

WEB PROGRAMMING

Anna, M.Kom

Pengenalan

► **Java Server Page (JSP)**



DAFTAR ISI

Halaman Sampul	i
Daftar Isi	ii
1. BAB 1 Pengenalan dan Instalasi Software	1
1.1.Pengenalan Pemrograman Java Web	1
1.2.Pengenalan dan Instalasi Software	2
1.3.Instalasi Plugin	10
2. BAB II Pengenalan HTML	15
2.1.Sejarah HTML	15
2.2.Struktur HTML	16
2.3.Memilih Text Editor	17
2.4.Penggunaan Script HTML pada Netbeans	18
2.5.Membuat Tabel	27
3. BAB III Pemrograman Java Server Page (JSP)	32
3.1.Apa itu Java Server Page?.....	32
3.2.JSP Pada Netbeans	32
3.3.Variabel	38
3.4.Tipe Data.....	40
3.5.Pengenalan Operator	41
3.6.Pembuatan JSP Form	47
3.7.Percabangan (Branching).....	57
3.8.Perulangan.....	63
Daftar Referensi	67

BAB I

PENGENALAN & INSTALASI *SOFTWARE*

Deskripsi:

Bab ini membahas tentang pengenalan pemrograman *Java Web* dan *software* yang digunakan, instalasi *software* dan menambahkan *plugin* sesuai dengan kebutuhan *software*.

Tujuan Pembelajaran:

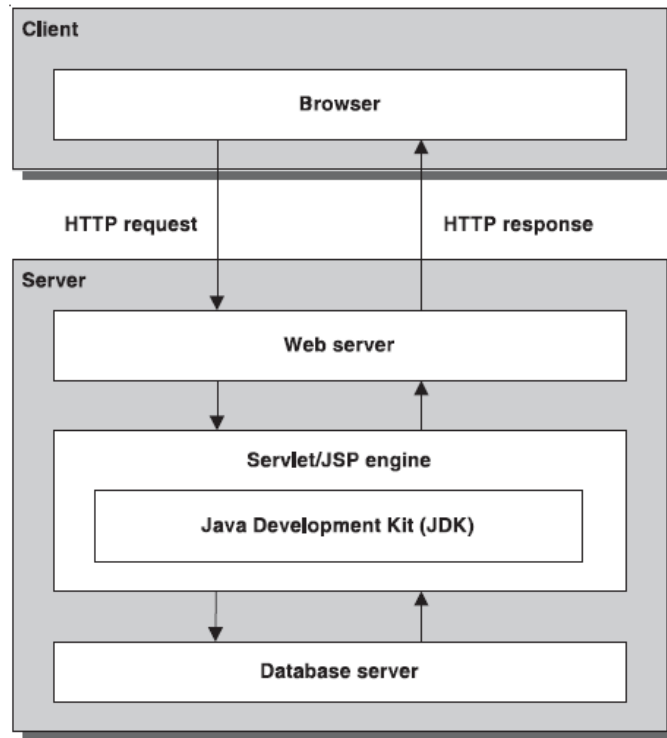
Setelah melakukan pembelajaran ini, pembaca diharapkan mampu:

1. Mengenal Pemrograman *Java Web* dan *software*.
2. Melakukan instalasi *software*.
3. Melakukan instalasi *plugin*.

1.1 Pengenalan Pemrograman Java Web

Pada bab ini, Anda akan dikenalkan pada konsep dan ketentuan yang dibutuhkan untuk memulai penggunaan Servlet dan Java Server Page (JSP) saat Anda hendak membuat aplikasi web. Ada beberapa *software* yang dibutuhkan untuk membuat sebuah aplikasi *Java website* dengan menggunakan *Java Server Page (JSP)*. *JSP* adalah halaman web dinamis berdasarkan kode-kode *HTML* yang disisipkan dengan kode-kode program *Java*. Cara kerja *JSP engine* sendiri harus dapat mengakses *Java Development Kit (JDK)* yang merupakan sebagai *Java Standard Edition (Java SE)*. *Java Development Kit* ini berisi *Java Compiler* dan *Java Runtime Environment* untuk menjalankan *Compile Class* pada *Java*.

Untuk menjalankan *Java Server Pages*, Anda harus melakukan beberapa proses instalasi, yaitu program *Java Versi J2SE JDK 7* ataupun *8*, *Glasfish server* yang sudah ter-install pada saat instalasi editor *Java Netbeans*, dan *XAMPP package* sebagai web server. Berikut adalah komponen dari Aplikasi *Java Web*.



Gambar 1.1 Komponen-komponen dari Aplikasi Java Web JSP.

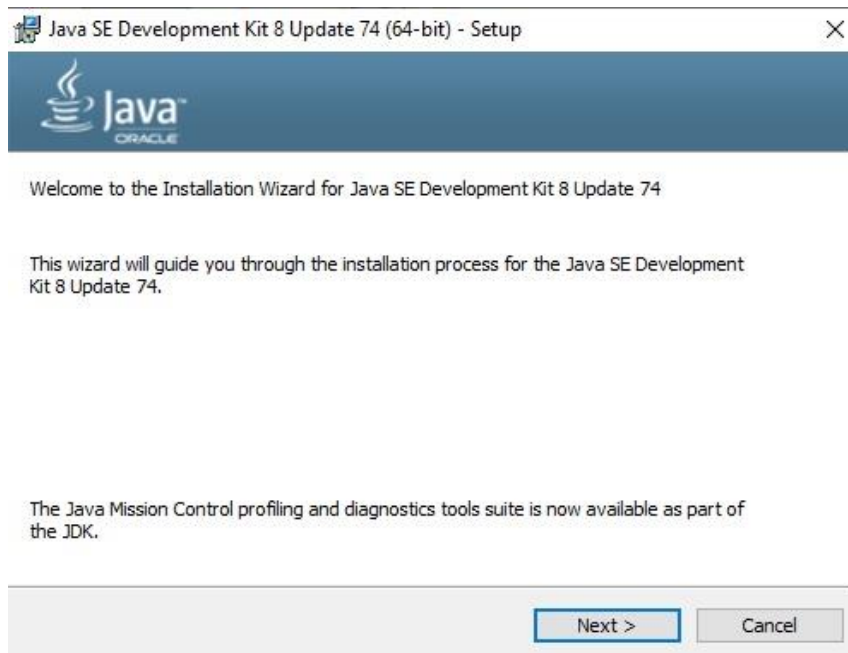
1.2 Pengenalan dan Instalasi *Software*

1.2.1 JDK (Java Development Kit)

Java Development Kit atau yang dikenal dengan JDK ini merupakan software yang digunakan untuk melakukan proses kompilasi dari Java code ke bytecode. Semua package seperti JDK/JRE, Netbeans akan terinstall secara bersamaan dan hanya membutuhkan satu kali proses instalasi. Jika ingin menggunakan Code Editor Netbeans maka diwajibkan untuk pemasangan JDK terlebih dahulu.

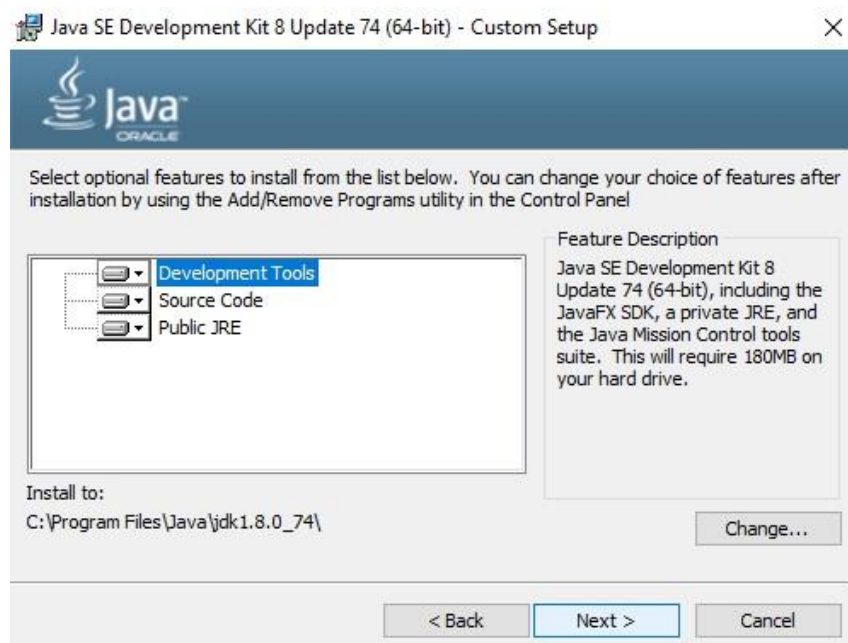
Setelah mengecek jenis windows pada laptop/komputer Anda. Barulah kita akan menginstal JDK yang sesuai dengan system type 64 bit atau 32 bit. Berikut cara untuk menginstal JDK.

Klik dua kali pada file **jdk-8u74-windows-x64**, maka akan tampil jendela seperti berikut.



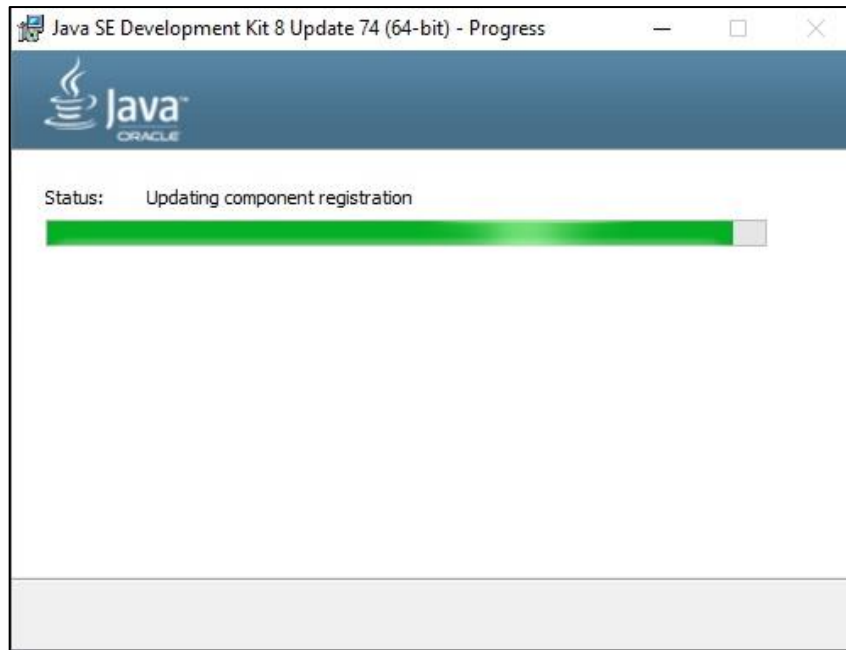
Gambar 1.2 Tampilan awal JDK saat Instalasi

Kemudian, klik **Next**, jika ingin mengubah partisi penyimpanan software jdk, silahkan klik **Change**, jika ingin tersimpan secara default di partisi C langsung klik **Next** saja.



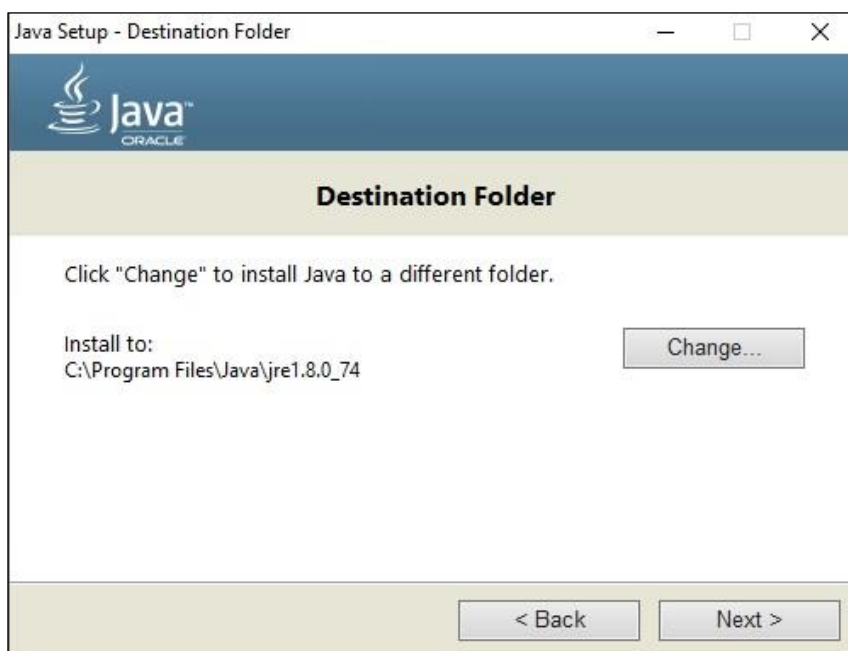
Gambar 1.3 Tampilan Penyimpanan Software JDK

Maka akan tampil proses seperti berikut.



Gambar 1.4 Tampilan Progress JDK

Jika ingin mengubah tempat penyimpanan file jdk, klik **Change** Jika ingin sesuai default, maka pilih **Next**.



Gambar 1.5 Tampilan Destination Folder JDK

Tunggu beberapa saat untuk proses instalasi JDK.



Gambar 1.6 Tampilan Proses Instalasi JDK

Berikut ini menyatakan Java Development Kit (JDK) telah berhasil di install.



Gambar 1.7 Tampilan Selesai Instalasi JDK

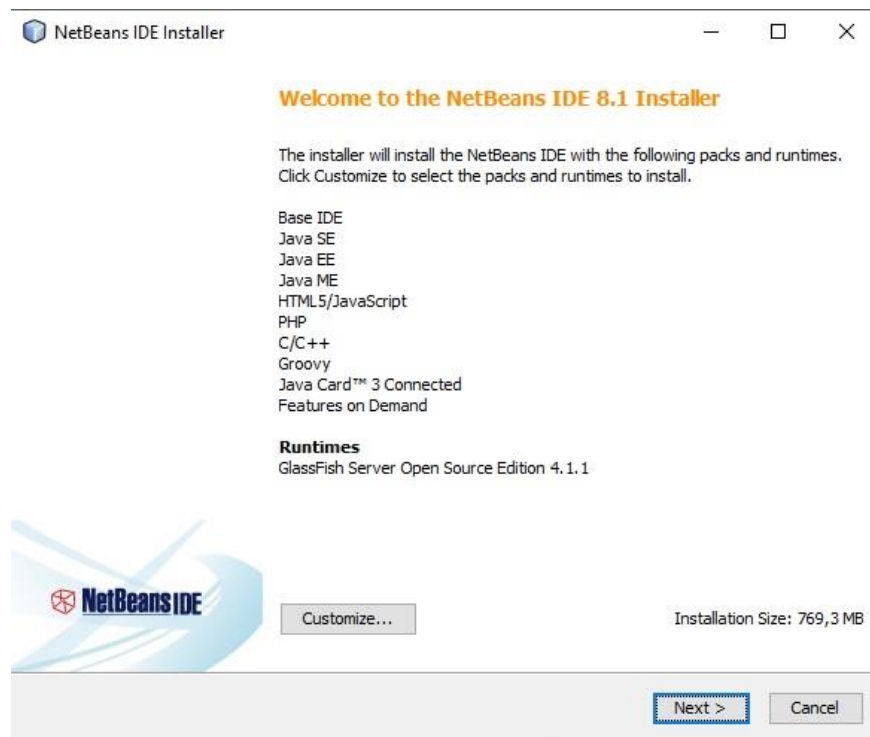
1.2.2 Netbeans

Selanjutnya yang dibutuhkan adalah text editor untuk tempat '*ngoding*'. Netbeans merupakan salah satu text editor untuk penulisan source code program dengan bahasa

Java. Pada umumnya tersedia berbagai macam text editor untuk mempermudah para programmer menuliskan kode programnya. Netbeans Integrated Development Environment (IDE) hadir dengan segala keunggulan fitur yang memungkinkan Anda untuk mengembangkan aplikasi desktop, mobile, dan maupun aplikasi web dengan bahasa Java, C/C++, dan bahkan dynamic languages seperti PHP, JavaScript, Groovy, dan Ruby.

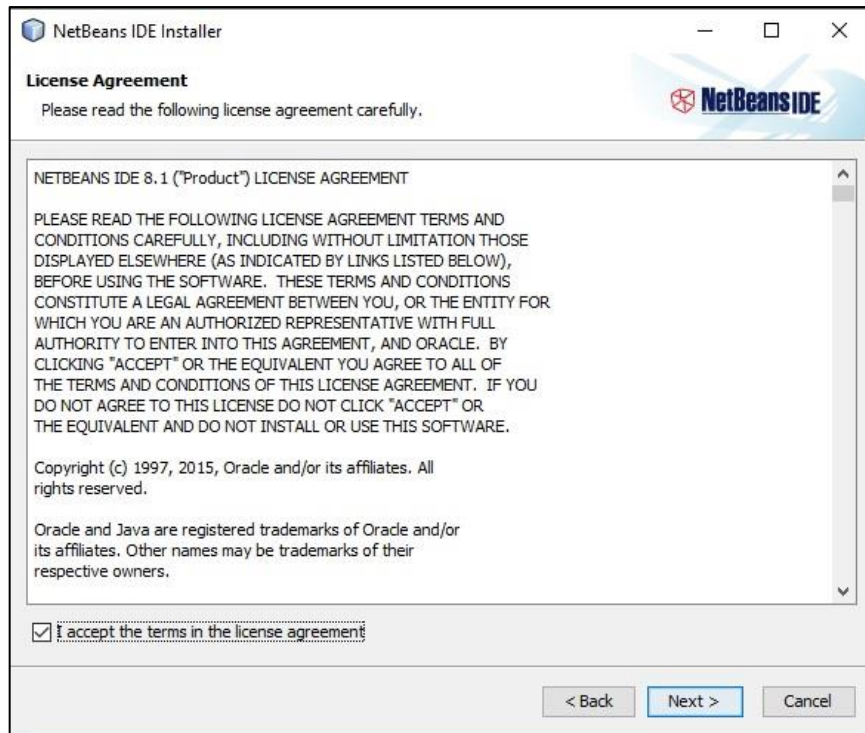
IDE mendukung pengembangan aplikasi dalam berbagai bahasa, termasuk Java, HTML5, PHP dan C++. IDE menyediakan dukungan terintegrasi untuk siklus pengembangan lengkap, mulai dari pembuatan proyek hingga debugging, profiling, dan deploy. IDE berjalan pada Windows, Linux, Mac OS X, dan sistem berbasis UNIX lainnya. Berikut cara instalasi Netbeans 8.1.

