


SISTEM KEPENDUDUKAN ONLINE PROVINSI DKI JAKARTA

Dalam mengembangkan fitur-fitur yang ada pada tugas kali ini, terdapat 5 tabel pada database yaitu Penduduk, Keluarga, Kota, Kelurahan dan Kecamatan

 Tugas1APAP [boot] [devtools]

▷ Spring Elements
 ▾ src/main/java
 ▷ com.example
 ▷ com.example.controller
 ▷ **com.example.dao**
 ▷ com.example.model
 ▷ com.example.service

A. PENDUDUK

1. Dao

- PendudukMapper.java

```
//PENDUDUK
@Select("select * from penduduk")
List<PendudukModel> selectAllPenduduk();

@Select("select * from penduduk where nik = #{nik}")
PendudukModel selectPenduduk (@Param("nik") String nik);

@Select("select * from penduduk where id_keluarga = #{id}")
List<PendudukModel> selectAnggotaKeluarga (@Param("id") String id);

@Select("select * from penduduk where nik LIKE #{nik}")
List<PendudukModel> selectNIK (@Param("nik") String nik);

//TAMBAH PENDUDUK
@Insert("insert into penduduk (id, nik, nama, tempat_lahir, tanggal_lahir,
jenis_kelamin, golongan_darah, agama, status_perkawinan, pekerjaan, is_wni, is_wafat,
id_keluarga, status_dalam_keluarga) "
+ "VALUES (#{id}, #{nik}, #{nama}, #{tempat_lahir}, #{tanggal_lahir},
#{jenis_kelamin}, #{golongan_darah}, #{agama}, #{status_perkawinan}, #{pekerjaan}, #{is_wni},
#{is_wafat}, #{id_keluarga}, #{status_dalam_keluarga})")
void addPenduduk (PendudukModel penduduk);

//UPDATE PENDUDUK
@Update("update penduduk SET nik = #{nik}, nama =#{nama}, tempat_lahir =
#{tempat_lahir}, tanggal_lahir = #{tanggal_lahir}, jenis_kelamin = #{jenis_kelamin},
golongan_darah = #{golongan_darah}, agama = #{agama}, status_perkawinan =
#{status_perkawinan}, pekerjaan = #{pekerjaan}, is_wni = #{is_wni}, is_wafat = #{is_wafat},
id_keluarga = #{id_keluarga}, status_dalam_keluarga = #{status_dalam_keluarga} WHERE id =
#{id}")
void updatePenduduk(PendudukModel penduduk);

//UPDATE KEMATIAN PENDUDUK
final String UPDATE_IS_WAFAT = "update penduduk set is_wafat = #{is_wafat} where id = #{id} ";
@Update(UPDATE_IS_WAFAT)
void updateIsWafat(@Param("is_wafat") String is_wafat, @Param("id") String id);

//CARI PENDUDUK
@Select("SELECT * from penduduk where id_keluarga IN (select id from keluarga where
id_kelurahan = #{id_kelurahan})")
List<PendudukModel> selectPendudukByKelurahan(@Param("id_kelurahan") String
id_kelurahan);
```

2. Model

- PendudukModel

```
import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class PendudukModel
{
    private String id;
    private String nik;
    private String nama;
    private String tempat_lahir;
    private String tanggal_lahir;
    private String jenis_kelamin;
    private String is_wni;
    private String id_keluarga;
    private String agama;
    private String pekerjaan;
    private String status_perkawinan;
    private String status_dalam_keluarga;
    private String golongan_darah;
    private String is_wafat;
}
```

3. Service

- PendudukService.java

```
import java.util.List;

public interface PendudukService
{
    List<PendudukModel> selectAllPenduduk();
    List<PendudukModel> selectAnggotaKeluarga(String id);
    List<PendudukModel> selectNIK(String nik);
    PendudukModel selectPenduduk(String nik);
    List<PendudukModel> selectPendudukByKelurahan(String id_kelurahan);

    //TAMBAH PENDUDUK
    void addPenduduk(PendudukModel penduduk);

    //UPDATE PENDUDUK
    void updatePenduduk(PendudukModel penduduk);

    //UPDATE KEMATIAN PENDUDUK
    void updateIsWafat(String is_wafat, String id);
}
```

- PendudukServiceDatabase.java

```
import java.util.List;

@Slf4j
@Service
public class PendudukServiceDatabase implements PendudukService
{
    @Autowired
    private PendudukMapper pendudukMapper;

    @Override
    public List<PendudukModel> selectAllPenduduk() {
        return pendudukMapper.selectAllPenduduk();
    }

    @Override
    public List<PendudukModel> selectAnggotaKeluarga(String id) {
        return pendudukMapper.selectAnggotaKeluarga(id);
    }

    @Override
    public List<PendudukModel> selectNIK(String nik) {
        return pendudukMapper.selectNIK(nik);
    }

    @Override
    public PendudukModel selectPenduduk(String nik) {
        return pendudukMapper.selectPenduduk(nik);
    }

    @Override
    public void addPenduduk(PendudukModel penduduk) {
        pendudukMapper.addPenduduk(penduduk);
    }

    @Override
    public void updatePenduduk(PendudukModel penduduk) {
        pendudukMapper.updatePenduduk(penduduk);
    }

    @Override
    public void updateIsWafat(String is_wafat, String id) {
        pendudukMapper.updateIsWafat(is_wafat, id);
    }

    @Override
    public List<PendudukModel> selectPendudukByKelurahan(String id_kelurahan) {
        return pendudukMapper.selectPendudukByKelurahan(id_kelurahan);
    }
}
```

B. KELUARGA

1. Dao

- PendudukMapper.java

```
//KELUARGA
@Select("select * from keluarga")
List<KeluargaModel> selectAllKeluarga ();

@Select("select * from keluarga where nomor_kk = #{nomor_kk}")
KeluargaModel selectKeluargaNKK (@Param("nomor_kk") String nomor_kk );

@Select("select * from keluarga where id = #{id_keluarga}")
KeluargaModel selectKeluarga (@Param("id_keluarga") String id_keluarga);

@Select("select * from keluarga where nomor_kk LIKE #{nomor_kk}")
List<KeluargaModel> selectNKK (@Param("nomor_kk") String nomor_kk);

//TAMBAH KELUARGA
@Insert("insert into keluarga (id, nomor_kk, alamat, RT, RW, id_kelurahan,
is_tidak_berlaku) "
+ "values (#{id}, #{nomor_kk}, #{alamat}, #{RT}, #{RW}, #{id_kelurahan},
#{is_tidak_berlaku})")
void addKeluarga(KeluargaModel keluarga);

//UPDATE KELUARGA
@Update("update keluarga set nomor_kk = #{nomor_kk}, alamat = #{alamat}, rt = #{rt},
rw = #{rw}, id_kelurahan = #{id_kelurahan}, is_tidak_berlaku = #{is_tidak_berlaku} where id =
#{id}")
void updateKeluarga(KeluargaModel keluarga);

//UPDATE KEMATIAN KELUARGA
final String UPDATE_IS_TIDAK_BERLAKU = "update keluarga set is_tidak_berlaku =
#{is_tidak_berlaku} where id = #{id}";
final String UPDATE = "";
@Update(UPDATE_IS_TIDAK_BERLAKU)
void updateIsTidakBerlaku(@Param("is_tidak_berlaku") String is_tidak_berlaku, @Param("id")
String id);
```

2. Model

- KeluargaModel

```
import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class KeluargaModel
{
    private String id;
    private String nomor_kk;
    private String alamat;
    private String RT;
    private String RW;
    private String id_kelurahan;
    private String is_tidak_berlaku;
}
```

3. Service

- KeluargaService.java

```
package com.example.service;

import java.util.List;

public interface KeluargaService
{
    List<KeluargaModel> selectAllKeluarga();
    List<KeluargaModel> selectNKK(String nomor_kk);
    KeluargaModel selectKeluarga (String id_keluarga);
    KeluargaModel selectKeluargaNKK(String nomor_kk);

    //TAMBAH KELUARGA
    void addKeluarga(KeluargaModel keluarga);

    //UPDATE KELUARGA
    void updateKeluarga(KeluargaModel keluarga);

    //UPDATE KEMATIAN KELUARGA
    void updateIsTidakBerlaku(String is_tidak_berlaku, String id);
}
```

- KeluargaServiceDatabase.java

```
import java.util.List;

@Slf4j
@Service
public class KeluargaServiceDatabase implements KeluargaService
{
    @Autowired
    private PendudukMapper pendudukMapper;

    @Override
    public KeluargaModel selectKeluarga(String id_keluarga) {
        log.info("select keluarga with id_keluarga {}", id_keluarga);
        return pendudukMapper.selectKeluarga(id_keluarga);
    }

    @Override
    public KeluargaModel selectKeluargaNKK(String nomor_kk) {
        log.info("select keluarga with nomor_kk {}", nomor_kk);
        return pendudukMapper.selectKeluargaNKK(nomor_kk);
    }

    @Override
    public List<KeluargaModel> selectAllKeluarga() {
        log.info("select semua keluarga {}");
        return pendudukMapper.selectAllKeluarga();
    }

    @Override
    public List<KeluargaModel> selectNKK(String nomor_kk) {
        return pendudukMapper.selectNKK(nomor_kk);
    }

    @Override
    public void addKeluarga(KeluargaModel keluarga) {
        pendudukMapper.addKeluarga(keluarga);
    }

    @Override
    public void updateKeluarga(KeluargaModel keluarga) {
        pendudukMapper.updateKeluarga(keluarga);
    }

    @Override
    public void updateIsTidakBerlaku(String is_tidak_berlaku, String id) {
        pendudukMapper.updateIsTidakBerlaku(is_tidak_berlaku, id);
    }
}
```

