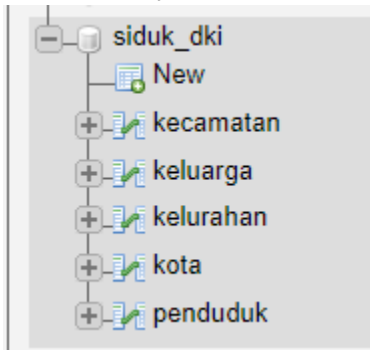


## Writeup Tugas 1 APAP

Proses pengembangan program berjalan sekitar 2 minggu, dengan mengerjakan *backend* dahulu (fungsionalitas) kemudian *frontend* (desain interface aplikasi). Pengerjaan aplikasi menggunakan program Eclipse Spring Boot Application dan database MySQL.



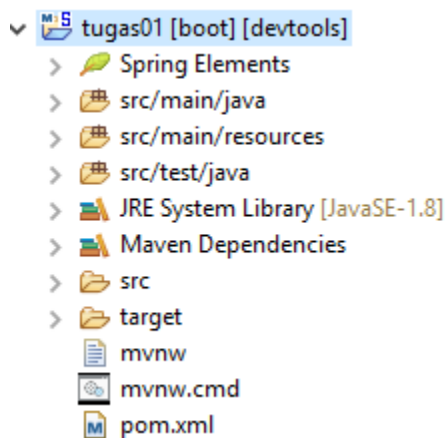
Program ini bertujuan untuk mengolah data penduduk dan keluarga Kota Jakarta. Program ini dapat menambah, menghapus, dan mengubah data pada database. Program disupport oleh Thymeleaf yang memudahkan output objek java pada interface di HTML.

Proses diawali dengan mengimport database, mengambil sebagian data pada database untuk memudahkan pengecekan fitur, pembuatan model yang berkaitan; yaitu PendudukModel, KeluargaModel, KelurahanModel, KecamatanModel, dan KotaModel, pembuatan fitur-fitur agar berjalan sesuai dengan *requirement*, dan akhirnya mendesain interface aplikasi menggunakan CSS Bootstrap.

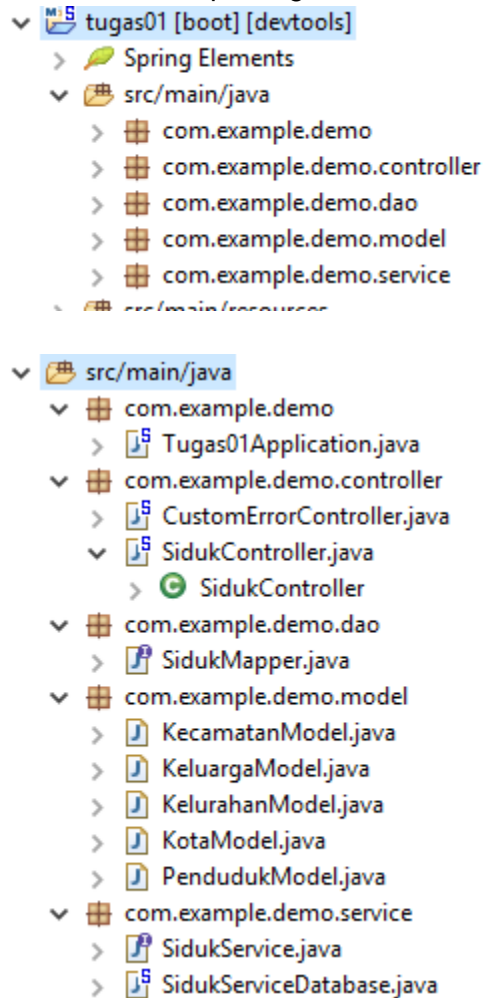
### 1. Konsep MVC pada program (struktur project dan package)

Struktur proyek terdiri dari Model, View, dan Controller. Model berfungsi untuk menyimpan atribut suatu tabel (data) di database, View berfungsi untuk menampilkan data, dan Controller berfungsi untuk mengolah data.

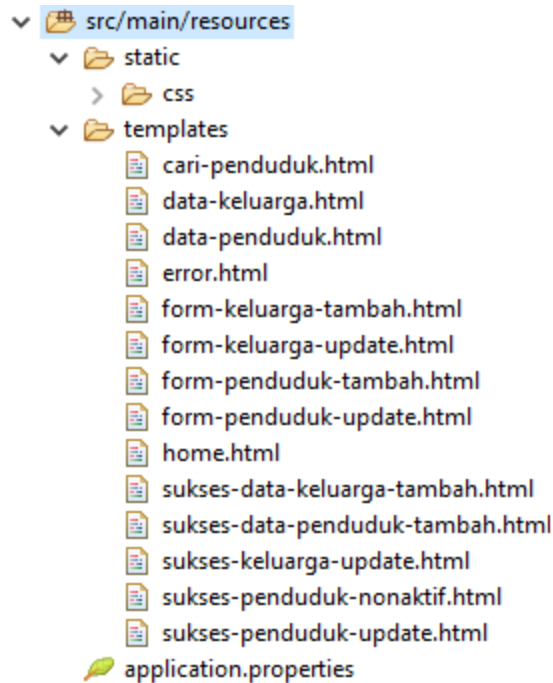
Package yang dibuat yaitu `src/main/java`, `src/main/resources`, dan `src/test/java`.



Main terdiri dari package Controller, DAO, Model, dan Service.



Resources terdiri dari static dan templates. Resources berguna untuk penampilan interface bagi para pengguna. Folder static menyediakan css untuk mengganti dan memanipulasi tampilan, dan templates berisi halaman-halaman html untuk memberikan output dari hasil olahan data.



Setiap fitur memiliki method yang bersesuaian dengan kebutuhannya, method dibuat melalui proses sebagai berikut:

1. Pembuatan interface di SidukService.java – membuat *constructor* method.
2. Pembuatan method di SidukMapper.java – menghubungkan method yang sudah di-*construct* dengan database dengan cara penggunaan anotasi @Select, @Update, dan sebagainya di atas method.
3. Pembuatan method di SidukServiceDatabase.java – pengisian method yang ada di SidukMapper.
4. Pembuatan method di SidukController.java – pembuatan RequestMapping dan pengaplikasian method sesuai *requirement* yang diinginkan.
5. Pembuatan halaman html yang berkaitan.

## 2. Penjelasan fitur (implementasi MVC)

### a. Tampilkan Data Penduduk Berdasarkan NIK

Initial Page: /

Form Request: GET, /penduduk

Response Page: /penduduk?nik={NIK}

Fitur dimulai dengan pengisian form di home.html (mapping /)

```
@RequestMapping("/")
public String home ()
{
    return "home";
}
```

dan kemudian merefer ke data-penduduk.html (mapping /penduduk). Method yang dibuat adalah SelectPenduduk. Hasil input di-GET dan dimasukkan ke

May Iffah Rizki  
1506757661  
APAP-A

method untuk dicarikan penduduk yang sesuai berdasarkan NIK yang diinput, serta penggunaan `changeDefault` yang ada di `PendudukModel` untuk mengganti data pada database agar sesuai dengan yang diminta (contoh `jenis_kelamin = 0` menjadi `jenis_kelamin = Pria`).

```
@RequestMapping("/penduduk")
public String penduduk (@RequestParam(value = "nik", required = false, defaultValue = "tes") String nik, Model model)
{
    if (sudahNonAktif == true) {
        model.addAttribute("nik", nik);
        sudahNonAktif = false;
        return "sukses-penduduk-nonaktif";
    } else {
        PendudukModel penduduk = sidukDAO.selectPenduduk(nik);
        penduduk.changeDefault(nik);
        model.addAttribute("penduduk", penduduk);
        return "data-penduduk";
    }
}
```



home.html

The screenshot shows a web browser window with the address bar at `localhost:8080`. The page has a dark header with the title "SIDUK" and navigation links: "Home", "Cari Penduduk", "Tambah Penduduk", and "Tambah Keluarga". The main content area has a light gray background and contains the following text and form elements:

- Header: "Sistem Informasi Kependudukan Online Provinsi DKI Jakarta"
- Greeting: "Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta"
- Form 1: "Masukkan Nomor Induk Kependudukan" with a text input field and a "Lihat" button.
- Form 2: "Masukkan Nomor Kartu Keluarga" with a text input field and a "Lihat" button.

Memasukkan nomor penduduk.

This screenshot shows the same web page as the previous one, but with the number "3101011405170001" entered into the "Masukkan Nomor Induk Kependudukan" input field. The "Lihat" button is still visible below the input field.

data-penduduk.html

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

- b. Tampilkan Data Keluarga Beserta Daftar Anggotanya Berdasarkan Nomor KK  
Initial Page: /  
Form Request: GET, /keluarga  
Response Page: /keluarga?nkk={NKK}

Fitur dimulai dengan pengisian form di home.html (mapping /)

```
@RequestMapping("/")
public String home ()
{
    return "home";
}
```

dan kemudian merefer ke data-keluarga.html (mapping /keluarga). Method yang dibuat adalah SelectKeluarga dan selectAnggotaKeluarga. Hasil input di-GET dan dimasukkan ke method untuk dicarikan keluarga yang sesuai berdasarkan

NKK yang diinput, serta penggunaan selectAnggotaKeluarga yang mencari penduduk sesuai dengan id keluarga yang ada di NKK.

```
@RequestMapping("/keluarga")
public String keluarga (@RequestParam(value = "nkk", required = false, defaultValue = "tes") String nkk, Model model)
{
    KeluargaModel keluarga = sidukDAO.selectKeluarga(nkk);
    int sizeAnggotaKeluarga = keluarga.getAnggotaKeluarga().size();
    if (sizeAnggotaKeluarga != 0) {
        for (int i = 0; i < sizeAnggotaKeluarga; i++) {
            PendudukModel index = keluarga.getAnggotaKeluarga().get(i);
            String nik = index.getNik();
            keluarga.getAnggotaKeluarga().get(i).changeDefault(nik);
        }
    }
    model.addAttribute("keluarga", keluarga);
    return "data-keluarga";
}
```

home.html

The screenshot shows a web browser window with the URL `localhost:8080`. The page title is "Sistem Informasi Kependudukan Online Provinsi DKI Jakarta". The navigation bar includes "SIDUK" and links for "Home", "Cari Penduduk", "Tambah Penduduk", and "Tambah Keluarga". The main content area has a heading "Sistem Informasi Kependudukan Online Provinsi DKI Jakarta" and a subtitle "Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta". There are two search sections: "Masukkan Nomor Induk Kependudukan" with a text input field and a "Lihat" button, and "Masukkan Nomor Kartu Keluarga" with a text input field and a "Lihat" button.

Memasukkan nomor kartu keluarga

This is a close-up of the "Masukkan Nomor Kartu Keluarga" search field. The text input field contains the value "3101010101050001". Below the input field is a "Lihat" button.

