

WRITE UP TUGAS 1

ARSITEKTUR DAN PERANCANGAN APLIKASI PERUSAHAAN

Sistem Kependudukan Online Provinsi DKI Jakarta

Oleh :

Lintang Matahari Hasani

1506689231

FAKULTAS ILMU KOMPUTER

UNIVERSITAS INDONESIA

2017

I. PENJELASAN FITUR

A. Menampilkan Data Penduduk Berdasarkan NIK

Sistem dapat menampilkan data penduduk berdasarkan NIK. User memasukkan NIK penduduk yang ingin ditampilkan pada field yang terletak pada laman utama (index.html) kemudian mengklik tombol Lihat. Sistem kemudian menampilkan data penduduk yang dicari.

The screenshot shows the main interface of the 'SIDUK JAKARTA' web application. The header includes the title 'Sistem Informasi Kependudukan' and navigation links 'Home' and 'Cari Data'. The main content area features a large title 'SIDUK JAKARTA' and a subtitle 'Selamat Datang di Sistem Informasi Kependudukan Provinsi DKI Jakarta'. Below this are three buttons: 'Tambah Penduduk', 'Tambah Keluarga', and 'Penduduk Kelurahan'. The 'Lihat Data Penduduk Berdasarkan NIK' section has a text input field containing '3101010101900001' and a 'Lihat' button. The 'Lihat Data Keluarga Berdasarkan NKK' section has a text input field labeled 'Masukkan NIK' and a 'Lihat' button. The 'Statistik Kependudukan' section displays three statistics: 'Jumlah Penduduk : 281473', 'Jumlah Keluarga : 63491', and 'Rasio Penduduk/Keluarga : 4'.

Mengisi field Lihat Data Penduduk di halaman utama

The screenshot shows the detailed profile of a resident with NIK 3101010101900001. The header is the same as the previous screenshot. The main content area is titled 'Lihat Data Penduduk - 3101010101900001'. It displays a list of personal and administrative details on the left, with corresponding values on the right. The details include NIK, Nama, Tempat/Tanggal Lahir, Alamat, RT/RW, Kelurahan/Desa, Kecamatan, Kota, Golongan Darah, Agama, Status Perkawinan, Pekerjaan, Kewarganegaraan, and Status Kematian. A 'Nonaktifkan' button is located at the bottom right of the details section.

Detail	Value
NIK	3101010101900001
Nama	Ridwan Hasim Marpaung
Tempat/Tanggal Lahir	Atambua/1990-01-01
Alamat	Kpg. Cihampelas No. 220
RT/RW	177/056
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	A
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	KETUA DPRD KABUPATEN SUMEDANG
Kewarganegaraan	VNI
Status Kematian	Hidup

Laman detail penduduk

Langkah Pengerjaan :

1. Membuat Class Model

Class Model yang perlu dibuat, antara lain: CitizenModel. Class CitizenModel memodelkan penduduk dan menyimpan atribut yang terkait dengan penduduk, seperti nama, NIK, tempat lahir, tanggal lahir, id keluarga, status perkawinan, pekerjaan dan status kematian. Class CitizenModel juga menyimpan atribut terkait penduduk seperti data domisili berupa kelurahan, kecamatan, dan kota tempat penduduk terdaftar pada kartu keluarga.

2. Membuat Interface CitizenService

Interface CitizenService memuat method terkait fitur pencarian berdasarkan NIK, seperti selectCitizen() dengan parameter NIK.

3. Membuat Interface CitizenMapper

Interface CitizenMapper memuat method terkait fitur pencarian berdasarkan NIK, seperti selectCitizen() dengan parameter NIK. Method ini kemudian akan mengirimkan query ke database MySQL sesuai masukan NIK pada parameter dan mengambil data pada tabel penduduk, keluarga, kelurahan, kecamatan, dan kota yang relevan.

4. Membuat Class CitizenServiceDatabase

Class CitizenServiceDatabase mengimplementasikan method selectCitizen() pada interface CitizenService. Method ini memanggil method selectcitizen() yang menerima parameter NIK penduduk

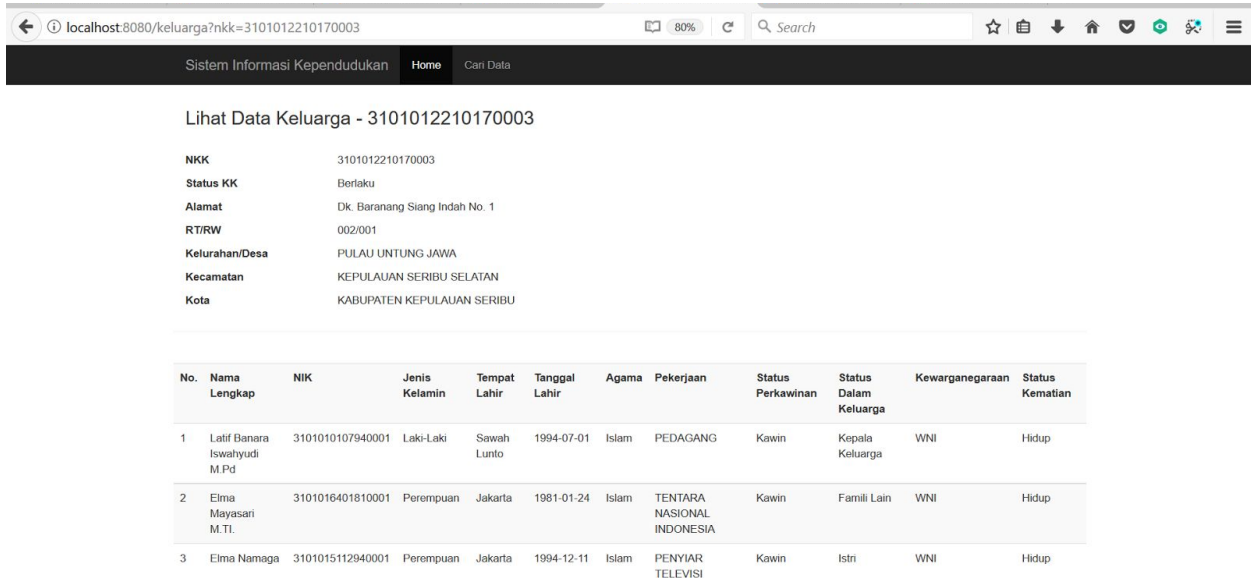
5. Membuat Class Controller

Class CitizenController memuat method yang menangani Request Mapping /index dan /penduduk. Method selectPenduduk() menerima parameter NIK penduduk yang dicari dan memproses converting data terkait Status kematian dan Kewarganegaraan penduduk yang dicari

6. Membuat view

B. Menampilkan Data Data Keluarga Berdasarkan Nomor KK

Sistem dapat menampilkan data keluarga berdasarkan NKK (Nomor Kartu Keluarga). User memasukkan NKK penduduk yang ingin ditampilkan pada field yang terletak pada laman utama (index.html) kemudian mengklik tombol Lihat. Sistem kemudian menampilkan data keluarga yang dicari.



