

**MODEL DAN IMPLEMENTASI E-FORUM BERBASIS J2EE MENGGUNAKAN EJB
DESIGN PATTERN DAN STRUTS APPLICATION FRAMEWORK**

Joko Lianto Buliali¹
Suhadi Lili²
Muhammad Zuber³

Email : joko@its-sby.edu, suhadi@its-sby.edu, zubby@telkom.net

ABSTRAKSI

Pengelolaan komunitas pada situs web organisasi memegang peranan penting untuk mencapai keberhasilan pengelolaan situs web tersebut. E-Forum merupakan salah satu tool yang memungkinkan untuk meningkatkan interaksi pengguna. Tujuan utama dari E-Forum adalah sebagai wadah untuk diskusi secara online melalui media Internet. Dalam penelitian ini sistem E-Forum dimodelkan dengan menggunakan notasi UML sedangkan data dimodelkan dengan menggunakan Conceptual Data Model (CDM) dan Physical Data Model (PDM). E-Forum dibangun berdasarkan standar J2EE agar dapat dihasilkan aplikasi yang dapat bekerja secara cross platform dan scalable. Dalam pengembangan aplikasi berbasis J2EE, terdapat beberapa pattern dan framework yang dapat mempercepat dan mempermudah proses pengembangan. Dalam pembuatan komponen EJB, pattern yang sering digunakan adalah Session Façade Pattern, Value Object / Data Transfer Object Pattern, Data Access Object Pattern, JDBC for Reading Pattern, EJHome Factory Pattern, dan Business Delegate Pattern. Struts framework, yang berbasis paradigma MVC, digunakan dalam pembuatan komponen web. Uji coba fungsional dilakukan untuk mengevaluasi fitur-fitur E-Forum. Uji coba non fungsional seperti internationalization support, clustering support dan multi database support juga dilakukan untuk mengevaluasi E-Forum yang dibuat. Hasil uji coba menunjukkan bahwa aplikasi E Forum berfungsi sesuai dengan tujuan yang ditetapkan.

Kata Kunci: Pengelolaan komunitas, E-Forum, J2EE, EJB, Struts Application Framework.

-
- 1. Dosen Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember**
Kampus ITS, Jl. Raya ITS, Sukolilo, Surabaya 60111
Telp. +(62) (31) 593 9214, Fax. +(62) (31) 599 6813
 - 2. Dosen Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember**
Kampus ITS, Jl. Raya ITS, Sukolilo, Surabaya 60111
Telp. +(62) (31) 593 9214, Fax. +(62) (31) 599 6813
 - 3. Dosen Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember**
Kampus ITS, Jl. Raya ITS, Sukolilo, Surabaya 60111
Telp. +(62) (31) 593 9214, Fax. +(62) (31) 599 6813

ABSTRACT

Community management in an organization's website plays an important role in achieving the success of the management of that website. E-Forum is one of the tools that allows the enhancement of the users' interaction. The main purpose of E-Forum is as a forum for online discussion through the use of the Internet. In this research E-Forum system is modelled by using UML notation while data is modelled by using Conceptual Data Model (CDM) and Physical Data Model (PDM). The E-Forum is built using J2EE standard so that cross-platform and scalable application can be obtained. In the development of applications based on J2EE, several patterns and frameworks can be used to make the development quicker and easier. In the development of EJB components, patterns that are often used are Session Façade Pattern, Value Object / Data Transfer Object Pattern, Data Access Object Pattern, JDBC for Reading Pattern, EJBHome Factory Pattern, and Business Delegate Pattern. Struts framework, which is based on MVC paradigm, is used in the development of web component. Functional testing is carried out to evaluate the features of E-Forum. Non-functional testing, such as on internationalization support, clustering support and multi database support, is also carried out to evaluate the E-Forum. The result of the testing showed that the E-Forum application works as the stated purpose.

Key words: Community management, E-Forum, J2EE, EJB, Struts Application Framework.

1. PENDAHULUAN

Perkembangan teknologi Internet dalam kurun waktu beberapa tahun terakhir ini sangatlah pesat. Banyak kegiatan bisnis kini dapat dilakukan melalui media Internet, seperti *E-Commerce*, *E-Banking*, *E-Bidding*, *E-Store*. *Electronic forum (E-Forum)* atau *message boards* merupakan turunan dari *bulletin boards system* dan *usenet news system* yang populer pada tahun 1990-an. *E-Forum* biasanya merupakan bagian dari suatu situs *web* yang memungkinkan pengguna untuk memulai suatu topik dan mendiskusikannya dengan yang lain. Secara umum, aplikasi *E-Forum* memungkinkan *webmaster* untuk mendefinisikan forum-forum yang merupakan tempat bagi pengguna untuk posting topik (*thread*). Pengguna yang lain dapat memberikan tanggapan terhadap topik tersebut atau membuat topik baru sesuai keinginannya.

Pengelolaan komunitas merupakan salah satu elemen pokok yang tidak boleh dikesampingkan pada pengelolaan situs web. Keberhasilan situs web, baik situs web organisasi maupun situs web perusahaan, tidak lepas dari keberhasilan pengelolaan komunitas dengan baik. *E-Forum* merupakan salah satu tool yang memungkinkan untuk meningkatkan interaksi pengguna. Disamping itu, *E-Forum* juga dapat meningkatkan komunikasi, pertukaran informasi dan kolaborasi baik bagi pelanggan, mitra kerja maupun karyawan. Fungsi utama *E-Forum* adalah sebagai wadah untuk tanya jawab dan berdiskusi secara online melalui media Internet. Disamping itu *E-Forum* juga seringkali dilengkapi dengan beberapa fitur lainnya seperti *attachment support*, *multi language*, *themes*, dan sebagainya.

