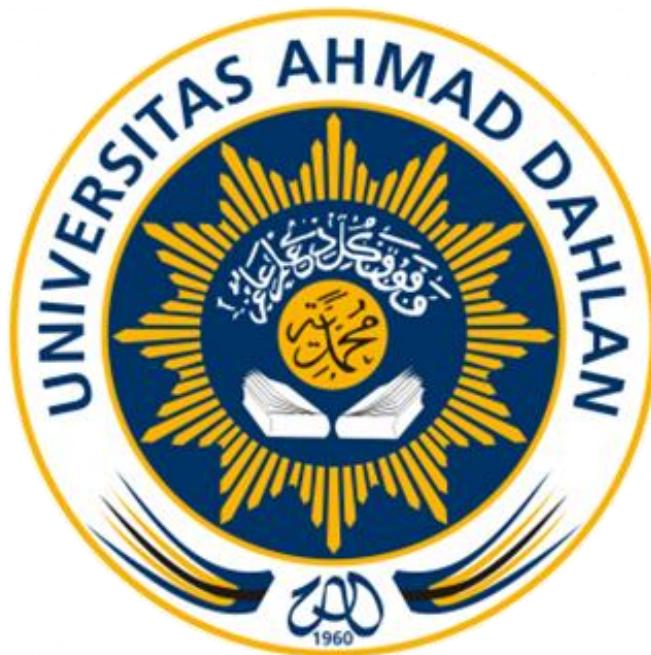


# LAPORAN PEMROGRAMAN BERORIENTASI OBJEK PERANCANGAN DAN IMPLEMENTASI SISTEM PEMESANAN MAKANAN ONLINE BERBASIS JAVA



Disusun oleh:

Muhammad Faisal Ahnav 2400018047

Ghiyas Arya Bimantara 2400018016

Reza Aditya Putra 2400018040

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS AHMAD DAHLAN  
DAERAH ISTIMEWA YOGYAKARTA**

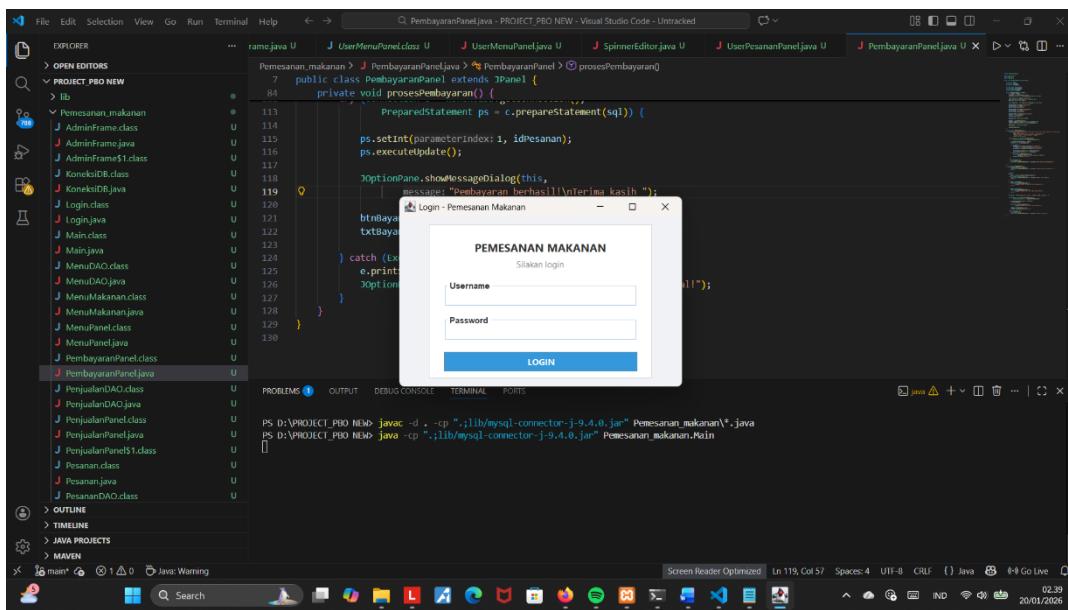
## A. Judul Proyek

Perancangan dan Implementasi Sistem Pemesanan Makanan Online Berbasis Java

Link Program:

<https://github.com/username/pemesanan-makanan-java>

Tampilan Awal:



## B. persoalan bisnis

- Sistem Pemesanan Manual: Banyak restoran masih menggunakan pencatatan manual menggunakan kertas sehingga rawan kesalahan.
- Risiko Kesalahan Pesanan: Kesalahan pencatatan menu, jumlah, dan harga sering terjadi.
- Proses Laporan Tidak Efisien: Rekap laporan penjualan dilakukan secara manual dan memakan waktu.
- Pelayanan Lambat: Proses pemesanan hingga pembayaran menjadi tidak efektif saat restoran ramai.
- Kebutuhan Digitalisasi: Perkembangan teknologi menuntut restoran untuk beradaptasi dengan sistem digital.

