

Tugas 1 ini sangat menantang dibandingkan dengan tutorial-tutorial sebelumnya. Hal yang pertama kali saya lakukan adalah membaca soal dengan baik, dan mulai memperkirakan apa saja yang akan dibutuhkan pada tugas 1. Saya memulai membuat model yang akan digunakan nanti, hal ini berdasarkan dengan relasi yang ada pada *database*. Ada lima model yaitu Penduduk, Keluarga, Kecamatan, Kelurahan, dan Kota. Masing-masing model saya sesuaikan dengan isi tabelnya masing-masing. Saya mengerjakan masing-masing fitur sesuai urutan dari 1 - 8.

1. Fitur 1

Untuk mengerjakan fitur 1, saya membuat PendudukController terlebih dahulu. Pada controller dibuat suatu method yang akan melakukan *request mapping* ke `/` dan `/penduduk?nik=`. Kemudian saya membuat *interface* PendudukService berisi method untuk memilih penduduk berdasarkan nik.

Method ini akan diterapkan oleh *class* PendudukServiceDatabase yang mengimplementasikan PendudukService.

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

Kemudian saya membuat PendudukMapper, di sana akan dilakukan pemanggilan *query* yang hasilnya berupa pendudukModel.

```
@Select("select nik, nama, tempat_lahir, tanggal_lahir, alamat, rt, rw, nama_kelurahan, nama_kecamatan, nama_kota, "
+ "golongan_darah, agama, status_perkawinan, pekerjaan, is_wni, is_wafat, id_keluarga, jenis_kelamin, "
+ "status_dalam_keluarga"
+ " from penduduk join keluarga k" + " on id_keluarga=k.id join kelurahan kl"
+ " on id_kelurahan = kl.id join kecamatan kc" + " on id_kecamatan = kc.id join kota ko"
+ " on id_kota = ko.id where nik=#{nik}")
PendudukModel selectPenduduk(@Param("nik") String nik);
```

Dikarenakan query view membutuhkan atribut tambahan seperti 'nama_kecamatan' dll, saya kemudian menambahkan atribut tersebut pada PendudukModel. Hasil tersebut akan dilempar kembali ke ServiceDatabase lalu ke Controller yang kemudian akan dicetak pada halaman `"view-penduduk.html"`. Setelah memastikan sudah bisa menampilkan detail penduduk, saya baru membuat halaman `"view.html"` yang akan muncul ketika *mapping* `/`.

2. Fitur 2

Untuk fitur 2, kurang lebih memiliki alur yang sama. Karena mapping yang berbeda, maka dibuat satu *controller* baru yaitu KeluargaController. Kemudian saya membuat KeluargaService yang akan diimplementasikan oleh *class* KeluargaServiceDatabase. Methodnya adalah sebagai berikut.

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

Lalu akan dihubungkan dengan KeluargaMapper, yang berisi

```

) @Select("select nomor_kk, alamat, rt, rw, nama_kelurahan, nama_kecamatan, nama_kota, k.id as id_kel"
+ " from keluarga k join kelurahan kl" + " on id_kelurahan = kl.id join kecamatan kc"
+ " on id_kecamatan = kc.id join kota ko" + " on id_kota = ko.id where 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"),
@Result(property = "pendukuks", column = "id_kel", javaType = List.class, many = @Many(select = "selectPendukuksByIdFam")) }
KeluargaModel selectKeluarga(@Param("nkk") String nkk);

) @Select("select nama, nik, jenis_kelamin, tempat_lahir, tanggal_lahir, agama, pekerjaan, status_perkawinan, status_dalam_keluarga, "
+ "is_wni, is_wafat from penduduk where id_keluarga=#{id_keluarga}")
List<PendudukModel> selectPendukuksByIdFam(@Param("id_keluarga") String id_keluarga);

```

Karena keluarga memiliki anggota yang banyak, maka diperlukan *query* yang berbeda. *Query* pertama untuk mendapatkan detail keluarga, *query* selanjutnya untuk mendapatkan list dari anggota keluarga tersebut. List ini nantinya akan disajikan dalam bentuk tabel pada “view-keluarga.html”.

