Rancang Bangun Keamanan *Web* *Service* Dengan Metode Ws-Security

**Ari Muzakir\*1, Ahmad Ashari2**

1Program Studi Teknik Informatika, Universitas Bina Darma; Jln. A Yani No 12 Palembang, Telp: +62711-515679,515581

2Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: **\*****1ariemuzakir@gmail.com**, 2ashari@ugm.ac.id

***Abstrak***

*Web service menggunakan teknologi XML dalam melakukan pertukaran data. Umumnya penggunaan web service terjadi pertukaran data ataupun informasi penting yang perlu dijaga keamanannya. Bentuk pengamanan yang diterapkan pada web services adalah dengan penggunaan teknik kriptografi kunci-publik. Adapun serangan dapat berupa pengintaian, perusakan maupun pencurian data. Salah satu cara penyelesaian terbaik adalah dengan membuat data tersebut tidak dapat dibaca orang lain.*

*Implementasi yang telah dilakukan dengan menggunakan library keamanan akan memberikan kemudahan dalam membangun keamanan web service karena dengan dukungan library XMLSEC sebagai library pendukung dan library class\_wss yang telah dibangun mampu mengatasi masalah keamanan pada jalur transport khususnya untuk otentikasi, otorisasi, dan konfidensialitas pesan SOAP request. Model WS-Security dengan menggunakan XML Signature, XML Encryption, serta Security Token yang memanfaatkan algoritma kriptografi RSA dengan panjang kunci 1024 bit mampu memberikan perlindungan terhadap transmisi data antara client dan server web service. Pengujian yang dilakukan pada web service dengan menerapkan model library class\_wss sebagai keamanan web service yang dibangun memberikan hasil yang baik, yaitu pesan SOAP request terenkripsi dan mampu didekripsi dengan baik serta dapat tertandatangani dan dicek keasliannya.*

***Kata Kunci :*** *Keamanan Web Service, XML Signature, XML Encryption, Security Token.*

***Abstract***

*Web service uses XML technology to exchange data in. Generally, the use of the web service exchanges data or important information that needs to be guarded security. Form of security is applied to web services is to use public-key cryptography techniques. The attack can be a reconnaissance, destruction or theft of data. One way the best solution is to create data that can not read anyone else, even if someone else managed to retrieve the data, he will not be able to read it.*

 *The implementation was done by using the security library will provide facilities in developing a web security service for the library support XMLSEC as library supporters and library class\_wss that have been built able to overcome the problem of security on the transport path, especially for authentication, authorization, and confidentiality request SOAP message. Model WS-Security using XML Signature, XML Encryption, and Security Token which utilizes the cryptographic algorithm RSA with 1024 bit key length to provide protection against transmission of data between client and server web service. Tests performed on the web service by implementing a security model class\_wss library web service that is built to give good results, the SOAP request message is encrypted and decrypted with a good and able to sign and check their authenticity too.*

***Keywords:*** *Web Service Security, XML Signature, XML Encryption, Security Token.*

1. PENDAHULUAN

S

aat ini *web service*s menjadi sangat populer di *enterprise* karena kemampuannya dalam mengintegrasikan aplikasi-aplikasi yang berbeda *platform* dengan menggunakan dokumen XML. XML (*eXtensible Markup Language*) adalah sebuah standar untuk mendefinisikan data dalam format yang sederhana dan fleksibel. *Web service* mendukung komunikasi antar aplikasi dan integrasi aplikasi dengan menggunakan XML dan *Web*. Faktor keamanan pada jalur komunikasi antara *client* ke *server* *web service* itu belum sepenuhnya terjamin. Hal ini dibuktikan dengan banyaknya faktor yang menimbulkan celah-celah ancaman terhadap *web service* tersebut seperti yang telah dilakukan oleh penelitian terdahulu.

Selain itu, pada kerahasiaan pesan yang dikirimkan melalui *web service* masih berupa data XML. Sehingga hal ini menyebabkan terjadinya data yang tidak asli ketika sampai di sisi penerima. Walaupun pesan telah di enkripsi menggunakan suatu algoritma maka bukan berarti bahwa pesan yang di terima oleh penerima benar-benar masih asli, karena bisa saja bahwa struktur pesan telah berubah ketika pesan dikirimkan atau ketika diterima.

