

WRITE UP TUGAS 1 APAP

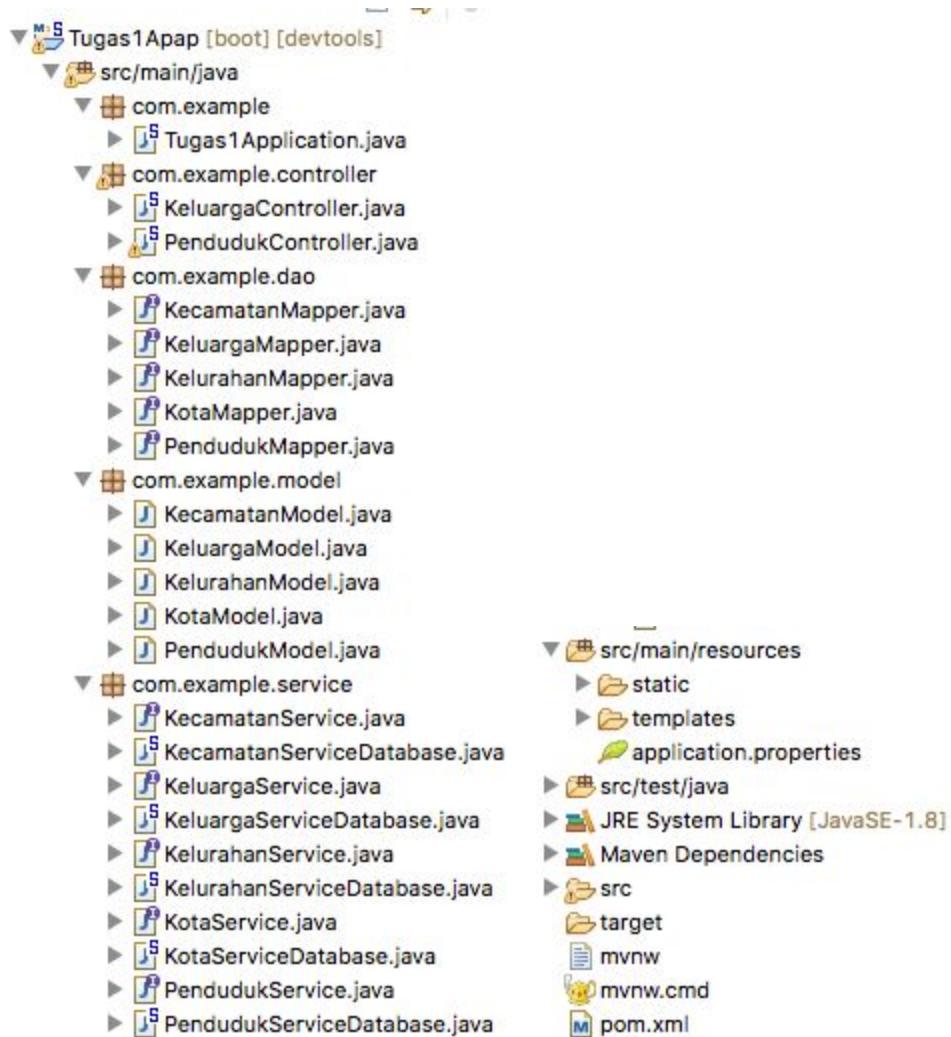
Mayanggita Rachmaryza - 1506732974 - Kelas B

A. Proses Pengembangan Tugas 1

Proses pengembangan tugas 1 ini dimulai dengan membuat model terlebih dahulu. Setelah itu saya membuat *front end* tanpa *bootstrap*, lalu membuat *controller* untuk mengatur *flow* dari program. Kemudian adalah membuat *mapper* dan *service*. Lalu mengimplementasikan fitur yang dibutuhkan satu persatu kemudian melakukan *testing* dan *debugging*. Dan yang terakhir adalah merapikan tampilan dengan menambahkan *bootstrap*.

1. Struktur

Struktur dari folder project tugas 1 adalah sebagai berikut:



2. *Package*

Project ini terdiri dari 5 *package*, yaitu:

- ***Package com.example***

Berisi `Tugas1Application.java` yang berfungsi sebagai *main class* dari *project*.

- ***Package com.example.controller***

Berisi *controller* untuk mengatur *flow* dari program. *Controller* digunakan untuk menerima url, kemudian menambahkan informasi yang ada ke dalam Model dan menampilkannya kembali ke halaman yang sesuai. *Controller* juga berfungsi untuk memanggil *method-method* yang ada di *service*. Pada *project* ini, saya memiliki 2 *controller*, yaitu `PendudukController.java` untuk yang berhubungan dengan penduduk dan `KeluargaController.java` untuk yang berhubungan dengan keluarga.

- ***Package com.example.dao***

Package ini berisi *interface mapper* untuk menghubungkan *project* dengan *database* yang ada (*database siduk*). Pada *project* ini, terdapat 5 *mapper*, yaitu:

1. PendudukMapper: berfungsi untuk mengambil informasi mengenai penduduk dari *database*
2. KeluargaMapper: berfungsi untuk mengambil informasi mengenai keluarga dari *database*
3. KecamatanMapper: berfungsi untuk mengambil informasi mengenai kecamatan dari *database*
4. KelurahanMapper: berfungsi untuk mengambil informasi mengenai kelurahan dari *database*
5. KotaMapper: berfungsi untuk mengambil informasi mengenai kota dari *database*

- **Package com.example.model**

Package ini berisi model-model untuk menampung informasi dari *database*. Pada *project* ini, terdapat 5 model, yaitu:

1. PendudukModel: berfungsi untuk menampung informasi dari database (tabel penduduk)
2. KeluargaModel: berfungsi untuk menampung informasi dari database (tabel keluarga)
3. KecamatanModel: berfungsi untuk menampung informasi dari database (tabel kecamatan)
4. KelurahanModel: berfungsi untuk menampung informasi dari database (tabel kelurahan)
5. KotaModel: berfungsi untuk menampung informasi dari database (tabel kota)

- **Package com.example.service**

Pada *Package* ini, terdapat 5 *interface* dan 5 kelas yang mengimplementasi *interface* tersebut (masing-masing 1), yaitu:

1. PendudukServiceDatabase yang mengimplementasi *interface* PendudukService
2. KeluargaServiceDatabase yang mengimplementasi *interface* KeluargaService
3. KecamatanServiceDatabase yang mengimplementasi *interface* KecamatanService
4. KelurahanServiceDatabase yang mengimplementasi *interface* KelurahanService
5. KotaServiceDatabase yang mengimplementasi *interface* KotaService

3. Implementasi MVC

- **Model**

Model merupakan kelas-kelas yang menggambarkan struktur data dari *database*. Yang termasuk *model* adalah kelas-kelas yang berada dalam Package Model (com.example.model).

- **View**

View merupakan kelas-kelas yang merepresentasikan *interface* dari program. Yang termasuk *view* adalah *file* di dalam src/main/resources, yaitu di *folder static* dan juga *templates*.

- **Controller**

Controller merupakan kelas-kelas yang mengatur *flow* dari program, yaitu kelas-kelas dalam *package controller, service* dan juga *dao*.

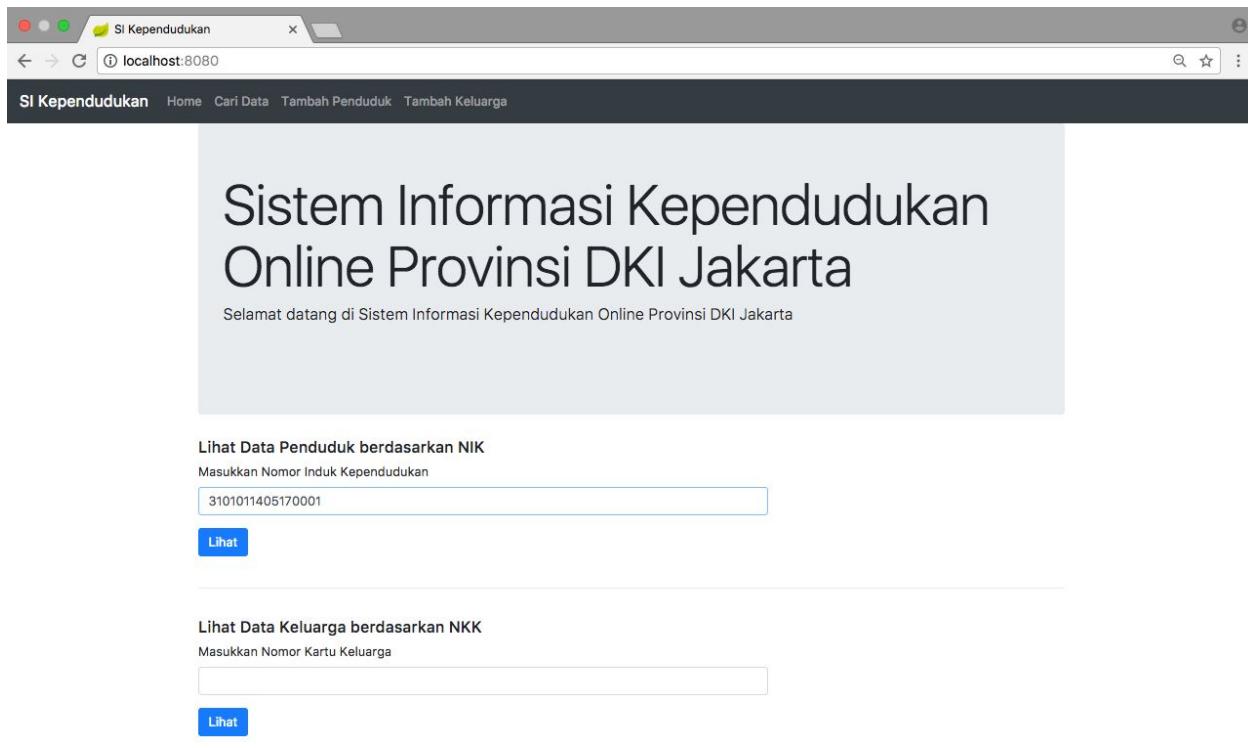
B. Penjelasan Fitur

1. Tampilkan Data Penduduk Berdasarkan NIK

Initial page: /

Form Request: GET, /penduduk

Response Page: /penduduk?nik={NIK}



Lihat data penduduk - 3101011405170001

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

[Nonaktifkan](#)

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Pada halaman index.html, terdapat *form* untuk menginput NIK penduduk yang akan dilihat datanya. *Form* ini dibuat dengan *method* “GET” yang akan meneruskan NIK yang sudah diinput kepada *controller*, yaitu PendudukController. Selanjutnya, controller akan mengecek apakah NIK nya ada atau tidak, jika tidak ada, maka akan menampilkan halaman *error*, jika NIK nya ada, maka data penduduk sesuai NIK tersebut akan diambil untuk kemudian menginisialisasi keluarga, kelurahan, kecamatan dan kota.

```

@RequestMapping("/penduduk")
public String pendudukViewNIK(Model model, @RequestParam(value="nik", required=true) String nik) {
    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
    if (penduduk == null) {
        model.addAttribute("errorMessage", "Penduduk dengan NIK " + nik + " tidak ditemukan");
        return "error/error404";
    } else {
        KeluargaModel keluarga = keluargaDAO.selectKeluargaById(penduduk.getId_keluarga());
        KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
        KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(kelurahan.getId_kecamatan());
        KotaModel kota = kotaDAO.selectKota(kecamatan.getId_kota());

        model.addAttribute("penduduk", penduduk);
        model.addAttribute("keluarga", keluarga);
        model.addAttribute("kelurahan", kelurahan);
        model.addAttribute("kecamatan", kecamatan);
        model.addAttribute("kota", kota);
        return "penduduk-detail";
    }
}

```

Setelah itu, controller akan memanggil service untuk menjalankan fungsi selectKeluargaById di KeluargaService (interface) dan KeluargaServiceDatabase, selectKelurahan di KelurahanService (interface) dan KelurahanServiceDatabase selectKecamatan di KecamatanService (interface) dan KecamatanServiceDatabase dan selectKota di KotaService dan KotaService Database. Dimana, method-method yang ada pada kelas tersebut akan memanggil kelas mapper (keluargaMapper, kelurahanMapper, kecamatanMapper, dan kotaMapper) untuk mencari data nya di database. Kemudian mengembalikan halaman penduduk-detail

2. Tampilkan Data Keluarga Beserta Daftar Anggotanya Berdasarkan Nomor KK

Initial Page: /

Form Request: GET, /keluarga

Response Page: /keluarga?nkk={NKK}

Lihat Data Penduduk berdasarkan NIK
Masukkan Nomor Induk Kependudukan

Lihat

Lihat Data Keluarga berdasarkan NKK
Masukkan Nomor Kartu Keluarga

Lihat

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Lihat Data Keluarga - 3101010101050001

NKK	:	3101010101050001
Alamat	:	Jln. Cikapayang No. 594
RT/RW	:	149/090
Kelurahan/Desa	:	PULAU PARI
Kecamatan	:	KEPULAUAN SERIBU SELATAN
Kota	:	KABUPATEN KEPULAUAN SERIBU

Show 10 entries Search:

No	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status dalam Kel
1	Taswir Rajata	3101010303890001	Laki-laki	Jakarta	1989-03-03	Islam	KARYAWAN HONORER	Kawin	Kepala Keluarga
2	Mulyono Hutagalung S.Gz	3101011408150001	Laki-laki	Jakarta	2015-08-14	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak
3	Yuliana Wahyuni	3101016312160001	Perempuan	Jakarta	2016-12-23	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak
4	Ifa Novitasari	3101014712830001	Perempuan	Jakarta	1983-12-07	Islam	PETANI/PEKEBUN	Kawin	Istri

Showing 1 to 4 of 4 entries Previous Next

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Sama seperti fitur pertama, pada halaman index.html, terdapat *form* untuk menginput NIK penduduk yang akan dilihat datanya. *Form* ini dibuat dengan *method* “GET” yang akan meneruskan NKK yang sudah diinput kepada *controller*, yaitu KeluargaController. Selanjutnya, controller akan mengecek apakah NKK nya ada atau tidak, jika tidak ada, maka akan menampilkan halaman *error*, jika NKK nya ada, maka data penduduk sesuai NKK tersebut akan diambil untuk kemudian menginisialisasi list anggota keluarga, kelurahan, kecamatan dan kota.

```

@RequestMapping("/keluarga")
public String viewKeluargaSubmit(Model model, @RequestParam(value="nkk", required=false) String nkk) {
    KeluargaModel keluarga = keluargaDAO.selectKeluarga(nkk);
    if (keluarga == null) {
        model.addAttribute("errorMessage", "Keluarga dengan NKK " + nkk + " tidak ditemukan");
        return "error/error404";
    } else {
        List<PendudukModel> anggotaKeluarga = pendudukDAO.selectAnggotaKeluarga(keluarga.getId());
        KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
        KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(kelurahan.getId_kecamatan());
        KotaModel kota = kotaDAO.selectKota(kecamatan.getId_kota());

        model.addAttribute("keluarga", keluarga);
        model.addAttribute("anggotaKeluarga", anggotaKeluarga);
        model.addAttribute("kelurahan", kelurahan);
        model.addAttribute("kecamatan", kecamatan);
        model.addAttribute("kota", kota);

        return "keluarga-detail";
    }
}

```

Setelah itu, controller akan memanggil service untuk menjalankan fungsi seletAnggotaKeluarga di PendudukService (interface) dan PendudukServiceDatabase, selectKelurahan di KelurahanService (interface) dan KelurahanServiceDatabase selectKecamatan di KecamatanService (interface) dan KecamatanServiceDatabase dan selectKota di KotaService dan KotaService Database. Dimana, method-method yang ada pada kelas tersebut akan memanggil kelas mapper (pendudukMapper, kelurahanMapper, kecamatanMapper, dan kotaMapper) untuk mencari data nya di database. Kemudian mengembalikan halaman keluarga-detail.

