DIGITAL RIGHTS MANAGEMENT SEBAGAI SOLUSI KEAMANAN DOKUMEN ELEKTRONIK

Agus Kurniawan

Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia
agusk@cs.ui.ac.id

Abstrak

Perkembangan dokumen elektronik yang pesat dan mudah diakses berdampak semakin sulitnya dalam membatasi informasi ketika ingin membatasi akses terhadap informasi pada dokumen elektronik. Oleh karena itu, diperlukan mekanisme untuk menjaga informasi pada dokumen elektronik hanya dapat diakses oleh yang berhak dan tersimpan dengan aman. Pendekatan solusi dapat menerapkan *Digital Rights* Management pada dokumen elektronik sehingga pemilik dokumen elektronik aka lebih aman menyimpannya ataupun melakukan *sharing* ke partnernya.

Kata kunci : *Digital Rights Managements*, *Security System*

1. Pendahuluan

Perkembangan komputer yang makin pesat mempengaruhi kebiasaan seseorang terutama pada kebutuhan dokumen. Dokumen konvensional yang ditulis diatas kertas lambat laun bergeser dengan dokumen elektronik [1,2]. Bahkan dokumen elektronik sekarang sudah mempunyai kekuatan hukum. Hal ini membuat naiknya pemakaian dokumen elektronik.

Sifat dokumen elektronik yang fleksibel untuk diedit, digandakan ataupun didistribusikan membuat semakin banyak orang cenderung bekerja pada dokumen berbasis elektronik dibandingkan dengan bekerja dengan dokumen konvensional.

Pada saat ini, begitu banyak dokumen elektronik yang dibuat dengan mudah melalui komputer seiring berkembangnya aplikasi pengolah data dokumen. Pada kasus tertentu, beberapa dokumen tersebut mungkin ada dokumen yang dapat dikategorikan sebagai dokumen rahasia. Dokumen ini hanya dapat dibaca atau diberikan pada orang-orang tertentu. Oleh karena itu diperlukan mekanisme atau metode bagaimana mengamankan dokumen elektronik.

1.1. Tujuan

Tujuan utama yang hendak dicapai adalah mendesain suatu model sistem untuk mengamankan dokumen eletronik dari orang yang tidak mempunyai hak mengakses.

2. Landasan Teori

2.1. *Digital Rights Management*

Digital Right Management (DRM) *system* adalah istilah yang digunakan untuk mengatur data digital dan memproteksinya dari user yang tidak mempunyai hak akses [3]. DRM dapat berasal dari banyak bentuk antara lain:

- Dokumen
- Gambar
- Musik
- Video
- dan sebagainya

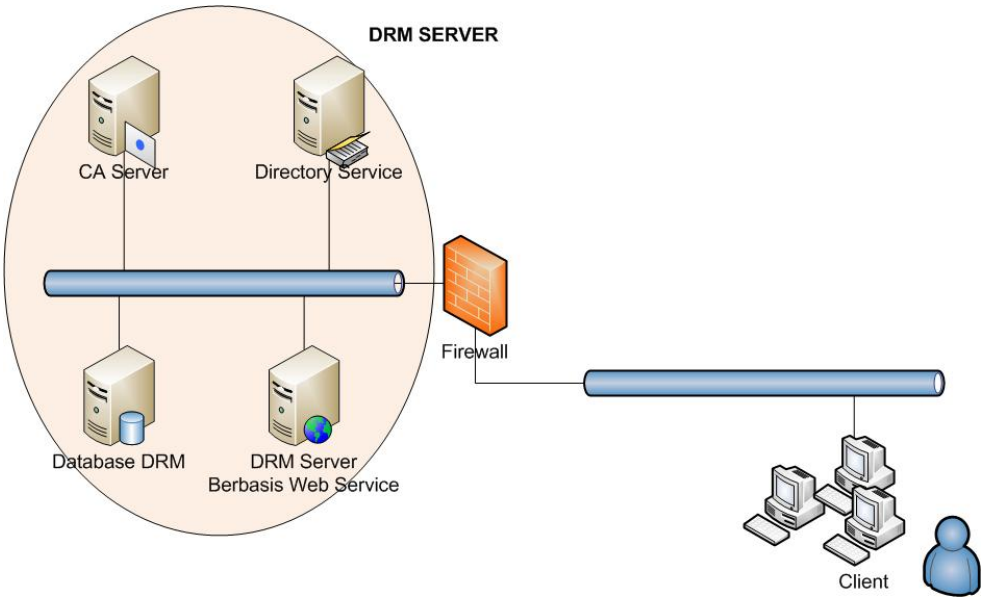
Data digital ini selanjutnya oleh pemiliknya diamankan agar hanya orang-orang tertentu saja yang dapat mengaksesnya.