C. KOTA

1. Dao

- PendudukMapper.java

```
//KOTA
@Select("select * from kota")
List<KotaModel> selectAllKota();

@Select("select * from kota where id = #{id_kota}")
KotaModel selectKota (@Param("id_kota") String id_kota);

@Select("select * from kota where nama_kota = #{nama_kota}")
KotaModel selectKotaByNama (@Param("nama_kota") String nama_kota);
```

2. Model

- KotaModel.java

```
import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class KotaModel
{
    private String id;
    private String kode_kota;
    private String nama_kota;
}
```

3. Service

- KotaService.java

```
import com.example.model.KotaModel;

public interface KotaService
{
    List<KotaModel> selectAllKota();
    KotaModel selectKota (String id_kota);
    KotaModel selectKotaByNama (String nama_kota);
}
```

- KotaServiceDatabase.java

```
import java.util.List;

@Slf4j
@Service
public class KotaServiceDatabase implements KotaService
{
    @Autowired
    private PendudukMapper pendudukMapper;

    @Override
    public KotaModel selectKota(String id_kota) {
        log.info("select kota with id_kota {}", id_kota);
        return pendudukMapper.selectKota(id_kota);
    }

    @Override
    public List<KotaModel> selectAllKota() {
        return pendudukMapper.selectAllKota();
    }

    @Override
    public KotaModel selectKotaByNama(String nama_kota) {
        return pendudukMapper.selectKotaByNama(nama_kota);
    }
}
```

D. KECAMATAN

1. Dao

- PendudukMapper.java

```
//KECAMATAN
@Select("select * from kecamatan")
List<KecamatanModel> selectAllKecamatan();

@Select("select * from kecamatan where id = #{id_kecamatan}")
KecamatanModel selectKecamatan (@Param("id_kecamatan") String id_kecamatan);

@Select("select * from kecamatan where id_kota = #{id_kota}")
List<KecamatanModel> selectKecamatanByKota(@Param("id_kota") String id_kota);

@Select("select * from kecamatan where nama_kecamatan = #{nama_kecamatan}")
KecamatanModel selectKecamatanByNama (@Param("nama_kecamatan") String nama_kecamatan);
```

2. Model

- KecamatanModel.java

```
import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class KecamatanModel
{
    private String id;
    private String id_kota;
    private String kode_kecamatan;
    private String nama_kecamatan;
}
```

3. Service

- KecamatanService.java

```
import java.util.List;

public interface KecamatanService
{
    KecamatanModel selectKecamatan (String id_kecamatan);
    KecamatanModel selectKecamatanByNama (String nama_kecamatan);
    List<KecamatanModel> selectAllKecamatan();
    List<KecamatanModel> selectKecamatanByKota(String id_kota);
}
```

- KecamatanServiceDatabase.java

```
import java.util.List;

@Slf4j
@Service
public class KecamatanServiceDatabase implements KecamatanService
{
    @Autowired
    private PendudukMapper pendudukMapper;

    @Override
    public KecamatanModel selectKecamatan(String id_kecamatan) {
        log.info ("select kecamatan with id_kecamatan {}", id_kecamatan);
        return pendudukMapper.selectKecamatan(id_kecamatan);
    }

    @Override
    public List<KecamatanModel> selectKecamatanByKota(String id_kota) {
        log.info ("select KecamatanByKota");
        return pendudukMapper.selectKecamatanByKota(id_kota);
    }

    @Override
    public List<KecamatanModel> selectAllKecamatan() {
        return pendudukMapper.selectAllKecamatan();
    }

    @Override
    public KecamatanModel selectKecamatanByNama(String nama_kecamatan) {
        return pendudukMapper.selectKecamatanByNama(nama_kecamatan);
    }
}
```

E. KELURAHAN

1. Dao

- PendudukMapper.java

```
//KELURAHAN
@Select("select * from kelurahan")
List<KelurahanModel> selectAllKelurahan();

@Select("select * from kelurahan where id = #{id_kelurahan}")
KelurahanModel selectKelurahan (@Param("id_kelurahan") String id_kelurahan);

@Select("select * from kelurahan where id_kecamatan = #{id_kecamatan}")
List<KelurahanModel> selectKelurahanByKecamatan(@Param("id_kecamatan") String
id_kecamatan);

@Select("select * from kelurahan where nama_kelurahan = #{nama_kelurahan}")
KelurahanModel selectKelurahanByNama (@Param("nama_kelurahan") String nama_kelurahan);
```

2. Model

- KelurahanModel.java

```
import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class KelurahanModel
{
    private String id;
    private String id_kecamatan;
    private String kode_kelurahan;
    private String nama_kelurahan;
    private String kode_pos;
}
```

3. Service

- KelurahanService.java

```
import java.util.List;

public interface KelurahanService
{
    List<KelurahanModel> selectAllKelurahan();
    List<KelurahanModel> selectKelurahanByKecamatan(String id_kecamatan);
    KelurahanModel selectKelurahan (String id_kelurahan);
    KelurahanModel selectKelurahanByNama (String nama_kelurahan);
}
```

- KelurahanServiceDatabase.java

```
import java.util.List;

@Slf4j
@Service
public class KelurahanServiceDatabase implements KelurahanService
{
    @Autowired
    private PendudukMapper pendudukMapper;

    @Override
    public KelurahanModel selectKelurahan(String id_kelurahan) {
        log.info("select kelurahan with id_kelurahan {} ", id_kelurahan);
        return pendudukMapper.selectKelurahan(id_kelurahan);
    }

    @Override
    public List<KelurahanModel> selectAllKelurahan() {
        return pendudukMapper.selectAllKelurahan();
    }

    @Override
    public List<KelurahanModel> selectKelurahanByKecamatan(String id_kecamatan) {
        return pendudukMapper.selectKelurahanByKecamatan(id_kecamatan);
    }

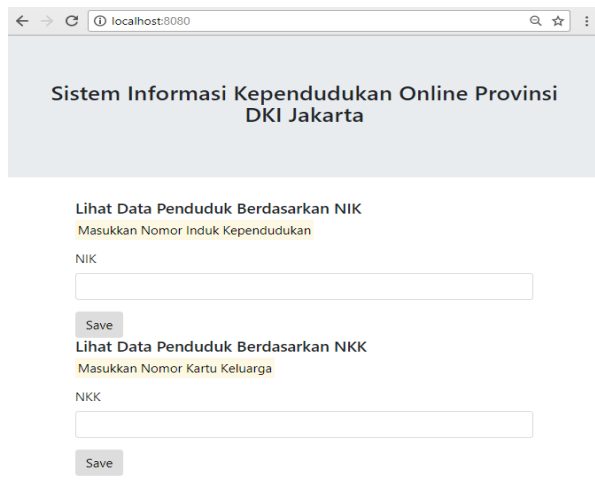
    @Override
    public KelurahanModel selectKelurahanByNama(String nama_kelurahan) {
        return pendudukMapper.selectKelurahanByNama(nama_kelurahan);
    }
}
```

1. MENAMPILKAN DATA PENDUDUK BERDASARKAN NIK

Initial Page: /

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

Jika pengguna mengakses localhost:8080 maka akan direturn ke halaman index yang akan memuat sebuah form untuk melihat data penduduk berdasarkan NIK seperti berikut :



Form Request: GET, /penduduk

Pada halaman index.html form diberikan request method GET untuk mengambil nik penduduk seperti dibawah ini

```
<h5>Lihat Data Penduduk Berdasarkan NIK</h5>
<p><mark>Masukkan Nomor Induk Kependudukan</mark></p>
<form action="/penduduk" method="get">
```

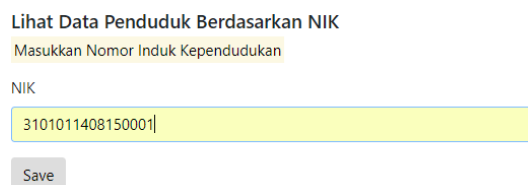
Response Page: /penduduk?nik={NIK}

Pada controller akan di ambil penduduk dengan nik tersebut, apabila ada penduduk dengan nik tersebut maka akan diambil id keluarga, id kelurahan, id kecamatan serta id kotanya seperti dibawah ini dengan memanggil method select untuk dapat mendapatkan data-data yang ada pada tabel lainnya.

```
@RequestMapping("/penduduk")
public String viewPenduduk (Model model, @RequestParam(value="nik", required = false) String nik)
{
    PendudukModel penduduk = pendudukDAO.selectPenduduk (nik);

    if (penduduk != null) {
        KeluargaModel keluarga = keluargaDAO.selectKeluarga(penduduk.getId_keluarga());
        KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
        KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(kelurahan.getId_kecamatan());
        KotaModel kota = kotaDAO.selectKota(kecamatan.getId_kota());
    }
}
```

Berikut tampilannya



The screenshot shows a web browser at localhost:8080/penduduk?nik=3101011408150001&action=save. The navigation bar includes 'Home', 'Mencari Penduduk', and 'Menambah Penduduk'. The main heading is 'Lihat Data Penduduk - 3101011408150001'. Below this is a table of resident details:

NIK	3101011408150001
Nama	Mulyono Hutagalung S.Gz
Tempat/Tanggal Lahir	Jakarta, 14 Agustus 2015
Alamat	Jln. Cikapayang No. 594
RT/RW	149/090
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	B-
Agama	Islam
Status Perkawinan	Belum Kawin
Pekerjaan	BELUM/TIDAK BEKERJA
Kewarganegaraan	WNI
Status Kematian	Wafat

At the bottom of the table is a red button labeled 'Nonaktifkan'.

