LAPORAN PRAKTIKUM
SonarQube

Disusun Oleh:
Dzakir Tsabit Asy Syafiq (241511071)
Jurusan Teknik Komputer dan Informatika

Program Studi D-3 Teknik Informatika
Politeknik Negeri Bandung
09/11/2025

Code sebelum nya (KasirUI.java):

```
package code;


import java.awt.*;

import java.text.NumberFormat;

import java.util.Locale;

import javax.swing.*;

import javax.swing.table.DefaultTableModel;


public class KasirUI extends JFrame {


    // Komponen GUI

    private JTable tabelProduk;

    private JTable tabelKeranjang;

    private JTextField qtyField;

    private JLabel labelTotal;

    private JLabel labelPoints;

    private JTextArea areaStruk;

    private DefaultTableModel modelProduk;

    private DefaultTableModel modelKeranjang;


    // Buttons - simpan sebagai instance variables

    private JButton btnAdd;

    private JButton btnHapus;

    private JButton btnCheckout;

    private JButton btnCetak;


    // Format mata uang

    private final NumberFormat currencyFormat = 
NumberFormat.getCurrencyInstance(new Locale("id", "ID"));


    // Data untuk struk terakhir

    private String strukTerakhir = "";
```

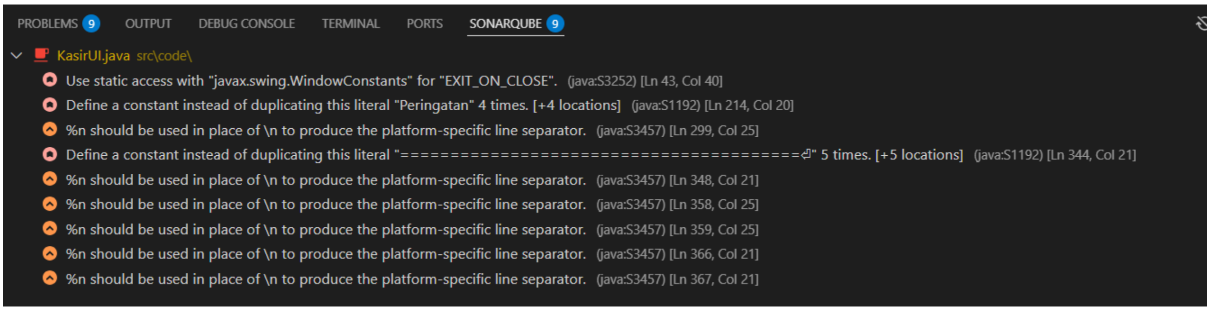
```
public KasirUI() {
    initializeFrame();
    initializeComponents();
    setupLayout();
    setupEventListeners();
    setVisible(true);
}

private void initializeFrame() {
    setTitle("GUI Dzakir Tsabit - 241511071");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(900, 600);
    setLocationRelativeTo(null);
    setLayout(new BorderLayout());
}

private void initializeComponents() {
    // Inisialisasi tabel produk
    String[][] dataProduk = {
            { "P001", "Air Mineral 600ml", "3000" },
            { "P002", "Kopi Sachet", "5000" },
            { "P003", "Roti isi", "8000" },
            { "P004", "Snack Keripik", "6000" },
            { "P005", "Minuman Botol", "12000" }
    };
    String[] kolomProduk = { "ID", "Nama Produk", "Harga (Rp)" };
    modelProduk = new DefaultTableModel(dataProduk, kolomProduk) {
        @Override
        public boolean isCellEditable(int row, int column) {
            return false; // Tidak bisa diedit manual
        }
    };
    tabelProduk = new JTable(modelProduk);
    tabelProduk.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
```

```
        // Inisialisasi tabel keranjang
        String[] kolomKeranjang = { "ID", "Nama Produk", "Qty", "Harga",
"Subtotal" };
        modelKeranjang = new DefaultTableModel(kolomKeranjang, 0) {
            @Override
            public boolean isCellEditable(int row, int column) {
                return false; // Tidak bisa diedit manual
            }
        };
        tabelKeranjang = new JTable(modelKeranjang);
        tabelKeranjang.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

        // Inisialisasi komponen lainnya
        qtyField = new JTextField("1", 5);
        labelTotal = new JLabel("Total: Rp 0");
        labelTotal.setFont(new Font("Segoe UI", Font.BOLD, 14));
        labelPoints = new JLabel("Points: 0");
        areaStruk = new JTextArea(8, 20);
        areaStruk.setEditable(false);
        areaStruk.setFont(new Font("Monospaced", Font.PLAIN, 11));

        // Inisialisasi buttons
        btnAdd = new JButton("Add to Cart");
        btnAdd.setBackground(new Color(46, 204, 113));
        btnAdd.setForeground(Color.black);
        btnAdd.setFocusPainted(false);

        btnHapus = new JButton("Hapus Item");
        btnHapus.setBackground(new Color(231, 76, 60));
        btnHapus.setForeground(Color.black);
        btnHapus.setFocusPainted(false);

        btnCheckout = new JButton("Checkout");
        btnCheckout.setBackground(new Color(52, 152, 219));
        btnCheckout.setForeground(Color.black);
        btnCheckout.setFocusPainted(false);

        btnCetak = new JButton("Cetak Struk");
        btnCetak.setBackground(new Color(155, 89, 182));
        btnCetak.setForeground(Color.black);
        btnCetak.setFocusPainted(false);
    }

    private void setupLayout() {
        JPanel mainPanel = new JPanel(new GridLayout(1, 2, 10, 0));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        // Panel kiri - Produk
        JPanel panelProduk = createPanelProduk();

        // Panel kanan - Keranjang
        JPanel panelKeranjang = createPanelKeranjang();

        mainPanel.add(panelProduk);
        mainPanel.add(panelKeranjang);

        add(mainPanel, BorderLayout.CENTER);
    }

    private JPanel createPanelProduk() {
        JPanel panel = new JPanel(new BorderLayout());
