



Pemrograman WEB

Versi 1.0
April 2007

<p><u>Pengarang</u> Joyce Avestro</p> <p><u>Team</u> Joyce Avestro Florence Balagtas Rommel Feria Reginald Hutcherson Rebecca Ong John Paul Petines Sang Shin Raghavan Srinivas Matthew Thompson</p> <p><u>Team Translator</u> Frans Thamura Eko Subiyantoro Cahya Kusuma Ratih Rizzatama N.S. Mir'atul Khusna Mufida Khansa Fitriannisa Sulistyaningtyas Fauzan Prasetyo E.P Allin Junikhah Bagus Chandra Badar Agung Nugroho Kadek Surya Pranata</p>	<p><u>Persyaratan-persyaratan untuk latihan laboratorium</u></p> <p><u>Sistem Operasi yang mendukung</u></p> <p>NetBeans IDE 5.5 berjalan pada sistem operasi yang mendukung Java VM. Dibawah ini merupakan daftar dari platform-platform yang digunakan:</p> <ul style="list-style-type: none"> • Microsoft Windows XP Professional SP2 atau yang terbaru • Mac OS X 10.4.5 atau yang terbaru • Red Hat Fedora Core 3 • Sistem Operasi Solaris™ 10 Update 1 (SPARC® and x86/x64 Platform Edition) <p>NetBeans Enterprise Pack juga dapat digunakan pada platform-platform berikut ini:</p> <ul style="list-style-type: none"> • Microsoft Windows 2000 Professional SP4 • Solaris™ 8 OS (SPARC and x86/x64 Platform Edition) and Solaris 9 OS (SPARC and x86/x64 Platform Edition) • Berbagai macam distribusi LINUX <p><u>Minimum konfigurasi Hardware</u></p> <p>Catatan: Minimum resolusi layar NetBeans IDE's adalah 1024x768 pixels.</p> <ul style="list-style-type: none"> • Sistem Operasi Microsoft Windows: <ul style="list-style-type: none"> ○ Prosesor: 500 MHz Intel Pentium III workstation atau setaranya ○ Memori: 512 MB ○ Disk space: 850 MB dari free disk space • Sistem operasi Linux: <ul style="list-style-type: none"> ○ Prosesor: 500 MHz Intel Pentium III workstation atau setaranya ○ Memori: 512 MB ○ Disk space: 450 MB dari free disk space • Solaris OS (SPARC): <ul style="list-style-type: none"> ○ Prosesor: UltraSPARC II 450 MHz ○ Memori: 512 MB ○ Disk space: 450 MB of free disk space • Solaris OS (x86/x64 Platform Edition): <ul style="list-style-type: none"> ○ Prosesor: AMD Opteron 100 Series 1.8 GHz ○ Memori: 512 MB ○ Disk space: 450 MB dari free disk space • Sistem Operasi Macintosh OS X: <ul style="list-style-type: none"> ○ Prosesor: PowerPC G4 ○ Memori: 512 MB ○ Disk space: 450 MB dari free disk space
---	---

	<p><u>Konfigurasi Hardware yang direkomendasikan</u></p> <ul style="list-style-type: none">• Sistem Operasi Microsoft Windows:<ul style="list-style-type: none">○ Prosesor: 1.4 GHz Intel Pentium III workstation atau setaranya○ Memori: 1 GB○ Disk space: 1 GB dari free disk space• Sistem Operasi Linux:<ul style="list-style-type: none">○ Prosesor: 1.4 GHz Intel Pentium III atau setaranya○ Memori: 1 GB○ Disk space: 850 MB dari free disk space• Solaris™ OS (SPARC®):<ul style="list-style-type: none">○ Prosesor: UltraSPARC IIIi 1 GHz○ Memori: 1 GB○ Disk space: 850 MB dari free disk space • Solaris™ OS (x86/x64 platform edition):<ul style="list-style-type: none">○ Prosesor: AMD Opteron 100 Series 1.8 GHz○ Memori: 1 GB○ Disk space: 850 MB dari free disk space• Sistem Operasi Macintosh OS X:<ul style="list-style-type: none">○ Prosesor: PowerPC G5○ Memori: 1 GB○ Disk space: 850 MB dari free disk space <p><u>Software yang diperlukan</u></p> <p>NetBeans Enterprise Pack 5.5 dapat diakses pada Java 2 Platform Standard Edition Development Kit 5.0 Update 1 atau di atasnya (JDK 5.0, versi 1.5.0_01 atau di atasnya), dimana berisi Java Runtime Environment plus developer tools untuk meng-compile, men-debug, and menjalankan aplikasi yang ditulis dengan bahasa Java. Sun Java System Application Server Platform Edisi 9 telah diuji dengan JDK 5.0 update 6.</p> <ul style="list-style-type: none">• Untuk Solaris, Windows, dan Linux, Anda dapat mendownload JDK untuk platform Anda pada http://java.sun.com/j2se/1.5.0/download.html• Untuk Mac OS X, memerlukan Java 2 Platform Standard Edition (J2SE) 5.0 Release 4. Anda dapat mendownload JDK dari Apple's Developer Connection site. Pada: http://developer.apple.com/java (untuk mendownload JDK Anda harus melakukan registrasi terlebih dahulu).
--	--

