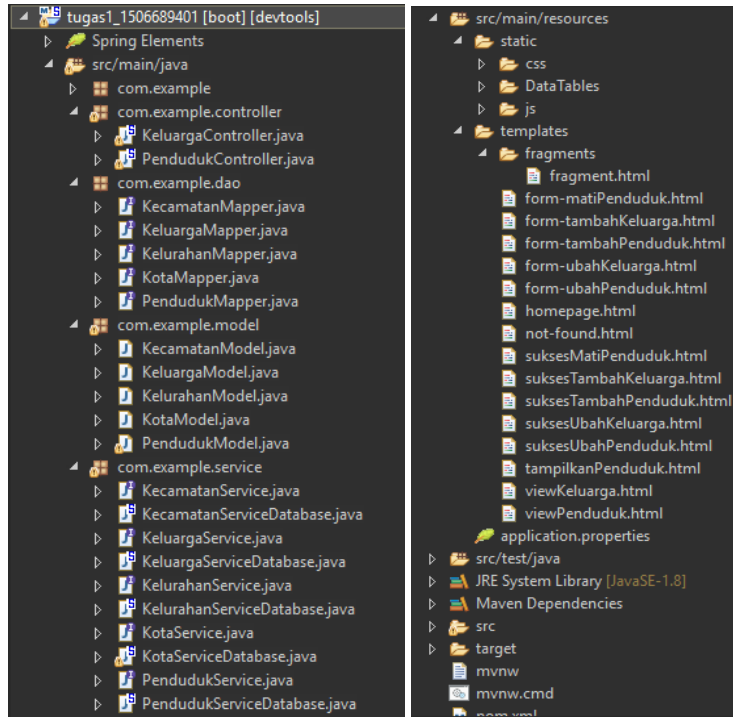


Write Up Tugas 1

1. STRUKTUR PROJECT



2. PACKAGE

Pada tugas 1 ini Saya memiliki 5 package, yaitu :

- **com.example**
 - Tugas11506689401Application.java

- **com.example.controller**
 - KeluargaController.java
 - @RequestMapping("/keluarga")

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

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

        //Buat object dari KelurahanModel agar bisa ditarik di view
        KelurahanModel kelurahanMdl = keluargaDAO.selectKelurahan(keluarga.getId_kelurahan());
        model.addAttribute("kelurahan", kelurahanMdl);

        //Buat object dari KecamatanModel agar bisa ditarik di view
        KecamatanModel kecamatanMdl = kecamatanDAO.selectKecamatan(kelurahanMdl.getId_kecamatan());
        model.addAttribute("kecamatan", kecamatanMdl);

        //Buat object dari KotaModel agar bisa ditarik di view
        KotaModel kotaMdl = kotaDAO.selectKota(kecamatanMdl.getId_kota());
        model.addAttribute("kota", kotaMdl);

        //List anggotaKeluarga yang isinya anggota keluarga dari sebuah keluarga, di ambil dari penduduk model
        List<PendudukModel> anggotaKeluarga = keluargaDAO.selectAnggotaKeluarga(keluarga.getId());
        model.addAttribute("anggota", anggotaKeluarga);

        return "viewKeluarga";
    }
    else {
        //Simpan JUDUL di notifikasi dan DETAIL di detail
        model.addAttribute("notifikasi", "NKK not found");
        model.addAttribute("detail", "NKK = " + nomor_kk + " tidak ditemukan");
        return "not-found";
    }
}
```

- @RequestMapping("/keluarga/tambah") (RequestMethod.GET)

```
@RequestMapping(value="/keluarga/tambah", method=RequestMethod.GET)
public String formTambahKeluarga() {
    return "form-tambahKeluarga";
}
```

- @RequestMapping("/keluarga/tambah") (RequestMethod.POST)

```
@RequestMapping(value="/keluarga/tambah", method=RequestMethod.POST)
public String tambahKeluarga(Model model,
    @RequestParam(value = "alamat", required=false) String alamat,
    @RequestParam(value = "rt", required=false) String rt,
    @RequestParam(value = "rw", required=false) String rw,
    @RequestParam(value = "kelurahan", required=false) String kelurahan,
    @RequestParam(value = "kecamatan", required=false) String kecamatan,
    @RequestParam(value = "kota", required=false) String kota
)
{
    KelurahanModel kelurahanMdl = keluargaDAO.selectKelurahan(kelurahan);
    //model.addAttribute("kelurahan", kelurahanMdl);
    KecamatanModel kecamatanMdl = kecamatanDAO.selectKecamatan(kelurahanMdl.getId_kecamatan());
    //model.addAttribute("kecamatan", kecamatanMdl);

    //masukkan kode kecamatan ke string agar bisa di substring
    String kodeKecamatan = kecamatanMdl.getKode_kecamatan();

    //ambil 6 kode awal dari kode kecamatan, ya 0 terakhir harus karena pasti akhirnya 0
    kodeKecamatan = kodeKecamatan.substring(0, 6);

    //ambil tanggal yang keacakan dengan hari joi, gunakan format yang sesuai, lalu simpan ke
    Date today = new Date();
    SimpleDateFormat tanggalFormatBenar = new SimpleDateFormat("ddMMyy");
    String tanggalHariIni = tanggalFormatBenar.format(today);

    //Buat nkk yang sesuai dengan ketentuan di wikipedia (6 digit kode kecamatan + 6 digit hari)
    String nkk = kodeKecamatan;
    nkk = nkk + tanggalHariIni;

    //panggil method yang sudah dibuat untuk mengecek kesamaan NIK
    int cekKesamaanNKK = keluargaDAO.cekNKKYangSama(nkk+"%");

    //By Default, NKK akan diaset akhirnya 0001, jika ada ya sama baru ditambahin 1
    cekKesamaanNKK = cekKesamaanNKK + 1;
    String kesamaanNKK = cekKesamaanNKK + "";

    //lakukan looping untuk mengecek seberapa banyak nkk yang sama
    int i=1;
    while(i > kesamaanNKK.length()) {
        kesamaanNKK = "0" + kesamaanNKK;
    }
    nkk = nkk + kesamaanNKK;

    int noFinalId = keluargaDAO.hitungId() + 1;
    String finalId = noFinalId + "";

    KeluargaModel keluarga = new KeluargaModel(finalId,nkk,alamat,rt,rw,kelurahan,0,null);
    keluargaDAO.tambahKeluarga(keluarga);
    model.addAttribute("keluarga", keluarga);

    return "suksesTambahKeluarga";
}
```

- @RequestMapping("/keluarga/ubah/{nomor_kk}")
(RequestMethod.GET)

