

PEMROGRAMAN BERORIENTASI OBJEK
“PENERAPAN INHERITANCE &
ENCAPSULATION”



Oleh

Ikmal Ali Azhari

(2309116087)

Dosen Matakuliah:
Putut Pamilih Widagdo, S.Kom., M.Kom

UNIVERSITAS MULAWARMAN
FAKULTAS TEKNIK
JURUSAN SISTEM INFORMASI
2024/2025

DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN	1
1.1. Penjelasan Studi Kasus	1
1.3. Tujuan	1
BAB II PEMBAHASAN.....	2
2.1 Inheritance	2
2.2 Encapsulation	3
2.3 Source Code.....	4
2.4 Mock Up	9
2.5 Output Program	12
BAB III PENUTUP	13
3.1. Kesimpulan	13

BAB I

PENDAHULUAN

1.1. Penjelasan Studi Kasus

Program kasir ini dibuat untuk mendukung operasional Pakde Fried Chicken, sebuah restoran yang dikenal dengan sajian ayam goreng berkualitas tinggi dan berbagai pilihan menu lainnya. Dalam dunia kuliner yang semakin kompetitif, efisiensi dan kecepatan dalam pelayanan pelanggan menjadi faktor kunci untuk menarik dan mempertahankan pelanggan.

Dengan menggunakan sistem kasir yang terkomputerisasi, Pakde Fried Chicken bertujuan untuk mempermudah proses pemesanan dan pembayaran. Program ini menyediakan fitur untuk melihat menu makanan dan minuman, serta mencatat pesanan pelanggan secara sistematis.

Sistem ini dibangun dengan menerapkan konsep pemrograman berorientasi objek (OOP), yang memungkinkan pengelolaan data menu makanan dan minuman melalui inheritance. Makanan dan minuman sebagai subclass dari kelas Menu memungkinkan penambahan jenis makanan dan minuman baru dengan mudah, menjadikan program ini fleksibel dan mudah dikembangkan di masa mendatang.

Diharapkan, dengan adanya program kasir ini, Pakde Fried Chicken dapat memberikan pelayanan yang lebih baik, mempercepat transaksi, dan meningkatkan kepuasan pelanggan, serta membantu dalam pengelolaan penjualan yang lebih efisien.


1.3. Tujuan

Tujuan dari program kasir Pakde Fried Chicken ini adalah untuk mempermudah proses pemesanan dan pembayaran bagi pelanggan, sehingga transaksi dapat dilakukan dengan lebih cepat dan efisien. Dengan adanya program ini, pencatatan pesanan menjadi otomatis dan akurat, mengurangi kemungkinan kesalahan dalam input data manual. Selain itu, program ini dirancang untuk


mendukung fleksibilitas pengelolaan menu, sehingga penambahan atau perubahan menu makanan dan minuman dapat dilakukan dengan mudah tanpa perlu mengubah banyak bagian dari sistem. Dengan sistem kasir yang terkomputerisasi, pengelolaan penjualan diharapkan dapat lebih efisien, yang pada akhirnya membantu meningkatkan kepuasan pelanggan melalui pelayanan yang lebih mudah, cepat, dan lancar.

BAB II PEMBAHASAN

2.1 Inheritance




```
1 public class Makanan extends Menu {  
2     public Makanan(String name, int price) {  
3         super(name, price, "Makanan");  
4     }  
5 }
```



```
1 public class Minuman extends Menu {  
2     public Makanan(String name, int price) {  
3         super(name, price, "Minuman");  
4     }  
5 }
```

2.2 Encapsulation



```
1  public class Menu {
2      private String name;
3      private int price;
4      private String category;
5
6
7      public Menu(String name, int price, String category) {
8          this.name = name;
9          this.price = price;
10         this.category = category;
11     }
12
13     public String getName() {
14         return name;
15     }
16
17
18     public void setName(String name) {
19         this.name = name;
20     }
21
22
23     public int getPrice() {
24         return price;
25     }
26
27
28     public void setPrice(int price) {
29         this.price = price;
30     }
31
32
33     public String getCategory() {
34         return category;
35     }
36
37
38     public void setCategory(String category) {
39         this.category = category;
40     }
41 }
```