Daftar Isi

- 1 Pengenalan Pemrograman Web
 - 1.1 Mengapa harus dengan Web?
 - 1.1.1 Teknologi Neutral Environment
 - 1.1.2 Arsitektur Client Server
 - 1.1.2.1 Thick dan Thin Clients
 - 1.1.2.2 Arsitektur Client Server dari Perspektif Web
 - 1.1.3 J2EE Web Tier Overview
 - 1.1.3.1 Servlets
 - 1.1.3.2 Java Server Pages
 - 1.1.3.3 Containers
 - 1.1.3.4 Struktur Dasar dari Aplikasi Web
- 2 Servlets
 - 2.1 Pendahuluan
 - 2.1.1 Definisi
 - 2.1.2 Gambaran Arsitektur Servlet
 - 2.1.3 Servlet Lifecycle
 - 2.1.3.1 Instantiation
 - 2.1.3.2 Initialization
 - 2.1.3.3 Service
 - 2.1.3.4 Destruction
 - 2.1.3.5 Garbage Collection
 - 2.1.4 Output Generation
 - 2.1.4.1 Konfigurasi Aplikasi Web
 - 2.1.4.2 Packaging Aplikasi Web
 - 2.1.4.3 Meng-generate file War dari Enterprise Project yang ada
 - 2.1.5 Pengenalan Ant
 - 2.1.6 Deployment dalam Server
 - 2.2 Ringkasan
- 3 Pembahasan Servlet Lanjutan
 - 3.1 Pengalihan Respon
 - 3.1.1 RequestDispatcher
 - 3.2 Pembatasan Object-Object
 - 3.2.1 Menyimpan dan mendapatkan kembali data dari suatu bidang
 - 3.2.2 Contoh Skenario
 - 3.3 Session Tracking dan Pengaturan
 - 3.3.1 Cookies
 - 3.3.2 Penulisan kembali alamat URL
 - 3.3.3 Form Fields tersembunyi
 - 3.3.4 Session tracking dalam servlets
 - 3.3.5 Mendapatkan sebuah instance dari object HttpSession
 - 3.3.6 Menyimpan dan mendapatkan data dalam sebuah session

