPULAUAN

Ubah

Gambar 2.10 Tombol ubah pada halaman view-keluarga

f. Navbar



Gambar 2.11 Navbar

- Home berfungsi mengalihkan ke halaman utama
- Tambah berisi dropdown untuk pilihan menambahkan penduduk maupun keluarga
- Cari berfungsi untuk mengalihkan ke fitur 8

3. Optimasi *Database*

a. Penambahan *Primary Key*

Pada tiap relasi/tabel pada *database*, belum terdapat *primary key*. Sehingga dapat dilakukan optimasi dengan menambahkan:

- Id sebagai *Primary Key* relasi penduduk
- Id sebagai *Primary Key* relasi keluarga
- Id sebagai *Primary Key* relasi kelurahan
- Id sebagai *Primary Key* relasi kecamatan
- Id sebagai *Primary Key* relasi kota

b. Penambahan *Index*

Terdapat *index* yang dapat diterapkan dalam relasi guna mempercepat pencarian dalam suatu relasi seperti pada operasi 'LIKE'. *Index* yang diterapkan adalah:

- Pada atribut nik pada relasi penduduk
- Pada atribut id_keluarga pada relasi penduduk
- Pada atribut tanggal_lahir pada relasi penduduk
- Pada atribut nomor_kk pada relasi keluarga
- Pada atribut id_kelurahan pada relasi keluarga
- Pada atribut kode pada relasi kelurahan
- Pada atribut id_kecamatan pada relasi kelurahan
- Pada atribut kode pada relasi kecamatan
- Pada atribut id_kota pada relasi kecamatan
- Pada atribut kode pada relasi kota

c. Penambahan *Auto Increment*

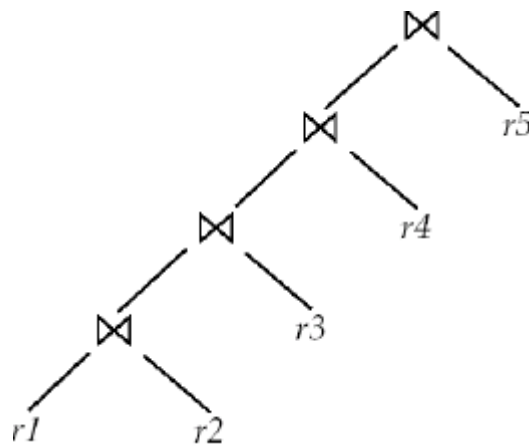
Auto Increment berfungsi untuk menambahkan nilai secara otomatis setiap bertambahnya relasi pada sebuah data. Hal ini memungkinkan agar *primary key* selalu terjaga dan tidak perlu untuk dilakukan penambahan *primary key* secara manual tiap bertambahnya data pada relasi. *Auto Increment* diterapkan pada:

- Atribut id pada relasi penduduk
- Atribut id pada relasi keluarga

4. Optimasi Query

a. Penerapan *Left Deep Join Trees*

Contoh dari *Query Tree*



Gambar 4.1 *Query Tree* untuk operasi $r1 \text{ join } r2 \text{ join } r3 \text{ join } r4 \text{ join } r5$ dengan penerapan *left deep join*

Dalam *Left Deep Join Tree*, anak di sebelah kanan merupakan relasi bukan merupakan hasil dari operasi *Join*. Hal ini bertujuan untuk mengurangi banyak *tuple* (baris data) yang dihasilkan dari tiap operasi *Join* yang ada.

b. Penerapan *Heuristic Optimization*

Optimasi ini menekankan kepada pengurangan *cost time* yang diperlukan dalam query yang ada dengan langkah-langkah :

- Mengubah conjunctive selections (WHERE) menjadi sebuah sekuense of operasi *selection* yang tunggal

$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

Gambar 4.2 conjunctive selections equivalence rule

Dimana E merupakan sebuah relasi dan theta merupakan kondisi yang diinginkan (ex: $\text{id_keluarga} = 2$)

