

SISTEM PARKIR BERBASIS GUI JAVA

Anggota Kelompok 1



Alifia Nur Huda
2310631250005



Arya Cakra Ramadhan
2310631250086



Yuuka Natasya Aji
2310631250079



Daffa Abyan Hakim
2310631250088

Latar Belakang

Perkembangan teknologi mendorong kebutuhan sistem otomatis, termasuk dalam pengelolaan parkir. Sistem parkir manual sering menimbulkan masalah seperti kesalahan pencatatan, antrean panjang, dan perhitungan tarif yang tidak efisien.

Kampus UNSIKA memiliki mobilitas kendaraan tinggi dan masih menggunakan sistem parkir manual. Solusinya adalah membangun sistem parkir desktop berbasis Java dan MySQL dengan fitur pencatatan kendaraan, perhitungan otomatis tarif, dan pencetakan tiket PDF.

Rumusan Masalah

- Bagaimana merancang sistem parkir desktop yang mencatat kendaraan masuk dan keluar secara otomatis dan akurat?
- Bagaimana sistem menghitung durasi dan biaya parkir secara otomatis?
- Bagaimana menghasilkan tiket dan struk parkir dalam bentuk PDF?
- Bagaimana sistem ini mempermudah administrasi parkir kampus?

Tujuan

1. Membangun sistem parkir desktop yang cepat dan akurat menggunakan Java & MySQL.
2. Mengembangkan fitur perhitungan otomatis durasi dan biaya parkir.
3. Menghasilkan tiket dan struk parkir dalam file PDF sebagai bukti transaksi sah.
4. Mempermudah tugas petugas parkir dan meningkatkan efisiensi sistem parkir di kampus.
5. Mendukung profesionalisme dan pemanfaatan teknologi informasi di lingkungan kampus.

Fitur- Fitur Aplikasi

Fitur - Fitur Aplikasi

1. Login
2. Menu Utama
3. Parkir Masuk
4. Parkir Keluar
5. Cetak Tiket Parkir (PDF)
6. Cetak Struk Parkir (PDF)
7. Koneksi Database

