

4

Pengenalan J2EE

Objektif :

- **Mengenal J2EE**
 - **Mengetahui komponen teknologi web**
 - **Mengetahui cara kerja aplikasi berbasis web**
 - **Mengenal lingkungan pemrograman web**
-

4.1. Definisi J2EE

Jika anda mempelajari pemrograman Java, mungkin anda pernah mendengar tentang beberapa versi platform yang ada pada *Software Development Kit* (SDK) bahasa pemrograman Java. Beberapa versi platform tersebut diantaranya adalah J2ME, J2SE, dan J2EE. Pada kesempatan kali ini kita akan mempelajari mengenai apa itu J2EE serta bagaimana penerapan J2EE. J2EE sendiri merupakan sebuah edisi platform dari bahasa pemrograman Java yang berstandar untuk menyelesaikan kompleksitas masalah pada perusahaan. J2EE menyediakan API (Application Program Interface) dan Run Time Environment untuk mengembangkan dan menjalankan perangkat lunak multiple network yang dapat menggunakan web service, secure network application, dan aplikasi skala besar lainnya. Beberapa API yang disediakan oleh J2EE diantaranya adalah API untuk berkomunikasi dengan web service dan untuk mengembangkan aplikasi berbasis web.

Pada zaman sekarang, kebutuhan sebuah perusahaan untuk memiliki sebuah aplikasi bukan hanya terbatas pada lingkup internal saja. Tetapi sering kali juga dibutuhkan aplikasi untuk melakukan promosi serta pelayanan perusahaan kepada konsumen. Dan salah satu aplikasi yang umum digunakan untuk hal tersebut adalah sebuah aplikasi berbasis website. Bukan hal yang asing rasanya dan sudah merupakan bagian hidup dari masyarakat modern untuk mengakses sebuah website dan menggunakan sebuah website. Pada kursus kali ini kita akan membahas secara khusus mengenai J2EE dalam mengembangkan sebuah aplikasi berbasis web. Hal yang akan dibahas diantaranya adalah bagaimana teknologi website itu bekerja dan bagaimana mengembangkan aplikasi berbasis web dengan J2EE.

4.2. Pendahuluan Aplikasi Berbasis Web

Sebelum kita melangkah lebih jauh terlebih dahulu kita harus mengetahui 2 jenis kategori website yang ada, yaitu web statis dan web dinamis. Kategori pertama adalah web statis, yaitu sebuah jenis web yang mana pengguna tidak bisa mengubah konten dari web tersebut secara langsung menggunakan browser. Interaksi yang terjadi antara pengguna dan server hanyalah seputar pemrosesan link saja. sebuah web statis sering kali hanya merupakan sebuah text yang diatur dengan mark-up language yang bernama HTML. Pada web statis tidak terdapat database dan sebuah pemrograman yang bertugas untuk mengolah konten pada halaman website.

Dan kategori kedua merupakan web dinamis, yaitu sebuah web dimana dapat terjadi interaksi antara pengguna dan server. Seseorang dapat mengubah konten dari halaman tertentu dengan menggunakan browser. Pada sebuah web dinamis terdapat sebuah pemrograman khusus yang dapat berjalan disisi server dan akan membuat konten pada web kita dapat berubah-ubah. Hal ini membuat sebuah web dinamis harus memiliki server untuk menjalankannya.

Jika anda pada saat ini belum mengerti mengenai web statis maupun web dinamis anda tidak perlu khawatir. Untuk saat ini hal yang perlu ditekankan adalah anda mengerti perbedaan dari web statis dan web dinamis. Serta poin bahwa web statis kontennya hanya statis dan untuk web dinamis kontennya dapat berubah-ubah. Semoga nanti ketika kita masuk kedalam pemrograman web itu sendiri, kita akan lebih memahaminya.

