Pengamanan Sertifikat Tanah Digital Menggunakan Digital Signature SHA-512 dan RSA

Leonardo Refialy [#1], Eko Sediyono [*2], Adi Setiawan [#3]
Magister Sistem Informasi dan Fakultas Sains dan Matematika, Universitas Kristen Satya WacanaUniversitas Kristen Satya Wacana dan Jl. Diponegoro 52-60, Salatiga 50711-Indonesia
[1]leo.refialy@gmail.com
[3]adi_setia_03@yahoo.com
[*] *Magister Sistem Informasi, Universitas Kristen Satya WacanaUniversitas Kristen Satya Wacana dan Jl. Diponegoro 52-60, Salatiga 50711-Indonesia*
[2]eko@staff.uksw.edu

Abstract — **Land trading as investment is sometimes find some barriers or problems. One of the problems is illegal or misclaimed of the land certificate. Badan Pertanahan Nasional (BPN) and Pejabat Pembuat Akta Tanah (PPAT) as institutions and correlated parties in the process of land certificate making need a system to handle the problem of falsification land certificate. The purpose of this research is to make secure system of digital land certificate document with a data protection system to detect if there is a falsification activity from certain party. This research hopefully can help BPN and PPAT as authorized institution of land certificate making and to lower the number of falsification. Digital signature SHA 512 and RSA are used in this research as a solution to keep data integrity of land certificate in a digital format, in this case, in an electronic book with.pdf format. Digital signature derived from xref table in.pdf file. This program has been consulted with BPN and PPAT officer in Salatiga. The result is, this program can be used and applied for authentication process of digital land certificate and help BPN to solve the problem of high number of falsification digital land certificate.**

Keywords—**Sertifikat Tanah Digital, Digital Signature, Hash SHA 512, RSA, Integritas Data, Xref-Table**

I. PENDAHULUAN

Sengketa tanah sangat marak terjadi di Indonesia. Berdasarkan data Badan Pertanahan Nasional (BPN), pada tahun 2012 saja telah terjadi 7.196 kasus pertanahan di seluruh Indonesia. Jumlah tersebut meningkat tajam jika dibandingkan dengan jumlah kasus yang terjadi pada tahun 2006, yaitu sebanyak 2.810 kasus. Beberapa tahun terakhir, persentase akumulasi perkara bidang pertanahan yang diajukan ke Mahkamah Agung diperkirakan berkisar antara 65% hingga 70% dari keseluruhan perkara yang ditangani setiap tahunnya [1].

Salah satu contoh kasus belakangan ini, Badan Pertanahan Nasional (BPN) telah melaporkan 60 sertifikat palsu ke Polres Jakarta Timur. Hingga saat ini, memang belum diketahui secara pasti siapa pelakunya. Namun, dapat dipastikan bahwa pelaku yang membuat sertifikat palsu itu telah melanggar Pasal 263 KUHP tentang pemalsuan dokumen dengan ancaman hukuman enam tahun penjara [1].

Sertifikat tanah diterbitkan dalam bentuk cetak kertas. Sertifikat dilengkapi dengan hologram berlogo BPN, yang ditambahkan untuk menghindari pemalsuan sertifikat. Pada sertifikat dalam bentuk digital, sangat mudah dilakukan proses penggandaan. Proses manipulasi pada data digital juga mudah dilakukan. Sehingga perlu mekanisme untuk mengetahui bahwa suatu sertifikat digital, tidak mengalami perubahan dari aslinya [2].

Untuk mengamankan suatu dokumen dari modifikasi yang tidak sah, digunakan suatu metode yang disebut dengan *digital signature*. *Digital signature* bekerja dengan cara meringkas isi dari dokumen yang diamankan, kemudian disandikan dengan suatu algoritma kriptografi, dan hasilnya disisipkan ke dalam dokumen tersebut. Sehingga dokumen digital dan tanda tangan digital tersebut akan selalu ada bersama-sama dalam satu *file*.

Terdapat tiga proses utama dalam *digital signature*, yaitu proses mendapatkan ringkasan isi dokumen, proses menyandikan ringkasan, dan terakhir adalah proses menyisipkan ringkasan terenkripsi. Proses meringkas suatu isi dokumen dapat dilakukan dengan menggunakan fungsi *hash*. *Output* dari fungsi *hash* disebut nilai *hash*.

Fungsi *hash* adalah salah satu alat penting dalam bidang kriptografi dan digunakan untuk mencapai sejumlah tujuan keamanan seperti keaslian, tanda tangan digital, *digital steganography*, dan lainnya [3].