May Iffah Rizki  
1506757661  
APAP-A

data-keluarga.html

Lihat Data Keluarga

New Tab

New Tab

localhost:8080/keluarga?nkk=3101010101050001

Apps Home | Bill Gates OpenStax College Student Centered e-L Top 30 OOPS Conce Dashboard | Khan Ac Perubahan nomor Co Single Sign On - Inte Konsep Pemrogram

SIDUK Home Cari Penduduk Tambah Penduduk Tambah Keluarga

Lihat Data Keluarga - 3101010101050001

NKK3101010101050001

AlamatJln. Cikapayang No. 594

RT/RW149/090

Kelurahan/DesaPULAU PARI

KecamatanKEPULAUAN SERIBU SELATAN

KotaKABUPATEN KEPULAUAN SERIBU

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

- c. Menambahkan Penduduk Baru Sebagai Anggota Keluarga  
Initial Page: /penduduk/tambah  
Form Request: POST, /penduduk/tambah  
Response Page: /penduduk/tambah

Fitur dimulai dengan pengisian form di form-penduduk-tambah.html (mapping /penduduk/tambah)

```
@RequestMapping("/penduduk/tambah")
public String pendudukTambah (Model model)
{
    PendudukModel penduduk = new PendudukModel();

    model.addAttribute ("penduduk", penduduk);
    return "form-penduduk-tambah";
}
```

dan kemudian merefer ke sukses-data-penduduk-tambah.html jika sukses ditambah (mapping /penduduk/tambah).

```
@RequestMapping(value= "/penduduk/tambah", method = RequestMethod.POST)
public String pendudukTambahSubmit (@ModelAttribute PendudukModel penduduk, Model model)
{
    int id = sidukDAO.countPendudukRow();
    penduduk.setId(id+1);
    long id_keluarga = penduduk.getId_keluarga();
    String idString = String.valueOf(id_keluarga);
    //cari id kelurahan sesuai id keluarga
    String kode_kelurahan = sidukDAO.selectKodeKelurahan(idString);
    //firstSix = ambil string 0,5 -> 6 digit pertama
    String firstSix = kode_kelurahan.substring(0, 6);
    String tanggal_lahir = penduduk.getTanggal_lahir();
    String[] tanggal_lahir_split = tanggal_lahir.split("-");
    String tanggal = tanggal_lahir_split[2];
    String bulan = tanggal_lahir_split[1];
    String tahun = tanggal_lahir_split[0].substring(2);
    StringBuilder tglAppend = new StringBuilder().append(tanggal).append(bulan).append(tahun);

    String tanggalFix = tglAppend.toString();
    String jenis_kelamin = penduduk.getJenis_kelamin();

    if (jenis_kelamin.equals("Wanita")) {
        long tanggalLahir = Long.parseLong(tanggal);
        tanggalLahir = tanggalLahir + 40;
        tanggal_lahir = String.valueOf(tanggalLahir);
        StringBuilder tglBaru = new StringBuilder().append(tanggal_lahir).append(bulan).append(tahun);
        tanggalFix = tglBaru.toString();
    }
    String secondSix = tanggalFix;
```

Method yang dibuat adalah addPenduduk, selectKodeKelurahan, dan countPendudukRow. Hasil input di-POST dan dimasukkan ke method untuk tambah ke tabel penduduk, untuk menggenerate NIK, diperlukan kode kelurahan yang diambil dari method selectKodeKelurahan untuk membuat 6 digit pertama, serta jenis\_kelamin dan tanggal\_lahir dari PendudukModel untuk membuat 6 digit kedua yang berupa tanggal lahir; jika wanita akan ditambah tanggal lahirnya, dan countPendudukRow untuk menentukan id penduduk baru (menambah 1 dari jumlah row table penduduk).



NIK final adalah 16 digit dengan 4 digit terakhir urutan kesamaan 12 digit awal NIK dengan penduduk yang sudah ada di database.

```
String nomor_urut = "0001";
int counter = 1;
StringBuilder nikAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut);
String nik = nikAppend.toString();
//generate nik -> append semua + 0001
//loop
boolean masihAda = true;
while (masihAda == true) {
    PendudukModel cekPenduduk = sidukDAO.selectPenduduk (nik);
    if (cekPenduduk != null) {
        counter+=1;
        int digit_counter = String.valueOf(counter).length();
        StringBuilder nomor_urut_baru = new StringBuilder();
        if (digit_counter == 1) {
            nomor_urut_baru = new StringBuilder().append("000").append(String.valueOf(counter));
        } else if (digit_counter == 2) {
            nomor_urut_baru = new StringBuilder().append("00").append(String.valueOf(counter));
        } else if (digit_counter == 3) {
            nomor_urut_baru = new StringBuilder().append("0").append(String.valueOf(counter));
        } else {
            nomor_urut_baru = new StringBuilder().append(String.valueOf(counter));
        }
        StringBuilder nikBaruAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut_baru.toString());
        String nikBaru = nikBaruAppend.toString();
        nik = nikBaru;
    } else {
        masihAda = false;
    }
}
penduduk.setNik(nik);
```

Kemudian, menyesuaikan data yang diinput di html dengan data yang akan dimasukkan ke database.

```
if (penduduk.getIs_wni().equals("WNI") || penduduk.getIs_wni().equals("wni")) {
    penduduk.setIs_wni("1");
} else {
    penduduk.setIs_wni("0");
}

if (penduduk.getIs_wafat().equals("Hidup")) {
    penduduk.setIs_wafat("0");
} else {
    penduduk.setIs_wafat("1");
}

if (penduduk.getJenis_kelamin().equals("Pria")) {
    penduduk.setJenis_kelamin("0");
} else {
    penduduk.setJenis_kelamin("1");
}

String tanggal_lahir_fix = penduduk.getTanggal_lahir();
String[] tanggal_lahir_split_fix = tanggal_lahir_fix.split("-");
String tanggal_fix = tanggal_lahir_split_fix[2];
String bulan_fix = tanggal_lahir_split_fix[1];
String tahun_fix = tanggal_lahir_split_fix[0];
StringBuilder tglAppendFix = new StringBuilder().append(tahun_fix).append("/").append(bulan_fix).append("/").append(tanggal_fix);

penduduk.setTanggal_lahir(tglAppendFix.toString());

sidukDAO.addPenduduk (penduduk);
model.addAttribute("nikbaru", nik);
return "sukses-data-penduduk-tambah";
```

May Iffah Rizki  
1506757661  
APAP-A

form-penduduk-tambah.html

The screenshot shows a web browser with three tabs: 'Tambah Penduduk', 'New Tab', and 'New Tab'. The address bar shows 'localhost:8080/penduduk/tambah'. The browser's bookmark bar includes 'Apps', 'Home | Bill Gates', 'OpenStax College', 'Student Centered e-L', 'Top 30 OOPS Concep', 'Dashboard | Khan Ac', and 'Perubahan nomor Ca'. The page has a dark navigation bar with 'SIDUK' and links to 'Home', 'Cari Penduduk', 'Tambah Penduduk', and 'Tambah Keluarga'. The main content area is titled 'Tambah Penduduk' and contains the following form fields:

- Nama:
- Tempat Lahir:
- Tanggal Lahir:
- Jenis Kelamin:
- Kewarganegaraan:
- Agama:
- Pekerjaan:
- Status Perkawinan:
- Status dalam Keluarga:
- Golongan Darah:

Below the main form, there is a section for 'Status Kematian:' with a dropdown menu set to 'Hidup' and an 'ID Keluarga:' field containing the value '0'. A 'Simpan' button is located at the bottom of this section.

May Iffah Rizki  
1506757661  
APAP-A

## Memasukkan data penduduk baru

Tambah Penduduk

New Tab

New Tab

localhost:8080/penduduk/tambah

Apps Home | Bill Gates OpenStax College Student Centered e-L Top 30 OOPS Concep Dashboard | Khan Ac Perubahan nom

SIDUK Home Cari Penduduk Tambah Penduduk Tambah Keluarga

### Tambah Penduduk

Nama:  
May Iffah

Tempat Lahir:  
Jakarta

Tanggal Lahir:  
04/19/1998

Jenis Kelamin:  
Wanita ▾

Kewarganegaraan:  
WNI

Agama:  
Islam

Pekerjaan:  
Pelajar

Status Perkawinan:  
Belum Kawin ▾

Status dalam Keluarga:  
Anak

Golongan Darah:  
A+ ▾

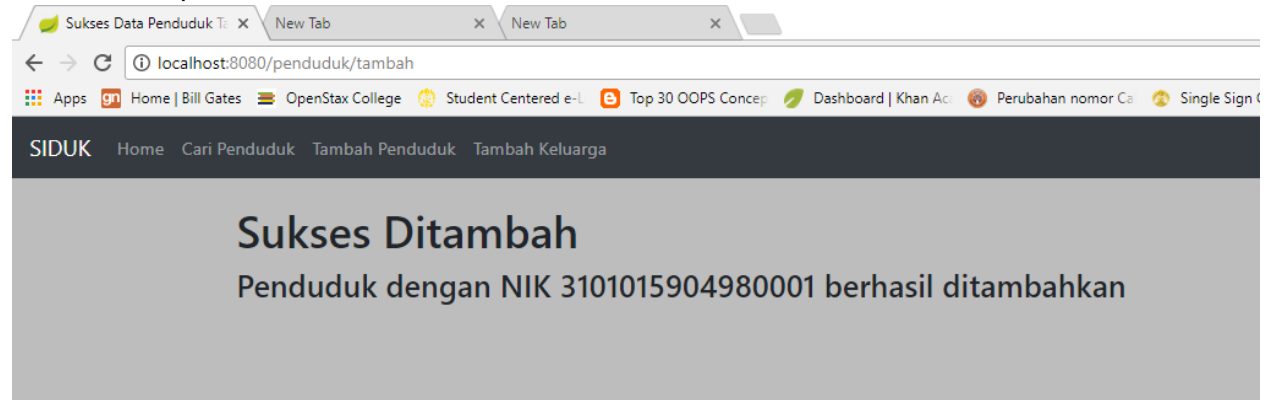
Status Kematian:  
Hidup ▾

ID Keluarga:  
1

Simpan

May Iffah Rizki  
1506757661  
APAP-A

sukses-data-penduduk-tambah.html



Penduduk baru:

281450	3101015904980001	May Iffah	Jakarta	1998-04-19	1	1	1	Islam	Pelajar	Belum Kawin
Anak		A+		0						

- d. Menambahkan Keluarga Baru  
Initial Page: /keluarga/tambah  
Form Request: POST, /keluarga/tambah  
Response Page: /keluarga/tambah

Fitur dimulai dengan pengisian form di form-keluarga-tambah.html (mapping /keluarga/tambah)

```
@RequestMapping("/keluarga/tambah")
public String keluargaTambah (Model model)
{
    KeluargaModel keluarga = new KeluargaModel();

    model.addAttribute ("keluarga", keluarga);
    return "form-keluarga-tambah";
}
```

dan kemudian merefer ke sukses-data-keluarga-tambah.html jika sukses ditambah (mapping /keluarga/tambah). Method yang dibuat adalah addKeluarga, selectIDKelurahan, dan countKeluargaRow. Hasil input di-POST dan dimasukkan ke method untuk tambah ke tabel keluarga, untuk generate NKK, diperlukan kode kelurahan yang diambil dari method selectKodeKelurahan untuk membuat 6 digit pertama, serta selectIDKelurahan untuk mencari kelurahan sesuai nama kelurahan yang diinput (dengan asumsi; nama kelurahan, kecamatan, dan kota sudah sesuai dengan yang ada di database). Kelurahan yang didapatkan dapat dicari kodenya dengan selectKodeKelurahan. Mencari tanggal saat ini digunakan untuk 6 digit kedua.

```
@RequestMapping(value= "/keluarga/tambah", method = RequestMethod.POST)
public String keluargaTambahSubmit (@ModelAttribute KeluargaModel keluarga, Model model)
{
    int id = sidukDAO.countKeluargaRow();
    keluarga.setId(id+1);
    String nama_kelurahan = keluarga.getKelurahan();
    //cari id kelurahan sesuai nama kelurahan
    String id_kelurahan = sidukDAO.selectIDKelurahan(nama_kelurahan);
    keluarga.setId_kelurahan(id_kelurahan);
    long id_kelurahan_long = Long.parseLong(id_kelurahan);
    KelurahanModel tes = sidukDAO.selectKelurahanByID(id_kelurahan_long);
    String kode_kelurahan = tes.getKode_kelurahan();
    //firstSix = ambil string 0,6 -> 6 digit pertama
    String firstSix = kode_kelurahan.substring(0, 6);

    //tanggal submit adalah tanggal sekarang
    DateFormat df = new SimpleDateFormat("dd/MM/yy");
    Date dateobj = new Date();
    String tanggal_submit = df.format(dateobj);

    String[] tanggal_split = tanggal_submit.split("/");
    String tanggal = tanggal_split[0];
    String bulan = tanggal_split[1];
    String tahun = tanggal_split[2];
    StringBuilder tglAppend = new StringBuilder().append(tanggal).append(bulan).append(tahun);

    String tanggalFix = tglAppend.toString();

    String secondSix = tanggalFix;
```

