

Purnomo, Rini D. (1992). *Pemanfaatan Energi Surya Untuk Penyulingan Air Laut Di Daerah Yogyakarta*. Seminar Analisis dan Aplikasi Perpindahan Panas dan Massa Menuju Era Industrialisasi, Yogyakarta: PAU.

Sears, Zemansky. (1983). *Fisika Untuk Universitas (Jilid III)*. Bandung: Binacipta.

Supranto. (1992). *Fundamental Transfer Panas*. Yogyakarta: Pusat Antar Universitas.

PERANCANGAN DAN IMPLEMENTASI SISTEM E-COMMERCE DENGAN TEKNOLOGI FRAMEWORK BERORIENTASI OBJEK

Oleh:
Herman Dwi Surjono
Staf Pengajar FT UNY

Abstract

The purpose of this research was to design and to implement an e-commerce (electronic commerce) system using components and object-oriented frameworks technology. The research begins with analyzing and identifying specifications, which are categorized functionally into modules and objects. The objects are grouped into three logical categories: presentation aspects, business rules and data, and objects that accept and interpret user requests and control the business objects. The result is an e-commerce system running on local server of J2EE with URL <http://localhost:8000/rbi/>. By accessing the system, it is assumed that the performance would be in line with the expected research goals. The important web pages of the system among others are main page, product category, product item list, detail item description, shopping cart, log-in and log-out page, order notes, and other supporting pages. This e-commerce system was designed to be adapted to other products easily.

Keywords: e-commerce, JavaServer Page (JSP), component, framework, object-oriented programming.

PENDAHULUAN

Pada satu dasa warsa terakhir terlihat bahwa kekuatan komputasi dan lebar jalur jaringan telah meningkat dengan pesat. Namun demikian perancangan dan implementasi perangkat lunak yang kompleks tetap dirasa mahal dan cenderung banyak terjadi

kesalahan. Sebagian besar biaya dan usaha dipakai untuk penemuan kembali konsep dan komponen yang terus-menerus. Terlebih, bila melihat kenyataan semakin bervariasinya arsitektur perangkat keras, sistem operasi dan *platform* komunikasi, dirasa semakin berat untuk membangun aplikasi yang portabel, efisien, dan murah.

Framework berorientasi objek (*object oriented frameworks*) merupakan teknologi yang menjanjikan dalam penggunaan kembali hasil rancangan dan implementasi perangkat lunak sehingga mengurangi biaya dan meningkatkan kualitas. Berbeda dengan teknik berorientasi objek sebelumnya yang berbasis pada pustaka kelas, maka *framework* ini berbasis pada unit dan domain aplikasi tertentu. *Framework* terdiri atas sekumpulan kelas yang dirancang untuk menyelesaikan masalah tertentu (Damian, 1999).

Seiring dengan semakin kompleksnya sistem perangkat lunak, teknologi *framework* berorientasi objek dirasa semakin penting peranannya, baik untuk industri maupun akademis. Kini banyak aplikasi *e-commerce* (*electronic commerce*) tersedia di pasaran, akan tetapi umumnya masih kurang terkemas dengan baik. Di samping itu, proses pengembangannya relatif masih kompleks dan cenderung sering terjadi kesalahan. Dengan teknologi *framework* berorientasi objek yang kini sedang berkembang diharapkan dapat diatasi beberapa kelemahan tersebut.

Tujuan penelitian ini adalah untuk mendesain dan mengimplementasikan suatu sistem *e-commerce* dengan teknologi komponen dan *framework* berorientasi objek. *Platform* Java 2 Edisi Enterprise (J2EE) dengan dukungannya terhadap Enterprise JavaBeans (EJB), Servlet dan JavaServer Page (JSP) memberikan lingkungan *framework* dan komponen berbasis Java yang tepat untuk pengembangan berbagai aplikasi bisnis dengan cepat dan mudah. Di samping itu, dengan sifat Java yang netral arsitektur akan menjamin aplikasi yang dikembangkan dapat berjalan pada berbagai *platform*.

E-commerce pada dasarnya merupakan bentuk transaksi bisnis secara elektronik yang menggunakan komputer dan jaringan telekomunikasi, baik melibatkan individu maupun perusahaan (Kamthan, 1999). Liew (1998) mendefinisikan *e-commerce* sebagai bentuk transaksi bisnis melalui media elektronik seperti surat elektronik, faks, dan internet.

