

PENANGANAN PROYEK PENGEMBANGAN PERANGKAT LUNAK DENGAN ALFRESCO

Choirul Ikhwan¹, Hendrik²

Laboratorium Sistem Informasi dan Rekayasa Perangkat Lunak (SIRKEL)
Jurusan Teknik Informatika, Fakultas Teknologi Industri,
Universitas Islam Indonesia
e-mail: ¹aristhu_oracle@yahoo.com, ²hendrik@fti.uii.ac.id

ABSTRAKSI

Komunikasi antar role dan divisi dalam lembaga, pembagian kerja yang jelas dan penetapan aturan untuk setiap role menjadi faktor penunjang kinerja suatu organisasi dalam melayani konsumen semaksimal mungkin. Demikian pula dengan suatu proyek pengembangan perangkat lunak, diperlukan sebuah tim yang dapat bekerja sama dan berkomunikasi dengan baik antar anggotanya agar menghasilkan hasil akhir/produk yang berkualitas. Untuk itu diperlukan adanya sistem terintegrasi yang dapat mengakomodir semua kebutuhan tersebut.

Alfresco, sebagai salah satu open source – Enterprise Content Management System (ECMS), dapat menjadi salah satu alternatif sistem terintegrasi ini.

Berdasarkan pengujian yang dilakukan, pemanfaatan Alfresco sebagai ECMS dalam mengelola suatu proyek pengembangan perangkat lunak cukup efektif, terutama berkaitan dengan collaborative work. Selain itu Alfresco cukup mudah digunakan.

Kata kunci: Enterprise Content Management System, Alfresco

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Seiring dengan perkembangan zaman, kebutuhan manusia semakin lama semakin kompleks. Hal ini menjadi tantangan bagi semua lembaga yang melayani masyarakat baik yang bersifat profit maupun non-profit, *government* maupun *non-government*, dalam memenuhi kebutuhan masyarakat tersebut.

Faktor utama yang menjadi fokus dalam memberikan pelayanan adalah kepuasan konsumen dalam ranah apapun. Ada beberapa faktor yang dapat menjadi penentu kepuasan konsumen di antaranya adalah kemudahan dan kecepatan proses pelayanan dari awal hingga akhir.

Efektivitas dan efisiensi kerja dari sisi sumber daya lembaga juga harus dipertimbangkan. Komunikasi antar *role* dan divisi dalam lembaga, pembagian kerja yang jelas dan penetapan aturan untuk setiap *role* menjadi faktor penunjang kinerja lembaga untuk melayani konsumen semaksimal mungkin.

Berdasarkan beberapa faktor di atas, perlu adanya sistem terintegrasi yang mengakomodir semua kebutuhan tersebut. Dalam hal ini *Enterprise Content Management System* (ECMS) dapat menjadi salah satu pilihan solusinya.

Salah satu ECMS yang dapat digunakan adalah Alfresco. Dengan berbagai fitur yang dimiliki, Alfresco dapat menjadi kakas untuk menunjang peningkatan kinerja lembaga secara internal maupun eksternal untuk melayani kebutuhan konsumen. Dalam tulisan ini, Alfresco digunakan dalam konteks penanganan proyek perusahaan

pengembangan perangkat lunak mulai dari tahap inisialisasi sampai finalisasi.

1.2 Rumusan Masalah

Proyek pengembangan perangkat lunak di sini berupa pengembangan sistem informasi akademik untuk suatu universitas.

Makalah ini akan memaparkan bagaimana menggunakan Alfresco untuk menangani alur kerja pembuatan perangkat lunak dalam internal tim proyek.

1.3 Batasan Masalah

Adapun batasan permasalahan yang dibahas meliputi:

- Pemanfaatan Alfresco hanya difokuskan pada internal tim dan tidak mencakup aktor-aktor di luar tim, seperti divisi lain yang kemungkinan terkait dengan proyek yang dikerjakan.
- Role* yang berperan dalam proyek adalah *role* umum yang ada dalam pengembangan perangkat lunak sesuai dengan fungsionalitasnya namun tidak semuanya diakomodir. Adapun *role* yang digunakan sebagai contoh di sini adalah *Project Manager*, *System Analyst*, *System Designer*, *Programmer*, dan *Database Engineer*.