Jika penduduk tidak ditemukan dengan nik yang tersimpan dalam database maka akan mengembalikan tampilan sebagai berikut

The screenshot shows a web browser at localhost:8080/penduduk?nik=3455666666&action=save. The navigation bar includes 'Home'. The main heading is 'Penduduk Tidak Ditemukan' with the subtitle 'NIK = 3455666666'.

Validasi input dilakukan dengan menambahkan required="true"

```
<div class="container">
  <h5>Lihat Data Penduduk Berdasarkan NIK</h5>
  <p><mark>Masukkan Nomor Induk Kependudukan</mark></p>
  <form action="/penduduk" method="get">
    <div class="form-group">
      <label for="nik">NIK</label> <input type="text" class="form-control" name="nik" required="true"/>
    </div>
    <div>
      <button type="submit" class="btn btn-default" name="action" value="save">Save</button>
    </div>
  </form>
</div>
```

Lihat Data Penduduk Berdasarkan NIK
Masukkan Nomor Induk Kependudukan

NIK

Save

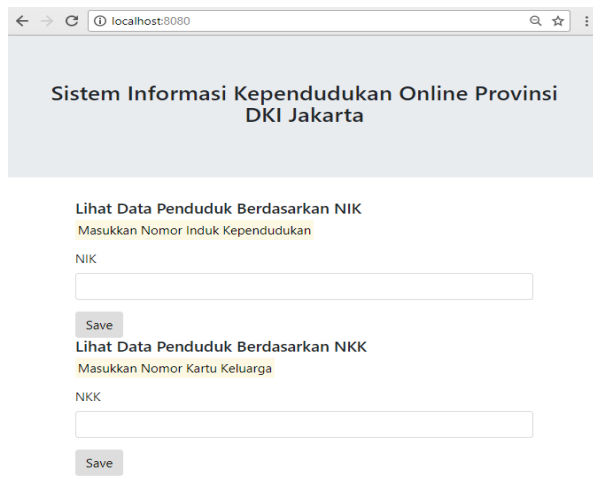
Please fill in this field.

2. MENAMPILKAN DATA KELUARGA BESERTA DAFTAR ANGGOTANYA BERDASARKAN NOMOR KK

Initial Page: /

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

Jika pengguna mengakses localhost:8080 maka akan direturn ke halaman index yang akan memuat sebuah form untuk melihat data penduduk berdasarkan NKK seperti berikut :



Form Request: GET, /keluarga

Pada halaman index.html form diberikan request method GET untuk mengambil nomor kk penduduk seperti dibawah ini

```
<div class="container">
<h5>Lihat Data Penduduk Berdasarkan NKK</h5>
<p><mark>Masukkan Nomor Kartu Keluarga</mark></p>
<form action="/keluarga" method="get">
```

Response Page: /keluarga?nkk={NKK}

Pada controller akan di ambil penduduk dengan nomor kk tersebut, apabila ada penduduk dengan nomor kk tersebut maka akan diambil data penduduk dari daftar penduduk yang memiliki id keluarga yang sama, serta mengambil id kelurahan, id kecamatan serta id kotanya seperti dibawah ini dengan memanggil method select untuk dapat mendapatkan data-data yang ada pada tabel lainnya.

```

@RequestMapping("/keluarga")
public String viewPendudukNKK (Model model, @RequestParam(value="nomor_kk", required = false) String nomor_kk)
{
    KeluargaModel keluarga = keluargaDAO.selectKeluargaNKK(nomor_kk);

    if (keluarga != null) {
        List<PendudukModel> penduduk = pendudukDAO.selectAnggotaKeluarga(keluarga.getId());
        KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
        KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(keluurahan.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 "viewall";
    } else {
        model.addAttribute("nomor_kk", nomor_kk);
        return "not-found";
    }
}

```

Berikut tampilannya

Lihat Data Penduduk Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

NKK

3101010201030001

Save

← → ↻ localhost:8080/keluarga?nomor_kk=3101010201030001&action=save 🔍 ☆ ⋮

Home Home Mencari Penduduk Menambah Penduduk

Lihat Data Keluarga - 3101010201030001

NKK	3101010201030001
Alamat	Ki. Panjaitan No. 548
RT/RW	031/028
Kelurahan/Desa	PULAU UNTUNG JAWA
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU

No.	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga	Kewarganegaraan
1	Wardaya Maulana	3101012402140001	Laki-laki	Jakarta	2014-02-24	Kristen	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI
2	Hasta Tugiman Zulkarnain S.Pt	3101011111760001	Laki-laki	Jakarta	1976-11-11	Islam	PEMBANTU RUMAH TANGGA	Cerai Mati	Pembantu	WNI
3	Wirda Uyainah S.Pd	3101015909170001	Perempuan	Jakarta	2017-09-19	Kristen	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI

Jika penduduk tidak ditemukan dengan nomor kk yang tersimpan di database maka akan mengembalikan tampilan sebagai berikut

← → ↻ localhost:8080/keluarga?nomor_kk=3101010303890001&action=save 🔍 ☆ ⋮

Home

Penduduk Tidak Ditemukan

Nomor KK = 3101010303890001

Validasi input dilakukan dengan menambahkan required = "true"

```
<div class="container">
<h5>Lihat Data Penduduk Berdasarkan NKK</h5>
<p><mark>Masukkan Nomor Kartu Keluarga</mark></p>
<form action="/keluarga" method="get">
  <div class="form-group">
    <label for="nomor_kk">NKK</label> <input type="text" class="form-control" name="nomor_kk" required="true"/>
  </div>
  <div>
    <button type="submit" class="btn btn-default" name="action" value="save">Save</button>
  </div>
</form>
</div>
```

3. MENAMBAHKAN PENDUDUK BARU SEBAGAI ANGGOTA KELUARGA

Initial Page: /penduduk/tambah

Pada controller maka dibuat method tambahPendudukBaru dengan RequestMethod adalah GET yang akan mengembalikan ke halaman form-add.html

```
@RequestMapping(value="/penduduk/tambah", method=RequestMethod.GET)
public String tambahPendudukBaru(Model model) {
    PendudukModel penduduk = new PendudukModel();

    penduduk.setJenis_kelamin("");
    penduduk.setGolongan_darah("");
    penduduk.setStatus_perkawinan("");
    penduduk.setIs_wni("");
    penduduk.setIs_wafat("");
    model.addAttribute("penduduk", penduduk);
    return "form-add";
}
```

Form Request: POST, /penduduk/tambah

Pada form-add.html dibuat sebuah form yang akan mengembalikan ke /penduduk/tambah dengan method POST seperti dibawah ini

```
<form method="post" action="/penduduk/tambah" th:object="${penduduk}">
```

