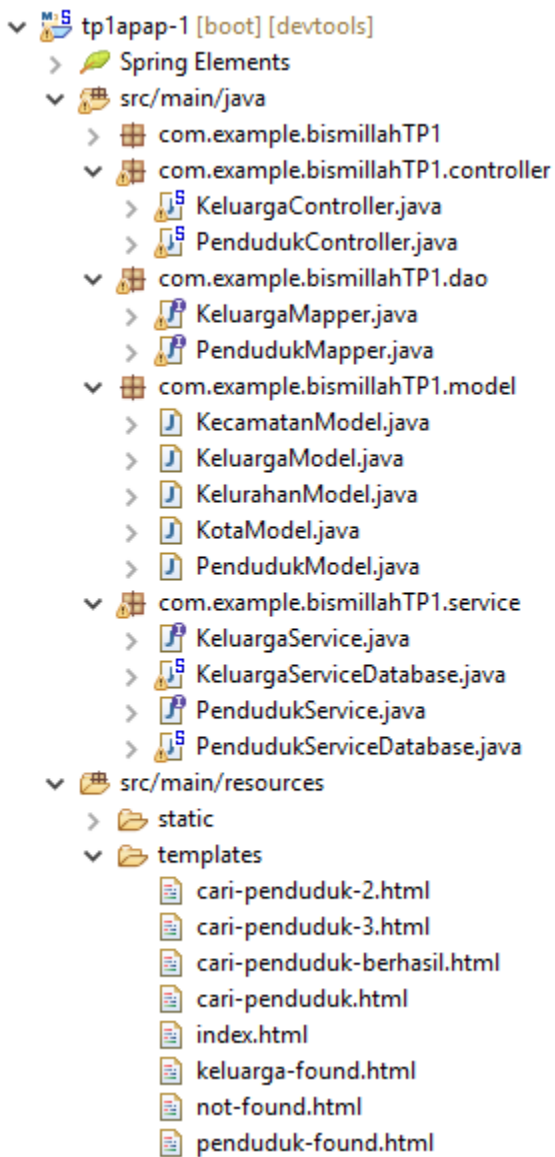


Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

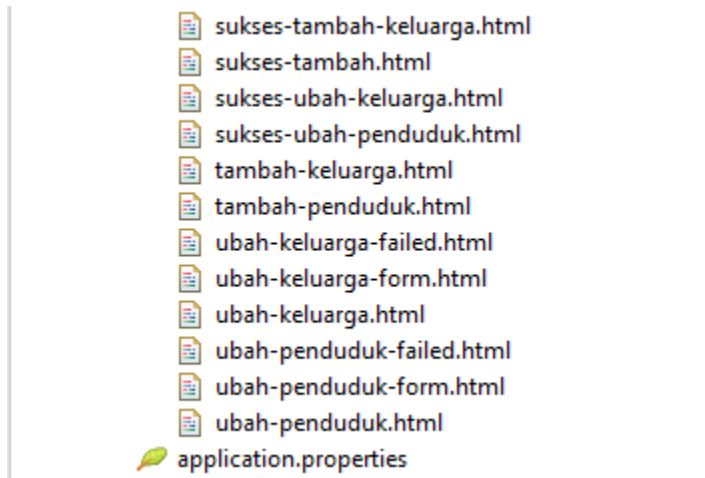
Pada pengerjaan tugas 1 APAP ini, hal pertama yang saya lakukan setelah membaca soal adalah melakukan *import file* siduk.sql ke *database*, kemudian saya membuat Spring Starter Project di Eclipse Jee-Oxygen dengan menggunakan *dependency* Web, Thymeleaf, MySQL, Lombok, MyBatis, dan DevTools. Setelah itu saya membuat kelas-kelas *model* yang diperlukan dan selanjutnya langsung *coding* fitur-fitur yang diminta.

STRUKTUR PROJECT

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B



Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B



Penjelasan:

- *Package Controller* yang berisi *PendudukController* dan *KeluargaController* berfungsi untuk mengatur segala hal yang berkaitan dengan *Penduduk* dan *Keluarga* serta sebagai penghubung antara tampilan (*view*) dengan *Service*, *Mapper*, dan *Model*
- *Package dao* berfungsi untuk melakukan *method* yang berkaitan dengan *database* sehingga *database* bisa dimanipulasi
- *Package Model* merepresentasikan objek-objek yang digunakan dalam program ini. *Model* utama yang digunakan pada program ini adalah *PendudukModel* dan *KeluargaModel*
- *Package Service* berfungsi sebagai *mediator* antara *controller* dan *database* serta menyimpan fitur-fitur apa saja yang bisa dilakukan objek di dalam *Service Interface* nya
- Dalam folder *templates* berisi semua *view* yang menjadi tampilan untuk keseluruhan fitur

A. Proses Pembuatan Masing-masing Fitur

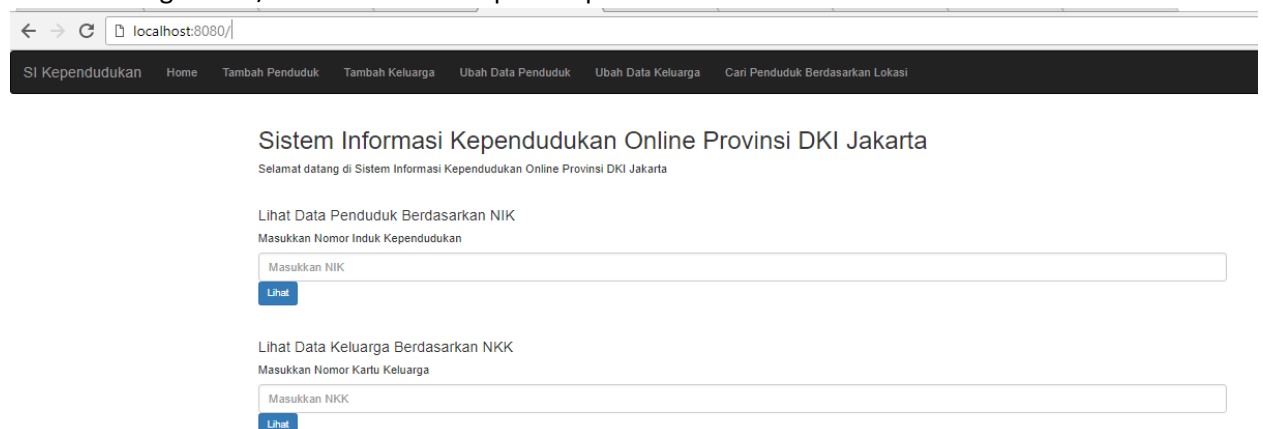
Untuk fitur 1 dan 2 diawali dari dengan halaman *index.html* dengan potongan kode seperti berikut:

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

```
<title>Sistem Kependudukan DKI</title>
</head>
<body>
  <nav class="navbar navbar-inverse">
    <div class="container-fluid">
      <div class="navbar-header">
        <a class="navbar-brand" href="/">SI Kependudukan</a>
      </div>
      <ul class="nav navbar-nav">
        <li><a href="/">Home</a></li>
        <li><a href="/penduduk/tambah">Tambah Penduduk</a></li>
        <li><a href="/keluarga/tambah">Tambah Keluarga</a></li>
        <li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>
        <li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>
        <li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>
      </ul>
    </div>
  </nav>
  <div class="container-fluid text-center">
    <div class="row content">
      <div class="col-sm-2"></div>
      <div class="col-sm-8 text-left">
        <h2>Sistem Informasi Kependudukan Online Provinsi DKI Jakarta</h2>
        <h5>Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta</h5>
        <hr>
        <h4>Lihat Data Penduduk Berdasarkan NIK</h4>
        <h5>Masukkan Nomor Induk Kependudukan</h5>
        <div class="container">
          <div class="row">
            <div class="search">
              <form method="get" action="/penduduk">
                <input type="text" name="nik" class="form-control" maxlength="64" placeholder="Masukkan NIK" />
                <button type="submit" class="btn btn-primary btn-sm">Lihat</button>
              </form>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
```

1. Tampilkan Data Penduduk Berdasarkan NIK

- Ketika mengakses "/" akan muncul tampilan seperti berikut



- Melihat data penduduk dengan NIK = 3101010303890001

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

Lihat Data Penduduk Berdasarkan NIK

Masukkan Nomor Induk Kependudukan

3101010303890001

Lihat

- Lalu klik *button* “lihat” sehingga muncul:

← → ↻

SI Kependudukan Home Tambah Penduduk Tambah Keluarga Ubah Data Penduduk Ubah Data Keluarga Cari Penduduk Berdasarkan Lokasi

Lihat Data Penduduk - 3101010303890001

Nonaktifkan

NIK	3101010303890001
nama	Taswir Rajata
Jenis Kelamin	Pria
Tempat/Tanggal Lahir	Jakarta Fri Mar 03 00:00:00 ICT 1989
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	Kawin
Kewarganegaraan	WNI
Status Kematian	Hidup

Kode:

- Membuat *query* pada PendudukMapper untuk memilih data penduduk sesuai NIK yang diminta

```
@Select("select P.nik, P.nama, P.tempat_lahir, P.tanggal_lahir, P.jenis_kelamin, "
+ "P.is_wni, P.agama, P.pekerjaan, P.status_perkawinan, P.status_dalam_keluarga, P.golongan_darah, P.is_wafat, P.id_keluarga, "
+ "K.alamat, K.RT, K.RW, KL.nama_kelurahan, KC.nama_kecamatan, KO.nama_kota FROM PENDUDUK P,"
+ "KELUARGA K, KELURAHAN KL, KECAMATAN KC, KOTA KO WHERE P.id_keluarga = K.id AND K.id_kelurahan = KL.id AND "
+ "KL.id_kecamatan = KC.id AND KC.id_kota = KO.id AND P.nik = #{nik}")
@Results(value = {
    @Result(property="nik", column="nik"),
    @Result(property="nama", column="nama"),
    @Result(property="tempat_lahir", column="tempat_lahir"),
    @Result(property="tanggal_lahir", column="tanggal_lahir"),
    @Result(property="jenis_kelamin", column="jenis_kelamin"),
    @Result(property="alamat", column="alamat"),
    @Result(property="rt", column="RT"),
    @Result(property="rw", column="RW"),
    @Result(property="nama_kelurahan", column="nama_kelurahan"),
    @Result(property="nama_kecamatan", column="nama_kecamatan"),
    @Result(property="nama_kota", column="nama_kota"),
    @Result(property="golongan_darah", column="golongan_darah"),
    @Result(property="agama", column="agama"),
    @Result(property="status_perkawinan", column="status_perkawinan"),
    @Result(property="status_dalam_keluarga", column="status_dalam_keluarga"),
    @Result(property="pekerjaan", column="pekerjaan"),
    @Result(property="is_wni", column="is_wni"),
    @Result(property="is_wafat", column="is_wafat"),
    @Result(property="id_keluarga", column="id_keluarga")
})
PendudukModel selectPenduduk (@Param("nik") String nik);
```

- Membuat *method* pada *interface* PendudukService

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

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

- Mengimplementasikan *method* tersebut ke PendudukServiceDatabase

```
@Override
public PendudukModel selectPenduduk(String nik) {
    Log.info("select penduduk with nik{}",nik);
    return pendudukMapper.selectPenduduk(nik);
}
```

