

# Dasar-Dasar Pemrograman 2

## Tugas Pemrograman 3 Corona



FAKULTAS  
ILMU  
KOMPUTER

---

## Corona



<https://www.instagram.com/p/B-MQBZXp6eM/?igshid=5aslbfpbp73c>

Saat ini, Indonesia sedang dilanda pandemi virus corona. Pandemi yang merugikan kita semua ini membuat masyarakat saling tolong menolong mulai dari memberikan donasi uang, memberikan konsumsi dan alat perlindungan diri (APD) untuk petugas medis, hingga membuat *platform* yang dapat membantu kita melawan pandemi virus corona ini seperti [SIGAP](#) dan [EndCorona](#).

Kamu yang merupakan mahasiswa DDP 2 tentu tidak ingin ketinggalan kesempatan untuk berkontribusi. Kamu ingin membuat sebuah program yang dapat melacak penyebaran virus corona pada suatu wilayah. Program ini menerima masukan berupa data-data objek apa saja yang mampu menjadi carrier dari virus corona pada wilayah tersebut beserta rincian kegiatan yang dilakukan oleh objek. Dalam menjalankan program ini, input dapat merupakan sebuah file dengan ekstensi .txt ataupun input secara manual melalui terminal begitu pula dengan outputnya.

**\*\*Terdapat revisi pada bagian yang di highlight kuning**

## Spesifikasi Program

Terdapat dua jenis carrier utama virus corona, yaitu Benda serta Manusia. Manusia dikelompokkan menjadi 5 kategori sesuai dengan pekerjaannya yaitu Ojol, Pekerja Jasa, Jurnalis, Petugas Medis, serta Cleaning Service. Benda dikelompokkan menjadi 4 jenis Benda yang memiliki kemungkinan menularkan virus corona yang tertinggi yaitu Pegangan Tangga, Angkutan Umum, Tombol Lift, serta Pintu. UML Class Diagram untuk memodelkan program tersebut dapat diakses di [bit.ly/UML-TP3](https://bit.ly/UML-TP3)

Ada beberapa hal yang perlu diperhatikan mengenai *class-class* yang ada pada program tersebut, antara lain:

- Atribut ‘totalKasusDisebabkan’ pada *class* Carrier menyimpan jumlah carrier, baik yang masih terinfeksi virus corona **maupun sudah sembuh** yang disebabkan Carrier tersebut, baik secara langsung maupun tidak langsung.
- Atribut ‘aktifKasusDisebabkan’ pada *class* Carrier menyimpan jumlah carrier yang masih terinfeksi virus corona yang disebabkan Carrier tersebut baik secara langsung maupun tidak langsung.
- Untuk kedua atribut ‘totalKasusDisebabkan’ dan ‘aktifKasusDisebabkan’ pada *class* Carrier, definisi **secara langsung** adalah apabila objek A merupakan objek positif corona dan berinteraksi dengan B sehingga objek B menjadi positif corona, artinya objek B merupakan carrier yang terinfeksi secara langsung oleh objek A.
- Untuk kedua atribut ‘totalKasusDisebabkan’ dan ‘aktifKasusDisebabkan’ pada *class* Carrier, definisi **secara tidak langsung** adalah apabila objek A merupakan objek positif corona dan berinteraksi dengan B sehingga B menjadi positif corona, lalu objek B berinteraksi dengan objek C sehingga objek C menjadi positif corona, artinya objek C merupakan carrier yang terinfeksi secara tidak langsung oleh objek A.

- Atribut ‘rantaiPenular’ pada *class* Carrier merupakan sebuah kontainer yang berisikan objek apa saja yang membuat objek tersebut tertular. Atribut ini digunakan untuk menyelesaikan query “RANTAI” nantinya.

Hint : Apabila mengambil contoh kasus sebelumnya, isi dari atribut ‘rantaiPenular’ pada objek C ialah [A,B]. Isi dari kontainer tersebut diurutkan dari yang paling awal menularkan hingga yang terakhir.

