

---

# Bab 5

---

## Menggunakan Java Bean

---

### POKOK BAHASAN:

- Karakteristik Java Bean
- Membuat Java Bean
- Mengeset nilai property Java Bean
- Menampilkan nilai property Java Bean

### TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. mengenal pengertian Java Bean
2. Mahasiswa mengenal sintak Java Bean
3. Mahasiswa mampu mengintegrasikan Java Bean dengan JSP

### 5.1 Karakteristik Java Bean

API Java Bean merupakan standar format dari class java. Dengan adanya Java Bean, komponen dapat digunakan sewaktu-waktu untuk membangun program. Penulisan kelas Java Bean mempunyai aturan tertentu sebagai berikut:

- a. Memiliki properti (variabel) dengan akses bukan public yang digunakan untuk menyimpan data.
- b. Memiliki method get dan set (memberi dan mendapatkan properti dari bean).
- c. Memiliki konstruktor kosong.

Berikut ini adalah langkah-langkah membuat kelas Java Bean:

- a. Mendefinisikan variabel dalam bean yang akan kita buat.

Variabel ini tidak boleh dideklarasikan public.

Misal : `private String nrp;`

- b. Untuk masing-masing variabel dibuat metode get dan set.

Misal : `getNrp()` dan `setNrp()`

Untuk boolean, gunakan method `setXxx()` untuk mengeset nilai boolean dan `isXxx` yang mengembalikan nilai boolean

Aturan membuat method:

```
public void set<namaVariabel>
```

```
public <tipe_variabel> get<namaVariabel>
```

```
public boolean is<namaVariabel>
```

- c. Menambahkan konstruktor kosong.

Berfungsi untuk inisialisasi

## 5.2 Membuat Java Bean

Memanggil javabean dalam halaman jsp memakai syntax:

```
<jsp:usebean id="name" class="package.Class"/>
```

Yang berarti pembuatan object class yang telah ditentukan dan dimasukkan ke dalam variable dengan nama yang ditentukan id.

Seumpama ada contoh syntax berikut :

```
<jsp:usebean id="book1" class="cwp.Book"/>
```

Secara normal hal sama dengan scriplet :

```
<% cwp.Book book1=new cwp.Book(); %>
```

Meskipun javabean ekivalen dengan membangun sebuah object, tapi dia mempunyai option tambahan yang membuatnya lebih powerfull. Kita dapat menentukan

scope yang mendefinisikan ruang batas dari obyek yang akan dibuat. Bisa saja scope bukan hanya pada page ini saja, tapi bisa lebih jauh. Jika ini dapat dishare tentunya akan sangat berguna, jadi kita akan membuat object baru jika dan hanya jika tidak memang benar benar tidak ada yang id dan scopenya sama.

### 5.3 Mengambil Nilai Property Java Bean

Dalam beberapa kasus mungkin kita ingin variable lokal kita mempunyai tipe yang sama sebagai object yang dibuat. Cara pengaksesan javabean adalah memakai syntax `jsp:getProperty`. Contoh pengaksesan sbb :

```
<jsp:getProperty name="book1" property="title" />
```

ekivalen dengan

```
<%= book1.getTitle() %>
```

### 5.4 Mengeset Nilai Property Java Bean

Untuk memodifikasi properti menggunakan syntax `jsp:setProperty`. Action ini mempunyai beberapa bentuk yang berbeda, tapi untuk yang paling simple perlu tiga atribut yaitu : name, property dan valuenya. Contoh syntax sederhananya adalah sbb :

```
<jsp:setProperty name="book1" property="title" value="Core Web Programming"/>
```

ekivalen dengan :

```
<%= book1.setTitle("Core Web Programmng") %>
```

Berikut ini adalah contoh sederhana Java Bean dengan JSP:

```

package coreservlets;
/** Contoh Java Bean dengan satu properti bertipe String
 * yang bernama message
 */
public class StringBean {
    private String message = "No message specified";

    public String getMessage() {
        return(message);
    }

    public void setMessage(String message) {
        this.message = message;
    }
}

```

Listing 4.1 StringBean.java

Dan pengaksesannya memakai JSP adalah sbb :