Klik 2x file Netbeans 8.1, kemudian akan tampilan awal jendela packaging dari Netbeans 8.1. seperti gambar berikut.



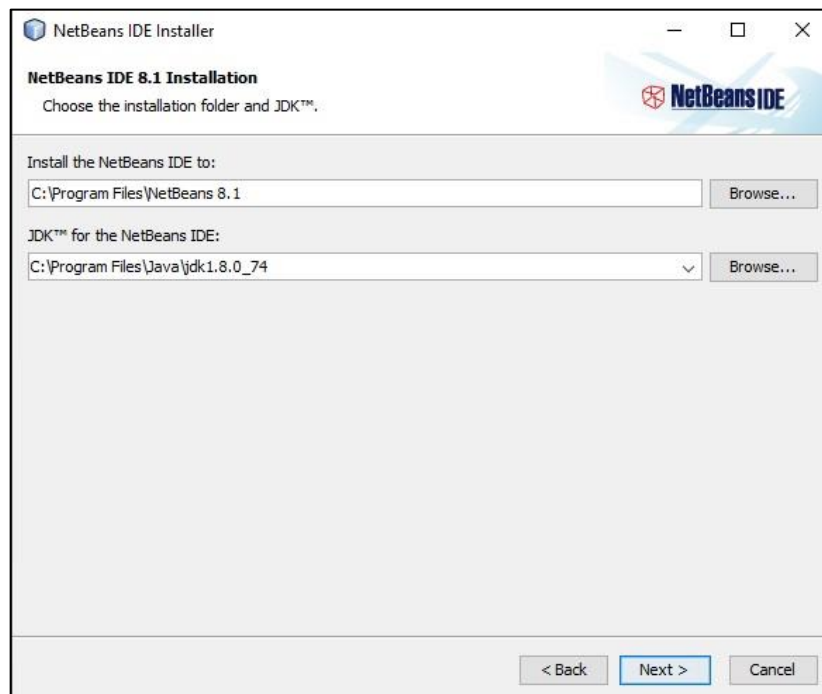
Gambar 1.8 Tampilan Awal Netbeans Installer

Kemudian lanjutkan dengan klik **Next**, maka akan tampil jendela license agreement seperti gambar di bawah ini. Klik centang pada pernyataan **“I accept the terms in the license agreement”** dan klik **Next**.



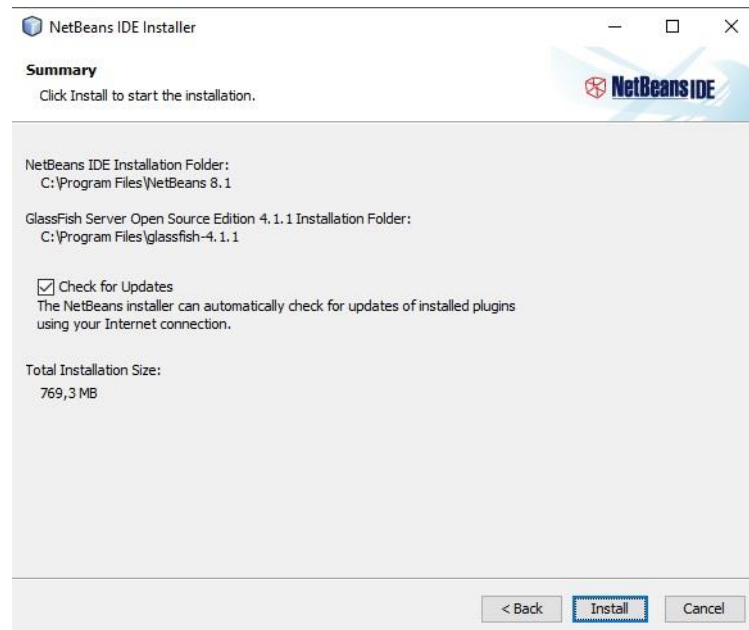
Gambar 1.9 Tampilan License Agreement Netbeans Installer

Setelah itu, akan tampil proses berikutnya yaitu memilih folder untuk penyimpanan file Netbeans dan JDK. Jika sudah sesuai maka dilanjutkan dengan klik **Next**.



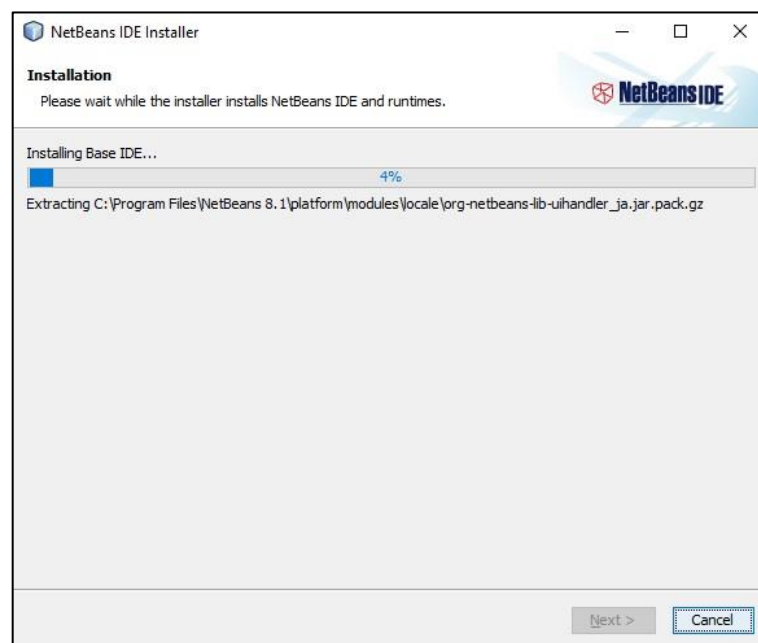
Gambar 1.10 Tampilan Pemilihan Folder Netbeans Installer

Setelah disesuaikan partisi penyimpanan untuk Netbeans Installer dan GlassFish 4.1.1. Installation, lanjutkan dengan klik **Next** dan akan tampil jendela instalasi.



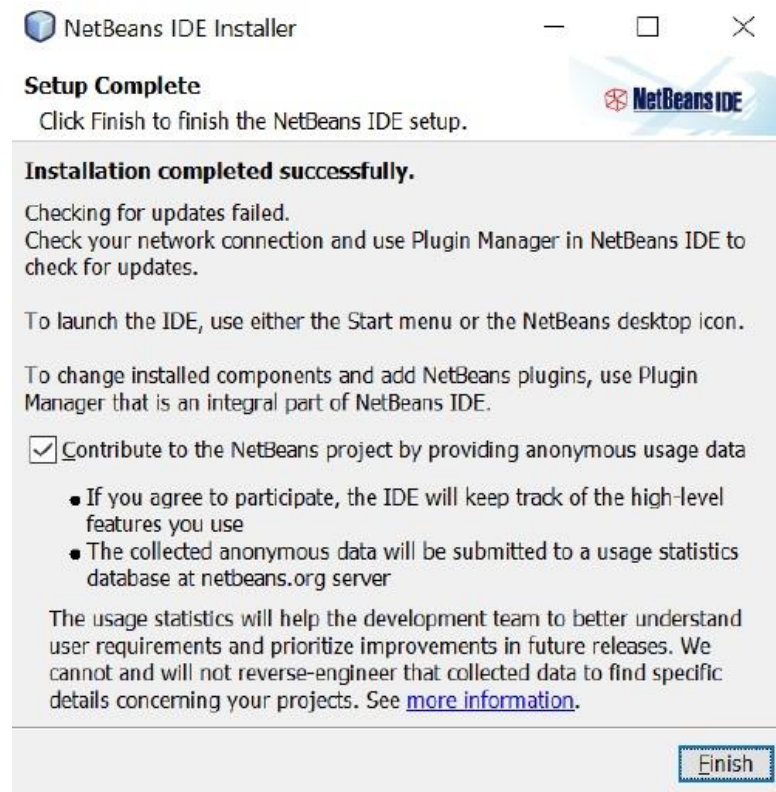
Gambar 1.11 Konfirmasi Penyimpanan File Instalasi Netbeans dan GlassFish

Selanjutnya klik **Install** untuk memulai proses instalasi Netbeans. Tunggu beberapa saat hingga proses selesai.



Gambar 1.12 Proses Instalasi Netbeans dan GlassFish

Tampilan berikut ini menandakan bahwa proses instalasi Netbeans 8.1 telah selesai dan klik **Finish**.

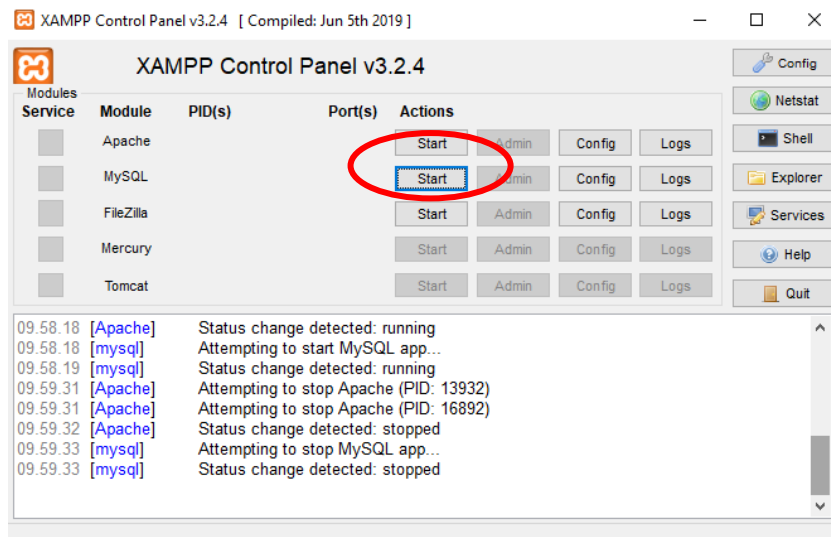


Gambar 1.13 Tampilan selesai Instalasi Netbeans

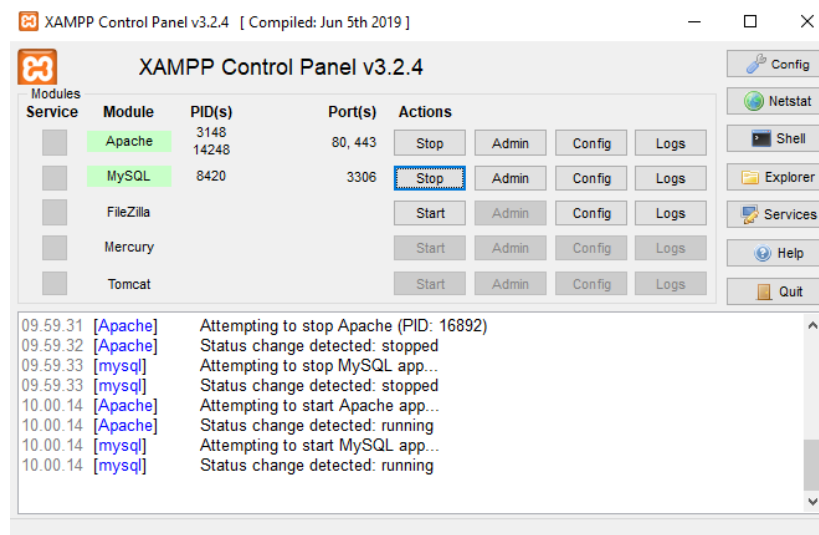
1.2.3 XAMPP

XAMPP adalah web server open source yang berjalan pada sistem operasi cross-platform (Windows, Linux, MacOS). Semua yang diperlukan untuk mengelola website tersedia di XAMPP seperti Apache, MySQL/MariaDB, PHP, dan Perl. XAMPP merupakan solusi bagi Anda yang ingin menjalankan web server dan database di localhost komputer. Jadi bisa memodifikasi website dalam versi offline.

XAMPP berguna untuk menjalankan Apache, MariaDB, dan PHP pada localhost atau komputer tanpa harus ada koneksi internet. Dengan Adanya XAMPP akan membantu Anda dalam mengembangkan, mendesain sehingga mempermudah pekerjaan frontend dan backend developer serta keperluan testing website. Berikut tampilan XAMPP yang sudah di instal.



Gambar 1.14 Tampilan Awal XAMPP

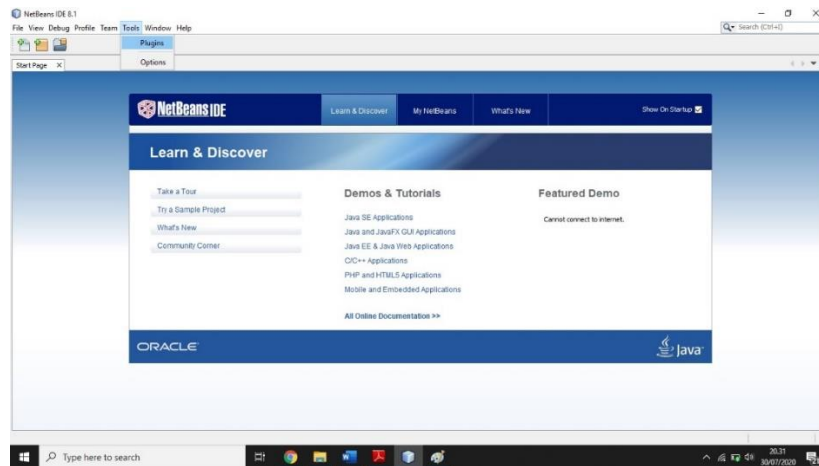


Gambar 1.15 Tampilan Running XAMPP

1.3 Instalasi *Plugin*

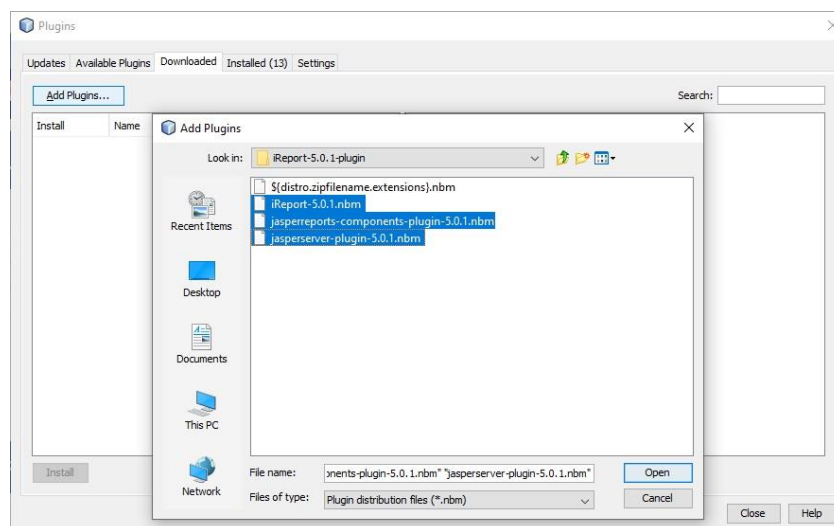
Pada pembuatan project java web, diperlukan plugin yang harus ditambahkan dan diinstal. Salah satu plugin yang akan kita gunakan yaitu iReport-5.5.0. Plugin ini digunakan membuat laporan. iReport adalah desainer laporan visual untuk JasperReports. Anda dapat mengelola bagan, gambar, sublaporan, dan crosstab. Data dapat diambil menggunakan JDBC, TableModels, JavaBeans, XML, Hibernate, CSV dan juga mendukung file dalam bentuk PDF, RTF, XML, XLS, CSV, dan HTM. Berikut cara menginstal iReport-5.5.0 :

Klik pada menu **Tools**, lalu pilih **Plugin**,



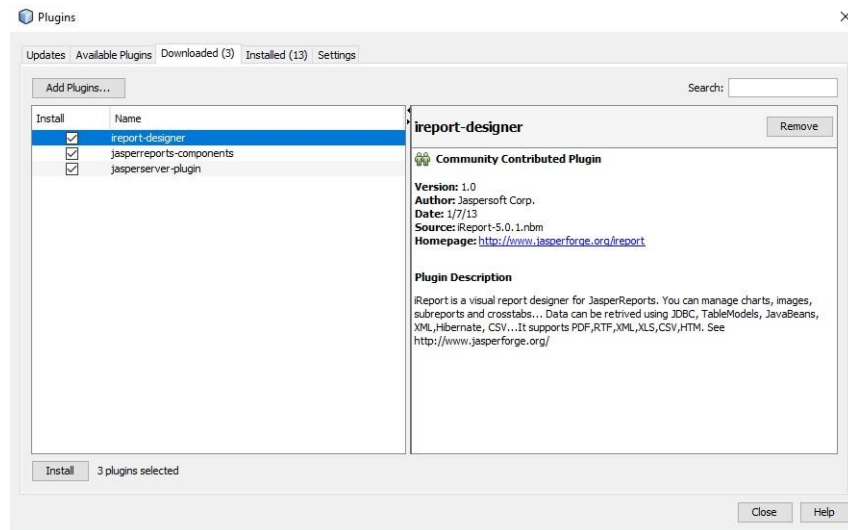
Gambar 1.16 Tampilan Awal Instalasi Plugin iReport

Kemudian muncul jendela Plugin, pilih tab **Downloaded** dan klik **Add Plugin**, maka akan muncul tampilan berikut.