Komponen utama yang harus ada dalam suatu sistem *e-commerce* adalah katalog produk dan troli belanjaan yang bersifat *on-line*. Katalog produk berisi daftar barang-barang yang disusun secara sistematis sehingga mudah diakses oleh pengguna. Troli belanjaan adalah suatu tempat untuk menaruh barang-barang yang telah dipilih pelanggan. Situs *e-commerce* yang baik menurut Browning (1999) adalah situs yang memberikan lingkungan yang

memudahkan bagi pengunjung sejak melihat-lihat katalog produk tersebut, memilih barang dan memasukkannya dalam troli belanjaan, hingga melakukan transaksi.

Metode pembayaran dalam sistem *e-commerce* umumnya menggunakan kartu kredit. Pemrosesan pembayaran melalui kartu kredit dimulai dengan authorisasi. Authorisasi secara sederhana berarti mengecek apakah kartu tidak dilaporkan sedang dicuri dan masih ada sejumlah kredit yang cukup setelah dikurangi dengan nilai transaksi. Authorisasi bisa diperoleh secara manual atau otomatis. Secara manual berarti dilakukan sebagaimana pada toko nyata yakni menggunakan metode *point of sale* (POS), sedangkan secara otomatis berarti server sistem *e-commerce* berkomunikasi secara langsung dengan perusahaan penjamin kartu kredit. Setelah proses authorisasi selesai, kartu kredit akan didebit dengan jaminan bahwa barang akan segera dikirim dalam rentang waktu yang disepakati.

Aplikasi bisnis perlu didesain, dibangun, dan diproduksi dengan waktu lebih cepat, dengan biaya yang lebih murah, dan dengan sumberdaya yang lebih sedikit dari sebelumnya. Guna mengatasi hal itu platform J2EE memberikan pendekatan berbasis komponen untuk mengembangkan dan mendistribusikan aplikasi bisnis. Platform J2EE memberikan model aplikasi terdistribusi *multi-tier*, kemampuan untuk menggunakan kembali komponen,

model keamanan yang menyatu, dan kontrol transaksi yang fleksibel (Pawlan, 2000).

Komponen adalah elemen perangkat lunak yang bersifat dapat digunakan kembali guna membentuk aplikasi lain yang lebih besar. Bentuk dan ukuran komponen sangat bervariasi. Suatu komponen dapat berupa sebuah antarmuka GUI yang kecil dan sederhana seperti sebuah tombol, atau dapat pula diimplementasikan sebagai aplikasi yang kompleks seperti sebuah fungsi untuk pengelolaan gaji. Arsitektur yang memberi landasan pengembangan dan interaksi antara komponen-komponen disebut model komponen (Thomas, 1998).

Framework berorientasi objek membawa paradigma berorientasi objek lebih jauh dengan memberikan infrastruktur dan fleksibilitas untuk mengirim teknologi berorientasi objek serta dengan memungkinkan penggunaan kembali komponen pada skala yang lebih besar (Bosch, et.al., 1997; Taligent, 1996). *Framework* merupakan sekumpulan kelas yang membentuk suatu perancangan guna menyelesaikan berbagai persoalan yang berkaitan serta mendukung penggunaan kembali objek yang lebih besar (Johnson, 1988). Dengan konsep *framework* dimungkinkan penggunaan kembali tidak hanya pada level kode tetapi juga pada level analisis dan desain. Menurut Fayad dan Schmidt (1997) keuntungan utama

dari *framework* berorientasi objek adalah: modularitas, pemanfaatan kembali, ekstensibilitas, dan pembalikan kontrol.

Sistem *e-commerce* yang akan dikembangkan dalam penelitian ini mengikuti model aplikasi *multi-tier*, karena pada tier menengah terdiri atas tier web yang berisi halaman-halaman JSP, Servlet, XML, JavaMail dan tier EJB yang berisi Bean enterprise dan JDBC. Teknologi JPS memberikan cara yang mudah dan efisien dalam mengembangkan presentasi dinamis berbasis servlet. Di samping itu diperoleh keuntungan lain yakni pemisahan antara isi dengan logika presentasi, sehingga masing-masing dapat dikembangkan tanpa saling terikat.

