

J2EE VS .NET PLATTFORM SOLUSI UNTUK WEB SERVICES

Muhammad Rudyanto Arief

Abstraksi

Script internet sudah mulai mengalami pergeseran dalam konsep dan teknologi. Ini semua tidak terlepas dari peranan dunia internet yang menawarkan banyak kemudahan bagi pelaku bisnis di dunia nyata. Dari yang hanya mendukung pemrograman web statis lalu muncul web dinamis yang melibatkan basis data di dalamnya. J2EE dan .NET muncul sebagai jawaban atas kebutuhan pelaku bisnis akan layanan web commerce yang mumpuni.

Kata Kunci : *script, J2EE, .NET, web services, Java*

Perkembangan bahasa pemrograman internet atau yang lebih sering disebut *script* sangat pesat. Hal ini tidak lepas dari pengaruh perkembangan teknologi jaringan internet beserta komunitas yang ikut didalamnya (*people*). Perkembangan *script* internet tidak terlepas pula dari berkembangnya konsep bisnis yang ada di dunia nyata "real world". Setelah internet diperkenalkan dengan segala kelebihan yang ada padanya maka para pelaku bisnis mulai melirik media ini sebagai salah satu metode untuk menjalankan transaksi bisnisnya. Untuk itu dibutuhkan pendekatan baru yang sesuai dengan teknologi dan konsep moder bisnis saat ini.

Lalu muncullah konsep bisnis e-commerce yang merupakan tindak lanjut dari munculnya internet. Untuk mendukung permintaan pasar akan teknologi ini maka para developer mulai membangun bahasa pemrograman yang berbasis internet. Generasi pertama *script* berbasis server mulai dimunculkan. Hal ini dengan adanya CGI (common gateway interface) yang merupakan generasi pertama *script* berbasis web yang berjalan pada sisi server. *Script* dalam pemrograman internet sendiri dapat dibagi menjadi 2 sisi yaitu pada sisi *client* dan pada sisi *server*.

Perbedaan keduanya sudah sangat jelas. Bahwa *script* yang digunakan pada sisi client yang berfungsi untuk membuat halaman web yang statis (tidak bisa berubah-ubah) contohnya adalah html, javascript sementara *script* yang digunakan pada sisi

server digunakan untuk membuat halaman web yang dinamis. Pendekatan-pendekatan baru digunakan untuk memenuhi permintaan dunia bisnis akan teknologi internet yang mendukung proses transaksi bisnis mereka. Mulai dari world wide web sampai saat ini muncul konsep web services saat ini.

Untuk membangun suatu web services tentunya harus didukung oleh kesiapan infrastruktur dan teknologi yang akan di implementasikan. Kalau dulu dikenal konsep client-server yaitu pembagian kerja antara client dengan server. Kita semua tahu bahwa untuk membangun sistem ini sangatlah mudah karena kita hanya membutuhkan aplikasi dan *script* yang bisa berjalan pada sisi client dan sisi server.

Pada sisi client kita hanya membutuhkan html untuk menampilkan halaman web yang statis lalu pada sisi servernya kita bisa menggunakan *script* berbasis server seperti CGI, PERL, ASP, PHP, XML lalu menambahkan database server sebagai tempat menyimpan datanya. Awalnya pendekatan client-server ini sudah dianggap cukup untuk menangani masalah ini tapi untuk saat ini pendekatan ini dirasa sudah tidak cukup lagi. Dengan adanya perusahaan yang secara full menerapkan aturan bisnis e-commerce untuk perusahaan mereka dan dengan berbagai macam transaksi yang terjadi tentunya dibutuhkan pendekatan baru untuk itu. Kali ini kita akan membandingkan 2 tools yang digunakan untuk membangun web services yaitu J2EE dan .NET.

Pendekatan J2EE dan .NET untuk web services

Jika kita ingin membangun sistem web services yang bermanfaat banyak hal yang harus dipertimbangkan. Web yang dibangun harus reliable, highly available, fault-tolerant, scalable, and must perform at acceptable levels. Hal ini tidak berbeda jika kita ingin membangun aplikasi bisnis lainnya.