## **DESKRIPSI PROYEK**

1. Nama Proyek:  
Sistem Pemesanan Makanan Online Berbasis Java
2. Tujuan:  
Membangun aplikasi pemesanan makanan berbasis Java untuk meningkatkan efisiensi pelayanan, akurasi data pesanan, dan kemudahan pengelolaan laporan.
3. Fitur Utama:
  - Manajemen Data Menu : Memudahkan manajemen Data menu yang tersedia
  - Proses Pemesanan Makanan : Memudahkan pemesanan makanan yg di lakukan oleh user secara online
  - Perhitungan Total Harga Otomatis : Memudahkan penghitungan total harga yang sesuai dengan menu makanan yg di pesan oleh user
  - Manajemen Pesanan : Dapat memanagement pesanan yg sedang di proses dan yg masih belum di proses
  - Laporan Penjualan : Memudahkan admin melihat laporan transaksi penjualan yg telah di lakukan oleh user
4. Teknologi & Metodologi:
  - Bahasa Pemrograman: Java, karena bersifat multiplatform dan memiliki struktur yang terorganisir
  - Metode Pengembangan: SDLC
  - Metode Pengujian: Blackbox Testing, yang berfokus pada verifikasi fungsionalitas aplikasi tanpa harus masuk ke dalam struktur kode internalnya

## **C. SPESIFIKASI TEKNIS**

- Bahasa Pemrograman: Java
- Konsep PBO: Encapsulation, Inheritance, Polymorphism
- Database: MySQL
- GUI: Java Swing

Fitur Sistem:

**- Input Data Pelanggan**

Mencatat data pelanggan yang melakukan pemesanan, seperti nama pelanggan dan ID pesanan sebagai identitas transaksi.

**- Manajemen Menu Makanan**

Menampilkan daftar menu makanan dan minuman yang tersedia lengkap dengan harga, serta memungkinkan admin untuk menambah, mengubah, dan menghapus data menu.

### - Proses Pemesanan Makanan

Pelanggan dapat memilih menu dan jumlah pesanan, kemudian sistem akan mencatat pesanan secara otomatis ke dalam sistem.

### - Kalkulasi Total Harga Otomatis

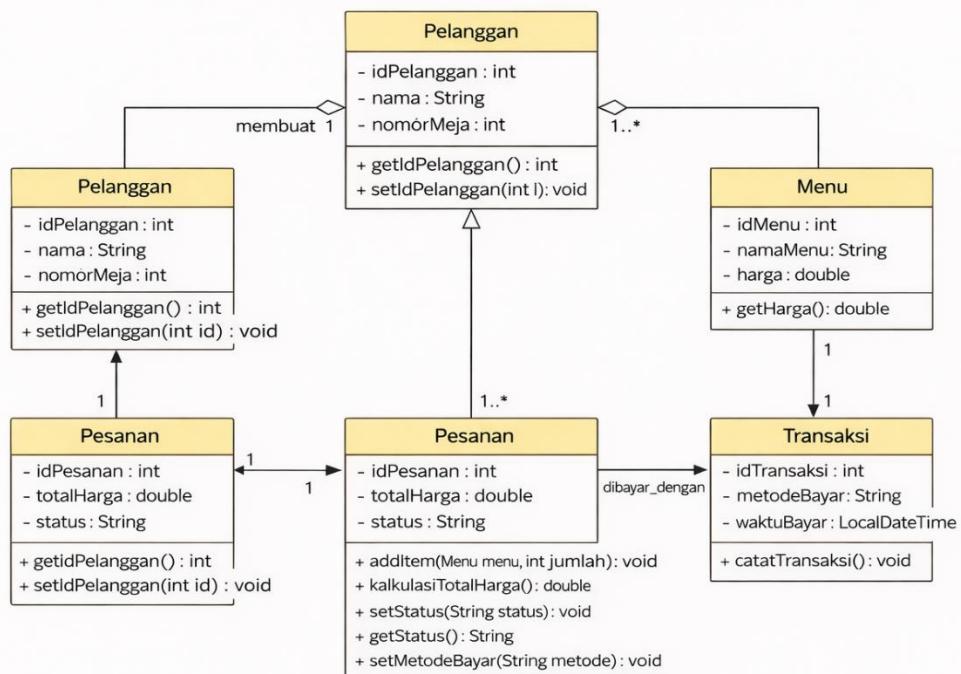
Menghitung total biaya pemesanan secara otomatis berdasarkan harga menu dan jumlah pesanan yang dipilih, sehingga meminimalkan kesalahan perhitungan.

### - Log Transaksi

Menampilkan riwayat transaksi pemesanan yang telah diproses secara **realtime** pada dashboard admin sebagai bahan monitoring dan laporan penjualan.

## D. RANCANGAN MODEL DIAGRAM UML

Sistem Pemesanan Makanan Online



Perancangan sistem menggunakan UML berupa:

- Use Case Diagram
- Class Diagram
- Sequence Diagram

Class utama yang digunakan:

- Class AdminFrame
- Class KoneksiDB
- Class Main
- Class Login
- Class UserFrame

## E. RANCANGAN ANTARMUKA BERBASIS GUI

Aplikasi **Pemesanan Makanan Online** ini dirancang berbasis GUI menggunakan pustaka **Java Swing** untuk memberikan pengalaman pengguna yang interaktif dan responsif. Desain tampilan difokuskan pada kemudahan navigasi konsumen dalam memilih menu. Rancangan GUI meliputi:

### 1. Bagian Input & Pemilihan (Sidebar/Main Panel)

Bagian ini dirancang untuk mempercepat proses pemesanan oleh konsumen:

- **Daftar Menu (List/Icons):** Menampilkan variasi makanan dan minuman yang tersedia beserta harganya.
- **Field Jumlah Pesanan (TextField):** Tempat konsumen memasukkan kuantitas (jumlah porsi) yang ingin dipesan.