Lihat Data Keluarga - 3101012210170003

NKK	3101012210170003
Status KK	Berlaku
Alamat	Dk. Baranang Siang Indah No. 1
RT/RW	002/001
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU

No.	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga	Kewarganegaraan	Status Kematian
1	Latif Banara Iswahyudi M.Pd	3101010107940001	Laki-Laki	Sawah Lunto	1994-07-01	Islam	PEDAGANG	Kawin	Kepala Keluarga	WNI	Hidup
2	Elma Mayasari M.Ti.	3101016401810001	Perempuan	Jakarta	1981-01-24	Islam	TENTARA NASIONAL INDONESIA	Kawin	Famili Lain	WNI	Hidup
3	Elma Namaga	3101015112940001	Perempuan	Jakarta	1994-12-11	Islam	PENYIAR TELEVISI	Kawin	Istri	WNI	Hidup

Tampilan laman detail keluarga

Langkah Pengerjaan :

1. Membuat Class Model

Class Model yang perlu dibuat, antara lain: FamilyModel. Class FamilyModel memodelkan keluarga dan menyimpan atribut yang terkait dengan keluarga, seperti NKK, alamat, RT, RW, id kelurahan, status berlaku, dan id. Class FamilyModel juga menyimpan atribut terkait keluarga seperti data domisili berupa kelurahan, kecamatan, dan kota tempat keluarga terdaftar pada kartu keluarga. Selain itu, class FamilyModel juga menyimpan atribut list CitizenModel yang merupakan anggota keluarga.

2. Memodifikasi Interface CitizenService

Interface CitizenService memuat method terkait fitur pencarian berdasarkan NKK, seperti selectFamily() dengan parameter NKK.

3. Memodifikasi Interface CitizenMapper

Interface CitizenMapper memuat method terkait fitur pencarian berdasarkan NKK, seperti selectFamily() dengan parameter NKK. Method ini kemudian akan mengirimkan query ke database MySQL sesuai masukan NKK pada parameter dan

mengambil data pada tabel keluarga, kelurahan, kecamatan, dan kota yang relevan dengan keluarga.

4. Memodifikasi Class CitizenServiceDatabase

Class CitizenServiceDatabase mengimplementasikan method selectFamily() pada interface CitizenService. Method ini memanggil method selectFamily() yang menerima parameter NKK penduduk

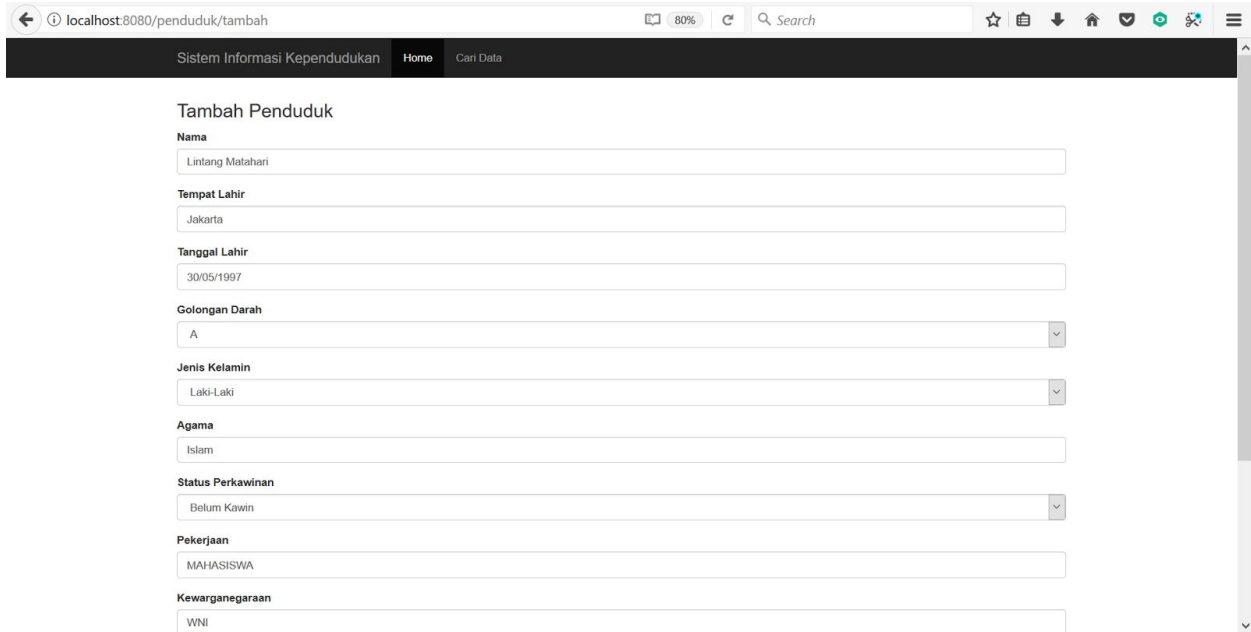
5. Memodifikasi Class Controller

Class CitizenController memuat method yang menangani Request Mapping /index dan /keluarga. Method selectKeluarga() menerima parameter NKK keluarga yang dicari dan memproses converting data terkait Status berlaku dan keluarga yang dicari

6. Membuat view

C. Menambahkan Penduduk Baru Sebagai Anggota Keluarga

Sistem dapat menambahkan data penduduk baru pada database. User memasukkan data penduduk yang ingin ditambahkan pada form yang terletak pada laman tambah penduduk (tambah-penduduk.html) kemudian mengklik tombol Tambah. Sistem kemudian menampilkan laman tambah-penduduk-berhasil.html yang menampilkan pesan bahwa penduduk sesuai informasi yang diisi telah ditambahkan. Fitur ini dapat diakses dengan mengakses path /penduduk/tambah atau menekan tombol Tambah Penduduk pada laman beranda.

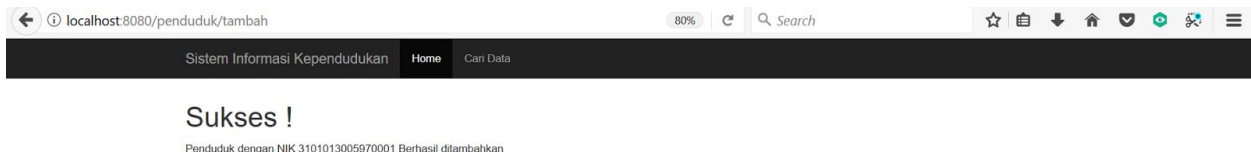


The screenshot shows a web browser at localhost:8080/penduduk/tambah. The page title is 'Sistem Informasi Kependudukan'. The form 'Tambah Penduduk' includes the following fields:

- Nama: Lintang Matahari
- Tempat Lahir: Jakarta
- Tanggal Lahir: 30/05/1997
- Golongan Darah: A
- Jenis Kelamin: Laki-Laki
- Agama: Islam
- Status Perkawinan: Belum Kawin
- Pekerjaan: MAHASISWA
- Kewarganegaraan: WNI

Tampilan laman tambah penduduk

Setiap penduduk baru yang ditambahkan memiliki NIK yang unik berdasarkan aturan pembuatan NIK. Jika NIK yang identik (memiliki tanggal lahir dan wilayah yang sama) ditemukan, sistem akan meng-increment digit pada 4 digit terakhir NIK.



