PERANCANGAN DAN UJICOBA SISTEM KEAMANAN WEB SERVICE DENGAN METODE WS-SECURITY

Ari Muzakir
Dosen Universitas Bina Darma
Jalan Jenderal Ahmad Yani No.12 Palembang
Pos-el: ariemuzakir@gmail.com

Abstract: *Web service uses XML technology to exchange data in. 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. The implementation was done by using the security library will provide facilities in developinga. Web security service for the library support XMLSEC as library supporters and library class_wssthat have been builtable to overcome the problem of security on the transport path, especially forauthentication, authorization, and confidentiality request SOAP message. Model WS-Security usingXMLSignature, XMLEncryption, andSecurityTokenwhich utilizesthe cryptographic algorithm RSA with1024 bitkey lengt hto provide protection against transmission of data between client and server web service. The results obtain edare SOAP request messages encrypted and decryptedable to pitch well and signed and checked their authenticity***.*

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

Abstrak: *Web service menggunakan teknologi XML dalam melakukan pertukaran data. Bentuk pengamanan yang diterapkan pada web services adalah dengan penggunaan teknik kriptografi kunci-publik. 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, sertaSecurity Token yang memanfaatkan algoritma kriptografi RSA dengan panjang kunci 1024 bit mampu memberikan perlindungan terhadap transmisi data antara client dan server web service.Hasil yang diperoleh yaitu pesan SOAP request terenkripsi dan mampu didekripsi dengan baik serta dapat ter tandatangani dan dicek keotentikannya.*

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

1. PENDAHULUAN

Saat 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. Dimana *web service* mendukung komunikasi antar aplikasi dan integrasi aplikasi dengan menggunakan XML dan *Web*. Faktor keamanan pada jalur komunikasi antara *client* ke *serverweb 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 (Rakhim, 2010). *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 *webservice* dalam kemampuannya untuk meneruskan pesan ke beberapa *webservice* lain secara bersamaan. Pengolahan model *webservice* 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 SSL/TLS serta *message-level security* seperti yang telah disediakan WS-*Security*(Zhang, 2009). WS-*Security* juga mengatur cara menyisipkan *security token* dalam pesan SOAP dalam bentuk *plaintext* maupun dalam bentuk biner, seperti sertifikat X.509 (Adriansyah Dkk, 2005). 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 *serviceclient* menggunakan keamanan berbasis SSL (*Security Socket Layer*) atau disebut *protocol* HTTPS.

Beberapa penelitian yang telah dilakukan berkenaan dengan keamanan *webservice* 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 CORBA dan RMI (Adriansyah Dkk, 2005).

Selanjutnya, analisa mengenai bagaimana mengatasi tantangan pada keamanan *webservice* 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 (Zhang,

2009). Penelitian mengenai penyajian suatu metode yang komprehensif untuk suatu jaminan layanan keamanan dalam SOA. dimana metode yang diusulkan mendefinisikan tiga tahap yaitu *securityanalysis*, arsitektur jaminan keamanan, dan identifikasi Standar WS-*Security* (Fareghzadeh, 2009).

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 (Kenali, 2010). 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 (Rakhim, 2010). 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 (Suteja, 2004).

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 (Supriyanto, 2007). 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 (Hartono, 2003).