### 2. Panel Kontrol (Tombol Aksi)

Terdapat tombol-tombol utama untuk menggerakkan alur transaksi:

- **Tombol TAMBAH PESANAN:** Tombol yang berfungsi memasukkan menu yang dipilih ke dalam daftar belanja sementara.

- **Tombol PROSES PESANAN:** Digunakan untuk melakukan *checkout* atau finalisasi pemesanan agar data terkirim ke sistem dapur/admin.

### 3. Bagian Output (Display Data & Tabbed Panel)

Bagian ini menggunakan sistem tampilan yang rapi untuk memisahkan fungsi operasional:

- **Tabel Keranjang Belanja (JTable):** Menampilkan ringkasan pesanan aktif yang berisi Nama Menu, Harga Satuan, Jumlah, dan Subtotal.
- **Log Struk Transaksi (TextArea):** Area teks khusus dengan font *monospaced* yang menampilkan detail rekapitulasi pembayaran, meliputi:
  - **Metode:** Menunjukkan jenis pembayaran yang dipilih konsumen.
  - **Total Bayar:** Kalkulasi akhir dari seluruh item yang dipesan.
  - **Timestamp:** Mencatat waktu pemesanan secara otomatis untuk keperluan antrean dapur.

## F. SKRIP PROGRAM DAN PENJELASANNYA

Skrip Program dan penjelasannya

Aplikasi Pemesanan Makanan Online berbasis Java dikembangkan menggunakan konsep OOP dengan beberapa kelas utama yang saling terintegrasi, yaitu:

- **Class AdminFrame** digunakan untuk merepresentasikan halaman utama admin pada aplikasi pemesanan makanan. Class ini berfungsi sebagai antarmuka grafis berbasis Java Swing yang menampilkan dashboard serta menyediakan navigasi ke menu makanan, data pesanan, dan data penjualan. Class ini juga mengatur pergantian tampilan panel dan proses logout admin, sehingga aktivitas pengelolaan data dapat dilakukan secara terstruktur dan mudah digunakan.

```
1 package Pemesanan_makanan;
2
3 import javax.swing.*;
4 import java.awt.*;
5
6 public class AdminFrame extends JFrame {
7
8     private JPanel contentPanel;
9     private JLabel lblTitle;
10
11    public AdminFrame() {
12        setTitle(title: "Admin Panel - Pemesanan Makanan");
13        setSize(width: 900, height: 550);
14        setLocationRelativeTo(c: null);
15        setDefaultCloseOperation(EXIT_ON_CLOSE);
16        setLayout(new BorderLayout());
17
18        // ===== SIDEBAR =====
19        JPanel sidebar = new JPanel();
20        sidebar.setBackground(new Color(r: 45, g: 52, b: 54));
21        sidebar.setPreferredSize(new Dimension(width: 200, height: 0));
22        sidebar.setLayout(new BoxLayout(sidebar, BoxLayout.Y_AXIS));
23
24        JLabel lblAdmin = new JLabel(text: "ADMIN");
25        lblAdmin.setForeground(Color.WHITE);
26        lblAdmin.setFont(new Font(name: "Segoe UI", Font.BOLD, size: 18));
27        lblAdmin.setAlignmentX(Component.CENTER_ALIGNMENT);
28        lblAdmin.setBorder(BorderFactory.createEmptyBorder(top: 20, left: 0, bottom: 20, right: 0));
29
30        sidebar.add(lblAdmin);
31
32        JButton btnMenu = createMenuButton(text: "Menu Makanan");
33        JButton btnPesanan = createMenuButton(text: "Data Pesanan");
34        JButton btnPenjualan = createMenuButton(text: "Data Penjualan");
35        JButton btnKeluar = createMenuButton(text: "Keluar");
36
37        sidebar.add(btnMenu);
```

- **Class KoneksiDB** digunakan untuk mengatur koneksi antara aplikasi pemesanan makanan dengan database MySQL. Class ini menyimpan konfigurasi database seperti URL, username, dan password, serta menyediakan method getConnection() untuk membuat dan mengembalikan koneksi database. Dengan menggunakan method statis, class ini memudahkan pemanggilan koneksi dari class lain dan membantu menjaga efisiensi serta konsistensi akses database.

```

1 package Pemesanan_makanan;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6 public class KoneksiDB {
7
8     private static final String URL = "jdbc:mysql://localhost:3306/pemesanan_makanan?useSSL=false&serverTimezone=UTC";
9     private static final String USER = "root";
10    private static final String PASS = ""; // isi kalau MySQL pakai password
11
12    public static Connection getConnection() {
13        try {
14            // Load driver MySQL
15            Class.forName(className: "com.mysql.cj.jdbc.Driver");
16
17            System.out.println(x: "Driver MySQL OK");
18
19            // Buat koneksi
20            Connection conn = DriverManager.getConnection(URL, USER, PASS);
21            System.out.println(x: "Koneksi database berhasil");
22
23            return conn;
24
25        } catch (Exception e) {
26            System.out.println(x: "KONEKSI DATABASE GAGAL");
27            e.printStackTrace();
28            return null;
29        }
30    }
31}
32