4.3. Komponen Teknologi Web

Teknologi web tidaklah dapat dijalankan oleh sebuah perangkat saja. Namun didalamnya harus terdapat beberapa komponen yang telah ditentukan yang disusun menggunakan topologi client server. Sebuah teknologi web merupakan teknologi yang berdiri diatas sebuah jaringan komputer. Dimana terdapat beberapa komponen yang memiliki tugas dan posisinya masing-masing. Beberapa komponen dalam teknologi web tersebut diantaranya yang akan dibahas adalah:

- Web client
- Protokol
- Web Server
- HTML

Dan untuk lebih jelasnya berikut merupakan penjelasannya serta fungsi dari masing-masing komponen:

1. Web Client

Web client merupakan sebuah aplikasi yang digunakan untuk mengirim request dan menerima respon dari web server. Jenis dari web client yang paling umum digunakan adalah web browser. Web browser digunakan untuk mengakses sebuah halaman web.



Gambar 4.1 – Lima web browser populer

Web browser mengakses sebuah website berdasarkan dari URL dari website itu sendiri. URL (Uniform Resource Locator) sendiri adalah sebuah alamat unik yang dirancang untuk mengakses sebuah halaman website. Sebuah URL terdiri dari beberapa bagian fragmentasi dengan fungsinya masing-masing yang berbeda. Berikut merupakan contoh serta penjelasan fragmentasi URL:

- Contoh URL:
 - `http://www.youtube.com/watch?v=YYqzz1dy3Ak`
- Penjelasan:
 - `http://` : protokol yang digunakan
 - `www.youtube.com` : domain name
 - `watch` : path file pada server
 - `?v= YYqzz1dy3Ak` : request dari client yang berupa method GET

2. Protokol

Protokol merupakan sebuah jalan tempat dimana lalu lintas komunikasi antara server dan client terjadi. Protokol untuk web yang standar adalah protokol HTTP. Protokol HTTP menjadi jembatan dimana client dapat melakukan request dan menyediakan jenis request apa saja yang dapat dilakukan didalamnya. Yang paling umum dilakukan oleh client dalam melakukan proses request ke pada server adalah method GET dan method POST. Masing-masing dari method tersebut memiliki keunggulan masing-masing. Berikut merupakan beberapa perbedaan antara method GET dengan method POST :

Tabel 4.1. Perbedaan method GET dan POST

Method GET	Method POST
keterbatasan panjang request yang dilakukan	Lebih panjang
hal yang direquest terlihat pada address bar	Hal yang di-request tidak terlihat pada browser
kurang aman	Lebih aman
Lebih cepat diproses	Lebih lambat diproses

3. Web Server

Web server merupakan server khusus web yang bertugas untuk menerima request dari client dan merespon permintaan dari web client. Pada web server terdapat beberapa komponen penting yang mendukung agar sebuah website dapat diolah dengan baik. Diantara komponen tersebut adalah web server application, helper application, dan database. Masing-masing dari komponen tersebut memiliki tugas masing-masing, berikut merupakan penjelasannya:

- Web server application

Berfungsi untuk menerima HTTP request dan mengirimkan respon kembali kepada user. Web server application memiliki kelemahan dengan tidak bisa dalam mengolah konten web dinamis. Beberapa orang menyebut juga web server application sebagai HTTP server.
- Helper application

Berfungsi untuk mengolah konten web dinamis serta berkomunikasi dengan database. Dalam mengembangkan aplikasi web dengan J2EE kita akan banyak berkuat pada bagian Helper Application.

- Database
Berfungsi untuk tempat penyimpanan data.