J2EE dan .NET adalah perkembangan dari teknologi aplikasi berbasis server yang sudah ada sebelumnya yang digunakan untuk membangun aplikasi bisnis serupa. Versi awal dari teknologi ini ternyata tidak digunakan untuk kepentingan pembangunan aplikasi berbasis web services. Namun dengan munculnya web services konsep maka kedua perusahaan yang mengembangkan J2EE dan .NET mengambil posisi untuk ikut terlibat dalam memberikan solusi bagi konsep web services. Mereka menamakan konsep ini sebagai platform yang dapat digunakan untuk membangun web services.

J2EE

Platform Java 2, Enterprise Edition (J2EE) di desain untuk menyederhanakan masalah kompleks yang kita miliki berhubungan dengan pengembangan, deployment, dan manajemen dari solusi bisnis multi-tier. J2EE adalah suatu standarisasi industri, yang dikembangkan oleh Sun Microsystems.

Yang penting untuk kita ketahui adalah bahwa J2EE adalah sebuah standarisasi, bukan sebuah produk. Kita tidak dapat men-download J2EE. Tidak seperti kita men-download aplikasi-aplikasi yang lain. Sepanjang 2 pihak mematuhi kontrak/ perjanjian J2EE, maka aplikasi dapat dikembangkan sendiri dalam lingkungan yang berbeda-beda sesuai keinginan pengembang.

Tujuan dari komunitas J2EE adalah untuk memberikan pilihan pada konsumen untuk memilih vendor-vendor produk dan tool-tool dan untuk mendorong memilih produk yang terbaik untuk masing-masing konsumen dalam memenuhi persaingan dengan pesaingnya. Hal ini bisa terwujud hanya dengan cara jika semua industri menggunakan platform yang sama yaitu J2EE. Untuk menjamin pembeliannya Sun bekerja sama dengan beberapa vendor yang bekerja dibidang eBusiness, seperti BEA, IBM, dan Oracle, dalam membuat definisi yang jelas tentang J2EE. Sun kemudian berinisiatif dengan membuat komunitas java (Java Community Process) untuk mengumpulkan ide-ide untuk meningkatkan kemampuan J2EE. Alasan Sun melakukan ini adalah karena mereka harus melakukan ini untuk meraih sukses dalam J2EE ini. Cara terbaik untuk mengamankan ide-ide ini adalah dengan langsung terlibat dengan komunitas lain untuk menjelaskan ide ini.

Java: The foundation for J2EE

Arsitektur J2EE didasarkan pada bahasa pemrograman Java. Yang menarik dari Java adalah kemampuannya untuk membolehkan perusahaan untuk menulis kembali kode-kode program mereka sekali dan mengembangkan kode-kode tersebut kedalam platform manapun. Prosesnya adalah sebagai berikut :

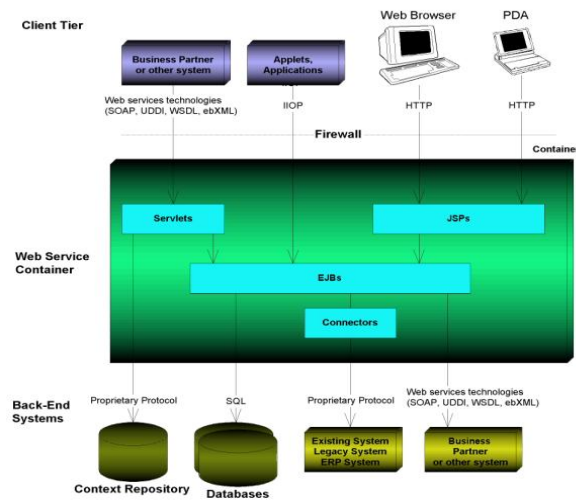
1. Pengembang menuliskan source code dengan menggunakan Java.
2. Kode Java di-compile menjadi bytecode, yang menjadi jembatan antar platform, setengah jalan antara source code dan bahasa mesin.
3. Ketika kode siap untuk dijalankan, Java Runtime Environment (JRE) menerjemahkan bytecode tersebut dan mengeksekusinya pada run-time.

J2EE adalah sebuah aplikasi dari Java. Komponen-komponen J2EE ditransformasikan menjadi bytecode dan dieksekusi oleh JRE pada saat runtime.

