

PEMBUATAN APLIKASI KASIR
UAS



DOSEN PENGAMPU:
SLAMET TRYANTO, S,ST

Disusun Oleh:

BELLA TRI FEBRI
202013002

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK KAMPAR
2021

KATA PENGANTAR

Puji syukur di ucapkan kehadirat Allah SWT atas segala rahmat nya sehingga laporan Tugas UAS PBO ini dapat tersusun sampai selesai, dengan judul project “PEMBUATAN APLIKASI KASIR“. Adapun tujuan penulisan ini adalah untuk melengkapi Tugas Perkuliahan.

Dalam proses pembuatan laporan ini, tentunya penulis mendapat bimbingan, arahan, koreksi, dan saran.untuk itu penulis mengucapkan terimakasih kepada Bapak SLAMET TRYANTO, S,ST selaku dosen pegampu Pemograman Berorientasi Objek.

Penulis menyadari bahwa baik dari segi penulisan maupun isi. Laporan ini masih jauh dari kata sempurna, oleh karena itu penulis sangat mengharapkan kritik yang membangun dan saran dari pembaca agar terbentuknya kesempurnaan Laporan ini. Atas partisipasinya penulis mengucapkan terimakasih.

Bangkinang, 1 Agustus 2021

Penulis

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI	3
DAFTAR GAMBAR	4
BAB I	5
PENDAHULUAN	5
1.1 Latar Belakang	5
1.2 Tujuan Pembahasan.....	5
BAB II	6
ISI	6
2.1 Tampilan FlowChart.....	6
2.2 Tampilan Source Code	7
2.1 Tampilan Menu Utama.....	13
KESIMPULAN	14

DAFTAR GAMBAR

1. 1 <i>Source Code</i> Pertama	7
1. 2 <i>Source Code</i> Kedua.....	8
1. 3 <i>Source Code</i> Ketiga.....	9
1. 4 <i>Source Code</i> Keempat.....	10
1. 5 <i>Source Code</i> Kelima	11
1. 6 <i>Source Code</i> Keenam.....	12
1. 7 Tampilan Menu Utama	13

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era sekarang ini, persaingan pada bisnis restoran semakin ketat. Selain rasa masakan, kualitas pelayanan juga menjadi hal yang krusial demi nama baik restoran tersebut. Beberapa restoran telah menggunakan sistem informasi untuk menyelesaikan permasalahan dalam proses bisnis yang umum terjadi pada restoran seperti proses pemesanan, pembayaran, pengaturan tempat hingga pengaturan keuangan, proses pembelian bahan, penentuan harga, dll.

Proses-proses diatas kebanyakan masih dilakukan secara manual oleh pemilik restoran. Tentunya hal ini sangat tidak efisien untuk restoran terutama yang berskala menengah hingga besar karena akan memakan waktu dan tenaga. Belum lagi faktor kesalahan individu atau “human error” sering terjadi bila tidak memiliki proses bisnis dan komitmen yang kuat dalam pelaksanaannya. Pemilihan teknologi untuk sistem harus disesuaikan dengan keadaan dan proses bisnis restoran. Hal ini berdampak langsung ke kecepatan pelayanan dan manajemen restoran.

