

Dasar-Dasar Pemrograman 2

Lab 06

Inheritance 2



FAKULTAS
ILMU
KOMPUTER

Versi Dokumen

- Versi 1 (13-04-2022; 16.00 WIB): Rilis Pertama
- **Versi 2** (13-04-2022; 17.27 WIB): Perintah menggunakan Array dan revisi *template*
- **Versi 3** (13-04-2022; 18.40 WIB): Perbaikan deskripsi perintah ADD
- **Versi 4** (14-04-2022; 11.29 WIB): Penambahan keterangan *handling* batas happiness

Sistem Interaksi Warga RS



(source : <https://static.wikia.nocookie.net/16044dbb-8913-4259-91a4-b02c2d271622>)

Setelah menjalani UTS, kamu kelelahan dan pergi ke Rumah Sakit untuk melakukan pemeriksaan kondisi kesehatan baik fisik maupun mental. Sesampainya di rumah sakit, kamu tidak sengaja berpapasan dengan Dek Depe yang ternyata merupakan anak dari pemilik rumah sakit tersebut. Dek Depe ingin mengetahui aktivitas apa saja yang dilakukan oleh warga rumah sakit, maka ia membuat sebuah program

untuk mencatat segala aktivitas yang dilakukan oleh warga rumah sakit. Melihat kamu yang hebat dalam membuat program, Dek Depe meminta bantuanmu untuk menyelesaikan program sistem interaksi warga rumah sakit.

Task

Terdapat 4 buah class yang harus kamu lengkapi:

- **Warga**

Class Warga merupakan *parent class* dari class Pasien dan Dokter. Lengkapilah *constructor* dan *method* yang ada pada *template*. Adapun atribut pada class Warga:

- **String nama**
 - Atribut ini merepresentasikan nama dari warga yang terdaftar pada sistem.
- **Warga[] logInteraksi**
 - Atribut ini menyimpan objek Warga yang berinteraksi dengan dirinya. **Inisiasi Array Warga sudah tersedia pada template.**

- **Pasien**

Class Pasien merupakan *subclass* dari Warga. Adapun atribut tambahan pada class Pasien adalah:

- **int happiness**
 - Atribut ini merepresentasikan tingkat kebahagiaan pasien karena berinteraksi dengan warga rumah sakit. Atribut ini memiliki **nilai awal 0, nilai minimal 0** (inklusif), dan **nilai maksimal 100** (inklusif). Jika tingkat kebahagiaan kurang dari 0(negatif) atau lebih dari 100, maka harus dihandle. Berikut merupakan perubahan tingkat kebahagiaan yang mungkin terjadi:
 - Jika penyakit pasien berhasil disembuhkan, tingkat kebahagiaannya akan **bertambah 20**.
 - Jika pasien berinteraksi dengan dokter yang ramah, tingkat kebahagiaannya akan **bertambah 10**.
 - Jika pasien berinteraksi dengan dokter yang tidak ramah, tingkat kebahagiaannya akan **berkurang 5**.
 - Jika pasien bertemu dengan pasien lain, tingkat kebahagiaannya akan **bertambah 5**.

Untuk melengkapi penjelasan, berikut merupakan ketentuan yang dapat digunakan untuk melakukan *handling* batasan nilai happiness:

- Prioritas penambahan atau pengurangan happiness adalah dari atas ke bawah.

- Setiap happiness mengalami perubahan nilai, pengecekan batas akan selalu berlaku.
- Jika 1 interaksi membuat nilai happiness mengalami perubahan sebanyak dua kali (misalnya karena pasien sembuh dan dokter yang menyembuhkan tidak ramah), maka pengecekan batas akan terjadi dua kali.
- Contoh:

Budi adalah dokter dengan penyakitKeahlian batuk dan merupakan seorang dokter tidak ramah. Andi adalah seorang pasien dengan penyakit batuk.

Happiness Andi setelah melakukan beberapa interaksi diketahui bernilai 90 dan penyakitnya masih belum sembuh (penyakitSembuh = false).

Jika andi bertemu Budi, maka penyakitnya akan sembuh dan menyebabkan happiness Budi akan bertambah 20 poin menjadi 110. **Karena melebihi 100, maka poin dijadikan 100.**

Karena Budi merupakan dokter tidak ramah, **maka happiness Andi akan berkurang 5 sehingga menjadi 95.**

- **String penyakit**
 - Atribut ini merepresentasikan penyakit yang dimiliki oleh pasien dan ingin disembuhkan di rumah sakit.
- **boolean pasienSembuh**
 - Atribut ini merepresentasikan status kesembuhan pasien. Pada awal inisiasi object, atribut ini bernilai **false**. Jika pasien sudah berinteraksi dengan dokter yang dapat menyembuhkan penyakitnya (tidak *case sensitive*), maka statusnya menjadi **true**.
- **Dokter**