Login

Fitur login berfungsi sebagai gerbang awal akses ke dalam aplikasi sistem parkir. Melalui tampilan login, pengguna diminta untuk memasukkan username dan password yang telah ditentukan. Validasi dilakukan secara langsung, dan jika data yang dimasukkan sesuai, maka pengguna akan diarahkan menuju halaman Menu Utama. Sebaliknya, jika terdapat kesalahan dalam memasukkan username atau password, sistem akan memunculkan pop up bahwa data tidak valid. Saat ini, sistem login menggunakan data statis yaitu username “admin” dan password “1234”, namun dapat dikembangkan lebih lanjut agar terintegrasi dengan database untuk mendukung manajemen pengguna yang lebih kompleks.



```
1 package Parkir1;
2
3 import java.awt.*;
4 import javax.swing.*;
5 import javax.swing.border.EmptyBorder;
6
7 public class Login extends JFrame {
8     private JTextField userField;
9     private JPasswordField passField;
10
11    public Login() {
12        setTitle("Login Parkir UNSIKA");
13        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14        setExtendedState(JFrame.MAXIMIZED_BOTH);
15        setMinimumSize(new Dimension(800, 500));
16        setLocationRelativeTo(null); // Pusat saat pertama dibuka
17
18        // Root panel dengan BorderLayout
19        JPanel root = new JPanel(new BorderLayout());
20        root.setBackground(new Color(245, 245, 245));
21        root.setBorder(new EmptyBorder(50, 50, 50, 50));
22
23        // Panel judul
24        JLabel title = new JLabel("Parkir UNSIKA", SwingConstants.CENTER);
25        title.setFont(new Font("Segoe UI", Font.BOLD, 36));
26        title.setForeground(new Color(40, 40, 120));
27        root.add(title, BorderLayout.NORTH);
28
```

```
28
29     // Panel tengah dengan GridBagLayout
30     JPanel center = new JPanel(new GridBagLayout());
31     center.setBackground(Color.WHITE);
32     center.setBorder(BorderFactory.createCompoundBorder(
33         BorderFactory.createLineBorder(new Color(100,100,150), 2, true),
34         new EmptyBorder(30, 30, 30, 30)
35     ));
36
37     GridBagConstraints gbc = new GridBagConstraints();
38     gbc.insets = new Insets(15, 15, 15, 15);
39     gbc.fill = GridBagConstraints.HORIZONTAL;
40
41     // Label "Login"
42     JLabel loginLabel = new JLabel("LOGIN", SwingConstants.CENTER);
43     loginLabel.setFont(new Font("Segoe UI", Font.BOLD, 24));
44     loginLabel.setForeground(new Color(60, 60, 100));
45     gbc.gridx = 0;
46     gbc.gridy = 0;
47     gbc.gridwidth = 2;
48     center.add(loginLabel, gbc);
49
```

```
1  gbc.gridx = 1;
2  gbc.gridy = 1;
3  center.add(new JLabel("Username:"), gbc);
4  gbc.gridx = 1;
5  userField = new JTextField();
6  userField.setFont(new Font("Segoe UI", Font.PLAIN, 16));
7  center.add(userField, gbc);
8
9  // Password
10 gbc.gridx = 0;
11 gbc.gridy = 2;
12 center.add(new JLabel("Password:"), gbc);
13 gbc.gridx = 1;
14 passField = new JPasswordField();
15 passField.setFont(new Font("Segoe UI", Font.PLAIN, 16));
16 center.add(passField, gbc);
17
18 // Panel tombol
19 JPanel btnPanel = new JPanel(new GridLayout(1, 2, 20, 0));
20 btnPanel.setBackground(Color.WHITE);
21 JButton loginBtn = new JButton("Login");
22 loginBtn.setFont(new Font("Segoe UI", Font.BOLD, 16));
23 loginBtn.setBackground(new Color(40, 120, 200));
24 loginBtn.setForeground(Color.WHITE);
25 JButton cancelBtn = new JButton("Cancel");
26 cancelBtn.setFont(new Font("Segoe UI", Font.BOLD, 16));
27 cancelBtn.setBackground(new Color(200, 40, 40));
28 cancelBtn.setForeground(Color.WHITE);
29 btnPanel.add(loginBtn);
30 btnPanel.add(cancelBtn);
31
32 gbc.gridx = 0;
33 gbc.gridy = 3;
34 gbc.gridwidth = 2;
35 center.add(btnPanel, gbc);
36
37 root.add(center, BorderLayout.CENTER);
38 setContentPane(root);
39
40 // Aksi tombol
41 loginBtn.addActionListener(e -> handleLogin());
42 cancelBtn.addActionListener(e -> { userField.setText(""); passField.setText(""); });
43 }
```

```
45     private void handleLogin() {
46         String user = userField.getText().trim();
47         String pass = new String(passField.getPassword());
48         if (user.equals("admin") && pass.equals("1234")) {
49             JOptionPane.showMessageDialog(this, "Login berhasil!", "Sukses", JOptionPane.INFORMATION_MESSAGE);
50             dispose();
51             new MenuUtamaParkir().setVisible(true);
52         } else {
53             JOptionPane.showMessageDialog(this, "Username atau Password salah.", "Error", JOptionPane.ERROR_MESSAGE);
54             userField.setText("");
55             passField.setText("");
56         }
57     }
58
59     public static void main(String[] args) {
60         SwingUtilities.invokeLater(() -> new Login().setVisible(true));
61     }
62 }
63 }
```