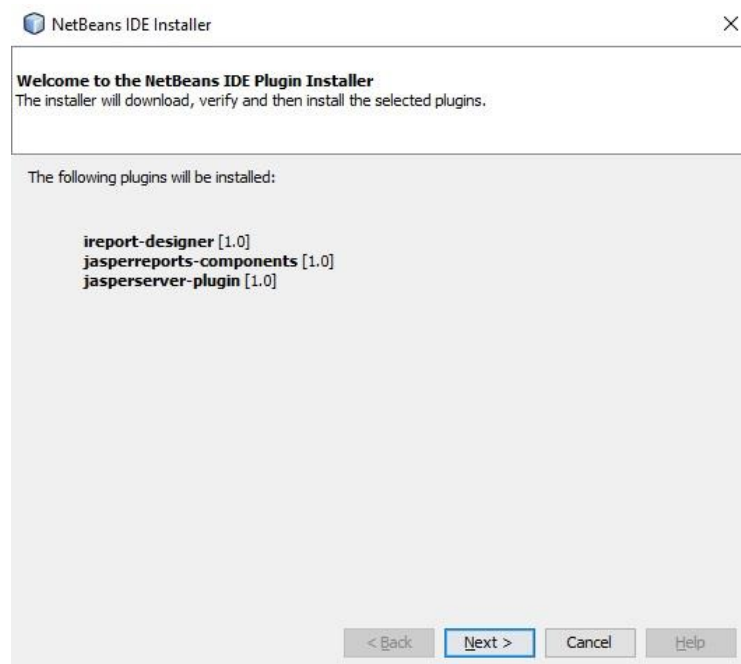
Gambar 1.17 Tampilan Add Plugin iReport

Setelah itu, blok ketiga file untuk plugin iReport dan jasperReport kemudian klik **Open**. Maka akan tampil jendela instalasi dan klik **Install**.



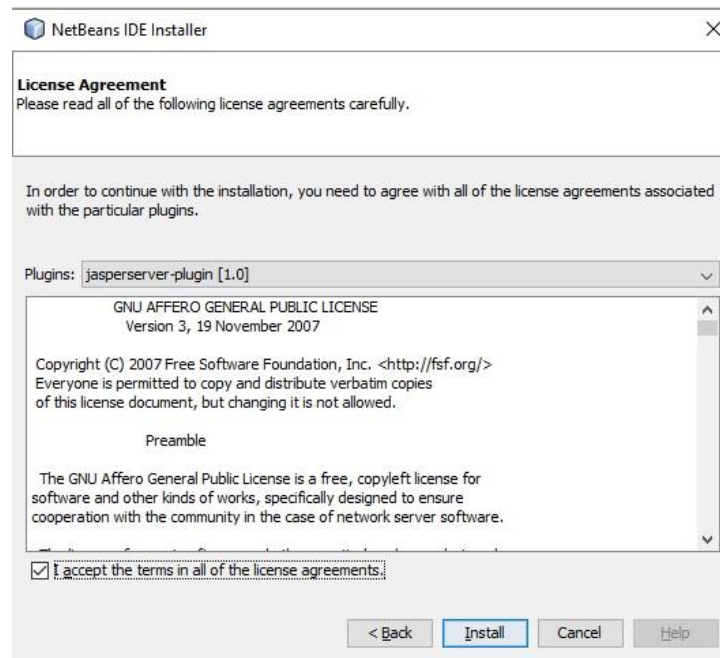
Gambar 1.18 Tampilan Instalasi Plugin iReport

Tunggu hingga selesai untuk proses instalasi Plugin iReport, kemudian klik **Next**.



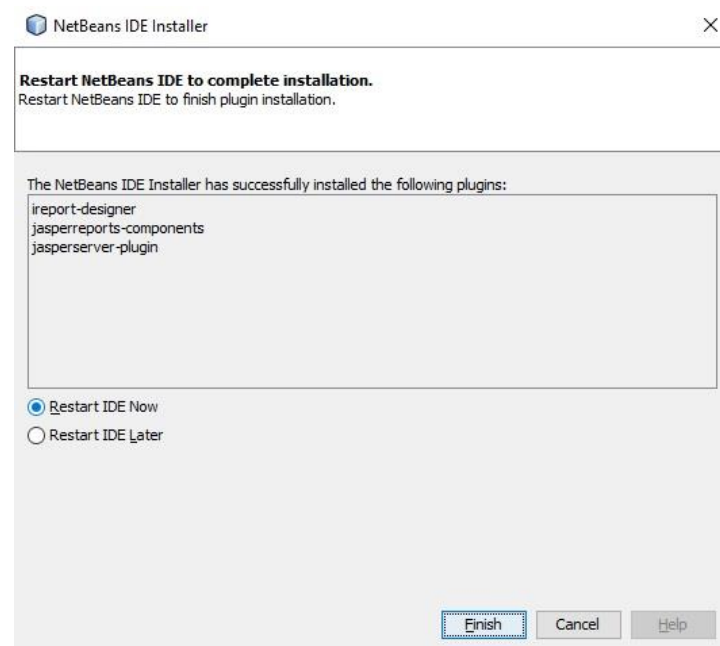
Gambar 1.19 Tampilan Proses Instalasi Plugin

Kemudian centang pada pernyataan **I Accept the terms in all of the license agreements** untuk lisensi agreement plugin. Lalu klik **Install** maka akan muncul jendela berikutnya.



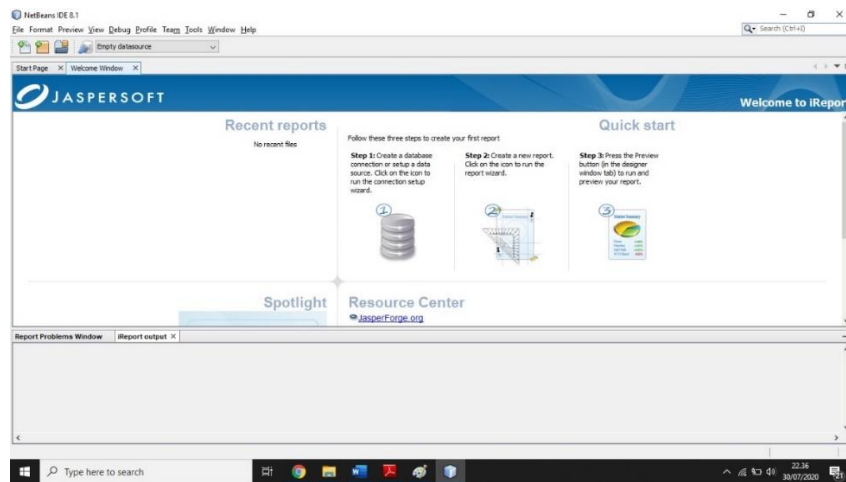
Gambar 1.20 Tampilan Persetujuan Lisensi Plugin

Kemudian klik **Continue** untuk melanjutkan tahapan instalasi. Setelah proses instalasi selesai maka akan tampil seperti berikut, pilih button **Restart IDE Now** dan klik **Finish**.



Gambar 1.21 Tampilan Instalasi Plugin Selesai

Setelah itu Netbeans 8.1 akan restart dan akan menampilkan tampilan seperti gambar dibawah ini.



Gambar 1.22 Tampilan Netbeans Setelah Install Plugin iReport

BAB II

PENGENALAN HTML

Deskripsi:

Bab ini membahas pengenalan tentang bahasa pemrograman HTML dan membuat form sederhana dengan HTML.

Tujuan Pembelajaran:

Setelah melakukan pembelajaran ini, pembaca diharapkan mampu:

4. Mengenal Bahasa Pemrograman HTML beserta fungsi tag-tag di dalamnya.
5. Membuat tabel sederhana dengan HTML.

2.1. Sejarah HTML

Hypertext Markup Language atau HTML ini merupakan salah satu bahasa pemrograman dasar dalam dunia programming. Bahasa HTML ini bisa dikatakan wajib dikuasai oleh seorang programmer karena digunakan sebagian besar situs web. HTML merupakan bahasa yang mendeskripsikan struktur dan nilai semantik konten dari dokumen Web. Lebih tepatnya, HTML ini digunakan untuk membuat halaman dan membuatnya berfungsi. Nah, kita kulik sedikit mengenai sejarah munculnya bahasa pemrograman yang satu ini.

HTML pertama kali dibuat oleh Tim Berners-Lee, Robert Cailliau, dan lainnya mulai tahun 1989. Hypertext yang menyatakan bahwa dokumen yang berisi link memungkinkan pembaca untuk berpindah ke tempat lain dalam dokumen atau ke dokumen lain secara bersamaan. Sedangkan bahasa Markup adalah cara komputer berbicara satu sama lain untuk mengontrol bagaimana teks diproses dan disajikan. Untuk melakukan ini HTML menggunakan dua hal, yakni tag dan atribut. Kedua hal ini menjadi dasar dari pembelajaran HTML.

Tag digunakan untuk menandai awal elemen HTML dan diapit oleh tanda kurung siku. Contoh dari sebuah tag adalah: `<tr>`. Setiap tag biasanya ada tag pembuka `<tr>` dan tag penutup yang dibedakan dengan tanda garis miring atau slash “/” seperti `</tr>` agar dapat berfungsi dengan sebagaimana mestinya.

Sedangkan atribut berupa tag pembuka dan berisi informasi tambahan yang ditempatkan di dalamnya. Contoh atribut seperti di bawah ini:

```
<tr width="1000" height="250">
```

Yang dikatakan info tambahan dalam contoh tersebut adalah Tag `<tr>` memiliki atribut “width” dan “height”.

2.2. Struktur HTML

Ketika akan memulai membuat sebuah website dengan Bahasa HTML, bentuk umum dari HTML tidak terlepas dari struktur dasar tag-tag berikut.

```
<html>
<head>
  <title></title>
</head>
<body>

</body>
</html>
```

Tag-tag di atas merupakan awalan ketika ingin menuliskan kode program pada HTML. Perhatikan setiap tag pembuka maka biasanya akan ada tag penutup. Berikut penjelasan detail dari jabaran tag di atas.

1. Tag pembuka `<html>` pada baris pertama dan tag penutup `</html>` pada baris terakhir. Tag ini menandakan bahwa kita akan mulai menulis dalam kode HTML.
2. Tag `<head> ... </head>` merupakan tag kepala web. Tag kepala ini akan terlebih dulu dieksekusi sebelum tag `<body>`. Di dalam tag ini berisi tag `<meta>` dan `<title>`. Tag `<meta>` merupakan informasi atau header suatu dokumen HTML. Di sinilah informasi tentang dokumen disimpan berupa pengkodean karakter, nama (konteks halaman), deskripsi. Atribut yang dimiliki oleh tag ini antara lain:
 - a. **HTTP_EQUIV**, berfungsi untuk menampilkan dokumen HTML secara otomatis dalam jangka waktu tertentu.
 - b. **CONTENT**, atribut ini berisi informasi tentang isi document HTML yang akan dipanggil.

- c. **NAME**, merupakan identifikasi dari meta itu sendiri. Tag `<meta>` dalam suatu dokumen HTML statusnya opsional boleh dipakai atau tidak sama sekali.
3. Tag `<title> ... </title>` adalah tag judul. Pada tag inilah tempat untuk memasukkan nama halaman karena akan muncul di bagian atas jendela browser atau tab. Setiap halaman web memiliki judul dituliskan di dalam `<title> ... </title>`.
 4. Tag `<body> ... </body>` disebut juga tag badan web yang berisi konten dari suatu halaman web. Pada tag ini lah tempat menambahkan konten web yang dirancang agar terlihat secara visual. Berikut tampilan struktur halaman HTML.



Source: <https://html.com>

Gambar 2.1 Tampilan Struktur Halaman HTML

Saat menggunakan beberapa tag, tag harus ditutup sesuai urutan pembukaannya. Sebagai contoh lainnya:

```
<tr><td>Isikan Nama Lengkap Anda</td></tr>
```

Yang dimaksud harus berurutan adalah seperti contoh di atas diawali dengan tag buka `<tr>` dan diakhiri dengan tag tutup `</tr>` pula, sementara tag buka `<td>` dan tag tutup `</td>` berada di dalamnya.

2.3. Memilh Text Editor

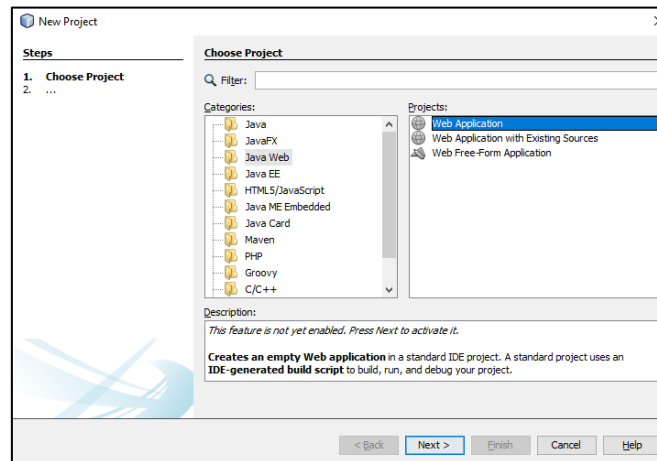
Bagian terpenting juga adalah terletak pada pemilihan text editor yang tepat dan membuat kita sebagai programmer nyaman untuk menggunakannya. Ada banyak pilihan untuk text editor dalam programming, jadi bebas saja mau menggunakan text editor yang mana. Diantaranya ada Sublime Text, Notepad++, Notepad, Komodo Edit, dan masih banyak lagi. Namun, untuk pembelajaran pada buku ini adalah mengenai Java Web, maka kita langsung

saja menggunakan Netbeans. Di awal pembahasan sudah kita tentukan untuk menggunakan Netbeans untuk menuliskan kode program atau source code.

2.4. Penggunaan Script HTML pada Netbeans

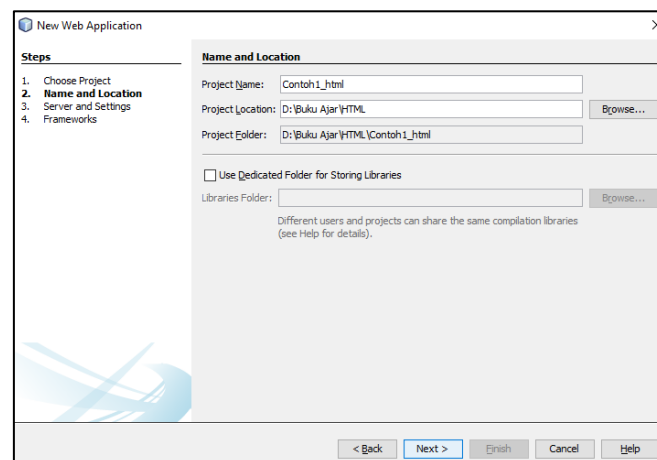
Berikut langkah-langkah dalam menuliskan kode program html pada Netbeans IDE 8.1.

1. Buka aplikasi Netbeans yang telah di install, kemudian pilih menu **File >> New Project**, maka akan tampil jendela berikut.



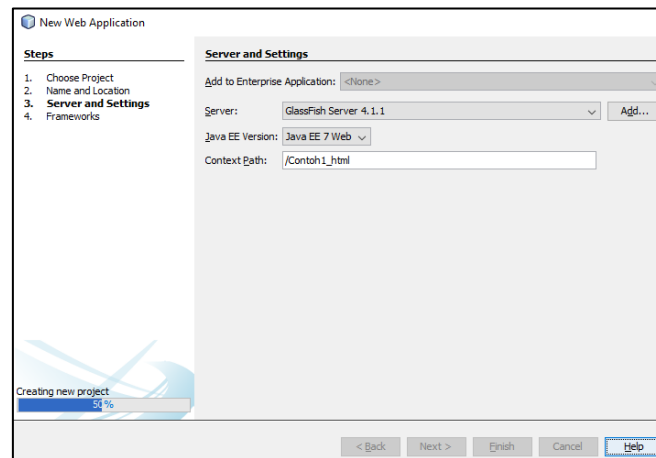
Gambar 2.2 Tampilan Membuat Project Baru pada Netbeans

2. Pilih pada bagian Categories dan Projects, dengan memilih folder **Java Web** dan Projects **Web Application**. Setelah itu klik **Next**.
3. Selanjutnya tentukan nama project dan penyimpanan lokasi project seperti gambar berikut.



