

Algoritma Struktur Data

Modul 03 Struct

Supermarket Dek Puti



Deskripsi

Dek Puti memiliki supermarket yang terkenal dan ramai pengunjung. Dek Puti senang dengan pencapaiannya, tetapi semakin hari ia semakin kewalahan

mengurus supermarketnya. Akhirnya, Dek Puti meminta mahasiswa Fatika(fakultas informatika) untuk mengerjakan *project* agar supermarketnya lebih efisien.

Setiap pelanggan dapat melakukan fungsi berikut:

- **ADD K B S**

- Fungsi ini menambahkan Order dengan barang B ke dalam keranjang pelanggan K sebanyak S buah.
- Pelanggan K tidak dapat mengambil barang B jika Stock kurang.
- Apabila Stock yang ingin ditambahkan melebihi berat maksimum keranjang (5000), maka banyak barang yang dapat masuk adalah semaksimal mungkin tetapi tidak melebihi kapasitas maksimum keranjang.
 - Misalnya total berat barang di keranjang saat ini adalah 1000, lalu ingin menambahkan barang B sebanyak 2 buah dengan berat masing-masing 4000. Barang baru yang akan masuk hanya 1.

- **TOTAL_HARGA K**

- Fungsi ini akan mengembalikan total harga belanjaan yang ada di keranjang K saat ini.

- **KASIR K**

- Fungsi ini untuk memproses pembelian, berarti semua yang di keranjang pelanggan K sekarang akan **direset** (karena sudah dibeli) dan uang dari K juga akan **berkurang** sebanyak total harga belanjaanyang dia bawa di keranjang.

- **CEK_UANG K**

- Fungsi ini akan mengembalikan uang yang dimiliki pelanggan K saat ini.

Spesifikasi

Akan ada empat file untuk setiap struct , yaitu:

1. Barang

Struct Barang memiliki 4 atribut, yaitu nama (String), harga (int), beratBarang (int), dan stock (int). Tugas Anda adalah melengkapi TODO dalam method di struct ini yang termasuk membuat constructor dan method setter/getter apabila diperlukan. Constructor akan menerima parameter nama, harga, beratBarang, dan stock. **(NOTE: Atribut struct harus memiliki visibility private)**

Struct Barang memiliki beberapa method:

- boolean cekStock(int stock)
 - Method ini akan mengecek stock yang akan ditambahkan ke dalam keranjang masih tersedia atau tidak
 - Method ini dipanggil di dalam method addBarang pada struct Pelanggan.

2. Pelanggan

Struct Pelanggan secara default memiliki 4 atribut, yaitu nama (String), uang (int), keranjang (Order[]), dan kapasitasKeranjang (int). Tugas Anda adalah melengkapi TODO dalam struct ini yang termasuk membuat constructor dan method setter/getter apabila diperlukan. Constructor akan menerima parameter nama, uang, dan banyak barang yang terdaftar di supermarket. Perlu diperhatikan bahwa **keranjang tidak dapat menampung berat total barang lebih dari 5000 (NOTE: Atribut struct harus memiliki visibility private)**

Struct Pelanggan memiliki beberapa method:

- String addBarang(Barang barang, int banyakBarang)
 - Method ini akan menambahkan barang ke keranjang dengan membuat instance Order baru apabila belum ada Order dengan barang yang sama dalam keranjang. Jika sudah ada Order dengan barang yang sama dengan yang ingin ditambahkan, maka banyak barang dari Order tersebut akan ditambahkan. Barang tidak akan bisa diambil apabila **stock-nya sudah habis atau melebihi berat maksimum keranjang** (manfaatkan method di struct Barang).
- int totalHargaBarang()

- Method ini akan mengembalikan total harga barang yang ada di keranjang.
- String cekUang()
 - Method ini akan mengecek uang yang dimiliki oleh seorang pelanggan saat ini.

3. Kasir

Untuk memeriksa kebenaran program Anda, silahkan jalankan *struct* Kasir. Namun sebelum itu lengkapi dulu **//TODO** yang ada. Tersedia template untuk menerima input user. Format output harus sama dengan contoh.

