NETWORK LABORATORY
Electrical Engineering Department, 2nd floor
Universitas Indonesia
Depok. 16424

TUGAS PENDAHULUAN
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
MODUL 8 – Web Integration

Pastikan anda memiliki koneksi internet saat mengerjakan tugas ini!
Pastikan program JHotel anda sudah berjalan dengan benar!
Pastikan anda sudah bisa mejalankan program JHotel anda di IntelliJ!

Part 1. Setup

PETUNJUK DI TUGAS INI MENGASUMSIKAN ANDA MENGGUNAKAN INTELLIJ, JIKA MENGGUNAKAN IDE LAIN MOHON DISESUAIKAN

1. Lakukan instalasi Gradle. Petunjuk instalasi dapat anda temukan di sini: https://gradle.org/install/
   Anda dapat melakukan instalasi dengan *package manager* atau secara manual.
   Setelah instalasi, buatlah *screenshot* yang menunjukkan versi Gradle anda dengan memasukkan perintah `gradle -v` pada command prompt.

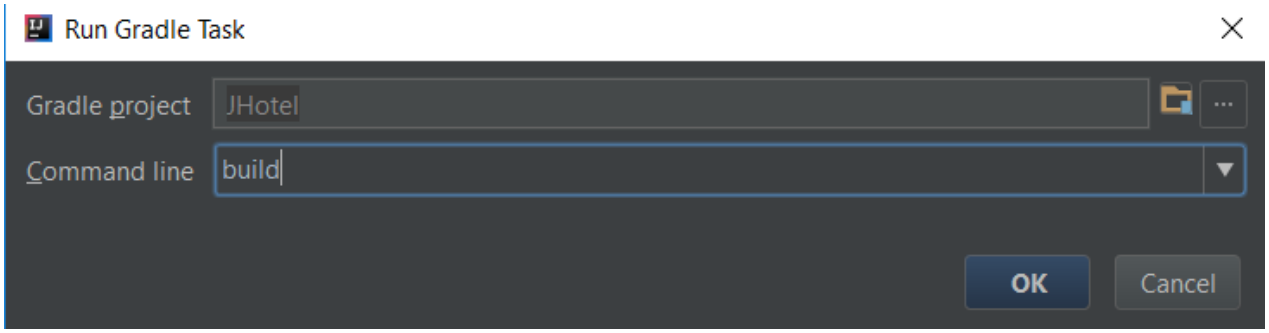
2. Backup folder JHotel anda ke tempat lain. Pindahkanlah kode JHotel anda dari /src ke folder /src/main/java/jhotel. Lalu masukkanlah file **build.gradle** berikut ini ke direktori utama JHotel anda (bukan src atau subdirektorinya).

   Link: https://github.com/gatitoneku/jhotel-8/blob/master/build.gradle

   File **build.gradle** berisi konfigurasi proyek yang menggunakan Gradle.

3. Pada IntelliJ buka project JHotel anda. Selanjutnya buka *File->New->Module From Existing Sources*, pilih file build.gradle yang baru anda buat, akan muncul layar *Import Module from Gradle*. Pilih opsi *Use auto-import*, *Use local gradle distribution*, lalu pada *Gradle home*, masukkan lokasi instalasi Gradle di komputer anda.

4. Pada IntelliJ, buka *View->Tool Windows->Gradle*, lalu pada window yang muncul, klik lambang Gradle (jika di *hover* akan muncul tulisan *Execute Gradle Task*). Pilih *Tasks->build->build* atau masukkan perintah seperti dibawah ini lalu klik OK. Berikan screenshot tulisan BUILD SUCCESSFUL.



Anda juga dapat melakukan langkah ini dengan command prompt. Pada folder JHotel anda masukkan perintah `gradle build`.

5. Pada Gradle Tool Window yang dibuka pada No. 5, dibawah menu JHotel, bukalah *Tasks->application->bootRun.* Ini akan menjalankan program pada IntelliJ anda. Screenshot-lah hasilnya!