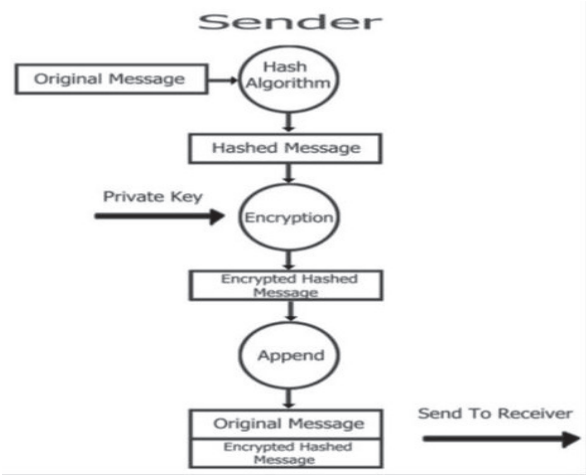
SHA-2 merupakan salah satu fungsi *hash* yang terbukti aman [4]. Proses kedua adalah menyandikan nilai *hash*, dengan algoritma kriptografi. Dengan menggunakan algoritma kunci publik (asimetrik), maka hanya pihak

pemilik kunci privat, yaitu pemilik dokumen, yang dapat melakukan enkripsi nilai *hash*, dan memberikan *digital signature*. Penerima dokumen, yang diasumsikan sebagai publik, dapat menggunakan kunci publik untuk melakukan verifikasi. Proses ketiga adalah proses penyisipan ke dalam dokumen tersebut.

Pada penelitian ini dikembangkan mekanisme *digital signature* untuk *file* PDF. Nilai *hash* tidak dihitung dari keseluruhan dokumen, namun hanya dari bagian *xref table (Cross Refference Table)* pada *file* PDF, karena *xref table* sendiri sudah mencerminkan keseluruhan isi *file* PDF. Fungsi *hash* yang digunakan adalah SHA-2, lebih tepatnya SHA 512. Algoritma kriptografi yang digunakan adalah RSA.

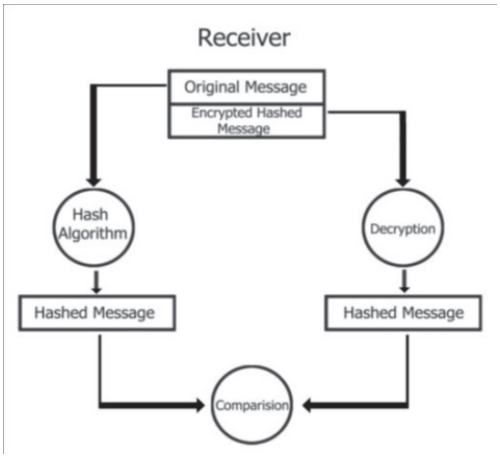
II. PENELITIAN TERKAIT

Noroozi, Daud, dan Sabouhi [5] dalam penelitiannya memberikan tinjauan literatur dan analisis sistem keamanan dan penekanannya adalah pada tanda tangan digital, algoritma *hash* pesan. Algoritma yang diusulkan memperkenalkan teknik baru untuk menghasilkan *output* berukuran kecil dari tanda tangan digital, hal ini menjadi skema baru yang berpotensi praktis dalam penandatanganan dan verifikasi tanda tangan yang cepat.



Gambar 1.Proses *Signing* yang dilakukan oleh S*ender*

Pada *original message* diproses dengan menggunakan algoritma *hash*, sehingga dihasilkan *hashed message*. *Hashed message* dienkripsi dengan menggunakan algoritma asimetrik. *Encrypted hashed message* ditempelkan pada *original message*, kemudian dikirimkan ke *receiver* [5].



Gambar 2. Proses *Verification* dilakukan oleh *Receiver*

Pada sisi *receiver* terdapat dua proses. Proses pertama yaitu mendekripsi *encrypted hashed message* yang menempel pada *original message*. Proses kedua adalah menghitung nilai *hash original message* yang diterima. Dari kedua proses diperoleh dua *hashed message*, yang akan sama nilainya jika dokumen tidak mengalami perubahan/manipulasi [5].

Dalam kriptografi, SHA-1 adalah fungsi *hash* kriptografi yang dirancang oleh *National Security Agency* Amerika Serikat dan diterbitkan oleh Amerika Serikat NIST (AS) *Federal Information Processing Standard*, sementara generasi berikutnya yaitu SHA-2 adalah satu set fungsi *hash* kriptografi (SHA-224, SHA-256, SHA-384, SHA-512) yang dirancang oleh *National Security Agency* (NSA) dan diterbitkan pada tahun 2001 oleh NIST sebagai US F*ederal Information Processing Standard* [6]. Dalam *sign* dan *verify* SHA2 yang mempunyai waktu paling cepat dan baik dalam melakukan proses otentikasi [7]. Dalam penelitian ini digunakan SHA-2 sebagai algoritma tanda tangan digital.

