

## TUGAS 1 APAP WRITEUP

Pada tugas 1 kali ini setelah melihat *data dictionary* yang diberikan saya memutuskan untuk membuat 5 model yaitu Penduduk, Keluarga, Kelurahan, Kecamatan, dan Kota.

Penduduk merupakan model yang menyimpan informasi-informasi yang dimiliki oleh seorang penduduk. Seperti nik, nama, tempat lahir, dll. Serta seorang Penduduk pasti tepat memiliki satu keluarga.

```
@Data
@AllArgsConstructor
public class PendudukModel
{
    private String nik;
    @NotEmpty
    private String nama;
    @NotEmpty
    private String tempat_lahir;
    private Date tanggal_lahir;
    private int jenis_kelamin;
    @NotEmpty
    private String agama;
    private KeluargaModel keluarga;
    @NotNull @NotEmpty
    private String pekerjaan;
    private String golongan_darah;
    private String status_perkawinan;
    private String status_dalam_keluarga;
    private int is_wni;
    private int is_wafat;

    public PendudukModel()
    {
        this.is_wni = 1;
        this.is_wafat = 0;
    }
}
```

Saya juga membuat konstruktor kosong yang mana jika suatu penduduk baru dibuat, maka dia memiliki default value untuk kewarganegaraan adalah WNI dan status kematiannya adalah HIDUP.

Sama halnya dengan Penduduk, Keluarga serta Kelurahan, Kecamatan dan Kota menyimpan data yang dimilikinya. Untuk sebuah Keluarga, mereka pasti tinggal tepat di satu Kelurahan yang mana Kelurahan itu tepat berada di satu Kecamatan dan begitu pula dengan Kecamatan yang berada tepat di satu Kota.

```

@Data
@AllArgsConstructor
public class KeluargaModel
{
    private String nomor_kk;
    @NotEmpty
    private String alamat;
    @NotEmpty @Size(min = 3, max = 3)
    private String rt;
    @NotEmpty @Size(min = 3, max = 3)
    private String rw;
    private KelurahanModel kelurahan;
    private int is_tidak_berlaku;
    private List<PendudukModel> anggotaKeluarga;

    public KeluargaModel() { this.is_tidak_berlaku = 0; }
}

```

Pada Keluarga saya membuat konstruktor kosong yang mana jika suatu keluarga baru dibuat maka masa berlakunya akan aktif.

```

@Data
public class KelurahanModel
{
    private String kode_kelurahan;
    private KecamatanModel kecamatan;
    private String nama_kelurahan;
    private String kode_pos;
}

```

```

@Data
public class KecamatanModel
{
    private String kode_kecamatan;
    private String nama_kecamatan;
    private KotaModel kota;
}

```

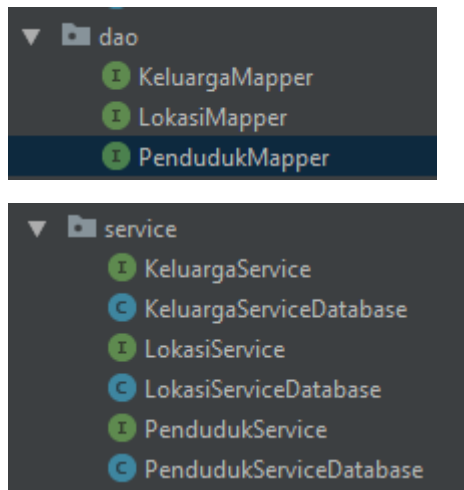
```

@Data
public class KotaModel
{
    private String kode_kota;
    private String nama_kota;
}

```

Selanjutnya saya menentukan terkait mapper apa saja yang akan saya buat. Saya memutuskan untuk membuat tiga (3) mapper. Yang pertama adalah Penduduk, kemudian Keluarga dan yang terakhir adalah Lokasi. Mapper Lokasi ini khusus untuk

melakukan query tersendiri pada model Kelurahan, Kecamatan dan Kota. Terkait Service dan Service Database, saya menyesuakannya dengan Mapper yang saya miliki.

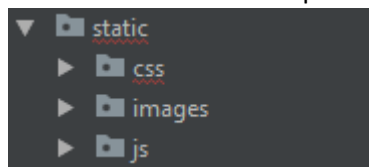


## FITUR

Pada bagian ini saya akan menjelaskan tahapan saya membuat fitur-fitur yang ada di soal

### 1. Tampilkan Penduduk dari NIK

Pertama-tama saya *download* terlebih dahulu terkait bootstrap dan jquery untuk membantu dalam pembuatan tampilan pada folder static.



Selanjutnya saya membuat PendudukController, yang nantinya semua controlling terkait penduduk akan dilakukan disana. Untuk RequestMapping ke index, saya buat di PendudukController.

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

Setelah itu saya membuat view index seperti yang terdapat pada soal dimana terdapat dua input pada index untuk mengerjakan fitur 1 dan fitur 2.

```

<!-- Page Content -->
<div class="container">
  <div class="row">
    <div>
      <h1 th:text="'Sistem Informasi Kependudukan Online Provinsi DKI Jakarta'">System</h1>
      <p th:text="'Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta'">Welcome</p>
      <hr />
      <form method="get" action="/penduduk">
        <div class="form-group col-md-5">
          <p class="search" th:text="'Lihat Data Penduduk Berdasarkan NIK'">Search by NIK</p>
          <label for="nik">Masukkan Nomor Induk Kependudukan</label>
          <input type="text" id="nik" class="form-control" name="nik" />
          <br />
          <button type="submit" class="btn btn-default" th:text="'Lihat'">Find</button>
        </div>
      </form>
      <hr />
      <form method="get" action="/keluarga">
        <div class="form-group col-md-5">
          <p class="search" th:text="'Lihat Data Penduduk Berdasarkan NKK'">Search by NKK</p>
          <label for="nkk" th:text="'Masukkan Nomor Kartu Keluarga'">NIK</label>
          <input type="text" id="nkk" class="form-control" name="nkk" />
          <br />
          <button type="submit" class="btn btn-default">Lihat</button>
        </div>
      </form>
    </div>
  </div>
</div>

```

## Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Lihat Data Penduduk Berdasarkan NIK

Masukkan Nomor Induk Kependudukan

Lihat

Lihat Data Penduduk Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

Lihat

Pada view Index terdapat satu form yang digunakan pada fitur 1. Nantinya form tersebut akan menerima input yaitu nik dan selanjutnya akan melakukan request ke “/penduduk” dengan method get.

```

<form method="get" action="/penduduk">
  <div class="form-group col-md-5">
    <p class="search" th:text="'Lihat Data Penduduk Berdasarkan NIK'">Search by NIK</p>
    <label for="nik">Masukkan Nomor Induk Kependudukan</label>
    <input type="text" id="nik" class="form-control" name="nik" />
    <br />
    <button type="submit" class="btn btn-default" th:text="'Lihat'">Find</button>
  </div>
</form>

