

DOKUMEN
APLIKASI PEMESAN MAKANAN



Disusun Oleh :
Kelompok 8 (Delapan)

NAMA:	NIM:
1. Muhammad Rafli Aldian Jamil	(32602300044)
2. Muhammad Veri Ramadhani	(32602300027)
3. Marwan Hamdany	(32602300073)

PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG

2025



UNIVERSITAS ISLAM SULTAN AGUNG
SEMARAN

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT, yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan dokumen laporan aplikasi "Pemesanan Makanan" ini dengan baik. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita, Nabi Muhammad SAW, beserta keluarga, sahabat, dan umatnya hingga akhir zaman.

Dokumen ini disusun sebagai salah satu bentuk laporan pengembangan aplikasi yang bertujuan untuk mempermudah proses pemesanan makanan, baik bagi pengguna maupun admin yang mengelola menu. Penulis berusaha memberikan penjelasan yang sistematis dan rinci mengenai tujuan, desain, fitur, dan diagram yang terkait dengan aplikasi ini.

Penulis menyadari bahwa dalam penyusunan laporan ini masih terdapat banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan dan penyempurnaan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, dalam proses pengembangan aplikasi ini. Semoga laporan ini dapat memberikan manfaat bagi pembaca, khususnya bagi mereka yang sedang mempelajari atau mengembangkan aplikasi serupa.



UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG

DAFTAR ISI

HALAMAN JUDUL	1
KATA PENGANTAR.....	3
DAFTAR ISI	5
DAFTAR GAMBAR	7
DOKUMEN APLIKASI PEMESAN MAKANAN	9
1.1. Tujuan.....	9
1.2. Pendahuluan	10
1.3. Fitur Utama	10
1.4. Diagram UML.....	10
1.3.1 Diagram <i>Use Case</i>	10
1.3.2 Diagram <i>Class</i>	11
1.3.3 Diagram Activity	13
1.3.4 Diagram <i>Deployment</i>	15
1.5. Skenario Bisnis	16
1.6. Implementasi Kode Program dan <i>Output</i> Program	18
1.7. Kesimpulan	30



UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG

DAFTAR GAMBAR

Gambar 1. 1 Diagram <i>Use Case</i>	11
Gambar 1. 2 Diagram <i>Class</i>	13
Gambar 1. 3 Diagram <i>Activity</i>	14
Gambar 1. 4 Diagram <i>Deployment</i>	16
Gambar 1. 5 <i>Output</i> Pesan Makanan.....	26
Gambar 1. 6 <i>Output</i> Admin Kelola Menu (Tambah Menu)	27
Gambar 1. 7 <i>Output</i> Admin Kelola Menu (Hapus Menu).....	28
Gambar 1. 8 <i>Output</i> Admin Kelola Menu (Update Menu)	29
Gambar 1. 9 <i>Output</i> Keluar	30



UNIVERSITAS ISLAM SULTAN AGUNG
SEMAR

DOKUMEN APLIKASI

PEMESAN MAKANAN

1.1. Tujuan

Aplikasi Pemesanan Makanan ini dikembangkan dengan tujuan sebagai berikut:

1. Mempermudah Proses Pemesanan Makanan

Memberikan solusi digital yang efisien bagi pengguna untuk melihat menu, memilih makanan, dan melakukan pemesanan tanpa harus datang langsung ke lokasi restoran atau rumah makan.

2. Meningkatkan Efisiensi Pengelolaan Menu oleh Admin

Memungkinkan admin untuk dengan mudah menambah, memperbarui, atau menghapus menu secara real-time, sehingga menu yang tersedia tetap relevan dan akurat.

3. Memberikan Pengalaman Pengguna yang Praktis dan Interaktif

Aplikasi ini dirancang agar mudah digunakan dengan antarmuka yang sederhana namun informatif, sehingga pengguna dapat menyelesaikan pesanan dengan cepat dan nyaman.

