**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 4**

**Object Oriented Programming Java 1**

Logo

Description automatically generated with medium confidence

**Disusun Oleh:**

**Gerald Eberhard**

**(105223002)**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN KOMPUTER**

**UNIVERSITAS PERTAMINA**

**2025**

1. **Pendahuluan**

Studi kasus praktikum ini adalah pembuatan sistem transaksi penjualan sederhana untuk sebuah toko. Sistem ini dirancang untuk mengelola data pelanggan, produk, dan transaksi penjualan menggunakan konsep Pemrograman Berorientasi Objek (PBO) dalam bahasa Java. Setiap transaksi mencakup informasi pelanggan (nama dan ID), produk (nama produk dan kode produk), dan detail transaksi (jumlah pembelian dan total harga). Program ini harus mampu memvalidasi input, memperbarui stok produk setelah transaksi, dan menampilkan laporan transaksi secara terstruktur.

Codingan ini terdiri dari empat kelas yaitu Main.java, Produk.java, Pelanggan.java, dan transaksi.java. Kelas Main.java akan berisi pemanggilan dan pembuatan objek serta pemanggilan perintah-perintah yang terdapat pada kelas produk, pelanggan, dan transaksi. Kemudian ada juga kelas Produk.java yang berisi informasi mengenai method-method dan informasi-informasi yang akan disimpan terkait produk. Kemudian ada juga kelas Pelanggan.java yang berisi informasi mengenai method-method dan informasi yang akan disimpan terkait pelanggan. Kemudian yang terakhir ada kelas transaksi yang bertujuan untuk menyimpan semua informasi terkait transaksi dan operasi-operasi yang akan dilakukan dalam transaksi.

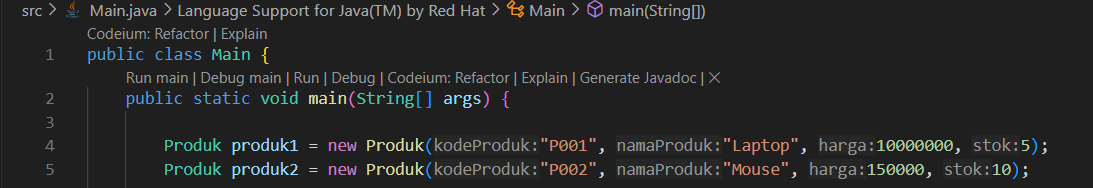
Pembuatan codingan ini dilandaskan berdasarkan materi yang telah dipelajari selama praktikum. Kemudian karena tidak ada detail spesifik tentang harga produk atau aturan bisnis lainnya dalam modul, maka saya akan membuat asumsi sendiri untuk harga produk dan jumlah stok.