```

## Lihat Data Penduduk Berdasarkan NIK

### Masukkan Nomor Induk Kependudukan

Lihat

Kemudian saya menambah method pada controller yang menerima request mapping dengan method get. Saya membuat query pada PendudukMapper untuk mendapatkan seluruh informasi yang dibutuhkan untuk menampilkan data penduduk nik tertentu.

```
@Select("SELECT * FROM PENDUDUK WHERE nik = #{nik}")
@Results(value = {
    @Result(property = "keluarga", column = "id_keluarga",
        javaType = KeluargaModel.class,
        one = @One(select = "selectKeluarga"))
})
PendudukModel selectPenduduk(@Param("nik") String nik);
```

```
@Select("SELECT * FROM KELUARGA WHERE id = #{id_keluarga}")
@Results(value = {
    @Result(property = "kelurahan", column = "id_kelurahan",
        javaType = KelurahanModel.class,
        one = @One(select = "selectKelurahan"))
})
KeluargaModel selectKeluarga(@Param("id_keluarga") String id_keluarga);
```

```
@Select("SELECT * FROM KELURAHAN WHERE id = #{id_kelurahan}")
@Results(value = {
    @Result(property = "kecamatan", column = "id_kecamatan",
        javaType = KelurahanModel.class,
        one = @One(select = "selectKecamatan"))
})
KelurahanModel selectKelurahan(@Param("id_kelurahan") String id_kelurahan);
```

```
@Select("SELECT * FROM KECAMATAN WHERE id = #{id_kecamatan}")
@Results(value = {
    @Result(property = "kota", column = "id_kota",
        javaType = KelurahanModel.class,
        one = @One(select = "selectKota"))
})
KecamatanModel selectKecamatan(@Param("id_kecamatan") String id_kecamatan);
```

```
@Select("SELECT * FROM KOTA WHERE id = #{id_kota}")
KotaModel selectKota(@Param("id_kota") String id_kota);
```

Kemudian saya juga menambahkan method pada service dan juga service databasenya.

```
PendudukModel selectPenduduk(String nik);
```

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

Lalu selanjutnya saya membuat tampilan pada viewnya yang akan menampilkan informasi terkait penduduk yang dicari.

```
<!-- Page Content -->
<div class="container">
  <div class="row">
    <div>
      <div th:if="${wafat}==true" success">
        <h2 th:text="'Sukses!'"></h2>
        <p th:text="'Penduduk dengan NIK ' + ${penduduk.nik} + ' sudah tidak aktif'"></p>
        <hr />
      </div>
      <h5 th:text="'Lihat Data Penduduk - ' + ${penduduk.nik} + 'Penduduk Dilihat'"></h5>
      <table class="penduduk">
        <tbody>
          <tr>
            <td class="jenis-data" th:text="NIK">NIK</td>
            <td th:text="${penduduk.nik}">NIK</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="Nama">NAMA</td>
            <td th:text="${penduduk.nama}">NAMA</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="Tempat/Tanggal Lahir">TTL</td>
            <td th:text="${penduduk.tempat_lahir} + ', ' + ${dates.format(penduduk.tanggal_lahir, 'dd MMM yyyy')}>TEMPAT, TANGGAL LAHIR</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="Alamat">ALAMAT</td>
            <td th:text="${penduduk.keluarga.alamat}">ALAMAT</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

## 2. Tampilkan Keluarga dari NKK

Pada fitur ini tidak jauh berbeda dengan fitur 1, melalui page index, terdapat form untuk melakukan request ke "/keluarga". Namun pada fitur ini, listanggotakeluarga yang terdapat pada model keluarga akan digunakan karena anggota keluarga nantinya akan ditampilkan pada fitur ini.

```
<form method="get" action="/keluarga">
  <div class="form-group col-md-5">
    <p class="search" th:text="'Lihat Data Penduduk Berdasarkan NKK'">Search by NKK</p>
    <label for="nkk" th:text="'Masukkan Nomor Kartu Keluarga'">NIK</label>
    <input type="text" id="nkk" class="form-control" name="nkk" />
    <br />
    <button type="submit" class="btn btn-default">Lihat</button>
  </div>
</form>
```

```
@GetMapping("/keluarga")
public String keluargaSubmit(Model model, @RequestParam(value = "nkk", required = true)String nkk)
{
  if(nkk.equals(""))
  {
    return "index";
  }
  KeluargaModel keluarga = keluargaDAO.selectKeluarga(nkk);

  if (keluarga != null)
  {
    model.addAttribute( attributeName: "keluarga", keluarga);
    return "view-keluarga";
  }
  model.addAttribute( attributeName: "nkk", nkk);
  model.addAttribute( attributeName: "model", attributeValue: "keluarga");
  return "not-found";
}
```

```

<!-- Page Content -->
<div class="container">
  <div class="row">
    <div class="col-md-6">
      <h5 th:text="'Lihat Data Keluarga - ' + ${keluarga.nomor_kk}">Keluarga Dilihat</h5>
      <table class="keluarga">
        <tbody>
          <tr>
            <td class="jenis-data" th:text="'NKK'">NKK</td>
            <td th:text="${keluarga.nomor_kk}">NKK</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="'Alamat'">ALAMAT</td>
            <td th:text="${keluarga.alamat}">ALAMAT</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="'RT/RW'">RT/RW</td>
            <td th:text="${keluarga.rt} + '/' + ${keluarga.rw}">RT/RW</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="'Kelurahan/Desa'">KELURAHAN</td>
            <td th:text="${keluarga.kelurahan.nama_kelurahan}">KELURAHAN</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="'Kecamatan'"></td>
            <td th:text="${keluarga.kelurahan.kecamatan.nama_kecamatan}">KECAMATAN</td>
          </tr>
          <tr>
            <td class="jenis-data" th:text="'Kota'"></td>
            <td th:text="${keluarga.kelurahan.kecamatan.kota.nama_kota}">KOTA</td>
          </tr>
          <td th:text="${keluarga.kelurahan.kecamatan.kota.nama_kota}">KOTA</td>
        </tbody>
      </table>
    </div>
    <div class="col-md-12 col-md-offset-6">
      <hr />
      <table class="table table-bordered">
        <thead>
          <tr>
            <th th:text="'No.'">NOMOR</th>
            <th th:text="'Nama Lengkap'">NAMA LENGKAP</th>
            <th th:text="'NIK'">NIK</th>
            <th th:text="'Jenis Kelamin'">JENIS KELAMIN</th>
            <th th:text="'Tempat Lahir'">TEMPAT LAHIR</th>
            <th th:text="'Tanggal Lahir'">TANGGAL LAHIR</th>
            <th th:text="'Agama'">AGAMA</th>
            <th th:text="'Pekerjaan'">PEKERJAAN</th>
            <th th:text="'Status Perkawinan'">STATUS PERKAWINAN</th>
            <th th:text="'Status dalam Keluarga'">STATUS DALAM KELUARGA</th>
            <th th:text="'Kewarganegaraan'">KEWARGANEGARAAN</th>
          </tr>
        </thead>
        <tbody th:each="penduduk, iterationStatus: ${keluarga.anggotaKeluarga}">
          <tr>
            <td th:text="${iterationStatus.index} + 1">INDEX</td>
            <td th:text="${penduduk.nama}">NAMA LENGKAP</td>
            <td th:text="${penduduk.nik}">NIK</td>
            <td th:text="(${penduduk.jenis_kelamin} == 0) ? 'Pria' : 'Wanita'">JENIS KELAMIN</td>
            <td th:text="${penduduk.tempat_lahir}">TEMPAT LAHIR</td>
            <td th:text="${#dates.format(penduduk.tanggal_lahir, 'dd-MM-YYYY')}">TANGGAL LAHIR</td>
            <td th:text="${penduduk.agama}">AGAMA</td>
            <td th:text="${penduduk.pekerjaan}">PEKERJAAN</td>
            <td th:text="${penduduk.status_perkawinan}">STATUS PERKAWINAN</td>
            <td th:text="${penduduk.status_dalam_keluarga}">STATUS DALAM KELUARGA</td>
            <td th:text="${penduduk.status_perkawinan}">STATUS PERKAWINAN</td>
            <td th:text="${penduduk.status_dalam_keluarga}">STATUS DALAM KELUARGA</td>
            <td th:text="(${penduduk.is_wni} == 1) ? 'WNI' : 'WNA'">KEWARGANEGARAAN</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>

```

