Jurnal Cendikia Vol. XVI Cendikia 2018 *P-ISSN:0216-9436*
Bandar Lampung, Oktober 2018 *E-ISSN:2622-6782*

DIGITAL SIGNATURE DENGAN ALGORITMA SHA-1 DAN RSA SEBAGAI AUTENTIKASI

Sugiyatno[1], Prima Dina Atika[2]
[1)2)] Teknik Informatika; Universitas Bhayangkara Jakarta Raya; Jl. Raya Perjuangan, Marga Mulya, Bekasi Utara, Marga Mulya, Bekasi Utara, Kota Bks, Jawa Barat 17121 (021) 88955882;
e-mail: sugiyatno@dsn.ubharajaya.ac.id.

ABSTRAKS
Untuk meningkatkan daya saing tenaga kerja Indonesia dengan tenaga kerja asing, pada era MEA saat ini diberlakukan Surat Keterangan Pendamping Ijazah (SKPI) di seluruh Institusi Perguruan Tinggi di Indonesia. SKPI ini nantinya akan menjadi acuan seseorang dalam melamar pekerjaan di suatu instansi / perusahaan. Dan biasanya dalam melamar suatu posisi diperusahaan diperlukan SKPI yang telah dilegalisasi, namun di era digitalisasi ini manipulasi terhadap gambar, teks, atau berkas-berkas termasuk dokumen atau sertifikat hasil test, sangat mudah dilakukan. Sehingga dapat memberikan celah untuk melaksanakan praktik pemalsuan dokumen sertifikat. Dokumen digital merupakan salah satu solusi untuk mengantisipasi terjadinya pemalsuan atau perubahan dalam suatu dokumen, tentunya dokumen digital ini pun masih perlu pengamanan untuk menghindari perubahan dari isi dokumen oleh pihak yang tidak bertanggung jawab dengan cara melakukan autentikasi pada dokumen tersebut. Penerapan *Digital Signature* dirasa sangatlah tepat sebagai autentikasi pada suatu dokumen, yang mana *Digital Signature* ini berfungsi sebagai penanda pada suatu dokumen yang memastikan bahwa dokumen tersebut adalah asli dan tidak pernah dimodifikasi. Untuk memberikan metode proteksi maksimum untuk keabsahan suatu digital signature. Adapun metode pembuatan digital signature adalah dengan menerapkan algoritma SHA-1 dan RSA yang dirasa cukup untuk memberikan jaminan otentikasi pengirim dan penerima dokumen digital yang telah didistribusikan.
.
Kata Kunci: Kriptografi, Algortima RSA, Fungsi HASH, Algoritma SHA-1 Message Digest dan Tanda Tangan Digital (Digital Signature).

1. PENDAHULUAN

1.1 Latarbelakang

Sesuai dengan peraturan pemerintah nomor 81 tahun 2014 Setiap Sekolah Tinggi atau universitas harus mengeluarkan SKPI (Surat Keterangan Pendamping Ijazah) pada setiap lulusannya. SKPI sebagai dokumen yang menampung informasi hasil pencapaian akademik atau sebagai Rekam Jejak Mahasiswa dalam selama mengikuti kuliah".

Sebuah SKPI sangat dibutuhkan karena dapat mempermudah suatu instansi atau perusahaan untuk mengetahui rekam jejak baik mahasiswa ataupun alumninya. Dengan perkembangan bidang teknologi informasi, manipulasi terhadap teks, gambar, atau dokumen termasuk sertifikat, dengan mudah dapat dilakukan. Sehingga dapat memberikan celah untuk melaksanakan praktik pemalsuan dokumen dan sertifikat dengan memanipulasi isi dokumen

Dokumen digital merupakan salah satu solusi untuk mengantisipasi terjadinya pemalsuan atau perubahan dalam suatu dokumen, tentunya dokumen digital ini pun masih perlu pengamanan untuk menghindari perubahan dari isi dokumen oleh pihak yang tidak bertanggung jawab dengan cara melakukan autentikasi pada dokumen tersebut. Penerapan teknologi *Digital Signature* dirasa sangatlah tepat sebagai autentikasi pada suatu dokumen, yang mana *Digital Signature* ini berfungsi sebagai penanda pada suatu dokumen yang memastikan bahwa dokumen tersebut adalah asli dan tidak pernah dimodifikasi. Terdapat berbagai algoritma untuk pembuatan *Digital Signature* diantaranya adalah algortima SHA, RSA, DSA dan Elgamal.

Untuk memberikan metode proteksi maksimum untuk keabsahan suatu *digital signature*, pada penelitian ini penulis akan meneliti mengenai metode kriptografi dengan menggunakan algoritma SHA-1 dan RSA. Dimana algoritma tersebut akan digunakan untuk membuat digital signature yang akan digunakan sebagai autentikasi dari keabsahan suatu dokumen. Adapun kelebihan dari algoritma RSA adalah sulitnya memfaktorkan bilangan besar menjadi faktor-faktor primanya.

Menurut Rivest dan kawan-kawan, usaha untuk mencari faktor prima dari bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun, sedangkan untuk bilangan 500 digit membutuhkan waktu 1025 tahun (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik) sedangkan kelebihan dari algoritma SHA-1 adalah algoritma ini dikatakan lebih aman (secure) karena ia dirancang sedemikian rupa sehingga secara komputasi tidak mungkin menemukan pesan yang berkoresponden dengan message digest yang diberikan

P-ISSN:0216-9436
E-ISSN:2622-6782

1.2 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* artinya *secret* (rahasia) dan graphia artinya writing (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain (Ariyus, 2008).

Kriptografi diperkenalkan oleh orang-orang mesir lewat *hieroglyph* 4000 tahun yang lalu. Dikisahkan pada saat Julius Caesar ingin mengirim pesan rahasia kepada seorang jendral di medan perang melalui seorang kurir. Karena pesan rahasia, Julius Caesar tidak ingin pesan tersebut terbuka di jalan. Untuk mengatasinya ia mengacak pesan hingga menjadi pesan yang tidak dapat dipahami oleh siapapun kecuali jendralnya saja dan sang jendral telah diberitahu sebelum membaca pesan acak tersebut dengan mengganti susunan *alfabet* dari a, b, c yaitu a menjadi d, b menjadi e, dan c menjadi f dan seterusnya. Dari ilustrasi tersebut yang dilakukan Julius Caesar dengan mengacak pesan, disebut dengan enkripsi. Saat sang jendral merapikan pesan yang teracak, disebut dekripsi. Pesan awal yang belum diacak dan pesan yang telah dirapikan disebut *plaintext*, Sedangkan pesan yang telah diacak disebut *chipertext* (Ariyus, 2008).