The screenshot shows a web browser at localhost:8080/penduduk/tambah. The page displays a success message: 'Sukses !' followed by 'Penduduk dengan NIK 3101013005970001 Berhasil ditambahkan'.

Tampilan laman berhasil tambah penduduk

Langkah Pengerjaan :

1. Memodifikasi Interface CitizenService

Interface CitizenService memuat method terkait fitur penambahan penduduk addCitizen() dengan parameter CitizenModel penduduk yang ingin ditambahkan.

2. Memodifikasi Interface CitizenMapper

Interface CitizenMapper memuat method terkait fitur penambahan penduduk. Method ini kemudian akan mengirimkan query ke database MySQL sesuai masukan object CitizenModel pada parameter dan memasukkan data pada tabel penduduk.

3. Memodifikasi Class CitizenServiceDatabase

Class CitizenServiceDatabase mengimplementasikan method addCitizen() pada interface CitizenService. Method ini memanggil method addCitizen() yang menerima parameter CitizenModel penduduk

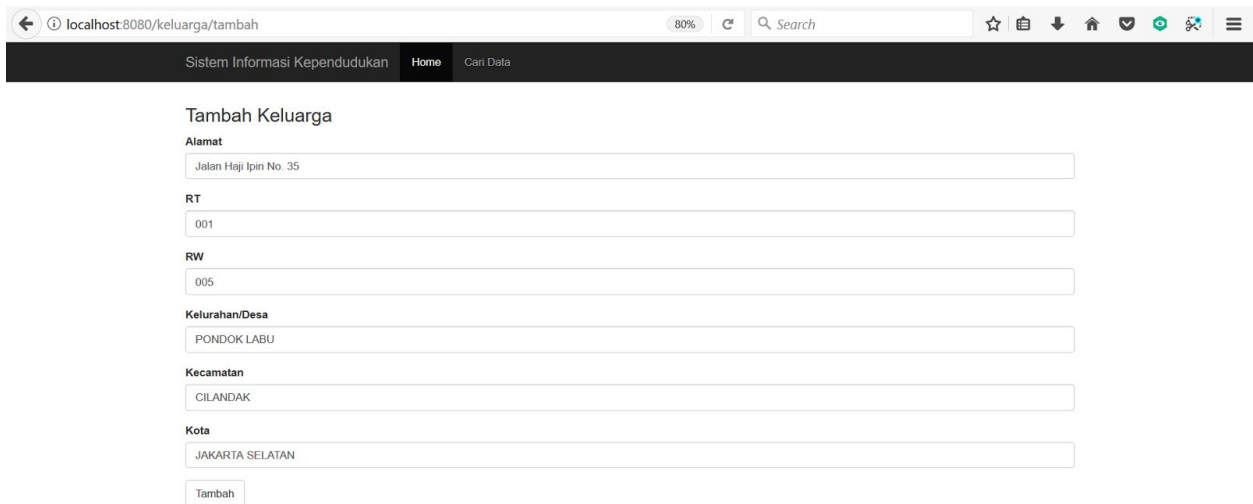
4. Memodifikasi Class Controller

Class CitizenController memuat method yang menangani Request Mapping /penduduk/tambah. Method addCitizen() mengembalikan laman tambah-penduduk.html yang memuat form penambahan penduduk baru. Method addCitizenSubmit() menerima parameter data penduduk yang ingin ditambah dan memproses converting data terkait Status Kematian dan Jenis Kelamin.

5. Membuat view

D. Menambahkan Keluarga Baru

Sistem dapat menambahkan data keluarga baru pada database. User memasukkan data keluarga yang ingin ditambahkan pada form yang terletak pada laman tambah keluarga (tambah-keluarga.html) kemudian mengklik tombol Tambah. Sistem kemudian menampilkan laman tambah-keluarga-berhasil.html yang menampilkan pesan bahwa keluarga sesuai informasi yang diisi telah ditambahkan. Fitur ini dapat diakses dengan mengakses path /keluarga/tambah atau menekan tombol Tambah Keluarga pada laman beranda.



Sistem Informasi Kependudukan Home Cari Data

Tambah Keluarga

Alamat
Jalan Haji Ipin No. 35

RT
001

RW
005

Kelurahan/Desa
PONDOK LABU


Kecamatan
CILANDAK

Kota
JAKARTA SELATAN

Tambah

Tampilan laman tambah penduduk

Setiap keluarga baru yang ditambahkan memiliki NKK yang unik berdasarkan aturan pembuatan NKK. Jika NKK yang identik (memiliki tanggal pembuatan dan wilayah yang sama) ditemukan, sistem akan meng-increment digit pada 4 digit terakhir NKK.



Sistem Informasi Kependudukan Home Cari Data

Sukses !

Keluarga dengan NKK 3171032210170001 Berhasil ditambahkan

Tampilan laman tambah penduduk berhasil

Langkah Pengerjaan :

1. Memodifikasi Interface CitizenService

Interface CitizenService memuat method terkait fitur penambahan keluarga addFamily() dengan parameter FamilyModel keluarga yang ingin ditambahkan.

2. Memodifikasi Interface CitizenMapper

Interface CitizenMapper memuat method terkait fitur penambahan keluarga. Method ini kemudian akan mengirimkan query ke database MySQL sesuai masukan object FamilyModel pada parameter dan memasukkan data pada tabel keluarga.