Pada penelitian ini, dibangun *E-Forum* dengan menggunakan *Java* dan dengan standar J2EE untuk menghasilkan aplikasi yang *reliable*, *robust* dan *multi platform*. J2EE menawarkan model aplikasi yang terdistribusi, kemudahan untuk memakai ulang komponen, komponen keamanan, dan komponen *transaction control* (Sun Microsystems, 2002). Dengan menggunakan standar J2EE, maka *E-Forum* yang dibuat dapat berjalan di berbagai platform sistem operasi (Windows, Linux, Unix, Solaris) dan dapat di-*deploy* ke berbagai *application server* (WebSphere, WebLogic, JRun, JBoss dan lain-lain) tanpa mengubah *source code*.

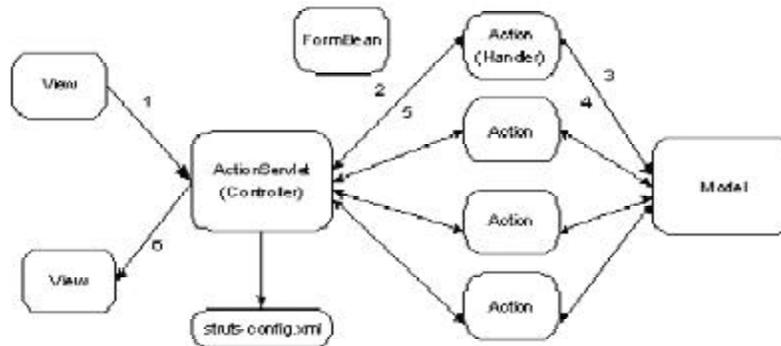
2. J2EE

Dalam pengembangan aplikasi berbasis J2EE, terdapat beberapa *pattern* dan *framework* yang dapat mempercepat dan mempermudah proses pembuatan aplikasi. Dalam hal pembuatan komponen EJB, beberapa *pattern* yang sering digunakan adalah *Session Façade Pattern*, *Value Object / Data Transfer Object Pattern*, *Data Access Object Pattern*, *JDBC for Reading Pattern*, *EJBHome Factory Pattern*, dan *Business Delegate Pattern* (Sun Microsystems, 2002) (Marinescu, 2002). Sedang dalam hal pembuatan aplikasi berbasis *web* dapat digunakan *Struts Application Framework*.

3. STRUTS APPLICATION FRAMEWORK

Struts Application Framework adalah suatu *framework* yang bersifat *open source* yang bermanfaat dalam pembuatan aplikasi berbasis *web* dengan menggunakan *servlet* dan JSP. Arsitektur Struts didasarkan pada paradigma perancangan Model-View-Controller (MVC) (Apache Software Foundation, 2007). Struts mengintegrasikan *servlet*, *JSP*, *custom tags*, *message resource*, dan *Java Bean* ke dalam satu *framework* sehingga dapat diperoleh hasil yang optimal baik dalam proses pembuatan maupun produk akhir. Struts dilengkapi pula dengan *flow control*, *error handling*, *resource-management* serta *internationalization support*.

Sesuai dengan paradigma MVC, Struts memiliki tiga komponen pokok, yaitu komponen *Controller*, komponen *Model*, dan komponen *View*. Dalam *Struts Application Framework*, *process flow* dari aplikasi diatur oleh *controller* yang terpusat (*ActionServlet*). Komponen *Controller* akan mendelegasikan *request*, dalam hal ini *HTTP request*, ke *handler (Action)* yang sesuai. Setiap *handler* berperan sebagai *adapter* antara *request* dan *model*. Komponen *Model* sendiri merepresentasikan atau mengenkapsulasi *business logic* atau *state*. Setelah proses pada *business logic* selesai, *flow* dilanjutkan kembali ke *handler* dan *controller* untuk selanjutnya ditampilkan ke komponen *View* yang sesuai. Alur pendelegasian ini ditentukan melalui suatu peta yang diletakkan pada file konfigurasi dari *Struts Application Framework*. Cara kerja *Struts Application Framework* dapat dilihat pada Gambar 1.

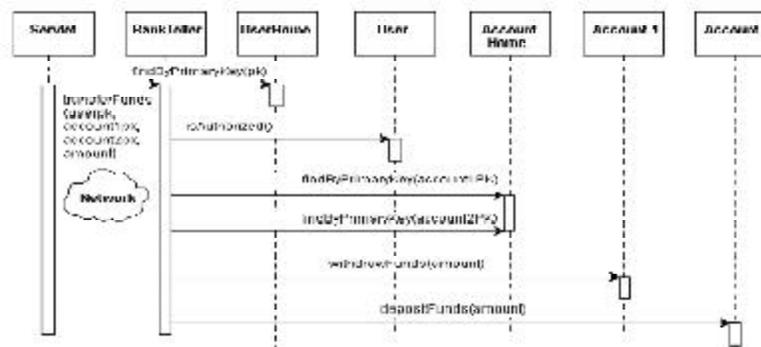
Gambar 1. *Process Flow* pada Struts Application Framework

4. EJB DESIGN PATTERN

Dalam hal pembuatan komponen EJB, beberapa *pattern* yang sering digunakan adalah *Session Façade Pattern*, *Value Object / Data Transfer Object Pattern*, *Data Access Object Pattern*, *JDBC for Reading Pattern*, *EJBHome Factory Pattern*, dan *Business Delegate Pattern*.

