**LAPORAN TUGAS PROYEK**

**PEMROGRAMAN BERIORIENTASI OBJEK**

*Minimarket*



**Disusun Oleh :**

|  |  |
| --- | --- |
| 11322045 | Nania Avantika Oligiviana Pangaribuan |
| 11322050 | Elisabeth Uli Tambunan |
| 11322060 | Vanessa Siahaan |
| 11322063 | Hagai Natasha Sianturi |

**Institut Teknologi Del**

**Fakultas Vokasi**

**D-III Teknologi Informasi**

**Laguboti**

**2023/2024**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc153661644)

[DAFTAR GAMBAR 3](#_Toc153661645)

[BAB 1 FUNCTIONAL DESCRIPTION 5](#_Toc153661646)

[1.1 Deskripsi dan Fungsi 5](#_Toc153661647)

[BAB 2 OOP IMPLEMENTATION 6](#_Toc153661648)

[2.1 Pengertian OOP 6](#_Toc153661649)

[2.2 Konsep OOP 6](#_Toc153661650)

[BAB 3 LINK YOUTUBE 11](#_Toc153661651)

# DAFTAR GAMBAR

[Gambar 1. Absraction 6](#_Toc153653262)

[Gambar 2. Encapsulation 7](#_Toc153653263)

[Gambar 3. Inheritance 8](#_Toc153653264)

[Gambar 4. Polymorphism 8](#_Toc153653265)

[Gambar 5. Interface 9](#_Toc153653266)

[Gambar 6. Exception handling 10](#_Toc153653267)

**Instruksi arahan untuk laporan Proyek PBO**

**JUDUL/TOPIK PER KELOMPOK**

Soal berdasarkan Nim anda. Nim % 5 = No tugas anda

Ada 5 topik dengan kasus

1. Minimarket
2. School
3. Hotel
4. Travel
5. Library

Setiap kelompok terdiri dari 4 orang. (NIM 062 pindah ke School, NIM 063 pindah ke Minimarket)

**RUBRIK PROYEK**

1. Beri deskripsi secara singkat ada berapa dan uraian mengenai fungsi yang anda buat dalam proyek ini (Minimal 5 fungsi).
2. Jelaskan penerapan konsep OOP yang anda terapkan pada proyek anda kemudian sertakan screenshot dari sistem yang anda bangun berdasarkan point berikut:
   1. Abstraction
   2. Encapsulation
   3. Inheritance
   4. Polymorphism
   5. Interface
   6. Exception handling
3. Penjelasan hasil implementasi keseluruhan
4. Rekam video penjelasan dari No 1 dan No 2. Upload pada Youtube dan masukkan link tersebut ke dalam laporan ini.
5. Pengumpulan laporan dan *source code* adalah hari Minggu, 17 Desember 2023, 23.59 WIB di E-Course. Template *filename*:

3digitbelakangNIM1\_3digitbelakangNIM2\_3digitbelakangNIM3\_3digitbelakangNIM4-Proyek PBO-Topik.

Contoh: 005\_010\_015\_020-Proyek PBO-Minimarket.zip  
Keterlambatan pengumpulan akan mendapatkan pengurangan poin sebanyak berapa menit anda terlambat.

# BAB 1 FUNCTIONAL DESCRIPTION

## 1.1 Deskripsi dan Fungsi

**1.Fungsi Login**Fungsi login digunakan oleh admin dan kasir untuk dapat masuk dan mengakses aplikasi minimarket.

**2. Fungsi Logout**Fungsi login digunakan oleh admin dan kasir untuk dapat keluar aplikasi minimarket.

**3. Fungsi Menambah Daftar Produk**Fungsi ini digunakan oleh admin untuk menambah produk-produk baru yang akan dijual.

**4. Fungsi Mengedit Daftar Produk**Fungsi ini digunakan oleh admin untuk mengedit data produk yang ada.

**5. Fungsi Menghapus Daftar Produk**Fungsi ini digunakan oleh admin untuk menghapus data produk-produk yang tidak digunakan lagi.

**6. Fungsi Menambah Daftar Kasir**Fungsi ini digunakan oleh admin untuk menambah daftar kasir yang bekerja.

**7. Fungsi Mengedit Daftar Kasir**Fungsi ini digunakan oleh admin untuk mengedit daftar kasir yang bekerja.

**8. Fungsi Menghapus Daftar Kasir**Fungsi ini digunakan oleh admin untuk menghapus daftar kasir yang sudah tidak bekerja lagi.