Selain berdasarkan sejarah yang membagi kriptografi menjadi kriptografi klasik dan kriptografi modern, maka berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan lagi menjadi kriptografi Kunci Simetris (Symmetric-key Cryptography) dan kriptografi Kunci Asimetris (Asymmetric-key Criptography) (Munir, 2006).

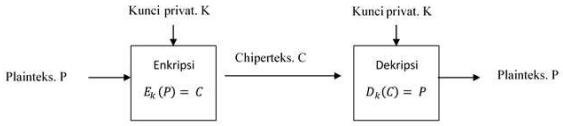
a. Kriptografi Kunci Simetris

Algoritma ini pengirim dan penerima pesan sudah mengetahui kuncinya sebelum mengirim pesan. Algoritma Simetris (*Symmetric Cryptography*) adalah suatu algoritma dimana kunci enkripsi yang digunakan sama dengan kunci dekripsi, sehingga algoritma ini disebut juga sebagai *single-key algorithm*. Keamanan pesan menggunakan algoritma ini tergantung pada kunci. Jika kunci diketahui orang lain, maka orang tersebut dapat melakukan enkripsi dan dekripsi terhadap pesan.

Algoritma yang memakai kunci simetris diantaranya adalah :

1. *Data encryption Standard* (DES)
2. *RC2, RC4, RC5, RC6*
3. *International Data Encryption Algorithm* (IDEA)
4. *Advanced Encryption Standard* (AES)
5. *One Time Pad* (OTP)

Berikut Ilustrasi penggunaan algoritma simetris :



Gambar 1. Skema Kriptografi Simetris
Sumber (Munir, 2006)

Sebelum pengiriman pesan, pengirim dan penerima memilih kunci yang sama untuk digunakan bersama, dan kunci bersifat rahasia (*secret-key algorithm*).

b. Kriptografi Kunci Asimetris

Kriptografi asimetris juga disebut dengan kriptografi kunci publik. Kunci yang digunakan enkripsi dan dekripsi berbeda dan setiap orang yang berkomunikasi mempunyai sepasang kunci yaitu :

1. Kunci umum (*public key*) yaitu kunci yang boleh semua orang tahu.
2. Kunci rahasia (*private key*) yaitu kunci yang dirahasiakan atau diketahui oleh satu orang saja.

Kunci tersebut berhubungan satu dengan yang lain. Walaupun kunci publik sudah diketahui, namun sangat sulit untuk mengetahui kunci private yang digunakan. Contoh algoritma kriptografi kunci publik diantaranya, RSA, Elgamal, DSA, dll.

1.3. *Digital Signature*

Secara garis besar *digital signature* adalah sebuah skema matematis yang secara unik mengidentifikasikan seorang pengirim, sekaligus untuk membuktikan keaslian dari pemilik sebuah pesan atau dokumen digital, sehingga sebuah digital signature yang autentik (sah), sudah cukup menjadi alasan bagi penerima untuk percaya bahwa sebuah pesan atau dokumen yang diterima adalah berasal dari pengirim yang telah diketahui.

Digital Signature menerapkan konsep Algoritma Sandi, dimana algoritma sandi itu sendiri adalah sebuah skema dengan tujuan kriptografis, adapun syarat yang harus dipenuhi dalam membuat algoritma sandi adalah sebagai berikut :

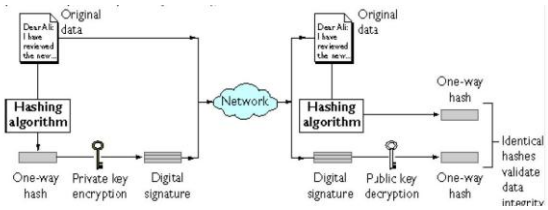
a. *Konfusi* / pembingungan, yaitu dari teks terang / jelas sehingga dibuat sulit untuk direkontruksikan secara langsung tanpa algoritma deskripsinya.
b. *Difusi* / peleburan, yaitu menghilangkan karakteristik dari sebuah teks terang / jelas.

Teknologi *digital signature* memanfaatkan teknologi kunci publik. Sepasang kunci *public privat* dibuat untuk keperluan seseorang. Kunci *privat* ada pada pemiliknya untuk membuat tanda tangan digital. Dan kunci publik diserahkan kepada siapa

P-ISSN:0216-9436
E-ISSN:2622-6782

saja yang ingin memeriksa tanda tangan digital pada suatu dokumen. Proses pembuatan dan pemeriksaan tanda tangan menggunakan teknik kriptografi seperti *hashing* (membuat "sidik jari" dokumen) dan enkripsi asimetris..

Berikut ini adalah bagan cara penggunaan *digital signature* pada suatu jaringan dan pesan :



Gambar 2. Bagan *Digital Signature* (Munir, 2006)

1.4. Algoritma *Rivest Shamir Adleman* (*RSA*)

Tingkat keamanan pada algoritma RSA terletak dalam sulitnya memfaktorkan bilangan menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan, maka keamanan algoritma RSA tetap terjamin. Algoritma RSA memiliki besaran sebagai berikut:

1. p dan q bilangan prima
2. n = p . q
3. Φ(n) = (p – 1) (q – 1)
4. e (kunci enkripsi)
5. d (kunci dekripsi)
6. m (plainteks)
7. c (cipherteks