- *Class* Petugas Medis memiliki *method* tambahan yaitu *method* obati(Manusia manusia). *Method* ini mengubah objek dari *class* Manusia yang ada pada parameter yang awalnya positif terkena virus Corona menjadi negatif.
- Untuk *class* Benda, tidak dapat berinteraksi dengan *class* Benda lainnya.
- Untuk seluruh *subclass* dari *class* Benda, apabila mereka berinteraksi dengan objek dari *class* Manusia yang positif corona tidak langsung berubah statusnya menjadi positif corona (tidak seperti seluruh *subclass* dari *class* Manusia). Setiap benda memiliki persentase menularnya masing-masing yang memiliki nilai *default* 0. Untuk Pintu, persentase tersebut bertambah sebesar 30 setiap kali berinteraksi dengan objek dari *class* Manusia yang memiliki status positif terkena virus corona. Untuk Angkutan Umum, persentase bertambah sebanyak 35. Untuk Pegangan Tangga dan Tombol Lift, persentase bertambah sebanyak 20. Suatu benda dikatakan positif virus corona apabila persentasenya telah berjumlah  $\geq 100$ .
- *Class* Cleaning Service memiliki *method* tambahan yaitu *method* bersihkan(Benda benda). *Method* ini mengubah objek dari *class* Benda yang ada pada parameter yang awalnya positif terkena virus Corona menjadi negatif. Apabila seorang Cleaning Service membersihkan sebuah benda yang memiliki persentase menular sebesar  $0 \leq x < 100$  maka persentase menular benda tersebut di-reset kembali menjadi 0, dan apabila persentase menular bernilai 100 atau lebih (Benda tersebut memiliki statusCovid positif), maka statusCovid akan diubah menjadi negatif dan persentase menular menjadi 0 kembali.
- Hint: Setiap *subclass* dari carrier dapat memiliki implementasi *method* toString()-nya masing-masing. *Method* tersebut mengembalikan String dengan format:

“[NAMA\_CLASS] [NAMA\_OBJEK]”

Dengan method `toString` ini yang akan memudahkan untuk keseragaman output dari object `Carrier` sesuai dengan contoh output.

- Nama dari seluruh objek dari *class* `Carrier` dipastikan berbeda satu sama lain dan **tidak** bersifat *case-sensitive*.
- Anda diperkenankan menambah method pada template yang telah diberikan sesuai dengan kebutuhan anda. Tetapi dilarang mengubah access modifier pada method dan atribut dari template yang sudah diberikan

Selain struktur dan implementasi dari *class-class* yang ada pada UML Class Diagram, ada 12 *query* pada *class* `InputOutput` yang harus kamu implementasikan.

Sebelum menjalankan *query* dibawah, program akan meminta input apakah inputan akan berasal dari *command prompt / terminal* atau dari *file*. Jika ke *file*, maka program akan meminta path *file txt* yang akan dibaca. Program juga akan meminta apakah output-nya ke *command prompt / terminal* atau ke suatu *file* dan ditulis path-nya, berikut adalah contohnya:

```
Selamat datang di program simulasi COVID-19!
Silahkan masukkan metode input yang anda inginkan (TEXT/TERMINAL):
terminal
Silahkan masukkan metode output yang anda inginkan (TEXT/TERMINAL):
terminal
█
```

**Notes:** ketika membaca ataupun menulis ke suatu *file*, jika kamu tidak menuliskan *absolute path*, maka *path* dari *file* tersebut akan relatif terhadap direktori **src/main/java**. Jadi, jika kamu menulis `InputTC1.txt`, program akan mencoba membaca **src/main/java/InputTC1.txt**. Jika menggunakan Gradle, *path* dari *file* tersebut relatif terhadap direktori **assignment3** (yang di luar **src** dan di dalamnya terdapat direktori **testcases**).

**Jika tidak yakin, kamu bisa menggunakan *absolute path*, contoh:**

**C:\Users\username\Downloads\berkasInputku.txt**

**Untuk lebih detailnya, silakan baca `README.md` pada direktori `assignment3`.**

1. **ADD [TIPE\_CARRIER] [NAMA\_OBJEK]**

*Query* ini membuat sebuah objek Carrier yang memiliki tipe yang sesuai dan nama yang sesuai pada input. Objek yang telah dibuat disimpan dalam List listCarrier yang terdapat pada *class* World. Dipastikan tidak ada objek carrier dengan nama “NAMA\_OBJEK” pada listCarrier sebelumnya.

2. **INTERAKSI [OBJEK\_1] [OBJEK\_2]**

*Query* ini membuat adanya interaksi antara kedua objek tersebut. Interaksi berjalan secara dua arah, artinya OBJEK\_1 berinteraksi dengan OBJEK\_2 dan sebaliknya. Perlu diingat, apabila OBJEK\_1 dan OBJEK\_2 keduanya merupakan objek dari subclass benda, maka tidak akan terjadi apa apa. Dipastikan Objek\_1 dan Objek\_2 telah dilakukan query ADD sebelumnya.

3. **POSITIFKAN [OBJEK\_1]**

*Query* ini mengubah status dari Objek\_1 menjadi positif terkena virus corona. *Query* ini berfungsi untuk menandakan objek apa yang tertular virus corona untuk pertama kalinya di dalam suatu wilayah. Objek\_1 dipastikan selalu merupakan seorang Manusia. Dipastikan Objek\_1 telah dilakukan query ADD sebelumnya.

