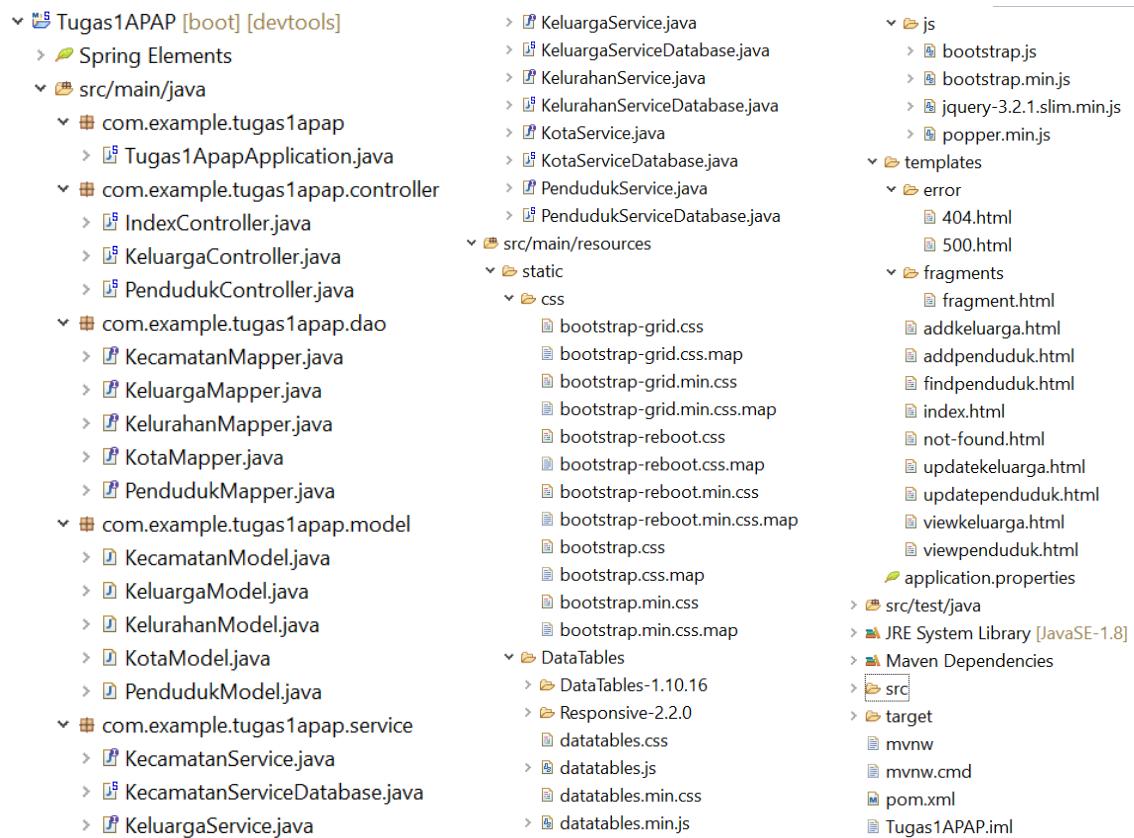


TUGAS 1

ARSITEKTUR DAN PERANCANGAN APLIKASI PERUSAHAAN

Write Up

1. STRUKTUR PROJECT



2. PACKAGE

Pada Tugas 1 ini, saya memiliki 5 *package* pada *project* ini. Yaitu:

- com.example.tugas1apap
 - Tugas1ApapApplication.java
- com.example.tugas1apap.controller
 - IndexController.java
 - @RequestMapping("/")

```
package com.example.tugas1apap.controller;
```

```
import org.springframework.stereotype.Controller;  
  
@Controller  
public class IndexController {
```

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

- KeluargaController.java
 - @RequestMapping("/keluarga")

```
@RequestMapping("/keluarga")  
public String index (@RequestParam(value = "nkk") String nkk, Model model) {  
    KeluargaModel km = keluargaDAO.selectKeluarga(nkk);  
  
    if(km != null) {  
        model.addAttribute("km", km);  
  
        KelurahanModel kelm = kelurahanDAO.selectKelurahan(km.getIdKelurahan());  
        model.addAttribute("kelm", kelm);  
  
        KecamatanModel kecm = kecamatanDAO.selectKecamatan(kelurahan.getIdKecamatan());  
        model.addAttribute("kecm", kecm);  
  
        KotaModel kotm = kotaDAO.selectKota(kecm.getIdKota());  
        model.addAttribute("kotm", kotm);  
  
        List<PendudukModel> selectAnggotaKeluarga = keluargaDAO.selectAnggotaKeluarga(km.getId());  
        model.addAttribute("listkeluarga", selectAnggotaKeluarga);  
  
        return "viewkeluarga";  
    } else {  
        model.addAttribute("nkk", nkk);  
        return "not-found";  
    }  
}
```

- @RequestMapping("/keluarga/tambah", GET)

```
@RequestMapping(value = "/keluarga/tambah", method = RequestMethod.GET)  
public String addKeluarga (PendudukModel penduduk, Model model) {  
    model.addAttribute("keluarga", new KeluargaModel());  
  
    List<KelurahanModel> kelurahans = kelurahanDAO.selectAllKelurahan();  
    model.addAttribute("kelurahans", kelurahans);  
  
    List<KecamatanModel> kecamatans = kecamatanDAO.selectAllKecamatan();  
    model.addAttribute("kecamatans", kecamatans);  
  
    List<KotaModel> kotas = kotaDAO.selectAllKota();  
    model.addAttribute("kotas", kotas);  
  
    return "addkeluarga";  
}
```

- **@RequestMapping("/keluarga/tambah", POST)**

```
@RequestMapping(value = "/keluarga/tambah", method = RequestMethod.POST)
public String addKeluargaPost (KeluargaModel keluarga, Model model) {
    model.addAttribute("keluarga", keluarga);

    String nkk = "";
    KelurahanModel kelm = kelurahanDAO.selectKelurahan(keluarga.getIdKelurahan());
    nkk += kelm.getKodeKelurahan().substring(0, 6);

    DateFormat dateFormat = new SimpleDateFormat("ddMMyy");
    Date date = new Date();

    String currentDate = dateFormat.format(date);
    nkk += currentDate;

    int offset = Integer.parseInt(keluargaDAO.findSimilarNKK(nkk)) + 1;
    if(offset < 10)
        nkk += "00" + offset;
    else if(offset < 100)
        nkk += "0" + offset;
    else if(offset < 1000)
        nkk += "0" + offset;
    else
        nkk += offset;
    keluarga.setNomorKk(nkk);
    keluarga.setIsTidakBerlaku("0");

    String lastID = keluargaDAO.selectLastID();
    keluarga.setId("0" + (Integer.parseInt(lastID) + 1));
    keluargaDAO.addKeluarga(keluarga);
    model.addAttribute("nkk", nkk);
    return "addkeluarga";
}
```

- **@RequestMapping("/keluarga/ubah/{NKK}", GET)**

```
@RequestMapping(value = "/keluarga/ubah/{NKK}", method = RequestMethod.GET)
public String updateKeluarga (Model model, @PathVariable(value = "NKK") String nkk)
{
    KeluargaModel keluarga = keluargaDAO.selectKeluarga(nkk);

    if (keluarga != null) {
        model.addAttribute("keluarga", keluarga);

        KelurahanModel currentKelurahan = kelurahanDAO.selectKelurahan(keluarga.getIdKelurahan());
        KecamatanModel currentKecamatan = kecamatanDAO.selectKecamatan(currentKelurahan.getIdKecamatan());
        KotaModel currentKota = kotaDAO.selectKota(currentKecamatan.getIdKota());

        List<KelurahanModel> kelurahans = kelurahanDAO.selectAllKelurahan();
        model.addAttribute("kelurahans", kelurahans);
        model.addAttribute("currentKelurahan", currentKelurahan.getId());

        List<KecamatanModel> kecamatans = kecamatanDAO.selectAllKecamatan();
        model.addAttribute("kecamatans", kecamatans);
        model.addAttribute("currentKecamatan", currentKecamatan.getId());

        List<KotaModel> kotas = kotaDAO.selectAllKota();
        model.addAttribute("kotas", kotas);
        model.addAttribute("currentKota", currentKota.getId());
    }
    else {
        model.addAttribute("nkk", nkk);
        return "not-found";
    }
}
```