4. Mengurangi Kesalahan dalam Pemesanan

Dengan adanya sistem terstruktur, aplikasi dapat meminimalkan kesalahan seperti pesanan yang salah atau informasi menu yang tidak valid.

5. Mendukung Pengelolaan Bisnis yang Lebih Modern

Aplikasi ini mendukung digitalisasi proses pemesanan, yang dapat membantu restoran atau rumah makan dalam meningkatkan layanan kepada pelanggan dan mengelola operasional secara lebih efektif.

6. Mengakomodasi Kemudahan Berbagai Peran

Aplikasi ini dirancang untuk mendukung dua peran utama, yaitu:

- Pengguna: Untuk memesan makanan dengan mudah.

- Admin: Untuk mengelola menu dan memastikan ketersediaan informasi terkini bagi pengguna.

1.2. Pendahuluan

Aplikasi ini dirancang untuk mensimulasikan proses pemesanan makanan secara online. Dengan antarmuka berbasis *command line Interface (CLI)* dengan menggunakan *software* NetBeans, pengguna dapat memilih menu, menambahkan item ke pesanan, dan melihat total biaya serta estimasi waktu pemrosesan.

1.3. Fitur Utama

1. Menampilkan Menu: Pengguna dapat melihat daftar menu makanan khas Indonesia beserta harga.
2. Menambahkan Pesanan: Pengguna dapat memilih item berdasarkan *ID* menu.
3. Melihat Rincian Pesanan: Setelah selesai memilih, pengguna dapat melihat daftar item yang dipesan, total harga, dan estimasi waktu pemrosesan.

1.4. Diagram UML

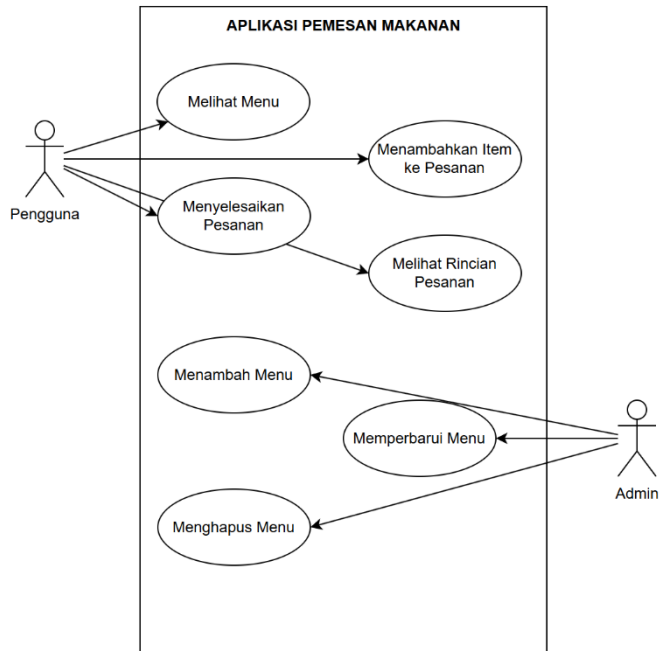
1.3.1 Diagram *Use Case*

Diagram ini menunjukkan interaksi pengguna dan admin dengan aplikasi:

- Aktor: Pengguna, Admin
- Fungsi Utama
 - 1) Pengguna
 - a. Melihat menu
 - b. Menambahkan item ke pesanan
 - c. Menyelesaikan pesanan
 - d. Melihat rincian pesanan

2) Admin

- a. Menambah Menu
- b. Menghapus Menu
- c. Memperbarui Menu



Gambar 1. 1 Diagram *Use Case*

1.3.2 Diagram *Class*

Diagram ini menunjukkan entitas yang terdiri dari atribut dan metode serta relasi dan kardinalitas yang ada di dalam aplikasi pesanan makanan

1. MenuItem

- Atribut:

- 1) Id
- 2) *Name*
- 3) *price*

- Metode:

- 1) *getId*

2) *getName*

3) *getPrice*

4) *toString*

2. *Order*

- Atribut:

1) *items*

- Metode:

1) *addItem*

2) *viewOrder*

3. *FoodOrderingApp*

- Metode:

1) *main(String[] args)*

4. *Admin*

- Atribut:

1) *username: String*

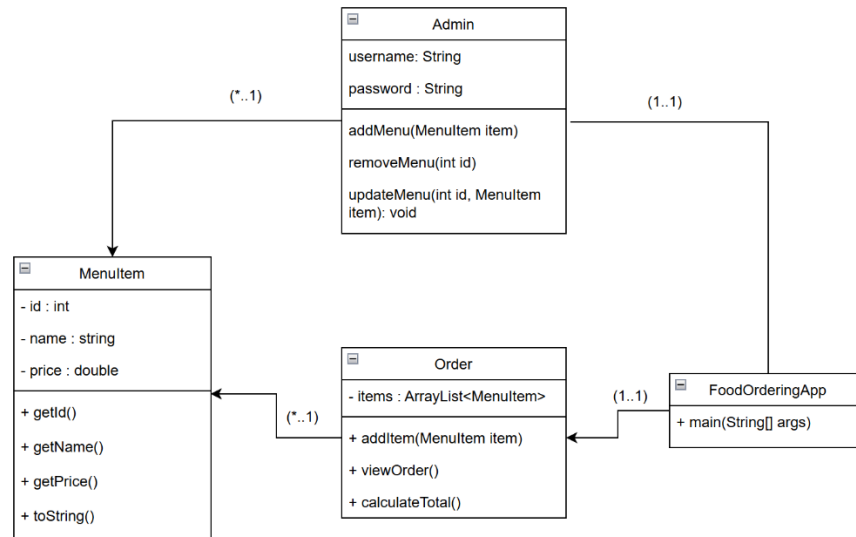
2) *password: String*

- Metode:

1) *addMenu(MenuItem item)*

2) *removeMenu(int id)*

3) *updateMenu(int id, MenuItem item)*



Gambar 1. 2 Diagram Kelas

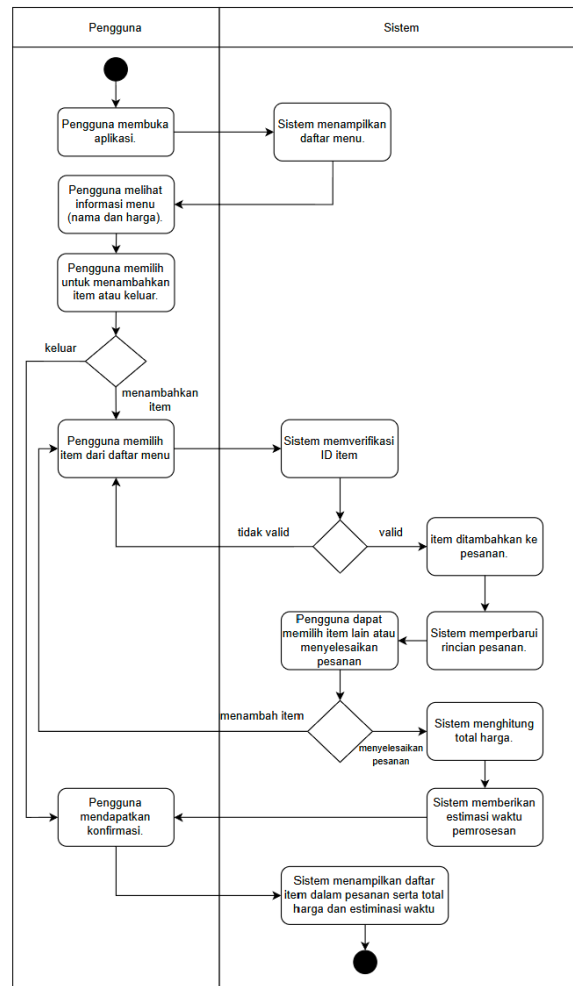
1.3.3 Diagram Activity

Diagram Activity digunakan untuk menggambarkan alur aktivitas atau proses yang terjadi di dalam aplikasi pemesanan makanan. Diagram ini menunjukkan bagaimana interaksi antar-aktor dilakukan serta langkah-langkah yang diikuti dalam proses pemesanan makanan, pengelolaan menu, dan penyelesaian pesanan.