4. **SEMBUHKAN [OBJEK\_1] [OBJEK\_2]**

*Query* ini berfungsi agar Objek\_1 mengobati Objek\_2. Objek\_1 pasti merupakan seorang Petugas Medis dan Objek\_2 pasti merupakan seorang Manusia yang berstatus positif terinfeksi virus corona. Dipastikan Objek\_1 dan Objek\_2 telah dilakukan query ADD sebelumnya dan Objek\_1 dipastikan merupakan nama dari objek bertipe PetugasMedis, dan dipastikan Objek\_2 merupakan objek Manusia ber statusCovid positif

5. **BERSIHKAN [OBJEK\_1] [OBJEK\_2]**

*Query* ini berfungsi agar Objek\_1 membersihkan Objek\_2. Objek\_1 pasti merupakan seorang Cleaning Service dan Objek\_2 pasti merupakan sebuah Benda. Dipastikan Objek\_1 dan Objek\_2 telah dilakukan query ADD

sebelumnya dan Objek\_1 dipastikan merupakan nama dari objek bertipe CleaningService

#### 6. RANTAI [OBJEK\_1]

Apabila Objek\_1 memiliki status covid negatif, maka *query* ini akan menghasilkan **BelumTertularException** dengan message sesuai contoh output. Jika berhasil, *Query* ini mencetak riwayat penularan hingga Objek\_1 menjadi tertular dengan format yang tertera pada contoh *output*. Dipastikan Objek\_1 dan Objek\_2 telah dilakukan query ADD sebelumnya. Dipastikan Objek\_1 telah dilakukan query ADD sebelumnya.

Hint : Manfaatkan atribut rantaiPenular yang terdapat pada class Carrier.

#### 7. TOTAL\_KASUS\_DARI\_OBJEK [OBJEK\_1]

*Query* ini mencetak jumlah seluruh kasus corona yang diakibatkan oleh Objek\_1, baik yang masih positif ataupun yang sudah sembuh sesuai dengan format yang tertera pada contoh *output*. Dipastikan Objek\_1 telah dilakukan query ADD sebelumnya.

#### 8. AKTIF\_KASUS\_DARI\_OBJEK [OBJEK\_1]

*Query* ini mencetak jumlah seluruh kasus corona yang diakibatkan oleh Objek\_1 dan masih positif, sesuai dengan format yang tertera pada contoh *output*. Dipastikan Objek\_1 telah dilakukan query ADD sebelumnya.

#### 9. TOTAL\_SEMBUH\_MANUSIA

*Query* ini mencetak jumlah manusia yang pernah terinfeksi virus corona lalu sembuh sesuai dengan format yang tertera pada contoh *output*.

#### 10. TOTAL\_SEMBUH\_PETUGAS\_MEDIS [OBJEK\_1]

*Query* ini mencetak jumlah manusia yang berhasil disembuhkan oleh Objek\_1 (yang pasti merupakan seorang Petugas Medis) sesuai dengan format yang tertera pada contoh *output*. Dipastikan Objek\_1 telah dilakukan query ADD sebelumnya dan merupakan nama dari objek bertipe PetugasMedis

## 11. TOTAL\_BERSIH\_CLEANING\_SERVICE [OBJEK\_1]

*Query* ini mencetak jumlah benda yang berhasil dibersihkan oleh Objek\_1 (yang pasti merupakan seorang Cleaning Service) sesuai dengan format yang tertera pada contoh *output*. Dipastikan Objek\_1 telah dilakukan query ADD sebelumnya dan merupakan nama dari objek bertipe CleaningService

## 12. EXIT

*Query* ini berfungsi untuk menghentikan alur program.

### PENJELASAN OUTPUT

Contoh input dan output dapat diakses pada folder testcases.

#### Contoh input 1:

```
ADD JURNALIS Krysta
ADD OJOL Brandies
ADD OJOL Joko
ADD OJOL Sueb
ADD PEKERJA_JASA Sharleen
ADD PEKERJA_JASA Rabbi
POSITIFKAN Joko
INTERAKSI Joko Rabbi
INTERAKSI Rabbi Sharleen
INTERAKSI Sharleen Sueb
INTERAKSI Rabbi Brandies
INTERAKSI Brandies Krysta
RANTAI Joko
RANTAI Sharleen
TOTAL_KASUS_DARI_OBJEK Rabbi
EXIT
```

Penjelasan :

- Diawal, terdapat input untuk memasukkan objek carrier pada listCarrier pada class World.java :

```
ADD JURNALIS Krysta
ADD OJOL Brandies
```

ADD OJOL Joko  
ADD OJOL Sueb  
ADD PEKERJA\_JASA Sharleen  
ADD PEKERJA\_JASA Rabbi

- Kemudian query selanjutnya adalah

POSITIFKAN Joko

Objek bernama Joko menjadi memiliki statusCovid positif.

- Kemudian query selanjutnya adalah

INTERAKSI Joko Rabbi

Karena Joko memiliki statusCovid positif, maka ketika Rabbi melakukan interaksi dengan Joko, statusCovid Rabbi menjadi positif.

- Kemudian query selanjutnya adalah

INTERAKSI Rabbi Sharleen

Karena Rabbi memiliki statusCovid positif, maka ketika Sharleen melakukan interaksi dengan Rabbi, statusCovid Sharleen menjadi positif.