3. Fitur 3

Pertama-tama saya membuat halaman “tambah-penduduk.html”. Halaman ini berisi form yang akan diisi jika ingin menambah penduduk. Saya memulainya dengan menambah method pada PendudukService dan PendudukServiceDatabase. Dengan method sebagai berikut

```

@Override
public boolean tambahPenduduk(PendudukModel penduduk) {
    if (penduduk.getNama() != null && penduduk.getNama() != null && penduduk.getTempat_lahir()
        != null && penduduk.getTanggal_lahir() != null && penduduk.getJenis_kelamin() != null && penduduk.getGolongan_darah()
        != null && penduduk.getAgama() != null && penduduk.getStatus_perkawinan() != null && penduduk.getPekerjaan()
        != null && penduduk.getIs_wni() != null && penduduk.getIs_wafat() != null && penduduk.getId_keluarga()
        != null && penduduk.getStatus_dalam_keluarga() != null) {
        String nik = this.validasiNik(penduduk);
        penduduk.setNik(nik);
        pendudukMapper.tambahPenduduk(penduduk);
        return true;
    } else {
        return false;
    }
}

```

Method ini bertipe boolean, akan dilakukan validasi terlebih dahulu apakah semua form sudah terisi apa belum. Jika memang sudah, akan menghitung nik dan memanggil mapper.

Query untuk menambah penduduk

```

@Insert("INSERT INTO penduduk (nik, nama, tempat_lahir, tanggal_lahir, jenis_kelamin, is_wni, id_keluarga, agama, pekerjaan,
status_perkawinan, status_dalam_keluarga, golongan_darah, is_wafat) VALUES ({nik}, #{nama}, #{tempat_lahir}, #{tanggal_lahir},
#{jenis_kelamin}, #{is_wni}, #{id_keluarga}, #{agama}, #{pekerjaan}, #{status_perkawinan}, #{status_dalam_keluarga},
#{golongan_darah}, #{is_wafat})")
void tambahPenduduk(PendudukModel penduduk);

```

Method validasiNIK

```

public String validasiNik(PendudukModel penduduk) {
    String idkeluarga = penduduk.getId_keluarga();
    KecamatanModel kec = daerahMapper.selectKode(idkeluarga);
    String nik = kec.getKode_kecamatan().substring(0, 6);
    String[] tgl = penduduk.getTanggal_lahir().split("-");
    tgl[0] = tgl[0].substring(2, 4);
    String tglfix = tgl[2] + tgl[1] + tgl[0];
    int kode = 1;

    if (penduduk.getJenis_kelamin().equals("1")) {
        int tgltmp = Integer.parseInt(tgl[2]) + 40;
        tglfix = tgltmp + tgl[1] + tgl[0];
    }

    String cek = nik + tglfix + "%";
    int add = 0;
    List<PendudukModel> a = pendudukMapper.selectNIK(cek);
    String tes = "" + a.size();

    log.info(tes);

    for (int i = 0; i < a.size(); i++) {
        String last = a.get(i).getNik().substring(12, 16);
        add = Integer.parseInt(last);
    }

    kode = kode + add;
    String akhir = "";
    if (kode < 10) {
        akhir = "000" + kode;
    } else if (kode >= 10 && kode <= 99) {
        akhir = "00" + kode;
    } else if (kode > 99 && kode <= 999) {
        akhir = "0" + kode;
    } else {
        akhir = "" + kode;
    }

    nik = nik + tglfix + akhir;

    return nik;
}

```

Setelah itu, saya membuat 2 method pada controller. Pertama yang biasa saja, dan yang kedua yang menerima request metode post. Setelah berhasil, maka controller yang sudah handle request POST akan melakukan redirect ke halaman sukses, jika ternyata tadi di servicedatabase mengambailkan false maka akan kembali ke halaman “tambah-penduduk” dan mengirimkan membawa pesan error.

4. Fitur 4

Kurang lebih sama seperti alur fitur 3 namun method ini belum diimplementasikan pengecekan apakah yang disubmit null atau tidak. Perbedaannya terdapat pada saat membuat nomor_kk. Pada keluarga, nkk membutuhkan tanggal pembuatan maka dari itu dibuat date dan dateformat.

```

@Override
public void tambahKeluarga(KeluargaModel keluarga) {
    DateFormat dateFormat = new SimpleDateFormat("ddMMyy");
    Date date = new Date();
    String datenow = dateFormat.format(date);

    keluarga.setIs_tidak_berlaku("0");
    String nkk = this.validasiNkk(keluarga, datenow);
    keluarga.setNomor_kk(nkk);

    keluargaMapper.tambahKeluarga(keluarga);
}