Penerapan DRM dapat melibatkan banyak metode bahkan setiap *vendor* besar seperti Sony, Microsoft, Apple, dan Adobe mempunyai mekanisme sendiri. Sebagai contoh, Microsoft menerapkan DRM pada produk Windows Media, *Operating System* Windows dan Microsoft Office [4].

2.2. Metadata

Beberapa metode untuk menerapkan DRM pada data digital dengan memasukkan informasi tertentu pada bentuk metadata kedalam data tersebut. Informasi metadata biasanya meliputi nama, informasi *account* atau *e-mail*.

Metadata juga diterapkan pada data komersial contohnya Apple's iTunes yang meletakan data DRM yang dimasukkan kedalam MPEG *standard metadata* [5].



Gambar 1. *Design logical* pada sistem DRM

3. Analisa

3.1. Design Umum

Setiap dokumen elektronik yang telah dibuat dan ingin memproteksi informasi didalamnya maka dokumen elektronik tersebut dapat dilakukan proses DRM. Proses ini melibatkan enkripsi berbasis *Cryptography* yang mempunyai *public key* dan *private key*. Sistem yang menghasilkan *public* dan *private key* dapat berperan sebagai *license server*.

Secara umum, implementasi aplikasi yang menerapkan DRM dapat menggunakan pendekatan arsitektur *client-server* seperti terlihat pada Gambar 1. *Server* disini menyediakan servis untuk melayani:

- Menghasilkan *public* dan *private key* setiap dokumen elektronik yang akan diterapkan DRM
- Menghasilkan *license* yang berisi informasi hak apa saja dapat ditujukan pada dokumen tersebut
- Menyimpan semua data DRM setiap dokumen. Dalam hal ini dilakukan pada bagian *database server*

Semua komponen yang terlibat dimasukkan ke dalam domain direktori yang sama sehingga sistem authentikasi dan authorisasi dapat dikontrol secara terpusat.

3.2. DRM *Server*

DRM *Server* merupakan bagian terpenting untuk melakukan proses DRM pada suatu dokumen elektronik. Komponen ini bertanggung jawab mengatur dokumen elektronik yang akan diamankan termasuk manajemen *cryptography key*. Komponen DRM *Server* meliputi;

- DRM *Server* berbasis *Web Service*
- DRM *Database Server*
- *Directory Service*
- CA *Server*

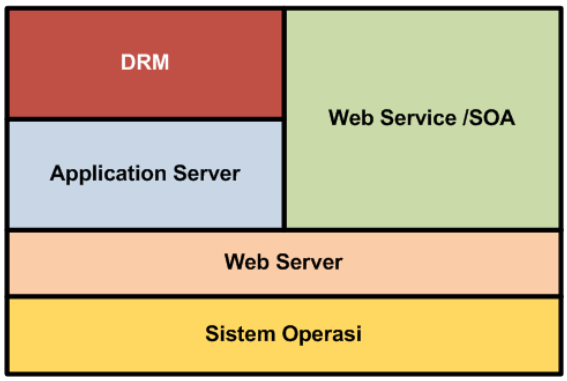
Masing-masing *item* akan dijelaskan pada subjudul selanjutnya.

3.2.1. DRM *Server* Berbasis *Web Service*

DRM *Server* berbasis *Web Service* berfungsi sebagia *web interface* antara *server* dan *client*. Penerapan model *Web Service* diharapkan sistem dapat berfungsi SOA (*Service-Oriented Architecture*). *Client* yang ingin berkomunikasi dengan DRM *Server* cukup menerapkan *standard* SOA yang telah didefinisikan.