1. **Variabel**

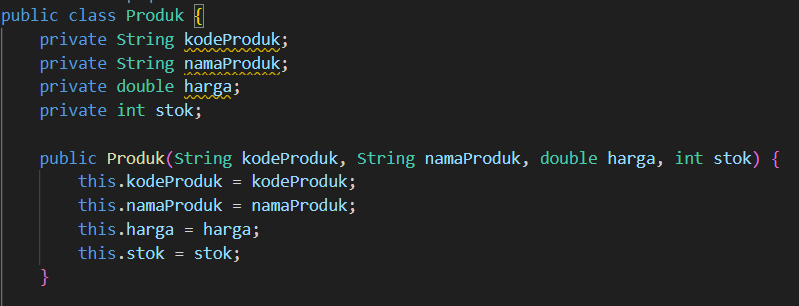
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Variabel** | **Tipe Data** | **Fungsi** |
| 1 | idProduk | String | Menyimpan ID unik produk dan terdapat di kelas Produk. |
| 2 | namaProduk | String | Menyimpan nama produk dan variabel ini terdapat di kelas Produk. |
| 3 | harga | double | Menyimpan harga per unit produk dan variabel ini terdapat di kelas Produk. |
| 4 | stok | Int | Menyimpan jumlah stok produk dan variabel ini terdapat di kelas Produk. |
| 5 | idPelanggan | String | Menyimpan kode identitas pelanggan. Terdapat di dalam kelas Pelanggan. |
| 6 | nama | String | Menyimpan nama pelanggan. Terdapat di dalam kelas pelanggan. |
| 7 | email | String | Menyimpan alamat email pelanggan. Terdapat di dalam kelas Pelanggan. |
| 8 | saldo | double | Menyimpan jumlah saldo yang dimiliki pelanggan dan variabel ini terdapat di kelas Pelanggan. |
| 9 | idTransaksi | String | Menyimpan kode identfikasi transaksi yang terdapat di dalam kelas Transaksi |
| 10 | pelanggan | Pelanggan (Kelas) | Menyimpan informasi terkait pelanggan yang melakukan transaksi. Variabel ini terdapat di dalam kelas Transaksi |
| 11 | Produk | Produk (Kelas) | Menyimpan informasi terkait produk yang dibeli oleh pelanggan. Variabel ini terdapat di dalam kelas Transaksi |
| 12 | jumlahBeli | Int | Menyimpan informasi mengenai jumlah barang yang dibeli. Variabel ini terdapat di dalam kelas Transaksi |
| 13 | totalHarga | Double | Menyimpan informasi mengenai jumlah total harga yang harus dubayarkan oleh pelanggan. Variabel ini terdapat di dalam kelas Transaksi. |
| 14 | Produk1 | Produk | Ini adalah variabel yang akan menampun nilai object product ke - 1 |
| 15 | Produk2 | Produk | Ini adalah variabel yang akan menampun nilai object product ke - 2 |
| 16 | pelanggan1 | Pelanggan | Ini adalah variabel yang akan menampun nilai object pelanggan ke - 1 |
| 17 | pelanggan2 | Pelanggan | Ini adalah variabel yang akan menampun nilai object pelanggan ke - 2 |
| 18 | transaksi1 | Transaksi | Ini adalah variabel yang akan menampun nilai object transaksi ke - 1 |
| 19 | transaksi2 | Transaksi | Ini adalah variabel yang akan menampun nilai object transaksi ke - 2 |
| 20 | transaksi3 | Transaksi | Ini adalah variabel yang akan menampun nilai object transaksi ke - 3 |
| 21 | jumlah | int | Menyimpan nilai jumlah barang yang telah dibeli pelanggan terdapat di dalam kelas Produk. |
| 22 | jumlah | double | Menyimpan nilai saldo yang ingin dikurangkan dari saldo rekening pelanggan terdapat di dalam kelas Pelanggan. |
| 23 | jumlah | double | Menyimpan nilai saldo yang ingin ditambahkan pada saldo rekening pelanggan terdapaat di dalam kelas Pelanggan. |

1. **Constructor dan Method**

|  |  |  |
| --- | --- | --- |
| **Nama Method** | **Jenis Method** | **Fungsi** |
| Pelanggan | Constructor | Untuk membuat objek pelanggan |
| TampilkanInfoPelanggan | Method/Proedur | Untuk menampilkan informasi terkait pelanggan |
| topUpSaldo | Method/Proedur | Untuk menambahkan saldo pelanggan |
| kurangiSaldo | Fungsi | Ini adalah fungsi untuk mengurangi saldo pelanggan dan mengembalikan nilai boolean |
| getSaldo | Fungsi | Ini adalah fungsi untuk mengembalikan nilai saldo pelanggan. |
| getIdPelanggan | Fungsi | Ini adalah fungsi untuk mengembalikan nilai ID pelanggan. |
| getNama | Fungsi | Ini adalah fungsi untuk mengembalikan nilai nama pelanggan. |
| Transaksi | Constructor | Untuk membuat objek transaksi |
| prosesTransaksi | Fungsi | Ini adalah fungsi untuk melakukan validasi apakah transaksinya dapat dilakukan dengan cara memanggil method kurangiStok dan mengembalikan nilai boolean. |
| tampilkanDetailTransaksi | Method/Proedur | Method ini bertujuan untuk mengeluarkan informasi detail terkait transaksi. Mirip dengan nota atau struk pembelian sederhana. |
| Produk | Constructor | Untuk membuat objek produk |
| tampilkanInfoProduk | Method | Method ini bertujuan untuk menampilkan indormasi produk |
| kurangiStok | Function | Fungsi ini bertujuan untuk mengurangi jumlah stok barang kemudian mengembalikan nilai boolean |
| getHarga | Function | Untuk mengembalikan harga produk |
| getStok | Function | Untuk mengembalikan jumlah stok produk |
| getKodeProduk | Function | Untuk mengembalikan informasi kode produk |
| getNama | Function | Untuk mengembalikan informasi nama produk |
| main | Method | Sebagai tempat dimana objek dibuat, dipanggil dan sebagainya. |