```

Kemudian dipassing ke method validasiNKK yang berisi

```
public String validasiNkk(KeluargaModel keluarga, String dateNow) {

    // String namaKel = "lcase(' " + keluarga.getNama_kelurahan() + " '");
    KelurahanModel kelurahan = keluargaMapper.getIdKel(keluarga.getNama_kelurahan());
    KecamatanModel kecamatan = keluargaMapper.getIdKec(keluarga.getNama_kecamatan());
    // KotaModel kota = keluargaMapper.getIdKota(keluarga.getNama_kota());

    String idKel = "" + kelurahan.getId();

    keluarga.setId_kelurahan(idKel);

    String nkk = kecamatan.getKode_kecamatan().substring(0, 6);
    log.info(nkk);
    int kode = 1;

    String cek = nkk + dateNow + "%";
    int add = 0;
    List<KeluargaModel> a = keluargaMapper.selectNKK(cek);
    String tes = a.size() + "";

    log.info(tes);

    for (int i = 0; i < a.size(); i++) {
        String last = a.get(i).getNomor_kk().substring(12, 16);
        add = Integer.parseInt(last);
    }

    kode = kode + add;
    String akhir = "";
    if (kode < 10) {
        akhir = "000" + kode;
    } else if (kode >= 10 && kode <= 99) {
        akhir = "00" + kode;
    } else if (kode > 99 && kode <= 999) {
        akhir = "0" + kode;
    } else {
        akhir = "" + kode;
    }

    nkk = nkk + dateNow + akhir;
    return nkk;
}
```

Setelah mendapat NKK yang sesuai maka akan dipanggil mapper untuk insert ke keluarga.

```
@Insert("insert into keluarga (nomor_kk, alamat, rt, rw, id_kelurahan, is_tidak_berlaku) values ({nomor_kk}, #{alamat}, #{rt},  
#{rw}, #{id_kelurahan}, #{is_tidak_berlaku})")  
void tambahKeluarga(KeluargaModel keluarga);
```

Kemudian saya membuat 2 controller. Jika berhasil menambahkan keluarga, maka akan menampilkan halaman sukses.

5. Fitur 5

Fitur ini untuk mengubah data penduduk. Pada PendudukServiceDatabase metod ubahPenduduk, akan diperiksa terlebih dahulu apakah ada yang null. Jika tidak ada, kemudian akan dilakukan pengecekan apakah ide keluarga/tanggal lahir/jenis kelamin terdapat perubahan. Jika ada, maka harus dilakukan validasi NIK terlebih dahulu. Methodnya sama seperti screenshot di atas.

```

@Override
public boolean ubahPenduduk(PendudukModel penduduk) {
    if (penduduk.getNama() != null && penduduk.getNama() != null && penduduk.getTempat_lahir()
        != null && penduduk.getTanggal_lahir() != null && penduduk.getJenis_kelamin() != null && penduduk.getGolongan_darah()
        != null && penduduk.getAgama() != null && penduduk.getStatus_perkawinan() != null && penduduk.getPekerjaan()
        != null && penduduk.getIs_wni() != null && penduduk.getIs_wafat() != null && penduduk.getId_keluarga()
        != null && penduduk.getStatus_dalam_keluarga() != null) {
        String nikLama = penduduk.getNik();
        PendudukModel pendudukLama = pendudukMapper.selectPenduduk(nikLama);

        if (!(pendudukLama.getId_keluarga().equals(penduduk.getId_keluarga()))
            || !(penduduk.getTanggal_lahir().equals(pendudukLama.getTanggal_lahir()))
            || !(penduduk.getJenis_kelamin().equals(pendudukLama.getJenis_kelamin()))) {
            String nik = this.validasiNik(penduduk);
            penduduk.setNik(nik);
        }

        pendudukMapper.ubahPenduduk(penduduk.getNik(), penduduk.getNama(), penduduk.getTempat_lahir(),
            penduduk.getTanggal_lahir(), penduduk.getGolongan_darah(), penduduk.getAgama(),
            penduduk.getStatus_perkawinan(), penduduk.getPekerjaan(), penduduk.getIs_wni(), penduduk.getIs_wafat(),
            penduduk.getId_keluarga(), penduduk.getJenis_kelamin(), penduduk.getStatus_dalam_keluarga(), nikLama);

        return true;
    } else {
        return false;
    }
}