- **@RequestMapping("/keluarga/ubah/{NKK}", POST)**

```
@RequestMapping(value = "/keluarga/ubah/{NKK}", method = RequestMethod.POST)
public String updateKeluargaPost (KeluargaModel keluarga, Model model, @PathVariable(value = "NKK") String nkk)
{
    model.addAttribute("keluarga", keluarga);
    String nkkLama = nkk;
    KeluargaModel keluargaOld = keluargaDAO.selectKeluarga(nkkLama);

    keluarga.setNomorKk(nkkLama);
    keluarga.setIsTidakBerlaku(keluargaOld.getIsTidakBerlaku());

    if(!keluargaOld.getIdKelurahan().equalsIgnoreCase(keluarga.getIdKelurahan())) {
        String nkkBaru = "";
        KelurahanModel kelm = kelurahanDAO.selectKelurahan(keluarga.getIdKelurahan());
        nkkBaru += kelm.getKodeKelurahan().substring(0, 6);

        DateFormat dateFormat = new SimpleDateFormat("ddMMyy");
        Date date = new Date();

        String currentDate = dateFormat.format(date);
        nkk += currentDate;

        int offset = Integer.parseInt(keluargaDAO.findSimilarNKK(nkkBaru)) + 1;
        if(offset < 10)
```

```

        if(offset < 10)
            nkkBaru += "000" + offset;
        else if(offset < 100)
            nkkBaru += "00" + offset;
        else if(offset < 1000)
            nkkBaru += "0" + offset;
        else
            nkkBaru += offset;
    keluarga.setNomorKK(nkkBaru);
}
keluargaDAO.updateKeluarga(keluarga, nkkLama);
model.addAttribute("nkkLama", nkkLama);
return "updatekeluarga";
}
}

```

- PendudukController.java

- `@RequestMapping("/penduduk")`

```

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

    if(pm != null) {
        String tanggalLahirLengkap = pm.getTanggalLahir();
        String tanggalLahir = tanggalLahirLengkap.substring(8, 10);
        String bulanLahir = tanggalLahirLengkap.substring(5, 7);
        String tahunLahir = tanggalLahirLengkap.substring(0, 4);

        model.addAttribute("pm", pm);
        model.addAttribute("tanggalLahir", tanggalLahir + "-" + bulanLahir + "-" + tahunLahir);
        int jenisKelamin = pm.getJenisKelamin();
        if(jenisKelamin == 0)
            model.addAttribute("jenisKelamin", "Pria");
        else
            model.addAttribute("jenisKelamin", "Wanita");

        if(pm.getIsWafat().equals("1"))
            model.addAttribute("statusKematian", "Wafat");
        else
            model.addAttribute("statusKematian", "Hidup");

        if(pm.getIsWni().equals("1"))
            model.addAttribute("isWni", "WNI");
        else
            model.addAttribute("isWni", "WNA");
    }
    KeluargaModel km = keluargaDAO.selectKeluargaWithID(pm.getIdKeluarga());
    model.addAttribute("km", km);

    KelurahanModel kem = kelurahanDAO.selectKelurahan(km.getIdKelurahan());
    model.addAttribute("kem", kem);

    KecamatanModel kec = kecamatanDAO.selectKecamatan(kem.getIdKecamatan());
    model.addAttribute("kec", kec);

    KotaModel kot = kotaDAO.selectKota(kec.getIdKota());
    model.addAttribute("kot", kot);

    return "viewpenduduk";
} else {
    model.addAttribute("nik", nik);
    return "not-found";
}
}

```

- `@RequestMapping("/penduduk/tambah", GET)`

```

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

```

- `@RequestMapping("/penduduk/tambah", POST)`

- ```

 @RequestMapping(value = "/penduduk/tambah", method = RequestMethod.POST)
 public String addPendudukPost (PendudukModel penduduk, Model model) {
 model.addAttribute("penduduk", penduduk);

 String nik = "";
 String tanggalLahirLengkap = penduduk.getTanggalLahir();
 String tanggalLahir = tanggalLahirLengkap.substring(8, 10);
 String bulanLahir = tanggalLahirLengkap.substring(5, 7);
 String tahunLahir = tanggalLahirLengkap.substring(2, 4);

 KeluargaModel kem = keluargaDAO.selectKeluargaWithID(penduduk.getIdKeluarga());
 KelurahanModel kelm = kelurahanDAO.selectKelurahan(kem.getIdKelurahan());
 nik += kelm.getKodeKelurahan().substring(0, 6);

 if(penduduk.getJenisKelamin() == 0)
 nik += tanggalLahir;
 else
 nik += (Integer.parseInt(tanggalLahir) + 40);

 nik += bulanLahir;
 nik += tahunLahir;

 int offset = Integer.parseInt(pendudukDAO.findSimilarNIK(nik)) + 1;

 if(offset < 10)
 nik += "000" + offset;
 else if(offset < 100)
 if(offset < 10)
 nik += "000" + offset;
 else if(offset < 100)
 nik += "00" + offset;
 else if(offset < 1000)
 nik += "0" + offset;
 else
 nik += offset;
 penduduk.setNik(nik);

 String lastID = pendudukDAO.selectLastID();
 penduduk.setId(Integer.parseInt(lastID) + 1);

 pendudukDAO.addPenduduk(penduduk);

 model.addAttribute("nik", nik);
 return "addpenduduk";
 }
}

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

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

■ @RequestMapping("/penduduk/ubah/{NIK}, POST)
 @RequestMapping(value = "/penduduk/ubah/{NIK}", method = RequestMethod.POST)
 public String updatePendudukPost (PendudukModel penduduk, Model model, @PathVariable(value = "NIK") String nik)
 {
 model.addAttribute("penduduk", penduduk);
 String niklama = nik;
 PendudukModel pendudukOld = pendudukDAO.selectPenduduk(niklama);

 penduduk.setNik(niklama);

 if(!pendudukOld.getTanggalLahir().equalsIgnoreCase(penduduk.getTanggalLahir()) ||
 pendudukOld.getJenisKelamin() != penduduk.getJenisKelamin() ||
 !pendudukOld.getIdKeluarga().equalsIgnoreCase(penduduk.getIdKeluarga())) {

 String nikBaru = "";
 String tanggalLahirLengkap = penduduk.getTanggalLahir();
 String tanggalLahir = tanggalLahirLengkap.substring(8, 10);
 String bulanLahir = tanggalLahirLengkap.substring(5, 7);
 String tahunLahir = tanggalLahirLengkap.substring(2, 4);

 KeluargaModel kem = keluargaDAO.selectKeluargaWithID(penduduk.getIdKeluarga());
 KelurahanModel kelm = kelurahanDAO.selectKelurahan(kem.getIdKelurahan());
 nikBaru += kelm.getKodeKelurahan().substring(0, 6);

 if(penduduk.getJenisKelamin() == 0)
 nikBaru += tanggalLahir;
 else
 nikBaru += (Integer.parseInt(tanggalLahir) + 40);
 }
 }
}

```

```

 --
 else
 nikBaru += (Integer.parseInt(tanggalLahir) + 40);

 nikBaru += bulanLahir;
 nikBaru += tahunLahir;

 int offset = Integer.parseInt(pendudukDAO.findSimilarNIK(nikBaru)) + 1;

 if(offset < 10)
 nikBaru += "000" + offset;
 else if(offset < 100)
 nikBaru += "00" + offset;
 else if(offset < 1000)
 nikBaru += "0" + offset;
 else
 nikBaru += offset;

 penduduk.setNik(nikBaru);
 }
}

```

- **@RequestMapping("/penduduk/mati", POST)**

```

@RequestMapping(value = "/penduduk/mati", method = RequestMethod.POST)
public String matikanPenduduk (String nik, RedirectAttributes redirectAttributes) {
 pendudukDAO.matikanPenduduk(nik);

 int countIsNotWafat = keluargaDAO.countIsNotWafat(pendudukDAO.selectPenduduk(nik).getIdKeluarga());
 if(countIsNotWafat == 0)
 keluargaDAO.setTidakBerlaku(pendudukDAO.selectPenduduk(nik).getIdKeluarga());

 redirectAttributes.addAttribute("nik", nik);
 redirectAttributes.addFlashAttribute("isWafat", true);
 redirectAttributes.addFlashAttribute("nik", nik);
 return "redirect:/penduduk";
}

```

- **@RequestMapping("/penduduk/cari")**

```

@RequestMapping("/penduduk/cari")
public String cariPenduduk (@RequestParam(value = "kt") Optional<String> kt,
 @RequestParam(value = "kc") Optional<String> kc,
 @RequestParam(value = "kl") Optional<String> kl,
 Model model) {
 List<KotaModel> kotas = kotaDAO.selectAllKota();
 model.addAttribute("kotas", kotas);

 if(kt.isPresent()) {
 String namaKota = kotaDAO.selectKota(kt.get()).getNamaKota();
 model.addAttribute("namaKota", namaKota);
 model.addAttribute("currentKota", kt.get());
 List<KecamatanModel> kecamatans = kecamatanDAO.selectKecamatanByKota(kt.get());
 model.addAttribute("kecamatans", kecamatans);
 }
 if(kc.isPresent()) {
 String namaKecamatan = kecamatanDAO.selectKecamatan(kc.get()).getNamaKecamatan();
 model.addAttribute("namaKecamatan", namaKecamatan);
 model.addAttribute("currentKecamatan", kc.get());
 List<KelurahanModel> kelurahans = kelurahanDAO.selectKelurahanByKecamatan(kc.get());
 model.addAttribute("kelurahans", kelurahans);
 }
 if(kl.isPresent()) {
 String namaKelurahan = kelurahanDAO.selectKelurahan(kl.get()).getNamaKelurahan();
 model.addAttribute("namaKelurahan", namaKelurahan);
 model.addAttribute("currentKelurahan", kl.get());
 }

 List<PendudukModel> pendudiks = pendudukDAO.selectPendudukByKelurahan(kl.get());
 model.addAttribute("pendudiks", pendudiks);

 PendudukModel youngestCitizen = pendudukDAO.selectPendudukTermuda(kl.get());
 PendudukModel oldestCitizen = pendudukDAO.selectPendudukTertua(kl.get());

 String tanggalLahirLengkap = youngestCitizen.getTanggalLahir();
 String tanggalLahir = tanggalLahirLengkap.substring(8, 10);
 String bulanLahir = tanggalLahirLengkap.substring(5, 7);
 String tahunLahir = tanggalLahirLengkap.substring(0, 4);
 String tanggalLahirBaru = tanggalLahir + "-" + bulanLahir + "-" + tahunLahir;

 model.addAttribute("tanggalLahirYoungest", tanggalLahirBaru);

 tanggalLahirLengkap = oldestCitizen.getTanggalLahir();
 tanggalLahir = tanggalLahirLengkap.substring(8, 10);
 bulanLahir = tanggalLahirLengkap.substring(5, 7);
 tahunLahir = tanggalLahirLengkap.substring(0, 4);
 tanggalLahirBaru = tanggalLahir + "-" + bulanLahir + "-" + tahunLahir;

 model.addAttribute("tanggalLahirOldest", tanggalLahirBaru);

 model.addAttribute("youngestCitizen", youngestCitizen);
 model.addAttribute("oldestCitizen", oldestCitizen);
}

return "findpenduduk";
}