Menu Utama Parkir

Setelah berhasil melakukan login, pengguna diarahkan ke tampilan Menu Utama. Pada halaman ini, disediakan tiga tombol utama yaitu “Parkir Masuk”, “Parkir Keluar”, dan “Logout”. Tombol “Parkir Masuk” digunakan untuk mencatat kendaraan yang masuk ke area parkir, sedangkan “Parkir Keluar” digunakan untuk mencatat data kendaraan yang keluar sekaligus menghitung biaya parkir. Tombol “Logout” digunakan untuk keluar dari sistem dan kembali ke halaman login. Tampilan Menu Utama dirancang secara sederhana dan responsif agar mudah digunakan oleh petugas parkir.



```
1 package Parkir1;
2
3 import java.awt.*;
4 import javax.swing.*;
5
6 public class MenuUtamaParkir extends JFrame {
7
8     public MenuUtamaParkir() {
9         setTitle("Menu Utama Parkir");
10        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        setSize(500, 400); // Ukuran awal
12        setMinimumSize(new Dimension(400, 300));
13        setLocationRelativeTo(null); // Pusat saat pertama dibuka
14        setResizable(true);
15
16        // Panel utama dengan layout GridBag untuk fleksibilitas
17        JPanel panelUtama = new JPanel(new GridBagLayout());
18        panelUtama.setBackground(Color.WHITE); // Latar belakang putih untuk tampilan bersih
19        panelUtama.setBorder(BorderFactory.createEmptyBorder(30, 50, 30, 50));
20
21        GridBagConstraints gbc = new GridBagConstraints();
22        gbc.gridx = 0;
23        gbc.fill = GridBagConstraints.HORIZONTAL;
24        gbc.insets = new Insets(10, 0, 10, 0); // Spasi antar komponen
25
26        // Label Judul
27        JLabel labelJudul = new JLabel("Parkir UNSIKA", SwingConstants.CENTER);
28        labelJudul.setFont(new Font("Segoe UI", Font.BOLD, 30));
29        labelJudul.setForeground(new Color(40, 40, 100)); // Biru tua
30        gbc.gridy = 0;
31        panelUtama.add(labelJudul, gbc);
32
33        // Tombol-tombol
34        JButton btnMasuk = new JButton("Parkir Masuk");
35        JButton btnKeluar = new JButton("Parkir Keluar");
36        JButton btnLogout = new JButton("Logout");
37
```

```
37
38     Dimension tombolUkuran = new Dimension(250, 45);
39     for (JButton btn : new JButton[]{btnMasuk, btnKeluar, btnLogout}) {
40         btn.setPreferredSize(tombolUkuran);
41         btn.setFont(new Font("Segoe UI", Font.PLAIN, 16));
42         gbc.gridx++;
43         panelUtama.add(btn, gbc);
44     }
45
46     // Tambahkan ke frame
47     setContentPane(panelUtama);
48
49     // Aksi tombol
50     btnMasuk.addActionListener(e -> {
51         new ParkirMasuk().setVisible(true);
52         dispose();
53     });
54
55     btnKeluar.addActionListener(e -> {
56         new ParkirKeluar().setVisible(true);
57         dispose();
58     });
59
60     btnLogout.addActionListener(e -> {
61         int confirm = JOptionPane.showConfirmDialog(this, "Yakin ingin logout?", "Konfirmasi Logout", JOptionPane.YES_NO_OPTION);
62         if (confirm == JOptionPane.YES_OPTION) {
63             dispose();
64             new Login().setVisible(true);
65         }
66     });
67 }
68
69 public static void main(String[] args) {
70     SwingUtilities.invokeLater(() -> new MenuUtamaParkir().setVisible(true));
71 }
72 }
73 }
```

Parkir Masuk

Fitur Parkir Masuk berfungsi untuk mencatat kendaraan yang baru memasuki area parkir. Petugas menginput data seperti nomor tiket, nomor polisi, dan jenis kendaraan, sementara tanggal serta jam masuk tercatat otomatis oleh sistem. Data yang telah dimasukkan akan tersimpan di database MySQL dan secara otomatis sistem mencetak tiket parkir dalam format PDF A6 berisi informasi kendaraan dan waktu masuk sebagai bukti parkir.