- Memindahkan operasi *selection* yang memungkinkan ke bagian bawah *query tree* untuk dieksekusi terlebih dahulu
- Mengeksekusi lebih dahulu *selection* dan operasi *join* yang akan menghasilkan relasi dengan jumlah *tuple* paling sedikit
- Mengganti operasi *Cartesian product* yang diikuti dengan *selection condition* dengan operasi *join*
- Mengambil atribut-attribut yang hanya diperlukan dari tiap relasi
- Mengidentifikasi *subtrees* yang operasinya dapat menggunakan pipelining lalu eksekusi menggunakan pipelining

Contoh dari optimasi *query* ini adalah mengubah *query* mendapatkan alamat lengkap suatu keluarga

```
@Select("select * "
+ "FROM kota kot, kecamatan kec, kelurahan kel "
+ "WHERE kot.id = kec.id_kota AND kec.id = kel.id_kecamatan AND kel.id = #{id_kelurahan}")
AlamatModel getAlamatLengkapbyIdKel(@Param("id_kelurahan")int id_kelurahan);
```

Gambar 4.3 Query yang belum dioptimasi

menjadi

```
@Select("select kot.id AS id_kota, kec.id AS id_kecamatan, kel.id AS id_kelurahan, "
+ "kel.nama_kelurahan, kec.nama_kecamatan, kot.nama_kota "
+ "FROM (SELECT id, nama_kota FROM kota) AS kot JOIN "
+ "(SELECT id, id_kota, nama_kecamatan FROM kecamatan) AS kec "
+ "ON kot.id = kec.id_kota JOIN "
+ "(SELECT id, id_kecamatan, nama_kelurahan FROM kelurahan WHERE id = #{id_kelurahan}) AS kel "
+ "ON kec.id = kel.id_kecamatan")
AlamatModel getAlamatLengkapbyIdKel(@Param("id_kelurahan")int id_kelurahan);
```

Gambar 4.4 Query yang telah dioptimasi

5. Stress Testing

Stress Testing dilakukan dengan menggunakan *thread = 1000* dengan *ram-up period = 10*

a. Fitur 1

View Results in Table									
Name: View Results in Table									
Comments:									
Write results to file / Read from file									
Filename				Browse...		Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes		Configure	
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	16:26:41.825	Thread Group 1-1	HTTP Request	16	✓	3740	146	15	4
2	16:26:41.873	Thread Group 1-5	HTTP Request	29	✓	3740	146	29	3
3	16:26:41.892	Thread Group 1-7	HTTP Request	10	✓	3740	146	10	2
4	16:26:41.862	Thread Group 1-4	HTTP Request	59	✓	3740	146	59	3
5	16:26:41.913	Thread Group 1-9	HTTP Request	34	✓	3740	146	34	1
6	16:26:41.923	Thread Group 1-...	HTTP Request	40	✓	3740	146	40	2
7	16:26:41.933	Thread Group 1-...	HTTP Request	75	✓	3740	146	75	1
8	16:26:41.954	Thread Group 1-...	HTTP Request	56	✓	3740	146	56	3
9	16:26:41.994	Thread Group 1-...	HTTP Request	30	✓	3740	146	30	0
10	16:26:41.984	Thread Group 1-...	HTTP Request	47	✓	3740	146	47	1
11	16:26:41.973	Thread Group 1-...	HTTP Request	68	✓	3740	146	68	2
12	16:26:41.963	Thread Group 1-...	HTTP Request	79	✓	3740	146	79	3
13	16:26:41.943	Thread Group 1-...	HTTP Request	111	✓	3740	146	110	1
14	16:26:42.024	Thread Group 1-...	HTTP Request	45	✓	3740	146	45	1
15	16:26:42.035	Thread Group 1-...	HTTP Request	37	✓	3740	146	36	2
16	16:26:42.004	Thread Group 1-...	HTTP Request	72	✓	3740	146	72	2
17	16:26:42.014	Thread Group 1-...	HTTP Request	75	✓	3740	146	75	3
18	16:26:41.902	Thread Group 1-8	HTTP Request	194	✓	3740	146	194	1
19	16:26:42.055	Thread Group 1-...	HTTP Request	45	✓	3740	146	44	2
20	16:26:42.065	Thread Group 1-...	HTTP Request	45	✓	3740	146	45	1
21	16:26:42.083	Thread Group 1-...	HTTP Request	31	✓	3740	146	31	1
22	16:26:42.097	Thread Group 1-...	HTTP Request	18	✓	3740	146	18	3
23	16:26:42.045	Thread Group 1-...	HTTP Request	82	✓	3740	146	82	2
24	16:26:41.841	Thread Group 1-2	HTTP Request	294	✓	3740	146	294	1
25	16:26:41.883	Thread Group 1-6	HTTP Request	266	✓	3740	146	265	2
26	16:26:41.848	Thread Group 1-3	HTTP Request	307	✓	3740	146	307	3
27	16:26:42.122	Thread Group 1-...	HTTP Request	60	✓	3740	146	59	2
<input type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples? No of Samples 1000 Latest Sample 802 Average 249 Deviation 336									