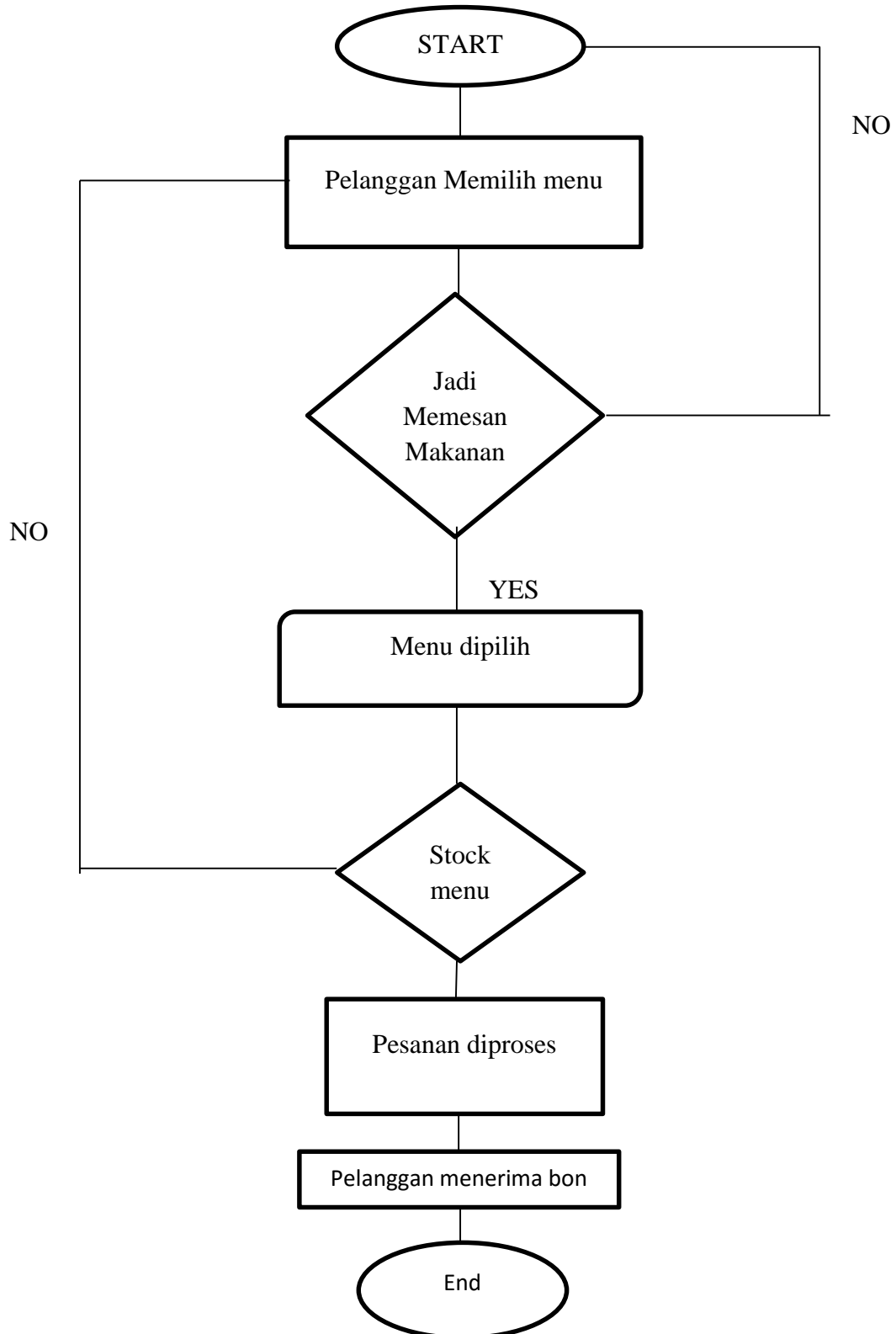
1.2 Tujuan Pembahasan

1. Membuat sebuah sistem informasi POS (Point of Sales) yang dapat membantu petugas restoran dalam mencatat transaksi penjualan yang dilakukan.
2. Membuat sistem informasi yang dapat menyediakan laporan yang dibutuhkan oleh pemilik restoran.
3. Membuat sistem informasi POS (Point of Sales) yang disesuaikan dengan proses bisnis restoran.

