XML *Key Management Specification* (XKMS) untuk Keamanan *Web Services*

Roni Sambiangga [NIM 13502025]

Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha No. 10, Bandung 40132
INDONESIA

Email: if12025@students.if.itb.ac.id

ABSTRAK

Penggunaan layanan *web service* semakin banyak dilakukan untuk proses pertukaran data ataupun informasi melalui jaringan internet. *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. Untuk pengelolaan kunci-publik yang digunakan pada web service diterapkan dalam penggunaan infrastruktur yang disebut *public-key infrastructure* (PKI). XML *Key Management Specification* (XKMS) merupakan bentuk pengembangan PKI yang berada pada lingkungan berbasis XML. Penerapan XKMS ditujukan untuk meningkatkan proses pengamanan *web services*.

Kata kunci: XML *Key Management Specification*, XMKS, keamanan *web services*.

PENDAHULUAN

Kebutuhan akan informasi telah menjadi tuntutan dasar dalam pengembangan teknologi informasi. Adanya teknologi internet menjadi salah satu hasil pengembangan teknologi informasi dalam menjawab tantangan/tuntutan tersebut. Penggunaan dan pengembangan teknologi informasi terkait dengan teknologi internet telah melahirkan *web services* pada jaringan internet. Pada komunitas bisnis dan para pengembang teknologi terdapat beragam jenis layanan, sistem ataupun komunikasi data yang dibutuhkan. Kehadiran *web services* diharapkan dapat menjembatani keanekaragaman teknologi tersebut. Yang menjadi dasar pemikiran munculnya *web services* adalah tersedianya suatu sistem, yang berada dalam suatu perusahaan ataupun berada pada perusahaan berbeda, yang dapat saling berkomunikasi satu dengan lainnya. Kebutuhan adanya bentuk komunikasi antarsistem merupakan hasil dari suatu "hubungan" antara perusahaan dengan para pelanggan, dan juga dengan rekan bisnis. *Web services* memiliki teknologi yang memungkinkan komunikasi (hubungan) tersebut dapat berjalan dengan lancar dan efektif.

Pada penerapan web services dibutuhkan teknologi-teknologi pendukung yang terdiri dari *eXtensible Markup Language* (XML), *Web Service Definition Language* (WSDL), *Simple Object Access Protocol* (SOAP), dan *Universal Description, Discovery, and Integration* (UDDI) – setiap teknologi pendukung tersebut berdasar pada standar yang terbuka dan umum digunakan.

- **XML**

  XML (*eXtensible Markup Language*) merupakan teknologi paling mendasar (inti) yang digunakan pada *web*

services. XML merupakan pengembangan dari HTML – untuk mendukung pembuatan, pengembangan, ataupun pengaturan data yang bersifat dinamis. Pada penggunaan XML dapat dilakukan pembuatan "elemen-elemen" dokumen yang disesuaiakan dengan kebutuhan; berbeda dengan HTML yang memiliki elemen (*tag*) yang sangat terbatas. Dengan adanya sifat XML yang fleksibel tersebut, maka dapat dibentuk suatu skema dokumen yang memiliki tipe data, struktur, isi, dan elemen-elemen lain yang dibutuhkan.

- **WSDL**

  Dalam *web services* akan terdapat komunikasi antarsistem yang berbeda-beda sehingga dibutuhkan suatu "bahasa" yang dapat digunakan dan dipahami oleh semua sistem. Untuk dapat melakukan komunikasi tersebut digunakan WSDL (*Web Service Definition Language*). Pada spesifikasi WSDL versi 1.1[4] dijelaskan bahwa WSDL adalah "*an XML grammar for describing network services as collections of communication end points capable of exchanging messages*". WSDL digunakan untuk menjelaskan/ mendeskripsikan suatu *web service* dan untuk meningkatkan kemampuan operasional antarsistem ataupun antaroperasi tanpa memperhatikan protokol jaringan ataupun skema yang digunakan oleh aplikasi-aplikasi atau sistem-sistem tersebut.

  Pada dasarnya dokumen WSDL menjelaskan cara penggunaan suatu *web service* dan penyediaan informasi yang dibutuhkan untuk saling dipertukarkan, pengiriman pesan, penyatuan protokol, dan lokasi layanan (*service*) yang dibutuhkan. Dalam dokumen WSDL terdapat tiga komponen utama, yaitu (a) *data type definitions*, yang berisi penjelasan mengenai struktur dan isi pesan; (b) *abstract operations*, yang menjelaskan operasi-operasi apa saja yang perlu dilakukan; (c) *service bindings*, berisi penjelasan mengenai jaringan transportasi yang akan digunakan untuk proses pengiriman pesan.

- **SOAP**

  Pada SOAP (*Simple Object Access Protocol*) terdapat kumpulan teknologi berbasis XML yang menjelaskan suatu *framework* untuk melakukan pertukaran pesan; protokol umum pengiriman data untuk komunikasi jaringan melalui internet. Dengan menggunakan SOAP, *client* ataupun aplikasi melakukan panggilan (*request*) kepada *server*. Panggilan (*request*) dan jawaban (*response*) yang dikirimkan menggunakan format dokumen XML dan pihak pengirim (*client*) haruslah terlebih dahulu mengetahui lokasi penerima (*server*) agar pesan (*request*) dapat dikirimkan – hal inilah yang mendasari UDDI.

