Pengembangan Aplikasi Pengamanan Dokumen Digital Memanfaatkan Algoritma *Advance Encryption Standard*, RSA *Digital Signature* dan *Invisible Watermarking*

Aji Setiyo Sukarno
Deputi III, Lembaga Sandi Negara
Jakarta, Indonesia
aji.setiyo.sukarno@gmail.com

***Abstrak*—Digitalisasi dokumen sekarang ini merupakan sebuah kebutuhan, hal ini akan mempermudah seseorang untuk mendistribusikan dokumen tersebut melalui berbagai media komunikasi elektronik. Dengan adanya kemudahan tersebut tentunya akan menimbulkan sebuah kerentanan, diantaranya adalah penduplikatan, dan publikasi dokumen digital tanpa izin pemilik dokumen asli hal ini tentunya melanggar HAKI seseorang. Terlebih lagi jika dokumen tersebut berkasifikasi rahasia dan bersifat strategis tentunya akan membawa dampak yang buruk jika terjadi kebocoran atau pengubahan content dari informasi tersebut. Kriptografi merupakan salah satu solusi untuk mengamankan pesan rahasia tetapi kriptografi hanya memberikan keamanan sampai tahap distribusi, maka dibutuhkan sebuah metode tambahan untuk memberikan aspek keamanan pasca distribusi salah satu metodenya adalah dengan memanfaatkan teknik watermarking. Dengan teknik ini akan memberikan proteksi terhadap penggunaan tidak sah dari materi digital, namun untuk menghilangkan kecurigaan dari pihak yang tidak memiliki hak akses terhadap materi digital watermarking yang digunakan adalah invisible watermarking. Pemanfaatan algoritma AES akan melindungi pesan saat proses distribusi, dan algoritma RSA Digital Signature dapat memeberikan jaminan otentikasi pengirim dan penerima dokumen digital yang didistribusikan dan akan memberikan layanan non repudiation.**

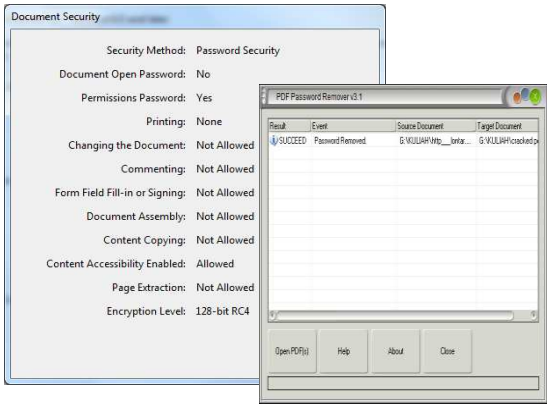
Kata kunci—invisible watermarking, digital signature, AES, RSA

I. PENDAHULUAN

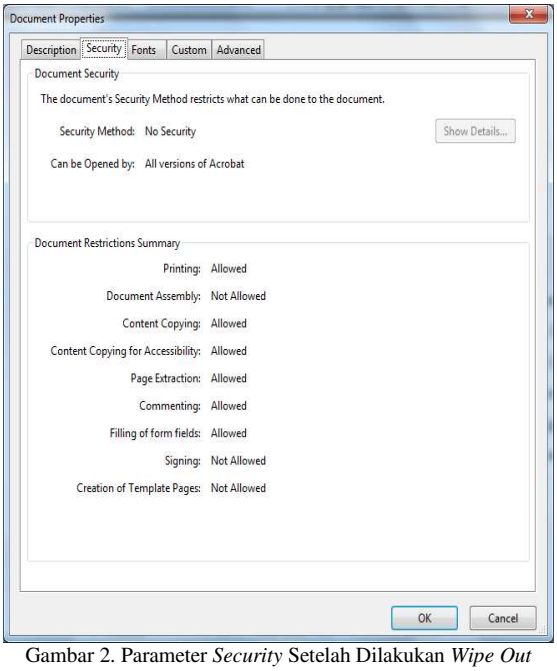
Proses distribusi dokumen digital melalui berbagai media elektronik baik secara *online* maupun *local* akan sangat memepermudah proses pertukaran data, namun dibalik kemudah tersebut terdapat beberapa kerentanan ancaman yang mungkin terjadi terlebih lagi jika dokumen digital yang didistribusikan bersifat kritis dan memiliki klasifikasi rahasia atau sangat rahasia. Untuk menjaga kerahasiaan dokumen digital yang didistribusikan dapat memanfaatkan kriptografi untuk memberikan layanan *confidentiality, data integrity, authentication,* dan *non repudiation.* Layanan kriptografi hanya memberikan layanan keamanan pada saat proses distribusi, sehingga diperlukan layanan lain yang dapat melindungi hak cipta dokumen digital tersebut. Layanan yang dapat digunakan untuk melindungi hak cipta dokumen salah satu diantaranya adalah *Watermarking*. *Watermark* dalam perlindungan hak cipta adalah sebagai bukti otentik atas hak kepemilikan pencipta atas sebuah *content* digital yang dibuat atau diproduksinya. *Watermarking* yang digunakan keberadaanya harus tidak merubah atau merusak konten dari host (dokumen yang disisipi *watermark*). Untuk menambahkan aspek *non repudiation* digunakan *Digital Signature* berbasis kriptografi Asimetrik RSA – 1024 bit .

