

# Dasar-Dasar Pemrograman 2

## Lab 06 Inheritance



FAKULTAS  
**ILMU**  
**KOMPUTER**

Pada tutorial/lab sebelumnya, kalian telah mengetahui dasar-dasar dari pemrograman berbasis object pada Java yang terdiri dari class, object, attribute, dan method. Kali ini, kalian akan mempelajari bagaimana cara memodelkan masalah dengan menggunakan inheritance.

### Inheritance

**Inheritance** dapat didefinisikan sebagai suatu proses di mana satu kelas memerlukan suatu properti (method atau atribut) dari kelas lainnya. Penggunaan inheritance sebuah informasi data dapat di-manage dalam struktur hierarchical. Suatu kelas yang meng-inherit properti dari kelas lain dikenal sebagai subclass dan kelas yang di-inherit dikenal sebagai superclass.

### Example of Inheritance

Mari kita coba mengimplementasikan inheritance pada program sederhana berikut ini :

```
↳ ProgramBangunDatar.java > BangunDatar
1  public class ProgramBangunDatar {
2      Run | Debug
3      public static void main(String[] args) {
4          Persegi persegi = new Persegi(3);
5          persegi.HitungLuas();
6          persegi.HitungKeliling();
7
7          PersegiPanjang persegiPanjang = new PersegiPanjang(4, 5);
8          persegiPanjang.HitungLuas();
9          persegiPanjang.HitungKeliling();
10     }
11 }
```

```

12
13 class BangunDatar {
14     float luas;
15     float keliling;
16
17     public void HitungLuas() {
18 }
19
20     public void HitungKeliling() {
21 }
22 }
23
24 class Persegi extends BangunDatar {
25     int x;
26
27     public Persegi(int x) {
28         this.x = x;
29     }
30
31     @Override
32     public void HitungLuas() {
33         this.luas = this.x * this.x;
34         System.out.println("Luas Persegi adalah : " + this.luas);
35     }
36
37     @Override
38     public void HitungKeliling() {
39         this.keliling = this.x * 4;
40         System.out.println("Keliling Persegi adalah : " + this.keliling);
41     }
42 }
43
44 class PersegiPanjang extends BangunDatar {
45     int x, y;
46
47     public PersegiPanjang(int x, int y) {
48         this.x = x;
49         this.y = y;
50     }
51
52     @Override
53     public void HitungLuas() {
54         this.luas = this.x * this.y;
55         System.out.println("Luas Persegi Panjang adalah : " + this.luas);
56     }
57
58     @Override
59     public void HitungKeliling() {
60         this.keliling = (this.x + this.y) * 2;
61         System.out.println("Keliling Persegi Panjang adalah : " + this.keliling);
62     }
63 }
```

Berikut contoh hasil kode diatas :

```
Luas Persegi adalah : 9.0
Keliling Persegi adalah : 12.0
Luas Persegi Panjang adalah : 20.0
Keliling Persegi Panjang adalah : 18.0
```

Perhatikan bahwa kelas *Persegi* dan *PersegiPanjang* (subclass) meng-inherit kelas *BangunDatar* (superclass), dan mengambil atribut *luas* serta *keliling* dan method *hitungKeliling()* serta *hitungLuas()* dari superclass-nya. Cara subclass meng-inherit yaitu dengan keyword *extends*. Kita juga dapat memanggil method superclass atau mengambil atribut superclass melalui suatu subclass dengan memakai keyword *super* yang akan dijelaskan pada bagian selanjutnya.

## Method Super

```
SubClass.java > ...
1  class SuperClass {
2      int age;
3
4      public SuperClass(int age) {
5          this.age = age;
6      }
7
8      public void getAge() {
9          System.out.println("Nilai dari 'age' pada SuperClass adalah : " + this.age);
10     }
11 }
12
13 public class SubClass extends SuperClass {
14
15     public SubClass(int age) {
16         super(age);
17     }
18
19     public static void main(String[] args) {
20         SubClass subClass = new SubClass(19);
21         subClass.getAge();
22     }
23 }
```

Perhatikan kembali bahwa method *super* pada constructor kelas *SubClass* digunakan untuk meng-construct kelas *SuperClass*. Ini sangat berguna ketika superclass mempunyai semua atribut yang ada di subclass sehingga kita hanya perlu memanggil method *super*. Dan terakhir, kita juga dapat mengecek apakah suatu kelas merupakan instance dari kelas yang lain dengan memakai keyword *instanceof*. Silahkan mengimplementasikan program sederhana berikut untuk lebih memahami fungsionalitas *instanceof*.

```
↳ ProgramAnimal.java > ...
1  public class ProgramAnimal {
2      Run | Debug
3      public static void main(String[] args) {
4          Animal cat = new Cat();
5          Animal bfly = new Butterfly();
6          System.out.println(cat instanceof Animal);
7          System.out.println(cat instanceof Insect);
8          System.out.println(cat instanceof Mammal);
9          System.out.println(bfly instanceof Animal);
10         System.out.println(bfly instanceof Insect);
11         System.out.println(bfly instanceof Mammal);
12     }
13
14     class Animal {}
15     class Insect extends Animal {}
16     class Mammal extends Animal {}
17     class Cat extends Mammal {}
18     class Butterfly extends Insect {}
```

Perhatikan bahwa output dari program akan menjadi seperti berikut:

```
true
false
true
true
true
false
```

Hal tersebut menunjukkan bahwa *cat* merupakan instance dari kelas *Animal* serta *Mammal*, dan *bfly* merupakan instance dari kelas *Animal* serta *Insect*. Keyword *instanceof* sangat berguna untuk mengetahui apakah suatu kelas meng-extend suatu kelas tertentu.

## Polymorphism?

**Polymorphism** dapat didefinisikan sebagai suatu konsep dimana suatu objek dapat memiliki banyak bentuk. Contohnya adalah  $H_2O$  dapat memiliki bentuk cair, padat, dan gas.

Pada OOP, **Polymorphism** adalah kemampuan suatu variabel atau argumen untuk merujuk kepada suatu objek dari bermacam-macam kelas [Meyer pp.224].