Gambar 2.3 Tampilan Penyimpanan Nama dan Lokasi Project

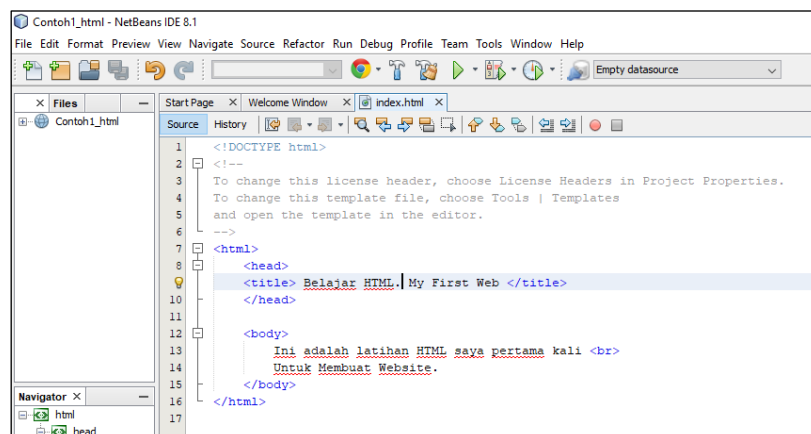
4. Setelah klik **Next**, lanjutkan dengan klik **Finish**. Silahkan tunggu beberapa saat proses pembuatan project baru selesai.




Gambar 2.4 Tampilan Server dan Settings

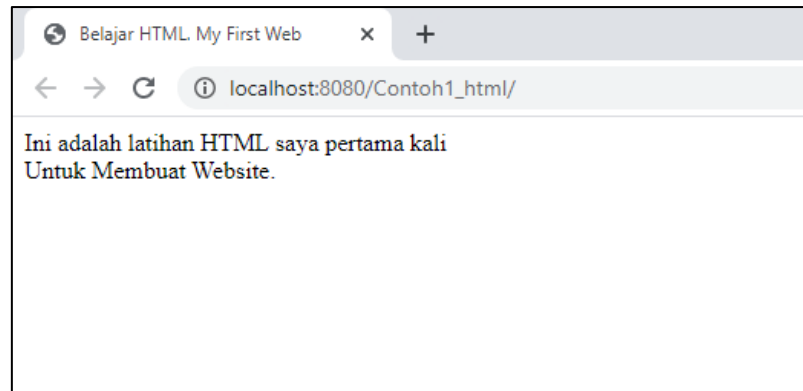
5. Tag <head>

Berikut kita akan mencoba bagian dasar dari penggunaan tag <head>. Tuliskan script HTML berikut pada **index.html**.



Gambar 2.5 Script HTML File index.html

Jalankan script di atas dengan klik **Run Project** atau icon  maka akan muncul pada browser Anda masing-masing seperti gambar di bawah ini:



Gambar 2.6 Tampilan Halaman Web index.html

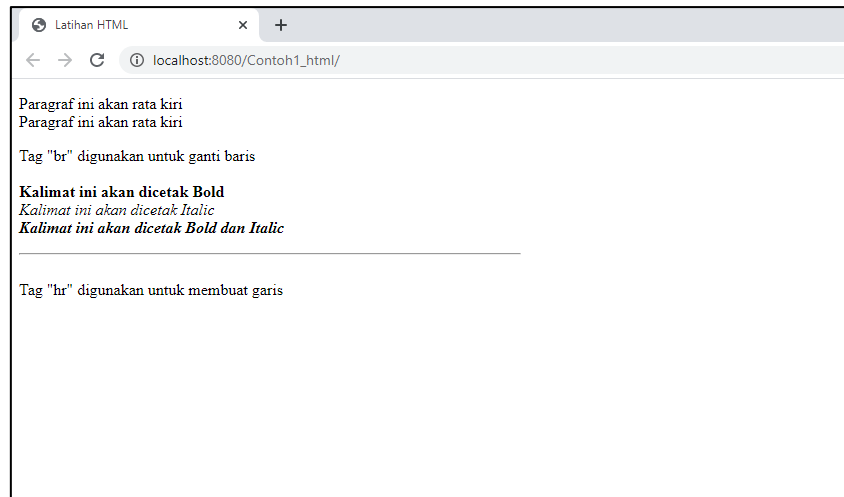
6. Tag <body>

Pada tag <body> ini sebagai tempat Anda menambahkan konten yang dirancang untuk dilihat oleh manusia secara visual. Termasuk di dalamnya akan tampil teks, gambar, tabel, formulir, dan segala sesuatu yang biasanya kita lihat di internet setiap hari. Berikut contoh penggunaan tag <body> yang di dalamnya berisi perintah-perintah tag dasar.

```
<html>
<head>
  <title> Latihan HTML </title>
</head>

<body>
  <p align="left">
    Paragraf ini akan rata kiri<br>
    Paragraf ini akan rata kiri
  <p>
    Tag "br" digunakan untuk ganti baris <br><br>
    <b>Kalimat ini akan dicetak Bold </b> <br>
    <i>Kalimat ini akan dicetak Italic </i> <br>
    <b><i>Kalimat ini akan dicetak Bold dan Italic</i></b>
  <hr width="500" align="left"><br>
  Tag "hr" digunakan untuk membuat garis
</body>
</html>
```

Berikut hasil running script di atas.



Gambar 2.7 Tampilan Running Tag <body>

7. Tag Heading

Dalam HTML, Tag Heading biasanya digunakan untuk penulisan judul atau label pada konten sebuah website, ditulis dalam elemen berikut:

- <h1>
- <h2>
- <h3>
- <h4>
- <h5>
- <h6>

Untuk tag <h1> dan <h2> harus digunakan untuk judul, sedangkan tag lainnya harus digunakan untuk sub-judul dan isi teks. Tag <h1> yang paling besar ukuran font nya, sementara <h6> ukuran font yang paling kecil dalam Heading. Contohnya silahkan dicoba script berikut.

```
<html>
<head>
  <title> Latihan HTML </title>
</head>

<body>
<h1>Selamat Datang di Website Portal Berita Nasional</h1>
<h2>Selamat Datang di Website Portal Berita Nasional</h2>
<h3>Selamat Datang di Website Portal Berita Nasional</h3>
<h4>Selamat Datang di Website Portal Berita Nasional</h4>
<h5>Selamat Datang di Website Portal Berita Nasional</h5>
<h6>Selamat Datang di Website Portal Berita Nasional</h6>
```

```
</body>
</html>
```

Berikut tampilan penggunaan tag heading setelah dijalankan.



Gambar 2.8 Tampilan Tag Heading

8. Tag Style

Berikut contoh penggunaan tag Style.

```
<html>
<body>

<h1 style="text-align:center">Heading di Tengah</h1>
<p style="text-align:center">Paragraf di Tengah.</p>

</body>
</html>
```

9. Tag List

Dalam HTML, tag list terdiri dari 2 jenis, ordered list (berurutan) dan unordered list (tidak berurutan). Berikut penjelasan mengenai tag list beserta fungsinya.

No.	Nama Tag	Fungsi
1		Untuk membuat tampilan daftar menggunakan simbol seperti bullet, cycle, dan lainnya.
2		Membuat daftar menggunakan nomor, huruf, romawi, dan lainnya.
3		Menentukan berbagai item yang ingin ditampilkan

4	<dl>	Mendefinisikan daftar deskripsi
5	<dt>	Mendefinisikan istilah deskripsi
6	<dd>	Menggambarkan istilah dalam daftar deskripsi
7	<type>	Menentukan jenis penomoran

Ordered List

Ordered list atau ditulis tag adalah daftar yang diurutkan dan ditampilkan dengan angka atau huruf.

Di dalam tag kami mencantumkan setiap item pada daftar yang ada di dalam tag . Berikut contoh script nya.

```
<ol>
  <li>Nasi Goreng</li>
  <li>Mie Ayam</li>
  <li>Bakso Urat</li>
  <li>Kwe Tiaw</li>
  <li>Sate Ayam/Kambing</li>
</ol>
```

Unordered List

Unordered List atau ditulis tag adalah daftar yang tidak berurutan. Tag ini lebih dikenal sebagai daftar bullet point dan bukan berupa angka. Berikut contoh script nya.

```
<ul>
  <li>Aneka Soft Drink</li>
  <li>Aneka Jus</li>
  <li>Es Teh</li>
  <li>Es Timun Serut</li>
  <li>Es Lidah Buaya</li>
</ul>
```

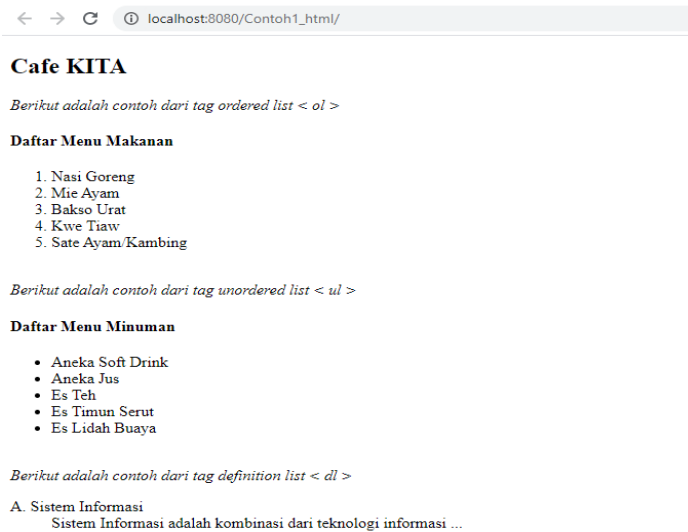
Definition List

Definition List ditandai dengan dengan tag <dl> yang berfungsi untuk mendefinisikan daftar deskripsi yakni tag <dt> dan menggambarkan istilah dalam daftar deskripsi yang ditandai dengan tag <dd>. Berikut contoh scriptnya.

```
<dt>A. Sistem Informasi</dt>
  <dd> Sistem Informasi adalah kombinasi dari teknologi informasi... </dd>
```

</dt>

Berikut hasil running program ketiga script tag list di atas jika digabungkan.



Gambar 2.9 Tampilan Tag List

10. Membuat Link pada HTML

Sering kita jumpai pada halaman-halaman website terdapat link. Internet sendiri tidak lepas dari adanya link. Fungsi link sendiri yaitu membawa Anda ke halaman lain dalam situs web yang Anda kunjungi atau ke situs eksternal. Link disertakan dalam atribut yang dibuka oleh tag <a> atau yang disebut anchor tag. Contohnya seperti di bawah ini.

```
<a href="index.html"> Kembali ke halaman Beranda </a>
```

Atribut href pada tag <a> menunjuk ke halaman yang akan terbuka setelah link yang bertuliskan “Kembali ke halaman Beranda” di klik.

11. Menambah Gambar

Di era yang serba digital ini, kebutuhan akan gambar dan video menjadi hal yang sangat penting dalam sebuah website. Hampir semua website yang telah kita jelajahi menampilkan gambar-gambar maupun video sehingga bisa dilihat dan dinikmati oleh user. Berikut tag-tag gambar beserta atribut dan fungsinya.

No.	Nama Tag	Fungsi
1		Elemen untuk mendefinisikan gambar
2	src	Atribut untuk menentukan URL gambar
3	alt	Mendefinisikan teks pada gambar, jika gambar tidak dapat ditampilkan
4	width – height	Menentukan ukuran gambar
5	float	Properti untuk float image pada CSS
6	<map>	Mendefinisikan gambar peta
7	<area>	Mendefinisikan area atau daerah-daerah gambar pada peta
8	<picture>	Menampilkan gambar yang berbeda untuk perangkat yang berbeda

Jenis file yang umumnya digunakan untuk file gambar online adalah: .jpg, .png, dan (less and less) .gif. Tag ditulis seperti contoh berikut.

```

```

Pada contoh script tag di atas, menggunakan atribut src, alt, width dan height. Berikut adalah elementor-elementor lain pada HTML.

No	Tag	Fungsi
1		Membuat teks tebal
2		Membuat teks penting
3	<i>	Membuat teks miring
4		Membuat teks ditekankan
5	<mark>	Membuat teks ditandai
6	<small>	Membuat teks kecil
7		Teks dihapus
8	<ins>	Teks dimasukkan
9	<sub>	Teks subscript
10	<sup>	Teks superscript
11	<p>	Membuat paragraf

12	 	Membuat garis baru
13	<pre>	Memformat teks atau kalimat
14	<hr>	Memisahkan konten (biasanya ditampilkan garis pembatas)
15	<strike>	Untuk menempatkan garis horizontal di seluruh teks
16	<u>	Digunakan untuk link atau highlight pada teks yang diberi garis bawah (underline)

12. Tag <form>

Tag <form> berguna untuk mengirimkan atau istilah lain *mem-post* variabel yang bersisi data ke file tujuan yang memproses variabel tersebut. Atribut form yang banyak digunakan adalah method dan action. Atribut method untuk menentukan jenis pengiriman data bisa berupa “POST” atau “GET”. Sedangkan action adalah file tujuan yang akan memproses data yang ada diantara tag <form>...</form>.

Method POST akan mengirimkan data atau nilai langsung ke action untuk ditampung, tanpa menampilkan pada URL. Sedangkan method GET akan menampilkan data/nilai pada URL, kemudian akan ditampung oleh action. Method POST data yang dikirim tidak terbatas. Sedangkan method GET tidak boleh lebih dari 2047 karakter.

Berikut adalah elementor yang berkaitan dengan tag <form>.

No.	Tag	Fungsi
1	<form>	Membuat formulir untuk mengumpulkan input pengguna
2	<input>	Membuat tipe inputan pada form yang akan dibuat
3	<textarea>	Elemen untuk mendefinisikan field input
4	<label>	Memberikan label pada elemen input
5	<fieldset>	Mengelompokkan elemen yang terdapat pada sebuah form
6	<select>	Membuat input dengan pilihan yang berbentuk list drop down
7	<optgroup>	Mengelompokkan beberapa pilihan pada daftar pilihan input
8	<option>	Mendefinisikan opsi yang bisa dipilih
9	<button>	Membuat Button
10	<datalist>	Membuat daftar pilihan untuk input data

11	<output>	Menampilkan hasil dari hitungan
----	----------	---------------------------------

Berikut contoh script penggunaan tag <form> dengan menggunakan method post.

```
<form action="coba.php" method="post"> </form>
```

2.5. Membuat Tabel

Tabel merupakan bagian paling penting dalam HTML, karena sering digunakan untuk merepresentasikan data atau daftar yang digambarkan dengan tabel. Pembuatan sebuah tabel tidak terlepas dari istilah baris dan kolom yang dibuat dengan tag <tr> dan <td>. Sebuah tabel dibuat berawal dari pembuatan baris terlebih dahulu kemudian diikuti dengan pembuatan kolom.

Berikut adalah tag-tag dan atribut yang berhubungan dengan pembuatan tabel.

No	Nama Tag	Fungsi
1	<table>	Membuat tabel pada web
2	<tr>	Membuat baris pada tabel
3	<td>	Membuat kolom pada tabel
4	<th>	Membuat judul pada kolom. Contohnya nama, kelas, dan alamat.
5	<caption>	Membuat judul tabel
6	border	Mengatur garis tabel
7	border-collapse	Mengatur batas garis tabel
8	padding	Mengatur padding pada cell
9	text-align	Mengatur perataan pada konten tabel
10	border-spacing	Mengatur jarak spasi garis tabel
11	colspan	Menggabungkan beberapa kolom. Kalau di office disebutnya Merge Cell
12	rowspan	Menggabungkan beberapa baris
13	id	Memberikan id pada tabel atau kolom

Contoh pembuatan tabel dengan script sebagai berikut.

Latihan 1.

```

<html>
<head>
  <title> Belajar Membuat Tabel </title>
</head>
<body>
<h1 align="center" >Data Pelanggan</h1>
<table border="1" align="center">
  <tr bgcolor="pink" height="40" align="center">
    <td>No.</td>
    <td>Nama Pelanggan</td>
    <td>Jenis Kelamin</td>
    <td>Alamat</td>
    <td>No.Hp</td>
  </tr>
  <tr>
    <td>1.</td>
    <td>Deni Ridwan</td>
    <td>Laki-laki</td>
    <td>Jeruju, Pontianak</td>
    <td>089637994002</td>
  </tr>
  <tr>
    <td>2.</td>
    <td>Agus Priyanto</td>
    <td>Laki-laki</td>
    <td>Kota Baru, Pontianak</td>
    <td>089537894002</td>
  </tr>
  <tr>
    <td>3.</td>
    <td>Fajrudin</td>
    <td>Laki-laki</td>
    <td>Siantan Hulu, Pontianak</td>
    <td>085237993002</td>
  </tr>
  <tr>
    <td>4.</td>
    <td>Rina Mirwan</td>
    <td>Perempuan</td>
    <td>Jeruju Pontianak</td>
    <td>082137993992</td>
  </tr>
  <tr>
    <td>5.</td>
    <td>Santi Ningrum</td>
    <td>Perempuan</td>
    <td>Adi Sucipto, Pontianak</td>
    <td>082139096002</td>
  </tr> <br>
</table>
<br>
<center><a href="index.html"> Kembali ke halaman Beranda</a></center>
</body>
</html>

```

Berikut hasil script di atas ketika dijalankan (running).