Penjelasan Diagram Activity:

1. Proses Pemesanan Makanan oleh Pengguna
 - a. Pengguna memulai aplikasi dan melihat daftar menu.
 - b. Pengguna memilih item makanan dan menambahkannya ke keranjang pesanan.
 - c. Setelah selesai memilih, pengguna dapat melihat rincian pesanan di keranjang.
 - d. Pengguna melakukan konfirmasi pesanan.
2. Pengelolaan Menu oleh Admin

- a. Admin login ke aplikasi untuk mengakses fitur pengelolaan.
 - b. Admin dapat menambahkan menu baru, mengubah informasi menu yang sudah ada, atau menghapus menu yang tidak relevan.
 - c. Perubahan yang dilakukan admin akan langsung diperbarui di aplikasi.
3. Penyelesaian Pesanan
- a. Setelah pengguna menyelesaikan pesanan, sistem memproses pesanan dan menyimpan datanya di database.
 - b. Pengguna mendapatkan notifikasi bahwa pesanan telah diterima oleh sistem.



Gambar 1. 3 Diagram Activity

1.3.4 Diagram *Deployment*

Diagram deployment menjelaskan arsitektur fisik dari aplikasi pemesanan makanan, mencakup bagaimana komponen aplikasi diimplementasikan pada perangkat keras. Berikut adalah rincian komponen dalam diagram deployment:

1. Desktop/Mobile Client (Pengguna/Admin)

- Pengguna:

Pengguna mengakses aplikasi melalui perangkat desktop atau mobile untuk melihat menu, menambahkan item ke pesanan, melihat rincian pesanan, dan menyelesaikan transaksi.

- Admin:

Admin menggunakan aplikasi untuk mengelola menu, seperti menambahkan, memperbarui, atau menghapus item pada menu.

2. Internet

Koneksi internet digunakan sebagai media komunikasi antara perangkat pengguna/admin dan server. Seluruh permintaan data dari client dikirim ke server melalui koneksi ini, kemudian server mengembalikan respon berupa data yang diminta.

3. Application Server

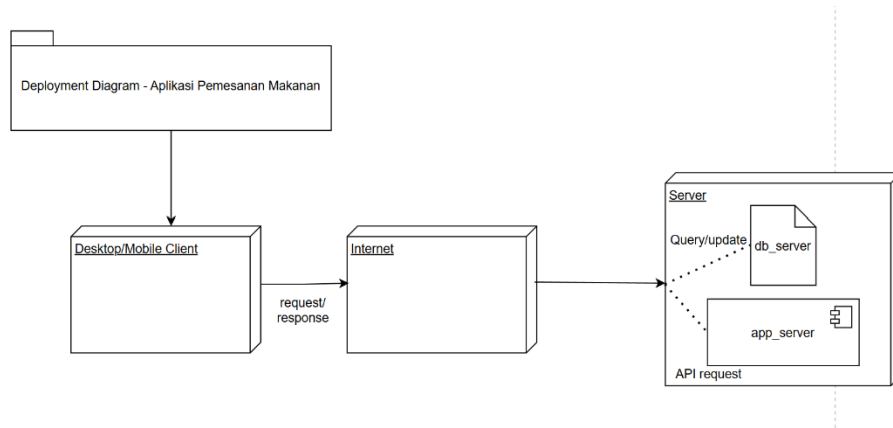
- Application server bertanggung jawab untuk memproses logika aplikasi, seperti memvalidasi permintaan dari pengguna, memproses pesanan, serta mengelola autentikasi pengguna dan admin.
- Server ini juga bertindak sebagai perantara antara client dan database server.