Perumusan Algoritma RSA
Algoritma RSA pada teorema Euler :

$$a^{(\varphi(n))}=1(\text{mod } n) \quad (2.1)$$

dengan syarat :

a harus relatif prima terhadap n

$\varphi(n)=n(1-1/p\_1\ )(1-1/p\_2\ )\ldots(1-1/p\_r\ )$, yang dalam hal ini $p\_1,p\_2,\ldots,p\_r$ adalah faktor prima dari n. φ(n) adalah fungsi yang menentukan berapa banyak dari bilangan – bilangan 1, 2, 3 …. , n yang relatif prima terhadap n.
Berdasarkan sifat $a^k=b^k$ (mod n) untuk k bilangan bulat ≥ 1, maka persamaan (2.1) dapat ditulis menjadi :

$$a^{(k\varphi(n))}=1^k (\text{mod } n)$$

atau

$$a^{(k\varphi(n))}=1(\text{mod } n) \quad (2.2)$$

bila a diganti dengan m, maka persamaan (2.2) dapat ditulis menjadi :

$$m^{(k\varphi(n))}=1(\text{mod} n) \quad (2.3)$$

berdasarkan sifat ac=bc (mod n), maka bila persamaan (2.3) dikali dengan m menjadi :

$$m^{(k\varphi(n)+1)}=m (\text{mod } n) \quad (2.4)$$

yang dalam hal ini m relatif prima terhadap n, misalkan e dan d dipilih sedemikian sehingga :

$$e .d=1 (\text{mod}\varphi(n)) \quad (2.5)$$

atau

$$e .d=k\varphi(n)+1 \quad (2.6)$$

sulihkan (2.6) ke dalam persamaan (2.4) menjadi :

$$m^{(e .\ d)}=m (\text{mod } n) \quad (2.7)$$

persamaan (2.7) dapat dituliskan kembali menjadi :

$$[\![(m]\!]^{e\ })^d=m (\text{mod } n) \quad (2.8)$$

Yang artinya, perpangkatan m dengan e diikuti dengan perpangkatan dengan d menghasilkan kembali m semula. Berdasarkan persamaan (2.8), maka enkripsi dan dekripsi dirumuskan sebagai berikut :

$$E\_e (m)=c\equiv m^e \text{ mod } n$$
$$D\_d (c)=m\equiv c^d \text{ mod } n$$

1.5. *Secure Hashing Algorithm* (SHA)

Dalam kriptografi terdapat sebuah fungsi yang digunakan untuk aplikasi keamanan seperti otentikasi dan integritas pesan. Fungsi tersebut adalah fungsi hash. Fungsi hash merupakan suatu fungsi yang menerima barisan hingga dengan panjang sembarang dan mengkonversinya menjadi barisan hingga keluaran yang panjangnya tertentu. Fungsi hash dapat menerima masukan barisan hingga apa saja. Jika barisan hingga menyatakan pesan M, maka sembarang pesan M berukuran bebas dimampatkan oleh fungsi hash H melalui persamaan : h = H(M).

Keluaran fungsi hash disebut juga nilai hash (hash - value) atau ringkasan pesan (message digest / MD). Pada persamaan di atas, h adalah nilai hash dari fungsi H untuk masukan M. Dengan kata lain, fungsi hash mengkompresi sembarang pesan yang berukuran berapa saja menjadi nilai hash yang panjangnya selalu tetap. Fungsi hash merupakan fungsi yang bersifat satu arah (one-way function) dimana jika dimasukkan data, maka keluarannya berupa sebuah fingerprint (sidik jari), Message Authentication Code (MAC). Fungsi hash satu arah adalah fungsi yang bekerja satu arah yaitu pesan yang sudah diubah menjadi ringkasan pesan maka tidak dapat dikembalikan lagi menjadi pesan semula.

Fungsi hash H satu arah mempunyai sifat sebagai berikut :

1. Jika diberikan M (M adalah suatu data dalam hal ini berupa pesan), maka H(M) = h mudah dihitung.
2. Untuk setiap h yang dihasilkan, tidak mungkin dikembalikan nilai M sedemikian sehingga H(M) = h. Fungsi H disebut fungsi Hash satu arah.
3. Jika diberikan M, tidak mungkin mendapatkan M* sedemikian sehingga H(M) = H(M*). Bila diperoleh pesan M* semacam ini maka disebut tabrakan (collision). Maka sangat sulit menemukan

P-ISSN:0216-9436
E-ISSN:2622-6782

dua pesan yang berbeda yang menghasilkan nilai hash yang sama.

4. Tidak mungkin mendapatkan dua pesan M dan M* sedemikian sehingga H(M) = H(M*).

Sebuah fungsi hash satu arah H(M) beroperasi pada prapeta pesan M dengan panjang sebarang dan mengembalikan nilai hash h yang memiliki panjang tetap. Fungi hash dikembangkan berdasarkan ide sebuah fungsi kompresi. Fungsi satu arah ini menghasilkan nilai hash berukuran n pada input sebesar b. Input tersebut berupa suatu fungsi kompresi blok pesan dan hasil blok pesan sebelumnya. Sehingga nilai hash suatu blok pesan M adalah : $h_i = f(M_i, h_{i-1})$.

Dimana : h_i = nilai hash saat ini

M_i = blok pesan saat ini

h_{i-1} = nilai hash pokok sebelumnya

SHA (Secure Hashing Algorithm) adalah salah satu jenis dari algoritma fungsi hash. SHA terdiri dari 4 macam yaitu SHA-1, SHA-256, SHA- 384, SHA-521. Pada skripsi ini digunakan fungsi hash SHA-1 karena mempunyai nilai hash yang paling kecil.

Tabel 2.1. Macam - Macam SHA

Algoritma	Naskah (bit)	Blok (bit)	Kata (bit)	Nilai hash/*message digest* (bit)
SHA-1	$< 2^{64}$	512	32	160
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-521	$< 2^{128}$	1024	64	512