3. Menambahkan Penduduk Baru Sebagai Anggota Keluarga

Initial Page: /penduduk/tambah

Form Request: POST, /penduduk/tambah

Response Page: /penduduk/tambah

Pada fitur ini, buka halaman “/penduduk/tambah” (terdapat di navbar). Maka akan terbuka halaman seperti ini, lalu isi semua field pada form.

The screenshot shows a web browser window with the title "Tambah Penduduk". The URL in the address bar is "localhost:8080/penduduk/tambah". The page has a dark header with the text "SI Kependudukan" and navigation links "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". Below the header, the main content area has a title "Tambah Penduduk". The form consists of several input fields:

- Masukkan nama: An input field for entering a name.
- Tempat Lahir: An input field for entering a birthplace.
- Tanggal Lahir: An input field for entering a birthdate, with a placeholder "mm/dd/yyyy".
- Jenis Kelamin: A dropdown menu currently set to "Perempuan".
- Golongan Darah: A dropdown menu currently set to "A+".
- Agama: An input field for entering a religion.
- Status Perkawinan: A dropdown menu currently set to "Kawin".

The screenshot shows a web browser window with the URL localhost:8080/penduduk/tambah/. The page title is "Tambah Penduduk". The form fields are as follows:

- Jenis Kelamin: Perempuan
- Golongan Darah: A+
- Agama:
- Status Perkawinan: Kawin
- Pekerjaan:
- Kewarganegaraan: WNI
- Status Kematian: Hidup
- Id Keluarga:
- Status dalam Keluarga:

A blue "Simpan" button is at the bottom left, and a footer note at the bottom right reads "Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza."

The screenshot shows a web browser window with the URL localhost:8080/penduduk/tambah/. The page title is "Tambah Penduduk". The navigation bar includes links for "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". The form fields are as follows:

- Masukkan nama: assy||
- Tempat Lahir: Bogor
- Tanggal Lahir: 10/10/2017
- Jenis Kelamin: Perempuan
- Golongan Darah: B+
- Agama: Islam
- Status Perkawinan: Belum Kawin

Jenis Kelamin
Perempuan

Golongan Darah
B+

Agama
Islam

Status Perkawinan
Belum Kawin

Pekerjaan
mahasiswa

Kewarganegaraan
WNI

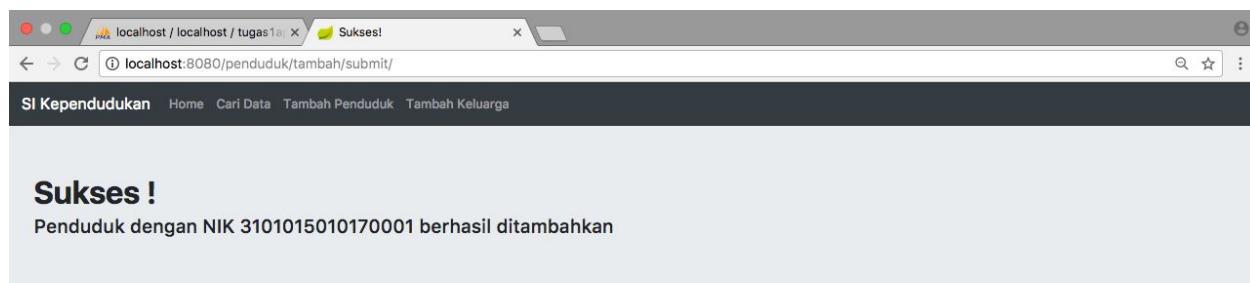
Status Kematian
Hidup

Id Keluarga
2

Status dalam Keluarga
anak

Simpan

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.



Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

```
@RequestMapping("/penduduk/tambah")
public String add (Model model)
{
    PendudukModel penduduk = new PendudukModel();
    model.addAttribute("penduduk", penduduk);
    return "penduduk-add";
}
```

Pada saat membuka halaman “/penduduk/tambah” , PendudukController akan menginisialisasi penduduk model terlebih dahulu sehingga nanti bisa di-passing menggunakan object. Setelah itu, menampilkan halaman penduduk-add.

```
@RequestMapping("/penduduk/tambah/submit")
public String addPendudukSubmit (
    @ModelAttribute PendudukModel penduduk, Model model)
{
    KeluargaModel keluarga = keluargaDAO.selectKeluargaById(penduduk.getId_keluarga());
    KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
    KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(kelurahan.getId_kecamatan());

    String kode = kecamatan.getKode_kecamatan();
    kode = kode.substring(0, 6);
    String nik = kode;

    String tanggal_lahir = penduduk.getTanggal_lahir();
    String[] tanggal_nik = tanggal_lahir.split("-");
    String tahun = tanggal_nik[0];
    String bulan = tanggal_nik[1];
    String tanggal = tanggal_nik[2];

    if(penduduk.getJenis_kelamin().equals("1")) {
        int temp = Integer.parseInt(tanggal);
        temp += 40;
        tanggal = "" + temp;
    }

    nik += tanggal + bulan + tahun.substring(2);

    List<PendudukModel> listNikSama = pendudukDAO.listNikSama(nik + "%");
    if(listNikSama.isEmpty()) {
        nik += "0001";
    } else {
        PendudukModel temp = listNikSama.get(listNikSama.size()-1);
        String nikTambah = temp.getNik().substring(12);
        int temp2 = Integer.parseInt(nikTambah);
        temp2++;

        String temp3 = "" + temp2;
        for(int i = 0; i < 4-temp3.length(); i++) {
            nik += "0";
        }
        nik += temp3;
    }
}

penduduk.setNik(nik);
pendudukDAO.addPenduduk(penduduk);
model.addAttribute("nik", nik);
return "penduduk-add-success";
}
```

Form yang terdapat di halaman penduduk-add memiliki method POST dengan action “penduduk/tambah/submit”. Setelah semua field diisi, maka halaman tersebut akan mengirimkan semua data ke controller. Controller kemudian akan menginisialisasi keluarga, kelurahan dan

kecamatan, yang diperlukan untuk men-generate NIK penduduk baru. Kemudian akan memanggil service, service database dan mapper untuk menjalankan method yang dipanggil di pendudukController.

Untuk mendapatkan NIK baru, gunakan kode_kecamatan, kemudian di-sub string 0 sampai 6 untuk mendapatkan 6 digit pertama dari kecamatan. Setelah itu, ambil tanggal_lahir penduduk, kemudian split string tanggal_lahir (menggunakan "-") menjadi tahun, bulan dan tanggal secara berurutan. Lalu cek apakah jenis kelamin penduduk perempuan atau laki-laki, apabila perempuan, maka tanggal_lahir akan ditambah 40. Kemudian untuk menentukan 4 digit terakhir, harus dicek apakah sudah ada penduduk yang berdomisili di tempat yang sama dan tanggal lahir yang sama (memiliki NIK dengan 12 digit pertama yang sama dengan penduduk baru ini). Apabila belum ada maka 4 digit terakhir dari NIK penduduk baru adalah 0001, namun jika sudah ada maka akan seperti nomor urut.

Setelah itu, maka set NIK dari penduduk baru, kemudian panggil penduduk service untuk menjalankan method addPenduduk yang nantinya memanggil pendudukServiceDatabase dan pendudukMapper. Dan terakhir, akan menampilkan halaman penduduk-add-success.

4. Menambahkan Keluarga Baru

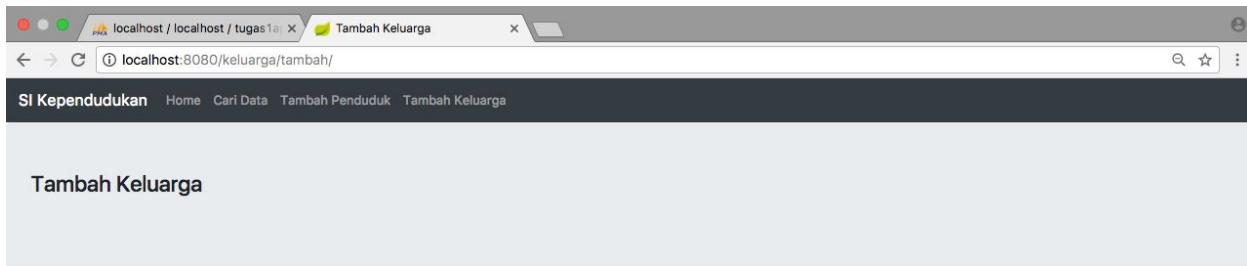
Initial Page: /keluarga/tambah

Form Request: POST, /keluarga/tambah

Response Page: /keluarga/tambah

Pada fitur ini, buka halaman “/keluarga/tambah” (terdapat di navbar). Maka akan terbuka halaman seperti ini, lalu isi semua field pada form.

Untuk field kota, kecamatan dan kelurahan, saya menggunakan *dropdown* seperti ini untuk menghindari user yang salah menginput data



Alamat:

jalan raya pari

RT

42

RW

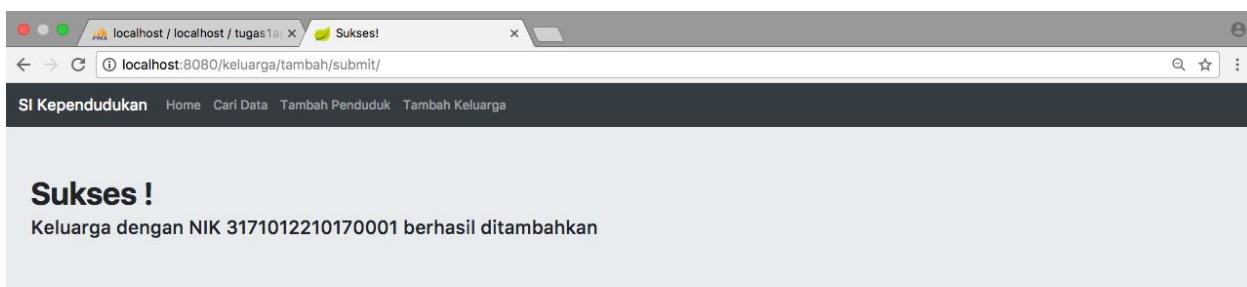
19

Kota / Kecamatan / Kelurahan

KOTA JAKARTA SELATAN / JAGAKARSA / CIGANJUR

Simpan

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.



Sukses !

Keluarga dengan NIK 3171012210170001 berhasil ditambahkan

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

```

63 @RequestMapping("/keluarga/tambah")
64 public String add(Model model)
65 {
66     KeluargaModel keluarga = new KeluargaModel();
67     keluarga.setId_kelurahan("");
68     model.addAttribute("keluarga", keluarga);
69
70     List<KelurahanModel> kelurahan = kelurahanDAO.selectAllKelurahan();
71     List<KecamatanModel> kecamatan = kecamatanDAO.selectAllKecamatan();
72     List<KotaModel> kota = kotaDAO.selectAllKota();
73     for(int i = 0; i < kelurahan.size(); i++){
74         for(int j = 0; j < kecamatan.size(); j++){
75             for(int k = 0 ; k < kota.size(); k++){
76                 if(kelurahan.get(i).getId_kecamatan().equals(kecamatan.get(j).getId())){
77                     if(kecamatan.get(j).getId_kota().equals(kota.get(k).getId())){
78                         String namaKelurahan = kota.get(k).getNama_kota() +
79                             " / " + kecamatan.get(j).getNama_kecamatan() +
80                             " / " + kelurahan.get(i).getNama_kelurahan();
81                         kelurahan.get(i).setNama_kelurahan(namaKelurahan);
82                     }
83                 }
84             }
85         }
86     }
87     model.addAttribute("kelurahan", kelurahan);
88     return "keluarga-add";
89 }

```

Hampir sama dengan fitur add penduduk, pada saat membuka halaman “/keluarga/tambah”, KeluargaController akan menginisialisasi keluarga model terlebih dahulu sehingga nanti bisa di-passing menggunakan object. Setelah itu, akan ditampilkan halaman keluarga-add. Kode pada baris 70-87 digunakan untuk menampilkan dropdown kota/kecamatan/kelurahan.

```
@RequestMapping("/keluarga/tambah/submit")
public String addKeluargaSubmit (
    @ModelAttribute KeluargaModel keluarga, Model model)
{
    KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());

    String nkk = kelurahan.getKode_kelurahan().substring(0, 6);

    String[] tanggal_isi = java.time.LocalDate.now().toString().split("-");
    nkk += tanggal_isi[2] + tanggal_isi[1] + tanggal_isi[0].substring(2);

    List<KeluargaModel> listNkkSama = keluargaDAO.listNkkSama(nkk + "%");
    if(listNkkSama.isEmpty()) {
        nkk += "0001";
    }else {
        KeluargaModel temp = listNkkSama.get(listNkkSama.size()-1);
        String nkkTambah = temp.getNomor_kk().substring(12);
        int temp2 = Integer.parseInt(nkkTambah);
        temp2++;

        String temp3 = "" + temp2;
        for(int i = 0; i < 4-temp3.length(); i++) {
            nkk += "0";
        }
        nkk += temp3;
    }

    keluarga.setNomor_kk(nkk);
    keluarga.setIs_tidak_berlaku("0");

    String rt = keluarga.getRT();
    String temp = "";
    for(int i = 0; i < 3-rt.length(); i++) {
        temp += "0";
    }
    rt = temp + rt;

    String rw = keluarga.getRW();
    String temp2 = "";
    for(int i = 0; i < 3-rw.length(); i++) {
        temp2 += "0";
    }
    rw = temp2 + rw;

    keluarga.setRT(rt);
    keluarga.setRW(rw);

    keluargaDAO.addKeluarga(keluarga);
    model.addAttribute("nkk", nkk);
    return "keluarga-add-success";
}
```

Form yang terdapat di halaman keluarga-add memiliki method POST dengan action “keluarga/tambah/submit”. Setelah semua field diisi, maka halaman tersebut akan mengirimkan semua data ke controller. Controller kemudian akan menginisialisasi kelurahan, yang diperlukan untuk men-generate NKK keluarga baru. Kemudian akan memanggil service, service database dan mapper untuk menjalankan method yang dipanggil di keluargaController.

Untuk mendapatkan NKK baru, ambil 6 digit pertama dari kode_kelurahan yang ada dengan melakukan substring. Kemudian mengambil tanggal hari ini sebagai tanggal pengisian untuk 4 digit berikutnya (menggunakan java.time.LocalDate.now()), kemudian di split untuk mendapatkan format tanggal yang diinginkan (hh/bb/tt). Kemudian untuk menentukan 4 digit terakhir, harus dicek apakah sudah ada keluarga yang berdomisili di tempat yang sama dan tanggal terbit yang sama (memiliki NKK dengan 12 digit pertama yang sama dengan keluarga yang baru ini). Apabila belum ada maka 4 digit terakhir dari NKK penduduk baru adalah 0001, namun jika sudah ada maka akan seperti nomor urut. Untuk RT dan RW, input harus dibuat menjadi 3 digit, maka harus di looping untuk menambahkan nol.

Setelah itu, maka set NKK, RT, RW dari keluarga baru, kemudian panggil keluarga service untuk menjalankan method addKeluarga yang nantinya memanggil keluargaServiceDatabase dan keluargaMapper. Dan terakhir, akan menampilkan halaman keluarga-add-success.

5. Mengubah Data Penduduk

Initial Page: /penduduk/ubah/{NIK}

Form Request: POST, /penduduk/ubah/{NIK}

ResponsePage: /penduduk/ubah/{NIK}

Pada saat mengakses /penduduk/ubah/{NIK}, akan langsung muncul data dari penduduk tersebut pada field yang ada. Dapat dilihat seperti gambar di bawah ini.