Kemudian masalah keamanan *web service* pada kasus-kasus sebelumnya kebanyakan penelitian dilakukan pada satu model keamanan atau standar keamanan untuk *web service*. Sehingga dengan adanya sistem keamanan yang seperti ini dirasakan masih kurang memberi suatu perlindungan yang maksimal terhadap ancaman keamanan *web service*  antara *client* ke *server service* sendiri walaupun secara umum sudah mampu mencukupi. Masih adanya kendala mengenai *web service* yaitu beberapa pihak yang masih merasa ragu untuk menerapkan *web service*, khususnya mereka yang menggunakan jaringan internet pada transaksinya. Keraguan ini dilihat dari tingkat keamanan dari teknologi *web service*. Aspek keamanan menjadi sangat penting untuk menjaga data atau informasi agar tidak disalahgunakan ataupun diakses secara sembarangan [1]. *Transport Layer Security* (TLS) yang digunakan untuk mengotentikasi dan *A*mengenkripsi pesan berbasis *web* tidak memadai untuk melindungi pesan SOAP karena dirancang untuk beroperasi antara dua *endpoint*. TLS tidak dapat mengakomodasi *web* *service* dalam kemampuannya untuk meneruskan pesan ke beberapa *web* *service* lain secara bersamaan. Pengolahan model *web* *service* membutuhkan kemampuan untuk dapat memberikan pengamanan pesan SOAP dan dokumen XML mulai dari *client, service provider, dan intermediary services*. Selanjutnya teknologi yang mungkin dapat dimanfaatkan untuk meningkatkan kerahasiaan dan integritas dari *web service* yaitu *Secure Socket Layer* / *Transport Layer Security* (SSL/TLS) serta *message-level security* seperti yang telah disediakan WS-*Security* [2]. WS-*Security* juga mengatur cara menyisipkan *security token* dalam pesan Simple Object Access Protocol(SOAP) dalam bentuk *plaintext* maupun dalam bentuk biner, seperti sertifikat X.509 [3].

Oleh karena itu, penelitian ini akan mencoba menghadirkan sebuah implementasi dari *prototype* keamanan *web service* berbasis pengamanan *service to services* yang dapat memproses dan mengamankan data yang diterima dari *client* sebelum di simpan ke *database server* dengan cara mengenkripsi dan menandatangani serta menyisipkan *security token* pada pesan SOAP *request* dan *response* dengan memanfaatkan dua standar keamanan yang telah ada yaitu XML *Encryption* dan XML *Signature*. Adapun jalur komunikasi antara user ke *service* *client* menggunakan keamanan berbasis SSL atau disebut *Hypertext Transfer Protocol Secure* (HTTPS).

Beberapa penelitian yang telah dilakukan berkenaan dengan keamanan *web* *service* ini diantaranya mengenai spesifikasi dari keamanan *web service*s dan bagaimana spesifikasi tersebut menanggulangi ancaman terhadap keamanan *web service*s. Baik dari segi *security web service* masih belum matang seperti *Common Object Request Broker Architecture* (CORBA) dan *Remote Method Invocation* (RMI) [3].

Selanjutnya, analisa mengenai bagaimana mengatasi tantangan pada keamanan *web* *service* dengan menyajikan keamanan kerangka atau *framework* terpadu yang didasarkan pada penggunaan otentikasi, otorisasi, kerahasiaan, dan mekanisme integritas pada *web service* serta untuk mengintegrasikan dan menerapkan mekanisme keamanan tersebut untuk membuat *web service* kuat terhadap serangan [2]. Penelitian mengenai penyajian suatu metode yang komprehensif untuk suatu jaminan layanan keamanan dalam SOA. dimana metode yang diusulkan mendefinisikan tiga tahap yaitu *security* *analysis*, arsitektur jaminan keamanan, dan identifikasi Standar WS-*Security* [4].