Teknologi servlet dikembangkan sebagai suatu mekanisme untuk menerima permintaan dari *browser*, mengakses data dari tier aplikasi atau basis data, melakukan logika aplikasi pada data, dan memformat data tersebut untuk ditampilkan dalam *browser* (biasanya dalam HTML). Servlet dan JSP kedua-duanya menjelaskan bagaimana memroses permintaan (dari klien HTTP) untuk kemudian membangkitkan respons. Servlet diekspresikan dengan bahasa pemrograman (Java), halaman JSP merupakan dokumen berbasis teks dengan memasukkan kombinasi elemen-elemen HTML, tag-tag JSP, kode Java, dan informasi lain (Sun, 2000).

Menurut Marchal (1997) servlet mempunyai empat buah keuntungan nyata atas CGI atau API yang lainnya, yaitu: familiaritas, portabilitas, keamanan, dan kinerja. Familiaritas diperoleh karena servlet ditulis dalam Java dan servlet juga dapat mengakses semua paket Java dan JavaBean yang ada di pasaran. Portabilitas diperoleh karena servlet dapat beroperasi lintas server. Keamanan servlet diperoleh dengan mewarisi sistem keamanan Java. Kinerja servlet lebih baik karena lebih efisien dibanding dengan CGI.

Cara Penelitian

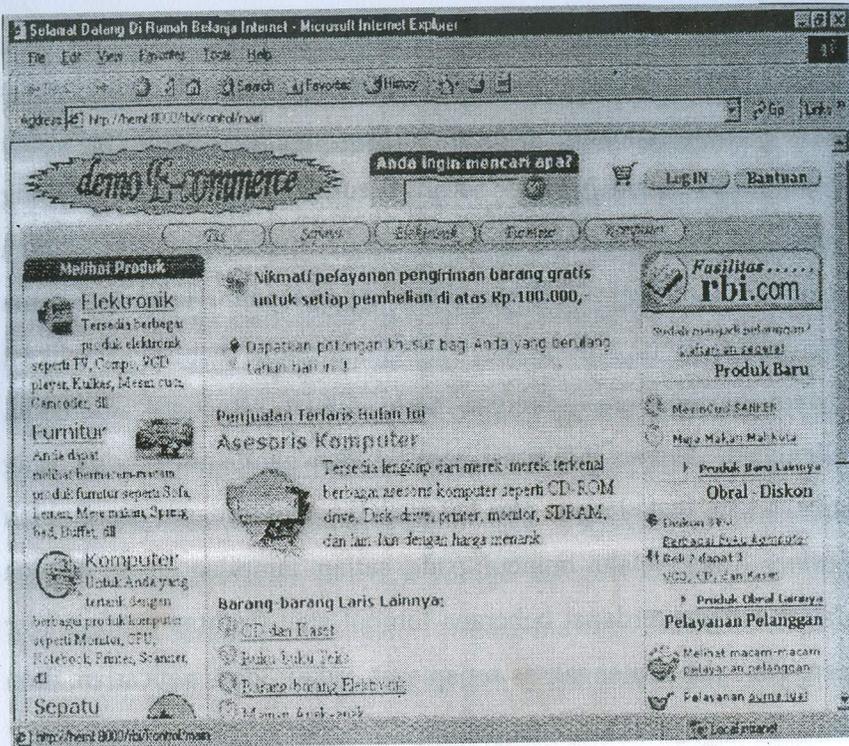
Perancangan sistem *e-commerce* dalam penelitian ini diawali dengan identifikasi persyaratan atau spesifikasi fungsi yang diharapkan dari sistem tersebut. Dari hasil identifikasi spesifikasi fungsi sistem secara keseluruhan tersebut dapat ditentukan spesifikasi untuk antarmuka pengguna. Fungsi-fungsi sistem secara keseluruhan dikategorisasi menjadi modul-modul berdasarkan fungsi yang berhubungan dan mempunyai kemiripan. Selanjutnya, menjabarkan keseluruhan sistem yang sudah dikategorisasi dalam modul-modul menjadi satuan-satuan logika bisnis, data, dan logika presentasi, serta memodelkan masing-masing satuan sebagai objek perangkat lunak.