4. Database Server

Database server menyimpan semua data penting aplikasi, termasuk:

- Data Menu: Informasi makanan dan minuman yang tersedia dalam aplikasi.
- Data Pesanan: Rincian setiap pesanan yang dibuat oleh pengguna.

- Data Akun: Informasi akun admin untuk autentikasi dan manajemen menu.



Gambar 1. 4 Diagram *Deployment*

1.5. Skenario Bisnis

Aplikasi pemesanan makanan ini dirancang untuk mendukung dua jenis pengguna utama, yaitu Pengguna (*User*) dan Admin, dengan skenario bisnis sebagai berikut:

1. Skenario Bisnis untuk Pengguna (*User*)

Pengguna adalah individu yang menggunakan aplikasi untuk memesan makanan. Berikut adalah langkah-langkah yang menggambarkan skenario bisnis:

- Melihat Menu:

Pengguna membuka aplikasi dan melihat daftar menu makanan dan minuman yang tersedia, lengkap dengan deskripsi dan harga.

- Menambahkan Item ke Pesanan:

Setelah menemukan menu yang diinginkan, pengguna dapat menambahkan item ke keranjang pesanan (*cart*).

- Melihat Rincian Pesanan:

Pengguna dapat mengecek kembali rincian pesanan di keranjang, termasuk jumlah item, total harga, dan pilihan tambahan lainnya.

- Menyelesaikan Pesanan:

Setelah memastikan rincian pesanan sudah sesuai, pengguna menyelesaikan transaksi dengan memilih metode pembayaran yang tersedia.

2. Skenario Bisnis untuk Admin

Admin adalah pihak yang bertanggung jawab untuk mengelola konten dan data pada aplikasi. Berikut adalah langkah-langkah skenario bisnis untuk admin:

- Menambah Menu Baru:

Admin dapat menambahkan item menu baru, lengkap dengan deskripsi, harga, dan kategori.

- Memperbarui Menu:

Jika terdapat perubahan harga, deskripsi, atau ketersediaan menu, admin dapat memperbarui data tersebut agar tetap akurat.

- Menghapus Menu:

Apabila suatu menu sudah tidak tersedia, admin dapat menghapus menu tersebut dari daftar.

3. Alur Utama dalam Skenario Bisnis

Berikut adalah alur utama yang menghubungkan pengguna, admin, dan aplikasi:

- Pengguna: Melakukan proses pemesanan makanan dengan berinteraksi langsung dengan aplikasi untuk melihat menu, membuat pesanan, dan menyelesaikan transaksi.
- Admin: Mengelola data menu melalui antarmuka aplikasi admin, memastikan data yang disajikan kepada pengguna selalu akurat dan terkini.
- Server: Berperan sebagai penghubung antara pengguna dan database, memastikan bahwa setiap permintaan dari pengguna/admin diproses dengan baik.

Manfaat dari Skenario Bisnis Ini

- Pengguna mendapatkan pengalaman yang mudah dan nyaman dalam memesan makanan.
- Admin dapat mengelola operasional dengan efisien tanpa perlu melakukan proses manual.
- Restoran atau bisnis makanan dapat memberikan layanan yang lebih cepat, akurat, dan modern kepada pelanggan.