- Kemudian query selanjutnya adalah

INTERAKSI Sharleen Sueb

Karena Sharleen memiliki statusCovid positif, maka ketika Sueb melakukan interaksi dengan Sharleen, statusCovid Sueb menjadi positif.

- Kemudian query selanjutnya adalah

INTERAKSI Rabbi Brandies

Karena Rabbi memiliki statusCovid positif, maka ketika Brandies melakukan interaksi dengan Rabbi, statusCovid Brandies menjadi positif.

- Kemudian query selanjutnya adalah

INTERAKSI Brandies Krysta

Karena Brandies memiliki statusCovid positif, maka ketika Krysta melakukan interaksi dengan Brandies, statusCovid Krysta menjadi positif.

- Kemudian query selanjutnya adalah

RANTAI Joko

Karena Joko merupakan objek yang pertama positif corona (tidak ditularkan dari siapapun). Maka, keluaran query ini akan sebagai berikut :

Rantai penyebaran OJOL Joko: OJOL Joko

- Kemudian query selanjutnya adalah



RANTAI Sharleen

Riwayat penularan yang mengakibatkan Sharleen tertular adalah : Joko menularkan kepada Rabbi, lalu Rabbi menularkan kepada Sharleen. Maka, keluaran query ini akan sebagai berikut :

Rantai penyebaran PEKERJA JASA Sharleen: OJOL Joko -> PEKERJA JASA Rabbi -> PEKERJA JASA Sharleen

Keterangan : Keluaran query rantai ini, dikeluarkan dalam satu baris, tidak ada enter. (Dapat dilihat di contoh output pada folder testcases)

- Kemudian query selanjutnya adalah

TOTAL\_KASUS\_DARI\_OBJEK Rabbi

Orang orang yang menjadi positif dikarenakan Rabbi adalah :

- Sharleen, Sharleen menjadi positif karena tertular langsung melalui interaksi dengan Rabbi
- Sueb, Sueb menjadi positif karena tertular oleh Sharleen, dan Sharleen tertular akibat Rabbi
- Brandies, Brandies menjadi positif karena tertular langsung melalui interaksi dengan Rabbi
- Krysta, Krysta menjadi positif karena tertular oleh Brandies, dan Brandies tertular akibat Rabbi

Total orang yang tertular akibat Rabbi ada sebanyak 4, maka keluaran query ini akan sebagai berikut :

PEKERJA JASA Rabbi telah menyebarkan virus COVID ke 4 objek

- Kemudian terdapat query EXIT, program selesai.

### Contoh Input 2:

ADD JURNALIS Ros  
ADD OJOL Upin  
ADD OJOL Ipin  
ADD PEKERJA\_JASA Ihsan  
ADD PETUGAS\_MEDIS Jarjit  
POSITIFKAN Ipin  
INTERAKSI Upin Ipin  
INTERAKSI Ros Upin  
RANTAI Ros  
TOTAL\_KASUS\_DARI\_OBJEK Ipin

```

AKTIF_KASUS_DARI_OBJEK Ipin
SEMBUHKAN Jarjit Upin
TOTAL_SEMBUH_PETUGAS_MEDIS Jarjit
TOTAL_KASUS_DARI_OBJEK Ipin
AKTIF_KASUS_DARI_OBJEK Ipin
TOTAL_SEMBUH_MANUSIA
INTERAKSI Ros Upin
INTERAKSI Upin Ihsan
Rantai Ihsan
TOTAL_KASUS_DARI_OBJEK Upin
AKTIF_KASUS_DARI_OBJEK Upin
EXIT

```

Penjelasan:

- Di awal, terdapat input untuk memasukkan objek carrier pada listCarrier pada class World.java :

```

ADD JURNALIS Ros
ADD OJOL Upin
ADD OJOL Ipin
ADD PEKERJA_JASA Ihsan
ADD PETUGAS_MEDIS Jarjit

```

- Kemudian query selanjutnya adalah

```
POSITIFKAN Ipin
```

Objek bernama Ipin menjadi memiliki statusCovid positif.

- Kemudian query selanjutnya adalah

```
INTERAKSI Upin Ipin
```

Karena Ipin memiliki statusCovid positif, maka ketika Upin melakukan interaksi dengan Ipin, statusCovid Upin menjadi positif.

- Kemudian query selanjutnya adalah

```
INTERAKSI Ros Upin
```

Karena Upin memiliki statusCovid positif, maka ketika Ros melakukan interaksi dengan Upin, statusCovid Ros menjadi positif.

