

# Dasar-Dasar Pemrograman 2

## Lab 04

### Objects & Classes



FAKULTAS  
ILMU  
KOMPUTER

---

#### Riwayat Versi

Versi	Timestamp	Keterangan	Warna
1	9-03-2023; 14.00 WIB	Rilis Pertama	-
2	09-03-2023; 17.42 WIB	Revisi Pertama	Kuning

# Diskonpedia



(Sumber: <https://rectmedia.com/10-cara-simpel-agar-online-shop-anda-lebih-menarik-dan-menonjol/>)

## Deskripsi

Di dunia yang lain, Dek Depe merupakan seorang mahasiswa Fasilkom UI yang sangat menyukai bisnis. Saat ini, Dek Depe mempunyai satu toko di Depok. Namun, karena hujan deras yang menimpa Depok belakangan ini, jumlah pelanggan yang datang ke tokonya berkurang secara signifikan. Oleh karena itu, Dek Depe memutuskan untuk membuka tokonya secara *online* sebagai alternatif dari toko *offline* nya, dengan diskon yang besar untuk menarik perhatian banyak orang.

Akan tetapi, Dek Depe sadar bahwa ia tidak dapat membuat suatu *online shop* karena ia seringkali tidak memperhatikan kelas akibat terlalu fokus dengan tokonya. Namun, Dek Depe mengingat bahwa ia mempunyai teman dekat, yaitu kamu, yang pastinya mau membantu Dek Depe dalam membuat suatu *online shop* 😊.

## Ketentuan Program

Program ini memiliki *template* yang bisa di akses pada [link](#) ini untuk mempermudah pengerjaan Lab 04. Terdapat empat perintah yang dapat dijalankan pada program *online shop* ini:

- **PESAN [NAMA] [BARANG] [JUMLAH]**

Perintah ini berguna bagi pembeli bernama **[NAMA]** untuk memesan barang bernama **[BARANG]** sejumlah **[JUMLAH]**. Barang yang dipesan **[NAMA]** akan dimasukkan ke dalam list pesanan dari pembeli **[NAMA]** dan stok **[BARANG]** akan berkurang. Jika barang **[BARANG]** sudah pernah dipesan, maka tambahkan **[JUMLAH]** ke pesanan barang yang lama. Ada tiga kondisi yang membuat pembeli **[NAMA]** tidak dapat memesan barang **[BARANG]**:

- Stok barang **[BARANG]** tidak mencukupi.
- Jumlah pesanan **[NAMA]** melebihi jumlah maksimal pesanan saat ditambah oleh barang berjumlah **[JUMLAH]** (maksimal pesanan yang dapat dibuat oleh seorang pembeli dalam satu waktu adalah 20 barang. Contoh: Jika pembeli

memesan 4 aqua dan 1 indomie, maka pembeli tersebut sudah memesan 5 barang sehingga pembeli tersebut tidak bisa membeli indomie sebanyak 16 buah karena akan melebihi jumlah maksimal pesanan yang bisa dibuat).

- Uang **[NAMA]** tidak mencukupi (Contoh: **[NAMA]** memiliki uang 10,000 rupiah dan sudah memesan barang dengan total harga 9,000. **[NAMA]** tidak bisa memesan barang dengan total harga 2,000.).
- **BAYAR [NAMA]**  
Perintah ini berguna untuk pembeli **[NAMA]** membeli dan membayar barang yang sudah dipesan. Perintah ini akan mencetak detail pesanan **[NAMA]**, total harga dari barang-barang yang dipesan **[NAMA]**, besar diskon yang didapat **[NAMA]** dalam uang, dan sisa uang yang dimiliki **[NAMA]**. Setelah perintah ini, list pesanan pembeli tersebut dikosongkan.
- **RESTOCK [BARANG] [JUMLAH\_BARU]**  
Perintah ini berguna untuk menambahkan stok barang **[BARANG]** sebesar **[JUMLAH\_BARU]**. Nilai **[JUMLAH\_BARU]** harus lebih besar atau sama dengan 1.
- **DISKON [NAMA]**  
Perintah ini berguna untuk mencari jumlah diskon yang didapat **[NAMA]** dalam persen.

