

LAPORAN UJIAN AKHIR SEMESTER 4
PERANCANGAN PROGRAM RENTAL PLAYSTATION

Disusun untuk Memenuhi Tugas Mata Kuliah Pemrograman Berorientasi Objek

Dosen Pengampu :

Taufik Ridwan, M.T.



Disusun oleh:

Abid Zeva Ghozali	(2310631250081)
Farih Rahmatullah	(2310631250055)
Farhan Ramadhan	(2310631250089)
Fatkhi Maulana	(2310631250015)

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SINGAPERBANGSA KARAWANG

2025

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmat Nya penyusun dapat menyelesaikan Laporan yang berjudul “ Perancangan Program Rental Playstation” tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan. Sholawat serta salam semoga tercurahkan kepada junjungan Nabi Besar Muhammad SAW. Semoga kita bisa mendapatkan syafaat di hari akhir nanti.

Dalam penyusunan laporan ini kami banyak mengalami hambatan, namun berkat bantuan dan kerja sama dari berbagai pihak, laporan ini dapat terselesaikan. Dengan penuh kerendahan hati, kami mengucapkan terimakasih kepada :

1. Prof. Dr. Ade Maman Suherman, S.H., M.Sc. selaku Rektor Universitas Singaperbangsa Karawang.
2. Dr. Oman Komarudin, S.Si., M.Kom. selaku Dekan Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang.
3. Taufik Ridwan, M.T. selaku Dosen Pengampu Mata Kuliah Pemrograman Berorientasi Objek.

yang telah memberi arahan, masukan dan saran dalam pembuatan laporan ini. Kami menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan karena keterbatasan kami. Maka dari itu kami selaku penyusun sangat mengharapkan kritik dan saran untuk menyempurnakan laporan ini. Semoga apa yang ditulis dapat bermanfaat bagi semua pihak yang membutuhkan.

Karawang, 22 Mei 2025

Kelompok 7 *Rental Playstation*

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	3
BAB I.....	4
PENDAHULUAN.....	4
1.1 Latar Belakang.....	4
1.2 Rumusan Masalah.....	5
1.3 Tujuan Masalah.....	5
BAB II.....	7
PEMBAHASAN.....	7
2.1 Fitur-Fitur.....	7
2.2 Penjelasan Konsep OOP di Kode Program.....	8
2.3 Unified Modeling Language.....	10
2.4 Implementasi Kode Program dan Pengujian.....	13
BAB III.....	25
KESIMPULAN.....	25
3.1 Kesimpulan.....	25
3.2 Saran.....	25

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat telah mendorong digitalisasi di berbagai sektor, termasuk sektor jasa hiburan seperti rental PlayStation. PlayStation merupakan salah satu konsol game yang sangat populer di kalangan masyarakat, khususnya remaja dan dewasa muda. Hal ini membuat bisnis rental PlayStation memiliki potensi pasar yang cukup besar. Namun, dalam praktiknya, masih banyak pelaku usaha rental yang mengelola data transaksi dan pelanggan secara manual, sehingga rawan terjadi kesalahan pencatatan, kehilangan data, serta kurang efisien dalam pengelolaannya.

Program yang dirancang akan mencakup fitur-fitur utama seperti pengelolaan data pelanggan, pencatatan transaksi penyewaan, pengembalian, dan pembuatan laporan. Sistem ini tidak hanya membantu pemilik usaha dalam mempermudah operasional, tetapi juga menjadi media pembelajaran dalam mengimplementasikan teori yang telah dipelajari secara praktis. Program ini dapat mengakumulasi Data pelanggan dari Nama, Jenis Playstation yang disewa, lama waktu yang di sewa, hingga makanan yang dipesan oleh pelanggan dan jumlah yang harus dibayarkan oleh pelanggan. Program ini juga sudah terhubung dengan Database sehingga data pelanggan tidak akan hilang selama database masih ada.

Melalui tugas ini, diharapkan kami sebagai mahasiswa dapat memahami bagaimana mengembangkan sebuah Sistem yang terstruktur, efisien, dan user-friendly, sekaligus memberikan solusi nyata terhadap permasalahan yang sering dihadapi dalam usaha rental PlayStation.

1.2 Rumusan Masalah

1. Bagaimana cara merancang sistem informasi yang dapat mengelola data pelanggan dan transaksi rental PlayStation secara efektif dan efisien ?
2. Bagaimana sistem dapat meminimalisir kesalahan pencatatan dan kehilangan data yang sering terjadi dalam pengelolaan manual ?
3. Fitur-fitur apa saja yang perlu disediakan dalam sistem untuk mendukung operasional rental PlayStation, seperti pencatatan penyewaan, pengembalian, pemesanan makanan, dan laporan ?
4. Bagaimana mengintegrasikan sistem dengan database agar data dapat tersimpan dengan aman dan mudah diakses?

1.3 Tujuan Masalah

1. Mengembangkan sistem informasi rental PlayStation yang mampu mengelola data pelanggan dan transaksi secara digital dan terstruktur.
2. Meningkatkan efisiensi dan akurasi dalam pencatatan data melalui otomatisasi proses bisnis.
3. Menyediakan fitur-fitur utama yang mendukung operasional rental, seperti pengelolaan data pelanggan, transaksi sewa dan pengembalian, pemesanan makanan, serta pembuatan laporan.
4. Mengintegrasikan sistem dengan database untuk memastikan keamanan dan keutuhan data pelanggan dan transaksi.

BAB II

PEMBAHASAN

2.1 Fitur-Fitur

Program yang kami kembangkan memiliki beberapa fitur utama yang mendukung operasional sistem rental PlayStation secara komprehensif. Sistem ini dirancang untuk memudahkan pengelolaan data pelanggan, transaksi, dan administrasi secara terintegrasi. Adapun fitur-fitur utama yang telah diimplementasikan dalam sistem ini antara lain:

1. Manajemen Data Pelanggan

Fitur ini memungkinkan admin untuk menambahkan data pelanggan secara lengkap, termasuk nama pelanggan, durasi waktu bermain yang diinginkan, serta pemesanan makanan atau minuman pendamping. Sistem ini dirancang dengan interface yang user-friendly agar proses pencatatan data menjadi lebih efisien, terorganisir, dan meminimalkan kesalahan input data. Selain itu, sistem juga dapat menyimpan informasi kontak pelanggan untuk keperluan komunikasi lebih lanjut.

2. Sistem Cetak Struk Otomatis

Setelah seluruh data transaksi dimasukkan oleh admin, sistem secara otomatis dapat mencetak struk sebagai bukti transaksi yang sah. Struk ini berisi detail lengkap pemesanan yang telah diinput, termasuk waktu mulai dan berakhirnya sesi bermain, biaya rental, serta total pembayaran. Hal ini memungkinkan pelanggan untuk mengecek kembali pesanan mereka dan berfungsi sebagai dokumentasi resmi transaksi.

3. Sistem Diskon Fleksibel

Sistem menyediakan opsi untuk menambahkan berbagai jenis diskon secara fleksibel. Fitur ini dapat dimanfaatkan dalam berbagai situasi seperti event khusus, promo musiman, program loyalitas pelanggan, atau diskon untuk member reguler. Admin dapat mengatur besaran diskon baik dalam bentuk persentase maupun nominal tetap sesuai dengan kebijakan bisnis dan kebutuhan promosi.

4. Ekspor Data ke Format PDF

Selain mencetak struk fisik, sistem juga memungkinkan admin untuk menyimpan data transaksi dalam format PDF yang dapat diakses secara digital. Fitur ini sangat memudahkan proses dokumentasi elektronik, pengarsipan transaksi jangka panjang, dan berbagi informasi dengan pihak terkait. Format PDF yang dihasilkan tetap mempertahankan format dan layout yang profesional.

5. Manajemen Riwayat Transaksi

Seluruh transaksi yang telah dilakukan akan tersimpan secara otomatis dalam database riwayat sistem. Admin memiliki akses penuh untuk melihat, mengedit, atau menghapus data transaksi sebelumnya sesuai kebutuhan. Fitur ini sangat membantu dalam berbagai aspek operasional seperti koreksi data yang salah input, pengecekan transaksi terdahulu, analisis pola pemesanan pelanggan, serta pelaporan keuangan berkala.

Keunggulan Sistem

Implementasi kelima fitur utama ini menjadikan sistem rental PlayStation kami sebagai solusi manajemen yang terintegrasi dan efisien. Sistem ini tidak hanya mempermudah proses administrasi harian, tetapi juga mendukung pengambilan keputusan bisnis melalui data transaksi yang tersimpan dengan baik. Dengan demikian, operasional rental PlayStation dapat berjalan lebih optimal dan profesional.