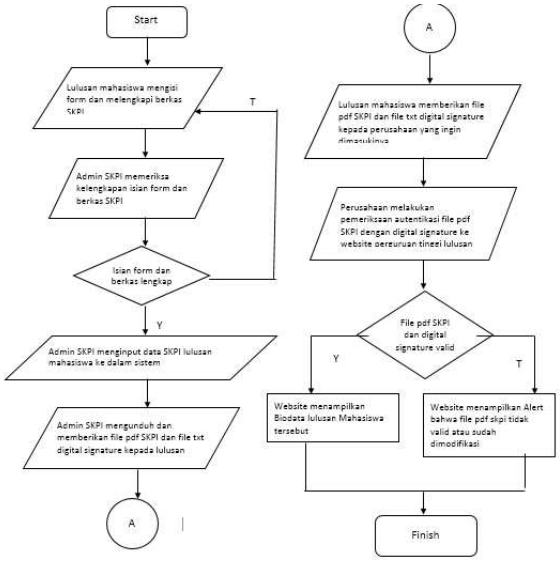
SHA–1 adalah algoritma hash yang paling banyak digunakan publik. SHA-1 merupakan salah satu jenis dari fungsi hash satu arah. SHA-1 menerima masukan berupa pesan dengan ukuran maksimum 2^{64} bit (2.147.483.648 gigabyte) dan menghasilkan nilai hash dengan panjang 160 bit. Nilai hash kemudian digunakan dalam DSA untuk menghitung tanda tangan digital pesan tersebut. Nilai hash pesan yang sama dapat diperoleh oleh penerima pesan ketika menerima pesan dari pengirim dengan cara memasukkan pesan tersebut pada fungsi SHA-1. SHA-1 dikatakan aman karena secara matematis tidak mungkin menemukan dua pesan yang berbeda yang menghasilkan nilai hash yang sama atau tidak mungkin menemukan pesan aslinya jika diberikan suatu nilai hash-nya. Pesan M dengan panjang L bit dimana $1 \leq L \leq 2^{64}$

Algoritma pada SHA-1 menggunakan :

1. Pesan yang tersusun pada kata 32-bit sebanyak 80 buah.
2. Lima variabel kerja masing – masing 32-bit.
3. Lima buah nilai hash masing – masing terdiri dari kata 32-bit $W_0, W_1, \ldots, W_{79}$. Lima variable kerja diberi label a, b, c, d dan e. Kata dari nilai hash diberi label $H_0^{(i)}, H_1^{(i)}, \ldots, H_4^{(i)}$ yang akan menampung nilai hash awal $H^{(0)}$ untuk diganti dengan nilai hash yang berurutan setelah blok pesan diproses $H^{(N)}$. (FIPS PUB 183-3, 2008)

2. PEMBAHASAN

2.1 Business Process Modelling

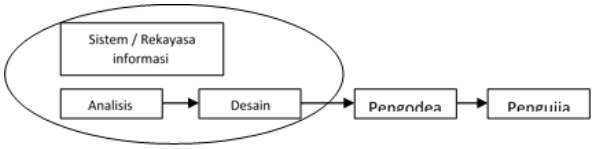


Gambar 3. Business Process Modelling

2.2 Metode Pengembangan Sistem

Model Waterfall (air terjun) sering disebut juga model sequensial linear atau alur hidup klasik. Model ini merupakan mode yang paling banyak dipakai oleh para pengembang software.

Berikut adalah gambar waterfall :



Gambar 4. Ilustrasi Model Waterfall
Sumber: (Rosa & M, 2011)

Adapun langkah-langkah pengembangan perangkat lunak dengan model waterfall yaitu :

1. Analisis kebutuhan perangkat lunak

P-ISSN:0216-9436
E-ISSN:2622-6782

Mengumpulkan kebutuhan user yang akan diterapkan pada perangkat lunak dan mendokumentasikan.

2. Perancangan.
   Mendesain perangkat lunak yang akan digunakan; seperti struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean dan mendokumentasikan.
3. Pembuatan kode program
   Desain harus ditranslasikan ke dalam program perangkat lunak dan mendokumentasikan
4. Pengujian
   Pengujian focus pada perangkat lunak secara dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalah dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.
5. Pendukung atau pemeliharaan
   Tahap pendukung dan pemelihara dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.3 Analisis

Tujuan dari analisa sistem adalah untuk menentukan masalah dan upaya untuk memperbaiki sistem, sehingga diharapkan dengan dilakukannya analisa sistem, maka permasalahan yang ada akan dapat teratasi.

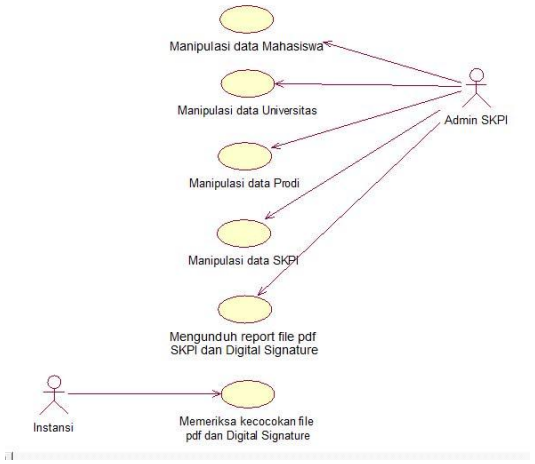
Permasalahan yang dihadapi adalah tentang bagaimana memberikan suatu authentikasi dalam suatu dokumen yang dianggap penting oleh pihak lain, agar tidak terjadi suatu kesalahan dalam memberikan suatu keputusan terhadap si pemilik dokumen tersebut. Untuk mengatasi hal ini penulis menerapkan sebuah metode kriptografi Digital Signature dalam sebuah dokumen, sebagai authentikasi dari keaslian dokumen tersebut, dimana dalam kriptografi ini menerapkan algoritma SHA-1 dan RSA dan untuk memudahkan dalam akses penggunaan aplikasi, agar pihak lain dapat mengakses aplikasi dimanapun maka dalam pembuatan aplikasi ini akan menggunakan bahasa pemrograman PHP.