4. HTML

HTML atau hypertext markup language merupakan sebuah markup language yang digunakan untuk menyusun sebuah halaman web. HTML digunakan untuk menyusun struktur tampilan pada website. HTML akan dikirimkan oleh webserver application dan di-render pada browser untuk dijadikan tampilan pada web browser. Pada web statis HTML sangat mendominasi dan hampir menjadi satu-satunya bahasa pemrograman/scripting yang digunakan. Penulisan HTML terdiri atas tag-tag tertentu yang memiliki fungsi dan maksud masing-masing. Setiap tag memiliki pembuka dan Penutup. Penutup tag pada HTML ditulis menggunakan tanda / setelah kurung buka. Penulisan tag HTML juga harus menggunakan kata yang sama persis, termasuk besar kecilnya huruf. Jadi, apabila anda menulis <body> dengan huruf kecil semua maka anda harus menutupnya dengan </body> HTML ditulis dengan struktur yang telah distandarisasi oleh w3 konsorsium. Berikut merupakan struktur standar HTML versi 5:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title dari HTML</title>
  </head>
  <body>
  </body>
</html>
```

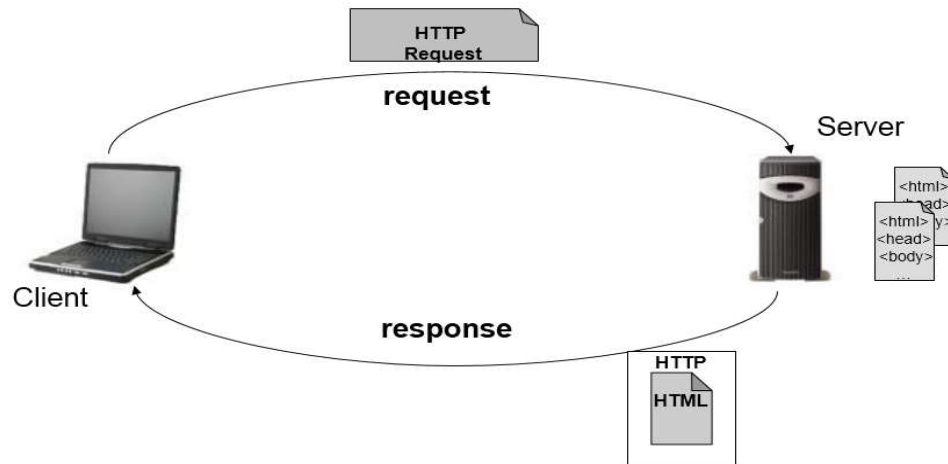
Kode HTML diatas tidak akan memberikan tampilan apapun pada browser. Satu-satunya bagian yang akan berubah hanya bagian title bar pada bagian atas browser yang namanya akan berubah. Terdapat banyak tag-tag standar yang telah distandarisasikan pada HTML. Berikut diantaranya beberapa tag yang sering digunakan dalam HTML dan penjelasannya:

Tabel 4.2. Tag – tag HTML

Tag HTML	Contoh Penggunaan	Penjelasan
<p>	<p>Hello World</p>	Membuat paragraf
 	Hello World	Memberikan baris baru
<hr>	Hello World <hr>	Memberikan garis baru
<h1>	<h1> Hello World</h1>	Memformat text menjadi judul tipe 1 (terbesar)
<h2>	<h2> Hello World</h2>	Memformat text menjadi judul tipe 2
<h3>	<h3> Hello World</h3>	Memformat text menjadi judul tipe 3
<h4>	<h4> Hello World</h4>	Memformat text menjadi judul tipe 4
<h5>	<h5> Hello World</h5>	Memformat text menjadi judul tipe 5
<h6>	<h6>Hello World</h6>	Memformat text menjadi judul tipe 6 (terkecil)
<title>	<title>Web Pertama Ku</title>	Memberi title pada web page
<i>	<i>Hello World</i>	Memiringkan text
<u>	<u>Hello World</u>	Memberi garis bawah pada text
	Hello World	Menebalkan text
<small>	<small>Hello World</small>	Mengecilkan text
<div>	<div>Hello World</div>	Mengelompokkan text atau bagian dari web

4.4. Cara kerja Aplikasi Berbasis web

Sebelumnya kita telah membahas komponen apa saja yang ada pada teknologi website serta fungsinya. Sekarang kita akan merangkai komponen tersebut agar kita mengetahui bagaimana proses yang terjadi didalamnya. Sebelum menjelaskan secara detail bagaimana proses yang terjadi, ada baiknya anda melihat terlebih dahulu ilustrasi pada gambar 1.2 supaya anda mendapatkan bayangan bagaimana proses yang terjadi.



Gambar 4.2 – Proses dalam teknologi website secara global

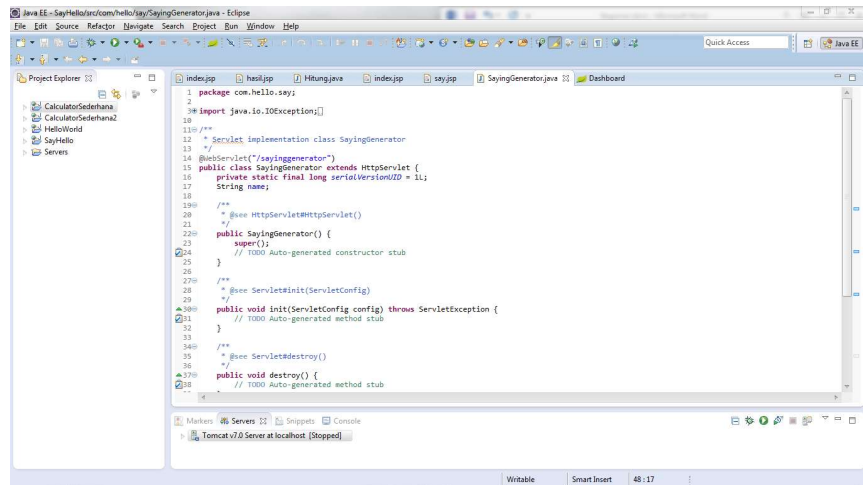
Setelah anda memperhatikan gambar 1.2 mari lihat rincian prosesnya dengan detail pada yang berikut ini:

- Diawali dengan web client mengirim permintaan kepada server melalui HTTP, proses request tersebut berbentuk sebagai HTTP request
- HTTP request diterima oleh server dan lalu diolah di dalam web server
- Selanjutnya web server akan mengirim kembali respon atas apa yang diminta oleh web client melalui HTTP kembali. Respon yang dikirim oleh web server berupa umumnya berupa HTML statis, atau tipe konten lainnya.

Untuk bagian proses pengolahan request pada server, akan lebih detail dijelaskan pada bab berikutnya.

4.5. Lingkungan Pemrograman