```

- com.example.tugas1apap.dao

- KecamatanMapper.java

```

import java.util.List;

@Mapper
public interface KecamatanMapper {
 @Select("select * " +
 "from kecamatan " +
 "where id = #{id}")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="idKota", column="id_kota"),
 @Result(property="kodeKecamatan", column="kode_kecamatan"),
 @Result(property="namaKecamatan", column="nama_kecamatan"),
 }) KecamatanModel selectKecamatan (@Param("id") String id);

 @Select("select * from kecamatan order by nama_kecamatan asc")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="kodeKecamatan", column="kode_kecamatan"),
 @Result(property="idKota", column="id_kota"),
 @Result(property="namaKecamatan", column="nama_kecamatan")
 })
 List<KecamatanModel> selectAllKecamatan ();

 @Select("select * from kecamatan where id_kota = #{idKota} order by nama_kecamatan asc")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="kodeKecamatan", column="kode_kecamatan"),
 @Result(property="idKota", column="id_kota"),
 @Result(property="namaKecamatan", column="nama_kecamatan")
 })
 List<KecamatanModel> selectKecamatanByKota (@Param("idKota") String idKota);
}

```

- KeluargaMapper.java

```

package com.example.tugas1apap.dao;

import java.util.List;

@Mapper
public interface KeluargaMapper {
 @Select("select * " +
 "from keluarga " +
 "where nomor_kk = #{nkk}")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="nomorKK", column="nomor_kk"),
 @Result(property="alamat", column="alamat"),
 @Result(property="rt", column="RT"),
 @Result(property="rw", column="RW"),
 @Result(property="idKelurahan", column="id_kelurahan"),
 @Result(property="isTidakBerlaku", column="is_tidak_berlaku")
 }) KeluargaModel selectKeluarga (@Param("nkk") String nkk);

 @Select("select * " +
 "from keluarga " +
 "where id = #{id}")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="nomorKK", column="nomor_kk"),
 @Result(property="alamat", column="alamat"),
 @Result(property="rt", column="RT"),
 @Result(property="rw", column="RW"),
 @Result(property="idKelurahan", column="id_kelurahan"),
 @Result(property="isTidakBerlaku", column="is_tidak_berlaku")
 }) KeluargaModel selectKeluargaWithID (@Param("id") String id);

 @Select("select * " +
 "from penduduk " +
 "where id_keluarga = #{id}")
 @Results(value = {
 @Result(property="nik", column="nik"),
 @Result(property="name", column="name"),
 @Result(property="tempatlahir", column="tempat_lahir"),
 @Result(property="tanggallahir", column="tangggal_lahir"),
 @Result(property="jeniskelamin", column="jenis_kelamin"),
 @Result(property="is_wni", column="is_wni"),
 @Result(property="idKeluarga", column="id_keluarga"),
 @Result(property="agama", column="agama"),
 @Result(property="occupation", column="pekerjaan"),
 @Result(property="statusPerekawinan", column="status_perekawinan"),
 @Result(property="statusDalamKeluarga", column="status_dalam_keluarga"),
 @Result(property="pendidikan", column="pendidikan"),
 @Result(property="golongan_darah", column="golongan_darah"),
 @Result(property="is_wafat", column="is_wafat")
 }) List<PendudukModel> selectAnggotaKeluarga(@Param("id") String id);

 @Select("select count(*) as nkk " +
 "from keluarga " +
 "where nomor_kk like #{nkkPart}")
 String findSimilarNKK (@Param("nkkPart") String nkkPart);

 @Select("SELECT id FROM keluarga ORDER BY ID DESC LIMIT 1")
 String selectlastID();

 @Insert("insert into keluarga (id, nomor_kk, alamat, RT, RW, id_kelurahan, is_tidak_berlaku) " +
 "+ values (#{keluarga.id}, #{keluarga.nomorKk}, #{keluarga.alamat}, #{keluarga.rt}, #{keluarga.rw}, " +
 "+ #{keluarga.idKelurahan}, #{keluarga.isTidakBerlaku})")
 void addKeluarga(@Param("Keluarga") KeluargaModel keluarga);
}

```

```

 @Update("update keluarga set nomor_kk=#{keluarga.nomorKk}, alamat=#{keluarga.alamat}, RT=#{keluarga.rt}, RW=#{keluarga.rw}, "
 + "id_kelurahan=#{keluarga.idKelurahan}, is_tidak_berlaku=#{keluarga.isTidakBerlaku} "
 + "where nomor_kk = #{nkkLama}")
 void updateKeluarga(
 @Param("keluarga") KeluargaModel keluarga,
 @Param("nkkLama") String nkkLama
);

 @Select("select count(*) from penduduk where id_keluarga=#{idKeluarga} and is_wafat=0")
 int countIsNotWafat(@Param("idKeluarga") String idKeluarga);

 @Update("update keluarga set is_tidak_berlaku=1 where id=#{idKeluarga}")
 void setTidakBerlaku(@Param("idKeluarga") String idKeluarga);
}

```

### ○ KelurahanMapper.java

```

public interface KelurahanMapper {
 @Select("select * " +
 "from kelurahan " +
 "where id = #{id}")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="kodeKelurahan", column="kode_kelurahan"),
 @Result(property="idKecamatan", column="id_kecamatan"),
 @Result(property="namaKelurahan", column="nama_kelurahan"),
 @Result(property="kodePos", column="kode_pos")
 }) KelurahanModel selectKelurahan (@Param("id") String id);

 @Select("select * from kelurahan order by nama_kelurahan asc")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="kodeKelurahan", column="kode_kelurahan"),
 @Result(property="idKecamatan", column="id_kecamatan"),
 @Result(property="namaKelurahan", column="nama_kelurahan"),
 @Result(property="kodePos", column="kode_pos")
 })
 List<KelurahanModel> selectAllKelurahan ();

 @Select("select * from kelurahan where id_kecamatan = #{idKecamatan} order by nama_kelurahan asc")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="kodeKelurahan", column="kode_kelurahan"),
 @Result(property="idKecamatan", column="id_kecamatan"),
 @Result(property="namaKelurahan", column="nama_kelurahan"),
 @Result(property="kodePos", column="kode_pos")
 })
 List<KelurahanModel> selectKelurahanByKecamatan (@Param("idKecamatan") String idKecamatan);
}

```

### ○ KotaMapper.java

```

package com.example.tugas1apap.dao;

import java.util.List;

@Mapper
public interface KotaMapper {
 @Select("select * " +
 "from kota " +
 "where id = #{id}")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="kodeKota", column="kode_kota"),
 @Result(property="namaKota", column="nama_kota")
 }) KotaModel selectKota (@Param("id") String id);

 @Select("select * from kota order by nama_kota asc")
 @Results(value = {
 @Result(property="id", column="id"),
 @Result(property="kodeKota", column="kode_kota"),
 @Result(property="namaKota", column="nama_kota")
 })
 List<KotaModel> selectAllKota ();
}