The screenshot shows a web application window titled "Ubah Data Penduduk". The URL in the address bar is "localhost:8080/penduduk/ubah/3101015210200001". The page header includes "SI Kependudukan" and navigation links for "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". The main content area is titled "Ubah Data Penduduk". Below it, there is a form with the following fields and their values:

Nama	assyi
Tempat Lahir	Bogor
Tanggal Lahir	10/12/2017
Jenis Kelamin	Perempuan
Golongan Darah	A
Agama	Islam
Status Perkawinan	Belum Kawin

The screenshot shows a web application window titled "Ubah Data Penduduk". The URL in the address bar is "localhost:8080/penduduk/ubah/3101015010170001". The form contains the following fields:

- Jenis Kelamin: Perempuan
- Golongan Darah: A
- Agama: Islam
- Status Perkawinan: Belum Kawin
- Pekerjaan: mahasiswa
- Kewarganegaraan: WNI
- Status Kematian: Hidup
- Id Keluarga: 2
- Status dalam Keluarga: anak

At the bottom left is a blue "Simpan" button, and at the bottom center is the text "Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza."

Misalnya, mengubah tanggal lahir dan id_keluarga.

Tanggal Lahir

10/12/2017

Id Keluarga

1

Klik submit

The screenshot shows a web application window titled "Sukses!". The URL in the address bar is "localhost:8080/penduduk/ubah/3101015010170001". The header includes "SI Kependudukan" and navigation links for Home, Cari Data, Tambah Penduduk, and Tambah Keluarga. The main content area displays the message "Sukses!" and "Penduduk dengan NIK 3101015010170001 berhasil diubah". At the bottom center is the text "Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza."

```
@RequestMapping("/penduduk/ubah/{nik}")
public String ubah (Model model, @PathVariable(value="nik") String nik) {
    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);

    if(penduduk == null) {
        model.addAttribute("errorMessage", "Penduduk dengan NIK " + nik + " tidak ditemukan, "
            + "mohon cek kembali Nomor Induk Kependudukan Anda.");
        return "error/error404";
    }else {
        model.addAttribute("penduduk", penduduk);
        return "penduduk-update";
    }
}
```

Pada saat membuka halaman “/penduduk/ubah” , PendudukController akan menginisialisasi penduduk model terlebih dahulu. Setelah itu, controller akan memeriksa, apakah NIK nya ada atau tidak di database, jika tidak, maka akan menampilkan error, jika ada, maka akan ditampilkan halaman penduduk-update.

```
@RequestMapping(value="/penduduk/ubah/{nik}", method = RequestMethod.POST)
public String ubahPendudukSubmit(Model model, @ModelAttribute PendudukModel penduduk) {
    PendudukModel pendudukLama = pendudukDAO.selectPenduduk(penduduk.getNik());
    String nikBaru = pendudukLama.getNik();

    //mengganti tanggal lahir
    if(!pendudukLama.getTanggal_lahir().equals(penduduk.getTanggal_lahir())) {
        String tanggal_lahir = penduduk.getTanggal_lahir();
        String[] tgl_lahir = tanggal_lahir.split("-");
        String tahun = tgl_lahir[0];
        String bulan = tgl_lahir[1];
        String tanggal = tgl_lahir[2];
        int temp;

        //jenis kelamin perempuan
        if(penduduk.getJenis_kelamin().equals("1")) {
            temp = Integer.parseInt(tanggal) + 40;
            tanggal = "" + temp;
        }
        nikBaru = nikBaru.substring(0,6) + tanggal + bulan + tahun + nikBaru.substring(12);

    }else {
        //jenis kelamin di update
        if(!pendudukLama.getJenis_kelamin().equals(penduduk.getJenis_kelamin())) {
            int tanggal = Integer.parseInt(nikBaru.substring(6, 8));
            //menambah menjadi laki-laki
            if(penduduk.getJenis_kelamin().equals("0")) {
                tanggal = tanggal-40;
            }else {
                tanggal += 40;
            }
            nikBaru = nikBaru.substring(0, 6) + tanggal + nikBaru.substring(8);
        }
    }
}
```

```
if(!pendudukLama.getId_keluarga().equals(penduduk.getId_keluarga())) {
    KeluargaModel keluargaBaru = keluargaDAO.selectKeluargaById(penduduk.getId_keluarga());
    KelurahanModel kelurahanBaru = kelurahanDAO.selectKelurahan(keluargaBaru.getId_kelurahan());
    KecamatanModel kecamatanBaru = kecamatanDAO.selectKecamatan(kelurahanBaru.getId_kecamatan());

    String kodeLokasiBaru = kecamatanBaru.getKode_kecamatan().substring(0,6);
    nikBaru = kodeLokasiBaru + nikBaru.substring(6);
}else {

}

if(!nikBaru.equals(pendudukLama.getNik())) {
    String nikDepan = nikBaru.substring(0, 12) + "%";
    List<PendudukModel> listNikSama = pendudukDAO.listNikSama(nikDepan + "%");

    if(listNikSama.isEmpty()) {
        nikBaru = nikBaru.substring(0, 12) + "0001";
    }else {
        PendudukModel temp = listNikSama.get(listNikSama.size()-1);
        String nikTambah = temp.getNik().substring(12);
        int temp2 = Integer.parseInt(nikTambah) + 1;
        String nol = "";
        String temp3 = "" + temp2;
        int counter = 4-temp3.length();
        for(int i = 0; i < counter; i++) {
            nol += "0";
        }
        nikBaru = nikBaru.substring(0, 12) + nol + temp3;
    }
}else {
}

penduduk.setNik(nikBaru);
penduduk.setId(pendudukLama.getId());

model.addAttribute("nikLama", pendudukLama.getNik());
pendudukDAO.updatePenduduk(penduduk);
return "penduduk-update-success";
```

Form yang terdapat di halaman penduduk-update memiliki method POST dengan action “penduduk/ubah/{NIK}”. Setelah semua field diisi, maka halaman tersebut akan mengirimkan semua data ke controller. Controller kemudian akan menginisialisasi pendudukLama. Kemudian akan memanggil service, service database dan mapper.

Setelah itu, maka akan dicek, apakah tanggal lahir pendudukLama sama dengan tanggal lahir penduduk yang baru, apabila tanggal lahirnya berbeda, maka tanggal lahir yang baru akan diubah sesuai dengan input, kemudian cek jenis kelaminnya, apabila perempuan maka ditambah 40. Apabila tanggal lahir tidak di update, maka cek jenis kelamin, apabila jenis kelamin diubah dari perempuan menjadi laki-laki maka tanggal lahir akan dikurang 40.

Kemudian cek apakah pendudukBaru memiliki id keluarga yang berbeda atau tidak, jika berbeda, maka akan menginisialisasi keluargaBaru, kelurahanBaru dan kecamatanBaru. Kemudian, mengambil 6 digit dari kode kecamatan untuk menjadi 6 digit pertama di NIK.

Terakhir, cek apakah ada penduduk lain yang memiliki 12 digit NIK terdepan yang sama dengan NIK penduduk baru. Jika iya, maka ambil list nya lalu buat urutannya seperti pada fitur addPenduduk. Lalu, set NIK dan Id penduduk baru dan return halaman penduduk-update-success.

6. Mengubah Data Keluarga

Initial Page: /keluarga/ubah/{NKK}

Form Request: POST, /keluarga/ubah/{NKK}

Response Page: /keluarga/ubah/{NKK}

Pada saat mengakses /keluarga/ubah/{NKK}, akan langsung muncul data dari keluarga tersebut pada field yang ada. Dapat dilihat seperti gambar di bawah ini.

Ubah Keluarga

Alamat
Jln. Cikapayang No. 594

RT
149

RW
090

Kota / Kecamatan / Kelurahan
KABUPATEN KEPULAUAN SERIBU / KEPULAUAN SERIBU SELATAN / PULAU TIDUNG

Simpan

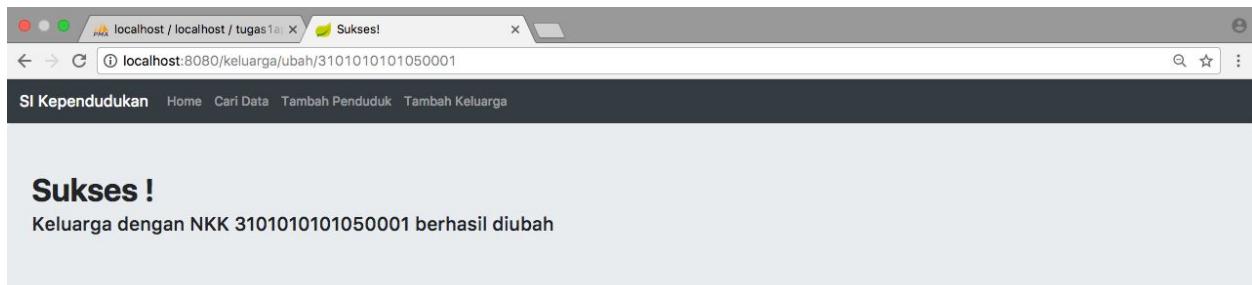
Misal, ubah RW dan Kota/Kecamatan/Kelurahan

RW
18

Kota / Kecamatan / Kelurahan
KOTA JAKARTA SELATAN / PASAR MINGGU / CILANDAK TIMUR

Simpan

Klik Simpan



```
@RequestMapping("/keluarga/ubah/{nomor_kk}")
public String ubah (Model model, @PathVariable(value="nomor_kk") String nomor_kk) {
    KeluargaModel keluarga = keluargaDAO.selectKeluarga(nomor_kk);

    if(keluarga == null) {
        model.addAttribute("errorMessage", "Keluarga dengan NKK " + nomor_kk + " tidak ditemukan, "
            + "mohon cek kembali Nomor Kartu Keluarga Anda.");
        return "error/error404";
    }
    model.addAttribute("keluarga", keluarga);
    List<KelurahanModel> kelurahan = kelurahanDAO.selectAllKelurahan();
    List<KecamatanModel> kecamatan = kecamatanDAO.selectAllKecamatan();
    List<KotaModel> kota = kotaDAO.selectAllKota();
    for(int i = 0; i < kelurahan.size(); i++){
        for(int j = 0; j < kecamatan.size(); j++){
            for(int k = 0 ; k < kota.size(); k++){
                if(kelurahan.get(i).getId_kecamatan().equals(kecamatan.get(j).getId())){
                    if(kecamatan.get(j).getId_kota().equals(kota.get(k).getId())){
                        String namaKelurahan = kota.get(k).getNama_kota()+
                            " / " + kecamatan.get(j).getNama_kecamatan() +
                            " / " + kelurahan.get(i).getNama_kelurahan();
                        kelurahan.get(i).setNama_kelurahan(namaKelurahan);
                    }
                }
            }
        }
    }
    model.addAttribute("kelurahan", kelurahan);
    return "keluarga-update";
}
```

Pada saat membuka halaman “/keluarga/ubah” , KeluargaController akan menginisialisasi keluarga model terlebih dahulu. Setelah itu, controller akan memeriksa, apakah NKK nya ada atau tidak di database, jika tidak, maka akan menampilkan error, jika ada, maka akan ditampilkan halaman keluarga-update.

```
@RequestMapping(value = "/keluarga/ubah/{nomor_kk}", method = RequestMethod.POST)
public String ubahKeluargaSubmit(Model model, @ModelAttribute KeluargaModel keluarga) {
    KeluargaModel keluargaLama = keluargaDAO.selectKeluarga(keluarga.getNomor_kk());
    KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
    String nkkBaru = keluargaLama.getNomor_kk();
    System.out.println(keluargaLama.toString());

    //cek rt
    if(!keluarga.getRT().equals(keluargaLama.getRT())) {
        String rt = keluarga.getRT();
        String temp = "";
        for(int i = 0; i < 3-rt.length(); i++) {
            temp += "0";
        }
        rt = temp + rt;
        keluarga.setRT(rt);
    }

    //cek rw
    if(!keluarga.getRW().equals(keluargaLama.getRW())) {
        String rw = keluarga.getRW();
        String temp2 = "";
        for(int i = 0; i < 3-rw.length(); i++) {
            temp2 += "0";
        }
        rw = temp2 + rw;
        keluarga.setRW(rw);
    }
}
```

Form yang terdapat di halaman keluarga-update memiliki method POST dengan action “keluarga/ubah/{NKK}”. Setelah semua field diisi, maka halaman tersebut akan mengirimkan semua data ke controller. Controller kemudian akan menginisialisasi keluargaLama dan kelurahan. Kemudian akan memanggil service, service database dan mapper.

Pertama, cek RT, apabila RT berubah, maka lakukan looping agar RT tetap terdiri dari 3 digit (menambahkan nol). Hal ini juga berlaku pada RW.

```
nkkBaru = kelurahan.getKode_kelurahan().substring(0,6);
String[] tanggal_isi = java.time.LocalDate.now().toString().split("-");
nkkBaru += tanggal_isi[2] + tanggal_isi[1] + tanggal_isi[0].substring(2);

List<KeluargaModel> listNkkSama = keluargaDAO.listNkkSama(nkkBaru + "%");
if(listNkkSama.isEmpty()) {
    nkkBaru += "0001";
} else {
    KeluargaModel temp = listNkkSama.get(listNkkSama.size()-1);
    String nkkTambah = temp.getNomor_kk().substring(12);
    int temp2 = Integer.parseInt(nkkTambah);
    temp2++;

    String temp3 = "" + temp2;
    for(int i = 0; i < 4-temp3.length(); i++) {
        nkkBaru += "0";
    }
    nkkBaru += temp3;
}
keluarga.setNomor_kk(nkkBaru);
keluarga.setId(keluargaLama.getId());
keluarga.setIs_tidak_berlaku("0");
```

Kemudian ambil kode kelurahan dan tanggal sekarang untuk tanggal pengisian. Lalu, ambil semua keluarga yang memiliki 12 digit pertama NKK yang sama dengan NKK keluarga baru. Jika tidak ada yang sama, maka nomor urut NKK baru adalah 0001, jika ada, maka nomor urut didapat dengan cara yang sama dengan fitur addKeluarga. Setelah itu, set NKK, id dan berlaku untuk keluarga baru.

```
List<PendudukModel> anggotaKeluarga = pendudukDAO.selectAnggotaKeluarga(keluarga.getId());
if(!anggotaKeluarga.isEmpty()) {
    for(int i = 0; i < anggotaKeluarga.size(); i++) {
        PendudukModel penduduk = anggotaKeluarga.get(i);

        String nikTengah = penduduk.getNik().substring(6, 12);
        String nikBaru = "";
        String nikDepan = kelurahan.getKode_kelurahan().substring(0, 6);
        nikBaru = "" + nikDepan + nikTengah;

        List<PendudukModel> listNikSama = pendudukDAO.listNikSama(nikBaru + "%");
        if(listNikSama.isEmpty()) {
            nikBaru = nikBaru.substring(0, 12) + "0001";
        }else {
            PendudukModel temp = listNikSama.get(listNikSama.size()-1);
            String nikTambah = temp.getNik().substring(12);
            int temp2 = Integer.parseInt(nikTambah) + 1;
            String nol = "";
            String temp3 = "" + temp2;
            int counter = 4-temp3.length();
            for(int j = 0; j < counter; j++) {
                nol += "0";
            }
            nikBaru = nikBaru.substring(0, 12) + nol + temp3;
        }
        penduduk.setNik(nikBaru);
        pendudukDAO.updatePenduduk(penduduk);
    }
}
model.addAttribute("nkkLama", keluargaLama.getNomor_kk());
keluargaDAO.updateKeluarga(keluarga);
return "keluarga-update-success";
```

Apabila keluarga mengalami perubahan domisili, maka NIK dari anggota keluarga juga akan berubah. Maka, ambil list dari anggota keluarga, jika listnya tidak kosong, maka ambil iterasi penduduk yang ada dalam list tersebut, dan ganti 6 digit NIK pertama supaya sama dengan 6 digit pertama NKK. Setelah itu, hitung juga nomor urut untuk masing-masing penduduk (sama seperti fitur addPenduduk). Lalu set NIK penduduk menjadi penduduk yang baru. Setelah itu panggil keluarga service untuk menjalankan method updateKeluarga yang akan memanggil keluargaServiceDatabase dan keluargaMapper. Kemudian return halaman keluarga-update-success.