2.2 Penjelasan Konsep OOP di Kode Program

1. Enkapsulasi (Encapsulation)

Konsep enkapsulasi dalam aplikasi Rental PS diimplementasikan melalui pembungkusan data dan fungsi yang berkaitan dalam satu kesatuan kelas. Implementasi enkapsulasi terlihat jelas pada beberapa komponen sistem.

Kelas Handler mendemonstrasikan enkapsulasi dengan menyimpan komponen-komponen form dan tabel sebagai atribut privat. Kelas ini menyediakan metode-metode publik seperti `handleSaveTransaction()` yang berfungsi sebagai antarmuka untuk memanipulasi data internal. Pendekatan ini memastikan bahwa data internal kelas terlindungi dari akses langsung dari luar kelas.

Kelas Transaksi juga menerapkan enkapsulasi dengan menyimpan data sensitif seperti nama pelanggan, jenis PlayStation, dan durasi sewa dalam atribut yang tertutup. Proses perhitungan biaya sewa dikelola melalui fungsi internal yang tidak dapat diakses secara langsung dari luar kelas, sehingga integritas data terjaga.

2. Pewarisan (Inheritance)

Konsep pewarisan diterapkan secara strategis pada komponen antarmuka pengguna (GUI) aplikasi. Kelas-kelas seperti `FormComponents` dan `TableComponents` mewarisi komponen dari Java Swing framework, khususnya dari kelas dasar seperti `JPanel` dan `JFrame`.

Implementasi pewarisan ini memberikan keunggulan dalam pengembangan modular dan efisiensi kode. Pengembang tidak perlu membangun ulang fitur-fitur dasar dari komponen GUI, melainkan dapat memanfaatkan fungsionalitas yang sudah tersedia dari kelas induk dan menambahkan fitur spesifik sesuai kebutuhan aplikasi.

3. Polimorfisme (Polymorphism)

Polimorfisme dalam aplikasi Rental PS diimplementasikan melalui penggunaan lambda expression dan method reference dalam penanganan aksi pengguna. Konsep ini memungkinkan penggunaan satu antarmuka untuk berbagai implementasi yang berbeda.

Contoh implementasi polimorfisme terlihat ketika pengguna melakukan interaksi dengan tombol dalam aplikasi. Method seperti `handleSaveTransaction()` dipanggil melalui `addActionListener`, dimana berbagai method yang berbeda dapat digunakan dalam konteks yang sama karena semuanya mengimplementasikan interface `ActionListener`. Fleksibilitas ini memungkinkan program untuk merespons berbagai jenis aksi pengguna dengan cara yang konsisten namun dapat disesuaikan.

4. Abstraksi (Abstraction)

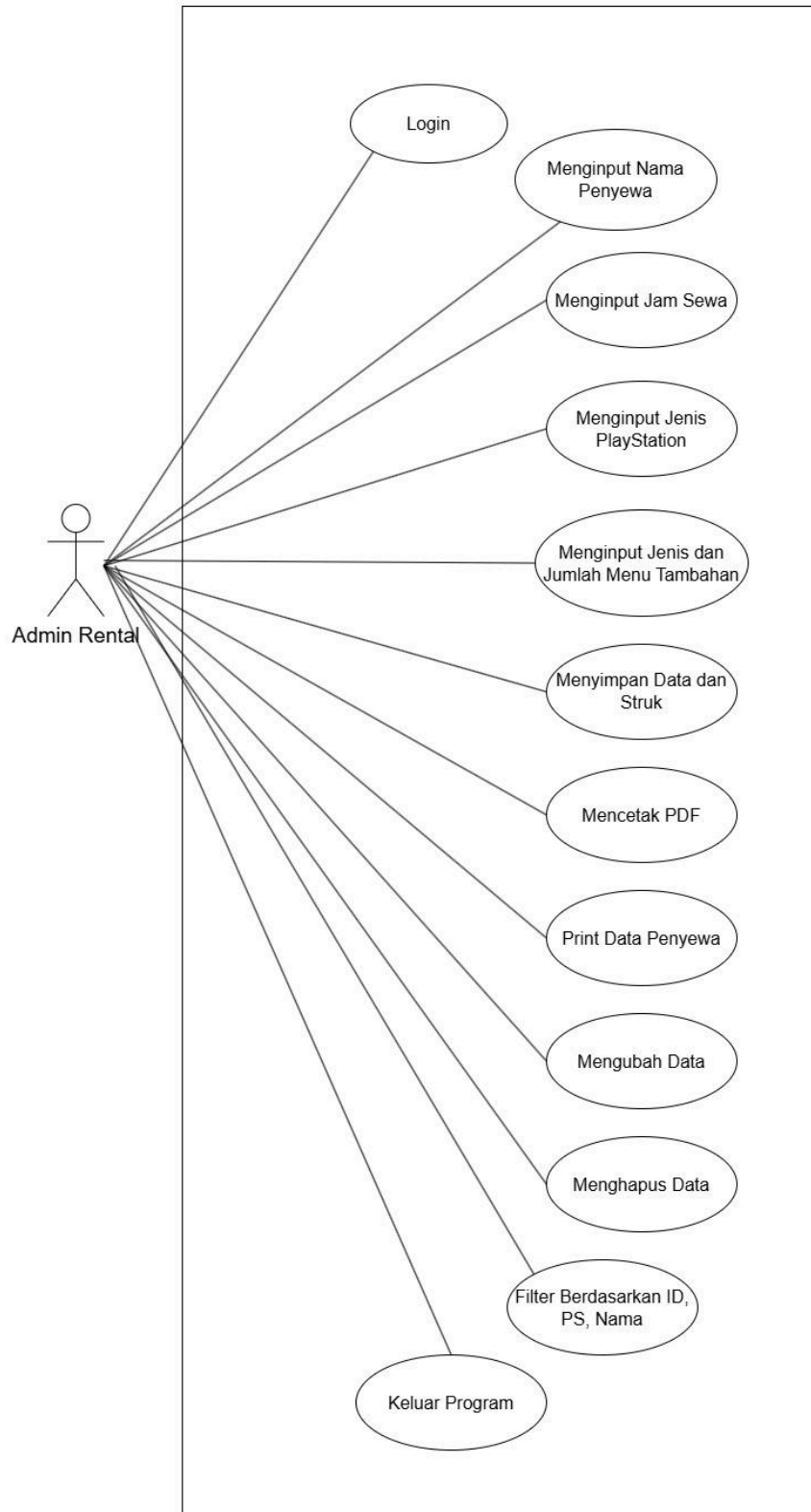
Abstraksi diterapkan melalui pemisahan berbagai aspek sistem ke dalam kelas-kelas yang memiliki tanggung jawab spesifik. Pembagian ini menciptakan struktur yang modular dan mudah dipelihara.

Kelas Handler bertugas menangani logika bisnis dan interaksi dengan input pengguna. Kelas `RentalPSAPP` bertanggung jawab untuk menyusun dan mengatur antarmuka program secara keseluruhan. Kelas `DBConnection` menyembunyikan kompleksitas detail koneksi database dan eksekusi query SQL dari bagian lain sistem. Sementara itu, kelas `Struk` memiliki tanggung jawab khusus untuk mencetak bukti transaksi.