Response Page: /penduduk/tambah

Pada controller dibuat sebuah method tambahPenduduk dengan RequestMethod adalah POST yang akan memproses data-data penduduk baru yang dimasukkan dan generate nomor nik penduduk secara otomatis sesuai dengan ketentuan yang diberikan. Pada method ini juga akan dipanggil method addPenduduk di PendudukService yang akan menambahkan data-data penduduk ke database.

```

@RequestMapping(value="/penduduk/tambah", method=RequestMethod.POST)
public String tambahPenduduk(@Valid @ModelAttribute PendudukModel penduduk, Model model) {

    KeluargaModel keluarga = keluargaDAO.selectKeluarga(penduduk.getId_keluarga());
    KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
    KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(kelurahan.getId_kecamatan());

    //mengambil tanggal lahir penduduk yang dimasukkan dan dimasukkan ke dalam array dan pemisahannya adalah tanda -
    String[] tanggalLahir = penduduk.getTanggal_lahir().split("-");
    //jika jenis kelamin penduduk adalah perempuan maka akan diambil array dengan indeks ke 2 dari tanggal lahirnya yaitu hari
    //hari tanggal lahir penduduk perempuan akan ditambah 40
    if(penduduk.getJenis_kelamin().equals("1")) {
        tanggalLahir[2] = (Integer.parseInt(tanggalLahir[2]) + 40) + "";
    }
    //membuat nik baru dengan mengambil kode kecamatan dari indeks ke 0-6 ditambahkan dengan (hari + bulan + tahun)
    //data tahun yang akan diambil hanya dari indeks ke 2
    String nik = kecamatan.getKode_kecamatan().substring(0,6) + tanggalLahir[2] + tanggalLahir[1] + tanggalLahir[0].substring(2);
    //memanggil method selectNIK yang akan memunculkan data dengan nik yang sama karena apabila penduduk lahir
    //pada tanggal yang sama maka harus dibuat nomor urutannya
    List<PendudukModel> sama = pendudukDAO.selectNIK(nik + "%");
    //nomor urutan dimulai dari 0001
    String nomor = "0001";
    if(sama.size() > 0) {
        int nomorSama = Integer.parseInt(sama.get(sama.size()-1).getNik().substring(12)) + 1;
        nomor = nomorSama + "";
    }

    int count = 4 - nomor.length();
    for(int i = 0; i < count; i++) {
        nomor = "0" + nomor;
    }
    //nik baru akan ditambahkan dengan nomor urutan
    nik = nik + nomor;
    //penduduk akan di set niknya
    penduduk.setNik(nik);
    //mengambil semua daftar penduduk
    List<PendudukModel> daftarPenduduk = pendudukDAO.selectAllPenduduk();
    //penduduk yang baru id penduduknya akan ditambahkan 1 dari banyaknya penduduk saat ini
    penduduk.setId("" + (daftarPenduduk.size() + 1));
    //memanggil method tambah penduduk untuk menambahkan data-data penduduk lainnya ke database
    pendudukDAO.addPenduduk(penduduk);
    return "view-tambah";
}

```

Berikut tampilannya

Jika penduduk yang ditambahkan adalah perempuan

localhost:8080/penduduk/tambah

Menambah Penduduk
Siapkan Memasukkan Data Penduduk Baru

Masukkan Nama	
<small>Raka Andriana</small>	
Tempat Lahir	
<small>Jakarta</small>	
Tanggal Lahir	
<small>10/22/2017</small>	
Jenis Kelamin	
<small>Perempuan</small>	▼
Golongan Darah	
<small>B</small>	▼
Agama	
<small>Islam</small>	
Status Perkawinan	
<small>Kawin</small>	▼
Pekerjaan	
<small>Pengajar</small>	
Kewarganegaraan	
<small>WNI</small>	▼
Status Kematian	
<small>Hidup</small>	▼
ID Keluarga	
<small>1</small>	
Status Dalam Keluarga	
<small>anak</small>	
<input type="button" value="Simpan"/>	

localhost:8080/penduduk/tambah

Menambah Penduduk

Silahkan Memasukkan Data Penduduk Baru

Masukkan Nama
Miana Tama

Tempat Lahir
Korea

Tanggal Lahir
10/22/2017

Jenis Kelamin
Perempuan ▼

Golongan Darah
B ▼

Agama
Islam ▼

Status Perkawinan
Kawin ▼

Pekerjaan
Mahasiswa ▼

Kewarganegaraan
WNI ▼

Status Kematian
Hidup ▼

ID Keluarga
1

Status Dalam Keluarga
anak

Hasil generate nomor nik jika ada yang memiliki tanggal lahir sama dan perempuan

20	Raisa Andriana	3101016210170001	Perempuan	Jakarta	2017-10-22	Islam	Penyanyi	Kawin	anak	WNI
21	Miana Tama	3101016210170002	Perempuan	Korea	2017-10-22	Islam	Mahasiswa	Kawin	anak	WNI

Jika penduduk yang ditambahkan adalah laki-laki

Menambah Penduduk

Silahkan Memasukkan Data Penduduk Baru

Masukkan Nama
Bahrun Sutjiono

Tempat Lahir
Jakarta

Tanggal Lahir
01/11/1965

Jenis Kelamin
Laki Laki ▼

Golongan Darah
B ▼

Agama
Katolik ▼

Status Perkawinan
Kawin ▼

Pekerjaan
Karyawan Swasta ▼

Kewarganegaraan
WNI ▼

Status Kematian
Hidup ▼

ID Keluarga
1

Status Dalam Keluarga
Ayah

Jika penduduk berhasil ditambahkan

← → ↻ localhost:8080/penduduk/tambah

Home Home Mencari Penduduk Menambah Penduduk

Sukses!

Penduduk dengan NIK 3101011101650001 berhasil ditambahkan

25	Bahrun Sutjiono	3101011101650001	Laki-laki	Jakarta	1965-01-11	Katolik	Karyawan Swasta	Kawin	Ayah	WNI
----	-----------------	------------------	-----------	---------	------------	---------	-----------------	-------	------	-----

Validasi input dilakukan dengan menambahkan `required="true"` pada setiap input

```
</hr>
<div class="container-fluid">
  <form method="post" action="/penduduk/tambah" th:object="${penduduk}">
    <div class="form-group">
      <label for="nama"><h6>Masukkan Nama</h6></label>
      <input type="text" class="form-control" name="nama" id="nama" maxlength="128" th:field="**{nama}" required="true" />
    </div>

    <div class="form-group">
      <label for="tempat_lahir"><h6>Tempat Lahir</h6></label>
      <input type="text" class="form-control" name="tempat_lahir" id="tempat_lahir" maxlength="128" th:field="**{tempat_lahir}" required="true" />
    </div>

    <div class="form-group">
      <label for="tanggal_lahir"><h6>Tanggal Lahir</h6></label>
      <input type="date" class="form-control" name="tanggal_lahir" id="tanggal_lahir" th:field="**{tanggal_lahir}" required="true" />
    </div>
  </form>
</div>
```

Menambah Penduduk

Silahkan Memasukkan Data Penduduk Baru

Masukkan Nama

Tempat Lahir

Please fill in this field.

Tempat Lahir

Tanggal Lahir

Please fill in this field.

Tanggal Lahir

Jenis Kelamin

Please fill in this field.

Pilih Jenis Kelamin

Golongan Darah

Please select an item in the list.

Pilih Golongan Darah

Agama

Please select an item in the list.

Agama

Status Perkawinan

Please fill in this field.

Status Perkawinan

Pekerjaan

Please select an item in the list.

Pekerjaan

Kewarganegaraan

Please fill in this field.

Kewarganegaraan

Status Kematian

Please select an item in the list.

4. MENAMBAHKAN KELUARGA BARU

Initial Page: /keluarga/tambah

Pada controller maka dibuat method tambahKeluargaBaru dengan RequestMethod adalah GET yang akan mengembalikan ke halaman form-add-keluarga.html

```
@RequestMapping(value="/keluarga/tambah", method=RequestMethod.GET)
public String tambahKeluargaBaru(Model model){
    KeluargaModel keluarga = new KeluargaModel();
    keluarga.setId_kelurahan("");
    model.addAttribute("keluarga", keluarga);
    //mengambil semua daftar kelurahan
    List<KelurahanModel> kelurahan = kelurahanDAO.selectAllKelurahan();
    //mengambil semua daftar kecamatan
    List<KecamatanModel> kecamatan = kecamatanDAO.selectAllKecamatan();
    //mengambil semua daftar kota
    List<KotaModel> kota = kotaDAO.selectAllKota();

    //melakukan looping untuk mendapatkan kelurahan yang id kecamatannya sama dengan id kecamatan yang tersedia
    //mendapatkan kecamatan yang id kotanya sama dengan id kota yang tersedia
    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())){
                        //Hasilnya akan disimpan di dalam variabel namaKelurahan
                        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 "form-add-keluarga";
}
```

Form Request: POST, /keluarga/tambah

Pada form-add-keluarga.html dibuat sebuah form yang akan mengembalikan ke /penduduk/tambah dengan method POST seperti dibawah ini

```
<form method="post" action="/keluarga/tambah" th:object="${keluarga}">
```

Response Page: /keluarga/tambah

Pada controller dibuat sebuah method tambahKeluarga dengan RequestMethod adalah POST yang akan memproses data-data keluarga baru yang dimasukkan dan