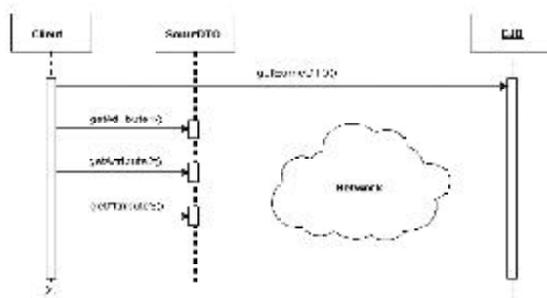
1. Session Façade Pattern

Session Façade Pattern merupakan teknik dengan menggunakan *Session Bean* yang bertindak selaku *intermediary* untuk akses ke *Entity Bean*. EJB *client* (baik yang merupakan aplikasi biasa ataupun aplikasi *web*) tidak diijinkan mengakses *Entity Bean* secara langsung karena mengakibatkan *network cost* menjadi tinggi. Gambar 2 menunjukkan skenario *Session Façade Pattern*.

Gambar 2. Skenario *Session Façade Pattern*

2. Data Transfer Object / Value Object Pattern

Data Transfer Object (DTO) Pattern merupakan teknik dengan menggunakan suatu *helper class* untuk mengenkapsulasi data baik yang dikirim dari suatu *EJB client* ke *server* ataupun sebaliknya. Dalam implementasinya DTO merupakan *serialized Java class* biasa yang mengenkapsulasi data untuk dapat ditransfer melalui *network*. Gambar 3 menunjukkan skenario *Data Transfer Object Pattern*.



Gambar 3. Skenario *Data Transfer Object Pattern*

3. Data Access Object Pattern

Data Access Object Pattern merupakan digunakan untuk memisahkan *business logic* dan *data access logic* sehingga memudahkan adaptasi akan terjadinya perubahan *resource database* yang digunakan. Dalam implementasinya ada beberapa tambahan *helper class* dan *interface* antara lain *DAOFactory* sebagai *generator* dari *DAO object*, *DAOInterface* sebagai sebagai kontrak untuk metode-metode yang akan digunakan, dan *DAOImplementation* sebagai implementasi dari *DAOInterface* untuk tiap *database* yang menjadi target dari aplikasi EJB yang dibuat.

4. JDBC For Reading Pattern

JDBC For Reading Pattern merupakan teknik untuk melakukan akses langsung ke *database* tanpa melalui EJB, dalam hal ini *Entity Bean. Pattern* ini khusus digunakan dalam operasi baca saja dimana transaction tidak diperlukan. *Pattern* ini sangat berguna menangani permintaan *listing data* dalam jumlah besar yang biasanya datang dari suatu *web application* yang merupakan *client* dari EJB.

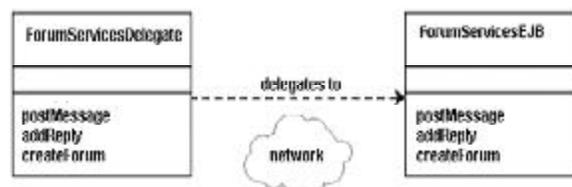
5. EJBHomeFactory Pattern

EJBHomeFactory Pattern merupakan teknik yang digunakan pada *client* dari EJB untuk melakukan koneksi maupun melakukan *caching* pada *EJBHome object*. *EJBHomeFactory* dapat diimplementasikan sebagai *singleton Java class* biasa yang bertugas untuk

melakukan *lookup* dan *caching* dari *EJBHome* yang dibutuhkan. Dengan demikian *lookup EJBHome* melalui JNDI hanya dilakukan pada saat pertama saja dan hasilnya kemudian diletakkan ke dalam *cache*, pemanggilan yang kedua dan seterusnya cukup mengambil data dari *cache*.

6. Business Delegate Pattern

Business Delegate Pattern merupakan teknik yang digunakan pada *client* EJB dengan memanfaatkan *helper class* sebagai penghubung antara *client* dan *session façade*. *Helper class* ini berfungsi untuk mengenkapsulasi serta mendelegasikan permintaan dari *client* ke *server* sekaligus menangani *exception* yang mungkin terjadi. Manfaat lain yang dapat diperoleh dengan mengimplementasikan *Business Delegate Pattern* ini adalah dapat melepaskan ketergantungan antara proses pembuatan dan pengujian *client application* dengan komponen EJB. Umumnya *Business Delegate Helper class* ini memiliki relasi satu satu dengan *session Bean* pada *session façade pattern*, artinya tiap satu *session Bean* memiliki koresponsensi dengan satu *Business Delegate Helper class*. Gambar 4 menunjukkan contoh *Business Delegate Helper class*.



Gambar 4. Contoh *Business Delegate Helper Class*

5. Perancangan MODEL Sistem

Sebelum melakukan perancangan terlebih dahulu didefinisikan kebutuhan *E-Forum* baik yang bersifat fungsional maupun non-fungsional. Kebutuhan yang bersifat fungsional diperoleh dengan mengidentifikasi pengguna yang dilambangkan sebagai *actor* pada notasi UML dan juga proses-proses yang ada dalam *E-Forum* yang dalam notasi UML disebut dilambangkan sebagai *use case*. Pengguna yang ada pada *E-Forum* dibedakan atas *Guest*, *Member*, *Moderator*, dan *Administrator*. Hak akses masing-masing kelompok pengguna ini dirancang untuk penggunaan dalam *E-Forum* pada umumnya. Berikut adalah uraian masing-masing kelompok pengguna tersebut.