Objek-objek dapat dikelompokkan menjadi tiga kategori logika, yakni objek yang berhubungan dengan aspek presentasi, objek yang menangani aturan bisnis dan data, serta objek yang menerima dan menginterpretasi permintaan pengguna dan mengontrol objek bisnis untuk memenuhi permintaan tersebut. Objek presentasi akan menempati dan ditangani oleh tier web atau klien. Objek yang berkaitan dengan kontrol berada pada setiap tier agar dapat melakukan koordinasi aksi di antara tier. Objek yang memodelkan data dan aturan bisnis berada di tier EJB.

Implementasi sistem *e-commerce* ini mengikuti arsitektur *Model/View/Controller* (MVC). Ketiga kategori objek yang telah dijelaskan berkaitan dengan masing-masing MVC tersebut. Tampilan (atau *view*) menentukan penyajian antarmuka pengguna aplikasi. Tampilan aplikasi ini diimplementasikan dengan halaman *JavaServer Pages* (JSP) dan EJB. Halaman JSP digunakan untuk membangkitkan respons HTML secara dinamis. Halaman-halaman layar ditampilkan melalui mekanisme *template*. Mekanisme *template* memberikan cara untuk memisahkan elemen-elemen yang merupakan bagian tiap halaman dengan elemen-elemen yang berubah setiap halaman. Dengan meletakkan semua yang umum dalam satu file, maka akan diperoleh konsistensi tampilan.

HASIL DAN PEMBAHASAN

Hasil penelitian ini berupa sistem *e-commerce* yang berada pada server lokal dengan <http://localhost:8000/rbi/> dan ditampilkan dalam *browser* secara berurutan sebagaimana seorang pengguna mengakses sistem tersebut mulai halaman utama (gambar 1) hingga halaman bukti pemesanan. Halaman utama merupakan pintu gerbang bagi pengguna untuk memasuki aplikasi sistem *e-commerce*, sehingga perlu dibuat semenarik mungkin, dilengkapi dengan informasi penting, dan gambaran produk yang ditawarkan. Pada bagian atas halaman tersebut terdapat *header* atau *banner* yang selalu muncul pada setiap tampilan layar, karena dalam *banner* terdapat beberapa tombol atau fasilitas penting agar pengguna dapat mengakses setiap saat, yakni: logo, pencarian, ikon troli, ikon Log-in/Out, ikon bantuan, dan tombol-tombol navigasi.



Gambar 1. Halaman Utama Sistem E-Commerce

Dengan mekanisme *template* yang diimplementasikan dalam halaman JSP, fasilitas *banner* tersebut selalu muncul pada setiap halaman yang ditampilkan. Mekanisme *template* memberikan cara untuk memisahkan elemen-elemen yang umum yang merupakan bagian pada setiap layar dengan elemen yang berubah setiap pergantian layar. Implementasi dengan halaman JSP yang diberi nama *template.jsp* adalah sebagai berikut.

```
<%@ taglib uri="WEB-INF/tlds/taglib.tld" prefix="j2ee" %>
<%@ include file="DefinisiLayar.jsp" %>
<html>
<head>
<title>
<j2ee:insert template="template" parameter="JudulHtml" />
</title>
</head>
<body bgcolor="white">
<j2ee:insert template="template" parameter="BannerHtml" />
<j2ee:insert template="template" parameter="IndeksHtml" />
<j2ee:insert template="template" parameter="BadanHtml" />
</body>
</html>
```

Dengan file *template.jsp* tersebut dapat dihasilkan bermacam-macam tampilan dengan cara melewatkan parameter yang berbeda. Parameter-parameter yang mungkin dalam seluruh tampilan sistem perlu didefinisikan terlebih dahulu dalam file *DefinisiLayar.jsp*. Selanjutnya, untuk menentukan layar apa yang akan ditampilkan diperlukan adanya pengontrol. Pengontrol menentukan pemilihan layar dengan menggunakan ID layar yang juga telah didefinisikan dalam file *DefinisiLayar.jsp* tersebut.