mengenerate nomor nkk secara otomatis sesuai dengan ketentuan yang diberikan. Pada method ini juga akan dipanggil method addKeluarga di KeluargaService yang akan menambahkan data-data keluarga ke database.

```
@RequestMapping(value="/keluarga/tambah", method=RequestMethod.POST)
public String tambahKeluarga(@Valid @ModelAttribute KeluargaModel keluarga, Model model) {
    //untuk mengenerate nomor nkk baru maka harus diatur tanggal KK diterbitkan
    //dengan method new Date() dimana tanggal akan diformat dalam tahun-bulan-hari
    Date sekarang = new Date();
    SimpleDateFormat tanggal = new SimpleDateFormat("yyyy-MM-dd");
    String tanggalSekarang = tanggal.format(sekarang);
    String[] date = tanggalSekarang.split("-");
    //mengambil id kelurahan dari data keluarga
    KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
    //nomor kk dibentuk dari kode kelurahan yang diambil dari indeks 0-6 ditambah dengan hari
    //bulan dan tahun yang diambil datanya hanya mulai dari indeks ke 2
    String nomor_kk = kelurahan.getKode_kelurahan().substring(0,6);
    nomor_kk += date[2] + date[1] + date[0].substring(2);
    //memanggil method selectNKK yang akan memunculkan data dengan nkk yang sama karena apabila
    //penduduk membuat KK pada tanggal yang sama maka harus dibuat nomor urutannya
    List<KeluargaModel> sama = keluargaDAO.selectNKK(nomor_kk + "%");
    //nomor urutan dimulai dari 0001
    String nomor = "0001";
    if(sama.size() > 0) {
        int nomorSama = Integer.parseInt(sama.get(sama.size()-1).getNomor_kk().substring(12)) + 1;
        nomor = nomorSama + "";
    }

    int count = 4 - nomor.length();
    for(int i = 0; i < count; i++) {
        nomor = "0" + nomor;
    }
    //nomor kk akan ditambahkan nomor urutan
    nomor_kk = nomor_kk + nomor;
    keluarga.setNomor_kk(nomor_kk);
    keluarga.setIs_tidak_berlaku("0");
    //mengambil semua daftar keluarga
    List<KeluargaModel> daftarKeluarga = keluargaDAO.selectAllKeluarga();
    //id keluarga ditambahkan 1 dari jumlah keluarga saat ini
    keluarga.setId("" + (daftarKeluarga.size() + 1));
    //memanggil method addKeluarga untuk menambahkan data-data keluarga pada database
    keluargaDAO.addKeluarga(keluarga);
    model.addAttribute("confirmation", "Penduduk dengan NKK " + nomor_kk + " berhasil ditambahkan" );
    return "view-tambah";
}
```

Berikut tampilannya :

localhost:8080/keluarga/tambah

Tambah Keluarga

Silahkan Memasukkan Data Keluarga Baru

Alamat
Jalan Manggis Raya Nomor 15

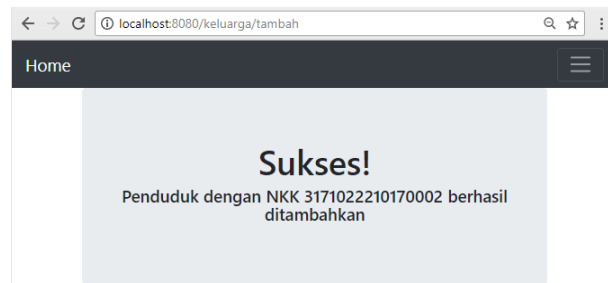
RT
006

RW
007

Kota / Kecamatan / Kelurahan
KOTA JAKARTA SELATAN / PASAR MINGGU / PEJATEN BARAT

Simpan

Jika berhasil ditambahkan maka akan muncul tampilan sebagai berikut

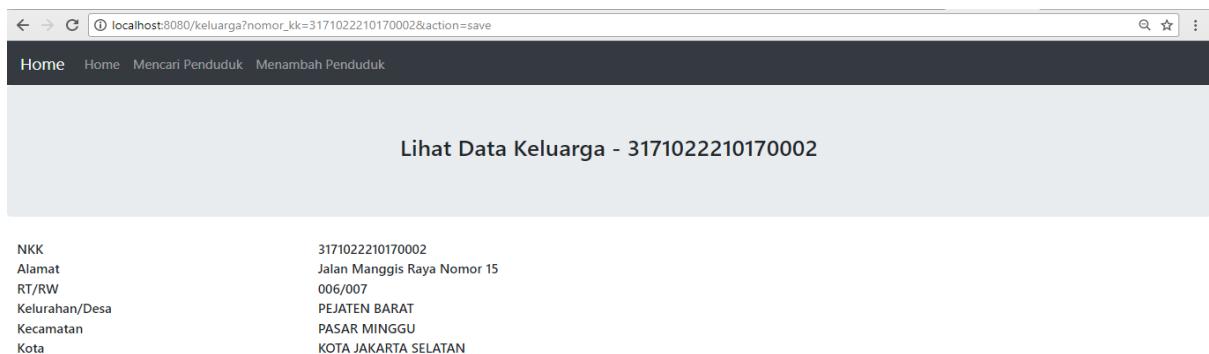


Lihat Data Penduduk Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

NKK

Save



Validasi input dengan menambahkan required="true" pada setiap input

```
<div class="container-fluid">
  <h3>Tambah Keluarga</h3>
  <p><mark>Silahkan Memasukkan Data Keluarga Baru</mark></p>
</div>

<hr/>
<div class="container-fluid">
  <form method="post" action="/keluarga/tambah" th:object="${keluarga}">
    <div class="form-group">
      <label for="alamat"><h6>Alamat</h6></label>
      <input type="text" class="form-control" name="alamat" id="alamat" maxlength="128" th:field="*{alamat}" required="true" />
    </div>

    <div class="form-group">
      <label for="RT"><h6>RT</h6></label>
      <input type="text" class="form-control" name="RT" id="RT" maxlength="128" th:field="*{RT}" required="true" />
    </div>

    <div class="form-group">
      <label for="RW"><h6>RW</h6></label>
      <input type="text" class="form-control" name="RW" id="RW" th:field="*{RW}" required="true"/>
    </div>


    <div class="form-group">
      <label for="id_kelurahan"><h6>Kota / Kecamatan / Kelurahan</h6></label>
      <select class="form-control" name="id_kelurahan" id="id_kelurahan" th:field="*{id_kelurahan}" required="true">
        <option value="" th:disabled="disabled" th:selected="selected">Pilih Kelurahan</option>
        <div th:each="kelurahan: ${kelurahan}">
          <option th:value="${kelurahan.id}" th:text="${kelurahan.nama_kelurahan}">Kelurahan yang mau dipilih</option>
        </div>
      </select>
    </div>
  </div>
</div>
```

Tambah Keluarga

Silahkan Memasukkan Data Keluarga Baru

Alamat

RT

 Please fill in this field.

Tambah Keluarga


Silahkan Memasukkan Data Keluarga Baru

Alamat

Jalan Haji Dimun Nomor 100

RT

RW

 Please fill in this field.

Tambah Keluarga

Silahkan Memasukkan Data Keluarga Baru

Alamat

Jalan Haji Dimun Nomor 100

RT

005

RW

|

Kota / Kecamatan / Kelurahan

Please fill in this field.

Tambah Keluarga

Silahkan Memasukkan Data Keluarga Baru

Alamat

Jalan Haji Dimun Nomor 100

RT

005

RW

009

Kota / Kecamatan / Kelurahan

Pilih Kelurahan

Simpan

Please select an item in the list.

5. MENGUBAH DATA PENDUDUK

Initial Page: /penduduk/ubah/{NIK}

Pada controller maka dibuat method ubahPenduduk dengan RequestMethod adalah GET yang akan mengembalikan ke halaman form-update-penduduk.html

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

    if (penduduk != null) {
        model.addAttribute("penduduk", penduduk);
    }else {
        model.addAttribute("nik", "nik");
        return "not-found";
    }
    return "form-update-penduduk";
}
```

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

Pada form-update-penduduk.html dibuat sebuah form yang akan mengembalikan ke /penduduk/ubah/{nik} dengan method POST seperti dibawah ini

```
<form method="post" th:action="${nik}" th:object="${penduduk}">
```

Response Page: /penduduk/ubah/{NIK}

Pada controller dibuat sebuah method ubah dengan RequestMethod adalah POST yang akan mengubah data penduduk yang telah ada dan generate nomor nik secara otomatis sesuai dengan ketentuan yang diberikan apabila tanggal lahir penduduk diubah, sehingga cara yang digunakan untuk generate nik sama dengan yang ada pada fitur sebelumnya. Pada method ini juga akan dipanggil method updatePenduduk di PendudukService yang akan mengubah data-data penduduk ke database.

```
@RequestMapping(value="/penduduk/ubah/{nik}", method=RequestMethod.POST)
public String ubah(@PathVariable(value = "nik")String nik, @Valid @ModelAttribute PendudukModel pendudukUpdate, Model model) {
    PendudukModel pendudukDb = pendudukDAO.selectPenduduk(nik);
    //untuk menyimpan nomor nik yang lama
    String nikLama = nik;
    pendudukDb.setName(pendudukUpdate.getName());
    pendudukDb.setTempat_lahir(pendudukUpdate.getTempat_lahir());
    pendudukDb.setTanggal_lahir(pendudukUpdate.getTanggal_lahir());
    pendudukDb.setJenis_kelamin(pendudukUpdate.getJenis_kelamin());
    pendudukDb.setIs_wni(pendudukUpdate.getIs_wni());
    pendudukDb.setId_keluarga(pendudukUpdate.getId_keluarga());
    pendudukDb.setAgama(pendudukUpdate.getAgama());
    pendudukDb.setPekerjaan(pendudukUpdate.getPekerjaan());
    pendudukDb.setStatus_perkawinan(pendudukUpdate.getStatus_perkawinan());
    pendudukDb.setStatus_dalam_keluarga(pendudukUpdate.getStatus_dalam_keluarga());
    pendudukDb.setGolongan_darah(pendudukUpdate.getGolongan_darah());