**9. Fungsi Menambah Transaksi**Fungsi ini digunakan oleh kasir untuk menambah transaksi.

**10. Fungsi Menghapus Transaksi**Fungsi ini digunakan oleh kasir untuk menghapus transaksi yang sudah ditambahkan.

# BAB 2 OOP IMPLEMENTATION

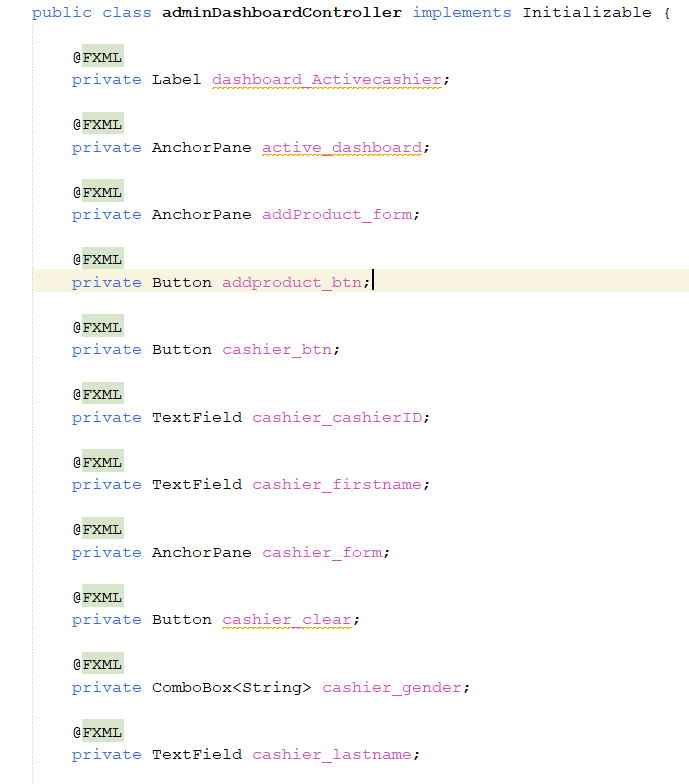
## 2.1 Pengertian OOP

OOP adalah paradigma pemrograman yang menggunakan konsep "objek" untuk merancang dan mengorganisir kode. Paradigma ini berfokus pada pengorganisasian kode ke dalam objek-objek yang merepresentasikan entitas atau konsep di dunia nyata.

## 2.2 Konsep OOP

**a. Abstraction**

Abstraksi adalah class-class yang memiliki informasi abstrak dan metode-metode dari sekumpulan data. Abstract Class tidak bisa di ubah dan berlaku juga sebagai kerangka dalam penciptaan berbagai subclass (berperan seperti Superclass yang di bahas di konsep Inheritance). Kelas adminDashboardController mencerminkan abstraksi dari logika dan tampilan dashboard admin



Gambar 1. Absraction

**b. Encapsulation**

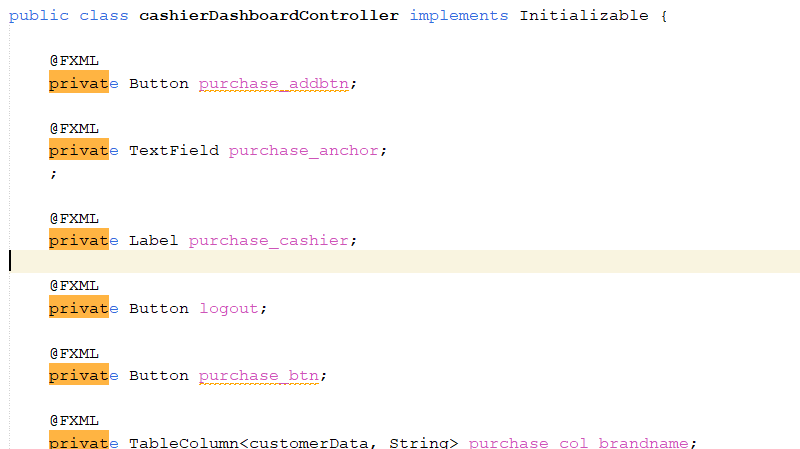
Encapsulasi adalah konsep tentang pengikatan data atau metode berbeda yang di satukan atau “dikapsulkan” menjadi satu unit data. Encapsulation dapat memudahkan dalam pembacaan kode karena informasi yang di sajikan tidak perlu di baca secara rinci dan sudah merupakan satu kesatuan. implementasi konsep encapsulation terutama terlihat dalam penggunaan akses modifier private untuk variabel-variabel kelas dan penggunaan metode-metode publik untuk berinteraksi dengan data tersebut