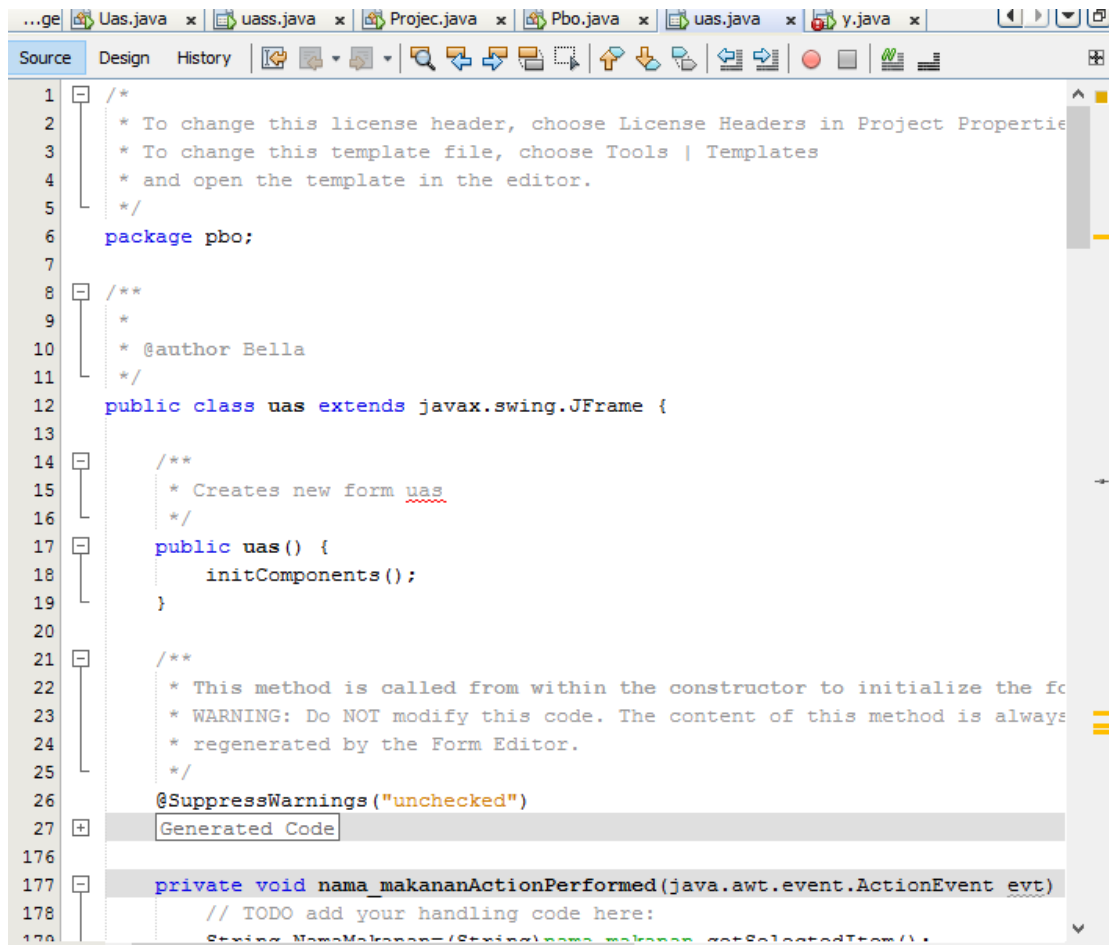
BAB II

ISI

2.1 Tampilan FlowChart

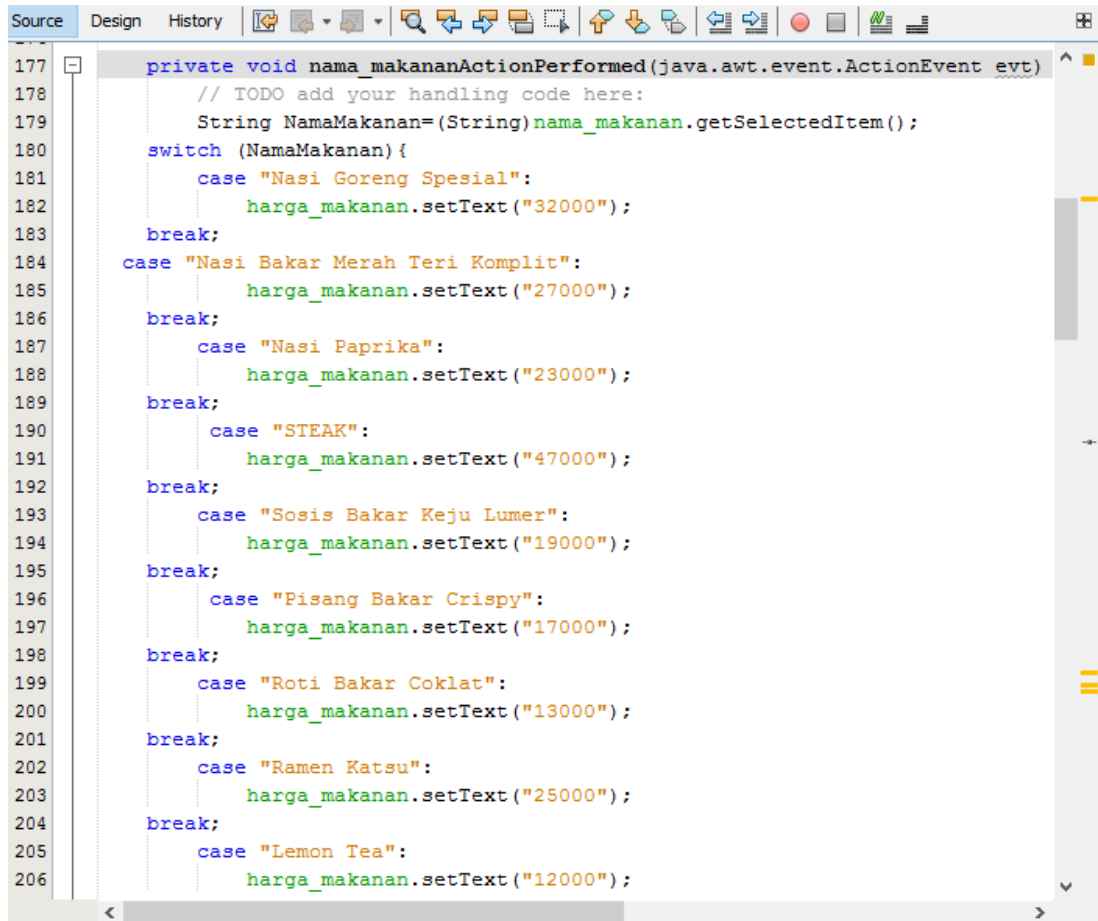


2.2 Tampilan Source Code



```
1  /**
2   * To change this license header, choose License Headers in Project Properties
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package pbo;
7
8  /**
9   *
10   * @author Bella
11   */
12  public class uas extends javax.swing.JFrame {
13
14      /**
15       * Creates new form uas
16       */
17      public uas() {
18          initComponents();
19      }
20
21      /**
22       * This method is called from within the constructor to initialize the form
23       * WARNING: Do NOT modify this code. The content of this method is always
24       * regenerated by the Form Editor.
25       */
26      @SuppressWarnings("unchecked")
27      Generated Code
28
29      private void nama_makananActionPerformed(java.awt.event.ActionEvent evt) {
30          // TODO add your handling code here:
31          String NamaMakanan = (String) nama_makanan.getSelectedItem();
32      }
33  }
```