2. METODOLOGI PENELITIAN

2.1 Analisis Sistem

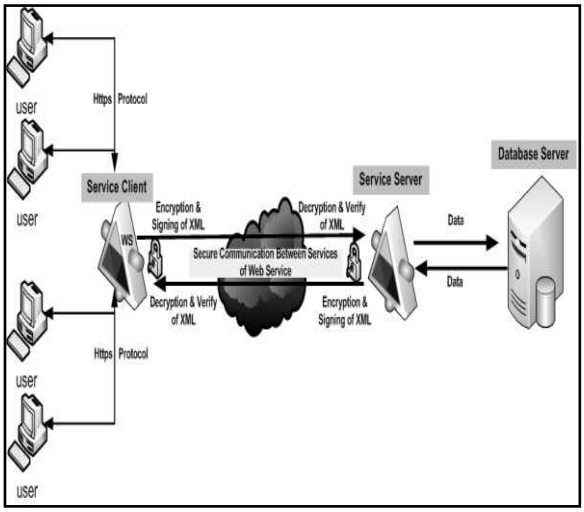
Pada penelitian ini, model keamanan *web service* menggunakan metode WS-*Security* memanfaatkan dua teknologi yang sudah ada yaitu XML *Signature* dan XML *Encryption* sebagai model pengamanan dokumen XML yang memanfaatkan kriptografi kunci publik RSA yang akan digunakan pada metode keamanan *client server web service*. Proses enkripsi dan dekripsi pada kriptografi ini terapkan pada *client service* dan *server service* untuk pengamanan jalur komunikasi yang mana menggunakan dua buah kunci yaitu kunci publik dan kunci rahasia. Metode dari *prototype* keamanan *web service* antara *client service* dan *server service* ini dimulai dari pengiriman data dari user menggunakan SLL ke *client service*, kemudian komunikasi antara *client service* dengan *server service* dari *web service*. Di mana 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 *databaseserver*.

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.

2.2 Perancangan Sistem

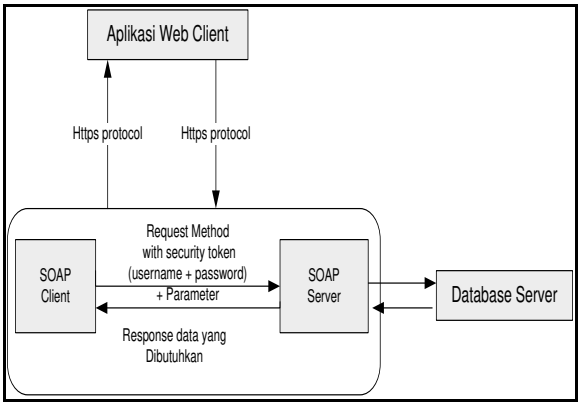
Sistem aplikasi yang akan dibangun memiliki arsitektur keamanan secara umum seperti pada gambar 1, di mana setiap *request* dari *client* akan dilakukan otentikasi, otorisasi, dan konfidensialitas. Otentikasi dilakukan ketika *client* berhasil melakukan *login* dan akan diberikan akses ke sumber daya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



Gambar1. Keamanan Antara *Client Service* dan *ServiceServer* dari *Web Service*

Gambar 1 memperlihatkan alur keamanan *web service* antara *client service* dan *server service*, dimana gambaran umum dari keamanan sistem ini dimulai dari pengiriman data dari user menggunakan SLL 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 *databaseserver*.

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. Mekanisme Otentikasi *User* Pada *Web Service*

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 *clientservice*dan *serverservice* di mana bertujuan untuk mengamanakan jalur transmisi pada *webservice* sendiri.

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 *webservice* pada jalur transport. Selain itu untuk dapat mencapai tujuan dari keamanan tersebut akan dilakukan modifikasi terhadap rutin dari fungsi-fungsi didalam *classlibrary* NuSOAP dan juga penambahan rutin program lainnya untuk keperluan keamanan *webservice*.

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 *webservice* 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 *digitalsignature* yang mana memanfaatkan agoritma kriptografi RSA dan RSAwithSHA-1 dengan panjang kunci 1024 bit.

Sedangkan perancangan sistem mengacu pada tiga tahapan pengembangan sistem dengan model prototipe, tahapan tersebut yaitu:

2.2.1 Mendefinisikan Tujuan dan Mengidentifikasi Kebutuhan Pemakai

Untuk mendefinisikan tujuan dan indentifikasi kebutuhan pemakai sendiri, dilihat dari alur kerja dalam implementasi sistem pada *web service* ini dibagi menjadi dua bagian, yaitu berdasarkan pengimplementasian di *client* dan di *server*.