1.6. Implementasi Kode Program dan *Output* Program

1. *MenuItem.java*

```
public class MenuItem {  
    private int id;  
    private String name;  
    private double price;  
  
    public MenuItem(int id, String name, double price) {  
        this.id = id;  
        this.name = name;  
        this.price = price;  
    }  
    public int getId() {  
        return id;  
    }  
    public String getName() {  
        return name;  
    }  
    public double getPrice() {  
        return price;  
    }  
    @Override  
    public String toString() {  
        return id + ". " + name + " - Rp " + price;  
    }  
}
```

```
    }  
}
```

2. *Order.java*

```
import java.util.ArrayList;  
  
public class Order {  
    private ArrayList<MenuItem> items = new ArrayList<>();  
  
    public void addItem(MenuItem item) {  
        items.add(item);  
    }  
  
    public void viewOrder() {  
        System.out.println("Pesanan Anda:");  
        double total = 0;  
        for (MenuItem item : items) {  
            System.out.println(item);  
            total += item.getPrice();  
        }  
        System.out.println("Total: Rp " + total);  
        System.out.println("\nDetail Pesanan: ");  
        System.out.println("Jumlah Item: " + items.size());  
        System.out.println("Waktu Pemrosesan: " +  
(items.size() * 5) + " menit");  
    }  
}
```

3. Admin.java

```
import java.util.ArrayList;
import java.util.Scanner;

public class Admin {
    private static Scanner scanner = new Scanner(System.in);

    public static void kelolaMenu(ArrayList<MenuItem> menu) {
        System.out.println("\nMasukkan username & password
admin untuk mengelola menu:");
        scanner.nextLine(); // Consume newline
        String password = scanner.nextLine();

        if (password.equals("admin123")) {
            while (true) {
                System.out.println("\nMenu Pengelolaan:");
                System.out.println("1. Tambah Menu");
                System.out.println("2. Hapus Menu");
                System.out.println("3. Update Menu");
                System.out.println("0. Kembali");
                int choice = scanner.nextInt();

                if (choice == 0) {
                    break;
                } else if (choice == 1) {
                    tambahMenu(menu);
                } else if (choice == 2) {
                    hapusMenu(menu);
                } else if (choice == 3) {
                    updateMenu(menu);
                } else {
                    System.out.println("Pilihan tidak valid.
Silakan coba lagi.");
                }
            }
        }
    }
}
```

```

        } else {
            System.out.println("Password salah!");
        }
    }

    // Fitur untuk menambahkan menu
    private static void tambahMenu(ArrayList<MenuItem> menu)
    {
        System.out.println("\nMasukkan ID menu baru:");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        System.out.println("Masukkan nama menu:");
        String name = scanner.nextLine();
        System.out.println("Masukkan harga menu:");
        double price = scanner.nextDouble();

        menu.add(new MenuItem(id, name, price));
        System.out.println("Menu " + name + " berhasil
ditambahkan.");
    }

    // Fitur untuk menghapus menu
    private static void hapusMenu(ArrayList<MenuItem> menu) {
        System.out.println("\nMasukkan ID menu yang ingin
dihapus:");
        int id = scanner.nextInt();
        boolean found = false;
        for (MenuItem item : menu) {
            if (item.getId() == id) {
                menu.remove(item);
                System.out.println("Menu " + item.getName() +
" berhasil dihapus.");
                found = true;
                break;
            }
        }
    }

```

```

        if (!found) {
            System.out.println("Menu dengan ID " + id + "
tidak ditemukan.");
        }
    }
    // Fitur untuk memperbarui menu
    private static void updateMenu(ArrayList<MenuItem> menu)
    {
        System.out.println("\nMasukkan ID menu yang ingin
diperbarui:");
        int id = scanner.nextInt();
        boolean found = false;
        for (MenuItem item : menu) {
            if (item.getId() == id) {
                scanner.nextLine(); // Consume newline
                System.out.println("Masukkan nama menu
baru:");
                String newName = scanner.nextLine();
                System.out.println("Masukkan harga menu
baru:");
                double newPrice = scanner.nextDouble();
                menu.set(menu.indexOf(item), new MenuItem(id,
newName, newPrice));
                System.out.println("Menu berhasil
diperbarui.");
                found = true;
                break;
            }
        }
        if (!found) {
            System.out.println("Menu dengan ID " + id + "
tidak ditemukan.");
        }
    }
}