```

@Select("SELECT * FROM KELUARGA WHERE nomor_kk = #{nomor_kk}")
@Results(value = {
    @Result(property = "anggotaKeluarga", column = "id",
        javaType = List.class,
        many = @Many(select = "selectAnggotaKeluarga")),
    @Result(property = "kelurahan", column = "id_kelurahan",
        javaType = KelurahanModel.class,
        one = @One(select = "selectKelurahan"))
})
KeluargaModel selectKeluarga(@Param("nomor_kk") String nomor_kk);

@Select("SELECT * FROM KELURAHAN WHERE id = #{id_kelurahan}")
@Results(value = {
    @Result(property = "kecamatan", column = "id_kecamatan",
        javaType = KelurahanModel.class,
        one = @One(select = "selectKecamatan"))
})
KelurahanModel selectKelurahan(@Param("id_kelurahan") String id_kelurahan);

@Select("SELECT * FROM KECAMATAN WHERE id = #{id_kecamatan}")
@Results(value = {
    @Result(property = "kota", column = "id_kota",
        javaType = KelurahanModel.class,
        one = @One(select = "selectKota"))
})
KecamatanModel selectKecamatan(@Param("id_kecamatan") String id_kecamatan);

@Select("SELECT * FROM KOTA WHERE id = #{id_kota}")
KotaModel selectKota(@Param("id_kota") String id_kota);

@Select("SELECT * FROM PENDUDUK WHERE id_keluarga = #{id_keluarga}")
List<PendudukModel> selectAnggotaKeluarga(@Param("id_keluarga") String id_keluarga);

KeluargaModel selectKeluarga(String nkk);

@Override
public KeluargaModel selectKeluarga(String nkk) { return keluargaDAO.selectKeluarga(nkk); }

```

### 3. Menambahkan Penduduk Baru

Pada fitur ini pertama-tama saya membuat method yang menerima request mapping get pada controller dan juga tampilan saat user mengakses halamannya.

```

@GetMapping("/penduduk/tambah")
public String pendudukAdd(Model model)
{
    model.addAttribute("attributeName: penduduk", new PendudukModel());
    return "penduduk-form";
}

```



```

<!-- Page Content -->
<div class="container">
  <div class="row">
    <div class="col-md-10">
      <div class="penduduk-add">
        <form action="/penduduk/tambah" th:object="${penduduk}" method="post">
          <h4 th:text="'Tambah Penduduk'"></h4>
          <label th:text="'Masukkan Nama'">NAMA</label>
          <input type="text" class="form-control" name="nama" th:field="*{nama}" />

          <label th:text="'Tempat Lahir'">TEMPAT LAHIR</label>
          <input type="text" class="form-control" name="tempat_lahir" th:field="*{tempat_lahir}" />

          <label th:text="'Tanggal Lahir'">TANGGAL LAHIR</label>
          <input type="date" class="form-control" name="tggl_lahir" />

          <label th:text="'Jenis Kelamin'">JENIS KELAMIN</label>
          <select class="form-control" name="golongan_darah" th:field="*{jenis_kelamin}">
            <option th:value="'0'" th:text="'Pria'"></option>
            <option th:value="'1'" th:text="'Wanita'"></option>
          </select>

          <label th:text="'Golongan Darah'">GOLONGAN DARAH</label>
          <select class="form-control" name="golongan_darah" th:field="*{golongan_darah}">
            <option th:value="'A+' " th:text="'A+' "></option>
            <option th:value="'A-' " th:text="'A-' "></option>
            <option th:value="'B+' " th:text="'B+' "></option>
            <option th:value="'B-' " th:text="'B-' "></option>
            <option th:value="'AB+' " th:text="'AB+' "></option>
            <option th:value="'AB-' " th:text="'AB-' "></option>
            <option th:value="'O+' " th:text="'O+' "></option>
            <option th:value="'O-' " th:text="'O-' "></option>
          </select>

          <label th:text="'Agama'">AGAMA</label>
          <input type="text" class="form-control" name="agama" th:field="*{agama}" />

          <label th:text="'Status Perkawinan'">STATUS PERKAWINAN</label>
          <select class="form-control" name="status_perkawinan" th:field="*{status_perkawinan}">
            <option th:value="'Kawin'" th:text="'Kawin'"></option>
            <option th:value="'Belum Kawin'" th:text="'Belum Kawin'"></option>
          </select>

          <label th:text="'Pekerjaan'">PEKERJAAN</label>
          <input type="text" class="form-control" name="pekerjaan" th:field="*{pekerjaan}" />

          <label th:text="'Kewarganegaraan'">KEWARGANEGERAAN</label>
          <select class="form-control" name="status_kewarganegaraan" th:field="*{is_wni}">
            <option th:value="'0'" th:text="'WNA'"></option>
            <option th:value="'1'" th:text="'WNI'"></option>
          </select>

          <label th:text="'Status Kematian'">STATUS KEMATIAN</label>
          <select class="form-control" name="status_kematian" th:field="*{is_wafat}">
            <option th:value="'0'" th:text="'Hidup'"></option>
            <option th:value="'1'" th:text="'Wafat'"></option>
          </select>

          <label th:text="'Id Keluarga'">ID KELUARGA</label>
          <input type="text" class="form-control" name="id_keluarga" />

          <label th:text="'Status Dalam Keluarga'">STATUS DALAM KELUARGA</label>
          <select class="form-control" name="status_dalam_keluarga" th:field="*{status_dalam_keluarga}">
            <option th:value="'Kepala Keluarga'" th:text="'Kepala Keluarga'"></option>
            <option th:value="'Istri'" th:text="'Istri'"></option>
            <option th:value="'Anak'" th:text="'Anak'"></option>
          </select>

          <br />
          <button type="submit" class="btn btn-primary">Simpan</button>
        </form>
      </div>
    </div>
  </div>
</div>

```

Saya menyimpan beberapa value dalam input dan di distribusikan kedalam objek penduduk. Sisanya saya menerima dalam suatu parameter. Seperti tanggal dan id\_keluarga.