Tampilan fitur ini juga dilengkapi tombol Edit dan Hapus. Tombol Edit memungkinkan petugas memperbaiki data kendaraan yang sudah tercatat, dengan pop-up konfirmasi sebelum perubahan disimpan dan notifikasi setelah data berhasil diubah. Tombol Hapus digunakan untuk menghapus data kendaraan tertentu, juga dilengkapi pop-up konfirmasi dan notifikasi keberhasilan. Fitur ini memberikan fleksibilitas dan akurasi dalam pengelolaan data kendaraan yang masuk, sehingga operasional parkir menjadi lebih efisien dan terkontrol.

```
1 package Parkir1;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import java.sql.*;
6 import java.time.*;
7 import java.time.format.DateTimeFormatter;
8 import javax.swing.*;
9 import javax.swing.border.TitledBorder;
10 import javax.swing.table.*;
11
12 public class ParkirKeluar extends JFrame {
13     private JTextField tfNomorTiket, tfNomorPolisi, tfJenisKendaraan, tfTanggalMasuk, tfJamMasuk;
14     private JTextField tfTanggalKeluar, tfDurasi, tfTotal, tfPembayaran;
15     private JTable table;
16     private DefaultTableModel tableModel;
17     private Timer timer;
18     private LocalDateTime waktuMasuk;
19     private long totalBiaya = 0;
20     private JButton btnBayar;
21
22
23     public ParkirKeluar() {
24         setTitle("Parkir Keluar");
25         setExtendedState(JFrame.MAXIMIZED_BOTH);
26         setMinimumSize(new Dimension(850, 500));
27         setLocationRelativeTo(null);
28         setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
29
30         JPanel panelUtama = new JPanel(new BorderLayout(20, 20));
31         panelUtama.setBackground(Color.decode("#f9f9f9"));
32         panelUtama.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
33 }
```

```
33
34     JPanel panelForm = new JPanel(new GridBagLayout());
35     panelForm.setPreferredSize(new Dimension(400, 600));
36     panelForm.setBorder(BorderFactory.createTitledBorder(
37         BorderFactory.createLineBorder(new Color(100, 100, 150), 1, true),
38         " DATA KENDARAAN KELUAR ", 
39         TitledBorder.CENTER, TitledBorder.TOP,
40         new Font("Segoe UI", Font.BOLD, 15), new Color(40, 40, 120)
41     ));
42     panelForm.setBackground(Color.WHITE);
43     GridBagConstraints gbc = new GridBagConstraints();
44     gbc.insets = new Insets(10, 10, 10, 10);
45     gbc.fill = GridBagConstraints.HORIZONTAL;
46     gbc.weightx = 1.0;
47
48     gbc.gridx = 0; gbc.gridy = 0;
49     panelForm.add(new JLabel("Nomor Tiket:"), gbc);
50     tfNomorTiket = new JTextField(10);
51     gbc.gridx = 1;
52     panelForm.add(tfNomorTiket, gbc);
53     JButton btnCek = new JButton("Cek");
54     gbc.gridx = 2;
55     panelForm.add(btnCek, gbc);
56
57     gbc.gridx = 0; gbc.gridy++;
58     panelForm.add(new JLabel("Nomor Polisi:"), gbc);
59     tfNomorPolisi = new JTextField(10); tfNomorPolisi.setEditable(false);
60     gbc.gridx = 1; gbc.gridwidth = 2;
```

```
1 panelForm.add(tfNomorPolisi, gbc); gbc.gridx = 1;
2
3 gbc.gridx = 0; gbc.gridy++;
4 panelForm.add(new JLabel("Jenis Kendaraan:"), gbc);
5 tfJenisKendaraan = new JTextField(10); tfJenisKendaraan.setEditable(false);
6 gbc.gridx = 1; gbc.gridwidth = 2;
7 panelForm.add(tfJenisKendaraan, gbc); gbc.gridwidth = 1;      36
8
9 gbc.gridx = 0; gbc.gridy++;
10 panelForm.add(new JLabel("Tanggal Masuk:"), gbc);           37      // Baris Total Biaya
11 tfTanggalMasuk = new JTextField(10); tfTanggalMasuk.setEditable(false);
12 gbc.gridx = 1; gbc.gridwidth = 2;                           38
13 panelForm.add(tfTanggalMasuk, gbc); gbc.gridwidth = 1;       39
14
15 gbc.gridx = 0; gbc.gridy++;
16 panelForm.add(new JLabel("Jam Masuk:"), gbc);             40
17 tfJamMasuk = new JTextField(10); tfJamMasuk.setEditable(false);
18 gbc.gridx = 1; gbc.gridwidth = 2;                           41      // TextField untuk Total Biaya
19 panelForm.add(tfJamMasuk, gbc); gbc.gridwidth = 1;         42
20
21 gbc.gridx = 0; gbc.gridy++;
22 panelForm.add(new JLabel("Tanggal Keluar:"), gbc);        43
23 tfTanggalKeluar = new JTextField(10); tfTanggalKeluar.setEditable(false);