```

Jika sudah, akan dipanggil Penduduk Mapper dengan query berikut.

```

@Update("update penduduk set nik=#{nik}, nama=#{nama}, tempat_lahir=#{tempat_lahir}, tanggal_lahir=#{tanggal_lahir},
golongan_darah=#{golongan_darah}, agama=#{agama}, status_perkawinan=#{status_perkawinan}, pekerjaan=#{pekerjaan},
is_wni=#{is_wni}, is_wafat=#{is_wafat}, id_keluarga=#{id_keluarga}, jenis_kelamin=#{jenis_kelamin}, status_dalam_keluarga =
#{status_dalam_keluarga} where nik=#{nikLama}")
void ubahPenduduk(@Param("nik") String nik, @Param("nama") String nama, @Param("tempat_lahir") String
tempat_lahir,
                    @Param("tanggal_lahir") String tanggal_lahir, @Param("golongan_darah") String golongan_darah,
                    @Param("agama") String agama, @Param("status_perkawinan") String status_perkawinan,
                    @Param("pekerjaan") String pekerjaan, @Param("is_wni") String is_wni, @Param("is_wafat") String
is_wafat,
                    @Param("id_keluarga") String id_keluarga, @Param("jenis_kelamin") String jenis_kelamin,
                    @Param("status_dalam_keluarga") String status_dalam_keluarga, @Param("nikLama") String
nikLama);

```

Jika sukses mengembalikan halaman sukses. , jika ternyata tadi di servicedatabase mengembalikan false maka akan kembali ke halaman “tambah-penduduk” dan mengirimkan membawa pesan error. Jika ternyata tadi di PendudukServiceDatabase mengembalikan false maka akan kembali ke halaman “tambah-penduduk” dan mengirimkan membawa pesan error.

6. Fitur 6

Fitur ini untuk mengubah keluarga. Ketika sudah mengubah dan di-post, pada KeluargaServiceDatabase akan melakukan pengecekan apakah yang diganti merupakan kelurahan dan apakah ada perbedaan tanggal. Jika ada maka akan melakukan validasi NKK dulu. Kemudian mapper akan dipanggil, mappernya adalah sebagai berikut

```

@Update("update keluarga set nomor_kk=#{nomor_kk}, alamat=#{alamat}, rt=#{rt}, rw=#{rw}, id_kelurahan=#{id_kelurahan}
where nomor_kk=#{nkkLama}")
void ubahKeluarga(@Param("nomor_kk") String nomor_kk, @Param("alamat") String alamat, @Param("rt") String rt,
@Param("rw") String rw,
                    @Param("id_kelurahan") String id_kelurahan, @Param("nkkLama") String nkkLama);

```

Kemudian juga sudah sukses akan mengembalikan halaman sukses.

7. Fitur 7

Fitur 7 mengubah status kematian penduduk. Saya memulainya dengan membuat form ygg dihidden pada halaman view-penduduk. Form tersebut disubmit pada penduduk/mati. Kemudian saya membuat service, dengan memanggil mapper untuk mengupdate is_wafat dari penduduk tersebut. Setelah melakukan update, saya menggunakan keluargaMapper untuk mengambil method yang bisa mengembalikan list Penduduk dalam keluarga tersebut. Kemudian dilakukan pengecekan, jika sudah wafat semua, maka akan dilakukan pengubahan is_tidak_berlaku dari keluarga. Setelah semua pengecekan selesai, selanjutnya saya menggunakan ModelAndView untuk redirect ke halaman /penduduk?nik= dan menggunakan RedirectAttributes untuk mengirim flag sebagai tanda kalau sudah melakukan non-aktif. Jika berhasil, maka akan muncul halaman sukses.

8. Fitur 8

Fitur 8 berfungsi untuk menampilkan list penduduk pada kelurahan tertentu. Yang pertama saya lakukan adalah membuat DaerahService, DaerahServiceDatabase, dan DaerahMapper. Mereka berfungsi untuk handle segala macam kegiatan yang berhubungan dengan kota, kelurahan, dan kecamatan. Saya membuat 4 tampilan. Tampilan pertama untuk menampilkan list kota "cari-1", method dibuat untuk mengambil semua list kota. Jika sudah didapatkan, maka akan disubmit, value dari kota yang dipilih akan menjadi paramater pada url. Selanjutnya menampilkan halaman "cari-2", isinya adalah kecamatan yang bersesuaian dengan kota yang sudah dipilih. Untuk kota, sudah tidak bisa diedit kembali. Kemudian saya membuat method selectKecamatans pada daerahService. Lalu muncul halaman "cari-3" Dengan drop down baru untuk memilih kelurahan. Jika kelurahan sudah terpilih semua maka akan menggunakan pendudukService. Penduduk service akan menggunakan keluargaMapper untuk mengambil list keluarga siapa saja yang ada pada kelurahan berserta dengan anggotanya. Kemudian, hasil dari query tersebut akan ditampilkan ke table yang sudah diimplementasikan dataTables.