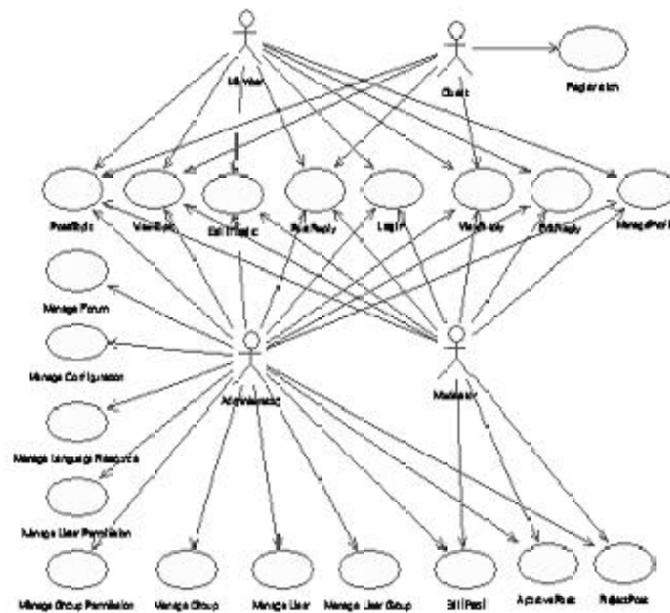
Guest adalah pengguna yang belum terdaftar pada *E-Forum*. Pengguna ini memiliki hak akses dengan level paling rendah pada aplikasi *E-Forum*. Hak dan kewenangan *guest* untuk melakukan *posting* atau *reading* dapat diatur secara global maupun secara spesifik pada tiap-tiap forum. *Guest* tidak memiliki hak untuk mengakses fitur-fitur yang berkenaan

dengan personalifikasi. Untuk dapat menjadi *Member* maka *Guest* harus melakukan pendaftaran terlebih dahulu.

Member adalah pengguna yang telah terdaftar pada *E-Forum*. Hak dan akses *Member* pada forum juga dapat diatur secara global maupun secara spesifik pada tiap-tiap forum. *Member* berhak untuk mengakses fitur yang berkaitan dengan personalifikasi seperti pengisian biodata, pemilihan layout/*theme* sesuai dengan yang disukai, dan memberikan *signature* pada *posting*.

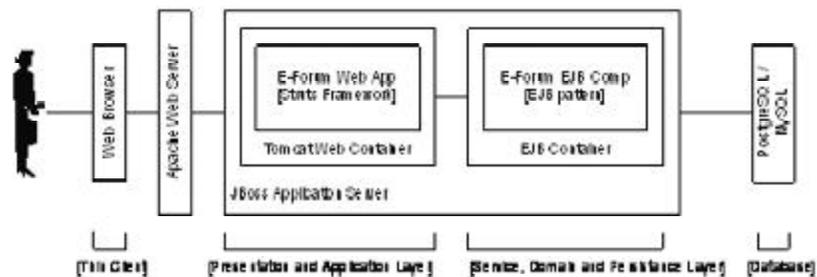
Moderator adalah pengguna yang ditunjuk oleh *Administrator* untuk memberikan verifikasi untuk setiap *posting* pada forum yang telah ditentukan. Verifikasi ini dapat berupa persetujuan/*approval*, penolakan/*rejection* maupun persetujuan dengan modifikasi pada setiap posting. Seorang *Member* dapat menjadi *Moderator* dari beberapa forum dan begitupun sebaliknya, setiap forum dapat memiliki satu atau lebih *Moderator*.

Administrator memiliki hak akses dengan level tertinggi pada *E-Forum*. Disamping memiliki hak untuk mengakses fitur-fitur umum yang dapat diakses baik oleh *Member* maupun *Moderator*, *Administrator* juga memiliki hak untuk mengakses fitur-fitur yang berkaitan dengan fungsi administrasi, seperti manajemen forum, manajemen *group*, manajemen pengguna, mengatur *permission*, mengatur bahasa, dan mengatur konfigurasi. Gambar 5 menunjukkan *use case E-Forum* yang dirancang dengan memperhatikan kebutuhan fungsional diatas.



Gambar 5. Use Case E-Forum

Kebutuhan yang bersifat non-fungsional adalah kebutuhan tambahan lainnya yang tidak langsung berhubungan dengan proses-proses yang ada pada E-Forum, yang meliputi kemampuan untuk *clustering*, mendukung *internationalization* serta dapat menggunakan berbagai *database*. Disini yang bertindak sebagai *client* adalah *web browser* atau biasa disebut *thin client*. Apache digunakan sebagai *load balancer* untuk mendistribusikan *request* ke beberapa simpul yang menjalankan E-Forum. JBoss bertindak sebagai *application server* yang di dalamnya terdapat Tomcat *web container* dan EJB *container* (JBoss Organization, 2002). Apabila dibutuhkan, Tomcat dapat pula berdiri sendiri sebagai *web container* yang terpisah dari JBoss (Apache Software Foundation, 2002). Database yang dapat digunakan adalah MySQL atau PostgreSQL, keduanya memiliki fungsi dan kapabilitas yang setara serta tingkat kompatibilitas yang tinggi. Arsitektur dari E-Forum secara umum dapat dilihat pada Gambar 6.



Gambar 6. Arsitektur E-Forum

Secara umum *layer* aplikasi pada E-Forum terbagi atas 5 bagian yaitu *presentation layer*, *application layer*, *service layer*, *domain layer*, dan *persistence layer*. *Presentation Layer* adalah *layer* yang memuat hal-hal yang berkenaan dengan *layout* dari E-Forum, diantaranya adalah HTML, JSP, Javascript, CSS, gambar, dan juga taglib yang berfungsi untuk pemformatan data.

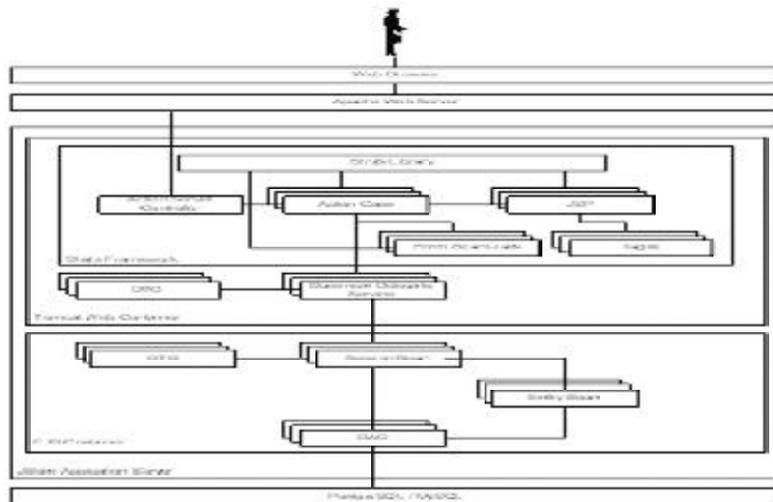
Application layer merupakan penghubung antara *presentation layer* dan *service layer*. *Layer* ini bertugas untuk mengatur *client state* (*HTTPSession*), melakukan validasi sintaks pada input *client*, dan melakukan *delegation* ke *service layer*.