6. Pada folder /src/main/java/jhotel, buatlah folder bernama *controller*.

Part 2: Coding

1. Pada semua class, tambahkan potongan kode
```
package jhotel;
```
Pada bagian paling atas.

2. Pada class JHotel anda, tambahkan potongan kode sebagai berikut pada bagian awal (sebelum deklarasi class JHotel):
```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
```
Annotation @SpringBootApplication memberi tahu main class dari aplikasi Spring Boot.

Kosongkan isi method main() anda lalu di dalamnya tambahkan potongan kode berikut:
```
SpringApplication.run(JHotel.class, args);
```
Nama class disesuaikan dengan program anda. Pada contoh ini nama class yang mengandung method main() adalah JHotel.

Buatlah file CustomerController.java pada folder controller dengan konten sebagai berikut:

Link: https://github.com/gatitoneku/jhotel-8/blob/master/CustomerController.java

Lalu jalankan program dengan bootRun. Pada bagian akhir program akan muncul tulisan seperti dibawah ini dan informasi lainnya tentang Spring Boot:

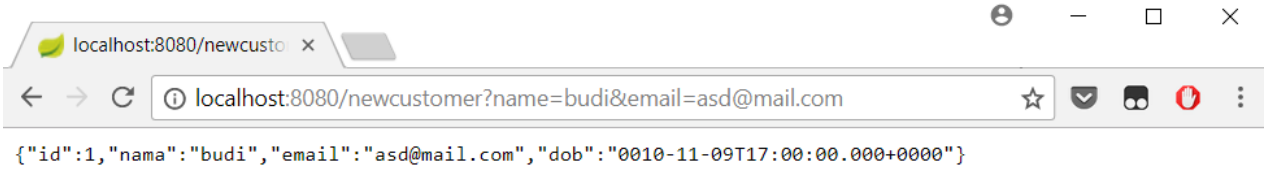
Jika berhasil, ketika anda scroll output program kebawah maka muncul log semacam ini:

Bukalah http://localhost:8080 pada browser anda. Akan muncul halaman seperti berikut:

Bukalah URL http://localhost:8080/?name=<nama_anda>, akan muncul tampilan seperti ini:

Berikan screenshot bahwa tampilan yang anda dapatkan sudah benar!

3. Masukkan URL
http://localhost:8080/newcustomer?name=budi&email=asd@mail.com
pada browser anda. Akses ke URL ini akan membuat customer baru dengan nama dan tahun lahir yang dapat anda tentukan. Seharusnya akan tampil seperti ini:



Ini merupakan tampilan JSON dari objek Customer tersebut.
Buatlah lagi customer baru dengan nama anda dan email pilihan anda lalu berikan screenshot bahwa tampilan yang anda dapatkan sudah benar!