- 3.3.7 Menghapus data yang tersimpan session
 - 3.3.8 Terminasi Session
 - 3.3.9 Melakukan URL-Rewriting
 - 3.4 Filters
 - 3.4.1 Membuat sebuah filter
 - 3.4.2 Rantai Filter
 - 3.4.3 Konfigurasi Filter
- 4 Dasar JSP
 - 4.1 Pengenalan
 - 4.2 Tujuan
 - 4.2.1 Apakah JSP itu?
 - 4.2.2 Kenapa menggunakan JSP?
 - 4.2.3 Contoh JSP
 - 4.2.4 Menggunakan IDE Enterprise
 - 4.2.5 Menggunakan Build Tool
 - 4.2.6 Alur JSP
 - 4.3 Syntax JSP dan Semantics
 - 4.3.1 Elemen-elemen dan Data Template
 - 4.3.2 Dua Tipe Syntax
 - 4.3.3 Scripting Elements
 - 4.3.4 Scriptlets
 - 4.3.5 Expressi
 - 4.3.6 Deklarasi
 - 4.3.7 Variabel – Variabel yang telah dikenal
 - 4.3.8 JSP Directives
 - 4.3.8.1 Page Directives
 - 4.3.8.2 Include Directives
 - 4.3.8.3 Tag-Lib Directives
 - 4.4 JavaBeans pada JSP
 - 4.4.1 JavaBeans dihubungkan dengan JSP Action
 - 4.5 Penanganan Error
- 5 SQL dan JDBC
 - 5.1 Pengenalan
 - 5.1.1 SELECT
 - 5.1.2 INSERT
 - 5.1.3 UPDATE
 - 5.2 JDBC
 - 5.2.1 java.sql.DriverManager
 - 5.2.2 javax.sql.DataSource
 - 5.2.3 Konfigurasi Data Source pada Sun Application Server 8.1
 - 5.2.4 Mendaftarkan file JAR
 - 5.2.5 Membuat Sebuah Connection Pool
 - 5.2.6 Mendaftarkan DataSouce
 - 5.2.7 Retrieving DataSource
 - 5.2.8 java.sql.Connection / java.sql.Statement
 - 5.2.9 java.sql.ResultSet
- 6 JSP Lanjutan

- 6.1 Pendahuluan
- 6.2 Expression dalam JSP
 - 6.2.1 Sytanx EL
 - 6.2.2 Mengakses variabel scope dan properties
 - 6.2.3 Object Implisit EL
 - 6.2.4 Notasi []
 - 6.2.5 JSTL
 - 6.2.6 Tag-tag Custom
 - 6.2.7 Memasukkan JSTL dalam aplikasi kita
 - 6.2.8 Core
 - 6.2.9 General Purpose Tags
 - 6.2.10 Iterasi
 - 6.2.11 Kondisi
- 7 Pengenalan Arsitektur MVC
 - 7.1 Pengenalan Arsitektur Model-View-Controller
 - 7.1.1 Motivasi
 - 7.1.2 Solusi
 - 7.2 Model
 - 7.3 View
 - 7.4 Controller
 - 7.5 Arsitektur MVC Untuk Web = Arsitektur Model 2
 - 7.5.1 Model 2 Diagram
 - 7.5.2 Struts
 - 7.5.3 Controller
 - 7.5.3.1 ActionServlet
 - 7.5.3.2 Action
 - 7.5.3.3 ActionForm
 - 7.5.3.4 struts-config.xml
 - 7.5.4 Model
 - 7.5.5 View
 - 7.5.5.1 struts-html
 - 7.6 Memandang Hal-hal yang ada secara keseluruhan
- 8 Advanced MVC
 - 8.1 Pendahuluan
 - 8.2 DynaActionForms
 - 8.3 Validators
 - 8.3.1 Konfigurasi Validator Plugins
 - 8.3.2 validator-rules.xml
 - 8.3.3 validation.xml
 - 8.3.3.1 Mengkonfigurasi file validation.xml
 - 8.3.3.2 Mendefinisikan resource bundle
 - 8.4 Tiles

- 8.4.1 Mempersiapkan tiles
 - 8.4.2 Membuat layout template
 - 8.4.3 Membuat Screen Definitions
 - 8.4.4 Membuat definition menggunakan konfigurasi file XML
 - 8.4.5 Menggunakan Screen Definitions
 - 8.4.6 Menambahkan Definitions
- 9 JSF
- 9.1 Pengenalan JSF
 - 9.1.1 Controller
 - 9.1.1.1 FacesServlet
 - 9.1.1.2 Action Handlers
 - 9.1.1.3 Method Aplikasi
 - 9.1.1.4 ActionListener
 - 9.1.1.5 faces-config.xml
 - 9.1.2 Model
 - 9.1.3 View
 - 9.1.3.1 Integrasi JSF-JSP
 - 9.1.3.2 Mengikat Nilai
 - 9.1.3.3 Mendaftarkan Action Handlers ke Komponen View
- 10 Advanced JSF
- 10.1 Pendahuluan
 - 10.2 FacesContext
 - 10.2.1 FacesContext dan Component Tree
 - 10.2.2 FacesContext dan External Context
 - 10.3 Validator
 - 10.3.1 Validator Standard JSF
 - 10.3.2 Menggunakan Validator Standard
 - 10.3.3 Customized Validation
 - 10.3.3.1 Menggunakan Method Validasi External
 - 10.3.3.2 Membuat Method Validasi External
 - 10.3.3.3 Pemakaian Method
 - 10.3.4 Membuat implementasi Validator yang terpisah
 - 10.3.4.1 Register komponen sebuah Validator
 - 10.3.4.2 Menggunakan komponen validator
 - 10.3.4.3 Menambahkan atribut pada validator kita
 - 10.4 Converter
 - 10.4.1 DateTimeConverter
 - 10.4.2 NumberConverter
 - 10.4.3 CustomConverter
 - 10.4.4 Method getAsObject
 - 10.4.5 Method getAsString
 - 10.4.6 Menggunakan Custom Converter
- 11 Keamanan Web