```
@PostMapping("/penduduk/tambah")
public String pendudukAddSubmit(@RequestParam(value = "tgl_lahir", required = true) String tgl_lahir,
                                @RequestParam(value = "id_keluarga", required = true) String id_keluarga,
                                @Valid PendudukModel penduduk,
                                BindingResult result,
                                Model model) throws Exception
{
    if(result.hasErrors() || tgl_lahir.equals("") || id_keluarga.equals(""))
    {
        model.addAttribute( attributeName: "penduduk", penduduk);
        return "penduduk-form";
    }
    pendudukDAO.addPenduduk(penduduk, id_keluarga, tgl_lahir);
    model.addAttribute( attributeName: "nik", penduduk.getNik());
    model.addAttribute( attributeName: "perintah", attributeValue: "insert");
    model.addAttribute( attributeName: "model", attributeValue: "penduduk");
    return "success";
}
```

Kemudian saya membuat method di service addPenduduk yang di override pada service database yang nantinya akan memanggil method di mapper untuk melakukan query insert.

```
void addPenduduk(PendudukModel penduduk, String id_keluarga, String tgl_lahir) throws Exception;

@Override
public void addPenduduk(PendudukModel penduduk, String id_keluarga, String tgl_lahir) throws Exception {
    penduduk.setTanggal_lahir(setTanggal(tgl_lahir));
    penduduk.setNik(createNIK(penduduk, id_keluarga));
    pendudukDAO.addPenduduk(id_keluarga, penduduk);
}

@Insert("INSERT INTO PENDUDUK " +
        "(nik, nama, tempat_lahir, tanggal_lahir, jenis_kelamin, is_wni, id_keluarga, agama, pekerjaan, status_perkawinan, status_dalam_keluarga, golongan_darah, is_wafat) VALUES " +
        "(#{penduduk.nik}, #{penduduk.nama}, #{penduduk.tempat_lahir}, #{penduduk.tanggal_lahir}, #{penduduk.jenis_kelamin}, #{penduduk.is_wni}, #{id_keluarga}, #{penduduk.agama}, " +
        " #{penduduk.pekerjaan}, #{penduduk.status_perkawinan}, #{penduduk.status_dalam_keluarga}, #{penduduk.golongan_darah}, #{penduduk.is_wafat})")
void addPenduduk(@Param("id_keluarga") String id_keluarga, @Param("penduduk") PendudukModel penduduk);
```

Pada service database, saya membuat method untuk melakukan parse tanggal lahir agar menjadi Date. Serta saya juga membuat method untuk generate nik.

```
public Date setTanggal(String tanggal) throws Exception
{
    return new SimpleDateFormat( pattern: "yyyy-MM-dd").parse(tanggal);
}

public String createNIK(PendudukModel penduduk, String id_keluarga) {
    String nik = nikTmp(penduduk, id_keluarga);
    return generateNIK(nik);
}

public String nikTmp(PendudukModel penduduk, String id_keluarga)
{
    KeluargaModel keluarga = pendudukDAO.selectKeluarga(id_keluarga);

    String kodePKK = keluarga.getKelurahan().getKecamatan().getKode_kecamatan().substring(0,6);
    String kodeTglLahir = new SimpleDateFormat( pattern: "ddMMyy").format(penduduk.getTanggal_lahir());
    if(penduduk.getJenis_kelamin() == 1)
    {
        kodeTglLahir = "" + (Integer.parseInt(kodeTglLahir.substring(0, 1)) + 4) + kodeTglLahir.substring(1);
    }
    return kodePKK + kodeTglLahir;
}
```

```

public String generateNIK(String nik)
{
    String result = pendudukDAO.nikMax(nik + "%");
    if(result == null)
    {
        return nik + "0001";
    }
    return (Long.parseLong(result) + 1) + "";
}



```

#### 4. Menambahkan Keluarga Baru

Pada fitur ini suatu keluarga baru akan dibuat, dimana form akan meminta data-data yang harus dimiliki oleh suatu keluarga. Pada fitur ini saya melakukan beberapa perubahan dimana list kota kecamatan dan kelurahan akan berbentuk select option. Pada fitur ini dan fitur 6 saya menggunakan lokasimapper untuk mendapatkan data kota kecamatan dan kelurahan sesuai dengan yang dibutuhkan.

```

@GetMapping("/keluarga/tambah")
public String keluargaAdd(Model model)
{
    model.addAttribute( attributeName: "keluarga", new KeluargaModel());
    model.addAttribute( attributeName: "listKota", lokasiDAO.selectAllKotaForm());
    model.addAttribute( attributeName: "listKecamatan", lokasiDAO.selectAllKecamatanForm());
    model.addAttribute( attributeName: "listKelurahan", lokasiDAO.selectAllKelurahanForm());

    return "keluarga-form";
}

@PostMapping("/keluarga/tambah")
public String keluargaAddSubmit(@RequestParam("id_kelurahan") String id_kelurahan,
                                @RequestParam("id_kecamatan") String id_kecamatan,
                                @Valid KeluargaModel keluarga,
                                BindingResult result,
                                Model model)
{
    if(result.hasErrors() || id_kelurahan.equals("") || id_kecamatan.equals(""))
    {
        model.addAttribute( attributeName: "keluarga", keluarga);
        model.addAttribute( attributeName: "listKota", lokasiDAO.selectAllKotaForm());
        model.addAttribute( attributeName: "listKecamatan", lokasiDAO.selectAllKecamatanForm());
        model.addAttribute( attributeName: "listKelurahan", lokasiDAO.selectAllKelurahanForm());

        return "keluarga-form";
    }
    keluargaDAO.addKeluarga(keluarga, id_kelurahan);

    model.addAttribute( attributeName: "perintah", attributeValue: "insert");
    model.addAttribute( attributeName: "nomor_kk", keluarga.getNomor_kk());
    model.addAttribute( attributeName: "model", attributeValue: "keluarga");
    return "success";
}

```

```

<!-- Page Content -->
<div class="container">
  <div class="row">
    <div class="col-md-10">
      <div class="penduduk-add">
        <form action="/keluarga/tambah" th:object="${keluarga}" method="post">
          <h4 th:text="'Tambah Keluarga'"></h4>
          <label th:text="'Alamat'">ALAMAT</label>
          <input type="text" class="form-control" name="alamat" th:field="*{alamat}" />

          <label th:text="'RT'">RT</label>
          <input type="text" class="form-control" name="rt" th:field="*{rt}" />

          <label th:text="'RW'">RW</label>
          <input type="text" class="form-control" name="rw" th:field="*{rw}" />

          <label for="kota" th:text="'Kota'">KOTA</label>
          <select class="form-control" id="kota">
            <option th:value="" th:text="'Pilih Kota/Kabupaten'"></option>
            <div th:each="kota, iterationStatus: ${listKota}">
              <option th:id="${kota.kode_kota}" th:text="${kota.nama_kota}"></option>
            </div>
          </select>

          <label for="kecamatan" th:text="'Kecamatan'">KECAMATAN</label>
          <select class="form-control" id="kecamatan">
            <option th:value="" th:text="'Pilih Kecamatan'"></option>
            <div th:each="kecamatan, iterationStatus: ${listKecamatan}">
              <option th:id="${kecamatan.kota.kode_kota}" th:value="${kecamatan.kode_kecamatan}" th:text="${kecamatan.nama_kecamatan}"></option>
            </div>
          </select>