- Kemudian query selanjutnya adalah

```
RANTAI Ros
```

Riwayat penularan yang mengakibatkan Ros tertular adalah : Ipin menularkan

kepada Upin, lalu Upin menularkan kepada Ros.  
Maka, keluaran query ini akan sebagai berikut :

Rantai penyebaran JURNALIS Ros: OJOL Ipin -> OJOL Upin -> JURNALIS Ros

- Kemudian query selanjutnya adalah

TOTAL\_KASUS\_DARI\_OBJEK Ipin

Orang orang yang menjadi positif dikarenakan Ipin adalah :

- Upin, Upin menjadi positif karena tertular langsung melalui interaksi dengan Ipin
- Ros, Ros menjadi positif karena tertular oleh Upin, dan Upin tertular akibat Ipin

Total orang yang tertular akibat Rabbi ada sebanyak 2, maka keluaran query ini akan sebagai berikut :

OJOL Ipin telah menyebarkan virus COVID ke 2 objek

- Kemudian query selanjutnya adalah

AKTIF\_KASUS\_DARI\_OBJEK Ipin

Orang orang yang disebutkan dalam query sebelumnya yang positif diakibatkan oleh Ipin, kesemuanya masih positif, sehingga banyaknya kasus aktif yang disebabkan oleh Ipin sama dengan banyaknya kasus total yang disebabkan oleh Ipin.

Maka keluaran query ini akan sebagai berikut :

OJOL Ipin telah menyebarkan virus COVID dan masih teridentifikasi positif sebanyak 2 objek

- Kemudian query selanjutnya adalah

SEMBUHKAN Jarjit Upin

Jarjit merupakan objek bertipe petugas medis, dan Upin memiliki statusCovid positif, maka setelah query ini, Upin akan disembuhkan oleh jarjit sehingga statusCovid dari Upin menjadi Negatif.

- Kemudian query selanjutnya adalah

TOTAL\_SEMBUH\_PETUGAS\_MEDIS Jarjit

Query ini akan meminta ada berapa banyak objek Manusia yang telah disembuhkan oleh petugas medis Jarjit.

Jarjit telah menyembuhkan 1 objek Manusia yaitu Upin. Maka, keluaran dari

query ini akan sebagai berikut :

PETUGAS MEDIS Jarjit menyembuhkan 1 manusia

- Kemudian query selanjutnya adalah

TOTAL\_KASUS\_DARI\_OBJEK Ipin

Orang orang yang menjadi positif dikarenakan Ipin masih sama dengan query Total Kasus sebelumnya. Maka, keluaran query ini akan sebagai berikut :

OJOL Ipin telah menyebarkan virus COVID ke 2 objek

- Kemudian query selanjutnya adalah

AKTIF\_KASUS\_DARI\_OBJEK Ipin

Dari 2 kasus covid yang disebabkan oleh Ipin, terdapat 1 objek, yaitu objek Upin yang telah sembuh dari covid, maka keluaran query ini akan sebagai berikut:

OJOL Ipin telah menyebarkan virus COVID dan masih teridentifikasi positif sebanyak 1 objek

- Kemudian query selanjutnya adalah

TOTAL\_SEMBUH\_MANUSIA

Query ini akan mengeluarkan banyaknya total kasus covid keseluruhan yang telah disembuhkan.

Terdapat 1 objek yang telah disembuhkan, yaitu Upin. Maka keluaran query ini akan sebagai berikut :

Total sembuh dari kasus COVID yang menimpa manusia ada sebanyak: 1 kasus

- Kemudian query selanjutnya adalah

INTERAKSI Ros Upin

Karena Ros memiliki statusCovid positif, maka ketika Ros melakukan interaksi dengan Upin, statusCovid Upin menjadi positif.

Perhatikan bahwa, disini objek yang telah sembuh dari covid tetap dapat tertular kembali.

- Kemudian query selanjutnya adalah

INTERAKSI Upin Ihsan

Karena Upin memiliki statusCovid positif, maka ketika Ihsan melakukan interaksi dengan Upin, statusCovid Ihsan menjadi positif.

- Kemudian query selanjutnya adalah

RANTAI Ihsan

Riwayat penularan yang mengakibatkan Sharleen tertular adalah : Ipin menularkan kepada Upin, lalu Upin menularkan kepada Ros, kemudian Ros menularkan kepada Upin, lalu Upin menularkan kepada Ihsan.

Maka, keluaran query ini akan sebagai berikut :

Rantai penyebaran PEKERJA JASA Ihsan: OJOL Ipin -> OJOL Upin -> JURNALIS Ros -> OJOL Upin -> PEKERJA JASA Ihsan

Perhatikan bahwa, Ojol Upin dicetak 2 kali. Hal ini karena memang riwayat penularannya seperti yang telah dijabarkan tadi.

- Kemudian query selanjutnya adalah

TOTAL\_KASUS\_DARI\_OBJEK Upin

Orang-orang yang menjadi positif dikarenakan Upin adalah :

- Ros, Ros menjadi positif karena tertular langsung melalui interaksi dengan Upin saat Upin positif pertama kali.
- Upin yang tertular untuk kedua kalinya, Upin menjadi positif karena tertular oleh Ros, dan Ros tertular akibat Upin.
- Ihsan, Ihsan tertular akibat berinteraksi langsung dengan Upin.

