**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**PACKAGE**

Logo

Description automatically generated with medium confidence

**Disusun Oleh:**

**Rayhan Surya Destian (105222024)**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN KOMPUTER**

**PEMROGRAMAN BERORIENTASI OBJEK**

**UNIVERSITAS PERTAMINA**

**2023/2024**

1. **Pendahuluan**

Dalam tugas ini, tujuan utama adalah untuk membuat sebuah sistem pemesanan makanan yang melibatkan beberapa modul dalam package yang berbeda. Sistem ini mengimplementasikan konsep pemrograman berorientasi objek dan penggunaan access modifiers untuk mengontrol akses terhadap data secara ketat. Sistem dibagi menjadi tiga package utama: menu, order, dan util, masing-masing menyimpan kelas yang sesuai dengan fungsi spesifik dalam sistem.

1. **Variabel**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Variabel | Tipe data | Fungsi |
| 1 | itemCount | int (static) | Menghitung item menu yang telah dibuat |
| 2 | orderCount | int (static) | Menghitung jumlah pesanan yang telah dibuat |
| 3 | id | String | Identifikasi unik untuk setiap item menu |
| 4 | name | String | Nama dari item menu |
| 5 | price | String | Harga dari item menu |
| 6 | menuItems | ArrayList<MenuItem> | Daftar untuk menyimpan item menu yang tersedia |
| 7 | orderId | String | Menyimpan identifikasi unik untuk setiap order |
| 8 | items | ArrayList<MenuItem> | Daftar untuk menyimpan item yang dipesan dalam suatu order |
| 9 | orders | ArrayList<Order> | Menyimpan jumlah malam untuk reservasi |

1. **Constructor dan Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Metode | Jenis Metode | Fungsi |
| 1 | IDGenerator() | Constructor | Inisialisasi generator ID |
| 2 | generateMenuItemID() | Functional | Menghasilkan ID unik untuk item menu |
| 3 | generateOrderID() | Functional | Menghasilkan ID unik untuk order |
| 4 | MenuItem(String, String, double) | Constructor | Inisialisasi objek MenuItem |
| 5 | MenuDatabase() | Constructor | Inisialisasi database menu |
| 6 | addMenuItem(MenuItem) | Procedural | Tambahkan item ke database menu |
| 7 | removeMenuItem(String) | Procedural | Hapus item menu berdasarkan ID |
| 8 | getMenuItems() | Functional | Kembalikan daftar item menu |
| 9 | findMenuItemById(String) | Functional | Cari dan kembalikan item menu berdasarkan ID |
| 10 | findMenuItemByName(String) | Functional | Cari dan kembalikan item menu berdasarkan nama |
| 11 | Order(String) | Constructor | Inisialisasi objek Order |
| 12 | addItem(MenuItem) | Procedural | Tambah item ke order |
| 13 | getItems() | Functional | Kembalikan daftar item dalam order |
| 14 | OrderManager() | Constructor | Inisialisasi manager order |
| 15 | addOrder(Order) | Procedural | Tambah order baru ke manager |
| 16 | removeOrder(String) | Procedural | Hapus order berdasarkan ID |
| 17 | findOrderById(String) | Functional | Cari dan kembalikan order berdasarkan ID |
| 18 | getOrders() | Functional | Kembalikan daftar orders |

1. **Dokumentasi dan Pembahasan Code**

Sistem pemesanan makanan online ini dibagi menjadi tiga package utama: `menu`, `order`, dan `util`.

- **Package `menu`**: Berisi kelas `MenuItem` yang menyimpan detail item makanan seperti ID, nama, dan harga. `MenuDatabase` mengelola koleksi dari `MenuItem` melalui `ArrayList`, termasuk fungsi untuk menambah, menghapus, dan mencari item.

- **Package `order`**: Terdiri dari kelas `Order` yang mencatat detail dari setiap pesanan dan `OrderManager` yang mengelola koleksi dari `Order`. Metode yang tersedia memungkinkan untuk menambah item ke dalam pesanan serta mengelola pesanan melalui penambahan dan penghapusan.

- **Package `util`**: Mengandung kelas `IDGenerator` yang berfungsi untuk menghasilkan ID unik untuk item menu dan order, memastikan tidak ada duplikasi ID.

1. **Kesimpulan**

Proyek ini mengilustrasikan implementasi prinsip dasar pemrograman berorientasi objek dalam Java, memanfaatkan struktur package dan modifikasi akses untuk keamanan data. Sistem ini memungkinkan manajemen data yang efisien melalui ArrayList, dan mencapai enkapsulasi yang baik serta isolasi antar komponen. Ini menunjukkan sebuah pendekatan modular yang dapat diperluas sesuai kebutuhan pengembangan lebih lanjut.

1. **Daftar Pustaka**

-