2. TINJAUAN PUSTAKA

2.1 Enterprise Content Management System (ECMS)

Definisi ECM menurut AIIM adalah: “*Enterprise Content Management (ECM) is the technologies used to Capture, Manage, Store, Preserve, and Deliver content and documents*

related to organizational processes. ECM tools and strategies allow the management of an organization's unstructured information, wherever that information exists". [2]

ECM merupakan teknologi yang digunakan untuk membuat, mengelola, menyimpan, menjaga dan mengirimkan dokumen yang berhubungan dengan organisasi. Selain itu ECM juga dapat digunakan untuk mengumpulkan informasi-informasi terkait yang belum terstruktur dari berbagai sumber.

Berdasarkan definisi di atas, setidaknya ada beberapa komponen dasar ECM, yaitu:

a. Capture

Capture menyediakan berbagai macam fungsionalitas dan komponen untuk memperoleh informasi analog maupun elektronik mulai dari informasi sederhana hingga kompleks.

b. Manage

Manage digunakan untuk mengelola pemrosesan dan penggunaan informasi yang diperoleh. Hal ini diimplementasikan dalam *Document Management, Collaboration, Web Content Management, Record Management, dan Workflow*.

c. Store

Store merupakan komponen yang digunakan sebagai tempat penyimpanan informasi arsip. *Store* dapat berupa:

- **Repositories**
Merupakan tempat penyimpanan yang disediakan oleh sistem.
- **Library Services**
Dalam ECMS, *library service* dapat berupa *version management*, atau mekanisme *check in/check out*.
- **Storage Technologies**
Dapat berupa *Tape Media* atau *Optical Media*.

d. Preserve

Komponen *preserve* digunakan untuk menangani informasi yang tidak digunakan sebagai arsip. *Preserve* dapat menggunakan media elektronik maupun non-elektronik.

e. Deliver

Komponen *deliver* digunakan untuk menyampaikan informasi dalam format tertentu. *Deliver* merupakan gabungan dari tiga komponen yang berbeda, yaitu:

- **Transformation Technologies**
Teknologi yang digunakan untuk mengontrol dan mengikuti informasi yang dikirimkan. Teknologi ini dilakukan sebagai proses *background*. Teknologi ini dapat berupa *XML, Personalization, Syndication*, dan lain-lain.

- **Security Technologies**

Teknologi yang digunakan untuk melakukan pengecekan terhadap otorisasi informasi yang dikirimkan.

- **Distribution**

Merupakan media tempat informasi dipublikasikan. *Distribution* dapat berupa: Internet, E-mail, dan lain-lain.

2.2 Alfresco

Alfresco merupakan suatu perangkat lunak *open source Enterprise Content Management System (ECMS)* yang dapat memberikan modularitas tinggi dan performa memadai. Alfresco dapat digunakan sebagai manajemen konten untuk dokumen, web, *record*, gambar dan *Collaborative Development Environment*.

Alfresco dibuat pada tahun 2005 oleh John Newton (pendiri Documentum, sebuah produk ECMS komersial) dan John Powell. Awalnya, Alfresco didukung oleh para pengembang dari Documentum dan Oracle dengan memfokuskan produknya pada *document management*. Pada Mei 2006, Alfresco difokuskan menjadi *web content management* dengan mengambil orang-orang dari Interwoven. Pada tahun 2007, Alfresco mengambil orang-orang dari Vignette. Pada Maret 2007, CMS Watch (perusahaan yang bertugas mengevaluasi perangkat lunak), memperkenalkan Alfresco sebagai salah satu *Web Content Management*. [1]

Alfresco berlisensi *General Public License (GPL)*. Alfresco tidak tergantung pada web browser, sistem operasi, server maupun DBMS tertentu.

Sebagai ECMS, Alfresco menyediakan beberapa fitur yang telah didefinisikan oleh AIIM untuk persyaratan dalam kategori ECM, antara lain :*Document Management, Web Content Management, Versioning, Record Management, Image Management, AJAX suport, Integrated Publishing, CIFS, Workflow, Multi-language support, Portable application packaging, Multi-platform support*.

Ada beberapa komponen utama Alfresco, yaitu:

- Space
- Pengguna/*User*, Roles, Group.
- Workflow
- Rules
- Content

Gabungan komponen-komponen utama tersebut yang membentuk Alfresco menjadi sebuah ECMS.

2.3 Pemodelan dengan UML

UML atau *Unified Modelling Language* merupakan bahasa pemodelan grafis yang digunakan untuk membantu dalam proses analisis dan perancangan dengan metode berorientasi objek. UML lahir dari penggabungan beberapa bahasa pemodelan yang berkembang pesat pada akhir 1980-an dan awal 1990-an.