3. Memodifikasi Class CitizenServiceDatabase

Class CitizenServiceDatabase mengimplementasikan method addFamily() pada interface CitizenService. Method ini memanggil method addFamily() yang menerima parameter FamilyModel keluarga

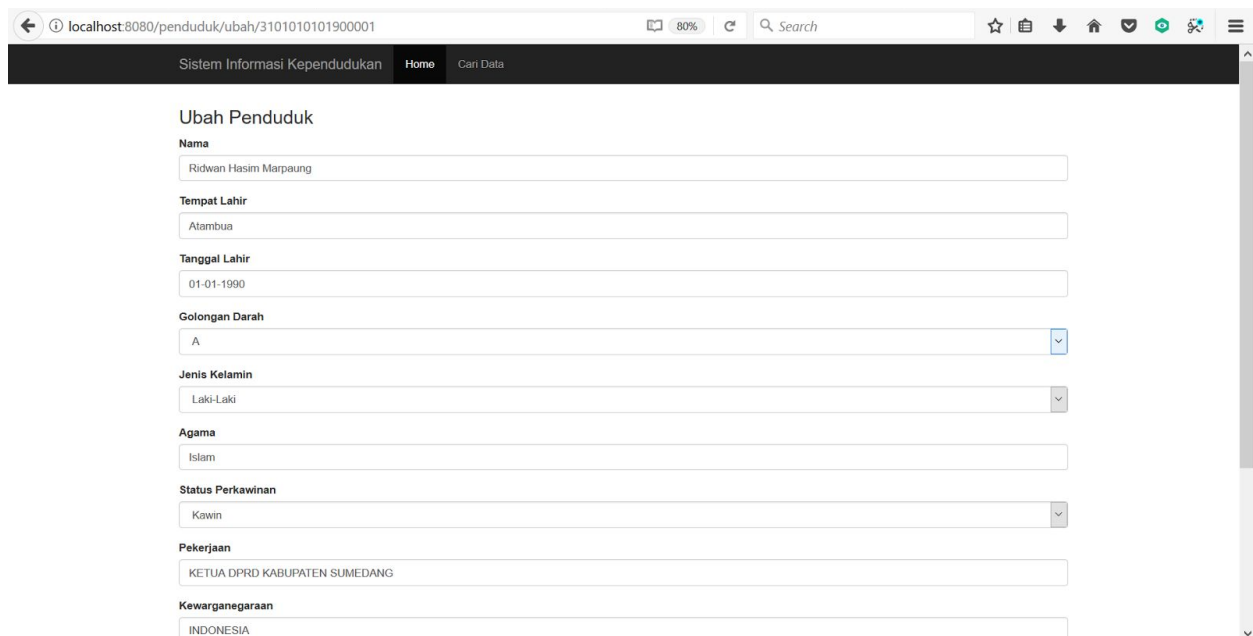
4. Memodifikasi Class Controller

Class CitizenController memuat method yang menangani Request Mapping /keluarga/tambah. Method add() mengembalikan laman tambah-keluarga.html yang memuat form penambahan keluarga baru. Method addSubmit() menerima parameter data keluarga yang ingin ditambah dan memproses converting data terkait Status Berlaku.

5. Membuat view

E. Mengubah Data Penduduk

Sistem dapat mengubah data penduduk pada database berdasarkan NIK. User memasukkan data baru pada penduduk yang ingin ditambahkan pada form yang terletak pada laman ubah penduduk (ubah-penduduk.html) kemudian mengklik tombol Tambah. Sistem kemudian menampilkan laman ubah-penduduk-berhasil.html yang menampilkan pesan bahwa keluarga sesuai informasi yang diisi telah ditambahkan beserta NIK lama penduduk. Fitur ini dapat diakses dengan mengakses path /penduduk/ubah/{NIK}.



The screenshot shows a web browser window with the URL `localhost:8080/penduduk/ubah/31010101900001`. The page title is "Ubah Penduduk". The form contains the following fields:

- Nama**: Text input with value "Ridwan Hasim Marpaung".
- Tempat Lahir**: Text input with value "Atambua".
- Tanggal Lahir**: Text input with value "01-01-1990".
- Golongan Darah**: Dropdown menu with value "A".
- Jenis Kelamin**: Dropdown menu with value "Laki-Laki".
- Agama**: Text input with value "Islam".
- Status Perkawinan**: Dropdown menu with value "Kawin".
- Pekerjaan**: Text input with value "KETUA DPRD KABUPATEN SUMEDANG".
- Kewarganegaraan**: Text input with value "INDONESIA".

Tampilan laman ubah penduduk

Setiap penduduk yang diubah memiliki NIK berdasarkan aturan pembuatan NIK. Jika NIK yang identik (memiliki tanggal lahir dan wilayah yang sama) ditemukan dan NIK baru berbeda dengan NIK lama, sistem akan meng-increment digit pada 4 digit terakhir NIK. Jika NIK identik dengan NIK lama, NIK tidak diubah



The screenshot shows a web browser window with the URL `localhost:8080/penduduk/ubah/31010101900001`. The page title is "Sukses !". The message displayed is:

Penduduk dengan NIK 31010101900001 Berhasil diubah

Tampilan laman ubah penduduk berhasil

Langkah Pengerjaan :

1. Memodifikasi Interface CitizenService
Interface CitizenService memuat method terkait fitur perubahan penduduk updateCitizen() dengan parameter CitizenModel penduduk yang ingin diubah.
2. Memodifikasi Interface CitizenMapper
Interface CitizenMapper memuat method terkait fitur perubahan penduduk. Method ini kemudian akan mengirimkan query ke database MySQL sesuai masukan object CitizenModel pada parameter dan memasukkan data pada tabel penduduk.
3. Memodifikasi Class CitizenServiceDatabase
Class CitizenServiceDatabase mengimplementasikan method updateCitizen() pada interface CitizenService. Method ini memanggil method updateCitizen() yang menerima parameter CitizenModel penduduk
4. Memodifikasi Class Controller
Class CitizenController memuat method yang menangani Request Mapping /penduduk/ubah/{NIK}. Method updateCitizen() mengembalikan laman ubah-penduduk.html yang memuat form perubahan penduduk. Method updateCitizenSubmit() menerima parameter data penduduk yang ingin diubah dan memproses converting data terkait Status Kematian dan Jenis Kelamin.
5. Membuat view

F. Mengubah Data Keluarga

Sistem dapat mengubah data keluarga pada database berdasarkan NKK. User memasukkan data baru pada keluarga yang ingin diubah pada form yang terletak pada laman ubah keluarga (ubah-keluarga.html) kemudian mengklik tombol Tambah. Sistem kemudian menampilkan laman ubah-keluarga-berhasil.html yang menampilkan pesan bahwa keluarga sesuai informasi yang diisi telah diubah beserta NKK lama keluarga. Fitur ini dapat diakses dengan mengakses path /keluarga/ubah/{NKK}.

The screenshot shows a web browser at localhost:8080/keluarga/ubah/3101012210170003. The page title is 'Ubah Keluarga'. It contains several input fields for address and location data: 'Alamat' (Dk. Baranang Siang Indah No. 1), 'RT' (002), 'RW' (001), 'Kelurahan/Desa' (PULAU UNTUNG JAWA), 'Kecamatan' (KEPULAUAN SERIBU SELATAN), and 'Kota' (KABUPATEN KEPULAUAN SERIBU). There is a 'Status Keberlakuan' dropdown menu set to 'Berlaku' and a 'Tambah' button at the bottom.

Tampilan laman ubah keluarga

Setiap penduduk yang diubah memiliki NKK berdasarkan aturan pembuatan NKK. Jika NKK yang identik (memiliki tanggal pembuatan dan wilayah yang sama) ditemukan dan NKK baru berbeda dengan NKK lama, sistem akan meng-increment digit pada 4 digit terakhir NKK. Jika NKK identik dengan NKK lama, NKK tidak diubah.