```

- **Class Login** digunakan untuk menangani proses autentikasi pengguna pada aplikasi pemesanan makanan. Class ini menyediakan antarmuka login berbasis Java Swing yang memungkinkan pengguna memasukkan username dan password. Data login diverifikasi melalui koneksi ke database MySQL, kemudian sistem menentukan hak akses pengguna berdasarkan peran (admin atau user). Jika login berhasil, pengguna akan diarahkan ke halaman yang sesuai, sedangkan jika gagal akan ditampilkan pesan kesalahan.

```

1 package Pemesanan_makanan;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.sql.*;
6
7 public class Login extends JFrame {
8
9     private JTextField txtUsername;
10    private JPasswordField txtPassword;
11
12    public Login() {
13        setTitle(title: "Login - Pemesanan Makanan");
14        setSize(width: 420, height: 280);
15        setLocationRelativeTo(c: null);
16        setDefaultCloseOperation(EXIT_ON_CLOSE);
17
18        // ===== BACKGROUND =====
19        JPanel mainPanel = new JPanel(new GridBagLayout());
20        mainPanel.setBackground(new Color(r: 240, g: 242, b: 245));
21
22        // ===== CARD =====
23        JPanel card = new JPanel();
24        card.setPreferredSize(new Dimension(width: 320, height: 220));
25        card.setBackground(Color.WHITE);
26        card.setLayout(new BoxLayout(card, BoxLayout.Y_AXIS));
27        card.setBorder(BorderFactory.createCompoundBorder(
28            BorderFactory.createLineBorder(new Color(r: 220, g: 220, b: 220)),
29            BorderFactory.createEmptyBorder(top: 20, left: 20, bottom: 20, right: 20)));
30
31        // ===== TITLE =====
32        JLabel lblTitle = new JLabel(text: "PEMESANAN MAKANAN");
33        lblTitle.setFont(new Font(name: "Segoe UI", Font.BOLD, size: 16));
34        lblTitle.setAlignmentX(Component.CENTER_ALIGNMENT);
35
36        JLabel lblSub = new JLabel(text: "Silakan login");
37        lblSub.setFont(new Font(name: "Segoe UI", Font.PLAIN, size: 12));

```

- **Class Main** merupakan class utama yang berfungsi sebagai titik awal (entry point) dari aplikasi pemesanan makanan. Class ini berisi method main() yang dijalankan pertama kali saat program dieksekusi, dengan tugas utama memanggil class Login agar aplikasi langsung menampilkan halaman login kepada pengguna.

```
1 package Pemesanan_makanan;
2
3 public class Main {
4     Run | Debug
5     public static void main(String[] args) {
6         new Login();
7     }
8 }
```

- **Class UserFrame** digunakan untuk menampilkan halaman utama pengguna (user) pada aplikasi pemesanan makanan. Class ini menyediakan antarmuka grafis berbasis Java Swing yang dilengkapi dengan sidebar navigasi untuk mengakses menu makanan, pesanan pengguna, dan pembayaran. Class ini juga menyimpan idUser untuk menampilkan data yang sesuai dengan pengguna yang sedang login, serta mengatur pergantian panel konten dan proses logout pengguna.

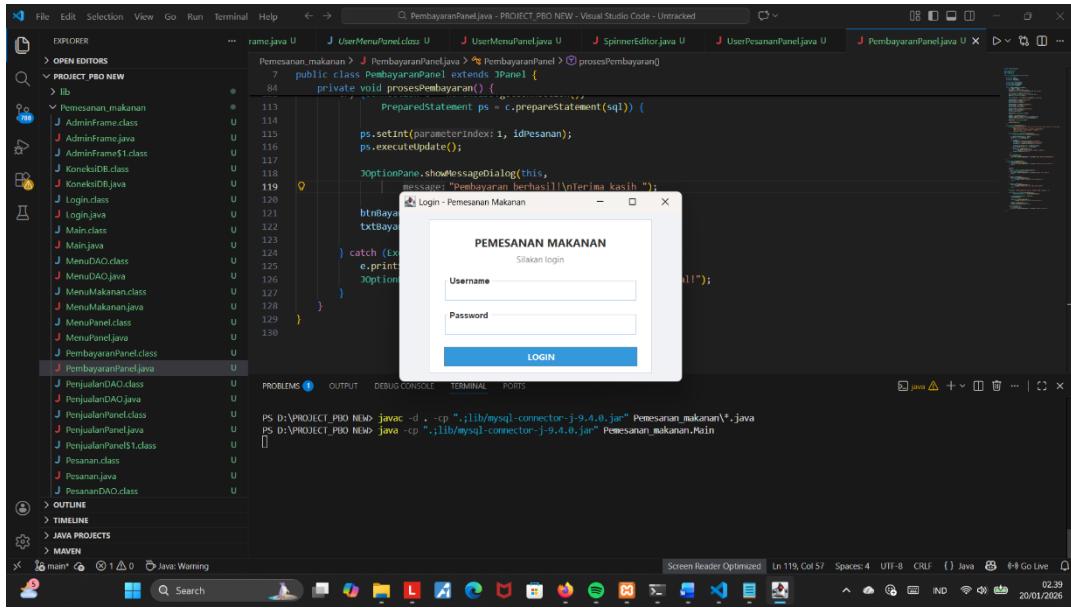
```
1 package Pemesanan_makanan;
2
3 import javax.swing.*;
4 import java.awt.*;
5
6 public class UserFrame extends JFrame {
7
8     private JPanel contentPanel;
9     private JLabel lblTitle;
10    private int idUser;
11
12    public UserFrame(int idUser) {
13        this.idUser = idUser;
14
15        setTitle(title: "User Dashboard - Pemesanan Makanan");
16        setSize(width: 900, height: 550);
17        setLocationRelativeTo(c: null);
18        setDefaultCloseOperation(EXIT_ON_CLOSE);
19        setLayout(new BorderLayout());
20
21        // ===== SIDEBAR =====
22        JPanel sidebar = new JPanel();
23        sidebar.setBackground(new Color(r: 52, g: 73, b: 94));
24        sidebar.setPreferredSize(new Dimension(width: 200, height: 0));
25        sidebar.setLayout(new BoxLayout(sidebar, BoxLayout.Y_AXIS));
26
27        JLabel lblUser = new JLabel(text: "USER MENU");
28        lblUser.setForeground(Color.WHITE);
29        lblUser.setFont(new Font(name: "Segoe UI", Font.BOLD, size: 18));
30        lblUser.setAlignmentX(Component.CENTER_ALIGNMENT);
31        lblUser.setBorder(BorderFactory.createEmptyBorder(top: 20, left: 0, bottom: 20, right: 0));
32
33        sidebar.add(lblUser);
34
35        JButton btnMenu = createMenuButton(text: "Menu Makanan");
36        JButton btnPesanan = createMenuButton(text: "Pesanan Saya");
37        JButton btnBayar = createMenuButton(text: "Pembayaran");
```

```
37     JButton btnBayar = createMenuButton(text: "Pembayaran");
38     JButton btnKeluar = createMenuButton(text: "Keluar");
39
40     sidebar.add(btnMenu);
41     sidebar.add(btnPesanan);
42     sidebar.add(btnBayar);
43     sidebar.add(Box.createVerticalGlue());
44     sidebar.add(btnKeluar);
45
46     add(sidebar, BorderLayout.WEST);
47
48     // ===== HEADER =====
49     JPanel header = new JPanel(new BorderLayout());
50     header.setPreferredSize(new Dimension(width: 0, height: 60));
51     header.setBackground(new Color(r: 236, g: 240, b: 241));
52
53     lblTitle = new JLabel(text: "Dashboard");
54     lblTitle.setFont(new Font(name: "Segoe UI", Font.BOLD, size: 18));
55     lblTitle.setBorder(BorderFactory.createEmptyBorder(top: 0, left: 20, bottom: 0, right: 0));
56     header.add(lblTitle, BorderLayout.WEST);
57
58     add(header, BorderLayout.NORTH);
59
60     // ===== CONTENT =====
61     contentPanel = new JPanel(new BorderLayout());
62     contentPanel.setBackground(Color.WHITE);
63     add(contentPanel, BorderLayout.CENTER);
64
65     showMenuMakanan();
66
67     // ===== ACTION =====
68     btnMenu.addActionListener(e -> showMenuMakanan());
69     btnPesanan.addActionListener(e -> showPesanan());
70     btnBayar.addActionListener(e -> showPembayaran());
71     btnKeluar.addActionListener(e -> logout());
```