```

4. *FoodOrderingApp.java*

```
import java.util.ArrayList;
import java.util.Scanner;

public class FoodOrderingApp {
    private static Scanner scanner = new Scanner(System.in);
    private static ArrayList<MenuItem> menu = new
ArrayList<>();
    private static Order order = new Order();

    public static void main(String[] args) {
        // Daftar menu makanan Indonesia
        menu.add(new MenuItem(1, "Nasi Goreng", 25000));
        menu.add(new MenuItem(2, "Sate Ayam", 20000));
        menu.add(new MenuItem(3, "Gado-Gado", 15000));
        menu.add(new MenuItem(4, "Es Teh Manis", 5000));
        menu.add(new MenuItem(5, "Bakso", 18000));

        System.out.println("Selamat Datang di Aplikasi
Pemesanan Makanan!");
        while (true) {
            System.out.println("\nMenu:");
            for (MenuItem item : menu) {
                System.out.println(item);
            }

            System.out.println("\nPilihan:");
            System.out.println("1. Pesan Makanan");
            System.out.println("2. Kelola Menu (Admin)");
            System.out.println("0. Keluar");
            System.out.println("Masukan ID pilihan anda: ");
            int choice = scanner.nextInt();

            if (choice == 0) {
                break;
            }
        }
    }
}
```

```

        } else if (choice == 1) {
            pesanan();
        } else if (choice == 2) {
            Admin.kelolaMenu(menu);
        } else {
            System.out.println("Pilihan tidak valid.
Silakan coba lagi.");
        }
    }

    System.out.println("Terima kasih telah menggunakan
aplikasi!");
}

// Fitur untuk melakukan pemesanan
private static void pesanan() {
    while (true) {
        System.out.println("\nMenu:");
        for (MenuItem item : menu) {
            System.out.println(item);
        }

        System.out.println("\nMasukkan ID menu untuk
ditambahkan ke pesanan Anda (0 untuk selesai):");
        int choice = scanner.nextInt();

        if (choice == 0) {
            break;
        }

        boolean found = false;
        for (MenuItem item : menu) {
            if (item.getId() == choice) {
                order.addItem(item);
            }
        }
    }
}

```



```
                System.out.println(item.getName() + "
berhasil ditambahkan ke pesanan.");
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("ID menu tidak valid.
Silakan coba lagi.");
        }
    }

    System.out.println("\nMenyelesaikan pesanan
Anda...");
    order.viewOrder();
}
}
```

- *Output*

```
run:
Selamat Datang di Aplikasi Pemesanan Makanan!

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0

Pilihan:
1. Pesan Makanan
2. Kelola Menu (Admin)
0. Keluar
Masukan ID pilihan anda:
1

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0

Masukkan ID menu untuk ditambahkan ke pesanan Anda (0 untuk selesai):
1
Nasi Goreng berhasil ditambahkan ke pesanan.

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0

Masukkan ID menu untuk ditambahkan ke pesanan Anda (0 untuk selesai):
0

Menyelesaikan pesanan Anda...
Pesanan Anda:
1. Nasi Goreng - Rp 25000.0
Total: Rp 25000.0

Detail Pesanan:
Jumlah Item: 1
Waktu Pemrosesan: 5 menit
```

Gambar 1. 5 *Output* Pesan Makanan

Pada Gambar 1.5 menunjukkan bahwa user bisa memesan makanan dengan memilih ID yang sesuai, user juga bisa memilih lebih dari 1 ID

```

run:
Selamat Datang di Aplikasi Pemesanan Makanan!

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0

Pilihan:
1. Pesan Makanan
2. Kelola Menu (Admin)
0. Keluar
Masukan ID pilihan anda:
2

Masukkan username & password admin untuk mengelola menu:
admin
admin123

Menu Pengelolaan:
1. Tambah Menu
2. Hapus Menu
3. Update Menu
0. Kembali
1

Masukkan ID menu baru:
6
Masukkan nama menu:
Siomay
Masukkan harga menu:
10000
Menu Siomay berhasil ditambahkan.

Menu Pengelolaan:
1. Tambah Menu
2. Hapus Menu
3. Update Menu
0. Kembali
0

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0
6. Siomay - Rp 10000.0