Selain itu penelitian terhadap keamanan *web service* juga pernah dilakukan pada integrasi data laporan kejadian perkara satuan reserse kriminal (satreskrim) yang dilengkapi dengan mekanisme keamanan internal, dimana yang dilakukan pada implementasi mekanisme keamanannya adalah menambahkan fungsi-fungsi keamanan pada *tool* NuSOAP yang mana digunakan sebagai otentikasi serta untuk kerahasiaan pesan SOAP menggunakan kriptografi AES 128[5]. Selanjutnya untuk implementasi terhadap otentikasi user untuk dokumen XML dengan menggunakan *username token* juga pernah dilakukan , melakukan pembuktian terhadap validasi dokumen XML dan melakukan pengujian terhadap dokumen XML [1]. Selanjutnya Untuk mengimplementasikan suatu XML *Signature* untuk memperoleh dokumen XML yang *secure* pada kasus transkrip *online*. Dengan cara memperoleh transkrip yang memiliki tipe format XML yang terdapat *digital Signature*-nya [6].

Kemudian untuk mengimplementasikan algoritma RSA untuk pembuatan pasangan kunci public dan kunci privat guna proses enkripsi dan dekripsi. Selain itu RSA juga berperan menunjukan jangkauan data yang dapat diproses. Selanjutnya mengimplementasikan *message digest* untuk fungsi hash SHA-1 yang digunakan untuk proses penandatanganan dokumen XML [7]. Penelitian lainya yaitu mengenai data XML yang dienkripsi menggunakan kunci publik dengan algoritma RSA dengan hasil implementasinya berupa dua buah program komputer yaitu findkey.exe dan crypto.exe yang dibuat menggunakan bahasa pemrograman C [8].

2. METODE PENELITIAN

## 2.1 Analisa Sistem

Secara umum sistem yang akan dibangun dalam penelitian ini adalah model *prototype* dari keamanan *web service* yang akan diujikan pada suatu data nilai. Pada penelitian ini, model keamanan *web service* menggunakan metode WS-*Security* memanfaatkan dua teknologi yang sudah ada yaitu XML *Signature* dan XML *Encryption* sebagai model pengamanan dokumen XML yang memanfaatkan kriptografi kunci publik RSA yang akan digunakan pada model keamanan *client server web service*. Proses enkripsi dan dekripsi pada kriptografi ini terapkan pada *client service* dan *server service* untuk pengamanan jalur komuikasi yang mana menggunakan dua buah kunci yaitu kunci publik dan kunci rahasia. Model dari *prototype* keamanan *web service* antara *client service* dan *server service* ini dimulai dari pengiriman data dari user menggunakan SSL ke *client service*, kemudian komunikasi antara *client service* dengan *server service* dari *web service*. Dimana data XML akan dienkripsi (*encrypt*) dan ditandatangani (*signing*) dari *client service* dan akan didekripsi (*decrypt*) serta diverifikasi ketika diterima oleh *server service*, selanjutnya data hasil dekripsi akan disimpan pada *database* *server*.

Analisa kebutuhan sistem menentukan bagaimana *user*, data, proses, dan teknologi informasi dapat saling terhubung, dengan analisa kebutuhan sistem diharapkan dapat diuraikan secara utuh menjadi komponen-komponen suatu sistem dengan tujuan identifikasi, mengevaluasi permasalahan dan kebutuhan sesuai dengan yang diharapkan, hal ini dibagi menjadi dua yaitu sebagai berikut:

*2.1.1 Analisa kebutuhan fungsional*

Kebutuhan fungsional merupakan kebutuhan terkait dengan fungsi dan kemampuan sistem, Didalam pengimplementasian sistem otentikasi *user* pada *web service* ini dibagi menjadi dua bagian, yaitu berdasarkan pengimplementasian di *client* dan di *server*.