Method countKeluargaRow untuk menentukan id keluarga baru (menambah 1 dari jumlah row table keluarga). NKK final adalah 16 digit dengan 4 digit terakhir urutan kesamaan 12 digit awal NKK dengan keluarga yang sudah ada di database.

```
String nomor_urut = "0001";
int counter = 1;
StringBuilder nkkAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut);
String nkk = nkkAppend.toString();
//generate nik -> append semua + 0001
//loop
boolean masihAda = true;
while (masihAda == true) {
    KeluargaModel cekKeluarga = sidukDAO.selectKeluarga (nkk);
    if (cekKeluarga != null) {
        counter+=1;
        int digit_counter = String.valueOf(counter).length();
        StringBuilder nomor_urut_baru = new StringBuilder();
        if (digit_counter == 1) {
            nomor_urut_baru = new StringBuilder().append("000").append(String.valueOf(counter));
        } else if (digit_counter == 2) {
            nomor_urut_baru = new StringBuilder().append("00").append(String.valueOf(counter));
        } else if (digit_counter == 3) {
            nomor_urut_baru = new StringBuilder().append("0").append(String.valueOf(counter));
        } else {
            nomor_urut_baru = new StringBuilder().append(String.valueOf(counter));
        }
        StringBuilder nkkBaruAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut_baru.toString());
        String nkkBaru = nkkBaruAppend.toString();
        nkk = nkkBaru;
    } else {
        masihAda = false;
    }
}
keluarga.setNomor_kk(nkk);
keluarga.setIs_tidak_berlaku("0");

sidukDAO.addKeluarga (keluarga);
model.addAttribute("nkkbaru", nkk);
return "sukses-data-keluarga-tambah";
}
```

form-keluarga-tambah.html

May Iffah Rizki  
1506757661  
APAP-A

The screenshot displays a web browser with three tabs: 'Tambah Keluarga', 'New Tab', and 'New Tab'. The address bar shows the URL 'localhost:8080/keluarga/tambah'. The browser's toolbar includes icons for Apps, Home, Bill Gates, OpenStax College, Student Centered e-L, Top 30 OOPS Concep, Dashboard | Khan Ac, and Peru.

The web page has a dark navigation bar with the following links: **SIDUK**, Home, Cari Penduduk, Tambah Penduduk, and Tambah Keluarga.

## Tambah Keluarga

Alamat:

RT:

RW:

Kelurahan/Desa:

Kecamatan:

Kota:

May Iffah Rizki  
1506757661  
APAP-A

## Menambahkan keluarga baru

Tambah Keluarga

Alamat:  
Jl. A no. 23

RT:  
8

RW:  
9

Kelurahan/Desa:  
MENTENG DALAM

Kecamatan:  
TEBET

Kota:  
KOTA JAKARTA SELATAN

Simpan

## sukses-data-keluarga-tambah.html

Sukses Ditambah

Keluarga dengan NKK 3171092210170001 berhasil ditambahkan

Keluarga baru:

63482	3171092210170001	Jl. A no. 23	8	9	57	0
-------	------------------	--------------	---	---	----	---



e. Mengubah Data Penduduk

Initial Page: /penduduk/ubah/{NIK}

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

Response Page: /penduduk/ubah/{NIK}

Fitur dimulai dengan pengisian form yang sudah terisi dengan data penduduk (sesuai dengan NIK yang dijadikan parameter dan penduduk sesuai dengan selectPenduduk) pada tiap input field di form-penduduk-ubah.html (mapping /penduduk/ubah/{NIK}) dan kemudian merefer ke sukses-data-penduduk-ubah.html jika sukses diubah (mapping /penduduk/ubah/{NIK}). Data penduduk yang dimasukkan pada awalnya sudah disesuaikan dari database sebelum muncul di halaman html.

```
@RequestMapping("/penduduk/ubah/{nik_lama}")
public String updatePenduduk (Model model, @PathVariable(value = "nik_lama") String nik_lama)
{
    PendudukModel penduduk = sidukDAO.selectPenduduk (nik_lama);
    if (penduduk.getIs_wni().equals("1")) {
        penduduk.setIs_wni("WNI");
    } else {
        penduduk.setIs_wni("WNA");
    }

    if (penduduk.getIs_wafat().equals("0")) {
        penduduk.setIs_wafat("Hidup");
    } else {
        penduduk.setIs_wafat("Mati");
    }

    String jenis_kelamin = penduduk.getJenis_kelamin();
    if (jenis_kelamin.equals("0")) {
        penduduk.setJenis_kelamin("Pria");
    } else {
        penduduk.setJenis_kelamin("Wanita");
    }
    model.addAttribute("penduduk", penduduk);
    return "form-penduduk-update";
}
```

Method yang dibuat adalah updatePenduduk. Hasil input di-POST dan dimasukkan ke method untuk tambah ke tabel penduduk, untuk generate NIK, diperlukan kode kelurahan yang diambil dari method selectKodeKelurahan untuk membuat 6 digit pertama, serta jenis\_kelamin dan tanggal\_lahir dari PendudukModel untuk membuat 6 digit kedua yang berupa tanggal lahir; jika wanita akan ditambah tanggal lahirnya, dan countPendudukRow untuk menentukan id penduduk baru (menambah 1 dari jumlah row table penduduk).

NIK final adalah 16 digit dengan 4 digit terakhir urutan kesamaan 12 digit awal NIK dengan penduduk yang sudah ada di database.

```
@RequestMapping(value = "/penduduk/ubah/{nik_lama}", method = RequestMethod.POST)
public String updatePendudukSubmit (@ModelAttribute PendudukModel penduduk, Model model, @PathVariable(value = "nik_lama") String nik_lama)
{
    String niklama= nik_lama;
    //sesuain yg diinput kayak di db
    //set wni sesuai kalo wni = 1
    //set iswafat sesuai kalo hidup = 0
    //set jenis kelamin sesuai-kalo cowok = 0

    if (penduduk.getIs_wni().equals("wni") || penduduk.getIs_wni().equals("wni")) {
        penduduk.setIs_wni("1");
    } else {
        penduduk.setIs_wni("0");
    }

    if (penduduk.getIs_wafat().equals("Hidup")) {
        penduduk.setIs_wafat("0");
    } else {
        penduduk.setIs_wafat("1");
    }

    String jenis_kelaminawal = penduduk.getJenis_kelamin();
    if (jenis_kelaminawal.equals("Pria")) {
        penduduk.setJenis_kelamin("0");
    } else {
        penduduk.setJenis_kelamin("1");
    }
    String jenis_kelamin = penduduk.getJenis_kelamin();

    //GENERATING NEW NIK
    boolean nikBerubah = false;
    PendudukModel pendudukDiDB = sidukDAO.selectPenduduk(nik_lama);
    penduduk.setNik_lama(nik_lama);
    String firstSixlama = pendudukDiDB.getNik().substring(0,6);
    String secondSixlama = pendudukDiDB.getNik().substring(6,12);
    String firstSix = firstSixlama;
    String secondSix = secondSixlama;

    long id_keluarga = penduduk.getId_keluarga();
    String idString = String.valueOf(id_keluarga);
    //cari id kelurahan sesuai id keluarga
    String kode_kelurahan = sidukDAO.selectKodeKelurahan(idString);
    String kodekelurahan1 = kode_kelurahan.substring(0, 6);

    long id_keluarga_DB = pendudukDiDB.getId_keluarga();
    String idStringDB = String.valueOf(id_keluarga_DB);
    String kode_kelurahan_DB = sidukDAO.selectKodeKelurahan(idStringDB);
    String kodekelurahan2 = kode_kelurahan_DB.substring(0, 6);

    boolean domisiliTetap = (kodekelurahan1.equals(kodekelurahan2));
    boolean tglTetap= penduduk.getTanggal_lahir().equals(pendudukDiDB.getTanggal_lahir());
    boolean jkTetap = penduduk.getJenis_kelamin().equals(pendudukDiDB.getJenis_kelamin());
    if (!domisiliTetap) {
        nikBerubah = true;
        firstSix = kodekelurahan1;
    }
    if (!tglTetap) {
        String tanggal_lahir = penduduk.getTanggal_lahir();
        String[] tanggal_lahir_split = tanggal_lahir.split("-");
        String tanggal = tanggal_lahir_split[2];
    }
}
```

May Iffah Rizki  
1506757661  
APAP-A

```
String bulan = tanggal_lahir_split[1];
String tahun = tanggal_lahir_split[0].substring(2);
StringBuilder tglAppend = new StringBuilder().append(tanggal).append(bulan).append(tahun);

String tanggalFix = tglAppend.toString();
secondSix = tanggalFix;
//jenis kelamin yg diinput
if (jenis_kelamin.equals("1")) {
    long tanggallahir = Long.parseLong(tanggal);
    tanggallahir = tanggallahir + 40;
    tanggal_lahir = String.valueOf(tanggallahir);
    StringBuilder tglBaru = new StringBuilder().append(tanggal_lahir).append(bulan).append(tahun);
    tanggalFix = tglBaru.toString();
}
secondSix = tanggalFix;
//tgl tetap sama
} else {
    String tanggal_lahir = secondSixlama.substring(0,2);
    String bulan = secondSixlama.substring(2,4);
    String tahun = secondSixlama.substring(4,6);
    //cek jenis kelamin pria atau wanita. jika wanita akan ditambahkan.
    if (!jkTetap) {
        long tanggallahir = Long.parseLong(tanggal_lahir);
        if (jenis_kelamin.equals("1")) {
            tanggallahir = tanggallahir + 40;
        } else {
            tanggallahir = tanggallahir - 40;
        }
        tanggal_lahir = String.valueOf(tanggallahir);
        StringBuilder tglBaru = new StringBuilder().append(tanggal_lahir).append(bulan).append(tahun);
        secondSix = tglBaru.toString();
    }
}
} else {
    if (!tglTetap) {
        nikBerubah = true;
        //generate tgl lahir
        //first six
        //secondsix =
        String tanggal_lahir = penduduk.getTanggal_lahir();
        String[] tanggal_lahir_split = tanggal_lahir.split("-");
        String tanggal = tanggal_lahir_split[2];
        String bulan = tanggal_lahir_split[1];
        String tahun = tanggal_lahir_split[0].substring(2);
        StringBuilder tglAppend = new StringBuilder().append(tanggal).append(bulan).append(tahun);

        String tanggalFix = tglAppend.toString();
        if (jenis_kelamin.equals("1")) {
            long tanggallahir = Long.parseLong(tanggal);
            tanggallahir = tanggallahir + 40;
            tanggal_lahir = String.valueOf(tanggallahir);
            StringBuilder tglBaru = new StringBuilder().append(tanggal_lahir).append(bulan).append(tahun);
            tanggalFix = tglBaru.toString();
        }
        secondSix = tanggalFix;
        //tgl tetap sama
    } else {
        String tanggal_lahir = secondSixlama.substring(0,2);
        String bulan = secondSixlama.substring(2,4);
        String tahun = secondSixlama.substring(4,6);
        //cek jenis kelamin pria atau wanita. jika wanita akan ditambahkan.

        if (!jkTetap) {
            nikBerubah = true;
            long tanggallahir = Long.parseLong(tanggal_lahir);
            if (jenis_kelamin.equals("1")) {
                tanggallahir = tanggallahir + 40;
            } else {
                tanggallahir = tanggallahir - 40;
            }
            tanggal_lahir = String.valueOf(tanggallahir);
            StringBuilder tglBaru = new StringBuilder().append(tanggal_lahir).append(bulan).append(tahun);
            secondSix = tglBaru.toString();
        }
    }
}
}
```