Struct Kasir memiliki satu method:

- static void kasir(Pelanggan K)
 - Method ini untuk memproses pembelian

4. Order

Struct Order secara default memiliki 2 atribut, yaitu barang (Barang) dan banyakBarang (int). Struct ini akan digunakan untuk menyimpan barang dan banyaknya barang yang akan dimasukkan ke dalam keranjang. InstanceOrder baru akan ditambahkan ke dalam array keranjang apabila belum ada Order dengan barang yang sama. Jika barang sudah ada, maka hanya perlu menambahkan banyaknya barang (banyakBarang) tersebut. **Lengkapi modifier yang kurang!**

Format Masukan

- Baris pertama berisi N, banyaknya barang yang ada
- N baris berikutnya mendeskripsikan setiap barang (nama, harga, berat, stock)
- Baris berikutnya berisi M, banyaknya pelanggan yang ada
- M baris berikutnya mendeskripsikan setiap pelanggan (nama, uang)
- Baris berikutnya berisi P, banyaknya kejadian
- P baris berikutnya mendeskripsikan setiap kejadian dengan format sesuai fungsinya masing-masing

Batasan

- Dijamin K dan B selalu ada di setiap perintah
- Input dijamin valid dan ada

Format Keluaran

- **ADD K B S:**
 - Apabila berhasil: **K berhasil menambahkan S B**
 - Apabila stock kurang dari yang akan diambil: **Stock B kurang**
 - Apabila barang terlalu berat: **Maaf S B terlalu berat, tetapi T B berhasil ditambahkan.** (Note: T adalah maksimal yang dapat dimasukkan).
- **TOTAL_HARGA K**
 - **Total harga belanjaan K adalah E** (E merupakan total harga semua barang yang ada di keranjang)
- **KASIR K**
 - Apabila tidak ada barang di keranjang: **Maaf tidak ada barang di keranjang K**
 - Apabila uang pelanggan tidak mencukupi: **Maaf K tidak memiliki cukup uang**
 - Apabila pembelian berhasil:
Pembelian K berhasil:
 - * **$B_i S_i = C_i$** (C adalah harga barang dikali jumlah barang yang ada di keranjang)
 - * ...
 - * ...
 - * **Total Belanjaan = E** (E merupakan Total dari C_i)
 - * **Sisa Uang = D** (D adalah uang K sekarang dikurangi Total Belanjaan)
- **CEK_UANG K**
 - **Uang K sekarang U** (U adalah uang pelanggan K sekarang)

Note: Keluaran adalah kalimat dengan format tebal (***bold***)

Contoh input dan output program ketika dijalankan

Contoh Masukan 1:

```
3
Akua 3000 100 4
Indiemie 3500 250 5
Minyak 15000 1000 10
2
Udin 50000
Bambang 20000
9
ADD Udin Akua 3
ADD Udin Minyak 5
TOTAL_HARGA Udin
ADD Bambang Akua 2
ADD Bambang Akua 1
ADD Bambang Indiemie 5
TOTAL_HARGA Bambang
ADD Bambang Minyak 5
ADD Bambang Indiemie 5
```

Contoh Keluaran 1:

```
Udin berhasil menambahkan 3 Akua
Maaf 5 Minyak terlalu berat, tetapi 4 Minyak berhasil ditambahkan
Total harga belanjaan Udin adalah 69000
Stock Akua kurang
Bambang berhasil menambahkan 1 Akua
Bambang berhasil menambahkan 5 Indiemie
Total harga belanjaan Bambang adalah 20500
Maaf 5 Minyak terlalu berat, tetapi 3 Minyak berhasil ditambahkan
Stock Indiemie kurang
```

Penjelasan 1:

Note: Isi keranjang di penjelasan merupakan representasi dari object Order

Kondisi Awal

Keranjang Udin Uang Udin = 50000 Kapasitas keranjang = 5000 Isi keranjang = []	Keranjang Bambang Uang Bambang = 20000 Kapasitas keranjang = 5000 Isi keranjang = []
---	---

ADD Udin Akua 3

Keranjang Udin Uang Udin = 50000 Kapasitas keranjang = 4700 Isi keranjang = [(Akua, 3)]	Keranjang Bambang Uang Bambang = 20000 Kapasitas keranjang = 5000 Isi keranjang = []
--	---

ADD Udin Minyak 5

Keranjang Udin Uang Udin = 50000 Kapasitas keranjang = 700 Isi keranjang = [(Akua, 3), (Minyak, 4)]	Keranjang Bambang Uang Bambang = 20000 Kapasitas keranjang = 5000 Isi keranjang = []
--	---

Hanya 4 minyak yang berhasil ditambahkan ke keranjang Udin.

TOTAL_HARGA Udin

Total harga pada keranjang Udin = $(3 * \text{harga_Akua}) + (4 * \text{harga_Minyak})$
 $= (3 * 3000) + (4 * 15000) = 69000$

ADD Bambang Akua 2

Keranjang Udin Uang Udin = 50000 Kapasitas keranjang = 700 Isi keranjang = [(Akua, 3), (Minyak, 4)]	Keranjang Bambang Uang Bambang = 20000 Kapasitas keranjang = 5000 Isi keranjang = []
--	---

Stock Akua kurang sehingga tidak bisa masuk keranjang

ADD Bambang Akua 1

Keranjang Udin Uang Udin = 50000 Kapasitas keranjang = 700 Isi keranjang = [(Akua, 3), (Minyak, 4)]	Keranjang Bambang Uang Bambang = 20000 Kapasitas keranjang = 4900 Isi keranjang = [(Akua, 1)]
--	--

Berhasil menambahkan 1 Akua ke keranjang Bambang.