1. *Client* menghasilkan *web service* *request* yang kemudian akan diterima oleh *client service* sebelum dilanjutkan ke *server service*. Tahap ini berkaitan dengan proses-proses yang dilakukan oleh *client* untuk melakukan *request* kepada *web service* dengan menggunakan *username token*. Terdapat beberapa langkah yang dilakukan oleh *client* yaitu :
2. Menginisialisasi *username token,* cara ini mengimplementasikan untuk kebutuhan inisialisasi *username token* kedalam SOAP request yang dikirimkan oleh *client service* ke *server service* untuk meminta layanan *web* *service* yang dituju serta penggunaan metode *username token* dengan cara menyisipkan ke dalam SOAP *Header*. Fungsi dari *username token* ini adalah untuk memvalidasi pesan SOAP request yang dikirimkan oleh *client*.
3. Melakukan pemanggilan metode yang dibutuhkan. Setelah menyisipkan *username* dilakukan pemanggilan salah satu metode yang telah disediakan pada *web service*. Jika metode yang dipanggil menggunakan input, maka akan disisipkan pada bagian SOAP *Body*.
4. *Server service* akan mengotentikasi *client service* dan mengembalikan respon ke *client*. Tahap ini menjelaskan beberapa proses yang dilakukan oleh *server* *web service* setelah menerima SOAP *Request* dari *client service*. Beberapa proses tersebut adalah:
5. Memastikan integritas pesan. Pilihan penggunaan untuk mengamankan komunikasi antara *client* dan *web service* menentukan bagaimana integritas pesan dibuat dan diverifikasi. Pada penggunaan keamanan *web service* dengan keamananan *end to end security* digunakan WS-*Security* untuk memastikan integritas pesan.

b. Mengotentikasi pengguna. Pada tahap ini, *web service* akan melakukan otentikasi pengguna melalui pesan SOAP. Jika otentikasi pengguna berhasil, maka *web service* akan memberikan *service* apa saja yang dapat digunakan berdasarkan method yang diminta.

1. Memvalidasi *password*. Pada tahap ini dilakukan validasi *password* pada SOAP request yang dikirimkan ke *web service* dengan mengecek *username token*. Jika validasi berhasil maka proses selanjutnya *web service* memberikan response ke *client* dengan *service* yang diminta sebelumnya.
2. Memberikan respon ke *client.* Pada tahap terakhir sesuai permintaan *client* mengenai *service* apa saja yang di minta sebelumnya, *server* melakukan respon tersebut dengan mengirimkan SOAP Response.
3. Mengenkripsi,mendekripsi, menandatangani, dan memverifikasi data XML. Pada tahap ini, baik *web service client* maupun *web service server* akan melakukan enkripsi pesan yang akan dikirimkan dari dan ke *web* *service* menggunakan XML *Encryption* dengan kriptografi RSA, namun sebelum dokumen di enkripsi terlebih dahulu dilakukan penandatangan *digital* menggunakan XML *Signature* dengan kriptografi RSAwithSHA-1. Hal ini diperlukan untuk pengamanan pada jalur transport antar *service* sehingga data dokumen yang telah di tanda tangani dan di enkripsi nantinya dapat diterima oleh pengguna yang memiliki hak otorisasi.

*2.1.2 Analisa kebutuhan non-fungsional*

Analisis kebutuhan non-fungsional dari penelitian ini adalah sebagai berikut:

1. Faktor waktu respon dari site internal maupun eksternal melalui *web service* yang tidak bisa diprediksi.
2. Proses enkripsi dan dekripsi pesan SOAP yang membutukan waktu yang tidak bisa diprediksi di sisi *web service*.

## 2. 2 Perancangan Sistem

 Sistem aplikasi yang akan dibangun memiliki arsitektur keamanan secara umum seperti pada Gambar 1, dimana setiap *request* dari *client* akan dilakukan otentikasi, otorisasi, dan konfidensialitas. Otentikasi dilakukan ketika *client* berhasil melakukan *login*  dan akan diberikan akses ke sumber daya sesuai dengan hak aksesnya dengan memberikan otorisasi layanan yang telah ditentukan pada *Header username token*, sedangkan konfidensialitas di gunakan pada proses enkripsi dan dekripsi.



Gambar 1 Model Keamanan Antara Client service dan Service Server dari Web Service

Gambar 1 memperlihatkan model dari keamanan *web service* antara *client service* dan *server service*, dimana gambaran umum dari keamanan sistem ini dimulai dari pengiriman data dari user menggunakan SSL ke *client service*, kemudian komunikasi antara *client service* dengan *server service* dari *web service*. Selanjutnya data XML akan dienkripsi (*encrypt*) dan ditandatangani (*signing*) serta menyertakan *username token* dari *client service* dan akan didekripsi (*decrypt*), di verifikasi serta di cek *username token* ketika diterima oleh *server service*, selanjutnya data hasil dekripsi akan disimpan pada *database* *server*.