May Iffah Rizki

1506757661

APAP-A

```
//JIKA NIK BERUBAH, MAKA GENERATE NIK ULANG
if (nikBerubah == true) {
    String nomor_urut = "0001";
    int counter = 1;
    StringBuilder nikAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut);
    String nik = nikAppend.toString();
    //generate nik -> append semua + 0001
    //loop
    boolean masihAda = true;
    while (masihAda == true) {
        PendudukModel cekPenduduk = sidukDAO.selectPenduduk (nik);
        if (cekPenduduk != null) {
            counter++;
            int digit_counter = String.valueOf(counter).length();
            StringBuilder nomor_urut_baru = new StringBuilder();
            if (digit_counter == 1) {
                nomor_urut_baru = new StringBuilder().append("000").append(String.valueOf(counter));
            } else if (digit_counter == 2) {
                nomor_urut_baru = new StringBuilder().append("00").append(String.valueOf(counter));
            } else if (digit_counter == 3) {
                nomor_urut_baru = new StringBuilder().append("0").append(String.valueOf(counter));
            } else {
                nomor_urut_baru = new StringBuilder().append(String.valueOf(counter));
            }
            StringBuilder nikBaruAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut_baru.toString());
            String nikBaru = nikBaruAppend.toString();
            nik = nikBaru;
        } else {
            masihAda = false;
        }
    }
    penduduk.setNik(nik);
} else {
    penduduk.setNik(nik_lama);
}

sidukDAO.updatePenduduk(penduduk);

model.addAttribute("niklama", niklama);
return "sukses-penduduk-update";
}
```

## form-penduduk-ubah.html

Update Penduduk

×

New Tab

×

New Tab

×

localhost:8080/penduduk/ubah/3101011405170001

Apps Home | Bill Gates OpenStax College Student Centered e-L Top 30 OOPS Conce Dashboard | Khan Ac Perubahan nomor C Single Sign On - Inte Konsep Pemrograman

SIDUK Home Cari Penduduk Tambah Penduduk Tambah Keluarga

### Update Penduduk

Nama:

Tempat Lahir:

Tanggal Lahir:

Jenis Kelamin:

Kewarganegaraan:

Agama:

Pekerjaan:

Status Perkawinan:

Status dalam Keluarga:

Golongan Darah:

May Iffah Rizki  
1506757661  
APAP-A

Status Kematian:

Hidup ▾

ID Keluarga:

172

Simpan

Mengubah nama dan jenis kelamin menjadi wanita

Update Penduduk

Nama:

Heri Haryanti

Tempat Lahir:

Jakarta

Tanggal Lahir:

05/14/2017

Jenis Kelamin:

Wanita ▾

Kewarganegaraan:

WNI

Agama:

Islam

Pekerjaan

BELUM/TIDAK BEKERJA

Status Perkawinan:

Belum Kawin ▾

Status dalam Keluarga:

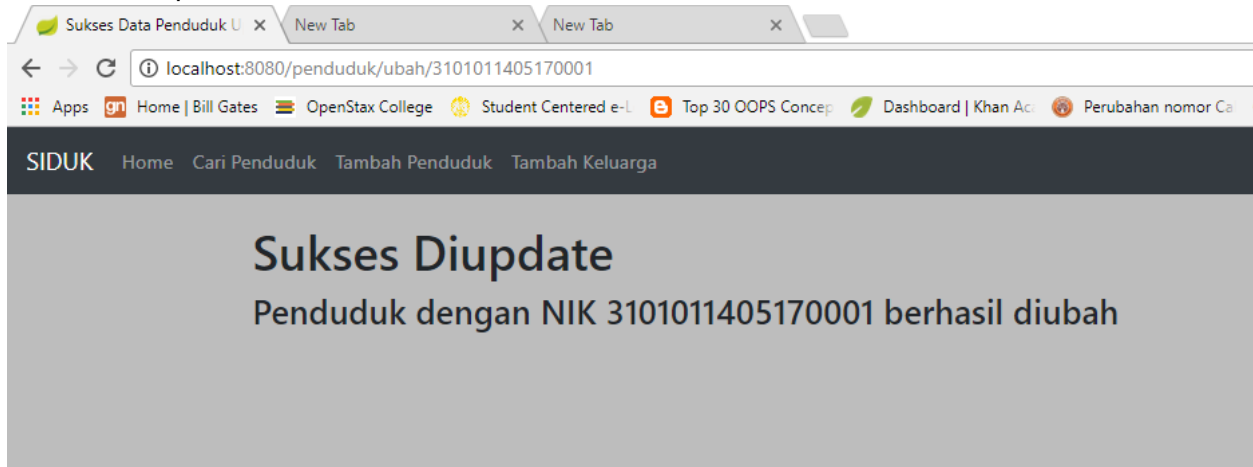
Anak

Golongan Darah:

A+ ▾

May Iffah Rizki  
1506757661  
APAP-A

sukses-data-penduduk-ubah.html



NIK baru:

id	nik	nama	tempat_lahir	tanggal_lahir	jenis_kelamin	is_wni	id_keluarga	agama	pekerjaan	status_perkawinan	statu
1	3101015405170003	Heri Haryanti	Jakarta	2017-05-14	1	1	172	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak

f. Mengubah Data Keluarga

Initial Page: /keluarga/ubah/{NKK}

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

Response Page: /keluarga/ubah/{NKK}

Fitur dimulai dengan pengisian form yang sudah terisi dengan data keluarga (sesuai dengan NKK yang dijadikan parameter dan keluarga sesuai dengan selectKeluarga) pada tiap input field di form-keluarga-ubah.html (mapping /keluarga/ubah/{NKK}) dan kemudian merefer ke sukses-data-keluarga-ubah.html jika sukses diubah (mapping /keluarga/ubah/{NKK}). Data keluarga yang dimasukkan pada awalnya sudah disesuaikan dari database sebelum muncul di halaman html.

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

    model.addAttribute("keluarga", keluarga);
    return "form-keluarga-update";
}
```

Method yang dibuat adalah updateKeluarga. Hasil input di-POST dan dimasukkan ke method untuk tambah ke tabel keluarga, untuk generate NKK, diperlukan kode kelurahan yang diambil dari method selectKodeKelurahan untuk membuat 6 digit pertama, serta selectIDKelurahan untuk mencari kelurahan sesuai nama kelurahan yang diinput (dengan asumsi; nama kelurahan, kecamatan, dan kota sudah sesuai dengan yang ada di database). Kelurahan yang didapatkan dapat dicari kodenya dengan selectKodeKelurahan. Mencari tanggal saat ini digunakan untuk 6 digit kedua. Method countKeluargaRow untuk menentukan id keluarga baru (menambah 1 dari jumlah row table keluarga). NKK final adalah 16 digit dengan 4 digit terakhir urutan kesamaan 12 digit awal NKK dengan keluarga yang sudah ada di database.

```
@RequestMapping(value = "/keluarga/ubah/{nomor_kk_lama}", method = RequestMethod.POST)
public String updatePendudukSubmit (@ModelAttribute KeluargaModel keluarga, Model model, @PathVariable(value = "nomor_kk_lama") String nomor_kk_lama)
{
    String nkklama= nomor_kk_lama;

    //GENERATING NEW NKK

    KeluargaModel keluargaDiDB = sidukDAO.selectKeluarga(nomor_kk_lama);

    keluarga.setNomor_kk_lama(nomor_kk_lama);

    String firstSixlama = keluargaDiDB.getNomor_kk().substring(0,6);
    String firstSix = firstSixlama;

    //tanggal submit adalah tanggal sekarang
    DateFormat df = new SimpleDateFormat("dd/MM/yy");
    Date dateobj = new Date();
    String tanggal_submit = df.format(dateobj);

    String[] tanggal_split = tanggal_submit.split("/");
    String tanggal = tanggal_split[0];
    String bulan = tanggal_split[1];
    String tahun = tanggal_split[2];
    StringBuilder tglAppend = new StringBuilder().append(tanggal).append(bulan).append(tahun);
    String tanggalFix = tglAppend.toString();

    String secondSix = tanggalFix;

    String nama_kelurahan = keluarga.getKelurahan();
    //cari id kelurahan sesuai nama kelurahan
    String id_kelurahan = sidukDAO.selectIDKelurahan(nama_kelurahan);
    keluarga.setId_kelurahan(id_kelurahan);
    long id_kelurahan_long = Long.parseLong(id_kelurahan);
    KelurahanModel tes = sidukDAO.selectKelurahanByID(id_kelurahan_long);
    String kode_kelurahan = tes.getKode_kelurahan();
    String kodekelurahan1 = kode_kelurahan.substring(0, 6);

    long id_kelurahan_lama_long = Long.parseLong(keluargaDiDB.getId_kelurahan());
    KelurahanModel tes2 = sidukDAO.selectKelurahanByID(id_kelurahan_lama_long);
    String kode_kelurahan_lama = tes2.getKode_kelurahan();
    String kodekelurahan2 = kode_kelurahan_lama.substring(0, 6);
    boolean domisiliTetap = (kodekelurahan1.equals(kodekelurahan2));

    if (!domisiliTetap) {
        firstSix = kodekelurahan1;
    }

    //GENERATE NKK ULANG
    String nomor_urut = "0001";
    int counter = 1;
    StringBuilder nkkAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut);
    String nkk = nkkAppend.toString();
    //generate nkk -> append semua + 0001
    //loop
    boolean masihAda = true;
    while (masihAda == true) {
        KeluargaModel cekKeluarga = sidukDAO.selectKeluarga (nkk);
        if (cekKeluarga != null) {
            counter+=1;
            int digit_counter = String.valueOf(counter).length();
            StringBuilder nomor_urut_baru = new StringBuilder();
            if (digit_counter == 1) {
                nomor_urut_baru = new StringBuilder().append("000").append(String.valueOf(counter));
            } else if (digit_counter == 2) {
                nomor_urut_baru = new StringBuilder().append("00").append(String.valueOf(counter));
            } else if (digit_counter == 3) {
                nomor_urut_baru = new StringBuilder().append("0").append(String.valueOf(counter));
            } else {
                nomor_urut_baru = new StringBuilder().append(String.valueOf(counter));
            }
            StringBuilder nkkBaruAppend = new StringBuilder().append(firstSix).append(secondSix).append(nomor_urut_baru.toString());
            String nkkBaru = nkkBaruAppend.toString();
            nkk = nkkBaru;
        } else {
            masihAda = false;
        }
    }
}