## G. PENJELASAN SCREENSHOT TAMPILAN APLIKASI

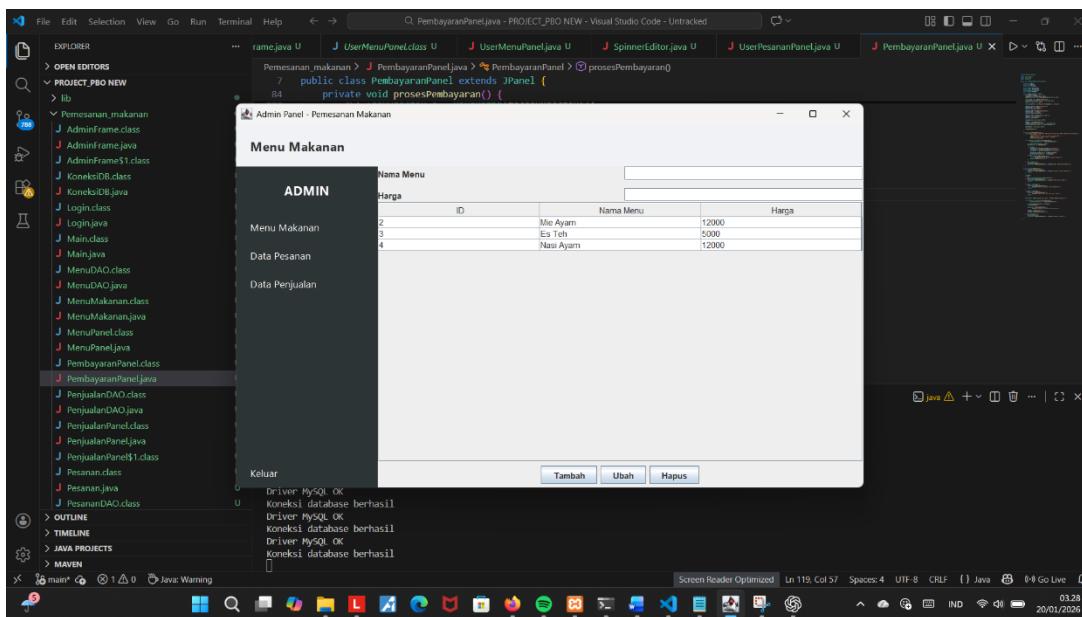
### 1. Form Login

Digunakan untuk autentikasi admin dan pelanggan.



### 2. Dashboard Admin

Mengelola menu, Mengubah Status pesanan dan melihat laporan penjualan.