2.3 Source Code

2.3.1 Class Main

```
package com.mycompany.utspbo2;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Pesanan pesanan = new Pesanan();

        Menu[] makanan = {

            new Makanan("Ayam Biasa", 6000),

            new Makanan("Geprek Sambal Matah", 11000),

            new Makanan("Geprek Sambal Ijo", 11000),

            new Makanan("Geprek Keju Mozarella", 15000),

            new Makanan("Kripik Usus", 10000)

        };

        Menu[] minuman = {

            new Minuman("Es Teh", 5000),

            new Minuman("Es Jeruk", 5000)

        };

        // Lanjutkan logika program

        boolean pesan = true;

        while (pesan) {
```

```

System.out.println("Pilih Menu:");
System.out.println("1. Lihat Menu");
System.out.println("2. Lihat Pesanan");
System.out.println("3. Keluar");
System.out.print("Pilihan Anda (1/2/3): ");
int mainChoice = scanner.nextInt();

if (mainChoice == 1) {

    System.out.println("Pilih Kategori:");
    System.out.println("1. Makanan");
    System.out.println("2. Minuman");
    System.out.print("Pilihan Anda (1/2): ");
    int categoryChoice = scanner.nextInt();

    Menu[] selectedMenu = null;

    if (categoryChoice == 1) {
        selectedMenu = makanan;
    } else if (categoryChoice == 2) {
        selectedMenu = minuman;
    } else {
        System.out.println("Pilihan tidak valid, coba lagi.");
        continue;
    }

    boolean kategoripesanan = true;
    while (kategoripesanan) {

```

```

        System.out.println("Menu yang tersedia:");
        for (int i = 0; i < selectedMenu.length; i++) {
            System.out.println((i + 1) + ". " + selectedMenu[i].getName() +
" : Rp" + selectedMenu[i].getPrice());
        }

        System.out.print("Pilih nomor menu yang ingin dipesan (atau 0
untuk kembali): ");

        int itemChoice = scanner.nextInt();

        if (itemChoice > 0 && itemChoice <= selectedMenu.length) {
            pesanan.addMenu(selectedMenu[itemChoice - 1]);
            System.out.println("Menu " + selectedMenu[itemChoice -
1].getName() + " berhasil dipesan.");
        } else if (itemChoice == 0) {
            kategoripesanan = false;
        } else {
            System.out.println("Pilihan tidak valid, coba lagi.");
        }
    }

    } else if (mainChoice == 2) {
        pesanan.displayOrder();
    } else if (mainChoice == 3) {
        System.out.println("Terima kasih telah memesan!");
        pesan = false;
    } else {
        System.out.println("Pilihan tidak valid, coba lagi.");
    }
}

```



```
    }  
}
```

2.3.2 Class Makanan

```
public class Makanan extends Menu {  
    public Makanan(String name, int price) {  
        super(name, price, "Makanan");  
    }  
}
```

2.3.3 Class Minuman

```
public class Minuman extends Menu {  
    public Minuman(String name, int price) {  
        super(name, price, "Minuman");  
    }  
}
```

2.3.4 Class Menu

```
package com.mycompany.utspbo2;  
  
public class Menu {  
    private String name;  
    private int price;  
    private String category;  
  
    public Menu(String name, int price, String category) {  
        this.name = name;
```

```
    this.price = price;
    this.category = category;
}
```

```
public String getName() {
    return name;
}
```

```
public void setName(String name) {
    this.name = name;
}
```

```
public int getPrice() {
    return price;
}
```