- **UDDI**

  UDDI (*Universal Description, Discovery, and Integration*) merupakan bagian dari *web services* yang digunakan untuk menentukan lokasi *web services* tersebut dan informasi lainnya terkait dengan layanan yang tersedia. Dengan menggunakan UDDI suatu *web service* dapat mempublikasikan ataupun mencari layanan-layanan yang tersedia.

Dengan adanya kemampuan dalam melakukan komunikasi dan integrasi antarsistem yang berbeda, penggunaan teknologi *middleware* yang umum beserta kelebihan lain yang diberikan, *web services* mulai banyak diterapkan untuk mendukung perusahaan-perusahaan yang memanfaatkan teknologi informasi dalam menjalankan proses bisnisnya. Namun, di sisi lain perlu diperhatikan juga aspek-aspek yang terkait dengan penerapan teknologi web services, salah satunya adalah mengenai keamanan. Perlu diperhatikan kembali bahwa *web services* merupakan suatu sistem tersebar dan terhubung dalam jaringan (internet) yang luas dan digunakan untuk melakukan proses pertukaran data ataupun informasi.

KEAMANAN WEB SERVICE

Aspek keamanan merupakan aspek yang perlu dipertimbangkan secara khusus dalam penerapan teknologi *web services*. Pada penggunaan *web services* terjadi proses pertukaran (transaksi) komunikasi data-data ataupun informasi yang penting – perlu diperhatikan bahwa data ataupun informasi suatu perusahaan akan terhubung dengan jaringan yang cukup luas. Dengan demikian aspek keamanan menjadi sangat penting untuk menjaga data ataupun informasi agar tidak disalahgunakan ataupun diakses secara sembarangan. Berikut ini merupakan isu-isu utama yang perlu diperhatikan terkait dengan keamanan *web services*.

Authentication (keabsahan). Pada penggunaan layanan yang disediakan pada web service, setiap pihak (*client* dan *server*) yang terkait di dalamnya haruslah dapat dijamin keabsahan para pengguna layanan tersebut.

Authorization (otorisasi). Untuk setiap penggunaan layanan pada web services haruslah terdapat suatu proses pemberian kuasa/hak (otorisasi) terhadap layanan yang akan digunakan. Dengan adanya otorisasi dapat ditetapkan siapa saja yang dapat menggunakan layanan yang tersedia, apa saja layanan yang dapat diakses dan sejauh mana layanan tersebut yang dapat digunakan.

Confidentiality (kerahasiaan). Dikarenakan komunikasi/transaksi yang terjadi dilakukan pada jaringan yang umum maka kerahasiaan data ataupun informasi merupakan salah satu kebutuhan utama pada *web services* sehingga dibutuhkan suatu jaminan bahwa data atau informasi menggunakan jalur yang aman.

Integrity (keutuhan). Dalam hal ini perlu dijamin bahwa pada penggunaan layanan *web services* data atau informasi yang dikirimkan oleh *client* atau *server* sama dengan data atau informasi yang diterima oleh *server* atau *client*.

Nonrepudiation (anti-penyangkalan). Dalam penggunaan layanan yang terdapat pada *web services*, diperlukan mekanisme yang dapat membuktikan penggunaan suatu layanan tertentu – pembuktian dapat dilakukan di kedua sisi: *client* (pengguna layanan) ataupun *server* (penyedia layanan).

Availability (ketersediaan). Pada *web services*, haruslah dapat dijamin ketersediaan layanan yang dapat digunakan; jaminan ketersediaan layanan juga ditujukan untuk menghindari adanya serangan *denial-of-services* (DoS).

End-to-end security. Dalam suatu jaringan komunikasi data ataupun informasi dibutuhkan pengamanan pada media/ saluran perantara pesan yang bersifat *end-to-end*. Diperlukan adanya mekanisme yang menjamin keamanan dan keutuhan data ataupun informasi yang dikirim atau diterima melalui saluran perantara karena data ataupun informasi tersebut dapat saja hilang ataupun rusak.

Selain isu-isu yang telah disebutkan di atas, terdapat hal-hal lain yang juga perlu diperhatikan. Hal-hal tersebut terkait dengan *web services* sebagai suatu sistem komputerisasi yang tersebar. Seiring dengan terus berkembangnya dan muncul penemuan baru dalam bidang teknologi informasi, terdapat hal-hal yang perlu diperhatikan untuk penerapan *web services*, yaitu:

- Banyaknya jumlah dan beragamnya spesifikasi standar yang digunakan antarsistem.
- Penemuan-penemuan spesifikasi standar keamanan yang baru.
- Bentuk-bentuk interaksi ataupun antarmuka data dan proses bisnis suatu perusahaan.
- Adanya standardisasi dokumen XML untuk pengamanan data.
- Jenis-jenis keamanan komunikasi yang digunakan (antaraplikasi, *end-to-end*, *just-one-context*).
- Kemampuan lintas-operasi pada jaringan.
- Kemungkinan dibutuhkannya suatu proses otomasi pengelolaan yang dilakukan antarmesin – tidak lagi harus dilakukan ataupun harus selalu diawasi oleh manusia.
- Ketersediaan pengelolaan proses bisnis yang dilakukan secara *online*.