        panel.setBorder(BorderFactory.createTitledBorder("Daftar Produk"));

        JScrollPane scrollProduk = new JScrollPane(tabelProduk);
        panel.add(scrollProduk, BorderLayout.CENTER);

        // Panel kontrol add
        JPanel panelAdd = new JPanel(new FlowLayout(FlowLayout.LEFT));
        panelAdd.add(new JLabel("Qty:"));
        panelAdd.add(qtyField);
        panelAdd.add(btnAdd);

        panel.add(panelAdd, BorderLayout.SOUTH);

        return panel;
    }

    private JPanel createPanelKeranjang() {
        JPanel panel = new JPanel(new BorderLayout());
        panel.setBorder(BorderFactory.createTitledBorder("Keranjang
Belanja"));

        JScrollPane scrollKeranjang = new JScrollPane(tabelKeranjang);
        panel.add(scrollKeranjang, BorderLayout.CENTER);

        // Panel bawah
        JPanel panelBawah = new JPanel(new BorderLayout());

        // Panel aksi dan info
        JPanel panelInfo = createPanelInfo();

        // Panel struk
        JPanel panelStruk = createPanelStruk();

        panelBawah.add(panelInfo, BorderLayout.NORTH);
        panelBawah.add(panelStruk, BorderLayout.CENTER);

        panel.add(panelBawah, BorderLayout.SOUTH);

        return panel;
    }
```

```
}

private JPanel createPanelInfo() {
    JPanel panel = new JPanel();
    panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
    panel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

    labelTotal.setAlignmentX(Component.CENTER_ALIGNMENT);
    labelPoints.setAlignmentX(Component.CENTER_ALIGNMENT);
    btnHapus.setAlignmentX(Component.CENTER_ALIGNMENT);
    btnCheckout.setAlignmentX(Component.CENTER_ALIGNMENT);
    btnCetak.setAlignmentX(Component.CENTER_ALIGNMENT);

    panel.add(labelTotal);
    panel.add(labelPoints);
    panel.add(Box.createRigidArea(new Dimension(0, 10)));
    panel.add(btnHapus);
    panel.add(Box.createRigidArea(new Dimension(0, 5)));
    panel.add(btnCheckout);
    panel.add(Box.createRigidArea(new Dimension(0, 5)));
    panel.add(btnCetak);

    return panel;
}

private JPanel createPanelStruk() {
    JPanel panel = new JPanel(new BorderLayout());
    panel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
    panel.add(new JLabel("Struk Belanja:"), BorderLayout.NORTH);
    JScrollPane scrollStruk = new JScrollPane(areaStruk);
    panel.add(scrollStruk, BorderLayout.CENTER);
    return panel;
}

private void setupEventListeners() {
```

```java
        // Event listener untuk setiap button
        btnAdd.addActionListener(e -> handleAddToCart());
        btnHapus.addActionListener(e -> handleHapusItem());
        btnCheckout.addActionListener(e -> handleCheckout());
        btnCetak.addActionListener(e -> handleCetakStruk());
    }

    private void handleAddToCart() {
        int selectedRow = tabelProduk.getSelectedRow();

        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this,
                    "Pilih produk terlebih dahulu!",
                    "Peringatan",
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        int qty;
        try {
            qty = Integer.parseInt(qtyField.getText().trim());
            if (qty <= 0) {
                throw new NumberFormatException();
            }
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this,
                    "Masukkan jumlah yang valid (angka positif)!",
                    "Error",
                    JOptionPane.ERROR_MESSAGE);
            return;
        }

        String id = (String) modelProduk.getValueAt(selectedRow, 0);
        String nama = (String) modelProduk.getValueAt(selectedRow, 1);
        int harga = Integer.parseInt((String)
modelProduk.getValueAt(selectedRow, 2));

        addOrUpdateKeranjang(id, nama, qty, harga);
        updateTotal();
        qtyField.setText("1");
    }

    private void addOrUpdateKeranjang(String id, String nama, int qty, int
harga) {
        // Cek apakah produk sudah ada di keranjang
        for (int i = 0; i < modelKeranjang.getRowCount(); i++) {
            if (modelKeranjang.getValueAt(i, 0).equals(id)) {
                // Update qty dan subtotal
                int qtyLama = (int) modelKeranjang.getValueAt(i, 2);
                int qtyBaru = qtyLama + qty;
                int subtotal = qtyBaru * harga;

                modelKeranjang.setValueAt(qtyBaru, i, 2);
                modelKeranjang.setValueAt(subtotal, i, 4);
                return;
            }
        }

        // Tambah produk baru
        int subtotal = qty * harga;
        modelKeranjang.addRow(new Object[] { id, nama, qty, harga, subtotal