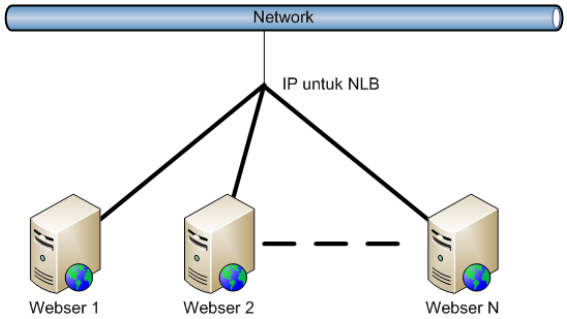
Desain untuk DRM *Server* berbasis *web service* dapat dilihat pada Gambar 2. Aplikasi DRM *server* berjalan diatas suatu *web server*. Realisasi DRM *server* memerlukan komponen antara lain

- *Application Server*
  Aplikasi ini sebagai *container* untuk aplikasi *web server* contohnya dalam hal ini adalah ASP.NET, PHP dan JSP
- DRM
  Aplikasi modul yang menyediakan hal-hal yang berhubungan dengan DRM
- *Web Service* / SOA
  Modul atau *library* yang menyediakan kontruksi dan dekontriksi skema SOA. Modul ini juga bertindak sebagai *interface* antara aplikasi *server* DRM dan *client* yang mengkonsumsi DRM.



Gambar 2. Design DRM *server* berbasis *web service*

Apabila *user* yang menggunakan aplikasi ini sangat banyak atau intensitas pembuatan dokumen yang diamankan cukup banyak maka DRM *server* pada *web server* dapat dipertimbangan untuk diterapkan *clustering*. Cara umum yang dapat digunakan adalah *Network Load Balancing* (NLB). Metode ini sangat cocok untuk *web server*. Implementasi dapat dilihat pada Gambar 3.



Gambar 3. DRM *Web server* untuk NLB

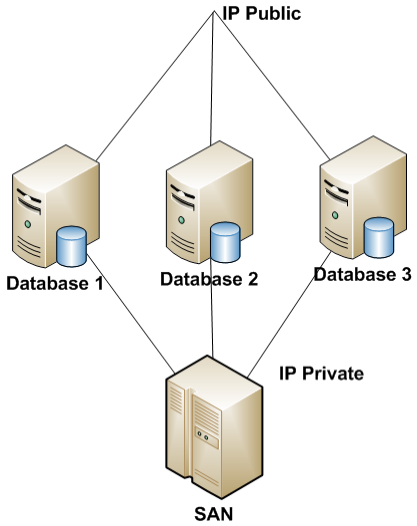
Implementasi NLB pada *web server* memerlukan satu IP virtual yang mana semua yang ingin mengakses *web server* harus melalui IP virtual ini.

3.2.2. DRM *Database Server*

DRM *Database Server* berfungsi sebagai penyimpan semua *key* yang dihasilkan oleh DRM. Selain itu dokumen *original* yang sudah terenkripsi dapat disimpan pada *database* ini.

Hal yang perlu diperhatikan dalam memilih *database* yaitu *performance* dan kapasitanya. Ini dikarenakan semua *key* dan dokumen tersimpan ke dalam *database* ini. Beberapa pilihan *database* yang dapat digunakan yaitu SQL *Server*, Oracle, dan MySQL.

Apabila *availability* merupakan hal penting dan utama maka *database* yang digunakan harus didesain sebagai *database cluster* atau *failover*. Dengan pendekatan ini, jika ada mesin *database* yang mati maka mesin *database* lainnya tetap dapat melayani.



Gambar 4. Penerapan *database* sebagai *failover cluster*

Realisasi *database cluster* dapat dilihat pada Gambar 4. Setiap mesin minimal mempunyai 2 *Network Interface Controller* (NIC) dimana satu NIC akan dikonfigurasi sebagai IP *public* atau *cluster* dan satu NIC untuk IP *private* yang berguna untuk mengakses *external storage* (misalkan SAN).