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Using JavaBeans with JSP</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER=5 ALIGN="CENTER">
      <TR><TH CLASS="TITLE"> Using JavaBeans with JSP
    </TABLE>
    <jsp:useBean id="stringBean"
class="coreservlets.StringBean"/>
    <OL>
      <LI>Initial value (getProperty):
      <I><jsp:getProperty name="stringBean"
        property="message" /></I>
      <LI>Initial value (JSP expression):
      <I><%= stringBean.getMessage() %></I>
      <LI><jsp:setProperty name="stringBean"
        property="message"
        value="Best string bean: Fortex" />
      Value after setting property with setProperty:
      <I><jsp:getProperty name="stringBean"
        property="message" /></I>
      <LI><%= stringBean.setMessage("My favorite:
        Kentucky Wonder"); %>
      Value after setting property with scriptlet:
      <I><%= stringBean.getMessage() %></I>
    </OL>

```

```
</BODY>
</HTML>
```

Listing 4.2 File JSP untuk StringBean

Berikut ini adalah contoh program pemakaian set dan get properti javabeen:

```
package coreservlets;
/** Bean sederhana untuk mengilustrasikan berbagai macam
 * bentuk jsp:setProperty.
 */

public class SaleEntry {
    private String itemID = "unknown";
    private double discountCode = 1.0;
    private int numItems = 0;

    public String getItemID() {
        return(itemID);
    }

    public void setItemID(String itemID) {
        if (itemID != null) {
            this.itemID = itemID;
        } else {
            this.itemID = "unknown";
        }
    }

    public double getDiscountCode() {
        return(discountCode);
    }

    public void setDiscountCode(double discountCode) {
        this.discountCode = discountCode;
    }

    public int getNumItems() {
        return(numItems);
    }

    public void setNumItems(int numItems) {
        this.numItems = numItems;
    }
    // Replace this with real database lookup.
    public double getItemCost() {
        double cost;
        if (itemID.equals("a1234")) {
            cost = 12.99*getDiscountCode();
        } else {
            cost = -9999;
        }
        return(roundToPennies(cost));
    }
}
```

```

private double roundToPennies(double cost) {
    return(Math.floor(cost*100)/100.0);
}

public double getTotalCost() {
    return(getItemCost() * getNumItems());
}
}

```

Listing 4.3 SaleEntry.java

Dan pengaksesannya memakai JSP adalah sebagai berikut :

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
  <HEAD><TITLE>Using jsp:setProperty</TITLE>
</HEAD>
  <BODY>
    <TABLE BORDER=5 ALIGN="CENTER">
      <TR><TH CLASS="TITLE">
        Using jsp:setProperty</TH>
      </TR>
      <TR>
        <td>
          <jsp:useBean id="entry"
            class="coreservlets.SaleEntry" />
          <jsp:setProperty name="entry"
            property="itemID"
            value='<%= request.getParameter("itemID") %>' />
          <%
            int numItemsOrdered = 1;
            try {
              numItemsOrdered =
                Integer.parseInt(request.getParameter("numItems"));
            } catch(NumberFormatException nfe) {}
          %>
          <jsp:setProperty name="entry"
            property="numItems"
            value="<%= numItemsOrdered %>" />
          <%
            double discountCode = 1.0;
            try {
              String discountString =
                request.getParameter("discountCode");
              // Double.parseDouble not available in JDK 1.1
              discountCode =
                Double.valueOf(discountString).doubleValue();
            } catch(NumberFormatException nfe) {}
          %>
          <jsp:setProperty name="entry"
            property="discountCode"
            value="<%= discountCode %>" />
        </td>
      </TR>
    </TABLE>
  </BODY>
</HTML>

```

```

        <TH>Item ID<TH>Unit Price<TH>Number
            Ordered<TH>Total Price
        <TR ALIGN="RIGHT">
        <TD><jsp:getProperty name="entry"
                                property="itemID" />
        <TD>${jsp:getProperty name="entry"
                                property="itemCost" />
        <TD><jsp:getProperty name="entry"
                                property="numItems" />
        <TD>${jsp:getProperty name="entry"
                                property="totalCost" />
    </TABLE>
</BODY>
</HTML>

```

Listing 4.4 File JSP untuk SaleEntry

Untuk menjalankan program akses url dengan cara sebagai berikut:

<http://localhost:8080/process/SaleEntry1.jsp?itemID=a1234&numItems=11&discountCode=0.95>

Perhatikan Listing 4.4 diatas digunakan ekspresi JSP untuk memberikan nilai pada atribut value. Nilai yang diperoleh dari pengambilan parameter adalah bernilai String. Bagaimana bila nilai hasil pengambilan parameter tersebut akan diberikan pada atribut yang bertipe numerik? Diperlukan konversi tipe data seperti yang dicontohkan pada atribut discountCode dan numItems diatas. Ingat method `getParameter(String param)` akan mengembalikan nilai String.