- Membuat method viewPenduduk dengan parameter NIK

```
@RequestMapping(method = RequestMethod.GET, value = "/penduduk")
public String viewPenduduk(Model model, @RequestParam(value = "nik", required = true) String nik) {
    PendudukModel penduduk = pendudukService.selectPenduduk(nik);
    if (penduduk == null) {
        model.addAttribute("nik", nik);
        return "not-found";
    } else {
        model.addAttribute("nik", nik);
        model.addAttribute("nama", penduduk.getNama());
        model.addAttribute("tempat_lahir", penduduk.getTempat_lahir());
        model.addAttribute("tanggal_lahir", penduduk.getTanggal_lahir());
        model.addAttribute("alamat", penduduk.getAlamat());
        model.addAttribute("rt", penduduk.getRt());
        model.addAttribute("rw", penduduk.getRw());
        model.addAttribute("nama_kelurahan", penduduk.getNama_kelurahan());
        model.addAttribute("nama_kecamatan", penduduk.getNama_kecamatan());
        model.addAttribute("nama_kota", penduduk.getNama_kota());
        model.addAttribute("golongan_darah", penduduk.getGolongan_darah());
        model.addAttribute("agama", penduduk.getAgama());
        model.addAttribute("status_perkawinan", penduduk.getStatus_perkawinan());
        model.addAttribute("pekerjaan", penduduk.getPekerjaan());

        if (penduduk.getIs_wni() == 1) {
            String wni = "WNI";
            model.addAttribute("is_wni", wni);
        } else {
            String wna = "WNA";
            model.addAttribute("is_wni", wna);
        }

        if (penduduk.getIs_wafat() == 0) {
            String hidup = "Hidup";
            model.addAttribute("is_wafat", hidup);
        } else {
            String mati = "Mati";
            model.addAttribute("is_wafat", mati);
        }

        if (penduduk.getJenis_kelamin() == 1) {
            String wanita = "Wanita";
            model.addAttribute("jenis_kelamin", wanita);
        } else {
            String pria = "Pria";
            model.addAttribute("jenis_kelamin", pria);
        }

        return "penduduk-found";
    }
}
```

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

- Membuat view untuk menampilkan hasilnya di penduduk-found.html

```
<div class="container-fluid text-center">
<div class="row content">
<div class="col-sm-2"></div>
<div class="col-sm-8 text-left">
<h2 th:text="'Lihat Data Penduduk - ' + ${nik}"></h2>
<table id="view-data" cellpadding="0" width="100%">
<tbody>
<tr>
<td><strong>NIK</strong></td>
<td th:text="${nik}"></td>
</tr>
<tr>
<td><strong>nama</strong></td>
<td th:text="${nama}"></td>
</tr>
<tr>
<td><strong>Jenis Kelamin</strong></td>
<td th:text="${jenis_kelamin}"></td>
</tr>
<tr>
<td><strong>Tempat/Tanggal Lahir</strong></td>
<td th:text="${tempat_lahir} + ' ' + ${tanggal_lahir}"></td>
</tr>
<tr>
<td><strong>Alamat</strong></td>
<td th:text="${alamat}"></td>
</tr>
<tr>
<td><strong>RT/RW</strong></td>
<td th:text="${rt} + '/' + ${rw}"></td>
</tr>
<tr>
<td><strong>Kelurahan/Desa</strong></td>
<td th:text="${nama_kelurahan}"></td>
</tr>
<tr>
<td><strong>Kecamatan</strong></td>
<td th:text="${nama_kecamatan}"></td>
</tr>
<tr>
<td><strong>Kota</strong></td>
<td th:text="${nama_kota}"></td>
</tr>
<tr>
<td><strong>Golongan Darah</strong></td>
<td th:text="${golongan_darah}"></td>
</tr>
<tr>
<td><strong>Agama</strong></td>
<td th:text="${agama}"></td>
</tr>
<tr>
<td><strong>Status Perkawinan</strong></td>
<td th:text="${status_perkawinan}"></td>
</tr>
<tr>
<td><strong>Kewarganegaraan</strong></td>
<td th:text="${is_wni}"></td>
</tr>
<tr>
<td><strong>Status Kematian</strong></td>
<td th:text="${is_wafat}"></td>
<td th:if="${is_wafat} == 'Hidup'">
<form action="/penduduk/mati" method="POST">
<input type="hidden" name="nik" th:value="${nik}"></input>
<button type="submit" class="btn btn-danger" name="action"
value="submit">Nonaktifkan</button>
</form>
</td>
</tr>
</tbody>
</table>
```

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

2. Tampilkan Data Keluarga Beserta Daftar Anggotanya Berdasarkan Nomor KK

- Melihat data keluarga dengan NKK = 3101010112010001

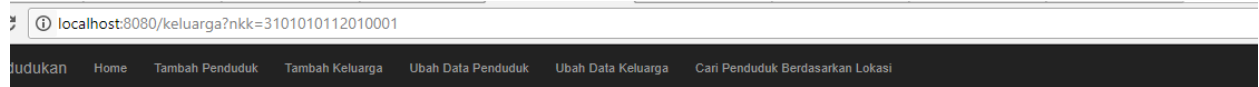
Lihat Data Keluarga Berdasarkan NKK

Masukkan Nomor Kartu Keluarga

3101010112010001

Lihat

- Lalu klik *button* "lihat" sehingga muncul:



Lihat Data Keluarga - 3101010112010001

NKK 3101010112010001
Alamat Psr. Pasir Koja No. 138
RT/RW 147/039
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	Halim Wahyudin	3101010508860001	Pria	Tangerang	Tue Aug 05 00:00:00 ICT 1986	Islam	PENATA RAMBUT	Kawin	Kepala Keluarga	WNI
2	Warta Prayoga M.TI.	3101011607650001	Pria	Kupang	Fri Jul 16 00:00:00 ICT 1965	Kristen	GURU	Kawin	Famili Lain	WNI

Kode:

- Membuat method selectKeluarga pada keluargaMapper

```
@Select("select K.nomor_kk, K.alamat, K.rt, K.rw, KL.nama_kelurahan, "
+ "KC.nama_kecamatan, KO.nama_kota FROM KELUARGA K, KELURAHAN KL, "
+ "KECAMATAN KC, KOTA KO WHERE K.id_kelurahan = KL.id AND KL.id_kecamatan = KC.id "
+ "AND KC.id_kota = KO.id AND K.nomor_kk = #{nkk}")
@Results(value = {
    @Result(property="nomor_kk", column="nomor_kk"),
    @Result(property="alamat", column="alamat"),
    @Result(property="rt", column="rt"),
    @Result(property="rw", column="rw"),
    @Result(property="nama_kelurahan", column="nama_kelurahan"),
    @Result(property="nama_kecamatan", column="nama_kecamatan"),
    @Result(property="nama_kota", column="nama_kota")
})
KeluargaModel selectKeluarga(@Param("nkk") String nkk);
```

- Membuat method selectKeluarga pada KeluargaService

```
KeluargaModel selectKeluarga(String nkk);
```


Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

- Mengimplementasikan method selectKeluarga di KeluargaServiceDatabase

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

- Membuat *method* viewKeluarga dalam KeluargaController

```
@RequestMapping(method = RequestMethod.GET, value="/keluarga")
public String viewKeluarga (Model model,
    @RequestParam(value="nkk", required = true) String nkk)
{
    KeluargaModel keluarga = keluargaService.selectKeluarga(nkk);
    List<PendudukModel> daftarKeluarga = keluargaService.selectDaftarKeluarga(nkk);

    if(keluarga == null) {
        model.addAttribute("nkk", nkk);
        return "not-found";
    } else {
        model.addAttribute("nkk", keluarga.getNomor_kk());
        model.addAttribute("alamat", keluarga.getAlamat());
        model.addAttribute("rt", keluarga.getRt());
        model.addAttribute("rw", keluarga.getRw());
        model.addAttribute("nama_kelurahan", keluarga.getNama_kelurahan());
        model.addAttribute("nama_kecamatan", keluarga.getNama_kecamatan());
        model.addAttribute("nama_kota", keluarga.getNama_kota());

        model.addAttribute("daftarKeluarga", daftarKeluarga);

        return "keluarga-found";
    }
}
```

- Membuat view keluarga-found.html untuk menampilkan hasilnya, berikut potongan kodenya

```
<div class="container">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>No.</th>
        <th>Nama Lengkap</th>
        <th>NIK</th>
        <th>Jenis Kelamin</th>
        <th>Tempat Lahir</th>
        <th>Tanggal Lahir</th>
        <th>Agama</th>
        <th>Pekerjaan</th>
        <th>Status Perkawinan</th>
        <th>Status Dalam Keluarga</th>
        <th>Kewarganegaraan</th>
      </tr>
    </thead>
    <div th:each="daftarKeluarga, iterationStatus: ${daftarKeluarga}">
    <tbody>
      <tr>
        <td th:text="${iterationStatus.count}"></td>
        <td th:text="${daftarKeluarga.nama}"></td>
        <td th:text="${daftarKeluarga.nik}"></td>
        <td th:text="${daftarKeluarga.jenis_kelamin} == 0 ? 'Pria' : 'Wanita'"></td>
        <td th:text="${daftarKeluarga.tempat_lahir}"></td>
        <td th:text="${daftarKeluarga.tanggal_lahir}"></td>
        <td th:text="${daftarKeluarga.agama}"></td>
        <td th:text="${daftarKeluarga.pekerjaan}"></td>
        <td th:text="${daftarKeluarga.status_perkawinan}"></td>
        <td th:text="${daftarKeluarga.status_dalam_keluarga}"></td>
        <td th:text="${daftarKeluarga.is_wni} == 1 ? 'WNI' : 'WNA'"></td>
      </tr>
    </tbody>
    </div>
  </table>
</div>
```

3. Menambahkan Penduduk Baru Sebagai Anggota Keluarga

- halaman ini bisa diakses melalui menu “Tambah Penduduk” di navbar

SI Kependudukan Home Tambah Penduduk Tambah Keluarga Ubah Data Penduduk Ubah Data Keluarga Cari Penduduk Berdasarkan Lokasi

- Ketika menu tersebut di-klik, akan muncul tampilan seperti berikut:

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

Submit

Tambah Penduduk

Masukkan Nama

Tempat Lahir

Tanggal Lahir

Jenis Kelamin

Wanita ▼

Golongan Darah

A ▼

Agama

Status Perkawinan

Kawin ▼

Status Dalam Keluarga

Pekerjaan

Kewarganegaraan

WNI ▼

Status Kematian

Hidup ▼

ID Keluarga

- Setelah semua data diisi dengan benar, misalnya seperti ini:

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

Tambah Penduduk

Masukkan Nama

Tempat Lahir

Tanggal Lahir

Jenis Kelamin

Golongan Darah

Agama

Status Perkawinan

Status Dalam Keluarga

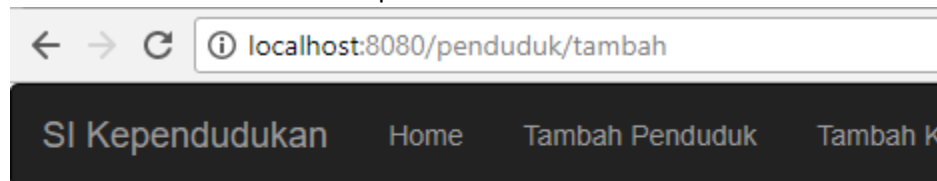
Pekerjaan

Kewarganegaraan

Status Kematian

ID Keluarga

Maka akan muncul halaman seperti berikut:



Sukses!