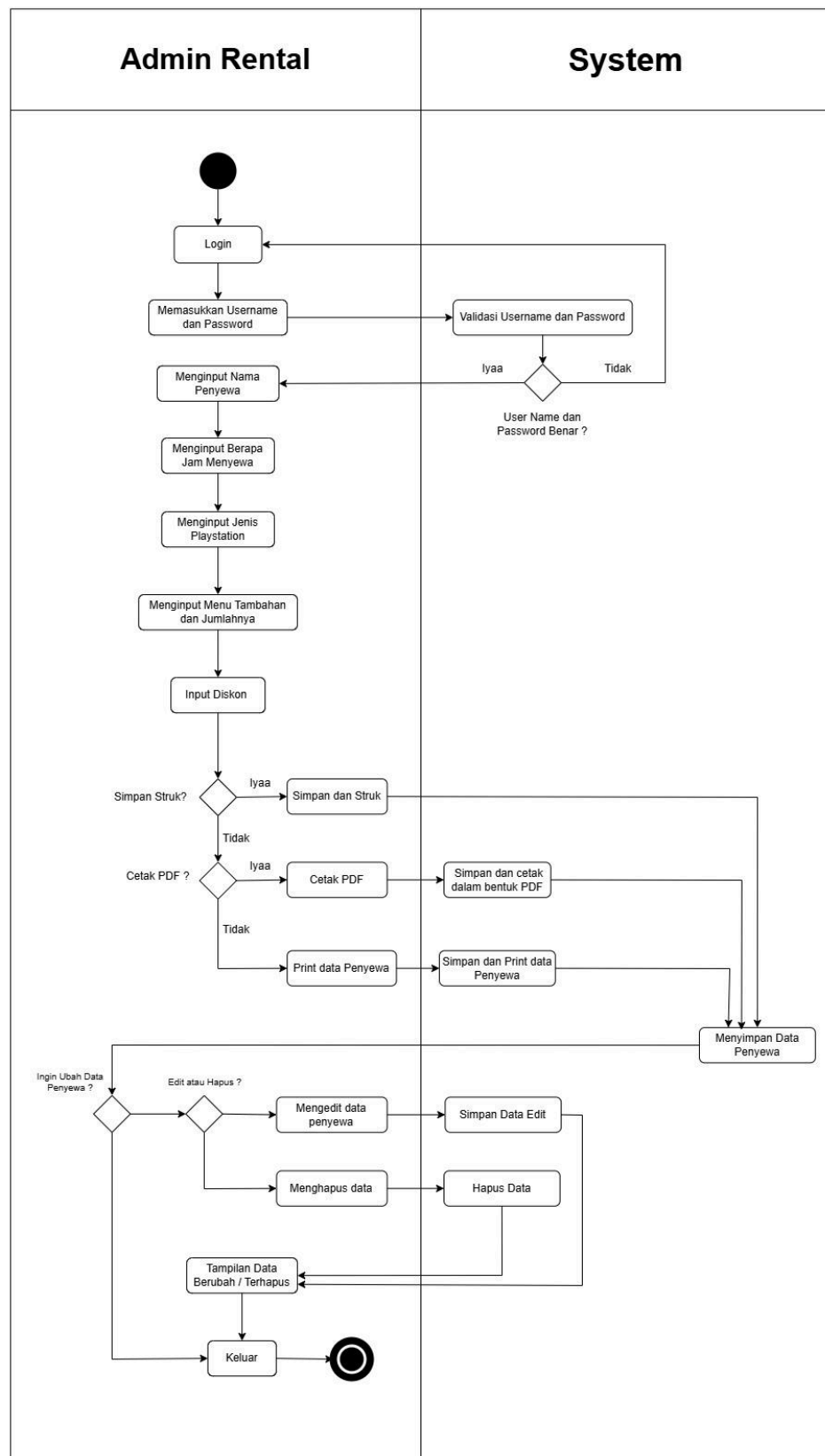
Pemisahan tanggung jawab ini memudahkan proses pengembangan dan pemeliharaan kode karena setiap komponen memiliki fungsi yang jelas dan tidak saling tumpang tindih. Hal ini juga memungkinkan pengembangan paralel dan modifikasi sistem tanpa mempengaruhi komponen lain.

2.3 Unified Modeling Language

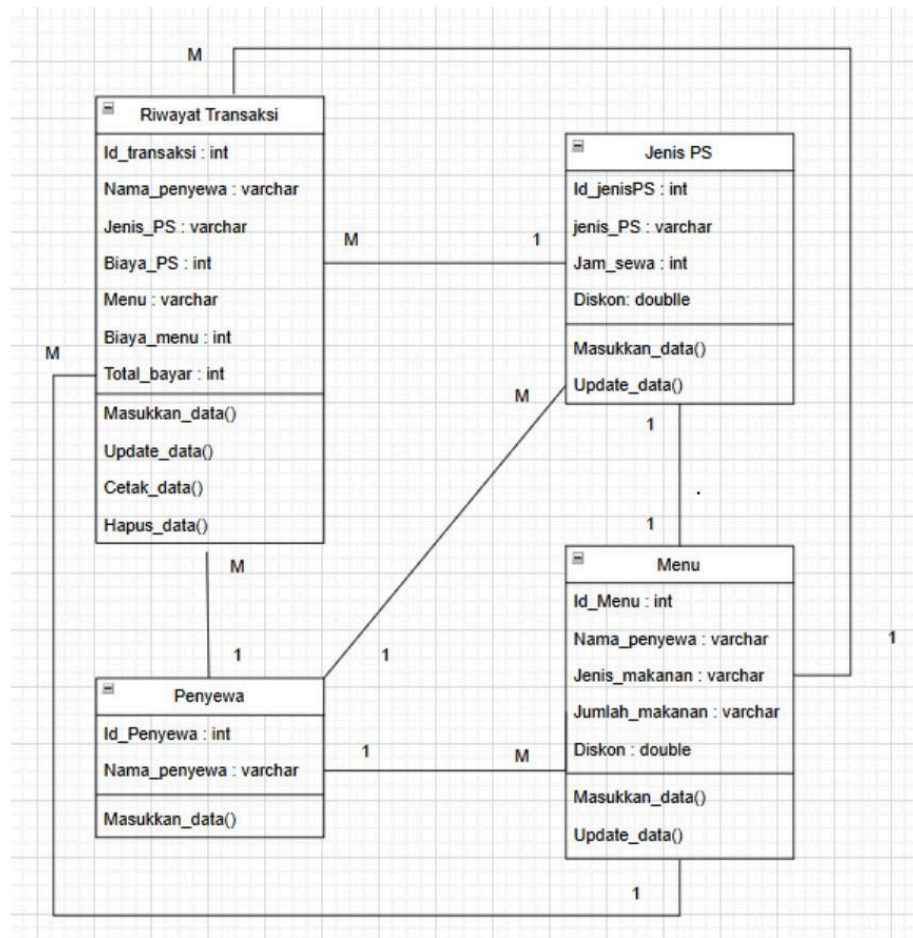
- Use Case



- Activity Diagram



- **Class Diagram**



2.4 Implementasi Kode Program dan Pengujian

- **Main.java**

```

1 package UAS;
2
3 public class Main {
4     public static void main(String[] args) {
5         // Set system properties untuk mendukung GUI di berbagai platform
6         System.setProperty("java.awt.headless", "false");
7
8         try {
9             // Jalankan aplikasi
10            RentalPSAPP.main(args);
11        } catch (Exception e) {
12            System.err.println("Error saat menjalankan aplikasi: " + e.getMessage());
13            e.printStackTrace();
14        }
15    }
16 }
  
```

● Login.java

```
1 package UAS;
2
3 import javax.swing.*;
4 import java.awt.*;
5
6 public class Login {
7     private static final String ADMIN_USERNAME = "admin";
8     private static final String ADMIN_PASSWORD = "admin123";
9
10    public static boolean authenticateUser() {
11        try {
12            for (UIManager.LookAndFeelInfo info : UIManager.getInstalledLookAndFeels()) {
13                if ("Nimbus".equals(info.getName())) {
14                    UIManager.setLookAndFeel(info.getClassName());
15                    break;
16                }
17            }
18        } catch (Exception e) {
19            System.out.println("Gagal menerapkan Nimbus LookAndFeel: " + e.getMessage());
20        }
21
22        JPanel panel = new JPanel();
23        panel.setLayout(new BorderLayout(panel, BorderLayout.Y_AXIS));
24        panel.setPreferredSize(new Dimension(400, 300));
25        panel.setBackground(Color.WHITE);
26        panel.setBorder(BorderFactory.createEmptyBorder(20, 30, 20, 30)); // padding luar
27
28        // Spasi dari atas
29        panel.add(Box.createVerticalStrut(20));
30
31        // Label SISTEM PARKIR
32        JLabel lblTitle = new JLabel("SISTEM PARKIR", SwingConstants.CENTER);
33        lblTitle.setFont(new Font("Arial", Font.BOLD, 22));
34        lblTitle.setForeground(new Color(33, 99, 255));
35        lblTitle.setAlignmentX(Component.CENTER_ALIGNMENT);
36        panel.add(lblTitle);
37
38        // Spasi antara judul dan form
39        panel.add(Box.createVerticalStrut(20));
40
41        // Form login
42        JPanel formPanel = new JPanel(new GridBagLayout());
43        formPanel.setOpaque(false);
44        GridBagConstraints gbc = new GridBagConstraints();
45        gbc.insets = new Insets(8, 8, 8, 8);
46        gbc.anchor = GridBagConstraints.WEST;
47
48        JLabel lblUser = new JLabel("Username:");
49        JTextField tfUser = new JTextField(15);
50        JLabel lblPass = new JLabel("Password:");
51        JPasswordField pfPass = new JPasswordField(15);
52
53        gbc.gridx = 0;
54        gbc.gridy = 0;
55        formPanel.add(lblUser, gbc);
56        gbc.gridx = 1;
57        formPanel.add(tfUser, gbc);
58
59        gbc.gridx = 0;
60        gbc.gridy = 1;
61        formPanel.add(lblPass, gbc);
62        gbc.gridx = 1;
63        formPanel.add(pfPass, gbc);
64
65        panel.add(formPanel);
66
67        // Spasi sebelum tombol login
68        panel.add(Box.createVerticalStrut(10));
69
70        // Tombol login
71        JButton btnLogin = new JButton("LOGIN");
72        btnLogin.setBackground(new Color(33, 99, 255));
73        btnLogin.setForeground(Color.WHITE);
74        btnLogin.setFocusPainted(false);
75        JPanel btnPanel = new JPanel();
76        btnPanel.setOpaque(false);
77        btnPanel.add(btnLogin);
78        panel.add(btnPanel);
79
80        // Tampilkan dialog
81        JDialog dialog = new JDialog((Frame) null, "Login", true);
82        dialog.setContentPane(panel);
83        dialog.pack();
84        dialog.setLocationRelativeTo(null);
85
86        final boolean[] loginSuccess = { false };
87
88        btnLogin.addActionListener(e -> {
89            String username = tfUser.getText().trim();
90            String password = new String(pfPass.getPassword());
91
92            if (ADMIN_USERNAME.equals(username) && ADMIN_PASSWORD.equals(password)) {
93                loginSuccess[0] = true;
94                dialog.dispose();
95            } else {
96                JOptionPane.showMessageDialog(dialog, "Username atau password salah.", "Login Gagal",
97                    JOptionPane.ERROR_MESSAGE);
98            }
99        });
100
101        dialog.setVisible(true);
102        return loginSuccess[0];
103    }
104
105    public static boolean confirmExit(JFrame parentFrame) {
106        int confirm = JOptionPane.showConfirmDialog(
107            parentFrame,
108            "Vakin ingin keluar?",
109            "Konfirmasi Keluar",
110            JOptionPane.YES_NO_OPTION);
111        return confirm == JOptionPane.YES_OPTION;
112    }
113 }
114
```