          <label for="kelurahan" th:text="'Kelurahan'">KELURAHAN</label>
          <select class="form-control" id="kelurahan" name="id_kelurahan">
            <option th:value="" th:text="'Pilih Kelurahan'"></option>
            <div th:each="kelurahan, iterationStatus: ${listKelurahan}">
              <option th:id="${kelurahan.kecamatan.kode_kecamatan}" th:value="${kelurahan.kode_kelurahan}" th:text="${kelurahan.nama_kelurahan}"></option>
            </div>
          </select>

          <br />
          <button type="submit" class="btn btn-primary">Simpan</button>
        </form>
      </div>
    </div>
  </div>
</div>

@Insert("INSERT INTO KELUARGA " +
        "(nomor_kk, alamat, rt, rw, id_kelurahan, is_tidak_berlaku) VALUE " +
        "(${keluarga.nomor_kk}, ${keluarga.alamat}, ${keluarga.rt}, ${keluarga.rw}, ${id_kelurahan}, ${keluarga.is_tidak_berlaku})")
void addKeluarga(@Param("id_kelurahan") String id_kelurahan, @Param("keluarga") KeluargaModel keluarga);

void addKeluarga(KeluargaModel keluarga, String id_kelurahan);

public String createNKK(String id_kelurahan) {
    String nkk = nkkTmp(id_kelurahan);
    return generateNKK(nkk);
}

```

```

public String nkkTmp(String id_kelurahan)
{
    String kodePKK = keluargaDAO.selectKelurahan(id_kelurahan).getKecamatan().getKode_kecamatan().substring(0, 6);
    String waktu = new SimpleDateFormat( "ddMMyy").format(new Date());
    return kodePKK + waktu;
}

public String generateNKK(String nkk)
{
    String result = keluargaDAO.nkkMax( nomor_kk: nkk + "%");
    if(result == null)
    {
        return nkk + "0001";
    }

    return (Long.parseLong(result) + 1) + "";
}

@Override
public void addKeluarga(KeluargaModel keluarga, String id_kelurahan) {
    keluarga.setNomor_kk(createNKK(id_kelurahan));
    keluargaDAO.addKeluarga(id_kelurahan, keluarga);
}

```

## 5. Mengubah Data Penduduk

Fitur ini bekerja tidak jauh berbeda dengan fitur menambahkan penduduk yang mana perbedaannya adalah form yang ditampilkan sudah terisi sesuai dengan data yang dimiliki oleh penduduk yang ingin diperbarui. Serta perbedaan yang lain adalah jika terdapat data yang mempengaruhi nik, yaitu id keluarga dan tanggal lahir. Maka nik akan diperbarui

```

@GetMapping("/penduduk/ubah/{nik}")
public String pendudukUpdate(@PathVariable(value = "nik", required = true) String nik,
                             Model model)
{
    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);

    if(penduduk == null)
    {
        model.addAttribute( attributeName: "nik", nik);
        model.addAttribute( attributeName: "model", attributeValue: "penduduk");
        return "not-found";
    }

    if(penduduk.getGolongan_darah().charAt(penduduk.getGolongan_darah().length() - 1) == '-')
    {
        penduduk.setGolongan_darah(penduduk.getGolongan_darah().substring(0, penduduk.getGolongan_darah().length() - 1) + "-");
    }
    model.addAttribute( attributeName: "penduduk", penduduk);

    String tggl_lahir = new SimpleDateFormat( pattern: "yyyy-MM-dd").format(penduduk.getTanggal_lahir());
    model.addAttribute( attributeName: "tggl_lahir", tggl_lahir);
    model.addAttribute( attributeName: "id_keluarga", pendudukDAO.selectIDKeluargaPenduduk(penduduk.getNik()));
    return "penduduk-form-update";
}

@PostMapping("/penduduk/ubah/{nik}")
public String pendudukUpdateSubmit(@RequestParam(value = "tggl_lahir", required = true) String tggl_lahir,
                                   @RequestParam(value = "id_keluarga", required = true) String id_keluarga,
                                   @Valid PendudukModel penduduk,
                                   BindingResult result,
                                   Model model) throws Exception
{
    if(result.hasErrors() || tggl_lahir.equals("") || id_keluarga.equals(""))
    {
        return new String( original: "redirect:/penduduk/ubah/" + penduduk.getNik());
    }
    String old_nik = penduduk.getNik();
    pendudukDAO.updatePenduduk(id_keluarga, old_nik, penduduk, tggl_lahir);

    model.addAttribute( attributeName: "nik", old_nik);
    model.addAttribute( attributeName: "perintah", attributeValue: "update");
    model.addAttribute( attributeName: "model", attributeValue: "penduduk");
    return "success";
}

```

```

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

<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-md-10">
            <div class="penduduk-add">
                <form th:action="/penduduk/ubah/" + ${penduduk.nik} th:object="${penduduk}" method="post">
                    <h4 th:text="'Ubah Penduduk'"></h4>
                    <label th:text="'NIK'">NIK</label>
                    <input type="text" class="form-control" th:field="*{nik}" readonly="readonly" />

                    <label th:text="'Masukkan Nama'">NAMA</label>
                    <input type="text" class="form-control" th:field="*{nama}" />

                    <label th:text="'Tempat Lahir'">TEMPAT LAHIR</label>
                    <input type="text" class="form-control" th:field="*{tempat_lahir}" />

                    <label th:text="'Tanggal Lahir'">TANGGAL LAHIR</label>
                    <input type="date" class="form-control" name="tggl_lahir" th:value="${tggl_lahir}" />

                    <label th:text="'Jenis Kelamin'">GOLONGAN DARAH</label>
                    <select class="form-control" th:field="*{jenis_kelamin}">
                        <option th:value="'0'" th:text="'Pria'"></option>
                        <option th:value="'1'" th:text="'Wanita'"></option>
                    </select>

                    <label th:text="'Golongan Darah'">GOLONGAN DARAH</label>
                    <select class="form-control" th:field="*{golongan_darah}">
                        <option th:value="'A+' th:text="'A+'></option>
                        <option th:value="'A-' th:text="'A-'></option>
                        <option th:value="'B+' th:text="'B+'></option>
                        <option th:value="'B-' th:text="'B-'></option>

```

```

        <option th:value="'0+'" th:text="'0+'"></option>
        <option th:value="'0-' th:text="'0-'"></option>
    </select>

    <label th:text="'Agama'">AGAMA</label>
    <input type="text" class="form-control" th:field="*{agama}" />

    <label th:text="'Status Perkawinan'">STATUS PERKAWINAN</label>
    <select class="form-control" th:field="*{status_perkawinan}">
        <option th:value="'Kawin'" th:text="'Kawin'"></option>
        <option th:value="'Belum Kawin'" th:text="'Belum Kawin'"></option>
    </select>

    <label th:text="'Pekerjaan'">PEKERJAAN</label>
    <input type="text" class="form-control" th:field="*{pekerjaan}" />

    <label th:text="'Kewarganegaraan'">KEWARGANEGERAAN</label>
    <select class="form-control" th:field="*{is_wni}">
        <option th:value="'0'" th:text="'WNA'"></option>
        <option th:value="'1'" th:text="'WNI'"></option>
    </select>