Penduduk dengan NIK 3101011310570001 berhasil ditambahkan

Kode:

- menu Tambah Penduduk pada navbar tersebut terdapat di index.html

```
<li><a href="/penduduk/tambah">Tambah Penduduk</a></li>
```

- ketika di-klik, akan me-refer ke bukaFormPenduduk di PendudukController

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

- kemudian akan me-refer lagi ke tambah-penduduk.html, yang menampilkan form untuk mengisi data penduduk baru. Berikut potongan kode yang ada di tambah-penduduk.html

```
<h1 class="page-header">Tambah Penduduk</h1>
<form action="/penduduk/tambah" method="POST">
    <div class="form-group">
        <label for="nama">Masukkan Nama</label> <input type="text"
            class="form-control" name="input_nama"/>
    </div>

    <div class="form-group">
        <label for="tempat_lahir">Tempat Lahir</label> <input type="text"
            class="form-control" id="tempat_lahir" name="input_tempat_lahir"/>
    </div>

    <div class="form-group">
        <label for="tanggal_lahir">Tanggal Lahir</label> <input type="date"
            class="form-control" id="tggl_lahir" name="input_tanggal_lahir" />
    </div>
    <div class="form-group">
        <label for="jenis_kelamin">Jenis Kelamin</label>
        <select name="input_jenis_kelamin" class="form-control">
            <option value="wanita">Wanita</option>
            <option value="pria">Pria</option>
        </select>
    </div>
    <div class="form-group">
        <label for="golongan_darah">Golongan Darah</label>
        <select name="input_gol_dar" class="form-control">
            <option value="A">A</option>
            <option value="B">B</option>
            <option value="A-">A-</option>
            <option value="B-">B-</option>
            <option value="A+">A+</option>
            <option value="B+">B+</option>
            <option value="AB">AB</option>
            <option value="O">O</option>
        </select>
    </div>

    <button type="submit" class="btn btn-default">Submit</button>
```

- Ketika *button* tersebut di-klik akan me-refer /penduduk/tambah dengan method POST. Berikut

potongan kode /penduduk/tambah

```
@RequestMapping(method = RequestMethod.POST, value="/penduduk/tambah")
public String tambahPenduduk (@RequestParam(value="input_nama", required = true) String nama,
    @RequestParam(value="input_tempat_lahir", required = true) String tempat_lahir,
    @RequestParam(value="input_tanggal_lahir", required = true) String tanggal_lahir_tmp,
    @RequestParam(value="input_jenis_kelamin", required = true) String jenis_kelamin_tmp,
    @RequestParam(value="input_goldar", required = true) String golongan_darah,
    @RequestParam(value="input_agama", required = true) String agama,
    @RequestParam(value="input_stts_kawin", required = true) String status_perkawinan,
    @RequestParam(value="input_stts_keluarga", required = true) String status_dalam_keluarga,
    @RequestParam(value="input_pekerjaan", required = true) String pekerjaan,
    @RequestParam(value="input_is_wni", required = true) String is_wni_tmp,
    @RequestParam(value="input_id_keluarga", required = true) String id_keluarga,
    @RequestParam(value="input_is_wafat", required = true) String is_wafat_tmp,
    Model model) throws Exception {

    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
    Date tanggal_lahir = df.parse(tanggal_lahir_tmp);

    int jenis_kelamin;
    int is_wafat;
    int is_wni;

    if(jenis_kelamin_tmp.equalsIgnoreCase("wanita")) {
        jenis_kelamin = 1;
    } else {
        jenis_kelamin = 0;
    }

    if(is_wni_tmp.equalsIgnoreCase("wni")) {
        is_wni = 1;
    } else {
        is_wni = 0;
    }
}
```

- Apabila data sukses ditambahkan, akan merefer ke halaman sukses-tambah.html

```
<body>
    <nav class="navbar navbar-inverse">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand" href="/">SI Kependudukan</a>
            </div>
            <ul class="nav navbar-nav">
                <li><a href="/">Home</a></li>
                <li><a href="/penduduk/tambah">Tambah Penduduk</a></li>
                <li><a href="/keluarga/tambah">Tambah Keluarga</a></li>
                <li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>
                <li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>
                <li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>
            </ul>
        </div>
    </nav>

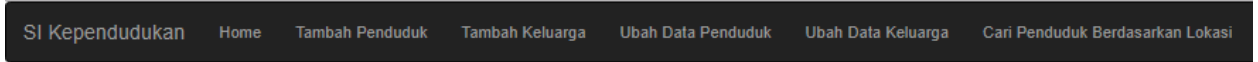
    <h3>Sukses!</h3>
    <h5 th:text="'Penduduk dengan NIK ' + ${nik} + ' berhasil ditambahkan'"></h5>

</body>
```

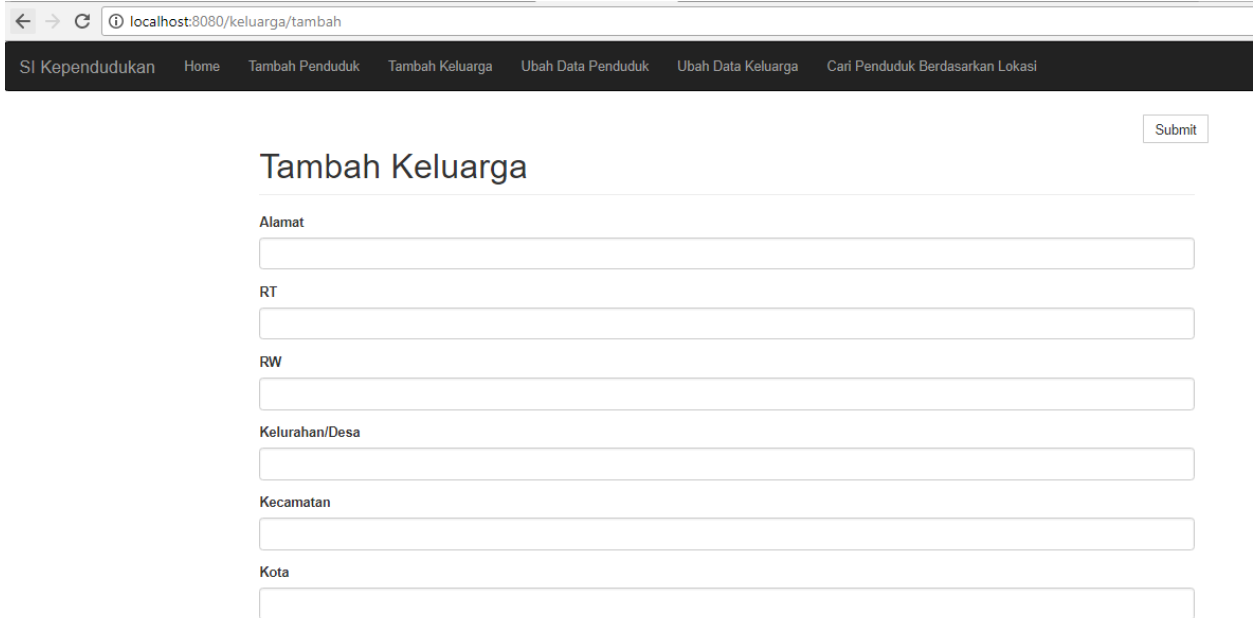
Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

4. Menambahkan Keluarga Baru

- menu Tambah Keluarga bisa diakses di menu di navbar yang tersedia



- setelah menu Tambah Keluarga di-klik, akan me-refer ke halaman form untuk mengisi data keluarga

A screenshot of a web browser showing the 'Tambah Keluarga' form. The browser's address bar shows 'localhost:8080/keluarga/tambah'. The page has a dark navigation bar at the top with the same menu items as the previous screenshot. The form itself is titled 'Tambah Keluarga' and has a 'Submit' button in the top right corner. The form contains six input fields, each with a label to its left: 'Alamat', 'RT', 'RW', 'Kelurahan/Desa', 'Kecamatan', and 'Kota'. All input fields are currently empty.

- Setelah data diisi dengan benar, misalnya seperti ini:

Submit

Tambah Keluarga

Alamat

Jl. Sedap malam

RT

5

RW

1

Kelurahan/Desa

Jagakarsa

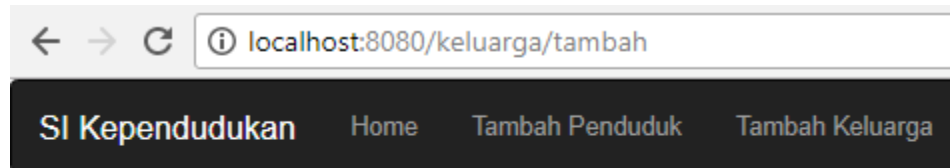
Kecamatan

Permai

Kota

Kota Jakarta Kota

- Maka setelah tombol di-submit, akan muncul halaman berikut:



Sukses!

Keluarga dengan NKK 3171012210170001 berhasil ditambahkan

kode:

- menu Tambah Keluarga di navbar disimpan pada halaman index.html seperti berikut:

```
<li><a href="/keluarga/tambah">Tambah Keluarga</a></li>
```

- Ketika menu Tambah Keluarga di-klik akan me-refer ke bukaFormKeluarga di KeluargaController

```
@RequestMapping(method = RequestMethod.GET, value="/keluarga/tambah")
public String bukaFormKeluarga (Model model) {
    model.addAttribute("keluarga", new KeluargaModel());
    return "tambah-keluarga";
}
```


- Selanjutnya bukaFormKeluarga akan me-refer ke tambah-keluarga.html yang mana akan menampilkan form untuk mengisi data keluarga yang baru, berikut potongan kode-nya:

```
<div class="container-fluid text-center">
<div class="row content">
<div class="col-sm-2"></div>
<div class="col-sm-8 text-left">
<h1 class="page-header">Tambah Keluarga</h1>
<form action="/keluarga/tambah" method="POST">
  <div class="form-group">
    <label for="alamat">Alamat</label> <input type="name"
      class="form-control" name="input_alamat"/>
  </div>

  <div class="form-group">
    <label for="rt">RT</label> <input type="name"
      class="form-control" name="input_rt"/>
  </div>

  <div class="form-group">
    <label for="rw">RW</label> <input type="name"
      class="form-control" name="input_rw" />
  </div>
  <div class="form-group">
    <label for="kelurahan">Kelurahan/Desa</label> <input type="name"
      class="form-control" name="input_kelurahan"/>
  </div>
  <div class="form-group">
    <label for="kecamatan">Kecamatan</label> <input
      type="name" class="form-control" name="input_kecamatan" />
  </div>
  <div class="form-group">
    <label for="kota">Kota</label> <input type="name"
      class="form-control" name="input_kota"/>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
```

- Ketika tombol submit yang ada pada form tersebut di-klik, akan me-refer ke /keluarga/tambah di KeluargaController dengan method POST sehingga akan mengambil semua data dari form

```
@RequestMapping(method = RequestMethod.POST, value="/keluarga/tambah")
public String tambahKeluarga (@RequestParam(value="input_alamat", required = true) String alamat,
    @RequestParam(value="input_rt", required = true) String rt,
    @RequestParam(value="input_rw", required = true) String rw,
    @RequestParam(value="input_kelurahan", required = true) String nama_kelurahan,
    @RequestParam(value="input_kecamatan", required = true) String nama_kecamatan,
    @RequestParam(value="input_kota", required = true) String kota, Model model) {

    KeluargaModel keluarga = new KeluargaModel();

    Long id_kelurahan = keluargaService.cariIdKelurahan(nama_kelurahan);
    String nomor_kk = keluargaService.createNKK(id_kelurahan);
    int is_tidak_berlaku = 0;

    keluarga.setNomor_kk(nomor_kk);
    keluarga.setAlamat(alamat);
    keluarga.setRt(rt);
    keluarga.setRw(rw);
    keluarga.setId_kelurahan(id_kelurahan);
    keluarga.setIs_tidak_berlaku(is_tidak_berlaku);

    keluargaService.insertKeluarga(keluarga);

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

- Setelah data berhasil dimasukkan ke database, akan me-refer ke sukses-tambah-keluarga.html dengan kode sebagai berikut:

```
<body>
    <nav class="navbar navbar-inverse">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand" href="/">SI Kependudukan</a>
            </div>
            <ul class="nav navbar-nav">
                <li><a href="/">Home</a></li>
                <li><a href="/penduduk/tambah">Tambah Penduduk</a></li>
                <li><a href="/keluarga/tambah">Tambah Keluarga</a></li>
                <li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>
                <li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>
                <li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>
            </ul>
        </div>
    </nav>