The screenshot shows the same browser address. The page displays a large 'Sukses !' message. Below it, a smaller text reads: 'Keluarga dengan Nomor KK 3101012210170003 Berhasil diubah'.

Tampilan laman ubah keluarga berhasil

Langkah Pengerjaan :

1. Memodifikasi Interface CitizenService

Interface CitizenService memuat method terkait fitur pengubahan penduduk updateCitizen() dengan parameter CitizenModel penduduk yang ingin diubah.

2. Memodifikasi Interface CitizenMapper

Interface CitizenMapper memuat method terkait fitur perubahan penduduk. Method ini kemudian akan mengirimkan query ke database MySQL sesuai masukan object CitizenModel pada parameter dan memasukkan data pada tabel penduduk.

3. Memodifikasi Class CitizenServiceDatabase

Class CitizenServiceDatabase mengimplementasikan method updateCitizen() pada interface CitizenService. Method ini memanggil method updateCitizen() yang menerima parameter CitizenModel penduduk

4. Memodifikasi Class Controller

Class CitizenController memuat method yang menangani Request Mapping /penduduk/ubah/{NIK}. Method updateCitizen() mengembalikan laman ubah-penduduk.html yang memuat form perubahan penduduk. Method updateCitizenSubmit() menerima parameter data penduduk yang ingin diubah dan memproses converting data terkait Status Kematian dan Jenis Kelamin.

5. Membuat view

G. Mengubah Status Kematian Penduduk

Sistem dapat mengubah status kematian penduduk ketika tombol Nonaktifkan pada laman penduduk.html.



Tampilan menonaktifkan penduduk

Langkah Pengerjaan :

1. Memodifikasi Interface CitizenService
Interface CitizenService memuat method terkait fitur perubahan status kematian penduduk setWafat() dengan parameter CitizenModel penduduk yang ingin dinonaktifkan.
2. Memodifikasi Interface CitizenMapper
Interface CitizenMapper memuat method terkait fitur perubahan status kematian penduduk. Method ini kemudian akan mengirimkan query ke database MySQL sesuai masukan object CitizenModel pada parameter dan mengubah data isWafat pada tabel penduduk.
3. Memodifikasi Class CitizenServiceDatabase
Class CitizenServiceDatabase mengimplementasikan method setWafat() pada interface CitizenService. Method ini memanggil method setWafat() yang menerima parameter CitizenModel penduduk yang akan dinonaktifkan.
4. Memodifikasi Class Controller
Class CitizenController memuat method yang menangani Request Mapping /penduduk/mati. Method setWafat() menerima parameter data penduduk yang ingin dinonaktifkan. Handling untuk fitur menonaktifkan kartu keluarga yang anggota keluarganya telah wafat semua ditangani pada method viewFamily() untuk fitur 2.
5. Membuat view

H. Menampilkan Data Penduduk Berdasarkan Kota/Kabupaten, Kecamatan, dan Kelurahan Tertentu

Fitur ini menampilkan data penduduk berdasarkan kelurahan tertentu. User mengakses laman /penduduk/cari. User kemudian akan diarahkan ke laman pilih-kota.html untuk memilih kota, kemudian ke laman pilih-kecamatan.html dan pilih-kelurahan.html untuk memilih kecamatan dan kelurahan. Kemudian sistem akan menampilkan data semua penduduk pada kelurahan yang dipilih pada laman penduduk-kelurahan.html.

The screenshot shows a web browser at localhost:8080/penduduk/cari?kt=2&kc=5. The page title is 'Sistem Informasi Kependudukan'. Below the navigation bar, the text reads 'Lihat Data Penduduk Berdasarkan Kecamatan di KOTA JAKARTA SELATAN, Kecamatan CILANDAK'. There are three dropdown menus: 'Kota' with 'KOTA JAKARTA SELATAN' selected, 'Kecamatan' with 'CILANDAK' selected, and 'Kelurahan' with 'PONDOK LABU' selected. A 'Tambah' button is at the bottom left.

Tampilan memilih kelurahan setelah memilih kota dan kecamatan

The screenshot shows the same web browser at localhost:8080/penduduk/cari?kt=2&kc=5&kl=21. The page title is 'Sistem Informasi Kependudukan'. Below the navigation bar, the text reads 'Lihat Data Penduduk di Kelurahan PONDOK LABU, CILANDAK, KOTA JAKARTA SELATAN'. There is a search bar and a table with 10 entries. The table has columns: No., NIK, Nama, and Jenis Kelamin. The data is as follows:

No.	NIK	Nama	Jenis Kelamin
1	3171036802900001	Gawati Kamila Puspasari M TI	Perempuan
2	3171035909170001	Vicky Kusnawati	Perempuan
3	3171034910950001	Zizi Dewi Mulyani	Perempuan
4	3171032206560001	Adika Napitupulu	Laki-Laki
5	3171030403160001	Legawa Mahendra	Laki-Laki
6	3171032207160001	Ajiman Latupono	Laki-Laki
7	3171030205870001	Kardi Nashiruddin	Laki-Laki
8	3171036310880001	Shakila Uyainah	Perempuan
9	3171032211830001	Mujur Wakiman Siregar	Laki-Laki
10	3171033103910001	Putu Adriansyah	Laki-Laki

At the bottom, it says 'Showing 1 to 10 of 873 entries' and has pagination links: Previous, 1, 2, 3, 4, 5, ..., 88, Next.

Tampilan lihat data penduduk kelurahan

Langkah Pengerjaan :

1. Membuat Class Model untuk Kota, Kecamatan, Kelurahan
Membuat Model Class untuk Kota (CityModel.java), Kecamatan (KecamatanModel.java), dan Kelurahan (KelurahanModel.java) yang memuat atribut

terkait id, nama, jumlah penduduk, dan list entitas wilayah dibawahnya (list keluarga untuk kelurahan).

2. Memodifikasi Interface CitizenService

Interface CitizenService memuat method terkait fitur pemilihan kota, kecamatan, dan kelurahan parameter id masing-masing wilayah yang dipilih. Method mengembalikan Class Model wilayah yang dipilih.

3. Memodifikasi Interface CitizenMapper

Interface CitizenMapper memuat method terkait fitur pemilihan kota, kecamatan, dan kelurahan dari database. Selain itu, terdapat method selectPendudukKelurahan() yang mengembalikan data penduduk pada kelurahan dengan masukkan id_kelurahan. Object penduduk kemudian disimpan ke dalam list pada object KelurahanModel.

4. Memodifikasi Class CitizenServiceDatabase

Class CitizenServiceDatabase mengimplementasikan method terkait fitur pemilihan kota, kecamatan dan kelurahan pada interface CitizenService.