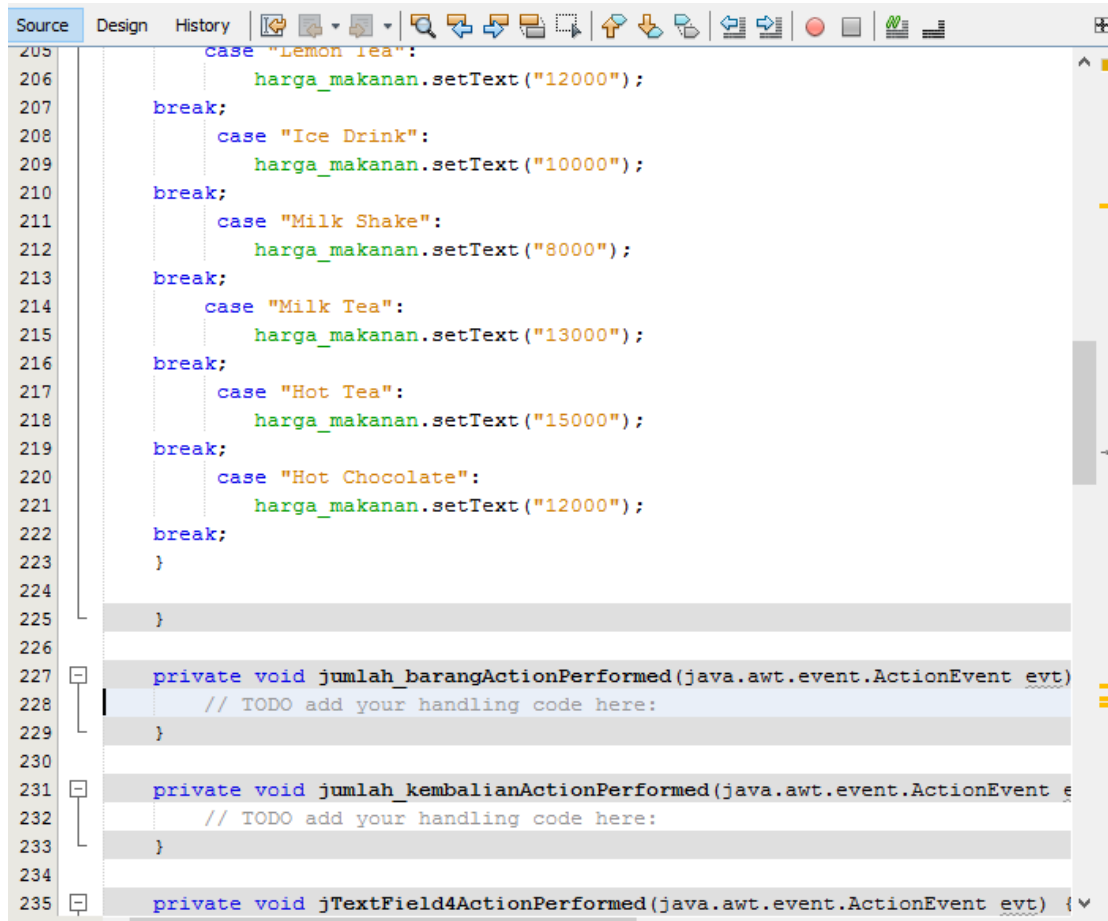
1. 1 Source Code Pertama



The image shows a screenshot of an IDE window with the 'Source' tab selected. The code is a Java method named `nama_makananActionPerformed` that takes a `java.awt.event.ActionEvent` parameter. It contains a `switch` statement that maps food names to their prices. The food names are: "Nasi Goreng Spesial", "Nasi Bakar Merah Teri Komplit", "Nasi Paprika", "STEAK", "Sosis Bakar Keju Lumer", "Pisang Bakar Crispy", "Roti Bakar Coklat", "Ramen Katsu", and "Lemon Tea". The prices are: 32000, 27000, 23000, 47000, 19000, 17000, 13000, 25000, and 12000 respectively. The code is as follows:

```
177 private void nama_makananActionPerformed(java.awt.event.ActionEvent evt)
178     // TODO add your handling code here:
179     String NamaMakanan=(String)nama_makanan.getSelectedItem();
180     switch (NamaMakanan) {
181         case "Nasi Goreng Spesial":
182             harga_makanan.setText("32000");
183         break;
184     case "Nasi Bakar Merah Teri Komplit":
185         harga_makanan.setText("27000");
186     break;
187         case "Nasi Paprika":
188             harga_makanan.setText("23000");
189         break;
190         case "STEAK":
191             harga_makanan.setText("47000");
192         break;
193         case "Sosis Bakar Keju Lumer":
194             harga_makanan.setText("19000");
195         break;
196         case "Pisang Bakar Crispy":
197             harga_makanan.setText("17000");
198         break;
199         case "Roti Bakar Coklat":
200             harga_makanan.setText("13000");
201         break;
202         case "Ramen Katsu":
203             harga_makanan.setText("25000");
204         break;
205         case "Lemon Tea":
206             harga_makanan.setText("12000");
```

1. 2 Source Code Kedua



```
205     case "Lemon Tea":
206         harga_makanan.setText("12000");
207     break;
208     case "Ice Drink":
209         harga_makanan.setText("10000");
210     break;
211     case "Milk Shake":
212         harga_makanan.setText("8000");
213     break;
214     case "Milk Tea":
215         harga_makanan.setText("13000");
216     break;
217     case "Hot Tea":
218         harga_makanan.setText("15000");
219     break;
220     case "Hot Chocolate":
221         harga_makanan.setText("12000");
222     break;
223     }
224 }
225
226
227 private void jumlah_barangActionPerformed(java.awt.event.ActionEvent evt)
228 {
229     // TODO add your handling code here:
230 }
231
232 private void jumlah_kembalianActionPerformed(java.awt.event.ActionEvent evt)
233 {
234     // TODO add your handling code here:
235 }
236
237 private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {
238     // TODO add your handling code here:
239 }
```