Service layer merupakan titik utama untuk akses ke EJB. *Application layer* akan menggunakan *layer* ini untuk menjalankan suatu fungsi bisnis pada *use case* tertentu. *Service layer* selanjutnya meneruskannya pada satu atau beberapa *domain object* yang terkait dengan fungsi tersebut. *Service layer* juga mengatur transaksi yang dibutuhkan dalam eksekusi suatu fungsi bisnis. Pada *layer* inilah *Session Façade pattern* dalam bentuk *Stateless Session Bean* akan digunakan.

Domain layer merupakan *layer* dimana semua obyek yang muncul pada masalah analisa bisnis berada. Obyek-obyek ini direpresentasikan dalam bentuk *Entity Bean*. *Service layer* mendelegasikan *request* yang diterimanya ke *domain layer* untuk diproses lebih lanjut.

Persistence layer adalah *layer* dimana *logic* yang berkaitan representasi *domain object* pada *database* berada. Pada *layer* ini operasi-operasi yang berkaitan dengan *database* dilakukan, seperti *Insert*, *Delete*, *Update*, maupun *Select*. *Layer* akan diimplementasikan dengan menggunakan DAO.

Gambar 7 menunjukkan arsitektur detail E-Forum yang dibuat.



Gambar 7. Arsitektur Detil E-Forum

Komponen *E-Forum* terbagi atas dua komponen pokok, yaitu komponen *web* dan komponen EJB. Komponen *web* direalisasikan dengan menggunakan *Struts Application Framework* serta *EJBHomeFactory* dan *Business Delegate Pattern*. Sedang komponen EJB direalisasikan dengan *Session Façade*, *EJBHomeFactory*, DAO serta *JDBC For Reading pattern*. Selanjutnya untuk transfer data antara komponen *web* dan komponen EJB akan digunakan *Data Transfer Object / Value Object pattern*.

Perancangan komponen Web

Komponen web dibangun dengan *Struts Application Framework*. Untuk mempermudah hubungan dengan komponen EJB juga digunakan *EJBHomeFactory* dan *Business Delegate Pattern*. Perancangan untuk masing-masing komponen dapat diuraikan sebagai berikut:

1. Model

Komponen model pada *E-Forum* meliputi *Form Bean*, *ValueObject*, dan *Business Delegate classes*.

a. FormBean Classes

Form bean merupakan representasi form-form dalam pada halaman JSP dalam bentuk Java Bean. Misal *LoginForm*, *SearchForm* dan lain-lain.

b. ValueObject/DTO Classes

Value Object adalah *Java class* biasa yang digunakan untuk mengenkapsulasi data yang ditransfer antara *web* dan komponen EJB. Terdapat dua tipe *value object* yang digunakan pada *E-Forum*, yaitu *Domain Value Object* dan *Custom Value Object*. Contoh *Domain Value Object* pada aplikasi ini adalah *UserVO* yang merupakan representasi *User Entity Bean* pada EJB. Sedang contoh *Custom Value Object* pada aplikasi ini adalah *UserIdentifier VO* yang merupakan mengenkapsulasi beberapa karakteristik dari *User*. Semua *Value Object* harus mengimplementasi *Serializable Interface*, sehingga memungkinkan untuk ditransfer melalui *network*.

c. BusinessDelegate Classes

Business Delegates disini merupakan penghubung / *intermediate* antara komponen *web* dan komponen EJB. *Business Delegate Classes* akan meneruskan *request client* ke komponen EJB, dalam hal ini *Session Façade EJB*. Misal *AuthService*, *PostService* dan lain-lain.

2. View

Komponen *View* pada *E-Forum* meliputi halaman-halaman JSP, *Javascript*, *images*, *Cascading Style Sheet (CSS)* dan beberapa *taglib* yang berhubungan dengan *layout data*. *E-Forum* dapat memiliki beberapa *layout* yang berbeda. Dalam *E-Forum* terdapat dua buah *layout* yakni *default* dan *classic*.

3. Controller

Komponen *Controller* pada *E-Forum* meliputi *ForumActionServlet*, *ForumDispatchAction*, *Action Handler Classes*.

a. ForumActionServlet

Class ini merupakan *Class* turunan dari *ActionServlet* bawaan Struts dengan dengan penambahan beberapa fungsi seperti mengakses konfigurasi *E-Forum* dari EJB, inisialisasi *data caching*, dan sebagainya.

b. Action Handler Classes

Class ini merupakan *Class* turunan dari *ForumDispatch Action* yang akan menangani suatu request untuk diteruskan ke *model*. Misal *PostHandlerAction*, *RegistrationHandler*

Action, *SearchHandlerAction* dan lain-lain. *ForumDispatchAction* merupakan *class* turunan dari *DispatchAction* bawaan *Struts* yang memungkinkan pengelompokan beberapa *handler/action* dalam satu *class*.

Selain ketiga komponen utama diatas, dalam pembuatan komponen *web* juga diperlukan beberapa *helper class*, *custom taglib* dan *filter*. Untuk masalah *message resource* yang digunakan, *E-Forum* melakukan perubahan terhadap *behaviour default* dari *Struts* yang menggunakan *property file* menjadi *database*. Untuk beberapa data yang sering diakses atau membutuhkan waktu eksekusi *query* yang cukup panjang, dilakukan *caching* dengan memanfaatkan *Global JNDI context*.

