

MANUAL BOOK



Aplikasi Pembelian Bahan Baku Makanan Pada Restoran Warung Pasta Bekasi

Website pembelian bahan baku makanan pada Restoran Warung Pasta Bekasi adalah website yang digunakan oleh perusahaan dalam pembelian bahan baku makanan pada restoran. Website ini digunakan sebagai kontrol bagi perusahaan dalam mengetahui pembelian bahan baku dan penyediaan stok bahan baku di restoran. Fitur yang dimiliki dari website ini diantaranya, mengelola data master user, data bahan baku, data supplier, data akun, mengelola transaksi serta laporan. Berikut adalah tampilan dari login website :



Pengguna dari website ini ada 3, yaitu bagian Purchasing, bagian Personal In charge (PIC) dan Manager. Untuk bagian Purchasing bertugas mengelola data master yaitu data akun, data supplier, data user, data transaksi yaitu pembelian dan pembayaran, serta pembuatan laporan bahan baku. Sedangkan untuk bagian PIC bertugas untuk mengelola data barang dan untuk bagian Manager melihat data laporan pembelian dan laporan akun Berikut adalah tampilan menu utama aplikasi :

1. Menu utama bagian Purchasing



2. Menu utama bagian PIC



3. Menu utama Manager



Pada menu utama terdapat menu-menu diantaranya :

1. Home

Menu file digunakan untuk beranda website.

2. Logout

Menu file digunakan untuk logout dari website.

3. Data Master

Menu master digunakan untuk mengelola master data, diantaranya data barang, data supplier, data akun dan data user.

4. Transaksi

Menu transaksi digunakan untuk mengelola transaksi, diantaranya transaksi pembelian, pembayaran dan jurnal umum.

5. Laporan

Menu laporan digunakan untuk membuat laporan-laporan dari data master dan data transaksi.

Tampilan kelola data user :

1) Menu data user

[Data Nama User Warung Pasta]



[Tambah User Baru](#)

Cari Data Akun :

ID User	Nama User	Password	Hak Akses	Action
MG001	Kristopher Kenji	*****	Manager	Edit Hapus
PC001	Soleh Yusro	*****	Pic	Edit Hapus
PR001	Michael Wirianto	*****	Purchasing	Edit Hapus
PR002	Prawira Kusuma	*****	Purchasing	Edit Hapus

2) Tambah data user

Tambah User Baru

ID User	PR002
Nama User	Prawira Kusuma
Password	
Hak Akses	Purchasing
Simpan	Batal

Data telah ditambahkan

Oke

3) Update data user

[Edit User Warung Pasta]

ID User	: PR002
Nama User	: Prawira Kusuma
Password	:
Hak Akses	: Purchasing
Update	

Data telah di Update

Oke

4) Hapus data user

Data telah di Hapus

Oke

Data *User* dikelola oleh bagian Admin Purchasing. Admin Purchasing dapat menambahkan data data user yang dapat menggunakan website pembelian bahan baku ini.

Tampilan kelola data *supplier* :

1) Menu data *supplier*

[Data Nama Supplier Warung Pasta]

 [Tambah Supplier Baru](#)

Cari Data Akun :

Kode Supplier	Nama Supplier	Alamat Supplier	No Telephone Supplier	Action
SP0001	CV. Alam Jaya	Jl. Cempaka Raya No. 02 Jakarta	081281239045	Edit Hapus
SP0002	PT. Bogasari Flour Mills	Jl. Raya Cililing No.01 Jakarta utara	021-29263800	Edit Hapus
SP0003	PT. Sukanda Djaya	Jl. Pasir Putih Kav. 1 Ancol Jakarta	021-6405678	Edit Hapus

1) Tambah data *supplier*

Tambah Supplier Baru

Kode Supplier	SP0001
Nama Supplier	CV. Alam Jaya
Alamat Supplier	Jl. Cempaka Raya No. 02 Jakarta
No Telpn Supplier	081281239045
Simpan	Batal