});
    }

    private void handleHapusItem() {
        int selectedRow = tabelKeranjang.getSelectedRow();

        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this,
                    "Pilih item yang ingin dihapus!",
                    "Peringatan",
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        int confirm = JOptionPane.showConfirmDialog(this,
                "Hapus item dari keranjang?",
                "Konfirmasi",
                JOptionPane.YES_NO_OPTION);

        if (confirm == JOptionPane.YES_OPTION) {
            modelKeranjang.removeRow(selectedRow);
            updateTotal();
        }
    }

    private void handleCheckout() {
        if (modelKeranjang.getRowCount() == 0) {
            JOptionPane.showMessageDialog(this,
                    "Keranjang kosong!",
                    "Peringatan",
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        int total = hitungTotal();
        int points = total / 1000;

        // Simpan data untuk struk
        generateStrukData(total, points);

        String message = String.format(
                "Checkout berhasil!\n\nTotal: %s\nPoints: %d\n\nKeranjang
telah dikosongkan.",
                currencyFormat.format(total),
                points);

        JOptionPane.showMessageDialog(this,
                message,
                "Checkout Berhasil",
                JOptionPane.INFORMATION_MESSAGE);

        // Kosongkan keranjang
        modelKeranjang.setRowCount(0);
        updateTotal();
    }

    private void handleCetakStruk() {
        if (strukTerakhir.isEmpty()) {
            JOptionPane.showMessageDialog(this,
                    "Belum ada transaksi yang di-checkout!",
                    "Peringatan",
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        areaStruk.setText(strukTerakhir);
    }

    private void updateTotal() {
        int total = hitungTotal();
        int points = total / 1000;

        labelTotal.setText("Total: " + currencyFormat.format(total));
        labelPoints.setText("Points: " + points);
    }
    private int hitungTotal() {
        int total = 0;
        for (int i = 0; i < modelKeranjang.getRowCount(); i++) {
            total += (int) modelKeranjang.getValueAt(i, 4);
        }
        return total;
    }

    private void generateStrukData(int total, int points) {
        StringBuilder struk = new StringBuilder();
        struk.append("========================================\n");
        struk.append("          TOKO DZAKIR TSABIT           \n");
        struk.append("           241511071                   \n");
        struk.append("========================================\n");
        struk.append(String.format("Tanggal: %s\n",
java.time.LocalDateTime.now()
                .format(java.time.format.DateTimeFormatter.ofPattern("dd/MM/y
yyy HH:mm"))));
        struk.append("========================================\n\n");

        for (int i = 0; i < modelKeranjang.getRowCount(); i++) {
            String nama = (String) modelKeranjang.getValueAt(i, 1);
            int qty = (int) modelKeranjang.getValueAt(i, 2);
            int harga = (int) modelKeranjang.getValueAt(i, 3);
            int subtotal = (int) modelKeranjang.getValueAt(i, 4);

            struk.append(String.format("%-25s\n", nama));
            struk.append(String.format("  %d x %s = %s\n\n",
                    qty,
                    currencyFormat.format(harga),
                    currencyFormat.format(subtotal)));
        }

        struk.append("========================================\n");
        struk.append(String.format("TOTAL:              %s\n",
currencyFormat.format(total)));
        struk.append(String.format("POINTS:             %d\n", points));
        struk.append("========================================\n");
        struk.append("      Terima kasih atas kunjungan      \n");
        struk.append("                Anda!                  \n");
        struk.append("========================================\n");

        strukTerakhir = struk.toString();
    }

    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName(
));
        } catch (ClassNotFoundException | InstantiationException |
IllegalAccessException | UnsupportedLookAndFeelException e) {
            JOptionPane.showMessageDialog(null,
                "Error setting look and feel: " + e.getMessage(),
                "Error",
                JOptionPane.ERROR_MESSAGE);
        }

        SwingUtilities.invokeLater(KasirUI::new);
    }
}
```