Melalui IP *public* ini digunakan sebagai IP yang dikenali oleh *client* atau *server* yang ingin mengakses *database* tersebut. Mesin *database* mana yang aktif akan ditangani oleh *database cluster* itu sendiri.

3.2.3. *Directory Service*

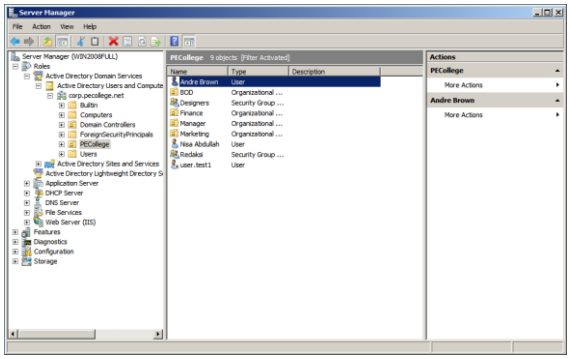
Directory Service berfungsi sebagai identitas dari entitas yang terlibat pada transaksi yang terjadi. Setiap *client* yang mengakses ke DRM *Server* sebelumnya akan dilakukan pengecekan apakah *client* tersebut memang *member* yang mempunyai hak untuk mengakses DRM *Server* atau tidak.

Directory service dapat dikatakan seperti *database* karena ini dapat menyimpanan informasi profil *user* seperti nama, alamat, email dan sebagainya. Selain berisi profil *user*, *Directory service* dapat digunakan untuk *security directory* dimana kita dapat mendefinisikan hak akses setiap *user* atau *group*.

Implementasi *Directory service* biasanya mengikuti standar yang ada yaitu LDAP/X.500. Beberapa perusahan besar juga mengimplementasi *Directory service* berbasis LDAP/X.500 seperti Microsoft, Novell, IBM, SUN, APACHE, RED HAT. Contoh implementasi *Directory service* pada Microsoft adalah *Active Directory* (AD). Ini dapat ditemukan pada sistem operasi Windows *Server* 2000, 2003, dan 2008. Sebagai contoh AD Windows

Server 2008 seperti pada Gambar 5.

Penggunaan *Directory service* untuk DRM berguna sebagai *identity management* artinya setiap kejadian pada DRM harus dikenali oleh *Directory service*. Oleh karena itu, setiap *user* akan mempunyai identitas yang tersimpan di dalam *Directory service*.



Gambar 5. Implementasi *Directory service* pada Windows *Server* 2008

3.2.4. *Certificate Authority*

CA (*Certificate Authority*) *Server* berfungsi *certificate approval* yang memberikan otoritas bahwa *certificate* yang digunakan untuk DRM pada dokumen itu valid atau tidak.

Disarankan CA yang digunakan untuk DRM dikeluarkan oleh suatu lembaga yang dapat dipercaya contohnya Verisign. CA akan diinstall pada DRM *server* terutama yang melakukan publikasi *certificate* ke *client*.

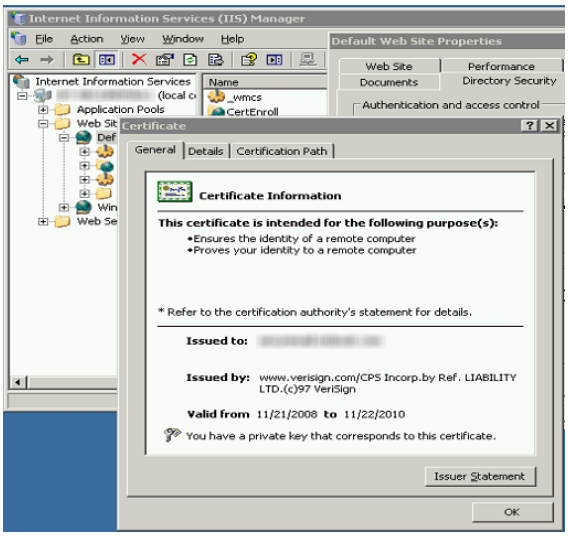
CA yang sudah diperoleh dari lembaga yang dipercaya dapat langsung di-*install* pada *web server* dimana DRM *server* terinstall contohnya seperti Gambar 6.