```
@RequestMapping(value="/keluarga/ubah/{nomor_kk}", method=RequestMethod.GET)
public String formUbahKeluarga(Model model, @PathVariable(value="nomor_kk") String nomor_kk) {
    KeluargaModel keluarga = keluargaDAO.selectNKKKeluarga(nomor_kk);

    if(keluarga != null) {
        model.addAttribute("keluarga", keluarga);
        return "form-ubahKeluarga";
    }
    else {
        model.addAttribute("nomor_kk", nomor_kk);
        model.addAttribute("notifikasiKeluarga", "Keluarga not found");
        model.addAttribute("detailKeluarga", "Keluarga dengan NKK " + nomor_kk + " tidak ditemukan");
        return "not-found";
    }
}
```

- @RequestMapping("/keluarga/ubah") (RequestMethod.POST)

```
@RequestMapping(value="/keluarga/ubah", method=RequestMethod.POST)
public String ubahKeluarga(Model model, KeluargaModel keluarga,
    @RequestParam(value="old_id_kelurahan", required=false) String old_id_kelurahan,
    @RequestParam(value="id", required=false) String id)
{
    //Kalo id kelurahan yg lama sama dgn id kelurahan yg baru
    if(!keluarga.getId_kelurahan().equalsIgnoreCase(old_id_kelurahan)) {
        List<PendudukModel> anggotaKeluarga = keluargaDAO.selectAnggotaKeluarga(id);

        /*
        * lakukan updated NKK seperti menambah NKK baru yg ada di method tambahKeluarga
        * ==tambahKeluarga di KeluargaController
        */
        KelurahanModel kelurahanMdl = kelurahanDAO.selectKelurahan(keluarga.getId_kelurahan());
        //model.addAttribute("kelurahan", kelurahanMdl);
        KecamatanModel kecamatanMdl = kecamatanDAO.selectKecamatan(keluurahanMdl.getId_kecamatan());
        //model.addAttribute("kecamatan", kecamatanMdl);

        //masukkan kode kecamatan ke string agar bisa di substring
        String kodeKecamatan = kecamatanMdl.getKode_kecamatan();

        //ambil 6 kode awal dari kode kecamatan. yg 0 terakhir diubah karena pasti akhirnya 0
        kodeKecamatan = kodeKecamatan.substring(0, 6);

        //ambil tanggal yang bersesuaian dengan hari ini. gunakan format yang sesuai. lalu simpan ke
        Date today = new Date();
        SimpleDateFormat tanggalFormatBenar = new SimpleDateFormat("ddMyy");
        String tanggalIni = tanggalFormatBenar.format(today);

        //Buat nkk yang sesuai dengan ketentuan di wikipedia (6 digit kode kecamatan + 6 digit hari
        String updatedNkk = kodeKecamatan + tanggalIni;
        updatedNkk = updatedNkk + tanggalIni;

        //panggil method yang sudah dibuat untuk mengecek kesamaan NIK
        int cekKesamaanNKK = keluargaDAO.cekNKKYangSama(updatedNkk+"%");

        //panggil method yang sudah dibuat untuk mengecek kesamaan NIK
        int cekKesamaanNKK = keluargaDAO.cekNKKYangSama(updatedNkk+"%");

        //By Default, NKK akan diset akhirnya 0001, jika ada yg sama baru ditambahin 1
        cekKesamaanNKK = cekKesamaanNKK + 1;
        String kesamaanNKK = cekKesamaanNKK + "";

        //lakukan looping untuk mengecek seberapa banyak nkk yang sama
        int i=4;
        while(i > kesamaanNKK.length()) {
            kesamaanNKK = "0" + kesamaanNKK;
        }
        updatedNkk = updatedNkk + kesamaanNKK;

        //Untuk mengambil nomor KK yang lama
        model.addAttribute("oldies_nomor_kk", keluarga.getNomor_kk());

        keluarga.setNomor_kk(updatedNkk);

        keluargaDAO.ubahKeluarga(keluarga);

        /* Updated NKK sesuai anak */

        //Looping utk update NIK penduduk baru
        for(int x=0; x<anggotaKeluarga.size();x++) {
            /*
            * lakukan pembuatan NIK baru seperti yang ada di updated NIK Penduduk
            * ada kita looping tiap nama keluarga. karena NIKnya disanti satu per
            * == method ubahPenduduk di PendudukController
            */

            //ambil tanggal lahir
            String tgllahir = anggotaKeluarga.get(x).getTanggal_lahir();

            //pisah tanggalnya biar bisa dipecahkan
            String pisahTanggal[] = tgllahir.split("-");
        }
    }
}
```

```
//ambil tanggal lahir
String tgllahir = anggotaKeluarga.get(x).getTanggal_lahir();

//pisah tanggalnya biar bisa diinputkan
String pisahTanggal[] = tgllahir.split("-");

//cek jenis kelamin. jika jenis kelaminnya wanita maka tanggal lahirnya ditambah 40, sedangkan
//lakukan parsing karena kita data asalnya string
//parse dulu string ke integer lalu parse lagi ke string
if(anggotaKeluarga.get(x).getJenis_kelamin().equals("1")) {
    pisahTanggal[2] = (Integer.parseInt(pisahTanggal[2]) + 40 + "");
}

//Buat nik yang sesuai dengan ketentuan di wikipedia (6 digit kode kecamatan + 2 digit tahun lahir)
String updatedNik = kodeKecamatan + pisahTanggal[2] + pisahTanggal[1] + pisahTanggal[0].substring(2);

//panggil method yang sudah dibuat untuk mengecek kesamaan NIK
int cekKesamaanNIK = pendudukDAO.cekNIKYangSama(updatedNik+"");

//By Default, NIK akan diikut alihannya 0001, jika ada ya sama baru ditambah 1
cekKesamaanNIK = cekKesamaanNIK + 1;
String kesamaanNIK = cekKesamaanNIK + "";

//lakukan looping untuk mengecek apakah benar nik yang sama
for(int j=4; j>kesamaanNIK.length(); j+=0) {
    kesamaanNIK = '0' + kesamaanNIK;
}
updatedNik = updatedNik + kesamaanNIK;
/* * Proses pembuatan NIK seperti menambahkan NIK baru berhenti sampai sini */

//Masukkan nik yang sudah dibuat ulang lalu set menjadi nik yang aktif
anggotaKeluarga.get(x).setNik(updatedNik);

pendudukDAO.ubahPenduduk(anggotaKeluarga.get(x));
//Membuat NIK baru dengan NIK baru sampai sini"