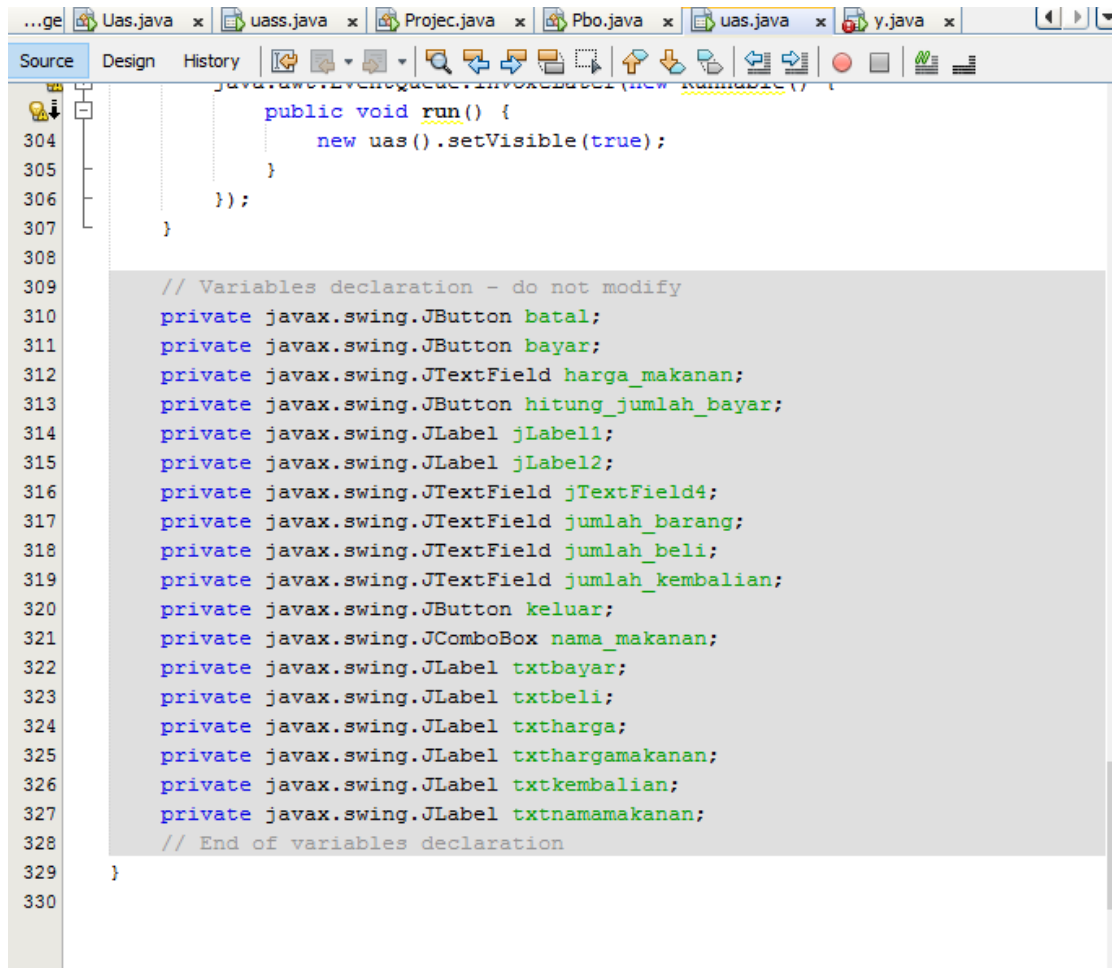
1.3 Source Code Ketiga

```
Source Design History
239 private void hitung_jumlah_bayarActionPerformed(java.awt.event.ActionEvent ^
240     // TODO add your handling code here:
241     int jumlahbeli=Integer.parseInt(jumlah_beli.getText());
242     int hargamakanan=Integer.parseInt(harga_makanan.getText());
243     jumlah_barang.setText(Integer.toString(hargamakanan*jumlahbeli));
244     //txthargamakanan =Integer.parseInt(harga_makanan.getText());
245
246 }
247
248 private void keluarActionPerformed(java.awt.event.ActionEvent evt) {
249     // TODO add your handling code here:
250     System.exit(0);
251 }
252
253 private void batalActionPerformed(java.awt.event.ActionEvent evt) {
254     // TODO add your handling code here:
255     harga_makanan.setText("");
256     jumlah_beli.setText("");
257     harga_makanan.setText("");
258     bayar.setText("");
259     jumlah_kembalian.setText("");
260     nama_makanan.setSelectedIndex(0);
261 }
262
263 private void bayarActionPerformed(java.awt.event.ActionEvent evt) {
264     // TODO add your handling code here:
265     int jumlahbayar=Integer.parseInt(hitung_jumlah_bayar.getText());
266     int jumlahharga=Integer.parseInt(jumlah_beli.getText());
267     if(jumlahbayar>=jumlahharga){
268         jumlah_kembalian.setText(Integer.toString(jumlahbayar-jumlahharga);
```