3.3. DRM *Client*

Pada dasarnya semua aplikasi dokumen yang ada pada komputer *client* dapat memanfaatkan fasilitas DRM dengan catatan aplikasi tersebut mengetahui proses DRM tersebut.

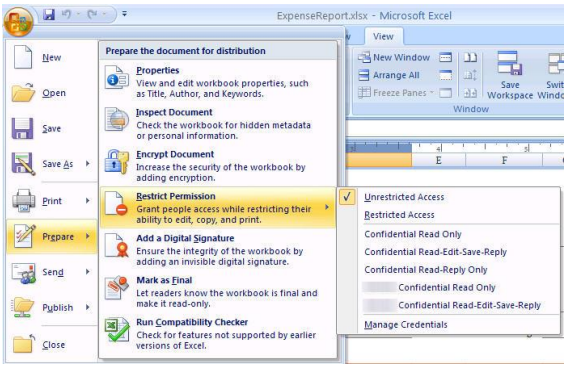
Apabila aplikasi tersebut tidak mengenal sistem DRM ini maka ketika aplikasi membuka dokumen tersebut maka akan terjadi kesalahan pembacaan karena isi dokumen tersebut sudah terenkripsi. Sedangkan aplikasi yang mengenal atau *aware* terhadap DRM maka aplikasi ini dapat berkomunikasi dengan DRM *Server* mulai dari mendapatkan *Certificate*, *Cryptography Key* dan proses enkripsi dan dekripsi.

Aplikasi *client* yang ada dapat juga diintegrasikan dengan DRM dengan pendekatan *Add-on* artinya menambah modul baru pada aplikasi tersebut.

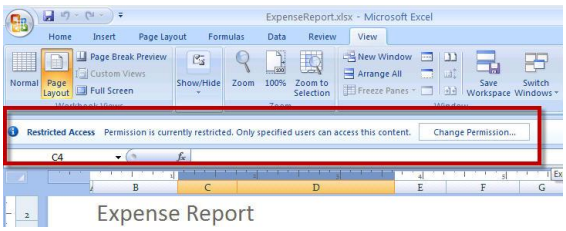


Gambar 6. Contoh sebuah CA yang ter-*install* pada *web server*

Beberapa aplikasi komersial contohnya Micrsooft Office 2007 sudah *support* untuk DRM melalui Microsoft Right Management Services seperti yang terlihat pada Gambar 7.



Gambar 7. DRM pada Microsoft Office Word 2007



Gambar 8. Informasi ketika dokumen DRM tidak dapat diakses oleh *user* yang tidak berhak

Dengan konsep *user interaction*, kita juga dapat meampilkan apabila ada dokumen yang mana *user* tidak mempunyai akan diinformasikan seperti pada Gambar 8.

3.4. Prinsip Kerja

Dokumen elektronik yang akan diterapkan DRM maka *user* pemilik dokumen akan melakukan proses sebagai berikut ini:

1. *User* yang bermaksud untuk memproteksi dokumen harus mendaftarkan komputer dan *domain user account* ke DRM CA. Ini dapat melakukan aplikasi *client* yang berbasis DRM. Mesin komputer yang sudah didaftarkan akan mendapatkan *machine certificate* sedangkan identitas *user* akan memperoleh *Rights Account Certificate*.
2. Selanjutnya *user* akan mengunduh *Certificate* dari DRM *Server* dan mengaktifkannya
3. Setelah user mengaktifkan *Certificate*, *user* membuat *Issuance License* yang berisi hak akses yang akan diterapkan pada dokumen tersebut seperti *Read-Only*, *Don't Forward*, *Editable*, *Print* dan sebagainya. *Issuance License* juga dapat diberlakukan masa aktifnya sesuai dengan kebutuhan.
4. Kemudian *Issuance License* yang telah dibuat dikirim ke DRM *Server* untuk dilakukan proses *signing* sehingga nantinya dapat didistribusikan apabila *user* lain mengakses dokumen ini. *User* pemilik *Issueance License* akan memperoleh *Owner License*.
5. Dengan menggunakan *Owner License*, *user* kemudian melakukan proses *editing* dokumen
6. Apabila selesai proses *editing* maka aplikasi akan melakukan enkripsi isi dokumen dan mengirimnya ke DRM *Server*