24 tfTanggalKeluar.setText(LocalDate.now().format(DateTimeFormatter.ISO_DATE)); 44
25 gbc.gridx = 1; gbc.gridwidth = 2;                           45
26 panelForm.add(tfTanggalKeluar, gbc); gbc.gridwidth = 1;     46
27
28 gbc.gridx = 0; gbc.gridy++;
29 panelForm.add(new JLabel("Durasi:"), gbc);                 47
30 tfDurasi = new JTextField(10); tfDurasi.setEditable(false);
31 gbc.gridx = 1;
32 panelForm.add(tfDurasi, gbc);
33 JButton btnOk = new JButton("OK");
34 gbc.gridx = 2;
35 panelForm.add(btnOk, gbc);                                48 // Tombol Bayar di sebelah tfTotal (gridx = 2)
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
```

```
36
37     // Baris Total Biaya
38 gbc.gridx = 0;
39 gbc.gridy++;
40 panelForm.add(new JLabel("Total Biaya:"), gbc);
41
42 // TextField untuk Total Biaya
43 tfTotal = new JTextField(10);
44 tfTotal.setEditable(false);
45 gbc.gridx = 1;
46 panelForm.add(tfTotal, gbc);
47
48 // Tombol Bayar di sebelah tfTotal (gridx = 2)
49 btnBayar = new JButton("Bayar");
50 gbc.gridx = 2;
51 panelForm.add(btnBayar, gbc);
52
53 // Action untuk tombol Bayar
54 btnBayar.addActionListener(e -> {
55     String input = JOptionPane.showInputDialog(this, "Silahkan masukkan pembayaran:");
56     if (input != null && !input.trim().isEmpty()) {
57         tfPembayaran.setText(input.trim());
58     }
59 });
60
61
```

```
1  gbc.gridx = 0; gbc.gridy++;
2      tfPembayaran = new JTextField(10);
3      tfPembayaran.setVisible(false);
4      gbc.gridx = 1; gbc.gridwidth = 2;
5      panelForm.add(tfPembayaran, gbc); gbc.gridwidth = 1;
6      gbc.gridx = 0; gbc.gridy++;
7      gbc.gridwidth = 3;
8      JButton btnCetakStruk = new JButton("Cetak Struk");
9      panelForm.add(btnCetakStruk, gbc);
10
11     gbc.gridy++;
12     JButton btnKembali = new JButton("Kembali Ke Menu Utama");
13     btnKembali.setBackground(new Color(220, 220, 220));
14     btnKembali.setFont(new Font("Segoe UI", Font.PLAIN, 14));
15     panelForm.add(btnKembali, gbc);
16     gbc.gridwidth = 1;
17
18     String[] kolom = {"Nomor Tiket", "Nomor Polisi", "Jenis Kendaraan", "Tanggal Masuk", "Jam Masuk", "Tanggal Keluar", "Durasi"};
19     tableModel = new DefaultTableModel(kolom, 0);
20     table = new JTable(tableModel);
21     table.setFont(new Font("Segoe UI", Font.PLAIN, 13));
22     table.setRowHeight(28);
23     JTableHeader header = table.getTableHeader();
24     header.setFont(new Font("Segoe UI", Font.BOLD, 14));
25     ((DefaultTableCellRenderer) header.getDefaultRenderer()).setHorizontalAlignment(JLabel.CENTER);
26
27     DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();
28     centerRenderer.setHorizontalAlignment(JLabel.CENTER);
29     for (int i = 0; i < table.getColumnCount(); i++) {
30         table.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
31     }
```

```
32
33     table.setSelectionBackground(new Color(210, 210, 255));
34     JScrollPane scrollPane = new JScrollPane(table);
35     scrollPane.setPreferredSize(new Dimension(600, 400));
36
37     loadTableData();
38
39
40     btnCek.addActionListener(e -> {
41         String tiket = tfNomorTiket.getText().trim();
42         try (Connection conn = DatabaseConnection.getConnection()) {
43             String query = "SELECT * FROM parkir WHERE nomor_tiket = ?";
44             PreparedStatement stmt = conn.prepareStatement(query);
45             stmt.setString(1, tiket);
46             ResultSet rs = stmt.executeQuery();
47             if (rs.next()) {
48                 tfNomorPolisi.setText(rs.getString("nomor_polisi"));
49                 tfJenisKendaraan.setText(rs.getString("jenis_kendaraan"));
50                 tfTanggalMasuk.setText(rs.getDate("tanggal_masuk").toString());
51                 tfJamMasuk.setText(rs.getTime("jam_masuk").toString());
52                 waktuMasuk = LocalDateTime.of(rs.getDate("tanggal_masuk").toLocalDate(), rs.getTime("jam_masuk").toLocalTime());
53
54                 if (timer != null) timer.stop();
55             }
56         }
57     });
58 }
```

```