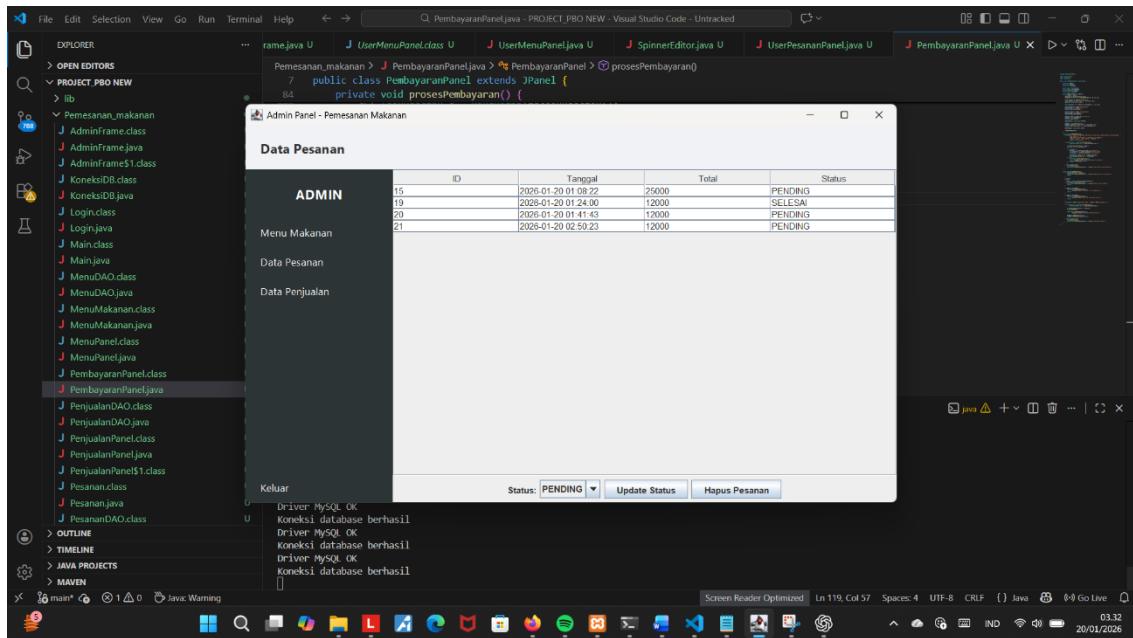


### 3. Dashboard User

Digunakan pelanggan untuk memilih menu, pembayaran dan jumlah pesanan.