```

- PendudukMapper.java

```

package com.example.tugas1apap.dao;

import java.util.List;

@Mapper
public interface PendudukMapper {
 @Select("select * " +
 "from penduduk " +
 "where nik = #{nik}")
 @Results(value = {
 @Result(property="nik", column="nik"),
 @Result(property="nama", column="nama"),
 @Result(property="tempatLahir", column="tempat_lahir"),
 @Result(property="tanggalLahir", column="tanggal_lahir"),
 @Result(property="jenisKelamin", column="jenis_kelamin"),
 @Result(property="isWni", column="is_wni"),
 @Result(property="idKeluarga", column="id_keluarga"),
 @Result(property="agama", column="agama"),
 @Result(property="pekerjaan", column="pekerjaan"),
 @Result(property="statusPerkawinan", column="status_perkawinan"),
 @Result(property="statusDalamKeluarga", column="status_dalam_keluarga"),
 @Result(property="golonganDarah", column="golongan_darah"),
 @Result(property="isWafat", column="is_wafat")
 }) PendudukModel selectPenduduk (@Param("nik") String nik);

 @Select("select count(*) as nik " +
 "from penduduk " +
 "where nik like #{nikPart}")
 String findSimilarNIK (@Param("nikPart") String nikPart);

 @Select("SELECT id FROM penduduk ORDER BY ID DESC LIMIT 1")
 String selectLastID();

 @Insert("insert into penduduk (id, 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.id}, #{penduduk.nik}, #{penduduk.nama}, #{penduduk.tempatLahir}, #{penduduk.tanggalLahir}, "
 + "#{penduduk.jenisKelamin}, #{penduduk.isWni}, #{penduduk.idKeluarga}, #{penduduk.agama}, #{penduduk.pekerjaan}, "
 + "#{penduduk.statusPerkawinan}, #{penduduk.statusDalamKeluarga}, #{penduduk.golonganDarah}, #{penduduk.isWafat}")
 void addPenduduk(@Param("penduduk") PendudukModel penduduk);

 @Update("update penduduk set nik=#{penduduk.nik}, nama=#{penduduk.name}, tempat_lahir=#{penduduk.tempatLahir}, tanggal_lahir=#{"
 + "where nik=#{niklama})"
 void updatePenduduk(
 @Param("penduduk") PendudukModel penduduk,
 @Param("niklama") String nikLama
);

 @Update("update penduduk set is_wafat=1 "
 + "where nik=#{nik}")
 void matikanPenduduk(@Param("nik") String nik);

 @Select("select p.nik, p.nama, p.jenis_kelamin from penduduk p join keluarga k on p.id_keluarga=k.id join kelurahan kel on k.id"
 + "Result(value = {
 @Result(property="nik", column="nik"),
 @Result(property="nama", column="nama"),
 @Result(property="jenisKelamin", column="jenis_kelamin"))
 List<PendudukModel> selectPendudukByKelurahan(@Param("kode_kelurahan") String kode_kelurahan);

 @Select("select p.nik, p.nama, p.tanggal_lahir from penduduk p join keluarga k on p.id_keluarga=k.id join kelurahan kel on k.id"
 + "Result(value = {
 @Result(property="tanggalLahir", column="tanggal_lahir"))
 PendudukModel selectPendudukTermauda(@Param("id_kelurahan") String id_kelurahan);

 @Select("select p.nik, p.nama, p.tanggal_lahir from penduduk p join keluarga k on p.id_keluarga=k.id join kelurahan kel on k.id"
 + "Result(value = {
 @Result(property="tanggalLahir", column="tanggal_lahir"))
 PendudukModel selectPendudukTertua(@Param("id_kelurahan") String id_kelurahan);
}

```

- com.example.tugas1apap.model

- KecamatanModel.java
- KeluargaModel.java
- KelurahanModel.java
- KotaModel.java
- PendudukModel.java

- com.example.tugas1apap.service

- KecamatanService.java
  - KecamatanModel selectKecamatan(String id);
  - List<KecamatanModel> selectAllKecamatan();
  - List<KecamatanModel> selectKecamatanByKota(String idKota);
- KecamatanServiceDatabase.java

- KeluargaService.java
  - KeluargaModel selectKeluarga(String nkk);
  - KeluargaModel selectKeluargaWithID(String id);
  - List<PendudukModel> selectAnggotaKeluarga(String id);
  - String findSimilarNKK(String nkkPart);
  - String selectLastID();
  - **void** addKeluarga(KeluargaModel keluarga);
  - **void** updateKeluarga(KeluargaModel keluarga, String nkkLama);
  - **int** countIsNotWafat(String idKeluarga);
  - **void** setTidakBerlaku(String idKeluarga);
- KeluargaServiceDatabase.java
- KelurahanService.java
  - KelurahanModel selectKelurahan(String id);
  - List<KelurahanModel> selectAllKelurahan();
  - List<KelurahanModel> selectKelurahanByKecamatan(String idKecamatan);
- KelurahanServiceDatabase.java
- KotaService.java
  - KotaModel selectKota(String id);
  - List<KotaModel> selectAllKota();
- KotaServiceDatabase.java
- PendudukService.java
  - PendudukModel selectPenduduk(String nik);
  - String findSimilarNIK(String nikPart);
  - **void** addPenduduk (PendudukModel penduduk);
  - String selectLastID();
  - **void** updatePenduduk (PendudukModel penduduk, String nikLama);
  - **void** matikanPenduduk (String nik);
  - List<PendudukModel> selectPendudukByKelurahan (String kode\_kelurahan);
  - PendudukModel selectPendudukTermuda(String id\_kelurahan);
  - PendudukModel selectPendudukTertua(String id\_kelurahan);
- PendudukServiceDatabase.java

### 3. IMPLEMENTASI MVC

Pada Tugas 1 ini saya menggunakan konsep MVC dengan *package-package* yang telah disebutkan diatas (**example, controller, dao, model, service**)

- **Example:** berfungsi untuk menjalankan keseluruhan program ini
- **Controller:** berfungsi untuk menjalankan fungsi yang pengguna telah lakukan pada *View*. Pada *Controller*, terdapat berbagai macam *logic* yang juga berinteraksi dengan *Model* dan juga *Service*. Pada *package* ini saya memiliki tiga *controller* utama yaitu **IndexController** (/) yang berguna untuk memproses perintah umum. Kemudian ada **PendudukController** (/penduduk) yang memroses seluruh aktivitas yang berhubungan dengan penduduk. Terakhir ada **KeluargaController** yang memroses seluruh aktivitas yang berhubungan dengan keluarga (/keluarga)
- **DAO:** yaitu Data Access Object (*mapper*) yang menghubungkan *database* dengan *service*. DAO ini yang melakukan *query* ke *database* kemudian memberikan hasilnya kepada *service*. Pada *package* ini saya memiliki lima *mapper* yaitu **PendudukMapper** yang berfungsi untuk mengakses data yang berkaitan dengan penduduk, kemudian ada **KotaMapper** yang berfungsi untuk mengakses data yang berkaitan dengan kota, kemudian ada **KelurahanMapper** yang berfungsi untuk mengakses data yang berkaitan dengan kelurahan, kemudian ada **KeluargaMapper** yang berfungsi untuk mengakses data yang berkaitan dengan keluarga, dan terakhir ada **KecamatanMapper** yang berfungsi untuk mengakses data yang berkaitan dengan kecamatan.
- **Model:** merupakan suatu *class* yang merepresentasikan *entity* dari *database*. Pada masing-masing Model memiliki atribut yang dimiliki oleh *entity* tersebut. Pada *package* ini saya memiliki lima Model yaitu **KecamatanModel**, **KeluargaModel**, **KelurahanModel**, **KotaModel**, **PendudukModel**.
- **Service:** berfungsi sebagai pemanggil fungsi yang telah dibuat dalam DAO serta implementasinya. Terdapat *interface* dan *class* yang diimplementasi dalam *package* ini. Pada *package* ini saya memiliki lima Service *interface* dan *class* yang bersesuaian dengan Model.

### 4. FITUR LAIN-LAIN

#### 4a. Mengganti *Text Field* dengan *Dropdown*

##### a. Konsep

Mengganti seluruh *input* yang memiliki *value* yang sudah pasti (seperti Kelurahan, Kecamatan, Kota, Golongan Darah, Jenis Kelamin, dll) dengan *dropdown*

##### b. Manfaat

Agar pengguna tidak dipusingkan dengan *input* yang diharuskan sama dengan *value* yang seharusnya dan juga agar tidak perlu banyak meng-*handle* kemungkinan kesalahan yang akan di-*input*-kan oleh pengguna

##### c. Cara Mengakses Fitur

Bisa diakses melalui fitur Tambah Penduduk, Tambah Keluarga, Update Penduduk, Update Keluarga

## 5. OPTIMASI DATABASE

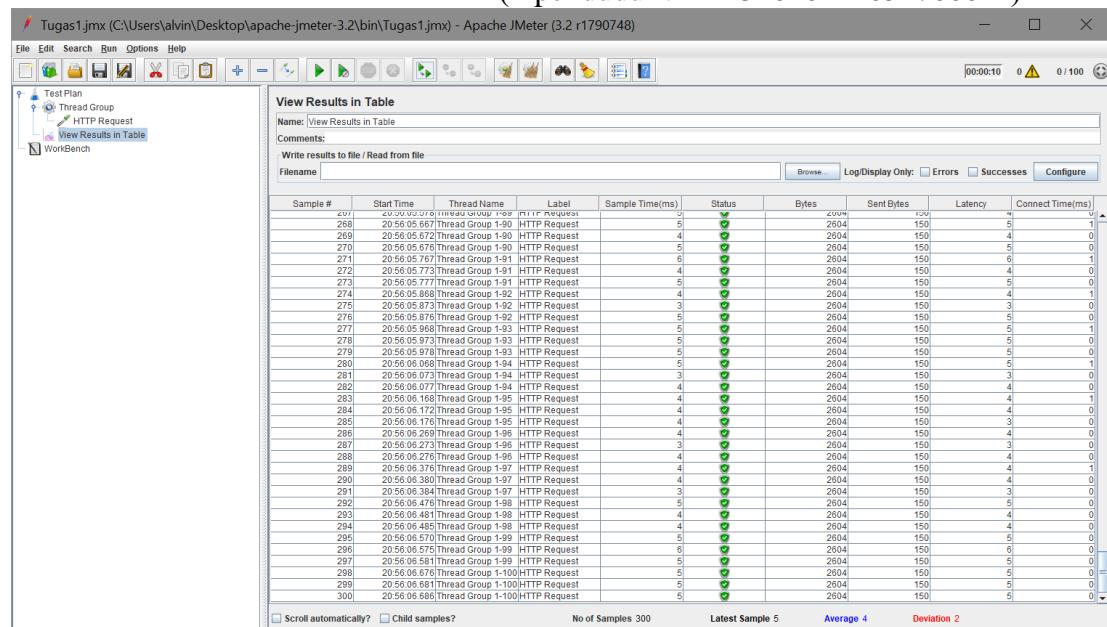
Saya melakukan beberapa optimasi berikut pada *database*:

- **PENDUDUK**: id (PRIMARY), nik (INDEX), id\_keluarga (INDEX)
- **KELUARGA**: id (PRIMARY), nomor\_kk (INDEX), id\_kelurahan (INDEX)
- **KELURAHAN**: id (PRIMARY), id\_kecamatan (INDEX)
- **KECAMATAN**: id (PRIMARY), id\_kota (INDEX)
- **KOTA**: id (PRIMARY), kode\_kota (INDEX)

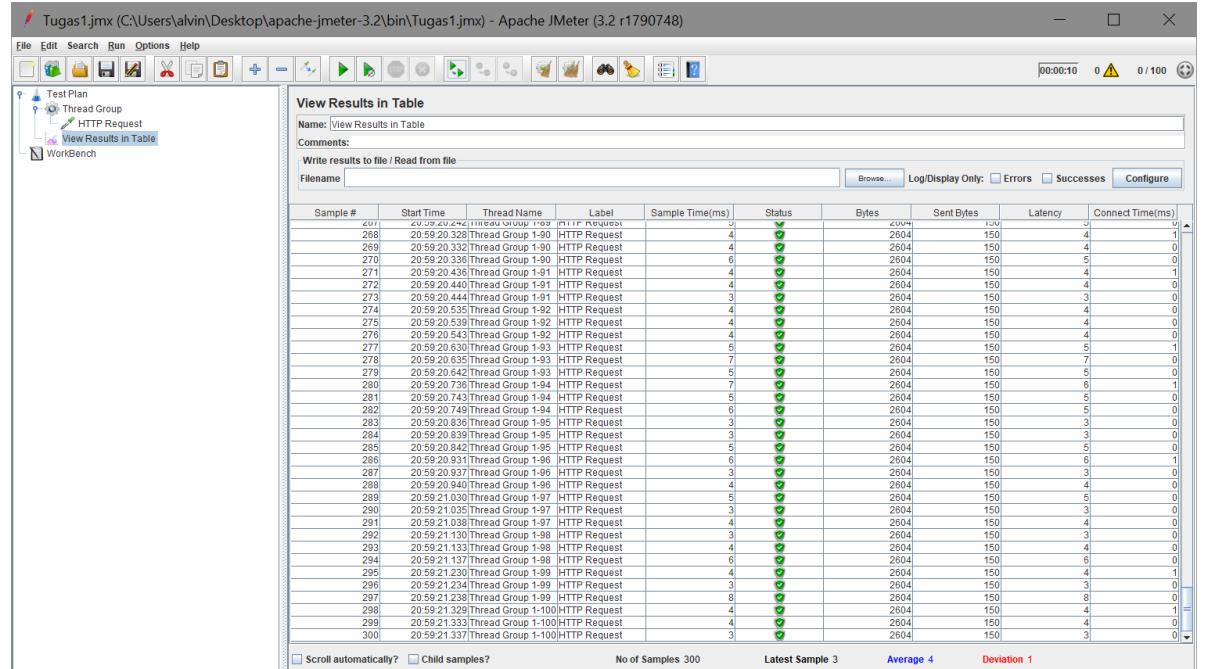
## 6. STRESS TESTING

Saya melakukan stress testing pada masing-masing fitur dengan **Number of Threads** berjumlah **100** dan **Ramp-up Period** selama **10 detik**

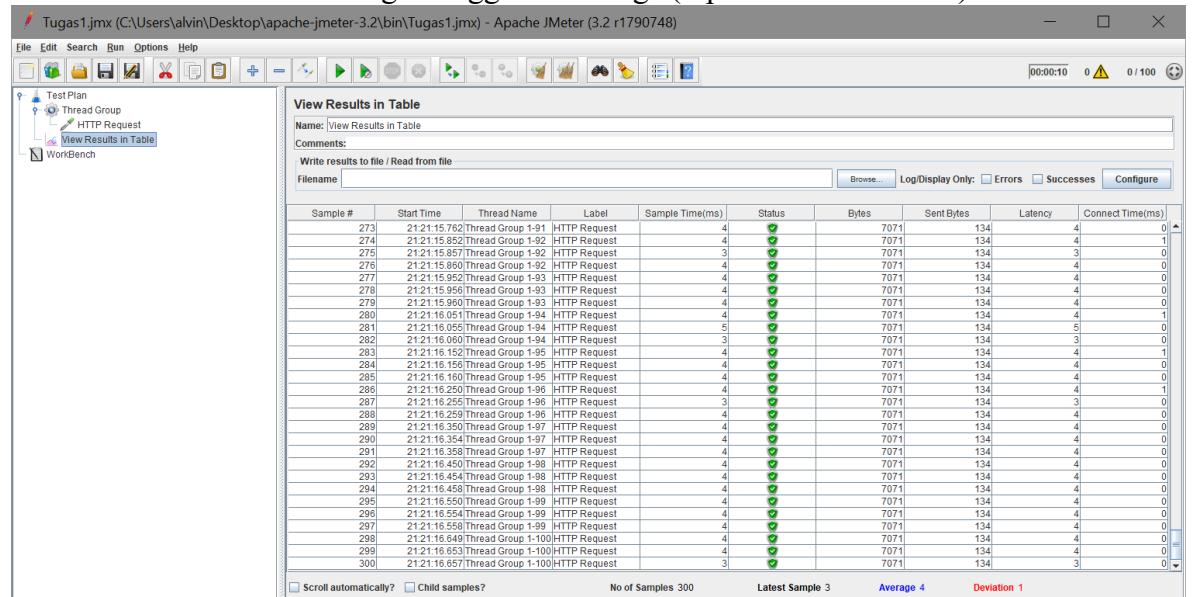
- Tampilkan Data Penduduk Berdasarkan NIK (“/penduduk?nik= 3101011405170001”)



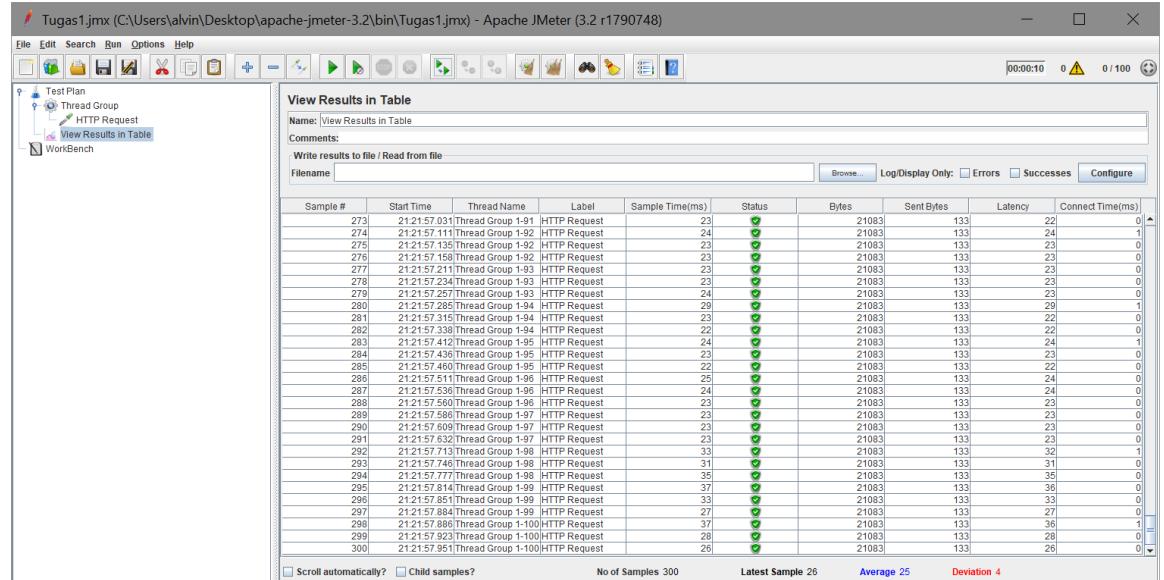
- Tampilkan Data Keluarga Beserta Daftar Anggota Berdasarkan Nomor KK (“/keluarga?nkk= 3101010101050001”)