Gambar 5.1 Stress Testing pada Fitur 1

Fitur ini cepat dan reliable bila diakses oleh banyak pengguna dalam waktu bersamaan

Abdul Raziq Andri
1506757560
Tugas 1 APAP

b. Fitur 2

View Results in Table										
Name: View Results in Table										
Comments:										
Write results to file / Read from file										
Filename						Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>			
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)	
1	16:45:06.322	Thread Group 1-7	HTTP Request	69	✓	6008	146	69	0	▲
2	16:45:06.353	Thread Group 1-...	HTTP Request	38	✓	6008	146	38	0	■
3	16:45:06.364	Thread Group 1-...	HTTP Request	30	✓	6008	146	29	0	
4	16:45:06.271	Thread Group 1-2	HTTP Request	126	✓	6008	146	126	0	
5	16:45:06.282	Thread Group 1-3	HTTP Request	115	✓	6008	146	115	3	
6	16:45:06.295	Thread Group 1-4	HTTP Request	109	✓	6008	146	109	0	
7	16:45:06.333	Thread Group 1-8	HTTP Request	71	✓	6008	146	71	3	
8	16:45:06.374	Thread Group 1-...	HTTP Request	37	✓	6008	146	37	1	
9	16:45:06.343	Thread Group 1-9	HTTP Request	68	✓	6008	146	68	1	
10	16:45:06.384	Thread Group 1-...	HTTP Request	33	✓	6008	146	32	1	
11	16:45:06.394	Thread Group 1-...	HTTP Request	29	✓	6008	146	29	1	
12	16:45:06.404	Thread Group 1-...	HTTP Request	26	✓	6008	146	26	1	
13	16:45:06.414	Thread Group 1-...	HTTP Request	18	✓	6008	146	18	1	
14	16:45:06.425	Thread Group 1-...	HTTP Request	16	✓	6008	146	16	1	
15	16:45:06.313	Thread Group 1-6	HTTP Request	130	✓	6008	146	130	0	
16	16:45:06.257	Thread Group 1-1	HTTP Request	186	✓	6008	146	186	12	
17	16:45:06.302	Thread Group 1-5	HTTP Request	152	✓	6008	146	152	0	
18	16:45:06.434	Thread Group 1-...	HTTP Request	25	✓	6008	146	25	2	
19	16:45:06.445	Thread Group 1-...	HTTP Request	15	✓	6008	146	15	2	
20	16:45:06.455	Thread Group 1-...	HTTP Request	18	✓	6008	146	18	2	
21	16:45:06.465	Thread Group 1-...	HTTP Request	15	✓	6008	146	15	1	
22	16:45:06.476	Thread Group 1-...	HTTP Request	14	✓	6008	146	14	1	
23	16:45:06.485	Thread Group 1-...	HTTP Request	10	✓	6008	146	10	1	
24	16:45:06.495	Thread Group 1-...	HTTP Request	13	✓	6008	146	13	1	
25	16:45:06.502	Thread Group 1-...	HTTP Request	9	✓	6008	146	8	1	
26	16:45:06.515	Thread Group 1-...	HTTP Request	8	✓	6008	146	8	1	
27	16:45:06.530	Thread Group 1-...	HTTP Request	q	✓	6008	146	q	1	▼
<input type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples? No of Samples 1000 Latest Sample 13 Average 13 Deviation 18										