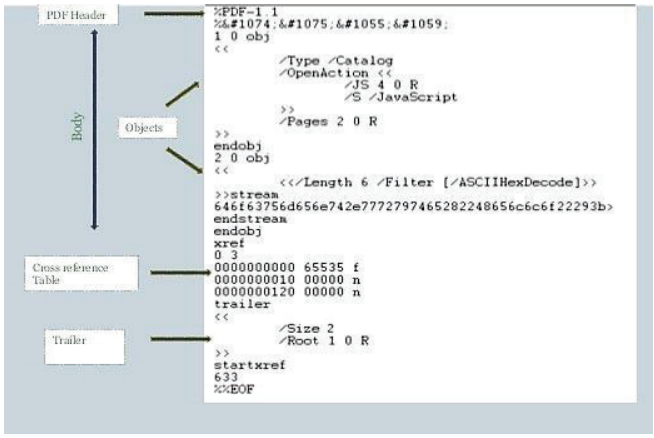
Dalam bidang kriptografi, RSA adalah sebuah algoritma pada enkripsi *public key*. RSA merupakan algoritma pertama yang cocok untuk *digital signature* seperti halnya enkripsi, dan RSA merupakan salah satu algoritma yang paling maju dalam bidang kriptografi *public key*. Algortima RSA dijabarkan pada tahun 1977 oleh tiga orang: Ron Rivest, Adi Shamir dan Len Adleman dari *Massachusetts Institute of Technology* [8]. Huruf RSA itu sendiri berasal dari inisial nama mereka (Rivest—Shamir—Adleman). RSA dipercaya dalam mengamankan dengan menggunakan kunci yang cukup panjang [9].

Dokumen atau *file* PDF telah menjadi standar dunia sebagai dokumen elektronik (digital) [10]. *file* berformat PDF terdiri dari 4 bagian yaitu: *header*, *body*, the *xref cross-reference table*. *Header* adalah baris pertama dari file PDF menentukan nomor versi yang digunakan dokumen PDF. Sementara untuk *body* pada *file* PDF berisi objek termasuk didalamnya aliran teks, gambar, dan objek multimedia lainnya dan digunakan untuk menyimpan semua data pada dokumen yang ditampilkan kepada pengguna.

Xref table adalah tabel referensi, yang berisi referensi ke semua objek yang ada di dalam dokumen. Tujuan dari *xref table* adalah untuk mempermudah akses secara acak ke suatu objek dalam *file*, sehingga tidak perlu dibaca keseluruhan dokumen PDF untuk menemukan objek tertentu. *Xref table* mencerminkan keseluruhan isi *file* PDF. *Trailer*: PDF *Trailer* menentukan bagaimana aplikasi membaca dokumen PDF harus menemukan tabel referensi silang dan objek-objek yang berada pada *file* pdf [11].

Gambar 3. Struktur *File* PDF

Pada bagian *Header* dalam file PDF menentukan nomor versi yang digunakan PDF. *Body* berisi objek-objek yang biasanya adalah teks, gambar, dan unsur multimedia lainnya. *Body* digunakan untuk menyimpan semua data dokumen yang ditampilkan kepada pengguna. *XRef Table* berisi referensi ke semua objek dalam dokumen. *Trailer* pada *file* PDF menentukan bagaimana aplikasi membaca dokumen PDF.



Gambar 4. Contoh Struktur *File* PDF

Dalam penelitian ini, bagian *file* pdf yang diekstrak menjadi fungsi *hash* adalah *XREF Table. XREF Table* dimulai dengan kata kunci *"xref"*. Tabel *Xref* adalah peta pengidentifikasi objek untuk menemukan setiap objek dalam *file* dokumen pdf. Bagian *file* pdf yang akan diekstrak untuk menjadi fungsi *hash* adalah pada bagian *XRef Tab* (*Cross Reference Table*). Tujuan dari tabel referensi silang adalah untuk memungkinkan akses acak ke objek dalam *file*, sehingga kita tidak perlu membaca seluruh dokumen PDF untuk menemukan objek tertentu. Setiap objek diwakili oleh satu entri dalam tabel referensi silang, yang selalu 20 *byte* panjangnya.