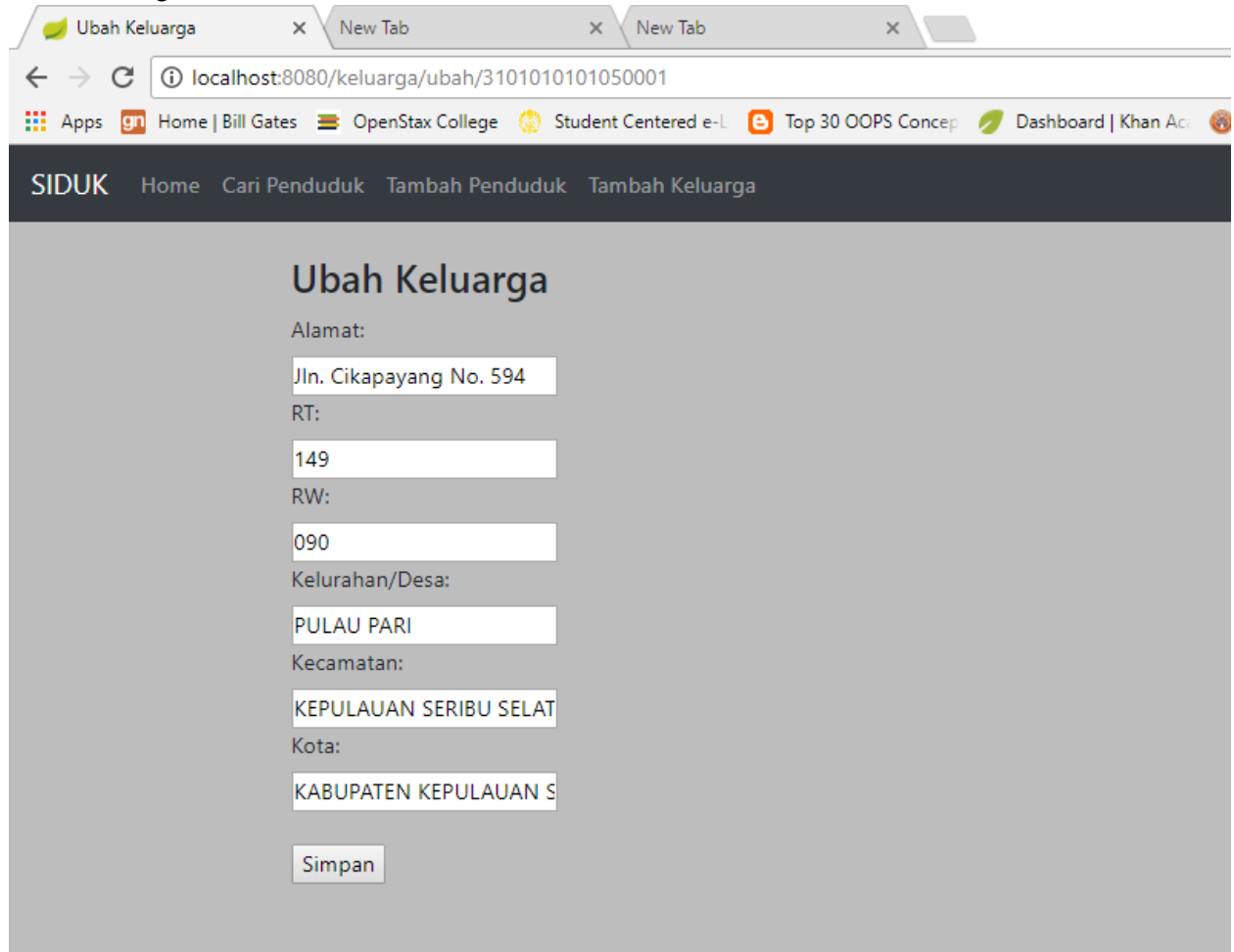
keluarga.setNomor_kk(nkk);
keluarga.setIs_tidak_berlaku("0");
sidukDAO.updateKeluarga(keluarga);

model.addAttribute("nkklama", nkklama);
return "sukses-keluarga-update";
}
```



May Iffah Rizki  
1506757661  
APAP-A

form-keluarga-ubah.html



The screenshot shows a web browser with three tabs: 'Ubah Keluarga', 'New Tab', and 'New Tab'. The address bar displays 'localhost:8080/keluarga/ubah/3101010101050001'. The browser's toolbar includes icons for Apps, Home, Bill Gates, OpenStax College, Student Centered e-L, Top 30 OOPS Concep, Dashboard, and Khan Ac.

The web page has a dark header with the text 'SIDUK' and navigation links: 'Home', 'Cari Penduduk', 'Tambah Penduduk', and 'Tambah Keluarga'.

The main content area is titled 'Ubah Keluarga' and contains the following form fields:

- Alamat:
- RT:
- RW:
- Kelurahan/Desa:
- Kecamatan:
- Kota:

At the bottom of the form is a 'Simpan' (Save) button.

May Iffah Rizki  
1506757661  
APAP-A

### Mengganti RT dan RW keluarga

Ubah Keluarga

Alamat:  
Jln. Cikapayang No. 594

RT:  
100

RW:  
200

Kelurahan/Desa:  
PULAU PARI

Kecamatan:  
KEPULAUAN SERIBU SELAT

Kota:  
KABUPATEN KEPULAUAN S

Simpan

### sukses-data-keluarga-ubah.html

Sukses Data Keluarga Up

Sukses Diupdate  
Keluarga dengan NKK 3101010101050001 berhasil diubah

NKK dan RT, RW baru:

id	nomor_kk	alamat	RT	RW	id_kelurahan	is_tidak_berlaku
1	3101012210170001	Jln. Cikapayang No. 594	100	200	2	0

- g. Mengubah Status Kematian Penduduk  
Initial Page: /penduduk?nik={NIK}  
Form Request: POST, /penduduk/mati  
Response Page: /penduduk?nik={NIK}

Fitur dimulai dengan melihat data penduduk di data-penduduk.html (mapping /penduduk?={NIK})

```
boolean sudahNonAktif = false;

@RequestMapping("/penduduk")
public String penduduk (@RequestParam(value = "nik", required = false, defaultValue = "tes") String nik, Model model)
{
    if (sudahNonAktif == true) {
        model.addAttribute("nik", nik);
        sudahNonAktif = false;
        return "sukses-penduduk-nonaktif";
    } else {
        PendudukModel penduduk = sidukDAO.selectPenduduk(nik);
        penduduk.changeDefault(nik);
        model.addAttribute("penduduk", penduduk);
        return "data-penduduk";
    }
}
```

dan kemudian merefer ke sukses-penduduk-nonaktif.html jika sukses dinonaktifkan (mapping /penduduk?={NIK}).

```
@RequestMapping(value = "/penduduk/mati", method = RequestMethod.POST)
public String nonAktif (@ModelAttribute PendudukModel penduduk)
{
    String nik = penduduk.getNik();
    long idKeluarga = penduduk.getId_keluarga();
    String nkk = sidukDAO.selectKeluargaNKKByID(idKeluarga);
    sidukDAO.nonAktifkan(nik);
    sudahNonAktif = true;
    int jumlahKeluargaAktif = 0;
    List<PendudukModel> anggotaKeluarga = sidukDAO.selectAnggotaKeluarga(nkk);
    int jumlahKeluarga = anggotaKeluarga.size();
    for (int i = 0; i < jumlahKeluarga; i++) {
        if (anggotaKeluarga.get(i).getIs_wafat().equals("0")) {
            jumlahKeluargaAktif++;
        }
    }
    if (jumlahKeluarga != 0 && jumlahKeluargaAktif == 0) {
        sidukDAO.nonAktifkanKeluarga(nkk);
    }

    return "redirect:/penduduk?nik=" + nik;
}
```

Method yang dibuat adalah nonAktifkan, nonAktifkanKeluarga, dan selectKeluargaNKKByID. Hasil submit tombol di-POST dan dimasukkan ke method untuk menonaktifkan penduduk sesuai dengan NIK dengan nonAktifkan. Untuk menonaktifkan keluarga dengan nonAktifkanKeluarga, diperlukan jumlah anggota keluarga dari keluarga penduduk. Diperlukan NKK untuk melihat anggota keluarga, sehingga perlu method selectKeluargaNKKByID dari id

keluarga milik penduduk. Jika keluarga memiliki anggota dan semua anggotanya mati, maka keluarga juga dinonaktifkan. Proses nonaktif ini dilakukan di mapping (/penduduk/mati) yang kemudian langsung meredirect ke (/penduduk?nik={NIK}).

```
boolean sudahNonAktif = false;

@RequestMapping("/penduduk")
public String penduduk (@RequestParam(value = "nik", required = false, defaultValue = "tes") String nik, Model model)
{
    if (sudahNonAktif == true) {
        model.addAttribute("nik", nik);
        sudahNonAktif = false;
        return "sukses-penduduk-nonaktif";
    } else {
        PendudukModel penduduk = sidukDAO.selectPenduduk(nik);
        penduduk.changeDefault(nik);
        model.addAttribute("penduduk", penduduk);
        return "data-penduduk";
    }
}
```

data-penduduk.html

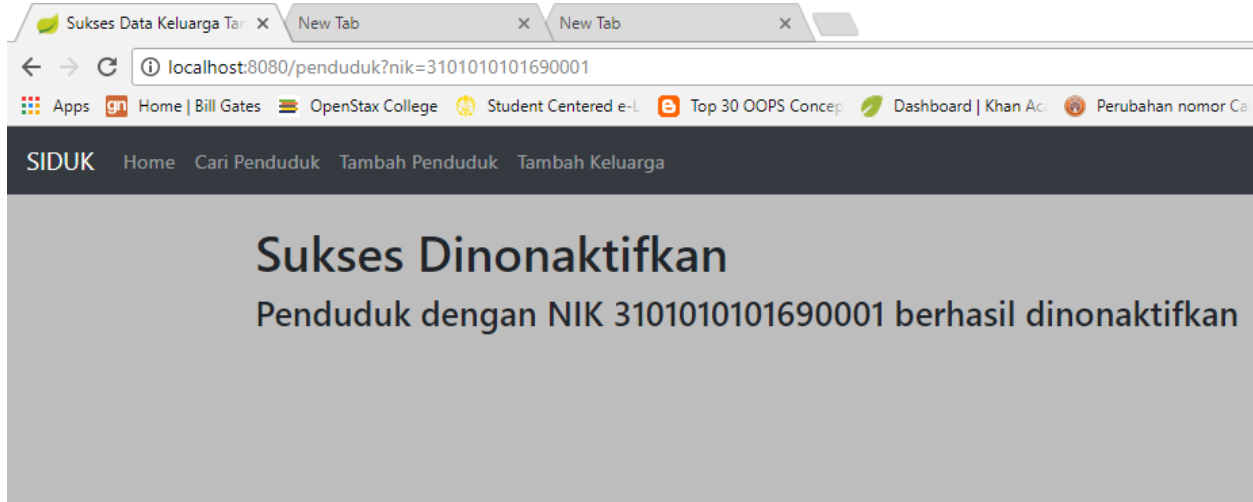
The screenshot shows a web browser with the URL `localhost:8080/penduduk?nik=3101010101690001`. The page title is "Lihat Data Penduduk - 3101010101690001". The page content includes a table with the following data:

NIK	3101010101690001
Nama	Ridwan Hasim Marpaung
Jenis Kelamin	Pria
Tempat/Tanggal Lahir	Atambua, 1969-01-01
Alamat	Kpg. Cihampelas No. 220
RT/RW	177/056
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	B-
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	PEMBANTU RUMAH TANGGA
Kewarganegaraan	WNI
Status Kematian	Hidup

Below the table is a button labeled "Nonaktifkan".

May Iffah Rizki  
1506757661  
APAP-A

sukses-penduduk-nonaktif.html



Status kematian di database:

id	nik	nama	tempat_lahir	tanggal_lahir	jenis_kelamin	is_wni	id_keluarga	agama	pekerjaan	status_perkawinan		
<input type="checkbox"/>	Edit  Copy  Delete	2	3101010101690001	Ridwan Hasim Marpaung	Atambua	1969-01-01	0	1	261	Islam	PEMBANTU RUMAH TANGGA	Kawin

nama	tempat_lahir	tanggal_lahir	jenis_kelamin	is_wni	id_keluarga	agama	pekerjaan	status_perkawinan	status_dalam_keluarga	golongan_darah	is_wafat
Ridwan Hasim Marpaung	Atambua	1969-01-01	0	1	261	Islam	PEMBANTU RUMAH TANGGA	Kawin	Pembantu	B-	1

- h. 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}