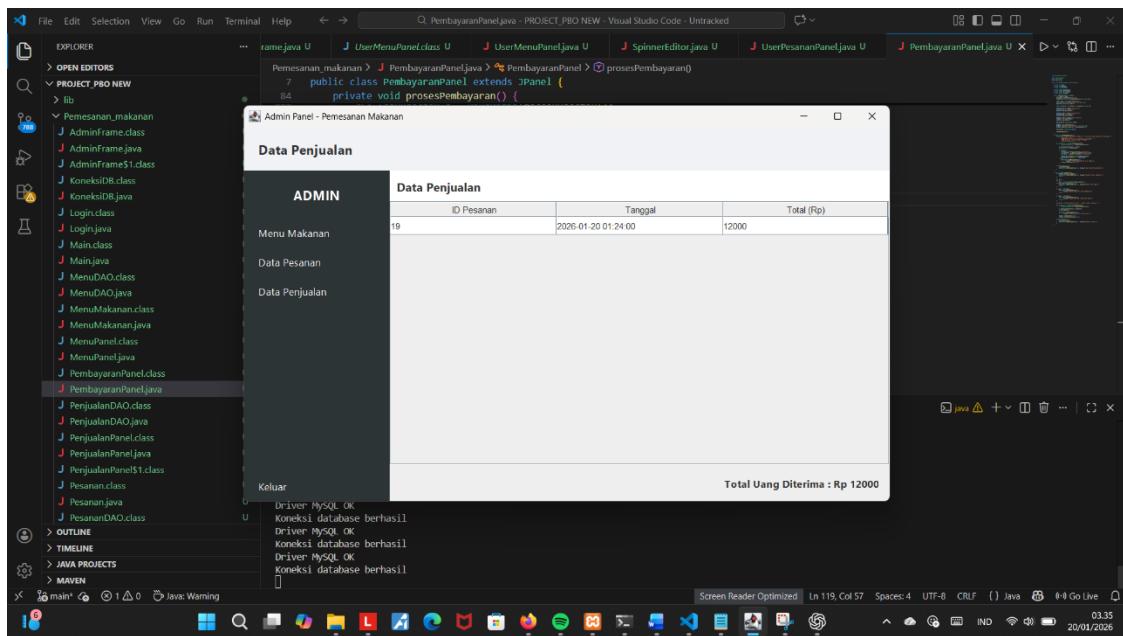
### 4. Data Pesanan Pada Admin

Digunakan untuk memonitoring pesanan yg belum/ sedang di proses



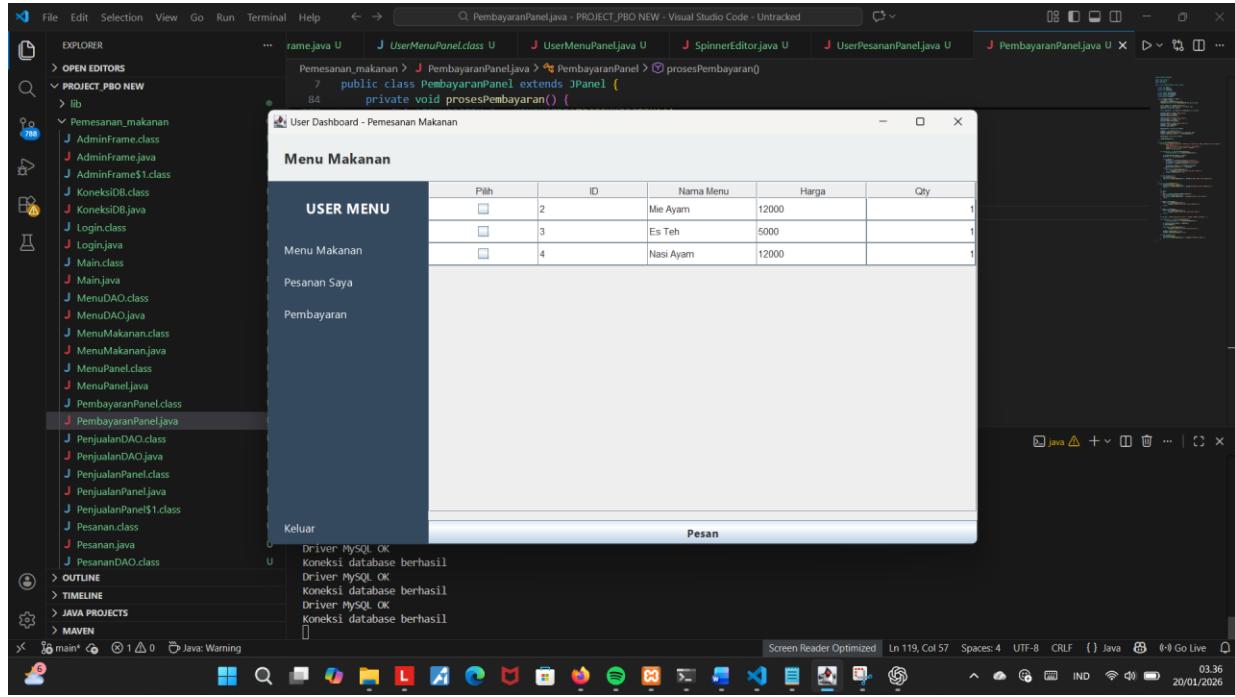
### 5. Data Penjualan Pada admin

Menampilkan laporan transaksi penjualan.