Sedangkan *user* lain yang ingin mengakses dokumen berbasis DRM maka proses yang dilakukan adalah

1. Ketika *user* membuka dokumen berbasis DRM maka aplikasi *client* yang *aware* terhadap DRM akan meminta *end-user License* ke DRM *Server* dengan dengan informasi pada dokumen DRM tersebut
2. Sebelum memperoleh *end-user License* maka *user* akan dicek melalui *Directory Service* apakah *user* tersebut *member* dari DRM *Server* atau tidak
3. Apabila proses pengecekan selesai dan ternyata bukan *member* DRM maka *user* tersebut tidak dapat memperoleh *end-user License*
4. Jika *user* merupakan *member* DRM maka *user* akan memperoleh *end-user License* dari DRM *Server*
5. *End-user License* berisi informasi hak akses yang dimiliki terhadap dokumen yang dibuka

Apabila *user* mempunyai hak akses minimal dapat melihat (*view*) maka aplikasi akan melakukan dekripsi melalui *public key* yang sudah disediakan

3.5. Integrasi dengan Sistem E-mail

Apabila dokumen DRM ini diletakan pada e-mail sebagai *attachment file* maka pada dasarnya aplikasi *client* e-mail akan melakukan *attachment* suatu dokumen yang terenkripsi.

Kemudian *user* mengirim e-mail dengan *attachment file* dokumen DRM ke target penerima *mail server*. Setelah sampai pada target penerima *mail server* maka *attachment* tetap dalam bentuk enkripsi DRM yang tersimpan pada *storage mail server* tersebut.

Dari analisa proses yang terjadi diatas maka kita dapat mengatakan dokumen DRM tetap tersimpan aman karena dokumen DRM tersebut tetap pada kondisi terenkripsi.

Ketika *user* yang menerima email yang di dalamnya ada *attachment* dokumen dan e-mail sudah masuk ke dalam aplikasi e-mail *client*. Kemudian user membuka *attachment* dokumen DRM. Apabila aplikasi e-mail *client* tersebut tidak mengenal format dokumen DRM maka aplikasi tetap tidak dapat membuka dokumen DRM.

3.6. Integrasi dengan Aplikasi *Email Client*

Agar aplikasi *email* dapat membuka dokumen DRM maka aplikasi *email* harus dibuat supaya *aware* terhadap dokumen DRM. Guna keperluan ini, kita dapat membuat aplikasi baru yang mengerti sistem DRM kita atau membuat *add-on* pada aplikasi yang ada.

Selain aplikasi *email client* yang *aware* terhadap DRM maka kita dapat membuat interaksi dengan *user* melalui pendekatan GUI yang komprehensif. Misalkan kita ingin *email* yang akan dikirim dapat diberlakukan sebagai

- *Read-only email*. Disini *email* hanya dapat dibaca tetapi tidak dapat disimpan atau di-*print*
- *Do not forward*. Ini *email* yang dikirim ke *user* tertentu maka penerima tidak dapat melakukan *forward* ke *email* lain

Sebagai contoh implementasi DRM pada aplikasi *client* email yaitu Microsoft Right Management Services (RMS) seperti pada Gambar 9.

3.7. DRM pada Sistem Blackberry

Era mobilitas teknologi informasi semakin meningkat ditambah dengan kehadiran sistem Blackberry dari RIM (*Research In Motion*). Sistem ini bekerja dengan melibatkan telekomunikasi *provider* sehingga ini dapat meningkatkan tingkat mobilitas. Dengan *tool* kita dapat mengakses email atau sekedar menjelajah dunia internet.

Gambar 9. Pemilihan keamanan dokumen pada aplikasi *client* email

Kalau kita melihat sistem email yang diberikan pada RIM maka sebenarnya di sisi RIM semua email akan menjadi transparan sehingga data atau dokumen email yang dikirim melalui RIM akan terbuka. Hal ini sangat menjadi kendala ketika mengamankan data atau dokumen pada Blackberry.