7. Mengubah Status Kematian Penduduk

Initial Page: /penduduk?nik={NIK}

Form Request: POST, /penduduk/mati

Response Page: /penduduk?nik={NIK}

Pada halaman penduduk?nik={NIK} terdapat tombol ‘Nonaktifkan’

Lihat data penduduk - 3101011405170001

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

Nonaktifkan

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Klik tombol ‘Nonaktifkan’

Sukses !

Penduduk dengan NIK 3101011405170001 sudah tidak aktif

Harap tunggu

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Muncul halaman seperti di atas, kemudian akan *redirect* ke halaman seperti di bawah ini. Dapat dilihat bahwa Status Kematian berubah dari hidup menjadi wafat.

The screenshot shows a web browser window titled "Detail Penduduk". The address bar displays "localhost:8080/penduduk?nik=3101011405170001". The page header includes "SI Kependudukan" and navigation links for "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". The main content area is titled "Lihat data penduduk - 3101011405170001". It lists the following data in a table:

NIK	:	3101011405170001
Nama	:	Heru Haryanto
Tempat/Tanggal Lahir	:	Jakarta, 2017-05-14
Jenis Kelamin	:	Laki-laki
Alamat	:	Ds. Adisumarmo No. 43
RT/RW	:	079/025
Kelurahan/Desa	:	PULAU TIDUNG
Kecamatan	:	KEPULAUAN SERIBU SELATAN
Kota	:	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	:	O-
Agama	:	Islam
Status Perkawinan	:	Belum Kawin
Pekerjaan	:	BELUM/TIDAK BEKERJA
Kewarganegaraan	:	WNI
Status Kematian	:	Wafat

A blue button labeled "Nonaktifkan" is located at the bottom right of the data table. At the very bottom of the page, a small footer reads "Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza."

```
@PostMapping("/penduduk/mati")
public String nonAktifPenduduk(@RequestParam(value="nik", required = true) String nik, Model model){
    if(nik == null) {
        model.addAttribute("errorMessage", "Masukkan NIK");
        return "error/error404";
    }

    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);

    if(penduduk == null) {
        model.addAttribute("errorMessage", "Nik tidak ditemukan");
        return "error/error404";
    }

    pendudukDAO.nonAktifkanPenduduk(nik);

    String id_keluarga = penduduk.getId_keluarga();
    KeluargaModel keluarga = keluargaDAO.selectKeluargaById(id_keluarga);
    List<PendudukModel> anggotaKeluarga = pendudukDAO.selectAnggotaKeluarga(id_keluarga);

    if(!anggotaKeluarga.isEmpty()) {
        int counter = 0;

        //hitung anggota keluarga yang sudah wafat
        for (int i = 0; i < anggotaKeluarga.size(); i++) {
            if(anggotaKeluarga.get(i).getIs_wafat().equals("1")) {
                counter++;
            }
        }

        //jika semua anggota keluarga sudah wafat, maka tidak berlaku
        if(anggotaKeluarga.size() == counter) {
            keluarga.setIs_tidak_berlaku("1");
        }
    }
    keluargaDAO.updateKeluarga(keluarga);
    model.addAttribute("nik", nik);
    return "penduduk-nonaktif-success";
}
```

Pertama cek apakah NIK null atau tidak, jika iya, maka tampilkan error. Lalu inisiasi penduduk dengan cara memanggil pendudukService untuk menjalankan method selectPenduduk. Setelah itu apabila penduduk tidak ditemukan, maka tampilkan pesan error, jika penduduk ditemukan, maka jalankan method nonAktifkanPenduduk(nik) yaitu memanggil pendudukService, pendudukServiceDatabase dan pendudukMapper.

Setelah itu, ambil id keluarga, kemudian inisiasi keluarga tersebut dengan cara mengambil keluarga dari penduduk tersebut. Lalu ambil list anggota keluarganya, apabila anggota keluarganya tidak kosong, maka hitung berapa banyak anggota keluarga yang sudah wafat di keluarga tersebut. Lalu hitung juga berapa banyak anggota keluarga yang dimiliki keluarga tersebut. Jika jumlah anggota keluarga yang sudah wafat sama dengan jumlah semua

anggota keluarga, maka set keluarga menjadi tidak berlaku. Kemudian jalankan method update keluarga dengan memanggil keluargaService, keluargaServiceDatabase dan keluargaMapper. Setelah itu, tampilkan halaman penduduk-nonaktif-success.

8. Tampilkan Data Penduduk Berdasarkan Kota/Kabupaten, Kecamatan dan Kelurahan Tertentu

Initial Page: /penduduk/cari

Request Method:

GET,/penduduk/cari?kt={ID_KOTA}&kc={ID_KECAMATAN}&kl={ID_KELURAHAN}

Response Page:

/penduduk/cari?kt={ID_KOTA}&kc={ID_KECAMATAN}&kl={ID_KELURAHAN}

Pertama buka halaman /penduduk/cari, kemudian pilih kota yang diinginkan, lalu klik lihat.

Lihat Data Penduduk

Kota
KABUPATEN KEPULAUAN SERIBU

Kecamatan

Kelurahan

Lihat

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Akan muncul seperti di bawah ini, option dari kecamatan adalah kecamatan yang ada di kota yang sudah dipilih. Klik lihat

Lihat Data Penduduk

Kota
KABUPATEN KEPULAUAN SERIBU

Kecamatan
KEPULAUAN SERIBU UTARA

Kelurahan

Lihat

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Setelah itu muncul seperti di bawah, dimana option dari kelurahan yang muncul adalah kelurahan yang berada di kecamatan yang sudah dipilih. Klik lihat

Lihat Data Penduduk

Kota
KABUPATEN KEPULAUAN SERIBU

Kecamatan
KEPULAUAN SERIBU UTARA

Kelurahan
PULAU KELAPA

Lihat

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Kemudian akan muncul daftar nama penduduk yang ada di kota, kecamatan dan kelurahan yang sudah dipilih.

No	NIK	Nama Lengkap	Jenis Kelamin
1	3101026710980002	Maya Riyanti	Perempuan
2	3101025002120001	Padmi Sadina Mardhiyah	Perempuan
3	3101024901170001	Uli Wijayanti	Perempuan
4	3101026711930001	Latika Melani S.T.	Perempuan
5	3101026711980001	Amalia Clara Mandasari S.E.I	Perempuan
6	3101025209130001	Victoria Sudati	Perempuan
7	3101026104170001	Amalia Widiaستuti	Perempuan
8	3101026801940001	Raisa Lestari M.Pd	Perempuan
9	3101026802670001	Puspa Rahimah	Perempuan
10	3101026802680001	Elisa Rahmi Puspasari	Perempuan

Showing 1 to 10 of 1,133 entries

Previous 1 2 3 4 5 ... 114 Next

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

```

@GetMapping("/penduduk/cari")
public String cariPenduduk(@RequestParam(value="kt", required=false) String kt,
                           @RequestParam(value="kc", required=false) String kc,
                           @RequestParam(value="kl", required=false) String kl,
                           Model model) {

    Boolean flag = false;

    //jika kelurahan tidak kosong
    if(kl != null) {
        model.addAttribute("kt_nama", kotaDAO.selectKota(kt).getNama_kota());
        model.addAttribute("kc_nama", kecamatanDAO.selectKecamatan(kc).getNama_kecamatan());
        model.addAttribute("kl_nama", kelurahanDAO.selectKelurahan(kl).getNama_kelurahan());
        model.addAttribute("flag_kelurahan", true);
        model.addAttribute("kl", kl);

        flag = true;
        List<PendudukModel> listPenduduk = pendudukDAO.selectPendudukByKelurahan(kl);
        model.addAttribute("listPenduduk", listPenduduk);
        return "cari-penduduk-hasil";
    }

    //buat list kota
    List<KotaModel> listKota = kotaDAO.selectAllKota();
    model.addAttribute("listKota", listKota);
}

```

```
//jika kota tidak kosong
if(kt != null) {
    model.addAttribute("kt_nama", kotaDAO.selectKota(kt).getNama_kota());
    model.addAttribute("flag_kota", true);
    model.addAttribute("kt", kt);

    if(kc == null) {
        List<KecamatanModel> listKecamatan = kecamatanDAO.selectKecamatanByIdKota(kt);
        model.addAttribute("listKecamatan", listKecamatan);
    }

    if(kc != null) {
        model.addAttribute("kc_nama", kecamatanDAO.selectKecamatan(kc));
        model.addAttribute("flag_kecamatan", true);
        model.addAttribute("kc", kc);
        if(kl == null) {
            List<KelurahanModel> listKelurahan = kelurahanDAO.selectAllKelurahan(kc);
            model.addAttribute("listKelurahan", listKelurahan);
        }
    }
}

return "form-cari-penduduk";
```

Pertama, inisiasi boolean flag. Kemudian jika kelurahan belum dipilih, maka inisiasi kota, kecamatan, dan kelurahan menggunakan service, serviceDatabase dan mapper dari kota, kecamatan dan kelurahan. Setelah itu, cari list penduduk yang tinggal di kelurahan yang sudah dipilih menggunakan method selectPendudukByIdKelurahan. Setelah list penduduk sudah didapatkan, return cari-penduduk-hasil.

Kemudian, ambil semua list kota yang ada dengan menggunakan method selectAllKota pada kota service.

Apabila kecamatan yang diinput di form masih kosong, maka ambil kota yang ada di form lalu ambil semua kecamatan berdasarkan kota yang sudah diinput.

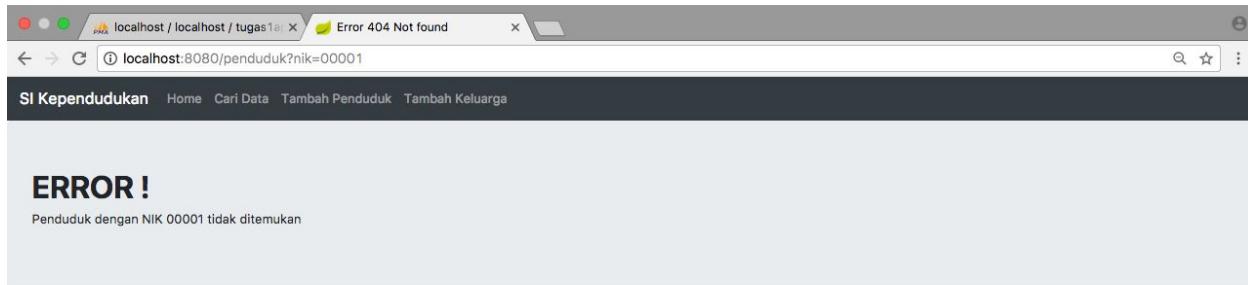
Apabila kecamatan sudah terisi, cek apakah kelurahan masih kosong atau tidak. Jika ya, maka teruskan kecamatan yang sudah dipilih tadi. Lalu cek lagi apakah kelurahan masih kosong atau tidak. Jika ya, cari semua list kelurahan dengan memanggil method select all kelurahan.

Setelah itu, return ke form-cari-penduduk.

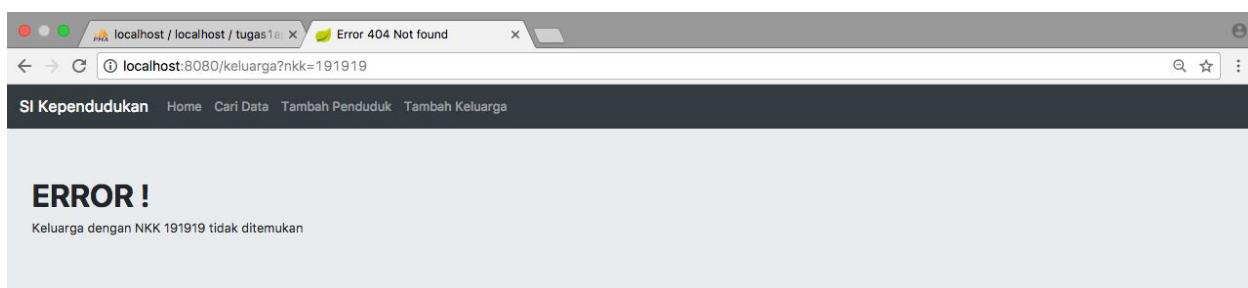
9. Menambahkan Error Page (Bonus)

Halaman error ini dibuat pada folder templates di dalam folder error dengan nama error404.html. Sehingga bisa langsung menampilkan pesan error yang ada di controller menggunakan \${errorMessage}

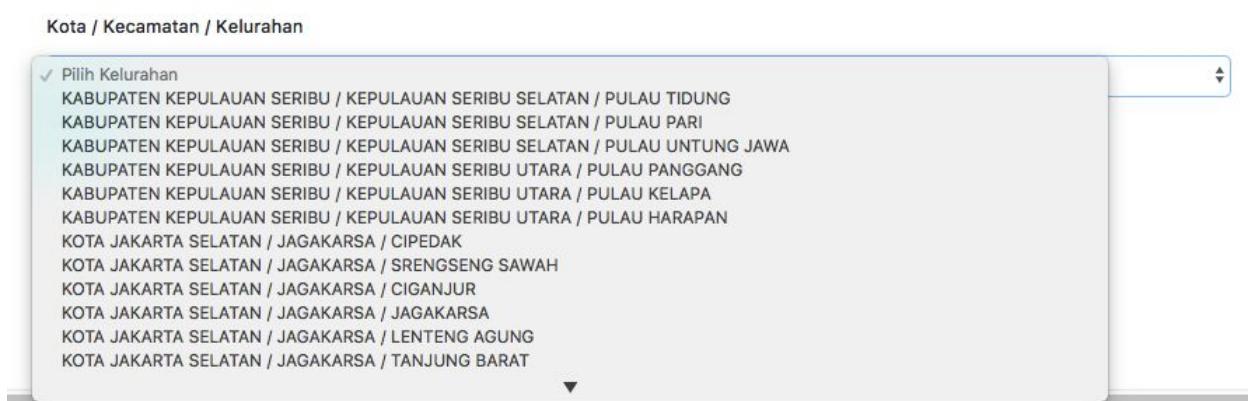
- NIK tidak ditemukan pada fitur 1 (Tampilkan Data Penduduk)



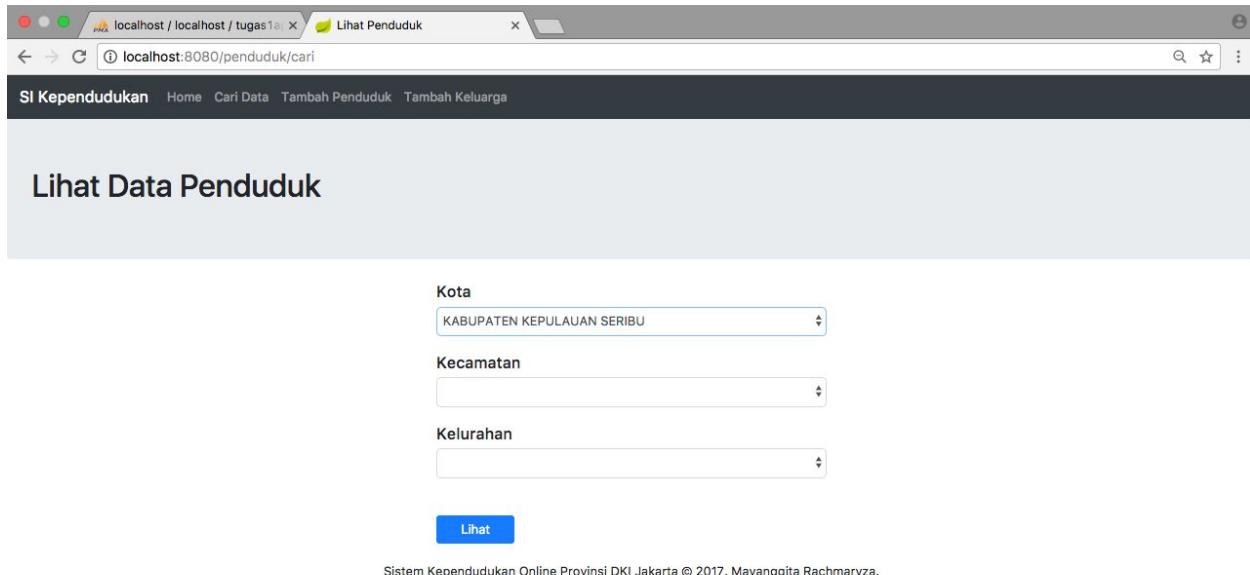
- NKK tidak ditemukan pada fitur 2 (Tampilkan Data Keluarga Beserta Daftar Anggotanya)



- Meng-handle kombinasi Kota, Kecamatan dan Kelurahan pada fitur 4



- Meng-handle kombinasi Kota, Kecamatan dan Provinsi pada fitur 8 dengan hanya menampilkan kecamatan yang sesuai dengan kota yang dipilih dan hanya menampilkan kelurahan yang sesuai dengan kecamatan yang dipilih



The screenshot shows a web browser window with the title "localhost / localhost / tugas1b Lihat Penduduk". The address bar displays "localhost:8080/penduduk/cari". The page header includes "SI Kependudukan" and navigation links for "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". The main content area is titled "Lihat Data Penduduk". It features three dropdown menus: "Kota" (selected value: "KABUPATEN KEPULAUAN SERIBU"), "Kecamatan" (empty), and "Kelurahan" (empty). A blue "Lihat" button is located below the dropdowns. At the bottom of the page, a footer note reads "Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza."