● DBConnection.java

```
1 package UAS;
2
3 import java.sql.*;
4 import javax.swing.table.DefaultTableModel;
5
6 public class DBConnection {
7     private static final String DB_URL = "jdbc:mysql://localhost:3306/db_rentalps";
8     private static final String DB_USER = "root";
9     private static final String DB_PASSWORD = "";
10
11     public static Connection getConnection() {
12         try {
13             Class.forName("com.mysql.cj.jdbc.Driver");
14             return DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
15         } catch (Exception e) {
16             e.printStackTrace();
17             return null;
18         }
19     }
20
21     public static void loadDataTable(DefaultTableModel tableModel) {
22         tableModel.setRowCount(0);
23         try (Connection conn = getConnection()) {
24             String sql = "SELECT * FROM riwayat_transaksi";
25             Statement stmt = conn.createStatement();
26             ResultSet rs = stmt.executeQuery(sql);
27             while (rs.next()) {
28                 tableModel.addRow(new Object[] {
29                     rs.getInt("id"), rs.getString("nama"), rs.getString("jenis_ps"),
30                     rs.getInt("jam"), rs.getInt("biaya_ps"), rs.getString("menu"),
31                     rs.getInt("biaya_menu"), rs.getInt("total_bayar")
32                 });
33             }
34         } catch (SQLException e) {
35             e.printStackTrace();
36         }
37     }
38
39     public static void searchData(String keyword, DefaultTableModel tableModel) {
40         tableModel.setRowCount(0);
41         try (Connection conn = getConnection()) {
42             String sql = "SELECT * FROM riwayat_transaksi WHERE nama LIKE ? OR jenis_ps LIKE ? OR id LIKE ?";
43             PreparedStatement ps = conn.prepareStatement(sql);
44             String query = "%" + keyword + "%";
45             ps.setString(1, query);
46             ps.setString(2, query);
47             ps.setString(3, query);
48             ResultSet rs = ps.executeQuery();
49             while (rs.next()) {
50                 tableModel.addRow(new Object[] {
51                     rs.getInt("id"), rs.getString("nama"), rs.getString("jenis_ps"),
52                     rs.getInt("jam"), rs.getInt("biaya_ps"), rs.getString("menu"),
53                     rs.getInt("biaya_menu"), rs.getInt("total_bayar")
54                 });
55             }
56         } catch (SQLException e) {
57             e.printStackTrace();
58         }
59     }
60
61     public static int insertTransaction(String nama, String jenisPS, int jam, int biayaPS,
62         String menu, int biayaMenu, int totalBayar) {
63         int lastId = 0;
64         try (Connection conn = getConnection()) {
65             String sql = "INSERT INTO riwayat_transaksi (nama, jenis_ps, jam, biaya_ps, menu, biaya_menu, total_bayar) VALUES (?, ?, ?, ?, ?, ?, ?)";
66             PreparedStatement ps = conn.prepareStatement(sql, Statement.RETURN_GENERATED_KEYS);
67             ps.setString(1, nama);
68             ps.setString(2, jenisPS);
69             ps.setInt(3, jam);
70             ps.setInt(4, biayaPS);
71             ps.setString(5, menu);
72             ps.setInt(6, biayaMenu);
73             ps.setInt(7, totalBayar);
74             ps.executeUpdate();
75             ResultSet rs = ps.getGeneratedKeys();
76             if (rs.next()) {
77                 lastId = rs.getInt(1);
78             }
79         } catch (SQLException e) {
80             e.printStackTrace();
81         }
82         return lastId;
83     }
84
85     public static boolean updateTransaction(int id, String nama, int jam) {
86         try (Connection conn = getConnection()) {
87             String sql = "UPDATE riwayat_transaksi SET nama=?, jam=? WHERE id=?";
88             PreparedStatement ps = conn.prepareStatement(sql);
89             ps.setString(1, nama);
90             ps.setInt(2, jam);
91             ps.setInt(3, id);
92             ps.executeUpdate();
93             return true;
94         } catch (SQLException e) {
95             e.printStackTrace();
96             return false;
97         }
98     }
99
100     public static boolean deleteTransaction(int id) {
101         try (Connection conn = getConnection()) {
102             String sql = "DELETE FROM riwayat_transaksi WHERE id=?";
103             PreparedStatement ps = conn.prepareStatement(sql);
104             ps.setInt(1, id);
105             ps.executeUpdate();
106             return true;
107         } catch (SQLException e) {
108             e.printStackTrace();
109             return false;
110         }
111     }
112 }
```

● Struk.java

```
1 package UAS;
2
3 import java.awt.*;
4 import java.awt.print.*;
5 import java.io.*;
6 import java.text.SimpleDateFormat;
7 import java.util.Date;
8 import javax.swing.*;
9 import com.itextpdf.text.*;
10 import com.itextpdf.text.Rectangle;
11 import com.itextpdf.text.pdf.*;
12
13 public class Struk {
14
15     public static String generateReceiptText(transaksi transaction, int transactionId) {
16         String waktu = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date());
17
18         return "\nSTRUK RENTAL PS\n" +
19             "ID Transaksi: " + transactionId + "\n" +
20             "Nama: " + transaction.getName() + "\n" +
21             "Jenis PS: " + transaction.getJenisPSFull() + "\n" +
22             "Jam: " + transaction.getJam() + " jam\n" +
23             "Biaya PS: Rp " + transaction.getBiayaPS() + "\n" +
24             "Menu: " + transaction.getMenu() + "\n" +
25             "Biaya Menu: Rp " + transaction.getBiayaMenu() + "\n" +
26             "Diskon: Rp " + transaction.getDiskon() + "\n" +
27             "Total Bayar: Rp " + transaction.getTotalBayar() + "\n" +
28             "Waktu: " + waktu + "\n";
29     }
30
31     public static void generatePDF(String receiptText) {
32         try {
33             JFileChooser fileChooser = new JFileChooser();
34             fileChooser.setDialogTitle("Simpan Struk PDF");
35             if (fileChooser.showSaveDialog(null) == JFileChooser.APPROVE_OPTION) {
36                 String filename = fileChooser.getSelectedFile().getAbsolutePath();
37                 if (!filename.toLowerCase().endsWith(".pdf")) {
38                     filename += ".pdf";
39                 }
40
41                 Document doc = new Document(new Rectangle(226, 600));
42                 PdfWriter.getInstance(doc, new FileOutputStream(filename));
43                 doc.open();
44
45                 Font fontTitle = new Font(Font.FontFamily.COURIER, 10, Font.BOLD);
46                 Font fontBody = new Font(Font.FontFamily.COURIER, 8);
47
48                 // Judul
49                 Paragraph title = new Paragraph("STRUK TRANSAKSI RENTAL PS\n", fontTitle);
50                 title.setAlignment(Element.ALIGN_CENTER);
51                 title.setSpacingAfter(5);
52                 doc.add(title);
53
54                 // Garis pembatas
55                 Paragraph separator = new Paragraph("-----", fontBody);
56                 separator.setAlignment(Element.ALIGN_CENTER);
57                 separator.setSpacingAfter(5);
58                 doc.add(separator);
59
60                 // Isi struk baris demi baris
61                 String[] lines = receiptText.split("\n");
62                 for (String line : lines) {
63                     if (!line.trim().isEmpty()) {
64                         Paragraph p = new Paragraph(line, fontBody);
65                         p.setAlignment(Element.ALIGN_LEFT);
66                         doc.add(p);
67                     }
68                 }
69
70                 // Garis bawah + ucapan
71                 doc.add(new Paragraph("-----", fontBody));
72                 Paragraph thanks = new Paragraph("Terima kasih atas kunjungan Anda!", fontBody);
73                 thanks.setAlignment(Element.ALIGN_CENTER);
74                 thanks.setSpacingBefore(10);
75                 doc.add(thanks);
76
77                 doc.close();
78
79                 // Buka file PDF setelah selesai
80                 if (Desktop.isDesktopSupported()) {
81                     Desktop.getDesktop().open(new File(filename));
82                 }
83
84                 JOptionPane.showMessageDialog(null, "Struk berhasil disimpan!");
85             } catch (Exception e) {
86                 e.printStackTrace();
87                 JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());
88             }
89         }
90     }
91
92     public static void printReceipt(String receiptText) {
93         try {
94             PrinterJob job = PrinterJob.getPrinterJob();
95             job.setPrintable(graphics, pageFormat, pageIndex -> {
96                 if (pageIndex > 0) {
97                     return Printable.NO_SUCH_PAGE;
98                 }
99
100                 Graphics2D g2d = (Graphics2D) graphics;
101                 g2d.translate(pageFormat.getImageableX(), pageFormat.getImageableY());
102
103                 String[] lines = receiptText.split("\n");
104                 int yPos = 100;
105
106                 for (String line : lines) {
107                     graphics.drawString(line, 50, yPos);
108                     yPos += 15;
109                 }
110
111                 return Printable.PAGE_EXISTS;
112             });
113
114             if (job.printDialog()) {
115                 job.print();
116             }
117         } catch (Exception ex) {
118             ex.printStackTrace();
119         }
120     }
121 }
```