    <h3>Sukses!</h3>
    <h5 th:text="'Keluarga dengan NKK ' + ${nomor_kk} + ' berhasil ditambahkan'"></h5>
</body>
```

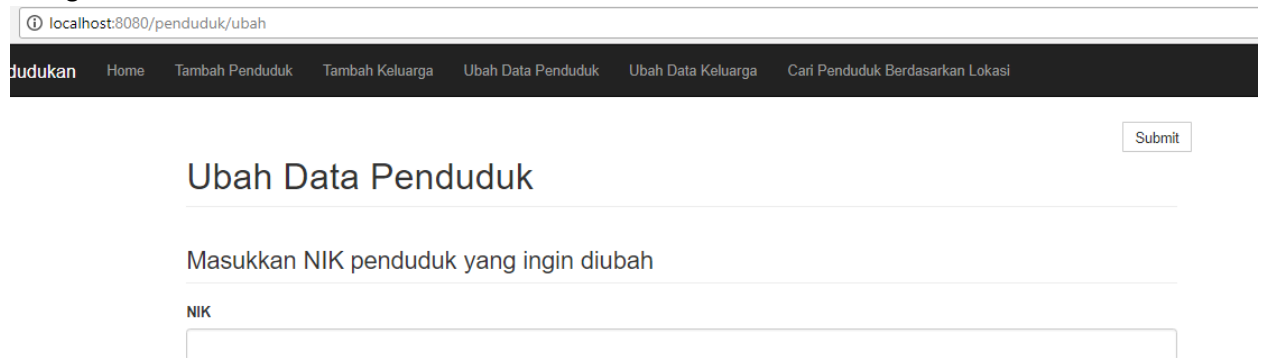
5. Mengubah Data Penduduk

- menu Ubah Data Penduduk dapat diakses pada navbar di index.html

SI Kependudukan Home Tambah Penduduk Tambah Keluarga Ubah Data Penduduk Ubah Data Keluarga Cari Penduduk Berdasarkan Lokasi

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

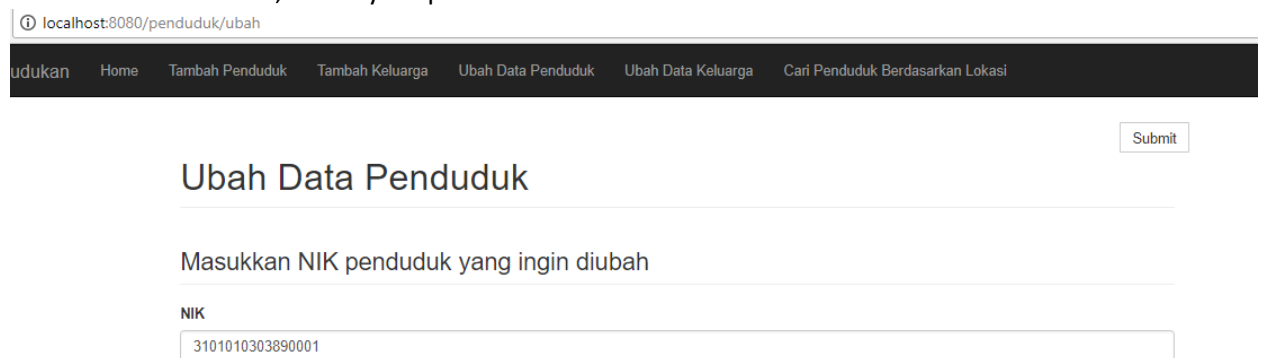
- Ketika menu tersebut di-klik akan me-refer ke halaman ubah-penduduk.html dengan tampilan sebagai berikut:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/penduduk/ubah'. The browser's navigation bar includes a dark menu with the following items: 'pendudukan', 'Home', 'Tambah Penduduk', 'Tambah Keluarga', 'Ubah Data Penduduk', 'Ubah Data Keluarga', and 'Cari Penduduk Berdasarkan Lokasi'. The main content area of the page has a light gray background and contains the following elements:

- A 'Submit' button in the top right corner.
- A heading 'Ubah Data Penduduk'.
- A text prompt 'Masukkan NIK penduduk yang ingin diubah'.
- A label 'NIK' above a text input field.
- An empty text input field for the NIK.

- Setelah field NIK diisi, misalnya seperti ini:



This screenshot is identical to the previous one, but the text input field for the NIK is now filled with the value '3101010303890001'.

- Maka setelah tombol submit di-klik, akan muncul halaman yang berisi form untuk mengubah data penduduk, dengan data-data sebelumnya yang sudah tertulis di field secara otomatis

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

Submit

Ubah Data Penduduk

NIK

3101010303890001

Masukkan Nama

Taswir Rajata

Tempat Lahir

Jakarta

Tanggal Lahir

03/03/1989

Jenis Kelamin

Pria

Golongan Darah

B+

Agama

Islam

Status Perkawinan

Kawin

Pekerjaan

KARYAWAN HONORER

Status Dalam Keluarga

Kepala Keluarga

Kewarganegaraan

WNI

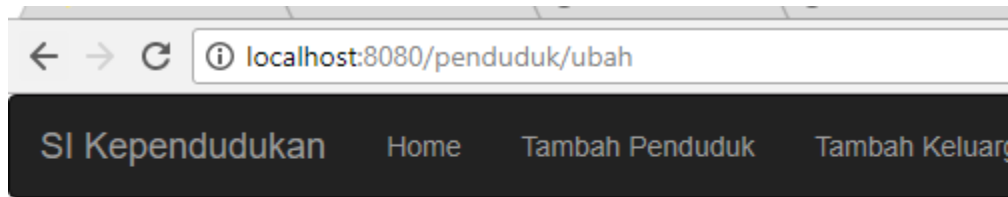
Status Kematian

Hidup

ID Keluarga

1

- Setelah itu dapat diubah data-data yang sudah sesuai dengan keperluan, kemudian setelah tombol submit di-klik akan memunculkan halaman seperti ini:



Sukses!

Penduduk dengan NIK 3101010303890001 berhasil diubah

Kode:

- menu Ubah Data Penduduk disimpan di dalam index.html seperti berikut

```
<li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>
```

- ketika menu Ubah Data Penduduk di-klik, akan me-refer ke PendudukController seperti berikut:

```
@RequestMapping("/penduduk/ubah")  
public String ubahPenduduk () {  
    return "ubah-penduduk";  
}
```

- yang kemudian akan mengembalikan halaman ubah-penduduk.html yang akan menampilkan form untuk mengisi NIK penduduk yang ingin diubah

```
<body>  
  <nav class="navbar navbar-inverse">  
    <div class="container-fluid">  
      <div class="navbar-header">  
        <a class="navbar-brand" href="/">SI Kependudukan</a>  
      </div>  
      <ul class="nav navbar-nav">  
        <li><a href="/">Home</a></li>  
        <li><a href="/penduduk/tambah">Tambah Penduduk</a></li>  
        <li><a href="/keluarga/tambah">Tambah Keluarga</a></li>  
        <li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>  
        <li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>  
        <li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>  
      </ul>  
    </div>  
  </nav>  
  
  <div class="container-fluid text-center">  
    <div class="row content">  
      <div class="col-sm-2"></div>  
      <div class="col-sm-8 text-left">  
        <h1 class="page-header">Ubah Data Penduduk</h1>  
        <h3 class="page-header">Masukkan NIK penduduk yang ingin diubah</h3>  
        <form action="/penduduk/ubah-submit" method="GET">  
          <div class="form-group">  
            <label for="nik">NIK</label> <input type="text" name="nik" class="form-control" />  
          </div>  
          <button type="submit" class="btn btn-default">Submit</button>  
        </form>
```

- Kemudian setelah NIK diisi dan tombol submit di-klik, akan me-refer ke /penduduk/ubah-submit yang terdapat dalam PendudukController dengan method POST serta parameter NIK

```
@RequestMapping(method = RequestMethod.GET, value="/penduduk/ubah-submit")
public String ubahPendudukSubmit(Model model, @RequestParam(value = "nik", required = true) String nik) {
    return "redirect:/penduduk/ubah/" + nik;
}
```

- setelah itu akan langsung refer ke /penduduk/ubah/{nik} dengan membawa parameter NIK yang tadi telah diambil:

```
@RequestMapping(value="/penduduk/ubah/{nik}")
public String ubahPendudukSubmitNIK (@PathVariable String nik, Model model){

    PendudukModel penduduk = pendudukService.selectPenduduk(nik);

    if(penduduk == null) {
        model.addAttribute("nik", nik);
        return "not-found";
    } else {
        SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
        String tanggal_lahir = df.format(penduduk.getTanggal_lahir());

        model.addAttribute("penduduk", penduduk);
        model.addAttribute("tgl_lahir", tanggal_lahir);

        return "ubah-penduduk-form";
    }
}
```

- Kemudian akan di-refer lagi ke ubah-penduduk-form.html yang akan menampilkan form untuk mengubah data penduduk. Berikut potongan kode ubah-penduduk-form:

```
<h1 class="page-header">Ubah Data Penduduk</h1>
<form th:object="${penduduk}" action="/penduduk/ubah" method="POST">
  <div class="form-group">
    <label for="nik">NIK</label>
    <input type="text" class="form-control" id="nik" name="nik" readonly="true"
      th:field="*{nik}" th:value="${penduduk.nik}"/>
  </div>
  <div class="form-group">
    <label for="nama">Masukkan Nama</label>
    <input type="text" class="form-control" id="nama" name="nama"
      th:field="*{nama}" th:value="${penduduk.nama}"/>
  </div>
  <div class="form-group">
    <label for="tempat_lahir">Tempat Lahir</label>
    <input type="text" class="form-control" id="tempat_lahir" name="tempat_lahir"
      th:field="*{tempat_lahir}" th:value="${penduduk.tempat_lahir}"/>
  </div>
  <div class="form-group">
    <label for="tanggal_lahir">Tanggal Lahir</label>
    <input type="date" class="form-control" id="tanggal_lahir" name="input_tanggal_lahir"
      th:value="${tgl_lahir}"/>
  </div>
  <div class="form-group">
    <label for="jenis_kelamin">Jenis Kelamin</label>
    <select class="form-control" name="jenis_kelamin" th:field="*{jenis_kelamin}"
      th:value="${penduduk.jenis_kelamin}">
      <option th:value="0" th:text="'Pria'"></option>
      <option th:value="1" th:text="'wanita'"></option>
    </select>
  </div>
  <div class="form-group">
    <label for="golongan_darah">Golongan Darah</label>
    <select name="golongan_darah" class="form-control" th:field="*{golongan_darah}"
      th:value="${penduduk.golongan_darah}">

```

- Setelah button submit tersebut di-klik, akan di refer ke /penduduk/ubah yang ada di PendudukController dengan method POST, lalu semua data yang diambil akan dijadikan parameter:

```
@RequestMapping(method = RequestMethod.POST, value="/penduduk/ubah")
public String ubahPendudukData (@RequestParam(value="nik", required = true) String nik_lama,
    @RequestParam(value="nama", required = true) String nama,
    @RequestParam(value="tempat_lahir", required = true) String tempat_lahir,
    @RequestParam(value="input_tanggal_lahir", required = true) String tanggal_lahir_tmp,
    @RequestParam(value="jenis_kelamin", required = true) String jenis_kelamin_tmp,
    @RequestParam(value="golongan_darah", required = true) String golongan_darah,
    @RequestParam(value="agama", required = true) String agama,
    @RequestParam(value="status_perkawinan", required = true) String status_perkawinan,
    @RequestParam(value="pekerjaan", required = true) String pekerjaan,
    @RequestParam(value="is_wni", required = true) String is_wni_tmp,
    @RequestParam(value="status_dalam_keluarga", required = true) String status_dalam_keluarga,
    @RequestParam(value="is_wafat", required = true) String is_wafat_tmp,
    @RequestParam(value="id_keluarga", required = true) String id_keluarga,
    Model model) throws Exception {
```

- Karena data yang baru akan mempengaruhi NIK, sehingga perlu dilakukan updateNIK dengan parameter nik sebelumnya, tanggal lahir, dan id keluarga

```
String nik = pendudukService.updateNIK(nik_lama, tanggal_lahir, id_keluarga);
```

- Lalu akan memanggil method updateNIK yang ada di PendudukService dan yang terimplementasi di PendudukServiceDatabase seperti ini:

```
public String updateNIK(String nik_lama, Date tanggal_lahir, String id_keluarga) {  
  
    Long id_kelurahan = pendudukMapper.selectIdKelurahan(id_keluarga);  
    Long id_kecamatan = pendudukMapper.selectIdKecamatan(id_kelurahan);  
    String kode_kecamatan = pendudukMapper.selectKodeKecamatan(id_kecamatan);  
    String kodeKecamatan = kode_kecamatan.substring(0, kode_kecamatan.length()-1);  
    SimpleDateFormat df = new SimpleDateFormat("ddMMyy");  
    String tanggalLahir = df.format(tanggal_lahir);  
    String nik_baru = kodeKecamatan + tanggalLahir;  
  
    if(nik_lama.substring(0, 11).equalsIgnoreCase(nik_baru)) {  
        return nik_lama;  
    } else {  
        nik_baru = nik_baru + "%";  
  