Saat ini UML telah menjadi standar bahasa pemodelan berorientasi objek dan berada di bawah kendali OMG (*Object Management Group*), yakni sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk guna menghasilkan standar-standar yang mendukung interoperabilitas, terutama interoperabilitas sistem berorientasi objek.[3]

3. ANALISIS DAN PERANCANGAN

3.1 Requirement Gathering

Pada proyek pengembangan perangkat lunak ini terdapat beberapa pengguna dengan *role* yaitu: *Project Manager* (PM), *System Designer* (SD), *Programmer* (Pr), *Database Engineer* (DBE), *Graphic Designer* (GD) dan *Project Officer* (PO).

PM melakukan inisialisasi proyek dengan melakukan *requirement gathering* bersama klien untuk mendapatkan kebutuhan sistem. Dari hasil *requirement gathering* tersebut, PM melakukan penjadwalan dan pengaturan sumber daya proyek dengan berkoordinasi dengan SA.

SA melakukan analisis terhadap kebutuhan sistem berupa bisnis proses, kemudian menuangkannya dalam bentuk dokumentasi sistem. SD mempelajari dokumentasi sistem yang dibuat oleh SA, kemudian menuangkannya dalam bentuk pemodelan UML. Sedangkan DBE membuat *Entity Relationship Diagram* (ERD) sebagai rancangan skema basisdata dengan berkoordinasi dengan SD.

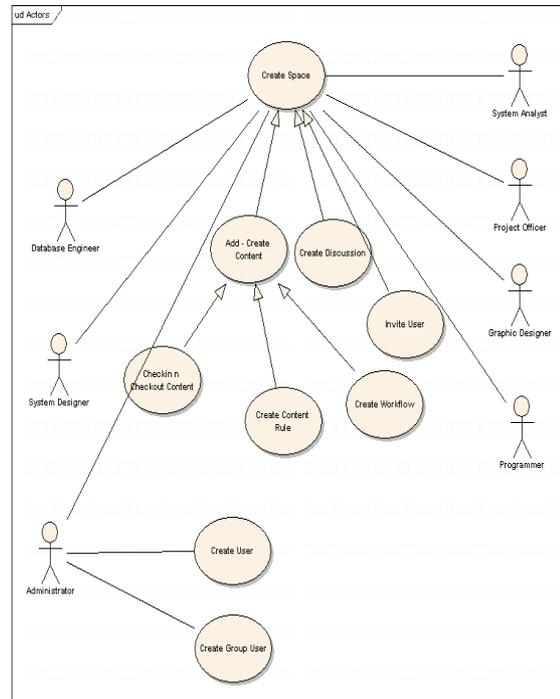
SD memberikan tugas kepada Pr untuk mengimplementasikan sistem dengan mengacu pada model yang telah dibuat. Dalam pembuatan aplikasi, Pr berkoordinasi dengan DBE terkait dengan query-query yang diperlukan pada sistem. Selain itu Pr berkoordinasi dengan GD terkait dengan antarmuka aplikasi yang akan dibuat.

Setiap satu modul selesai, PO bertugas melakukan pengujian modul untuk melihat kemungkinan-kemungkinan adanya *bug* atau kesalahan alur program. Setelah proses pengembangan selesai, PM dan beberapa tim teknis yang diperlukan melakukan instalasi perangkat lunak di tempat klien.

3.2 Diagram use-case

Diagram *use-case* digunakan untuk menggambarkan fungsionalitas sistem dari sudut pandang pengguna. Pada gambar 1 terlihat bahwa terdapat aktor-aktor sesuai *role* yang berperan dalam proyek pengembangan perangkat lunak. Selain itu nampak bahwa aktor-aktor tersebut berinteraksi dengan *use-case* yang menggambarkan fungsionalitas yang dapat dilakukan pengguna dengan menggunakan Alfresco, di antaranya adalah membuat *space* yang merupakan generalisasi dari *use-case add-create content*, *create discussion*, dan *invite pengguna*. *Add-create content* konten sendiri merupakan bentuk general dari *use-case Check-in n Checkout content*, *create content rule*, dan *create workflow*. Adapun untuk aktor admin, yang dalam hal ini mengelola penggunaan Alfresco, berinteraksi