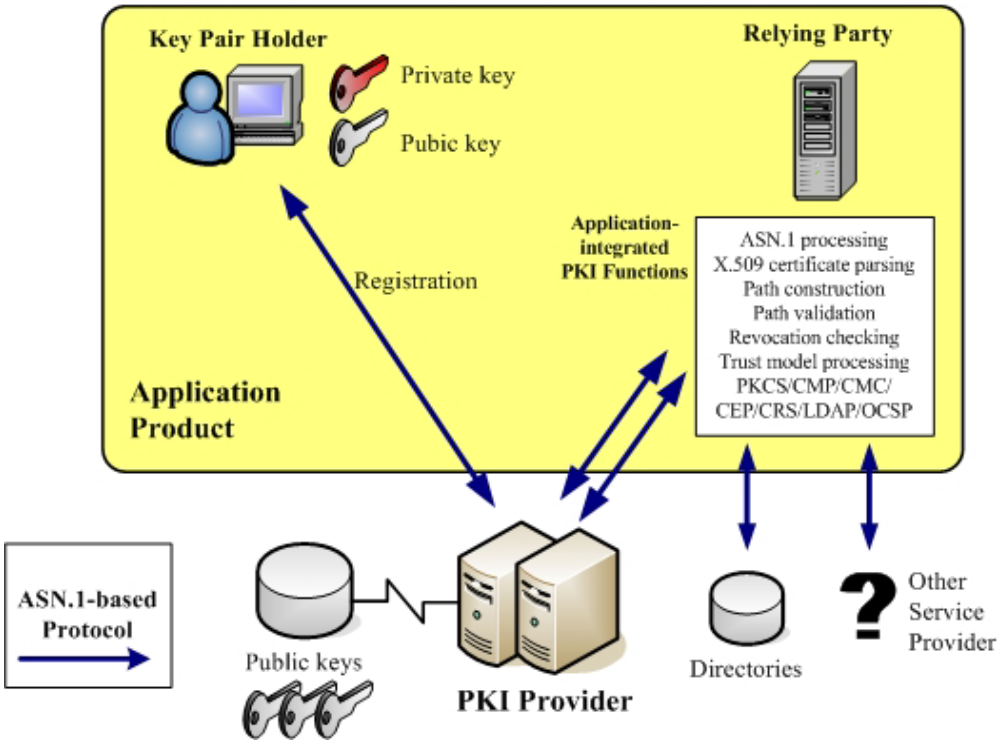
Dengan demikian dapat dipahami bahwa masalah keamanan merupakan hal yang sangat penting dalam penerapan dan penggunaan aplikasi yang berbasis *web* (internet). Oleh karenanya, dibutuhkan suatu standar yang disepakati dan digunakan secara umum. Untuk pengelolaan dan pembuatan

standar tersebut adalah W3C (*World Wide Wed Consortium*) dan OASIS (*Organization for the Advancement of Structured Information Standards*).

Saat ini, bentuk pengamanan yang digunakan pada *web services* adalah dengan penggunaan sistem kriptografi kunci-publik (*public-key*). Untuk menjamin keamanan kunci-publik tersebut dilakukan juga proses pemberian sertifikat digital terhadap kunci-publik tersebut. Sertifikat digital digunakan untuk mengatasi masalah pengaman kunci-publik yaitu serangan dengan bentuk penyamaran kepemilikan kunci-publik (*impersonation attack*). Sertifikat digital tersebut hanya dapat dikeluarkan/diterbitkan (*issued*) oleh pihak yang merupakan pemegang otoritas sertifikasi dan disebut sebagai *Certification Authority* (CA) – VeriSign, Inc. merupakan *certification authority* yang terkemuka saat ini. Dalam dokumen sertifikat digital berisi informasi mengenai nama subjek pemilik sertifikat, kunci-publik subjek, waktu kadaluarsa sertifikat, nomor seri sertifikat, dan lain-lain[5]. Untuk melakukan pengelolaan kunci-publik dan sertifikat digital dibentuklah suatu sistem/cara yang disebut sebagai *public-key infrastructure* (PKI).