Solusi untuk mengamankan data atau dokumen pada *email* di Blackberry adalah dengan membuat aplikasi *middleware* yang dapat berkomunikasi dengan Blackberry Enterprise *Server* (BES) dan DRM *Server*. Contoh solusinya dapat dilihat seperti Gambar 10. Solusi ini memerlukan dua aplikasi *middleware* yaitu:

- DRM-Blackberry *Server*
- DRM Agent pada BES

Proses yang terjadi dengan pendekatan solusi seperti Gambar 9 sebagai berikut:

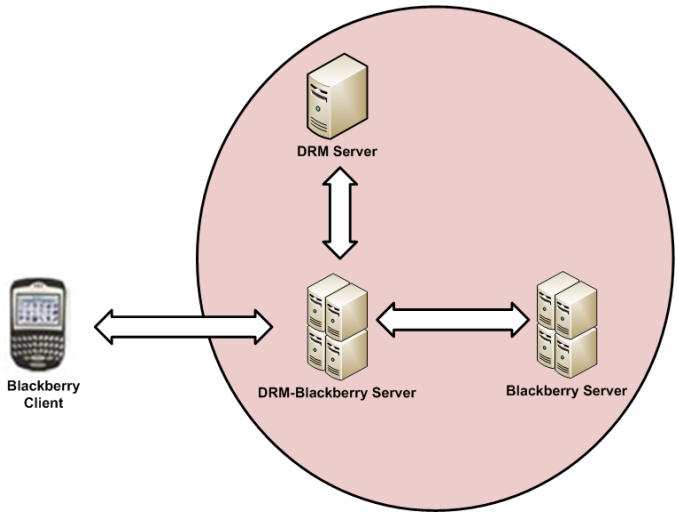
1. Pada Blackberry *client* menerima *email* yang tidak ada DRM maka Blackberry *client* akan langsung berkomunikasi dengan BES.
2. Apabila Blackberry *client* menerima *email* yang didalamnya terdapat dokumen DRM maka ketika email ini dibuka Blackberry *client* akan meminta dokumen DRM ke BES dengan meminta *permission* ke DRM *server* melalui DRM Agent pada BES dan DRM-Blackberry *server*.
3. DRM-Blackberry ini yang akan bertanggung jawab apakah *user* pada Blackberry *client* mempunyai hak akses atau tidak. Selain itu, DRM-Blackberry juga melakukan enkripsi/dekripsi data atau dokumen DRM.
4. Untuk *user* yang akan mengirim *email* dengan DRM maka ketika *email* sampai di BES maka DRM agent pada BES akan melakukan enkripsi melalui DRM-Blackberry *server*. Setelah selesai, *email* tersebut dikirim ke RIM.

Beberapa perusahan IT juga sudah membuat aplikasi pada Blackberry supaya dalam memproses data yang berbasis DRM. Salah satu *vendor* yang fokus terhadap ini adalah GigaTrust [6].