## Lebih Lanjut tentang Polymorphism

```
class A {  
    // ..  
    public void foo() {  
        System.out.println("foo A");  
    }  
  
    public static String cetak() {  
        return "Class A";  
    }  
}  
  
class B extends A {  
    // ..  
    public void foo() {  
        System.out.println("foo B");  
    }  
  
    public void bar() {  
        System.out.println("bar B");  
    }  
  
    public static String cetak() {  
        return "Class B";  
    }  
}
```

### Penjelasan

- Kelas A merupakan superclass dari kelas B.
- Kelas B merupakan subclass dari kelas A.

### Hasil simulasi kedua kelas tersebut adalah:

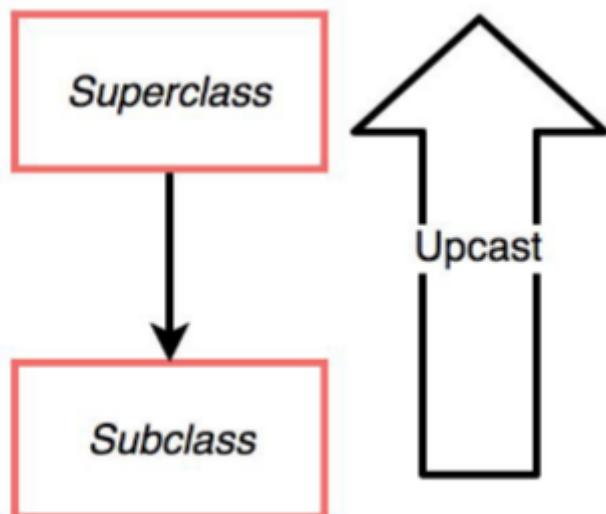
```
A a = new A();  
B b = new B();  
  
a.foo(); // foo A  
b.foo(); // foo B  
  
b.bar(); // bar B
```

Perhatikan bahwa *foo* terimplementasi pada class *A* serta *B*, sedangkan *bar* hanya pada class *B* saja. Maka dari itu hanya instance dari *B* yang dapat memanggil *bar*.

## Object Type Casting

Object type casting merupakan sebuah casting yang membuat suatu object dapat bertipe lain selain dari tipe asli object itu sendiri, tipe yang dapat digunakan antara lain adalah superclass atau interfacenya, namun pada kali ini hanya akan dibahas superclass nya saja. Ada 2 tipe casting, yaitu **Upcasting** dan **Downcasting**.

### Upcasting



```
A a = new A();
B b = new B();

a.foo(); // foo A

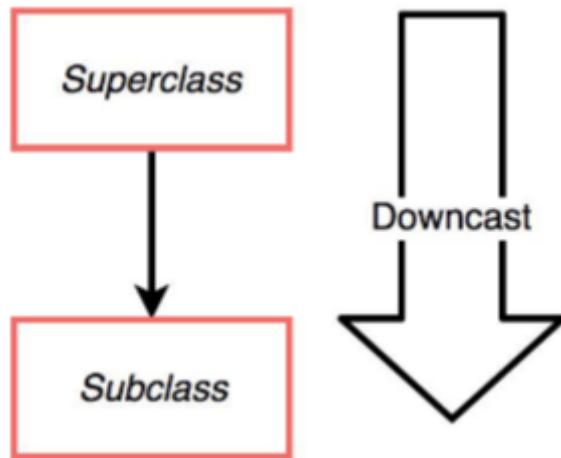
a = b; // upcasting

a.foo(); // foo B
```

Upcasting adalah melakukan proses pengubahan referensi subclass ke superclass. Jadi secara singkat kita dapat mendefinisikan object baru (subclass) ke sebuah variabel bertipe superclass, seperti contoh kita menyimpan *b* yang merupakan bertipe *B* ke dalam variabel *a* yang bertipe *A*.

**Perhatikan** bahwa pada kondisi diatas, kita tidak dapat memanggil *a.bar()* karena Kelas *A* tidak memiliki method tersebut. Dengan begitu kita dapat menyimpulkan bahwa method yang spesifik pada kelas *B* akan menjadi **inactive** .

## Downcasting



Downcasting adalah melakukan proses pengubahan referensi superclass ke subclass.

```
A a = new B();  
B b = (B) a; // Downcasting  
  
b.bar(); // bar B
```

Proses downcasting dari variabel a ke variabel bertipe Kelas B memberi kita akses untuk memanggil method bar() yang khusus pada Kelas B. **PERHATIKAN** bahwa kita bisa memanggil b.bar() namun tidak dapat memanggil a.bar() (lihat penjelasan pada **upcasting**).

## Static Binding

Static type binding ini terjadi ketika proses compile terjadi. Hal ini karena perubahan type terjadi pada method yang static. Jika kita perhatikan method cetak pada class A dan class B.

class A:

```
public static String cetak() {  
    return "Class A";  
}
```

class B:

```
public static String cetak() {  
    return "Class B";  
}
```

Bila kita melakukan simulasi pada kelas-kelas tersebut:

```
System.out.println(A.cetak()); // class A  
System.out.println(B.cetak()); // class B
```

Memang terlihat normal, **namun** untuk dapat memanggil *cetak()* yang sesuai harus dilakukan dengan menspesifikasi class A atau B, hal ini disebabkan karena method static tidak dapat di **Overridden** sehingga jika pada class B dideklarasikan static *cetak* baru maka kita tidak akan bisa memanggil *super().cetak()* pada class B.