Data Pelanggan				
No.	Nama Pelanggan	Jenis Kelamin	Alamat	No.Hp
1.	Deni Ridwan	Laki-laki	Jeruju, Pontianak	089637994002
2.	Agus Priyanto	Laki-laki	Kota Baru, Pontianak	089537894002
3.	Fajrudin	Laki-laki	Siantan Hulu, Pontianak	085237993002
4.	Rina Mirwan	Perempuan	Jeruju Pontianak	082137993992
5.	Santi Ningrum	Perempuan	Adi Sucipto, Pontianak	082139096002

[Kembali ke halaman Beranda](#)

Gambar 2.10 Tampilan Pembuatan Tabel Latihan 1

Latihan 2.

```
<html>
<head>
  <title> Belajar Membuat Form</title>
</head>
<body>
<h1> Daftar Menu CAFE KITA</h1>
<table border="0" align="left">
  <tr bgcolor="pink" height="40" align="center">
    <td colspan="5"><b>PILIHAN MENU</b></td>
  </tr>
  <tr align="center">
    <td colspan="2"><b>Makanan</b></td>
    <td width="30"></td>
    <td colspan="2"><b>Minuman</b></td>
  </tr>
  <tr>
    <td>Nasi Goreng</td>
    <td>Rp 15.000</td>
    <td></td>
    <td>Es Teh</td>
    <td>Rp 5.000</td>
  </tr>
  <tr>
    <td>Mie Tiaw</td>
    <td>Rp 18.000</td>
    <td></td>
    <td>Es Jeruk</td>
    <td>Rp 8.000</td>
  </tr>
  <tr>
    <td>Bakso Urat</td>
    <td>Rp 15.000</td>
    <td></td>
    <td>Es Serut Timun</td>
    <td>Rp 8.000</td>
```

```

        </tr>
        <tr>
        <td>Mie Ayam</td>
        <td>Rp 16.000</td>
        <td></td>
        <td>Es Limau</td>
        <td>Rp 5.000</td>
        </tr>
        <tr>
        <td>Ayam Geprek</td>
        <td>Rp 12.000</td>
        <td></td>
        <td>Air Mineral</td>
        <td>Rp 4.000</td>
        </tr>
        <tr>
        <td colspan="5"></td>
        </tr>
        <tr>
        <td colspan="5" align="center"><input type="submit" value="PESAN
SEKARANG"></td>
        </tr>
</table>
</body>
</html>

```



Gambar 2.11 Tampilan Pembuatan Tabel Latihan 2

BAB III

PEMROGRAMAN JAVA SERVER PAGE (JSP)

Deskripsi:

Bab ini membahas pengenalan tentang Pemrograman Java Server Page (JSP) dan membuat JSP Form.

Tujuan Pembelajaran:

Setelah melakukan pembelajaran ini, pembaca diharapkan mampu:

3. Mengenali bahasa pemrograman Java Server Page (JSP).
4. Menerapkan penggunaan variabel dan operator ke dalam bahasa JSP.
5. Membuat JSP Form dengan konsep percabangan dan perulangan.

3.1. Apa itu Java Server Page?

Java Server Page (JSP) merupakan aplikasi web yang umumnya diproses penampilannya di klien. JSP umumnya memanggil suatu kelas yang disebut bean, yang diproses di web server. Tujuan dari Java Server Page adalah untuk mempermudah pembuatan dan manajemen halaman web dinamis dengan memisahkan logika bisnis dengan presentasi atau tampilan. Dokumen JSP terdiri dari tag html/xml standar dan tag script JSP. Perbedaan antara JSP dari servlet adalah:

1. Servlet dieksekusi di server dan mengembalikan permintaan kepada browser.
2. JSP memisahkan logika dan presentasi di aplikasi web serta memisahkan isi web dinamis dan statis.
3. Servlet perlu untuk menggabungkan kode dengan isi statis, sedangkan JSP dapat menggunakan bean dengan cakupan tertentu untuk memisahkan kode atau extension tag.

JSP merupakan solusi dari servlet. Teknologi JSP sebagai perluasan dari teknologi servlet untuk membangun aplikasi web dengan menyatukan kode java ke dalam halaman HTML, sehingga lebih mudah dan cepat dalam membuat aplikasi berbasis web. Dengan konsep JSP ini dapat membuat bagian static yang berupa tag-tag HTML akan terpisah dari kode JSP. Jadi kita membuat halaman web static dengan HTML di web editor, kemudian disisipkan

dengan kode JSP untuk membuat halaman web dinamis. Umumnya kode JSP di dalam kode HTML menggunakan simbol `<%=...%>`.

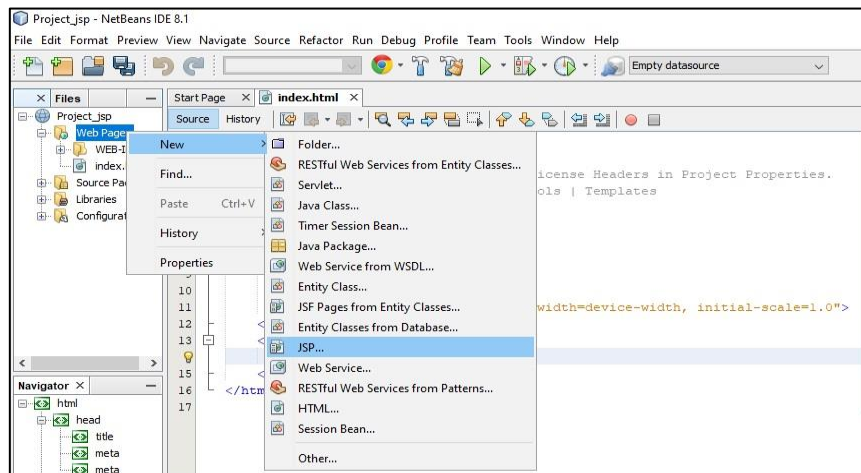
Beberapa alasan JSP banyak digunakan oleh para web developer diantaranya JSP menggunakan bahasa Java yang memiliki kemampuan OOP dan tingkat reuseability tinggi, sehingga tidak perlu mempelajari bahasa pemrograman lain untuk membuat aplikasi web. Selain itu, JSP juga bersifat multiplatform, artinya dapat dijalankan pada berbagai container servlet yang kompatibel dengan mengabaikan sistem operasinya.

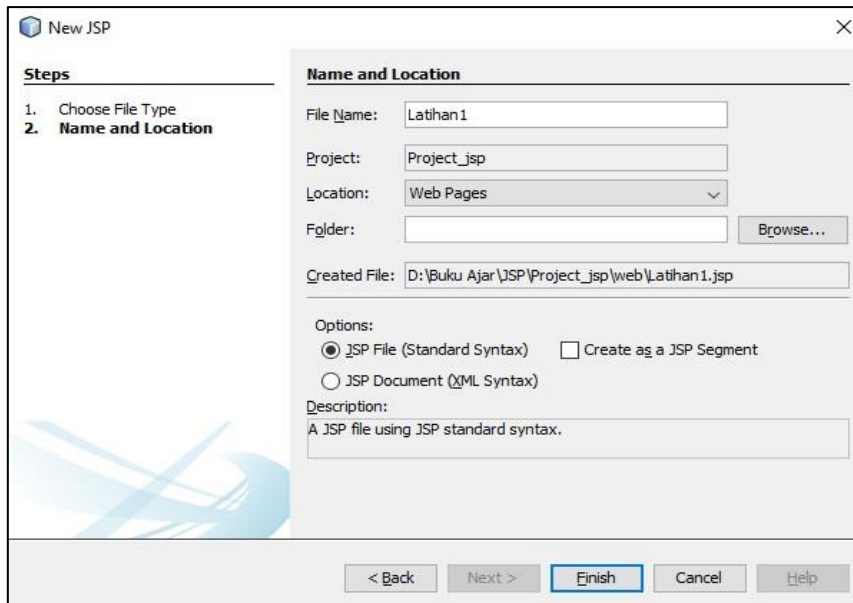
3.2. JSP Pada Netbeans

Untuk memulai membuat project dengan JSP, Anda memerlukan beberapa tools pendukung yang telah disampaikan pada chapter awal. Salah satunya adalah text editor Netbeans IDE 8.1 ini. Berikut cara membuat JSP pada Netbeans.

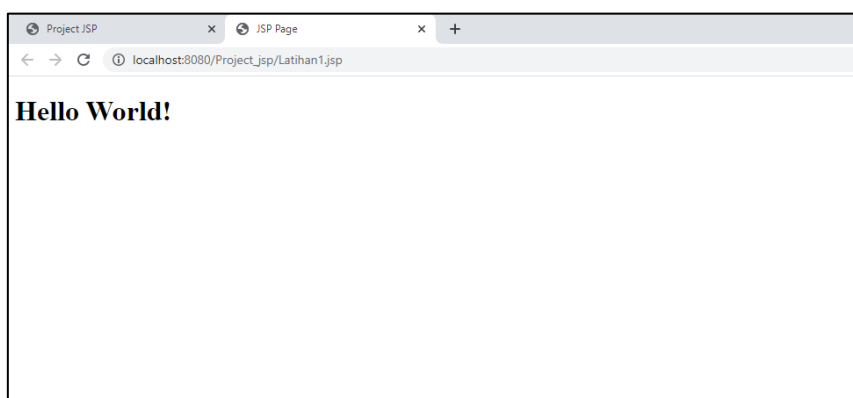
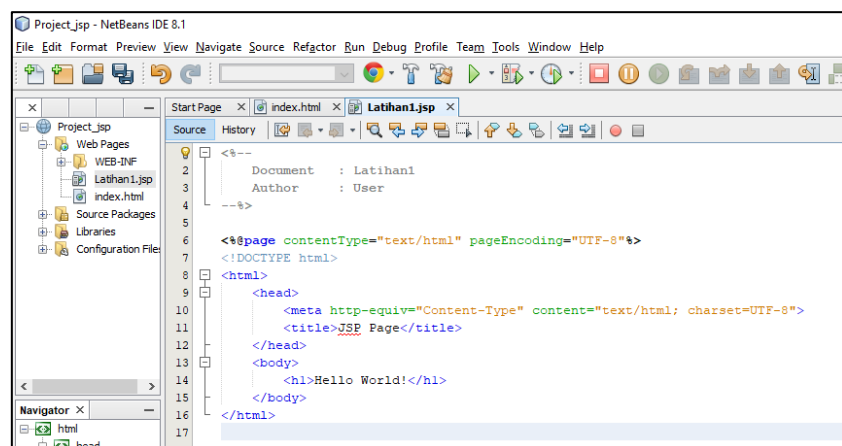
1. Membuat Project Baru

Buka text editor Netbeans 8.1, kemudian pilih menu **File >> New Project**, buatlah project baru dengan nama **Project_jsp**. Kemudian klik kanan pada **Web Pages >> Pilih New >> JSP**. Isikan nama file dengan **Latihan1**. Lihat gambar di bawah ini.





Kemudian klik **Finish** maka akan tampil file baru JSP dengan nama **Latihan1.jsp**.



Tampilan di atas adalah ketika file jsp dijalankan pada browser.

2. Dasar-dasar JSP

Berikutnya kita akan mempelajari mengenai Syntax pada JSP. Berikut elemen-elemen JSP.

a. Scriptlet

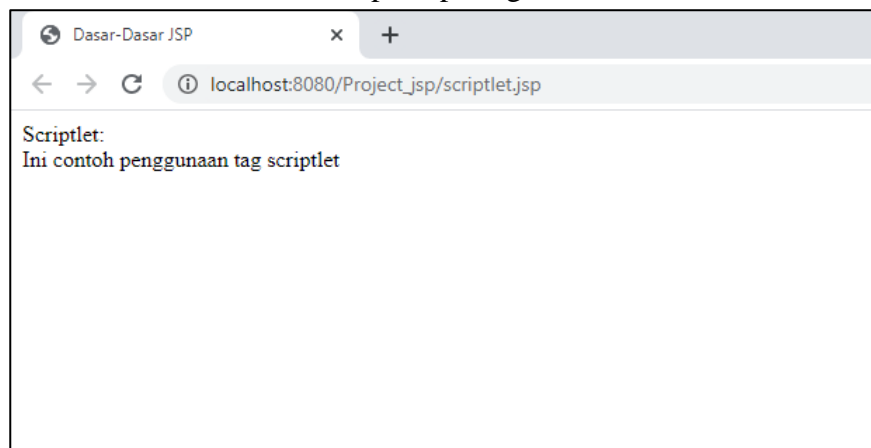
Scriptlet dapat berisi sejumlah kode program bahasa Java, variabel atau metode deklarasi, atau ekspresi yang valid setiap kali halaman jsp dipanggil. Pada bagian ini kita bisa memasukkan kode-kode program java yang telah dipahami. Berikut format penulisan syntax scriptlet:

```
<% code fragment %>
```

Ketikkan kode program berikut dan simpan dengan nama **scriptlet.jsp**

```
<html>
  <head>
    <title>Dasar-Dasar JSP</title>
  </head>
  <body>
    Scriptlet:
    <br>
    <%
      out.println("Ini contoh penggunaan tag scriptlet");
    %>
  </body>
</html>
```

Jalankan file tersebut maka akan tampil seperti gambar di bawah ini.



b. Deklarasi (Declarations)

Tag ini digunakan untuk mendeklarasikan satu atau lebih variabel dan bisa disebut sebagai metode yang dapat digunakan dalam kode program Java. Anda harus

mendeklarasikan variabel sebelum menggunakannya di file JSP. Berikut ini adalah syntax dasar untuk declarations.

```
<%! declaration; [ declaration; ]+ ... %>
```

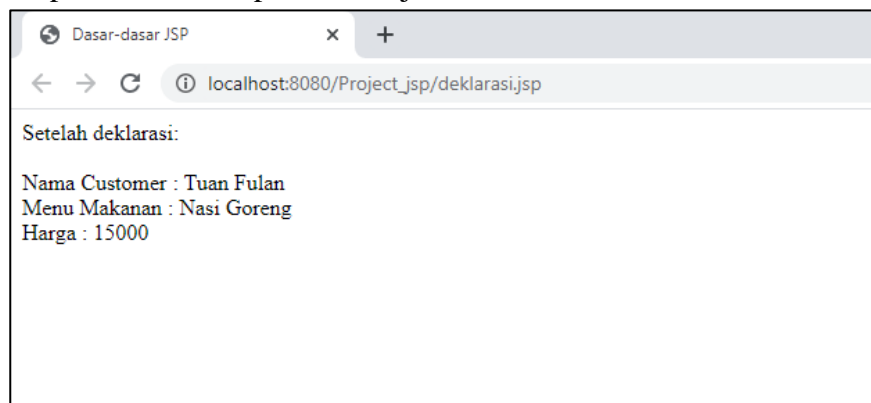
Contoh penggunaan deklarasi seperti kode program berikut, simpan dengan nama **deklarasi.jsp**.

```
<html>
  <head>
    <title>Dasar-dasar JSP</title>
  </head>
  <body>
    <%--Proses Pendeklarasian:--%>
    <%!
      String nama;
      String makanan;
      int harga = 15000;
    %>

    Setelah deklarasi: <br><br>
    <%
      nama = "Tuan Fulan";
      makanan = "Nasi Goreng";

      out.println("Nama Customer : " + nama + "<br>");
      out.println("Menu Makanan : " + makanan + "<br>");
      out.println("Harga : " + harga);
    %>
  </body>
</html>
```

Berikut tampilan ketika script di atas dijalankan.



c. Ekspresi (Expression)

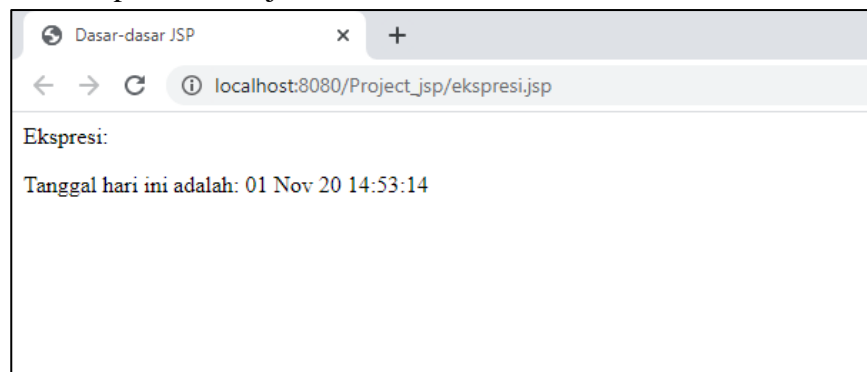
Elemen ekspresi merupakan satu baris perintah yang digunakan untuk mengeksekusi perintah dan sekaligus menampilkan ekspresi dalam halaman web. Penggunaan ekspresi mirip seperti ketika kita memanggil fungsi/method pada Java. Elemen ekspresi dapat berisi ekspresi apapun yang valid sesuai dengan spesifikasi bahasa Java, dan uniknya ekspresi tidak menggunakan titik koma di akhir script. Berikut adalah syntax dari ekspresi.

```
<%= expression %>
```

Contoh penggunaan tag ekspresi dengan nama file **ekspresi.jsp**.

```
<html>
<head>
<title>Dasar-dasar JSP</title>
</head>
<body>
Ekspresi:
  <p>
    Tanggal hari ini adalah: <%= (new java.util.Date()).toLocaleString()%>
  </p>
</body>
</html>
```

Berikut hasil script setelah dijalankan.



d. Komentar (Comments)

Elemen komentar bisa dikatakan sangat penting dalam hal '*ngoding*'. Baris komentar digunakan untuk menyembunyikan atau mengomentari kode-kode program JSP. Komentar tidak dieksekusi ketika program running, namun tertulis pada script. Setiap bahasa pemrograman memiliki ciri khas tag komentar masing-masing dalam memberikan keterangan pada baris kode program. Berikut ini adalah syntax komentar pada JSP.

```
<%-- JSP comments --%>
```

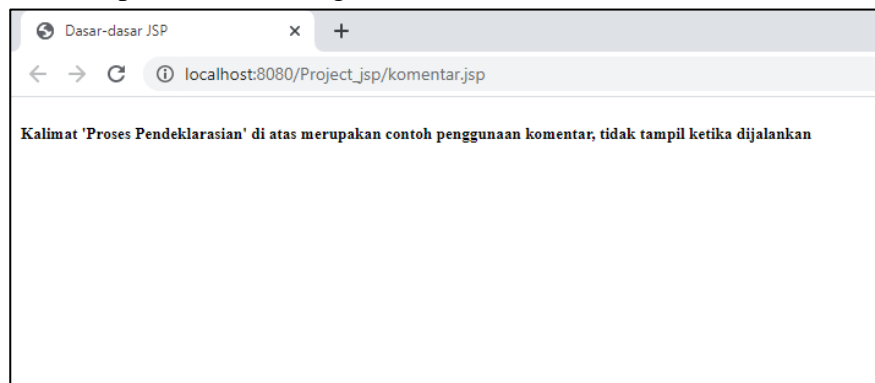
Apabila berada dalam blok kode program `<% ... %>`, pemberian komentar juga bisa menggunakan double slash `"// "`.