ADD Bambang Indiemie 5

Keranjang Udin	Keranjang Bambang
Uang Udin = 50000	Uang Bambang = 20000
Kapasitas keranjang = 700	Kapasitas keranjang = 3650
Isi keranjang = [(Akua, 3), (Minyak, 4)]	Isi keranjang = [(Akua, 1), (Indiemie, 5)]

Berhasil menambahkan 5 Indiemie ke keranjang Bambang.

TOTAL_HARGA Bambang

Total harga pada keranjang Udin = $(1 * \text{harga_Akua}) + (5 * \text{harga_Indiemie})$
 $= (1 * 3000) + (5 * 3500) = 20500$

ADD Bambang Minyak 5

Keranjang Udin	Keranjang Bambang
Uang Udin = 50000	Uang Bambang = 20000
Kapasitas keranjang = 700	Kapasitas keranjang = 650
Isi keranjang = [(Akua, 3), (Minyak, 4)]	Isi keranjang = [(Akua, 1), (Indiemie, 5), (Minyak, 3)]

Hanya 3 minyak yang berhasil ditambahkan, karena 5 minyak terlalu berat (melebihi kapasitas berat keranjang).

ADD Bambang Indiemie 5

Keranjang Udin	Keranjang Bambang
Uang Udin = 50000	Uang Bambang = 20000
Kapasitas keranjang = 700	Kapasitas keranjang = 650
Isi keranjang = [(Akua, 3), (Minyak, 4)]	Isi keranjang = [(Akua, 1), (Indiemie, 5), (Minyak, 3)]

Stock Indiemie sudah habis sehingga tidak ada yang bisa ditambahkan ke keranjang Bambang.

Contoh Masukan 2:

4
Akua 3000 100 4
Indiemie 3500 250 5
Minyak 15000 1000 10
Gula 10000 500 4
2
Udin 90000
Bambang 30000
14
ADD Udin Akua 3
ADD Udin Minyak 3
ADD Udin Gula 5
TOTAL_HARGA Udin
KASIR Udin
KASIR Udin
TOTAL_HARGA Udin
ADD Bambang Akua 1
ADD Bambang Indiemie 5
KASIR Bambang
ADD Bambang Minyak 1
KASIR Bambang
CEK_UANG Bambang
TOTAL_HARGA Bambang

Contoh Keluaran 2:

Udin berhasil menambahkan 3 Akua
Udin berhasil menambahkan 3 Minyak
Stock Gula kurang
Total harga belanjaan Udin adalah 54000
Pembelian Udin berhasil:
* Akua 3 = 9000
* Minyak 3 = 45000
* Total Belanjaan = 54000
* Sisa Uang = 36000
Maaf tidak ada barang di keranjang Udin
Total harga belanjaan Udin adalah 0
Bambang berhasil menambahkan 1 Akua
Bambang berhasil menambahkan 5 Indiemie

Pembelian Bambang berhasil:

* Akua 1 = 3000

* Indiemie 5 = 17500

* Total Belanjaan = 20500

* Sisa Uang = 9500

Bambang berhasil menambahkan 1 Minyak

Maaf Bambang tidak memiliki cukup uang

Uang Bambang sekarang 9500

Total harga belanjaan Bambang adalah 15000

Penjelasan 2:

Kondisi Awal

Keranjang Udin	Keranjang Bambang
Uang Udin = 90000	Uang Bambang = 30000
Kapasitas keranjang = 5000	Kapasitas keranjang = 5000
Isi keranjang = []	Isi keranjang = []

ADD Udin Akua 3

Keranjang Udin	Keranjang Bambang
Uang Udin = 90000	Uang Bambang = 30000
Kapasitas keranjang = 4700	Kapasitas keranjang = 5000
Isi keranjang = [(Akua, 3)]	Isi keranjang = []

Berhasil menambahkan 3 Akua.

ADD Udin Minyak 3

Keranjang Udin	Keranjang Bambang
Uang Udin = 90000	Uang Bambang = 30000
Kapasitas keranjang = 1700	Kapasitas keranjang = 5000
Isi keranjang = [(Akua, 3), (Minyak, 3)]	Isi keranjang = []

Berhasil menambahkan 3 Minyak.