SonarQube problems :



```
KasirUI.java src\code\
Use static access with "javax.swing.WindowConstants" for "EXIT_ON_CLOSE". (java:S3252) [Ln 43, Col 40]
Define a constant instead of duplicating this literal "Peringatan" 4 times. [+4 locations] (java:S1192) [Ln 214, Col 20]
%n should be used in place of \n to produce the platform-specific line separator. (java:S3457) [Ln 299, Col 25]
Define a constant instead of duplicating this literal "========================================⏎" 5 times. [+5 locations] (java:S1192) [Ln 344, Col 21]
%n should be used in place of \n to produce the platform-specific line separator. (java:S3457) [Ln 348, Col 21]
%n should be used in place of \n to produce the platform-specific line separator. (java:S3457) [Ln 358, Col 25]
%n should be used in place of \n to produce the platform-specific line separator. (java:S3457) [Ln 359, Col 25]
%n should be used in place of \n to produce the platform-specific line separator. (java:S3457) [Ln 366, Col 21]
%n should be used in place of \n to produce the platform-specific line separator. (java:S3457) [Ln 367, Col 21]
```

SonarQube Penjelasan :

A. Move this file to a named package / package error (screenshot sebelumnya)

- **Issue:** Sonar/IDE mengingatkan agar file berada di package (tidak di default package). Kamu sudah menambahkan package code; tetapi IDE sempat memprotes karena file belum berada di source root yang benar.
- **Perubahan:** Tetap gunakan package code; (lihat baris pertama). Pastikan file berada di src/code/KasirUI.java (atau konfigurasi proyek VS Code sumbernya mengarah ke folder yang berisi code).
- **Alasan:** Java menghubungkan struktur package ke struktur folder. Menaruh class di package memudahkan modularisasi dan mencegah nama bentrok.
- **Pengujian:** Jika proyek di-build dari src sebagai source root, error “declared package does not match expected package” hilang. Kalau tidak ingin repot, bisa hapus deklarasi package — tapi itu bukan best practice.

B. Use static access with 'javax.swing.WindowConstants' for "EXIT_ON_CLOSE"

- **Issue:** Menggunakan JFrame.EXIT_ON_CLOSE (yang sebenarnya berasal dari WindowConstants interface). Sonar merekomendasikan referensi yang lebih eksplisit.
- **Perubahan:** Ganti ke setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
- **Alasan:** Mengakses konstanta dari tempat deklarasinya dianggap lebih jelas/tepat.
- **Dampak:** Tidak mengubah behavior, hanya gaya akses konstanta. Pengujian: jalankan aplikasi, klik tombol close → aplikasi berhenti (sama seperti sebelumnya).

C. Define a constant instead of duplicating this literal "Peringatan" 4 times.

- **Issue:** Literal "Peringatan" muncul berulang-ulang pada judul dialog.
- **Perubahan:** Tambah private static final String TITLE_WARNING = "Peringatan"; dan gunakan konstanta itu di semua JOptionPane.
- **Alasan:** Mengurangi duplikasi, mempermudah perubahan satu tempat bila ingin ganti kata.
- **Dampak:** Tidak mengubah fungsionalitas, tapi memperbaiki maintainability dan memenuhi aturan kualitas.

D. Define a constant instead of duplicating this literal "==" 5 times.

- **Issue:** Baris pembatas (line separator visual) muncul berulang.
- **Perubahan:** Tambah LINE_SEPARATOR constant dan pakai di generateStrukData.
- **Alasan:** Konsistensi, memudahkan perubahan layout struk di satu tempat.
- **Dampak:** Struk masih sama, tapi kode lebih bersih.