II. LATAR BELAKANG

Kemudahan dalam pengiriman berita secara elektronik memberikan sebuah celah keamanan, terutama jika dokumen elektronik tersebut merupakan dokumen yang berklasifikasi, dan tidak dikirimkan melalui *secure channel.* Melihat kerentanan tersebut diperlukan sebuah aplikasi yang dapat memebrikan aspek *security* yang melindungi pada saat distribusi maupun pasca distribusi dokumen elektronik (perlindungan *digital right*). Saat ini terdapat aplikasi yang bersifat komersial yang dapat memberikan layanan *copy protection* dan *edit protection* namun setelah penulis melakukan beberapa percobaan menggunakan aplikasi *cracking* "PDF *Password Remover*", dapat melakukan "*wipe out"* seluruh parameter kemanan yang ada pada dokumen berekstensi *.pdf dengan parameter keamanan berikut :



Gambar 1. Parameter *Security* Dokumen *.pdf



Gambar 2. Parameter *Security* Setelah Dilakukan *Wipe Out*

Dari gambar diatas dapat kita lihat bahwa aplikasi PDF *Password Remover* dapat melakukan *wipe out* seluruh parameter kemanan termasuk enkripsi yang digunakan yaitu RC4 128 bit dan *copy protection* dari sebuah dokumen pdf, oleh karena itu dibutuhkan sebuah aplikasi yang dapat memeberikan tingkat layanan keamanan lebih tinggi. Selain itu pengembangan aplikasi ini juga bertujuan untuk memberikan aspek keamanan berupa integritas data jika terjadi pemalsuan tanda tangan digital pada saat pengiriman dokumen.

III. LANDASAN TEORI

A. Kriptografi

Kriptografi merupakan sebuah teknik/seni untuk mengamankan sebuah berita/pesan. Tujuan digunakannya Kriptografi (*cryptography*) adalah suatu seni dan ilmu pengetahuan untuk menjaga keamanan suatu pesan [1]. Kriptografi adalah suatu studi tentang teknik-teknik matematis yang berhubungan dengan aspek-aspek keamanan informasi seperti kerahasiaan (*confidentiality*), keutuhan data (*data integrity*), dan otentikasi (*authentication*) [2].

Kriptografi sendiri bertujuan untuk memberikan layanan keamanan, layanan keamanan tersebut terdiri dari empat aspek sebagai berikut: [2]

a) Kerahasiaan (*confidentiality),* adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
b) Integritas data (*data integrity)*, adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
c) Otentikasi (*authentication*), adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengidentifikasi kebenaran sumber pesan (*data origin authentication*).
d) Nirpenyangkalan (*non-repudiation),* adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman pesan atau penerima pesan menyangkal telah menerima pesan.

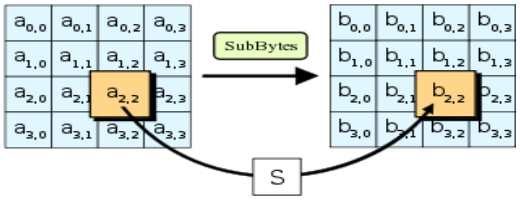
Berdasarkan kunci yang digunakan algoritma kriptografi dibagi menjadi menjadi dua yaitu algoritma simetrik dan algoritma asimetrik. Algoritma simetrik merupakan algoritma kriptografi yang menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi, sedangkan algoritma kriptografi asimetrik menggunakan dua kunci kunci yang berbeda untuk enkripsi dan dekripsi, kunci yang digunakan untuk mengenkripsi pesan disebut kunci publik, sedangkan kunci yang digunakan untuk mendekripsi sandi disebut kunci privat.

B. *Advanced Encryption Standard* (AES)-256

AES (*Advance Encryption Standard*) merupakan suatu standar dalam dalam dunia algoritma. Dalam usaha mencari standar ini tercatat 5 algoritma yang menjadi kandidat AES diantaranya Twofish, RC6, Rijndael, *Serpent,* MARS. Pada akhirnya algoritma Rijndael yang diajukan oleh Vincent Rijment dan Joan Daemen dinobatkan sebagai standar AES.

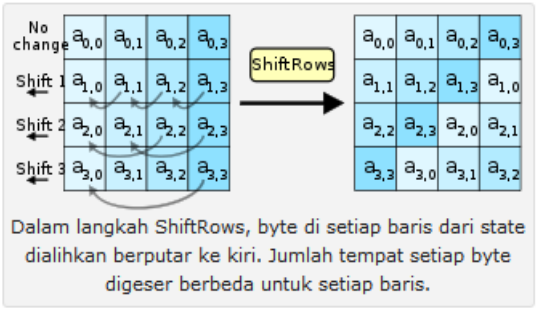
Rijndael adalah algoritma simetri berbasis *block cipher* yang mengenkripsi / mendekripsi blok berukuran 128 bit dengan panjang kunci yang beragam, yaitu 128, 192, dan 256 bit [1]. Sehingga dikenal tiga tipe AES berdasarkan panjang kuncinya, yaitu AES-128, AES-192, dan AES-256. Algoritma AES beroperasi dalam *byte* bukan dalam bentuk bit. Secara garis besar, algoritma AES adalah sebagai berikut :

a) *AddRoundKey,* yaitu melakukan XOR antara *state* awal (plainteks) dengan *cipher key.* Tahap ini juga disebut sebagai *initial round.*
b) Putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   (i) *SubBytes,* yaitu substitusi *byte* dengan menggunakan tabel substitusi (S-*box*). Dalam langkah *SubBytes*, setiap *byte* dalam *array* diperbarui menggunakan 8-bit S-box Rijndael. Operasi ini menyediakan sifat non-linearitas pada teks sandi. S-box yang digunakan berasal dari perkalian invers GF (2^8),yang dikenal mempunyai sifat non-linearitas.