    KeluargaModel keluarga = keluargaDAO.selectKeluarga(pendudukDb.getId_keluarga());
    KelurahanModel kelurahan = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
    KecamatanModel kecamatan = kecamatanDAO.selectKecamatan(kelurahan.getId_kecamatan());
    String[] tanggalLahir = pendudukDb.getTanggal_lahir().split("-");
    if(pendudukDb.getJenis_kelamin().equals("1")) {
        tanggalLahir[2] = (Integer.parseInt(tanggalLahir[2]) + 40) + "";
    }
    String newNik = kecamatan.getKode_kecamatan().substring(0,6) + tanggalLahir[2] + tanggalLahir[1] + tanggalLahir[0].substring(2);
    List<PendudukModel> sama = pendudukDAO.selectNIK(newNik + "%");
    String nomor = "0001";
    if(sama.size() > 0) {
        int nomorSama = Integer.parseInt(sama.get(sama.size()-1).getNik().substring(12)) + 1;
        nomor = nomorSama + "";
    }

    int count = 4 - nomor.length();
    for(int i = 0; i < count; i++) {
        nomor = "0" + nomor;
    }
    newNik = newNik + nomor;
    pendudukDb.setNik(newNik);

    pendudukDAO.updatePenduduk(pendudukDb);
    model.addAttribute("confirmation", "Penduduk dengan NIK " + nikLama + " berhasil diubah");
    return "view-tambah";
}
```

Sebelum diubah

Lihat Data Keluarga - 3101010101050001

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

No.	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga
1	Rai Indhi	3101010803170001	Perempuan	Bogor	2017-03-28	Islam	mahasiswa	Kawin	anak

Berikut tampilannya saat mengakses localhost:8080/penduduk/ubah/{nik}

localhost:8080/penduduk/ubah/3101016803170001

Mengubah Data Penduduk
3101016803170001

Masukkan Nama
Rai Indri

Tempat Lahir
Bogor

Tanggal Lahir
05/08/2017

Jenis Kelamin
Perempuan ▼

Golongan Darah
A ▼

Agama
Islam

Status Perkawinan
Kawin ▼

Pekerjaan
mahasiswa

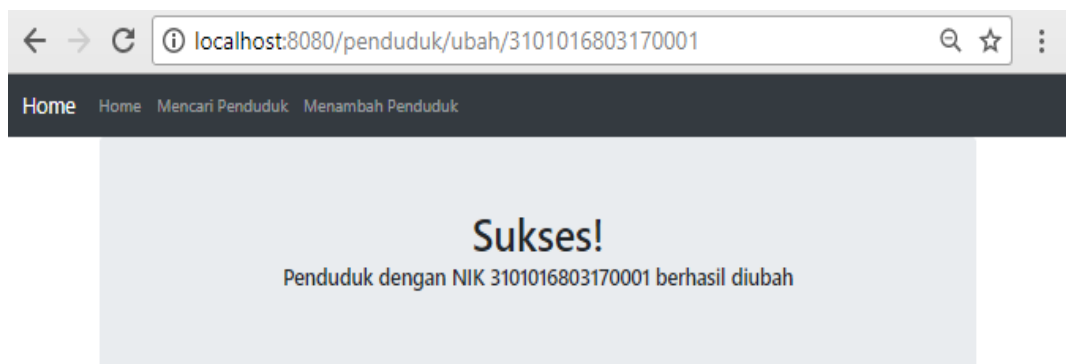
Kewarganegaraan
WNI ▼

Status Kematian
Hidup ▼

ID Keluarga
1

Status Dalam Keluarga
anak

Jika berhasil diubah maka akan ditampilkan pesan sukses dengan nik yang lama



Data penduduk dengan NIK 3101016803170001 berhasil diubah

Lihat Data Keluarga - 3101010101050001									
NIK		3101010101050001							
Alamat		Jln. Cikapayang No. 594							
RT/RW		149/090							
Kelurahan/Desa		PULAU PARI							
Kecamatan		KEPULAUAN SERIBU SELATAN							
Kota		KABUPATEN KEPULAUAN SERIBU							
No.	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga
1	Sinta Johana	3101015209170001	Perempuan	Arab	2017-09-12	Islam	mahasiswa	Belum Kawin	anak

Validasi input dengan menambahkan required="true" pada setiap input

```
<hr/>
<div class="container-fluid">
  <form method="post" th:action="{nik}" th:object="{penduduk}">
    <div class="form-group">
      <label for="nama"><h6>Masukkan Nama</h6></label>
      <input type="text" class="form-control" name="nama" id="nama" maxlength="128" th:value="{nama}" required="true" />
    </div>

    <div class="form-group">
      <label for="tempat_lahir"><h6>Tempat Lahir</h6></label>
      <input type="text" class="form-control" name="tempat_lahir" id="tempat_lahir" maxlength="128" th:value="{tempat_lahir}" required="true" />
    </div>
  </form>
</div>
```

Mengubah Data Penduduk
3101010102940001

Masukkan Nama

Tempat Lahir

Please fill in this field.

6. MENGUBAH DATA KELUARGA

Initial Page: /keluarga/ubah/{NKK}

Pada controller maka dibuat method ubahKeluarga dengan RequestMethod adalah GET yang akan mengembalikan ke halaman form-update.html

```
@RequestMapping(value="/keluarga/ubah/{nkk}", method = RequestMethod.GET)
public String ubahKeluarga(@PathVariable(value = "nkk") String nkk, Model model) {
    KeluargaModel keluargaModel = keluargaDAO.selectKeluargaNKK(nkk);
    if(keluargaModel != null) {
        model.addAttribute("nkk", nkk);
        model.addAttribute("keluarga", keluargaModel);
        //mengambil semua daftar kelurahan
        List<KelurahanModel> kelurahan = kelurahanDAO.selectAllKelurahan();
        //mengambil semua daftar kecamatan
        List<KecamatanModel> kecamatan = kecamatanDAO.selectAllKecamatan();
        //mengambil semua daftar kota
        List<KotaModel> kota = kotaDAO.selectAllKota();

        //melakukan looping untuk mendapatkan kelurahan yang id kecamatannya sama dengan id kecamatan yang tersedia
        //mendapatkan kecamatan yang id kotanya sama dengan id kota yang tersedia
        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);
                            if(kelurahan.get(i).getId().equals(keluargaModel.getId_kelurahan())) {
                                model.addAttribute("current_kelurahan", kelurahan.get(i));
                            }
                        }
                    }
                }
            }
        }

        model.addAttribute("kelurahan", kelurahan);
        //jika nkk ditemukan mengembalikan ke halaman form-update
        return "form-update";
    }
    else {
        //jika nkk tidak ditemukan maka akan ditampilkan pesan nkk tidak ditemukan
        model.addAttribute("nomor_kk", nkk);
        return "not-found";
    }
}
```

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

Pada form-update.html dibuat sebuah form yang akan mengembalikan ke /penduduk/ubah/{nkk} dengan method POST seperti dibawah ini

```
<form method="post" th:action="{nkk}" th:object="{keluarga}">
```

Response Page: /keluarga/ubah/{NKK}

Pada controller dibuat sebuah method ubahKeluargaku dengan RequestMethod adalah POST yang akan mengubah data keluarga yang telah ada dan generate nomor nkk secara otomatis sesuai dengan ketentuan yang diberikan apabila id kelurahan diubah, sehingga cara yang digunakan untuk generate nkk sama dengan yang ada pada fitur sebelumnya. Pada method ini juga akan dipanggil method update di KeluargaService yang akan mengubah data-data keluarga ke database.

```
@RequestMapping(value="/keluarga/ubah/{nkk}", method = RequestMethod.POST)
public String ubahKeluargaku(@PathVariable(value = "nkk") String nkk
    ,@Valid @ModelAttribute KeluargaModel keluargaUpdate, Model model) {
    //mengambil keluarga yang sesuai nkk
    KeluargaModel keluargaDb = keluargaDAO.selectKeluargaNKK(nkk);
    //menyimpan nilai nkk lama yang belum berubah
    String nkkLama = nkk;
    //mengubah data menjadi data yang baru
    keluargaDb.setAlamat(keluargaUpdate.getAlamat());
    keluargaDb.setRT(keluargaUpdate.getRT());
    keluargaDb.setRW(keluargaUpdate.getRW());
    keluargaDb.setIs_tidak_berlaku(keluargaUpdate.getIs_tidak_berlaku());

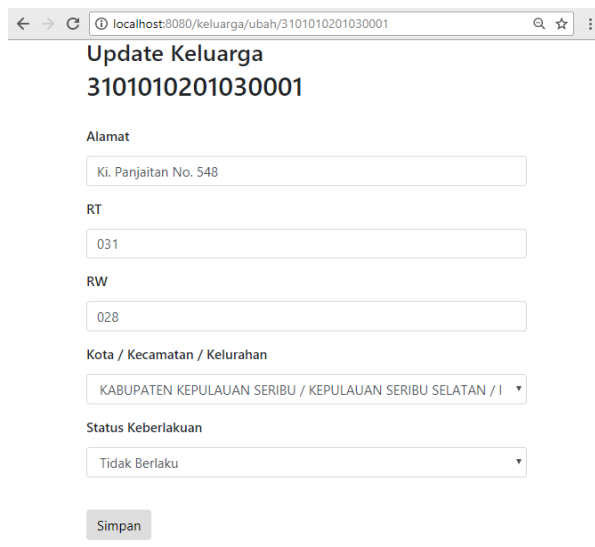
    //jika ganti id kelurahan
    if(!keluargaUpdate.getId_kelurahan().equals(keluargaDb.getId_kelurahan())) {
        Date sekarang = new Date();
        SimpleDateFormat tanggal = new SimpleDateFormat("yyyy-MM-dd");
        String tanggalSekarang = tanggal.format(sekarang);
        String[] date = tanggalSekarang.split("-");
        KelurahanModel kelurahan = keluargaDAO.selectKelurahan(keluargaUpdate.getId_kelurahan());
        String nomor_kk = kelurahan.getKode_kelurahan().substring(0,6);
        nomor_kk += date[2] + date[1] + date [0].substring(2);