Komposisi halaman utama dengan nama file *index.jsp* terdiri atas tiga bagian yang diwujudkan dengan tiga file terpisah, yakni: *kiri.jsp*; *tengah.jsp*; dan *kanan.jsp* yang masing-masing menduduki tempat sesuai nama filenya. Hal ini bisa diwujudkan dengan fasilitas yang diberikan teknologi JSP. File: *index.jsp* tersebut adalah sebagai berikut:

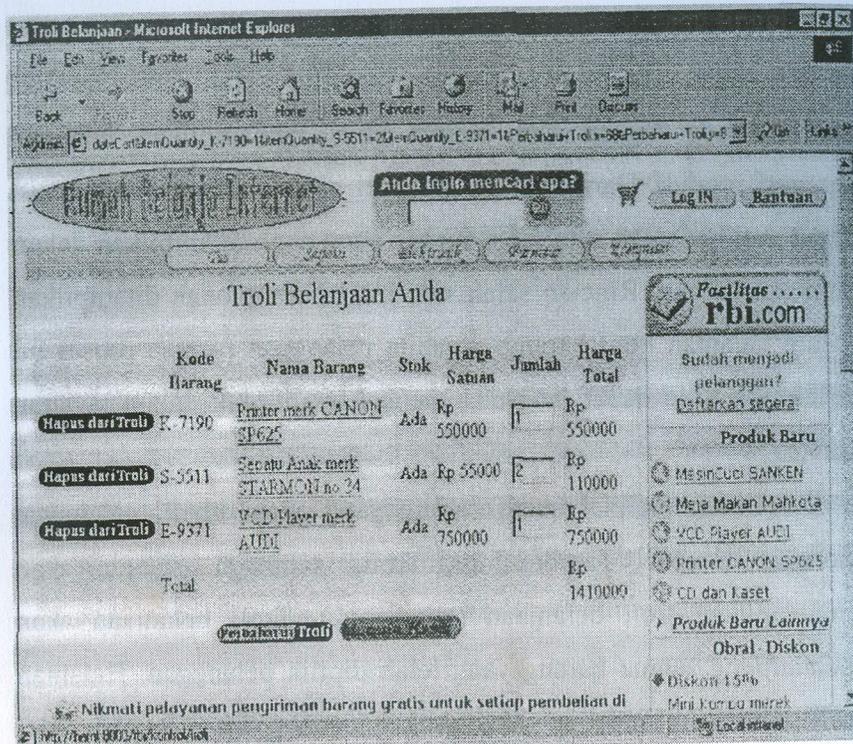
```
<table width="100%">
  <tr>
    <td vAlign=top>
      <%@ include file="kiri.jsp"%>
    </td>
    <td vAlign=top>
      <%@ include file="kanan.jsp"%>
    </td>
  </tr>
</table>
```

File `kiri.jsp` yang akan menduduki posisi kiri dari layar berisi menu untuk melihat berbagai kategori produk yang ditawarkan. Pada bagian tengah ditampilkan berbagai kelengkapan dan promosi yang berkaitan dengan fasilitas sistem. Pada sistem *e-commerce* ini bagian tengah tersebut diisi informasi mengenai fasilitas pelayanan pengiriman barang secara gratis, potongan khusus bagi pelanggan yang pada hari ini berulang tahun, serta produk-produk yang menjadi unggulan karena paling banyak dibeli pelanggan. Pemisahan bagian ini dalam suatu file tersendiri yakni file: `tengah.jsp` akan memudahkan pengembang untuk mengedit atau memperbaharui tampilan atau fasilitas lainnya. Demikian juga file: `kanan.jsp` yang menduduki posisi bagian kanan dengan mudah dapat diedit untuk memperbaharui isi.

Pada halaman utama ini pelanggan dapat memulai mencari barang-barang dengan memilih kategori tertentu, atau dapat langsung memilih produk yang termasuk dalam tawaran khusus.

Misalnya, produk yang terlaris, produk yang diobral, atau produk-produk baru. Apabila pelanggan memilih salah satu kategori produk pada bagian kiri, sistem akan menampilkan halaman kategori produk. Dari halaman kategori produk tersebut pelanggan dapat melihat salah satu jenis barang yang akan ditampilkan dalam halaman barang. Rincian salah satu jenis barang akan ditampilkan dalam halaman detail barang. Apabila pelanggan merasa proses ini terlalu lama, ia dapat langsung mengakses produk dalam tawaran khusus.