2.4 Perancangan.

Penelitian ini menggunakan *Unified Manipulation Language* (UML) untuk menggambarkan bisnis proses, adapun bisnis proses yang akan diuraikan adalah *use case diagram, activty diagram, sequence diagram* dan *class diagram*. Aplikasi yang digunakan dalam membuat UML adalah *Rational Rose.*

1. *Use Case Diagram*

Dalam sistem ini akan dibangun *use case* untuk sistem *digital signature*, untuk lebih jelasnya *actor* yang berinteraksi dengan *use case* dalam sistem yang diajukan dapat dilihat pada gambar di bawah ini :

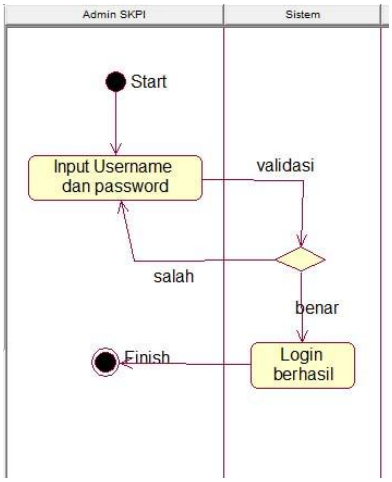


Gambar 5. *Use Case Diagram* SKPI

2. *Activity Diagram*

Activity diagram menggambarkan alur proses pada tiap-tiap *use case*. Berikut penulis uraikan dari setiap *activity diagram* :

a. *Activity Diagram Login*
   Activity diagram ini merupakan salah satu pengaman agar hanya orang-orang tertentu saja yang bisa masuk atau *login* ke dalam sistem ini, agar data yang bersifat rahasia dapat terjamin dan tidak ada penyalahgunaan terhadap sistem aplikasi yang telah dibuat. Di bawah ini merupakan gambar tampilan *activity diagram login.*

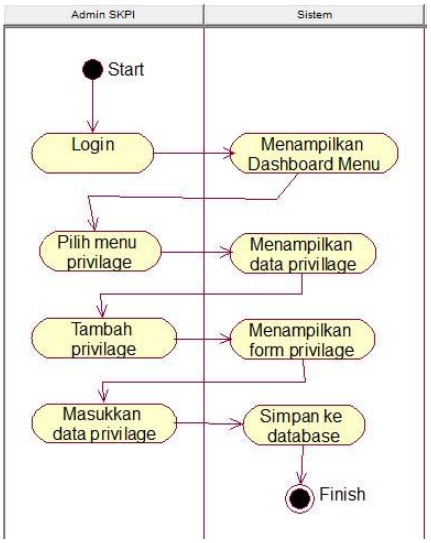


Gambar 6. *Activity Diagram Login*

b. *Activity Diagram* Manipulasi *Privilage User*
   Pada *activity diagram* manipulasi *privilage user*, aktor admin diharuskan *login* terhadap

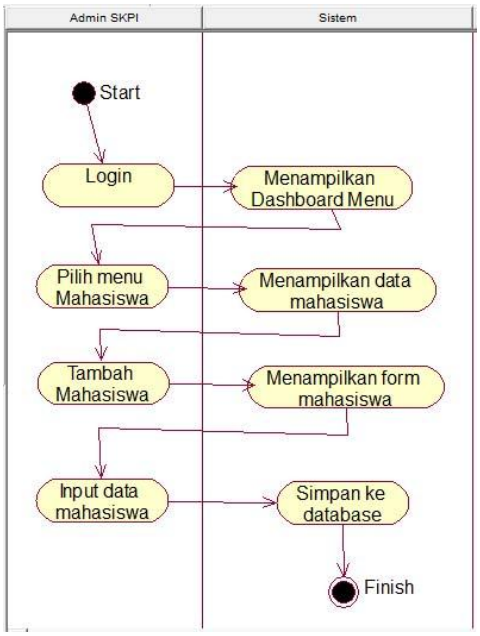
P-ISSN:0216-9436
E-ISSN:2622-6782

sistem untuk melakukan aktifitas manipulasi data *privilage user*. Adapun *activity diagram* manipulasi *privilage user* yaitu aktor admin membuka aplikasi → sistem menampilkan halaman *login* → aktor admin memasukkan data *username* dan *password* → sistem memeriksa apakah *username* dan *password* sudah benar atau belum, apabila sudah benar sistem akan menampilkan halaman menu dashboard dan apabila salah sistem akan menampilkan kembali halaman *login* → aktor admin memilih menu privilage → sistem menampilkan halaman data *privilage* → aktor admin dapat melakukan manipulasi data tambah data, edit data dan hapus data → apabila inputan data sudah terisi semua maka sistem akan menyimpannya kedalam database. Untuk lebih jelasnya merujuk pada gambar di bawah



Gambar 7. *Activity Diagram* Manipulasi *Privilage User*

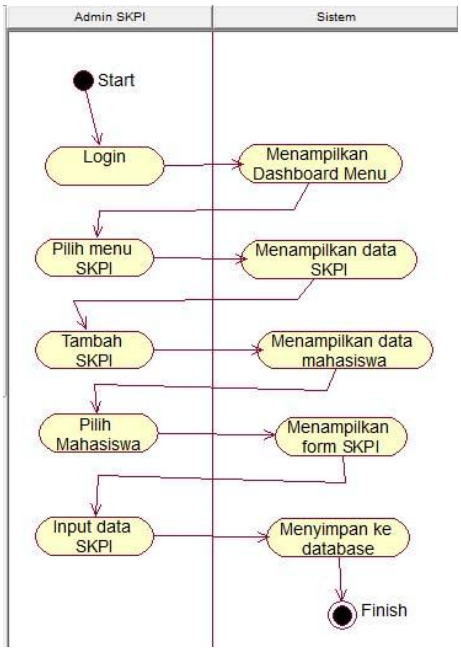
c. *Activity Diagram* Manipulasi Data Mahasiswa
Pada *activity diagram* manipulasi data mahasiswa, aktor admin diharuskan *login* terhadap sistem untuk melakukan aktifitas manipulasi data mahasiswa. Adapun *activity diagram* manipulasi data mahasiswa yaitu aktor admin membuka aplikasi → sistem menampilkan halaman *login* → aktor admin memasukkan data *username* dan *password* → sistem memeriksa apakah *username* dan *password* sudah benar atau belum, apabila sudah benar sistem akan menampilkan halaman menu dashboard dan apabila salah sistem akan menampilkan kembali halaman *login* → aktor admin memilih menu mahasiswa → sistem menampilkan halaman data mahasiswa → aktor admin dapat melakukan manipulasi data tambah data, edit data dan hapus data → apabila inputan data sudah terisi semua maka sistem akan menyimpannya kedalam database. Untuk lebih jelasnya merujuk pada gambar 8.