- 11.1 Pendahuluan
- 11.2 SSL
 - 11.2.1 Mengaktifkan SSL pada aplikasi
 - 11.2.2 Certificates
 - 11.2.3 Membuat Certificate private key
 - 11.2.4 Membuat Certificate
 - 11.2.5 Mengatur Certificate
 - 11.2.6 Membuat Secure HTTP Listener
- 11.3 10 Celah Keamanan Pada Aplikasi Web

- 14 AJAX
 - 14.1 Pendahuluan
 - 14.2 AJAX
 - 14.2.1 AJAX sebagai sebuah teknologi
 - 14.2.2 AJAX sebagai sebuah arsitektur
 - 14.2.3 Proses yang dilakukan oleh AJAX
 - 14.2.4 Menyiapkan halaman form
 - 14.2.5 Membuat Sebuah Instance Object XMLHttpRequest
 - 14.2.6 Menggunakan object XMLHttpRequest untuk berkomunikasi dengan server
 - 14.2.7 Fungsi Callback

- 15 Design Pattern
 - 15.1 Pengenalan Design Pattern
 - 15.1.1 Apakah Design Pattern?
 - 15.1.2 Keuntungan Design Pattern?
 - 15.2 View Helper
 - 15.2.1 Masalah
 - 15.2.2 Solusi
 - 15.2.3 View Helpers memisahkan detail dari pembacaan data
 - 15.2.4 View Helpers menyimpan data model lanjutan
 - 15.2.5 View Helpers menyediakan atau memodifikasi data model yang telah ada
 - 15.3 Session Facade
 - 15.3.1 Masalah
 - 15.3.2 Solusi
 - 15.4 Data Transfer Object
 - 15.4.1 Masalah
 - 15.4.2 Solusi
 - 15.5 Data Access Objects
 - 15.5.1 Masalah
 - 15.5.2 Solusi

- 16 Studi Kasus Aplikasi

- 16.1 Ikhtisar Aplikasi
- 16.2 Membuat Domain Object
- 16.3 Membuat Model
 - 16.3.1 Bekerja pada use case Add new User
 - 16.3.2 Membuat implementasi facade untuk use case Add User
- 16.4 Mengimplementasikan UserDAO
 - 16.4.1 Penulisan Kode DAOFactory
 - 16.4.2 Bekerja pada use case yang berbeda
 - 16.4.3 Menggali fungsi umum lainnya
 - 16.4.4 Mengimplementasikan SeminarDAO
 - 16.4.5 Meringkas Sejauh Apa yang telah kita lakukan
- 16.5 Membuat Komponen View dan Controller
 - 16.5.1 Membuat halaman login
 - 16.5.2 Membuat ActionForm untuk halaman login
 - 16.5.3 Membuat action handler halaman login
 - 16.5.4 Berbagai aktifitas implementasi halaman login
 - 16.5.5 Menambahkan sebuah action forward
 - 16.5.6 Menambahkan masukan pada file properties ApplicationResources
 - 16.5.7 Menambahkan validasi LoginForm kita
 - 16.5.7.1 Validasi Sisi Client
 - 16.5.7.2 Menambahkan validasi sisi server
 - 16.5.8 View Seminar List Page
 - 16.5.9 Halaman Detail Seminar
- 16.6 Kesimpulan