}

return "suksesUbahKeluarga";
}
else {
    keluargaDAO.ubahKeluarga(keluarga);
    model.addAttribute("oldies_nomor_kk", keluarga.getNomor_kk());
    return "suksesUbahKeluarga";
}
```

- PendudukController.java

- @RequestMapping("/")

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

- @RequestMapping("/penduduk")

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

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

        //mencoba mengubah output dari tanggal lahir
        String tanggalahir = penduduk.getTanggal_lahir().toString();
        model.addAttribute("tanggal_lahir", tanggalahir);

        //Buat object dari KeluargaModel agar bisa ditarik di view
        //karena ada model satu keluarga, maka akan sesuai dengan id_keluarga yang ada di database
        //Biar bisa dengan method karena di database ini
        KeluargaModel keluargaMdl = keluargaDAO.selectKeluarga(penduduk.getId_keluarga());
        model.addAttribute("keluarga", keluargaMdl);

        //Buat object dari KelurahanModel agar bisa ditarik di view
        KelurahanModel kelurahanMdl = kelurahanDAO.selectKelurahan(keluargaMdl.getId_kelurahan());
        model.addAttribute("kelurahan", kelurahanMdl);

        //Buat object dari KecamatanModel agar bisa ditarik di view
        KecamatanModel kecamatanMdl = kecamatanDAO.selectKecamatan(kelurahanMdl.getId_kecamatan());
        model.addAttribute("kecamatan", kecamatanMdl);

        //Buat object dari KelurahanModel agar bisa ditarik di view
        KotaMdl = kotaDAO.selectKota(kecamatanMdl.getId_kota());
        model.addAttribute("kota", kotaMdl);

        return "viewPenduduk";
    }
    else {
        model.addAttribute("notifikasi", "Penduduk not found");
        model.addAttribute("detail", "NIK = " + nik + " tidak ditemukan");
        return "not-found";
    }
}
```

- @RequestMapping("/penduduk/tambah") (RequestMethod.GET)

```
@RequestMapping(value="/penduduk/tambah", method=RequestMethod.GET)
public String formTambahPenduduk() {
    return "form-tambahPenduduk";
}
```

▪ @RequestMapping("/penduduk/tambah") (RequestMethod.POST)

```
@RequestMapping(value="/penduduk/tambah", method=RequestMethod.POST)
public String tambahPenduduk(Model model,
    @RequestParam(value = "nama", required=false) String nama,
    @RequestParam(value = "tempat_lahir", required=false) String tempat_lahir,
    @RequestParam(value = "tanggal_lahir", required=false) String tanggal_lahir,
    @RequestParam(value = "golongan_darah", required=false) String golongan_darah,
    @RequestParam(value = "agama", required=false) String agama,
    @RequestParam(value = "status_perkawinan", required=false) String status_perkawinan,
    @RequestParam(value = "pekerjaan", required=false) String pekerjaan,
    @RequestParam(value = "is_wni", required=false) Integer is_wni,
    @RequestParam(value = "is_wafat", required=false) Integer is_wafat,
    @RequestParam(value = "id_keluarga", required=false) String id_keluarga,
    @RequestParam(value = "jenis_kelamin", required=false) String jenis_kelamin,
    @RequestParam(value = "status_dalam_keluarga", required=false) String status_dalam_keluarga
)
{
    KeluargaModel keluargaMdl = keluargaDAO.selectKeluarga(id_keluarga);
    //model.addAttribute("keluarga", keluargaMdl);
    KelurahanModel kelurahanMdl = kelurahanDAO.selectKelurahan(keluargaMdl.getId_kelurahan());
    //model.addAttribute("kelurahan", kelurahanMdl);
    KecamatanModel kecamatanMdl = kecamatanDAO.selectKecamatan(kelurahanMdl.getId_kelurahan());
    //model.addAttribute("kecamatan", kecamatanMdl);

    //Masukkan kode kecamatan ke string agar bisa di substring
    String kodeKecamatan = kecamatanMdl.getKode_kecamatan();

    //ambil 6 kode awal dari kode kecamatan, yg @ terakhir diubah karena abstrak pasti @
    kodeKecamatan = kodeKecamatan.substring(0, 6);

    //ambil tanggal lahir
    String tgllahir = tanggal_lahir;
    //System.out.println(tgllahir); //cek di console keluarnya gimana : tahun-bulan-tanggal

    //pisah tanggalnya biar bisa dipecah
    String[] pisahTanggal = tgllahir.split("-");

    if(jenis_kelamin.equals("M")) {
        pisahTanggal[2] = (Integer.parseInt(pisahTanggal[2]) + 48 + "");
    }

    //Buat nik yang sesuai dengan ketentuan di atasnya (6 digit kode kecamatan + 2 digit tgl lahir + 2 digit bulan lahir + 2 digit
    String nik = kodeKecamatan;
    nik = nik + pisahTanggal[1] + pisahTanggal[2] + pisahTanggal[0].substring(2);

    //panggil method yang sudah dibuat untuk mengecek kesamaan nik
    int cekKesamaanNIK = pendudukDAO.cekNIKYangSama(nik+"");

    //By Default, NIK akan dibuat abstrak 0001, jika ada 22 maka akan ditambah 1
    cekKesamaanNIK = cekKesamaanNIK + 1;
    String kesamaanNIK = cekKesamaanNIK + "";

    //lakukan looping untuk mengecek seberapa banyak nik yang sama
    for(int i=1; i<kesamaanNIK.length(); i++) {
        kesamaanNIK = "0" + kesamaanNIK;
    }
    nik = nik + kesamaanNIK;

    //Masuk lagi dari looping diatas
    /* jik i=1
    while(i<kesamaanNIK.length()) {
        kesamaanNIK="0" + kesamaanNIK;
    }*/

    int finalId = pendudukDAO.hitungId() + 1;

    PendudukModel penduduk = new PendudukModel(finalId, nik, nama, tempat_lahir, tanggal_lahir, jenis_kelamin, is_wni, id_keluarga, agama,
    pekerjaan, status_perkawinan, status_dalam_keluarga);
    pendudukDAO.tambahPenduduk(penduduk);
    model.addAttribute("penduduk", penduduk);

    return "suksesTambahPenduduk";
}
```

▪ @RequestMapping("/penduduk/ubah/{nik}") (RequestMethod.GET)

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

    if (penduduk != null) {
        model.addAttribute("penduduk", penduduk);
        return "form-ubahPenduduk";
    }
    else {
        model.addAttribute("nik", nik);
        model.addAttribute("notifikasiPenduduk", "Penduduk not found");
        model.addAttribute("detailPenduduk", "Penduduk dengan NIK " + nik + " tidak ditemukan");

        return "not-found";
    }
}
```

- @RequestMapping("/penduduk/ubah") (RequestMethod.POST)