Contoh penggunaan tag komentar pada JSP, dengan menggunakan nama **komentar.jsp**.

```
<html>
  <head>
    <title>Dasar-dasar JSP</title>
  </head>
  <body>
    <%--Proses Pendeklarasian --%>
    <%!
      String nama;
      String makanan;
      int harga = 15000;
    %>

    <h5>Kalimat 'Proses Pendeklarasian' di atas merupakan contoh penggunaan komentar,
    tidak tampil ketika dijalankan</h5>
  </body>
</html>
```

Berikut hasil script setelah running.



e. Directive

Direktif mempengaruhi keseluruhan struktur kelas servlet, digunakan untuk mengirimkan pesan ke JSP container dan melakukan setting nilai global seperti deklarasi class atau method. Berikut bentuk umum syntax direktif yang biasa digunakan.

```
<% @ directive attribute="value" %>
```

Tag direktif terbagi menjadi 3 jenis, yaitu:

Tag	Deskripsi
-----	-----------

<code><%@ page ... %></code>	Tag Page , digunakan untuk mendefinisikan atribut-atribut yang terdapat pada halaman jsp. Misalnya atribut language, import, info, error page dan lain-lain.
<code><%@ include ... %></code>	Tag Include , digunakan untuk menyisipkan suatu berkas atau mengimpor suatu kelas.
<code><%@ taglib ... %></code>	Taglib , digunakan untuk mendeklarasikan tag library yang dibuat sendiri oleh programmer pada halaman web.

3.3. Variabel

Variabel adalah sebuah tempat penyimpanan data dalam bahasa pemrograman berupa nilai-nilai tertentu yang bersifat sementara atau bisa diubah-ubah. Pendeklarasian variabel setiap bahasa pemrograman berbeda-beda. Ada yang harus diawali dengan penyebutan tipe data terlebih dahulu, ada juga yang langsung mendeklarasikan variabel menggunakan simbol tertentu tanpa menyebutkan tipe data. Contohnya pada bahasa pemrograman PHP tidak memerlukan pendeklarasian tipe data suatu variabel secara eksplisit hanya ditentukan saat runtime program. Namun, pada bahasa JSP sendiri harus menggunakan tipe data untuk mendeklarasikan suatu variabel.

1. Membuat Variabel pada JSP

Beberapa poin-poin penting dalam pembuatan sebuah variabel, diantaranya:

- Nama variabel hanya dapat menampung satu jenis nilai.
- Pemberian nama variabel diawali dengan huruf, tidak boleh diawali dengan angka, simbol-simbol, ataupun tanda baca seperti: @ ; ! & dan lain-lain.
- Apabila ingin menggunakan dua suku kata, nama variabel tidak boleh mengandung spasi, namun disambung menggunakan underscore (_).
- Nama variabel tidak boleh mengandung operator aritmatika.

Berikut contoh pendeklarasian variabel pada bahasa Java.

```
String nm_pelanggan;
String makanan;
Double total;
```

Berdasarkan contoh di atas, string dan double merupakan tipe data, sedangkan nm_pelanggan, makanan dan total adalah nama variabel yang dibuat oleh programmer.

2. Pemberian Nilai & Pemanggilan Variabel

Setelah variabel dideklarasikan, variabel dapat diberi nilai. Caranya dengan menggunakan operator sama dengan (=).

Bentuk umumnya:

```
Tipe_data Nama_variabel = nilai;
```

Contoh:

```
int jumlah_barang = 10;  
String nama = "Aji Pangestu";
```

Pemanggilan nama variabel diikuti dengan pemberian tanda plus (+) di depannya. Hal penting yang perlu diingat adalah nama variabel yang dipanggil harus sama dengan nama variabel yang telah dideklarasikan sebelumnya. Dengan menggunakan perintah **out.println**.

Contoh pada script berikut ini.

```
<html>  
  <head>  
    <title>Dasar-dasar JSP</title>  
  </head>  
  <body>  
    <!--Proses Pendeklarasian-->  
    <%!  
      String nama;  
      String makanan;  
      int harga = 15000;  
    %>  
  
    Setelah deklarasi:  
    <%  
      nama = "Tuan Fulan";  
      makanan = "Nasi Goreng";  
  
      out.println("Nama Customer : " +nama+ "<br>");  
      out.println("Menu Makanan : " +makanan+ "<br>");  
      out.println("Harga : " +harga);  
    %>  
  </body>  
</html>
```

3.4. Tipe Data

Tipe data dalam bahasa pemrograman adalah jenis data yang tersimpan dalam variabel. Suatu data yang dikumpulkan pasti memiliki struktur yang berbeda-beda, contohnya seperti penggunaan huruf dan angka. Secara garis besar, terdapat 2 kelompok tipe data dalam bahasa Java, yakni tipe data sederhana (primitive data types), dan tipe data kompleks/object (non-primitive data types).

1. Tipe Data Sederhana (Primitive)

Tipe data jenis ini digunakan untuk manipulasi data. Ini adalah tipe data paling dasar dalam bahasa Java. Berikut ini macam-macam tipe data sederhana:

Tipe Data (Primitive)	Kategori	Jangkauan (Range)
byte	Bilangan Bulat	-128 s.d. 127
short		-32,768 s.d. 32,767
int (integer)		-32,768 to 32,767 atau -2,147,483,648 s.d. 2,147,483,647
long		-2,147,483,648 s.d. 2,147,483,647
float	Bilangan Pecahan	-3.4E38 s.d. 3.4E38
double		-1.7E308 s.d. 1.7E308
char	Karakter	-128 to 127 atau 0 s.d. 255
boolean	-	-

- **Tipe Bilangan Bulat**

Kategori tipe data ini hanya terdiri dari angka bilangan bulat dari 0 s.d. 9. Perbedaan masing-masing tipe data terletak pada jangkauan nilai yang dicakup. Terbagi 4 macam tipe, yaitu byte, short, int, dan long. Keempat jenis ini digunakan untuk operasi data bilangan bulat dan perhitungan aritmatika. Namun yang paling sering digunakan hanya int dan long. Contohnya: 34, 56, 100, dll.

- **Tipe Bilangan Pecahan**

Disebut juga dengan Floating Point. Pada dasarnya digunakan ketika kita mempunyai situasi dimana mendapatkan hasil atau output dalam bentuk desimal. Kelompok tipe data bilangan pecahan ini yaitu float dan double. Contohnya: 3.14 atau 89.65.

- **Tipe Data Karakter**

Tipe data karakter atau char ini hanya untuk menampung satu karakter, bisa berupa simbol, huruf, ataupun angka.
Contohnya: 'a', 'Z', atau '%'.

- **Tipe Data Boolean**

Tipe data Boolean hanya digunakan untuk menyimpan dua kemungkinan nilai yaitu benar (true) atau salah (false). True berarti bernilai 1, sedangkan false bernilai 0.

2. Tipe Data Kompleks (Non-Primitive)

Berikut beberapa jenis tipe data kompleks:

- **Tipe Data String**

Tipe data yang merupakan kumpulan dari karakter sehingga membentuk teks. Contohnya: “Ilmu Komputer”, “Belajar JSP”, dan lain-lain.

- **Tipe Data Array**

Tipe data untuk kumpulan tipe data lain yang sejenis.

- **Tipe Data Object**

Tipe data khusus yang bisa menampung beragam data, termasuk memiliki function/method sendiri.

3.5. Pengenalan Operator

Hal yang tidak kalah penting dalam pemrograman adalah operator. Operator digunakan untuk mengolah data. Operator ini merupakan instruksi yang diberikan untuk mendapatkan hasil dari proses operasi terhadap nilai operand. Sementara operand adalah nilai yang dilibatkan dalam operasi oleh operator. Terdapat beberapa jenis operator pada bahasa pemrograman Java yaitu sebagai berikut.

1. Operator Aritmatika

Operator ini digunakan untuk melakukan perhitungan atau operasi aritmatika. Aritmatika merupakan cabang ilmu matematika yang membahas perhitungan sederhana seperti kali, bagi, tambah dan kurang. Pada bahasa Java juga memiliki operasi *modulo division* atau disebut dengan *mod* yang dipakai untuk mencari sisa hasil bagi.

Operator	Penjelasan	Contoh
+	Penambahan	$x = 10 + 2$
-	Pengurangan	$x = 10 - 2$
*	Perkalian	$x = 10 * 2$
/	Pembagian	$x = 10 / 2$
%	Sisa Hasil Bagi	$x = 10 \% 2$

Contoh penggunaan dalam script. Buatlah file jsp baru dengan nama **operator_aritmatika.jsp**.

```
<html>
<head>
  <title>Latihan Operator</title>
</head>
<body>
  Berikut penggunaan Operator Aritmatika: <br>
  <%!
    int nilai1 = 10;
    int nilai2 = 2;
    int tambah,kurang, kali, mod;
    float bagi;
  %>
```

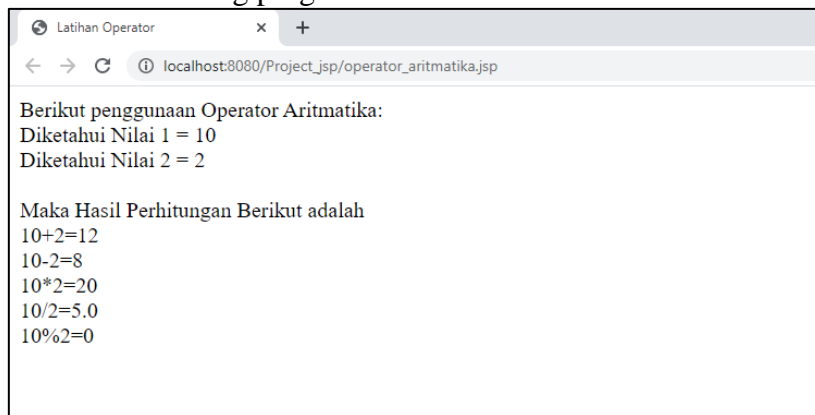
```

<%
    tambah=nilai1+nilai2;
    kurang=nilai1-nilai2;
    kali=nilai1*nilai2;
    bagi=nilai1/nilai2;
    mod=nilai1%nilai2;

    out.println("Diketahui Nilai 1 = " +nilai1+ "<br>");
    out.println("Diketahui Nilai 2 = " +nilai2+ "<br><br>");
    out.println("Maka Hasil Perhitungan Berikut adalah <br>");
    out.println(nilai1+ "+" +nilai2+ "=" +tambah+ "<br>");
    out.println(nilai1+ "-" +nilai2+ "=" +kurang+ "<br>");
    out.println(nilai1+ "*" +nilai2+ "=" +kali+ "<br>");
    out.println(nilai1+ "/" +nilai2+ "=" +bagi+ "<br>");
    out.println(nilai1+ "%" +nilai2+ "=" +mod);
%>
</body>
</html>

```

Di bawah ini adalah hasil running program di atas.



2. Operator Logika

Operator logika dipakai untuk menghasilkan nilai boolean true atau false dari 2 kondisi atau lebih.

Operator	Nama	Penjelasan
&&	And	Akan menghasilkan true jika kedua operand true (benar)
	Or	Akan menghasilkan true jika salah satu operand true (benar)
!	Not	Akan menghasilkan true jika operand false

Contoh penggunaan dalam script. Buatlah file jsp baru dengan nama **operator_logika.jsp**.

```

<html>
<head>
<title>Latihan Operator</title>
</head>
<body>
    Berikut Penggunaan Operator Logika: <br>
    <%!
        int nilai1 = 10;
        int nilai2 = 2;
        String a, b;
        Boolean hasil_or, hasil_and, hasil_negasi,p1,p2;
    %>

```

```

<%
a="JSP";
b="jsp";
p1= nilai1 != nilai2;
p2= a==b;

hasil_or=(nilai1 != nilai2) || (a==b);
hasil_and=(nilai1 != nilai2) && (a==b);
hasil_negasi=! (a==b);

out.println("Diketahui Pernyataan 1 (P1) : " +nilai1+ " != " +nilai2+ " , maka hasil P1 = "+p1+ "<br>");

out.println("Diketahui Pernyataan 2 (P2) : " +a+ " == " +b+ " , maka hasil P2 = " +p2+ "<br><br>");

out.println("Maka hasil logika dari kedua pernyataan tersebut adalah : <br>");

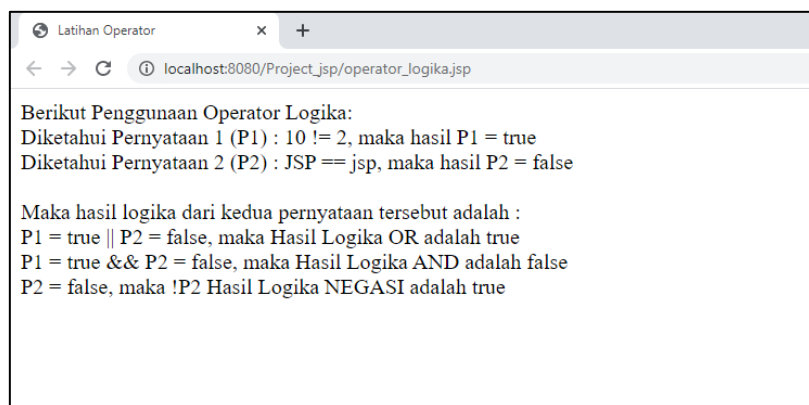
out.println("P1 = " +p1+ " || P2 = "+p2+ " , maka Hasil Logika OR adalah " +hasil_or+ "<br>");

out.println("P1 = " +p1+ " && P2 = "+p2+ " , maka Hasil Logika AND adalah " +hasil_and+ "<br>");

out.println("P2 = "+p2+ " , maka !P2 Hasil Logika NEGASI adalah " +hasil_negasi);
%>
</body>
</html>

```

Berikut hasil script di atas ketika dijalankan.



Jadi, bisa dikatakan logika And itu ‘si perfeksionis’ dan logika Or ‘si legowo’. Jika masih belum paham dari contoh di atas, silahkan cek lagi tabel kebenaran dari logika AND dan OR di bawah ini.

Pernyataan 1	Pernyataan 2	Logika AND	Logika OR
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

3. Operator Perbandingan

Operator ini digunakan sebagai pembandingan dari dua nilai yang hasil akhirnya adalah boolean true atau false. Operator ini sangat berguna dalam pemrograman karena bisa menentukan arah pemrograman. Berikut keenam operator perbandingan beserta contoh dan hasil.

Operator	Nama / Penjelasan	Contoh	Hasil
==	Sama dengan	7 == 7	true
!=	Tidak sama dengan	7 != 7	false
>	Lebih besar	7 > 5	true
<	Lebih kecil	7 < 5	false
>=	Lebih besar sama dengan	5 >= 7	false
<=	Lebih kecil sama dengan	5 <= 7	true

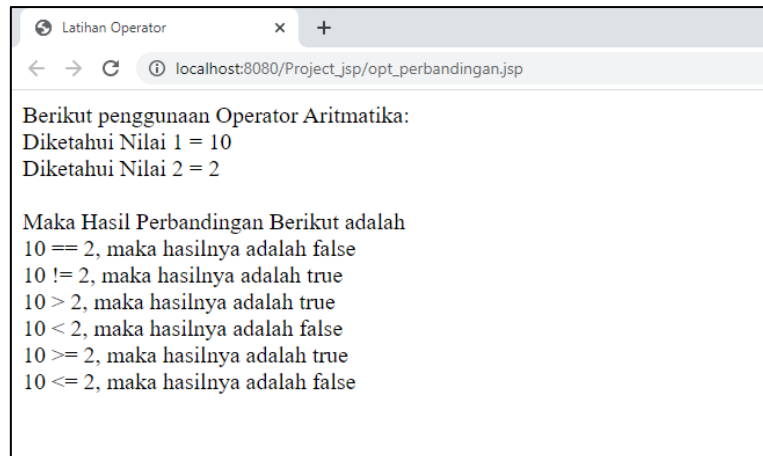
Contoh penggunaan operator perbandingan dalam script dengan nama file **opt_perbandingan.jsp**.

```
<html>
<head>
  <title>Latihan Operator</title>
</head>
<body>
  Berikut penggunaan Operator Aritmatika: <br>
  <%!
    int nilai1 = 10;
    int nilai2 = 2;
    Boolean hasil;
  %>

  <%
    out.println("Diketahui Nilai 1 = " + nilai1 + "<br>");
    out.println("Diketahui Nilai 2 = " + nilai2 + "<br><br>");
    out.println("Maka Hasil Perbandingan Berikut adalah <br>");

    hasil= nilai1 == nilai2;
    out.println(nilai1 + " == " + nilai2 + ", maka hasilnya adalah "+hasil+ "<br>");
    hasil= nilai1 != nilai2;
    out.println(nilai1 + " != " + nilai2 + ", maka hasilnya adalah "+hasil+ "<br>");
    hasil= nilai1 > nilai2;
    out.println(nilai1 + " > " + nilai2 + ", maka hasilnya adalah "+hasil+ "<br>");
    hasil= nilai1 < nilai2;
    out.println(nilai1 + " < " + nilai2 + ", maka hasilnya adalah "+hasil+ "<br>");
    hasil= nilai1 >= nilai2;
    out.println(nilai1 + " >= " + nilai2 + ", maka hasilnya adalah "+hasil+ "<br>");
    hasil= nilai1 <= nilai2;
    out.println(nilai1 + " <= " + nilai2 + ", maka hasilnya adalah "+hasil+ "<br>");
  %>
</body>
</html>
```

Berikut hasil script di atas ketika dijalankan.



