



---

---

## PEMROGRAMAN BERORIENTASI OBJEK 2017/2018

### KUNCI CASE STUDY

### MODUL 4: ABSTRACTION, INHERITANCE, & POLYMORPHISM

Untuk melanjutkan program JHotel yang dibuat, harap untuk melihat UML Modul 4 yang telah disediakan. **Praktikan diharapkan menyesuaikan program JHotel sesuai dengan UML Modul 4!**

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 masing-masing batas waktu adalah **8 Maret 2018 18:00**

#### **Tugas 1: Menyesuaikan kelas Room**

Sesuaikan kelas room dengan ketentuan sebagai berikut:

1. Sesuaikan dengan UML untuk modul 4, terdapat beberapa perubahan yang terjadi untuk modul ini, misalnya *Access Modifier*, *Non-Access Modifier*, penambahan *variable* dan/atau *method*, perubahan parameter, atau *constructor*
2. Tambahkan pencetakan tipe kamar pada method `printData()`
3. Hapus pencetakan nama pelanggan yang ada pada method `printData()`
4. Sesuaikan pencetakan daily tariff pada method `printData()`



### **Tugas 2: Membuat kelas subclass SingleRoom**

Buatlah sub-kelas **SingleRoom** dengan ketentuan sebagai berikut:

1. Isi konstanta TIPE\_KAMAR dengan enum tipe kamar single
2. Untuk constructor: assign hotel, nomor\_kamar, isAvailable, dan status\_kamar ke superclass yang dimiliki
3. Sesuaikan sisanya dengan UML untuk modul 4

### **Tugas 3: Membuat kelas subclass DoubleRoom**

Buatlah sub-kelas **DoubleRoom** dengan ketentuan sebagai berikut:

1. Isi konstanta TIPE\_KAMAR dengan enum tipe kamar double
2. Untuk constructor: assign hotel, nomor\_kamar, isAvailable, dan status\_kamar ke superclass yang dimiliki
3. Isi setter dan getter untuk customer2
4. Sesuaikan sisanya dengan UML untuk modul 4

### **Tugas 4: Membuat kelas subclass PremiumRoom**

Buatlah sub-kelas **PremiumRoom** dengan ketentuan sebagai berikut:

1. Isi konstanta TIPE\_KAMAR dengan enum tipe kamar premium
2. Untuk constructor: assign hotel, nomor\_kamar, isAvailable, dan status\_kamar ke superclass yang dimiliki
3. Isi konstanta discount dengan nilai 0.3
4. Isi body method setDailyTariff dengan men-set variable dailytariff yang di-input \* DISCOUNT
5. Sesuaikan sisanya dengan UML untuk modul 4



### **Tugas 5: Menyesuaikan kelas Pesanan**

Sesuaikan kelas **Pesanan** dengan ketentuan sebagai berikut

1. Sesuaikan dengan UML untuk modul 4, terdapat beberapa perubahan yang terjadi untuk modul ini, misalnya *Access Modifier*, *Non-Access Modifier*, penambahan *variable* dan/atau *method*, perubahan parameter, atau *constructor*
2. Pada method `setBiaya()` besar biaya yang perlu dibayarkan, didapatkan dari `dailytariff` dikalikan dengan `jumlahHari`
3. Pada constructor isi variabel biaya dengan cara penghitungan yang sama dengan poin 2
4. Hapus pencetakan `tipekamar` yang ada pada method `printData()`
5. Sesuaikan cara pencetakan nama pelanggan pada method `printData()`
6. Tambahkan pencetakan jumlah hari pada method `printData()`
7. Tambahkan pencetakan biaya pada method `printData()`

### **Tugas 7: Pengujian Sistem JHotel**

Pada tugas ini akan melakukan pengujian sistem JHotel yang dilakukan di kelas **JHotel**.

- a. Buat objek **Hotel**, objek **SingleRoom** (dengan tipe data Room), objek **Customer**, objek **Lokasi**, dan objek **Pesanan**. Jangan lupa set `dailytariff` dan hitung biaya yang perlu dibayarkan  
Nilai untuk masing-masing objek tersebut dibiarkan.
- b. Print data object lokasi, customer, dan hotel.
- c. Jalankan method `pesananDitugaskan()` yang ada di kelas Administrasi dan kemudian lakukan pencetakan untuk kamar dan pesanan yang telah dibuat
- d. Lakukan pengecekan apakah objek **SingleRoom** yang ada poin 7a merupakan **DoubleRoom**. Apabila benar, lakukan pencetakan “Benar Double Room”, bila tidak lakukan pencetakan “Salah, bukan Double Room”

**Hint: Gunakan instanceof**



- e. Buatlah objek **DoubleRoom** (dengan tipe data Room), dan objek **Pesanan** baru untuk pelanggan dan hotel yang telah dibuat pada point 7a
- f. Set dailytariff, hitung biaya yang perlu dibayarkan, dan jalankan method pesananDitugaskan() yang ada di kelas Administrasi
- g. Lakukan pencetakan untuk kamar dan pesanan yang telah dibuat
- h. Lakukan kembali langkah poin 7d untuk objek DoubleRoom yang dibuat pada poin 7e
- i. Lakukan **commit** pada Github dengan message “Modul 4”.