 Rancangan mekanisme otentiksi *user* bertujuan untuk membuktikan otentikasi identitas dari *user* yang melakukan login ke sistem dan meminta layanan keamanan data. Pada Gambar 2, disajikan sebuah mekanisme otentikasi *user* terhadap sistem.



Gambar 2 Rancangan Mekanisme Otentikasi User Pada Web Service

Perancangan mekanisme keamanan data ini bertujuan untuk memberikan gambaran mengenai kerahasiaan data dalam proses enkripsi dan proses dekripsi yang melibatkan algortima kunci publik. Selain itu, mekanisme keamanan data juga berupa penandatangan *digital* atau *signing* serta *username token*. Enkripsi dan *signing* terjadi antara *client service* dan *server service* dimana bertujuan untuk mengamanakan jalur transmisi pada *web* *service* sendiri. Rancangan ini dapat diperlihatkan pada Gambar 3.



Gambar 3 Rancangan Mekanisme Kerahasiaan Data User Pada Web Service

Pada penelitian ini rancangan keamanan tersebut dititikberatkan pada beberapa mekanisme keamanan, yaitu:

1. Otentikasi, yaitu sebuah mekanisme keamanan yang bertujuan untuk membuktikan atau verifikasi otentikasi *user*/mesin (*client*) yang meminta layanan, sehingga dengan demikian *service provider* (*server*) selanjutnya dapat menentukan apakah *user*/mesin (subjek) diperbolehkan untuk dilayani atau tidak. Mekanisme otentikasi pesan akan dilakukan dengan cara menyediakan fungsi-fungsi dan rutin program untuk keperluan pembuatan elemen *username token,* konfigurasi elemen *Header* SOAP *request* serta pengecekan otentikasi dari *security token* tersebut pada *server service.* Selain itu otentikasi juga dilakukan dengan cara menandatangani atau *signing* pesan SOAP *request* maupun *response* dengan menggunakan XML *Signature*. Fungsinya adalah untuk mengetahui keaslian data ketika dalam proses transmisi.
2. Konfidensialitas, yaitu menjaga kerahasiaan pesan dari orang yang tidak memiliki hak otorisasi yang ada pada jaringan *web*, maka solusinya adalah dengan melakukan enkripsi menggunakan XML *Encryption* yang menggunakan kriptografi algoritma RSA.

Implementasi terhadap rancangan arsitektur keamanan pesan SOAP akan disesuaikan dengan mekanisme *framework* NuSOAP dan menambahkan suatu *library* yang berisi beberapa fungsi yang dipergunakan dalam menunjang keamanan *web* *service* pada jalur transport. Selain itu untuk dapat mencapai tujuan dari keamanan tersebut akan dilakukan modifikasi terhadap rutin dari fungsi-fungsi didalam *class* *library* NuSOAP dan juga penambahan rutin program lainnya untuk keperluan keamanan *web* *service*.

Penambahan rutin program dan fungsi-fungsi keamanan dimaksudkan untuk pencapaian keamanan pesan. Dimana agar dapat melakukan hal-hal sebagai berikut:

1. Kemampuan untuk dapat mengamankan jalur transmisi data pada *web* *service* dengan menggunakan *security token* yang disertakan pada *Header* SOAP *request*, tujuannya adalah untuk otentikasi identitas user yang meminta layanan serta kendali akses untuk menentukan apakah user tersebut dilayani atau tidak.
2. Kemampuan untuk menjaga kerahasiaan serta keaslian data didalam pesan SOAP *request* dan SOAP *response*. Kemampuan ini ditunjang dengan penambahan beberapa *library* dari XMLSEC untuk keperluan enkripsi, dekripsi, serta *digital* *signature*  yang mana memanfaatkan agoritma kriptografi RSA dan RSAwithSHA-1 dengan panjang kunci 1024 bit.

Perancangan sistem yang akan dibuat diimplementasikan dalam sebuah perangkat lunak yang diharapkan memudahkan pemakainya (*user friendly*). Sebelum sistem di implementasikan, perlu diketahui aliran data dalam sistem. Aliran data di representasikan ke dalam bentuk diagram, yakni *Data flow diagram* (DFD)*.* DFDadalah sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output.