```

Gambar 1. 6 *Output* Admin Kelola Menu (Tambah Menu)

Pada Gambar 1.6 menunjukkan bahwa Admin bisa menambahkan sebuah menu yakni dengan memasukan ID, nama, dan harga menu yang diinginkan

```
run:
Selamat Datang di Aplikasi Pemesanan Makanan!

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0

Pilihan:
1. Pesan Makanan
2. Kelola Menu (Admin)
0. Keluar
Masukan ID pilihan anda:
2

Masukkan username & password admin untuk mengelola menu:
admin
admin123

Menu Pengelolaan:
1. Tambah Menu
2. Hapus Menu
3. Update Menu
0. Kembali
2

Masukkan ID menu yang ingin dihapus:
5
Menu Bakso berhasil dihapus.

Menu Pengelolaan:
1. Tambah Menu
2. Hapus Menu
3. Update Menu
0. Kembali
0

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
```

Gambar 1. 7 *Output* Admin Kelola Menu (Hapus Menu)

Pada gambar 1.7 menunjukkan bahwa Admin bisa menghapus menu dengan memasukkan ID menu yang ingin dihapus

```
run:
Selamat Datang di Aplikasi Pemesanan Makanan!

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0

Pilihan:
1. Pesan Makanan
2. Kelola Menu (Admin)
0. Keluar
Masukan ID pilihan anda:
2

Masukkan username & password admin untuk mengelola menu:
admin
admin123

Menu Pengelolaan:
1. Tambah Menu
2. Hapus Menu
3. Update Menu
0. Kembali
3

Masukkan ID menu yang ingin diperbarui:
1
Masukkan nama menu baru:
Mie Goreng
Masukkan harga menu baru:
20000
Menu berhasil diperbarui.

Menu Pengelolaan:
1. Tambah Menu
2. Hapus Menu
3. Update Menu
0. Kembali
0

Menu:
1. Mie Goreng - Rp 20000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0
```

Gambar 1. 8 *Output Admin Kelola Menu (Update Menu)*

Pada gambar 1.8 menunjukkan bahwa Admin bisa mengubah menu dengan memasukkan ID, nama, menu maka ID yang disebutkan akan berubah

```
run:
Selamat Datang di Aplikasi Pemesanan Makanan!

Menu:
1. Nasi Goreng - Rp 25000.0
2. Sate Ayam - Rp 20000.0
3. Gado-Gado - Rp 15000.0
4. Es Teh Manis - Rp 5000.0
5. Bakso - Rp 18000.0

Pilihan:
1. Pesan Makanan
2. Kelola Menu (Admin)
0. Keluar
Masukan ID pilihan anda:
0
Terima kasih telah menggunakan aplikasi!
BUILD SUCCESSFUL (total time: 3 seconds)
```

Gambar 1. 9 *Output* Keluar

Pada gambar 1.9 menunjukkan bahwa user dan Admin dapat langsung keluar dari aplikasi dengan mengetik ID 0

1.7. Kesimpulan

Aplikasi ini menyajikan simulasi dasar untuk pemesanan makanan secara online dengan berbagai pilihan menu khas Indonesia. Dengan menggunakan struktur kode yang terorganisir secara modular, aplikasi ini dirancang agar mudah untuk dikembangkan lebih lanjut. Beberapa fitur tambahan yang dapat diimplementasikan meliputi sistem pembayaran online, integrasi dengan sistem login untuk personalisasi pengguna, dan pengelolaan transaksi yang lebih kompleks. Hal ini memberikan fleksibilitas untuk menyesuaikan aplikasi dengan berbagai kebutuhan dan fungsionalitas yang lebih luas di masa depan.