Diskon dapat diterima oleh pembeli berdasarkan total harga pesanan dan jumlah pesanan yang dibuat pembeli. Ketentuan **cara mencari persentase diskon** yang didapat seorang pembeli adalah sebagai berikut:

- Diskon dihitung dari jumlah barang yang dipesan. Jumlah barang yang dipesan menjadi persentase diskonnya.  
Contoh: Jika seorang pembeli memesan 15 barang, maka ia mendapat diskon 15%

Terdapat empat class yang ada pada file yang berbeda

### 1. Diskonpedia

Class Diskonpedia adalah class utama dari program ini. Class ini berguna untuk menjalankan program kalian dan juga menyimpan list barang dan pembeli yang ada di Diskonpedia. **Pada class ini, kalian tidak perlu membuat proses untuk menerima input karena sudah disediakan oleh template.** Lengkapilah **//TODO** yang ada!

Method yang ada pada class ini selain method main adalah

- void pesan(String namaPembeli, String namaBarang, int jumlah)  
Method ini digunakan untuk perintah PESAN. Method ini akan mencetak string yang dihasilkan saat menjalankan perintah PESAN. **Manfaatkan method tambahPesanan() yang ada di class Pembeli.**
- void bayar(String namaPembeli)

Method ini digunakan untuk perintah BAYAR dan memproses pembelian dan pembayaran barang yang dipesan seorang pembeli. Method ini akan mencetak string-string yang dihasilkan dari perintah BAYAR. Setelah selesai membayar, list pesanan dari pembeli yang membayar akan kosong (diatur ulang).

- void restock(String namaBarang, int jumlah)  
Method ini digunakan untuk perintah RESTOCK dan menambah stok sebuah barang. Method ini akan mencetak string-string yang dihasilkan dari perintah RESTOCK.
- void diskon(String namaPembeli)  
Method ini digunakan untuk perintah DISKON dan mencari persentase diskon yang didapat seorang pembeli. Method ini akan mencetak string-string yang dihasilkan dari perintah DISKON.
- int hitungDiskon(Pembeli pembeli)  
Method ini digunakan untuk menghitung persentase diskon yang didapatkan seorang pembeli. **Implementasikan perhitungan diskon disini.**
- Barang cariBarang(String nama)  
Method ini digunakan untuk mencari barang berdasarkan nama
- Pembeli cariPembeli(String nama)  
Method ini digunakan untuk mencari pembeli berdasarkan nama.

## 2. Pembeli

Class ini merepresentasikan pembeli yang ada pada Diskonpedia. Class ini memiliki empat attribute, yaitu **String nama**, **long jumlahUang**, **Pesanan[] listPesanan**, dan **int MAKS\_JUMLAH\_BARANG** yang semuanya memiliki **visibility private**.

Implementasi **constructor** dan **//TODO** yang ada di class ini.

Method yang ada pada class ini adalah:

- String tambahPesanan(Barang barang, int jumlah)  
Method ini digunakan untuk menambah pesanan ke list pesanan yang ada di pembeli. **Sebelum menambahkan pesanan** ke dalam list pesanan, ada **tiga hal** yang akan **dicek dengan urutan** sebagai berikut:
  1. Apakah **stok barang masih tersedia** atau tidak
  2. Apakah **jumlah pesanan pembeli melebihi jumlah maksimal** saat ditambah dengan pesanan baru
  3. Apakah **uang pembeli cukup untuk membayar pesanan yang lama dan yang baru**

Saat pesanan baru lulus ketiga pengecekan di atas, maka pesanan tersebut akan dimasukkan ke list pesanan **dan stok barang yang dipesan akan berkurang**. **Jika pembeli sudah pernah memesan barang yang sama, tambahkan** pesanan baru tersebut **ke pesanan yang lama**

- void resetPesanan()

Method ini akan mengosongkan isi dari list pesanan.

### 3. Barang

Class ini merepresentasikan barang yang dijual di Diskonpedia. Class ini memiliki tiga attribute dengan visibility private, yaitu long harga, String nama, dan int stok. Class ini sudah diimplementasi pada template.

### 4. Pesanan

Class ini merepresentasikan pesanan yang dibuat oleh pembeli di Diskonpedia. Class ini memiliki dua attribute dengan visibility private, yaitu Barang barang dan int jumlah. Class ini sudah diimplementasi pada template

Method yang ada pada class ini adalah:

- long totalHarga()

Method ini mengembalikan total harga dari suatu pesanan.