DFD digunakan untuk menyajikan sebuah sistem atau perangkat lunak pada setiap abstraksi dan dapat dibagi kedalam tingkat-tingkat yang merepresentasikan aliran informasi yang bertambah. DFD memberikan suatu mekanisme bagi pemodelan fungsional dan pemodelan aliran informasi. DFD Rancang Bangun *Library* Keamanan *Web Service* dapat dilihat pada Gambar 4. Pada Gambar 4, DFD level 1 proses menunjukkan adanya 2 proses yaitu proses *client service*, dan proses *server service*.



Gambar 4 DFD Level 1

*2.2 Implementasi Sitem*

## Setelah proses perancangan sistem dilakukan, tahap selanjutnya adalah membuat implementasi Rancang Bangun Library Keamanan Web Service. Persiapan tahap implementasi dari prototype keamanan web service ini menggunakan aplikasi open source yaitu Web Server Apache, Database MySQL, Scripting Language PHP, HTML, CSS, library dari xmlsec, dan NuSOAP yang akan bertindak menangani interaksi dan komunikasi antara client dan server side. Sedangkan untuk implementasi dari keamanan web service ini, maka dirancang arsitektur dan skenario dalam alur yang akan diterapkan. Arsitektur dan skenario dari keamanan web service ini dapat diperlihatkan pada Gambar 5.



Gambar 5 Arsitektur Skenario Keamanan Web Service

3. HASIL DAN PEMBAHASAN

Pengujian sistem merupakan elemen kritis dalam pengembangan sebuah perangkat lunak (*software*) karena akan merepresentasikan hasil akhir dari spesifikasi kebutuhan aplikasi, perancangan dan implementasi. Tujuan utama dari pengujian sistem adalah untuk memastikan bahwa hubungan antarmodul aplikasi telah memenuhi spesifikasi kebutuhan dan berjalan sesuai dengan skenario yang telah dideskripsikan sebelumnya.

* 1. *Pengujian Otentikasi*

Pada tahap pengujian otentikasi disini adalah bagaimana pengguna dapat menggunakan sistem seperti yang telah dirancang sebelumnya. Pada menu utama atau *index*, pengguna akan mendapati *form login* ke sistem sesuai dengan hak akses yang telah diberikan. Gambar 6 memperlihatkan bagaimana pengguna harus memasukkan *username* dan *password* yang telah diberikan ketika mendaftar.



Gambar 6 Otentikasi Login

 Sedangkan otentikasi antara *client* dengan *server* dinyatakan dengan menggunakan *security token* pada pesan SOAP *request*. Jika *username token* di *client service* sama dengan *username token* di *server service*, maka *client service* dapat diizinkan untuk mengakses layanan sesuai dengan nilai parameter yang telah disisipkan pada *Header*. *Username token* sendiri akan di enkripsi menggunakan algoritma SHA1, hasilnya seperti yang ditunjukkan pada Gambar 7.

#

Gambar 7 Hasil SOAP Request dengan Username Token

Selain itu otentikasi juga dapat dilakukan dengan cara mengecek keaslian pesan SOAP (verifikasi) yang dikirimkan berupa *digital* signature, hasil yang didapatkan yaitu valid dan tidak valid. Gambar 8 memperlihatkan tampilan dari proses otentikasi dengan cara pegecekan *username token* serta verifikasi keaslian data yang diterima di *server web service*. Hasil dari proses otentikas dan verifikasi ini akan ditulisakan pada sebuah file yaitu ” logverifikasi.txt”.



Gambar 8 Hasil Log Pengecekan Otentikasi Security Token dan Verifikasi Elemen Reference pada XML Signature

 Kemudian dengan menggunakan metode XML Signature yang merupakan metode untuk keaslian data, maka pada pesan SOAP request akan disisipkan Signature untuk memastikan bahwa data XML yang dikirimkan tidak berubah ketika proses pengiriman. Hal ini dapat dilihat pada Gambar 9.



Gambar 9 Hasil Penyisipan XML Signature Pada Pesan SOAP Request

* 1. *Pengujian Otorisasi Pengguna*

Jika tahap login terlewatkan maka pengguna dapat mengakses halaman berikutnya sesuai hak akses. Jika pada tahap otentikasi *user* maupun admin tidak dapat menginputkan *username* dan *password* secara benar, maka otentikasi login dinyatakan gagal dan harus mencoba ulang, artinya bahwa *user* tidak memiliki otorisasi untuk mengakses sumber daya yang ada di *database*.