```
long idKt = 0;
long idKc = 0;
long idKl = 0;

@RequestMapping("/penduduk/cari")
public String cariPenduduk (Model modelKota, @RequestParam(value = "kt", required = false) String kt,
    Model modelKec, Model modelAda, @RequestParam(value = "kc", required = false) String kc,
    Model modelKel, Model modelAda2,
    Model modelAdaSemua, @RequestParam(value = "kl", required = false) String kl,
    Model modelPenduduk,
    Model modelJudul)
{
    boolean adaSemua = false;
```

Fitur menggunakan method `selectAllKota` untuk memilih semua kota, `selectKecamatanByIDKota`, `selectKelurahanByIDKecamatan`, `selectPendudukByDomisili`. Fitur ini menggunakan `thymeleaf if unless operation` untuk menentukan dropdown mana yang harus ditampilkan atau tidak, dropdown mana yang harus `readonly` optionnya atau tidak.

1. Memilih Kota/Kabupaten pada halaman "/penduduk/cari"

Pada halaman ini terdapat dropdown Kota/Kabupaten di DKI Jakarta dan tombol submit. Pilih salah satu Kota/Kabupaten kemudian klik submit. Request akan dikirim.

```
if (kt == null) {
    List<KotaModel> resultKota = sidukDAO.selectAllKota();
    modelKota.addAttribute("kota", resultKota);
    modelJudul.addAttribute("judul", "Lihat Data Penduduk Berdasarkan Kota");
```

2. Aplikasi mengembalikan halaman "/penduduk/cari?kt={ID\_KOTA}"

Memilih Kecamatan pada halaman "/penduduk/cari?kt={ID\_KOTA}"

Pada halaman ini terdapat dropdown Kecamatan di Kota/Kabupaten yang telah di pilih sebelumnya, dropdown Kota/Kabupaten yang `readonly` (tidak bisa diubah), dan tombol submit. Pilih salah satu Kecamatan kemudian klik submit. Request akan dikirim.

```
else {  
    long idKota = Long.parseLong(kt);  
    idKt = idKota;  
    KotaModel kota = sidukDAO.selectKotaByID(idKota);  
    modelKota.addAttribute("kota", kota);  
    modelJudul.addAttribute("judul", "Lihat Data Penduduk Berdasarkan Kecamatan di " + kota.getNama_kota());  
    if (kc == null) {  
        List<KecamatanModel> resultKec = sidukDAO.selectKecamatanByIDKota(idKota);  
        modelAda.addAttribute("blmAdaKelAdaKota", true);  
        modelKec.addAttribute("kecamatan", resultKec);  
    }
```

3. Aplikasi mengembalikan halaman “/penduduk/cari?kt={ID\_KOTA}&kc={ID\_KEC}”. Memilih Kelurahan pada halaman. Pada halaman ini terdapat dropdown Kelurahan di Kecamatan yang telah dipilih sebelumnya, dropdown Kota/Kabupaten yang disabled, dropdown Kecamatan yang read-only, dan tombol submit. Pilih salah satu Kelurahan kemudian klik submit. Request akan dikirim.

```
} else {  
    long idKec = Long.parseLong(kc);  
    idKc = idKec;  
    KecamatanModel kec = sidukDAO.selectKecamatanByID(idKec);  
    //bakal ada kelurahan setelah ini  
    modelAda.addAttribute("blmAdaKelAdaKota", false);  
    modelKec.addAttribute("kecamatan", kec);  
    modelJudul.addAttribute("judul", "Lihat Data Penduduk Berdasarkan Kelurahan di " + kota.getNama_kota() + ", KECAMATAN " + kec.getNama_kecamatan());  
    if (kl == null) {  
        List<KelurahanModel> resultKel = sidukDAO.selectKelurahanByIDKecamatan(idKec);  
        modelAda2.addAttribute("blmAdaKelAdaKotaAdaKec", true);  
        modelKel.addAttribute("kelurahan", resultKel);  
    }
```

4. Aplikasi lalu mengembalikan halaman “/penduduk/cari?kt={ID\_KOTA}&kc={ID\_KEC}&kl={ID\_KEL}” dan menampilkan semua penduduk yang ada pada Kelurahan yang dipilih. Pada halaman ini menampilkan semua penduduk yang terdaftar pada kelurahan yang dipilih, baik yang masih hidup atau mati.

```
} else {  
    long idKel = Long.parseLong(kl);  
    idKl = idKel;  
    KelurahanModel kel = sidukDAO.selectKelurahanByID(idKel);  
    //bakal ada kelurahan setelah ini  
    modelAda2.addAttribute("blmAdaKelAdaKotaAdaKec", false);  
    modelAdaSemua.addAttribute("adaSemua", true);  
    modelKel.addAttribute("kelurahan", kel);  
    modelJudul.addAttribute("judul", "Lihat Data Penduduk Berdasarkan Kelurahan di " + kota.getNama_kota() + ", KECAMATAN " + kec.getNama_kecamatan() + ", KELURAHAN " + kel.getNama_kelurahan());  
    adaSemua = true;  
}  
  
}  
  
if (adaSemua) {  
    List<PendudukModel> penduduk = sidukDAO.selectPendudukByDomisili(idKt, idKc, idKl);  
    modelPenduduk.addAttribute("penduduk", penduduk);  
}  
return "cari-penduduk";  
}
```

May Iffah Rizki  
1506757661  
APAP-A

cari-penduduk.html

SIDUK Home Cari Penduduk Tambah Penduduk Tambah Keluarga

## Lihat Data Penduduk Berdasarkan Kota

Kota:

Pilih Kota ▼

Cari

Memilih kota Jakarta Selatan

SIDUK Home Cari Penduduk Tambah Penduduk Tambah Keluarga

## Lihat Data Penduduk Berdasarkan Kecamatan di KOTA JAKARTA SELATAN

Kota:

KOTA JAKARTA SELATAN ▼

Kecamatan:

Pilih Kecamatan ▼

Cari



May Iffah Rizki  
1506757661  
APAP-A

## Memilih kecamatan Tebet

The screenshot shows a web browser window with three tabs: 'Cari Penduduk', 'New Tab', and 'New Tab'. The address bar displays 'localhost:8080/penduduk/cari?kt=28&kc=11'. The browser's bookmark bar includes links to 'Apps', 'Home | Bill Gates', 'OpenStax College', 'Student Centered e-L', 'Top 30 OOPS Concep', 'Dashboard | Khan Ac', 'Perubahan nomor C', 'Single Sign On - Inter', and 'Konsep Pemrograman'. The application's navigation bar features the 'SIDUK' logo and links to 'Home', 'Cari Penduduk', 'Tambah Penduduk', and 'Tambah Keluarga'. The main content area has a title 'Lihat Data Penduduk Berdasarkan Kelurahan di KOTA JAKARTA SELATAN, KECAMATAN TEBET'. Below the title, there are four dropdown menus for filtering: 'Kota:' (set to 'KOTA JAKARTA SELATAN'), 'Kecamatan:' (set to 'TEBET'), 'Kelurahan:' (set to 'Pilih Kelurahan'), and a 'Cari' button.

**Lihat Data Penduduk Berdasarkan Kelurahan di KOTA JAKARTA SELATAN, KECAMATAN TEBET**

Kota:  
KOTA JAKARTA SELATAN ▼

Kecamatan:  
TEBET ▼

Kelurahan:  
Pilih Kelurahan ▼

Cari

May Iffah Rizki  
1506757661  
APAP-A

## Memilih Kelurahan Menteng Dalam

Cari Penduduk

New Tab

New Tab

localhost:8080/penduduk/cari?kt=2&kc=11&kj=57

Apps Home | Bill Gates OpenStax College Student Centered e-L Top 30 OOPS Concep Dashboard | Khan Ac Perubahan nomor Ca Single Sign On - Inte Konsep Pem

SIDUK Home Cari Penduduk Tambah Penduduk Tambah Keluarga

Lihat Data Penduduk Berdasarkan Kelurahan di KOTA JAKARTA SELATAN, KECAMATAN TEBET, KELURAHAN MENTENG DALAM

Kota:

KOTA JAKARTA SELATAN

Kecamatan:

TEBET

Kelurahan:

MENTENG DALAM

Cari

Show 10 entries

Search:

NIK	Nama	Jenis Kelamin
3171090101640001	Danuja Vega Marbun	Pria
3171090101930001	Panji Prabowo	Pria
3171090102130003	Bagas Hardiansyah	Pria
3171090103120001	Lantar Firmansyah	Pria
3171090103160002	Laswi Mahendra	Pria
3171090103800001	Ilyas Nyana Marpaung	Pria
3171090103920001	Koko Suryono	Pria
3171090104820001	Pangestu Gilang Sihotang	Pria
3171090104870001	Asmianto Slamet Hutapea	Pria
3171090106860001	Jaeman Prakasa S.Farm	Pria

Showing 1 to 10 of 796 entries

Previous 1 2 3 4 5 ... 80 Next

- i. [Bonus] Validasi Input Untuk Semua Form POST
- j. [Bonus] Menampilkan Penduduk Paling Muda dan Paling Tua di Suatu Kelurahan
  - Menampilkan penduduk paling muda dan tua di kelurahan pada fitur 8, berdasarkan tanggal lahir penduduk.

```
if (adaSemua) {
    List<PendudukModel> penduduk = (ArrayList<PendudukModel>) sidukDAO.selectPendudukByDomisili(idKt, idKc, idKl);
    modelPenduduk.addAttribute("penduduk", penduduk);
    PendudukModel random1 = penduduk.get(0);
    PendudukModel random2 = penduduk.get(penduduk.size() - 1);

    PendudukModel tertua = random1;
    PendudukModel termuda = random2;
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    for (int i = 0; i < penduduk.size(); i++) {
        Date tua = sdf.parse(tertua.getTanggal_lahir());
        Date muda = sdf.parse(termuda.getTanggal_lahir());
        PendudukModel indexmodel = penduduk.get(i);
        Date index = sdf.parse(indexmodel.getTanggal_lahir());

        //jika index before tertua, tertua = index
        if (index.before(tua)) {
            tertua = indexmodel;
        }
        //jika index after termuda, termuda = index
        if (index.after(muda)) {
            termuda = indexmodel;
        }
    }
    modelPenduduk.addAttribute("termuda", termuda);
    modelPenduduk.addAttribute("tertua", tertua);
}
```

Cari Penduduk

localhost:8080/penduduk/cari?kt=2&kc=11&kl=57

Apps Home | Bill Gates OpenStax College Student Centered e-L Top 30 OOPS Concept Dashboard | Khan Ac Perubahan nomor C Single Sign On - Inter Konse

SIDUK Home Cari Penduduk Tambah Penduduk Tambah Keluarga

Lihat Data Penduduk Berdasarkan Kelurahan di KOTA JAKARTA SELATAN, KECAMATAN TEBET, KELURAHAN MENTENG DALAM

Kota:  
KOTA JAKARTA SELATAN  
Kecamatan:  
TEBET  
Kelurahan:  
MENTENG DALAM

Cari

Penduduk dengan Usia Tertua

NIK 3171092901530001  
Nama Martaka Antanto Suryono S.P  
Tanggal Lahir 1953-01-28

Penduduk dengan Usia Termuda

NIK 3171091609170001  
Nama Martaka Cakrawangsa Wacana S.P  
Tanggal Lahir 2017-09-16

Showing 10 entries

Search:

NIK	Nama	Jenis Kelamin
3171090101640001	Danuja Vega Marbun	Prta
3171090101930001	Parji Prabowo	Prta
3171090102130003	Bagas Hardiansyah	Prta
3171090103120001	Lantar Firmansyah	Prta
3171090103160002	Lassei Mahendra	Prta
3171090103800001	Ilyas Nyana Marpaung	Prta
3171090103920001	Koko Suryono	Prta
3171090104820001	Pangestu Gilang Sihotang	Prta
3171090104870001	Asmianto Slamet Hutapea	Prta
3171090106860001	Jaeman Prakasa S.Farm	Prta

Showing 1 to 10 of 796 entries Previous 1 2 3 4 5 ... 80 Next

## k. [Bonus] Menambahkan Error Page

- Menampilkan error-page jika parameter tidak sesuai dengan yang diminta.