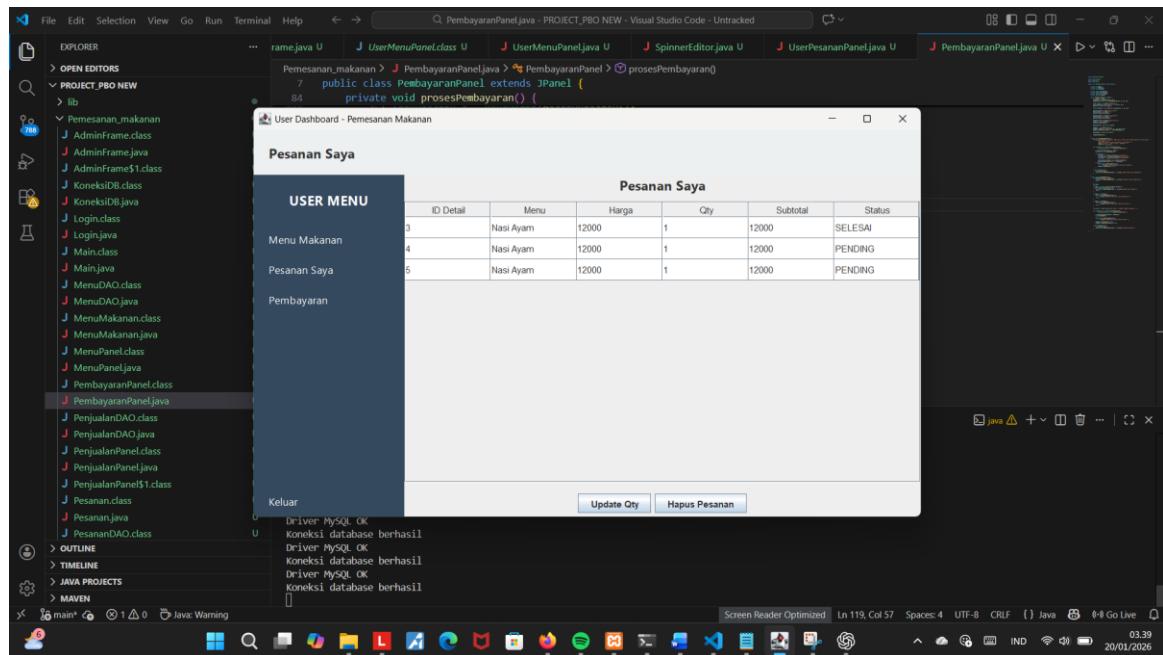
## 6. Menu Makanan Pada User

Menampilkan menu serta melakukan pemesanan



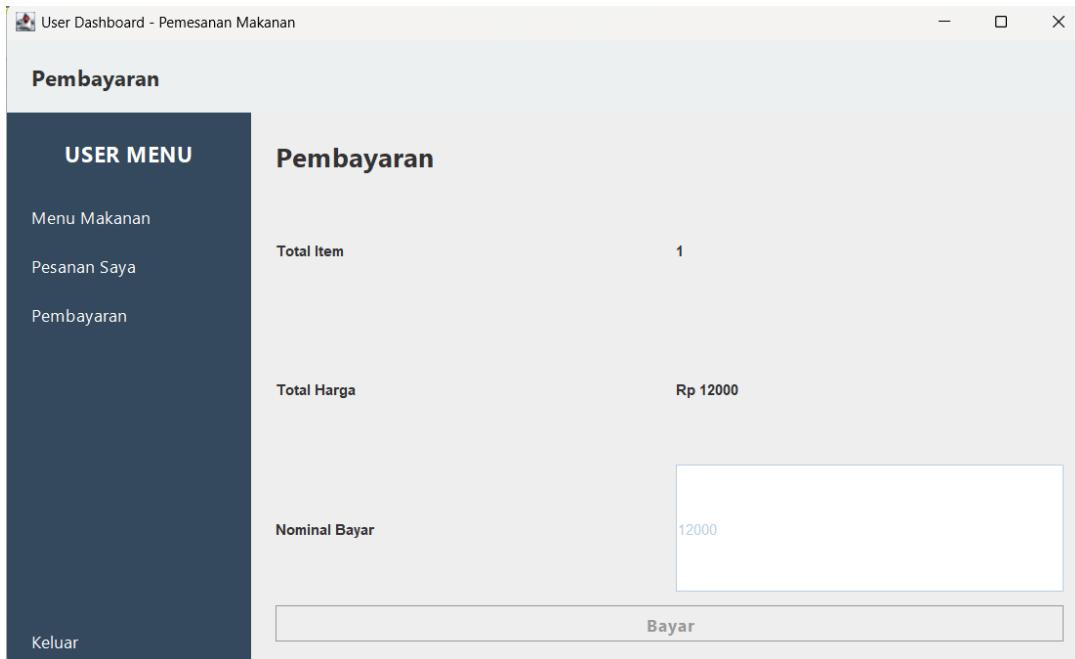
## 7. Menu Pesanan Saya pada User

Menampilkan menu yg sedang di pesan oleh user, dan user bisa menghapus jika status masih pending



8. Menu Pembayaran pada User

Menampilkan cara pembayaran, memasukan nominal yg sesuai



## H. PENJELASAN Screenshot STATUS UNGGAH DI GITHUB

Proyek sistem pemesanan makanan online telah berhasil diunggah ke GitHub hingga versi final. Repository berisi source code Java, struktur folder proyek, dan riwayat commit sebagai bukti pengerjaan bertahap.

Name	Last commit message	Last commit date
..		2 hours ago
AdminFrame\$1.class	Add package-info.java for internal X Protocol tests	2 hours ago
AdminFrame.class	Add package-info.java for internal X Protocol tests	2 hours ago
AdminFrame.java	Add package-info.java for internal X Protocol tests	2 hours ago
KoneksiDB.class	Add package-info.java for internal X Protocol tests	2 hours ago
KoneksiDB.java	Add package-info.java for internal X Protocol tests	2 hours ago
Login.class	Add package-info.java for internal X Protocol tests	2 hours ago
Login.java	Add package-info.java for internal X Protocol tests	2 hours ago
Main.class	Add package-info.java for internal X Protocol tests	2 hours ago
Main.java	Add package-info.java for internal X Protocol tests	2 hours ago
MenuDAO.class	Add package-info.java for internal X Protocol tests	2 hours ago

## I. ANALISIS PENGERJAAN PROYEK

### Waktu Pengerjaan

Pengembangan sistem pemesanan makanan online dilaksanakan selama kurang lebih dua minggu. Proses pengerjaan dimulai dari analisis kebutuhan dan perancangan alur sistem, dilanjutkan dengan implementasi program menggunakan bahasa Java, serta diakhiri dengan tahap pengujian untuk memastikan sistem berjalan dengan baik sebelum pengumpulan UAS.

### Ketercapaian Spesifikasi

Secara keseluruhan, sistem yang dikembangkan telah memenuhi tujuan dan spesifikasi awal yang direncanakan. Aplikasi mampu menampilkan daftar menu makanan, memproses pemesanan oleh pengguna, menyimpan data pesanan ke dalam database, serta menyediakan fitur pengelolaan pesanan dan data penjualan untuk admin. Penerapan konsep **Object Oriented Programming (OOP)** membantu sistem menjadi lebih terstruktur dan mudah dikembangkan.

### Biaya yang Dibutuhkan

Dalam proses pengembangan proyek ini tidak diperlukan biaya tambahan (Rp0), karena seluruh tools dan teknologi yang digunakan bersifat gratis dan open-source, seperti Java Development Kit (JDK), MySQL Server, serta lingkungan pengembangan VS Code.

**Kendala:**

- a. Terjadi beberapa kendala saat melakukan pengambilan dan pengolahan data dari database, terutama pada proses sinkronisasi data pesanan antara user dan admin.
- b. Pengaturan tampilan antarmuka pengguna (GUI) menjadi tantangan tersendiri, khususnya dalam menyesuaikan komponen agar tetap proporsional saat ukuran jendela berubah.
- c. Kesulitan dalam memastikan alur pemesanan berjalan berurutan, mulai dari pemilihan menu hingga proses pembayaran.

**Tantangan Masa Depan**

Untuk pengembangan selanjutnya, sistem pemesanan makanan online ini masih dapat ditingkatkan dengan menambahkan fitur pembayaran digital, sistem pelacakan status pesanan, serta notifikasi otomatis kepada pengguna. Selain itu, pengembangan ke versi berbasis web atau aplikasi mobile juga menjadi tantangan sekaligus peluang agar sistem dapat digunakan secara lebih luas.