Gambar 8. *Activity Diagram* Manipulasi Data Mahasiswa

d. *Activity Diagram* Manipulasi Data SKPI
Pada *activity diagram* manipulasi data SKPI, aktor admin diharuskan *login* terhadap sistem untuk melakukan aktifitas manipulasi data SKPI. Adapun *activity diagram* manipulasi data SKPI yaitu aktor admin membuka aplikasi → sistem menampilkan halaman *login* → aktor admin memasukkan data *username* dan *password* → sistem memeriksa apakah *username* dan *password* sudah benar atau belum, apabila sudah benar sistem akan menampilkan halaman menu dashboard dan apabila salah sistem akan menampilkan kembali halaman *login* → aktor admin memilih menu SKPI → sistem menampilkan halaman data SKPI → aktor admin dapat melakukan manipulasi data tambah data, edit data dan hapus data → apabila inputan data sudah terisi semua maka sistem akan menyimpannya kedalam database. Untuk lebih jelasnya merujuk pada gambar 9.

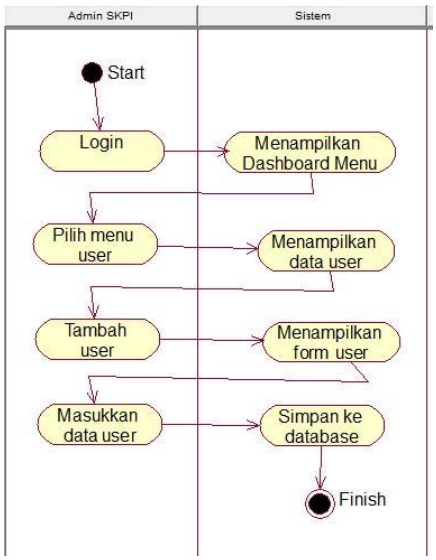
P-ISSN:0216-9436
E-ISSN:2622-6782



Gambar 9. *Activity Diagram* Manipulasi Data SKPI

e. *Activity Diagram* Membuat *Report File* PDF SKPI dan *Digital Signature*

Pada *activity diagram* membuat *report file* pdf SKPI dan *digital signature* ini akan diterapkan algoritma hashing SHA-1 dan juga proses pembangkitan kunci *private key* dan *public key* dan proses enkripsi dengan menggunakan algoritma kriptografi RSA. Pada *activity* ini aktor admin diharuskan *login* terhadap sistem untuk melakukan aktifitas membuat *report file* pdf SKPI dan *digital signature*. Adapun *activity diagram* membuat *report file* pdf SKPI dan *digital signature* yaitu aktor admin membuka aplikasi → sistem menampilkan halaman *login* → aktor admin memasukkan data *username* dan *password* → sistem memeriksa apakah *username* dan *password* sudah benar atau belum, apabila sudah benar sistem akan menampilkan halaman menu dashboard dan apabila salah sistem akan menampilkan kembali halaman *login* → aktor admin memilih menu SKPI → sistem menampilkan halaman data SKPI → aktor admin memilih salah satu data yang akan di buat *report file* pdf → sistem menampilkan *report file* pdf → aktor admin memilih mengunduh *file* pdf → sistem mengunduh *file* pdf → aktor admin kembali ke halaman sebelumnya → sistem menampilkan halaman data SKPI → aktor admin memilih buat *digital signature* pada data yang sudah di unduh *file* pdfnya → sistem meminta *file* pdf yang sudah di unduh sebelumnya → aktor admin mengunggah file pdf yang sudah di unduh sebelumnya → sistem membuat *digital signature* dengan menerapkan algoritma hashing SHA-1 dan algoritma kriptografi RSA pada file pdf yang telah diunggah tadi → sistem menempatkan public key dan *digital signature* dalam file txt → sistem mengunduh file txt. Untuk lebih jelasnya merujuk pada gambar 10.

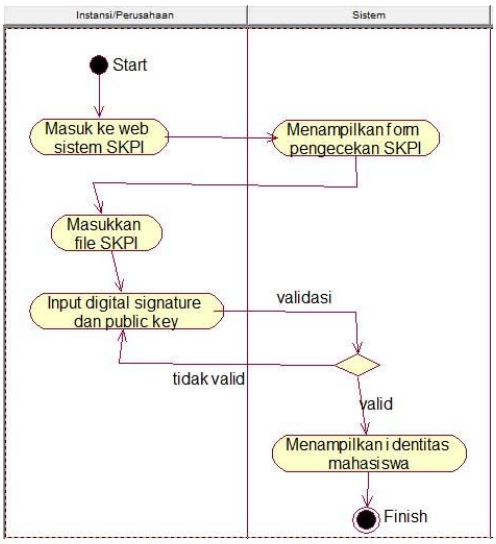


Gambar 10. *Activity Diagram* Membuat *Report File* PDF dan *gital Signature*

f. *Activity Diagram* Memeriksa Kecocokan *File* PDF dan *Digital Signature*

Pada *activity diagram* memeriksa kecocokan *file* pdf dan *digital signature*, aktor instansi/perusahaan tidak perlu *login* terhadap sistem untuk melakukan aktifitas memeriksa kecocokan *file* pdf dan *digital signature*. Adapun *activity diagram* memeriksa kecocokan *file* pdf dan *digital signature* yaitu aktor instansi/perusahaan membuka aplikasi → sistem menampilkan halaman form pemeriksaan SKPI → aktor instansi/perusahaan memasukkan data nomer SKPI, *public key* dan *digital signature* yang ada pada *file* txt dan mengunggah *file* pdf SKPI yang akan diperiksa → sistem memeriksa apakah *file* pdf dan *digital signature* cocok, apabila sudah cocok sistem akan menampilkan identitas mahasiswa pemegang SKPI tersebut dan apabila tidak cocok sistem akan menampilkan kembali halaman form pemeriksaan SKPI. Untuk lebih jelasnya merujuk pada gambar 11.