## 5.4 Percobaan

1. Tuliskan kembali program di atas dan amati output yang dihasilkan.
2. Tulis Program di bawah ini amati output apa yang dihasilkan.

Listing 4.5 dan Listing 4.6 adalah alternatif dari praktikum Listing 4.4.

Perhatikan bahwa pada alternatif ini tidak diperlukan konversi tipe data.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Menggunakan jsp:setProperty</TITLE>
</HEAD>
<BODY>
<TABLE BORDER=5 ALIGN="CENTER">
  <TR><TH CLASS="TITLE">
    Using jsp:setProperty</TABLE>
<jsp:useBean id="entry" class="cwp.SaleEntry" />

```

```

<jsp:setProperty
  name="entry"
  property="itemID"
  param="itemID" />
<jsp:setProperty
  name="entry"
  property="numItems"
  param="numItems" />
<%-- WARNING! JDK 1.0.1 dan Java Web Server
      mempunyai bug yang menyebabkan kegagalan
      konversi tipe data
--%>
<jsp:setProperty
  name="entry"
  property="discountCode"
  param="discountCode" />
<BR>
<TABLE ALIGN="CENTER" BORDER=1>
<TR CLASS="COLORED">
  <TH>Item ID<TH>Unit Price<TH>Number Ordered<TH>Total Price
<TR ALIGN="RIGHT">
  <TD><jsp:getProperty name="entry" property="itemID" />
  <TD>$<jsp:getProperty name="entry" property="itemCost" />
  <TD><jsp:getProperty name="entry" property="numItems" />
  <TD>$<jsp:getProperty name="entry" property="totalCost" />
</TABLE>
</BODY>
</HTML>

```

Listing 4.5 Alternatif pertama file JSP untuk SaleEntry

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!--
Contoh penggunaan jsp:setProperty
-->
<HTML>
<HEAD>
<TITLE>Using jsp:setProperty</TITLE>
<LINK REL=STYLESHEET
      HREF="JSP-Styles.css"
      TYPE="text/css">
</HEAD>

<BODY>
<TABLE BORDER=5 ALIGN="CENTER">
  <TR><TH CLASS="TITLE">
    Using jsp:setProperty</TABLE>
<jsp:useBean id="entry" class="cwp.SaleEntry" />
<%-- WARNING! JDK 1.0.1 dan Java Web Server
      mempunyai bug yang menyebabkan kegagalan
      konversi tipe data secara otomatis ke tipe
      data double
--%>

```



```

<jsp:setProperty name="entry" property="*" />
<BR>
<TABLE ALIGN="CENTER" BORDER=1>
<TR CLASS="COLORED">
  <TH>Item ID<TH>Unit Price<TH>Number Ordered<TH>Total Price
<TR ALIGN="RIGHT">
  <TD><jsp:getProperty name="entry" property="itemID" />
  <TD>${jsp:getProperty name="entry" property="itemCost" />
  <TD><jsp:getProperty name="entry" property="numItems" />
  <TD>${jsp:getProperty name="entry" property="totalCost" />
</TABLE>
</BODY>
</HTML>

```

Listing 4.6 Alternatif kedua file JSP untuk SaleEntry

3. Tulis Program di bawah ini dan amati hasil percobaan di bawah ini.

```

package coreservlets;
/** Simple bean to illustrate sharing beans through
 * use of the scope attribute of jsp:useBean.
 */

public class AccessCountBean {
  private String firstPage;
  private int accessCount = 1;

  public String getFirstPage() {
    return(firstPage);
  }

  public void setFirstPage(String firstPage) {
    this.firstPage = firstPage;
  }

  public int getAccessCount() {
    return(accessCount++);
  }
}