```
@RequestMapping(value="/penduduk/ubah", method = RequestMethod.POST)
public String ubahPenduduk(Model model,
    @RequestParam(value="nik", required = false) String nik, @Valid @ModelAttribute PendudukModel updatedPenduduk)
{
    //Buat dua buah penduduk, satu utk ambil yang lama, satu yg baru untuk diupdate
    PendudukModel oldPenduduk = pendudukDAO.selectPenduduk(nik);

    //cek jika atribut yang dapat berubah nik diganti (id keluarga, tgl lahir, jenis kelamin) ada yang berbeda, maka buat nik dari baru
    if(oldPenduduk.getId_keluarga() != updatedPenduduk.getId_keluarga() || oldPenduduk.getTanggal_lahir() != updatedPenduduk.getTanggal_lahir() || oldPenduduk.getJenis_kelamin() != updatedPenduduk.getJenis_kelamin()) {

        /*
        * Karena penduduk lama diupdate, data baru, lakukan proses pembuatan nik penduduk dari awal lagi
        */
        KeluargaModel keluargaMdl = keluargaDAO.selectKeluarga(updatedPenduduk.getId_keluarga());
        KelurahanModel kelurahanMdl = kelurahanDAO.selectKelurahan(keluargaMdl.getId_kelurahan());
        KecamatanModel kecamatanMdl = kecamatanDAO.selectKecamatan(kelurahanMdl.getId_kecamatan());

        //masukkan kode kecamatan ke string agar jika di substring
        String kodekecamatan = kecamatanMdl.getKode_kecamatan();

        //ambil 6 kode awal dari kode kecamatan, yg 0 terakhir kosong, karena akhirnya pasti 0
        kodekecamatan = kodekecamatan.substring(0, 6);
        System.out.println(kodekecamatan);

        //ambil tanggal lahir
        String tglLahir = updatedPenduduk.getTanggal_lahir();

        //pisah tanggalnya biar bisa dipisahin
        String pisahTanggal[] = tglLahir.split("-");

        //cek jenis kelamin, jika jenis kelaminnya wanita maka tanggal lahirnya ditambah 40, sedangkan pria tidak diubah
        //lakukan parsing karena kita data awalnya string
        //parse dari string ke integer lalu parse lagi ke string
        if(updatedPenduduk.getJenis_kelamin().equals("1")) {
            pisahTanggal[2] = (Integer.parseInt(pisahTanggal[2]) + 40 + "");
        }

        //parse dari string ke integer lalu parse lagi ke string
        if(updatedPenduduk.getJenis_kelamin().equals("1")) {
            pisahTanggal[2] = (Integer.parseInt(pisahTanggal[2]) + 40 + "");
        }

        //Buat nik yang sesuai dengan ketentuan di wikipedia (6 digit kode kecamatan + 2 digit tgl lah
        String updatedNik = kodekecamatan + pisahTanggal[2] + pisahTanggal[1] + pisahTanggal[0].substring(2);

        //lakukan method yang sudah dibuat untuk mengecek kesamaan NIK
        int cekKesamaanNIK = pendudukDAO.cekNIKYangSama(updatedNik+"");

        //By Default, NIK akan diaset akhirnya 0001, jika ada yg sama baru ditambahin 1
        cekKesamaanNIK = cekKesamaanNIK + 1;
        String kesamaanNIK = cekKesamaanNIK + "";

        //lakukan looping untuk mengecek seberapa banyak nik yang sama
        for(int i=1; i<kesamaanNIK.length(); i++) {
            kesamaanNIK = "0" + kesamaanNIK;
        }
        updatedNik = updatedNik + kesamaanNIK;
        /*
        * Proses pembuatan NIK seperti menambahkan NIK baru berhenti sampai sini */

        //Masukkan nik yang sudah dibuat ulang lalu set menjadi nik yang aktif
        updatedPenduduk.setNik(updatedNik);

        pendudukDAO.ubahPenduduk(updatedPenduduk);
        model.addAttribute("nik", nik);
        model.addAttribute("oldPenduduk", oldPenduduk);
        return "suksesUbahPenduduk";
    }
    else {
        pendudukDAO.ubahPenduduk(updatedPenduduk);
        model.addAttribute("nik", nik);
        model.addAttribute("updatedPenduduk", updatedPenduduk);
        return "suksesUbahPenduduk";
    }
}
```

- @RequestMapping("/penduduk/mati") (RequestMethod.POST)

```
@RequestMapping(value="/penduduk/mati", method=RequestMethod.POST)
public String suksesMatiPenduduk(Model model, @RequestParam(value="nik") String nik) {
    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
    model.addAttribute("penduduk", penduduk);
    return "form-matiPenduduk";
}
```

- @RequestMapping("/penduduk") (RequestMethod.POST)

```
@RequestMapping(value="/penduduk", method=RequestMethod.POST)
public String formMatiPenduduk(Model model, @RequestParam(value="nik") String nik) {
    PendudukModel penduduk = pendudukDAO.selectPenduduk(nik);
    //Matikan penduduk
    pendudukDAO.matikanPenduduk(nik);

    int hitungAnggota = pendudukDAO.hitungNotWafat(penduduk.getId_keluarga());

    KeluargaModel keluarga = keluargaDAO.selectKeluarga(penduduk.getId_keluarga());

    if(hitungAnggota == 0) {
        keluarga.setIs_tidak_berlaku(1);
    }
    keluargaDAO.keluargaTidakAktif(keluarga.getId());
    model.addAttribute("nik", nik);
    return "suksesMatiPenduduk";
}
```

▪ @RequestMapping("/penduduk/cari") (RequestMethod.GET)

```
@RequestMapping(value="/penduduk/cari", method=RequestMethod.GET)
public String cariPenduduk(Model model,
    @RequestParam Optional<String> kt,
    @RequestParam Optional<String> kc,
    @RequestParam Optional<String> kl)
{
    List<KotaModel> allKota = kotaDAO.selectSemuaKota();
    model.addAttribute("allKota", allKota);
    KecamatanModel kecamatan = null;
    KelurahanModel kelurahan = null;

    if(kt.isPresent()) {
        model.addAttribute("idKotaNow", kt.get());
        List<KecamatanModel> kecamatanByKota = kecamatanDAO.selectKecamatanByKota(kt.get());
        model.addAttribute("kecamatanByKota", kecamatanByKota);

        //Untuk mengambil nama Kota
        String judulKota = kotaDAO.selectKota(kt.get()).getNama_kota();
        model.addAttribute("judulKota", judulKota);
    }

    if(kc.isPresent()) {
        model.addAttribute("idKecamatanNow", kc.get());
        List<KelurahanModel> kelurahanByKecamatan = kelurahanDAO.selectKelurahanByIdKecamatan(kc.get());
        model.addAttribute("kelurahanByKecamatan", kelurahanByKecamatan);