## Format Masukan

- Baris pertama berisi **[BanyakBarang]**, banyaknya barang yang ada
- [BanyakBarang] baris berikutnya berisi keterangan setiap barang berupa nama barang, harga barang, dan stok awal barang
- Baris berikutnya berisi [BanyakPembeli], banyaknya pelanggan yang ada
- [BanyakPembeli] baris berikutnya berisi keterangan setiap pembeli berupa nama pembeli dan jumlah uang yang dimiliki pembeli
- Baris berikutnya berisi [JumlahPerintah], banyaknya perintah yang dijalankan
- [JumlahPerintah] baris berikutnya berisi perintah-perintah yang dijalankan sesuai deskripsi setiap perintah

## Batasan

- Barang dan pembeli yang muncul di perintah dijamin ada di dalam list (Contoh: Jika ada perintah *PESAN Andi Marjan 10*, pembeli yang bernama Andi pasti ada di list pembeli dan barang yang bernama Marjan pasti ada di list barang)
- Jumlah barang pada perintah PESAN dijamin  $\geq 1$
- Perintah BAYAR tidak akan muncul saat list pesanan seorang pembeli kosong
- Nama barang dan pembeli dijamin valid, unik, dan hanya akan terdiri dari satu kata saja.

## Format Keluaran

Kalimat keluaran adalah kalimat yang dicetak tebal (bold)

- **PESAN [NAMA] [BARANG] [JUMLAH]**
  - Saat berhasil: **[NAMA] berhasil memesan [BARANG] sebanyak [JUMLAH] buah**
  - Saat jumlah stok barang kurang: **Tidak bisa memesan [BARANG] sebanyak [JUMLAH] buah. Stok barang tidak cukup**
  - Saat list pesanan melebihi kapasitas: **Tidak bisa memesan [BARANG] sebanyak [JUMLAH] buah. List pesanan [BARANG] melebihi kapasitas**
  - Saat uang tidak cukup: **Tidak bisa memesan [BARANG] sebanyak [JUMLAH] buah. Uang [NAMA] tidak cukup**
- **BAYAR [NAMA]**

**[NAMA] berhasil melakukan pembelian barang dan pembayaran!**

**##### Detail Pembayaran #####**

**[BARANG]1: [HARGA]1 \* [JUMLAH]1 = [TOTAL\_HARGA\_BARANG]1**

**....**

**[BARANG]i: [HARGA]i \* [JUMLAH]i = [TOTAL\_HARGA\_BARANG]i**

---

**Total harga = [TOTAL\_HARGA]**

**Diskon = ([TOTAL\_DISKON])**

**Harga bayar = [HARGA\_YANG\_DIBAYAR]**

**Sisa uang = [SISA\_UANG\_PEMBELI\_SETELAH\_BAYAR]**

**#####**

keterangan:

[BARANG]i → nama barang ke-i

[HARGA]i → harga barang ke-i

[JUMLAH]i → jumlah barang ke-i yang dipesan

[TOTAL\_HARGA\_BARANG]i → hasil dari [HARGA]i dikali dengan [JUMLAH]i

[TOTAL\_HARGA] → total harga dari pesanan pembeli


[TOTAL\_DISKON] → diskon yang didapatkan oleh pembeli

[HARGA\_YANG\_DIBAYAR] → jumlah yang perlu dibayar pembeli  
([TOTAL\_HARGA] - [TOTAL\_DISKON])

[SISA\_UANG\_PEMBELI\_SETELAH\_BAYAR] → sisa uang pembeli setelah membayar

- **RESTOCK [BARANG] [JUMLAH\_BARU]**
  - Saat berhasil: **Berhasil menambahkan stok barang [BARANG]. Sisa stok sekarang = [SISA\_STOK].** [SISA\_STOK] adalah sisa stok dari barang [BARANG]

- Saat nilai [JUMLAH\_BARU] tidak valid: **Maaf, stok tambahan yang dimasukkan tidak valid**
- **DISKON [NAMA]**
  - **[NAMA] mendapatkan diskon sebesar [BESAR\_DISKON]%. [BESAR\_DISKON] adalah persentase diskon yang didapat oleh [NAMA].**

 Untuk melihat format keluaran dengan lebih jelas, kalian bisa melihat keluaran dari *test case* yang diberikan.