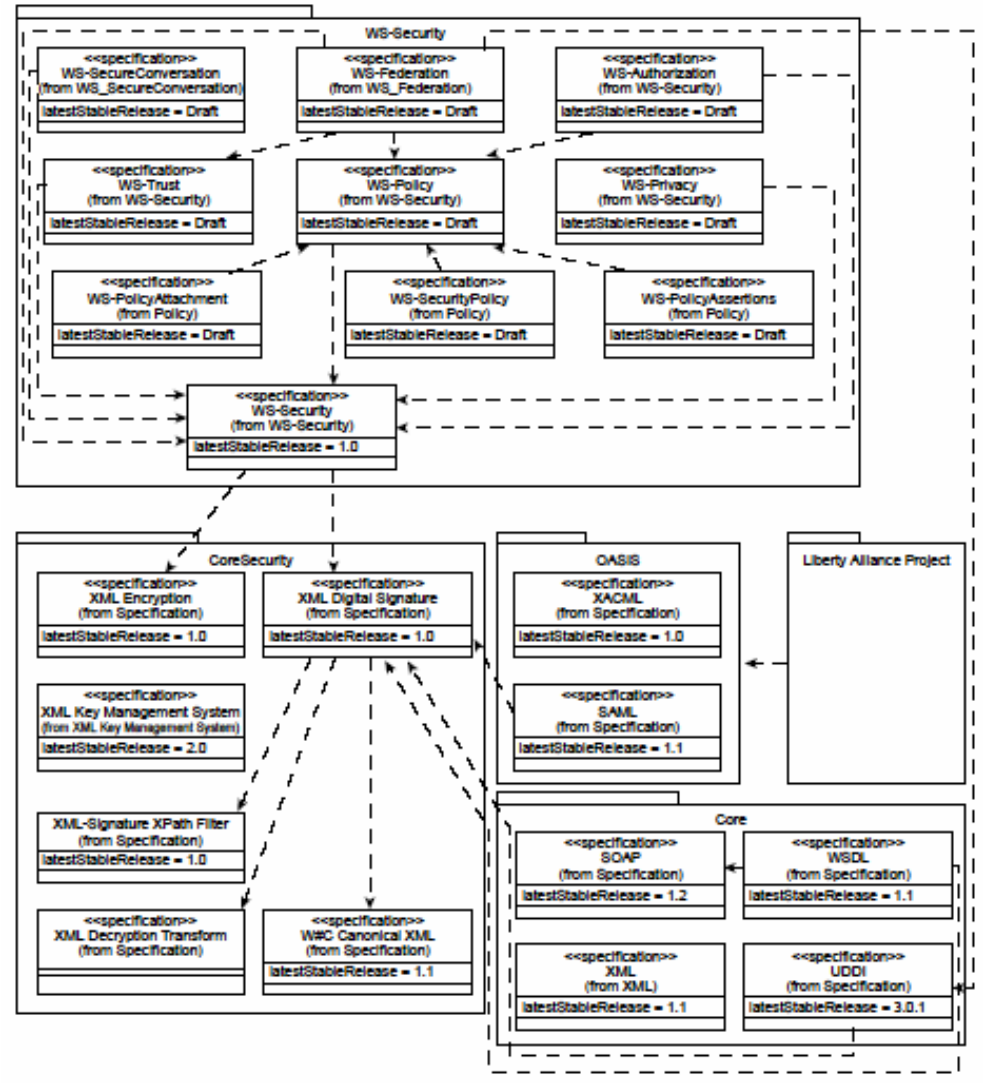
Public-key infrastructure (PKI) memungkinkan pengguna – pada jaringan publik yang kurang aman seperti internet – dapat melakukan transaksi perbankan ataupun pertukaran data penting secara aman dengan menggunakan sistem kriptografi kunci-publik (pasangan kunci-publik (*public-key*) dan kunci-privat (*private-key*)) yang diperoleh dan disebarluaskan oleh pihak yang berwenang. Pada PKI terdapat komponen-komponen sebagai berikut:

- *Certificate Authority* (CA), yang melakukan penerbitan dan pemeriksaan sertifikat digital.
- *Registration Authority* (RA), yang melakukan pemeriksaan sebelum *certificate authority* mengeluarkan serifikat digital.
- *Directories*, tempat penyimpanan sertifikat digital berserta kunci-pubik yang digunakan.
- *Certificate Management System* (sistem pengelolaan sertifikat).



Gambar 1 *Public-key infrastructure* dengan standar X.509[2]

Pada penggunaan sertifikat digital saat ini standar yang digunakan adalah X.509. Akan tetapi, standar X.509 belum merupakan standar yang telah ditetapkan , masih berupa suatu bentuk rekomendasi sehingga dalam penerapannya dilakukan dengan cara-cara yang berbeda-beda. Sebagai gambaran, perusahaan M dan perusahaan N mengimplementasikan X.509 pada masing-masing *web server* dan *browser*. Tetapi sertifikat yang dibangkitkan oleh produk perusahaan M tidak dapat dibaca oleh produk perusahaan N dan begitu juga sebaliknya[7]. Hal inilah yang menjadi salah satu hambatan yang dihadapi dalam penerapan PKI saat ini. Penerapan PKI dengan menggunakan X.509 (PKIX) dijelaskan pada gambar 1. Berdasarkan gambar 1, pihak *client* (*relying party*) harus melakukan proses yang cukup banyak meliputi pemrosesan terhadap ASN.1, pemrosesan terhadap sertifikat X.509, validasi jalur komunikasi yang digunakan, dan lain sebagainya. Dengan demikian terdapat beberapa hal yang menjadi permasalahan pada penerapan PKI saat ini (PKIX)[2], yaitu:



Gambar 2 Struktur standar pengamanan pada *web services*[1]

- Beban pada sisi client yang mengharuskan melakukan operasi coding/encoding protokol ASN.1, verifikasi sertifikat, validasi jaringan yang akan digunakan, pemeriksaan pembatalan, dsb.
- Kesulitan dan beban terhadap antarmuka aplikasi dengan infrastruktur layanan pada PKI; untuk infrastruktur yang berbeda maka dibutuhkan solusi yang berbeda juga.
- Kesulitan dalam melakukan pemrosesan terhadap X.509.

Penggunaan teknologi XML memungkinkan suatu pengembangan pengamanan pada *web services*. Untuk penerapan *web services* saat ini terdapat standar pengamanan *web services* yang sedang dikembangkan lebih lanjut. Struktur pada standar pengamanan *web services* tersebut dapat dilihat pada gambar 2. Berdasarkan struktur standar pengamanan *web service* [gambar 2], terdapat 5(lima) kelompok spesifikasi keamanan yang dikembangkan[1], yaitu:

- ***Core***: spesifikasi-spesifikasi yang mendasari web services yang terdiri dari *eXtensible Markup Language* (XML), *Web Service Definition Language* (WSDL), *Simple Object Access Protocol* (SOAP), dan *Universal Description, Discovery, and Integration* (UDDI).
- ***Core security***: standar-standar yang digunakan untuk pengamanan paling mendasar pada XML yang meliputi XML *encryption*, XML *digital signature*, XML *key management system*.
- **OASIS**: spesifikasi-spesifikasi keamanan yang dikembangkan oleh OASIS (*Organization for the Advancement of Structured Information Standards*).
- ***WS-security***: kelompok spesifikasi yang dikembangkan oleh Microsoft dan IBM dan dilakukan berdasar pada standardisasi OASIS.
- ***Liberty Alliance Project***: kumpulan spesifikasi yang dikembangkan oleh *Liberty Alliance Project*.