J2EE and Web Services

J2EE sendiri mempunyai sejarah pernah digunakan sebagai sebuah arsitektur untuk pengembangan aplikasi berbasis server dalam bahasa pemrograman Java. J2EE dapat digunakan untuk membangun website tradisional, komponen-komponen perangkat lunak, atau paket-paket aplikasi. J2EE telah memiliki "extended" yang disertakan untuk mendukung pengembangan web services berbasis XML dengan baik. Web services ini dapat beroperasi dengan web services-web services lainnya yang mungkin saja ditulis menggunakan standar J2EE.

Model pengembangan web services J2EE ditunjukkan di gambar 1 berikut :



Gambar 1 pengembangan web services dengan J2EE

J2EE application ditempatkan dalam sebuah "container", yang menyediakan kualitas layanan (qualities of service /QoS) yang dibutuhkan untuk aplikasi, seperti transaksi, keamanan, dan "persistence services".

The business layer melakukan pengolahan bisnis dan data logic. Untuk skala besar aplikasi J2EE, logika bisnisnya dibuat menggunakan komponen Enterprise JavaBeans (EJB). Lapisan ini melakukan pengelolaan data logic dan bisnis. Koneksinya ke basis data menggunakan Java Database Connectivity (JDBC) atau SQL/J, atau system yang sudah ada dengan menggunakan Java Connector Architecture (JCA). Juga dapat melakukan komunikasi/ koneksi ke rekanan bisnis menggunakan teknologi web services (SOAP, UDDI, WSDL, ebXML) melalui Java APIs untuk XML (the JAX APIs).

Business partners dapat melakukan koneksi dengan aplikasi J2EE melalui teknologi web services (SOAP, UDDI, WSDL, ebXML). Servlet, yang merupakan obyek request/response oriented Java object, dapat menerima request web services dari rekanan bisnis. Servlet menggunakan JAX APIs untuk melakukan operasi web services.

Traditional 'thick' clients seperti applets atau aplikasi berhubungan secara langsung ke lapisan EJB melalui Internet Inter-ORB Protocol (IIOP) tidak lagi melalui web services, sejak aplikasi client ditulis oleh perusahaan yang sama-sama menggunakan aplikasi J2EE dan untuk selanjutnya tidak memerlukan lagi kolaborasi web service berbasis XML.

Web browsers and wireless devices berhubungan ke JavaServer Pages (JSPs) yang melakukan rendering antarmuka pengguna dalam HTML, XHTML, atau WML.

Layanan Tambahan

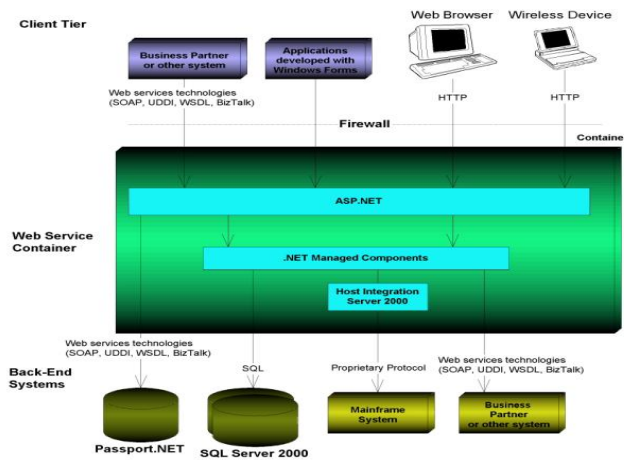
Semua vendor yang menawarkan platform J2EE telah menyediakan fitur-fitur tambahan yang tidak terdapat dalam J2EE standar. Beberapa dari mereka bahkan menerapkan konsep portability, menyediakan komponen-komponen e-commerce, atau bahkan integrasi B2B secara menyeluruh.

IV. Microsoft's .NET Platform

Microsoft.NET[®] adalah sebuah produk yang di desain untuk membolehkan perusahaan membangun web services smart, enterprise-class. Satu hal yang perlu diingat adalah : .NET adalah sebuah produk strategi sementara J2EE adalah sebuah standarisasi untuk produk yang menggunakannya.

Microsoft.NET secara garis besar adalah Windows DNA yang ditulis ulang, yang merupakan platform yang digunakan oleh Microsoft's sebelumnya untuk membangun aplikasi enterprise. Windows DNA termasuk dalam banyak teknologi yang sudah terbukti dan diproduksi oleh Microsoft hari ini, termasuk didalamnya Microsoft Transaction Server (MTS) dan COM+, Microsoft Message Queue (MSMQ), dan database Microsoft SQL Server..NET Framework yang terbaru menggantikan teknologi ini dan memasukkan lapisan web services sebaik dukungannya terhadap bahasa pemrograman yang ada.

