



PEMROGRAMAN BERORIENTASI OBJEK 2017/2018

CASE STUDY

MODUL 5: toString, Date, dan RegEx

Untuk melanjutkan program JHotel yang dibuat, harap untuk melihat UML Modul 5 yang telah disediakan. **Praktikan diharapkan menyesuaikan program JHotel sesuai dengan UML Modul 5!**

Perlu diperhatikan bahwa pada modul ini sistem JHotel masih dalam tahap **single-user**, sehingga masing-masing database hanya perlu menyimpan satu data, yaitu satu **Customer**, satu **Room**, dan satu **Pesanan**.

Pada modul ini data yang dikumpulkan adalah :

- ☐ **1 commit** pada github dengan batas waktu adalah 22 Maret 2018 pukul 18:00.
- ☐ **1 dokumentasi** yang berisikan screenshot dari setiap step, beserta penjelasannya.

Tugas 1: Fail-safe mechanism

1. Sesuaikan dengan UML untuk modul 5!
2. Jika pada UML terdapat suatu Class yang memiliki variable baru, buatlah accessor/mutator untuk variable tersebut.
3. Pada mutator variable E-Mail pada customer, buatlah mekanisme pengecekan input-an, sehingga parameter alamat E-Mail yang diterima hanyalah yang memenuhi ketentuan sebagai berikut:
 - ☐ Berbentuk *local@domain* eg. etd@ui.ac.id
 - ☐ Pada bagian local, dapat menggunakan alfabet (Upper maupun Lower)
 - ☐ Pada bagian local, dapat menggunakan digit numeral 0-9
 - ☐ Boleh menggunakan titik (.) asalkan TIDAK BERADA pada awal alamat dan TIDAK BERURUTAN dengan titik lain (Contoh salah : .etd@ui.ac.id atau et.d@ui.ac.id)
 - ☐ Boleh menggunakan special char



- ☐ Pada bagian domain, dapat menggunakan alfabet (Upper maupun Lower)
 - ☐ Pada bagian domain, dapat menggunakan digit numeral 0-9
 - ☐ Pada bagian domain, dapat menggunakan (-) asalkan TIDAK BERADA pada awal/permulaan bagian domain
 - ☐ Pada kedua bagian tidak diperbolehkan adanya whitespace.
4. Lakukan pengujian terhadap mekanisme pengecekan yang telah anda buat pada poin ke 3. Buat 3 objek Customer, dan pada tiap objek panggil method mutator E-Mail dan berikan masukkan e.t.d@, Ee_.td@ui.ac.id, dan {namaPraktikan}.modul5@netlab.com. Jelaskan hasil yang didapatkan.

Tugas 2:

1. Pastikan anda sudah membuat accessor dan mutator untuk variable tanggalPesanan (Pesanan) dan dob (Customer).
2. Pada kelas Customer, ubah Constructor sehingga menerima parameter berupa tanggal, bulan, tahun yang nantinya dapat digunakan untuk menginisiasi variable dob.
3. Gunakan metode overloading untuk Constructor milik Customer, sehingga untuk membuat objek customer baru bisa juga menggunakan perintah

```
Customer cust1 = new Customer(yourId, yourName, new GregorianCalendar  
(Year,Month,Date).getTime());
```
4. Pada contoh diatas, mengapa sebaiknya parameter Date yang dibutuhkan didapatkan dari 'new GregorianCalendar (Year,Month,Date).getTime()', bukan 'new Date (Year,Month,Date)' ?
5. Pada method getDOB, lakukan pemrograman sehingga DOB milik objek tersebut dapat dicetak. JANGAN gunakan println, dan pastikan hasil cetakan mirip contoh berikut: "DOB: 20 March 2018"
6. Ubah method diatas namun kali ini GUNAKAN println!
7. Lakukan poin 1 hingga 3 untuk variable tanggalPesanan pada class Pesanan!



Tugas 3:

Prelude:

Pada modul kali ini, data type list_pesanan pada DatabasePesanan menjadi Pesanan, dan sesuaikan metode-metode yang ada didalam kelas tersebut. Lakukan juga proses tersebut pada kelas DatabaseCustomer.

Selanjutnya, buatlah constructor pada semua kelas enum, dan tambahkan method toString() yang menghasilkan (me-return) String deskripsinya.

Selanjutnya:

1. Ubah metode printData() pada class Lokasi dan Hotel, menjadi toString yang menampilkan nilai-nilai dari instance variable class tersebut.
2. Ubah metode printData() pada Class Customer menjadi toString(). Hasil dari toString() nantinya dibagi menjadi 2 kondisi, saat Pelanggan memiliki pesanan (DatabasePesanan.getPesanan() tidak menghasilkan null) dan saat Pelanggan tersebut tidak memiliki pesana.

Pastikan toString() tersebut akan menghasilkan String seperti pada gambar dibawah ini saat memiliki pesanan

```
Customer ID      : custID
Name             : custName
E-Mail          : custEmail
Date of birth    : custDOB
Booking order is in progress
```

Sedangkan jika Pelanggan tidak memiliki pesanan, pastikan String yang dihasilkan seperti pada gambar dibawah ini

```
Customer ID      : custID
Name             : custName
E-Mail          : custEmail
Date of birth    : custDOB
```

3. Untuk class Room ubah printData() menjadi toString. String yang dihasilkan bergantung apakah kamar tersebut ada yang menggunakan atau tidak. Jika kosong, maka String yang dihasilkan adalah

```
Nama Hotel      :
Tipe Kamar      :
Harga           :
Status Kamar    : .
```

Sedangkan jika ada penggunanya maka



Nama Hotel :
Tipe Kamar :
Harga :
Status Kamar :
Pelanggan :

4. Ubah method `printData()` pada kelas `Pesanan` menjadi method `toString()`. Deklarasikan variable lokal dengan tipe `String` yang bernama `final_status` dengan nilai "KOSONG". `String` yang dihasilkan oleh metode `toString()` akan dibagi menjadi beberapa kondisi dengan ketentuan:
- status diproses dan belum selesai, maka `final_Status` akan bernilai "DIPROSES".
 - Jika status tidak diproses dan tidak selesai, maka `final_status` akan berisi "KOSONG".
 - Jika status tidak diproses dan selesai, maka fi
 - nal_status "SELESAI".

Selanjutnya, tentukan `String` output menjadi

("Dibuat oleh " + `namaPelanggan` + ". Proses booking untuk " + `namaHotel` + " kamar nomor " + `nomorKamar` + " dengan tipe kamar yang diinginkan " + `tipeKamar` + ". Status: " + `final_status` + ".")

5. Berikan contoh hasil dari `toString` pada poin 2, 3 dan 4 untuk masing-masing kondisi.