Selanjutnya yang akan dibahas pada dokumen ini hanyalah kelompok spesifikasi keamanan *core security* yaitu XML *encryption*, XML *digital signature*, dan XML *key management system*.

Penggunaan teknologi XML memberikan berbagai macam keuntungan di antaranya adalah dokumen dalam format XML merupakan dokumen yang berisi data terstruktur sehingga memungkinkan untuk dilihat oleh siapa saja dan/atau berbagai aplikasi. Dengan kemampuan dan kemudahan tersebut, XML digunakan sebagai format yang umum digunakan dalam komunikasi lintas aplikasi ataupun lintas sistem. Selain itu dikarenakan dokumen dengan format XML yang akan digunakan dapat ditentukan struktur data yang terdapat di dalamnya, maka cukup dengan menggunakan satu dokumen XML saja dapat digunakan oleh berbagai pihak yang membutuhkan dokumen tersebut. Akan tetapi dengan adanya kemudahaan tersebut, perlu diperhatikan juga proses pengamanan akses data dalam dokumen XML. Hal ini dikarenakan belum tentu semua komponen yang terdapat di dalam dokumen XML tersebut boleh diakses oleh siapa saja sehingga dibutuhkan proses pengamanan dengan penggunaan teknik kriptografi. Kriptografi saat ini sudah tidak hanya digunakan sebatas untuk menyembunyikan suatu data ataupun informasi saja. Teknik *Message Digest* digunakan untuk mengetahui keutuhan suatu pesan, tanda tangan digital (*digital signature*) digunakan untuk mendukung pembuktian keabsahan pengirim pesan, dan masih terdapat mekanisme lain pada kriptografi yang digunakan untuk menjaga suatu dokumen digital hanya dapat berasal dari pengirim yang unik dengan memiliki kunci enkripsi yang sesuai untuk dokumen digital tersebut. Dengan demikian pengirim tidak dapat melakukan penyangkalan terhadap suatu dokumen karena pada dokumen tersebut tanda tangan digital untuk menentukan keaslian dokumen. Pada proses penandatanganan digital juga terdapat proses enkripsi dokumen sehingga teknik enkripsi dan tanda tangan digital merupakan teknik yang saling melengkapi dan terkait. Hal ini menjadikan XML *encryption* dan XML *digital signature* menjadi teknologi yang mendasari keamanan pada XML (XML *security*).

XML ENCRYPTION

Pada dasarnya, teknik enkripsi dibutuhkan untuk melakukan enkripsi/dekripsi data digital pada dokumen XML dengan penggunaan sintaks XML untuk menentukan bagian dokumen yang merupakan data yang telah terenkripsi dan informasi mengenai proses dekripsi yang perlu dilakukan terhadap data yang terenkripsi tersebut. Pendekatan terstruktur yang digunakan pada XML bertujuan agar terhadap dokumen tersebut dapat dilakukan proses pencarian data-data yang memang dibutuhkan saja. Proses enkripsi yang umum dilakukan sebelumnya adalah dengan melakukan enkripsi terhadap dokumen XML sehingga seluruh isi dokumen XML tersebut telah terenkripsi sebagai satu kesatuan yang utuh. Hal ini menjadi permasalahan apabila dokumen XML akan digunakan oleh berbagai pihak terkait tetapi hanya untuk penggunaan komponen/elemen tertentu saja yang terdapat pada dokumen yang sama. Permasalahan tersebutlah yang menjadi dasar pemikiran pembentukan XML *encryption* oleh W3C. XML *encryption* (enkripsi XML) yang dikeluarkan oleh W3C tersebut merupakan sebuah model untuk melakukan proses enkripsi, dekripsi, dan bentuk representasi pada dokumen XML secara keseluruhan, elemen tunggal dalam dokumen XML, dan juga isi elemen XML. Pada XML *encryption* terdapat tambahan *tag* XML yang akan disisipkan ke dalam pesan yang akan dikirimkan. Hal utama pada proses enkripsi dokumen XML terdapat pada proses penambahan elemen **`EncryptedData`** untuk menyatakan bagian data yang terenkripsi. Dengan demikian, dimungkinkan untuk melakukan proses enkripsi terhadap data-data yang tersebar di dalam dokumen XML – dengan penggunaan beberapa *tag* tesebut di dalam satu dokumen XML.