9. Optimisasi

Untuk optimisasi, saya sudah menambahkan primary key pada setiap relasi. Semua tabel memiliki ide sebagai primary key-nya. Kemudian, untuk mempercepat query, saya juga menambahkan beberapa indeks. Yaitu pada keluarga dengan nomor_kk sebagai indeks dan penduduk dengan ide_keluarga dan nik sebagai indeks. Dengan penambahan indeks ini bisa mempercepat pada saat melakukan select, bisa dilihat hasilnya pada stress testing fitur 1 dan fitur 2.

10. Stress Testing

Stress testing dilakukan dengan 1000 thread of users dan 10 ramp of seconds.

Fitur 1

<http://localhost:8080/penduduk?nik=3101011405170003>. Menghasilkan rata-rata waktu 6 ms.

View Results in Table

Name:

Comments:

Write results to file / Read from file

File name:

Log/Display Only:
☐ Errors
☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
972	20:53:09.382	Thread Group ...	HTTP Request	9		2727	147	9	0
973	20:53:09.393	Thread Group ...	HTTP Request	8		2727	147	8	0
974	20:53:09.402	Thread Group ...	HTTP Request	6		2727	147	6	0
975	20:53:09.412	Thread Group ...	HTTP Request	10		2727	147	10	1
976	20:53:09.422	Thread Group ...	HTTP Request	6		2727	147	6	1
977	20:53:09.431	Thread Group ...	HTTP Request	9		2727	147	9	1
978	20:53:09.444	Thread Group ...	HTTP Request	9		2727	147	9	1
979	20:53:09.452	Thread Group ...	HTTP Request	7		2727	147	6	1
980	20:53:09.462	Thread Group ...	HTTP Request	8		2727	147	8	0
981	20:53:09.472	Thread Group ...	HTTP Request	5		2727	147	5	0
982	20:53:09.485	Thread Group ...	HTTP Request	6		2727	147	6	1
983	20:53:09.493	Thread Group ...	HTTP Request	6		2727	147	6	0
984	20:53:09.503	Thread Group ...	HTTP Request	5		2727	147	5	0
985	20:53:09.513	Thread Group ...	HTTP Request	5		2727	147	5	1
986	20:53:09.526	Thread Group ...	HTTP Request	6		2727	147	6	0
987	20:53:09.535	Thread Group ...	HTTP Request	7		2727	147	7	1
988	20:53:09.544	Thread Group ...	HTTP Request	6		2727	147	6	1
989	20:53:09.553	Thread Group ...	HTTP Request	5		2727	147	5	1
990	20:53:09.563	Thread Group ...	HTTP Request	5		2727	147	5	0
991	20:53:09.576	Thread Group ...	HTTP Request	7		2727	147	7	1
992	20:53:09.585	Thread Group ...	HTTP Request	6		2727	147	6	0
993	20:53:09.594	Thread Group ...	HTTP Request	7		2727	147	7	0
994	20:53:09.604	Thread Group ...	HTTP Request	5		2727	147	5	0
995	20:53:09.618	Thread Group ...	HTTP Request	6		2727	147	6	1
996	20:53:09.626	Thread Group ...	HTTP Request	6		2727	147	6	0
997	20:53:09.636	Thread Group ...	HTTP Request	5		2727	147	5	1
998	20:53:09.647	Thread Group ...	HTTP Request	5		2727	147	5	0
999	20:53:09.662	Thread Group ...	HTTP Request	6		2727	147	6	1
1000	20:53:09.676	Thread Group ...	HTTP Request	7		2727	147	7	1

☐ Scroll automatically?
☐ Child samples?
No of Samples 1000
Latest Sample 7
Average 6
Deviation 3

Fitur 2

Keluarga?nkk=31010101990001 dapat menghasilkan rata-rata waktu 8 ms.