1. **Dokumentasi dan Pembahasan Code  
   **

Pada awal mula codingan pada kelas main. Akan dibuat objek produk dengan memanfaatkan konstruktor dari kelas produk berikut:



Kemudian setelah itu dengan memanfaatkan method tampilkanInfoProduk pada kelas produk berikut ini:

Sebuah gambar berisi teks, cuplikan layar, Font

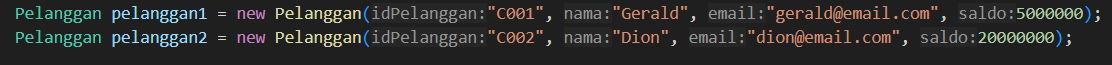
Konten yang dihasilkan AI mungkin salah.

Pada kelas main akan dijalankan perintah berikut untuk menampilkan informasi produk pada object produk1 dan produk2 sebagai berikut:

Sebuah gambar berisi teks, Font, cuplikan layar

Konten yang dihasilkan AI mungkin salah.

Kemudian selanjutnya kelas main akan memanggil contructor dari kelas pelanggan untuk membuat object pelanggan1 dan pelanggan2 sebagai berikut:



Hal ini dapat dilakukan dengan memanfaatkan method pelanggan yang telah di buat pada kelas pelanggan sebagai berikut:

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Kemudian dengan memanfaatkan method tampilkanInfoPelanggan yang telah dibuat sebagaimana telampir pada gambar di bawah ini maka program main akan memanggil program tersebut untuk menampilkan informasi terkait dengan pelanggan.

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Berikut ini pemanggilan method ini yang dilakukan pada kelas main:

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Konten yang dihasilkan AI mungkin salah.

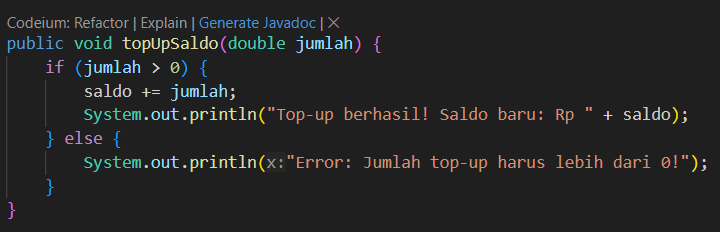
Kemudian saya merasa bahwa saldo yang dimiliki oleh objek pelanggan1 sepertinya terlalu sedikit oleh karena itu saya ingin menambahkan jumlah saldonya dengan memanggil method topUpSaldo yang telah saya buat pada kelas pelanggan. Agar dapat melakukan hal ini saya pun menuliskan bari codingan seperti ini pada kelas main:

Sebuah gambar berisi teks, Font, cuplikan layar, Grafis

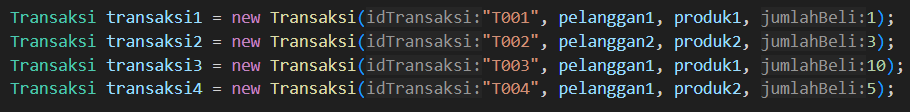
Konten yang dihasilkan AI mungkin salah.

oia tidak lupa juga saya selalu menggunakan System.out.println() untuk menambahkan bari baru sehingga hasil output pada terminal akan terlihat lebih rapih karena ada jaraknya. Begitu juga saya lakukan pada saat melakukan pemanggilan informasi pada object-object produk dan object-object pelanggan.