* 1. *Pengujian Konfidensialitas*

Pada tahap pengujian konfidensialitas ini, *client service* akan mengenkripsi pesan SOAP yang akan dikirimkan yaitu pada data yang akan dikirim dengan memanggil fungsi yang enkripsi yang ada di *server* dan menggunakan kunci publik dari *client*, proses enkripsi menggunakan algoritma RSA dengan panjang kunci 1024 bit. Sedangkan proses dekripsi dilakukan pada *server service* dengan menggunakan kunci privat. Selanjutnya untuk melihat hasil pesan SOAP *request* ini yang berisi data terenkripsi dengan menggunakan metode XML Encryption dapat diperlihatkan pada Gambar 10.



Gambar 10 Hasil Pesan SOAP Request Dengan Model Keamanan Menggunakan XML Encryption

 Hasil yang diperoleh dari Gambar 10 diatas adalah seluruh data tersebut akan dienkripsi oleh *client service* untuk menjamin kerahasiaan data pada jalur transmisi ke *server* *web* *service*. Kemudian dapat dilihat bahwa ketika data dikirimkan, maka *client* akan memanggil fungsi keamanan yang ada di *client service* yaitu *library* class\_wss.php, selanjutnya ketika data dikirimkan dari *client service*, maka data SOAP akan disisipkan *username token* yang mana akan dicocokkan dengan *username token* miliknya *server service*, selain itu pesan SOAP akan di *digital* *signature* dan dienkripsi data. Hasil enkripsi dari data XML ini dapat dilihat dari elemen <*EncryptedData*> pada baris ke 2 dan </*EncryptedData*> pada baris ke 22. Kemudian pesan SOAP yang berisi data yang telah dienkripsi terlihat pada elemen <*CipherData*> pada baris ke 16 dan </*CipherData*> pada baris ke 21 . Elemen ini mengindikasikan bahwa data telah berhasil dienkripsi.

Berbeda jika pesan SOAP yang dikirimkan tidak dilakukan apa-apa, artinya bahwa data tidak dienkripsi sama sekali, maka data yang dihasilkan akan dapat terbaca struktur dari elemen SOAP *request* seperti yang ditunjukkan pada contoh pada Gambar 11.



Gambar 11 Hasil Pesan SOAP Request Tanpa Model Keamanan

 Sedangkan lain halnya jika pesan SOAP request yang dikirimkan oleh web service yang tanpa disisipkan mekanisme keamanan seperti pada penelitian ini, data yang dihasilkan murni sebagai data XML, dilihat dari struktur elemenya. Dari Gambar 10 dan Gambar 11 terlihat jelas bagaimana perbedaan dari pesan SOAP *request* yang diberikan mekanisme keamanan serta tanpa mekanisme keamanan pada elemen <SOAP-ENV:Body> , data XML pada pesan SOAP *request* yang tanpa keamanan pada elemen <SOAP-ENV:Body> pada baris ke 1 sampai ke 16 maka data yang terkandung didalamnya dapat terbaca, sehingga terdapat kemungkinan terjadi penyangkalan serta modifikasi data oleh pelaku kejahatan.

4. KESIMPULAN

Berdasarkan pembahasan yang telah diuraikan pada bagian-bagian sebelumnya, maka diambil beberapa kesimpulan sebagai berikut:

1. Desain dan implementasi modul yang telah dilakukan dengan menggunakan *library* keamanan serta dukungan *library* XMLSEC sebagai *library* pendukung dan *library* class\_wss yang dibangun mampu mengatasi masalah keamanan pada proses pengirimanyaitu keamanan otentikasi, otorisasi, dan konfidensialitas pesan SOAP *request* yang dihasilkan.
2. Hasil dari implementasi mengindikasikan bahwa otentikasi, otorisasi, serta konfidensialitas dapat terpecahkan dengan menerapkan konsep keamanan berbasis *library* keamanan yaitu dengan memanfaatkan XML *Signature*, XML *Encryption*, serta *Security* *Token.* Hasil pesan SOAP *request* pada proses pengiriman dapat memenuhi standar keamanan *web service,* dimana data ketika dikirimkan dalam keadaan terenkripsi dan tertandatangani dengan menggunakan *library* class\_wss yang telah dibangun.
3. Pengujian yang dilakukan pada *web* *service* dengan menerapkan model *library* class\_wss sebagai *library* keamanan *web service* yang dibangun memberikan hasil yang baik, yaitu pesan SOAP *request* pada saat dikirimkan dalam bentuk terenkripsi dan mampu didekripsi serta dapat tertandatangani dan diperiksa keasliannya, walaupun belum didukung dengan otentikasi berbasis sertifikat seperti X509 atau karberos.
4. *Library* class\_wss yang dibangun dengan memanfaatkan *library* XMLSEC untuk keamanan pada *web service* ini dapat diimplementasikan pada server dengan Sistem Operasi Windows maupun Linux, sehingga pengguna yang menginginkan sistem keamanan pada *web service* dapat dengan mudah menggunakannya.
5. Model WS-Security dengan menggunakan XML Signature, XML Encryption, serta Security Token yang memanfaatkan algoritma kriptografi RSA dengan panjang kunci 1024 bit mampu memberikan perlindungan terhadap transmisi data antara *client* dan *server web service*.

5. SARAN

Dengan segala keterbatasan kemampuan dan waktu yang tersedia serta melihat potensi perkembangan dan kebutuhan ke depan dalam sistem keamanan pada jaringan internet, baik *web service* maupun jaringan dengan teknologi lainnya maka dari hasil penelitian ini disarankan beberapa *future work* sebagai berikut :

1. Pertukaran kunci pada penelitian ini hanya sebatas *prototype* sehingga untuk kedepannya perlu pertukaran kunci yang lebih aman, misalnya kunci *server* disimpan pada suatu repositori *database* *server* tersendiri agar lebih terjaga keamanannya.
2. Model otentikasi pada penelitian ini hanya sebatas pada penggunaan *Security Token* dan *Digital Signature* tanpa menyisipkan sertifikat otorisasi sehingga kedepannya perlu diharapkan ditambahkan suatu sertifikat yang dapat menjamin otentikasi pesan pada pertukaran pesan antara *client* dan *server* *web* *service* seperti X509 *certificate* atau karberos.

# UCAPAN TERIMA KASIH

Alhamdulillah, maha besar Allah yang telah memberikan karunia bagi penulis sehingga dapat menyelesaikan penelitian ini. Terimakasih juga penulis ucapkan kepada semua pihak atas dukungan nya, terutama kepada Bapak Dr. Ahmad Ashari, M.Kom selaku pembimbing, untuk arahan, kesabaran, dan motivasi yang diberikan.

DAFTAR PUSTAKA

[1] Rakhim, R, T, 2010, Keamanan *Web* *Service* Menggunakan Token, Tesis S2 Magister Ilmu Komputer, Universitas Gadjah Mada.

[2] Zhang, W., 2009, Integrated *Security* Framework for Secure *Web* *Service*s, Research Institute of Applied Computer Technology, China Women’s University.

[3] Adriansyah, A, Arifandi,W, dan, Wicaksono, N , 2005 ,*Keamanan Web Service*, Teknik Informatika, Institut Teknologi Bandung.

[4] Fareghzadeh, N,(2009), *Web Service Security Method To SOA Development,* World Academy of Science, Engineering and Technology, No.49, 10 hal.

[5] Kenali, E., W., ,2010, Implementasi *Web* *Service* untuk Integrasi Data Satuan Reserse Kriminal (Studi Kasus Polda Lampung), Tesis S2 Magister Ilmu Komputer, Universitas Gadjah Mada.

[6] Suteja, B ,2004, Implementasi XML Signature untuk Secure XML Pada Kasus Integritas Transkrip Online, Tesis S2 Magister Ilmu Komputer, Universitas Gadjah Mada.

[7] Supriyanto,A., 2007, Otentikasi Dokumen XML menggunakan Algoritma RSA dan Hash SHA-1, Tesis S2 Magister Ilmu Komputer, Universitas Gadjah Mada.

[8] Hartono, B., 2003, Pemakaian kriptografi kunci publik dengan algoritma RSA untuk keamanan data XML, S2 Ilmu Komputer, Universitas Gadjah Mada.