Model pengembangan untuk web services dengan Microsoft.NET ditunjukkan gambar 2 berikut :



The **.NET application** ditempatkan dalam sebuah "container", yang menyediakan "qualities of service" yang diperlukan oleh aplikasi enterprise seperti transaksi, keamanan dan "messaging services".

The **business layer** dari aplikasi .NET dibangun menggunakan komponen-komponen .NET managed. Lapisan ini melakukan pengelolaan data logic dan bisnis. Koneksinya ke database menggunakan Active Data Objects (ADO.NET) and system yang sudah ada menggunakan layanan yang disediakan oleh Microsoft Host Integration Server 2000, seperti COM Transaction Integrator (COM TI). Juga dapat melakukan koneksi ke rekanan bisnis menggunakan teknologi web services (SOAP, UDDI, WSDL).

Business partners dapat melakukan koneksi dengan aplikasi .NET melalui teknologi web services (SOAP, UDDI, WSDL, BizTalk).

Traditional 'thick' clients, web browsers, wireless devices melakukan koneksi ke Active Server Pages (ASP.NET) yang me-render user interfaces dalam HTML, XHTML, atau WML. user interfaces yang lebih kompleks dibangun menggunakan tampilan Windows.

The .NET Framework

Microsoft.NET menawarkan language-independence dan language-interoperability. Yang berarti bahwa kita bebas untuk membangun aplikasi sesuai bahasa yang kita gunakan dan masing-masing dapat berdiri sendiri tanpa bergantung satu sama lain. Tidak peduli bahasa pemrograman apa yang kita gunakan untuk membangun aplikasi maka .NET dapat menangani masalah tersebut. Ini merupakan salah satu aspek mendasar yang dimiliki oleh platform .NET. Satu komponen .NET dapat ditulis, contohnya sebagian menggunakan VB.NET, versi visual basic dari .NET, dan sebagian lagi menggunakan C#, yang merupakan bahasa pemrograman berorientasi obyek terbaru yang dimiliki oleh Microsoft.

How does this work?

Pertama, source code diterjemahkan kedalam **Microsoft Intermediate Language**, kadang-kadang disingkat **MSIL**, bisa juga **IL**. IL kode ini adalah bahasa netral dan dapat disamakan dengan Java bytecode.

IL kode kemudian diterjemahkan menjadi file eksekusi. .NET Framework menyertakan **Common Language Runtime (CLR)**, dapat disamakan dengan Java Runtime Environment (JRE), yang dapat memenuhi tujuan ini. CLR adalah yang menjembatani antara pengembang source code .NET dan hardware, dan semua kode .NET berjalan dalam CLR.

CLR ini menyediakan banyak fitur-fitur yang menarik yang tidak terdapat pada versi windows DNA sebelumnya seperti automatic garbage collection, exception handling, cross-language inheritance, debugging, and "side-by-side" execution dari versi yang berbeda untuk komponen .NET yang sama.

Kesimpulan

Understanding J2EE and .NET by analogy

Untuk membantu memahami 2 model ini, tabel berikut menawarkan analogi antara teknologi J2EE dan .NET

Feature	J2EE	.NET
Type of technology	Standard	Product
Middleware Vendors	30+	Microsoft
Interpreter	JRE	CLR
Dynamic Web Pages	JSP	ASP.NET
Middle-Tier	EJB	.NET Managed

Components		Components
Database access	JDBC SQL/J	ADO.NET
SOAP, WSDL, UDDI	Yes	Yes
Implicit middleware (load-balancing, etc)	Yes	Yes

Daftar Pustaka :

Disarikan dari beberapa sumber di internet :

- The great debate: .Net vs. J2EE (<http://www.javaworld.com/javaworld/jw-03-2002/jw-0308-j2eenet.html>).
- J2EE vs. Microsoft.NET A comparison of building XML-based web services (<http://www.theserverside.com/articles/article.tss?l=J2EE-vs-DOTNET>)
- A Strategic Comparison for Technical Executives and Managers (www.kantega.no/kurs/kursliste/kursinfo.asp?thisId=1060687289)
- http://www-1.ibm.com/linux/files/middleware_final7.pdf?ca=vgr-isv24J2EEvsNet