4. Operator Penugasan

Disebut juga sebagai assignment operator yang digunakan untuk memberikan tugas atau nilai ke dalam suatu variabel. Berikut yang termasuk ke dalam operator penugasan.

Operator	Nama	Contoh
=	Pemberian nilai	a = 10
+=	Pengisian dan Penambahan	b +=a
-=	Pengisian dan Pengurangan	a -= c
*=	Pengisian dan Perkalian	a *= b
/=	Pengisian dan Pembagian	c /= a
%=	Pengisian dan Sisa Bagi	a %= b

Contoh penggunaan operator penugasan dalam script dengan nama file **opt_penugasan.jsp**.

```
<html>
<head>
  <title>Latihan Operator</title>
</head>
<body>
  Berikut penggunaan Operator Penugasan/Assignment: <br>
  <% !
    int a;
    int b;
    //int tambah, kurang, kali, mod ;
    //float bagi;
  %>

  <%
    //Pemberian NILAI dengan operator " = "
    a = 10;
    b = 5;

    out.println("Diketahui Nilai awal a = " +a+ "<br>");
    out.println("Diketahui Nilai awal b = " +b+ "<br><br>");
    out.println("Maka hasil perhitungan berikut: <br>");

    //Penambahan
    a+=b;
    out.println("Setelah Penambahan a+=b, maka nilai a sekarang= " + a + "<br>");

    //Pengurangan
    a-=b;
```

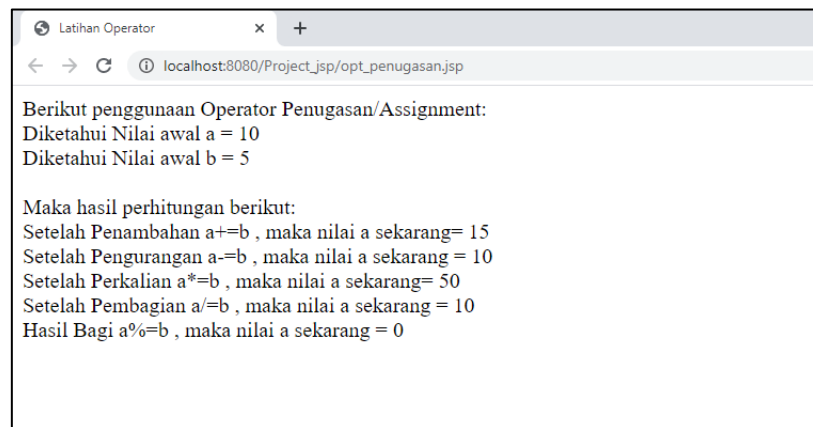
```

out.println("Setelah Pengurangan a-=b, maka nilai a sekarang = " + a + "<br>");
//Perkalian
a*=b;
out.println("Setelah Perkalian a*=b, maka nilai a sekarang= " + a + "<br>");
//Pembagian
a/=b;
out.println("Setelah Pembagian a/=b, maka nilai a sekarang = " + a + "<br>");
//Sisa Hasil Bagi
a%=b;
out.println("Hasil Bagi a%=b, maka nilai a sekarang = " + a + "<br>");

%>
</body>
</html>

```

Berikut hasil dari perubahan nilai variabel a setelah menggunakan operator penugasan.



5. Operator Bitwise

Operator bitwise adalah operator khusus digunakan untuk menangani operasi logika bilangan biner yang akan menghitung dari bit ke bit. Bilangan biner sendiri merupakan jenis bilangan yang hanya terdiri dari 2 jenis angka, yakni 0 dan 1. Jika nilai asal yang dipakai bukan bilangan biner, akan dikonversi secara otomatis oleh compiler Java menjadi bilangan biner. Operator ini berlaku untuk tipe data int, long, short, char, dan byte. Operator bitwise terdiri dari:

Operator	Nama
&	And
	Or
^	Xor
~	Negasi/Kebalikan
<<	Left Shift
>>	Right Shift

3.6. Pembuatan JSP Form

Formulir atau yang lebih dikenal dengan form dalam pemrograman adalah komponen dasar yang membantu terjadinya interaksi antara user (human) dengan halaman web (computer). Form digunakan untuk mengumpulkan data dari pengunjung web atau user. Dalam proses pembuatan form, browser menggunakan metode GET dan POST untuk meneruskan informasi ke web server. Metode atau method ini merupakan atribut yang terdapat pada tag form.

1. Komponen Form

Dalam Form HTML terdapat beberapa komponen yang bisa digunakan, antara lain:

a. Text Box

Text box atau input text berfungsi untuk memasukkan inputan data berupa angka maupun teks/string. Penulisan teks seperti berikut.

```
<input type="text" name="nama_variabel" value="value">
```

Contoh input text dengan label nama:

Nama : <input type="text"/>

b. Radio Button

Radio button digunakan untuk hanya memilih satu pernyataan dari beberapa pernyataan yang disediakan. Penulisan radio button seperti berikut. Radio Button akan efektif jika opsinya kurang dari 6 atau 5. Jika opsinya terlalu banyak maka gunakan Select Box.

```
<input type="radio" name="nama_variabel" value="value">
```

Contoh dengan label jenis kelamin:

Jenis Kelamin : <input type="radio"/> Laki-laki <input type="radio"/> Perempuan

c. Text Area

Text Area digunakan untuk menginput string ataupun angka yang terdiri atas banyak baris. Textarea lebih cocok untuk menuliskan sesuatu yang panjang karena memiliki textbox yang lebih tinggi dan lebih panjang. Penulisannya seperti berikut.

```
<textarea name="nama_variabel" rows=" " cols=" " placeholder="Isikan alamat lengkap Anda"></textarea>
```

Contoh dengan label alamat:

Alamat :

Isikan alamat lengkap Anda

d. Select Box

Select box atau biasa sering disebut combo box ini digunakan untuk menampilkan banyak pilihan data. Select box memiliki indikator anak panah di sebelah kanan, ketika di klik akan muncul beberapa pilihan yang disediakan oleh sistem dan user hanya bisa memilih satu opsi saja. Penulisannya seperti berikut.

```
<select name="nama_variabel">
<option value="" ">Combo 1</option>
<option value="" ">Combo 2</option>
<option value="" ">Combo 3</option>
</select>
```

Contoh dengan label jenis pendidikan:

Pendidikan :

== PILIH ==

== PILIH ==

SMA

S1

S2

S3

e. Checkbox

Checkbox digunakan untuk memilih satu pilihan atau lebih dari beberapa pernyataan yang disediakan. Penulisannya seperti berikut.

```
<input type="checkbox" name="nama_variabel" value="value">Pilihan1
```

Contoh dengan label pilihan hobi:

Hobi :

☐ Travelling
 ☐ Baca

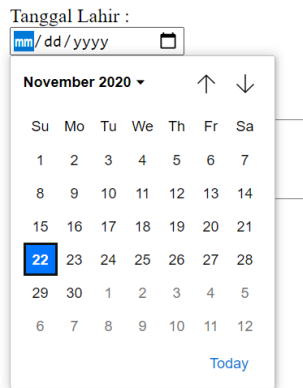
☐ Badminton
 ☐ Cooking

f. Datepicker

Komponen ini digunakan untuk memilih tanggal. Terdapat icon kalender di sebelah kanan untuk user memilih tanggal dengan meng-klik icon tersebut kemudian akan muncul kalender. Berikut penulisannya.

```
<input type="date" name="nama_variabel">
```

Contoh penggunaan datepicker dengan objek tanggal lahir.



g. Submit & Reset

Submit berfungsi untuk mengirimkan semua variabel data pada komponen-komponen form yang ada. Sedangkan reset untuk membatalkan semua penginputan yang telah dituliskan. Berikut penulisannya.

```
<input type="submit" name="submit" value="submit">
<input type="reset" name="reset" value="reset">
```

Contoh penggunaan tombol submit dan reset:

2. Metode GET & POST

Semua komponen form dapat dikenali sebagai variabel yang bernilai data, dan harus diletakkan di antara tag buka dan tutup `<form>...</form>`. Di dalam properti tag form, terdapat 2 atribut, yaitu action dan method. Action adalah nama file web tujuan yang akan menerima variabel data yang dikirimkan melalui form. Nah, sedangkan atribut method adalah jenis pengiriman informasi atau variabel data, bisa menggunakan metode GET atau POST.

a. Metode GET

Metode GET digunakan untuk mengirim informasi dari browser ke web server sehingga akan tampak pada URL browser. Setiap informasi dipisahkan oleh karakter "?". Method GET lebih cocok untuk pengiriman variabel disertai argumen yang panjang dan tidak membutuhkan keamanan lebih, makanya metode ini biasa digunakan untuk query pencarian. Sebaiknya hindari untuk pengiriman informasi sensitif seperti kata sandi, dan lain-lain. Metode GET hanya bisa menampung 1024 karakter. Struktur penulisannya bisa dilihat di bawah ini.

```
<form method="GET" action="tujuan.jsp">
komponen-komponen form
```

```
</form>
```

b. Metode POST

Metode POST digunakan untuk mengirimkan data dalam jumlah besar ke web server. Metode ini lebih cocok untuk pengiriman informasi yang sensitif karena informasi yang dikirimkan oleh browser ke web server tidak akan terlihat pada URL browser. Struktur penulisan metode POST bisa dilihat di bawah ini.

```
<form method="POST" action="tujuan.jsp">  
komponen-komponen form  
</form>
```

Ada beberapa method yang digunakan untuk membaca atau menangkap data form pada JSP, yaitu:

- **getParameter()**
Method ini berfungsi untuk mendapatkan nilai parameter dari form.
- **getParameterValues()**
Method ini digunakan apabila parameter muncul lebih dari sekali dan mengirimkan banyak nilai, misalnya pada komponen **checkbox**.
- **getParameterNames()**
Pemanggilan method ini apabila menginginkan list yang lengkap dari semua parameter pada request yang bersamaan.
- **getInputStream()**
Method ini digunakan untuk menangkap data biner yang dikirim user.

3. Metode GET pada Form

Berikut adalah kode program untuk pembuatan form menggunakan metode GET dengan membuat 2 file baru sebagai file input dan ouput.

Ketikkan script file input berikut dengan nama **method_get.jsp**.

```
<html>  
<head>  
<title>Latihan Pembuatan Form</title>  
</head>  
<body>  
<h3>Metode GET</h3>  
<form action="tampil_get.jsp" method="GET">  
<table>  
<tr>  
<td>Nama</td>  
<td><input type="text" name="nama"></td>  
</tr>  
</table>
```

```

        <td>Domisili</td>
        <td><input type="text" name="domisili"></td>
    </tr>
    <tr>
        <td>Pekerjaan</td>
        <td><input type="text" name="job"></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="SEND">
            <input type="reset" value="CANCEL">
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Dan buat file output menggunakan nama **tampil_get.jsp**.

```

<html>
<head>
    <title>Latihan Pembuatan Form</title>
</head>
<body>
    <h3>JSP Form Metode GET</h3>
    <table border="0">
        <tr>
            <td width="70px">Nama</td>
            <td width="10px">:</td>
            <td><%= request.getParameter("nama") %></td>
        </tr>
        <tr>
            <td>Domisili</td>
            <td>:</td>
            <td><%= request.getParameter("domisili") %></td>
        </tr>
        <tr>
            <td>Pekerjaan</td>
            <td>:</td>
            <td><%= request.getParameter("job") %></td>
        </tr>
        <tr>
            <td colspan="3" align="center"><a href="method_get.jsp"> Back to Page</a>
        </td>
        </tr>
    </table>
</body>
</html>

```

Berikut hasil eksekusi form input **method_get.jsp**.

Berikut hasil eksekusi file output **tampil_get.jsp**. Nah, perhatikan lingkaran merah pada gambar di bawah ini. Terlihat jelas bahwa apabila kita menggunakan metode GET, maka semua data/record yang kita kirim juga akan tampak pada URL browser.

4. Metode POST pada Form

Berikut adalah kode program untuk pembuatan form menggunakan metode POST dengan membuat 2 file baru sebagai file input dan ouput.

Ketikkan script file input berikut dengan nama **method_post.jsp**.

```
<html>
<head>
  <title>Latihan Pembuatan Form</title>
</head>
<body>
  <h3>Metode POST</h3>
  <form action="tampil_post.jsp" method="POST">
    <table>
      <tr>
        <td>Nama</td>
        <td><input type="text" name="nama"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

```

        <td>Domisili</td>
        <td><input type="text" name="domisili"></td>
    </tr>
    <tr>
        <td>Pekerjaan</td>
        <td><input type="text" name="job"></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="SEND">
            <input type="reset" value="CANCEL">
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Dan buat file output menggunakan nama **tampil_post.jsp**.

```

<html>
<head>
    <title>Latihan Pembuatan Form</title>
</head>
<body>
    <h3>JSP Form Metode POST</h3>
    <table border="0">
        <tr>
            <td width="70px">Nama</td>
            <td width="10px">:</td>
            <td><%= request.getParameter("nama") %></td>
        </tr>
        <tr>
            <td>Domisili</td>
            <td>:</td>
            <td><%= request.getParameter("domisili") %></td>
        </tr>
        <tr>
            <td>Pekerjaan</td>
            <td>:</td>
            <td><%= request.getParameter("job") %></td>
        </tr>
        <tr>
            <td colspan="3" align="center"><a href="method_post.jsp">Back to Page</a>
        </td>
        </tr>
    </table>
</body>
</html>

```

Berikut hasil eksekusi form input **method_post.jsp**.

Metode POST

Nama :

Domisili :

Pekerjaan :

Berikut hasil eksekusi file output **tampil_post.jsp**.

JSP Form Metode POST

Nama : Dayang Maharani

Domisili : Pontianak, Kalbar

Pekerjaan : Programmer

Nah, perhatikan lingkaran merah pada gambar di atas. Jika menggunakan **method POST**, maka semua data/record yang dikirim tidak akan terlihat pada URL browser. Hanya nama file jsp yang terlihat, dengan kata lain data terenkripsi.

5. Membuat Form pada JSP

Buat file untuk menampilkan form input dan output pada JSP. Pada latihan pembuatan form ini menggunakan method POST.

Ketikkan script **data_in.jsp** berikut.

```
<html>
  <head>
    <title>Membuat Form pada JSP</title>
  </head>
  <body>
    <h2 align="left">Formulir Pendaftaran Seminar Programming</h2>
    <hr align="left" width="500" color="blue">
    <form action="data_out.jsp" method="POST">
    <table align="left" border="0">
      <tr>
        <td>Nama Lengkap</td>
        <td>:</td>
        <td><input type="text" name="nama" size="30"></td>
      </tr>
      <tr>
        <td>Jenis Kelamin</td>
        <td>:</td>
        <td><input type="text" name="kelamin" size="30"></td>
      </tr>
    </table>
  </body>
</html>
```



```

        <input type="radio" name="jkel" value="Laki-laki">Laki-laki
        <input type="radio" name="jkel" value="Perempuan">Perempuan
    </td>
</tr>
<tr>
    <td>Alamat</td>
    <td>:</td>
    <td><textarea name="alamat" cols="30" rows="4" placeholder="Isikan Alamat Lengkap
Anda"></textarea>
    </td>
</tr>
<tr>
    <td>Pekerjaan</td>
    <td>:</td>
    <td><select name="job">
        <option value=""> -- PILIH -- </option>
        <option value="Mahasiswa"> Mahasiswa </option>
        <option value="Dosen"> Dosen </option>
        <option value="Programmer"> Programmer </option>
    </select></td>
</tr>
<tr>
    <td>No.HP</td>
    <td>:</td>
    <td><input type="text" name="nohp" size="20"></td>
</tr>
<tr>
    <td colspan="3" align="center">
        <input type="submit" value="SUBMIT">
        <input type="reset" value="CANCEL">
    </td>
</tr>
</table>
</form>
</body>
</html>

```

Ketikkan script **data_out.jsp** di bawah ini.