Gambar 5.2 Stress Testing pada Fitur 2

Fitur ini cepat dan reliable bila diakses oleh banyak pengguna dalam waktu bersamaan

c. Fitur 8 dan 10

View Results in Table										
Name: View Results in Table										
Comments:										
Write results to file / Read from file										
Filename						Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>			
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)	
1	18:18:38.368	Thread Group 1-2	HTTP Request	286	✓	99075	146	279	1	▲
2	18:18:38.379	Thread Group 1-3	HTTP Request	307	✓	99075	146	303	1	■
3	18:18:38.394	Thread Group 1-4	HTTP Request	319	✓	99075	146	313	0	
4	18:18:38.357	Thread Group 1-1	HTTP Request	378	✓	99075	146	288	5	
5	18:18:38.465	Thread Group 1-...	HTTP Request	303	✓	99075	146	298	1	
6	18:18:38.435	Thread Group 1-8	HTTP Request	350	✓	99075	146	345	1	
7	18:18:38.445	Thread Group 1-9	HTTP Request	344	✓	99075	146	278	1	
8	18:18:38.403	Thread Group 1-5	HTTP Request	393	✓	99075	146	362	1	
9	18:18:38.455	Thread Group 1-...	HTTP Request	477	✓	99075	146	473	1	
10	18:18:38.750	Thread Group 1-...	HTTP Request	706	✓	99075	146	702	1	
11	18:18:38.527	Thread Group 1-...	HTTP Request	1020	✓	99075	146	1011	0	
12	18:18:38.515	Thread Group 1-...	HTTP Request	1035	✓	99075	146	1029	1	
13	18:18:38.535	Thread Group 1-...	HTTP Request	1027	✓	99075	146	1023	1	
14	18:18:38.426	Thread Group 1-7	HTTP Request	1140	✓	99075	146	1135	1	
15	18:18:38.556	Thread Group 1-...	HTTP Request	1024	✓	99075	146	1020	1	
16	18:18:38.505	Thread Group 1-...	HTTP Request	1084	✓	99075	146	1081	1	
17	18:18:38.616	Thread Group 1-...	HTTP Request	983	✓	99075	146	975	1	
18	18:18:38.678	Thread Group 1-...	HTTP Request	949	✓	99075	146	942	1	
19	18:18:38.476	Thread Group 1-...	HTTP Request	1180	✓	99075	146	1175	1	
20	18:18:38.760	Thread Group 1-...	HTTP Request	924	✓	99075	146	707	1	
21	18:18:38.606	Thread Group 1-...	HTTP Request	1112	✓	99075	146	1106	1	
22	18:18:38.495	Thread Group 1-...	HTTP Request	1226	✓	99075	146	1095	1	
23	18:18:38.627	Thread Group 1-...	HTTP Request	1101	✓	99075	146	982	0	
24	18:18:38.646	Thread Group 1-...	HTTP Request	1089	✓	99075	146	1013	1	
25	18:18:38.485	Thread Group 1-...	HTTP Request	1322	✓	99075	146	1295	1	
26	18:18:38.577	Thread Group 1-...	HTTP Request	1379	✓	99075	146	1376	1	
27	18:18:38.545	Thread Group 1-...	HTTP Request	1422	✓	99075	146	1419	1	▼
<input type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples? No of Samples 1000 Latest Sample 27204 Average 15755 Deviation 7757										

Gambar 5.3 Stress Testing pada Fitur 8 dan 10

Fitur ini cukup cepat (mengingat banyaknya data yang diambil) dan cukup reliable bila diakses oleh banyak pengguna dalam waktu bersamaan