Data telah ditambahkan

2) *Update data supplier*

[Edit Supplier Warung Pasta]

Kode supplier	: SP0001
Nama supplier	: CV. Alam Jaya
Alamat	: Jl. Cempaka Raya No. 02 Ji
No Telephone	: 081281239045
Update	

Data telah di Update

3) *Hapus data supplier*

Data telah di Hapus

Data supplier dikelola oleh bagian Admin Purchasing. Admin Purchasing dapat menambahkan data supplier yang bekerja sama dengan perusahaan untuk memenuhi kebutuhan bahan baku makanan.

Tampilan kelola data akun :

1) Menu data akun

[Data Nama Akun Warung Pasta]

 [Tambah Akun Baru](#)

Cari Data Akun :

Kode Akun	Nama Akun	Jenis Akun	Saldo Normal	Action
1-1100	Bank Mandiri	Kas/Bank	Debet	Edit Hapus
1-1300	Persediaan Bahan Baku	Aktiva Lancar	Debet	Edit Hapus

2) Tambah data akun

Tambah Akun Baru



Kode Akun	1-1100
Nama Akun	Bank Mandiri
Jenis Akun	Kas/Bank
Saldo Normal	Debet
Simpan	Batal

Data telah ditambahkan

3) Edit data akun

[Edit Akun Warung Pasta]	
Kode Akun	: 1-1100
Nama Akun	: Bank Mandiri
Jenis Akun	: Kas/Bank
Saldo Normal	: Debet
Update	

Data telah di Update

4) Hapus data akun

Data telah di Hapus

Data akun dikelola oleh bagian Admin Purchasing. Admin Purchasing dapat menambahkan data akun-akun apa saja yang digunakan dalam transaksi pembelian bahan baku ini.

Tampilan kelola data bahan baku :

1) Menu data bahan baku

[Data Nama Bahan Baku Warung Pasta]				
 Tambah Bahan Baku Baru				
Cari Data Akan : Cari				
Kode Bahan Baku	Nama Bahan Baku	Harga Beli	Stok Bahan Baku	Action
BB0001	Aqua Botol @Dus	37000	2	Edit Hapus
BB0002	Supu Full Cream @pack	14500	4	Edit Hapus
BB0003	Pasta Spaghetti @pack	18500	5	Edit Hapus
BB0004	Pasta Fettucine @pack	17000	7	Edit Hapus
BB0005	Pasta Fusili @pack	17000	9	Edit Hapus
BB0006	Ice Cream @pack	30000	12	Edit Hapus

2) Tambah data bahan baku

Tambah Bahan Baku Baru	
	
Kode Bahan Baku	BB0001
Nama Bahan Baku	Aqua Botol
Harga Beli	37000
Stok Bahan Baku	2
Simpan	Batal

Data telah ditambahkan

3) *Update* data bahan baku

[Edit Bahan Baku Warung Pasta]

Kode Bahan Baku	: BB0001
Nama Bahan Baku	: Aqua Botol
Harga Beli	: 37000
Stok Bahan Baku	: 2
Update	

Data telah di Update

Data telah di Hapus

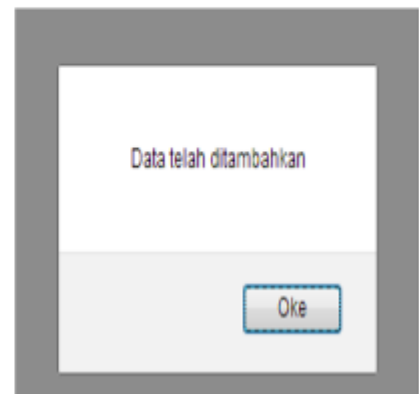
4) Hapus data bahan baku

Data bahan baku dikelola oleh bagian PIC. PIC dapat menambahkan data bahan baku sesuai dengan bahan baku yang diterima oleh restoran dari supplier. Biasanya data bahan baku dibutuhkan saat terjadi proses pembelian bahan baku.