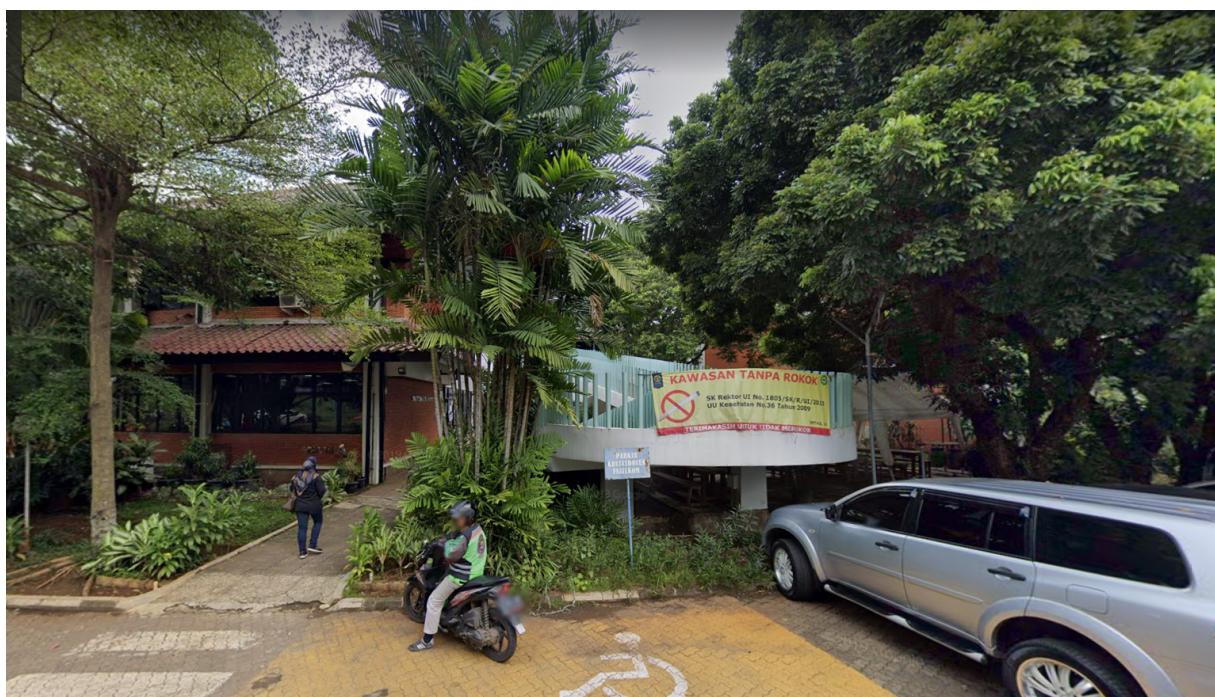
-----EOF Materi-----

# Soal Lab 06

---

## Pacilian Constructor

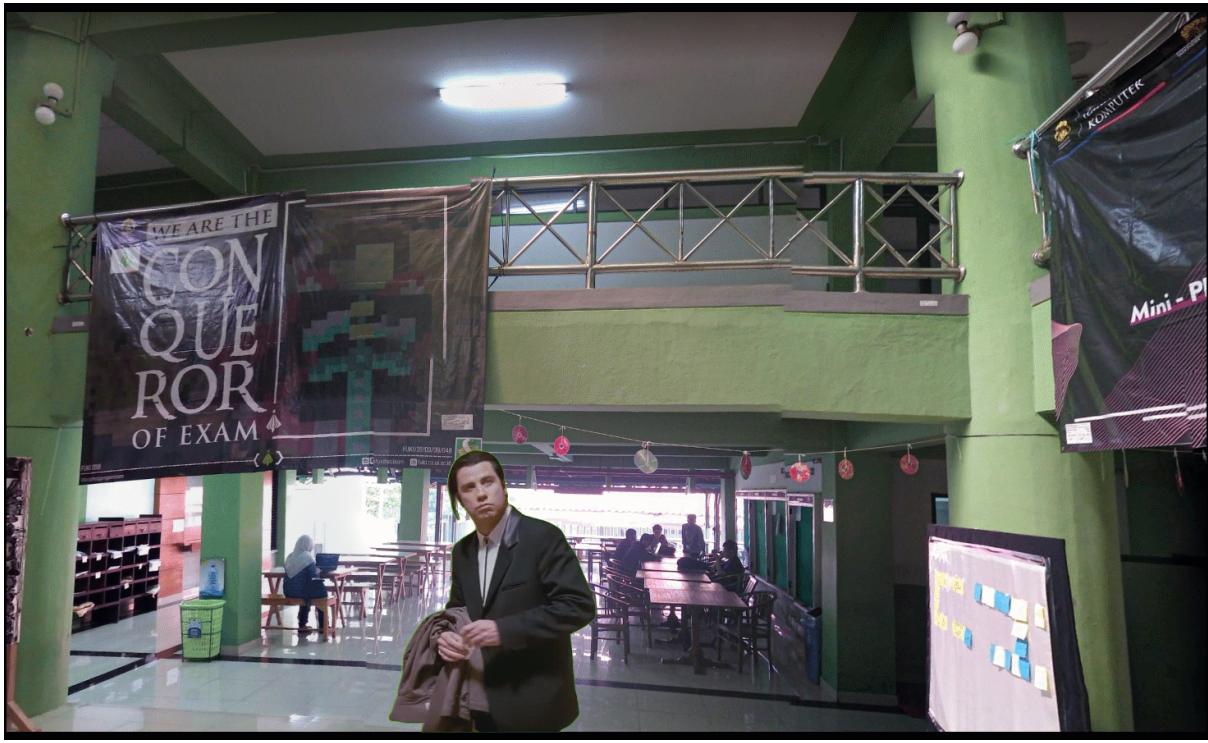
Waktu menunjukkan pukul 07.00 pagi. Kamu yang sudah bersiap untuk kuliah, berangkat menuju Fasilkom UI. 30 menit kemudian, kamu sampai di tempat *drop off*, di belalai Yuli depan Yoshinoya. Lalu kamu turun dari ojol, berterimakasih, lalu mengecek HP untuk melihat jadwal kelas jam 8 nanti.



“Oh, ternyata jam 8 pagi ini saya ada kelas DDP2 di 2402”, ucapmu dalam hati.

Lalu, kamu pun berjalan masuk ke gedung B lewat belalai Yuli. Sesampainya di Sekre Square (kamu belum tahu tempat ini bernama Sekre Square) lalu kamu melihat sekeliling. Ada banyak orang di sini, ada yang lagi diskusi, nugas, ada juga yang lagi lari terbirit-birit ke toilet mau setoran habis dari perjalanan KRL Tangerang-UI. Setelah itu, kamu sadar kalau tidak tahu ruang 2402 itu di mana.