        List<KeluargaModel> sama = keluargaDAO.selectNKK(nomor_kk + "%");
        String nomor = "0001";
        if(sama.size() > 0) {
            int nomorSama = Integer.parseInt(sama.get(sama.size()-1).getNomor_kk().substring(12)) + 1;
            nomor = nomorSama + "";
        }

        int count = 4 - nomor.length();
        for(int i = 0; i < count; i++) {
            nomor = "0" + nomor;
        }
        nomor_kk = nomor_kk + nomor;
        keluargaDb.setNomor_kk(nomor_kk);
    }

    keluargaDb.setId_kelurahan(keluargaUpdate.getId_kelurahan());
    //memanggil method update pada KeluargaService
    keluargaDAO.update(keluargaDb);
    model.addAttribute("confirmation", "Penduduk dengan NKK " + nkkLama + " berhasil diubah" );
    return "view-tambah";
}
```


Berikut tampilannya :



← → ↻ localhost:8080/keluarga/ubah/3101010201030001 🔍 ☆ ⋮

Update Keluarga 3101010201030001

Alamat

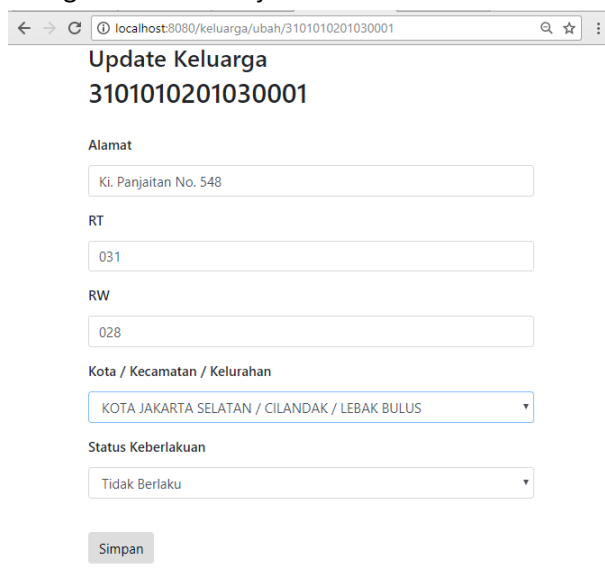
RT

RW

Kota / Kecamatan / Kelurahan

Status Keberlakuan

Mengubah data menjadi



← → ↻ localhost:8080/keluarga/ubah/3101010201030001 🔍 ☆ ⋮

Update Keluarga 3101010201030001

Alamat

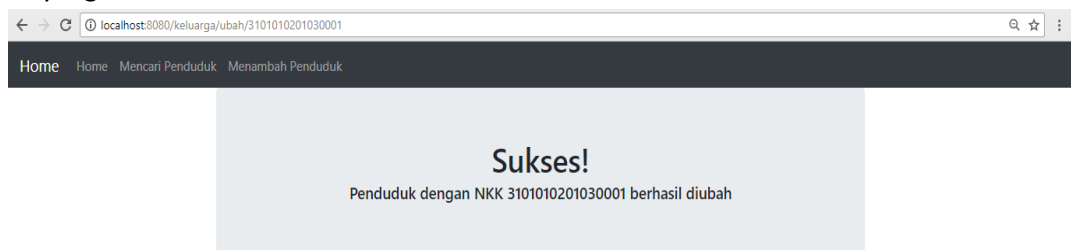
RT

RW

Kota / Kecamatan / Kelurahan

Status Keberlakuan

Jika data keluarga berhasil diubah maka akan ditampilkan pesan sukses dengan nomor kk yang lama



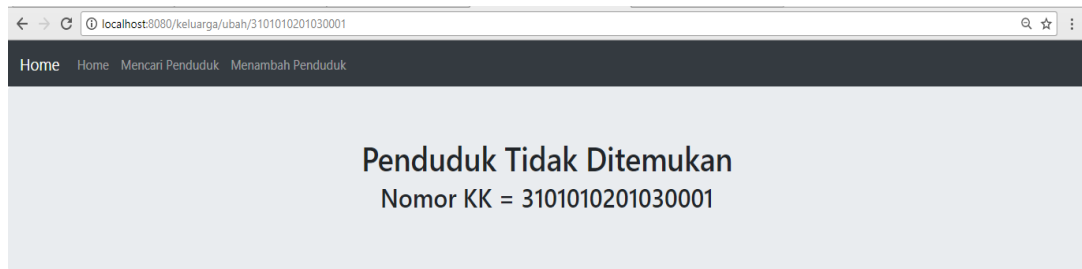
← → ↻ localhost:8080/keluarga/ubah/3101010201030001 🔍 ☆ ⋮

Home Home Mencari Penduduk Menambah Penduduk

Sukses!

Penduduk dengan NKK 3101010201030001 berhasil diubah

Jika kita ingin mengubah data dengan NKK yang tadi maka tidak dapat dilakukan karena sudah diganti



Data di database berubah dengan mengenerate nkk yang baru

19 3171032210170002 Ki. Panjaitan No. 548 031 028 20 0

Validasi dengan menambahkan required="true" pada setiap input

```
<div class="container-fluid">
  <form method="post" th:action="{nkk}" th:object="{keluarga}">
    <div class="form-group">
      <label for="nama"><h6>Alamat</h6></label>
      <input type="name" class="form-control" name="alamat" id="nama" maxlength="128" th:value="{alamat}" required="true" />
    </div>

    <div class="form-group">
      <label for="RT"><h6>RT</h6></label>
      <input type="name" class="form-control" name="RT" id="RT" maxlength="128" th:value="{RT}" required="true" />
    </div>
  </form>
```

Mengubah Data Keluarga

3174053004000001

Alamat

Dk. Abang No. 182

RT

|

RW

Please fill in this field.

020

Kota / Kecamatan / Kelurahan

KOTA JAKARTA BARAT / TAMBORA / KALIANYAR

Status Keberlakuan

Tidak Berlaku

Simpan

Mengubah Data Keluarga 3174053004000001

Alamat

RT

RW

Kota / Kecamatan / Kelurahan Please fill in this field.

Status Keberlakuan

7. MENGUBAH STATUS KEMATIAN PENDUDUK

Initial Page: /penduduk/{NIK} /penduduk?nik={NIK}

Form Request: POST, /penduduk/mati

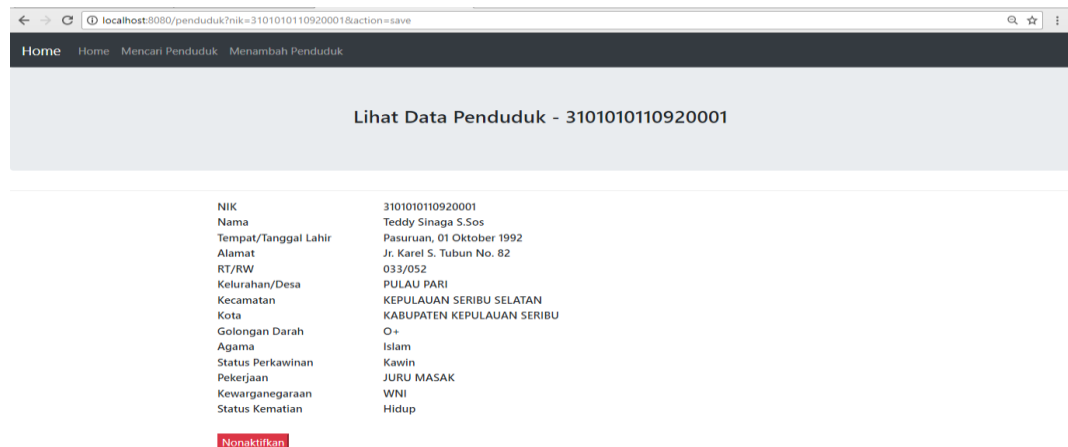
Response Page: /penduduk/{NIK} /penduduk?nik={NIK}

Pada controller dibuat sebuah method pendudukWafat yang akan muncul saat kita mengakses pencarian penduduk dengan NIK. Pada method ini juga akan dipanggil method updateIsTidakBerlaku di KeluargaService apabila penduduk yang sudah wafat tidak lagi memiliki anggota keluarga, lalu juga akan dipanggil method updateIsWafat di PendudukService.