Lihat Data Penduduk

Kota
KABUPATEN KEPULAUAN SERIBU

Kecamatan
KEPULAUAN SERIBU UTARA

Kelurahan

Lihat

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

Lihat Data Penduduk

Kota
KABUPATEN KEPULAUAN SERIBU

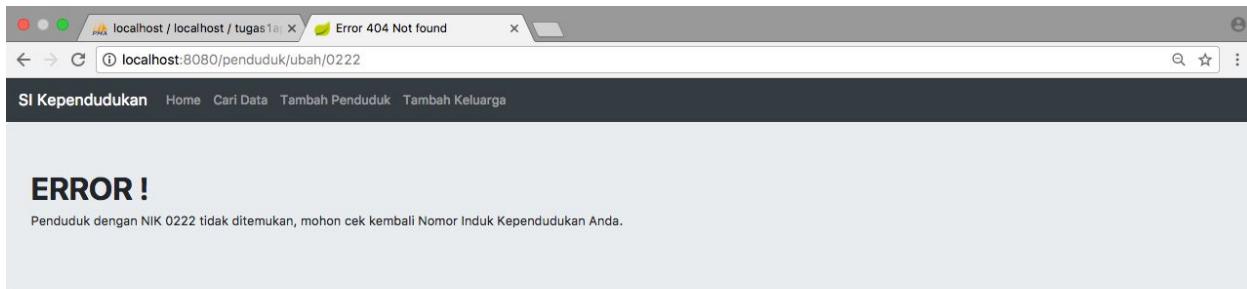
Kecamatan
KEPULAUAN SERIBU UTARA

Kelurahan
PULAU KELAPA

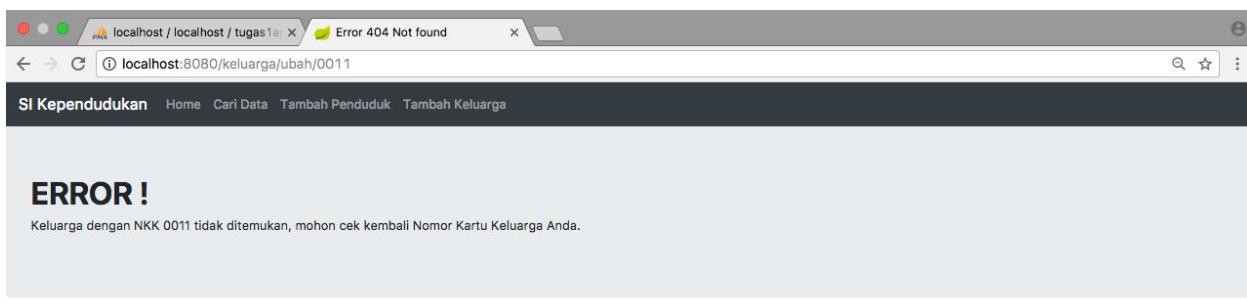
Lihat

Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza.

- NIK tidak ditemukan pada fitur 5 (Mengubah Data Penduduk)



- NKK tidak ditemukan pada fitur 6 (Mengubah Data Keluarga)



10. Menampilkan Penduduk Paling Muda dan Paling Tua di Suatu Kelurahan

The screenshot shows a browser window with the title "localhost / localhost / tugas18 Lihat Penduduk". The URL in the address bar is "localhost:8080/penduduk/cari?kt=2&kc=5&kl=21". The page content includes a navigation bar with "SI Kependudukan" and links to "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". Below this is a heading "Lihat Data Penduduk di Kota KOTA JAKARTA SELATAN, Kecamatan CILANDAK, Kelurahan PONDOK LABU".

Penduduk Dengan Usia Termuda		Penduduk Dengan Usia Tertua	
NIK	3171035909170001	NIK	3171036909520001
Nama	Vicky Kusmawati	Nama	Dalima Hastuti
Tanggal Lahir	2017-09-19	Tanggal Lahir	1952-09-29

[Lihat Penduduk](#) [Lihat Penduduk](#)

Show 10 entries Search:

No	NIK	Nama Lengkap	Jenis Kelamin
1	3171035003130001	Melinda Purwanti	Perempuan
2	3171037106810001	Dina Nasiyidah	Perempuan
3	3171035707930001	Diah Rika Wahyuni	Perempuan
4	3171034402170003	Azalea Betania Kusmawati	Perempuan
5	3171035706170002	Ani Padmasari	Perempuan
6	3171035708960001	Malika Safina Mulyani M.Pd	Perempuan

```
List<PendudukModel> listPenduduk = pendudukDAO.selectPendudukByKelurahan(kl);

PendudukModel termuda = listPenduduk.get(0);
PendudukModel tertua = listPenduduk.get(0);

for(int i = 0; i < listPenduduk.size(); i++) {
    if(termuda.getTanggal_lahir().compareTo(listPenduduk.get(i).getTanggal_lahir()) < 0) {
        termuda = listPenduduk.get(i);
    }
    if(tertua.getTanggal_lahir().compareTo(listPenduduk.get(i).getTanggal_lahir()) > 0) {
        tertua = listPenduduk.get(i);
    }
}
model.addAttribute("listPenduduk", listPenduduk);
model.addAttribute("tertua", tertua);
model.addAttribute("termuda", termuda);
return "cari-penduduk-hasil";
```

Untuk mencari penduduk tertua dan termuda bisa dilakukan dengan cara melakukan iterasi pada penduduk di listPenduduk yang sudah didapatkan dari fitur 8. Lalu membandingkan tanggal lahir penduduk yang ada dalam list.

11. Menambahkan Button Lihat Penduduk pada Penduduk dengan Usia Termuda dan Penduduk Dengan Usia Tertua

Lihat Data Penduduk di Kota KOTA JAKARTA SELATAN, Kecamatan CILANDAK, Kelurahan PONDOK LABU

No	NIK	Nama Lengkap	Jenis Kelamin
1	3171035003130001	Melinda Purwanti	Perempuan
2	3171037105810001	Dina Nasiyidah	Perempuan
3	3171035707930001	Diah Rika Wahyuni	Perempuan

Jika button “Lihat Penduduk” ditekan, maka akan redirect ke halaman penduduk?nik={NIK}, misalnya:

- Untuk penduduk termuda

The screenshot shows a web browser window with the title "localhost / localhost / tugas1e Detail Penduduk". The URL in the address bar is "localhost:8080/penduduk?nik=3171035909170001". The page header includes "SI Kependudukan" and navigation links for "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". The main content is titled "Lihat data penduduk - 3171035909170001". Below this, there is a table of居民 information:

NIK	:	3171035909170001
Nama	:	Vicky Kusmawati
Tempat/Tanggal Lahir	:	Jakarta, 2017-09-19
Jenis Kelamin	:	Perempuan
Alamat	:	Jr. Veteran No. 479
RT/RW	:	181/006
Kelurahan/Desa	:	PONDOK LABU
Kecamatan	:	CILANDAK
Kota	:	KOTA JAKARTA SELATAN
Golongan Darah	:	A-
Agama	:	Islam
Status Perkawinan	:	Belum Kawin
Pekerjaan	:	BELUM/TIDAK BEKERJA
Kewarganegaraan	:	WNI
Status Kematian	:	Hidup

At the bottom right of the table is a blue button labeled "Nonaktifkan". Below the table, a small note reads "Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza."

- Untuk penduduk tertua

The screenshot shows a web browser window with the title "localhost / localhost / tugas1e Detail Penduduk". The URL in the address bar is "localhost:8080/penduduk?nik=3171036909520001". The page header includes "SI Kependudukan" and navigation links for "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". The main content is titled "Lihat data penduduk - 3171036909520001". Below this, there is a table of居民 information:

NIK	:	3171036909520001
Nama	:	Dalima Hastuti
Tempat/Tanggal Lahir	:	Serang, 1952-09-29
Jenis Kelamin	:	Perempuan
Alamat	:	Dk. Bagas Pati No. 205
RT/RW	:	085/096
Kelurahan/Desa	:	PONDOK LABU
Kecamatan	:	CILANDAK
Kota	:	KOTA JAKARTA SELATAN
Golongan Darah	:	O-
Agama	:	Kristen
Status Perkawinan	:	Cerai Hidup
Pekerjaan	:	PERANGKAT DESA
Kewarganegaraan	:	WNI
Status Kematian	:	Hidup

At the bottom right of the table is a blue button labeled "Nonaktifkan". Below the table, a small note reads "Sistem Kependudukan Online Provinsi DKI Jakarta © 2017. Mayanggita Rachmaryza."

C. Optimasi Database

Untuk mengoptimasi *database*, dilakukan:

a. Penambahan *Primary Key*

Penambahan *primary key* dilakukan karena belum terdapat *primary key* pada *database*. Penambahkan *primary key* dilakukan pada setiap tabel, yaitu:

- Pada tabel keluarga, *primary key* nya adalah Id
- Pada tabel penduduk, *primary key* nya adalah Id
- Pada tabel kecamatan, *primary key* nya adalah Id
- Pada tabel kelurahan, *primary key* nya adalah Id
- Pada tabel kota, *primary key* nya adalah Id

b. Penambahan *Auto Increment*

Penambahan *auto increment* ini dilakukan untuk menambahkan nilai secara otomatis setiap ada penambahan relasi pada sebuah data. Sehingga tidak perlu untuk melakukan penambahan *primary key* secara manual setiap ada penambahan data pada relasi. *Auto increment* ini juga membuat *primary key* selalu terjaga. Penambahan *auto increment* dilakukan pada:

- Atribut id pada tabel keluarga
- Atribut id pada tabel penduduk

c. Penambahan *Index*

Penambahan *index* dilakukan untuk mempercepat pencarian dalam suatu tabel/relasi. Penambahan *index* dilakukan pada:

- Pada tabel penduduk, penambahan *index* dilakukan pada atribut nik, id_keluarga dan tanggal_lahir
- Pada tabel keluarga, penambahan *index* dilakukan pada atribut nomor_kk dan id_kelurahan
- Pada tabel kecamatan, penambahan *index* dilakukan pada atribut id_kota dan kode_kecamatan

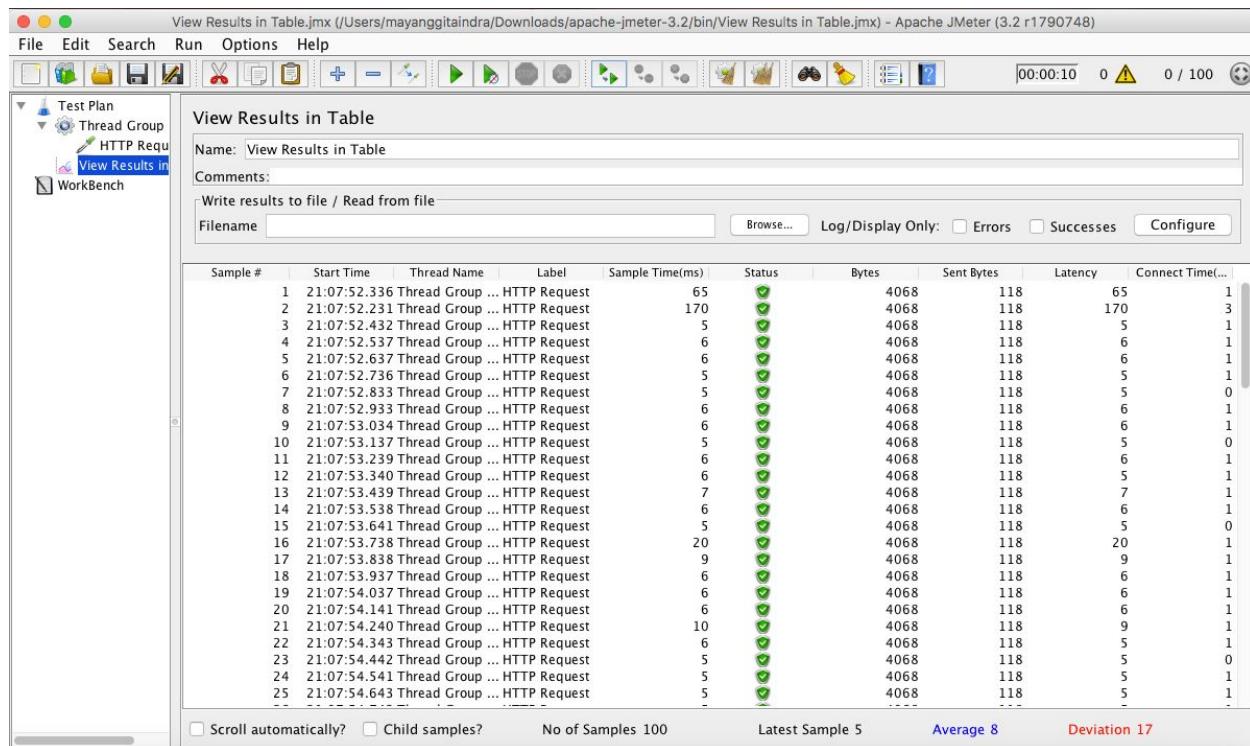
- Pada tabel kelurahan, penambahan *index* dilakukan pada atribut `id_kecamatan` dan `kode_kelurahan`
- Pada tabel kota, penambahan *index* dilakukan pada atribut `kota`