1  tfDurasi.setText(jam + " jam " + menit + " menit");
2      });
3      timer.start();
4  } else {
5      JOptionPane.showMessageDialog(this, "Nomor Tiket tidak ditemukan.");
6  }
7 } catch (Exception ex) {
8     JOptionPane.showMessageDialog(this, "Gagal mengambil data dari database.");
9 }
10 );
11
12
13
14 btnOk.addActionListener(e -> {
15 if (timer != null) timer.stop();
16 try {
17     LocalDateTime keluar = LocalDateTime.now();
18     Duration durasi = Duration.between(waktuMasuk, keluar);
19     long totalMenit = durasi.toMinutes();
20
21 // Pembulatan ke atas dan minimal 1 jam
22 long jamDibulatkan = (long) Math.ceil(totalMenit / 60.0);
23 if (jamDibulatkan == 0) jamDibulatkan = 1;
24
25 // Hitung tarif berdasarkan jenis kendaraan
26 long tarif = tfJenisKendaraan.getText().equalsIgnoreCase("Mobil") ? 5000 : 3000;
27 totalBiaya = jamDibulatkan * tarif;
28 tfDurasi.setText(jamDibulatkan + " jam");
29 tfTotal.setText("Rp " + totalBiaya);
30
31 try (Connection conn = DatabaseConnection.getConnection()) {
32     String update = "UPDATE parkir SET tanggal_keluar=? , durasi=? , total=? WHERE nomor_tiket=?";
33     PreparedStatement stmt = conn.prepareStatement(update);
34     stmt.setDate(1, java.sql.Date.valueOf(keluar.toLocalDate()));
35     stmt.setString(2, tfDurasi.getText());
36     stmt.setLong(3, totalBiaya);
37     stmt.setString(4, tfNomorTiket.getText());
38     stmt.executeUpdate();
39     JOptionPane.showMessageDialog(this, "Data parkir keluar berhasil disimpan.");
40     loadTableData();
41 }
```

```
42 } catch (Exception ex) {
43     JOptionPane.showMessageDialog(this, "Gagal menghitung durasi atau menyimpan data.");
44 }
45 );
46
47 btnCetakStruk.addActionListener(e -> {
48 if (tfPembayaran.getText().isEmpty()) {
49     JOptionPane.showMessageDialog(this, "Masukkan jumlah pembayaran!");
50     return;
51 }
52
53 try {
54     long pembayaran = Long.parseLong(tfPembayaran.getText());
55     if (pembayaran < totalBiaya) {
56         JOptionPane.showMessageDialog(this, "Pembayaran kurang!");
57         return;
58     }
59     long kembalian = pembayaran - totalBiaya;
```

```
1 CetakStruk.cetakStruk(
2             tfNomorTiket.getText(),
3             tfNomorPolisi.getText(),
4             tfJenisKendaraan.getText(),
5             tfTanggalMasuk.getText(),
6             tfJamMasuk.getText(),
7             tfTanggalKeluar.getText(),
8             tfDurasi.getText(),
9             tfTotal.getText(),
10            String.format("Rp %d", pembayaran),
11            String.format("Rp %d", kembalian)
12        );
13    } catch (NumberFormatException ex) {
14        JOptionPane.showMessageDialog(this, "Masukkan pembayaran dengan angka yang valid!");
15    }
16 });
17
18 btnKembali.addActionListener(e -> {
19     dispose();
20     new MenuUtamaParkir().setVisible(true);
21 });
22
23 panelUtama.add(panelForm, BorderLayout.WEST);
24 panelUtama.add(scrollPane, BorderLayout.CENTER);
25 setContentPane(panelUtama);
26 }
27
28 private void loadTableData() {
29     try (Connection conn = DatabaseConnection.getConnection()) {
30         tableModel.setRowCount(0);
31         Statement stmt = conn.createStatement();
32         ResultSet rs = stmt.executeQuery("SELECT * FROM parkir");
33         while (rs.next()) {
34             Object[] data = {
35                 rs.getString("nomor_tiket"),
36                 rs.getString("nomor_polisi"),
37                 rs.getString("jenis_kendaraan"),
38                 rs.getString("tanggal_masuk"),
39                 rs.getString("jam_masuk"),
40                 rs.getString("tanggal_keluar") != null ? rs.getString("tanggal_keluar") : "",
41                 rs.getString("durasi") != null ? rs.getString("durasi") : ""
42             };
43             tableModel.addRow(data);
44         }
45     } catch (Exception e) {
46         JOptionPane.showMessageDialog(this, "Gagal memuat data parkir keluar dari database.");
47     }
48 }
49
50 public static void main(String[] args) {
51     SwingUtilities.invokeLater(() -> new ParkirKeluar().setVisible(true));
52 }
53 }
```