        //Untuk mengambil nama Kecamatan
        kecamatan = kecamatanDAO.selectKecamatan(kc.get());
        String judulKecamatan = kecamatan.getNama_kecamatan();
        model.addAttribute("judulKecamatan", judulKecamatan);
    }

    if(kl.isPresent()) {
        model.addAttribute("idKelurahanNow", kl.get());

        if(kl.isPresent()) {
            model.addAttribute("idKelurahanNow", kl.get());

            //Untuk mengambil nama Kelurahan
            //Ini akannya akan punya list of keluarga di sebuah kelurahan
            //akannya akan punya list of penduduk di sebuah keluarga
            kelurahan = kelurahanDAO.selectKelurahan(kl.get());
            String judulKelurahan = kelurahan.getNama_kelurahan();
            model.addAttribute("kelurahan", kelurahan);

            //Inisiasi penduduk (tapi isi)
            List<PendudukModel> penduduk = new ArrayList<PendudukModel>();
            /*
            * Contoh Object Oriented
            */
            //Buat sebuah ArrayList kosong yang nanti menampung penduduk-penduduk dari keluarga yang ada
            for (KeluargaModel keluarga : kelurahan.getKeluargas()) {
                for (PendudukModel penduduk : keluarga.getPendukuks()) {
                    penduduk.add(penduduk);
                }
            }
            model.addAttribute("pendukuks", penduduk);
        }
        return "tampilkanPenduduk";
    }
}
```

- **com.example.dao**
 - KotaMapper.java (**Interface**)
 - KecamatanMapper.java (**Interface**)
 - KelurahanMapper.java (**Interface**)
 - KeluargaMapper.java (**Interface**)
 - PendudukMapper.java (**Interface**)
- **com.example.model**
 - KotaModel.java
 - KecamatanModel.java
 - KelurahanModel.java
 - KeluargaModel.java
 - Penduduk.java
- **com.example.service**
 - KotaService.java
 - KotaServiceDatabase.java
 - KecamatanService.java
 - KecamatanServiceDatabase.java
 - KelurahanService.java
 - KelurahanServiceDatabase.java

- KeluargaService.java
- KeluargaServiceDatabase.java
- PendudukService.java
- PendudukServiceDatabase.java

**Note : Untuk kelengkapan source code dari package yang lain dapat dilihat melalui kode program*

3. IMPLEMENTASI MVC

Pada program ini Saya mengimplementasikan MVC melalui beberapa package yaitu *example, controller, dao, model, dan service*.

- **Example**

Package ini berfungsi untuk menjalankan seluruh program. Package ini otomatis terbuat saat kita menggunakan project Spring Tools.

- **Controller**

Package ini untuk mengatur dan menjalankan fungsi yang dilakukan. Fungsi yang kemudian dijalankan akan ditampilkan dalam view (berupa *.html*). Pada tugas ini saya membuat dua controller yaitu **KeluargaController** (yang mengatur jalannya fungsi fitur-fitur yang berkaitan dengan Keluarga) dan **PendudukController** (yang mengatur jalannya fitur-fitur yang berkaitan dengan penduduk).

- **DAO**

DAO (*Data Access Object*) digunakan untuk mengakses data dari object. Package ini berkaitan dengan database dan menjalankan query-query yang berkaitan dengan database, seperti select, insert, update, dan delete. Pada package ini saya membuat 5 macam DAO yaitu **KotaMapper**, **KecamatanMapper**, **KelurahanMapper**, **KeluargaMapper**, dan **PendudukMapper**.

- **Model**

Package ini memodelkan attribute/entity yang ada dalam database ke dalam program, pada package ini setiap entity didefinisikan dan sesuai dengan entity yang ada di database. Saya membuat 5 model yaitu **KotaModel**, **KecamatanModel**, **KelurahanModel**, **KeluargaModel**, dan **PendudukModel**.

- **Service**

Fungsi dari service yaitu memanggil fungsi yang telah dibuat dalam DAO/Mapper serta implementasinya. Pada program ini saya membuat 5 interface dan class yang sesuai dengan 5 macam model yang sudah saya sebutkan diatas.

4. FITUR LAIN-LAIN

5. OPTIMASI DATABASE

Pada tugas ini saya melakukan optimasi database pada table yang ada dengan beberapa cara berikut :

#PENDUDUK

Membuat entity *id* menjadi **PRIMARY KEY**, membuat entity *nik* menjadi **UNIQUE**, membuat entity *id_keluarga* menjadi **INDEX**.