III. METODE PENELITIAN

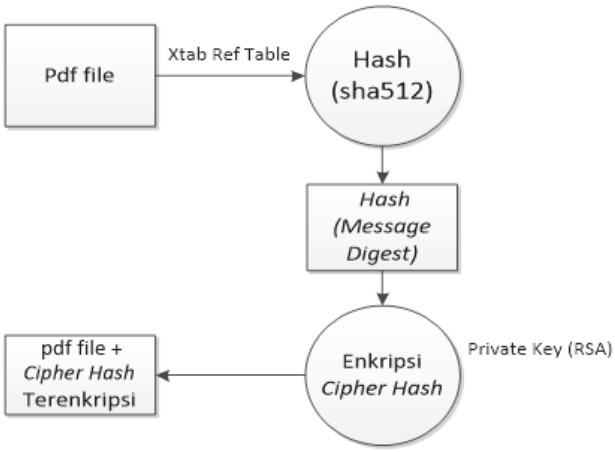
Penelitian yang dilakukan, diselesaikan melalui tahapan penelitian yang terbagi dalam empat tahapan, yaitu: (1) Identifikasi Masalah, (2) Perancangan sistem, (3) Implementasi sistem, dan (4) Pengujian sistem dan analisis hasil pengujian.

Gambar 5. Tahapan penelitian

Tahapan penelitian terbagi ke dalam 4 bagian. (1) Identifikasi Masalah, (2) Perancangan sistem, (3) Implementasi sistem, dan (4) Pengujian sistem dan analisis hasil pengujian.

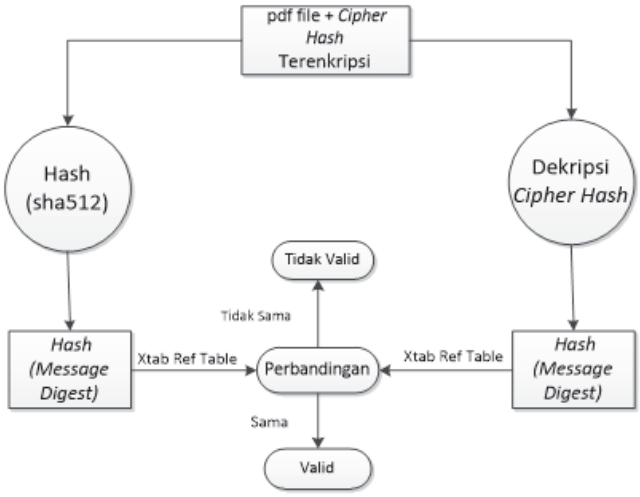
Sistem yang dirancang pada penelitian ini terdiri dari dua bagian utama. Proses *embedding* dan proses verifikasi. Pada proses *embedding*, proses awal adalah membaca *xref table* pada pdf, kemudian menghitung nilai *hash* dari *xref table* tersebut. Nilai *hash* dienkripsi sehingga menghasilkan *digital signature*.

Proses verifikasi terbagi menjadi dua proses, yaitu proses ekstraksi dan proses menghitung nilai *hash*. Proses ekstraksi bertujuan untuk membaca nilai *digital signature* yang ada di dalam dokumen. *Digital signature* didekripsi sehingga diperoleh nilai *hash* dari *xref table*.



Gambar 6. Pada *embedding* pada dokumen pdf

Proses awal dalam proses *embedding* adalah membaca *xref table* pada pdf, kemudian menghitung nilai *hash* dari *xref table* tersebut. Nilai *hash* dienkripsi sehingga menghasilkan *digital signature*.



Gambar 7. Proses verifikasi pada *file* pdf

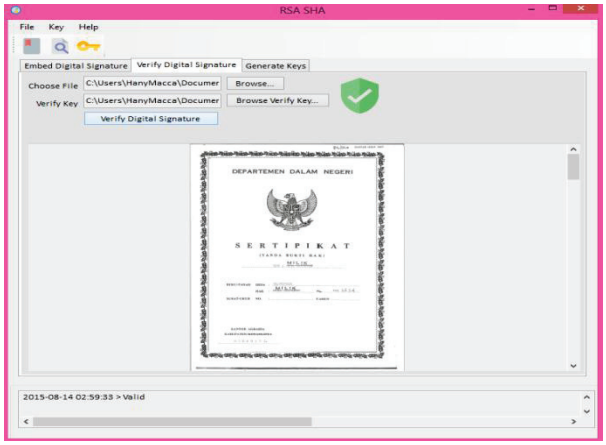
Proses Verifikasi terbagi menjadi dua proses, yaitu proses ekstraksi dan proses menghitung nilai *hash*. Proses ekstraksi bertujuan untuk membaca nilai *digital signature* yang ada di dalam dokumen. *Digital signature* didekripsi sehingga diperoleh nilai *hash* dari *xref table*.