Di sekre square, kamu langsung kebingungan karena ada banyak sekali jalan yang belum pernah kamu lewati. Kamu juga tidak tahu di mana tangga terdekat (dan apakah lift boleh dipakai mahasiswa), jadi kamu menghabiskan sekitar 10 menit berkeliling sambil bersiu.



Setelah bertanya ke beberapa orang, akhirnya kamu berhasil sampai ke ruang 2402, dan ternyata kode 2402 itu adalah: Gedung 2 (B) Lantai 4 Ruang 02. Tidak apa apa, namanya juga first timer 😊.

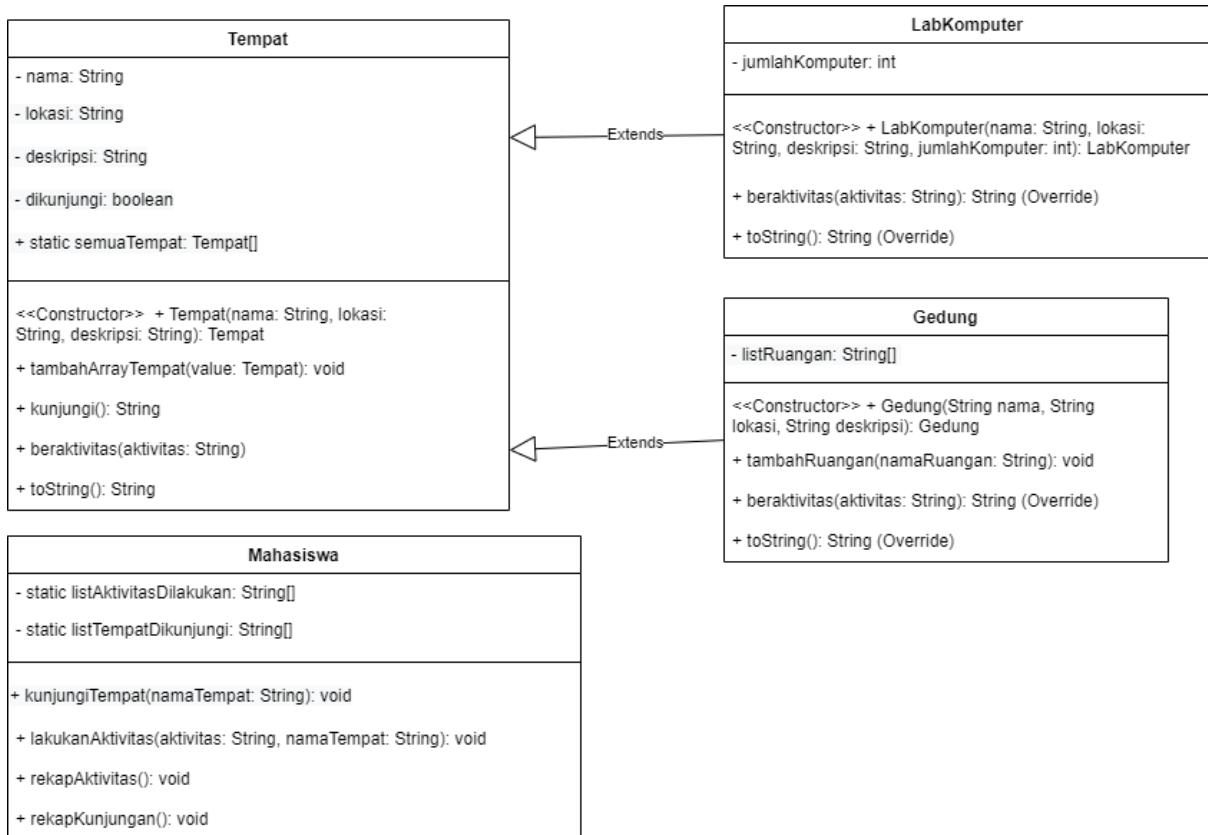
Tidak ingin mengulangi hal yang sama, kamu pun bertekad untuk menghafalkan setiap tempat yang ada di Fasilkom UI, mulai dari nama gedung, hingga apa saja yang bisa dilakukan di suatu tempat tertentu. Kamu pun bertanya kepada seorang kating yang ternyata seorang Asdos DDP2 sekaligus Panitia PMB. Ia pun lalu menjelaskan seluk beluk Fasilkom UI, apa saja yang ada di sini, termasuk beberapa trivia menarik seputar Fasilkom UI.

Lalu kamu pulang dengan membawa ilmu yang sangat banyak. Tidak ingin melupakannya, kamu pun lantas ngide untuk membuat sebuah program Virtual Tour untuk Fasilkom UI, dengan menggunakan CLI. "Supaya saya bisa ingat semua yang diceritakan tadi, aku akan membuat program ini, dan jika aku lupa aku hanya perlu mendatanginya secara virtual dan saya bisa langsung ingat". Begitulah rencananya.

Namun, karena terlalu banyak hal yang terdapat di Fasilkom UI (mulai dari penamaan kantin yang tertukar: balgebun/belakang gedung bundar versus kantin tivi, dan sebagainya) kamu jadinya membatasi apa yang bisa dilakukan oleh program tersebut, menjadi seperti spesifikasi Soal di bawah. Setelah selesai, kamu tidak hanya akan menggunakannya untuk diri sendiri, namun kamu juga akan membagikannya kepada teman-teman lain dan adik tingkat yang baru saja diterima di Fasilkom UI.

TLDR; Kamu ditugaskan untuk membuat program yang dapat digunakan untuk melakukan kunjungan ke Fasilkom UI secara virtual. Nantinya seorang mahasiswa (nanti ada objeknya) yang akan mewakili kalian di dunia virtual tersebut untuk berkunjung kesana sini dan melakukan aktivitas.

Berikut adalah UML Diagram yang menggambarkan relasi dari setiap kelas yang akan dibuat:



Tugas kamu adalah menyelesaikan program sesuai dengan UML Diagram diatas. Template yang disediakan sudah berisi setiap file dengan beberapa implementasi, kalian cukup menyelesaikan bagian yang diberi tanda TODO.