        List<String> NIKs = pendudukMapper.selectNIK(nik_baru);  
        nik_baru = nik_baru.substring(0, nik_baru.length()-1);  
  
        if(NIKs.size() == 0) {  
            nik_baru += "0001";  
        } else {  
            Long nikMAX = (long) 0;  
            for(int i=0; i<NIKs.size(); i++) {  
                Long tmp = Long.parseLong(NIKs.get(i));  
                if(nikMAX < tmp) {  
                    nikMAX = tmp;  
                }  
            }  
            nik_baru = Long.toString(nikMAX + 1);  
        }  
        return nik_baru;  
    }  
}
```

- dalam method tersebut akan memanggil selectIdKelurahan, selectIdKecamatan, dan selectKodeKecamatan yang ada pada pendudukMapper. Ketiga hal tersebut penting untuk membuat NIK yang baru sesuai data yang diubah”

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

```
@Select("SELECT id_kelurahan FROM keluarga WHERE id = #{id_keluarga}")
@Results(value = {
    @Result(property="id_kelurahan", column="id_kelurahan")
})
Long selectIdKelurahan(String id_keluarga);

@Select("SELECT id_kecamatan FROM kelurahan WHERE id = #{id_kelurahan}")
@Results(value = {
    @Result(property="id_kecamatan", column="id_kecamatan")
})
Long selectIdKecamatan(Long id_kelurahan);

@Select("SELECT kode_kecamatan FROM kecamatan WHERE id = #{id_kecamatan}")
@Results(value = {
    @Result(property="kode_kecamatan", column="kode_kecamatan")
})
String selectKodeKecamatan(Long id_kecamatan);
```

- Kemudian setelah mendapatkan NIK yang baru, akan dilakukan update data penduduk NIK yang lama dengan data-data baru

```
penduduk.setNik(nik);
penduduk.setNama(nama);
penduduk.setTempat_lahir(tempat_lahir);
penduduk.setTanggal_lahir(tanggal_lahir);
penduduk.setJenis_kelamin(jenis_kelamin);
penduduk.setIs_wni(is_wni);
penduduk.setId_keluarga(id_keluarga);
penduduk.setAgama(agama);
penduduk.setPekerjaan(pekerjaan);
penduduk.setStatus_perkawinan(status_perkawinan);
penduduk.setGolongan_darah(golongan_darah);
penduduk.setIs_wafat(is_wafat);
penduduk.setIs_wni(is_wni);
penduduk.setStatus_dalam_keluarga(status_dalam_keluarga);

pendudukService.updatePenduduk(penduduk, nik_lama);
```

- query updatePenduduk pada pendudukMapper adalah sebagai berikut:

```
@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 = #{penduduk.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 = #{nik_lama}")
void updatePenduduk(@Param("penduduk") PendudukModel penduduk, @Param("nik_lama") String nik_lama);
```

- Setelah data berhasil di-update, akan di-refer ke halaman sukses-ubah-penduduk.html

```
model.addAttribute("nik_lama", nik_lama);

return "sukses-ubah-penduduk";
```

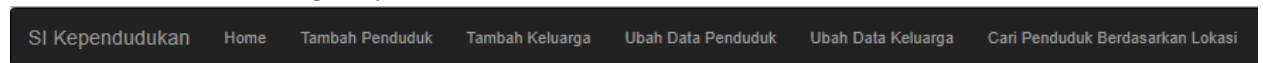
- Berikut potongan kode sukses-ubah-penduduk.html

```
<body>
  <nav class="navbar navbar-inverse">
    <div class="container-fluid">
      <div class="navbar-header">
        <a class="navbar-brand" href="/">SI Kependudukan</a>
      </div>
      <ul class="nav navbar-nav">
        <li><a href="/">Home</a></li>
        <li><a href="/penduduk/tambah">Tambah Penduduk</a></li>
        <li><a href="/keluarga/tambah">Tambah Keluarga</a></li>
        <li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>
        <li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>
        <li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>
      </ul>
    </div>
  </nav>

  <h3>Sukses!</h3>
  <h5 th:text="'Penduduk dengan NIK ' + ${nik_lama} + ' berhasil diubah'"></h5>
  . . .
```

6. Mengubah Data Keluarga

- menu Ubah Data Keluarga dapat diakses melalui navbar



- setelah menu Ubah Data Keluarga di-klik , akan me-refer ke halaman ubah-keluarga.html dengan tampilan seperti berikut

A screenshot of a web browser showing the 'Ubah Data Keluarga' page. The browser's address bar shows 'localhost:8080/keluarga/ubah'. The page has a dark navbar at the top with the same menu items as the previous image. Below the navbar, there is a 'Submit' button in the top right corner. The main heading is 'Ubah Data Keluarga'. Below it is a text input field with the placeholder text 'Masukkan NKK keluarga yang ingin diubah'. Underneath the input field, the label 'NKK' is displayed above an empty text input box.

- kemudian setelah field NKK tersebut diisi, misalnya seperti ini:

Submit

Ubah Data Keluarga

Masukkan NKK keluarga yang ingin diubah

NKK

3101010101050001

- Maka setelah tombol submit di-klik, akan mengarah ke halaman ubah-penduduk-form.html yang akan menampilkan form ubah data keluarga beserta data data sebelumnya yang otomatis terisi di dalam field

Submit

Ubah Data Keluarga

NKK

3101010101050001

Alamat

Jln. Cikapayang No. 594

RT

149

RW

090

Kelurahan/Desa

PULAU PARI

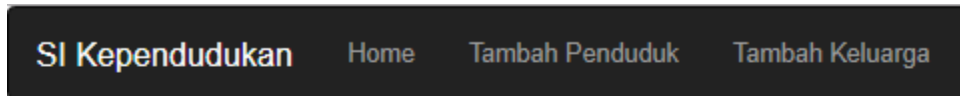
Kecamatan

KEPULAUAN SERIBU SELATAN

Kota

KABUPATEN KEPULAUAN SERIBU

- lalu setelah data diubah sesuai keperluan, kemudian tombol submit di-klik, akan muncul halaman seperti ini:



Sukses!

Keluarga dengan NKK 3101010101050001 berhasil diubah

Kode:

- menu Ubah Data Keluarga disimpan di navbar seperti berikut:

```
<li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>
```

- setelah menu Ubah Data Keluarga di-klik akan me-refer ke /keluarga/ubah pada KeluargaController yang berisi kode sebagai berikut:

```
@RequestMapping("/keluarga/ubah")  
public String ubahKeluarga () {  
    return "ubah-keluarga";  
}
```

- kemudian akan mengarah ke ubah-keluarga.html yang berisi form untuk memasukkan NKK yang akan diubah datanya

```
<div class="container-fluid text-center">  
  <div class="row content">  
    <div class="col-sm-2"></div>  
    <div class="col-sm-8 text-left">  
      <h1 class="page-header">Ubah Data Keluarga</h1>  
      <h3 class="page-header">Masukkan NKK keluarga yang ingin diubah</h3>  
      <form action="/keluarga/ubah-submit" method="GET">  
        <div class="form-group">  
          <label for="nkk">NKK</label> <input type="text" name="nkk" class="form-control" />  
        </div>  
        <button type="submit" class="btn btn-default">Submit</button>  
      </form>  
    </div>  
  </div>  
</div>
```

- setelah field NKK diisi, akan mengarah ke /keluarga/ubah-submit dengan method GET serta NKK sebagai parameternya

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

```
@RequestMapping(method = RequestMethod.GET, value="/keluarga/ubah-submit")
public String ubahKeluargaSubmit(Model model, @RequestParam(value = "nkk", required = true) String nkk) {
    return "redirect:/keluarga/ubah/" + nkk;
}
```

- Setelah itu akan me-refer lagi ke /keluarga/ubah/{nkk} dengan NKK yang tadi telah diambil dari input

```
@RequestMapping(value="/keluarga/ubah/{nkk}")
public String ubahKeluargaSubmitNKK (@PathVariable String nkk, Model model){

    KeluargaModel keluarga = keluargaService.selectKeluarga(nkk);

    if(keluarga == null) {
        model.addAttribute("nkk", nkk);
        return "not-found";
    } else {
        model.addAttribute("keluarga", keluarga);
        return "ubah-keluarga-form";
    }
}
```

- apabila NKK tersebut valid, akan diarahkan lagi ke ubah-keluarga-form.html yang akan berisi form untuk mengubah data-data keluarga. Berikut potongan kode ubah-keluarga-form.html

```
<h1 class="page-header">Ubah Data Keluarga</h1>
<form th:object="${keluarga}" action="/keluarga/ubah" method="POST">
    <div class="form-group">
        <label for="nomor_kk">NKK</label>
        <input type="text" class="form-control" id="nomor_kk" name="nomor_kk" readonly="true"
            th:field="*{nomor_kk}" th:value="${keluarga.nomor_kk}"/>
    </div>
    <div class="form-group">
        <label for="alamat">Alamat</label>
        <input type="text" class="form-control" id="alamat" name="alamat"
            th:field="*{alamat}" th:value="${keluarga.alamat}"/>
    </div>
    <div class="form-group">
        <label for="rt">RT</label>
        <input type="text" class="form-control" id="rt" name="rt"
            th:field="*{rt}" th:value="${keluarga.rt}"/>
    </div>
    <div class="form-group">
        <label for="rw">RW</label>
        <input type="text" class="form-control" id="rw" name="rw"
            th:field="*{rw}" th:value="${keluarga.rw}"/>
    </div>
    <div class="form-group">
        <label for="nama_kelurahan">Kelurahan/Desa</label>
        <input type="text" class="form-control" id="nama_kelurahan" name="nama_kelurahan"
            th:field="*{nama_kelurahan}" th:value="${keluarga.nama_kelurahan}"/>
    </div>
    <div class="form-group">
        <label for="nama_kecamatan">Kecamatan</label>
        <input type="text" class="form-control" id="nama_kecamatan" name="nama_kecamatan"
            th:field="*{nama_kecamatan}" th:value="${keluarga.nama_kecamatan}"/>
    </div>
    <div class="form-group">
        <label for="nama_kota">Kota</label>
        <input type="text" class="form-control" id="nama_kota" name="nama_kota"
            th:field="*{nama_kota}" th:value="${keluarga.nama_kota}"/>
    </div>
</form>
```

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

- setelah tombol submit yang tersedia di-klik, akan merefer ke /keluarga/ubah pada KeluargaController dengan method POST dengan input yang telah diambil sebagai parameter-nya

```
@RequestMapping(method = RequestMethod.POST, value="/keluarga/ubah")
public String ubahKeluargaData (@RequestParam(value="nomor_kk", required = true) String nomor_kk_lama,
    @RequestParam(value="alamat", required = true) String alamat,
    @RequestParam(value="rt", required = true) String rt,
    @RequestParam(value="rw", required = true) String rw,
    @RequestParam(value="nama_kelurahan", required = true) String nama_kelurahan,
    @RequestParam(value="nama_kecamatan", required = true) String nama_kecamatan,
    @RequestParam(value="nama_kota", required = true) String nama_kota, Model model) {
```

- setelah itu akan diubah NKK nya sesuai dengan id_kelurahan yang dicari dari nama_kelurahan (karena dari id_kelurahan dapat diketahui kecamatan dan kotanya)

```
Long id_kelurahan = keluargaService.cariIdKelurahan(nama_kelurahan);

String nomor_kk = keluargaService.createNKK(id_kelurahan);
```

- Berikut kode createNKK yang ada di KeluargaService:

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

```
@Override
public String createNKK(Long id_kelurahan) {

    Long id_kecamatan = keluargaMapper.selectIdKecamatan(id_kelurahan);
    String kode_kecamatan = keluargaMapper.selectKodeKecamatan(id_kecamatan);

    String kodeKecamatan = kode_kecamatan.substring(0, kode_kecamatan.length()-1);

    SimpleDateFormat df = new SimpleDateFormat("ddMMyy");
    Date date = new Date();