Gambar 2. Encapsulation

**C. Inheritance**

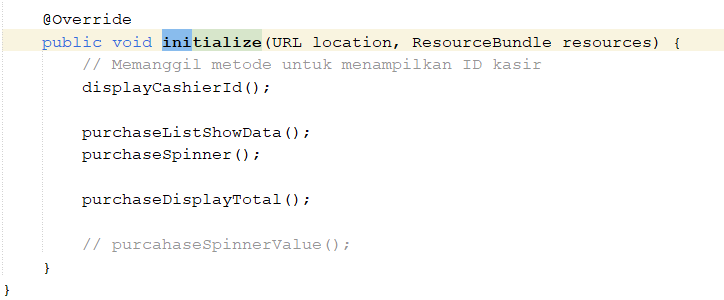
Inheritance adalah konsep di mana sebuah kelas dapat mewarisi properti dan metode dari kelas lain. Ini memungkinkan penggunaan kembali kode dan membentuk hubungan hierarki antar kelas. Inheritance dalam kelas cashierDashboardController terwujud melalui implementasi antarmuka Initializable untuk mewarisi dan mengkustomisasi metode initialize



Gambar 3. Inheritance

**D. Polymorphism**

Polymorphism adalah konsep di mana suatu objek berbeda-beda dapat di akses melalui satu interface. Sebuah objek polymorphic dapat beradaptasi dengan metode apapun yang di implementasikan pada objek tersebut, dan setiap class memiliki interpretasinya tersendiri terhadap interfacenya. Metode initialize di kelas cashierDashboardController menggantikan atau "meng-override" metode dengan nama yang sama dari antarmuka Initializable. Dalam konteks ini, metode initialize dari antarmuka Initializable memberikan fungsi khusus yang akan dijalankan saat suatu objek dibuat.



Gambar 4. Polymorphism

**E. Interface**

Interface adalah kontrak yang menyediakan antarmuka umum untuk kelas-kelas yang mengimplementasikannya. Interface hanya berisi deklarasi metode tanpa implementasi.



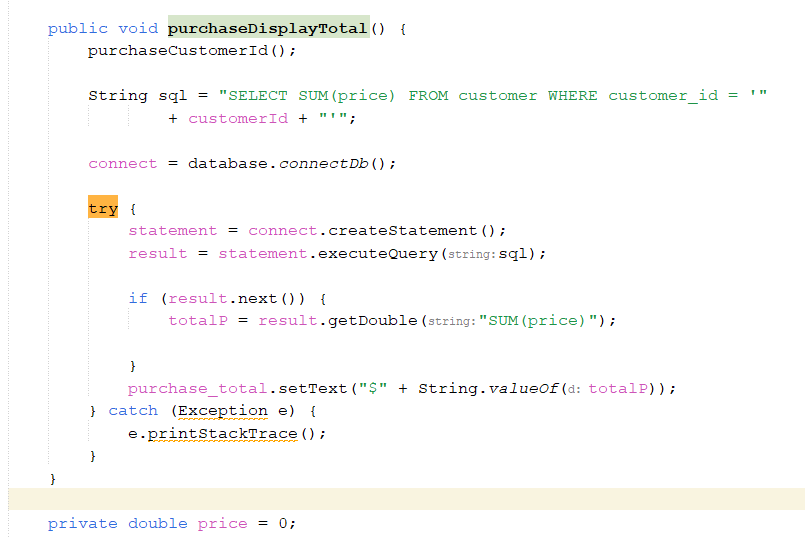
Gambar 5. Interface

**F. Exception handling**

Exception handling adalah suatu penanganan mekanisme penanganan kesalahan. Event yang terjadi ketika program menemui kesalahan saat instruksi program dijalankan, exception digunakan biasanya dalam sumberdaya non memori (untuk menangani masalah yang mirip atau ringan) penggunaan block catch untuk suatu blok try.Exception dalam satu program bisa mengatasi banyak exception. Checked exception hanya dapat ditangani dalam try block atau method yang method yang didesain didesain untuk melempar melempar exception.

* Compiler akan memberitahu jika checked exception tidak ditangani secara tepat **Contoh :** IOException.
* Unchecked exception tidak memerlukan penanganan langsung. Pada saat di compile tidak ada pemberitahuan kesalahan.

**Contoh :** RunTimeException dan turunannya

* Blok finally digunakan untuk mendefinisikan kode program yang selalu di dieksekusi baik ada exception yang terjadi maupun bila tidak terjadi exception sama sekali. Dengan menggunakan blok try-catch, kode dapat menanggapi kondisi yang tidak terduga atau kesalahan yang mungkin terjadi saat aplikasi berjalan, memastikan bahwa aplikasi tetap dapat berjalan dengan mulus meskipun ada masalah.
* 

Gambar 6. Exception handling

# BAB 3 LINK YOUTUBE

Berikut Merupakan Link dari penjelasan kami terkait tugas proyek Minimarket:

<https://youtu.be/4ysmBrFcje0>