D. Stress Testing

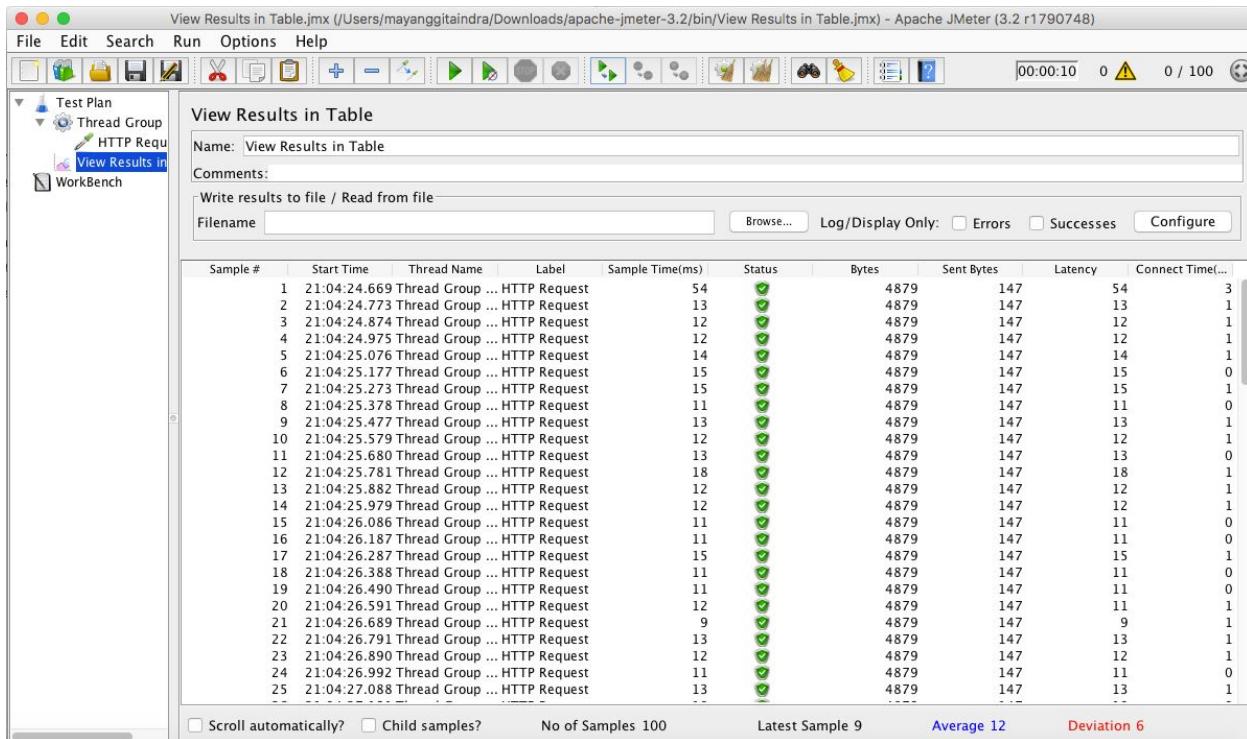
Ramp up: 10 s

Thread: 100

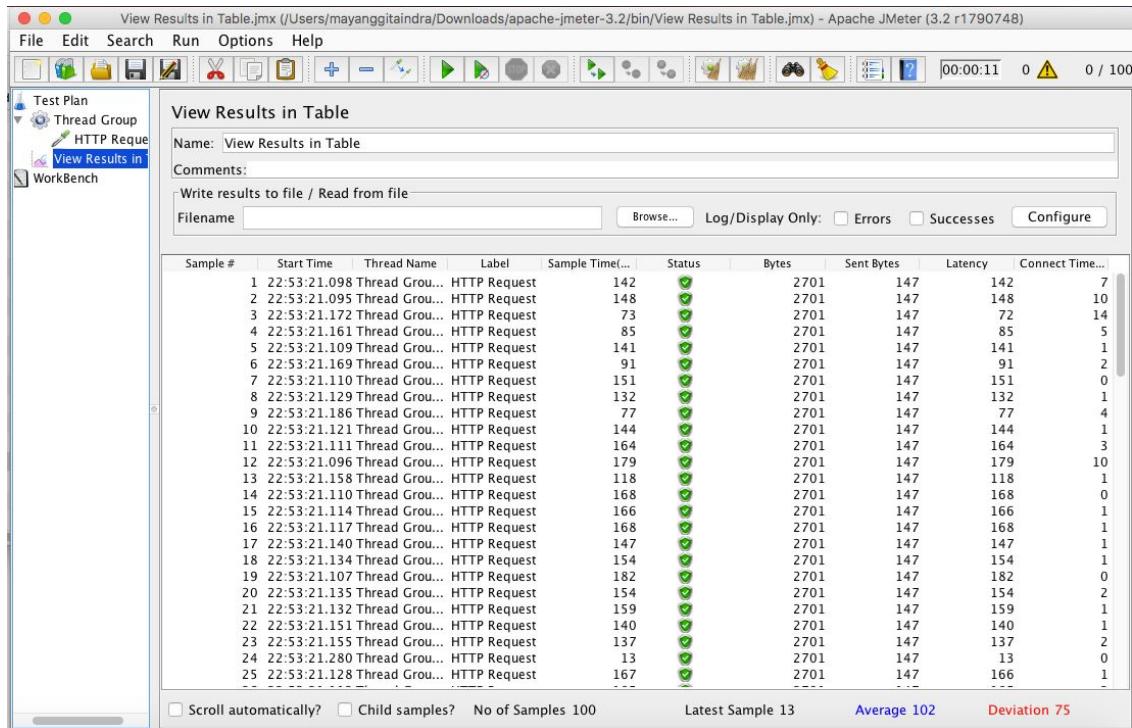
1. Membuka halaman index (<http://localhost:8080/>)



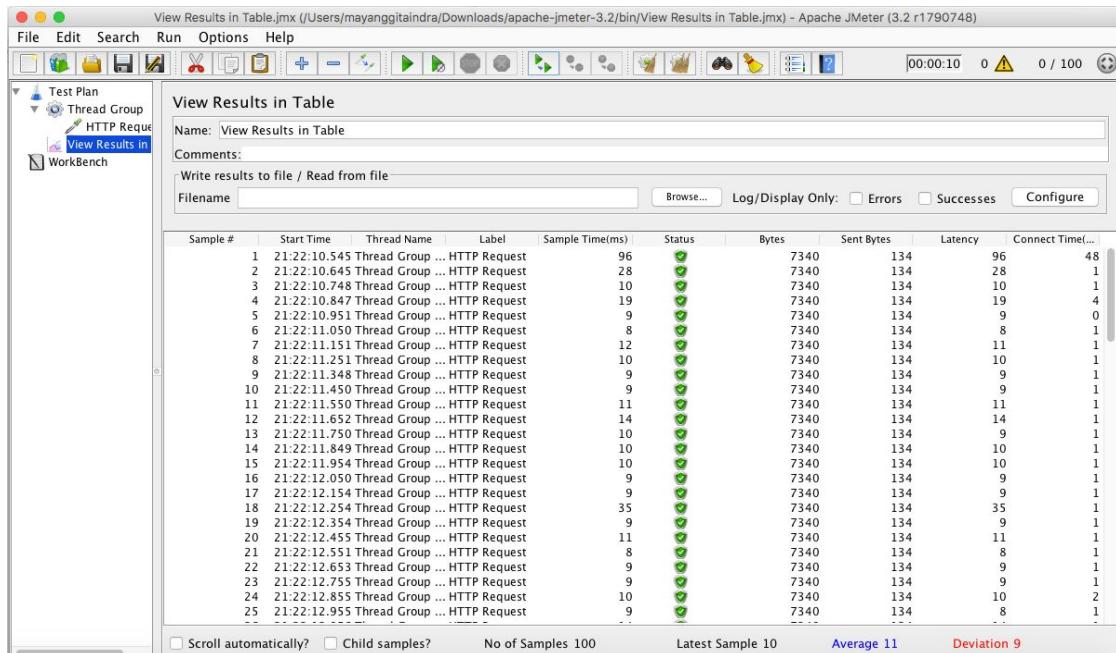
2. Membuka halaman lihat detail penduduk
(<http://localhost:8080/penduduk?nik=3101011405170001>)



3. Membuka halaman lihat detail keluarga
 (http://localhost:8080/keluarga?nkk=3171022210170001)



4. Membuka halaman tambah penduduk (http://localhost:8080/penduduk/tambah/)



5. Men-submit halaman tambah penduduk

HTTP Request

Method: POST Path: http://localhost:8080/penduduk/tambah Content encoding:

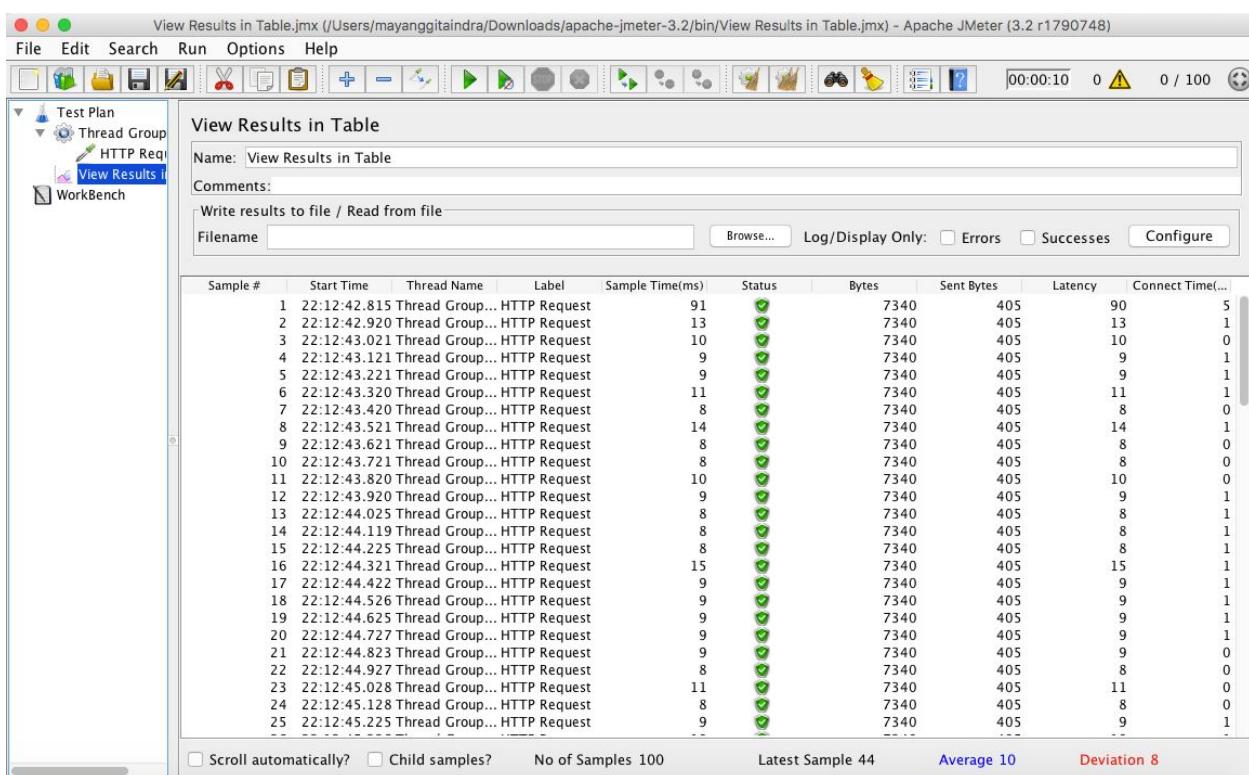
Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data for POST Browser-compatible headers

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

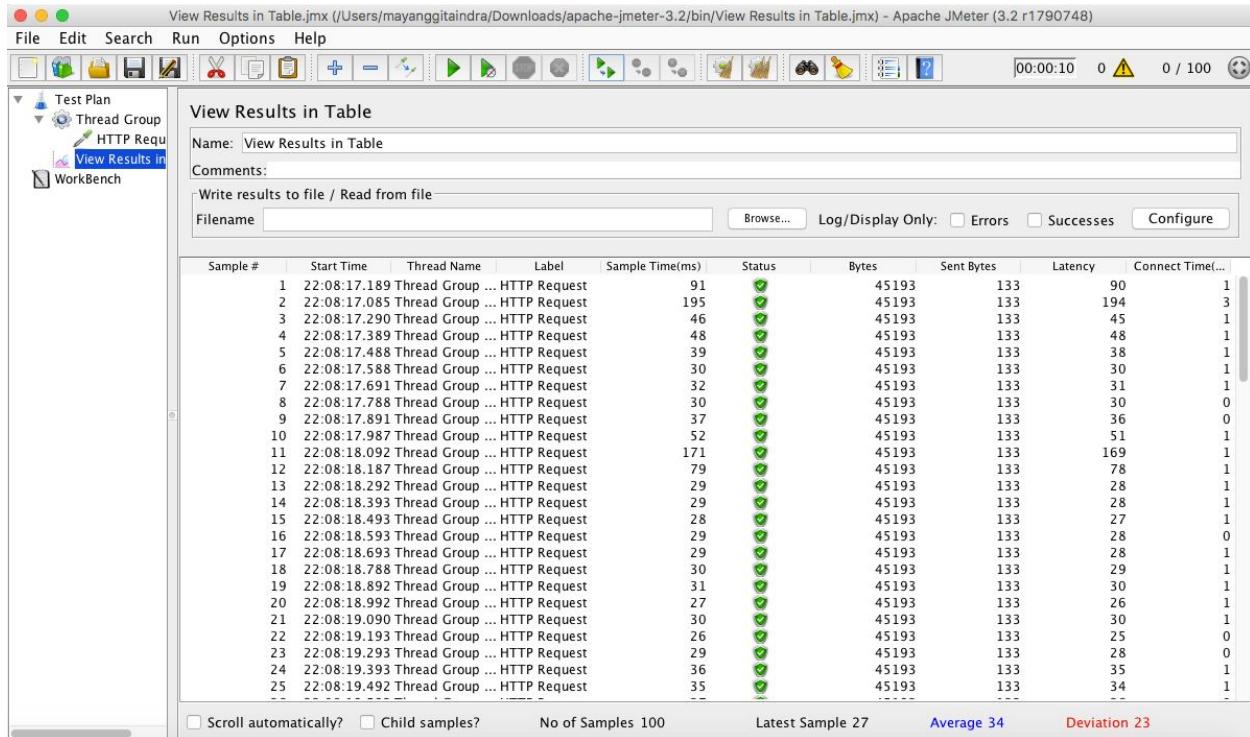
Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
nama	apo	<input type="checkbox"/>	<input checked="" type="checkbox"/>
tempat_lahir	jakarta	<input type="checkbox"/>	<input checked="" type="checkbox"/>
tanggal_lahir	1987-08-08	<input type="checkbox"/>	<input checked="" type="checkbox"/>
jenis_kelamin	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
golongan_darah	O-	<input type="checkbox"/>	<input checked="" type="checkbox"/>
agama	islam	<input type="checkbox"/>	<input checked="" type="checkbox"/>
status_perkawinan	kawin	<input type="checkbox"/>	<input checked="" type="checkbox"/>
pekerjaan	guru	<input type="checkbox"/>	<input checked="" type="checkbox"/>
is_wni	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
is_wafat	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
id_keluarga	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>
status_dalam_keluarga	anak	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Detail **Add** **Add from Clipboard** **Delete** **Up** **Down**



6. Membuka halaman tambah keluarga



7. Men-submit halaman tambah keluarga

HTTP Request

Method: POST	Path: http://localhost:8080/keluarga/tambah	Content encoding: <input type="text"/>																				
<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																						
<input type="radio"/> Parameters <input type="radio"/> Body Data <input type="radio"/> Files Upload																						
Send Parameters With the Request:																						
<table border="1"> <thead> <tr> <th>Name:</th> <th>Value</th> <th>Encode?</th> <th>Include Equals?</th> </tr> </thead> <tbody> <tr> <td>alamat</td> <td>jl paus</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>rt</td> <td>199</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>rw</td> <td>23</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>id_kelurahan</td> <td>2</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table>			Name:	Value	Encode?	Include Equals?	alamat	jl paus	<input type="checkbox"/>	<input checked="" type="checkbox"/>	rt	199	<input type="checkbox"/>	<input checked="" type="checkbox"/>	rw	23	<input type="checkbox"/>	<input checked="" type="checkbox"/>	id_kelurahan	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Name:	Value	Encode?	Include Equals?																			
alamat	jl paus	<input type="checkbox"/>	<input checked="" type="checkbox"/>																			
rt	199	<input type="checkbox"/>	<input checked="" type="checkbox"/>																			
rw	23	<input type="checkbox"/>	<input checked="" type="checkbox"/>																			
id_kelurahan	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>																			