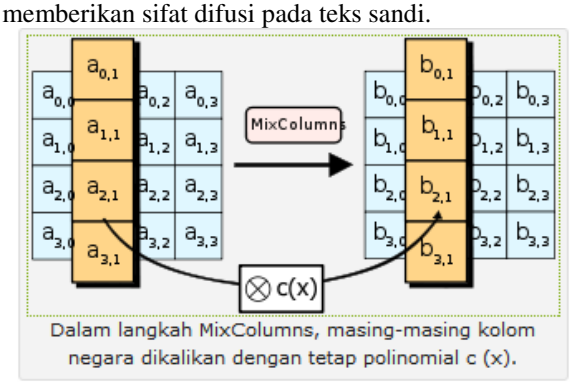
Gambar 3. SubBytes

   (ii) *ShiftRows,* yaitu pergeseran baris-baris *array state* secara *wrapping*. Dalam langkah *ShiftRows*, *byte* di setiap baris dari *state* dialihkan putaran ke kiri. Jumlah tempat setiap *byte* digeser berbeda untuk setiap baris.



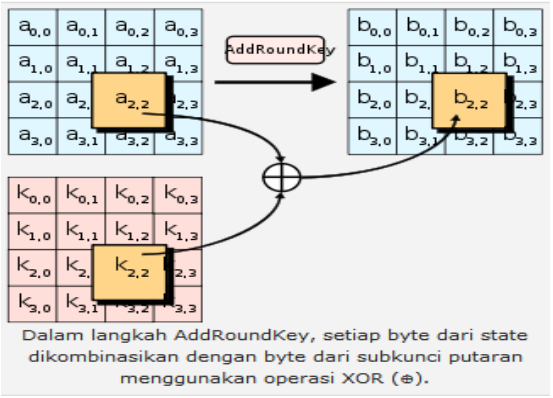
Gambar 4. ShiftRows

(iii) *MixColumns,* yaitu mengacak data di masing-masing kolom *array state.* Dalam langkah *MixColumns*, empat *byte* dari setiap kolom *state* digabungkan dengan menggunakan transformasi *linier invertible*. Fungsi *MixColumns* mengambil empat *byte* sebagai masukan dan keluaran empat *byte*, dimana setiap masukan *byte* mempengaruhi semua keluaran empat *byte*. Bersama dengan ShiftRows, MixColumns memberikan sifat difusi pada teks sandi.



Gambar 5. MixColumns

(iv) *AddRoundKey,* yaitu melakukan XOR antara *array state* dengan *round key.*



Gambar 6. AddRoundKey

c) *Final round*, yaitu proses untuk putaran terakhir yang meliputi:
   1. *SubBytes.*
   2. *ShiftRows.*
   3. *AddRoundKey*

C. RSA *Digital Signature* [2]

Dimisalkan entitas A akan menadatangani sebuah pesan (m) maka hal yang harus dilakukan adalah sebagai berikut:

1) Hitung (~m) = R(m), sebuah integer pada kisaran [0, n-1]
2) Hitung $s = (\sim m)^d \bmod n$.
3) Tanda tangan A untuk pesan m adalah s

Asumsikan dari proses pembuatan kunci tersebut A telah menandatangani pesan m dengan kunci publik B. Dan B mendapat pesan m dari A. B dapat melakukan verifikasi terhadap tanda tangan digital tersebut.

Cara untuk memverifikasi tanda tangan digital oleh B adalah dengan cara sebagai berikut:

1) Ambil kunci publik yaitu n dan e
2) Hitung $\sim m = s^e \bmod n$
3) Dengan mengetahui ~m, B dapat mendapatkan m.

Tanda tangan digital berfungsi pada kasus sebagai berikut: jika s adalah tanda tangan untuk pesan m, maka $s \equiv (\sim m)^d \bmod n$ dimana (~m) = R(m). Karena $ed \equiv 1 \pmod{\Phi(n)}$, $s^e \equiv \sim m^{ed} \equiv \sim m \pmod n$. Terakhir, $R^{-1}(\sim m) = R^{-1}(R(m)) = m$. Nilai $d \neq e$, nilai e dikirimkan oleh pihak pengirim (A) ke B, nilai e tidak bersifat rahasia sehingga dapat dikirimkan dalam saluran komunikasi publik.

Proses penyisipan dan pengecekan tanda tangan digital dilakukan dengan proses sebagai berikut:

1) Ketika A akan menandatangani pesan m, A menghitung $s \equiv m^d \pmod n$.
2) B memverifikasi bahwa A telah menandatangani pesan dengan menghitung $s^e \pmod n$ dan membandingkannya dengan m.