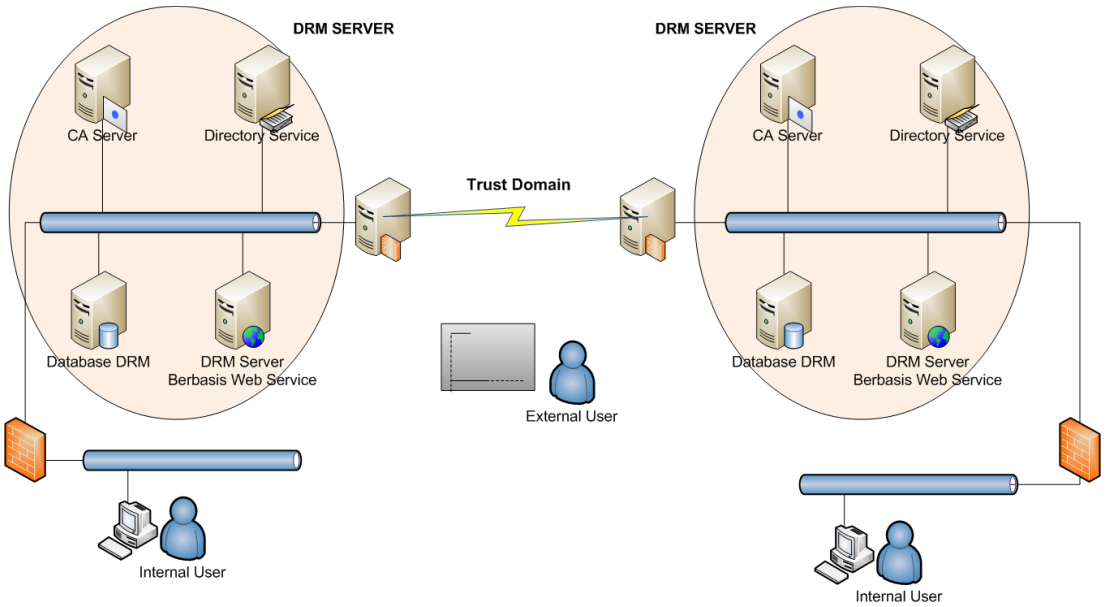
3.8. Klasifikasi Dokumen

Selain mengamankan dokumen elektronik supaya tidak diakses oleh orang yang tidak berhak, kita juga harus memikirkan untuk menentukan dokumen apa yang dapat dikatakan sebagai dokumen yang layak untuk diamankan. Hal ini sangat penting supaya sistem dapat digunakan secara efisien.

Ada banyak metode untuk mengklasifikasi dokumen dapat dikategorikan dokumen penting atau tidak. Pada umumnya klasifikasi dokumen berdasarkan kebutuhan dan jenis lembaga/institusi/perusahan yang akan menerapkan DRM ini.



Gambar 10. Solusi DRM pada Blackberry



Gambar 11. Implementasi DRM untuk B2B

Apabila kita sudah melakukan klasifikasi jenis dan tipe dokumen yang termasuk dokumen yang harus diamankan, kita selanjutnya dapat membakukan sebagai standar baik dalam bentuk SOP (*Standard Operational Procedure*) atau undang-undang. Ini diharapkan dapat melancarkan proses implementasi sistem ini.

Jika klasifikasi sudah ditentukan maka kita juga dapat menerapkan pada aplikasi *client* DRM dengan memberikan label dokumen, contohnya dapat diterapkan pada Microsoft Office Word 2007 seperti Gambar 6. Label ini akan memberikan keamanan sesuai dengan yang ditentukan.

3.9. Kolaborasi Dokumen DRM untuk Kebutuhan B2B

Implikasi untuk kolaborasi dokumen berbasis DRM untuk lingkungan B2B (*Business-to-Business*) dapat diterapkan apabila semua entitas menerapkan *trust domain* terhadap terjadi transaksi yang terjadi baik pada *Directory Service*, DRM *Server* dan CA *Server*.

Semua pihak yang terlibat kolaborasi dokumen DRM harus memberikan hak akses sistem lain untuk melakukan authentikasi dan verifikasi *user* dari setiap sistem. Secara umum desain untuk kolaborasi DRM untuk B2B dapat dilihat pada Gambar 11.

4. Kesimpulan

Penerapan dokumen elektronik yang berbasis DRM diharapkan melindungi informasi didalamnya terhadap orang-orang yang tidak mempunyai kepentingan. Tingkat kesuksesan penerapan dokumen berbasis DRM juga ditentukan kesiapan *user*.

REFERENSI

[1] M. K. Buckland, “What is a "document"?”, *Journal of the American Society of Information Science*, 48 (9), 1997.

[2] Michael Buckland, “What is a digital document"?, *Document Numérique (Paris) 2*, no. 2 (1998): 221-230, 1998

[3] INDICARE, Consumer's guide to Digital Rights Management, INDICARE Project, April 2006.

[4] Microsoft, Windows Rights Management Services, http://www.microsoft.com/rms

[5] Nilay Patel, iTunes Plus DRM-free, not free of annoying glitches, http://www.engadget.com/ 2007/05/31/itunes-plus-drm-free-not-free-of-annoying-glitches/

[6] GigaTrust, www.gigatrust.com,
Roberto Garcia Gonzalez, A Semantic *Web* approach to Digital Rights Managements, PhD. Thesis, Universitat Pompeu Fabra, November 2005.