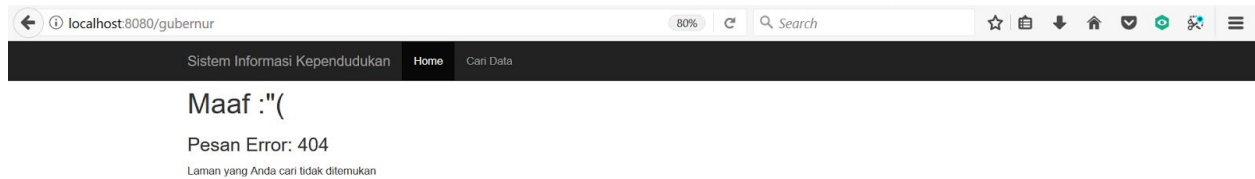
5. Memodifikasi Class Controller

Class Controller memuat method searchCitizen() yang memuat parameter id_kota, id_kecamatan, dan id_kelurahan. Method kemudian memproses input yang diterima dan mengembalikan laman pemilihan wilayah (pilih-kota.html, pilih-kecamatan.html, dan pilih-kelurahan.html).

6. Membuat view

I. Error Page

Sistem menyediakan fitur error page yang menampilkan pesan error ketika terjadi error. Error page yang tersedia adalah untuk jenis error 404 (Not Found), 500 (Internal Server), dan 400 (Bad Request).



Contoh pesan error

Langkah Pengerjaan :

1. Membuat Folder Error pada Folder Templates
Membuat folder error yang akan memuat laman html pesan error pada folder templates.
2. Membuat Error Page
Membuat error page 404.html, 500.html, dan 400.html di dalam folder error dan menampilkan pesan error sesuai jenis error

J. Statistik Kependudukan

Sistem menyediakan fitur statistik kependudukan yang mencakup data jumlah total penduduk dan keluarga. Selain itu terdapat data jumlah penduduk dan keluarga di setiap kota beserta rasio penduduk per keluarga.

The screenshot shows a web browser at localhost:8080 displaying the 'Sistem Informasi Kependudukan' application. The navigation bar includes 'Home' and 'Cari Data'. The main content area has a section 'Lihat Data Keluarga Berdasarkan NKK' with a search bar and a 'Lihat' button. Below this is the 'Statistik Kependudukan' section, which displays three summary statistics: 'Jumlah Penduduk : 281474', 'Jumlah Keluarga : 63492', and 'Rasio Penduduk/Keluarga : 4'. A table below these statistics shows data for six cities, with columns for 'Kota', 'Jumlah Penduduk', and 'Jumlah Keluarga'. The table is paginated, showing 'Showing 1 to 6 of 6 entries'.

Kota	Jumlah Penduduk	Jumlah Keluarga
Kabupaten Kepulauan Seribu	7210	1639
Kota Jakarta Barat	63229	14045
Kota Jakarta Pusat	43465	9873
Kota Jakarta Selatan	66472	15025
Kota Jakarta Timur	66006	15055
Kota Jakarta Utara	35092	7855

Tampilan statistik kependudukan

Langkah Pengerjaan :

1. Memodifikasi Interface CitizenMapper
Membuat method `countPenduduk()`, `countkeluarga()`, `countPendudukKota(id_kota)` dan `countKeluargaKota(id_kota)` yang mengembalikan jumlah penduduk atau keluarga.
2. Memodifikasi Interface CitizenService
Membuat method `countPenduduk()`, `countkeluarga()`, `countPendudukKota(id_kota)` dan `countKeluargaKota(id_kota)`. Method untuk penghitungan penduduk atau keluarga setiap kota menerima parameter `id_kota`.
3. Memodifikasi Class CitizenServiceDatabase
Mengimplementasikan method `countPenduduk()`, `countkeluarga()`, `countPendudukKota(id_kota)` dan `countKeluargaKota(id_kota)`. Method memanggil method terkait pada CitizenMapper.

4. Memodifikasi Controller

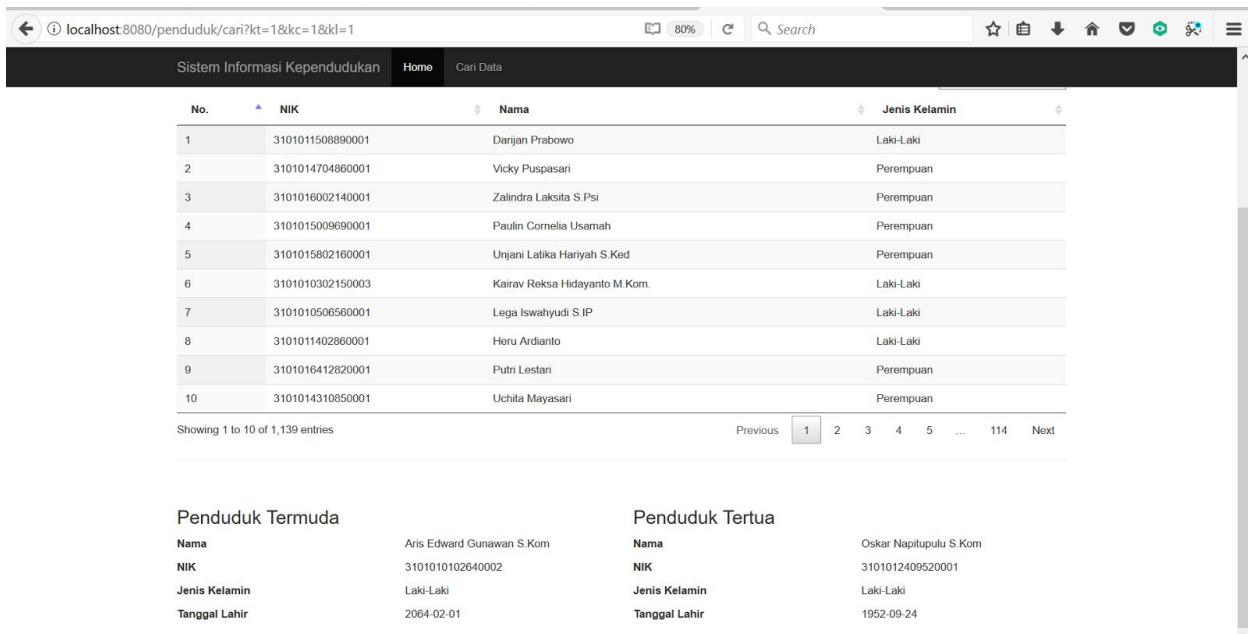
Memodifikasi method `index()` yang mengembalikan laman `index.html`. Pada controller dipanggil method terkait penghitungan jumlah penduduk dan keluarga yang kemudian akan disimpan dalam model terkait

5. Memodifikasi tampilan `index.html`

Memodifikasi laman `index.html` dengan menambahkan section Statistik Penduduk yang menampilkan tabel jumlah penduduk dan keluarga di setiap kota dan jumlah total di Jakarta.