Pada modul ini kita akan menggunakan IDE Eclipse sebagai teks editor dan tempat dimana kita akan menjalankan web server. Web sever yang akan digunakan adalah web server Tomcat versi 7. Sedang untuk versi JDK yang digunakan adalah versi 1.7.0_40. Semua aplikasi yang digunakan dapat anda unduh secara gratis pada masing-masing web official software tersebut.



Gambar 4.3 – Tampilan IDE Eclipse



5

Servlet

Objektif :

- Mengetahui apa itu servlet
 - Mengetahui arsitektur web server dan servlet
 - Mengetahui kegunaan dan cara kerja servlet
 - Mampu membuat program servlet sederhana
-

5.1. Definisi Servlet

Servlet merupakan sebuah class pada Java yang dapat digunakan untuk membuat aplikasi servlet. Aplikasi servlet digunakan untuk menerima request dan merespon HTTP. Pada bab sebelumnya kita menyebut servlet dengan *helper application*, karena sifatnya memang berguna untuk membantu *web server application*. Sebuah program servlet ditulis dengan ekstensi standar java yaitu .java.

Peranan sebuah servlet adalah sebagai pembantu dari web server application dalam mengolah layanan web dinamis. Tetapi servlet sendiri juga tidak bisa berjalan begitu saja, karena apabila anda perhatikan pada servlet tidak terdapat main method seperti program java pada umumnya. Untuk itu pada web server perlu terdapat sebuah komponen yang akan menangani jalannya servlet. Komponen untuk membantu servlet berjalan tersebut disebut dengan container. Container juga bertanggung jawab dalam membuat thread untuk menangani HTTP request yang datang pada waktu yang bersamaan. Setiap satu HTTP request yang datang container akan membuat sebuah thread baru agar semua proses bisa berjalan secara simultan.

5.2. Arsitektur Web Server dan Servlet

Sebelumnya kita telah mengetahui bahwa ada sebuah container yang bertugas untuk membantu jalannya program servlet pada web server. Lalu bagaimanakah susunan komponen-komponen tersebut pada sebuah web server? Hal ini akan kita bahas pada sub bab ini.

Sebuah web server secara umum dapat kita bagi menjadi 4 bagian besar yaitu: web server application, container, servlets, dan database. Pada bab sebelumnya kita telah menyinggung dan memberikan beberapa penjelasan bagi beberapa fungsi komponen pada web server. Dan untuk fungsi lebih detailnya, anda dapat melihatnya pada yang berikut ini:

- a. Web server application
 - Menerima HTTP request
 - Men-forward HTTP request dari client kepada container
 - Mengirim response yang telah diolah maupun HTML statis

b. Container

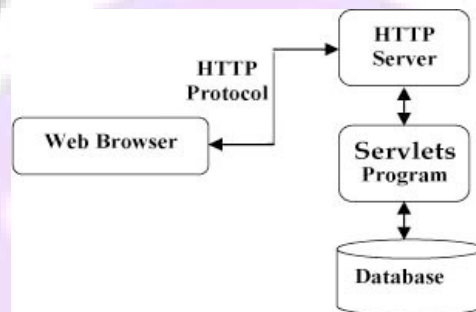
- Mengatur life cycle dari program servlets
- Communication support yaitu menerjemahkan HTTP request agar dimengerti servlet serta menerjemahkan sebaliknya
- Manajemen multi-threading yaitu menciptakan thread baru apabila terdapat client baru yang me-request layanan dari web dinamis
- Security yaitu salah satunya membuat letak direktori dari program servlet sendiri tidak mudah terdeteksi oleh user
- JSP support (akan dibahas pada bab berikutnya)

c. Servlets

- Enkapsulasi method pada HTTP
- Pusat dari logika bisnis aplikasi berbasis web pada java (program utama)

d. Database

- Tempat menyimpan data

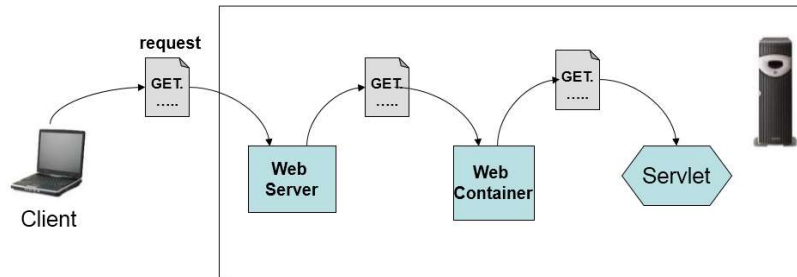


(*) container berada tepat pada servlets program

Gambar 5.1 - Arsitektur pada web server

5.3. Cara Kerja Servlet