XML DIGITAL SIGNATURE

Tanda tangan digital merupakan salah satu teknik kriptografi yang digunakan untuk menjamin keutuhan (*integrity*) serta menghindari penyangkalan (*nonrepudiation*) suatu pesan tertentu. Penggunaan tanda tangan digital dapat diterapkan pada berbagai jenis data digital, begitu juga untuk dokumen XML. XML *digital signature*, biasa disingkat dengan sebutan "XML Dsig", merupakan proses penandatangan digital pada dokumen XML. Pada XML *digital signature* didefinisikan mengenai cara pemberian tanda tangan digital pada dokumen XML dan penyisipan tanda tangan digital beserta informasi yang dibutuhkan berdasarkan skema yang digunakan pada dokumen XML. Seperti halnya pada XML *encryption*, XML *digital signature* juga dimungkinkan untuk penggunaan tanda tangan digital pada elemen-elemen tertentu dalam dokumen XML. Pemberian tanda tangan digital dilakukan dengan penambahan elemen **`SignedInfo`** dan juga **`SignatureMethod`** untuk menjelaskan algoritma yang digunakan untuk proses penandatanganan dokumen XML[7]. Tanda tangan digital XML dapat digunakan tidak hanya terhadap satu jenis tipe data saja tetapi juga dapat dilakukan penandatangan digital terhadap data dengan sandi karakter (HTML), data dengan sandi biner (data format gambar seperti BMP, JPG, GIF, dan sebagainya), data dengan sandi XML, ataupun bagian-bagian tertentu dari suatu dokumen XML.

Pemberian tanda tangan dan proses verifikasi dijelaskan pada spesifikasi XML *digital signature*. Seperti halnya pada XML *encryption*, XML *digital signature* juga merupakan suatu spesifikasi yang bersifat *technology-independent* sehingga dibutuhkan mekanisme tambahan untuk bentuk penerapan tanda tangan digital pada pertukaran pesan *web services*. Aplikasi yang akan menggunakan spesifikasi XML *digital signature* beserta juga proses enkripsi haruslah memenuhi hal-hal yang terkait dengan keamanan, yaitu:

- Ketika data dalam bentuk *ciphered* (tersandikan), simbol-simbol ataupun *digest* (penyingkatan) yang digunakan pada data tersebut haruslah tersandikan juga. Hal ini perlu diperhatikan untuk mengantisipasi serangan dengan menebak/menerka *plaintext* (*guessing plaintext attack*).
- Penggunaan proses transformasi XML *Decryption Transform* pada saat melakukan verifikasi tanda tangan digital.

```
<PersonalInfo>
  <ClothingSizes>
    <ShoeSize>11-D</ShoeSize>
    <WaistSize>34 inches</WaistSize>
  </ClothingSizes>
</PersonalInfo>
```

***Source Code* 1** Contoh dokumen XML

```
<PersonalInfo>
  <ClothingSizes Id="clothing-sizes-001">
    <ShoeSize>11-D</ShoeSize>
    <WaistSize>34 inches</WaistSize>
  </ClothingSizes>
  <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
    <dsig:SignedInfo>
      <dsig:CanonicalizationMethod
          Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <dsig:Reference URI="#clothing-sizes-001">
        <dsig:Transforms>
          <dsig:Transform Algorithm="http://www.w3.org/2002/07/decrypt#XML"/>
        </dsig:Transforms>
        <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <dsig:DigestValue>pIiwg5Z0wxE2EWxeDwGYWAWk/Nc=</dsig:DigestValue>
      </dsig:Reference>
    </dsig:SignedInfo>
    <dsig:SignatureValue>
        MAQTdjH541HIHsATjo3mIS3czScHf8rQz0qNV3qKvOHAAaEPyQ8uLy24RPteSAB3
        6fncZ8Q9eVKsxSe1P3EBMw==
    </dsig:SignatureValue>
    <dsig:KeyInfo>
      <dsig:X509Data>
        <dsig:X509SubjectName>CN=Ari,O=Phaos,L=New York,S=NYC=US</dsig:X509SubjectName>
      </dsig: X509Data >
    </dsig:KeyInfo>
  </dsig:Signature>
</PersonalInfo>
```

***Source Code* 2** Contoh dokumen XML dengan pemberian tanda tangan digital

```
<PersonalInfo>
  <ClothingSizes Id="clothing-sizes-001">
    <ShoeSize>11-D</ShoeSize>
    <WaistSize>
      <xenc:EncryptedData
          Type="http://www.w3.org/2001/04/xmlenc#Content"
          xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
        <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <dsig:KeyInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
          <dsig:KeyName>Acme Tailor Shop</dsig:KeyName>
        </dsig:KeyInfo>
        <xenc:CipherData>
          <xenc:CipherValue>N8SRrCrOjkHb==</xenc:CipherValue>
        </xenc:CipherData>
      </xenc:EncryptedData>
    </WaistSize>
  </ClothingSizes>
  <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
    ...
      <dsig:Reference URI="#clothing-sizes-001">
        <dsig:Transforms>
          <dsig:Transform Algorithm="http://www.w3.org/2002/07/decrypt#XML"/>
        </dsig:Transforms>
        <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <dsig:DigestValue>pIiwg5Z0wxE2EWxeDwGYWAWk/Nc=</dsig:DigestValue>
      </dsig:Reference>
    ...
  </dsig:Signature>
</PersonalInfo>
```

***Source Code* 3** Contoh dokumen XML terenkripsi

XML KEY MANAGEMENT SPECIFICATION (XKMS)