## Test Case

### Contoh Input 1

```
1
Aqua 20000 10
1
Udin 150000
7
PESAN Udin Aqua 4
PESAN Udin Aqua 1
PESAN Udin Aqua 3
DISKON Udin
PESAN Udin Aqua 20
BAYAR Udin
RESTOCK Aqua 100
```

### Contoh Output 1

```
Udin berhasil memesan Aqua sebanyak 4 buah
Udin berhasil memesan Aqua sebanyak 1 buah
Tidak bisa memesan Aqua sebanyak 3 buah. Uang Udin tidak cukup
Udin mendapatkan diskon sebesar 5%
Tidak bisa memesan Aqua sebanyak 20 buah. Stok barang tidak cukup
Udin berhasil melakukan pembelian barang dan pembayaran!
##### Detail Pembayaran #####
Aqua: 20000 * 5 = 100000

Total harga = 100000
Diskon = (5000)
Harga bayar = 95000
Sisa uang = 55000
#####
```

Berhasil menambahkan stok barang Aqua. Sisa stok sekarang = 105
---

Penjelasan 1

### Kondisi Awal

Aqua  Harga = 20000 Stok = 10	Udin  Uang = 150000 list pesanan = []
--	--

### PESAN Udin Aqua 4

Aqua  Harga = 20000 Stok = 6	Udin  Uang = 150000 list pesanan = [Aqua 4]
---------------------------------------	--

Udin berhasil memesan 4 Aqua.

### PESAN Udin Aqua 1

Aqua  Harga = 20000 Stok = 5	Udin  Uang = 150000 list pesanan = [Aqua 5]
---------------------------------------	--

Udin berhasil memesan 1 Aqua. Pesanan tersebut dimasukkan ke pesanan yang sudah ada karena Udin sudah pernah memesan Aqua sebelumnya.

### PESAN Udin Aqua 3

Udin tidak berhasil memesan 3 Aqua karena uang Udin tidak cukup.  
 $100000$  (total harga barang yang dipesan Udin) +  $60000$  (total harga pesanan baru Udin) =  $160000 > 150000$ .  
Kondisi barang dan pembeli sama seperti perintah sebelumnya.

### DISKON Udin

Udin mendapatkan diskon 5% karena total barang yang sedang dipesan Udin adalah 5 barang.

### PESAN Udin Aqua 20

Udin tidak berhasil memesan 20 Aqua karena stok Aqua tidak cukup.  
 $20 > 5$  (stok Aqua sekarang).



Kondisi barang dan pembeli sama seperti perintah sebelumnya.

### **BAYAR Udin**

Aqua	Udin
Harga = 20000	Uang = 150000
Stok = 5	list pesanan = []

Udin berhasil membayar dengan rincian:

Total harga = Aqua \* 5 = 20000 \* 5 = 100000

Diskon = 5% \* 100000 = 5000

Harga yang perlu dibayar (harga bayar) = 100000 - 5000 = 95000

Sisa uang Udin = 150000 - 95000 = 55000

### **RESTOCK Aqua 100**

Aqua berhasil di-restock. Sisa stok Aqua sekarang = 5 (stok awal) + 100 (stok baru) = 105.

### Contoh Input 2

```
2
Aqua 10000 10
Indomie 2500 5
2
Udin 60000
Ucok 10000
7
PESAN Udin Aqua 5
RESTOCK Aqua -10
PESAN Udin Indomie 1
PESAN Ucok Aqua 1
DISKON Ucok
BAYAR Udin
BAYAR Ucok
```

### Contoh Output 2

```
Udin berhasil memesan Aqua sebanyak 5 buah
Maaf, stok tambahan yang dimasukkan tidak valid
Udin berhasil memesan Indomie sebanyak 1 buah
Ucok berhasil memesan Aqua sebanyak 1 buah
```

Ucok mendapatkan diskon sebesar 1 %  
 Udin berhasil melakukan pembelian barang dan pembayaran!  
 ##### Detail Pembayaran #####  
 Aqua:  $10000 * 5 = 50000$   
 Indomie:  $2500 * 1 = 2500$

---

Total harga = 52500  
 Diskon = (3150)  
 Harga bayar = 49350  
 Sisa uang = 10650

#####  
 Ucok berhasil melakukan pembelian barang dan pembayaran!  
 ##### Detail Pembayaran #####  
 Aqua:  $10000 * 1 = 10000$

---

Total harga = 10000  
 Diskon = (100)  
 Harga bayar = 9900  
 Sisa uang = 100

#####

## Penjelasan 2

### Kondisi Awal

Aqua  harga = 10000 stok = 10	Indomie  harga = 2500 stok = 5
Udin  uang = 60000 list pesanan=[]	Ucok  uang = 10000 list pesanan = []