Servlet hanya akan bekerja ketika sebuah HTTP server membutuhkan pengolahan web yang dinamis dan membutuhkan database. Sebuah servlet mulai bekerja ketika web server application / HTTP server menerima dan meneruskan HTTP request kepada container. Container pada saat ini bertugas untuk menerjemahkan HTTP *request* menjadi *request* yang dapat dimengerti oleh program servlet. Setelah proses konversi request tersebut, container memanggil program servlet yang manakah yang dibutuhkan pada proses request tersebut dan menjalankannya. Lalu program servlet akan mengolah request dinamis apa yang diinginkan oleh web client. Dan hasil dari proses pengolahan tersebut akan dikirim kembali dari container ke web server application. Dan pada akhirnya, web server application akan mengirimkan respon hasil pengolahan tersebut kepada web client melalui HTTP. Umumnya apabila request tersebut menginginkan tipe file berupa sebuah halaman website, maka akan direspon dengan mengirimkan website statis / HTML.



Gambar 5.2 – Ilustrasi pada proses pada web server

5.4. Method pada Servlet

Program servlet terdiri atas beberapa method yang memiliki fungsi dan tugas masing-masing yang dapat berasosiasi dengan web container. Dan untuk beberapa method diantaranya adalah ada yang bertugas untuk mewakili siklus hidup dari sebuah program servlet. Beberapa method tersebut adalah:

- **Method init**
Method init merupakan sebuah method yang akan dijalankan pada saat awal program servlet dijalankan oleh container. Digunakan untuk menginisialisasi proses yang akan dikerjakan pada method lainnya pada servlet.
- **Method service**
Method service merupakan method override dari semua method enkapsulasi dari method HTTP request. Method service akan secara otomatis menentukan apakah bentuk dari method apa yang digunakan dalam proses request. Method service berjalan setelah proses dari method init dijalankan oleh container.
- **Method destroy**
Method destroy adalah sebuah method yang akan dijalankan oleh container setelah semua proses pengolahan request yang diminta telah selesai. Method destroy akan otomatis menghancurkan program dan memasukkannya ke garbage collection setelah program servlet berjalan.

Selain method yang telah kita bahas terdapat juga method yang merupakan abstraksi method dari method HTTP. Jika anda pernah mengembangkan aplikasi berbasis web dengan menggunakan bahasa pemrograman seperti PHP, mungkin anda pernah mendengar serta menggunakan method POST dan GET. Pada servlet anda diberi kemampuan lebih dalam melakukan penggunaan method-method tersebut. Kita tidak hanya dapat menggunakan method POST dan GET tetapi lebih banyak dari itu. Berikut merupakan list method HTTP yang dapat kita gunakan pada servlet serta method apa yang akan menangani request tersebut:

- POST - doPost
- GET - doGet
- PUT - doPut
- HEAD - doHead

- DELETE - doDelete
- OPTION - doOptions
- TRACE - doTrace

Untuk method yang akan kita pelajari pada kesempatan kali ini terlebih dahulu kita hanya membahas 2 method yang paling umum, yaitu doGet dan doPost.

5.5. Implementasi Servlet

Untuk mengimplementasikan servlet pada pemrograman yang kita buat, terlebih dahulu anda perlu terlebih dahulu meng-*import* standard extension class dari `javax.servlet` dan `javax.servlet.http` ke program java Anda. `javax.servlet` sendiri merupakan sebuah class berisi yang berisikan *framework* dasar servlet dimana `javax.servlet.http` digunakan sebagai ekstensi dari framework servlet bagi servlet yang akan menjawab HTTP request. Dan pada class yang anda gunakan sebagai program servlet anda harus menggunakan `extends HttpServlet` agar dapat menggunakan semua kemampuan servlet pada java. Berikut merupakan contoh program sederhana yang dapat anda buat dengan menggunakan java servlet:

ServletExample.java