● Transaksi.java

```
1 package UAS;
2
3 public class Transaksi {
4     private String nama;
5     private String jenisPS;
6     private int jam;
7     private int biayaPS;
8     private String menu;
9     private int biayaMenu;
10    private int totalBayar;
11    private int diskon;
12
13    public Transaksi(String nama, String jenisPS, int jam, int mie, int teh, int kopi, String diskonType) {
14        this.nama = nama;
15        this.jenisPS = jenisPS;
16        this.jam = jam;
17
18        // Hitung biaya PS
19        int tarif = jenisPS.contains("PS3") ? 5000 : jenisPS.contains("PS4") ? 8000 : 12000;
20        this.biayaPS = tarif * jam;
21
22        // Hitung biaya menu
23        this.biayaMenu = mie * 8000 + teh * 5000 + kopi * 6000;
24
25        // Format menu string
26        StringBuilder menuBuilder = new StringBuilder();
27        if (mie > 0)
28            menuBuilder.append("Mie x").append(mie).append(", ");
29        if (teh > 0)
30            menuBuilder.append("Es Teh x").append(teh).append(", ");
31        if (kopi > 0)
32            menuBuilder.append("Es Kopi x").append(kopi).append(", ");
33
34        if (menuBuilder.length() > 0) {
35            this.menu = menuBuilder.substring(0, menuBuilder.length() - 2);
36        } else {
37            this.menu = "-";
38        }
39
40        // Hitung diskon
41        int total = this.biayaPS + this.biayaMenu;
42        if (diskonType.contains("10%")) {
43            this.diskon = total * 10 / 100;
44        } else if (diskonType.contains("-5000")) {
45            this.diskon = 5000;
46        } else {
47            this.diskon = 0;
48        }
49
50        this.totalBayar = total - this.diskon;
51    }
52
53    // Getters
54    public String getName() {
55        return nama;
56    }
57
58    public String getJenisPS() {
59        return jenisPS.split("-")[0];
60    }
61
62    public String getJenisPSFull() {
63        return jenisPS;
64    }
65
66    public int getJam() {
67        return jam;
68    }
69
70    public int getBiayaPS() {
71        return biayaPS;
72    }
73
74    public String getMenu() {
75        return menu;
76    }
77
78    public int getBiayaMenu() {
79        return biayaMenu;
80    }
81
82    public int getTotalBayar() {
83        return totalBayar;
84    }
85
86    public int getDiskon() {
87        return diskon;
88    }
89
90    // Setters
91    public void setName(String nama) {
92        this.nama = nama;
93    }
94
95    public void setJenisPS(String jenisPS) {
96        this.jenisPS = jenisPS;
97    }
98
99    public void setJam(int jam) {
100        this.jam = jam;
101    }
102 }
```

● RentalPSAPP.java

```
1 package UAS;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 public class RentalPSAPP {
7     private JFrame frame;
8     private GUI.FormComponents formComponents;
9     private GUI.TableComponents tableComponents;
10    private Handler eventHandlers;
11
12    public RentalPSAPP() {
13        initializeApplication();
14    }
15
16    private void initializeApplication() {
17        // Authenticate user first
18        if (!Login.authenticateUser()) {
19            System.exit(0);
20        }
21
22        // Initialize components
23        frame = new JFrame("Rental PS - Kasir");
24        frame.setSize(1100, 800);
25        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26
27        formComponents = new GUI.FormComponents();
28        tableComponents = new GUI.TableComponents();
29        eventHandlers = new Handler(formComponents, tableComponents);
30
31        // Setup GUI
32        GUI.setupLayout(frame, formComponents, tableComponents);
33        GUI.applyTheme(frame);
34
35        // Setup event handlers
36        setupEventHandlers();
37
38        // Load initial data
39        DBConnection.loadDataToTable(tableComponents.tableModel);
40
41        // Show frame
42        frame.setLocationRelativeTo(null);
43        frame.setVisible(true);
44    }
45
46    private void setupEventHandlers() {
47        // Get buttons from frame
48        JButton btnSimpan = findButton("Simpan + Struk");
49        JButton btnPDF = findButton("PDF");
50        JButton btnPrint = findButton("Cetak");
51        JButton btnCari = findButton("Cari");
52        JButton btnUbah = findButton("Ubah");
53        JButton btnHapus = findButton("Hapus");
54        JButton btnKeluar = findButton("Keluar");
55
56        // Add event listeners
57        if (btnSimpan != null)
58            btnSimpan.addActionListener(eventHandlers::handleSaveTransaction);
59        if (btnPDF != null)
60            btnPDF.addActionListener(eventHandlers::handleGeneratePDF);
61        if (btnPrint != null)
62            btnPrint.addActionListener(eventHandlers::handlePrintReceipt);
63        if (btnCari != null)
64            btnCari.addActionListener(eventHandlers::handleSearch);
65        if (btnUbah != null)
66            btnUbah.addActionListener(eventHandlers::handleUpdate);
67        if (btnHapus != null)
68            btnHapus.addActionListener(eventHandlers::handleDelete);
69        if (btnKeluar != null)
70            btnKeluar.addActionListener(e -> handleExit());
71
72        // Add table selection listener
73        tableComponents.table.addMouseListener(new MouseAdapter() {
74            public void mouseClicked(MouseEvent e) {
75                eventHandlers.handleTableSelection();
76            }
77        });
78    }
79
80    private JButton findButton(String text) {
81        return findButtonRecursive(frame, text);
82    }
83
84    private JButton findButtonRecursive(java.awt.Container container, String text) {
85        for (java.awt.Component component : container.getComponents()) {
86            if (component instanceof JButton) {
87                JButton button = (JButton) component;
88                if (text.equals(button.getText())) {
89                    return button;
90                }
91            } else if (component instanceof java.awt.Container) {
92                JButton result = findButtonRecursive((java.awt.Container) component, text);
93                if (result != null) {
94                    return result;
95                }
96            }
97        }
98        return null;
99    }
100
101    private void handleExit() {
102        if (Login.confirmExit(frame)) {
103            System.exit(0);
104        }
105    }
106
107    public static void main(String[] args) {
108        SwingUtilities.invokeLater(() -> {
109            new RentalPSAPP();
110        });
111    }
112 }
```