Penjelasan untuk UML Diagram diatas adalah sebagai berikut:

1. Program utama yang dijalankan adalah **Mahasiswa.java**, dimana kamu akan bertindak sebagai seorang mahasiswa di dunia Virtual tersebut.
2. Jika **menggunakan template** yang sudah disediakan, Mahasiswa **hanya perlu** mengerjakan bagian yang diberi // TODO: saja. Semua bagian lain sudah diimplementasikan untuk mempermudah penggerjaan dan pemahaman materi. Jika mahasiswa memilih **tidak menggunakan Template**, maka output program harus sesuai dengan yang diminta di **poin 7 sampai 11**.

3. Panah segitiga putih adalah simbol Inheritance.
  4. Tanda + menunjukkan bahwa variabel/method tersebut bersifat Public, tanda - berarti Private, dan <>Constructor>> menunjukkan bahwa method tersebut adalah method Constructor.
  5. Format penulisan variabel/atribut adalah <Visibility Variabel: TipeData>. Contohnya "+ nama: String", dan format penulisan method adalah <visibility> NamaMethod(TipeData Argumen): Return Type. Contohnya:  
" + beraktivitas(String aktivitas): void"
6. Setiap objek tempat dipastikan memiliki value "nama" yang unik, dengan konvensi Camel Case. Contoh: Gedung B, Lab 1101.
  7. Ketika mahasiswa memilih untuk mengunjungi suatu tempat, akan terjadi hal berikut:
    - Tempat tersebut akan ditandai sebagai sudah dikunjungi (perhatikan bahwa ada variabel boolean dikunjungi);
    - Nama Tempat tersebut akan masuk ke listTempatDikunjungi;
    - Sebuah aktivitas baru dengan teks "Kamu mengunjungi <Nama Tempat>." akan dicetak ke terminal dan ditambahkan ke listAktivitasDilakukan.

Setelah mengunjungi suatu tempat, kamu bisa melakukan sebuah aktivitas di tempat tersebut, atau mengunjungi tempat lain.

Jika tempat yang ingin dikunjungi tidak ada, program akan mencetak "Maaf, tempat tersebut tidak ada di Fasilkom UI." lalu program akan kembali ke daftarMenu (Untuk output ini tidak perlu dimasukkan ke listTempatDikunjungi maupun listAktivitasDilakukan).

8. Ketika mahasiswa memilih untuk melakukan suatu aktivitas di suatu tempat, akan ada beberapa kemungkinan:
  - a. **Tempatnya ada dan aktivitasnya ada.** Pada kasus ini, program akan mencetak "Saya <aktivitas> di <nama Tempat>.". Lalu teks yang sama akan disimpan di listAktivitasDilakukan. Tidak ada batasan seberapa banyak mahasiswa melakukan suatu aktivitas. Setelah itu program akan kembali ke daftarMenu. Jika kamu baru pertama kali melakukan sesuatu di tempat itu(dan tempat itu belum dikunjungi) maka variabel dikunjungi diubah jadi True.
  - b. **Tempatnya ada, namun aktivitasnya tidak ada.** Pada kasus ini, program akan mencetak "Di <nama tempat>, <aktivitas> tidak bisa dilakukan.". Lalu teks yang sama akan disimpan di listAktivitasDilakukan. Contoh kasus nyata: aktivitas "makan nasi

- padang” di “Laboratorium Komputer 1103” tidak bisa dilakukan. Setelah itu program akan kembali ke daftarMenu.
- c. **Tempatnya tidak ada.** Disini, aktivitasnya diabaikan. Program akan mencetak “Tidak ada tempat bernama <nama tempat>.” Setelah itu program akan kembali ke daftarMenu.
9. Mahasiswa boleh tidak melakukan apa apa selain datang/mengunjungi suatu tempat, karena mengunjungi sudah termasuk melakukan aktivitas(yaitu mengunjungi).
  10. Sebuah Lab Komputer bisa memiliki komputer yang sangat banyak. Jika mahasiswa menggunakan komputer, String outputnya adalah “Saya menggunakan komputer di <nama lab>”, sesuai dengan format pada poin 7. Jika `toString()` dipanggil, maka akan mengembalikan Nama, Lokasi, Deskripsi, Jumlah Komputer, dan boolean Dikunjungi.
  11. Sebuah Gedung bisa memiliki ruangan yang sangat banyak. Jika mahasiswa memilih belajar di gedung, String outputnya adalah “Saya belajar di ruangan <nama ruangan>”, sesuai dengan format pada poin 7. Jika `toString()` dipanggil, maka akan mengembalikan Nama, Lokasi, Deskripsi, Jumlah Ruangan, dan boolean Dikunjungi.
  12. Data untuk setiap tempat (termasuk data seperti ruangan) sudah di-generate di file denah-fasilkom.txt dan tinggal dibaca oleh program. Tenang, fungsi untuk menerjemahkannya menjadi objek siap panggil sudah disediakan di template :D namun kalian boleh menambahkan sendiri tempat baru sesuai keinginan di file tersebut sebelum “Virtual Tour”-nya dimulai.
  13. Semua objek Tempat akan disimpan di static variable semuaTempat di kelas Tempat. Objek mahasiswa tinggal memanggil static method `getTempat` di kelas Tempat untuk mendapatkan suatu objek Tempat(sudah disediakan di template). Semua tempat yang ingin dikunjungi belum tentu ada di Fasilkom UI, kalian bisa menerapkan konsep defensive programming disini.
  14. Setelah selesai berkunjung, kamu tinggal ketik “selesai” dan program akan mengeluarkan rekap semua tempat yang sudah dikunjungi, dan aktivitas apa saja yang dilakukan di tempat tersebut(terurut secara kronologis). Implementasinya dibebaskan kepada mahasiswa, namun supaya tidak perlu memanggil setiap objek Tempat lagi, bisa memanfaatkan array yang sudah diberikan di Template.
  15. Setiap kelas memiliki method Getter dan Setter yang berfungsi untuk memungkinkan objek/kelas lain berinteraksi dengan variabel yang dimiliki suatu kelas tersebut.