### PESAN Udin Aqua 5

Aqua  harga = 10000 stok = 5	Indomie  harga = 2500 stok = 5
Udin	Ucok

uang = 60000 list pesanan=[Aqua 5]	uang = 10000 list pesanan = []
---------------------------------------	-----------------------------------

Udin berhasil memesan 5 Aqua

### RESTOCK Aqua -10

Tidak berhasil menambahkan stok Aqua karena stok tambahan < 1, kondisi barang dan pembeli sama seperti perintah sebelumnya

### PESAN Udin Indomie 1

Aqua  harga = 10000 stok = 5	Indomie  harga = 2500 stok = 4
Udin  uang = 60000 list pesanan=[Aqua 5, Indomie 1]	Ucok  uang = 10000 list pesanan = []

Udin berhasil memesan Indomie sebanyak 1.

### PESAN Ucok Aqua 1

Aqua  harga = 10000 stok = 4	Indomie  harga = 2500 stok = 3
Udin  uang = 60000 list pesanan=[Aqua 5, Indomie 1]	Ucok  uang = 10000 list pesanan = [Aqua 1]

Ucok berhasil memesan Aqua sebanyak 1.

### DISKON Ucok

Diskon yang didapatkan Ucok sebesar 1% karena Ucok hanya baru memesan 1 barang saja.

### BAYAR Udin

Aqua  harga = 10000 stok = 4	Indomie  harga = 2500 stok = 3
---------------------------------------	---

Udin	Ucok
uang = 10650 list pesanan=[]	uang = 10000 list pesanan = [Aqua 1]

Udin berhasil membayar dengan rincian:

Total harga = Aqua \* 5 + Indomie \* 1 = 10000 \* 5 + 2500 \* 1 = 52500

Diskon = 6% \* 52500 = 3150

Harga yang perlu dibayar (harga bayar) = 52500 - 3150 = 49350

Sisa uang Udin = 60000 - 49350 = 10650

## BAYAR Ucok

Aqua	Indomie
harga = 10000 stok = 4	harga = 2500 stok = 3
Udin	Ucok
uang = 10650 list pesanan=[]	uang = 10000 list pesanan = [Aqua 1]

Ucok berhasil membayar dengan rincian:

Total harga = Aqua \* 1 = 10000 \* 1 = 10000

Diskon = 1% \* 10000 = 100

Harga yang perlu dibayar (harga bayar) = 10000 - 100 = 9900

Sisa uang Udin = 10000 - 9900 = 100

## Komponen Penilaian

- 80% Kebenaran hasil *output* program
- 10% Dokumentasi
- 10% Standar penulisan kode sesuai dengan [Code Conventions for the Java Programming Language](#)

## Keterangan Tambahan

- Disarankan menggunakan *template* yang sudah disediakan pada [link](#) ini. Akan tetapi, kalian boleh membuat semua class dari awal asalkan sesuai ketentuan (setiap class berada dalam satu file .java berbeda dengan nama sesuai ketentuan dan memiliki attribute dan method yang sudah diberikan pada spesifikasi program).
- Diperbolehkan untuk mengubah *return type* dan *parameter type* dari semua method.

- Diperbolehkan untuk menambahkan method/attribute baru, tetapi **tidak diperbolehkan** untuk **mengurangi method/attribute yang sudah ada di *template***.
- Kalian **disarankan untuk mencetak keluaran program dengan menggunakan attribute *out*** (Cara penggunaan: `out.print()`, `out.printf()`, dan `out.println()`) yang ada di class **Diskonpedia** supaya program akan melakukan pencetakan keluaran program setelah baris masukan program selesai. Akan tetapi, kalian boleh menggunakan `System.out.print()` seperti biasa. Namun, **kalian harus konsisten** (Tidak boleh sebagian pakai attribute *out* (`out.print()`) sebagian lagi `System.out.print()`).
- **Boleh menggunakan ArrayList**. Kalian boleh mengganti semua implementasi *array primitive* dengan `ArrayList`, tetapi tetap semua attribute dan method yang sudah terimplementasi di *template* menggunakan implementasi *array primitive*

### Pengumpulan Berkas

Kumpulkan **4 berkas .java** (**Diskopedia.java**, **Barang.java**, **Pesanan.java**, **Pembeli.java**) yang telah di-zip dengan format penamaan seperti berikut.  
**[KodeAsdos]\_[Kelas]\_[NPM]\_[NamaLengkap]\_Lab04.zip**

Contoh:

**DDP\_A\_1234567890\_DekDePe\_Lab04.zip**