ADD Udin Gula 5

Keranjang Udin	Keranjang Bambang
Uang Udin = 90000	Uang Bambang = 30000
Kapasitas keranjang = 1700	Kapasitas keranjang = 5000
Isi keranjang = [(Akua, 3), (Minyak, 3)]	Isi keranjang = []

Tidak ada perubahan karena Gula gagal masuk

TOTAL_HARGA Udin

Total harga pada keranjang Udin = $(3 * \text{harga_Akua}) + (3 * \text{harga_Minyak}) = (3*3000) + (3*15000) = 54000$

KASIR Udin

Keranjang Udin	Keranjang Bambang
Uang Udin = 36000	Uang Bambang = 30000
Kapasitas keranjang = 5000	Kapasitas keranjang = 5000
Isi keranjang = []	Isi keranjang = []

Uang Udin berkurang karena dipakai untuk membayar barang yang dibelinya, dan keranjang Uding dikosongkan.

KASIR Udin

Keranjang Udin	Keranjang Bambang
Uang Udin = 36000	Uang Bambang = 30000
Kapasitas keranjang = 5000	Kapasitas keranjang = 5000
Isi keranjang = []	Isi keranjang = []

Tidak ada perubahan karena Udin tidak sedang membawa apa-apa.

TOTAL_HARGA Udin

Total harga pada keranjang Udin = 0

ADD Bambang Akua 1

Keranjang Udin	Keranjang Bambang
Uang Udin = 36000	Uang Bambang = 30000
Kapasitas keranjang = 5000	Kapasitas keranjang = 4900
Isi keranjang = []	Isi keranjang = [(Akua, 1)]

Berhasil menambahkan 1 Akua ke keranjang Bambang.

ADD Bambang Indiemie 5

Keranjang Udin	Keranjang Bambang
Uang Udin = 36000	Uang Bambang = 30000
Kapasitas keranjang = 5000	Kapasitas keranjang = 3650
Isi keranjang = []	Isi keranjang = [(Akua, 1), (Indiemie, 5)]

Berhasil menambahkan 5 Indiemie ke keranjang Bambang.

KASIR Bambang

Keranjang Udin Uang Udin = 36000 Kapasitas keranjang = 5000 Isi keranjang = []	Keranjang Bambang Uang Bambang = 9500 Kapasitas keranjang = 5000 Isi keranjang = []
---	--

Uang Bambang berkurang untuk bayar, dan keranjang Bambang dikosongkan.

ADD Bambang Minyak 1

Keranjang Udin Uang Udin = 36000 Kapasitas keranjang = 5000 Isi keranjang = []	Keranjang Bambang Uang Bambang = 9500 Kapasitas keranjang = 4000 Isi keranjang = [(Minyak, 1)]
---	---

Berhasil menambahkan 1 Minyak ke keranjang Bambang.

KASIR Bambang

Keranjang Udin Uang Udin = 36000 Kapasitas keranjang = 5000 Isi keranjang = []	Keranjang Bambang Uang Bambang = 9500 Kapasitas keranjang = 4000 Isi keranjang = [(Minyak, 1)]
---	---

Tidak ada perubahan karena uang Bambang tidak cukup.

CEK_UANG Bambang

Uang Bambang sekarang 9500

TOTAL_HARGA Bambang

Total harga pada keranjang Bambang = $(1 * \text{harga_Minyak}) = 15000$

Komponen Penilaian

- 40% Kebenaran kode
- 40% Implementasi struct Pelanggan, Barang, dan Kasir
- 10% Dokumentasi
- 10% Standar penulisan kode*

*Standar penulisan kode yang harus dipenuhi yaitu:

1. Penamaan Module, Struct , Method, dan Variabel yang tidak ambigu

Keterangan Tambahan

- Diperbolehkan menambahkan atribut dan method pada setiap struct sesuai kebutuhan masing-masing.
- Return type dari setiap method yang telah disediakan dapat diubah-ubah sesuai dengan kebutuhan kalian.
- **Boleh menggunakan vector!** Untuk detail vector, bisa dilihat dari [Vector in C++ STL - GeeksforGeeks](#)

Template dapat diakses di sini:

https://telkomuniversityofficial-my.sharepoint.com/:u:/g/personal/ammarpvl_student_telkomuniversity_ac_id/Ef5ECX-YIhBGhEhWx4PIXJIBgh-C_syxMqijgj5KCb5OMg?e=1sPUjA

Kumpulkan berkas .cpp + screenshot hasil compile yang benar (bisa dilihat dicontoh) yang telah di-zip dengan format penamaan seperti berikut.

[Kelas]_[NIM]_[NamaLengkap]_Modul03.zip

Contoh:

S1IT-KJ-23-001_1234567890_Diddy_Modul03.zip