Total objek yang tertular akibat Upin ada sebanyak 3, maka keluaran query ini akan sebagai berikut :

OJOL Upin telah menyebarkan virus COVID ke 3 objek

- Kemudian query selanjutnya adalah

AKTIF\_KASUS\_DARI\_OBJEK Upin

Orang-orang yang disebutkan dalam query sebelumnya yang positif diakibatkan oleh Upin, kesemuanya masih positif, sehingga banyaknya kasus aktif yang disebabkan oleh Upin sama dengan banyaknya kasus total yang disebabkan oleh Upin.

Maka keluaran query ini akan sebagai berikut :

OJOL Upin telah menyebarkan virus COVID dan masih teridentifikasi positif sebanyak 3 objek

- Kemudian terdapat query EXIT, program selesai.

### Contoh input 3:

```
ADD PEKERJA_JASA Borman
ADD PEKERJA_JASA Sasari
ADD PEKERJA_JASA Norman
ADD PEKERJA_JASA Elang
ADD CLEANING_SERVICE DekDepe
ADD TOMBOL_LIFT LiftTimur
ADD PEGANGAN_TANGGA TanggaBarat
POSITIFKAN Borman
INTERAKSI Borman LiftTimur
INTERAKSI Sasari LiftTimur
INTERAKSI Borman Norman
INTERAKSI Norman LiftTimur
INTERAKSI Borman LiftTimur
INTERAKSI Sasari LiftTimur
Rantai Sasari
INTERAKSI Norman LiftTimur
INTERAKSI Borman LiftTimur
INTERAKSI Sasari LiftTimur
Rantai Sasari
BERSIHKAN DekDepe LiftTimur
INTERAKSI Elang LiftTimur
Rantai Elang
INTERAKSI Sasari TanggaBarat
INTERAKSI Norman TanggaBarat
INTERAKSI Borman TanggaBarat
BERSIHKAN DekDepe TanggaBarat
INTERAKSI Sasari TanggaBarat
INTERAKSI Norman TanggaBarat
INTERAKSI Borman TanggaBarat
INTERAKSI Elang TanggaBarat
Rantai Elang
TOTAL_BERSIH_CLEANING_SERVICE DekDepe
EXIT
```

Penjelasan :

- Diawal, terdapat input untuk memasukkan objek carrier pada listCarrier pada class World.java :

```
ADD PEKERJA_JASA Borman
ADD PEKERJA_JASA Sasari
ADD PEKERJA_JASA Norman
ADD PEKERJA_JASA Elang
```

ADD CLEANING\_SERVICE DekDepe  
ADD TOMBOL\_LIFT LiftTimur  
ADD PEGANGAN\_TANGGA TanggaBarat

- Kemudian query selanjutnya adalah

POSITIFKAN Borman

Objek bernama Borman menjadi memiliki statusCovid positif.

- Kemudian query selanjutnya adalah

INTERAKSI Borman LiftTimur

Karena Borman memiliki statusCovid positif, maka ketika LiftTimur melakukan interaksi dengan Borman, persentase menular dari LiftTimur akan bertambah 20 (karena LiftTimur bertipe TombolLift), yang awalnya 0 menjadi 20.

- Kemudian query selanjutnya adalah

INTERAKSI Sasari LiftTimur

Karena Sasari memiliki statusCovid negatif, maka ketika Sasari melakukan interaksi dengan LiftTimur, persentase menular dari LiftTimur akan tetap 20 dan Sasari akan tetap memiliki statusCovid Negatif.

- Kemudian query selanjutnya adalah

INTERAKSI Borman Norman

Karena Borman memiliki statusCovid positif, maka ketika Norman melakukan interaksi dengan Borman, statusCovid Norman menjadi positif.

- Kemudian query selanjutnya adalah

INTERAKSI Norman LiftTimur

Karena Norman memiliki statusCovid positif, maka ketika LiftTimur melakukan interaksi dengan Borman, persentase menular dari LiftTimur akan bertambah 20 (karena LiftTimur bertipe TombolLift), yang awalnya 20 menjadi 40.

- Kemudian query selanjutnya adalah

INTERAKSI Borman LiftTimur

Karena Borman memiliki statusCovid positif, maka ketika LiftTimur melakukan interaksi dengan Borman, persentase menular dari LiftTimur akan bertambah 20 (karena LiftTimur bertipe TombolLift), yang awalnya 40 menjadi 60.

- Kemudian query selanjutnya adalah

INTERAKSI Sasari LiftTimur

Karena Sasari memiliki statusCovid negatif, maka ketika Sasari melakukan interaksi dengan LiftTimur, persentase menular dari LiftTimur akan tetap 60 dan Sasari akan tetap memiliki statusCovid Negatif.