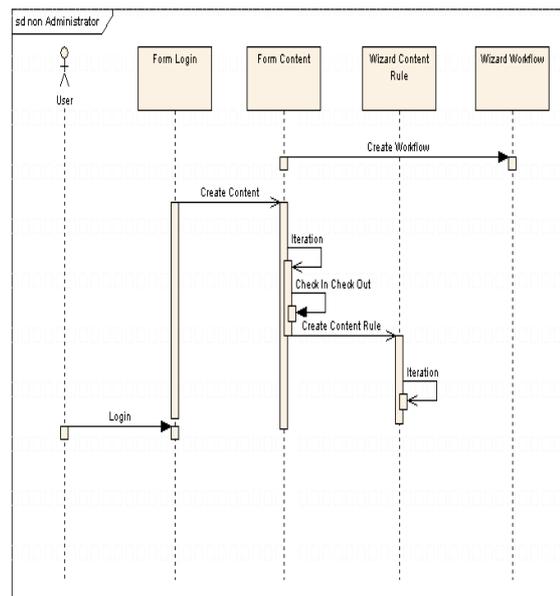
dengan *use-case create pengguna* dan *create group pengguna*.



Gambar 1. Diagram *use-case* pemanfaatan Alfresco dalam pengelolaan proyek pengembangan perangkat lunak.

3.3 Sequence Diagram

Diagram ini digunakan untuk menggambarkan urutan interaksi pengguna dengan sistem berdasarkan waktu. Pada gambar 2, ditunjukkan diagram *sequence* untuk aktor-aktor non admin.



Gambar 2. Diagram *sequence* untuk aktor-aktor non admin

4. IMPLEMENTASI DAN PENGUJIAN

Implementasi yang dilakukan merujuk pada tutorial dan dokumentasi-dokumentasi dari situs resmi Alfresco di antaranya adalah [5], [6], dan [7].

4.1 Implementasi Space

Space dibuat berdasarkan *role* yang terlibat pada fase proyek.

Selain itu, *space* dibuat sebagai lokasi penyimpanan hasil kerja semua setiap *role*. Ada beberapa *space* terkait dengan hal ini, yaitu : **Design System** sebagai *workspace* bagi role SD, **Documentation** untuk penyimpanan semua dokumentasi proyek, **Implementation** sebagai *workspace* untuk penyimpanan kode-kode modul dan query, **UI design** sebagai *workspace* untuk penyimpanan template HTML, CSS dan Javascript dan **Discussion** untuk *space* diskusi dari semua anggota tim.

Pembuatan *space* baik untuk *role* maupun sebagai *workspace* dilakukan oleh administrator.

4.2 Implementasi pengguna

Implementasi pengguna dilakukan dengan membuat beberapa pengguna yang ditempatkan pada *space* sesuai *role*-nya masing-masing. Area kerja pengguna ini berada pada *space* project yang telah didefinisikan sebelumnya.

4.3 Implementasi Roles

Masing-masing pengguna yang telah dibuat diberi *role* tertentu, sehingga pengguna tersebut memiliki hak akses pada *space* yang di-grant-kan kepadanya. Implementasi dilakukan dengan memberikan *privilege* tertentu pada *role* yang telah dibuat, yakni sebagai kontributor terhadap *space* yang menjadi haknya. Sebagai contoh, PM, SA, dan PO memiliki *role privilege* sebagai kontributor pada **Documentation space**. SD dan DBE memiliki *role privilege* sebagai kontributor pada **Design System**. Pr memiliki *role privilege* sebagai kontributor pada **Implementation**, dan semua *role* memiliki *privilege* sebagai kontributor pada **Discussion**. Sedangkan administrator merupakan koordinator proyek. Pengesetan *privilege* pada *role* tertentu dilakukan melalui fasilitas *manage space* pengguna.

4.4 Implementasi Versioning

Versioning dilakukan pada konten yang dibuat pada *space* untuk konten secara *inline* maupun konten yang di-upload dari *local drive*. Untuk melakukan *versioning* dalam Alfresco dilakukan dengan melihat detail *content space* dan aktifkan link **Allow Versioning History**. Gambar 3 merupakan *screenshot* contoh implementasi *versioning*. File default yang ditampilkan sistem adalah file dengan revisi terbaru. Tetapi semua file lama yang di-record oleh sistem masih dapat dipanggil kembali.

4.5 Implementasi Rule

Rule pada contoh ini dapat diimplementasikan dengan misalnya seorang Pr membuat *sub space* pada *space* miliknya dan meletakkan *working copy* file-file proyeknya pada *sub space* tersebut.

Agar Pr tidak perlu menggandakan file yang dibuatnya pada *working copy* ke *repository*, maka perlu dibuat sebuah *rule* yang melakukan penggandaan setiap konten yang masuk pada *sub space* ke *repository*.