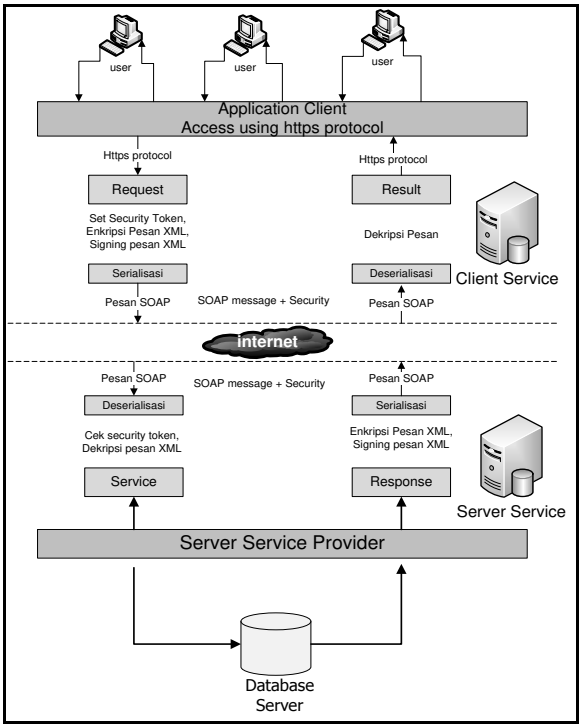
Tahapan pertama yaitu *Client* menghasilkan *web servicerequest* yang kemudian akan diterima oleh *client service* sebelum dilanjutkan ke *server service*. Tahap ini berkaitan dengan proses-proses yang dilakukan oleh *client* untuk melakukan *request* kepada *web service* dengan menggunakan *username token*. Terdapat beberapa langkah yang dilakukan oleh *client* yaitu: 1) 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 *webservice* 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*. 2) 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*.

Tahap kedua yaitu *server service* akan mengotentikasi *client service* dan mengembalikan respon ke *client*. Tahap ini menjelaskan beberapa proses yang dilakukan oleh *serverweb service* setelah menerima SOAP

Request dari *client service*. Beberapa proses tersebut adalah: 1) 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. 2) 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. 3) 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. 4) 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. 5) 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 *webservice* 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.

2.2.2 Melakukan Perancangan Secara Cepat untuk Membuat Prototipe

Alur perancangan yang dibuat dalam penelitian ini adalah dengan cara memberikan gambaran alur kerja sistem keamanan web service sendiri. Pada gambar 3 berikut memberikan gambaran mengenai mekanisme keamanan yang dimulai dari user mengirimkan data sampai user menerima data.



Gambar 3. Alur Mekanisme Keamanan Data *User* Pada *WebService*

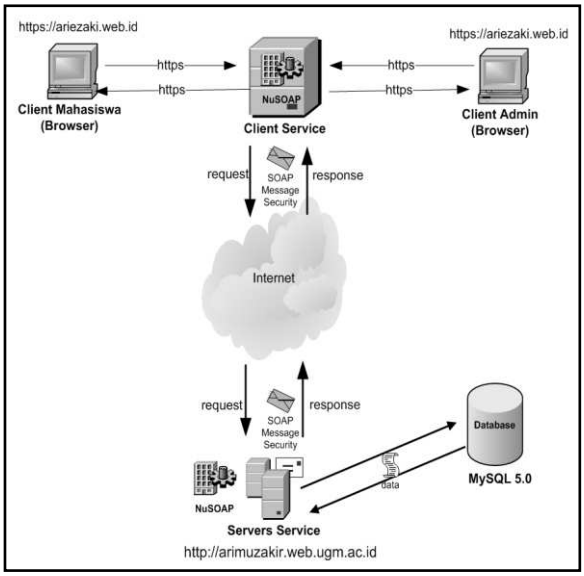
Pada penelitian ini rancangan keamanan tersebut dititikberatkan pada beberapa mekanisme keamanan, yaitu: 1) Otentikasi, yaitu sebuah mekanisme keamanan yang bertujuan untuk membuktikan atau verifikasi otentikasi *user*/mesin (*client*) yang meminta layanan, sehingga dengan demikian *service provider*

(*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.

2.2.3 Menguji Coba dan Mengevaluasi Prototipe

Setelah proses perancangan sistem dilakukan, tahap selanjutnya adalah membuat implementasi dan menguji sistem. Tujuannya adalah untuk mematikan bahawa seluruh komponen yang dibangun dapat bekerja sesuai rencana.