Parkir Keluar

Fitur parkir keluar digunakan untuk mengelola data kendaraan yang keluar dari area parkir. Petugas akan memasukkan nomor tiket yang digunakan saat kendaraan masuk. Setelah itu, sistem akan mengambil data yang sesuai dari database dan menampilkan informasi kendaraan beserta waktu masuk. Selanjutnya, sistem akan mencatat waktu keluar secara otomatis, menghitung durasi parkir berdasarkan selisih waktu masuk dan keluar, serta menghitung total biaya yang harus dibayarkan. Setelah pembayaran dilakukan, sistem juga menghitung jumlah kembalian dan mencatat seluruh transaksi untuk dicetak dalam bentuk struk PDF

Cetak Tiket

Setiap kali kendaraan masuk ke area parkir, sistem akan secara otomatis mencetak tiket parkir sebagai bukti transaksi. Proses ini dilakukan oleh kelas CetakPdf.java yang menggunakan library iText untuk menghasilkan file PDF berukuran A6. Di dalam tiket tersebut tercantum informasi penting seperti nomor tiket, nomor polisi kendaraan, jenis kendaraan, tanggal, dan jam masuk. Tiket ini menjadi acuan ketika kendaraan akan keluar dan sebagai identitas kendaraan selama berada di area parkir.



```
1 package Parkir1;
2
3 import com.itextpdf.text.*;
4 import com.itextpdf.text.pdf.*;
5
6 import java.io.FileOutputStream;
7 import java.time.LocalDate;
8 import java.time.LocalDateTime;
9 import java.time.format.DateTimeFormatter;
10
11 public class CetakTiket {
12     public static void cetakTiket(String nomorTiket, String nomorPolisi, String jenisKendaraan, String tanggal, String jam) {
13         Document document = new Document(PageSize.A6);
14         try {
15             String fileName = "Tiket_" + nomorTiket + ".pdf";
16             PdfWriter.getInstance(document, new FileOutputStream(fileName));
17             document.open();
18
19             Font fontHeader = new Font(Font.FontFamily.COURIER, 12, Font.BOLD);
20             Font fontText = new Font(Font.FontFamily.COURIER, 10, Font.NORMAL);
21
22             Paragraph title = new Paragraph("TIKET PARKIR", fontHeader);
23             title.setAlignment(Element.ALIGN_CENTER);
24             document.add(title);
25
26             Paragraph subTitle = new Paragraph("UNIVERSITAS SINGAPERBANGSA KARAWANG", fontHeader);
27             subTitle.setAlignment(Element.ALIGN_CENTER);
28             document.add(subTitle);
29
30             document.add(new Paragraph("=====", fontText));
31             document.add(new Paragraph("No : " + nomorTiket, fontText));
32             document.add(new Paragraph("Plat : " + nomorPolisi, fontText));
33             document.add(new Paragraph("Jenis : " + jenisKendaraan, fontText));
34             document.add(new Paragraph("Tanggal : " + tanggal, fontText));
35             document.add(new Paragraph("Jam : " + jam, fontText));
36             document.add(new Paragraph("=====", fontText));
37
38             Paragraph thankYou = new Paragraph("Terima Kasih", fontText);
39             thankYou.setAlignment(Element.ALIGN_CENTER);
40             document.add(thankYou);
41
42         } catch (Exception e) {
43             e.printStackTrace();
44         } finally {
45             document.close();
46         }
47     }
48 }
```