```

<html>
    <head>
        <title>Membuat Form pada JSP</title>
    </head>
    <body>
        <%
            //Mendapatkan Data dari form JSP

String xNama=request.getParameter("nama");
String xJkel=request.getParameter("jkel");
String xAlamat=request.getParameter("alamat");
String xJob=request.getParameter("job");
String xNo=request.getParameter("nohp");
%>

<h2 align="left">Data Peserta Seminar Programming</h2>
<hr align="left" width="380" color="blue">
<form action="data_out.jsp" method="POST">
<table align="left" border="1">
    <tr>
        <td width="140px">Nama Lengkap</td>
        <td>:</td>
        <td><% out.println(xNama); %></td>
    </tr>

```

```

<tr>
    <td>Jenis Kelamin</td>
    <td>:</td>
    <td><% out.println(xJkel); %></td>
</tr>
<tr>
    <td>Alamat</td>
    <td>:</td>
    <td><% out.println(xAlamat); %></td>
</tr>
<tr>
    <td>Pekerjaan</td>
    <td>:</td>
    <td><% out.println(xJob); %></td>
</tr>
<tr>
    <td>No.HP</td>
    <td>:</td>
    <td><% out.println(xNo); %></td>
</tr>
<tr>
    <td colspan="3" align="center">
<a href="data_in.jsp">Back To Form</a>
</td>
</tr>
</table>
</form>
</body>
</html>

```

Berikut hasil running file input **data_in.jsp**.

The screenshot shows a web browser window with the title 'Membuat Form pada JSP'. The address bar shows 'localhost:8080/Project_jsp/data_in.jsp'. The main content is a form titled 'Formulir Pendaftaran Seminar Programming'. The form contains the following elements:

- Nama Lengkap :** A text input field.
- Jenis Kelamin :** Two radio buttons labeled 'Laki-laki' and 'Perempuan'.
- Alamat :** A text area with the placeholder text 'Isikan Alamat Lengkap Anda'.
- Pekerjaan :** A dropdown menu with the text '-- PILIH --'.
- No.HP :** A text input field.
- Buttons:** Two buttons labeled 'SUBMIT' and 'CANCEL' at the bottom.

Dan di bawah ini adalah hasil running file output **data_out.jsp**

Data Peserta Seminar Programming	
Nama Lengkap	: Dayang Maharani
Jenis Kelamin	: Perempuan
Alamat	: Jl.Khatulistiwa No.12 Pontianak
Pekerjaan	: Programmer
No.HP	: 089678920390
Back To Form	

3.7. Percabangan (Branching)

Percabangan yang juga dikenal dengan Control Flow ini adalah bentuk struktur kondisi yang ada dalam pemrograman yang berfungsi untuk mengatur jalannya alur program. Dalam pembuatan program, sebagian besar menggunakan percabangan. Cara kerja struktur percabangan ini adalah dengan menentukan suatu opsi, apabila kondisi terpenuhi maka akan dijalankan perintah atau pernyataan dan jika tidak, maka jalankan perintah lainnya. Pada bahasa pemrograman Java, percabangan terbagi menjadi beberapa macam, yaitu:

1. Percabangan IF

Bentuk percabangan yang paling sederhana, yakni kondisi IF dengan aturan penulisan sebagai berikut.

```
if (kondisi)
{
    pernyataan (statement);
}
```

Prinsip kerjanya adalah perintah di atas akan dikerjakan apabila kondisi bernilai true atau terpenuhi, sedangkan jika kondisi salah (false) maka statement tidak akan dikerjakan.

2. IF-Else

Pada bentuk percabangan IF dan Else ini memiliki dua kondisi dan pernyataan. Else digunakan untuk memberikan alternatif perintah apabila kondisi pertama bernilai salah (false). Penulisan sebagai berikut.

```
if (kondisi 1)
{
    pernyataan 1(statement);
}
else
{
```

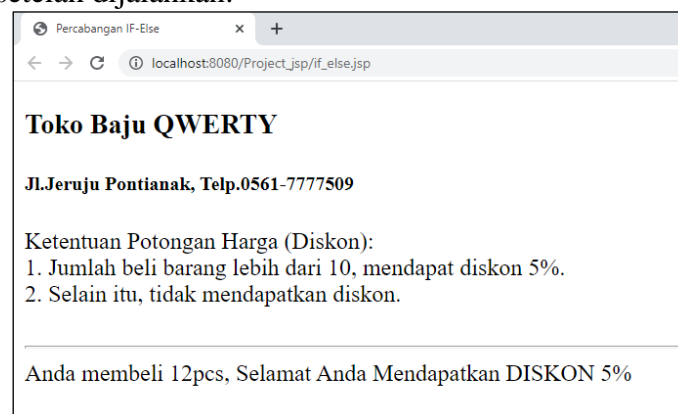
```
    pernyataan 2(statement);  
}
```

Prinsip kerja IF-Else adalah jika kondisi 1 bernilai true atau terpenuhi, maka akan dieksekusi pernyataan 1, namun jika kondisi 1 salah (false) maka pernyataan 2 yang akan dikerjakan.

Berikut contoh script percabangan IF-Else, simpan dengan nama file **if_else.jsp**.

```
<html>  
<head>  
    <title>Percabangan IF-Else</title>  
</head>  
<body>  
    <h3>Toko Baju QWERTY</h3>  
    <h5>Jl.Jeruju Pontianak, Telp.0561-7777509</h5>  
    Ketentuan Potongan Harga (Diskon):<br>  
    1.Jumlah beli barang lebih dari 10, mendapat diskon 5%.<br>  
    2.Selain itu, tidak mendapatkan diskon.<br><br>  
    <hr>  
    <%!  
        int jumlah_beli;  
    %>  
    <%  
        jumlah_beli=12;  
        if(jumlah_beli > 10)  
        {  
            out.println("Anda membeli " +jumlah_beli+ "pcs, Selamat Anda Mendapatkan DISKON 5%");  
        }  
        else  
        {  
            out.println("Pembelian Barang " +jumlah_beli+ "pcs belum mencapai syarat, Maaf Anda Tidak Mendapatkan DISKON");  
        }  
    %>  
</body>  
</html>
```

Hasil script di atas setelah dijalankan.



3. IF Majemuk (IF, Elseif, Else)

Percabangan ini digunakan apabila dalam sebuah studi kasus memiliki lebih dari 2 kondisi dan pernyataan. Penulisannya sebagai berikut.

```
if (kondisi 1)
{
    pernyataan 1;
}
elseif (kondisi 2)
{
    pernyataan 2;
}
elseif (kondisi 3)
{
    pernyataan 3;
}
elseif (kondisi 4)
{
    pernyataan 4;
}
else
{
    pernyataan n;
}
```

Prinsip kerja IF-Elseif-Else di atas hampir sama dengan percabangan IF-Else, hanya saja percabangan ini lebih banyak pilihan kondisi dan pernyataan yang akan dieksekusi. Berikut contoh script percabangan di atas.

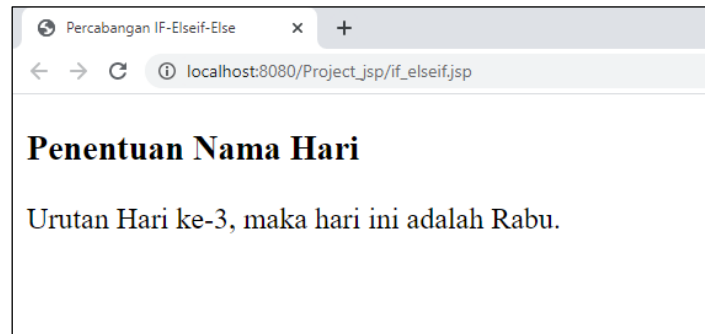
```
<html>
<head>
<title>Percabangan IF-Elseif-Else</title>
</head>
<body>
<h3>Penentuan Nama Hari</h3>
<%
String hari = "3";

if(hari == "1")
    out.println("Urutan Hari ke-"+hari+ " , maka hari ini adalah Senin.");
else if (hari == "2")
    out.println("Urutan Hari ke-"+hari+ " , maka hari ini adalah Selasa.");
else if (hari == "3")
    out.println("Urutan Hari ke-"+hari+ " , maka hari ini adalah Rabu.");
else if (hari == "4")
    out.println("Urutan Hari ke- "+hari+ " , maka hari ini adalah Kamis.");
else if (hari == "5")
    out.println("Urutan Hari ke- "+hari+ " , maka hari ini adalah Jumat.");
else if (hari == "6")
    out.println("Urutan Hari ke- "+hari+ " , maka hari ini adalah Sabtu.");
else if (hari == "7")
    out.println("Urutan Hari ke- "+hari+ " , maka hari ini adalah Ahad.");
```

```

else
    out.println("Urutan Hari ke- "+hari+ ", Tidak Tersedia!");
%>
</body>
</html>

```



4. IF Bersarang

IF bersarang atau disebut juga Nested IF merupakan struktur percabangan yang paling kompleks karena merupakan kombinasi dari berbagai struktur IF lainnya. Biasa dikatakan ada IF di dalam IF. Berikut bentuk penulisannya.

```

if (kondisi 1)
{
    if (kondisi 1a){
        Statement jika kondisi 1 dan 1a terpenuhi;
    }
    elseif (kondisi 1b){
        Statement jika kondisi 1 dan 1b terpenuhi;
    }
    else{
        Statement jika hanya kondisi 1 terpenuhi;
    }
}
else{
    Statement jika_semua_kondisi tidak terpenuhi;
}

```

Berikut contoh sederhana penerapan percabangan nested-if.

```

<html>
<head>
<title>Latihan Percabangan</title>
</head>
<body>
<h3>Percabangan IF-Nested</h3>
<%!
    int nilai = 10;
%>
<%
    if (nilai == 10) //if yang pertama

```

```

{
    out.println("Diketahui: <br>");
    out.println("Nilai yang dipilih = "+nilai+ "<br>");
    out.println("Nilai Pembanding = 15 <br><br>");
    out.println("MAKA : <br>");

    if (nilai < 15) //if yang ke2, if di dalam if
        out.println("Nilai " +nilai+ " lebih kecil dari 15");
    else
        out.println("Nilai " +nilai+ " lebih besar dari 15");
}
else
{
    out.println("Nilai yang Dipilih Tidak Sesuai Syarat!");
}
}
%>
</body>
</html>

```

Hasil running script di atas adalah seperti berikut.



5. Switch-Case

Percabangan Switch-Case pada Java hanya mengenal perbandingan, yaitu sama dengan (==). Mirip dengan kondisi IF–Elseif–Else, Switch-Case juga memberikan banyak alternatif pilihan pernyataan, namun disini menggunakan *break* untuk mengakhiri pernyataan. Berikut bentuk percabangan Switch-Case.

```

switch (variabel) {
    case nilai_1 :
        pernyataan 1;
break;
    case nilai_2 :
        pernyataan 2;
break;
    case nilai_3 :
        pernyataan 3;

```

```
break;
default:
    pernyataan.n;
}
```

Berikut contoh sederhana penerapan percabangan switch-case.

```
<html>
  <head>
    <title>Latihan Percabangan</title>
  </head>
  <body>
    <h3>Penentuan Hari dengan Switch-Case</h3>
    <%!
      int hari = 4;
    %>
    <%
      switch(hari)
      {
        case 1:
          out.println("Hari ke-" +hari+" adalah Senin.");
          break;
        case 2:
          out.println("Hari ke-" +hari+" adalah Selasa.");
          break;
        case 3:
          out.println("Hari ke-" +hari+" adalah Rabu.");
          break;
        case 4:
          out.println("Hari ke-" +hari+" adalah Kamis.");
          break;
        case 5:
          out.println("Hari ke-" +hari+" adalah Jumat.");
          break;
        case 6:
          out.println("Hari ke-" +hari+" adalah Sabtu.");
          break;
        case 7:
          out.println("Hari ke-" +hari+" adalah Ahad.");
          break;
        default:
          out.println("Hari Tidak Tersedia!");
      }
    %>
  </body>
</html>
```

Hasil eksekusi script percabangan di atas.



3.8. Perulangan

Perulangan atau biasa dikenal dengan istilah *Looping* ini adalah konstruksi pemrograman dasar untuk menangani tugas dengan menjalankan kode tertentu berulang kali. Prinsip kerja dari *looping*, sebuah perintah/pernyataan akan dieksekusi berulang kali hingga sesuai dengan kondisi tertentu yang telah ditetapkan. Perulangan terbagi menjadi 3 tipe dasar di pemrograman Java, yaitu:

1. For

For merupakan pernyataan perulangan (*looping*) dasar yang sangat umum digunakan. Setiap kali melalui *looping*, indeks *looping* akan memiliki nilai yang berbeda. Dan Anda dapat menggunakannya untuk mengakses data yang berbeda dalam kumpulan data, seperti penggunaan array. Berikut bentuk umum penulisan *looping*.

```
for (nilai awal; kondisi perulangan; iterasi)
{
    statement/pernyataan;
}
```

Penjelasan:

- Pemberian nilai awal (*initialization_expression*) dijalankan sebelum *looping* dieksekusi pertama kali.
- Kondisi perulangan (*end condition*) akan diperiksa setiap kali melalui *looping*, dan ketika nilainya menjadi false maka *looping* berakhir.
- Iterasi (*iteration_expression*) dijalankan setelah setiap kali melalui *looping* dan biasanya berupa operator *increment/decrement*.

Script untuk perulangan for dengan nama **contoh_for.jsp**.

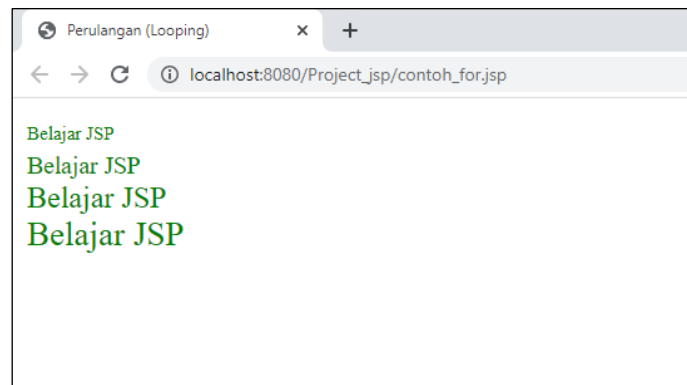
```
<html>
<head>
<title>Perulangan (Looping)</title>
</head>
```

```

<body>
<%!
    int cobafont;
%>
<%
for (cobafont = 1; cobafont <=4; cobafont++)
{
%>
    <font color="green" size="<%= cobafont %>">
        Belajar JSP
    </font><br />
<%
}
%>
</body>
</html>

```

Berikut adalah hasil perulangan for setelah dijalankan.



2. While

Perulangan while melakukan pengecekan kondisi di awal blok perulangan. Selama kondisi bernilai true atau terpenuhi, maka perulangan akan terus berlanjut dan sebaliknya jika kondisi tidak terpenuhi lagi (false) maka perulangan akan dihentikan. Berikut bentuk umum dari while.

```

while (kondisi perulangan)
{
    statement/pernyataan;
}

```

Berikut contoh penggunaannya.

```

<html>
<head>
    <title>Perulangan While</title>
</head>
<body>
<%
    int nilai = 1;
    while (nilai <=10)
    {
        out.println("Angka yang dihasilkan : " + nilai++ + "<br>");
    }
%>

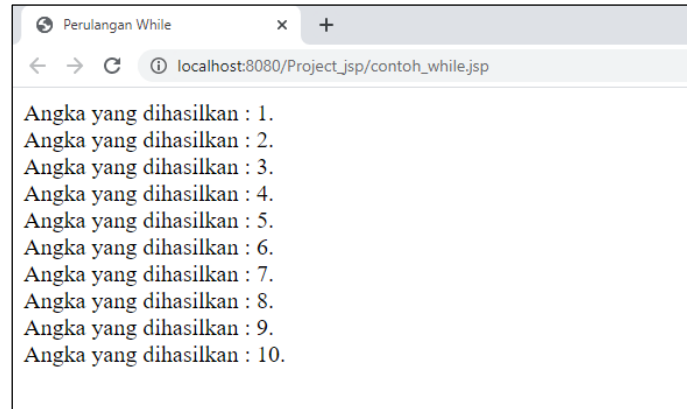
```

```

}
%>
</body>
</html>

```

Hasil setelah program while dijalankan adalah.



3. Do ... While

Perulangan do-while hampir sama seperti while, hanya saja pada perulangan do-while ini melakukan proses pengecekan kondisi di akhir blok perulangan setelah perintah eksekusi terlebih dahulu. Berikut bentuk penulisan do-while.

```

do
{
    statement/ Pernyataan;
}
while (kondisi perulangan);

```

Berikut contoh penggunaannya.

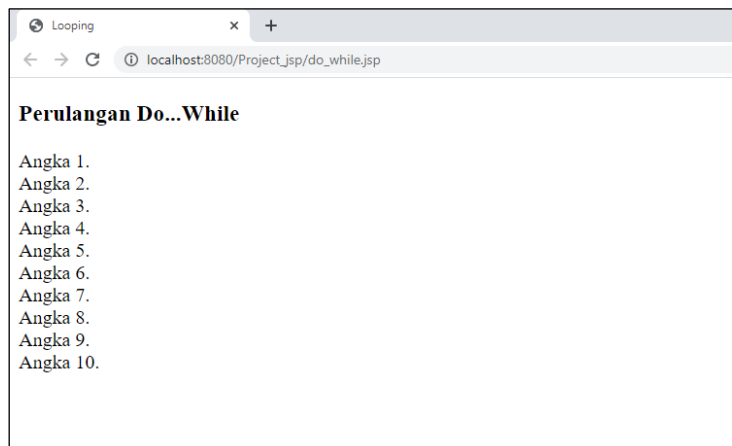
```

<html>
<head>
<title>Looping</title>
</head>
<body>
<h3>Perulangan Do...While</h3>
<%
int i = 1;
do
{
    out.println ("Angka " +i+ ".<br>");
    i = i + 1;
}

while(i <= 10);
%>
</body>
</html>

```

Hasil setelah program do...while dijalankan.



DAFTAR REFERENSI

1. Budiharto, Widodo. Membangun Aplikasi Web Handal dengan J2EE dan MVC. Andi Yogyakarta. 2013.
2. JSP tutorialpoints, www.tutorialpoints.com
3. Murach, Joel dan Andrea Steelman. Java Servlets & JSP, 2008.
4. Sidik, Betha dan Husni I.Pohan. Pemrograman Web dengan HTML. Informatika Bandung. 2014.