Untuk melihat suatu barang yang akan dibeli, pelanggan memijit tombol Tambah ke Troli, sehingga sistem akan menampilkan troli belanjaan (gambar 2). Troli belanjaan akan menampung semua barang yang telah dipilih pelanggan. Halaman troli belanjaan tersebut dibangkitkan secara dinamis oleh file: `troli.jsp` dengan menggunakan komponen EJB. Pada halaman ini selain terdapat tabel troli yang menjadi fokus utama, terdapat pula file `tengah.jsp` yang menempati posisi kiri bawah, dan file `kanan.jsp` yang menempati posisi kanan. Halaman-halaman lain yang selanjutnya dilalui pelanggan antara lain: pengecekan, log-IN, pemesanan, bukti pemesanan, log-OUT, hasil pencarian, rekening baru, rekening berhasil, kesalahan duplikasi, dan bantuan.



Gambar 2. Halaman Trolis Belanjaan

Sistem *e-commerce* yang dirancang dalam platform J2EE ini bersifat mudah dikembangkan untuk produk-produk lain tanpa harus mengubah kode pemrogramannya. Hal ini akan menjawab tuntutan bisnis terutama yang berbasis internet, yakni pengembangan aplikasi yang cepat dan murah sehingga bisa mengikuti perkembangan dunia bisnis. Beberapa file yang perlu

dimodifikasi agar sistem *e-commerce* ini dapat digunakan untuk memasarkan produk lain dapat disebutkan berikut ini.

1. Lima buah file jsp, yang meliputi:
 - banner.jsp, untuk menampilkan tombol navigasi produk pada bagian atas setiap layar.
 - topindex.jsp, untuk menghubungkan tombol navigasi produk dengan data.
 - kiri.jsp, tengah.jsp, dan kanan.jsp, untuk menampilkan isi.
2. Sebuah file data yang berkaitan dengan produk
3. Semua file gambar yang berkaitan dengan produk.

Pengembangan beberapa fasilitas pendukung pada sistem *e-commerce* meliputi: Monitoring Aktivitas Transaksi, Pengelolaan Aktivitas Pembayaran, Pelayanan Purna Transaksi, Langkah Pengamanan Data Pelanggan, dan Internasionalisasi Sistem *E-commerce*.

1. Monitoring aktivitas transaksi

Umumnya pengelola atau administrator sistem tidak terlibat dalam proses transaksi. Oleh karena itu, perlu ada mekanisme agar pengelola dalam hal ini pemilik bisnis mengetahui siapa saja yang telah melakukan transaksi, informasi kartu kredit, perincian barang-

barang, dan informasi alamat pengiriman, sehingga transaksi tersebut dapat ditindak lanjuti oleh pengelola.

Untuk mengatasi hal tersebut telah dikembangkan pula aplikasi klien dengan Microsoft Excel yang berkomunikasi dengan Server J2EE melalui protokol HTTP. Data dikirim oleh server dalam format *Extended Markup Language (XML)*

2. Pengelolaan aktivitas pembayaran

Aktivitas pembayaran dalam sistem *e-commerce* perlu dikelola dengan baik. Mekanisme pengelolaan tersebut dapat dirumuskan sesuai dengan aturan yang baku dalam sistem *e-commerce*. Mekanisme pembayaran dari pihak pelanggan kepada perusahaan *e-commerce* dan sebaliknya, dalam kasus bila ada ketidakpuasan pelanggan terutama untuk model pembayaran secara *on-line*, selalu melibatkan pihak ketiga seperti lembaga perbankan atau perusahaan lain.

Beberapa cara pembayaran yang dikenal dalam sistem *e-commerce* adalah dengan kartu kredit, cek, tunai, sedangkan cara pengiriman pembayarannya dapat dilakukan secara *on-line*, melalui pos, telepon atau faks. Pemrosesan pembayaran menggunakan kartu kredit atau sering disebut dengan authorisasi kartu kredit dapat dilakukan secara otomatis (*real time*) atau secara manual (*batching*). Agar sistem *e-commerce* dapat menerima pembayaran

secara *on-line* melalui kartu kredit diperlukan pengelolaan atas tiga hal, yakni: blanko pemesanan atau troli belanjaan, perangkat lunak pemroses pembayaran, dan rekening pengusaha.