View Results in Table

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
972	20:55:32.803	Thread Group ...	HTTP Request	8	✓	3623	147	8	0
973	20:55:32.810	Thread Group ...	HTTP Request	8	✓	3623	147	8	1
974	20:55:32.818	Thread Group ...	HTTP Request	8	✓	3623	147	7	1
975	20:55:32.829	Thread Group ...	HTTP Request	7	✓	3623	147	7	0
976	20:55:32.839	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
977	20:55:32.851	Thread Group ...	HTTP Request	8	✓	3623	147	8	0
978	20:55:32.859	Thread Group ...	HTTP Request	7	✓	3623	147	7	1
979	20:55:32.870	Thread Group ...	HTTP Request	7	✓	3623	147	7	0
980	20:55:32.880	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
981	20:55:32.893	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
982	20:55:32.901	Thread Group ...	HTTP Request	7	✓	3623	147	7	0
983	20:55:32.909	Thread Group ...	HTTP Request	7	✓	3623	147	7	1
984	20:55:32.920	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
985	20:55:32.929	Thread Group ...	HTTP Request	6	✓	3623	147	6	1
986	20:55:32.944	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
987	20:55:32.951	Thread Group ...	HTTP Request	7	✓	3623	147	7	1
988	20:55:32.960	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
989	20:55:32.970	Thread Group ...	HTTP Request	7	✓	3623	147	7	0
990	20:55:32.980	Thread Group ...	HTTP Request	7	✓	3623	147	7	0
991	20:55:32.992	Thread Group ...	HTTP Request	7	✓	3623	147	7	1
992	20:55:33.002	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
993	20:55:33.011	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
994	20:55:33.021	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
995	20:55:33.034	Thread Group ...	HTTP Request	7	✓	3623	147	7	1
996	20:55:33.042	Thread Group ...	HTTP Request	7	✓	3623	147	7	1
997	20:55:33.051	Thread Group ...	HTTP Request	7	✓	3623	147	7	0
998	20:55:33.062	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
999	20:55:33.071	Thread Group ...	HTTP Request	6	✓	3623	147	6	0
1000	20:55:33.084	Thread Group ...	HTTP Request	7	✓	3623	147	7	0

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 1000
 Latest Sample 7
 Average 8
 Deviation 9

Fitur 8 merupakan fitur yang paling berat, karena mengambil hasil yang sangat banyak. Berikut ini saya tampilkan beberapa hasil dari jmeter fitur 8 dengan link penduduk/cari?kt=3&kc=1&kl=74

50 Thread of Users

View Results in Table									
Name: View Results in Table									
Comments:									
Write results to file / Read from file									
Filename		Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	Configure			
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
22	23:05:56.480	Thread Group ...	HTTP Request	2976	✓	119352	147	2968	2
23	23:05:56.886	Thread Group ...	HTTP Request	2676	✓	119352	147	2643	1
24	23:05:57.081	Thread Group ...	HTTP Request	2643	✓	119352	147	2635	1
25	23:05:56.280	Thread Group ...	HTTP Request	3490	✓	119352	147	3484	1
26	23:05:56.682	Thread Group ...	HTTP Request	3094	✓	119352	147	3081	1
27	23:05:57.485	Thread Group ...	HTTP Request	2558	✓	119352	147	2549	1
28	23:05:58.084	Thread Group ...	HTTP Request	2041	✓	119352	147	2033	1
29	23:05:57.689	Thread Group ...	HTTP Request	2672	✓	119352	147	2648	1
30	23:05:57.885	Thread Group ...	HTTP Request	2809	✓	119352	147	2636	1
31	23:05:58.288	Thread Group ...	HTTP Request	2506	✓	119352	147	2476	1
32	23:05:58.687	Thread Group ...	HTTP Request	2379	✓	119352	147	2366	1
33	23:05:58.889	Thread Group ...	HTTP Request	2245	✓	119352	147	2239	1
34	23:05:58.485	Thread Group ...	HTTP Request	2714	✓	119352	147	2665	1
35	23:05:59.087	Thread Group ...	HTTP Request	2171	✓	119352	147	2164	1
36	23:05:59.489	Thread Group ...	HTTP Request	1771	✓	119352	147	1765	1
37	23:05:59.688	Thread Group ...	HTTP Request	1796	✓	119352	147	1790	1
38	23:05:59.291	Thread Group ...	HTTP Request	2630	✓	119352	147	2624	1
39	23:06:00.292	Thread Group ...	HTTP Request	2293	✓	119352	147	1978	0
40	23:05:59.888	Thread Group ...	HTTP Request	2798	✓	119352	147	2236	1
41	23:06:00.090	Thread Group ...	HTTP Request	2800	✓	119352	147	2701	0
42	23:06:00.491	Thread Group ...	HTTP Request	2521	✓	119352	147	2486	1
43	23:06:00.690	Thread Group ...	HTTP Request	2412	✓	119352	147	2402	1
44	23:06:01.092	Thread Group ...	HTTP Request	2033	✓	119352	147	1976	1
45	23:06:00.895	Thread Group ...	HTTP Request	2294	✓	119352	147	2278	1
46	23:06:01.292	Thread Group ...	HTTP Request	2082	✓	119352	147	2075	2
47	23:06:01.495	Thread Group ...	HTTP Request	1989	✓	119352	147	1976	1
48	23:06:01.898	Thread Group ...	HTTP Request	1613	✓	119352	147	1607	1
49	23:06:02.095	Thread Group ...	HTTP Request	1481	✓	119352	147	1474	1
50	23:06:01.693	Thread Group ...	HTTP Request	1901	✓	119352	147	1883	1