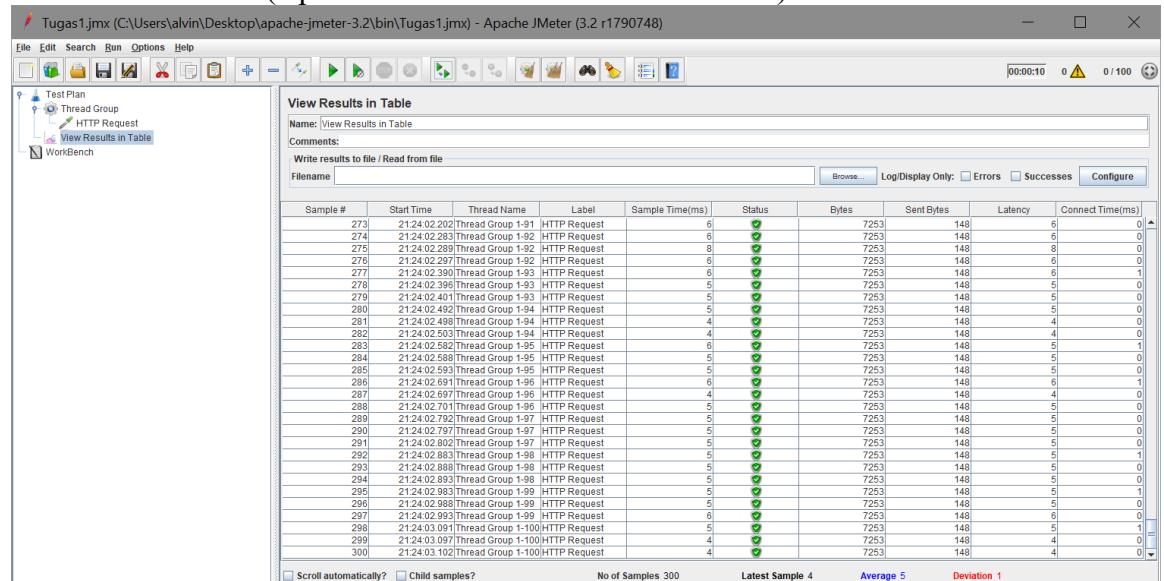
- Menambahkan Penduduk Baru Sebagai Anggota Keluarga (“/penduduk/tambah”)



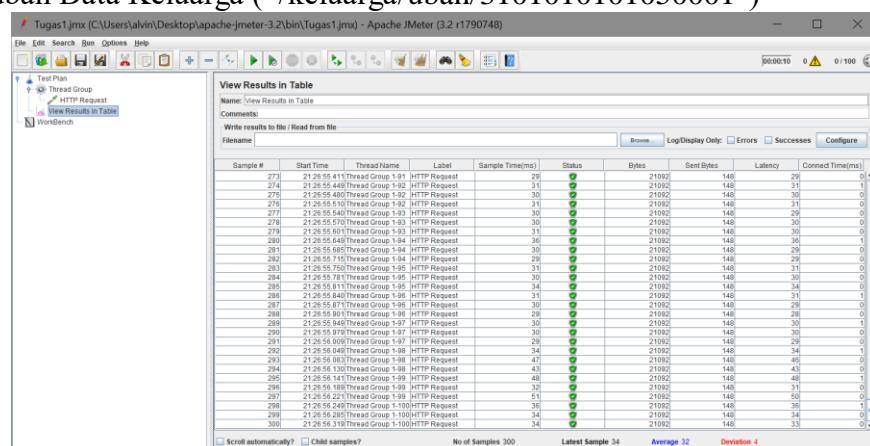
- Menambahkan Keluarga Baru (“/keluarga/tambah”)



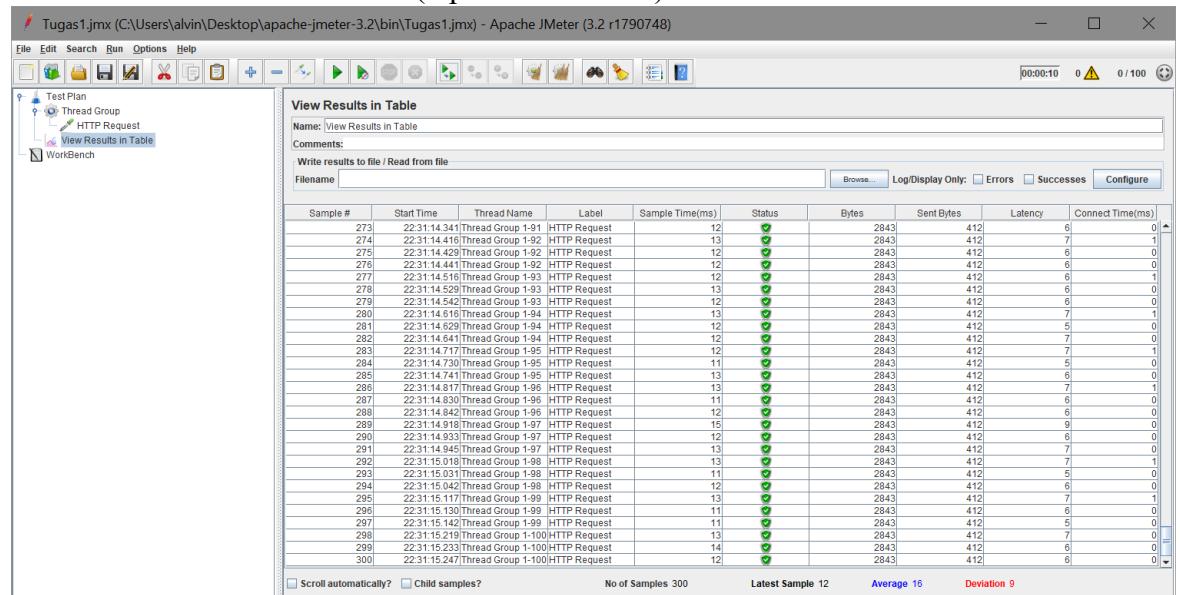
- Mengubah Data Penduduk (“/penduduk/ubah/3101011405170001”)



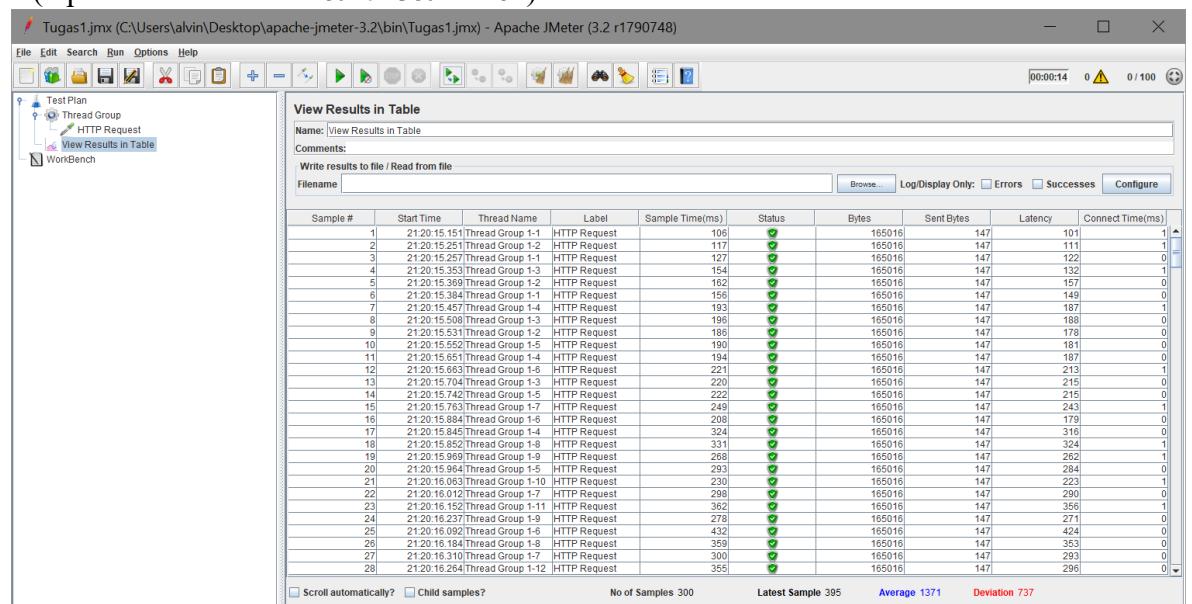
- Mengubah Data Keluarga (“/keluarga/ubah/3101010101050001”)



- Mengubah Status Kematian Penduduk (“/penduduk/mati”)



- Tampilkan Data Penduduk Berdasarkan Kota/Kabupaten, Kecamatan, dan Kelurahan Tertentu (“/penduduk/cari?kt=2&kc=3&kl=10”)



Berdasarkan pengamatan dengan jMeter, respon dari masing-masing halaman tergolong cepat dan *reliable* karena tidak memiliki waktu tunggu yang lama. Waktu cepat tersebut juga dipengaruhi oleh *database* yang udah dioptimasi.

## 7. PROSES MENGEMBANGKAN TUGAS

Saya mengerjakan tugas ini dengan berdiskusi bersama teman dan juga dengan melakukan berbagai macam pencarian di Google untuk mencari solusi dari permasalahan yang saya dapatkan.