Perancangan komponen EJB

Komponen EJB pada *E-Forum* terdiri atas dua tipe EJB, *Stateless Session Bean* dan *BMP Entity Bean*. *Stateless Session Bean* (SLSB) bertindak sebagai *session façade* yang merupakan *service layer* pada *E-Forum*. *BMP Entity Bean* berada pada *domain layer*. Sedang untuk *persistence layer* digunakan DAO. Berikut adalah uraian perancangan untuk masing-masing komponen.

1. Session Façade

Session façade merupakan penghubung antara komponen *web* dengan *Entity Bean*. *Session façade* biasanya menerima *request* dari *Business Delegates* yang berada pada sisi *client* untuk selanjutnya diteruskan pada satu atau lebih *Entity Bean*. *Business Delegates* yang berada pada sisi *client* memiliki korespondensi satu-satu dengan *Session Façade*. Misal *UserServiceEJB*, *AuthServiceEJB*, dan lain-lain. Setiap *Session Façade* dibentuk oleh dari tiga buah elemen: *LocalHome Interface*, *Local Interface*, dan *Bean class*-nya sendiri. Alasan menggunakan *local* untuk *home* dan *interface* dari *Session Façade EJB* adalah karena disini *web container* yang digunakan berjalan dalam *Virtual Machine* (VM) yang sama dengan *EJB container*.

2. BMP Entity Bean

BMP entity bean merupakan representasi dari obyek-obyek yang muncul pada masalah analisa bisnis. Pada *E-Forum*, *Entity Bean* tipe BMP dipilih karena memungkinkan pengembang aplikasi memiliki kontrol sepenuhnya terhadap *Entity Bean* yang dibuat, juga tidak tergantung pada *EJB container* tertentu, misalnya *UserEJB*, *ForumEJB*, dan lain-lain. Seperti halnya SLSB *Session Façade*, setiap *BMP Entity Bean* juga dibentuk oleh dari tiga buah elemen: *LocalHome Interface*, *Local Interface* dan *Bean class*-nya sendiri.

6. IMPLEMENTASI PERANGKAT LUNAK

Seperti telah dijelaskan pada bagian perancangan bahwa *E-Forum* tersusun atas 2 komponen yaitu komponen *web* dan komponen EJB. Komponen *web* dikembangkan dengan menggunakan *Struts Application Framework*. *Class* pembentuk komponen *web* maupun elemen-elemen lainnya seperti *deployment descriptor*, halaman JSP dan lain-lain dapat dilihat pada Gambar 8.

Komponen EJB dikembangkan dengan menggunakan beberapa *pattern* seperti yang telah dijelaskan dalam perancangan. *Class* pembentuk komponen EJB maupun elemen-elemen lainnya seperti *deployment descriptor*, file konfigurasi, SQL file, dan lain-lain dapat dilihat pada Gambar 9.

Untuk *class* yang digunakan baik oleh komponen *web* maupun komponen EJB, diletakkan dalam *net.forum.common*. Komponen *web* akan dikemas ke dalam file WAR (*Web Archive*) yaitu *web-forum.war*. Sedang untuk komponen EJB akan dikemas ke dalam file JAR (*Java Archive*) yaitu *ejb-forum.jar*. Keduanya kemudian digabungkan dalam satu file EAR (*Enterprise Archive*) yaitu *forum.ear* dan siap di-*deploy* ke *JBoss application server*.



Gambar 8. Class-class Pembentuk Komponen Web



Gambar 9. Class-class Pembentuk Komponen EJB

7. Hasil Uji Coba

Uji coba terhadap *E-Forum* terbagi atas dua jenis. Pengujian untuk beberapa fitur-fitur *E-Forum* yang bersifat fungsional seperti : autentikasi pengguna/login, pembuatan forum baru, pengaturan konfigurasi, penambahan topik baru dan pengaturan wewenang / hak akses pengguna. Disamping itu pengujian terhadap fitur-fitur yang bersifat non fungsional seperti *internationalization support*, *clustering support* dan *multi database support* juga dilakukan. Akan dievaluasi apakah fitur-fitur yang dimiliki *E-Forum* baik secara fungsional maupun non-fungsional telah berjalan dengan baik.

Untuk keperluan uji coba ini digunakan komputer dengan spesifikasi *hardware* AMD 2G dan memori 256M dengan sistem operasi *Windows 2000*. Aplikasi-aplikasi pendukung seperti *Apache 1.3.9*, *Tomcat 5.1*, *JBoss 3.2.3 Tomcat 4.1.29*, serta *database MySQL 4.1* dan *PostgreSQL 7.2* telah disiapkan sebelumnya. Pada komputer client disiapkan *browser Internet Explorer* dan *Mozilla*. Dalam uji coba fungsional akan dilakukan terhadap beberapa proses dan fitur yang ada pada *E-Forum*.

Login

Sebelum melakukan proses *login* maka semua pengguna dianggap sebagai *Guest* oleh *E-Forum*. Hak dan kewenangan yang dimiliki juga masih pada tingkat *Guest*. Setelah melakukan *login* maka pengguna tersebut memperoleh hak dan kewenangan baik sebagai *Member*, *Moderator*, atau *Administrator*. Menu yang ditampilkan pada bagian atas halaman disesuaikan dengan hak dan kewenangan pengguna. Halaman untuk *login* dapat dilihat pada Gambar 10. Menu untuk masing-masing kelompok pengguna dapat dilihat pada Gambar 11.