IV. HASIL DAN PEMBAHASAN

Gambar 8. GUI untuk proses *embedding*

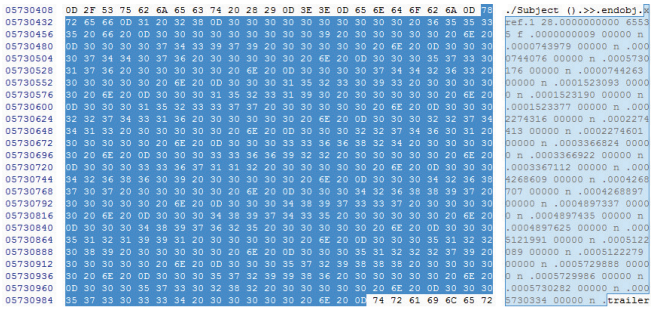
Gambar 8 *Interface* pada sistem yang mana menyediakan kolom-kolom untuk *inputan file* dokumen pdf, *file* kunci, dan *preview* tampilan sebelum dan sesudah hasil penyisipan, *interface* proses *embed* yang berfungsi untuk melakukan proses *embedding digital signature* pada dokumen sertifikat tanah digital. Sebagai masukan pada proses *embed* dokumen yaitu *file* sertifikat untuk disisipkan *digital signature* data (hasil *hashing* pada *xref table*) dan kunci publik yang langkah selanjutnya dilakukan proses enkripsi untuk mengenkripsi *file* data tersebut sehingga menjadi dokumen sertifikat yang memiliki *signature.*

Proses selanjutnya adalah proses verifikasi. Setelah ditentukan *file* data yang terenkripsi dan kunci pribadi yang *diinput* dari lokasi penyimpanan, langkah selanjutnya dilakukan proses verifikasi (ekstraksi) untuk mengecek keabsahan *file* dokumen sertifikat tersebut sehingga dapat diketahui dokumen digital sertifikat tanah tersebut asli atau dipalsukan.



Gambar 9. GUI untuk proses verifikasi

Gambar 9 menunjukkan *interface* proses *Verifying Digital Signature* yang berhasil. Proses ini berfungsi untuk melakukan proses verifikasi pengecekan apakah *file* yang diuji *valid (original)* atau tidak. Sebagai masukan pada proses verifikasi ini adalah sertifikat tanah digital yang *tersignature* dan *private key*.



Gambar 10. Contoh *Xref table* dengan panjang 570 *byte* pada *file* PDF.

Dimulai dengan kata kunci *xref*, dan berakhir sebelum kata kunci *trailer*. *Xref table* dengan panjang 570 byte dari total 5731074 byte (0.0099%).

Hash message dari XREF table pada Gambar 10 adalah:

```
32D36F3707DC39A7C0590392A61F17C6DDC35CDAB729D81E
654914B6530398A3AFF308BA436FD7FB616888115D38C24ACA
91E7E372E1489BDADAFDF6C0C2004A
```

Encrypted Hash message diperoleh dengan cara mengenkripsi *hash message* dengan menggunakan algoritma RSA. Kunci yang digunakan adalah kunci *private* yang hanya dimiliki oleh pemilik dokumen. *Encrypted hash message* yang dihasilkan adalah:

```
mSIpjbmLJbn4UgiftJhGpsnYuHMoPyf274JvAH3uiZnzVACizg
VN5KHhDiD3xoUiGbc6BHGdhGTF0+/uRTgwNeM7YoEHKsC0MTjT
wO5DFdQbgAEH66uuTUaYkhtGX6Rp2wMTIf8Vy0i8BTLIcfjrfY
5wAnZwcxoJ3f9noAaMU6E=
```