Fitur-fitur dari Tugas 1:

The screenshot shows a web browser window titled "Home Page" with the URL "localhost:8080". The page header includes "SI Kependudukan", "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". The main content area is titled "Sistem Informasi Kependudukan Online Provinsi DKI Jakarta" and displays a welcome message: "Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta". Below this, there are two sections: "Lihat Data Penduduk Berdasarkan NIK" with a form field "Nomor NIK" and a button "Lihat", and "Lihat Data Keluarga Berdasarkan NKK" with a form field "Nomor NKK" and a button "Lihat".

### Halaman Utama

#### 1. Tampilkan Data Penduduk Berdasarkan NIK

The screenshot shows the same web browser window as before, but now displaying search results. The "Nomor NIK" field contains "3101010101690001" and the "Lihat" button is highlighted in green. The page content below shows the results for this search, which are currently not visible in the screenshot.

Lihat Data Penduduk - 3101010101690001

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

Jika tidak ada,

Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

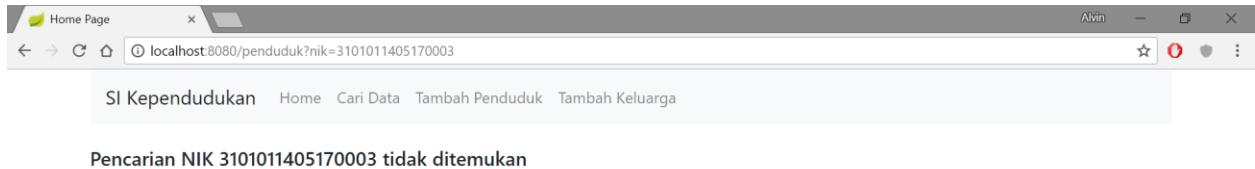
Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta

Lihat Data Penduduk Berdasarkan NIK

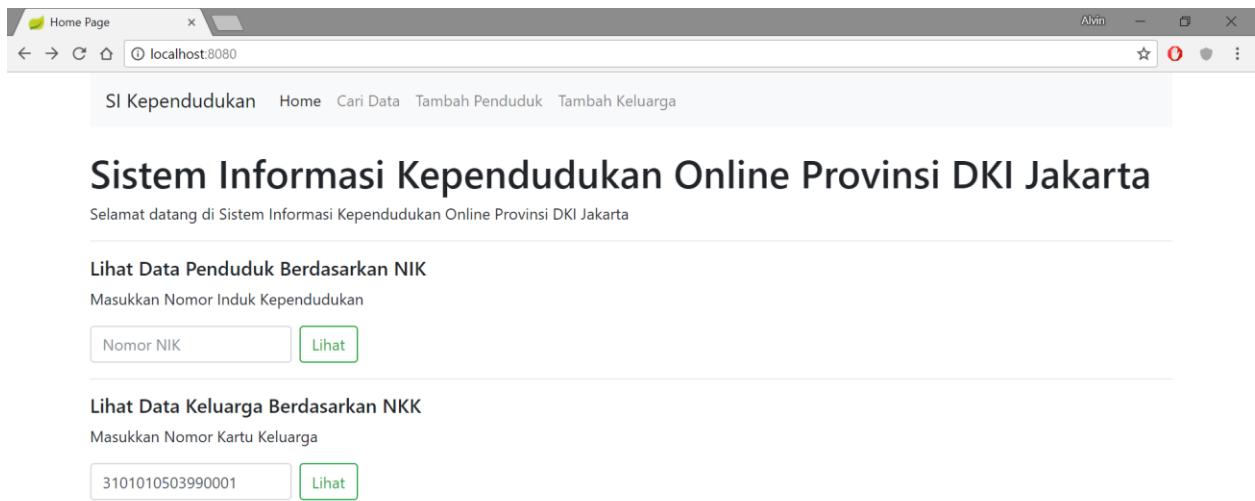
Masukkan Nomor Induk Kependudukan

Lihat Data Keluarga Berdasarkan NKK

Masukkan Nomor Kartu Keluarga



## 2. Tampilkan Data Keluarga Beserta Daftar Anggotanya Berdasarkan Nomor KK



Lihat Data Keluarga - 3101010503990001

| No | Nama Lengkap                | NIK              | Jenis Kelamin | Tempat Lahir | Agama | Pekerjaan             | Status Perkawinan | Status Dalam Keluarga | Kewarganegaraan |
|----|-----------------------------|------------------|---------------|--------------|-------|-----------------------|-------------------|-----------------------|-----------------|
| 1  | Hartaka Lukita Santoso M.Pd | 3101011603950001 | Laki-laki     | Banjarbaru   | Islam | PENTERJEMAH           | Kawin             | Kepala Keluarga       | WNI             |
| 2  | Uli Kuswandari              | 3101016606810001 | Perempuan     | Jakarta      | Islam | PEMBANTU RUMAH TANGGA | Cerai Mati        | Pembantu              | WNI             |

Jika tidak ditemukan,

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga

## 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  
 Nomor NIK

**Lihat Data Keluarga Berdasarkan NKK**  
Masukkan Nomor Kartu Keluarga  
 123

The screenshot shows a web browser window titled "Home Page". The address bar displays "localhost:8080". The page header includes the title "SI Kependudukan" and navigation links for "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". Below the header, a large heading reads "Sistem Informasi Kependudukan Online Provinsi DKI Jakarta". A sub-header says "Selamat datang di Sistem Informasi Kependudukan Online Provinsi DKI Jakarta". There are two search forms: one for "Lihat Data Penduduk Berdasarkan NIK" (NIK number input: "Nomor NIK" with value "123", "Lihat" button) and one for "Lihat Data Keluarga Berdasarkan NKK" (KK number input: "Masukkan Nomor Kartu Keluarga" with value "123", "Lihat" button).

The screenshot shows a web browser window titled "Home Page". The address bar displays "localhost:8080/keluarga?nkk=123". The page header includes the title "SI Kependudukan" and navigation links. Below the header, a message states "Pencarian NKK 123 tidak ditemukan" (Search for NKK 123 not found).

### 3. Menambahkan Penduduk Baru Sebagai Anggota Keluarga

Tambah Penduduk

Nama

Tempat Lahir

Tanggal Lahir  
 dd----yyyy

Jenis Kelamin  
 Laki-laki

Status Dalam Keluarga  
 Anak

Golongan Darah

Tambah Penduduk

Nama  
 Zoro

Tempat Lahir  
 Jakarta

Tanggal Lahir  
 25-Dec-1997

Jenis Kelamin  
 Laki-laki

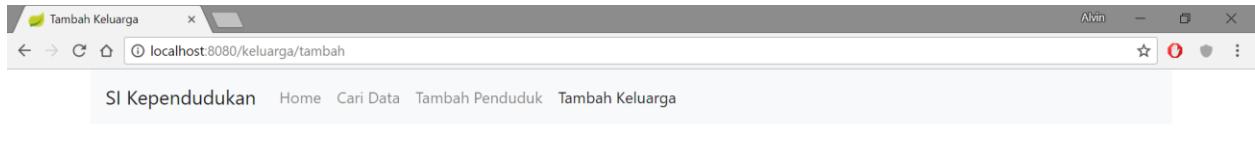
Status Dalam Keluarga  
 Anak

Golongan Darah



#### 4. Menambahkan Keluarga Baru

A screenshot of a web browser window titled "Tambah Keluarga". The address bar shows "localhost:8080/keluarga/tambah". The page title is "Tambah Keluarga". The form contains fields for "Alamat" (Jalan Akses UI), "RT" (002), "RW" (002), "Kelurahan/Desa" (JAGAKARSA), "Kecamatan" (JAGAKARSA), and "Kota/Kabupaten" (KOTA JAKARTA SELATAN). A blue "Simpan" button is at the bottom.



## 5. Mengubah Data Penduduk

The screenshot shows a web browser window titled "Update Data Penduduk". The address bar displays "localhost:8080/penduduk/ubah/3101010101690001". The page title is "Update Penduduk". The form contains the following fields:

- Masukkan Nama: Ridwan Hasim Marmaung
- Tempat Lahir: Atambua
- Tanggal Lahir: 01-Jan-1969
- Jenis Kelamin: Laki-laki
- Status Dalam Keluarga: Pembantu
- Golongan Darah: A+
- Agama: Agama



## 6. Mengubah Data Keluarga

A screenshot of a web browser window titled "Update Data Keluarga". The address bar shows "localhost:8080/keluarga/ubah/3171012210170002". The page title is "Update Keluarga". The form fields are as follows:

- Alamat: Jalan Akses UI
- RT: 002
- RW: 004
- Kelurahan/Desa: JAGAKARSA
- Kecamatan: JAGAKARSA
- Kota/Kabupaten: KOTA JAKARTA SELATAN

A blue "Simpan" button is located at the bottom left of the form.

