

Soal Lab 2 (Senin)
Pengantar Class dan Object

Dasar-Dasar Pemrograman 2
CSGE601021
Semester Genap 2016/2017

Batas waktu pengumpulan:

Senin, 20 Februari 2017 pukul 18.00 Waktu Scele (Lab Senin)

Peringatan: Jangan mengumpulkan pekerjaan beberapa menit menjelang batas waktu pengumpulan karena ada kemungkinan pengumpulan gagal dilakukan atau koneksi internet terputus!

Soal Tutorial

Nabil adalah seorang mahasiswa Fasilkom yang sangat memerhatikan kesehatannya. Pada suatu hari, dia menyadari bahwa pola makannya selama ini sangat tidak teratur dan perlu diperbaiki. Selain itu, Nabil juga sudah jarang melakukan olah raga. Oleh karena itu, Nabil memiliki ide untuk membuat suatu program yang dapat menghitung informasi kalori, BMR (*basal metabolic rate*), serta kebutuhan kalori harian agar pola makan dan olah raganya dapat direncanakan secara lebih teratur. Selain itu, program ini juga bisa digunakan untuk orang lain yang membutuhkannya. Nabil membuat programnya dengan memanfaatkan pemrograman berorientasi objek menggunakan bahasa Java, namun karena Nabil belum begitu lama melakukan pemrograman berorientasi objek, dia mengalami kesulitan dalam membuat kelas-kelas serta objek-objek. Oleh karena itu, kamu perlu membantu Nabil dalam menyelesaikan programnya.

Terdapat kelas **Manusia**, **Makanan**, **Gym**, dan **Kelas Utama**. Berikut adalah daftar kelas serta penjelasan *method* yang dimilikinya.

Manusia.java Kelas yang merepresentasikan seorang manusia.	
hitungBMR	Menghitung BMR*. Gunakan rumus hasil revisi Mifflin dan St Jeor pada tahun 1990*.
hitungKaloriHarian	Menghitung kebutuhan kalori harian* dengan asumsi entitas manusia tersebut melakukan <i>light exercise</i> setiap harinya.
makan	<p>Menambah total kalori yang dimiliki objek manusia dengan jumlah kalori pada makanan serta mencetak keluaran dengan format: [nama manusia] memakan [nama makanan] yang rasanya [rasa].</p> <p>Atau jika kalori yang akan ditambahkan ke total kalori objek manusia akan melebihi kebutuhan kalori hariannya, maka tolak dan cetak keluaran dengan format: [nama manusia] tidak bisa memakan [nama makanan] karena akan melebihi batas kebutuhan kalori hariannya.</p>
cetakInformasi	<p>Mencetak informasi objek manusia dengan format:</p> <pre>[nama] Umur : [umur] Tahun Berat : [berat] KG Tinggi : [tinggi] CM Kalori : [kalori] BMR : [hitungBMR] KKH : [hitungKaloriHarian] Kalori/hari</pre> <p>Contoh:</p> <pre>John Smith Umur : 25 Tahun Berat : 60 KG Tinggi : 180 CM Kalori : 0</pre>

	BMR : 1618 KKH : 2207 Kalori/hari
--	--------------------------------------

*) Referensi: https://en.wikipedia.org/wiki/Harris%E2%80%93Benedict_equation

Makanan.java Kelas yang merepresentasikan suatu makanan.
--

Gym.java Kelas yang merepresentasikan suatu gym.	
gunakanTreadmill	<p>Mengurangi totalKalori dari objek manusia sebanyak kalkulasi kalori, tolak jika totalKalori yang ada kurang dari kalori yang akan dibakar.</p> <p>Serta, mencetak keluaran dengan format: [nama manusia] menggunakan treadmill selama [waktu] dengan kecepatan [kecepatan]</p> <p>Atau jika totalKalori yang ada kurang dari kalori yang akan dibakar: [nama manusia] tidak dapat menggunakan treadmill karena kekurangan energi.</p>
hitungKaloriTreadmill	Mengembalikan hasil perhitungan kalori yang dibakar jika menggunakan treadmill.
gunakanStationaryBike	<p>Mengurangi totalKalori dari objek manusia sebanyak kalkulasi kalori, tolak jika totalKalori yang ada kurang dari kalori yang akan dibakar.</p> <p>Serta, mencetak keluaran dengan format: [nama manusia] menggunakan stationary bike selama [waktu].</p> <p>Atau jika totalKalori yang ada kurang dari kalori yang akan dibakar: [nama manusia] tidak dapat menggunakan stationary bike karena kekurangan energi.</p>
hitungKaloriStationaryBike	Mengembalikan hasil perhitungan kalori yang dibakar jika menggunakan stationary bike.

Terdapat *template* program yang berisi kelas utama dan kelas-kelas di atas beserta *instance variable* yang dimilikinya. Kalian diminta untuk melengkapi program tersebut.

1. Lengkapi tipe data yang sesuai untuk setiap *instance variable* dari setiap kelas.
2. Lengkapi parameter dan isi dari *constructor* setiap kelas.

3. Lengkapi isi *method* dari tiap kelas sesuai dengan fungsionalitasnya.
4. Tambahkan setter dan getter method pada setiap kelas.
5. Instansiasi setiap objek sesuai dengan yang diminta pada kelas utama.

Format Pengumpulan

- KELAS_NPM_TUTORIAL2.zip contoh: B_1606123456_TUTORIAL2.zip
- Isi file KELAS_NPM_TUTORIAL2.zip:
 - DDP2Tutorial2.java (kelas utama yang digunakan untuk menjalankan program)
 - Manusia.java
 - Makanan.java
 - Gym.java