```
package com.example.demo.controller;
import org.springframework.boot.autoconfigure.web.ErrorController;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class CustomErrorController implements ErrorController {

    private static final String PATH = "/error";

    @RequestMapping(PATH)
    public String error ()
    {
        return "error";
    }

    @Override
    public String getErrorPath() {
        return PATH;
    }
}

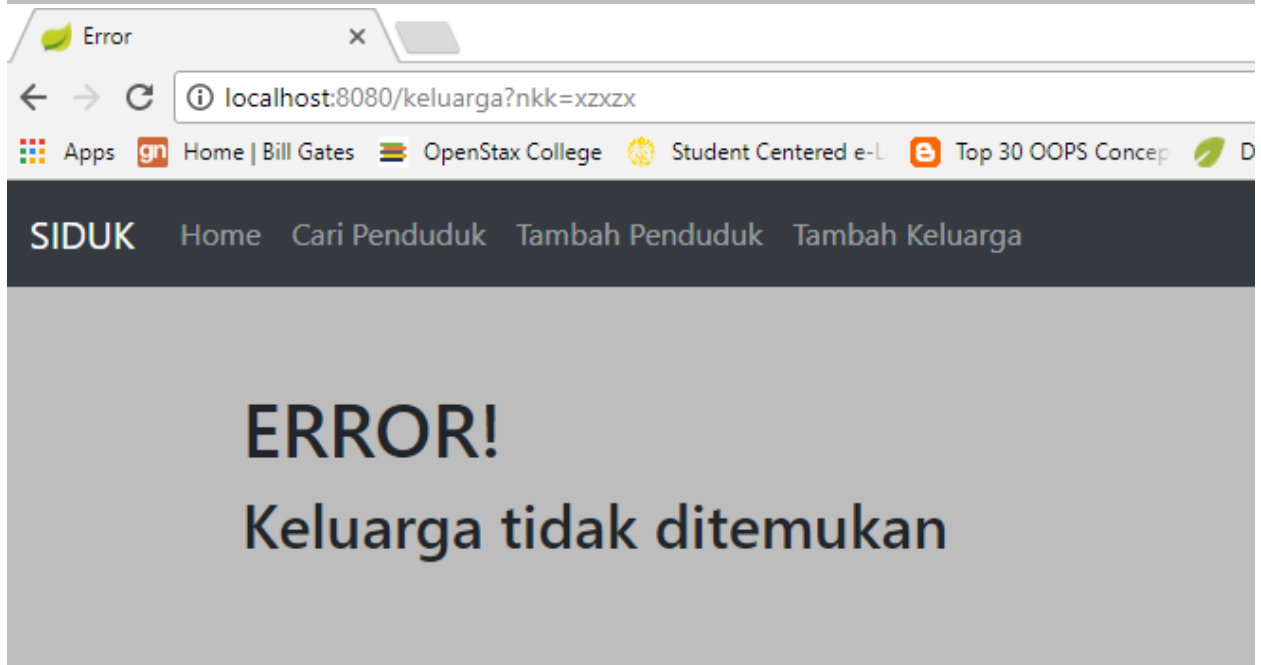
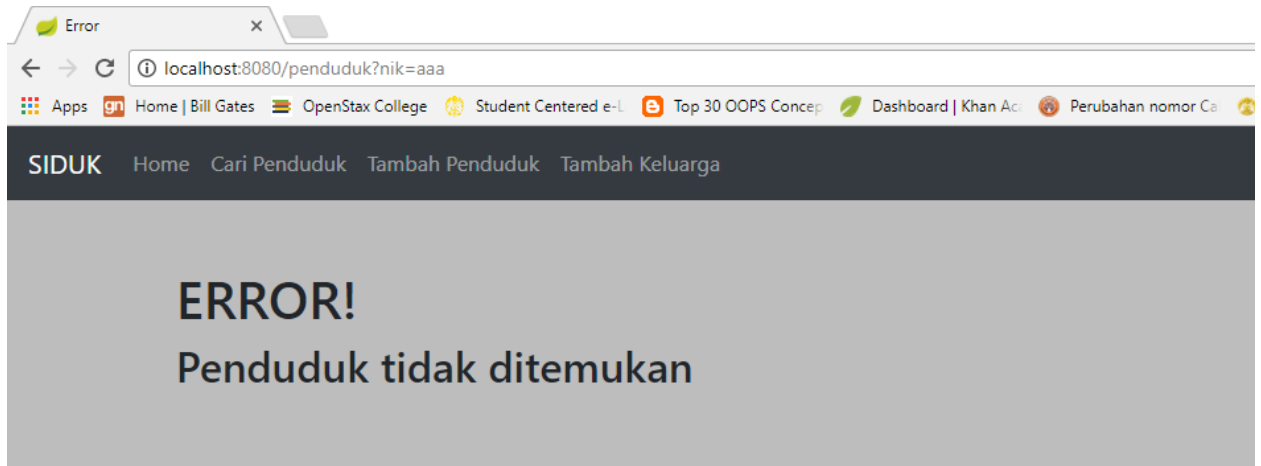
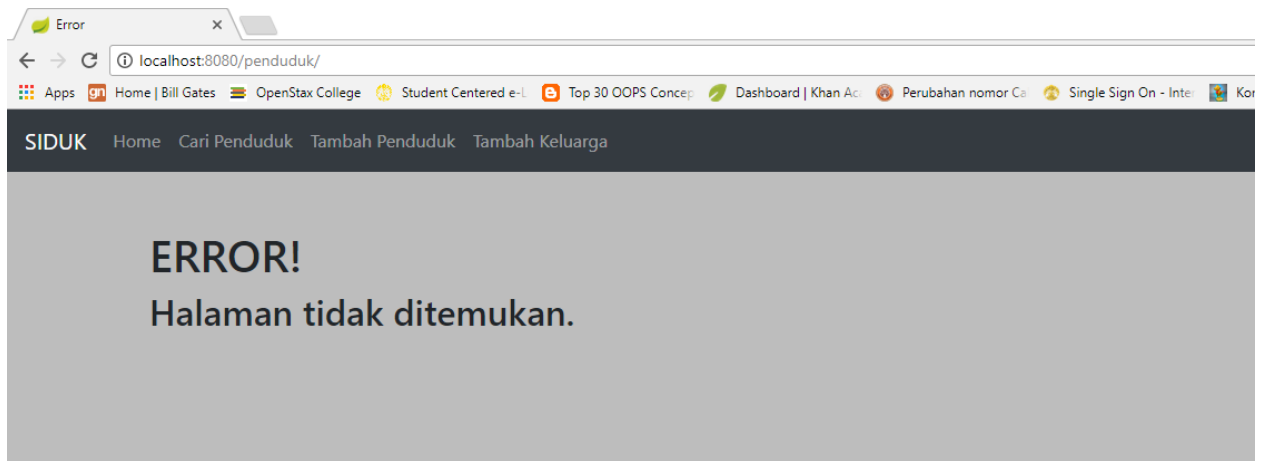
if (nik.equals("tes")) {
    model.addAttribute("message", "NIK tidak ada");
    return "error";
}
PendudukModel penduduk = sidukDAO.selectPenduduk(nik);

if (penduduk == null) {
    model.addAttribute("message", "Penduduk tidak ditemukan");
    return "error";
}

if (nkk.equals("tes")) {
    model.addAttribute("message", "NKK tidak ada");
    return "error";
}
KeluargaModel keluarga = sidukDAO.selectKeluarga(nkk);