● Handler.java

```
1 package UAS;
2
3 import javax.swing.*;
4 import java.awt.event.ActionEvent;
5
6 public class Handler {
7     private GUI.FormComponents formComponents;
8     private GUI.TableComponents tableComponents;
9     private String currentReceiptText = "";
10
11     public Handler(GUI.FormComponents formComponents,
12         GUI.TableComponents tableComponents) {
13         this.formComponents = formComponents;
14         this.tableComponents = tableComponents;
15     }
16
17     public void handleSaveTransaction(ActionEvent e) {
18         try {
19             // Validate input
20             if (formComponents.tfNama.getText().trim().isEmpty()) {
21                 JOptionPane.showMessageDialog(null, "Nama penyewa harus diisi!");
22                 return;
23             }
24
25             if (formComponents.tfJam.getText().trim().isEmpty()) {
26                 JOptionPane.showMessageDialog(null, "Jam sewa harus diisi!");
27                 return;
28             }
29
30             // Get input values
31             String nama = formComponents.tfNama.getText().trim();
32             int jam = Integer.parseInt(formComponents.tfJam.getText());
33             String jenisPS = formComponents.cbPS.getSelectedItem().toString();
34             int mie = (int) formComponents.spMie.getValue();
35             int teh = (int) formComponents.spTeh.getValue();
36             int kopi = (int) formComponents.spKopi.getValue();
37             String diskonType = formComponents.cbDiskon.getSelectedItem().toString();
38
39             // Create transaction model
40             Transaksi transaction = new Transaksi(nama, jenisPS, jam, mie, teh, kopi, diskonType);
41
42             // Save to database
43             int transactionId = DBConnection.insertTransaction(
44                 transaction.getNama(),
45                 transaction.getJenisPS(),
46                 transaction.getJam(),
47                 transaction.getBiayaPS(),
48                 transaction.getMenu(),
49                 transaction.getBiayaMenu(),
50                 transaction.getTotalBayar());
51
52             // Generate receipt
53             currentReceiptText = Struk.generateReceiptText(transaction, transactionId);
54             formComponents.areaOutput.setText(currentReceiptText);
55
56             // Refresh table
57             DBConnection.loadDataToTable(tableComponents.tableModel);
58
59             // Clear form
60             formComponents.clearForm();
61
62             JOptionPane.showMessageDialog(null, "Transaksi berhasil disimpan.");
63
64         } catch (NumberFormatException ex) {
65             JOptionPane.showMessageDialog(null, "Jam sewa harus berupa angka!");
66         } catch (Exception ex) {
67             JOptionPane.showMessageDialog(null, "Terjadi kesalahan: " + ex.getMessage());
68             ex.printStackTrace();
69         }
70     }
71
72     public void handleGeneratePDF(ActionEvent e) {
73         if (currentReceiptText.isEmpty()) {
74             JOptionPane.showMessageDialog(null,
75                 "Tidak ada struk untuk dicetak. Silakan buat transaksi terlebih dahulu.");
76             return;
77         }
78         Struk.generatePDF(currentReceiptText);
79     }
80 }
```

```

80
81
82 public void handlePrintReceipt(ActionEvent e) {
83     if (currentReceiptText.isEmpty()) {
84         JOptionPane.showMessageDialog(null,
85             "Tidak ada struk untuk dicetak. Silakan buat transaksi terlebih dahulu.");
86         return;
87     }
88     Struk.printReceipt(currentReceiptText);
89 }
90
91 public void handleSearch(ActionEvent e) {
92     String keyword = formComponents.tfCarI.getText().trim();
93     if (!keyword.isEmpty()) {
94         DBConnection.searchData(keyword, tableComponents.tableModel);
95     } else {
96         DBConnection.loadDataToTable(tableComponents.tableModel);
97     }
98 }
99
100 public void handleUpdate(ActionEvent e) {
101     try {
102         if (formComponents.tfIdHidden.getText().isEmpty()) {
103             JOptionPane.showMessageDialog(null, "Pilih data yang akan diubah dari tabel!");
104             return;
105         }
106         if (formComponents.tfNama.getText().trim().isEmpty()) {
107             JOptionPane.showMessageDialog(null, "Nama penyewa harus diisi!");
108             return;
109         }
110         if (formComponents.tfJam.getText().trim().isEmpty()) {
111             JOptionPane.showMessageDialog(null, "Jam sewa harus diisi!");
112             return;
113         }
114     }
115     int id = Integer.parseInt(formComponents.tfIdHidden.getText());
116     String nama = formComponents.tfNama.getText().trim();
117     int jam = Integer.parseInt(formComponents.tfJam.getText());
118     if (DBConnection.updateTransaction(id, nama, jam)) {
119         DBConnection.loadDataToTable(tableComponents.tableModel);
120         formComponents.clearForm();
121         JOptionPane.showMessageDialog(null, "Data berhasil diubah.");
122     } else {
123         JOptionPane.showMessageDialog(null, "Gagal mengubah data.");
124     }
125 }
126
127 catch (NumberFormatException ex) {
128     JOptionPane.showMessageDialog(null, "ID dan Jam harus berupa angka!");
129 }
130 catch (Exception ex) {
131     JOptionPane.showMessageDialog(null, "Terjadi kesalahan: " + ex.getMessage());
132     ex.printStackTrace();
133 }
134 }
135
136 public void handleDelete(ActionEvent e) {
137     try {
138         if (formComponents.tfIdHidden.getText().isEmpty()) {
139             JOptionPane.showMessageDialog(null, "Pilih data yang akan dihapus dari tabel!");
140             return;
141         }
142     }
143     int confirm = JOptionPane.showConfirmDialog(
144         null,
145         "Yakin ingin menghapus data ini?",
146         "Konfirmasi Hapus",
147         JOptionPane.YES_NO_OPTION);
148     if (confirm == JOptionPane.YES_OPTION) {
149         int id = Integer.parseInt(formComponents.tfIdHidden.getText());
150         if (DBConnection.deleteTransaction(id)) {
151             DBConnection.loadDataToTable(tableComponents.tableModel);
152             formComponents.clearForm();
153             JOptionPane.showMessageDialog(null, "Data berhasil dihapus.");
154         } else {
155             JOptionPane.showMessageDialog(null, "Gagal menghapus data.");
156         }
157     }
158 }
159
160 catch (NumberFormatException ex) {
161     JOptionPane.showMessageDialog(null, "ID harus berupa angka!");
162 }
163 catch (Exception ex) {
164     JOptionPane.showMessageDialog(null, "Terjadi kesalahan: " + ex.getMessage());
165     ex.printStackTrace();
166 }
167 }
168
169 public void handleTableSelection() {
170     try {
171         int selectedRow = tableComponents.table.getSelectedRow();
172         if (selectedRow >= 0) {
173             // Fill form with selected data
174             formComponents.tfIdHidden.setText(tableComponents.table.getValueAt(selectedRow, 0).toString());
175             formComponents.tfNama.setText(tableComponents.table.getValueAt(selectedRow, 1).toString());
176             formComponents.tfJam.setText(tableComponents.table.getValueAt(selectedRow, 3).toString());
177
178             // Set PS type
179             String psType = tableComponents.table.getValueAt(selectedRow, 2).toString();
180             String biayaPS = tableComponents.table.getValueAt(selectedRow, 4).toString();
181             String fullPSType = psType + " - Rp " + biayaPS;
182
183             for (int i = 0; i < formComponents.cbPS.getItemCount(); i++) {
184                 if (formComponents.cbPS.getItemAt(i).startsWith(psType)) {
185                     formComponents.cbPS.setSelectedIndex(i);
186                     break;
187                 }
188             }
189
190             formComponents.areaOutput
191                 .setText("Data siap diedit atau dihapus. ID: " + formComponents.tfIdHidden.getText());
192         }
193     } catch (Exception ex) {
194         ex.printStackTrace();
195     }
196 }
197 }