Sedangkan untuk implementasi dari keamanan *webservice* ini, maka dirancang arsitektur dan skenario dalam alur yang akan diterapkan. Arsitektur dan skenario dari keamanan *webservice* ini dapat diperlihatkan pada gambar 4.



Gambar 4. Alur Uji Coba Keamanan *WebService*

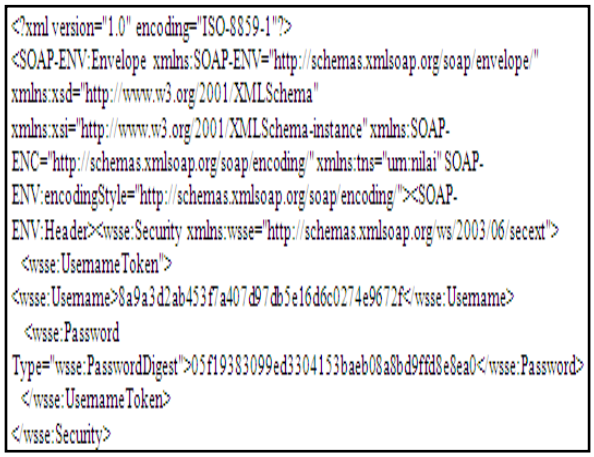
3. HASIL DAN PEMBAHASAN

Hasil akhir dari penelitian ini adalah ujicoba terhadap setiap modul yang dibangun dari sistem keamanan *web service* ini. Kemudian selain itu juga untuk memastikan bahwa hubungan antarmodul aplikasi telah memenuhi spesifikasi kebutuhan dan berjalan sesuai dengan skenario yang telah dideskripsikan pada gambar 4.

3.1 Otentikasi Sistem

Pada tahap otentikasi ini adalah bagaimana pengguna dapat menggunakan sistem seperti yang telah dirancang sebelumnya. 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

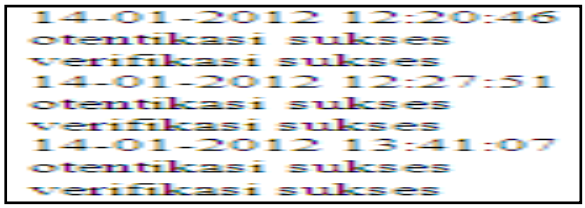
dengan nilai parameter yang telah disisipkan pada *Header*. *Username token* sendiri akan di enkripsi menggunakan algoritma SHA1, hasilnya seperti yang ditunjukkan gambar 5.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:nilai" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Header><wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
  <wsse:UsernameToken">
<wsse:Username>8a9a3d2ab453f7a407d97db5e16d6c0274e9672f</wsse:Username>
  <wsse:Password
Type="wsse:PasswordDigest">05f19383099ed3304153baeb08a8bd9ffd8e8ea0</wsse:Password>
  </wsse:UsernameToken>
</wsse:Security>
```


Gambar 5. 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 yangdidapatkan yaitu valid dan tidak valid. Gambar 6 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”.



```
14-01-2012 12:20:46
otentikasi sukses
verifikasi sukses
14-01-2012 12:27:51
otentikasi sukses
verifikasi sukses
14-01-2012 13:41:07
otentikasi sukses
verifikasi sukses
```


Gambar 6. 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 7.

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo><ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315">
        </ds:CanonicalizationMethod><ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1">
                </ds:SignatureMethod>
<ds:Reference URI=""><ds:Transforms>    <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"></ds:Transform>
        </ds:Transforms><ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
                </ds:DigestMethod>
        <ds:DigestValue>VFYsffQoQ9Q2U6ZaBUcouKyVGx4=</ds:DigestValue>
                </ds:Reference></ds:SignedInfo>
        <ds:SignatureValue>qefTd2Ysv389G8XddHbfgT8ZZolcQGQuOOPJzpHkdiNSd
VIYrh/yo4GwzYRDTkzGAnO+dx7GjIHBSXlGZrj4aFRHcEyO2T0T3o9TZ6hh8e/HNWoxm
/nK0moeBt2rCgQkGIo+VvKSWKWjh1hqnFJ0C+x4LT8/OPKlxoXw6LR8dWg=</ds:Signat
ureValue>           <ds:KeyInfo>
<ds:KeyValue>
<ds:RSAKeyValue><ds:Modulus>
vitPjejTFTjXtDJORSIB0t77MYjiDX+rRuZ9oTI0RpDI2OCxYgtf8dT0YIAWqN1w4psogk4u
2/77PCsol3PySYwuPuwDrG7VIYlZ/UfPiX3Spg+fQq0d6O8P4OQGljf/0XI7zd6Nf5+EWDJ
OEjzxLhAt3GJt0ZMmrJLxgno1/e0=
</ds:Modulus><ds:Exponent>AQAB</ds:Exponent>
</ds:RSAKeyValue>
</ds:KeyValue></ds:KeyInfo></ds:Signature>
```