Dalam hal ini, **condition** untuk *rule* adalah **inbound All Item**, yaitu berlaku untuk semua item yang masuk pada *space*. Sedangkan **action**-nya adalah **copy content to specified space**. Adapun tampilan implementasi *rule* ini dapat dilihat pada gambar 4.



Version	Notes	Author	Date	Actions
1.0		choiril	12 May 2007 06:48	View
1.1		choiril	12 May 2007 06:54	View

Gambar 3. Tampilan sejarah versi sebuah konten



Title	Description	Local	Created Date	Modified Date	Actions
-------	-------------	-------	--------------	---------------	---------

Gambar 4. Tampilan halaman pembuatan *rule* sebuah konten

4.6 Implementasi Workflow

Dalam contoh penanganan proyek pengembangan perangkat lunak ini, semua *workflow* yang didefinisikan oleh sistem dapat digunakan. Misalnya *workflow Adhoc-task* antar *role*. Contoh kasus, misalnya SD memberikan *task* pada Pr untuk mengerjakan suatu modul sesuai dengan *diagram class* yang dibuat oleh SD, seperti ditunjukkan pada gambar 5.

Contoh lain misalnya *workflow Review & Approve* dokumentasi yang diberikan oleh PO untuk PM. Apabila dokumentasi yang diberikan telah sesuai dengan POC, maka dokumentasi bisa diberikan kepada klien.



Description	Type	Id	Created	Completed on	Outcome	Actions
Buat Implementasi kode untuk diagram class berikut	Adhoc Task	2	12 May 2007 15:00	12 May 2007	Task Done	

Gambar 5. Tampilan halaman implementasi *workflow*

4.7 Implementasi Discussion

Semua topik diskusi diletakkan dalam *discussion space*. Hanya pengguna sebagai *contributor*, *collaborator* dan *coordinator* saja yang dapat memberikan tanggapan pada suatu topik diskusi. Untuk mengaktifkan diskusi, pengguna cukup memilih menu **Start Discussion**.

4.8 Implementasi Syndication

Sindikasi berita dapat diterapkan pada *space* maupun konten. Sindikasi dilakukan dengan mengaktifkan *RSS syndication* pada halaman detail dari suatu *space* atau konten. Agar bisa diakses oleh pengguna apapun, selain mengaktifkan *RSS syndication*, *owner* juga harus meng-*invite user guest* pada *space* tersebut. Kemudian alamat url lokasi sindikasi di letakkan pada perangkat lunak pembaca *RSS syndication*. Berikut contoh url dari suatu *space*:

```
http://localhost:8080/alfresco/guestTemplate/w  
orkspace/SpacesStore/ec72fff4-002d-11dc-89e1-  
3dcd5d24250e/workspace/SpacesStore/82d0ffe1-  
ff6b-11db-8c97-  
3bccf2e8553b/rss.xml?mimetype=text%2Fxml
```

5. KESIMPULAN

Dari pembahasan dan analisis yang dilakukan, dapat disimpulkan bahwa:

1. Penggunaan Alfresco untuk penanganan proyek pengembangan perangkat lunak cukup efektif khususnya yang berkaitan dengan *collaborative work*.
2. Alfresco cukup mudah digunakan, namun dari sisi *development/kustomisasi* relatif lebih sulit karena setiap penambahan/perubahan pada modul harus me-*restart* server aplikasi agar perubahan yang dilakukan efektif.

PUSTAKA

- [1] Alfresco, di <http://wiki.alfresco.com>, diakses tanggal 20 April 2007.
- [2] *Electronic Content Management System*, di <http://en.wikipedia.org/wiki/ecms>, diakses tanggal 20 April 2007.
- [3] Fowler, Martin. *UML Distilled, Edisi 3: Panduan Singkat Bahasa Pemodelan Objek Standar*. Yogyakarta: Penerbit Andi, 2005.
- [4] Miller, Randy, *Practical UML: A Hands-On Introduction for Developer*, di <http://bdn.borland.com/article/0,1410,31863,00.html>, diakses tanggal 13 April 2007.
- [5] Shariff, Munwar TT, *Alfresco Enterprise Content Management Implementation*, di <http://packtpub.com/alfresco/book> (diakses: 13 April 2007).
- [6] _____, *Alfresco Getting Started Guide, 2006*
- [7] _____, *Creating and Customizing Workflow, 2006*.