100 Thread of users

View Results in Table									
Name: View Results in Table									
Comments:									
Write results to file / Read from file									
Filename		Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	Configure			
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
72	23:06:35.273	Thread Group ...	HTTP Request	10877	✓	119352	147	10817	0
73	23:06:40.237	Thread Group ...	HTTP Request	5953	✓	119352	147	5887	2
74	23:06:39.937	Thread Group ...	HTTP Request	6402	✓	119352	147	6235	1
75	23:06:40.136	Thread Group ...	HTTP Request	6217	✓	119352	147	6211	1
76	23:06:41.340	Thread Group ...	HTTP Request	5035	✓	119352	147	5029	2
77	23:06:37.995	Thread Group ...	HTTP Request	8404	✓	119352	147	8397	0
78	23:06:41.141	Thread Group ...	HTTP Request	5267	✓	119352	147	5261	1
79	23:06:38.498	Thread Group ...	HTTP Request	7946	✓	119352	147	7940	1
80	23:06:39.333	Thread Group ...	HTTP Request	7113	✓	119352	147	7092	1
81	23:06:35.471	Thread Group ...	HTTP Request	10987	✓	119352	147	10940	1
82	23:06:41.243	Thread Group ...	HTTP Request	5236	✓	119352	147	5102	2
83	23:06:37.295	Thread Group ...	HTTP Request	9196	✓	119352	147	9189	1
84	23:06:40.041	Thread Group ...	HTTP Request	6467	✓	119352	147	6420	1
85	23:06:40.638	Thread Group ...	HTTP Request	5887	✓	119352	147	5791	1
86	23:06:38.803	Thread Group ...	HTTP Request	7735	✓	119352	147	7729	1
87	23:06:38.922	Thread Group ...	HTTP Request	7628	✓	119352	147	7622	1
88	23:06:39.635	Thread Group ...	HTTP Request	6933	✓	119352	147	6926	0
89	23:06:40.841	Thread Group ...	HTTP Request	5728	✓	119352	147	5722	1
90	23:06:40.741	Thread Group ...	HTTP Request	5870	✓	119352	147	5864	1
91	23:06:38.703	Thread Group ...	HTTP Request	7918	✓	119352	147	7912	1
92	23:06:40.947	Thread Group ...	HTTP Request	5683	✓	119352	147	5673	1
93	23:06:40.339	Thread Group ...	HTTP Request	6324	✓	119352	147	6317	1
94	23:06:41.039	Thread Group ...	HTTP Request	5630	✓	119352	147	5619	1
95	23:06:39.020	Thread Group ...	HTTP Request	7668	✓	119352	147	7656	1
96	23:06:40.542	Thread Group ...	HTTP Request	6148	✓	119352	147	6142	0
97	23:06:40.438	Thread Group ...	HTTP Request	6257	✓	119352	147	6249	1
98	23:06:41.440	Thread Group ...	HTTP Request	5269	✓	119352	147	5262	1
99	23:06:39.534	Thread Group ...	HTTP Request	7185	✓	119352	147	7181	1
100	23:06:39.127	Thread Group ...	HTTP Request	7592	✓	119352	147	7588	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 7592 Average 5485 Deviation 2047