#KELUARGA

Membuat entity *id* menjadi **PRIMARY KEY**, membuat entity *nomor_kk* menjadi **INDEX**, membuat entity *id_kelurahan* menjadi **INDEX**.

#KELURAHAN

Membuat entity *id* menjadi **PRIMARY KEY**, membuat entity *id_kecamatan* menjadi **INDEX**.

#KECAMATAN

Membuat entity *id* menjadi **PRIMARY KEY**, membuat entity *id_kota* menjadi **INDEX**.

#KOTA

Membuat entity *id* menjadi **PRIMARY KEY**, membuat entity *kode_kota* menjadi **INDEX**.

6. STRESS TESTING

7. PROSES MENGEMBANGKAN TUGAS

Pada proses pengembangan tugas ini, saya melakukan diskusi dengan teman-teman dan asisten dosen terkait beberapa fitur pada tugas. Berikut ini adalah flow dari program sesuai dengan requirement dari deskripsi soal.

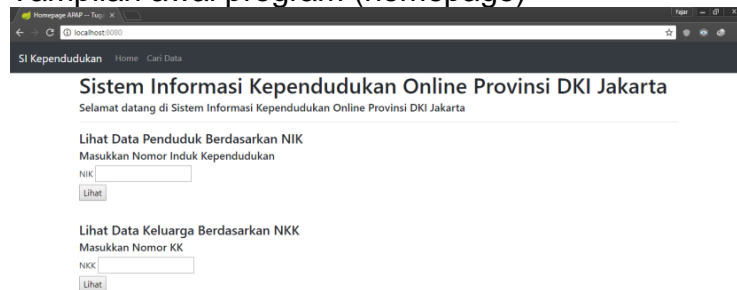
1. Tampilkan Data Penduduk Berdasarkan NIK

Initial Page : /

Form Request : GET, /penduduk

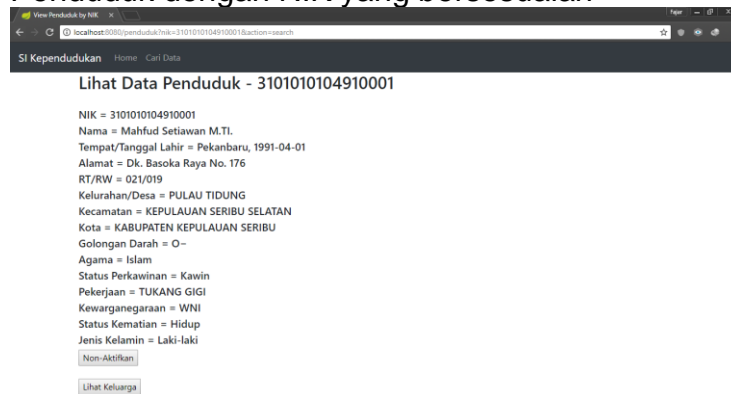
Response Page: / penduduk?nik={NIK}

- Tampilan awal program (homepage)



The screenshot shows the homepage of the 'Sistem Informasi Kependudukan Online Provinsi DKI Jakarta'. The page has a dark header with the title 'SI Kependudukan' and navigation links 'Home' and 'Cari Data'. Below the header, the main content area is white and contains two search sections. The first section is titled 'Lihat Data Penduduk Berdasarkan NIK' and includes a label 'Masukkan Nomor Induk Kependudukan', a text input field for 'NIK', and a 'Lihat' button. The second section is titled 'Lihat Data Keluarga Berdasarkan NKK' and includes a label 'Masukkan Nomor KK', a text input field for 'NKK', and a 'Lihat' button.

- User memasukkan NIK yang didapat dari database untuk melihat Data Penduduk dengan NIK yang bersesuaian



The screenshot shows the 'View Penduduk by NIK' page. The browser address bar shows the URL 'localhost:3000/penduduk/NIK=3101010104910001?action=search'. The page title is 'Lihat Data Penduduk - 3101010104910001'. The main content area displays a list of personal details for the individual with NIK 3101010104910001. The details include: Nama = Mahfud Setiawan M.TI, Tempat/Tanggal Lahir = Pekanbaru, 1991-04-01, Alamat = Dk. Basoka Raya No. 176, RT/RW = 021/019, Kelurahan/Desa = PULAU TIDUNG, Kecamatan = KEPULAUAN SERIBU SELATAN, Kota = KABUPATEN KEPULAUAN SERIBU, Golongan Darah = O-, Agama = Islam, Status Perkawinan = Kawin, Pekerjaan = TUKANG GIGI, Kewarganegaraan = WNI, Status Kematian = Hidup, and Jenis Kelamin = Laki-laki. At the bottom of the details, there are two buttons: 'Non-Aktifkan' and 'Lihat Keluarga'.

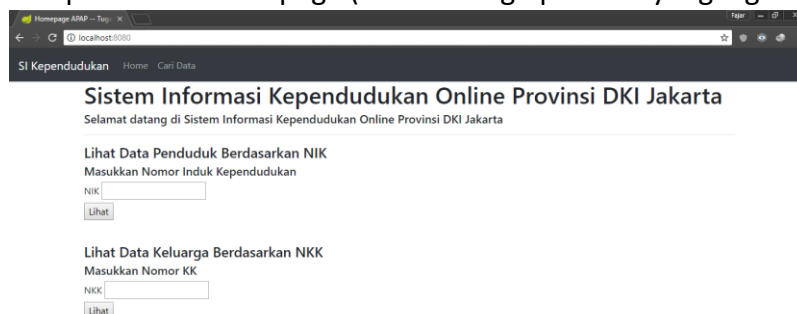
2. Tampilkan Data Keluarga Beserta Daftar Anggotanya Berdasarkan Nomor KK

Initial Page : /

Form Request : GET, /keluarga

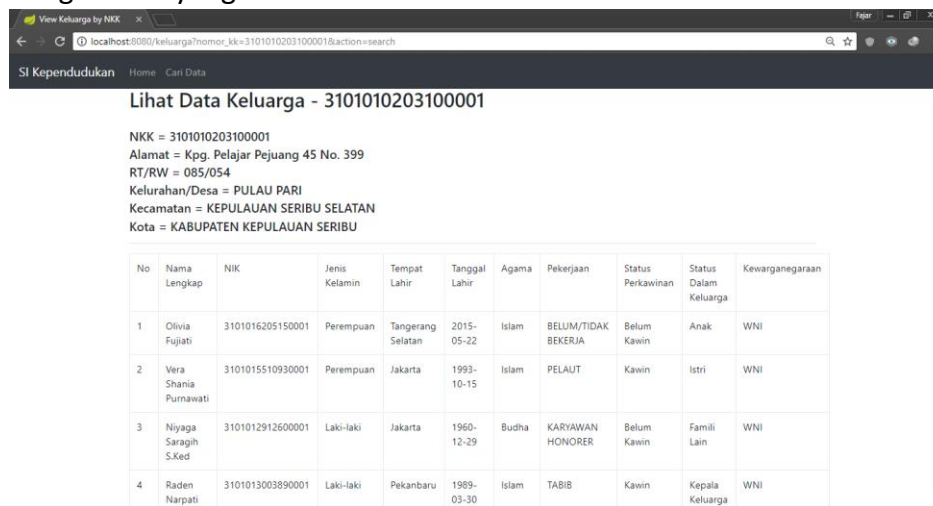
Response Page: /keluarga?nkk={NKK}

- Tampilan awal homepage (untuk menginput NKK yang ingin dilihat data keluarganya)



This screenshot is identical to the one above, showing the homepage of the 'Sistem Informasi Kependudukan Online Provinsi DKI Jakarta'. It features the same dark header, white main content area, and two search sections for NIK and NKK.

- User memasukkan NKK yang didapat dari database untuk melihat Data Keluarga dengan NKK yang bersesuaian



View Keluarga by NKK

localhost:8080/keluarga?nomor_nk=3101010203100001&action=search

SI Kependudukan Home Cari Data

Lihat Data Keluarga - 3101010203100001

NKK = 3101010203100001
Alamat = Kpg. Pelajar Pejuang 45 No. 399
RT/RW = 085/054
Kelurahan/Desa = PULAU PARI
Kecamatan = KEPULAUAN SERIBU SELATAN
Kota = KABUPATEN KEPULAUAN SERIBU

No	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga	Kewarganegaraan
1	Olivia Fujiati	3101016205150001	Perempuan	Tangerang Selatan	2015-05-22	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI
2	Vera Shania Purnawati	3101015510930001	Perempuan	Jakarta	1993-10-15	Islam	PELAUT	Kawin	Istri	WNI
3	Niyaga Saragih S.Ked	3101012912600001	Laki-laki	Jakarta	1960-12-29	Budha	KARYAWAN HONORER	Belum Kawin	Famili Lain	WNI