Proses selanjutnya yaitu *encrypted hash message* disisipkan ke dalam *file* PDF, dapat dilihat pada gambar 11.

```
005772E0  32 37 20 30 20 52 0D 3E 3E 0D 25 25 7B 6D 53 49  27 0 R.>>.%%{mSI
005772F0  70 6A 62 6D 4C 4A 62 6E 34 55 67 69 66 74 4A 68  pjbmLJbn4UgiftJh
00577300  47 70 73 6E 59 75 48 4D 6F 50 79 66 32 37 34 4A  GpsnYuHMoPyf274J
00577310  76 41 48 33 75 69 5A 6E 7A 56 41 43 69 7A 67 56  vAH3uiZnzVACizgV
00577320  4E 35 4B 48 68 44 69 44 33 78 6F 55 69 47 62 63  N5KHhDiD3xoUiGbc
00577330  36 42 48 47 64 68 47 54 46 30 2B 2F 75 52 54 67  6BHGdhGTF0+/uRTg
00577340  77 4E 65 4D 37 59 6F 45 48 4B 73 43 30 4D 54 6A  wNeM7YoEHKsC0MTj
00577350  54 77 4F 35 44 46 64 51 62 67 41 45 48 36 36 75  TwO5DFdQbgAEH66u
00577360  75 54 55 61 59 6B 68 74 47 58 36 52 70 32 77 4D  uTUaYkhtGX6Rp2wM
00577370  54 49 66 38 56 79 30 69 38 42 54 4C 49 63 66 6A  TIf8Vy0i8BTLIcfj
00577380  72 66 59 35 77 41 6E 5A 77 63 78 6F 4A 33 66 39  rfY5wAnZwcxoJ3f9
00577390  6E 6F 41 61 4D 55 36 45 3D 7D 0D 0A 73 74 61 72  noAaMU6E=}..star
005773A0  74 78 72 65 66 0D 35 37 33 30 34 33 31 0D 25 25  txref.5730431.%%
005773B0  45 4F 46 0D                                      EOF.
```

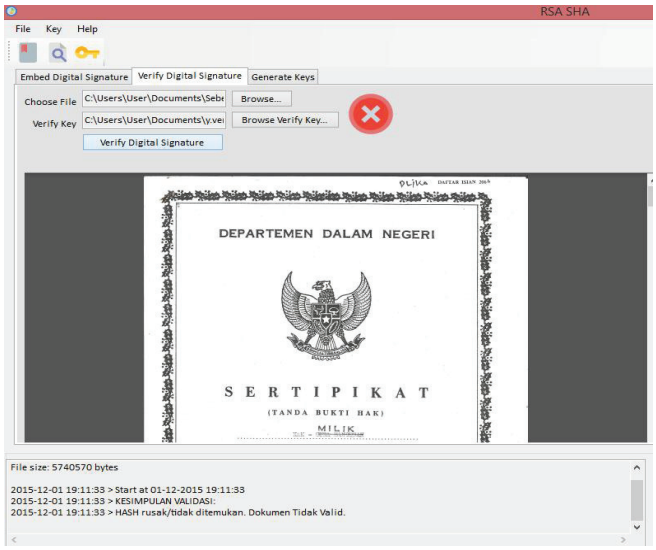
Gambar 11. Contoh hasil penyisipan *digital signature* pada *file* dokumen pdf.

Pada contoh tersebut, *digital signature* disisipkan dengan panjang 172 *byte*. Sehingga ukuran *file* PDF bertambah sebesar 172 *byte*.

Kode Program 1 Perintah untuk membaca *XREF Table*.

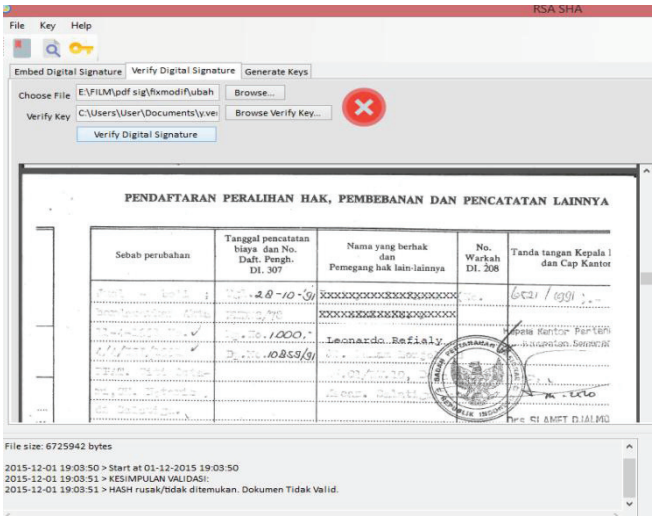
```
1.      internal static byte[]
2.          GetXREFTable(byte[] pdf)
3.      {
4.          byte[] keywordXREF =
5.          Encoding.Default.GetBytes("xref");
6.          byte[] keywordTrailer =
7.          Encoding.Default.GetBytes("trailer")
8.          ;
9.          int mulai = ArrayTool.Search(pdf,
10.         keywordXREF);
11.         int selesai = ArrayTool.Search(pdf,
12.         keywordTrailer);
13.         byte[] xrefTable = new byte[selesai
14.         - mulai];
15.         Array.Copy(pdf, mulai, xrefTable, 0,
16.         xrefTable.Length);
17.         return xrefTable;
        }
```

Kode Program 1 digunakan untuk mengambil potong*an XREF table* pada suatu dokumen PDF. Potongan dibatasi dengan kata kunci "*xref*". Proses pembacaan berakhir sampai dengan kata kunci "*trailer*".