E. %n should be used in place of \n to produce the platform-specific line separator.

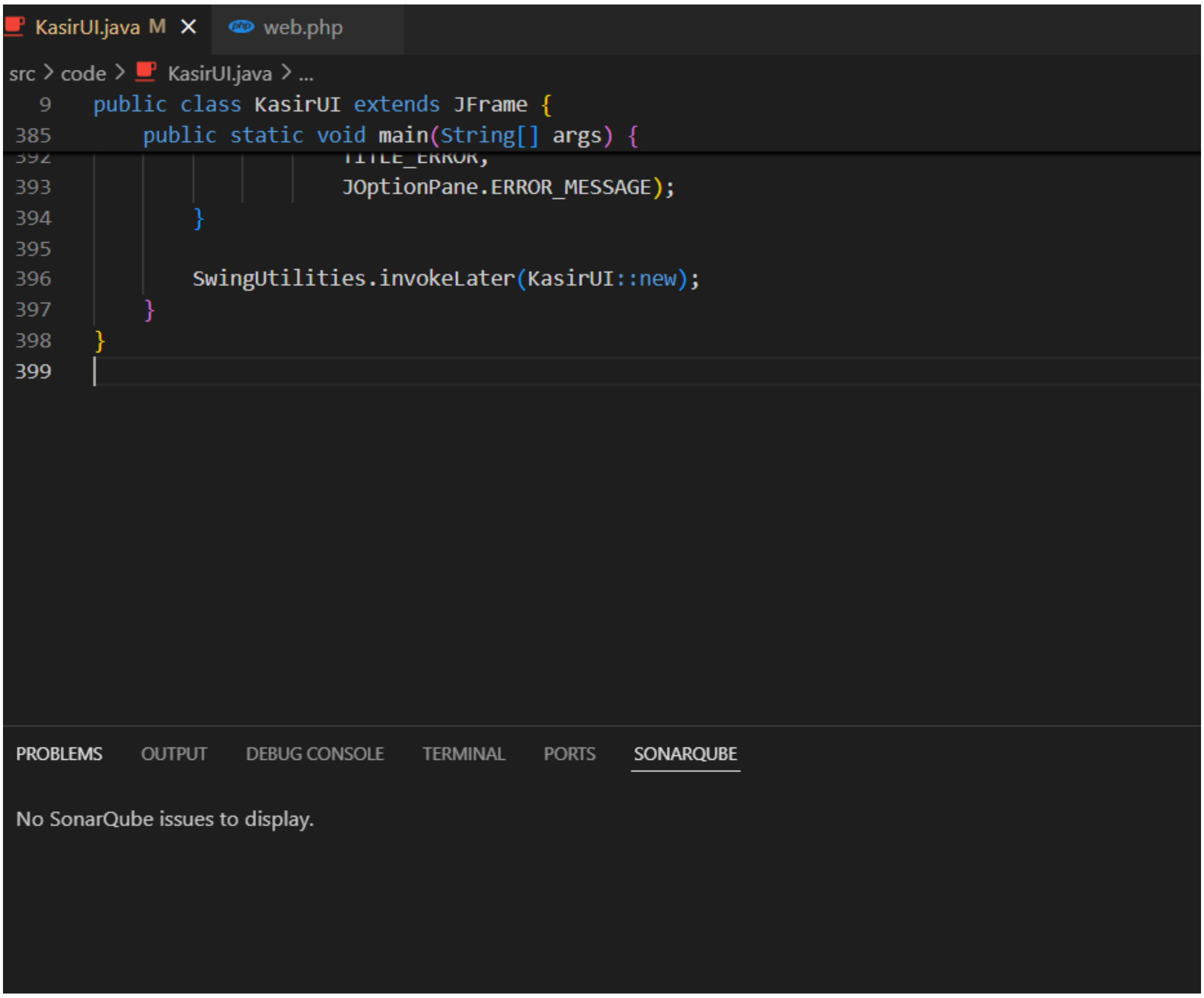
- **Issue:** Banyak penggunaan \n di String.format(...) atau append("...\n"). \n tidak menjamin kompatibilitas newline pada Windows (\r\n) vs Unix (\n).
- **Perubahan:**
  - Di String.format() ganti \n menjadi %n (sebagai placeholder platform-specific newline).
  - Untuk StringBuilder.append(...) ganti "\n" dengan .append(System.lineSeparator()).
- **Alasan:** String.format(..., "%n") dan System.lineSeparator() menyesuaikan dengan OS. Ini menghindari potensi masalah tampilan struk di Windows vs Linux.
- **Dampak:** Struk yang dicetak ke JTextArea akan memiliki newline yang benar pada OS yang bersangkutan. Tidak ada perubahan pada isi struk, hanya cara pembuatan newline.