Dengan memilih nilai exponen publik yang kecil, kecepatan verifikasi tanda tangan bisa menjadi lebih cepat daripada kecepatan menandatanganinya.

D. *Watermarking*

Watermarking biasanya digunakan untuk memberikan layanan *copy protection* pada sebuah dokumen digital. *Watermark* sengaja ditanamkan secara permanen pada data digital sedemikian hingga pengguna yang berwenang dapat dengan mudah membacanya, disisi lain *watermark* tersebut haruslah tidak mengubah isi media kecuali sedikit atau perubahan tersebut tidaklah tampak atau kurang begitu tampak bagi indera manusia[3]. *Watermark* yang ditambahkan dalam sebuah media digital dapat berupa watermark yang tampak secara kasat mata (*visible watermarking*) dan watermark yang tidak tampak secara kasat mata *(invisible watermarking*). Teknik yang termasuk dalam ranah frekuensi memperlakukan media penyimpan sebagai suatu frekuensi yang dapat diolah secara analog maupun digital. Teknik ini umumnya menghasilkan suatu watermark yang tahan terhadap modifikasi media penyimpannya atau yang dikenal sebagai sifat *robust*. Dengan sifat ketahanan terhadap modifikasi tersebut, maka teknik ini seringkali digunakan untuk dijadikan sebagai *copyright protection* pada media yang disisipi.

Invisible watermarking akan membawa keuntungan jika diaplikasikan untuk *copyright protection* dokumen digital, dengan digunakannya *invisible watermarking* seorang yang berniat untuk menyalah gunakan dokumen digital tersebut tidak akan sadar bahwa dalam dokumen digital tersebut terdapat informasi ekstra yang tersembunyi [4]. Dengan memanfaatkan *invisible watermarking* juga akan memperkecil peluang seorang penyerang untuk melakukan penghapusan watermarking tanpa dilakukan analisis terhadap dokumen digital terlebih dahulu. Namun jika yang digunakan adalah *visible watermarking* maka seseorang dapat dengan mudah dan menyadari bahwa dalam suatu dokumen terdapat *watermark*, tentunya akan lebih mempermudah untuk mengganti atau memodifikasi konten dari *watermark* tersebut.

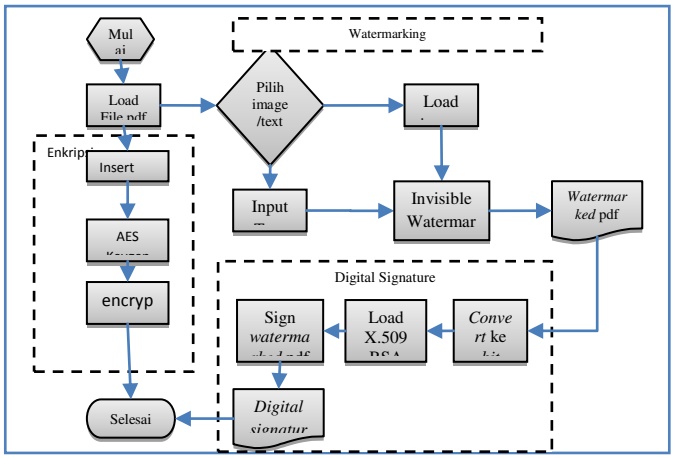
IV. PERANCANGAN APLIKASI

Aplikasi pengamanan dokumen yang dikembangkan merupakan aplikasi berbasis java, untuk saat ini aplikasi pengamanan dokumen dengan memanfaatkan *invisible watermarking* hanya dapat digunakan pada file berekstensi *.pdf. Skema yang diusulkan dibagi menjadi 2 Skema yaitu Skema proteksi dan skema verifikasi.

Berikut adalah penjelasan dari Skema proteksi dan skema verifikasi pada aplikasi pengamanan dokumen digital dengan memanfaatkan *invisible watermarking* dan RSA *Digital Signature*:

A. Skema Proteksi

Pada pemberian skema pemberian *watermark* dan otentikasi dilakukan dengaN cara yang sederhana seperti terlihat pada Gambar 7. Secara umum pemberian *watermarking* dapat dilakukan dengan memanfaatkan teks atau sebuah citra sebagai *watermark*-nya. File citra yang akan dijadikan watermark dapat diambil dari sembarang citra dengan ukuran sembarang. Pada saat implementasinya, citra tersebut akan diletakkan pada bagian tengah setiap halaman pdf.



Gambar 7. Skema Proteksi Dokumen

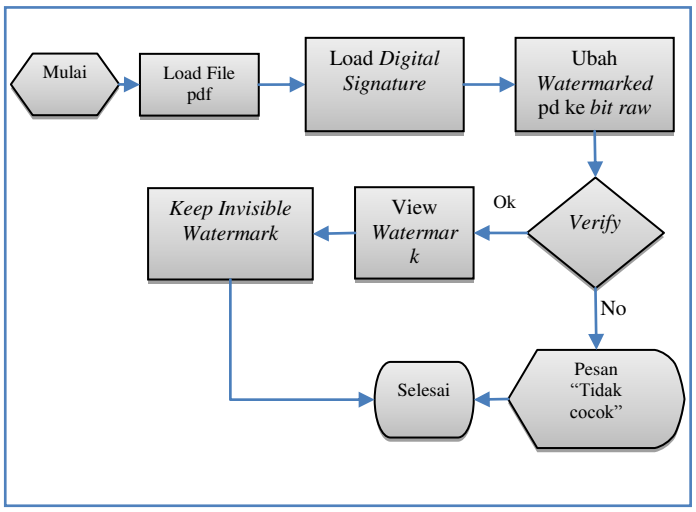
Pemberian *invisible watermark* dimaksudkan untuk memberikan tanda kepemilikan dan keotentikan dokumen digital tersebut.