- Kemudian query selanjutnya adalah

Rantai Sasari

Karena Sasari memiliki statusCovid negatif, maka query Rantai ini akan mendapatkan **BelumTertularException** . Dan message dari exception akan dicetak. Maka, keluaran program akan sebagai berikut:

assignments.assignment3.BelumTertularException: PEKERJA JASA Sasari berstatus negatif

- Kemudian query selanjutnya adalah

INTERAKSI Norman LiftTimur

Karena Norman memiliki statusCovid positif, maka ketika LiftTimur melakukan interaksi dengan Norman, persentase menular dari LiftTimur akan bertambah 20 (karena LiftTimur bertipe TombolLift), yang awalnya 60 menjadi 80.

- Kemudian query selanjutnya adalah

INTERAKSI Borman LiftTimur

Karena Borman memiliki statusCovid positif, maka ketika LiftTimur melakukan interaksi dengan Borman, persentase menular dari LiftTimur akan bertambah 20 (karena LiftTimur bertipe TombolLift), yang awalnya 80 menjadi 100. Karena persentase menular dari LiftTimur telah mencapai 100, maka statusCovid dari LiftTimur akan menjadi positif.

Perhatikan bahwa riwayat penularan dari LiftTimur akan dicatat bahwa LiftTimur ditulari oleh Borman, karena Borman yang membuat LiftTimur berubah status dari negatif ke positif.

- Kemudian query selanjutnya adalah

INTERAKSI Sasari LiftTimur

Karena LiftTimur memiliki statusCovid positif, maka ketika Sasari melakukan interaksi dengan LiftTimur, statusCovid Sasari menjadi positif.

- Kemudian query selanjutnya adalah

Rantai Sasari

Riwayat penularan yang mengakibatkan Sasari tertular adalah : Borman mengakibatkan LiftTimur menjadi positif, lalu LiftTimur melakukan interaksi dengan Sasari sehingga Sasari memiliki tertular. Sehingga, keluaran dari query ini adalah sebagai berikut:

Rantai penyebaran PEKERJA JASA Sasari: PEKERJA JASA Borman ->



TOMBOL LIFT LiftTimur -> PEKERJA JASA Sasari

- Kemudian query selanjutnya adalah

BERSIHKAN DekDepe LiftTimur

LiftTimur yang memiliki statusCovid Positif, telah dibersihkan oleh DekDepe, sehingga statusCovid dari LiftTimur akan menjadi negatif dan persentase menular dari LiftTimur akan kembali lagi menjadi 0.

- Kemudian query selanjutnya adalah

INTERAKSI Elang LiftTimur

Karena Elang memiliki statusCovid negatif, maka ketika Elang melakukan interaksi dengan LiftTimur, persentase menular dari LiftTimur akan tetap 0 dan Elang akan tetap memiliki statusCovid Negatif.

- Kemudian query selanjutnya adalah

Rantai Elang

Karena Elang memiliki statusCovid negatif, maka query Rantai ini akan mendapatkan **BelumTertularException** . Dan message dari exception akan dicetak. Maka, keluaran program akan sebagai berikut:

assignments.assignment3.BelumTertularException: PEKERJA JASA Elang berstatus negatif

- Kemudian query selanjutnya adalah

INTERAKSI Sasari TanggaBarat

Karena Sasari memiliki statusCovid positif, maka ketika TanggaBarat melakukan interaksi dengan Sasari, persentase menular dari TanggaBarat akan bertambah 20 (karena TanggaBarat bertipe PeganganTangga), yang awalnya 0 menjadi 20.

- Kemudian query selanjutnya adalah

INTERAKSI Norman TanggaBarat

Karena Norman memiliki statusCovid positif, maka ketika TanggaBarat melakukan interaksi dengan Norman, persentase menular dari TanggaBarat akan bertambah 20 (karena TanggaBarat bertipe PeganganTangga), yang awalnya 20 menjadi 40.

- Kemudian query selanjutnya adalah

INTERAKSI Borman TanggaBarat

Karena Borman memiliki statusCovid positif, maka ketika TanggaBarat melakukan interaksi dengan Borman, persentase menular dari TanggaBarat akan

bertambah 20 (karena TanggaBarat bertipe PeganganTangga), yang awalnya 40 menjadi 60.

- Kemudian query selanjutnya adalah

BERSIHKAN DekDepe TanggaBarat

DekDepe membersihkan TanggaBarat, persentase menular dari TanggaBarat menjadi 0 lagi.

- Kemudian query selanjutnya adalah

INTERAKSI Sasari TanggaBarat

Karena Sasari memiliki statusCovid positif, maka ketika TanggaBarat melakukan interaksi dengan Sasari, persentase menular dari TanggaBarat akan bertambah 20 (karena TanggaBarat bertipe PeganganTangga), yang awalnya 0 menjadi 20.

- Kemudian query selanjutnya adalah

INTERAKSI Norman TanggaBarat

Karena Norman memiliki statusCovid positif, maka ketika TanggaBarat melakukan interaksi dengan Norman, persentase menular dari TanggaBarat akan bertambah 20 (karena TanggaBarat bertipe PeganganTangga), yang awalnya 20 menjadi 40.