XML *key management specification* (XMKS) merupakan sebuah spesifikasi infrastruktur yang digunakan untuk pengamanan transaksi berbasis XML. Pada *web services* digunakan format komunikasi data berbasis XML dan untuk keamanan data-data tersebut digunakan teknik kriptografi kunci-publik. Pengelolaan terhadap kunci-publik ditenttukan dengan adanya *public-key infrastructure* (PKI). XKMS merupakan bentuk pengembangan berikutnya dari PKI yang ada saat ini (PKIX) dan juga melakukan perubahan standar PKI sebagai salah satu bentuk *web services*. Dengan demikian XKMS dapat melakukan proses registrasi pasangan kunci-publik (*private-key* dan *public-key*), penentuan lokasi penyimpanan kunci-publik, validasi kunci-publik, pencabutan (*revoke*) kunci-publik, dan pemulihan (*recover*) kunci-publik. Oleh karena itu, keseluruhan struktur PKI akan dikembangkan ke dalam lingkungan berbasis XML. XML *Key Management Specification* yang diterapkan sebagai *web service* akan mengurangi bentuk "ketergantungan" terhadap fungsi PKI yang terintegrasi dalam aplikasi. Sebelumnya penyedia PKI haruslah mengembangkan fungsi-fungsi khusus yang diterapkan pada produk aplikasi yang akan digunakan sedangkan dengan adanya XKMS sebagai *web service*, pada pengembangan produk aplikasi cukup dibuat fungsi untuk menentukan pengguna (*client*) yang mengakses fungsi/layanan yang disediakan oleh XKMS. Fungsi-fungsi pada XMKS meliputi:

- *Registration* (registrasi). Layanan pada XKMS dapat digunakan untuk mendaftarkan (registrasi) pasangan kunci dengan menggunakan fungsi "*register*". Pembangkitan pasangan kunci-publik dapat dilakukan oleh *client* ataupun layanan. Pada saat kunci-kunci telah terdaftarkan, layanan XKMS akan melakukan pengelolaan pencabutan ataupun pemulihan kunci-kunci, yang dibangkitkan oleh *server* ataupun *client*.
- *Locating*. Pada XKMS terdapat fungsi yang digunakan untuk mendapatkan kembali kunci-publik yang terdaftar.
- *Validation* (validasi). Fungsi validasi digunakan untuk memastikan bahwa kunci-publik yang telah didaftarkan dengan layanan XKMS valid dan tidak kadaluarsa ataupun telah dicabut.

Keunggulan utama XKMS dibandingkan PKI yang ada sebelumnya adalah proses enkapsulasi kerumitan yang ada pada PKI menjadi komponen pada sisi *server*. Dengan demikian pihak *client* hanya perlu mengetahui cara mengakses antarmuka yang disediakan. Model penerapan XML *Key Management Specification* (XKMS) dapat dilihat pada gambar 3. Sebagai pengembangan lebih lanjut dari PKI yang ada saat ini, XKMS memiliki karakteristik[2] kompatibel (selaras) dengan infrastruktur kunci-publik yang menggunakan X.509 (PKIX); dapat mendukung perubahan struktur dasar kebijakan yang akan digunakan pada PKI, contohnya Federal Bridge CA, ataupun perpaduan X.509 dengan non-509; proses pembangkitan dengan menggunakan XML *signature* dan *encryption*; dapat dikembangluaskan. Untuk produk aplikasi yang akan menggunakan spesifikasi XKMS haruslah melakukan implementasi operasi penandaan ataupun verifikasi paling dasar (*basic*) dan juga dapat mengelola kunci-privat yang dimiliki oleh *client* kemudian dapat melakukan pembangkitan dan pemrosesan terhadap transaksi-transaksi berbasis XML. Selain itu, berbeda dengan produk aplikasi pada PKIX, produk aplikasi yang akan menerapkan XKMS tidak perlu melakukan pemrosesan terhadap protokol ASN.1 ataupun sertifikasi X.509. Dengan bentuk penggunaan XKMS sebagai *web service*, maka produk aplikasi "terbebaskan" dari kebutuhan untuk memahami dan menerapkan PKI tradisional (PKIX) – cukup dengan memanggil dan menggunakan layanan yang disediakan oleh *web service* XKMS.

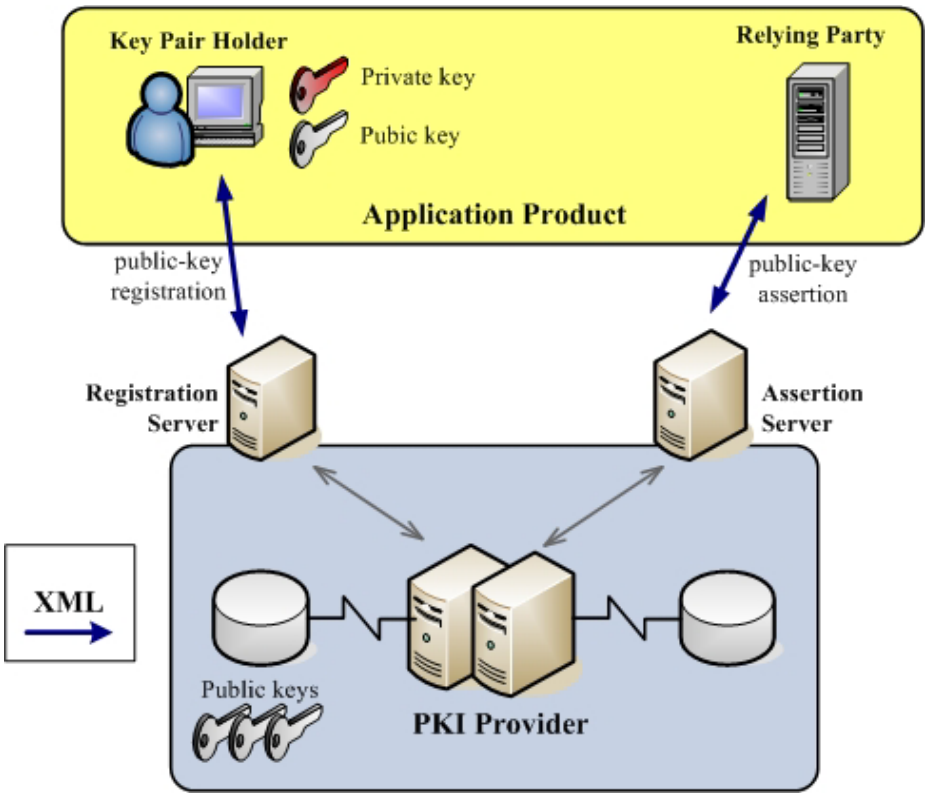
Pada gambar 3 dapat dilihat perubahan mendasar jika dibandingkan dengan PKI tradisional (PKIX) [gambar 1]. Dengan penerapan XKMS sebagai *web service*, pendaftaran kunci oleh pemilik pasangan kunci (*key pair holder*) dapat dilakukan dengan menggunakan layanan registrasi. Perbedaan yang paling tampak adalah pada sisi yang akan menggunakan kunci-publik (*relying party*). Untuk penggunaan kunci-publik yang dibutuhkan, dilakukan dengan meminta layanan PKI *provider* untuk