4	Raden Narpati	3101013003890001	Laki-laki	Pekanbaru	1989-03-30	Islam	TABIB	Kawin	Kepala Keluarga	WNI

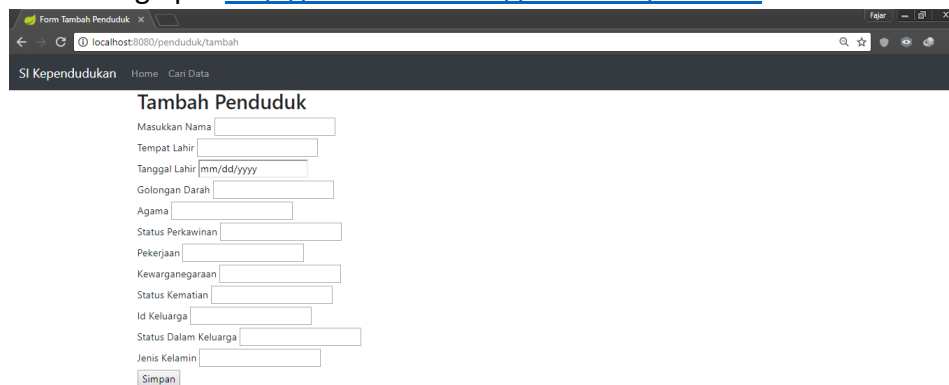
3. Menambahkan Penduduk Baru sebagai Anggota Keluarga

Initial Page : /penduduk/tambah

Form Request : POST, /penduduk/tambah

Response Page: /penduduk/tambah

- User menginput <http://localhost:8080/penduduk/tambah> untuk membuka page



Form Tambah Penduduk

localhost:8080/penduduk/tambah

SI Kependudukan Home Cari Data

Tambah Penduduk

Masukkan Nama

Tempat Lahir

Tanggal Lahir

Golongan Darah

Agama

Status Perkawinan

Pekerjaan

Kewarganegaraan

Status Kematian

Id Keluarga

Status Dalam Keluarga

Jenis Kelamin

- Masukkan data yang bersesuaian dengan yang diminta form (sesuai permintaan di deskripsi tugas)
(note : untuk input WNI, Status Kematian, dan Jenis Kelamin masih menggunakan 0 dan 1)

Form Tambah Penduduk

Masukkan Nama Fajar Bardan Salam

Tempat Lahir Jakarta

Tanggal Lahir 03/24/1997

Golongan Darah A-

Agama Islam

Status Perkawinan Belum Kawin

Pekerjaan Mahasiswa

Kewarganegaraan WNI

Status Kematian Hidup

Id Keluarga 24

Status Dalam Keluarga Anak

Jenis Kelamin Laki-laki

Simpan

- Submit dengan mengklik tombol simpan, operasi akan redirect ke halaman "/penduduk/tambah" dan menampilkan pesan bahwa operasi penambahan penduduk berhasil dilakukan

Add New Penduduk

localhost:8080/penduduk/tambah

SI Kependudukan Home Cari Data

Sukses!

Penduduk dengan NIK 3101012403970001 berhasil ditambahkan

- Kemudian cek melalui database, cari id keluarga yang bersesuaian dan copy nkk dari id keluarga yang bersesuaian untuk mengetahui apakah data penduduk berhasil dimasukkan. Tampilan berikut menunjukkan data berhasil dimasukkan ke database (**penduduk no 5**)

NIK = 3101010203100001

Alamat = Kpg. Pelajar Pejuang 45 No. 399

RT/RW = 085/054

Kelurahan/Desa = PULAU PARI

Kecamatan = KEPULAUAN SERIBU SELATAN

Kota = KABUPATEN KEPULAUAN SERIBU

No	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga	Kewarganegaraan
1	Olivia Fujiati	3101016205150001	Perempuan	Tangerang Selatan	2015-05-22	Islam	BELUM/TIDAK BEKERJA	Belum Kawin	Anak	WNI
2	Vera Shania Purnawati	3101015510930001	Perempuan	Jakarta	1993-10-15	Islam	PELAUT	Kawin	Istri	WNI
3	Niyaga Saragih S.Ked	3101012912600001	Laki-laki	Jakarta	1960-12-29	Budha	KARYAWAN HONORER	Belum Kawin	Famili Lain	WNI
4	Raden Narpati	3101013003990001	Laki-laki	Pekanbaru	1989-03-30	Islam	TABIB	Kawin	Kepala Keluarga	WNI
5	Fajar Bardan Salam	3101012403970001	Laki-laki	Jakarta	1997-03-24	Islam	Mahasiswa	Belum Kawin	Anak	WNI

4. Menambahkan Keluarga Baru

Initial Page : /keluarga/tambah

Form Request : POST, /keluarga/tambah

Response Page: / keluarga/tambah

- Halaman yang muncul ketika kita membuka “/keluarga/tambah”



Form Tambah Keluarga

Alamat

RT

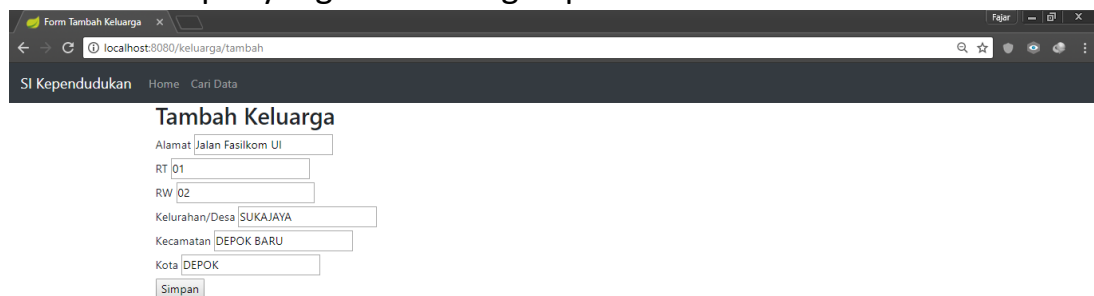
RW

Kelurahan/Desa

Kecamatan

Kota

- Masukkan input yang sesuai dengan permintaan soal



Form Tambah Keluarga

Alamat

RT

RW

Kelurahan/Desa

Kecamatan

Kota

- Submit dengan mengklik tombol simpan, operasi akan redirect ke halaman “/keluarga/tambah” dan menampilkan pesan bahwa operasi penambahan keluarga berhasil dilakukan

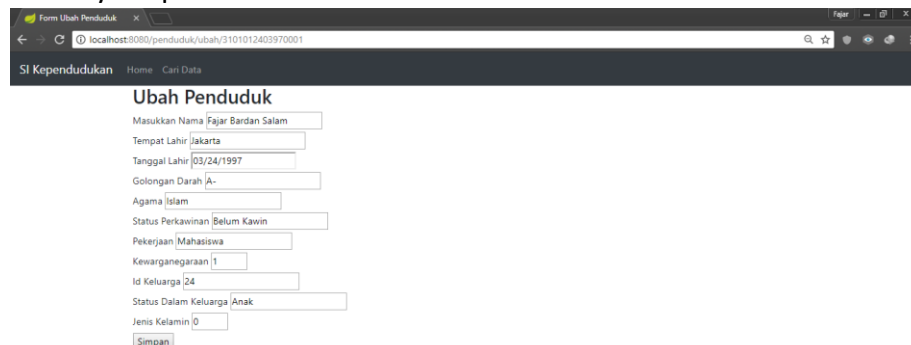
5. Mengubah Data Penduduk

Initial Page : /penduduk/ubah/{NIK}

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

Response Page: /penduduk/ubah/{NIK}

- Buka page dengan perintah “penduduk/ubah/{nik}” dengan nik yang ingin dicari, misalnya seperti berikut



Form Ubah Penduduk

Masukkan Nama

Tempat Lahir

Tanggal Lahir

Golongan Darah

Agama

Status Perkawinan

Pekerjaan

Kewarganegaraan

Id Keluarga

Status Dalam Keluarga

Jenis Kelamin

- Setelah diklik maka page akan langsung menampilkan data yang sebelumnya sudah ada, kemudian ganti data yang ingin diubah, misalnya mengganti id keluarga (yang akan membuat NIK berubah karena aturan NIK, yaitu ada beberapa data yang mempengaruhi NIK sehingga NIK juga berubah) (**ubah id keluarga jadi 25, dimana NIK sebelumnya : 3101012403970001**). Kemudian akan mengeluarkan tampilan berikut :



Sukses!

Penduduk dengan NIK 3101012403970001 berhasil diubah