    String tanggalKK = df.format(date);

    String nkk = kodeKecamatan + tanggalKK + "%";
    List<String> NKKs = keluargaMapper.selectNKK(nkk);
    nkk = nkk.substring(0, nkk.length()-1);

    if(NKKs.size() == 0) {
        nkk += "0001";
    } else {
        Long nkkMAX = (long) 0;
        for(int i=0; i<NKKs.size(); i++) {
            Long tmp = Long.parseLong(NKKs.get(i));
            if(nkkMAX < tmp) {
                nkkMAX = tmp;
            }
        }
        nkk = Long.toString(nkkMAX + 1);
    }
    log.info("nkk {}", nkk);

    return nkk;
}
```

- Setelah mendapatkan NKK yang baru, akan dilakukan update data keluarga

```
keluarga.setNomor_kk(nomor_kk);
keluarga.setAlamat(alamat);
keluarga.setRt(rt);
keluarga.setRw(rw);
keluarga.setId_kelurahan(id_kelurahan);
keluarga.setIs_tidak_berlaku(is_tidak_berlaku);

keluargaService.updateKeluarga(keluarga, nomor_kk_lama);
```

- berikut method updateKeluarga yang ada pada KeluargaService

```
@Override
public void updateKeluarga(KeluargaModel keluarga, String nomor_kk_lama) {
    keluargaMapper.updateKeluarga(keluarga, nomor_kk_lama);
}
```

- Lalu berikut adalah query untuk meng-update data keluarga, dengan nik lama sebagai

parameternya

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

- setelah sudah di-update akan di-refer ke halaman sukses-ubah-keluarga.html dengan kode sebagai berikut

```
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="/">SI Kependudukan</a>
    </div>
    <ul class="nav navbar-nav">
      <li><a href="/">Home</a></li>
      <li><a href="/penduduk/tambah">Tambah Penduduk</a></li>
      <li><a href="/keluarga/tambah">Tambah Keluarga</a></li>
      <li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>
      <li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>
      <li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>
    </ul>
  </div>
</nav>

<h3>Sukses!</h3>
<h5 th:text="'Keluarga dengan NKK ' + ${nomor_kk_lama} + ' berhasil diubah'"></h5>

</body>
```

7. Mengubah Status Kematian Penduduk

- button ubah status kematian penduduk dapat diakses ketika melihat data penduduk berdasarkan NIK. Apabila status kematian penduduk masih hidup, tombol akan muncul

Lihat Data Penduduk - 3101010303890002

Nonaktifkan

NIK	3101010303890002
nama	Taswir Rajata
Jenis Kelamin	Pria
Tempat/Tanggal Lahir	Jakarta Fri Mar 03 00:00:00 ICT 1989
Alamat	Jln. Cikapayang No. 594
RT/RW	149/091
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	B+
Agama	Islam
Status Perkawinan	Kawin
Kewarganegaraan	WNA
Status Kematian	Hidup

namun apabila penduduk sudah wafat, tombol tidak akan tersedia

Lihat Data Penduduk - 3101011408150001

NIK	3101011408150001
nama	Mulyono Hutagalung S.Gz
Jenis Kelamin	Pria
Tempat/Tanggal Lahir	Jakarta Fri Aug 14 00:00:00 ICT 2015
Alamat	Jln. Cikapayang No. 594
RT/RW	149/091
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	B-
Agama	Islam
Status Perkawinan	Belum Kawin
Kewarganegaraan	WNI
Status Kematian	Mati

- ketika tombol Nonaktifkan di-klik, maka status is_wafat penduduk akan diubah dan akan menampilkan notifikasi serta tombol Nonaktifkan akan hilang

Lihat Data Penduduk - 3101010303890002

NIK	3101010303890002
nama	Taswir Rajata
Jenis Kelamin	Pria
Tempat/Tanggal Lahir	Jakarta Fri Mar 03 00:00:00 ICT 1989
Alamat	Jln. Cikapayang No. 594
RT/RW	149/091
Kelurahan/Desa	PULAU PARI
Kecamatan	KEPULAUAN SERIBU SELATAN
Kota	KABUPATEN KEPULAUAN SERIBU
Golongan Darah	B+
Agama	Islam
Status Perkawinan	Kawin
Kewarganegaraan	WNA
Status Kematian	Mati

Penduduk dengan NIK 3101010303890002 sudah tidak aktif

Kode:

- tombol Nonaktifkan ada di halaman penduduk-found.html yang mana akan menampilkan data penduduk berdasarkan NIK yang dicari

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

```
<tr>
  <td><strong>Status Kematian</strong></td>
  <td th:text="{is_wafat}"></td>
  <td th:if="{is_wafat} == 'Hidup'">
    <form action="/penduduk/mati" method="POST">
      <input type="hidden" name="nik" th:value="{nik}"></input>
      <button type="submit" class="btn btn-danger" name="action"
        value="submit">Nonaktifkan</button>
    </form>
  </td>
</tr>
```

- setelah button tersebut di-klik, akan di-refer ke /penduduk/mati dengan method POST

```
@RequestMapping(method = RequestMethod.POST, value="/penduduk/mati")
public String pendudukMati (@RequestParam(value="nik", required = true) String nik,
    RedirectAttributes redirectAtt) {

    pendudukService.ubahIsWafat(nik);
    pendudukService.nonAktifkanNKK(nik);
}
```

- setelah itu akan diubah status is_wafat nya dengan menggunakan method ubahIsWafat yang ada di PendudukService seperti ini:

```
@Override
public void ubahIsWafat(String nik) {
    pendudukMapper.ubahIsWafat(nik);
}
```

- dan query Update di PendudukMapper seperti ini:

```
@Update("UPDATE penduduk SET is_wafat = '1' WHERE nik = #{nik}")
void ubahIsWafat(String nik);
```

- setelah sudah di-update status is_wafat nya, akan dicek juga apakah dari keluarga tersebut masih ada penduduk yang hidup, karena kalau tidak ada status is_tidak_berlaku keluarga tersebut akan di-non aktifkan. Method NonaktifkanNKK dipanggil dari PendudukController seperti berikut:

```
pendudukService.nonAktifkanNKK(nik);
```

- kemudian berikut kode nonAktifkanNKK di PendudukServiceDatabase

```
@Override
public void nonAktifkanNKK(String nik) {

    String idKeluarga = pendudukMapper.selectIdKeluarga(nik);
    List<Integer> cek = pendudukMapper.cekStatusKematian(idKeluarga);

    int hidup = 0;

    for(int i = 0; i < cek.size(); i++) {
        if(cek.get(i) == 0) {
            hidup++;
            i++;
        } else {
            i++;
        }
    }

    if(hidup == 0) {
        pendudukMapper.nonAktifkanNKK(idKeluarga);
    }

}
```

- di method tersebut dipanggil juga method selectIdKeluarga dan cekStatusKematian yang ada pada pendudukMapper, hal ini berfungsi untuk mencari tahu daftar status kematian dari sebuah keluarga

```
@Select("SELECT id_keluarga FROM penduduk WHERE nik = #{nik}")
String selectIdKeluarga(String nik);

@Select("SELECT P.is_wafat FROM penduduk P, keluarga K WHERE P.id_keluarga = K.id AND K.id = #{id_keluarga_cari}")
List<Integer> cekStatusKematian(String id_keluarga_cari);
```

- setelah status is_wafat pada Penduduk dan status is_tidak_berlaku pada Keluarga berhasil diubah, akan di-refer ke halaman /penduduk?nik="" lagi dengan flash attribute

```
redirectAtt.addFlashAttribute("wafat", "berhasil");
return "redirect:/penduduk?nik=" + nik;
```

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

- menu Cari Penduduk Berdasarkan Lokasi dapat diakses di navbar

SI Kependudukan Home Tambah Penduduk Tambah Keluarga Ubah Data Penduduk Ubah Data Keluarga Cari Penduduk Berdasarkan Lokasi

- Ketika tombol tersebut di-klik akan muncul form untuk memilih nama kota dengan menu dropdown

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

Submit

Cari Data Penduduk

Pilih Kota

KABUPATEN KEPULAUAN SERIBU

- setelah kota dipilih, klik button submit, akan muncul pilihan nama kecamatan sesuai dengan nama kota yang dipilih, dan nama kota sudah tidak bisa diubah lagi

Submit

Cari Data Penduduk

Pilih Kota

KABUPATEN KEPULAUAN SERIBU

Pilih Kecamatan

KEPULAUAN SERIBU SELATAN

- setelah nama kecamatan dipilih, akan muncul form pilihan nama kelurahan, dan nama kota serta nama kecamatan tidak bisa diubah lagi

Submit

Cari Data Penduduk

Pilih Kota

KABUPATEN KEPULAUAN SERIBU

Pilih Kecamatan

KEPULAUAN SERIBU SELATAN

Pilih Kelurahan

PULAU TIDUNG

- setelah menentukan nama kota, kecamatan, dan kelurahan, button submit di-klik dan akan menampilkan data penduduk berdasarkan lokasi yang dipilih

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

Lihat Data Penduduk di KABUPATEN KEPULAUAN SERIBU, kecamatan KEPULAUAN SERIBU SELATAN, kelurahan PULAU TIDUNG

Show **10** entries Search:

NIK	Nama	Jenis Kelamin
3101010101170001	Caraka Marbun	Laki-laki
3101010102640001	Aris Edward Gunawan S.Sos	Laki-laki
3101010103810001	Karma Samosir	Laki-laki
3101010103920001	Wakiman Sitorus	Laki-laki
3101010103940001	Jarwi Sihombing S.E.I	Laki-laki
3101010104890001	Taswir Hutasoit	Laki-laki
3101010104910001	Mahfud Setiawan M.TI.	Laki-laki
3101010105890001	Radika Dabukke	Laki-laki
3101010109130001	Emong Pranowo M.M.	Laki-laki
3101010109810001	Jamal Najmudin S.T.	Laki-laki

Showing 1 to 10 of 272 entries Previous **1** 2 3 4 5 ... 28 Next

Kode:

- button Cari Penduduk Berdasarkan Lokasi terdapat di navbar

```
<li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>
```

- ketika button tersebut di-klik akan me-refer ke /penduduk/cari-form pada PendudukController

```
@RequestMapping("/penduduk/cari-form")
public String cariPendudukForm (Model model){

    List<String> semuaKota = pendudukService.selectAllKota();
    model.addAttribute("semuaKota", semuaKota);

    return "cari-penduduk";
}
```

- lalu akan memanggil method selectAllKota yang ada pada PendudukServiceDatabase untuk mengambil semua nama kota yang akan ditampilkan pada halaman form.