Tahap pembuatan *digital signature* dilakukan setelah pemberian *watermark*. Hal ini dimaksudkan agar pada proses autentikasi dan non repudiasi juga dapat menjaga integritas data digital tersebut. Karena algoritma dalam *digital signature* memerlukan masukan berupa bit raw, maka data dokumen file pdf diubah terlebih dahulu ke dalam format bit raw. Selanjutnya bit raw tersebut diberikan digital signature dengan menggunakan kunci privat pengirim yang memiliki panjang kunci 1024 bit, dan menggunakan format Publik *Key* Infrastruktur (PKI) X.509. Nilai yang dihasilkan dari algoritma tersebut selanjutnya disimpan dalam suatu file yang terpisah dari file pdf yang dihitung. Setelah dilakukan proses signing pada dokumen pdf yang telah diberi *watermark* dienkripsi menggunakan algoritma *Advance Encryption Standard* dengan panjang kunci 256-bit. Kunci AES 256 tersebut didapat dari *password* yang diinputkan oleh *user* pada aplikasi yang selanjutnya *password* tersebut diproses menggunakan *key generator* sehingga didapatkan kunci enkripsi dengan panjang 256-bit.

Secara umum dapat disimpulkan bahwa *input* dari skema diatas adalah dokumen file pdf, kunci privat, citra atau teks *watermark*, dan *password*. Sedangkan *output* dari skema tersebut adalah dokumen berwatermark terenkripsi dan file teks berisi *digital signature*. Selanjutnya kedua *output* dokumen tersebut dapat dikirimkan terpisah atau bersama-sama untuk kebutuhan autentikasi, untuk memverifikasi tandatangan digital pihak pengirim harus mengirimkan kunci publiknya pada pihak yang akan melakukan verivikasi.

B. Skema Verifikasi

Untuk melakukan verifikasi dilakukan proses terbalik dari proses signing. File pdf yang akan diverifikasi diubah terlebih dahulu ke dalam bentuk barisan biner yang selanjutnya dilakukan proses dekripsi pesan menggunakan kunci yang telah disepakati kedua belah pihak sebelumnya. Setelah proses tersebut selanjutnya dilakukan perhitungan digital signature dengan memanfaatkan kunci publik yang telah diberikan oleh pengirim dokumen digital. Proses verifikasi adalah proses perhitungan nilai *digital signature* terhadap barisan biner dengan menggunakan kunci publik. Jika tidak cocok maka proses selanjutnya adalah pemberitahuan informasi bahwa telah terjadi perubahan pada file pdf yang akan diuji. Untuk pengecekan keaslian dokumen dilakukan ekstraksi *invisible watermarking*, hasil ekstraksi ini berupa citra digital yang diekstrak dari dokumen asli citra ini dapat digunakan sebagai bukti identitas kepemilikan dokumen asli , jika suatu saat ada pihak – pihak yang menyebarkan dan mengakui kepemilikan dokumen asli tersebut.



Gambar 7. Skema Verifikasi Dokumen

a) Penggunaan Sarana Implementasi

TABEL I. PERANGKAT YANG DIGUNAKAN

Jenis Perangkat	Spesifikasi
Prosesor	Intel® Core™ i5 2.1 GHz
Memori	6 GB
Hard Disk	500GB
VGA Card	2 GB
Sistem Operasi	Windows 7
Software App	Java Runtime Enviroment 1.6 Library Bouncycastle, iText

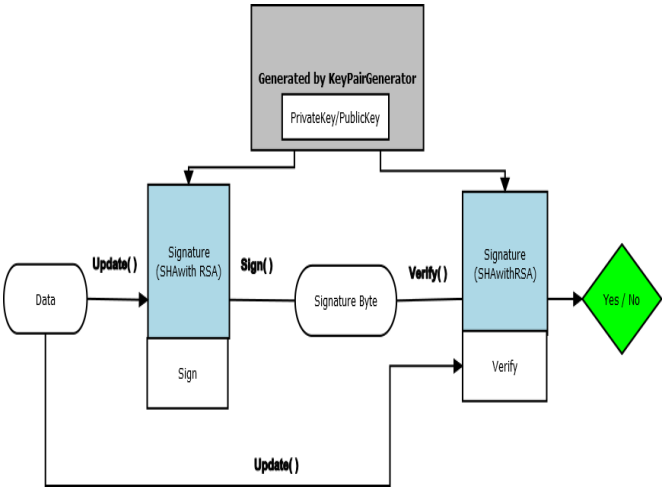
Pada percobaan implementasi digunakan komputer laptop dengan spesifikasi seperti terlihat pada Tabel I. Pada implementasi dan percobaan menjalankan aplikasi berjalan dengan baik dan tidak menemui kendala terhadap running program. Kebutuhan minimal aplikasi tersebut dapat berjalan pada Windows atau sistem operasi lain yang mendukung java. Untuk pembangunan aplikasi ini penulis menggunakan IDE Netbeans 6.9.

b) Implementasi *Signing*

Implementasi dari skema pada Gambar 7 dapat dilihat pada Gambar 8. Masukan aplikasi dibuat dengan mudah untuk menentukan file pdf yang akan dilakukan proses *signing*. Pemilihan watermark utama yang digunakan dapat dilakukan dengan pilihan *combo box text watermark* ataupun *image watermark*.