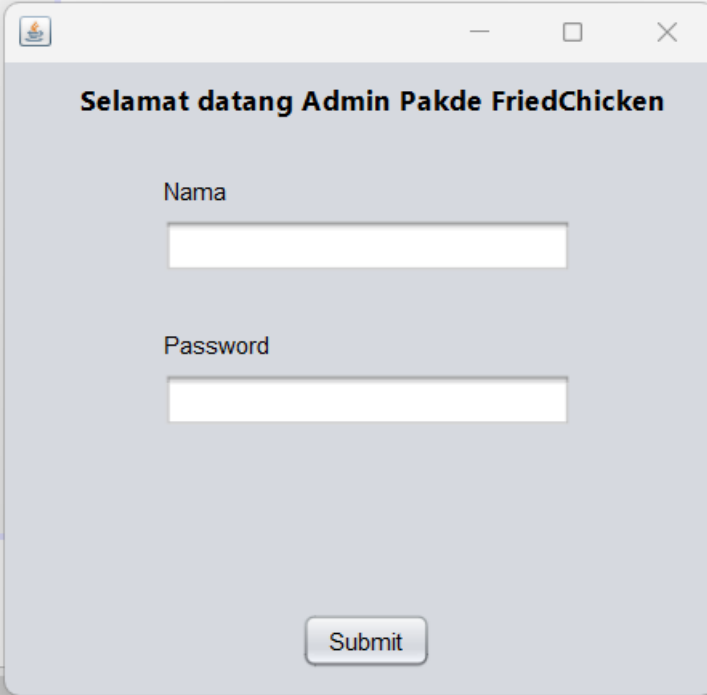
```
public void setPrice(int price) {
    this.price = price;
}
```

```
public String getCategory() {
    return category;
}
```

```
public void setCategory(String category) {  
    this.category = category;  
}  
}
```


2.4 Mock Up

2.4.1 Form Login



The image shows a mockup of a login form window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area has a light gray background. At the top, the text "Selamat datang Admin Pakde FriedChicken" is displayed in a bold, black font. Below this, there are two input fields. The first is labeled "Nama" and the second is labeled "Password". Both labels are in a standard black font. The input fields are white with a thin gray border. At the bottom center of the form, there is a "Submit" button with a rounded rectangular shape and a light gray background.

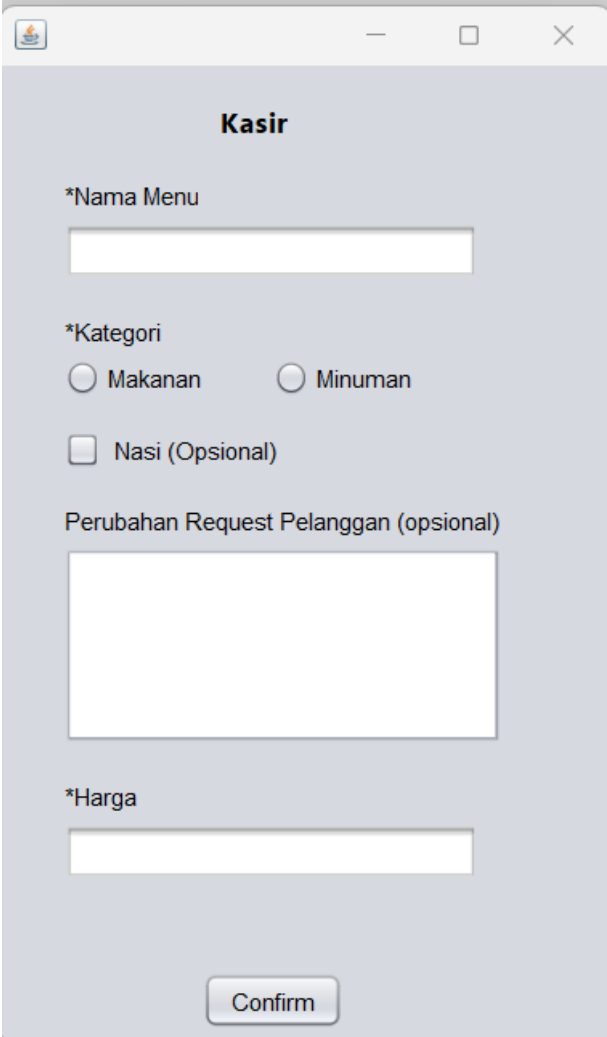
2.4.2 Form Lihat Menu



The screenshot shows a window titled "List Menu" with a standard Windows-style title bar (minimize, maximize, close buttons). The window is divided into two main sections: "Makanan" (Food) on the left and "Minuman" (Beverages) on the right. The "Makanan" section contains a list of five items with their prices, displayed in a scrollable area. The "Minuman" section contains a list of two items with their prices, also in a scrollable area.