P-ISSN:0216-9436
E-ISSN:2622-6782

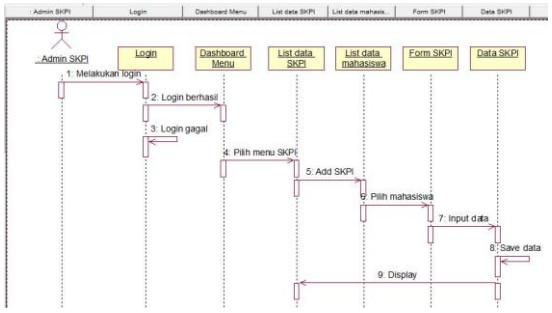


Gambar 11. *Activity Diagram* Memeriksa Kecocokan *File* Pdf dan *Digital Signature*

3. *Sequence Diagram*

a. *Sequence Diagram* Manipulasi Data SKPI

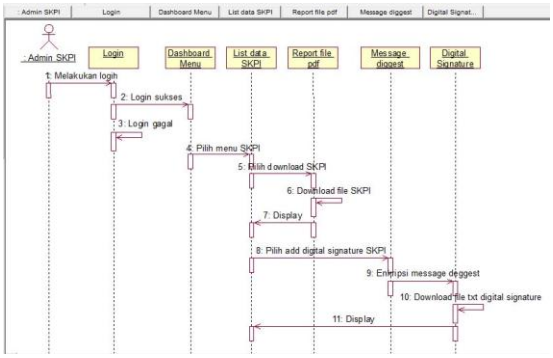
Hal pertama untuk melakukan manipulasi data SKPI adalah aktor admin harus melakukan login terhadap sistem, apabila login berhasil maka sistem akan menampilkan halaman menu dashboard dan apabila salah sistem akan menampilkan kembali halaman *login* → aktor admin memilih menu SKPI → sistem menampilkan halaman data SKPI → aktor admin menambahkan SKPI baru → sistem menampilkan halaman data mahasiswa yang sudah diinputkan sebelumnya tapi belum dibuatkan SKPI → aktor admin memilih salah satu mahasiswa yang akan dibuatkan SKPI → sistem menampilkan form SKPI → aktor admin menginputkan data SKPI dengan benar → sistem menyimpannya ke dalam database. Lebih jelas merujuk pada gambar 3.19.



Gambar 3.12. *Sequence Diagram* Manipulasi Data SKPI

b. *Sequence Diagram* Membuat *Report File* PDF SKPI dan *Digital Signature*

SKPI dan *digital signature* adalah aktor admin harus melakukan login terhadap sistem, apabila login berhasil maka sistem akan menampilkan halaman menu dashboard dan apabila salah sistem akan menampilkan kembali halaman *login* → aktor admin memilih menu SKPI → sistem menampilkan halaman data SKPI → aktor memilih salah satu data SKPI yang akan dibuat *report file* pdf → sistem menampilkan report pdf → aktor admin mengunduh file pdf → sistem mengunduh file → aktor admin kembali kehalaman sebelumnya → sistem menampilkan halaman data SKPI → aktor admin memilih create digital signature pada data SKPI yang telah diunduh *file* pdf sebelumnya → sistem meminta file pdf yang sudah diunduh sebelumnya → aktor admin mengunggah file pdf → sistem membuat *message deggest* dengan memanfaatkan algoritma hashing SHA -1 → kemudian dengan memanfaatkan algoritma kriptografi RSA sistem membangkitkan kunci *private keys* dan *public key* kemudian *public key* digunakan untuk mengenkripsi *message deggest* tersebut dan terbentuklah *digital signature* → sistem menempatkan *message deggest* dan *public key* ke dalam *file* txt → sistem mengunduh file txt tersebut. Lebih jelas merujuk pada gambar 13



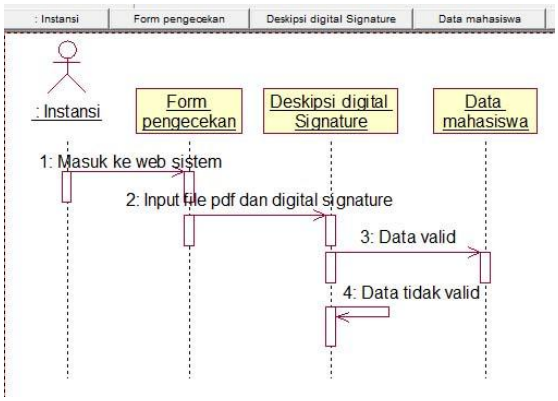
Gambar 13. *Sequence Diagram* Membuat *Report File* PDF SKPI dan *Digital Signature*

c. Sequence Diagram Memeriksa Kecocokan *File* PDF dan *Digital Signature*

Hal pertama untuk memeriksa kecocokan *file* pdf dan *digital signature* adalah aktor instansi/perusahaan tidak perlu login terlebih dahulu terhadap sistem → aktor instansi/perusahaan atau perusahaan masuk ke web sistem pemeriksaan SKPI → sistem menampilkan halaman form pemeriksaan SKPI → aktor instansi/perusahaan memasukkan data nomer SKPI, *public key* dan *digital signature* yang ada pada *file* txt dan mengunggah *file* pdf SKPI yang akan diperiksa → sistem mendekripsi *digital signature* dengan *public*

P-ISSN:0216-9436
E-ISSN:2622-6782

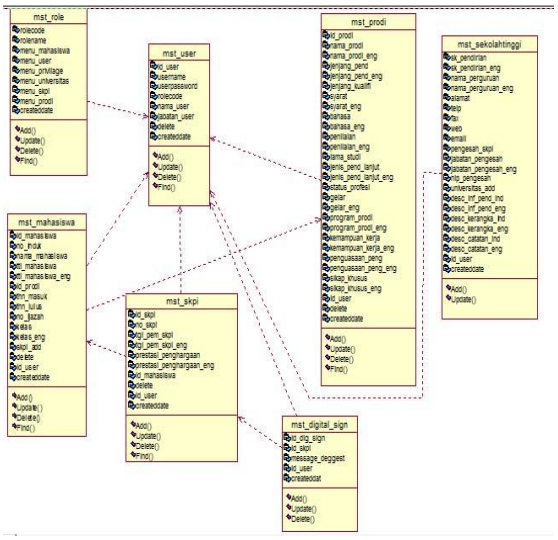
key dengan menggunakan algoritma kriptografi RSA → sistem memeriksa digital signature yang telah di dekripsi dengan file pdf yang telah di hashing dengan algoritma SHA -1, apabila cocok maka sistem akan menampilkan identitas dari pemilik SKPI tersebut dan apabila gagal maka sistem akan menampilkan kembali form pemeriksaan SKPI. Lebih jelas merujuk pada gambar 3.21.



