

Dasar-Dasar Pemrograman 2

Lab 06

Inheritance 2



FAKULTAS
ILMU
KOMPUTER

Riwayat Versi

Versi	Timestamp	Keterangan	Warna
1	13-04-2023; 15.00 WIB	Rilis Pertama	-
2	13-04-2023; 15.35 WIB	Revisi batasan input dan pilihan user serta perbaikan deskripsi properti hasMilk	
3	13-04-2023; 19.20 WIB	Revisi deskripsi atribut size dari beverage.java	



Template soal dapat diakses pada link [ini](#)

Starcil



Image : Hotpot.AI

Deskripsi

Setelah membangun startup PacilWorks, Dek Depe ingin mencoba peruntungannya dalam bidang minuman yang bernama Starcil. Saat ini, bisnis minuman Dek Depe hanya mengandalkan perhitungan manual setiap kali pembayaran dilakukan. Namun, Dek Depe mulai kesulitan ketika banyak pelanggan yang datang ke tempat minumannya. Sebagai teman yang baik, kamu berencana untuk membantu Dek Depe membuat sebuah sistem pembayaran minuman untuk bisnis Starcil.

Task

Pada Starcil, terdapat 4 buah *class* yang perlu dilengkapi, yaitu:

- **Beverage.java**

Class Beverage merupakan *parent class* dari spesialisasi *class* di bawahnya, yaitu **Coffee** dan **Tea**. Lengkapilah *attribute*, *constructor*, dan *method* yang ada pada *template*.

Properti	Type	Deskripsi
name	String	<i>Attribute</i> ini merepresentasikan nama dari minuman yang ada pada bisnis minuman Starcil
size	String	<i>Attribute</i> ini merepresentasikan ukuran dari minuman yang ada pada bisnis minuman Starcil. Adapun ukuran-ukuran yang tersedia pada bisnis minuman Starcil, yaitu Tall , Grande , dan Venti (Case Insensitive) .
isCold	boolean	<i>Attribute</i> ini merepresentasikan minuman dengan tipe dingin atau panas. <i>Attribute</i> ini akan bernilai true apabila minuman bertipe dingin, sedangkan false apabila minuman bertipe panas.
price	int	<i>Attribute</i> ini merepresentasikan harga dari minuman yang dijual pada bisnis minuman Starcil.

- Harga dari minuman tergantung pada ukuran dan jenis dari minuman tersebut yang dapat dirincikan melalui tabel berikut,

Ukuran/ Jenis	Coffee	Tea
Tall	20000	15000
Grande	25000	20000
Venti	30000	25000

- **Coffee.java**

Class **Coffee** merupakan salah satu spesialisasi dari **Beverage**. Lengkapilah *attribute*, *constructor*, dan *method* yang ada pada *template*. Adapun *attribute* tambahan yang ada pada class **Coffee**:

Properti	Type	Deskripsi
hasWhipCream	boolean	<p><i>Attribute</i> ini merepresentasikan minuman <i>coffee</i> yang memiliki tambahan <i>whip cream</i> atau tidak.</p> <ul style="list-style-type: none"> • <i>Attribute</i> ini mempunyai nilai true apabila minuman <i>coffee</i> memiliki tambahan <i>whip cream</i>, sedangkan false apabila minuman <i>coffee</i> tidak memiliki tambahan <i>whip cream</i>. • Penambahan <i>whip cream</i> akan meningkatkan harga dari <i>coffee</i> sebanyak 5000.

- **Tea.java**

Class **Tea** merupakan salah satu spesialisasi dari **Beverage**. Lengkapilah *attribute*, *constructor*, dan *method* yang ada pada *template*. Adapun *attribute* tambahan yang ada pada class **Tea**:

Properti	Type	Deskripsi
hasMilk	boolean	<p><i>Attribute</i> ini merepresentasikan minuman <i>tea</i> yang memiliki tambahan <i>milk</i> atau tidak.</p> <ul style="list-style-type: none"> • <i>Attribute</i> ini mempunyai nilai true apabila minuman <i>tea</i> memiliki tambahan <i>milk</i>, sedangkan false apabila minuman tea tidak memiliki tambahan <i>milk</i>. • Penambahan <i>milk</i> akan meningkatkan harga dari <i>tea</i> sebanyak 7000.

- **Starcil.java**

Class **Starcil** merupakan *class* yang akan memuat *main method*. Adapun atribut pada class **Starcil**:

Properti	Type	Deskripsi
daftarMinuman	Beverage[]	<i>Attribute</i> ini akan menyimpan seluruh daftar minuman yang ada pada bisnis minuman Starcil. Inisiasi Array daftarMinuman sudah tersedia pada template pengerjaan.

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 COFFEE [NAMA] [UKURAN] [ES]**
 - ◆ Menambahkan minuman coffee dengan nama **[NAMA]** dengan ukuran **[UKURAN]** (TALL/GRANDE/VENTI) dan apakah minuman dingin atau tidak **[ES]** (YES/NO) ke dalam Array daftarMinuman
 - **ADD TEA [NAMA] [UKURAN] [ES]**
 - ◆ Menambahkan minuman tea dengan nama **[NAMA]** dengan ukuran **[UKURAN]** (TALL/GRANDE/VENTI) dan apakah minuman dingin atau tidak **[ES]** (YES/NO) ke dalam Array daftarMinuman
 - **TOPPING [TOPPING]**
 - ◆ Menambahkan topping ke minuman dengan nama **[TOPPING]**, topping akan menyesuaikan dengan jenis minuman yang dipilih
 - **ORDERAN [JENIS]**
 - ◆ Mengeluarkan semua orderan dengan jenis **[JENIS]** (COFFEE/TEA) yang ada di sistem.



Input dan pilihan user dijamin valid, sehingga tidak perlu melakukan validasi input.

Test Case

Contoh Interaksi 1

```
4
ADD TEA test VENTI YES
TOPPING test
ORDERAN COFFEE
ORDERAN TEA
OUTPUT
COLD VENTI test with Milk Rp. 32000,-
```

Penjelasan TC 1

ADD TEA VENTI COLD → Set harga teh menjadi Rp.25000

Add topping teh → Set harga 25000 + 70000

ORDERAN COFFEE → tidak menghasilkan output karena tidak ada menu coffee yang tersedia

ORDERAN TEA → Print Teh test yang sudah dibuat dan ditambahkan topping

Contoh Interaksi 2

```
4
ADD COFFEE test VENTI YES
TOPPING test
TOPPING test
ORDERAN COFFEE
OUTPUT
COLD VENTI test with Whip Cream Rp. 35000,-
```

Penjelasan TC 2

ADD COFFEE VENTI COLD → Set harga kopi test menjadi 30000

TOPPING test → Add topping ke test sebanyak 1x → Harga menjadi 35000

TOPPING test → Karena test sudah memiliki topping, harga tidak bertambah

ORDERAN COFFEE → print output dari coffee yang sudah dibentuk

Komponen Penilaian

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

Pengumpulan Berkas

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.
[KodeAsdos]_[Kelas]_[NPM]_[NamaLengkap]_Lab06.zip

Contoh:

DDP_A_1234567890_DekDePe_Lab06.zip