Code setelah di perbaiki (sonarqube) :

```
package code;


import java.awt.*;
import java.text.NumberFormat;
import java.util.Locale;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;


public class KasirUI extends JFrame {


    // Konstanta yang dipakai berulang
    private static final String LINE_SEPARATOR = 
"========================================";
    private static final String TITLE_WARNING = "Peringatan";
    private static final String TITLE_ERROR = "Error";
    private static final String TITLE_CHECKOUT = "Checkout 
Berhasil";


    // Komponen GUI
    private JTable tabelProduk;
    private JTable tabelKeranjang;
    private JTextField qtyField;
    private JLabel labelTotal;
    private JLabel labelPoints;
    private JTextArea areaStruk;
    private DefaultTableModel modelProduk;
    private DefaultTableModel modelKeranjang;

    // Buttons - simpan sebagai instance variables
    private JButton btnAdd;
    private JButton btnHapus;
    private JButton btnCheckout;
    private JButton btnCetak;

    // Format mata uang
    private final NumberFormat currencyFormat =
NumberFormat.getCurrencyInstance(new Locale("id", "ID"));

    // Data untuk struk terakhir
    private String strukTerakhir = "";

    public KasirUI() {
        initializeFrame();
        initializeComponents();
        setupLayout();
        setupEventListeners();
        setVisible(true);
    }

    private void initializeFrame() {
        setTitle("GUI Dzakir Tsabit - 241511071");
        // akses yang lebih eksplisit ke konstanta exit
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(900, 600);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());
    }

    private void initializeComponents() {
        // Inisialisasi tabel produk
        String[][] dataProduk = {
                { "P001", "Air Mineral 600ml", "3000" },
                { "P002", "Kopi Sachet", "5000" },
                { "P003", "Roti isi", "8000" },
                { "P004", "Snack Keripik", "6000" },
                { "P005", "Minuman Botol", "12000" }
        };
        String[] kolomProduk = { "ID", "Nama Produk", "Harga
(Rp)" };
        modelProduk = new DefaultTableModel(dataProduk,
kolomProduk) {
            @Override
            public boolean isCellEditable(int row, int column) {
                return false; // Tidak bisa diedit manual
            }
        };
        tabelProduk = new JTable(modelProduk);
        tabelProduk.setSelectionMode(ListSelectionModel.SINGLE_S
ELECTION);

        // Inisialisasi tabel keranjang
        String[] kolomKeranjang = { "ID", "Nama Produk", "Qty",
"Harga", "Subtotal" };
        modelKeranjang = new DefaultTableModel(kolomKeranjang,
0) {
            @Override
            public boolean isCellEditable(int row, int column) {
                return false; // Tidak bisa diedit manual
            }
        };
        tabelKeranjang = new JTable(modelKeranjang);
        tabelKeranjang.setSelectionMode(ListSelectionModel.SINGL
E_SELECTION);

        // Inisialisasi komponen lainnya
        qtyField = new JTextField("1", 5);
        labelTotal = new JLabel("Total: Rp 0");
        labelTotal.setFont(new Font("Segoe UI", Font.BOLD, 14));
        labelPoints = new JLabel("Points: 0");
        areaStruk = new JTextArea(8, 20);
        areaStruk.setEditable(false);
        areaStruk.setFont(new Font("Monospaced", Font.PLAIN,
11));

        // Inisialisasi buttons
        btnAdd = new JButton("Add to Cart");
        btnAdd.setBackground(new Color(46, 204, 113));
        btnAdd.setForeground(Color.black);
        btnAdd.setFocusPainted(false);

        btnHapus = new JButton("Hapus Item");
        btnHapus.setBackground(new Color(231, 76, 60));
        btnHapus.setForeground(Color.black);
        btnHapus.setFocusPainted(false);

        btnCheckout = new JButton("Checkout");
        btnCheckout.setBackground(new Color(52, 152, 219));
        btnCheckout.setForeground(Color.black);
        btnCheckout.setFocusPainted(false);

        btnCetak = new JButton("Cetak Struk");
        btnCetak.setBackground(new Color(155, 89, 182));
        btnCetak.setForeground(Color.black);
        btnCetak.setFocusPainted(false);
    }

    private void setupLayout() {
        JPanel mainPanel = new JPanel(new GridLayout(1, 2, 10,
0));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(10,
10, 10, 10));

        // Panel kiri - Produk
        JPanel panelProduk = createPanelProduk();

        // Panel kanan - Keranjang
        JPanel panelKeranjang = createPanelKeranjang();

        mainPanel.add(panelProduk);
        mainPanel.add(panelKeranjang);

        add(mainPanel, BorderLayout.CENTER);
    }
    private JPanel createPanelProduk() {
        JPanel panel = new JPanel(new BorderLayout());