Tampilan kelola transaksi pembelian :

[From Pembelian]

Kode Pembelian	PM00000003
Tanggal Beli	hh / bb / tttt
Kode Bahan Baku	BB0004 Pasta Fettucine @pack ▼
Kode Supplier	SP0001 CV. Alam Jaya ▼
Satuan	Pack
Harga Beli	17000
Quantity	6
Sub Total	102000
Simpan	



Kode Pembelian	Tanggal	Kode Barang	Kode Supplier	Satuan	Harga Beli	Quantity	Sub Total
PM00000001	2019-07-02	BB0001	SP0001	Dus	37000	4	148000
PM00000002	2019-07-02	BB0003	SP0002	Pack	18500	5	92500

Tampilan kelola transaksi pembayaran :

[Form Pembayaran]

Kode Pembayaran	PB00000002
Kode Pembelian	PM00000002
Tanggal Bayar	02 / 07 / 2019
Jumlah Bayar	92500
Simpan	

Kode Pembayaran	Kode Pembelian	Tanggal	Jumlah Bayar
PB00000001	PM00000001	2019-07-02	148000

Data telah ditambahkan

Oke

Tampilan kelola transaksi jurnal umum :

Jurnal Umum Warung Pasta:

No Referensi	JU000000002		
Tanggal Transaksi	03 / 07 / 2019		
Kode Pembelian	PM000000001		
Keterangan	Pembelian Agua Botol		

	Kode dan Nama Akun	Debet	Kredit
☐	1-1100 Bank Mandiri	0	0
☐	Kode Akun Nama Akun	0	0
☐	Kode Akun Nama Akun	0	0
☐	Kode Akun Nama Akun	0	0
Total		0	0
Balance			0

SimpanBatal

Data telah ditambahkan
Oke

Data Jurnal Umum Warung Pasta:

No.Referensi	Tanggal Transaksi	Kode Pembelian	Keterangan	Kode Akun	Debet	Kredit
JU000000002	2019-07-03	PM000000001	Pembelian Aqua Botol	1-1100	50000	0
JU000000002	2019-07-03	PM000000001	Pembelian Aqua Botol	1-1300	0	50000

Transaksi pembelian, pembayaran dan jurnal umum dikelola oleh Admin Purchasing. Admin Purchasing menginput transaksi pembelian sesuai dengan permintaan dari pihak restoran untuk pengadaan bahan baku makanan di operasional. Transaksi pembayaran diinput jika bahan baku sudah diterima oleh pihak restoran dan Admin Purchasing mengelola pembayaran kepada Supplier. Untuk transaksi jurnal umum diinput berdasarkan transaksi pembelian yang sudah dilakukan sehari-hari.

Tampilan laporan data bahan baku dan laporan pembelian :

[Laporan Data Bahan Baku]

[Cetak Laporan Data Bahan Baku](#)

Warung Pasta

Jl. Sudap Malam No. 504, Jaka Setia, Bekasi Selatan
Email: kalimalang@warung-pasta.com
Telp: (021) 82419731



LAPORAN DATA BAHAN BAKU

Tanggal Cetak : Tuesday 10 September 2019

Kode Bahan Baku	Nama Bahan Baku	Harga Beli	Stok Bahan Baku
BB0001	Aqua Botol @Dus	37000	7
BB0002	Susu Full Cream @pack	14500	9
BB0003	Pasta Spaghetti @pack	18500	15
BB0004	Pasta Fettucine @pack	17000	14
BB0005	Pasta Fusili @pack	17000	9
BB0006	Saus Cream @pack	20000	12
BB0007	Saus Tomat @pack	18500	6

[Laporan Pembelian]

Masukkan Bulan :	Juli
Masukkan Tahun :	2019
Cetak	

Warung Pasta

Jl. Sedap Malam No. 504, Jaka Setia, Bekasi Selatan
Email: kalimalang@warung-pasta.com
Telp: (021) 82419731