```
@Override
public List<String> selectAllKota() {
    List<String> semuaKota = pendudukMapper.selectAllKota();
    return semuaKota;
}
```

- berikut query untuk mengambil semua nama kota

```
@Select("SELECT nama_kota FROM kota")  
List<String> selectAllKota();
```

- setelah semua nama kota diambil akan di-refer ke halaman cari-penduduk.html
`model.addAttribute("semuaKota", semuaKota);`

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

- kemudian di halaman cari-penduduk.html akan ditampilkan nama-nama kotanya

```
<div class="container-fluid text-center">  
  <div class="row content">  
    <div class="col-sm-2"></div>  
    <div class="col-sm-8 text-left">  
      <h1 class="page-header">Cari Data Penduduk</h1>  
      <form action="/penduduk/cari-kecamatan" method="GET">  
        <div class="form-group">  
          <label for="kt">Pilih Kota</label>  
          <select class="form-control"  
            name="kt">  
            <option th:each="semuaKota, iterationStatus: ${semuaKota}"  
              th:text="${semuaKota}"></option>  
          </select>  
        </div>  
        <button type="submit" class="btn btn-default">Submit</button>  
      </form>  
    </div>  
  </div>  
</div>
```

- setelah nama kota dipilih dan tombol submit di-klik, akan me-refer ke /penduduk/cari-kecamatan dengan method GET serta nama kota sebagai paramaternya

```
@RequestMapping(method = RequestMethod.GET, value="/penduduk/cari-kecamatan")  
public String cariPendudukKecamatan (Model model, @RequestParam(value = "kt", required = true) String nama_kota) {
```

- setelah itu dicari id kota sesuai dengan nama kota yang dipilih, baru kemudian diambil semua nama kecamatan berdasarkan id kota yang ada

```
Long id_kota = pendudukService.cariIdKota(nama_kota);  
  
List<String> semuaKecamatan = pendudukService.selectAllKecamatan(id_kota);
```

- berikut method untuk selectAllKecamatan yang ada di PendudukServiceDatabase serta query yang digunakan yang ada pada PendudukMapper

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

```
@Override
public List<String> selectAllKecamatan(Long id_kota) {
    List<String> semuaKecamatan = pendudukMapper.selectAllKecamatan(id_kota);
    return semuaKecamatan;
}

>Select("SELECT nama_kecamatan FROM kecamatan WHERE id_kota = #{id_kota_cari}")
List<String> selectAllKecamatan(Long id_kota_cari);
```

- setelah semua nama kecamatan didapatkan, akan ditampilkan ke halaman cari-penduduk-2.html yang akan menampilkan daftar nama kecamatan beserta nama kota yang telah dipilih

```
model.addAttribute("nama_kota", nama_kota);
model.addAttribute("semuaKecamatan", semuaKecamatan);

return "cari-penduduk-2";
```

- berikut potongan kode cari-penduduk-2

```
<div class="container-fluid text-center">
    <div class="row content">
        <div class="col-sm-2"></div>
        <div class="col-sm-8 text-left">
            <h1 class="page-header">Cari Data Penduduk</h1>
            <form action="/penduduk/cari-kelurahan" method="GET">
                <div class="form-group">

                    <label for="kt">Pilih Kota</label>
                    <input type="text" class="form-control" name="kt" readonly="true"
                        th:value="${nama_kota}"/>

                    <label for="kc">Pilih Kecamatan</label>
                    <select class="form-control"
                        name="kc">
                        <option th:each="semuaKecamatan, iterationStatus: ${semuaKecamatan}"
                            th:text="${semuaKecamatan}"></option>
                    </select>

                </div>
                <button type="submit" class="btn btn-default">Submit</button>

            </form>
        </div>
    </div>
</div>
```

- lalu setelah button submit di-klik akan di refer ke halaman /penduduk/cari-kelurahan dengan method GET serta parameter nama kecamatan dan nama kota yang dipilih.

```
@RequestMapping(method = RequestMethod.GET, value="/penduduk/cari-kelurahan")
public String cariPendudukKelurahan (Model model, @RequestParam(value = "kc", required = true) String nama_kecamatan,
    @RequestParam(value = "kt", required = true) String nama_kota) {

    Long id_kecamatan = pendudukService.cariIdKecamatan(nama_kecamatan);

    List<String> semuaKelurahan = pendudukService.selectAllKelurahan(id_kecamatan);
    model.addAttribute("nama_kota", nama_kota);
    model.addAttribute("nama_kecamatan", nama_kecamatan);
    model.addAttribute("semuaKelurahan", semuaKelurahan);

    return "cari-penduduk-3";
}
```

- kemudian setelah itu akan dilakukan selectAllKelurahan menggunakan parameter id kecamatan yang telah didapatkan dari nama kecamatan yang sudah dipilih. Setelah nama seluruh kelurahan didapatkan, akan di-refer ke halaman cari-penduduk-3 yang akan menampilkan pilihan nama kelurahan

```
<div class="container-fluid text-center">
    <div class="row content">
        <div class="col-sm-2"></div>
        <div class="col-sm-8 text-left">
            <h1 class="page-header">Cari Data Penduduk</h1>

            <form action="/penduduk/cari" method="GET">
                <div class="form-group">

                    <label for="kt">Pilih Kota</label> <input type="text"
                        class="form-control" name="kt" readonly="true"
                        th:value="${nama_kota}" /> <label for="kc">Pilih
                        Kecamatan</label> <input type="text" class="form-control" name="kc"
                        readonly="true" th:value="${nama_kecamatan}" /> <label for="kl">Pilih
                        Kelurahan</label> <select class="form-control" name="kl">
                            <option
                                th:each="semuaKelurahan, iterationStatus: ${semuaKelurahan}"
                                th:text="${semuaKelurahan}"></option>
                        </select>

                </div>
                <button type="submit" class="btn btn-default">Submit</button>

            </form>
        </div>
    </div>
</div>
```

- setelah button submit di-klik akan di-refer ke /penduduk/cari dengan method GET serta parameter nama kota, nama kelurahan, dan nama kecamatan

```
@RequestMapping(method = RequestMethod.GET, value="/penduduk/cari")
public String cariPenduduk (Model model, @RequestParam(value = "kt", required = true) String nama_kota,
    @RequestParam(value = "kc", required = true) String nama_kecamatan,
    @RequestParam(value = "kl", required = true) String nama_kelurahan) {
```

- setelah itu akan dilakukan pencarian nama-nama penduduk berdasarkan id_kelurahan yang telah didapatkan dari nama kelurahan yang telah dipilih, berikut kode untuk mencari penduduk berdasarkan kelurahan yang ada di PendudukServiceDatabase dan PendudukMapper:

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

```
@Override
public List<PendudukModel> cariPendudukKelurahan (Long id_kelurahan) {
    List<PendudukModel> penduduk = pendudukMapper.cariPendudukKelurahan(id_kelurahan);
    return penduduk;
}

>Select("select P.nik, P.nama, P.jenis_kelamin "
+ "FROM PENDUDUK P, KELUARGA K, KELURAHAN KL WHERE P.id_keluarga = K.id AND "
+ "K.id_kelurahan = KL.id AND KL.id = #{id_kelurahan}")
List<PendudukModel> cariPendudukKelurahan (Long id_kelurahan);
```

- setelah semua nama penduduk telah didapatkan, akan di-refer ke halaman cari-penduduk-berhasil.html dengan nama kota, kecamatan, kelurahan dan pendudukModel sebagai parameter

```
model.addAttribute("nama_kota", nama_kota);
model.addAttribute("nama_kecamatan", nama_kecamatan);
model.addAttribute("nama_kelurahan", nama_kelurahan);
model.addAttribute("penduduk", penduduk);

return "cari-penduduk-berhasil";
```

- berikut potongan kode cari-penduduk-berhasil.html yang telah mengimplementasi datatables

```
<div class="container-fluid text-center">
    <div class="row content">
        <div class="col-sm-2"></div>
        <div class="col-sm-8 text-left">
            <h3 class="page-header">
                th:text="Lihat Data Penduduk di ' + ${nama_kota} + ', kecamatan ' + ${nama_kecamatan} + ', kelurahan ' + ${nama_kelurahan}"></h3>

            <table id="view-data" class="display" cellspacing="0" width="100%">
                <thead>
                    <tr>
                        <th>NIK</th>
                        <th>Nama</th>
                        <th>Jenis Kelamin</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        th:each="penduduk, iterationStatus: ${penduduk}">
                            <td th:text="${penduduk.nik}">NIK</td>
                            <td th:text="${penduduk.nama}">Nama Lengkap</td>
                            <td
                                th:text="${penduduk.jenis_kelamin == 1} ? 'Perempuan' : 'Laki-Laki'"></td>
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
```

9. ****BONUS**** Validasi Input Untuk Semua Form POST
[TIDAK DIKERJAKAN]

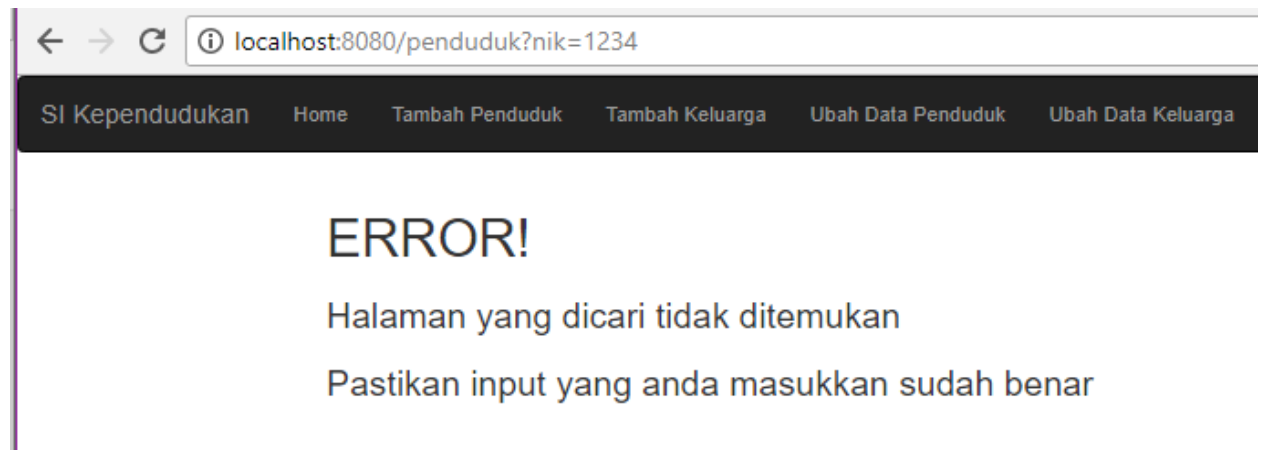
10. ****BONUS**** Menampilkan Penduduk Paling Muda dan Paling Tua di Suatu Kelurahan
[TIDAK DIKERJAKAN]

11. ****BONUS**** Menambahkan Error Page

- Dalam program ini telah dibuat halaman error yang akan muncul apabila *user* mengakses halaman yang tidak tersedia yang bisa disebabkan ketika mencari data penduduk/keluarga dengan nik/nkk yang tidak sesuai, mengubah data penduduk/keluarga dengan nik/nkk yang tidak sesuai atau mencari penduduk/keluarga di lokasi yang tidak tersedia. Berikut contohnya:

Lihat Data Penduduk Berdasarkan NIK

Masukkan Nomor Induk Kependudukan



kode:

```
<title>Sistem Kependudukan DKI</title>
</head>

<body>
  <nav class="navbar navbar-inverse">
    <div class="container-fluid">
      <div class="navbar-header">
        <a class="navbar-brand" href="/">SI Kependudukan</a>
      </div>
      <ul class="nav navbar-nav">
        <li><a href="/">Home</a></li>
        <li><a href="/penduduk/tambah">Tambah Penduduk</a></li>
        <li><a href="/keluarga/tambah">Tambah Keluarga</a></li>
        <li><a href="/penduduk/ubah">Ubah Data Penduduk</a></li>
        <li><a href="/keluarga/ubah">Ubah Data Keluarga</a></li>
        <li><a href="/penduduk/cari-form">Cari Penduduk Berdasarkan Lokasi</a></li>
      </ul>
    </div>
  </nav>

  <div class="container-fluid text-center">
    <div class="row content">
      <div class="col-sm-2"></div>
      <div class="col-sm-8 text-left">

        <h1>ERROR!</h1>
        <h3>Halaman yang dicari tidak ditemukan</h3>
        <h3>Pastikan input yang anda masukkan sudah benar</h3>