```
@RequestMapping(value = "/penduduk/mati", method=RequestMethod.POST)
public String pendudukWafat(Model model, @RequestParam(value="nik") String nik) {
    PendudukModel pendudukModel = pendudukDAO.selectPenduduk(nik);
    //menyimpan nik penduduk
    String nikMati = nik;
    if(pendudukModel != null) {
        //mengambil data semua penduduk yang termasuk ke dalam anggota suatu keluarga
        List<PendudukModel> pendudukModels = pendudukDAO.selectAnggotaKeluarga(pendudukModel.getId_keluarga());
        int counter = 1;
        for(PendudukModel x : pendudukModels) {
            if(x.getIs_wafat().equals("1")) {
                counter++;
            }
        }
        if(counter >= pendudukModels.size()) {
            keluargaDAO.updateIsTidakBerlaku("1",pendudukModel.getId_keluarga());
        }
        pendudukDAO.updateIsWafat("1",pendudukModel.getId());
        model.addAttribute("confirmation", "Penduduk dengan NIK " + nikMati + " sudah tidak aktif");
        return "view-tambah" ;
    } else{
        model.addAttribute("nik", nik);
        return "not-found-penduduk";
    }
}
```

Berikut tampilannya :

Jika penduduk masih hidup dan ternyata sudah wafat maka dapat dinonaktifkan dengan mengklik tombol nonaktifkan



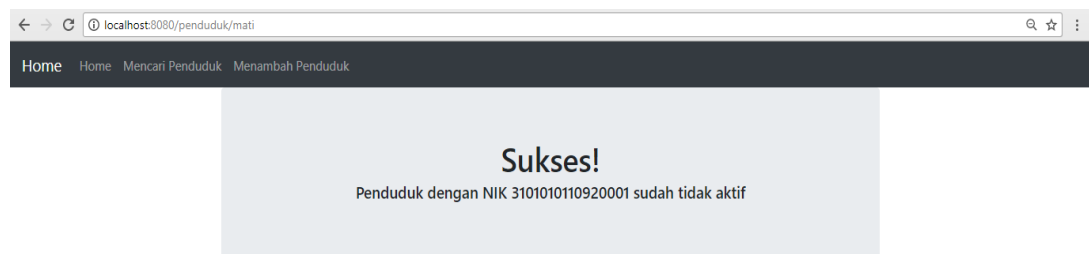
Home Home Mencari Penduduk Menambah Penduduk

Lihat Data Penduduk - 3101010110920001

NIK	3101010110920001
Nama	Teddy Sinaga S.Sos
Tempat/Tanggal Lahir	Pasuruan, 01 Oktober 1992
Alamat	Jr. Karel S. Tubun No. 82
RT/RW	033/052
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	O+
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	JURU MASAK
Kewarganegaraan	WNI
Status Kematian	Hidup

[Nonaktifkan](#)

Jika berhasil maka akan ditampilkan pesan sukses



Home Home Mencari Penduduk Menambah Penduduk

Sukses!

Penduduk dengan NIK 3101010110920001 sudah tidak aktif

Status kematian penduduk berubah menjadi wafat



Home Home Mencari Penduduk Menambah Penduduk

Lihat Data Penduduk - 3101010110920001

NIK	3101010110920001
Nama	Teddy Sinaga S.Sos
Tempat/Tanggal Lahir	Pasuruan, 01 Oktober 1992
Alamat	Jr. Karel S. Tubun No. 82
RT/RW	033/052
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	O+
Agama	Islam
Status Perkawinan	Kawin
Pekerjaan	JURU MASAK
Kewarganegaraan	WNI
Status Kematian	Wafat

8. MENAMPILKAN 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}

```
<form action="/penduduk/cari" method="get">
```

Response Page:

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

Pada controller dibuat sebuah method cariPenduduk yang akan mencari daftar data semua penduduk yang ada di database berdasarkan daerah tempat tinggalnya di kota, kecamatan dan kelurahan mana. Pada method ini juga akan dipanggil method selectPendudukByKelurahan di PendudukService yang akan memberikan data semua penduduk dengan id kelurahan yang sama.

```
@RequestMapping("/penduduk/cari")
private String cariPenduduk(Model model,
    @RequestParam(value = "id_kota", required = false) String id_kota,
    @RequestParam(value = "id_kecamatan", required = false) String id_kecamatan,
    @RequestParam(value = "id_kelurahan", required = false) String id_kelurahan
){
    //mengambil seluruh daftar kota
    List <KotaModel> kota = kotaDAO.selectAllKota();
    model.addAttribute("kota", kota);
    if (id_kota != null)
    {
        //mengambil daftar kecamatan berdasarkan id kota
        List <KecamatanModel> kecamatan = kecamatanDAO.selectKecamatanByKota(id_kota);
        model.addAttribute("id_kota", id_kota);
        model.addAttribute("kecamatan", kecamatan);
    }
    else
    {
        model.addAttribute("id_kota", null);
    }
    if (id_kecamatan != null)
    {
        //mengambil daftar kelurahan berdasarkan id kecamatan
        List <KelurahanModel> kelurahan = kelurahanDAO.selectKelurahanByKecamatan(id_kecamatan);
        model.addAttribute("kelurahan", kelurahan);
        model.addAttribute("id_kecamatan", id_kecamatan);
    }
    else
    {
        model.addAttribute("id_kecamatan", id_kecamatan);
    }
    if (id_kota != null && id_kecamatan != null && id_kelurahan != null)
    {
        //mengambil daftar penduduk berdasarkan kelurahan
        List <PendudukModel> penduduk = pendudukDAO.selectPendudukByKelurahan(id_kelurahan);
        model.addAttribute("penduduk", penduduk);
        return "sukses-cari";
    }
    return "cari-penduduk";
}
```

Berikut tampilannya :

← → 🔄 localhost:8080/penduduk/cari?id_kota=1 🔍 ☆ ⋮

Mencari Penduduk

Kota

KABUPATEN KEPULAUAN SERIBU ▼

Kecamatan

Pilih Kecamatan ▼

Cari

← → 🔄 localhost:8080/penduduk/cari?id_kota=1&id_kecamatan=1 🔍 ☆ ⋮

Mencari Penduduk

Kota

KABUPATEN KEPULAUAN SERIBU ▼

Kecamatan

KEPULAUAN SERIBU SELATAN ▼

Kelurahan/Desa

Pilih Kelurahan ▼

Cari

← → 🔄 localhost:8080/penduduk/cari?id_kota=1&id_kecamatan=1 🔍 ☆ ⋮

Home Home Mencari Penduduk Menambah Penduduk

Mencari Penduduk

Kota

KABUPATEN KEPULAUAN SERIBU ▼

Kecamatan

KEPULAUAN SERIBU SELATAN ▼

Kelurahan/Desa

PULAU UNTUNG JAWA ▼

Cari

Hasilnya akan ditampilkan dalam bentuk tabel dalam datatables

← → ↻ local:8080/penduduk/cari?id_kota=1&id_kecamatan=1&id_kelurahan=3&action=

Home Home Mencari Penduduk Menambah Penduduk

Daftar Penduduk

Show 10 entries Search:

NIK	Nama	Jenis Kelamin
3101010101690001	Ridwan Hasim Marpaung	Laki-laki
3101010101800001	Laksana Saragih	Laki-laki
3101010102930001	Erik Ganep Budiman	Laki-laki
3101010102940001	Umay Ozy Prasasta	Laki-laki
3101010103160001	Marsito Samosir	Laki-laki
3101010103660001	Mahesa Ardianto	Laki-laki
3101010104730001	Pranata Mansur	Laki-laki
3101010104820001	Mujur Sitompul	Laki-laki
3101010104890002	Saka Wibowo	Laki-laki
3101010104930001	Radit Prasetya S.Psi	Laki-laki

Showing 1 to 10 of 1,189 entries Previous 1 2 3 4 5 ... 119 Next

Validasi input dengan menambahkan required="true" pada setiap input

```
<div class="form-group">
<div th:if="{id_kecamatan != null}">
<label for="kelurahan" >Kelurahan/Desa</label> |
<select class="form-control" name = "id_kelurahan" required="{true}">
<option value="" th:disabled="{disabled}" th:selected="{selected}">Pilih Kelurahan</option>
<div th:each = "kelurahan: {kelurahan}">
<option th:value="{kelurahan.id}" th:text="{kelurahan.nama_kelurahan}" ></option>
</div>
</select>
</div>
</div>
```

Home

Mencari Penduduk

Kota

Pilih Kota

Cari

Please select an item in the list.

Home

Mencari Penduduk

Kota

KABUPATEN KEPULAUAN SERIBU

Kecamatan

Pilih Kecamatan

Cari

Please select an item in the list.

OPTIMASI DATABASE

Optimasi yang dapat dilakukan adalah dengan menambahkan primary key pada setiap tabel

1. PENDUDUK
Menjadikan nik sebagai primary key
2. KELUARGA
Menjadikan nomor_kk sebagai primary key
3. KOTA
Menjadikan kode_kota sebagai primary key
4. KECAMATAN
Menjadikan kode_kecamatan sebagai primary key
5. KELURAHAN
Menjadikan kode_kelurahan sebagai primary key