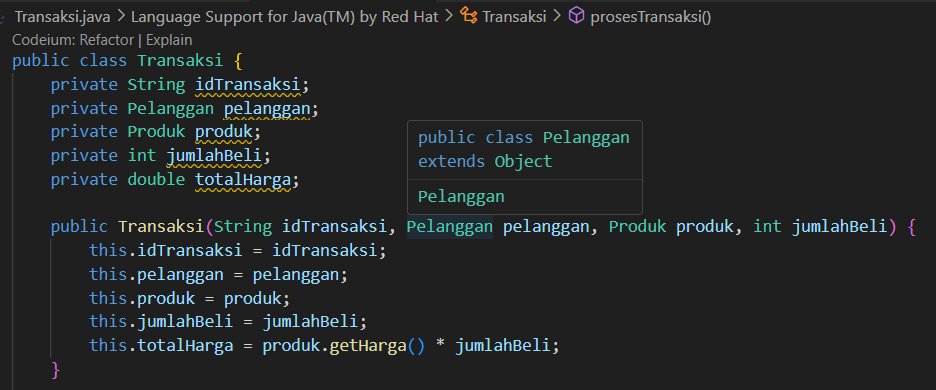
Berikut ini adalah method topUpSaldo yang digunakan pada kelas pelanggan:



Kemudian program berlanjut, setelah menambahkan saldo pada pelanggan 1 saya ingin memulai transaksi oleh karena itu langkah berikut yang akan saya lakukan adalah saya akan membuat object transaksi dengan menggunakan constructor yang telah dibuat pada kelas transaksi sebagai berikut:



Method Transaksi yang telah saya buat pada kelas Transaksi adalah sebagai berikut:



Codingan pun berlanjut kembali pada kelas main, saya akan melanjutkan proses transaksi. Berbagai method dari kelas transaksi pun dipanggil di dalam kelas main sebagai berikut:

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Object transaksi1, transaksi2, transaksi3, dan transaksi4 pun akan dibuat. Disini untuk melakukan proses transaksi pada setiap object setiap object akan ditransaksikan di dalam method prosesTransaksi sebagai berikut:   
Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Method proses transaksi ini sendiri merupakan fungsi yang akan melakukan operasi operasi mengurangi stok barang karena barang tersebut akan diambil atau istilahnya sebelum barang ini diserahkan ke penjual barang itu pasti akan dikeluarkan dari inventory terlebih dahulu. Misalnya ada pembeli datang mau beli pistol maka pasti penjual akan mengeluarkan beberapa koleksi pistol dari displaynya. Nah inilah yang saya asumsikan sehingga akan dijalan produk.kurangiStok. Namun disini saya menggunakan tanda seru hal ini berhubungan dengan konsep dari method kurangiStok yang telah saya buat pada kelas produk sebagai berikut:

Sebuah gambar berisi teks, cuplikan layar, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Okay demikianlah logic pada method kurangi stok kemudian otomatis apabila barang tersebut masih tersedia maka program pada struktur if ajan dijalankan dan otomatis stok akan dikurangi dengan jumlah barang yang dibeli.Namun, apabila jumlah stok tidak mencukupi maka akan muncur pesan eror stok tidak mencukupi.

Jika demikian kita akan kembali lagi ke method prosesTransaksi pada kelas Transaksi Berikut:

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Nah jangan bingung pada bagian ini ya kok objek yang di validasi adalah object pelanggan tapi kok ketika validasinya bernilai false program malah mengurangi stok. Nah teman-teman ingat bahwa tadi kan pistolnya sudah dikeluarkan untuk ditunjukkan kepada pembeli. Namun apa yang harus dilakukan kalo seandainya pembeli tidak jadi membeli produk kita? Yap benar produk tersebut akan disimpan kembali. Karena sebelumnya logik dari kurangiStok adalah stok-jumlah. Maka sekarang agar mengembalikan jumlah stok ke jumlah semula maka kita harus menambahkan stok dengan jumlah. Hal ini dapat dilakukan dengan mengubah jumlah menjadi minus sehingga operasinya menjadi stok – (- jumlah). Oke kalau hal ini sudah jelas. Saya akan menjelaskan bagian fungsi untuk validasi yaitu fungsi kurangiSaldo dari kelas pelanggan. Programnya itu sebagai berikut:

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Mirip seperti fungsi mengurangi stok demikian juga apabila barang tersebut terjual maka program akan mengurangi jumlah saldo yang dimiliki oleh object pelanggan. Tapi bagaimana jika saldo yang dimiliki oleh pelanggan tidak cukup. Maka akan muncul pesan eror yang menandakan saldonya tidak cukup.