500 thread of users

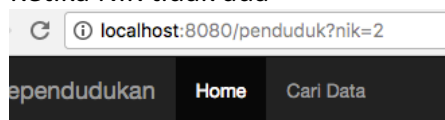
View Results in Table									
Name: View Results in Table									
Comments:									
Write results to file / Read from file									
Filename		Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	Configure			
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
472	23:07:55.853	Thread Group ...	HTTP Request	83455	✓	119352	147	83443	1
473	23:07:55.288	Thread Group ...	HTTP Request	84033	✓	119352	147	84027	0
474	23:07:56.378	Thread Group ...	HTTP Request	82943	✓	119352	147	82832	0
475	23:07:56.340	Thread Group ...	HTTP Request	83106	✓	119352	147	83100	1
476	23:07:55.490	Thread Group ...	HTTP Request	83958	✓	119352	147	83952	1
477	23:07:55.186	Thread Group ...	HTTP Request	84304	✓	119352	147	84295	1
478	23:07:55.976	Thread Group ...	HTTP Request	83565	✓	119352	147	83536	1
479	23:07:55.530	Thread Group ...	HTTP Request	84022	✓	119352	147	84016	1
480	23:07:55.651	Thread Group ...	HTTP Request	83943	✓	119352	147	83937	1
481	23:07:55.954	Thread Group ...	HTTP Request	83673	✓	119352	147	83668	1
482	23:07:55.893	Thread Group ...	HTTP Request	83768	✓	119352	147	83763	1
483	23:07:55.693	Thread Group ...	HTTP Request	83970	✓	119352	147	83965	1
484	23:07:56.236	Thread Group ...	HTTP Request	83476	✓	119352	147	83426	1
485	23:07:56.417	Thread Group ...	HTTP Request	83304	✓	119352	147	83297	1
486	23:07:56.096	Thread Group ...	HTTP Request	83627	✓	119352	147	83438	1
487	23:07:55.064	Thread Group ...	HTTP Request	84674	✓	119352	147	84669	1
488	23:07:56.256	Thread Group ...	HTTP Request	83485	✓	119352	147	83478	1
489	23:07:55.732	Thread Group ...	HTTP Request	84036	✓	119352	147	84016	1
490	23:07:56.298	Thread Group ...	HTTP Request	83470	✓	119352	147	83450	1
491	23:07:56.018	Thread Group ...	HTTP Request	83755	✓	119352	147	83750	1
492	23:07:56.116	Thread Group ...	HTTP Request	83664	✓	119352	147	83658	1
493	23:07:55.934	Thread Group ...	HTTP Request	83853	✓	119352	147	83703	1
494	23:07:55.592	Thread Group ...	HTTP Request	84200	✓	119352	147	84046	1
495	23:07:55.673	Thread Group ...	HTTP Request	84128	✓	119352	147	84122	0
496	23:07:55.613	Thread Group ...	HTTP Request	84190	✓	119352	147	84182	1
497	23:07:55.813	Thread Group ...	HTTP Request	84000	✓	119352	147	83993	1
498	23:07:56.137	Thread Group ...	HTTP Request	83697	✓	119352	147	83691	1
499	23:07:55.408	Thread Group ...	HTTP Request	84446	✓	119352	147	84440	1
500	23:07:55.793	Thread Group ...	HTTP Request	84068	✓	119352	147	84061	1
<input type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples? No of Samples 500 Latest Sample 84068 Average 55336 Deviation 20648									

Jika dilihat, semakin banyak yang mengakses maka semakin lama pula rata-rata waktu yang akan digunakan untuk mengakses halaman tersebut. Padahal, jika user hanya 1 hanya menggunakan waktu 528 ms. Fitur 8 ini sebenarnya sebelum diberikan index, sangat amat lebih lama dari yang sekarang. Index sudah memberikan keringanan untuk mengakses fitur 8 ini.

501	23:13:36.502	Thread Group ...	HTTP Request	528	✓	119352	147	524	3
-----	--------------	------------------	--------------	-----	---	--------	-----	-----	---

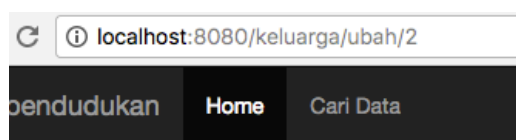
11. Error Page

Ketika NIK tidak ada



Tidak Ditemukan!

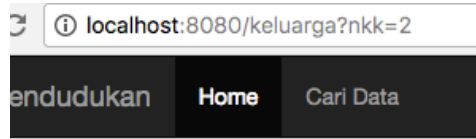
Penduduk tidak ditemukan



Tidak Ditemukan!

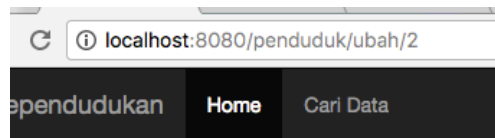
Keluarga Tidak Ditemukan

Ketika NKK tidak ada



Tidak Ditemukan!

Keluarga Tidak Ditemukan



Tidak Ditemukan!

Penduduk Tidak Ditemukan