Gambar 8. Implementasi *Signing*

Untuk melakukan *input privat key* dengan format PKCS#8 (*Private-Key Information Syntax Standard*) yang digunakan untuk penandatanganan digital dilakukan dengan mengklik tombol *load*. PKCS adalah suatu standar untuk kriptografi kunci publik yang dibuat dan diterbitkan oleh RSA *Security*. Proses selanjutnya adalah mencari letak lokasi file privat *key* yang akan digunakan. Privat *key* disimpan dalam suatu file berekstensi *.key yang disimpan dengan menggunakan encoding Base64.

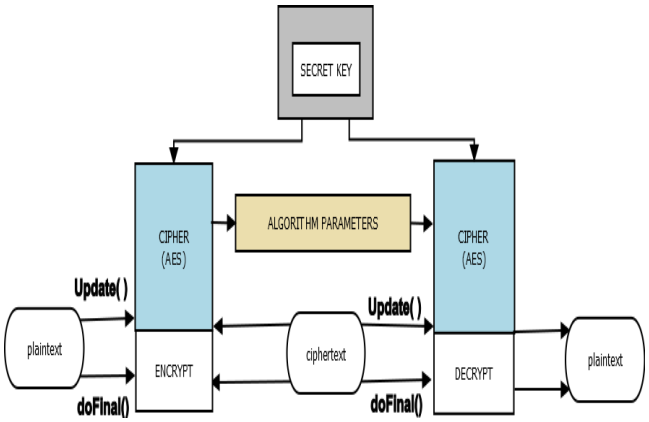


Gambar 9. Proses *Signing* RSA Memanfaatkan JCE[1]

Setelah proses *watermarking* dan proses penandatanganan digital dilakukan proses selanjutnya adalah proses enkripsi dokumen proses enkripsi memanfaatkan algoritma enkripsi AES 256- bit. Dengan memanfaatkan fungsi *key generator* AES pada Java, *password* yang diinputkan oleh *user* dijadikan sebagai *seed* untuk membangkitkan kunci enkripsi. Gambaran mengenai proses

[1,2]http://lampwww.epfl.ch/java/jdk1.7/docs/technotes/guides/security/crypto/CryptoSpec.html

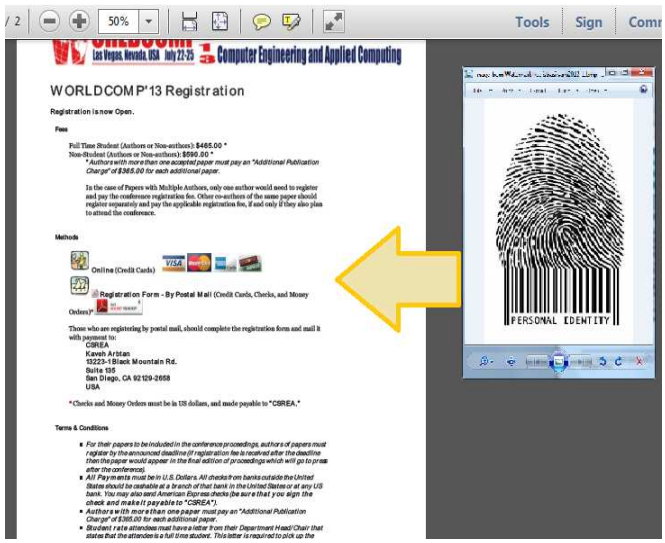
enkripsi AES menggunakan JCE pada Java dapat dilihat pada Gambar 10.



Gambar 10. Proses Enkripsi AES Memanfaatkan JCE[2]

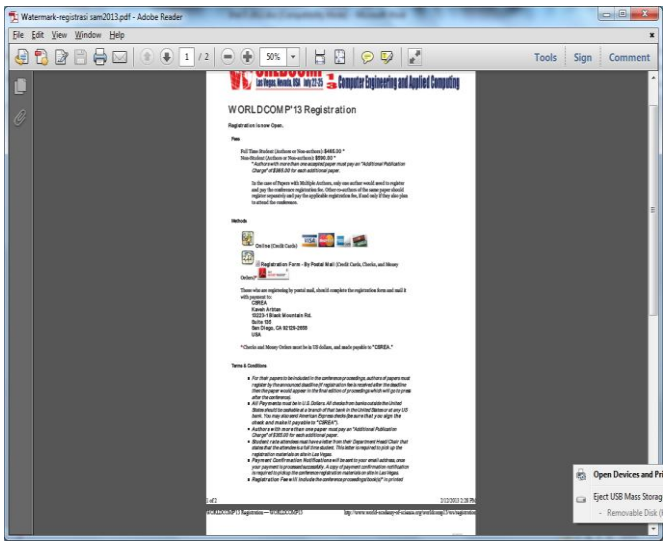
Output dari proses skema ini selanjutnya disimpan pada lokasi terpisah atau dengan nama lain dari file aslinya. File hasil proses proteksi disimpan dalam bentuk file berekstensi *.rhs. File berkestensi rhs ini dapat dikirimkan ke pihak yang dituju disertai dengan *file digital signature* serta kunci publik.