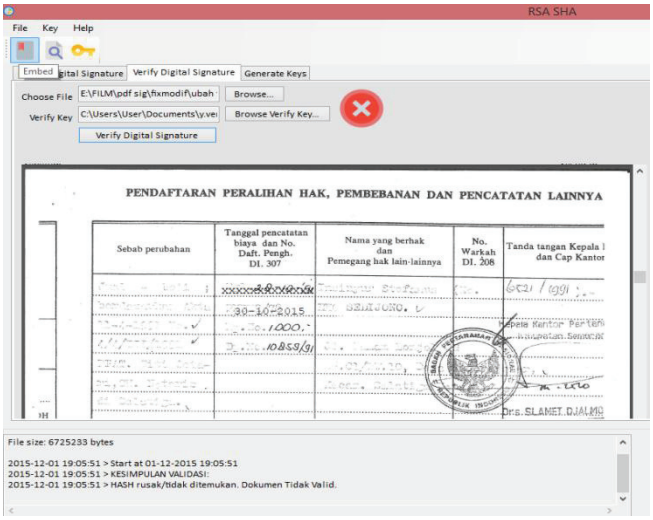


Gambar 12. Hasil verivikasi pada *file* pdf yang tidak valid

Pada Gambar 12 terlihat proses pengujian jenis manipulasi pada *file* pdf karena dalam proses verifikasi tidak ditemukan adanya *signature* atau *signature* mengalami kerusakan akibat modifikasi pada *file* pdf yang menunjukan bahwa *file* dokumen sertifikat tersebut terbukti tidak asli *(original)* atau palsu. Adapun jenis manipulasi yang diuji yaitu ubah nama pada dokumen, tambah *comment* pada *file* dokumen, ubah tanggal pada sertifikat, *highlight text* pada dokumen. Dalam proses pengujian terlihat sistem dapat dengan baik mendeteksi adanya perubahan pada dokumen pdf karena *signature* yang ada pada dokumen pdf mengalami kerusakan akibat proses modifikasi (dapat dilihat proses deteksi manipulasi pada Gambar 13, 14, 15 dan 16).

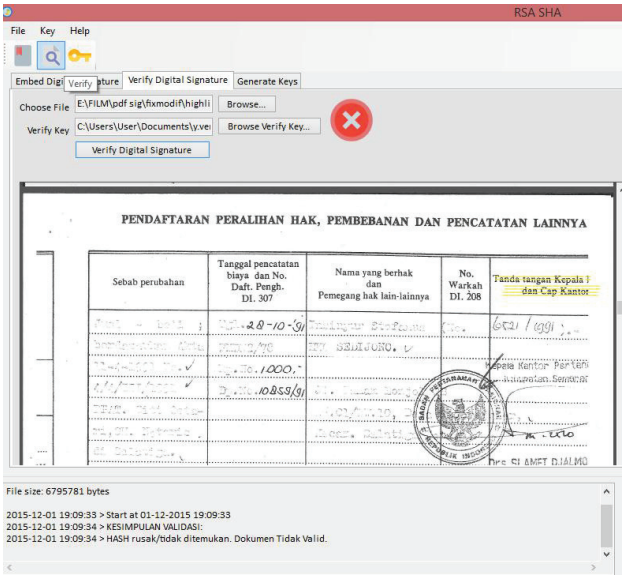


Gambar 13. Proses modifikasi ubah nama pada *file* dokumen pdf yang terdeteksi pada sistem.

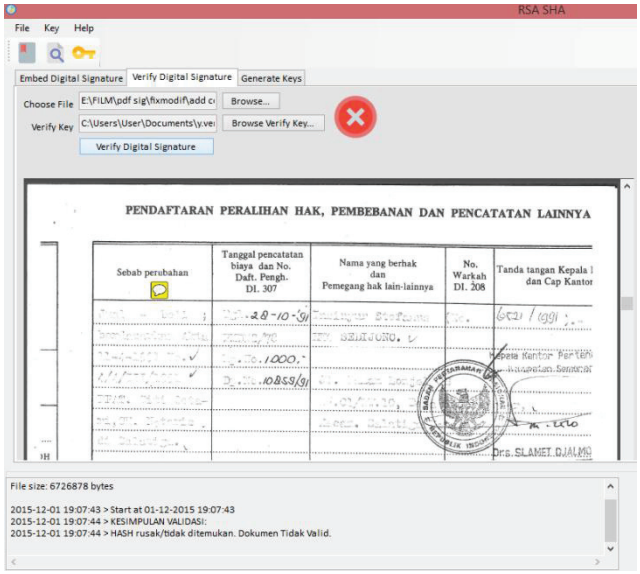


Gambar 14. Proses modifikasi ubah tanggal pada file dokumen pdf yang terdeteksi pada sistem.

e-ISSN : 2443-2229



Gambar 15. Proses modifikasi highlight text pada file dokumen pdf yang terdeteksi pada sistem.