LAPORAN PEMBELIAN

Bulan : July
Tahun : 2019

Tanggal Cetak Tuesday 10 September 2019

Kode Pembelian	Tanggal	Nama Bahan Baku	Nama Supplier	Satuan	Harga Beli	Qty	Subtotal
PM00000001	02/07/2019	Aqua Botol @Dus	CV. Alam Jaya	Pack	37000	4	148000
PM00000002	02/07/2019	Pasta Spaghetti @pack	PT. Bogasari Flour	Pack	18500	5	92500

Laporan bahan baku dapat diakses oleh bagian Admin Purchasing dan Manager. Sedangkan untuk laporan pembelian hanya diakses oleh Manager. Laporan dapat dicetak berdasarkan periode tanggalnya dan berdasarkan bulannya. Untuk mencetak laporan, dapat memilih dulu laporan yang ingin dicetak berdasarkan periode atau bulan. Setelah itu tekan tombol cetak dan laporan bahan baku dan pembelian pun tercetak.

Source Code JSP

```
<sql:query var="qryuser" dataSource="${dataSource}">
  <!-- perubahan muai dari where ...-->
  SELECT * FROM pembelian WHERE kode_pembelian
  LIKE '%${param.cari}%'
  OR tgl
  LIKE '%${param.cari}%'
</sql:query>
```

```

<%@page import="java.sql.*, model.Supplier, model.Barang, model.Pembelian"%>

<%
    Supplier supplier = new Supplier();
    Barang barang = new Barang();
    Pembelian pembelian = new Pembelian();
    //--koneksi database--
    Connection koneksi = null;
    Statement stmt = null;
    ResultSet rs = null;
    Class.forName("com.mysql.jdbc.Driver");
    koneksi = DriverManager.getConnection("jdbc:mysql://localhost/warung_pasta",
    "root", "");
    stmt = koneksi.createStatement();
    rs = stmt.executeQuery("SELECT * FROM pembelian"
        + " WHERE kode_pembelian");
%>

<h1>[ From Pembelian ]</h1>
<form action="pembelian1Servlet" method="post">
    <style type="text/css">
        table {
            font-family: Arial, Helvetica, sans-serif;
            background-color: #ff9900;
            border-style: ridge;
            border-color: #c40b00;
            border-collapse: separate;
            text-align: left;
            padding: 4px;
            color: white;
            font-size: 12px;
        }
        a {
            color: black;
            font: "Times New Roman",sans-serif;

```

```

    font-size: 20px;
}
h1 {
    border-style: inset;
    border-collapse: separate;
    background-color: #c40b00;
    width: 430px;
    height: 50px;
    border-color: #c40b00;
    font-family: 'cambria', sans-serif;
    color: white;
    letter-spacing: 6px;
    font-size: 35px;
    margin-bottom: 10px;
    margin-top: 10px;
}
input {
    border-color: white;
    color: #ff9900;
    font-family: Georgia, Arial, sans-serif;
    font-size: 16px;
    padding: 3px;
}
input:hover {
    background-color: #ff9900;
    color: #ffffff;
}
input.button {
    background-color: white;
    color: #ff9900;
    font-family: 'Roboto', sans-serif;
    margin-right: 10px;
}
input.button:hover {
    background-color: #ff9900;

```



```

        color: #ffffff;
    }
    select {
        border-color: white;
        color: #ff9900;
        font-family: Georgia, Arial, sans-serif;
    }
    select:hover {
        background-color: #ff9900;
        color: #ffffff;
    }
    select option [selected="selected"]
    {
        font-weight: bold;
        background-color: #ff9900;
        color: #ff9900;
    }
    tr:nth-child(even){background-color: #ffffff; color: black;}
</style>
<script type="text/javascript">
    function showEmp(emp_value)
    {
        if (document.getElementById("emp_id").value != "-1")
        {
            xmlhttp = GetXmlHttpRequestObject()
            if (xmlhttp == null)
            {
                alert("Browser does not support HTTP Request")
                return
            }
            var url = "getbarang.jsp"
            url = url + "?emp_id=" + emp_value

            xmlhttp.onreadystatechange = stateChanged
            xmlhttp.open("GET", url, true)

