PERANCANGAN DAN UJI COBA KEAMANAN PADA JALUR TRANSPORT WEB SERVICE MENGGUNAKAN METODE XML SIGNATURE DAN XML ENCRYPTION

[1]Ari Muzakir

[1]*Program Studi Teknik Informatika, Universitas Bina Darma, Jl. A. Yani No. 12 Palembang 30264*
e-mail: ariemuzakir@gmail.com

Abstrak. *Web service menggunakan teknologi XML dalam melakukan pertukaran data. Umumnya penggunaan web service terjadi pertukaran data ataupun informasi penting yang perlu dijaga keamanannya. Pengamanan tersebut dapat dilakukan dengan menggunakan salah satu teknik kriptografi yaitu enkripsi dan dekripsi. Teknik kriptografi yang digunakan adalah kriptografi asymmetric RSA yang diimplementasikan pada jalur pengiriman. Implementasi yang telah dilakukan dengan kemudian 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 dan SOAP response. Dengan menggunakan metode XML Signature dan XML Encryption 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 ter tandatangani dan dicek keotentikannya.*

Kata kunci: keamanan web service, XML Signature, XML Encryption

1. Pendahuluan

Perkembangan teknologi web saat ini semakin pesat dengan bermunculannya teknologi informasi yang semakin modern dan kebutuhan akan data yang cepat dan akurat melalui jaringan internet. Salah satu hal yang sedang fenomena saat ini adalah *web service*, dimana banyak kelebihan yang ditawarkan terutaman interoperabilitas yang tinggi serta mendukung komunikasi antar aplikasi dan integrasi aplikasi dengan menggunakan *eXtensible Markup Language* (XML). Namun, faktor keamanan pada *web service* terutama pada jalur komunikasi antara *client* dan *server web service* belum sepenuhnya terjamin. Hal ini dibuktikan dari hasil penelitian-penelitian terdahulu yang menyebutkan adanya celah-celah ancaman terhadap data XML pada jalur *transport* antara komunikasi *client* dan *server 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 *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 SSL/TLS serta *message-level security* seperti yang telah disediakan WS-*Security* (Zhang, 2009).

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 (Zhang, 2009). 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* (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).

2. Metodologi Penelitian

2.1 Analisa Sistem

Sistem yang akan dibangun merupakan model *prototype* keamanan *web service*, dimana akan diujicobakan pada data nilai mahasiswa. Model keamanan transport pada penelitian ini memanfaatkan XML *Signature* (XMLSig) dan XML *Encryption* (XMLEnc) pada model pengamanan dokumen XML dengan teknologi kriptografi kunci asimetris atau kunci public RSA dengan panjang kunci 1024 bit untuk proses enkripsi dan dekripsi.

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 *Secure Socket Layer* (SLL) 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:

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*.

a) *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.*
b) *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.*

2) Analisa kebutuhan non-fungsional

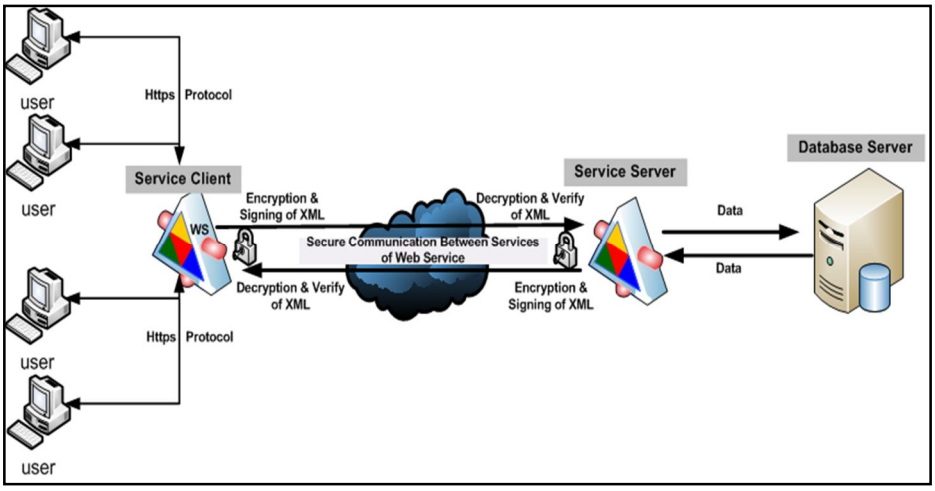
Analisis kebutuhan non-fungsional dari penelitian ini adalah sebagai berikut:

a) Faktor waktu respon dari site internal maupun eksternal melalui *web service* yang tidak bisa diprediksi.
b) 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 memperlihatkan model dari keamanan *web service* antara *client service* dan *server service*, dimana gambaran umum dari keamanan sistem ini dimulai

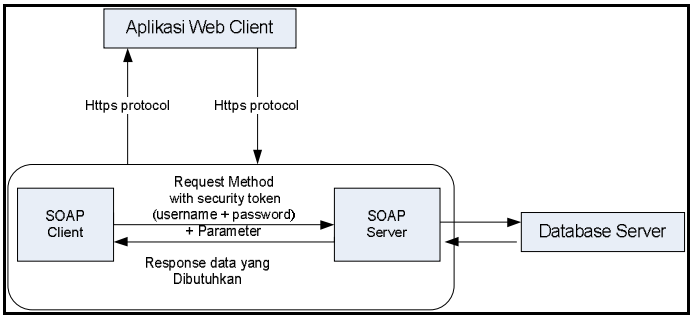


***Gambar 1**. Model Keamanan Transport Client dan Server dari Web Service*

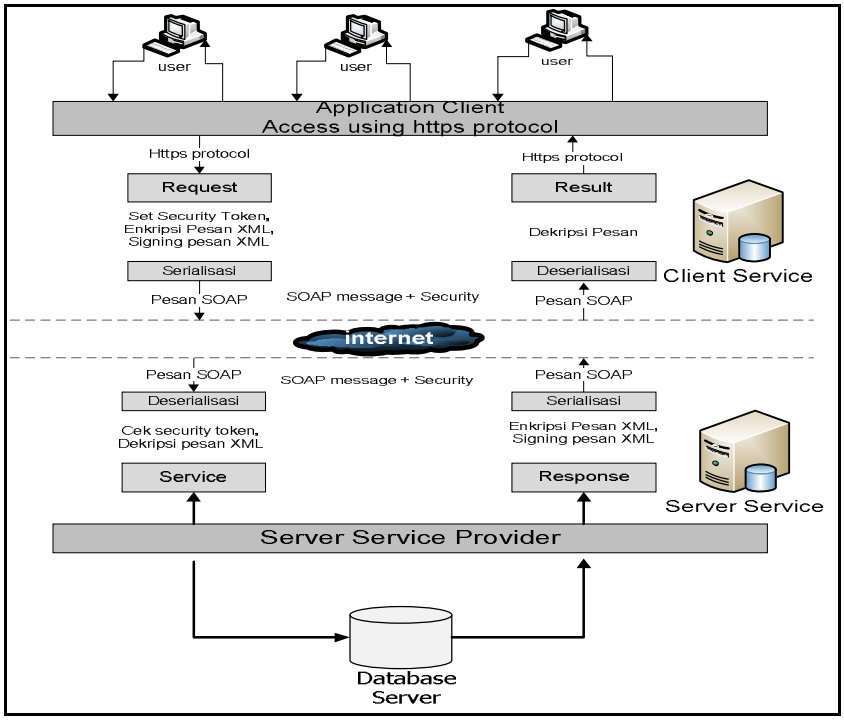
dari pengiriman data dari user menggunakan SLL ke *client service*, kemudian komunikasi antara *client service* dengan *server service* dari *web service*. Selanjutnya data XML akan dienkripsi (*encrypt*) dan ditandatangani (*signing*) serta menyertakan *username token* dari *client service* dan akan didekripsi (*decrypt*), di verifikasi serta di cek *username token* ketika diterima oleh *server service*, selanjutnya data hasil dekripsi akan disimpan pada *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.