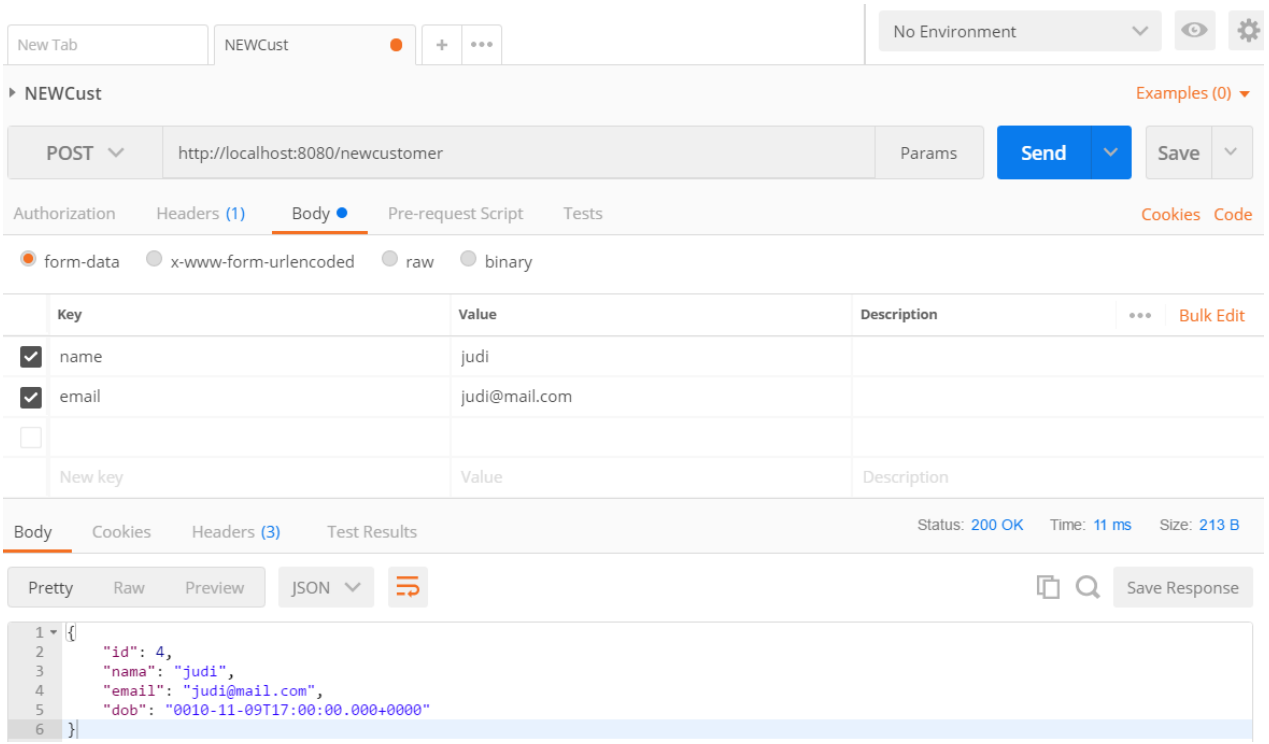
4. Masukkan URL http://localhost:8080/getcustomer/{id_anda} ke browser, dengan {id_anda} merupakan id dari customer yang anda buat di No. 3. Berikan screenshot yang membuktikan bahwa customer yang ditampilkan sama dengan di No. 3!

   Catatan: Jika ingin mengubah kode perlu me-*restart* program. Anda dapat mematikan program dengan tombol kotak merah di sebelah kanan atas IntelliJ.

5. Anda sudah berhasil membuat Customer baru, namun masih menggunakan method GET. Metode ini tidak aman untuk operasi *create* atau *update* (ingat CRUD). Ubah method newcust() beserta annotation @RequestMapping-nya sehingga menjadi POST.

```
@RequestMapping(value = "/newcustomer", method = RequestMethod.POST)
    public Customer newCust(@RequestParam(value="name") String name,
@RequestParam(value="email") String email) {
        Customer customer = new Customer(name, 10, 10, 10, email);
        try {
            DatabaseCustomer.addCustomer(customer);
        } catch(Exception ex) {
            ex.getMessage();
            return null;
        };
        return customer;
    }
```

Selanjutnya, jalankan lagi program anda. Lalu bukalah Postman dan buatlah request ke URL http://localhost:8080/newcustomer dengan contoh key dan value sebagai berikut:



Jika Customer berhasil terbuat, maka datanya akan tampil pada bagian *Body*. Selanjutnya bukalah browser anda, dan masukkan url http://localhost:8080/getcustomer/{id_anda} dimana {id_anda} merupakan id dari Customer yang barusan anda buat.
Screenshotlah hasil request POST newcustomer anda di Postman dan getcustomer!

6. Sebelumnya anda dapat membuat Customer, namun jika anda perhatikan anda hanya dapat menentukan nama dan email sehingga atribut lainnya selalu sama. Ubahlah method newcust() agar anda dapat menentukan atribut tahun lahirnya dengan nilai default adalah 2000! Buktikan dengan screenshot!

7. Lakukan commit dengan pesan “Modul 8 TP” lalu push ke Github.