        panel.setBorder(BorderFactory.createTitledBorder("Daftar Produk"));

        JScrollPane scrollProduk = new JScrollPane(tabelProduk);
        panel.add(scrollProduk, BorderLayout.CENTER);

        // Panel kontrol add
        JPanel panelAdd = new JPanel(new FlowLayout(FlowLayout.LEFT));
        panelAdd.add(new JLabel("Qty:"));
        panelAdd.add(qtyField);
        panelAdd.add(btnAdd);

        panel.add(panelAdd, BorderLayout.SOUTH);

        return panel;
    }

    private JPanel createPanelKeranjang() {
        JPanel panel = new JPanel(new BorderLayout());
        panel.setBorder(BorderFactory.createTitledBorder("Keranjang Belanja"));

        JScrollPane scrollKeranjang = new JScrollPane(tabelKeranjang);
        panel.add(scrollKeranjang, BorderLayout.CENTER);

        // Panel bawah
        JPanel panelBawah = new JPanel(new BorderLayout());
        // Panel aksi dan info
        JPanel panelInfo = createPanelInfo();

        // Panel struk
        JPanel panelStruk = createPanelStruk();

        panelBawah.add(panelInfo, BorderLayout.NORTH);
        panelBawah.add(panelStruk, BorderLayout.CENTER);

        panel.add(panelBawah, BorderLayout.SOUTH);

        return panel;
    }

    private JPanel createPanelInfo() {
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        panel.setBorder(BorderFactory.createEmptyBorder(10, 10,
10, 10));

        labelTotal.setAlignmentX(Component.CENTER_ALIGNMENT);
        labelPoints.setAlignmentX(Component.CENTER_ALIGNMENT);
        btnHapus.setAlignmentX(Component.CENTER_ALIGNMENT);
        btnCheckout.setAlignmentX(Component.CENTER_ALIGNMENT);
        btnCetak.setAlignmentX(Component.CENTER_ALIGNMENT);

        panel.add(labelTotal);
        panel.add(labelPoints);
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(btnHapus);
        panel.add(Box.createRigidArea(new Dimension(0, 5)));
        panel.add(btnCheckout);
        panel.add(Box.createRigidArea(new Dimension(0, 5)));
        panel.add(btnCetak);

        return panel;
    }

    private JPanel createPanelStruk() {
        JPanel panel = new JPanel(new BorderLayout());
        panel.setBorder(BorderFactory.createEmptyBorder(10, 10,
10, 10));
        panel.add(new JLabel("Struk Belanja:"),
BorderLayout.NORTH);
        JScrollPane scrollStruk = new JScrollPane(areaStruk);
        panel.add(scrollStruk, BorderLayout.CENTER);
        return panel;
    }

    private void setupEventListeners() {
        // Event listener untuk setiap button
        btnAdd.addActionListener(e -> handleAddToCart());
        btnHapus.addActionListener(e -> handleHapusItem());
        btnCheckout.addActionListener(e -> handleCheckout());
        btnCetak.addActionListener(e -> handleCetakStruk());
    }

    private void handleAddToCart() {
        int selectedRow = tabelProduk.getSelectedRow();

        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this,
                    "Pilih produk terlebih dahulu!",
                    TITLE_WARNING,
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        int qty;
        try {
            qty = Integer.parseInt(qtyField.getText().trim());
            if (qty <= 0) {
                throw new NumberFormatException();
            }
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this,
                    "Masukkan jumlah yang valid (angka 
positif)!",
                    TITLE_ERROR,
                    JOptionPane.ERROR_MESSAGE);
            return;
        }

        String id = (String) modelProduk.getValueAt(selectedRow, 
0);
        String nama = (String) 
modelProduk.getValueAt(selectedRow, 1);
        int harga = Integer.parseInt((String) 
modelProduk.getValueAt(selectedRow, 2));

        addOrUpdateKeranjang(id, nama, qty, harga);
        updateTotal();
        qtyField.setText("1");
    }
    private void addOrUpdateKeranjang(String id, String nama,
int qty, int harga) {
        // Cek apakah produk sudah ada di keranjang
        for (int i = 0; i < modelKeranjang.getRowCount(); i++) {
            if (modelKeranjang.getValueAt(i, 0).equals(id)) {
                // Update qty dan subtotal
                int qtyLama = (int) modelKeranjang.getValueAt(i,
2);
                int qtyBaru = qtyLama + qty;
                int subtotal = qtyBaru * harga;

                modelKeranjang.setValueAt(qtyBaru, i, 2);
                modelKeranjang.setValueAt(subtotal, i, 4);
                return;
            }
        }

        // Tambah produk baru
        int subtotal = qty * harga;
        modelKeranjang.addRow(new Object[] { id, nama, qty,
harga, subtotal });
    }

    private void handleHapusItem() {
        int selectedRow = tabelKeranjang.getSelectedRow();

        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this,