- Cek di database apakah NIK berubah, jika dicari dengan NIK yang lama maka akan menampilkan NIK yang lama tidak ada



Penduduk not found

Penduduk dengan NIK 3101012403970001 tidak ditemukan

- Setelah dicari dengan id keluarga 25 (didatabase) kemudian mencari keluarga dengan NKK yang bersesuaian dengan id keluarga 25, maka dapat dilihat perubahan nik berikut



Lihat Data Keluarga - 3101010204020001

NKK = 3101010204020001

Alamat = Jln. Abang No. 391

RT/RW = 013/069

Kelurahan/Desa = PULAU PARI

Kecamatan = KEPULAUAN SERIBU SELATAN

Kota = KABUPATEN KEPULAUAN SERIBU

No	Nama Lengkap	NIK	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Pekerjaan	Status Perkawinan	Status Dalam Keluarga	Kewarganegaraan
1	Lega Natsir	3101010810940001	Laki-laki	Denpasar	1994-10-08	Budha	SENIMAN	Kawin	Kepala Keluarga	WNI
2	Rini Safitri	3101015105830001	Perempuan	Jakarta	1983-05-11	Budha	PEDAGANG	Kawin	Istri	WNI
3	Fajar Bardan Salam	3101012403970002	Laki-laki	Jakarta	1997-03-24	Islam	Mahasiswa	Belum Kawin	Anak	WNI

6. Mengubah Data Keluarga

Initial Page : /keluarga/ubah/{NKK}

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

Response Page : /keluarga/ubah/{NKK}

- Ubah data keluarga yang bersangkutan, kemudian akan ditampilkan NKK sukses diubah (ingat aturan bahwa ada data yang dapat menyebabkan NKK berubah)

Ubah Keluarga

Alamat Jln. Abang No. 391

RT 013

RW 069

Kelurahan/Desa

Kecamatan

Kota

Simpan

Ubah Id keluarga jadi 4 (misalnya), maka akan menampilkan

Ubah Keluarga

Alamat Jln. Abang No. 391

RT 013

RW 069

Kelurahan/Desa

Kecamatan

Kota

Simpan

Sukses!

Keluarga dengan NKK 3101010204020001 berhasil diubah

- Jika NKK yang lama dicari maka akan menampilkan not found karena NKK berubah

Penduduk NOT found

Keluarga not found

Keluarga dengan NKK 3101010204020001 tidak ditemukan

7. Mengubah Status Kematian Penduduk

Initial Page: /penduduk?nik={NIK}

Form Request : POST, /penduduk/mati

Response Page: /penduduk?nik={NIK}

- Cari data dari penduduk yang ingin dinonaktifkan. Ubah status kematian penduduk dengan mengklik tombol non aktifkan.



- Nonaktifkan penduduk kemudian akan menampilkan bahwa penduduk tersebut telah mati (page akan redirect ke halaman pertama, pada proses ini ada jeda waktu untuk menunggu hasil nonaktif keluar)



- Penduduk berhasil dinonaktifkan (lihat status kematian = mati)

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

Initial Page: /penduduk/cari

Request Method: GET,

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

Response Page:

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

- Buka halaman “/penduduk/cari” kemudian pilih kota yang ingin dicari data penduduknya. Setelah memilih kota maka akan keluar pilihan untuk memilih kecamatan. Setelah memilih kecamatan maka akan keluar pilihan untuk memilih kelurahan.

Fajar Bardan Salam // 1506689401
APAP-A // Tugas 1



- Kemudian program akan menampilkan data penduduk yang ada pada kelurahan yang bersangkutan

The screenshot shows a web browser window with the URL `localhost:8080/penduduk/cari?kt=3&kc=19&kl=111`. The page title is "LIHAT DATA PENDUDUK DI KOTA JAKARTA TIMUR, KECAMATAN DUREN SAWIT KELURAHAN DUREN SAWIT". Below the title, there is a "Show" dropdown menu with "10" selected and a "Search:" input field. Below this is a table with 3 columns: "NIK", "Nama", and "Jenis Kelamin". The table contains 10 rows of data. At the bottom, there is a pagination bar showing "Showing 1 to 10 of 1,264 entries" and a set of buttons for "Previous", "1", "2", "3", "4", "5", "...", "127", and "Next".

NIK	Nama	Jenis Kelamin
3172070101820001	Nasab Wasita	Laki-laki
3172070102150001	Cakrabirawa Setiawan	Laki-laki
3172070102150002	Kamidin Saragih	Laki-laki
3172070103160001	Cager Habibi	Laki-laki
3172070103700001	Endra Teddy Uwais	Laki-laki
3172070104170001	Kawaca Napitupulu	Laki-laki
3172070104650001	Ganda Rajasa	Laki-laki
3172070105150001	Anggabaya Wahyudin	Laki-laki
3172070105810001	Emil Pandu Sihotang	Laki-laki
3172070105960001	Opan Rajata	Laki-laki