List Menu	
Makanan	Minuman
1. Ayam Biasa : Rp6000	1. Es Teh : Rp5000
2. Geprek Sambal Matah : Rp11000	2. Es Jeruk : Rp5000
3. Geprek Sambal Ijo : Rp11000	
4. Geprek Keju Mozzarella : Rp15000	
5. Kripik Usus : Rp10000	

2.4.3 Form Inputan Kasir



Kasir

*Nama Menu

*Kategori

☐ Makanan ☐ Minuman

☐ Nasi (Opsional)

Perubahan Request Pelanggan (opsional)

*Harga

2.5 Output Program

```
Pilih Menu:
1. Lihat Menu
2. Lihat Pesanan
3. Keluar
Pilihan Anda (1/2/3): 1
Pilih Kategori:
1. Makanan
2. Minuman
Pilihan Anda (1/2): 1
Menu yang tersedia:
1. Ayam Biasa : Rp6000
2. Geprek Sambal Matah : Rp11000
3. Geprek Sambal Ijo : Rp11000
4. Geprek Keju Mozarella : Rp15000
5. Kripik Usus : Rp10000
Pilih nomor menu yang ingin dipesan (atau 0 untuk kembali): 1
Menu 'Ayam Biasa' berhasil dipesan.

Pilih Menu:
1. Lihat Menu
2. Lihat Pesanan
3. Keluar
Pilihan Anda (1/2/3): 1
Pilih Kategori:
1. Makanan
2. Minuman
Pilihan Anda (1/2): 2
Menu yang tersedia:
1. Es Teh : Rp5000
2. Es Jeruk : Rp5000
Pilih nomor menu yang ingin dipesan (atau 0 untuk kembali): 2
Menu 'Es Jeruk' berhasil dipesan.
```

Opsi Menu 1

```
Pilih Menu:
1. Lihat Menu
2. Lihat Pesanan
3. Keluar
Pilihan Anda (1/2/3): 2
Pesanan Anda:
- Ayam Biasa : Rp6000
- Geprek Sambal Matah : Rp11000
- Es Jeruk : Rp5000
Total Harga: Rp22000
Pilih Menu:
1. Lihat Menu
2. Lihat Pesanan
3. Keluar
Pilihan Anda (1/2/3): 3
Terima kasih telah memesan!
```

Opsi Menu 2 dan 3

BAB III PENUTUP

3.1. Kesimpulan

program kasir yang dikembangkan untuk Pakde Fried Chicken berperan penting dalam meningkatkan efisiensi operasional restoran. Dengan memanfaatkan konsep pemrograman berorientasi objek (OOP) dan fitur-fitur seperti pencatatan pesanan secara otomatis serta pengelolaan menu yang fleksibel, program ini membantu mempercepat proses pemesanan dan pembayaran. Selain itu, sistem ini meminimalisir kesalahan dalam pencatatan dan pengelolaan pesanan, yang pada akhirnya meningkatkan kualitas pelayanan dan kepuasan pelanggan. Program ini juga memberikan kemudahan dalam pengelolaan penjualan, sehingga Pakde Fried Chicken dapat beradaptasi lebih baik dalam menghadapi persaingan di industri kuliner. manajemen lalu lintas jaringan, memungkinkan pengaturan prioritas bandwidth untuk berbagai ruangan yang memerlukan lebih banyak sumber daya.

Desain jaringan diatas juga mempertimbangkan keandalan, di mana kegagalan pada satu perangkat tidak akan memengaruhi keseluruhan jaringan. Aspek keamanan diperhatikan melalui pengaturan akses yang ketat, sehingga hanya perangkat yang berwenang yang dapat mengakses jaringan. Dengan struktur yang terorganisir dan sistematis, konfigurasi ini mendukung kebutuhan komunikasi internal antar departemen, serta memastikan stabilitas dan keamanan jaringan secara keseluruhan, sehingga dapat beroperasi dengan efisien sesuai dengan kebutuhan organisasi.