Pada pengembangan aplikasi ini belum dilakukan proses management kunci sehingga pengelolaan kunci Asimetrik yaitu RSA, dan kunci simetrik AES seutuhnya dikelola oleh entitas yang melakukan pertukaran data. Kunci publik RSA bersifat publik dapat di pertukarkkan melalui saluran komunikasi publik namun kunci enkripsi AES 256 dan kunci privat RSA yang digunakan untuk menandatangani pesan bersifat rahasia. Untuk kunci AES 256 hanya diketahui oleh kedua belah pihak yang berkomunikasi, dan kunci privat RSA hanya boleh diketahui oleh pihak yang melakukan penandatanganan terhadap dokumen digital.



Gambar 11. Citra Disisipkan pada Dokumen pdf

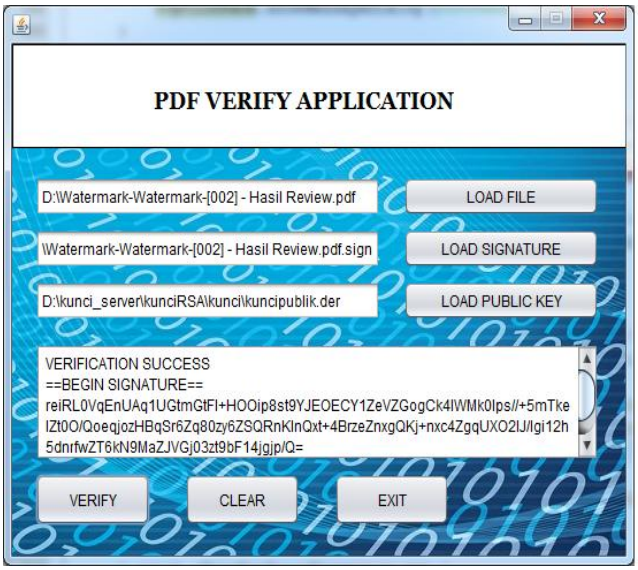
[2]http://lampwww.epfl.ch/java/jdk1.7/docs/technotes/guides/security/crypto/CryptoSpec.html



Gambar 12. Dokumen pdf yang Telah Disisipi *Invisible Watermarking*

c) Implementasi Verifikasi

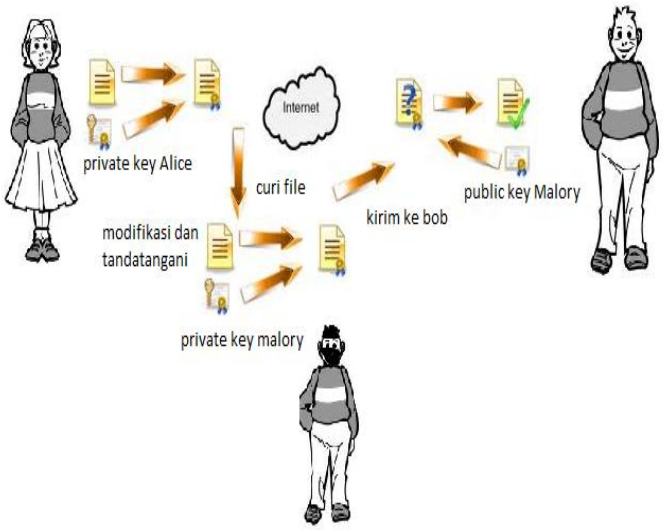
Implementasi skema verifikasi dapat dilihat pada Gambar 13. Aplikasi verifikasi dibuat terpisah dengan aplikasi signing. Hal ini dikarenakan pengguna aplikasi *signing* dan aplikasi verifikasi umumnya berbeda. Pemisahan ini juga memiliki keuntungan lain yaitu besarnya file aplikasi yang sangat kecil. Dengan demikian, aplikasi tersebut dapat dikirim via *email* dalam bentuk aplikasi yang dapat dipanggil langsung dengan *platform* Java.