Gambar 10. Halaman untuk Login



Gambar 11. Menu untuk Masing-masing Kelompok Pengguna

Membuat Forum Baru

Fasilitas ini hanya dapat diakses oleh *Administrator*. Disini *Administrator* akan mengisi *parent* dari forum, nama, deskripsi, *order*, *forum moderation*, hak akses *Guest* dan hak akses *Member*. Apabila disini dipilih opsi *default* untuk *forum moderation*, hak akses *Guest* dan hak akses *Member* yang berlaku adalah aturan *parent*. Bila pada *parent* juga dipilih opsi *default* maka yang berlaku adalah aturan yang ditetapkan secara global di konfigurasi *E-Forum*. Halaman penambahan forum baru dapat dilihat pada Gambar 12.

Parent	: Windows Programming
Name	: Programming pada handphone Nokia
Description	: Tipe dan trik data hai programming di Nokia.
Order	: 11
Forum Moderation	: disabled
Guest Permission	: Default
User Permission	: Post

Save | Reset | Back

Gambar 12. Halaman Penambahan Forum Baru

Mengatur Konfigurasi E-Forum

Fasilitas ini hanya dapat diakses oleh *Administrator*. Disini *Administrator* dapat mengatur konfigurasi umum dari *E-Forum*. Konfigurasi mengenai SMTP server yang digunakan, konfirmasi pada pendaftaran, *default pagination*, *filter* yang diaktifkan, hak akses, pengaturan *moderation*, *default theme* serta *mode display* dapat diatur disini. Halaman konfigurasi *E-Forum* dapat dilihat pada Gambar 13.

Administrator Email	: rubby@localhost
SMTP Server	: localhost
Confirmation Required	: yes
Pagination	: 20
Filter escapeHTML	: yes
Filter NiZar	: yes
Filter BBCode	: yes
Filter Smilies	: yes
Forum Moderation	: disabled
Guest Permission	: Post
User Permission	: Post
Attachment File	: enabled
Attachment Max. Size	: 10240 kb
Default Template	: Default
Display Mode	: Threaded

Gambar 13. Halaman Konfigurasi E-Forum

Penambahan Topik Baru

Disini pengguna dapat menambahkan topik baru pada *E-Forum*. Pengguna mengisi *subject* beserta *content* dari topik yang akan di-*posting*. Pengguna juga dapat melakukan *upload* beberapa *file* yang berhubungan dengan topik. Gambar form untuk menambahkan topik baru dapat dilihat pada Gambar 14. Fasilitas HTML *toolbar* dan *smilies* serta *attachment* dapat dinonaktifkan pada konfigurasi *E-Forum*. Hasil dari topik yang telah ditambahkan dapat dilihat pada Gambar 15.



Gambar 14. Halaman untuk Penambahan Topik Baru



Gambar 15. Topik yang Telah Ditambahkan

Pengaturan Hak Akses User Pada Forum

Hak akses pada forum adalah bersifat hirarki, artinya secara *default* hak akses pada *parent* juga berlaku ke subforum-subforum dibawahnya. Kecuali bila pada masing-masing subforum dilakukan pengaturan hak akses tersendiri. Hal ini dapat dilihat pada Gambar 16.

Modify User Permission					
Name : zubby					
Username : zubby					
No.	Forum Name	Permission			
		Default	Deny	Read	Post
1.	Begining Java	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2.	Desktop Programming	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.	SWING Tips and Trick	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4.	SWT Technology	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5.	J2EE Development	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6.	Enterprise Java Bean (EJB)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7.	Servlets	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8.	Java Server Pages (JSP)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9.	Application Server	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10.	JBOS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11.	SONAS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12.	IBM Websphere	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13.	Bea WebLogic	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14.	Others	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15.	Wireless Programming	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Gambar 16. Pengaturan Hak Akses Pengguna

Fitur Internationalization

Dengan fitur ini, pengguna dapat memilih bahasa yang digunakan pada E-Forum, yaitu bahasa Inggris atau bahasa Indonesia. Gambar 17 menunjukkan tampilan halaman utama *E-Forum* ketika pengaturan bahasa pada *browser client* masih menggunakan bahasa Inggris. Selanjutnya Gambar 18 menunjukkan tampilan *E-Forum* saat *browser client* sudah diubah ke bahasa Indonesia. Uji coba ini dilakukan pada *browser Internet Explorer* dan *Mozilla*.

Welcome to E-Forum.
This forum is built using J2EE architecture with STRUTS Web Application Framework.
This application running under JBOSS 3.2.3 Application server and TOMCAT 5.0.16.

Forum	Topic	Post	Last Post
Begining Java New to java ? This forum will suits you.	1	1	04/07/2004 06:32:10 By : zubby
Desktop Programming Tips and trick to build desktop apps in java. Learn about SWING, SWT or other java technologies related to GUI here.	2	6	05/07/2004 21:26:30 By : zubby
SWING Tips and Trick	0	0	No Post
SWT Technology	0	0	No Post
J2EE Development Discuss about J2EE architecture and development, tools, best practices and everything related to enterprise and distributed computing here.	2	4	21/07/2004 13:08:29 By : zubby
Enterprise Java Bean (EJB)	0	0	No Post
Servlets	1	1	21/07/2004 13:08:29 By : zubby
Java Server Pages (JSP)	0	0	No Post
Application Server	0	0	No Post
Wireless Programming Want to develop java apps in mobile environment ? This forum is built for you.	0	0	No Post