if (keluarga == null) {
    model.addAttribute("message", "Keluarga tidak ditemukan");
    return "error";
}
```

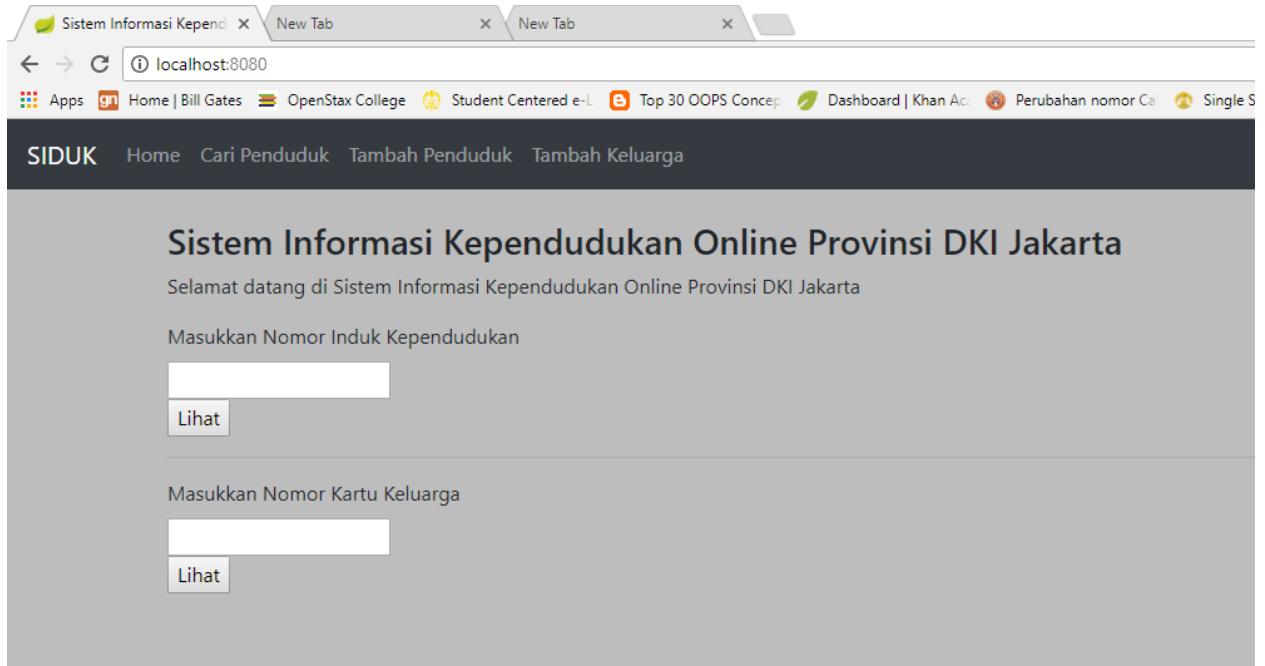
May Iffah Rizki  
1506757661  
APAP-A



May Iffah Rizki  
1506757661  
APAP-A

I. [Bonus] Fitur-Fitur Lain yang Mendukung Aplikasi

- Membuat navigasi bar pada setiap halaman memudahkan redirect ke halaman lain.



3. Optimasi Database

Optimasi Database dilakukan dengan penambahan primary key pada ID di setiap table, serta menambah index pada NIK di table penduduk dan NKK di table keluarga.

May Iffah Rizki

1506757661

APAP-A

#### 4. Stress Testing

Number of Threads (Users) = 100

Ramp-up Period = 5

Loop Count = 1

Berdasarkan dari hasil stress testing,

#### 1. Halaman pada / (home.html) (cepat) dan (reliable)

The image displays two screenshots of the Apache JMeter 3.3.1 interface. The top screenshot shows the configuration for an HTTP Request, and the bottom screenshot shows the results of a stress test.

**HTTP Request Configuration:**

- Name: HTTP Request
- Web Server: Protocol (http), Server Name or IP: localhost, Port Number: 8080
- Method: GET, Path: (empty)
- Content encoding: (empty)
- Parameters: (empty table)

**View Results in Table:**

Write results to file / Read from file: (empty)

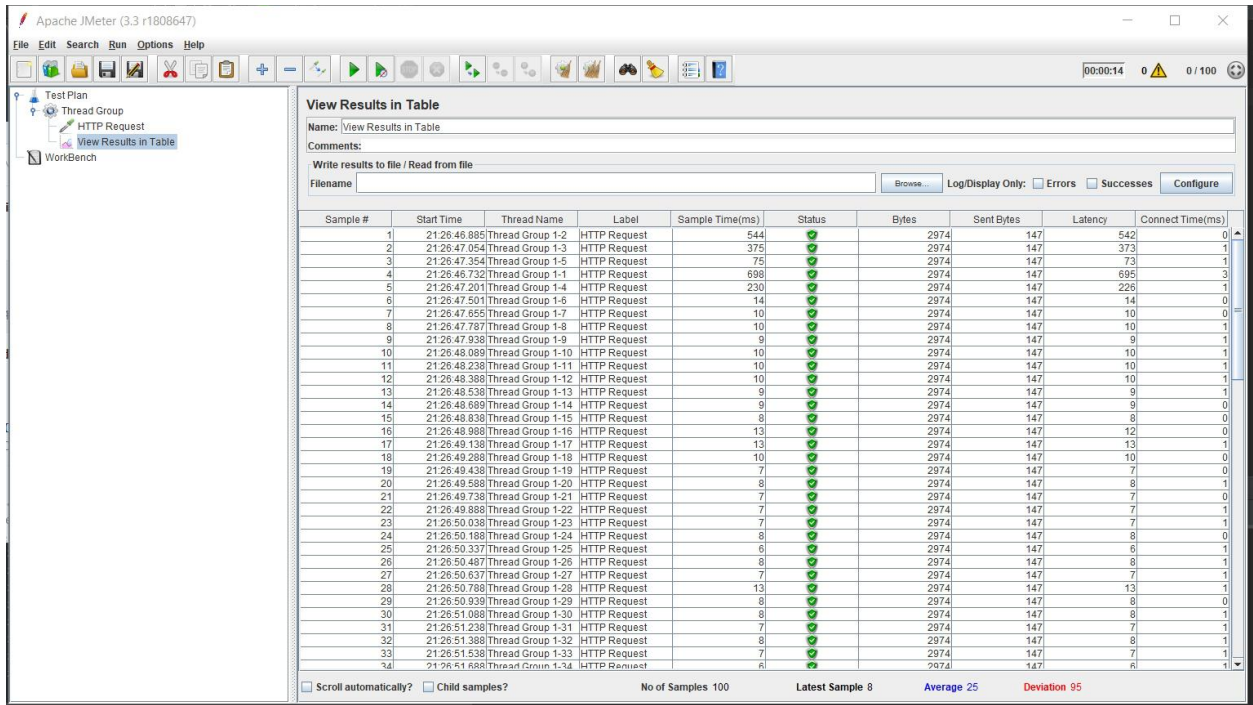
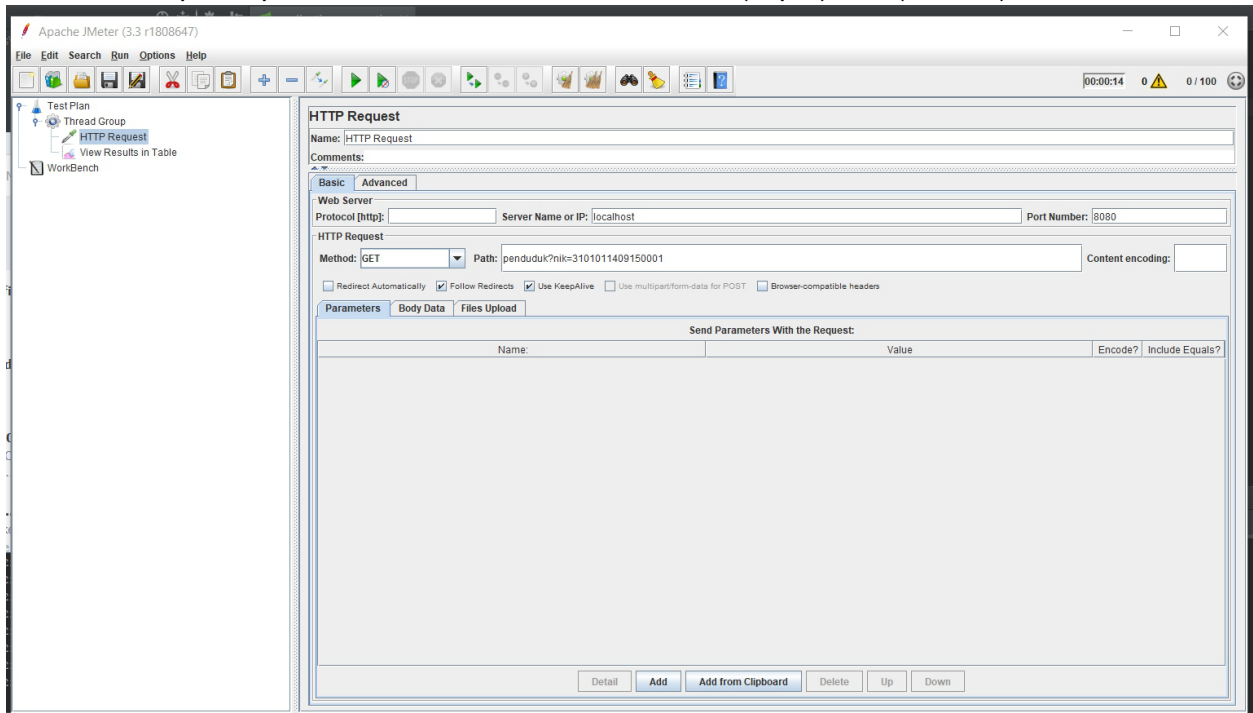
Log/Display Only: ☐ Errors ☐ Successes ☐ Configuration

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	21:17:00.599	Thread Group 1-3	HTTP Request	181	✓	2160	118	142	195
2	21:17:00.571	Thread Group 1-2	HTTP Request	209	✓	2160	118	170	134
3	21:17:00.769	Thread Group 1-4	HTTP Request	11	✓	2160	118	5	1
4	21:17:00.571	Thread Group 1-1	HTTP Request	209	✓	2160	118	170	134
5	21:17:00.901	Thread Group 1-5	HTTP Request	6	✓	2160	118	6	1
6	21:17:01.070	Thread Group 1-6	HTTP Request	6	✓	2160	118	6	1
7	21:17:01.201	Thread Group 1-7	HTTP Request	5	✓	2160	118	5	1
8	21:17:01.349	Thread Group 1-8	HTTP Request	5	✓	2160	118	5	1
9	21:17:01.499	Thread Group 1-9	HTTP Request	5	✓	2160	118	5	1
10	21:17:01.650	Thread Group 1-10	HTTP Request	6	✓	2160	118	6	1
11	21:17:01.800	Thread Group 1-11	HTTP Request	9	✓	2160	118	9	1
12	21:17:01.949	Thread Group 1-12	HTTP Request	5	✓	2160	118	5	1
13	21:17:02.100	Thread Group 1-13	HTTP Request	5	✓	2160	118	5	1
14	21:17:02.250	Thread Group 1-14	HTTP Request	5	✓	2160	118	5	1
15	21:17:02.399	Thread Group 1-15	HTTP Request	5	✓	2160	118	5	0
16	21:17:02.550	Thread Group 1-16	HTTP Request	5	✓	2160	118	5	1
17	21:17:02.699	Thread Group 1-17	HTTP Request	6	✓	2160	118	6	1
18	21:17:02.849	Thread Group 1-18	HTTP Request	4	✓	2160	118	4	0
19	21:17:02.999	Thread Group 1-19	HTTP Request	4	✓	2160	118	4	0
20	21:17:03.149	Thread Group 1-20	HTTP Request	5	✓	2160	118	5	1
21	21:17:03.300	Thread Group 1-21	HTTP Request	6	✓	2160	118	6	1
22	21:17:03.450	Thread Group 1-22	HTTP Request	3	✓	2160	118	3	0
23	21:17:03.599	Thread Group 1-23	HTTP Request	4	✓	2160	118	4	1
24	21:17:03.749	Thread Group 1-24	HTTP Request	5	✓	2160	118	5	1
25	21:17:03.900	Thread Group 1-25	HTTP Request	4	✓	2160	118	4	0
26	21:17:04.050	Thread Group 1-26	HTTP Request	4	✓	2160	118	4	0
27	21:17:04.199	Thread Group 1-27	HTTP Request	5	✓	2160	118	5	1
28	21:17:04.348	Thread Group 1-28	HTTP Request	6	✓	2160	118	6	1
29	21:17:04.499	Thread Group 1-29	HTTP Request	5	✓	2160	118	5	0
30	21:17:04.649	Thread Group 1-30	HTTP Request	5	✓	2160	118	4	0
31	21:17:04.798	Thread Group 1-31	HTTP Request	5	✓	2160	118	5	1
32	21:17:04.949	Thread Group 1-32	HTTP Request	4	✓	2160	118	4	0
33	21:17:05.099	Thread Group 1-33	HTTP Request	4	✓	2160	118	3	1
34	21:17:05.249	Thread Group 1-34	HTTP Request	4	✓	2160	118	4	1

Scroll automatically? ☐ Child samples? ☐ No of Samples 100 Latest Sample 4 Average 10 Deviation 33

May Iffah Rizki  
1506757661  
APAP-A

## 2. Halaman pada /penduduk?nik= 3101011409150001 (cepat) dan (reliable)





May Iffah Rizki

1506757661

APAP-A

Hasil dari optimasi saya berhasil karena database yang awalnya tidak memiliki primary key, diberikan primary key. Primary key berpengaruh pada performa MySQL karena primary key menyimpan baris bersama dan memberi urutan, primary key dapat mengidentifikasi sebuah baris secara unik. Primary key merepresentasikan kolom yang digunakan untuk query yang sangat penting dan sering digunakan, ia juga memiliki index yang berguna untuk performa pencarian query yang cepat, ia juga bekerja sebagai pointer untuk memilih baris ketika sebuah tabel digabungkan dengan tabel yang lainnya.

Primary key sudah pasti memiliki indexing. Selanjutnya, indexing berguna untuk mengecek data yang sudah disort. Baris yang dicek akan lebih sedikit. Hal ini mempercepat pencarian data.