```

```

        xmlHttp.send(null)
    } else
    {
        alert("-");
    }
}

function stateChanged()
{
    document.getElementById("ename").value = "";
    document.getElementById("emp_id").value = "";
    if (xmlHttp.readyState == 4 || xmlHttp.readyState == "complete")
    {

        var showdata = xmlHttp.responseText;
        var strar = showdata.split(":");

        if (strar.length == 1)
        {
            document.getElementById("emp_id").focus();
            alert("-");
            document.getElementById("ename").value = " ";
            document.getElementById("emp_id").value = " ";
        } else if (strar.length > 1)
        {
            document.getElementById("ename").value = strar[1];
        }

    }
}

function GetXmlHttpRequestObject()
{
    var xmlHttp = null;
    try

```

```

    {
        xmlHttp = new XMLHttpRequest();
    } catch (e)
    {
        try
        {
            xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e)
        {
            xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return xmlHttp;
}
</script>
<table border="1">
|
|  |

```

```

        int noLong = nomorpm.length();

        for (int a = 1; a < 9 - noLong; a++) {
            nomorpm = "0" + nomorpm;
        }
        String nomerpm = "PM" + nomorpm;
        out.println("<input type='hidden' name='kode_pembelian'
value='" + nomerpm + "'/>");
        out.println("<input type='text' disabled='disabled' value='" +
nomerpm + "'/>");
    }
}
} catch (Exception e) {
    out.println(e);
}
%>
</td>
</tr>
<tr>
    <td>Tanggal Beli</td>
    <td><input type="date" name="tgl"/></td>
</tr>
<tr style="background-color: #ff9900; color:
white;"> <td>Kode Bahan Baku</td>
<td>
        <select name="kode_barang" onchange="showEmp(this.value);">
            <option value="">Pilih Kode Bahan Baku</option>
        <%
            rs = stmt.executeQuery("SELECT * FROM barang
"); while (rs.next()) {
                barang.setKode_barang(rs.getString("kode_barang"));
                barang.setNama_barang(rs.getString("nama_barang"));
            %>

```

```

        <option
value="<%=barang.getKode_barang()%>"><%=barang.getKode_barang()%> ||
<%=barang.getNama_barang()%></option>
        <% }%>
    </select>

</td>
</tr>
<tr>
    <td>Kode Supplier</td>
    <td>
        <select name="kode_supplier">
            <option value="">Pilih Kode Supplier</option>
            <%
                rs = stmt.executeQuery("SELECT * FROM
supplier"); while (rs.next()) {
                    supplier.setKode_supplier(rs.getString("kode_supplier"));
                    supplier.setNama_supplier(rs.getString("nama_supplier"));

                %>
                <option
value="<%=supplier.getKode_supplier()%>"><%=supplier.getKode_supplier()%> ||
<%=supplier.getNama_supplier()%></option>
                <% }%>
            </select>
        </td>
<tr style="background-color: #ff9900; color:
white;"> <td>Satuan</td>
        <td><input type="text" name="satuan"
size="25"/></td> </tr>
<tr style="background-color: #ff9900; color:
white;"> <td>Harga Beli</td>
        <td id="emp_id" value="" ><input type="text" name="harga_beli"
id="ename" value="" onchange="beli()" readonly/></td>
    </tr>