3. Pelayanan purna transaksi

Setelah pelanggan melakukan transaksi dan sistem menampilkan halaman bukti pemesanan, tentunya pelanggan ingin agar barang-barang yang dipesan segera dikirim, kemudian sampai di alamat tujuan dengan cepat serta kondisi barang yang diterima sesuai dengan harapannya. Oleh karena itu, perusahaan *e-commerce* perlu mengantisipasi hal tersebut dengan pelayanan purna transaksi sebaik-baiknya. Kepuasan pelanggan terhadap pelayanan purna transaksi ini menjadi tolok ukur keberhasilan perusahaan dalam menjalankan bisnis *e-commerce*.

Adapun macam-macam kegiatan yang termasuk dalam pelayanan purna transaksi adalah: pembungkusan barang, pengiriman barang, pemantauan kemajuan pengiriman, jaminan uang kembali, pelayanan pelanggan *on-line*, dan pembuatan situs tentang FAQ.

4. Langkah pengamanan data pelanggan

Terdapat beberapa sistem yang dapat digunakan untuk mengamankan data informasi kartu kredit pelanggan. Sistem yang paling banyak digunakan adalah *Secure Sockets Layer (SSL)*.

Protokol SSL yang didukung oleh hampir semua *browser* dan *Internet Service Provider (ISP)* ini bertujuan untuk (a) melakukan enkripsi data kartu kredit yang sedang dikirim lewat jaringan, dan (b) melakukan sertifikasi bahwa pesan adalah datang dari sumber yang seharusnya.

Adapun langkah-langkah yang dilakukan untuk mendapatkan dan menjalankan SSL adalah (a) melakukan kontak dengan ISP untuk meletakkan halaman web pada server yang aman, (b) melakukan kontak dengan salah satu lembaga sertifikasi untuk mendapatkan sertifikat digital, (c) meminta ISP mengeluarkan *Certificate Signing Request* dan mengirimkannya kepada lembaga sertifikasi yang telah dipilih, dan (d) ISP selanjutnya dapat menginstall sertifikat tersebut untuk sistem *e-commerce*.

5. Internasionalisasi sistem *e-commerce*

Internasionalisasi adalah proses penyiapan aplikasi untuk mendukung berbagai bahasa di dunia, sedangkan lokalisasi adalah proses penyesuaian aplikasi internasional untuk mendukung bahasa tertentu. Proses internasionalisasi suatu aplikasi meliputi identifikasi dan isolasi bagian-bagian yang berisi data untuk pengguna, sehingga data tersebut dapat dikumpulkan dan kemudian diakses dari sebuah file. Selanjutnya, proses lokalisasi adalah

menerjemahkan data tersebut ke dalam suatu bahasa tertentu dan menyimpannya dalam sebuah file yang dapat diakses oleh aplikasi.

Sistem *e-commerce* yang sudah diinternasionalisasi akan dengan mudah diadaptasikan ke dalam bahasa lain, yakni hanya dengan menerjemahkan data yang tersimpan dalam satu file. Penanganan data merupakan satu bagian dari aplikasi yang paling dipengaruhi oleh internasionalisasi, dengan pengaruh pada tiga bidang, yaitu: input data, penyimpanan data, dan presentasi data yang independen terhadap lokal. Beberapa data yang peka terhadap lokal seperti format tanggal, waktu, mata uang, dan nomor perlu dibuat dalam format yang independen terhadap lokal.

SIMPULAN

Dari hasil penelitian dan pembahasan di depan dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem *e-commerce* dalam penelitian ini dikembangkan dengan teknologi komponen dan *framework* berorientasi objek dalam *platform Java 2 Edisi Enterprise (J2EE)*.
2. Dari seluruh rangkaian pengaksesan sistem *e-commerce* mulai halaman utama hingga selesainya proses transaksi dapat disimpulkan bahwa unjuk kerja sistem sudah sesuai dengan tujuan yang diharapkan.

3. Halaman-halaman penting dari sistem adalah halaman utama, kategori produk, daftar barang, detil barang, troli belanjaan, log-IN/log-OUT, bukti pesanan, fasilitas pencarian, dan bantuan serta halaman pendukung lainnya.
4. Sebagai kelengkapan dikembangkan pula aplikasi monitoring bagi pemilik bisnis, sehingga dapat diketahui aktivitas transaksi yang terjadi dan dapat dilakukan tindak lanjut pengiriman barang kepada pelanggan.
5. Sistem *e-commerce* ini dirancang agar mudah digunakan untuk memasarkan produk-produk lain. Untuk itu, perlu dimodifikasi lima buah file jsp, sebuah file data, dan file-file gambar yang berkaitan dengan produk.
6. Pengembangan beberapa fasilitas pendukung pada sistem *e-commerce* meliputi: Monitoring Aktivitas Transaksi, Pengelolaan Aktivitas Pembayaran, Pelayanan Purna Transaksi, Pengamanan Data, dan Internasionalisasi.