Class Dokter merupakan *subclass* dari Warga. Adapun atribut tambahan pada *class* Dokter adalah:

 - **int jumlahPasienDitemui**
 - Atribut ini menghitung berapa pasien yang berinteraksi dengan dokter tersebut.
 - **boolean dokterRamah**
 - Atribut ini akan memberikan tanda apakah dokter merupakan dokter yang ramah atau tidak.
 - **String penyakitKeahlian**

- Atribut ini akan menyimpan nama penyakit yang dapat disembuhkan oleh dokter.
- **RumahSakit**

RumahSakit merupakan *class* yang akan memuat *main method*. Berikut merupakan atribut yang dimiliki oleh *class* RumahSakit:

 - **Warga[] daftarWarga**
 - Atribut ini akan menyimpan *instance* Warga yang ada pada rumah sakit, terdiri dari dokter dan pasien. **Inisiasi Array Warga sudah tersedia pada template.**

Catatan: Anda diperbolehkan menambah atribut dan *method* baru pada setiap *class* apabila diperlukan, tetapi semua atribut dan *method* yang sudah didefinisikan pada dokumen soal dan template tidak boleh diubah.

Format masukan

- Baris pertama berisi N , banyaknya perintah yang akan dilakukan.
- N Baris berikutnya mendeskripsikan setiap perintah yang dilakukan. Adapun list perintah yang mungkin dimasukkan adalah sebagai berikut:
 - **ADD DOKTER X P R**
 - ◆ Menambahkan seorang dokter bernama X dengan penyakit keahlian P dan keramahan R (Yes/No) ke dalam **Array daftarWarga**
 - **ADD PASIEN X P**
 - ◆ Menambahkan seorang pasien bernama X dengan penyakit P ke dalam **Array daftarWarga**
 - **INTERAKSI X Y**
 - ◆ Warga X melakukan interaksi dengan Y dan sebaliknya. **X dan Y dijamin berbeda dan sudah ada pada sistem saat perintah ini dilakukan.**
 - **LOG X**
 - ◆ Jika X merupakan seorang dokter, cetak **jumlah pasien** yang berinteraksi dengannya dan cetak **nama-nama warga** (termasuk dokter) yang berinteraksi dengannya.
 - ◆ Jika Y merupakan seorang pasien, cetak **tingkat kebahagiaannya, status kesembuhannya, dan nama-nama warga** yang berinteraksi dengannya.

Format keluaran

- Format keluaran untuk LOG X terdapat pada *template*.

Test Case

Input 1

```
8
ADD DOKTER Andi Batuk Yes
ADD DOKTER Charlie Batuk No
ADD PASIEN Budi Batuk
INTERAKSI Andi Budi
INTERAKSI Andi Charlie
LOG Andi
LOG Budi
LOG Charlie
```

Output 1

```
1
Budi
Charlie
-----
30
true
Andi
-----
0
Andi
-----
```

Input 2

```
13
ADD PASIEN Dekdepe Demam
ADD PASIEN Eva Pusing
INTERAKSI Dekdepe Eva
ADD DOKTER Fira Diabetes Yes
ADD DOKTER Galih Demam No
INTERAKSI Dekdepe Fira
INTERAKSI Dekdepe Galih
INTERAKSI Eva Galih
INTERAKSI Galih Eva
LOG Dekdepe
LOG Eva
LOG Fira
LOG Galih
```

Output 2

```
30
true
Eva
Fira
Galih
-----
0
false
Dekdepe
Galih
Galih
-----
1
Dekdepe
-----
3
Dekdepe
Eva
Eva
-----
```

Komponen Penilaian

- 40% Kebenaran kode
- 20% Implementasi *inheritance*
- 20% *Overriding method*
- 10% Dokumentasi dan kerapian kode
- 10% Standar penulisan kode

* Ketentuan yang harus dipenuhi yaitu:

1. Indentasi yang konsisten ([sumber](#))
2. Aturan penamaan harus mengikuti Java Naming Convention ([sumber](#))
3. Penamaan *module*, *class*, *method*, dan variabel yang tidak ambigu

Template

Template dapat diakses [di sini](#).

Revisi

1. Memperbaiki keterangan soal terkait tipe data dari atribut logInteraksi pada class Warga dari **ArrayList** menjadi **Array biasa**. Terdapat revisi pada template soal.
2. Memperbaiki keterangan perintah **ADD DOKTER X P R** dan **ADD PASIEN X P**.
3. Menambahkan keterangan cara *handling* batasan nilai happiness agar tidak ambigu.

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.

[KodeAsdos]_[Kelas]_[NPM]_[NamaLengkap]_Lab06.zip

Contoh:

ORI_A_1234567890_AhmadHaroriZakiIchsan_Lab06.zip