```

● GUI.java

```
1 package UAS;
2
3 import javax.swing.*;
4 import javax.swing.table.DefaultTableModel;
5 import java.awt.*;
6
7 public class GUI {
8
9     public static class FormComponents {
10         public JTextField tfNama;
11         public JTextField tfJam;
12         public JComboBox<String> cbPS;
13         public JSpinner spMie;
14         public JSpinner spTeh;
15         public JSpinner spKopi;
16         public JComboBox<String> cbDiskon;
17         public JTextArea areaOutput;
18         public JTextField tfIdHidden;
19         public JTextField tfCar1;
20
21         public FormComponents() {
22             initializeComponents();
23         }
24
25         private void initializeComponents() {
26             tfNama = new JTextField();
27             tfJam = new JTextField();
28
29             String[] pilihanPS = { "PS3 - Rp 5000", "PS4 - Rp 8000", "PS5 - Rp 12000" };
30             cbPS = new JComboBox<>(pilihanPS);
31
32             spMie = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1));
33             spTeh = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1));
34             spKopi = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1));
35
36             String[] diskonOptions = { "Tanpa Diskon", "Diskon Pelajar (10%)", "Paket 2 Jam + Minuman -5000" };
37             cbDiskon = new JComboBox<>(diskonOptions);
38
39             areaOutput = new JTextArea();
40             areaOutput.setEditable(false);
41             areaOutput.setBorder(BorderFactory.createTitledBorder("Struk"));
42
43             tfIdHidden = new JTextField();
44             tfIdHidden.setVisible(false);
45
46             tfCar1 = new JTextField();
47         }
48
49         public void clearForm() {
50             tfNama.setText("");
51             tfJam.setText("");
52             cbPS.setSelectedIndex(0);
53             spMie.setValue(0);
54             spTeh.setValue(0);
55             spKopi.setValue(0);
56             cbDiskon.setSelectedIndex(0);
57             areaOutput.setText("");
58             tfIdHidden.setText("");
59         }
60     }
61
62     public static class TableComponents {
63         public DefaultTableModel tableModel;
64         public JTable table;
65
66         public TableComponents() {
67             String[] columns = { "ID", "Nama", "PS", "Jam", "Biaya PS", "Menu", "Biaya Menu", "Total Bayar" };
68             tableModel = new DefaultTableModel(columns, 0);
69             table = new JTable(tableModel);
70             table.setRowHeight(22); // Reduced from 25
71         }
72     }
73
74     public static void setupLayout(JFrame frame, FormComponents form, TableComponents tableComp) {
75         frame.setLayout(null);
76
77         // Optimized constants for better spacing
78         int leftMargin = 15;
79         int topMargin = 10;
80         int panelHeight = 55; // Reduced from 60
81         int buttonHeight = 28; // Reduced from 30
82         int fieldHeight = 24; // Reduced from 25
83
84         int currentY = topMargin;
85
86         // === ROW 1: Basic Information ===
87         JPanel basicInfoPanel = new JPanel();
88         basicInfoPanel.setBorder(BorderFactory.createTitledBorder("Form Rental PS"));
89         basicInfoPanel.setLayout(null);
90         basicInfoPanel.setBounds(leftMargin, currentY, 1050, panelHeight);
91
92         // Nama Penyewa
93         addLabelAndComponent(basicInfoPanel, "Nama Penyewa:", form.tfNama, 15, 22,
94             100, 200, fieldHeight);
95
96         // Jam Sewa
97         addLabelAndComponent(basicInfoPanel, "Jam Sewa:", form.tfJam, 370, 22,
98             60, 80, fieldHeight);
99
100         // Jenis PS
101         addLabelAndComponent(basicInfoPanel, "Jenis PS:", form.cbPS, 580, 22,
102             55, 180, fieldHeight);
103
104         frame.add(basicInfoPanel);
105         currentY += panelHeight + 5; // Reduced gap
106
107         // === ROW 2: Menu Selection ===
108         JPanel menuPanel = new JPanel();
109         menuPanel.setBorder(BorderFactory.createTitledBorder("Pilihan Menu"));
110         menuPanel.setLayout(null);
111         menuPanel.setBounds(leftMargin, currentY, 1050, panelHeight);
112
113         addLabelAndComponent(menuPanel, "Mie Instan (Rp 8000):", form.spMie, 15, 22,
114             120, 60, fieldHeight);
115
116         addLabelAndComponent(menuPanel, "Es Teh (Rp 5000):", form.spTeh, 250, 22,
117             100, 60, fieldHeight);
118
119         addLabelAndComponent(menuPanel, "Es Kopi (Rp 6000):", form.spKopi, 465, 22,
120             110, 60, fieldHeight);
121
122         frame.add(menuPanel);
123         currentY += panelHeight + 5;
124     }
```

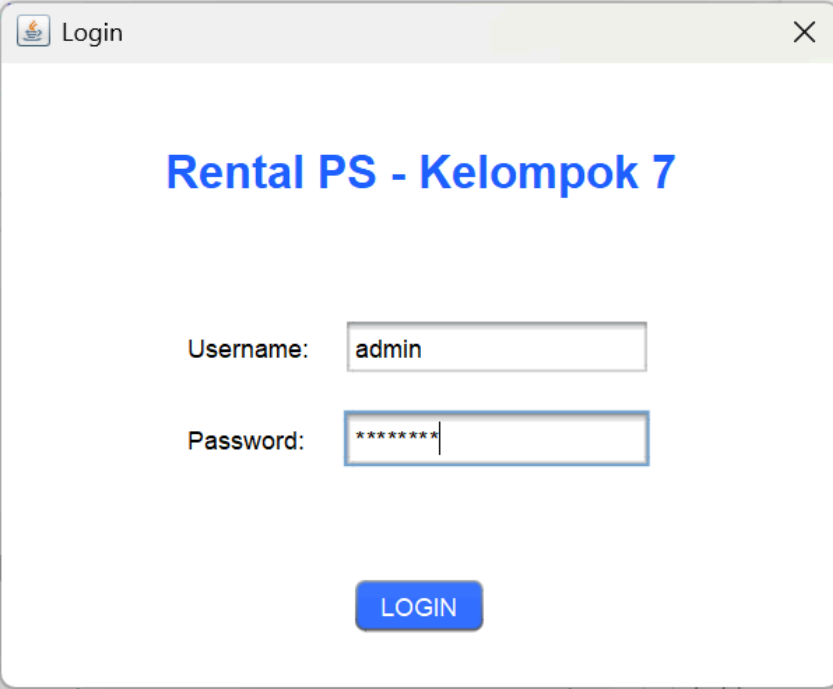
```

124
125 // === ROW 3: Discount and Action Buttons ===
126 JPanel actionPanel = new JPanel();
127 actionPanel.setBorder(BorderFactory.createTitledBorder("Diskon & Cetak"));
128 actionPanel.setLayout(null);
129 actionPanel.setBounds(leftMargin, currentY, 1050, panelHeight);
130
131 addLabelAndComponent(actionPanel, "Diskon:", form.cbDiskon, 15, 22,
132     60, 250, fieldHeight);
133
134 // Action buttons with proper spacing
135 JButton btnSimpan = new JButton("Simpan + Struk");
136 btnSimpan.setBounds(400, 22, 130, buttonHeight);
137 actionPanel.add(btnSimpan);
138
139 JButton btnPDF = new JButton("PDF");
140 btnPDF.setBounds(540, 22, 70, buttonHeight);
141 actionPanel.add(btnPDF);
142
143 JButton btnPrint = new JButton("Cetak");
144 btnPrint.setBounds(620, 22, 70, buttonHeight);
145 actionPanel.add(btnPrint);
146
147 frame.add(actionPanel);
148 currentY += panelHeight + 5;
149
150 // === Output Area (Reduced height) ===
151 JScrollPane areaScroll = new JScrollPane(form.areaOutput);
152 areaScroll.setBounds(leftMargin, currentY, 1050, 90); // Reduced from 120
153 frame.add(areaScroll);
154 currentY += 95; // Reduced gap
155
156 // === Data Table (Reduced height) ===
157 JPanel tablePanel = new JPanel();
158 tablePanel.setBorder(BorderFactory.createTitledBorder("Data Transaksi"));
159 tablePanel.setLayout(new BorderLayout());
160
161 JScrollPane tableScroll = new JScrollPane(tableComp.table);
162 tablePanel.add(tableScroll, BorderLayout.CENTER);
163 tablePanel.setBounds(leftMargin, currentY, 1050, 140); // Reduced from 180
164 frame.add(tablePanel);
165 currentY += 145; // Reduced gap
166
167 // === Search and Management Panel ===
168 JPanel managementPanel = new JPanel();
169 managementPanel.setBorder(BorderFactory.createTitledBorder("Manajemen Data"));
170 managementPanel.setLayout(null);
171 managementPanel.setBounds(leftMargin, currentY, 1050, panelHeight);
172
173 addLabelAndComponent(managementPanel, "Cari (Nama/PS/ID):", form.tfCari, 15, 22,
174     130, 200, fieldHeight);
175
176 JButton btnCari = new JButton("Cari");
177 btnCari.setBounds(360, 22, 80, buttonHeight);
178 managementPanel.add(btnCari);
179
180 JButton btnUbah = new JButton("Ubah");
181 btnUbah.setBounds(450, 22, 80, buttonHeight);
182 managementPanel.add(btnUbah);
183
184 JButton btnHapus = new JButton("Hapus");
185 btnHapus.setBounds(540, 22, 80, buttonHeight);
186 managementPanel.add(btnHapus);
187
188 frame.add(managementPanel);
189 currentY += panelHeight + 10;
190
191 // === Exit Button ===
192 JButton btnKeluar = new JButton("Keluar");
193 int btnWidth = 150;
194 int frameWidth = 1100;
195 int posX = (frameWidth / 2) - (btnWidth / 2);
196 btnKeluar.setBounds(posX, currentY, btnWidth, buttonHeight);
197 frame.add(btnKeluar);
198
199 // Add hidden field
200 frame.add(form.tfIdHidden);
201
202 // Set optimized frame size
203 frame.setMinimumSize(new Dimension(1100, currentY + 50));
204 frame.setPreferredSize(new Dimension(1100, currentY + 50));
205 }
206
207 private static void addLabelAndComponent(Container parent, String labelText, JComponent component,
208     int x, int y, int labelWidth, int componentWidth, int height) {
209     JLabel label = new JLabel(labelText);
210     label.setBounds(x, y, labelWidth, height);
211     parent.add(label);
212
213     component.setBounds(x + labelWidth + 10, y, componentWidth, height);
214     parent.add(component);
215 }
216
217 // Overloaded method for backward compatibility
218 private static void addLabelAndComponent(JFrame frame, String labelText, JComponent component,
219     int x, int y, int labelWidth, int componentWidth) {
220     addLabelAndComponent(frame, labelText, component, x, y, labelWidth, componentWidth, 25);
221 }
222
223 public static void applyTheme(JFrame frame) {
224     try {
225         UIManager.setLookAndFeel(new com.formdev.flatlaf.FlatLightLaf());
226         SwingUtilities.updateComponentTreeUI(frame);
227     } catch (Exception ex) {
228         System.err.println("Gagal mengatur tema FlatLaf");
229     }
230 }
231 }