View Results in Table.jmx (/Users/mayanggitaindra/Downloads/apache-jmeter-3.2/bin/View Results in Table.jmx) - Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - View Results**
- WorkBench

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

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

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	22:14:43.308	Thread Group...	HTTP Request	59	✓	45193	245	57	1
2	22:14:43.204	Thread Group...	HTTP Request	163	✓	45193	245	161	3
3	22:14:43.410	Thread Group...	HTTP Request	51	✓	45193	245	51	1
4	22:14:43.506	Thread Group...	HTTP Request	46	✓	45193	245	45	1
5	22:14:43.608	Thread Group...	HTTP Request	51	✓	45193	245	51	1
6	22:14:43.711	Thread Group...	HTTP Request	41	✓	45193	245	40	0
7	22:14:43.810	Thread Group...	HTTP Request	39	✓	45193	245	38	1
8	22:14:43.910	Thread Group...	HTTP Request	53	✓	45193	245	52	1
9	22:14:44.010	Thread Group...	HTTP Request	49	✓	45193	245	48	1
10	22:14:44.110	Thread Group...	HTTP Request	39	✓	45193	245	39	0
11	22:14:44.210	Thread Group...	HTTP Request	40	✓	45193	245	39	0
12	22:14:44.312	Thread Group...	HTTP Request	38	✓	45193	245	37	1
13	22:14:44.412	Thread Group...	HTTP Request	47	✓	45193	245	46	1
14	22:14:44.509	Thread Group...	HTTP Request	41	✓	45193	245	40	1
15	22:14:44.607	Thread Group...	HTTP Request	46	✓	45193	245	45	1
16	22:14:44.713	Thread Group...	HTTP Request	41	✓	45193	245	40	0
17	22:14:44.814	Thread Group...	HTTP Request	42	✓	45193	245	41	0
18	22:14:44.909	Thread Group...	HTTP Request	41	✓	45193	245	40	1
19	22:14:45.012	Thread Group...	HTTP Request	39	✓	45193	245	39	1
20	22:14:45.113	Thread Group...	HTTP Request	42	✓	45193	245	41	1
21	22:14:45.211	Thread Group...	HTTP Request	37	✓	45193	245	36	0
22	22:14:45.312	Thread Group...	HTTP Request	46	✓	45193	245	46	1
23	22:14:45.412	Thread Group...	HTTP Request	67	✓	45193	245	66	1
24	22:14:45.514	Thread Group...	HTTP Request	38	✓	45193	245	38	1
25	22:14:45.613	Thread Group...	HTTP Request	38	✓	45193	245	37	0

Scroll automatically? Child samples? No of Samples 100 Latest Sample 27 Average 35 Deviation 15

8. Membuka dan men-submit halaman update penduduk

View Results in Table.jmx (/Users/mayanggitaindra/Downloads/apache-jmeter-3.2/bin/View Results in Table.jmx) - Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - View Results**
- Thread Group
 - HTTP Request
 - View Results**
- WorkBench

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

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

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	22:21:08.664	bad Group 2-1	HTTP Request	160	✓	2750	147	160	1
2	22:21:08.748	Thread Grou...	HTTP Request	77	✓	7300	148	77	1
3	22:21:08.765	Thread Grou...	HTTP Request	61	✓	2750	147	59	1
4	22:21:08.647	Thread Grou...	HTTP Request	179	✓	7300	148	177	2
5	22:21:08.847	Thread Grou...	HTTP Request	10	✓	7300	148	10	1
6	22:21:08.883	Thread Grou...	HTTP Request	5	✓	2750	147	5	1
7	22:21:08.948	Thread Grou...	HTTP Request	8	✓	7300	148	8	0
8	22:21:08.966	Thread Grou...	HTTP Request	7	✓	2750	147	7	0
9	22:21:09.048	Thread Grou...	HTTP Request	18	✓	7300	148	18	0
10	22:21:09.067	Thread Grou...	HTTP Request	6	✓	2750	147	6	1
11	22:21:09.151	Thread Grou...	HTTP Request	9	✓	7300	148	9	1
12	22:21:09.166	Thread Grou...	HTTP Request	6	✓	2750	147	6	1
13	22:21:09.249	Thread Grou...	HTTP Request	12	✓	7300	148	12	1
14	22:21:09.270	Thread Grou...	HTTP Request	4	✓	2750	147	4	0
15	22:21:09.353	Thread Grou...	HTTP Request	9	✓	7300	148	9	0
16	22:21:09.366	Thread Grou...	HTTP Request	5	✓	2750	147	5	1
17	22:21:09.452	Thread Grou...	HTTP Request	10	✓	7300	148	10	1
18	22:21:09.470	Thread Grou...	HTTP Request	5	✓	2750	147	5	1
19	22:21:09.552	Thread Grou...	HTTP Request	9	✓	7300	148	9	1
20	22:21:09.568	Thread Grou...	HTTP Request	5	✓	2750	147	5	0
21	22:21:09.650	Thread Grou...	HTTP Request	18	✓	7300	148	18	1
22	22:21:09.673	Thread Grou...	HTTP Request	6	✓	2750	147	6	0
23	22:21:09.752	Thread Grou...	HTTP Request	8	✓	7300	148	8	0
24	22:21:09.767	Thread Grou...	HTTP Request	6	✓	2750	147	6	1
25	22:21:09.849	Thread Grou...	HTTP Request	11	✓	7300	148	11	1

Scroll automatically? Child samples? No of Samples 200 Latest Sample 7 Average 10 Deviation 17

HTTP Request

Method: POST Path: <http://localhost:8080/penduduk/ubah/310101808190> 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
nama	hai	<input type="checkbox"/>	<input checked="" type="checkbox"/>
tempat_lahir	jakarta	<input type="checkbox"/>	<input checked="" type="checkbox"/>
tanggal_lahir	1993-08-08	<input type="checkbox"/>	<input checked="" type="checkbox"/>
jenis_kelamin	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
golongan_darah	A-	<input type="checkbox"/>	<input checked="" type="checkbox"/>
agama	islam	<input type="checkbox"/>	<input checked="" type="checkbox"/>
status_perkawinan	kawin	<input type="checkbox"/>	<input checked="" type="checkbox"/>
pekerjaan	pegawai swasta	<input type="checkbox"/>	<input checked="" type="checkbox"/>
is_wni	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
is_wafat	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
id_keluarga	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>
status_dalam_keluarga	anak	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Detail **Add** **Add from Clipboard** **Delete** **Up** **Down**

View Results in Table.jmx (/Users/mayanggitaindra/Downloads/apache-jmeter-3.2/bin/View Results in Table.jmx) - Apache JMeter (3.2 r1790748)

File Edit Search Run Options Help

Test Plan Thread Group View Results in Table WorkBench

View Results in Table

Name: View Results in Table

Comments:

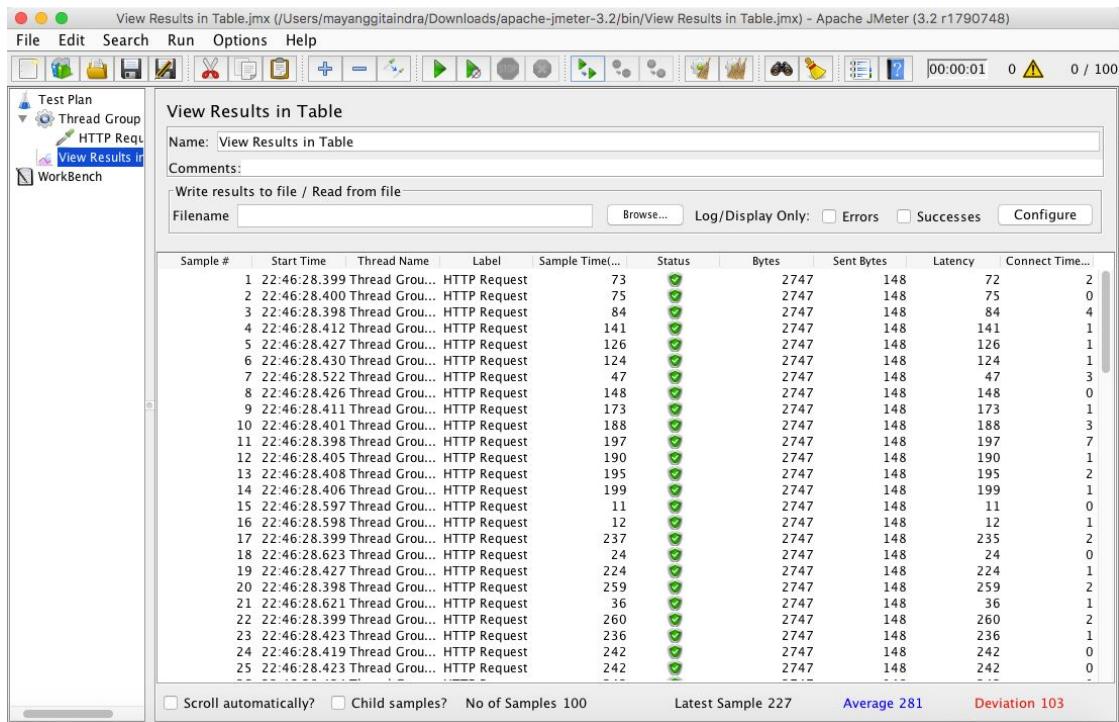
Write results to file / Read from file

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

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	22:51:24.682	Thread Group	HTTP Request	9372	✓	3028	221	9372	1
2	22:51:24.711	Thread Group	HTTP Request	9991	✓	3028	221	9991	1
3	22:51:24.728	Thread Group	HTTP Request	9990	✓	3028	221	9990	1
4	22:51:24.714	Thread Group	HTTP Request	10004	✓	3028	221	10004	1
5	22:51:25.039	Thread Group	HTTP Request	9704	✓	3028	221	9704	1
6	22:51:25.102	Thread Group	HTTP Request	9645	✓	3028	221	9645	1
7	22:51:25.071	Thread Group	HTTP Request	9717	✓	3028	221	9717	1
8	22:51:24.694	Thread Group	HTTP Request	10101	✓	3028	221	10101	2
9	22:51:25.068	Thread Group	HTTP Request	9781	✓	3028	221	9781	0
10	22:51:24.686	Thread Group	HTTP Request	10318	✓	3028	221	10317	2
11	22:51:24.716	Thread Group	HTTP Request	10296	✓	3028	221	10296	1
12	22:51:24.682	Thread Group	HTTP Request	10597	✓	3028	221	10597	1
13	22:51:25.083	Thread Group	HTTP Request	10293	✓	3028	221	10293	0
14	22:51:24.692	Thread Group	HTTP Request	10684	✓	3028	221	10684	1
15	22:51:24.725	Thread Group	HTTP Request	10662	✓	3028	221	10662	2
16	22:51:24.950	Thread Group	HTTP Request	10443	✓	3028	221	10443	1
17	22:51:24.755	Thread Group	HTTP Request	10643	✓	3028	221	10643	2
18	22:51:24.721	Thread Group	HTTP Request	10677	✓	3028	221	10677	1
19	22:51:24.684	Thread Group	HTTP Request	10719	✓	3028	221	10719	1
20	22:51:24.713	Thread Group	HTTP Request	10690	✓	3028	221	10690	1
21	22:51:24.888	Thread Group	HTTP Request	10550	✓	3028	221	10549	1
22	22:51:24.731	Thread Group	HTTP Request	10707	✓	3028	221	10707	1
23	22:51:24.717	Thread Group	HTTP Request	10725	✓	3028	221	10725	2
24	22:51:24.772	Thread Group	HTTP Request	10670	✓	3028	221	10670	1
25	22:51:24.726	Thread Group	HTTP Request	10720	✓	3028	221	10720	1

Scroll automatically? Child samples? No of Samples 100 Latest Sample 10871 Average 10709 Deviation 355

9. Membuka dan men-submit halaman update keluarga



HTTP Request

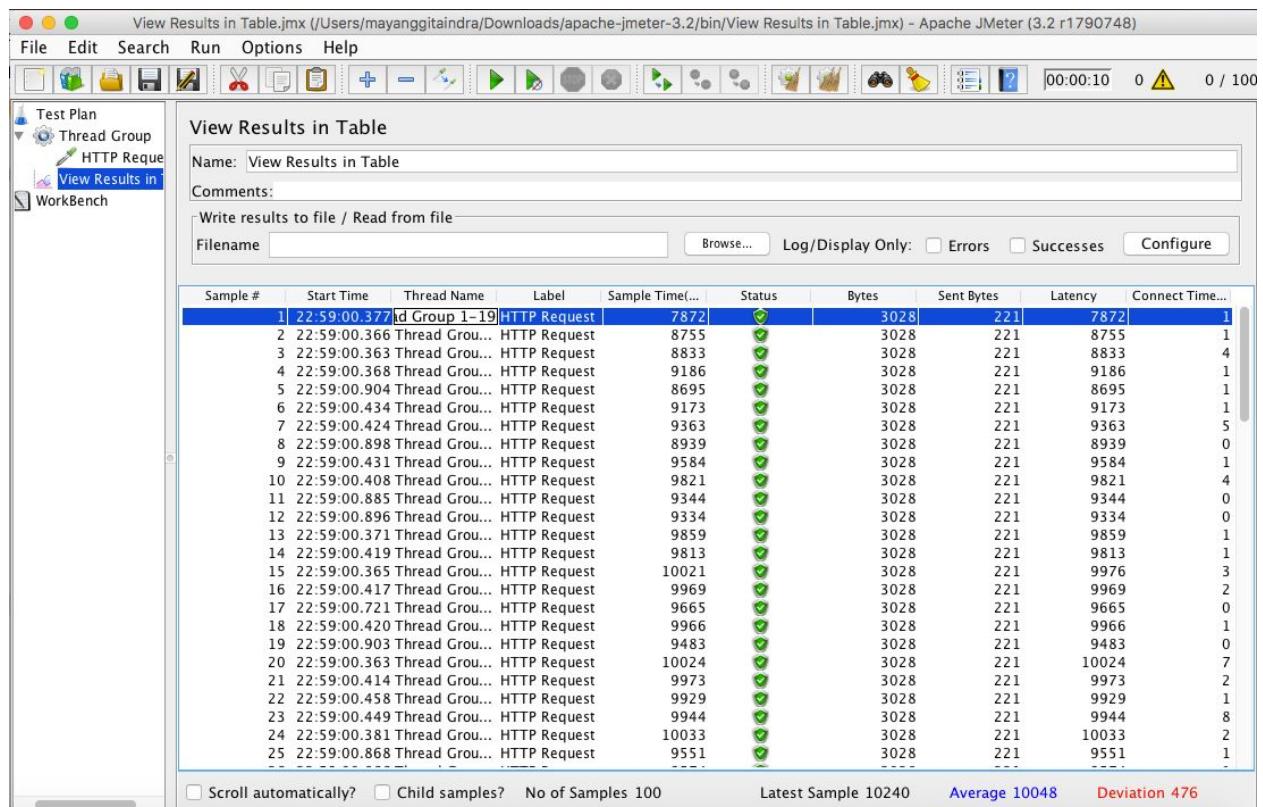
Method: POST Path: http://localhost:8080/keluarga/ubah/3101012210170001 Content encoding:

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

Parameters Body Data Files Upload

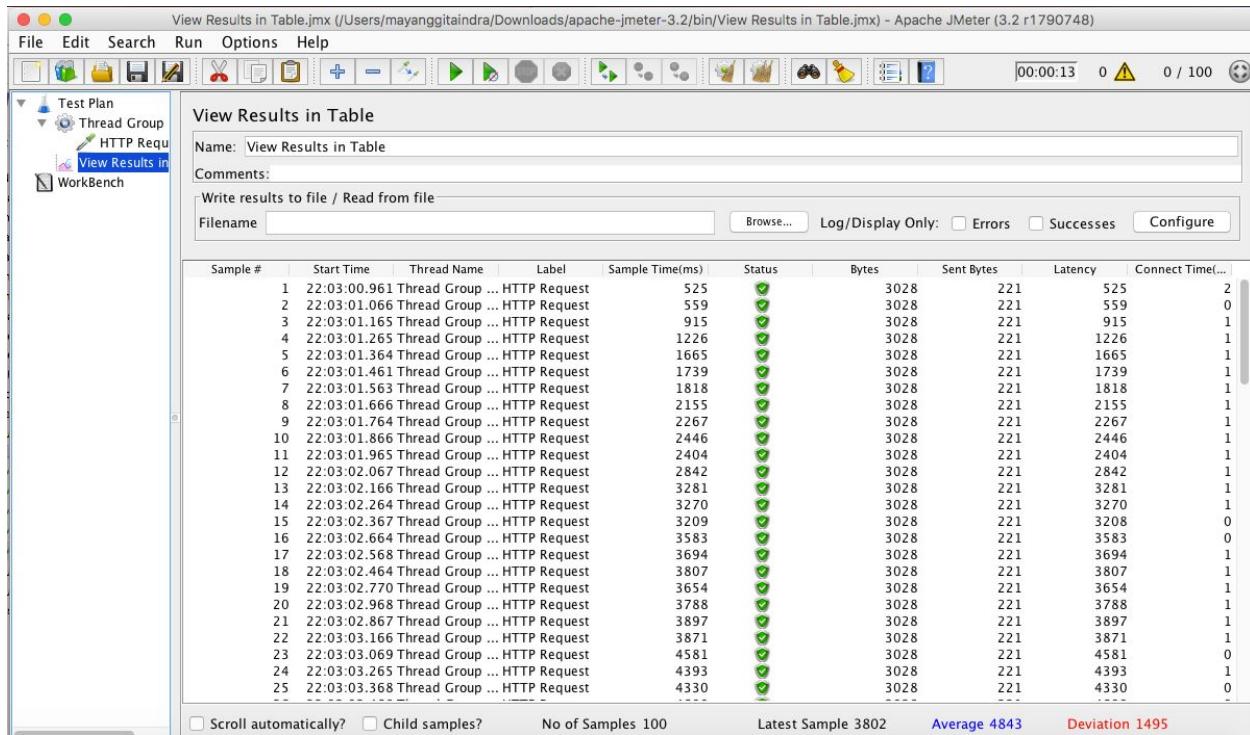
Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
alamat	jl puyuh	<input type="checkbox"/>	<input checked="" type="checkbox"/>
rt	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
rw	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>
id_kelurahan	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>



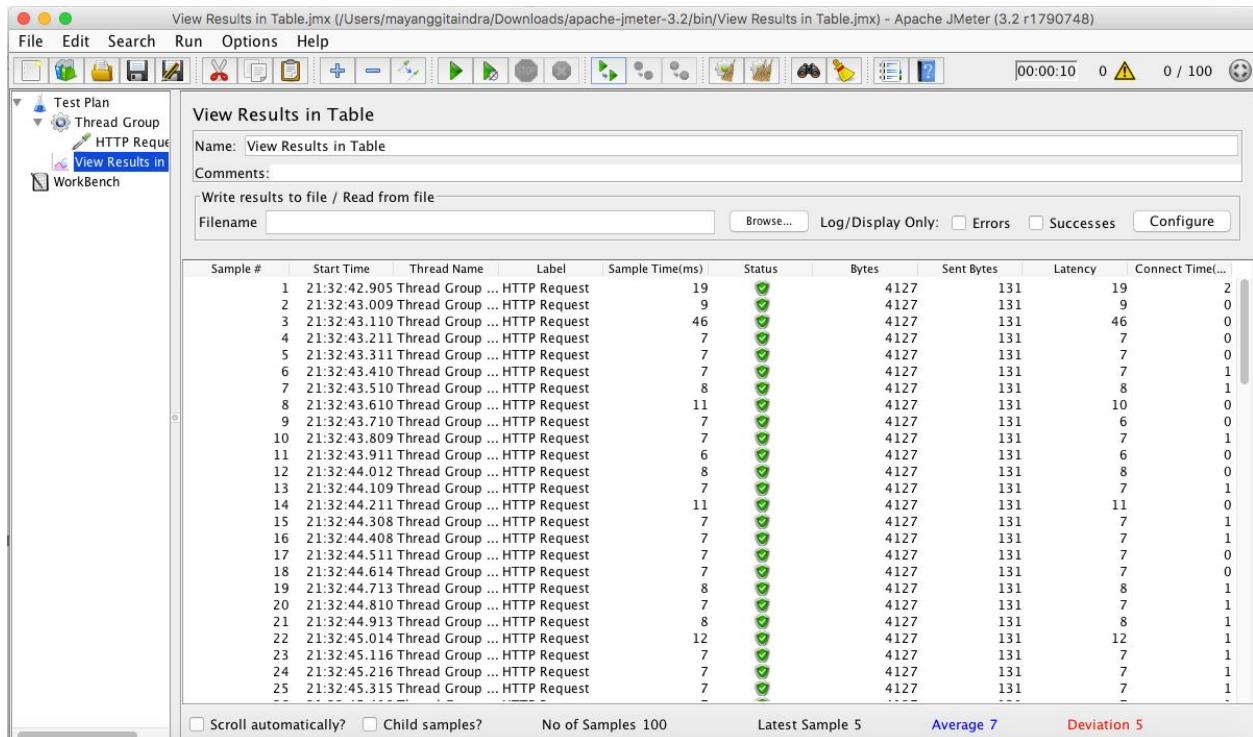
10. Menonaktifkan penduduk

The screenshot shows the 'HTTP Request' configuration dialog. The method is set to 'POST' and the path is 'http://localhost:8080/penduduk/mati'. Other options like 'Follow Redirects' and 'Use KeepAlive' are checked. The 'Parameters' tab is selected, showing a parameter named 'nik' with the value '3101011405170001'. The 'Encode?' checkbox is unchecked, while 'Include Equals?' is checked.



11. Mencari penduduk per kelurahan

- Membuka halaman cari penduduk (<http://localhost:8080/penduduk/cari>)



b. Pilih kota, lihat (<http://localhost:8080/penduduk/cari?kt=2>)

The screenshot shows the Apache JMeter interface with a test plan containing a single thread group and a view results in table listener. The results table displays 100 samples, each representing an HTTP request. The columns include Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, Sent Bytes, Latency, and Connect Time(...). All requests are successful (Status column has green checkmarks) and have a response size of 4349 bytes and a latency of 136 ms.

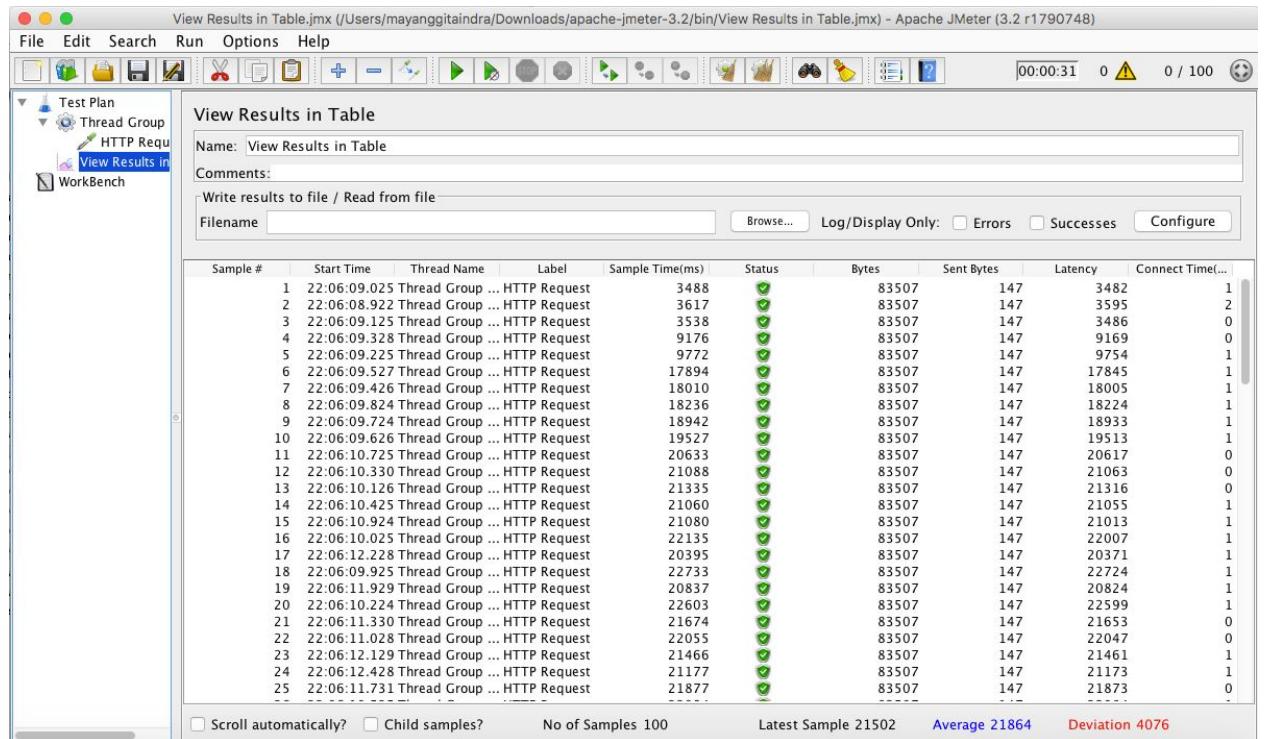
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	21:34:56.203	Thread Group ...	HTTP Request	12	✓	4349	136	12	2
2	21:34:56.308	Thread Group ...	HTTP Request	9	✓	4349	136	9	1
3	21:34:56.409	Thread Group ...	HTTP Request	7	✓	4349	136	6	0
4	21:34:56.509	Thread Group ...	HTTP Request	10	✓	4349	136	10	1
5	21:34:56.609	Thread Group ...	HTTP Request	8	✓	4349	136	8	1
6	21:34:56.706	Thread Group ...	HTTP Request	8	✓	4349	136	8	0
7	21:34:56.811	Thread Group ...	HTTP Request	7	✓	4349	136	7	0
8	21:34:56.908	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
9	21:34:57.008	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
10	21:34:57.108	Thread Group ...	HTTP Request	7	✓	4349	136	7	0
11	21:34:57.208	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
12	21:34:57.307	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
13	21:34:57.408	Thread Group ...	HTTP Request	6	✓	4349	136	6	0
14	21:34:57.507	Thread Group ...	HTTP Request	6	✓	4349	136	6	0
15	21:34:57.613	Thread Group ...	HTTP Request	7	✓	4349	136	7	0
16	21:34:57.713	Thread Group ...	HTTP Request	7	✓	4349	136	7	0
17	21:34:57.813	Thread Group ...	HTTP Request	6	✓	4349	136	6	0
18	21:34:57.912	Thread Group ...	HTTP Request	7	✓	4349	136	7	0
19	21:34:58.012	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
20	21:34:58.112	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
21	21:34:58.212	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
22	21:34:58.313	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
23	21:34:58.413	Thread Group ...	HTTP Request	7	✓	4349	136	7	1
24	21:34:58.508	Thread Group ...	HTTP Request	8	✓	4349	136	8	1
25	21:34:58.613	Thread Group ...	HTTP Request	7	✓	4349	136	7	1

c. Pilih kecamatan, lihat (<http://localhost:8080/penduduk/cari?kt=2&kc=4>)

The screenshot shows the Apache JMeter interface with a test plan containing a single thread group and a view results in table listener. The results table displays 100 samples, each representing an HTTP request. The columns include Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, Sent Bytes, Latency, and Connect Time(...). All requests are successful (Status column has green checkmarks) and have a response size of 4278 bytes and a latency of 141 ms.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	21:36:34.369	Thread Group ...	HTTP Request	13	✓	4278	141	13	2
2	21:36:34.473	Thread Group ...	HTTP Request	9	✓	4278	141	8	1
3	21:36:34.574	Thread Group ...	HTTP Request	8	✓	4278	141	8	0
4	21:36:34.675	Thread Group ...	HTTP Request	8	✓	4278	141	7	0
5	21:36:34.772	Thread Group ...	HTTP Request	8	✓	4278	141	8	1
6	21:36:34.873	Thread Group ...	HTTP Request	8	✓	4278	141	8	0
7	21:36:34.971	Thread Group ...	HTTP Request	8	✓	4278	141	8	0
8	21:36:35.071	Thread Group ...	HTTP Request	9	✓	4278	141	9	1
9	21:36:35.172	Thread Group ...	HTTP Request	9	✓	4278	141	9	1
10	21:36:35.271	Thread Group ...	HTTP Request	10	✓	4278	141	10	1
11	21:36:35.373	Thread Group ...	HTTP Request	8	✓	4278	141	8	1
12	21:36:35.473	Thread Group ...	HTTP Request	7	✓	4278	141	7	1
13	21:36:35.573	Thread Group ...	HTTP Request	6	✓	4278	141	6	0
14	21:36:35.673	Thread Group ...	HTTP Request	7	✓	4278	141	7	0
15	21:36:35.773	Thread Group ...	HTTP Request	7	✓	4278	141	7	1
16	21:36:35.877	Thread Group ...	HTTP Request	8	✓	4278	141	8	1
17	21:36:35.977	Thread Group ...	HTTP Request	8	✓	4278	141	8	1
18	21:36:36.074	Thread Group ...	HTTP Request	7	✓	4278	141	7	0
19	21:36:36.177	Thread Group ...	HTTP Request	8	✓	4278	141	8	1
20	21:36:36.274	Thread Group ...	HTTP Request	7	✓	4278	141	7	1
21	21:36:36.375	Thread Group ...	HTTP Request	7	✓	4278	141	7	1
22	21:36:36.477	Thread Group ...	HTTP Request	8	✓	4278	141	8	1
23	21:36:36.577	Thread Group ...	HTTP Request	7	✓	4278	141	7	1
24	21:36:36.673	Thread Group ...	HTTP Request	8	✓	4278	141	8	1
25	21:36:36.773	Thread Group ...	HTTP Request	6	✓	4278	141	6	1

d. Isi kelurahan, lihat (<http://localhost:8080/penduduk/cari?kt=2&kc=4&kl=18>)



Analisis Jmeter

- Apabila hanya menampilkan view berupa form atau html dan css sederhana (tidak mengambil data dari *database*), program akan tetap responsif walaupun diakses banyak user. Misalnya saat saat membuka halaman index, lihat detail penduduk, lihat detail keluarga, halaman tambah penduduk, men-submit tambah penduduk, halaman tambah keluarga, men-submit tambah keluarga, membuka halaman ubah penduduk, membuka halaman cari penduduk, pilih kota, dan pilih kecamatan.
- Apabila menerima input dan harus mengakses data di database namun dengan query yang sederhana, sistem akan cenderung lebih lama, tetapi hal ini bisa dibantu dengan penggunaan index atau optimasi database lainnya. Misalnya membuka halaman ubah keluarga, dan menonaktifkan penduduk.
- Apabila menerima data yang diakses cukup banyak dan query-nya kompleks, maka program akan menjadi tidak optimal (contoh: pada fitur lihat semua penduduk di suatu kelurahan). Walaupun sudah melakukan optimasi, program masih berjalan cukup lama. Misalnya saat men-submit

halaman ubah penduduk, men-submit halaman ubah keluarga dan saat mencari list penduduk yang ada di kelurahan tertentu.