Cetak Struk

Setelah kendaraan keluar dan proses pembayaran selesai dilakukan, sistem akan mencetak struk sebagai bukti pembayaran parkir. Fitur ini diatur oleh kelas CetakStruk.java, juga menggunakan library iText untuk mencetak file PDF. Struk yang dicetak mencakup informasi lengkap seperti nomor tiket, nomor kendaraan, jenis kendaraan, tanggal dan jam masuk, tanggal keluar, durasi parkir, total biaya, nominal pembayaran, dan kembalian. Struk ini dapat digunakan sebagai arsip atau bukti resmi bagi pengguna layanan parkir.

```
● ● ●  
1 package Parkir1;  
2  
3 import com.itextpdf.text.*;  
4 import com.itextpdf.text.pdf.*;  
5  
6 import java.io.FileOutputStream;  
7  
8 public class CetakStruk {  
9     public static void cetakStruk(String nomorTiket, String nomorPolisi, String jenisKendaraan,  
10                                String tanggalMasuk, String jamMasuk, String tanggalKeluar,  
11                                String durasi, String total, String pembayaran, String kembalian) {  
12         Document document = new Document(PageSize.A6);  
13         try {  
14             String fileName = "Struk_" + nomorTiket + ".pdf";  
15             PdfWriter.getInstance(document, new FileOutputStream(fileName));  
16             document.open();  
17  
18             Font fontHeader = new Font(Font.FontFamily.COURIER, 12, Font.BOLD);  
19             Font fontText = new Font(Font.FontFamily.COURIER, 10, Font.NORMAL);  
20  
21             Paragraph title = new Paragraph("STRUK PARKIR", fontHeader);  
22             title.setAlignment(Element.ALIGN_CENTER);  
23             document.add(title);  
24  
25             Paragraph subTitle = new Paragraph("UNIVERSITAS SINGAPERBANGSA KARAWANG", fontHeader);  
26             subTitle.setAlignment(Element.ALIGN_CENTER);  
27             document.add(subTitle);  
28  
29             document.add(new Paragraph("=====", fontText));  
30             document.add(new Paragraph("No : " + nomorTiket, fontText));  
31             document.add(new Paragraph("Plat : " + nomorPolisi, fontText));  
32             document.add(new Paragraph("Jenis : " + jenisKendaraan, fontText));  
33             document.add(new Paragraph("Masuk : " + tanggalMasuk + " " + jamMasuk, fontText));  
34             document.add(new Paragraph("Keluar : " + tanggalKeluar, fontText));  
35             document.add(new Paragraph("Durasi : " + durasi, fontText));  
36             document.add(new Paragraph("Total : " + total, fontText));  
37             document.add(new Paragraph("Pembayaran : " + pembayaran, fontText));  
38             document.add(new Paragraph("Kembalian : " + kembalian, fontText));  
39             document.add(new Paragraph("=====", fontText));  
40  
41             Paragraph thankYou = new Paragraph("Terima Kasih", fontText);  
42             thankYou.setAlignment(Element.ALIGN_CENTER);  
43             document.add(thankYou);  
44  
45         } catch (Exception e) {  
46             e.printStackTrace();  
47         } finally {  
48             document.close();  
49         }  
50     }  
51 }
```

Database Connection

Aplikasi ini terhubung dengan database MySQL bernama parkir_db untuk menyimpan dan mengelola data parkir secara terstruktur. Koneksi ke database dikelola oleh kelas DatabaseConnection.java, yang menyederhanakan proses pengambilan dan penyimpanan data antara aplikasi dan server database. Dengan sistem ini, semua data kendaraan yang masuk dan keluar akan tercatat dengan baik, sehingga memudahkan proses rekapitulasi maupun pencarian data secara real-time.



```
1 package Parkir1;  
2  
3 import java.sql.Connection;  
4 import java.sql.DriverManager;  
5 import java.sql.SQLException;  
6  
7 public class DatabaseConnection {  
8     private static final String URL = "jdbc:mysql://localhost:3306/parkir_db";  
9     private static final String USER = "root";  
10    private static final String PASSWORD = "";  
11  
12    public static Connection getConnection() throws SQLException {  
13        return DriverManager.getConnection(URL, USER, PASSWORD);  
14    }  
15 }
```

Terimakasih