    <label th:text="'Status Kematian'">STATUS KEMATIAN</label>
    <select class="form-control" th:field="*{is_wafat}">
        <option th:value="'0'" th:text="'Hidup'"></option>
        <option th:value="'1'" th:text="'Wafat'"></option>
    </select>

    <label th:text="'Id Keluarga'">ID KELUARGA</label>
    <input type="text" class="form-control" name="id_keluarga" th:value="${id_keluarga}" />

    <label th:text="'Status Dalam Keluarga'">STATUS DALAM KELUARGA</label>
    <select class="form-control" th:field="*{status_dalam_keluarga}">
        <option th:value="'Kepala Keluarga'" th:text="'Kepala Keluarga'"></option>
        <option th:value="'Istri'" th:text="'Istri'"></option>

        <label th:text="'Status Dalam Keluarga'">STATUS DALAM KELUARGA</label>
        <select class="form-control" th:field="*{status_dalam_keluarga}">
            <option th:value="'Kepala Keluarga'" th:text="'Kepala Keluarga'"></option>
            <option th:value="'Istri'" th:text="'Istri'"></option>
            <option th:value="'Anak'" th:text="'Anak'"></option>
        </select>

        <br />
        <button type="submit" class="btn btn-primary">Simpan</button>
    </form>

    void updatePenduduk(String id_keluarga, String old_nik, PendudukModel penduduk, String tggl_lahir) throws Exception;

    @Override
    public void updatePenduduk(String id_keluarga, String old_nik, PendudukModel penduduk, String tggl_lahir) throws Exception {
        penduduk.setTanggal_lahir(setTanggal(tggl_lahir));
        penduduk.setNik(updateNIK(penduduk, id_keluarga));
        pendudukDAO.updatePenduduk(id_keluarga, old_nik, penduduk);
    }

```



```

public String updateNIK(PendudukModel penduduk, String id_keluarga) {
    String nik = nikTmp(penduduk, id_keluarga);

    if (penduduk.getNik().substring(0, 12).equalsIgnoreCase(nik))
    {
        return penduduk.getNik();
    }

    return generateNIK(nik);
}

@Update("UPDATE PENDUDUK " +
        "SET nik = #{penduduk.nik}, nama = #{penduduk.nama}, tempat_lahir = #{penduduk.tempat_lahir}, " +
        "tanggal_lahir = #{penduduk.tanggal_lahir}, jenis_kelamin = #{penduduk.jenis_kelamin}, " +
        "is_wni = #{penduduk.is_wni}, id_keluarga = #{id_keluarga}, agama = #{penduduk.agama}, " +
        "pekerjaan = #{penduduk.pekerjaan}, status_perkawinan = #{penduduk.status_perkawinan}, " +
        "status_dalam_keluarga = #{penduduk.status_dalam_keluarga}, golongan_darah = #{penduduk.golongan_darah}, " +
        "is_wafat = #{penduduk.is_wafat} " +
        "WHERE nik = ${old_nik}")
void updatePenduduk(@Param("id_keluarga") String id_keluarga, @Param("old_nik") String old_nik, @Param("penduduk") PendudukModel penduduk);

```

## 6. Mengubah Data Keluarga

Tidak jauh berbeda saat membuat keluarga baru, namun pada fitur ini data yang sudah dimiliki suatu keluarga akan di tampilkan pada form yang akan diubah.

```

@GetMapping("/keluarga/ubah/{nkk}")
public String keluargaUpdate(@PathVariable(value = "nkk", required = true) String nkk,
                             Model model)
{
    KeluargaModel keluarga = keluargaDAO.selectKeluarga(nkk);
    if(keluarga == null)
    {
        model.addAttribute("nkk", nkk);
        model.addAttribute("model", "keluarga");
        return "not-found";
    }

    model.addAttribute("keluarga", keluarga);

    model.addAttribute("listKota", lokasiDAO.selectAllKotaForm());
    model.addAttribute("listKecamatan", lokasiDAO.selectAllKecamatanForm());
    model.addAttribute("listKelurahan", lokasiDAO.selectAllKelurahanForm());

    model.addAttribute("id_kelurahan", lokasiDAO.getIdKelurahan(keluarga.getKelurahan().getNama_kelurahan()));
    model.addAttribute("id_kecamatan", lokasiDAO.getIdKecamatan(keluarga.getKelurahan().getKecamatan().getNama_kecamatan()));
    model.addAttribute("id_kota", lokasiDAO.getIdKota(keluarga.getKelurahan().getKecamatan().getKota().getNama_kota()));

    return "keluarga-form-update";
}

@PostMapping("/keluarga/ubah/{nkk}")
public String keluargaUpdateSubmit(@RequestParam("id_kelurahan") String id_kelurahan,
                                   @RequestParam("id_kecamatan") String id_kecamatan,
                                   @Valid KeluargaModel keluarga,
                                   BindingResult result, Model model) throws Exception
{
    if(result.hasErrors())
    {
        return new String("redirect:/keluarga/ubah/" + keluarga.getNomor_kk());
    }

    String oldNKK = keluarga.getNomor_kk();
    KeluargaModel oldKeluargaData = keluargaDAO.selectKeluarga(oldNKK);
    String oldIdKecamatan = lokasiDAO.getIdKecamatan(oldKeluargaData.getKelurahan().getKecamatan().getNama_kecamatan());
    keluargaDAO.updateKeluarga(id_kelurahan, keluarga, oldNKK);

    if(!id_kecamatan.equals(oldIdKecamatan))
    {
        ArrayList<PendudukModel> anggotaKeluarga = new ArrayList<>(oldKeluargaData.getAnggotaKeluarga());
        if (anggotaKeluarga.size() > 0)
        {
            String id_keluarga = pendudukDAO.selectIDKeluargaPenduduk(anggotaKeluarga.get(0).getNik());

            for (PendudukModel penduduk: anggotaKeluarga)
            {
                pendudukDAO.updatePenduduk(id_keluarga, penduduk.getNik(), penduduk, new SimpleDateFormat("yyyy-MM-dd").format(penduduk.getTanggal_lahir()));
            }
        }
    }

    model.addAttribute("perintah", "update");
    model.addAttribute("nomor_kk", oldNKK);
    model.addAttribute("model", "keluarga");
    return "success";
}

@Update("UPDATE KELUARGA " +
        "SET nomor_kk = #{keluarga.nomor_kk}, alamat = #{keluarga.alamat}, rt = #{keluarga.rt}, rw = #{keluarga.rw}, id_kelurahan = #{id_kelurahan} " +
        "WHERE nomor_kk = ${oldNKK}")
void updateKeluarga(@Param("id_kelurahan") String id_kelurahan, @Param("keluarga") KeluargaModel keluarga, @Param("oldNKK") String oldNKK);

void updateKeluarga(String id_kelurahan, KeluargaModel keluarga, String oldNKK);

```



```

public String updateNKK(String id_kelurahan, String oldNKK) {
    String nkk = nkkTmp(id_kelurahan);
    log.info("Old NKK: {}", oldNKK);
    if (oldNKK.substring(0, 12).equalsIgnoreCase(nkk))
    {
        return oldNKK;
    }
    return generateNKK(nkk);
}

@Override
public void updateKeluarga(String id_kelurahan, KeluargaModel keluarga, String oldNKK) {
    keluarga.setNomor_kk(updateNKK(id_kelurahan, oldNKK));
    keluargaDAO.updateKeluarga(id_kelurahan, keluarga, oldNKK);
}

```

## 7. Mengubah Status Kematian

Pada fitur ini saat sedang menampilkan halaman penduduk, jika penduduk masih hidup maka akan terdapat tombol untuk mengubah status kematian penduduk tersebut yang selanjutnya jika anggota keluarga lainnya sudah meninggal, maka status kartu keluarga akan menjadi tidak aktif. Selanjutnya mapping tersebut akan di redirect ke halaman tampilan penduduk.