```
package com.example.tutorial;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/ServletExample")
public class ServletExample extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        PrintWriter out = response.getWriter();
```

```
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Aplikasi Hello World</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Pada contoh program diatas, kita menampilkan Hello World pada browser dan mengganti title bar pada browser menjadi Aplikasi Hello World. Untuk mengeluarkan tampilan pada browser menggunakan servlet terlebih dahulu harus dideklarasikan sebuah objek dari class `PrintWriter` (`PrintWriter out = response.getWriter()` ;). Objek yang kita deklarasikan tersebut bernama `out`, dan memiliki kemampuan untuk menggunakan method `println` yang dimiliki oleh class `PrintWriter`.

Review Singkat

Kita telah melihat bagaimana servlet diterapkan dan bagaimana kegunaannya, namun ada sedikit yang salah disini. Mengapa kita harus memasukkan sebuah html tag kedalam sebuah program java? Hal ini tidak lazim dan tidak mudah dilakukan. Terlebih apabila kita telah memiliki kode HTML yang sangat banyak dan beribu-ribu baris panjangnya. Web yang kita kembangkan akan menemukan masalah, yaitu sangat sulit untuk mendeteksi kesalahan dalam kode dan sulit untuk di-*maintenance*. Apa gunanya kita memiliki performa yang baik tetapi sulit untuk dipelihara dan digunakan? Tunggu dulu, kita belum melihat seluruh pengembangan aplikasi web dengan basis Java. Pada java terdapat sebuah teknologi yang memungkinkan kita untuk menulis bahasa pemrograman java didalam sebuah kode HTML. Kita mempelajarinya pada bab berikutnya.