      </div>
    </div>
  </div>
```

12. ****BONUS**** Fitur-Fitur Lain yang Mendukung Aplikasi [TIDAK DIKERJAKAN]

B. Optimasi Database

Hal yang dilakukan agar database lebih optimal:

- Menjadikan atribut penting seperti ID dari setiap table sebagai primary key
- Melakukan indexing untuk semua table berdasarkan ID nya agar mempermudah dan mempercepat proses searching
- Membuat ID di table penduduk dan keluarga menjadi auto increment sehingga ketika data ditambah menjadi lebih otomatis

C. Stress Testing

Dalam melakukan *stress testing* saya menggunakan Apache JMeter dengan Number of Threads = 100, dan Ramp-up Periode = 50, dengan begitu akan dibutuhkan waktu 50 detik untuk mencapai jumlah thread menjadi 100, sehingga thread baru akan muncul setiap 0.5 detik. Berikut adalah hasilnya:

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

- <http://localhost:8080/penduduk?nik=3101011408150001>

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	23:32:28.579	Thread Group 1-8	HTTP Request	2451	✔	2946	147	2417	3
2	23:32:26.140	Thread Group 1-3	HTTP Request	4890	✔	2946	147	4857	2
3	23:32:28.071	Thread Group 1-7	HTTP Request	2959	✔	2946	147	2926	2
4	23:32:27.068	Thread Group 1-5	HTTP Request	3962	✔	2946	147	3926	3
5	23:32:29.071	Thread Group 1-9	HTTP Request	1959	✔	2946	147	1923	2
6	23:32:25.633	Thread Group 1-2	HTTP Request	5397	✔	2946	147	5360	480
7	23:32:27.669	Thread Group 1-6	HTTP Request	3361	✔	2946	147	3327	3
8	23:32:26.547	Thread Group 1-4	HTTP Request	4483	✔	2946	147	4450	5
9	23:32:25.633	Thread Group 1-1	HTTP Request	5397	✔	2946	147	5361	480
10	23:32:30.078	Thread Group 1-...	HTTP Request	2552	✔	2946	147	2551	2
11	23:32:30.592	Thread Group 1-...	HTTP Request	2365	✔	2946	147	2364	4
12	23:32:29.573	Thread Group 1-...	HTTP Request	3404	✔	2946	147	3404	3
13	23:32:31.077	Thread Group 1-...	HTTP Request	2377	✔	2946	147	2377	3
14	23:32:31.651	Thread Group 1-...	HTTP Request	1924	✔	2946	147	1924	3
15	23:32:32.136	Thread Group 1-...	HTTP Request	2549	✔	2946	147	2548	2
16	23:32:34.138	Thread Group 1-...	HTTP Request	571	✔	2946	147	571	4
17	23:32:32.633	Thread Group 1-...	HTTP Request	2077	✔	2946	147	2077	4
18	23:32:33.638	Thread Group 1-...	HTTP Request	1078	✔	2946	147	1077	3
19	23:32:33.138	Thread Group 1-...	HTTP Request	1603	✔	2946	147	1603	2
20	23:32:34.631	Thread Group 1-...	HTTP Request	258	✔	2946	147	257	2
21	23:32:35.636	Thread Group 1-...	HTTP Request	1025	✔	2946	147	1025	3
22	23:32:36.137	Thread Group 1-...	HTTP Request	525	✔	2946	147	525	2
23	23:32:35.135	Thread Group 1-...	HTTP Request	1535	✔	2946	147	1535	2
24	23:32:36.637	Thread Group 1-...	HTTP Request	187	✔	2946	147	186	3
25	23:32:37.637	Thread Group 1-...	HTTP Request	1722	✔	2946	147	1722	3
26	23:32:38.636	Thread Group 1-...	HTTP Request	723	✔	2946	147	723	3
27	23:32:37.154	Thread Group 1-...	HTTP Request	2330	✔	2946	147	2329	4
<div><input checked="" type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples? No of Samples 100 Latest Sample 1193 Average 1729 Deviation 1289</div>									

- <http://localhost:8080/keluarga?nkk=31010102170001>

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	23:36:21.007	Thread Group 1-1	HTTP Request	871	✓	3238	147	871	6
2	23:36:21.508	Thread Group 1-2	HTTP Request	883	✓	3238	147	883	3
3	23:36:22.008	Thread Group 1-3	HTTP Request	809	✓	3238	147	809	3
4	23:36:23.015	Thread Group 1-5	HTTP Request	796	✓	3238	147	795	4
5	23:36:22.511	Thread Group 1-4	HTTP Request	1301	✓	3238	147	1299	3
6	23:36:24.026	Thread Group 1-7	HTTP Request	585	✓	3238	147	585	7
7	23:36:23.524	Thread Group 1-6	HTTP Request	1252	✓	3238	147	1252	3
8	23:36:24.526	Thread Group 1-8	HTTP Request	1078	✓	3238	147	1078	5
9	23:36:25.027	Thread Group 1-9	HTTP Request	1332	✓	3238	147	1332	3
10	23:36:25.527	Thread Group 1....	HTTP Request	2354	✓	3238	147	2354	3
11	23:36:26.043	Thread Group 1....	HTTP Request	1844	✓	3238	147	1844	4
12	23:36:26.544	Thread Group 1....	HTTP Request	2506	✓	3238	147	2505	7
13	23:36:27.027	Thread Group 1....	HTTP Request	2943	✓	3238	147	2943	3
14	23:36:28.029	Thread Group 1....	HTTP Request	2805	✓	3238	147	2805	17
15	23:36:28.528	Thread Group 1....	HTTP Request	2660	✓	3238	147	2659	2
16	23:36:29.029	Thread Group 1....	HTTP Request	2159	✓	3238	147	2158	4
17	23:36:29.546	Thread Group 1....	HTTP Request	1647	✓	3238	147	1647	5
18	23:36:27.567	Thread Group 1....	HTTP Request	3626	✓	3238	147	3620	5
19	23:36:30.032	Thread Group 1....	HTTP Request	2319	✓	3238	147	2319	2
20	23:36:31.030	Thread Group 1....	HTTP Request	1475	✓	3238	147	1475	2
21	23:36:30.582	Thread Group 1....	HTTP Request	1962	✓	3238	147	1962	9
22	23:36:31.528	Thread Group 1....	HTTP Request	1029	✓	3238	147	1029	3
23	23:36:32.031	Thread Group 1....	HTTP Request	1555	✓	3238	147	1555	5
24	23:36:33.047	Thread Group 1....	HTTP Request	1850	✓	3238	147	1849	4
25	23:36:32.613	Thread Group 1....	HTTP Request	2286	✓	3238	147	2286	3
26	23:36:33.529	Thread Group 1....	HTTP Request	1370	✓	3238	147	1370	3
27	23:36:34.531	Thread Group 1....	HTTP Request	1677	✓	3238	147	1676	7

☐ Scroll automatically?

☐ Child samples?

No of Samples 34

Latest Sample 2106

Average 1795

Deviation 693

Tugas 1 APAP
Savira Nurul Ahila
1506689692
Kelas APAP - B

- <http://localhost:8080/penduduk/ubah/3101011408150001>

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	23:37:58.637	Thread Group 1-1	HTTP Request	304	✓	4755	148	304	11
2	23:37:59.132	Thread Group 1-2	HTTP Request	399	✓	4755	148	399	4
3	23:37:59.640	Thread Group 1-3	HTTP Request	226	✓	4755	148	225	8
4	23:38:00.145	Thread Group 1-4	HTTP Request	406	✓	4755	148	406	7
5	23:38:00.643	Thread Group 1-5	HTTP Request	251	✓	4755	148	251	5
6	23:38:01.144	Thread Group 1-6	HTTP Request	364	✓	4755	148	364	8
7	23:38:01.656	Thread Group 1-7	HTTP Request	201	✓	4755	148	201	8
8	23:38:02.148	Thread Group 1-8	HTTP Request	571	✓	4755	148	570	4
9	23:38:02.650	Thread Group 1-9	HTTP Request	179	✓	4755	148	179	3
10	23:38:03.157	Thread Group 1-...	HTTP Request	318	✓	4755	148	318	7
11	23:38:03.649	Thread Group 1-...	HTTP Request	212	✓	4755	148	212	6
12	23:38:04.170	Thread Group 1-...	HTTP Request	383	✓	4755	148	383	6
13	23:38:04.653	Thread Group 1-...	HTTP Request	254	✓	4755	148	254	6
14	23:38:05.151	Thread Group 1-...	HTTP Request	378	✓	4755	148	377	14
15	23:38:05.650	Thread Group 1-...	HTTP Request	174	✓	4755	148	174	2
16	23:38:06.151	Thread Group 1-...	HTTP Request	344	✓	4755	148	344	8
17	23:38:06.659	Thread Group 1-...	HTTP Request	254	✓	4755	148	254	15
18	23:38:07.164	Thread Group 1-...	HTTP Request	442	✓	4755	148	442	8
19	23:38:07.656	Thread Group 1-...	HTTP Request	163	✓	4755	148	162	3
20	23:38:08.172	Thread Group 1-...	HTTP Request	624	✓	4755	148	624	8
21	23:38:08.657	Thread Group 1-...	HTTP Request	204	✓	4755	148	204	3
22	23:38:09.653	Thread Group 1-...	HTTP Request	253	✓	4755	148	253	3
23	23:38:09.157	Thread Group 1-...	HTTP Request	749	✓	4755	148	749	5
24	23:38:10.671	Thread Group 1-...	HTTP Request	346	✓	4755	148	346	3
25	23:38:10.154	Thread Group 1-...	HTTP Request	863	✓	4755	148	863	2
26	23:38:11.159	Thread Group 1-...	HTTP Request	1391	✓	4755	148	1391	5
27	23:38:11.659	Thread Group 1-...	HTTP Request	891	✓	4755	148	891	8

☐ Scroll automatically? ☐ Child samples? No of Samples 33 Latest Sample 418 Average 473 Deviation 325

- <http://localhost:8080/keluarga/ubah/3101010101050001>

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	23:39:08.418	Thread Group 1-1	HTTP Request	519	✓	1719	148	518	15
2	23:39:08.981	Thread Group 1-2	HTTP Request	137	✓	1719	148	137	2
3	23:39:09.482	Thread Group 1-3	HTTP Request	211	✓	1719	148	211	4
4	23:39:10.017	Thread Group 1-4	HTTP Request	219	✓	1719	148	219	3
5	23:39:10.487	Thread Group 1-5	HTTP Request	156	✓	1719	148	156	3
6	23:39:10.985	Thread Group 1-6	HTTP Request	119	✓	1719	148	119	5
7	23:39:11.486	Thread Group 1-7	HTTP Request	141	✓	1719	148	141	2
8	23:39:11.914	Thread Group 1-8	HTTP Request	117	✓	1719	148	117	3
9	23:39:12.518	Thread Group 1-9	HTTP Request	187	✓	1719	148	187	4
10	23:39:12.987	Thread Group 1-...	HTTP Request	116	✓	1719	148	114	5
11	23:39:13.416	Thread Group 1-...	HTTP Request	146	✓	1719	148	146	2
12	23:39:13.985	Thread Group 1-...	HTTP Request	109	✓	1719	148	109	6
13	23:39:14.485	Thread Group 1-...	HTTP Request	289	✓	1719	148	289	3
14	23:39:14.986	Thread Group 1-...	HTTP Request	185	✓	1719	148	185	3
15	23:39:15.416	Thread Group 1-...	HTTP Request	129	✓	1719	148	129	2
16	23:39:15.986	Thread Group 1-...	HTTP Request	315	✓	1719	148	314	3
17	23:39:16.416	Thread Group 1-...	HTTP Request	186	✓	1719	148	186	2
18	23:39:16.917	Thread Group 1-...	HTTP Request	116	✓	1719	148	116	3
19	23:39:17.488	Thread Group 1-...	HTTP Request	119	✓	1719	148	119	2
20	23:39:17.988	Thread Group 1-...	HTTP Request	119	✓	1719	148	119	4
21	23:39:18.489	Thread Group 1-...	HTTP Request	238	✓	1719	148	237	3
22	23:39:18.988	Thread Group 1-...	HTTP Request	102	✓	1719	148	101	3
23	23:39:19.487	Thread Group 1-...	HTTP Request	147	✓	1719	148	146	3
24	23:39:19.988	Thread Group 1-...	HTTP Request	102	✓	1719	148	102	2
25	23:39:20.420	Thread Group 1-...	HTTP Request	140	✓	1719	148	140	2
26	23:39:20.987	Thread Group 1-...	HTTP Request	99	✓	1719	148	99	3
27	23:39:21.488	Thread Group 1-...	HTTP Request	283	✓	1719	148	283	3

☐ Scroll automatically? ☐ Child samples? No of Samples 30 Latest Sample 102 Average 174 Deviation 87

Kesimpulan stress testing:

Berdasarkan stress testing tersebut, menurut saya program yang dibuat tidak terlalu cepat namun tidak terlalu lambat juga. Program ini akan reliable apabila data yang digunakan tidak terlalu banyak, namun akan tidak reliable apabila data yang digunakan jumlahnya sangat banyak.