```

@PostMapping("/penduduk/mati")
public String updateWafat(@RequestParam(value = "nik", required = true) String nik,
    RedirectAttributes redirectAttributes)
{
    pendudukDAO.updateWafat(nik);
    keluargaDAO.updateBerlaku(pendudukDAO.selectIDKeluargaPenduduk(nik));
    redirectAttributes.addFlashAttribute(attributeName: "wafat", attributeValue: "success");
    return "redirect:/penduduk?nik=" + nik;
}

<div th:if="${wafat} == 'success'">
    <h2 th:text="'Sukses!'"></h2>
    <p th:text="'Penduduk dengan NIK ' + ${penduduk.nik} + ' sudah tidak aktif'"></p>
    <hr />
</div>

<td>
    <form action="/penduduk/mati" th:if="${penduduk.is_wafat} == 0" method="post">
        <input name="nik" th:value="${penduduk.nik}" hidden="hidden" />
        <button type="submit" class="btn btn-danger">Nonaktifkan</button>
    </form>
</td>

void updateWafat(String nik);

@Override
public void updateWafat(String nik) { pendudukDAO.updateWafat(nik); }

@Update("UPDATE PENDUDUK " +
    "SET is_wafat = 1 " +
    "WHERE nik = #{nik}")
void updateWafat(@Param("nik") String nik);

@Update("UPDATE KELUARGA SET is_tidak_berlaku = 1 WHERE id = #{id}")
void updateBerlaku(@Param("id") String id);

boolean updateBerlaku(String id);

```

```

@Override
public boolean updateBerlaku(String id) {
    ArrayList<PendudukModel> anggotaKeluarga = new ArrayList<PendudukModel>(keluargaDAO.selectAnggotaKeluarga(id));
    for (PendudukModel penduduk: anggotaKeluarga)
    {
        if(penduduk.getIs_wafat() == 0)
        {
            return false;
        }
    }
    keluargaDAO.updateBerlaku(id);
    return true;
}

```

## 8. Menampilkan Data Penduduk Kelurahan

Pada fitur ini pertama tama saya membuat tiga tampilan form yang nantinya akan digunakan untuk memilih kota > kecamatan > kelurahan.

```

<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-md-10">
            <div class="penduduk-add">
                <form action="/penduduk/cari" method="get">
                    <h4 th:text="'Lihat Data Penduduk Berdasarkan Kota/Kabupaten'"></h4>
                    <label for="kota" th:text="'Kota/Kabupaten'">KOTA</label>
                    <select class="form-control" id="kota" name="kt">
                        <option th:value="" th:text="'Pilih Kota/Kabupaten'"></option>
                        <div th:each="kota, iterationStatus: ${listKota}">
                            <option th:value="${kota.kode_kota}" th:text="${kota.nama_kota}"></option>
                        </div>
                    </select>
                    <br />
                    <button type="submit" class="btn btn-primary">Lihat</button>
                </form>
            </div>
        </div>
    </div>
</div>
<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-md-10">
            <div class="penduduk-add">
                <form action="/penduduk/cari" method="get">
                    <h4 th:text="'Lihat Data Penduduk Berdasarkan Kecamatan di ' + ${strings.capitalizeWords(strings.toLowerCase(kota.nama_kota))}'"></h4>
                    <label for="kota" th:text="'Kota'">KOTA</label>
                    <select class="form-control" id="kota" disabled="disabled">
                        <option th:text="${kota.nama_kota}"></option>
                    </select>
                    <input name="kt" th:value="${id_kota}" hidden="hidden" />
                    <label for="kecamatan" th:text="'Kecamatan'">KECAMATAN</label>
                    <select class="form-control" id="kecamatan" name="kc">
                        <option th:value="" th:text="'Pilih Kecamatan'"></option>
                        <div th:each="kecamatan, iterationStatus: ${listKecamatan}">
                            <option th:if="${kecamatan.kode_kota} == ${id_kota}" th:value="${kecamatan.kode_kecamatan}" th:text="${kecamatan.nama_kecamatan}"></option>
                        </div>
                    </select>
                    <br />
                    <button type="submit" class="btn btn-primary">Lihat</button>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

<div class="container">
<div class="row">
<div class="col-md-10">
<div class="penduduk-add">
<form action="/penduduk/cari" method="get">
<h4 th:text="Lihat Data Penduduk Berdasarkan Kelurahan di " + ${#strings.capitalizeWords(#strings.toLowerCase(kota.nama_kota))} + " " + ${#strings.capitali
<label for="kota" th:text="Kota"=${kota}</label>
<select class="form-control" id="kota" disabled="disabled">
<option th:text="${kota.nama_kota}"></option>
</select>
<input name="kt" th:value="${id_kota}" hidden="hidden" />

<label for="kecamatan" th:text="Kecamatan"=${kecamatan}</label>
<select class="form-control" id="kecamatan" name="kc" disabled="disabled">
<option th:text="${kecamatan.nama_kecamatan}"></option>
</select>
<input name="kc" th:value="${id_kecamatan}" hidden="hidden" />

<label for="kelurahan" th:text="Kelurahan"=${kelurahan}</label>
<select class="form-control" id="kelurahan" name="kl">
<option th:value="" th:text="Pilih Kelurahan"></option>
<div th:each="kelurahan, iterationStatus: ${listKelurahan}">
<option th:if="${kelurahan.kecamatan.kode_kecamatan} == ${id_kecamatan}" th:value="${kelurahan.kode_kelurahan}" th:text="${kelurahan.nama_kelurahan}
</div>
</select>
<br />

<button type="submit" class="btn btn-primary">Lihat</button>
</form>
</div>
</div>
</div>
</div>

```

```

@GetMapping("/penduduk/cari")
public String cariPendudukKotaKecamatan(@RequestParam(value = "kt", required = false) String id_kota,
                                         @RequestParam(value = "kc", required = false) String id_kecamatan,
                                         @RequestParam(value = "kl", required = false) String id_kelurahan,
                                         Model model)
{
    if(id_kota == null || id_kota.equals(""))
    {
        model.addAttribute( attribute: "listKota", lokasiDAO.selectAllKotaForm());
        return "penduduk-search-form";
    }

    KotaModel kota = lokasiDAO.selectKotaID(id_kota);

    if(kota != null)
    {
        model.addAttribute( attribute: "kota", kota);

        if(id_kecamatan == null || id_kecamatan.equals(""))
        {
            model.addAttribute( attribute: "id_kota", id_kota);
            model.addAttribute( attribute: "listKecamatan", lokasiDAO.selectAllKecamatanForm());
            return "penduduk-search-kecamatan-form";
        }