**Catatan:** Anda diperbolehkan menambah atribut atau method baru apabila diperlukan, tapi semua atribut dan method yang sudah ada tidak boleh diubah. Untuk mengecek apakah semua kelas yang Anda implementasikan sudah benar atau belum, bisa menggunakan opsi 0 pada daftarMenu. Opsi debug ini akan mencetak `toString()` semua objek Tempat yang ada. Contoh keluarannya ada dibawah Contoh Input/Output.

## Contoh Input/Output 1

(Tulisan yang di **highlight biru** adalah user input):

```
Solusi$ java Mahasiswa  
===== Virtual Tour Fasilkom UI =====
```

```
Selamat datang di Aplikasi Virtual Tour Fasilkom UI!  
Sedang mengisi Database dengan data dari file denah-fasilkom.txt  
Selesai mengisi Database. Selamat menikmati Virtual Tournya!
```

```
----- MENU -----
```

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: **1**

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

**list tempat**

Nama: Gedung A

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung A Fasilkom UI, di lantai 1 terdapat beberapa lab komputer dan lab Babe dan di lantai 2 terdapat ruang Dosen dan lab KI.

Jumlah Ruangan: 1

Sudah dikunjungi: false

Nama: Gedung B

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung B Fasilkom UI, di lantai 1 terdapat Perpustakaan Fasilkom UI(disingkat Perpucil), Aula, Gudang Belakang Aula, dan ruang Lift. Di lantai 2 terdapat 2 Mushola, Ruang BEM, Ruang Ristek, Loket Sekretariat, Ruang Fasum, Sekre Square, Kafe Sekre, jalan Belalai Yuli dan Kantin, dan lift. Lantai 3-6 adalah ruang kelas. Pada lantai 5 ada area belajar dan balkon, lantai 6 ada Lab 2601 dan rooftop di lantai 7.

Jumlah Ruangan: 12

Sudah dikunjungi: false

Nama: Gedung C

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung C Fasilkom UI, di lantai 1 terdapat beberapa ruang kelas dan ruang khusus, di lantai 2 dan 3 terdapat ruang Dosen dan Staf Fasilkom UI.

Jumlah Ruangan: 2

Sudah dikunjungi: false

Nama: Lab 1101/1103

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang kedua adalah Lab 1105/1107.

Jumlah Komputer tersedia: 120

Sudah dikunjungi: false

Nama: Lab 1105/1107

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang pertama adalah Lab 1101/1103.

Jumlah Komputer tersedia: 100

Sudah dikunjungi: false

Nama: Lab 2601

Lokasi: Lantai 6

Deskripsi: Lab Komputer yang paling jarang digunakan, karena waktu buka nya hanya dari jam 8 pagi sampai jam 5 sore.

Jumlah Komputer tersedia: 80

Sudah dikunjungi: false

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: **1**

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

**Gedung A**

----- BERKUNJUNG -----

Saya mengunjungi Gedung A.

Apakah kamu ingin melakukan sesuatu disini?

Jika kamu mengetik ya, maka program akan mencetak aktivitas yang bisa dilakukan di tempat tersebut,  
dan kamu harus mengetik aktivitas tersebut untuk melakukannya.  
Jika kamu mengetik yang lain, maka program akan kembali ke menu utama.

**ya**

Di Gedung, kamu bisa Belajar. List semua ruangan:  
[Babe]

Apa yang ingin kamu lakukan? **belajar Babe**

----- BERAKTIVITAS -----

Saya belajar di ruang Babe.

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: **1**

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

Gedung B

----- BERKUNJUNG -----

Saya mengunjungi Gedung B.

Apakah kamu ingin melakukan sesuatu disini?  
Jika kamu mengetik ya, maka program akan mencetak aktivitas yang bisa dilakukan di tempat tersebut,  
dan kamu harus mengetik aktivitas tersebut untuk melakukannya.  
Jika kamu mengetik yang lain, maka program akan kembali ke menu utama.

ya

Di Gedung, kamu bisa Belajar. List semua ruangan:  
[2301, 2302, 2303, 2304, 2305, 2306, 2401, 2402, 2403, 2404, 2405, 2406]

Apa yang ingin kamu lakukan? belajar 2304

----- BERAKTIVITAS -----

Saya belajar di ruang 2304.

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: 2

Mau beraktivitas apa, dan dimana? Format: Aktivitas;Nama Tempat  
Jika ingin melihat apa saja yang bisa dilakukan di suatu tempat,  
ketik:

"List Aktivitas".

Ketik "List Tempat" untuk melihat semua tempat yang ada.

belajar 2406;Gedung B

----- BERAKTIVITAS -----

Saya belajar di ruang 2406.

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: 2

Mau beraktivitas apa, dan dimana? Format: Aktivitas;Nama Tempat  
Jika ingin melihat apa saja yang bisa dilakukan di suatu tempat,  
ketik:

"List Aktivitas".

Ketik "List Tempat" untuk melihat semua tempat yang ada.

belajar 2602;Gedung B

----- BERAKTIVITAS -----

Tidak ada ruangan bernama 2602.

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: 3

Melakukan Rekap Kunjungan...Selama Virtual tour, kamu telah  
mengunjungi tempat berikut:

Gedung A

Gedung A

Gedung B

Gedung B  
Gedung B  
Gedung B

Melakukan Rekap Aktivitas...Selama Virtual tour, aktivitas yang telah kamu lakukan adalah:

Saya mengunjungi Gedung A.  
Saya belajar di ruang Babe.  
Saya mengunjungi Gedung B.  
Saya belajar di ruang 2304.  
Saya belajar di ruang 2406.  
Tidak ada ruangan bernama 2602.

Virtual Tournya selesai!  
Sampai jumpa!

#### **Contoh output debug:**

=====DEBUGGING=====

Tempat yang berhasil didaftarkan:

Nama: Gedung A

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung A Fasilkom UI, di lantai 1 terdapat beberapa lab komputer dan lab Babe dan di lantai 2 terdapat ruang Dosen dan lab KI.