```

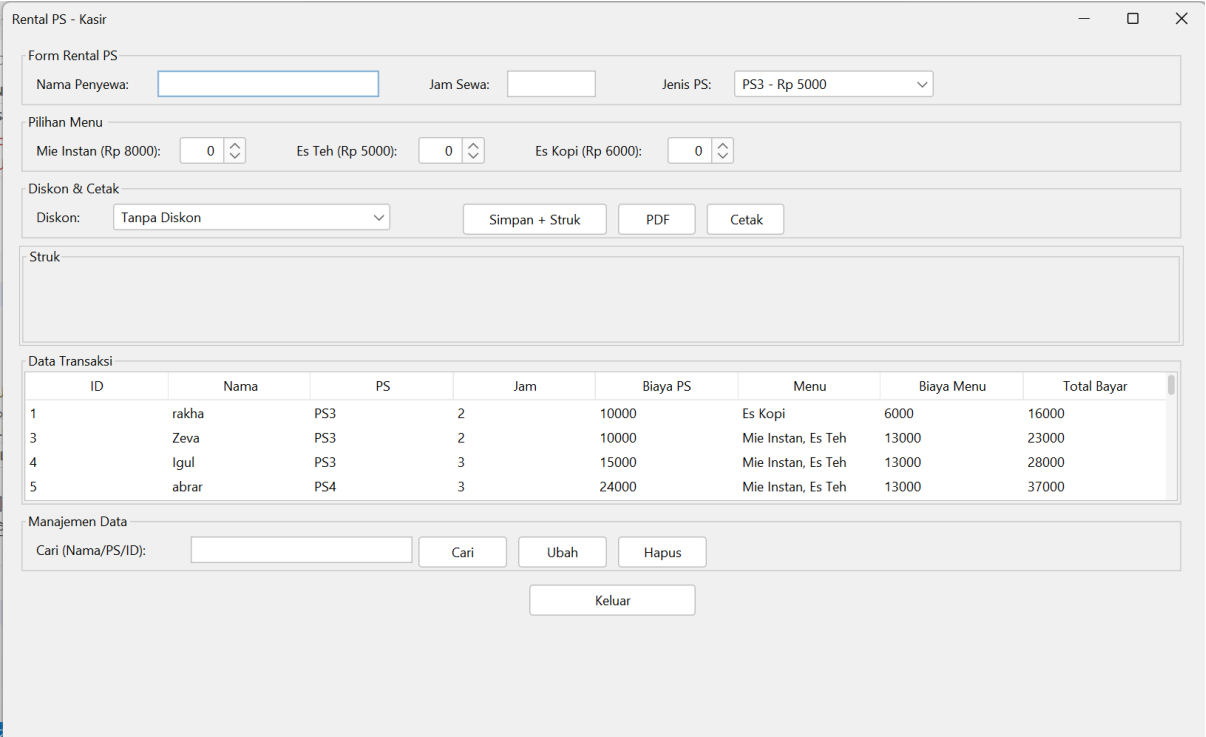
Output Kode Program

- Halaman Login



The screenshot shows a login window titled "Login" with a close button (X) in the top right corner. The window has a light gray border and a white background. In the center, the text "Rental PS - Kelompok 7" is displayed in a large, bold, blue font. Below this, there are two input fields: "Username:" with the text "admin" and "Password:" with a masked password "*****". A blue button labeled "LOGIN" is positioned below the password field.

- Halaman Sistem Rental PS



The screenshot shows a window titled "Rental PS - Kasir" with standard window controls (minimize, maximize, close) in the top right corner. The window is divided into several sections:

- Form Rental PS:** Contains input fields for "Nama Penyewa:" (empty), "Jam Sewa:" (empty), and "Jenis PS:" (dropdown menu showing "PS3 - Rp 5000").
- Pilihan Menu:** Contains three input fields with spinners: "Mie Instan (Rp 8000):" (0), "Es Teh (Rp 5000):" (0), and "Es Kopi (Rp 6000):" (0).
- Diskon & Cetak:** Contains a "Diskon:" dropdown menu (showing "Tanpa Diskon"), a "Simpan + Struk" button, a "PDF" button, and a "Cetak" button.
- Struk:** A large empty rectangular area for the receipt.
- Data Transaksi:** A table with 8 columns: ID, Nama, PS, Jam, Biaya PS, Menu, Biaya Menu, and Total Bayar. It contains 5 rows of data.
- Manajemen Data:** Contains a "Cari (Nama/PS/ID):" input field, "Cari", "Ubah", and "Hapus" buttons, and a "Keluar" button at the bottom.

ID	Nama	PS	Jam	Biaya PS	Menu	Biaya Menu	Total Bayar
1	rakha	PS3	2	10000	Es Kopi	6000	16000
3	Zeva	PS3	2	10000	Mie Instan, Es Teh	13000	23000
4	Igul	PS3	3	15000	Mie Instan, Es Teh	13000	28000
5	abrar	PS4	3	24000	Mie Instan, Es Teh	13000	37000

BAB III

KESIMPULAN

3.1 Kesimpulan

Sistem informasi rental PlayStation yang telah dirancang dan diimplementasikan terbukti mampu menjawab berbagai permasalahan dalam pengelolaan usaha rental yang sebelumnya dilakukan secara manual. Sistem ini menghadirkan fitur-fitur lengkap dan terintegrasi seperti manajemen data pelanggan, pencatatan transaksi penyewaan dan pengembalian, pemesanan makanan dan minuman, pencetakan struk otomatis, pemberian diskon yang fleksibel, ekspor data transaksi ke format PDF, serta penyimpanan dan pengelolaan riwayat transaksi secara digital. Seluruh fitur ini tidak hanya meningkatkan efisiensi dan akurasi proses bisnis, tetapi juga mendukung kegiatan operasional harian agar lebih profesional dan terdokumentasi dengan baik. Dari sisi teknis, sistem telah memanfaatkan prinsip-prinsip utama pemrograman berorientasi objek (Object-Oriented Programming) seperti enkapsulasi, pewarisan, polimorfisme, dan abstraksi yang diterapkan secara tepat guna menciptakan struktur kode yang bersih, modular, dan mudah dipelihara. Hal ini menjadikan sistem sebagai solusi yang tidak hanya bermanfaat dalam praktik langsung di lapangan, tetapi juga sebagai sarana pembelajaran yang aplikatif bagi mahasiswa dalam memahami implementasi pengembangan perangkat lunak secara nyata.

3.2 Saran

Meskipun sistem ini telah memiliki fitur yang cukup lengkap dan telah membantu dalam operasional rental PlayStation, masih terdapat ruang untuk peningkatan dan pengembangan lebih lanjut agar sistem menjadi lebih andal dan kompetitif. Disarankan untuk melakukan pengujian menyeluruh secara fungsional dan non-fungsional untuk memastikan sistem berjalan optimal dalam berbagai situasi, termasuk skenario dengan banyak pengguna atau transaksi dalam waktu bersamaan. Untuk meningkatkan pengalaman pengguna, tampilan antarmuka dapat didesain lebih menarik dan responsif, dengan mempertimbangkan aspek UI/UX yang modern. Dari sisi keamanan, sangat penting untuk menambahkan fitur enkripsi data pelanggan, sistem login yang aman, serta backup otomatis agar data tidak mudah hilang atau disalahgunakan. Selain itu, sistem juga dapat dikembangkan lebih lanjut dengan integrasi metode pembayaran digital seperti e-wallet atau QRIS agar pelanggan memiliki lebih banyak pilihan transaksi. Penambahan fitur notifikasi otomatis melalui email atau pesan singkat untuk mengingatkan pelanggan tentang waktu pengembalian atau promo khusus juga bisa menjadi nilai tambah yang meningkatkan loyalitas pelanggan. Terakhir, fitur analitik sederhana berbasis data transaksi dapat

ditambahkan untuk membantu pemilik usaha dalam mengambil keputusan bisnis berdasarkan data yang telah terakumulasi.