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.



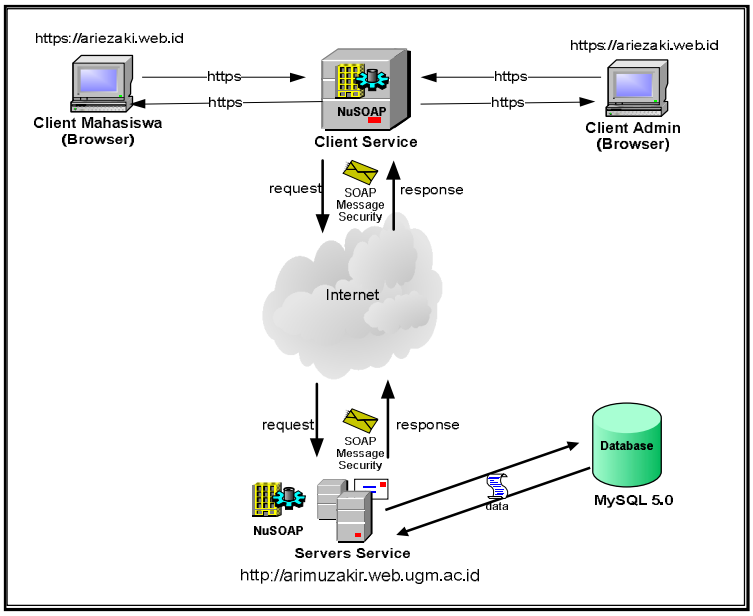
Gambar 2. *Rancangan Mekanisme Otentikasi User Pada Web Service*



Gambar 3. *Rancangan Mekanisme Kerahasiaan Data User Pada Web Service*

2.3 Implementasi Sistem

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, *Script*ing 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. Data 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.

3.1 Pengujian Otentikasi

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 6.

```
<?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="um: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 6. *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 7 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”.

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 8.

```
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 7. *Hasil Log Pengecekan Otentikasi Security Token dan Verifikasi Elemen Reference pada XML Signature*

```
<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 8. *Hasil Penyisipan XML Signature Pada Pesan SOAP Request*

3.2 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 9 berikut.

```
1     <SOAP-ENV:Body>
2     <EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#Element">
5     <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
6         <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
7         <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
8         <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
9                 <KeyName/>          </KeyInfo>
10                <CipherData>
11        <CipherValue>ill9Dz2IXlxJuQ9Nrkdqup/4I/npZeywQIfivHO4MTEmXRY
IBOViSpNROeKKcBNRLthH8ROQsdpTiN/7kO+ppfCBqjX+j9mGEHJOS+U0Tp9Kqx
FeN4YR3bJW4Lt/VOpxvVy+3TnGTv4cZxB0emNcR3H2BUYtjMen65aIflP5wa8=
14  </CipherValue>  </CipherData>
15  </EncryptedKey>  </KeyInfo>
16        <CipherData>
17        <CipherValue>5BzJItoZ9gMD8pmjNbGg2ynQZR9jYVtmz3YQVHaE6jyM
ECV1MXlFNKya13EoO7nCojzq3z03mSXXHC2CFrQkBW3R6pDFHTJqzc8i6he1VLO
pcfz+J29AHlswlndVR50RGW2emplnvGawsAD4zlVO15Uy9n010ZLDYL9sOB/MHe+Pk
/hwTn0bu1Cel7tk5kiBqH1865H386mmVerrKJaISjgHulhqZtX</CipherValue>
21        </CipherData>
22  </EncryptedData>
23  </SOAP-ENV:Body>
```

Gambar 9. *Hasil Pesan SOAP Request Dengan Model Keamanan Menggunakan XML Encryption*

Hasil yang diperoleh dari Gambar 9 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.

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

pengiriman yaitu 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* dan XML *Encryption.* 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. Daftar Pustaka

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

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

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

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.

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

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

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