```

```

        <tr style="background-color: #ff9900; color:
            white;"> <td>Quantity</td>
            <td><input type="text" name="qty" size="25" id="qty"
onchange="jumlah()"/></td>
        </tr>
        <tr>
            <td>Subtotal</td>
            <td><input type="text" name="subtotal" size="25" id="subtotal"
readonly/></td>
        </tr>
        <tr>
            <td colspan="3" align="center"><input type="submit" name="aksi"
value="Simpan" class="button"/></td>
        </tr>
    </table>
</form>

<script type="text/javascript">
    function jumlah() {
        var jumlah = document.getElementById("qty").value; var
        beli = document.getElementById("ename").value; var
        subtotal = jumlah * beli;
        document.getElementById("subtotal").value = subtotal;
    }
</script>
<form action="#" method="post"><br>
    Cari Data Pembelian : <input type="text" name="cari" value="{param.cari}"/>
<%-- modif value --%>
    <input type="submit" value=" Cari " class="button"/><br>
</form>
<table border="5" cellpadding="5" cellspacing="5">
    <tr class="head">
        <td>Kode Pembelian</td>
        <td>Tanggal</td>
        <td>Kode Barang</td>

```

```

        <td>Kode Supplier</td>
        <td>Satuan</td>
        <td>Harga Beli</td>
        <td>Quantity</td>
        <td>Subtotal</td>

    </tr>
    <c:forEach var="rowakun"
        items="${qryuser.rowsByIndex}"> <tr class="isi">
        <td align='center'>${rowakun[0]}</td>
        <td align='center'>${rowakun[1]}</td>
        <td align='center'>${rowakun[2]}</td>
        <td align='center'>${rowakun[3]}</td>
        <td align='center'>${rowakun[4]}</td>
        <td align='center'>${rowakun[5]}</td>
        <td align='center'>${rowakun[6]}</td>
        <td align='center'>${rowakun[7]}</td>

    </tr>
    </c:forEach>
</table>
</body>
</html>

```

Source Code Control Servlet

```

package control;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.Supplier; //mengakses parameter
import model.Barang;

```

```

import model.Pembelian;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.annotation.WebServlet;

/**
 *
 * @author michael
 */
@WebServlet(name = "pembelian1Servlet", urlPatterns = {"/pembelian1Servlet"})
public class pembelian1Servlet extends HttpServlet {

    /**
     * Processes requests for both HTTP GET
and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException, ClassNotFoundException, SQLException
    {
        response.setContentType("text/html;charset=UTF-8");

        Pembelian pembelian = new Pembelian(); //--Inisialisasi Object--

        String URL = "jdbc:mysql://localhost:3306/warung_pasta";
        String USERNAME = "root";
        String PASSWORD = "";

```



```

Connection koneksi = null;
PreparedStatement pstmt = null;
PreparedStatement pstmt2 = null;
int result = 0;
try (PrintWriter out = response.getWriter()) {
    /* TODO output your page here. You may use following sample code.
    */ Class.forName("com.mysql.jdbc.Driver");
    koneksi = DriverManager.getConnection(URL, USERNAME, PASSWORD);

    String aksi = request.getParameter("aksi");
    if (aksi != null) {
        pembelian.setKode_pembelian(request.getParameter("kode_pembelian"));
        pembelian.setTanggal(request.getParameter("tgl"));
        pembelian.setKode_barang(request.getParameter("kode_barang"));
        pembelian.setKode_supplier(request.getParameter("kode_supplier"));
        pembelian.setSatuan(request.getParameter("satuan"));
        pembelian.setHarga_beli(request.getParameter("harga_beli"));
        pembelian.setQty(request.getParameter("qty"));
        pembelian.setSubtotal(request.getParameter("subtotal")); switch (aksi) {

            case "Simpan":

                if (pembelian.getKode_pembelian().equals("") ||
                    pembelian.getTanggal().equals("") || pembelian.getKode_barang().equals("") ||
                    pembelian.getKode_supplier().equals("") || pembelian.getSatuan().equals("") ||
                    pembelian.getHarga_beli().equals("") || pembelian.getQty().equals("") ||
                    pembelian.getSubtotal().equals("")){
                    out.println("<script>alert('Gagal... masih ada data yang belum terisi, Silahkan Ulangi!!!')");
                    + "</script><meta http-equiv='refresh'
content='1;beranda1.jsp?hal=pembelian'/>"); }
                else {
                    pstmt = koneksi.prepareStatement("INSERT INTO pembelian "
                        + " VALUES(?, ?, ?, ?, ?, ?, ?, ?, 'tidak tampil')");