- Kemudian query selanjutnya adalah

INTERAKSI Borman TanggaBarat

Karena Borman memiliki statusCovid positif, maka ketika TanggaBarat melakukan interaksi dengan Borman, persentase menular dari TanggaBarat akan bertambah 20 (karena TanggaBarat bertipe PeganganTangga), yang awalnya 40 menjadi 60.

- Kemudian query selanjutnya adalah

INTERAKSI Elang TanggaBarat

Karena Elang memiliki statusCovid negatif, maka ketika Elang melakukan interaksi dengan TanggaBarat, persentase menular dari TanggaBarat akan tetap 0 dan Elang akan tetap memiliki statusCovid Negatif.

- Kemudian query selanjutnya adalah

Rantai Elang

Karena Elang memiliki statusCovid negatif, maka query Rantai ini akan mendapatkan **BelumTertularException** . Dan message dari exception akan dicetak. Maka, keluaran program akan sebagai berikut:

assignments.assignment3.BelumTertularException: PEKERJA JASA Elang

berstatus negatif

- Kemudian query selanjutnya adalah

TOTAL\_BERSIH\_CLEANING\_SERVICE DekDepe

Total dari query bersihkan yang telah dilakukan DekDepe adalah sebanyak 2 kali, yaitu membersihkan LiftTimur dan TanggaBarat. Maka output dari query ini adalah sebagai berikut:

CLEANING SERVICE DekDepe membersihkan 2 benda

- Kemudian terdapat query EXIT, dan program selesai

## Komponen penilaian utama

- 10% Implementasi *interface* Status dan *concrete class*-nya
- 30% Implementasi *class* Carrier
- 20% Implementasi *class* InputOutput
- 10% Implementasi *concrete class* dari Carrier
- 20% Demonstrasi program
- 10% Kerapian dan dokumentasi

## Komponen penilaian bonus

- +10% implementasi bonus

**Bonus: Implementasikan GUI untuk program ini.**

Tidak ada ketentuan khusus mengenai bentuk GUI program. Silakan berkreasi dalam implementasi *layout*, *widgets*, dsb. Usahakan untuk memanfaatkan ulang (*reuse*) kode yang sudah ada sebaik mungkin agar kamu tidak perlu mengimplementasikan ulang fitur-fitur yang ada. Kamu boleh mengimplementasikan GUI hanya untuk sebagian fitur saja (misalnya hanya seleksi antara output TEXT/TERMINAL), tetapi nilai sempurna hanya akan diberikan jika tiap *query* dapat dilakukan melalui GUI.

**Pastikan program asli (tanpa GUI) tetap bekerja sebagaimana mestinya dan tetap lulus tes.**

Usahakan agar program GUI terpisah dari program Simulator utama. Kamu boleh menambahkan opsi input **GUI** pada Simulator yang jika dipilih akan langsung memulai SimulatorGUI, atau jika tidak juga tidak apa-apa (jadi harus membuka SimulatorGUI jika ingin pakai GUI).

Namun, intinya di luar hal yang disebutkan di atas, program Simulator dan komponen-komponennya tidak perlu tahu mengenai adanya implementasi GUI. Usahakan agar implementasi GUI kamu "membungkus" program yang sudah ada. Program SimulatorGUI boleh tahu tentang isi program Simulator secara keseluruhan, tetapi tidak sebaliknya. Kamu bebas menambahkan *class* dan berkas-berkas yang diperlukan untuk implementasi GUI (tetapi hanya untuk keperluan GUI dan tidak dilibatkan pada kode Simulator biasa). Dengan begini, seharusnya akan lebih mudah untuk menjaga bahwa program asli kamu (tanpa GUI) tetap bekerja sebagaimana mestinya.

Berikut beberapa contoh ide GUI yang bisa diimplementasikan.

Misalnya untuk query **ADD**:

- Klik tombol Add
- Muncul dialog yang terdiri atas pilihan dropdown tipe Carrier dan input box untuk nama Carrier
- Jika diisi dan diklik OK, muncul konfirmasi bahwa Carrier dengan tipe dan nama tersebut berhasil ditambahkan

Kemudian untuk query **INTERAKSI**, bisa berupa salah satu dari ide berikut:

- Dibuat satu *input box* untuk memasukkan nama Objek1 dan Objek2 dipisah dengan spasi.
- Dibuat dua input box untuk nama Objek1 dan Objek2 secara terpisah.
- Diberikan dua dropdown untuk Objek1 dan Objek2, sehingga pengguna tidak perlu mengetik.

Ini hanya contoh, kamu bebas berkreasi sesuka kamu. Kalau ingin hanya ada satu tombol dan user langsung mengetik query dalam satu text box besar, ini juga bisa saja. Namun, hasil penilaiannya mungkin tidak akan sebaik jika kamu lakukan terpisah seperti contoh di atas (karena GUI nya kurang "terasa").