K. Menampilkan Penduduk Tertua dan Termuda di Suatu Kelurahan

Sistem dapat menampilkan penduduk termuda dan tertua di suatu kelurahan saat user mengakses laman penduduk/cari dan menampilkan penduduk berdasarkan kelurahan.



The screenshot shows a web application interface for a population information system. The main part of the page displays a table of residents with columns for No., NIK, Nama, and Jenis Kelamin. Below this table is a pagination control showing 'Showing 1 to 10 of 1,139 entries' and a set of page numbers (1, 2, 3, 4, 5, ..., 114, Next). At the bottom of the page, there are two side-by-side tables. The left table is titled 'Penduduk Termuda' and shows details for Aris Edward Gunawan S.Kom. The right table is titled 'Penduduk Tertua' and shows details for Oskar Napitupulu S.Kom.

No.	NIK	Nama	Jenis Kelamin
1	310101150890001	Danjan Prabowo	Laki-Laki
2	3101014704860001	Vicky Puspasari	Perempuan
3	3101016002140001	Zalindra Laksita S.Psi	Perempuan
4	3101015009690001	Paulin Cornelia Usamah	Perempuan
5	3101015802160001	Unjani Latika Hariyah S.Ked	Perempuan
6	3101010302150003	Kairav Reksa Hidayanto M.Kom.	Laki-Laki
7	3101010506560001	Lega Iswahyudi S.IP	Laki-Laki
8	3101011402860001	Heru Ardianto	Laki-Laki
9	3101016412820001	Putri Lestari	Perempuan
10	3101014310850001	Uchita Mayasari	Perempuan

Showing 1 to 10 of 1,139 entries

Nama	Aris Edward Gunawan S.Kom
NIK	3101010102640002
Jenis Kelamin	Laki-Laki
Tanggal Lahir	2064-02-01

Nama	Oskar Napitupulu S.Kom
NIK	3101012409520001
Jenis Kelamin	Laki-Laki
Tanggal Lahir	1952-09-24

Tampilan penduduk tertua dan termuda

Langkah Pengerjaan :

1. Memodifikasi Interface CitizenMapper

Membuat method `selectOldest(id_kelurahan)` dan `selectYoungest(id_kelurahan)` yang mengembalikan jumlah penduduk tertua dan termuda di suatu kelurahan dari database.

2. Memodifikasi Interface CitizenService

Membuat Membuat method selectOldest(id_kelurahan) dan selectYoungest(id_kelurahan). Method untuk pencarian penduduk tertua dan termuda setiap kelurahan menerima parameter id_kelurahan.

3. Memodifikasi Class CitizenServiceDatabase

Mengimplementasikan method selectOldest(id_kelurahan) dan selectYoungest(id_kelurahan). Method memanggil method terkait pada CitizenMapper.

4. Memodifikasi Controller

Memodifikasi method searchCitizen() yang pada akhirnya akan mengembalikan laman penduduk-kelurahan.html. Pada method ini dipanggil method terkait pencarian penduduk termuda dan tertua yang kemudian hasilnya akan disimpan dalam model terkait

5. Memodifikasi tampilan penduduk-kelurahan.html

Memodifikasi laman penduduk-kelurahan.html dengan menambahkan section Penduduk Tertua dan Termuda yang menampilkan nama, tanggal lahir serta NIK dari penduduk termuda dan tertua.

II. OPTIMASI DATABASE DAN STRESS TESTING

Pada sistem ini, optimasi database dilakukan dengan menambahkan index pada atribut yang sering diakses pada database. Index dibuat di semua tabel. Berikut merupakan daftar index yang dibuat:

No.	Nama Index	Nama Tabel	Nama Atribut
1	idx_id_kecamatan	Kecamatan	id
2	idx_id_kota	Kecamatan	id_kota
3	idx_id_keluarga	Keluarga	id
4	idx_nomor_kk	Keluarga	nomor_kk
5	idx_id_kelurahan	Keluarga	id_keluarga
6	idx_id_kelurahan	Kelurahan	id
7	idx_id_kecamatan	Kelurahan	id_kecamatan
8	idx_id_kota	Kota	id
9	idx_nama_kota	Kota	nama_kota
10	idx_id_penduduk	Penduduk	id
11	idx_id_keluarga	Penduduk	id_keluarga

Berikut merupakan hasil Stress Testing pada fitur 1 Menampilkan Data Penduduk Berdasarkan NIK :

1) Spesifikasi Thread Properties yang digunakan



The screenshot shows a 'Thread Properties' dialog box with the following settings:

- Number of Threads (users): 50
- Ramp-Up Period (in seconds): 10
- Loop Count: ☐ Forever, ☒ 1
- ☐ Delay Thread creation until needed
- ☐ Scheduler

2) Informasi Path, Server dan Port Number untuk sampler HTTP Request

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=310101140517000 Content encoding:

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

Parameters **Body Data** **Files Upload**

Send Parameters With the Request:

Name	Value	Encode?	Include Equals?
------	-------	---------	-----------------

3) Hasil Stress Testing sebelum menggunakan index

File Edit Search Run Options Help

06:04:18 0 0/50

Test Plan

- Thread Group
- HTTP Request
- View Results in Table
- WondBench

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(ms)
1	16.48.32.684	Thread Group 1-32	HTTP Request	69376	✓	2967	147	69376	1
2	16.48.30.683	Thread Group 1-22	HTTP Request	71427	✓	2967	147	71427	1
3	16.48.34.684	Thread Group 1-43	HTTP Request	73560	✓	2967	147	73560	1
4	16.48.34.485	Thread Group 1-41	HTTP Request	74109	✓	2967	147	74109	1
5	16.48.34.684	Thread Group 1-42	HTTP Request	73950	✓	2967	147	73950	1
6	16.48.33.264	Thread Group 1-35	HTTP Request	79470	✓	2967	147	79470	1
7	16.48.30.252	Thread Group 1-20	HTTP Request	82513	✓	2967	147	82513	1
8	16.48.33.484	Thread Group 1-36	HTTP Request	79285	✓	2967	147	79285	1
9	16.48.28.002	Thread Group 1-9	HTTP Request	84698	✓	2967	147	84698	1
10	16.48.28.482	Thread Group 1-11	HTTP Request	84315	✓	2967	147	84315	1
11	16.48.32.284	Thread Group 1-30	HTTP Request	80527	✓	2967	147	80527	1
12	16.48.30.684	Thread Group 1-23	HTTP Request	81844	✓	2967	147	81844	1
13	16.48.27.881	Thread Group 1-8	HTTP Request	84949	✓	2967	147	84949	2
14	16.48.26.684	Thread Group 1-2	HTTP Request	124419	✓	2967	147	124419	1
15	16.48.29.003	Thread Group 1-14	HTTP Request	122034	✓	2967	147	122034	0
16	16.48.31.483	Thread Group 1-26	HTTP Request	119646	✓	2967	147	119646	1
17	16.48.27.285	Thread Group 1-5	HTTP Request	123855	✓	2967	147	123855	0
18	16.48.28.240	Thread Group 1-10	HTTP Request	123669	✓	2967	147	123669	2
19	16.48.33.084	Thread Group 1-34	HTTP Request	118859	✓	2967	147	118859	1
20	16.48.32.684	Thread Group 1-33	HTTP Request	120820	✓	2967	147	120820	1
21	16.48.31.883	Thread Group 1-28	HTTP Request	124180	✓	2967	147	124180	1
22	16.48.33.885	Thread Group 1-38	HTTP Request	192230	✓	2967	147	192230	1
23	16.48.34.285	Thread Group 1-40	HTTP Request	200450	✓	2967	147	200450	1
24	16.48.35.485	Thread Group 1-45	HTTP Request	203205	✓	2967	147	203205	1
25	16.48.27.681	Thread Group 1-7	HTTP Request	226446	✓	2967	147	226446	2
26	16.48.26.440	Thread Group 1-1	HTTP Request	227682	✓	2967	147	227682	2
27	16.48.30.083	Thread Group 1-19	HTTP Request	224075	✓	2967	147	224074	1
28	16.48.27.083	Thread Group 1-4	HTTP Request	227095	✓	2967	147	227095	1
29	16.48.32.486	Thread Group 1-31	HTTP Request	221705	✓	2967	147	221703	0
30	16.48.36.285	Thread Group 1-50	HTTP Request	217924	✓	2967	147	217924	1
31	16.48.28.681	Thread Group 1-12	HTTP Request	225528	✓	2967	147	225528	2
32	16.48.32.083	Thread Group 1-29	HTTP Request	222139	✓	2967	147	222139	1
33	16.48.29.882	Thread Group 1-18	HTTP Request	224341	✓	2967	147	224340	1

☐ Scroll automatically? ☐ Child samples? No of Samples 50 Latest Sample 249138 Average 173096 Deviation 68680

Diperoleh data sebagai berikut :

Average : 173096

Deviation : 68680

4) Hasil Stress Testing setelah menggunakan index (pada fitur dan spesifikasi yang sama)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Send Bytes	Latency	Connect Time(ms)
1	17.32.21.961	Thread Group 1-13	HTTP Request	48021	Success	2900	147	46020	1
2	17.32.21.561	Thread Group 1-11	HTTP Request	46526	Success	2900	147	46526	1
3	17.32.23.962	Thread Group 1-23	HTTP Request	44137	Success	2900	147	44137	1
4	17.32.24.961	Thread Group 1-28	HTTP Request	43222	Success	2900	147	43222	1
5	17.32.23.361	Thread Group 1-20	HTTP Request	44877	Success	2900	147	44877	1
6	17.32.26.161	Thread Group 1-44	HTTP Request	40113	Success	2900	147	40113	2
7	17.32.21.161	Thread Group 1-9	HTTP Request	47128	Success	2900	147	47128	1
8	17.32.25.561	Thread Group 1-31	HTTP Request	42791	Success	2900	147	42791	2
9	17.32.22.361	Thread Group 1-15	HTTP Request	46003	Success	2900	147	46003	1
10	17.32.28.361	Thread Group 1-45	HTTP Request	49012	Success	2900	147	49012	1
11	17.32.26.162	Thread Group 1-34	HTTP Request	42273	Success	2900	147	42273	1
12	17.32.29.362	Thread Group 1-50	HTTP Request	39084	Success	2900	147	39084	1
13	17.32.22.561	Thread Group 1-16	HTTP Request	45885	Success	2900	147	45885	1
14	17.32.21.764	Thread Group 1-12	HTTP Request	46597	Success	2900	147	46597	1
15	17.32.27.162	Thread Group 1-39	HTTP Request	41315	Success	2900	147	41315	1
16	17.32.20.961	Thread Group 1-4	HTTP Request	47530	Success	2900	147	47530	1
17	17.32.20.762	Thread Group 1-7	HTTP Request	47748	Success	2900	147	47748	1
18	17.32.20.163	Thread Group 1-4	HTTP Request	46359	Success	2900	147	46359	2
19	17.32.23.561	Thread Group 1-21	HTTP Request	44951	Success	2900	147	44951	3
20	17.32.21.362	Thread Group 1-10	HTTP Request	47173	Success	2900	147	47173	1
21	17.32.26.372	Thread Group 1-5	HTTP Request	48172	Success	2900	147	48172	1
22	17.32.27.761	Thread Group 1-42	HTTP Request	49807	Success	2900	147	49807	1
23	17.32.20.575	Thread Group 1-6	HTTP Request	48015	Success	2900	147	48015	1
24	17.32.19.762	Thread Group 1-2	HTTP Request	48891	Success	2900	147	48891	1
25	17.32.19.605	Thread Group 1-1	HTTP Request	49148	Success	2900	147	49148	43
26	17.32.22.762	Thread Group 1-17	HTTP Request	46003	Success	2900	147	46003	3
27	17.32.27.361	Thread Group 1-40	HTTP Request	41426	Success	2900	147	41426	1
28	17.32.27.951	Thread Group 1-43	HTTP Request	49907	Success	2900	147	49907	1
29	17.32.25.364	Thread Group 1-30	HTTP Request	55291	Success	2900	147	55291	1
30	17.32.22.161	Thread Group 1-14	HTTP Request	58514	Success	2900	147	58514	1
31	17.32.23.161	Thread Group 1-19	HTTP Request	57524	Success	2900	147	57524	1
32	17.32.22.961	Thread Group 1-18	HTTP Request	57844	Success	2900	147	57844	1
33	17.32.18.962	Thread Group 1-3	HTTP Request	60857	Success	2900	147	60857	1

☐ Scroll automatically? ☐ Child samples? No of Samples: 50 Latest Sample: 67922 Average: 52976 Deviation: 9817

Diperoleh data sebagai berikut :

Average : 52976

Deviation : 9817

III. IMPLEMENTASI MVC

Pada Sistem ini telah diimplementasikan konsep Model-View-Controller (MVC). Pada sistem ini, dibuat model yang menyimpan atribut terkait entitas yang terkait dengan fungsionalitas sistem, view yang merupakan tampilan dari fitur-fitur sistem, dan controller yang memproses mapping, input data serta instansiasi model.

Model yang dibuat pada sistem ini, antara lain CitizenModel (Penduduk), FamilyModel (Keluarga), KelurahanModel, KecamatanModel, dan CityModel. Masing-masing model menyimpan atribut yang terkait atribut pada tabel di database ditambah atribut berisi entitas yang dimiliki (Contoh: FamilyModel memiliki atribut List CitizenModel anggota keluarga).

Pada Sistem ini terdapat interface CitizenMapper yang berperan memetakan data ke database. Query MySQL untuk mencari, memperoleh, mengubah, dan menambah data pada database dibuat di interface CitizenMapper.