```

```

        pstmt.setString(1, pembelian.getKode_pembelian());
        pstmt.setString(2, pembelian.getTanggal());
        pstmt.setString(3, pembelian.getKode_barang());
        pstmt.setString(4, pembelian.getKode_supplier());
        pstmt.setString(5, pembelian.getSatuan());
        pstmt.setString(6, pembelian.getHarga_beli());
        pstmt.setString(7, pembelian.getQty());
        pstmt.setString(8, pembelian.getSubtotal());

        result = pstmt.executeUpdate();
        pstmt2 = koneksi.prepareStatement("UPDATE barang SET
stok_barang = stok_barang + ? WHERE kode_barang=?");
        pstmt2.setString(1, pembelian.getQty());
        pstmt2.setString(2, pembelian.getKode_barang());

        result = pstmt2.executeUpdate();
        if (result > 0) {
            out.println("<script> "
                + "alert('Data telah ditambahkan');"
                + "document.location='beranda1.jsp?hal=pembelian';"
                + " </script>");
        }
    }
    break;
case "Delete":
    pstmt = koneksi.prepareStatement("DELETE FROM"
        + " pembelian WHERE kode_pembelian = ?");
    pstmt.setString(1, pembelian.getKode_pembelian());
    result = pstmt.executeUpdate();
    if (result > 0) {
        out.println("<script> "
            + "alert('Data telah di Hapus');"
            + "document.location='beranda1.jsp?hal=pembelian';"
            + " </script>");
    }
}

```

```

        break;
    default:
        break;
    }
}
}
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(pembelian1Servlet.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SQLException ex) {
        Logger.getLogger(pembelian1Servlet.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

/**
 * Handles the HTTP <code>POST</code> method.

```

*

```

    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        try {
            processRequest(request, response);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(pembelian1Servlet.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (SQLException ex) {
            Logger.getLogger(pembelian1Servlet.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }

    /**
     * Returns a short description of the servlet.
    */
    *
    * @return a String containing servlet description
    */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

Source Code Model Java

```
package model;
```

```
/**
```

```

*
* @author user
*/
public class Pembelian {
    private String kode_pembelian;
    private String tanggal;
    private String kode_barang;
    private String kode_supplier;
    private String satuan;
    private String harga_beli;
    private String qty;
    private String subtotal;

    public String getKode_pembelian() {
        return kode_pembelian;
    }

    public void setKode_pembelian(String kode_pembelian)
    { this.kode_pembelian = kode_pembelian;
    }

    public String getTanggal() {
        return tanggal;
    }

    public void setTanggal(String tanggal) {
        this.tanggal = tanggal;
    }

    public String getKode_barang() {
        return kode_barang;
    }

    public void setKode_barang(String kode_barang)
    { this.kode_barang = kode_barang;

```

}

```
public String getKode_supplier() {  
    return kode_supplier;  
}
```

```
public void setKode_supplier(String kode_supplier)  
    { this.kode_supplier = kode_supplier;  
}
```

```
public String getSatuan() {  
    return satuan;  
}
```

```
public void setSatuan(String satuan) {  
    this.satuan = satuan;  
}
```

```
public String getHarga_beli() {  
    return harga_beli;  
}
```

```
public void setHarga_beli(String harga_beli) {  
    this.harga_beli = harga_beli;  
}
```

```
public String getQty() {  
    return qty;  
}
```

```
public void setQty(String qty) {  
    this.qty = qty;  
}
```

```
public String getSubtotal() {
```

```
        return subtotal;
    }

    public void setSubtotal(String subtotal) {
        this.subtotal = subtotal;
    }
}
```