                    "Pilih item yang ingin dihapus!",
                    TITLE_WARNING,
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        int confirm = JOptionPane.showConfirmDialog(this,
                "Hapus item dari keranjang?",
                "Konfirmasi",
                JOptionPane.YES_NO_OPTION);

        if (confirm == JOptionPane.YES_OPTION) {
            modelKeranjang.removeRow(selectedRow);
            updateTotal();
        }
    }

    private void handleCheckout() {
        if (modelKeranjang.getRowCount() == 0) {
            JOptionPane.showMessageDialog(this,
                    "Keranjang kosong!",
                    TITLE_WARNING,
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        int total = hitungTotal();
        int points = total / 1000;

        // Simpan data untuk struk
        generateStrukData(total, points);
```

```
        String message = String.format(
                "Checkout berhasil!%n%nTotal: %s%nPoints:
%d%n%nKeranjang telah dikosongkan.",
                currencyFormat.format(total),
                points);

        JOptionPane.showMessageDialog(this,
                message,
                TITLE_CHECKOUT,
                JOptionPane.INFORMATION_MESSAGE);

        // Kosongkan keranjang
        modelKeranjang.setRowCount(0);
        updateTotal();
    }

    private void handleCetakStruk() {
        if (strukTerakhir.isEmpty()) {
            JOptionPane.showMessageDialog(this,
                    "Belum ada transaksi yang di-checkout!",
                    TITLE_WARNING,
                    JOptionPane.WARNING_MESSAGE);
            return;
        }

        areaStruk.setText(strukTerakhir);
    }

    private void updateTotal() {
        int total = hitungTotal();
        int points = total / 1000;
        labelTotal.setText("Total: " +
currencyFormat.format(total));
        labelPoints.setText("Points: " + points);
    }

    private int hitungTotal() {
        int total = 0;
        for (int i = 0; i < modelKeranjang.getRowCount(); i++) {
            total += (int) modelKeranjang.getValueAt(i, 4);
        }
        return total;
    }

    private void generateStrukData(int total, int points) {
        StringBuilder struk = new StringBuilder();
        struk.append(LINE_SEPARATOR).append(System.lineSeparator
());
        struk.append("          TOKO DZAKIR
TSABIT           ").append(System.lineSeparator());
        struk.append("           241511071                   ").
append(System.lineSeparator());
        struk.append(LINE_SEPARATOR).append(System.lineSeparator
());

        struk.append(String.format("Tanggal: %s%n",
                java.time.LocalDateTime.now()
                        .format(java.time.format.DateTimeFormatt
er.ofPattern("dd/MM/yyyy HH:mm"))));
        struk.append(LINE_SEPARATOR).append(System.lineSeparator
());

        for (int i = 0; i < modelKeranjang.getRowCount(); i++) {
            String nama = (String) modelKeranjang.getValueAt(i,
1);
            int qty = (int) modelKeranjang.getValueAt(i, 2);
            int harga = (int) modelKeranjang.getValueAt(i, 3);
            int subtotal = (int) modelKeranjang.getValueAt(i,
4);

            struk.append(String.format("%-25s%n", nama));
            struk.append(String.format("  %d x %s = %s%n%n",
                    qty,
                    currencyFormat.format(harga),
                    currencyFormat.format(subtotal)));
        }

        struk.append(LINE_SEPARATOR).append(System.lineSeparator
());
        struk.append(String.format("TOTAL:              %s%n",
currencyFormat.format(total)));
        struk.append(String.format("POINTS:             %d%n",
points));
        struk.append(LINE_SEPARATOR).append(System.lineSeparator
());
        struk.append("      Terima kasih atas
kunjungan      ").append(System.lineSeparator());
        struk.append("               Anda!                  ").
append(System.lineSeparator());
        struk.append(LINE_SEPARATOR).append(System.lineSeparator
());

        strukTerakhir = struk.toString();
    }

    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException
                | UnsupportedLookAndFeelException e) {
            JOptionPane.showMessageDialog(null,
                    "Error setting look and feel: " + e.getMessage(),
                    TITLE_ERROR,
                    JOptionPane.ERROR_MESSAGE);
        }

        SwingUtilities.invokeLater(KasirUI::new);
    }
}
```

SonarQube (fixed) :



```java
public class KasirUI extends JFrame {
    public static void main(String[] args) {
                    TITLE_ERROR,
                    JOptionPane.ERROR_MESSAGE);
        }

        SwingUtilities.invokeLater(KasirUI::new);
    }
}

```

No SonarQube issues to display.