DAFTAR PUSTAKA

- Bosch, J., Molin, P., Mattson, M. and Bengtson, P. (1997). *Objects-Oriented Frameworks – Problems & Experiences*. Proceedings of the 23rd International Conference in Technology of Object-Oriented Languages and Systems, TOOLS '97 USA, Santa Barbara, California, July 28-Agustus 1, 203-214.
- Browning, B. (1999). *Electronic Commerce Tutorial*. London: Textor Webmasters Ltd.
- Damian, A. (1999). *An Object Oriented Framework for the Simulation of Network Models*. Master's thesis, University of Calgary.
- Fayad, E.M. and Schmidt, D.C. (1997). "Object-Oriented Application Frameworks". *Communications of the ACM*, 40(10), October, 32-38.
- Johnson, R.E. and Foote, B. (1988). "Designing Reusable Classes". *The Journal of Object-Oriented Programming*, 1(2), 22-35.
- Kamthan, P. (1999). *E-Commerce on the WWW: A Matter of Trust*. [Online]. Tersedia pada <http://www.ibm.com/software/developer/>.
- Liew, W. (1998). *Object-Oriented Application Frameworks for Internet Commerce*. MSc. Thesis, University of Alberta.
- Marchal, B. (1997). *Dynamic WebPage in Java Servlet*. CA: Digital Cat, LLC.
- Pawlan, M. (2000). *J2EE™ Blueprints Digest*. Sun Microsystems, Inc.
- Sun Microsystems. (2000). *Designing Enterprise Applications for the Java™ 2Platform, Enterprise Edition*. Sun Microsystems, Inc.

- Taligent. (1996). *Leveraging Object-Oriented Frameworks – A Technology Primer from Taligent*. Taligent White Papers, [Online] <http://www.ibm.com/java/ooleveraging/index.html>.
- Thomas, A. (1998). *Enterprise JavaBeans Technology-Server Component Model for the Java Platform*. Patricia Seybold Group, 1998.

EVALUASI PASCA HUNI GEDUNG PERKULIAHAN DAN ADMINISTRASI JURUSAN PENDIDIKAN TEKNIK BANGUNAN FAKULTAS TEKNIK UNY

Oleh:
Bambang Sugestiyadi
Staf Pengajar FT UNY

Abstract

The study of the post occupancy evaluation of the lecturing and administration buildings of the Department of the Construction Engineering Teacher Training of the Faculty of Technique of Yogyakarta State University is limited at the indicative level of the lecturing and administration buildings. It aims at investigating the technical and functional aspects of the lecturing and administration rooms of the Department of the Construction Engineering Teacher Training of the Faculty of Technique of Yogyakarta State University. It is conducted in 5 months of the lecturing period 1999/2000, with the analysis unit of the Department of the Construction Engineering with all of its activity aspects. The observing focuses of the study are a) the technical aspect, including: 1) lighting, 2) ventilation, 3) security, 4) fire fighting, 5) maintenance; b) the functional aspect, including: 1) room organization, 2) room capacity, 3) circulation, 4) communication, 5) the possibility to change. The data is collected using interview, field observation and documentation. The content validity is tested on relational judgmental basis. The results of the study show that in almost 50% of the lecturing period of the odd semester of 1999/2000 the lecturing room capacity is "less". It is also found that the ideal and comfortable lecturing rooms are those with the capacity of 30-35 individuals per classroom. Based on the results of the study, it is necessary to consider the addition of the lecturing rooms with the "big" capacity of 100 individuals. And, based on the results of the evaluation of the capacity of the administration and lecturer rooms show that 90% of the capacity of the rooms do not meet the working room standard. The condition of the comfort and the security of the rooms do not meet the requirements. Based on the results of the study it is necessary to develop the administration and the lecturer rooms into a "vertical" direction that they meet the comfortable working room standard.

Keywords: Building, evaluation, comfort