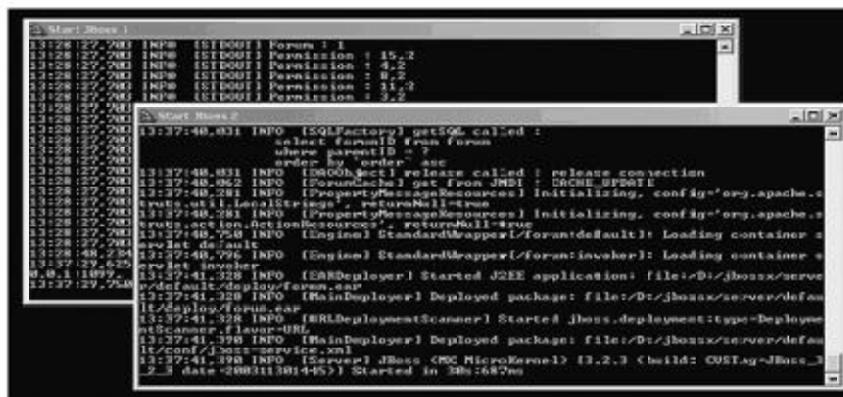
Gambar 17. Halaman Utama dalam Bahasa Inggris



Gambar 18. Halaman Utama dalam Bahasa Indonesia

Fitur Clustering

Untuk melakukan simulasi ini maka dijalankan 2 buah *instances JBoss* secara bersama pada komputer uji coba. Telah dilakukan penyesuaian pada JBoss kedua dalam hal port yang digunakan agar tidak terjadi konflik dengan JBoss pertama. Hal ini dapat dilihat pada Gambar 19. Terlihat bahwa *request* yang dilakukan beberapa pengguna terdistribusi secara merata pada kedua *instances JBoss*. Hal ini menunjukkan fungsi *load balancing* telah berjalan dengan baik. Uji coba juga dilakukan dengan mematikan *instances JBoss* kedua pada saat terdapat user yang telah *login*. Secara otomatis *instance JBoss* yang pertama mengambil alih fungsi *instance JBoss* kedua yang dimatikan, dengan tetap menjaga *state* dari pengguna. Hal ini menunjukkan fungsi *HttpSession replication* telah berjalan dengan baik.

Gambar 19. Dua *Instances JBoss* yang Berjalan Bersamaan

Fitur Multi Database

Database yang disupport oleh *E-Forum* adalah *MySQL* dan *PostgreSQL*. Untuk dapat menggunakan suatu *database* tertentu maka perlu dilakukan beberapa perubahan secara manual pada konfigurasi *E-Forum* dan *JBoss*. Kemudian dilakukan *packaging* dan *deploy* ulang aplikasi *E-Forum* ke *JBoss*. Setelah dilakukan uji coba terhadap masing-masing *database*, aplikasi *E-Forum* dapat berjalan dengan baik untuk masing-masing *database*, baik *MySQL* maupun *PostgreSQL*. Untuk dapat menggunakan *database* lainnya maka perlu dibuat implementasi dari DAO untuk masing-masing *database* yang ingin digunakan.

6. KESIMPULAN

Kesimpulan yang dapat diambil dari perancangan dan pembuatan *E-Forum* ini adalah sebagai berikut:

1. Penggunaan EJB *Design Pattern* dalam pengembangan aplikasi dapat mempermudah pengembang aplikasi dalam pembuatan komponen EJB pada *E-Forum*.
2. Dalam pembuatan komponen EJB *E-Forum* digunakan *Stateless Session Bean* sebagai *Session Façade* dan *Entity Bean* tipe BMP yang memungkinkan kontrol yang lebih baik terhadap *database*.
3. Fitur-fitur *E-Forum* secara umum berjalan dengan baik sesuai dengan yang ditetapkan.
4. *Clustering support* berjalan dengan baik, sehingga fungsi *load balancing* dan *HttpSession replication* juga berjalan baik.
5. *Internationalization support* berjalan dengan baik untuk pemilihan bahasa yang digunakan pada *E-Forum*, yaitu bahasa Inggris atau bahasa Indonesia pada *browser Internet Explorer* dan *Mozilla*.
6. *E-Forum* yang dibuat dapat menggunakan dua *database*, yakni *MySQL* dan *PostgreSQL*. Untuk dapat menggunakan *database* lainnya maka perlu dibuat implementasi dari DAO untuk masing-masing *database* yang ingin digunakan.
7. Dengan menggunakan struktur hirarki dalam pengaturan hak akses memungkinkan *E-Forum* untuk melakukan pengaturan *security* dan pendistribusian wewenang kepada pengguna.

DAFTAR PUSTAKA

1. Apache Software Foundation (2002). *The Tomcat 4 Servlet/JSP Container-Documentation*. Diakses pada 8 Februari 2008 dari : <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/index.html>.
2. Apache Software Foundation (2004). *The Apache HTTP Server Project*. Diakses pada 8 Februari 2008 dari : <http://http.apache.org/>.
3. Apache Software Foundation (2007). *The Struts User's Guide*. Diakses pada 5 Februari 2008 dari : <http://jakarta.apache.org/struts/userGuide/index.html>.
4. Hall, Marty, et. al. (2007). *Core Servlets and Javasever Pages: Advanced Technologies, Vol. 2*. Second edition. Prentice Hall.
5. JBoss Organization (2002). *JBoss 2.4+ Documentation*. Diakses pada 24 Februari 2008 dari : www.huihoo.org/jboss/online_manual/2.4/index.html.
6. Marinescu, Floyd (2002). *EJB Design Patterns: Advanced Patterns, Processes, and Idioms*. Wiley Computer Publishing.
7. PostgreSQL Global Development Group (2007). *The PostgreSQL Reference Manual Volume 3: Server Administration Guide (The Postgresql Reference Manual)*. Network Theory Ltd.
8. Roman, Ed, et. al. (2004). *Mastering Enterprise Java Beans*. Thrid Edition. Wiley Computer Publishing.
9. Sun Microsystems (2002). *J2EE 1.3 Tutorial*. Diakses pada 3 Februari 2008 dari : http://java.sun.com/j2ee/tutorial/1_3-fcs/.
10. Widenius, Michael, et. al. (2002). *MySQL Reference Manual*. O'Reilly Media, Inc.