        KecamatanModel kecamatan = lokasiDAO.selectKecamatanID(id_kecamatan);

        if(kecamatan != null && kecamatan.getKota().getNama_kota().equals(kota.getNama_kota()))
        {
            model.addAttribute( attribute: "kecamatan", kecamatan);

            if(id_kelurahan == null || id_kelurahan.equals(""))
            {
                model.addAttribute( attribute: "id_kota", id_kota);

```

```

        if(id_kelurahan == null || id_kelurahan.equals(""))
        {
            model.addAttribute( attributeName: "id_kota", id_kota);
            model.addAttribute( attributeName: "id_kecamatan", id_kecamatan);
            model.addAttribute( attributeName: "listKelurahan", lokasiDAO.selectAllKelurahanForm());
            return "penduduk-search-kelurahan-form";
        }
    } else
    {
        model.addAttribute( attributeName: "model", attributeValue: "lokasi");
        return "not-found";
    }
}

KelurahanModel kelurahan = lokasiDAO.selectKelurahanID(id_kelurahan);

if(kelurahan !=null && kelurahan.getKecamatan().getNama_kecamatan().equals(kecamatan.getNama_kecamatan()))
{
    List<PendudukModel> listPenduduk = pendudukDAO.selectPendudukKelurahan(id_kelurahan);
    model.addAttribute( attributeName: "kelurahan", kelurahan);
    model.addAttribute( attributeName: "listPenduduk", listPenduduk);
    List<PendudukModel> listTuaMuda = pendudukDAO.palingTuaMuda(listPenduduk);

    model.addAttribute( attributeName: "listTuaMuda", listTuaMuda);
    return "view-penduduk-kelurahan";
} else
{
    model.addAttribute( attributeName: "model", attributeValue: "lokasi");
    return "not-found";
}
}

model.addAttribute( attributeName: "model", attributeValue: "lokasi");
return "not-found";
}
}

```

Dimana selanjutnya jika kelurahan sudah dipilih, maka id\_kelurahan akan digunakan untuk mencari penduduk yang berada pada kelurahan tersebut. Pada hal ini pertama saya membuat query di mapper yang selanjutnya membuat service untuk mendapatkan list serta service databasenya.

```

@Select("SELECT P.nik, P.nama, P.jenis_kelamin, P.tanggal_lahir FROM PENDUDUK P " +
        "LEFT JOIN KELUARGA K ON P.id_keluarga = K.id " +
        "LEFT JOIN KELURAHAN KEL ON K.id_kelurahan = KEL.id " +
        "WHERE KEL.id = #{id_kelurahan}")
List<PendudukModel> selectPendudukKelurahan(@Param("id_kelurahan") String id_kelurahan);

List<PendudukModel> selectPendudukKelurahan(String id_kelurahan);

```

Selanjutnya pada tampilannya karena menggunakan datatable, maka saya menambahkan file java script agar datatable dapat terbaca.

```

<div class="col-md-12">
  <table id="table-id" class="display">
    <thead>
      <tr>
        <th th:text="'NIK'">NIK</th>
        <th id="nama" th:text="'Nama'">NAMA</th>
        <th th:text="'Jenis Kelamin'">JENIS KELAMIN</th>
      </tr>
    </thead>
    <tbody>
      <tr th:each="penduduk, iterationStatus: ${listPenduduk}">
        <td th:text="${penduduk.nik}">NIK</td>
        <td th:text="${penduduk.nama}">NAMA</td>
        <td th:text="(${penduduk.jenis_kelamin} == 0) ? 'Pria' : 'Wanita'">JENIS KELAMIN</td>
      </tr>
    </tbody>
  </table>
</div>

$(document).ready(function () {
  // Disable search and ordering by default
  $.extend( $.fn.dataTable.defaults, {
    searching: true,
    ordering: false
  });

  // For this specific table we are going to enable ordering
  // (searching is still disabled)
  $('#table-id').DataTable( {
    ordering: true
  });
})

```

## BONUS

Selain itu fitur bonus terkait validasi juga dilakukan dengan cara mencegah suatu nilai kosong atau null. Serta pada fitur terakhir mencegah ketidak sinkronan antara id kota, kecamatan dan kelurahan.

```

@NotNull @NotEmpty
private String pekerjaan;

```

```

kelurahan.getKecamatan().getNama_kecamatan().equals(kecamatan.getNama_kecamatan())

```

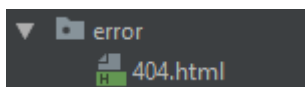
Kemudian fitur yang menampilkan penduduk termuda juga saya buat dimana listpenduduk di suatu keluarga akan digunakan untuk selanjutnya mengembalikan list yang berisikan tertua dan termuda.

```

@Override
public List<PendudukModel> palingTuaMuda(List<PendudukModel> listPenduduk) {
    List<PendudukModel> result = new ArrayList<PendudukModel>();
    result.add(index 0, listPenduduk.get(0));
    result.add(index 1, listPenduduk.get(0));
    for (PendudukModel penduduk: listPenduduk)
    {
        Date tanggal_lahir = penduduk.getTanggal_lahir();
        if(tanggal_lahir.after(result.get(0).getTanggal_lahir()))
        {
            result.set(0, penduduk);
        } else if(tanggal_lahir.before(result.get(1).getTanggal_lahir()))
        {
            result.set(1, penduduk);
        }
    }
    log.info("termuda {} tertua {}", result.get(0).getTanggal_lahir(), result.get(1).getTanggal_lahir());
    return result;
}

```

Kemudian terkait error page not found juga saya buat dengan membuat folder error di templates dengan file bernama 404.html



Terkait fitur lainnya yang mendukung sudah saya sebutkan diantaranya adalah select option pada saat memilih kota kecamatan dan kelurahan pada form keluarga serta saya juga menambahkan navigasi di navbar dan juga terdapat tombol update saat sedang melihat data penduduk

SI Kependudukan
Home
Cari Data
Tambah Penduduk
Tambah Keluarga

### Tambah Keluarga

Alamat

RT

RW

Kota

Pilih Kota/Kabupaten ▼

Kecamatan

Pilih Kecamatan ▼

Kelurahan

Pilih Kelurahan ▼

Lihat Data Penduduk - 310

NIK

Nama

Tempat/Tanggal Lahir

Alamat

RT/RW

Kelurahan/Desa

Kecamatan

Kota

Golongan Darah

Agama

Status Perkawinan

Pekerjaan

Kewarganegaraan

Status Kematian

Update

## **OPTIMASI DATABASE**

Salah satu cara yang dilakukan saya pertama kali terkait optimasi database adalah menambahkan primary key pada setiap table dan melakukan auto increment pada primary key di table penduduk dan keluarga. Serta saya juga membuat index pada setiap kolom yang ada di table

## **STRESS TESTING**

Setelah melakukan stress testing saya mengambil kesimpulan bahwa masih terdapat banyak fitur yang memiliki average time tinggi sehingga program berjalan lambat saat digunakan oleh ribuan orang. Sistem ini perlu diperbaiki lagi agar nantinya dapat digunakan oleh banyak orang dalam waktu yang singkat.