Gambar 14. *Sequence Diagram* Memeriksa Kecocokan *File* PDF *dan Digital Signature*

4. *Class Diagram*

Class Diagram akan menggambarkan atau menampilkan struktur dari sebuah sistem. Sistem tersebut akan menampilkan sistem kelas, atribut dan hubungan antar kelas ketika suatu sistem telah selesai membuat *diagram*. Untuk lebih jelasnya dapat dilihat pada gambar 3.22 berikut:

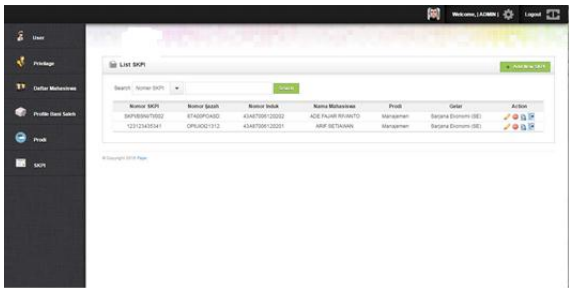


Gambar 15. *Class Diagram* Authentikasi Surat Keterangan Pendamping Ijazah (SKPI)

2.5 Hasil Pengujian.

a. Menu Tampilan Data SKPI

Halaman data SKPI ini berfungsi untuk melihat seluruh data SKPI yang telah dibuat untuk mahasiswa yang telah dinyatakan lulus. Lebih jelas merujuk pada gambar 16



Gambar 16. Tampilan Halaman Data SKPI

b. Menu Tampilan PDF *Online* SKPI

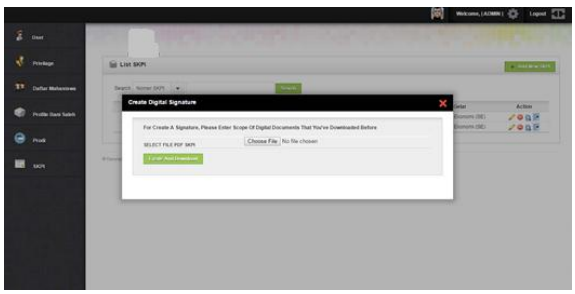
c.

Halaman PDF *online* SKPI ini berfungsi untuk menampilkan data SKPI dari seorang mahasiswa dalam bentuk pdf yang nantinya bisa di download atau dicetak. Lebih jelas merujuk pada gambar 17.

Gambar 17. Tampilan menu PDF *Online* SKPI

d. Form Create Digital Signature

Form Create Digital Signature ini berfungsi untuk membuat digital signature pada file pdf SKPI. Lebih jelas merujuk pada gambar 4.17



Gambar. 18. *Form Create Digital Signature*

P-ISSN:0216-9436
E-ISSN:2622-6782

3. KESIMPULAN

1. Algoritma *hashing* SHA -1 dan algoritma kriptografi *Rivest Shamir Adleman* (RSA) dapat dikombinasikan dengan baik dalam membuat sebuah *digital signature* pada file pdf.
2. Aplikasi *digital signature* sebagai autentikasi pada Surat Keterangan Pendamping Ijazah (SKPI) terbukti mampu diandalkan dalam autentikasi file pdf SKPI dari tindak pemalsuan dan modifikasi data.
3. Aplikasi ini dapat membantu pihak instansi / perusahaan dari rasa khawatir akan modifikasi atau pemalsuan data pada Surat Keterangan Pendamping Ijazah (SKPI).

PUSTAKA

A. Suhendar and Hariman Gunadi. 2002. *Visual Modeling Menggunakan UML dan Rational Rose*. Informatika. Bandung.

A.S. Rosa dan Salahuddin M. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Modula. Bandung.

Daqili, Ibnu. 2011. *Framework Codeigniter: Sebuah Panduan dan Best Practice*. E-book: koder.web.id. Pekanbaru.

Falani, A.Z. 2014. *Sistem Pengaman File dengan Menggunakan Metode RSA Kriptografi & Digital Signature*. Fakultas Ilmu Komputer, Universitas Narotama, Surabaya.

Lethbridge, T. and Laganiere, R. (2002). *Object-Oriented Software Engineering: Practical Software Development using UML and Java.* McGraw-Hill Education. New York.

Lycho, A.P. 2012. *Analisis Penerapan Digital Signature Sebagai Pengamanan Pada Fitur Workflow – DMS (Document Management System).* Program Studi Teknik Informatika Institut Teknologi Bandung.

Munir, Rinaldi. 2006. *Kriptografi*. Program Studi Teknik Informatika Institut Teknologi Bandung.

Munir, Rinaldi. 2004. "*Otentikasi dan Tandatangan Digital (Authentication and Digital Signature*", IF5054 Kriptografi, Program Studi Teknik Informatika Institut Teknologi Bandung.

Saipul. 2010. *Implementasi Tanda Tangan Digital Menggunakan Fungs Hash Algoritma SHA-256 dan RSA dalam Proses Otentikasi Data*. Universitas Ahmad Dahlan, Yogyakarta.

Sinlae, A.A.J. 2012. *Analisis Kriptosistem Menggunakan Digital Signature berbasis Algoritma SHA-512 dan RSA*. Magister Sistem Informasi, Universitas Kristen Satya Wacana, Salatiga.

Supriyanto, Aji. 2009. *Pemakaian Kriptografi Kunci Publik untuk Proses Enkripsi dan Tanda Tangan Digital pada Dokumen E-Mail*. Fakultas Teknologi Informasi, Universitas Stikubank, Semarang.

Wahyuni, Ana. 2011. *Aplikasi Kriptografi untuk Pengamanan E-Dokumen dengan Metode Hybrid : Biometrik Tandatangan dan DSA (Digital Signature Algorithm)*. Universitas Diponegoro, Semarang.

Whitten, J.L. et.al. 2004. *Systems Analysis and Design Methods.* McGraw-Hill Education. New York.

Yuana, Rosihan Ari. 2015. *Panduan Praktis OOP di PHP*. E-book: blog.rosihanari.net. Solo.