Gambar 16. Proses modifikasi *add comment* pada *file* dokumen pdf yang terdeteksi pada sistem.

TABEL I
PENGUJIAN DETEKSI MANIPULASI

No	Jenis Manipulasi	Hasil Verifikasi
1	Ubah nama pada sertifikat.	Perubahan terdeteksi
2	Tambah *comment* pada dokumen	Perubahan terdeteksi
3	Ubah tanggal pada sertifikat	Perubahan terdeteksi
4	Highlight text	Perubahan terdeteksi

Pada Tabel I terlihat proses pengujian jenis manipulasi pada *file* pdf, dapat dilihat jenis manipulasi yang terdeteksi pada program, sehingga sistem dapat mendeteksi adanya tiap perubahan pada dokumen pdf karena *signature* mengalami kerusakan akibat proses modifikasi pada *file* pdf.

V. SIMPULAN

Berdasarkan penelitian yang telah dibuat, untuk mengamankan data sertifikat tanah digital yakni dengan menggunakan *digital signature* SHA-512 sebagai proses *hashing* bagian *xref table* dari *file* pdf sertifikat, selanjutnya dienkripsi dengan RSA menjadi *signature* kemudian akan disisipi ke dalam dokumen sertifikat tanah digital maka dihasilkan simpulan sebagai berikut:

1. Sistem pengamanan sertifikat tanah digital dapat mengidentifikasi ada tidaknya perubahan pada *file* dokumen sertifikat digital sehingga dapat disimpulkan bahwa sistem dapat memverifikasi keaslian dari *file* dengan menggunakan SHA 512 dan RSA
2. Tidak terjadi perubahan yang signifikan terhadap *file* dokumen sertifikat sebelum dan sesudah *signing (hashing)*, sehingga secara kasat mata kedua *file* terlihat sama
3. Sistem pengamanan *signature* pada sertifikat tanah digital sangat bermanfaat karena dapat mengetahui adanya proses manipulasi pada dokumen sertifikat dengan cepat sehingga dapat menyelesaikan masalah pemalsuan pada sertifikat Tanah oleh BPN ataupun pada PPAT tanpa perlu proses pengecekan manual yang lebih memakan waktu
4. Dengan menggunakan *xref table*, proses menghitung nilai *hash* tidak perlu dilakukan terhadap keseluruhan dokumen pdf sertifikat, sehingga lebih mempercepat proses verifikasi.

DAFTAR PUSTAKA

[1] Waspada, Banyak Sertifikat Tanah Palsu Yang Beredar. http://www.dppncw.com/2014/01/waspada-banyak-sertifikat-tanah-palsu.html. Diakses 10 Agustus 2015.
[2] Santoso, Urip. (2010). Hukum Agraria dan Hak-Hak Atas Tanah. Jakarta: Kencana.
[3] Rajeev Sobti, G. Geetha "Cryptographic Hash Functions: A Review" IJCSI, Volume 9, issue 2, March 2012, ISSN (online): 1694-0814
[4] Kumar Raghuvanshi, K., Khurana, P. & Bindal, P. 2014. Study and Comparative Analysis of Different Hash Algorithm. Journal of Engineering Computers & Applied Sciences 3, 1–3.
[5] Noroozi, E., Daud, S. M. & Sabouhi, A. 2014. Enhancing Secured Data Hiding Using Dynamic Digital Signature for Authentication Purpose. Jurnal Teknologi 68.
[6] Shah, M. A., Swaminathan, R. & Baker, M. 2008. Privacy-Preserving Audit and Extraction of Digital Contents. IACR Cryptology ePrint Archive 2008, 186.
[7] Prakash, Purohit, (2013) An Efficient implementation of PKI architecture based Digital Signature using RSA and various hash functions (MD5 and SHA variants
[8] Rivest, R. L., Shamir, a. & Adleman, L. 1978. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 120–126. (doi:10.1145/359340.359342)archive/macros/latex/contrib/supported/IEEEtran/
[9] Burrows, James, 2005, Secured Hash Standard, USA: US National Institute and Technology.
[10] J. Wolf. OMG WTF PDF. Chaos Communication Congress (CCC), Dec. 2010.
[11] Itabashi, K. 2011. Portable document format malware. Symantec white paper