```

Listing 4.7 AccessCountBean.java

4. Dan pengaksesannya memakai JSP adalah sebagai berikut :

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Shared Access Counts: Page 1</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER=5 ALIGN="CENTER">
      <TR><TH CLASS="TITLE">
        Shared Access Counts: Page 1</TABLE>

```

```

<P>
<jsp:useBean id="counter"
              class="coreservlets.AccessCountBean"
              scope="application">
<jsp:setProperty name="counter"
                 property="firstPage"
                 value="SharedCounts1.jsp" />

</jsp:useBean>
Of SharedCounts1.jsp (this page), <A HREF=
"SharedCounts2.jsp">SharedCounts2.jsp</A>,
and <A HREF="SharedCounts3.jsp">
SharedCounts3.jsp</A>,
<jsp:getProperty name="counter"
                 property="firstPage" />
was the first page accessed.
<P>Collectively, the three pages have been
accessed
<jsp:getProperty name="counter"
                 property="accessCount" /> times.

</BODY>
</HTML>

```

Listing 4.8 File JSP untuk AccessCountBean

## 5. Simpan Listing 4.9 di package com.chocolove.

```

/* paket dari class */
package com.chocolove;
public class CounterBean {
    int count = 3; /* nilai awal properti */
    /* method untuk mengambil nilai properti */
    public int getCount() {
        return count;
    }
    /* method untuk mengisi nilai properti */
    public void setCount(int c) {
        count = c;
    }
    /* method untuk menaikkan nilai properti */
    public int increaseCount() {
        count++;
        return count;
    }
}

```

Listing 4.9 CounterBean.java

```
<html>
  <head><title>JSP</title></head>
  <body>
    <jsp:useBean id="cacah"
                 class="com.chocolove.CounterBean"
                 scope="page">
      Nilai awal, langsung dari Action getProperty tanpa set
      lewat JSP :
      <jsp:getProperty name="cacah"
                      property="count" /><br>
      <jsp:setProperty name="cacah"
                      property="count" value="8" />
    </jsp:useBean>
    <%
      out.print("Nilai setelah diset dengan setProperty
: "+cacah.getCount());
      cacah.setCount(5);
      cacah.increaseCount();
      out.println("<br>");
      out.println("Nilai setelah dicacah
: "+cacah.getCount());
    %>
    <br>Nilai dari Action getProperty :
    <jsp:getProperty name="cacah" property="count" /><br>
  </body>
</html>
```

Listing 4.10 File JSP untuk CounterBean

## 5.5 Soal Latihan

1. Terangkan secara ringkas cara pengaksesan properti Java Bean.
2. Terangkan secara ringkas cara seting properti Java Bean.
3. Amati program-program pada dasar teori di atas output apa yang akan dihasilkan jika program diatas akan dijalankan
4. Buat Java Bean yang memiliki properti sebagai berikut:
  - a. noKTP : String
  - b. nama : String
  - c. menikah : boolean
  - d. umur : int
6. Buat file JSP yang digunakan untuk mengeset dan menampilkan semua properti Java Bean yang telah dibuat pada percobaan 4.1.
7. Buatlah package cwp. Simpan Listing 4.11 di package cwp.

```
package cwp;
public class Pesan{
    private String message;

    public Pesan() {}

    public void setMessage(String message) {
        this.message = message;
    }

    public String getMessage() {
        return(message);
    }
}
```

Listing 4.11 Pesan.java

Dan pengaksesannya memakai JSP adalah sebagai berikut :

```
<HTML>
  <HEAD>
    <TITLE>JavaBean in JSP</TITLE>
  </HEAD>
  <BODY>
    <p>
      <jsp:useBean id="coba"
                  class="cwp.Pesan" />
      <jsp:setProperty name="coba"
                      property="message"
                      value="5" />
      Mengambil nilai yang sudah di set sebelumnya:
      <jsp:getProperty name="coba" property="message" />
    </p>
  </BODY>
</HTML>
```

## Listing 4.12 File JSP untuk Pesan

Selanjutnya cobalah untuk membuat aplikasi yang mengakses data numerik (misal: int atau double)!

Atribut **value** pada `<jsp:setProperty>` digunakan untuk memberikan nilai yang bertipe String. Jika nilai pada value tersebut diberikan pada atribut numerik maka secara otomatis akan terjadi konversi nilai.