Kemudian program akan berlanjut ke bagian berikut :

Sebuah gambar berisi teks, cuplikan layar, Font

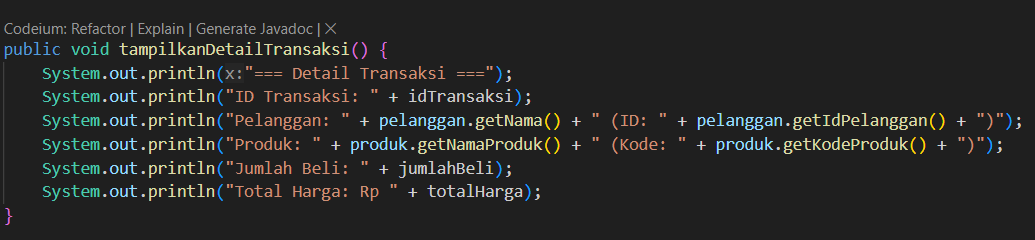
Konten yang dihasilkan AI mungkin salah.

Apabila semuanya lancar-lancar saja maka transaksi berhasil. Akan ada mengeluarkan output string menyatakan bahwa “Transaksi Berhasil”. Kemudian kita akan kembali lagi ke kelas main.

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Apabila proses transaksi berhasil maka dari konsep if yang ada di atas. Maka nanti akan ada dijalankan method tampilkanDetailTransaksi. Method ini terdapat pada kelas Transaksi isinya juga hanya method untuk menampilkan output. Berikut codingannya:



Dalam melakukan proses print nota atau informasi transaksi atau yang juga bisa disebut struk transaksi ini ada digunakan beberapa method yang berfungsi untuk melakukan return nilai dari kelas produk dan pelanggan sebagai berikut:

Dari kelas produk sebagai berikut:

Sebuah gambar berisi teks, cuplikan layar, Font

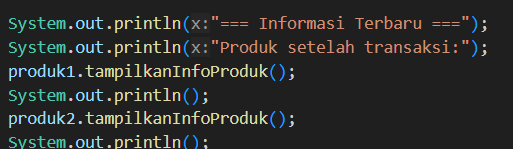
Konten yang dihasilkan AI mungkin salah.

Dari kelas Pelanggan sebagai berikut:

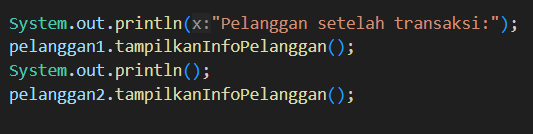
Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Kemudian program di kelas main akan berlanjut dengan line code berikut:



Disini informasi terkait produk akan ditampilkan lagi dengan menggunakan method tampilkanInfoProduk. Line Codingan juga akan berlanjut dengan menampilkan status pelanggan setelah melakukan transaksi sebagai berkut:



Demikian penjelasan codingan saya. Sekian dan terimakasih

1. **Kesimpulan**

Praktikum ini berhasil mengimplementasikan sistem transaksi penjualan sederhana dengan konsep PBO seperti *encapsulation* (melalui atribut private dan getter/setter) dan hubungan antar kelas. Program memenuhi kebutuhan studi kasus dengan mengelola data produk, pelanggan, dan transaksi secara terstruktur. Pengalaman ini sangat membantu untuk memperdalam pemahaman saya tentang penerapan materi PBO dalam menyelesaikan masalah nyata. Sekian dan terima kasih.

1. **Daftar Pustaka**

Modul 4: Obejct Oriented Programming Java 1