Gambar 13. Implementasi Verifikasi

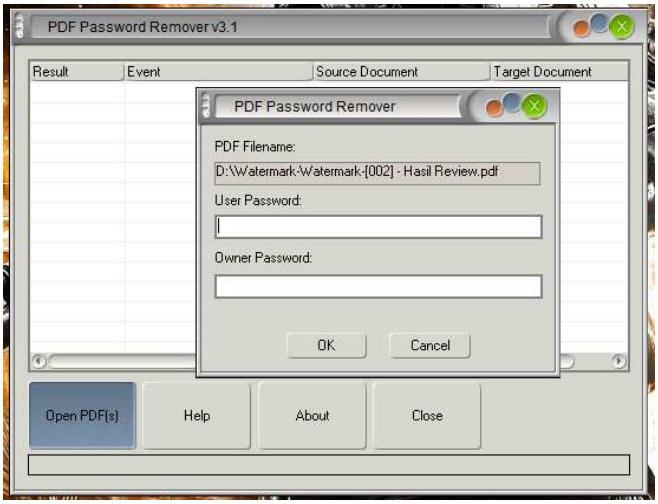
Untuk menjalankan aplikasi verifikasi dibutuhkan 3 (tiga) masukan, yaitu file dokumen pdf yang akan diverifikasi, *digital signature* dan kunci publik. Ketiga masukan tersebut harus ada untuk dapat menjalanan aplikasi tersebut.Pada proses ini *user* diminta untuk me-*load* file terproteksi, *digital signature*, publik key RSA yang diberikan oleh pengirim, dan menekan tombol verify dengan sendirinya aplikasi akan

melakukan verifikasi *digital signature* dan ekstraksi *watermarking*. Keluaran dari aplikasi tersebut adalah sebuah pesan dalam *message box* yang memberikan informasi bahwa dokumen yang diverifikasi sama dengan aslinya atau tidak, sebuah citra hasil ekstraksi dari dokumen digital yang asli, dan nilai *signature*. Jika dalam sebuah kasus seorang penyerang memiliki password untuk membuka enkripsi pada sebuah skema *man in the middle attack* dan penyerang tersebut memodifikasi konten dari dokumen asli serta mengubah *digital signature*, lalu mengirimkan kunci publik palsu dan dokumen tersebut ke penerima aslinya tanpa mengetahui terdapat *invisible watermark* dalam dokumen tersebut, penerima file masih dapat memverifikasi dokumen tersebut dengan cara melakukan ekstraksi *invisible watermarking*. Indikatornya adalah jika sebuah file tersebut telah dimodifikasi maka proses ekstraksi citra *invisible watermark* tidak dapat dilakukan, walaupun verifikasi *digital signature* dapat diverifikasi karena penyerang telah mengirimkan kunci publik palsu pada penerima. Berikut adalah ilustrasi pada penyerangan *man in the middle attack* :



Gambar 14. Skema *Man in The Middle Attack*

Setelah dilakukan proses implementasi *invisible watermarking, digital signature* dan enkripsi dokumen menggunakan AES dengan panjang kunci 256 bit, dilakukan percobaan *cracking password wipe out* terhadap dokumen terproteksi dan didapat hasil seperti pada Gambar 15.



Gambar 15. Percobaan *Password Wipe Out* pada Dokumen Terproteksi

Password remover tidak dapat melakukan *wipe out password* dan dibutuhkan *password* untuk membuka *file* terproteksi tersebut. Hal ini membuktikan bahwa dengan metode yang diajukan bersifat *robust* terhadap serangan password wipe out.

V. KESIMPULAN

Kesimpulan yang dapat diambil dari hasil penelitian ini adalah:

1. Aplikasi pengamanan dokumen digital memanfaatkan aspek kriptografi *confidentiality, authentication, data integrity,* dan *non repudiation.*
2. Aplikasi pengamanan dokumen digital sementara hanya bisa digunakan untuk file berekstensi *.pdf
3. Proses manajemen kunci simetrik dan asimetrik masih bersifat manual sehingga perlu dilakukan pengembangan lebih lanjut pada aplikasi ini.
4. Aplikasi pengamanan ini memanfaatkan fungsi kriptografi berupa enkripsi menggunakan AES dengan panjang kunci 256 bit dan RSA 1024 bit serta invisible watermark yang berfungsi sebagai digital stamp dan *copyright protection.*
5. *Invisible watermark* dapat digunakan untuk verifikasi pada kasus *man in the middle attack,* dengan pemalsuan tandatangan digital.
6. Metode yang diajukan bersifat *robust* terhadap serangan *password wipe out.*

DAFTAR PUSTAKA

[1] B. Schneier, Applied Cryptography, Canada: John Willey and Sons, Inc, 1996.

[2] Menezes, Handbook of Applied Cryptography, Florida: CRC Press LLC, 1997.

[3] Barni et al, A DCT domain system for robust image water marking, Signal Proccessing, 1998.

[4] K. Tsunoda, Digital Watermarking, 2006.

[5] Danang Jaya,dkk, Digital Right Management Citra Berbayar dengan Kriptografi pada Viewer Berbasis Mobile Device, Makassar: IDSECCONF, 2012.

[6] D. Jaya, Pemanfaatan Watermarking untuk Pengamanan Multimedia Digital, Jakarta: Seminar fungsional Sandiman, Lembaga Sandi Negara, 2011.

[7] D. J. &. Wildan, WildStego: Aplikasi Steganografi berbasis MMS, Jakarta: LKTI Lembaga Sandi Negara, 2010.

[8] Lembaga Sandi Negara, Jelajah Kriptologi, Jakarta: Lembaga Sandi Negara, 2007.

[9] W. Stallings, Cryptography and Network Security: Principles and Practices, Prentice Hall, 2003.

[10] A. Setiyo, Secure USB Flash Drive Dongle Menerapkan Konsep Two Factor Authentication, Password Base Encryption dan Digital Signature Sebagai Solusi Anti Piracy Software, Denpasar: KNS&I 2012, STIMIK STIKOM BALI, 2012.