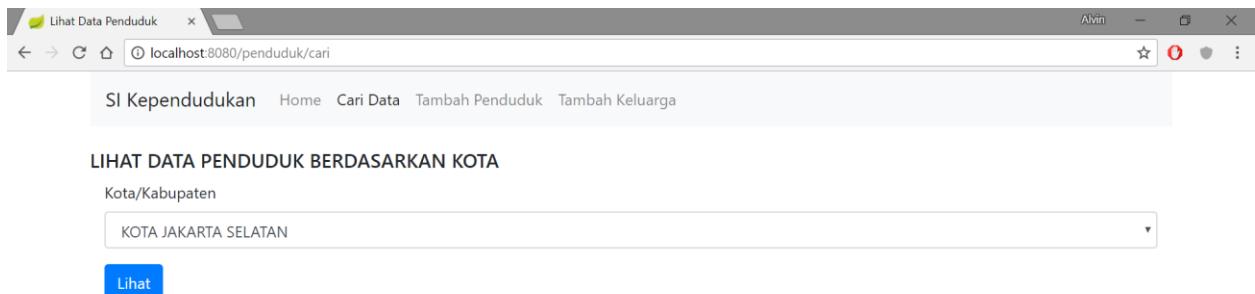


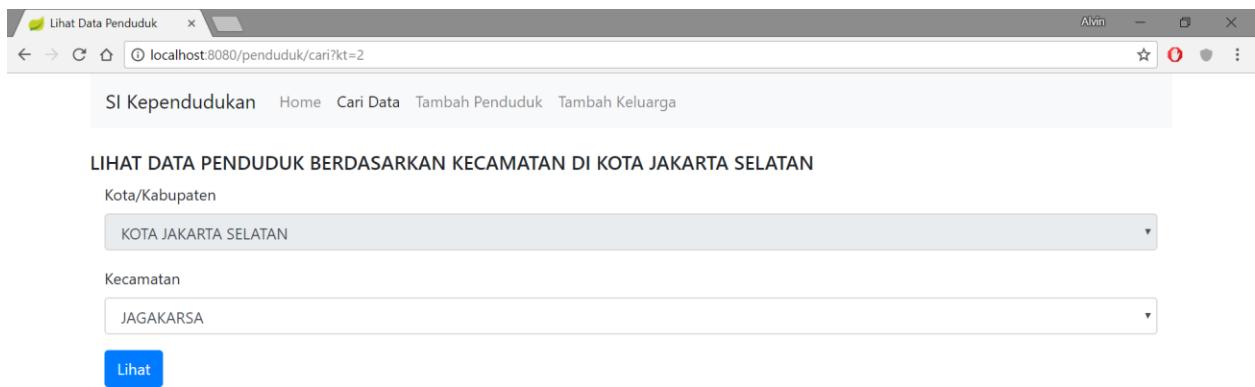
## 7. Mengubah Status Kematian Penduduk

|                             |                            |
|-----------------------------|----------------------------|
| NIK                         | 3101011307980001           |
| Nama                        | Alvin                      |
| Tempat/Tanggal Lahir        | Bogor, 13-07-1998          |
| Alamat                      | Jln. Cikapayang No. 594    |
| RT/RW                       | 149/090                    |
| Kelurahan/Desa              | PULAU PARI                 |
| Kecamatan                   | KEPULAUAN SERIBU SELATAN   |
| Kota                        | KABUPATEN KEPULAUAN SERIBU |
| Golongan Darah              | AB+                        |
| Agama                       | Islam                      |
| Status Perkawinan           | Belum Kawin                |
| Pekerjaan                   | Mahasiswa                  |
| Kewarganegaraan             | WNI                        |
| Status Kematian             | Hidup                      |
| <a href="#">Nonaktifkan</a> |                            |



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





Lihat Data Penduduk

localhost:8080/penduduk/cari?kt=2

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga

LIHAT DATA PENDUDUK BERDASARKAN KECAMATAN DI KOTA JAKARTA SELATAN

Kota/Kabupaten

KOTA JAKARTA SELATAN

Kecamatan

JAGAKARSA

Lihat

This screenshot shows a dropdown menu for selecting a district in Jakarta Selatan. The dropdown is open, displaying 'KOTA JAKARTA SELATAN'. Above the dropdown, the title 'LIHAT DATA PENDUDUK BERDASARKAN KECAMATAN DI KOTA JAKARTA SELATAN' is visible, along with navigation links for SI Kependudukan, Home, Cari Data, Tambah Penduduk, and Tambah Keluarga.



Lihat Data Penduduk

localhost:8080/penduduk/cari?kt=2&kc=3

SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga

LIHAT DATA PENDUDUK BERDASARKAN KELURAHAN DI KOTA JAKARTA SELATAN, KECAMATAN JAGAKARSA

Kota/Kabupaten

KOTA JAKARTA SELATAN

Kecamatan

JAGAKARSA

Kelurahan

JAGAKARSA

Lihat

This screenshot shows a dropdown menu for selecting a village in Jagakarsa, Jakarta Selatan. The dropdown is open, displaying 'JAGAKARSA'. Above the dropdown, the title 'LIHAT DATA PENDUDUK BERDASARKAN KELURAHAN DI KOTA JAKARTA SELATAN, KECAMATAN JAGAKARSA' is visible, along with navigation links for SI Kependudukan, Home, Cari Data, Tambah Penduduk, and Tambah Keluarga.

The screenshot shows a web browser window titled "Lihat Data Penduduk". The URL is "localhost:8080/penduduk/cari?kt=2&kc=3&kl=10". The page header includes "SI Kependudukan", "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". Below this is a section titled "LIHAT DATA PENDUDUK DI KOTA JAKARTA SELATAN, KECAMATAN JAGAKARSA, KELURAHAN JAGAKARSA". It displays two boxes: "Penduduk Dengan Usia Termuda" and "Penduduk Dengan Usia Tertua". The "Penduduk Dengan Usia Termuda" box contains three entries:

|               |                     |
|---------------|---------------------|
| NIK           | 3171015209170002    |
| Nama          | Shania Wahyuni S.Pd |
| Tanggal Lahir | 12-09-2017          |

The "Penduduk Dengan Usia Tertua" box also contains three entries:

|               |                           |
|---------------|---------------------------|
| NIK           | 3171016012520001          |
| Nama          | Carla Wani Yolanda M.Farm |
| Tanggal Lahir | 20-12-1952                |

Below these boxes is a search bar with "Show 10 entries" and a "Search:" input field. The main content area is a table with columns "No", "NIK", "Nama", and "Jenis Kelamin". The data is as follows:

| No | NIK              | Nama                     | Jenis Kelamin |
|----|------------------|--------------------------|---------------|
| 1  | 3171014405810001 | Puji Rahimah S.H.        | Perempuan     |
| 2  | 3171016111090001 | Kiandra Winarsih         | Perempuan     |
| 3  | 3171012005890001 | Galak Samsul Wijaya M.M. | Laki-laki     |
| 4  | 3171010704890001 | Ikin Manqunsong          | Laki-laki     |

## 8. BONUS

- Dapat menampilkan penduduk tertua dan termuda pada suatu kelurahan

The screenshot shows a web browser window titled "Lihat Data Penduduk". The URL is "localhost:8080/penduduk/cari?kt=1&kc=1&kl=2". The page header includes "SI Kependudukan", "Home", "Cari Data", "Tambah Penduduk", and "Tambah Keluarga". Below this is a section titled "LIHAT DATA PENDUDUK DI KABUPATEN KEPULAUAN SERIBU, KECAMATAN KEPULAUAN SERIBU SELATAN, KELURAHAN PULAU PARI". It displays two boxes: "Penduduk Dengan Usia Termuda" and "Penduduk Dengan Usia Tertua". The "Penduduk Dengan Usia Termuda" box contains three entries:

|               |                              |
|---------------|------------------------------|
| NIK           | 3101012009170002             |
| Nama          | Carub Hairyanto Kusumo M.Ti. |
| Tanggal Lahir | 20-09-2017                   |

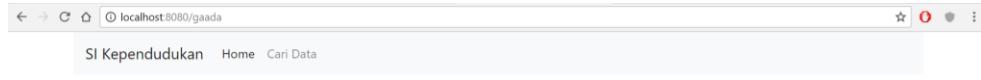
The "Penduduk Dengan Usia Tertua" box also contains three entries:

|               |                         |
|---------------|-------------------------|
| NIK           | 3101010611520001        |
| Nama          | Cahyanto Cemani Gunawan |
| Tanggal Lahir | 06-11-1952              |

Below these boxes is a search bar with "Show 10 entries" and a "Search:" input field. The main content area is a table with columns "No", "NIK", "Nama", and "Jenis Kelamin". The data is as follows:

| No | NIK              | Nama                    | Jenis Kelamin |
|----|------------------|-------------------------|---------------|
| 1  | 3101010303890001 | Taswir Rajata           | Laki-laki     |
| 2  | 3101011408150001 | Mulyono Hutagalung S.Gz | Laki-laki     |
| 3  | 3101016312160001 | Yuliana Wahyuni         | Perempuan     |

- Saya menambahkan *error handling* pada beberapa kasus *error* yang mungkin terjadi pada tugas ini.



Error 404 - Halaman Tidak Ditemukan



SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga

Pencarian NIK 123 tidak ditemukan



SI Kependudukan Home Cari Data Tambah Penduduk Tambah Keluarga

Pencarian NKK 123 tidak ditemukan