1. 4 Source Code Keempat

```
Source Design History
239 private void hitung_jumlah_bayarActionPerformed(java.awt.event.ActionEvent ^
240     // TODO add your handling code here:
241     int jumlahbeli=Integer.parseInt(jumlah_beli.getText());
242     int hargamakanan=Integer.parseInt(harga_makanan.getText());
243     jumlah_barang.setText(Integer.toString(hargamakanan*jumlahbeli));
244     //txthargamakanan =Integer.parseInt(harga_makanan.getText());
245
246 }
247
248 private void keluarActionPerformed(java.awt.event.ActionEvent evt) {
249     // TODO add your handling code here:
250     System.exit(0);
251 }
252
253 private void batalActionPerformed(java.awt.event.ActionEvent evt) {
254     // TODO add your handling code here:
255     harga_makanan.setText("");
256     jumlah_beli.setText("");
257     harga_makanan.setText("");
258     bayar.setText("");
259     jumlah_kembalian.setText("");
260     nama_makanan.setSelectedIndex(0);
261 }
262
263 private void bayarActionPerformed(java.awt.event.ActionEvent evt) {
264     // TODO add your handling code here:
265     int jumlahbayar=Integer.parseInt(hitung_jumlah_bayar.getText());
266     int jumlahharga=Integer.parseInt(jumlah_beli.getText());
267     if(jumlahbayar>=jumlahharga){
268         jumlah_kembalian.setText(Integer.toString(jumlahbayar-jumlahharga);
```

1. 5 Source Code Kelima



```
...ge Uas.java x uass.java x Projec.java x Pbo.java x uas.java x y.java x
Source Design History
public void run() {
    new uas().setVisible(true);
}
});
}

// Variables declaration - do not modify
private javax.swing.JButton batal;
private javax.swing.JButton bayar;
private javax.swing.JTextField harga_makanan;
private javax.swing.JButton hitung_jumlah_bayar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jumlah_barang;
private javax.swing.JTextField jumlah_beli;
private javax.swing.JTextField jumlah_kembalian;
private javax.swing.JButton keluar;
private javax.swing.JComboBox nama_makanan;
private javax.swing.JLabel txtbayar;
private javax.swing.JLabel txtbeli;
private javax.swing.JLabel txtharga;
private javax.swing.JLabel txthargamakanan;
private javax.swing.JLabel txtkembalian;
private javax.swing.JLabel txtnamamakanan;
// End of variables declaration
}
```

1. 6 Source Code Keenam

2.1 Tampilan Menu Utama

Design Preview [uas]



The interface features a pink background with a cartoon chef character in the center. The title "BELLA RESTO" is at the top. On the left, there are labels for various input fields: "NAMA MAKANAN", "HARGA MAKANAN", "JUMLAH BELI", "JUMLAH BARANG", "JUMLAH BAYAR", and "JUMLAH KEMBALIAN". The input fields are white with rounded corners. A dropdown menu labeled "PILIH MENU" is positioned next to the "NAMA MAKANAN" label. A grey button labeled "HITUNG JUMLAH BAYAR" is located between the "JUMLAH BELI" and "JUMLAH BARANG" fields. At the bottom right, there are two grey buttons labeled "BATAL" and "KELUAR". A "BAYAR" button is located below the "JUMLAH BAYAR" field.

BELLA RESTO

NAMA MAKANAN

HARGA MAKANAN

JUMLAH BELI

JUMLAH BARANG

JUMLAH BAYAR

JUMLAH KEMBALIAN

1. 7 Tampilan Menu Utama

KESIMPULAN

Berdasarkan uraian yang telah dibahas pada bab-bab sebelumnya dan hasil pembahasan maka dapat diambil kesimpulan

1. Aplikasi ini telah berhasil dibuat dan berjalan dengan baik sesuai fungsinya sebagai Aplikasi Kasir Bella Resto
2. Dari hasil analisis, desain, implementasi dan testing, dapat dibangun sebuah aplikasi sebagai media yang memberi informasi. Aplikasi Kasir Bella Resto menyediakan Berbagai menu makanan yang disediakan.