Jumlah Ruangan: 1

Sudah dikunjungi: false

Nama: Gedung B

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung B Fasilkom UI, di lantai 1 terdapat Perpustakaan Fasilkom UI(disingkat Perpucil), Aula, Gudang Belakang Aula, dan ruang Lift. Di lantai 2 terdapat 2 Mushola, Ruang BEM, Ruang Ristek, Loket Sekretariat, Ruang Fasum, Sekre Square, Kafe Sekre, jalan Belalai Yuli dan Kantin, dan lift. Lantai 3-6 adalah ruang kelas.

Pada lantai 5 ada area belajar dan balkon, lantai 6 ada Lab 2601 dan rooftop di lantai 7.

Jumlah Ruangan: 12

Sudah dikunjungi: false

Nama: Gedung C

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung C Fasilkom UI, di lantai 1 terdapat beberapa ruang kelas dan ruang khusus, di lantai 2 dan 3 terdapat ruang Dosen dan Staf Fasilkom UI.

Jumlah Ruangan: 2

Sudah dikunjungi: false

Nama: Lab 1101/1103

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang kedua adalah Lab 1105/1107.

Jumlah Komputer tersedia: 120

Sudah dikunjungi: false

Nama: Lab 1105/1107

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang pertama adalah Lab 1101/1103.

Jumlah Komputer tersedia: 100

Sudah dikunjungi: false

Nama: Lab 2601

Lokasi: Lantai 6

Deskripsi: Lab Komputer yang paling jarang digunakan, karena waktu buka nya hanya dari jam 8 pagi sampai jam 5 sore.

Jumlah Komputer tersedia: 80

Sudah dikunjungi: false

=====DEBUGGING=====

## Contoh Input/Output 2

(Tulisan yang di **highlight biru** adalah user input):

Solusi\$ java Mahasiswa

===== Virtual Tour Fasilkom UI =====

Selamat datang di Aplikasi Virtual Tour Fasilkom UI!

Sedang mengisi Database dengan data dari file denah-fasilkom.txt

Selesai mengisi Database. Selamat menikmati Virtual Tournya!

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: 0

=====DEBUGGING=====

Tempat yang berhasil didaftarkan:

Nama: Gedung A

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung A Fasilkom UI, di lantai 1 terdapat beberapa lab komputer dan lab Babe dan di lantai 2 terdapat ruang Dosen dan lab KI.

Jumlah Ruangan: 1

Sudah dikunjungi: false

Nama: Gedung B

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung B Fasilkom UI, di lantai 1 terdapat Perpustakaan Fasilkom UI(disingkat Perpucil), Aula, Gudang Belakang Aula, dan ruang Lift. Di lantai 2 terdapat 2 Mushola, Ruang BEM, Ruang Ristek, Loket Sekretariat, Ruang Fasum, Sekre Square, Kafe Sekre, jalan Belalai Yuli dan Kantin, dan lift. Lantai 3-6 adalah ruang kelas.

Pada lantai 5 ada area belajar dan balkon, lantai 6 ada Lab 2601 dan rooftop di lantai 7.

Jumlah Ruangan: 12

Sudah dikunjungi: false

Nama: Gedung C

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung C Fasilkom UI, di lantai 1 terdapat beberapa ruang kelas dan ruang khusus, di lantai 2 dan 3 terdapat ruang Dosen dan Staf Fasilkom UI.

Jumlah Ruangan: 2

Sudah dikunjungi: false

Nama: Lab 1101/1103

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang kedua adalah Lab 1105/1107.

Jumlah Komputer tersedia: 120

Sudah dikunjungi: false

Nama: Lab 1105/1107  
Lokasi: Lantai 1  
Deskripsi: Lab Komputer terluas di Fasilkom UI, yang pertama adalah Lab 1101/1103.  
Jumlah Komputer tersedia: 100  
Sudah dikunjungi: false

Nama: Lab 2601  
Lokasi: Lantai 6  
Deskripsi: Lab Komputer yang paling jarang digunakan, karena waktu buka nya hanya dari jam 8 pagi sampai jam 5 sore.  
Jumlah Komputer tersedia: 80  
Sudah dikunjungi: false

=====DEBUGGING=====

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: **1**

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

**list tempat**

Nama: Gedung A

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung A Fasilkom UI, di lantai 1 terdapat beberapa lab komputer dan lab Babe dan di lantai 2 terdapat ruang Dosen dan lab KI.

Jumlah Ruangan: 1

Sudah dikunjungi: false

Nama: Gedung B

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung B Fasilkom UI, di lantai 1 terdapat Perpustakaan Fasilkom UI(disingkat Perpucil), Aula, Gudang Belakang Aula, dan ruang Lift. Di lantai 2 terdapat 2 Mushola, Ruang BEM, Ruang Ristek,

Loket Sekretariat, Ruang Fasum, Sekre Square, Kafe Sekre, jalan Belalai Yuli dan Kantin, dan lift. Lantai 3-6 adalah ruang kelas. Pada lantai 5 ada area belajar dan balkon, lantai 6 ada Lab 2601 dan rooftop di lantai 7.

Jumlah Ruangan: 12

Sudah dikunjungi: false

Nama: Gedung C

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung C Fasilkom UI, di lantai 1 terdapat beberapa ruang kelas dan ruang khusus, di lantai 2 dan 3 terdapat ruang Dosen dan Staf Fasilkom UI.

Jumlah Ruangan: 2

Sudah dikunjungi: false

Nama: Lab 1101/1103

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang kedua adalah Lab 1105/1107.

Jumlah Komputer tersedia: 120

Sudah dikunjungi: false

Nama: Lab 1105/1107

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang pertama adalah Lab 1101/1103.

Jumlah Komputer tersedia: 100

Sudah dikunjungi: false

Nama: Lab 2601

Lokasi: Lantai 6

Deskripsi: Lab Komputer yang paling jarang digunakan, karena waktu buka nya hanya dari jam 8 pagi sampai jam 5 sore.

Jumlah Komputer tersedia: 80

Sudah dikunjungi: false

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur

#### 4. Keluar

Pilih: 1

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

Gedung C

----- BERKUNJUNG -----

Saya mengunjungi Gedung C.