Gambar 7. Hasil Penyisipan XML Signature pada Pesan SOAP Request

3.2 Otorisasi Pengguna Sistem

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*.

3.3 Konfidensialitas Sistem

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 8 berikut.

```
<SOAP-ENV:Body>
<EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#Element">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-
cbc"/>
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
                <EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>        <KeyInfo
xmlns="http://www.w3.org/2000/09/xmldsig#">
                <KeyName/>        </KeyInfo>
                <CipherData>
        <CipherValue>ill9Dz2IXlxJuQ9Nrkdqup/4I/npZeywQIfivHO4MTEmXRYIBOVi
SpNROeKKcBNRLthH8ROQsdpTiN/7kO+ppfCBqjX+j9mGEHJOS+U0Tp9KqxFeN4YR3bJ
W4Lt/VOpxvVy+3TnGTv4cZxB0emNcR3H2BUYtjMen65aIfIP5wa8=</CipherValue>
                </CipherData>
                </EncryptedKey>
        </KeyInfo>        <CipherData>
        <CipherValue>5BzJItoZ9gMD8pmjNbGg2ynQZR9jYVtmz3YQVHaE6jyMECV1
MXlFNKya13EoO7nCojzq3z03mSXXHC2CFrQkBW3R6pDFHTJqzc8i6he1VLOpcfz+J29A
HlswlndVR50RGW2emplnvGawsAD4zlVO15Uy9n010ZLDYL9sOB/MHe+Pk/hwTn0bu1Cel
7tk5kiBqH1865H386mmVemKJaISjgHulhqZtX</CipherValue>
        </CipherData>
</EncryptedData>
</SOAP-ENV:Body>
```


Gambar 8. Hasil Pesan SOAP Request Dengan Model Keamanan Menggunakan XML Encryption

Hasil yang diperoleh dari gambar 8 di atas adalah seluruh data tersebut akan dienkripsi oleh *client service* untuk menjamin kerahasiaan data pada jalur transmisi ke *serverwebservice*. 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 *digitalsignature* dan dienkripsi data. Hasil enkripsi dari data XML ini dapat dilihat dari elemen <*EncryptedData*> dan </*EncryptedData*>. Kemudian pesan SOAP yang berisi data yang telah dienkripsi terlihat pada elemen <*CipherData*> dan </*CipherData*>. Elemen ini mengindikasikan bahwa data telah berhasil dienkripsi.

4. SIMPULAN

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

1) 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 *SecurityToken.*
2) Hasil yang dilakukan pada *webservice* dengan menerapkan model *library* class_wss sebagai *library* keamanan *web service* yang dibangun memberikan respon yang baik, yaitu pesan SOAP *request* pada saat dikirimkan dalam bentuk terenkripsi dan mampu didekripsi serta dapat tertandatangani dan diperiksa keasliannya.
3) 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*.

DAFTAR RUJUKAN

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

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

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

Kenali, E., W. 2010. *Implementasi WebService untuk Integrasi Data Satuan Reserse Kriminal (Studi Kasus Polda Lampung*. Tesis S2 Magister Ilmu Komputer, Universitas Gadjah Mada. Yogyakarta.

Rakhim, R, T. 2010. *Keamanan WebService Menggunakan Token*. Tesis S2 Magister Ilmu Komputer, Universitas Gadjah Mada. Yogyakarta.

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

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

Zhang, W. 2009. *Integrated Security Framework for Secure WebServices*. Research Institute of Applied Computer Technology, China Women's University.