Apakah kamu ingin melakukan sesuatu disini?  
Jika kamu mengetik ya, maka program akan mencetak aktivitas yang  
bisa dilakukan di tempat tersebut,  
dan kamu harus mengetik aktivitas tersebut untuk melakukannya.  
Jika kamu mengetik yang lain, maka program akan kembali ke menu  
utama.

belajar 3112

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: 1

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

Gedung C

----- BERKUNJUNG -----

Saya mengunjungi Gedung C.

Apakah kamu ingin melakukan sesuatu disini?  
Jika kamu mengetik ya, maka program akan mencetak aktivitas yang  
bisa dilakukan di tempat tersebut,  
dan kamu harus mengetik aktivitas tersebut untuk melakukannya.

Jika kamu mengetik yang lain, maka program akan kembali ke menu utama.

ya

Di Gedung Gedung C, kamu bisa Belajar. List semua ruangan:

[3112, 3113]

Apa yang ingin kamu lakukan? **Belajar 3112**

----- BERAKTIVITAS -----

Saya belajar di ruang 3112.

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: **1**

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

**Lab 1101/1103**

----- BERKUNJUNG -----

Saya mengunjungi Lab 1101/1103.

Apakah kamu ingin melakukan sesuatu disini?

Jika kamu mengetik ya, maka program akan mencetak aktivitas yang bisa dilakukan di tempat tersebut,  
dan kamu harus mengetik aktivitas tersebut untuk melakukannya.  
Jika kamu mengetik yang lain, maka program akan kembali ke menu utama.

ya

Di LabKomputer ini, kamu bisa Gunakan Komputer.

Apa yang ingin kamu lakukan? **Gunakan Komputer**

----- BERAKTIVITAS -----

Saya menggunakan Komputer di Lab 1101/1103

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: 1

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

list tempat

Nama: Gedung A

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung A Fasilkom UI, di lantai 1 terdapat beberapa lab komputer dan lab Babe dan di lantai 2 terdapat ruang Dosen dan lab KI.

Jumlah Ruangan: 1

Sudah dikunjungi: false

Nama: Gedung B

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung B Fasilkom UI, di lantai 1 terdapat Perpustakaan Fasilkom UI(disingkat Perpucil), Aula, Gudang Belakang Aula, dan ruang Lift. Di lantai 2 terdapat 2 Mushola, Ruang BEM, Ruang Ristek, Loket Sekretariat, Ruang Fasum, Sekre Square, Kafe Sekre, jalan Belalai Yuli dan Kantin, dan lift. Lantai 3-6 adalah ruang kelas.

Pada lantai 5 ada area belajar dan balkon, lantai 6 ada Lab 2601 dan rooftop di lantai 7.

Jumlah Ruangan: 12

Sudah dikunjungi: false

Nama: Gedung C

Lokasi: Fasilkom UI Lama

Deskripsi: Gedung C Fasilkom UI, di lantai 1 terdapat beberapa ruang kelas dan ruang khusus, di lantai 2 dan 3 terdapat ruang Dosen dan Staf Fasilkom UI.

Jumlah Ruangan: 2

Sudah dikunjungi: true

Nama: Lab 1101/1103

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang kedua adalah Lab 1105/1107.

Jumlah Komputer tersedia: 120

Sudah dikunjungi: true

Nama: Lab 1105/1107

Lokasi: Lantai 1

Deskripsi: Lab Komputer terluas di Fasilkom UI, yang pertama adalah Lab 1101/1103.

Jumlah Komputer tersedia: 100

Sudah dikunjungi: false

Nama: Lab 2601

Lokasi: Lantai 6

Deskripsi: Lab Komputer yang paling jarang digunakan, karena waktu buka nya hanya dari jam 8 pagi sampai jam 5 sore.

Jumlah Komputer tersedia: 80

Sudah dikunjungi: false

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: **1**

Mau berkunjung kemana? Jika ingin melihat semua tempat yang ada,  
Ketik "List Tempat"

**Gedung B**

----- BERKUNJUNG -----

Saya mengunjungi Gedung B.

Apakah kamu ingin melakukan sesuatu disini?

Jika kamu mengetik ya, maka program akan mencetak aktivitas yang bisa dilakukan di tempat tersebut,

dan kamu harus mengetik aktivitas tersebut untuk melakukannya. Jika kamu mengetik yang lain, maka program akan kembali ke menu utama.

ya

Di Gedung Gedung B, kamu bisa Belajar. List semua ruangan: [2301, 2302, 2303, 2304, 2305, 2306, 2401, 2402, 2403, 2404, 2405, 2406]

Apa yang ingin kamu lakukan? belajar 2401

----- BERAKTIVITAS -----

Saya belajar di ruang 2401.

----- MENU -----

Silakan pilih menu:

1. Kunjungi suatu Tempat
2. Lakukan sebuah aktivitas di Tempat tertentu
3. Selesaikan tur
4. Keluar

Pilih: 3

Melakukan Rekap Kunjungan...Selama Virtual tour, kamu telah mengunjungi tempat berikut:

Gedung C

Gedung C

Gedung C

Lab 1101/1103

Lab 1101/1103

Gedung B

Gedung B

Melakukan Rekap Aktivitas...Selama Virtual tour, aktivitas yang telah kamu lakukan adalah:

Saya mengunjungi Gedung C.

Saya mengunjungi Gedung C.

Saya belajar di ruang 3112.

Saya mengunjungi Lab 1101/1103.

Saya menggunakan Komputer di Lab 1101/1103

Saya mengunjungi Gedung B.

Saya belajar di ruang 2401.

Virtual Tournya selesai!  
Sampai jumpa!

-- End of Contoh Input/Output --

### Komponen Penilaian

- 30% Implementasi *class* Tempat
- 25% Implementasi *class* Gedung
- 20% Implementasi *class* LabKomputer
- 15% Implementasi *class* Mahasiswa
- 10% Kerapian kode dan dokumentasi

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

**Lab06\_[Kelas]\_[KodeAsdos]\_[NPM]\_[NamaLengkap].zip**

Contoh:

**Lab06\_A\_LN\_1234567890\_DekDepe.zip**

Selamat mengerjakan! 😊