memberikan kunci-publik yang dibutuhkan – pihak *relying party* tidak perlu melakukan proses verifikasi sertifikat dan pemrosesan terhadap protokol yang digunakan serta pemrosesan lainnya seperti pada PKI tradisional (PKIX).

Pada spesifikasi XKMS terdapat 2(dua) komponen utama yaitu:

- **X-KISS** (*XML Key Information Service Specification*), merupakan sebuah protokol yang digunakan untuk mendukung proses pendelegasian aplikasi kepada layanan untuk melakukan pemrosesan informasi pada kunci (*key information*) terkait dengan XML *Digital Signature*, XML *Encryption*, ataupun pemrosesan lain yang membutuhkan penggunaan kunci-publik oleh aplikasi berbasis XML. Fungsi-fungsi pada XKISS meliputi lokasi kunci-publik yang dibutuhkan dan juga proses pengkaitan (*binding*) antara kunci dengan informasi identifikasi.
- **X-KRSS** (*XML Key Registration Service Specification*), merupakan protokol yang digunakan untuk mendukung proses pendaftaran (registrasi) pasangan kunci oleh pemilik pasangan kunci (*key pair holder*) dengan maksud bahwa pasangan kunci tersebut kemudian akan dibutuhkan dan dapat digunakan pada XKMS – oleh layanan XML *Key Information* dan/atau XML *Trust Assertion*.

Selain kedua komponen di atas terdapat satu komponen lainnya yaitu **X-BULK** (XML *Key Management Specification Bulk Operation*)[2]. Komponen ini digunakan untuk proses registrasi dalam jumlah yang sangat besar (*bulk*), contohnya untuk sertifikasi perangkat (*smartcard*, modem-kabel, dsb.). Perbedaan utama X-BULK dibandingkan dengan X-KRSS adalah X-BULK dibutuhkan untuk menghubungkan kumpulan *requests* dan *responses* yang tidak dapat dilakukan oleh X-KRSS. Untuk beberapa modul perangkat keras, X-KRSS tidak memiliki dukungan terhadap format pendaftaran kunci yang umum digunakan.



Gambar 3 Model penerapan XKMS[2]

X-KISS (XML *Key Information Service Specification*)

X-KISS memungkinkan client untuk melakukan pendelegasian kepada *Trust service* terhadap sebagian ataupun seluruh *tasks* yang dibutuhkan untuk pemrosesan elemen XML *Digital Signature*. Tujuan utama perancangan protokol pada X-KISS adalah untuk mengurangi kerumitan aplikasi yang akan menggunakan XML *Digital Signature*. Dengan menggunakan layanan *trust* (*trust service*), aplikasi akan terbebaskan dari kerumitan dan sintaks PKI yang mendasari pembentukan hubungan (*relationship*) *trust* – hubungan tersebut memungkinkan penggunaan spesifikasi seperti X.509/PKIX, SPKI, ataupun PGP. Selain itu, penggunaan X-KISS dapat membantu pengembang XML untuk mendapatkan kriptografi informasi kunci terkait dengan dokumen XML yang telah ditandatangani dan/atau dienkripsi.

Berdasarkan rancangan yang ada, XML *Signature Specification* tidak mengharuskan penggunaan satu kebijakan *trust* (kepercayaan) tertentu. Sang penandatangan dokumen tidak diharuskan untuk mencantumkan informasi kunci tetapi dapat mencantumkan elemen **`<ds:KeyInfo>`** yang berisi penjelasan kunci tersebut, seperti nama kunci, sertifikat yang digunakan, dan lain sebagainya. Bahkan pada elemen **`<ds:KeyInfo>`** dapat saja dituliskan *link* (URL) lokasi tempat untuk mendapatkan informasi lengkap dari elemen **`<ds:KeyInfo>`** tersebut. Pada rancangan X-KISS digunakan protokol yang terdiri dari sepasang pesan yang digunakan, yaitu pesan (*request*) yang dikirimkan oleh aplikasi kepada layanan *trust* (*trust service*) dan pesan (*respond*) yang berasal dari layanan *trust* (*trust service*).

Elemen-elemen Data

Pada X-KISS ditentukan jenis isi dan format pesan berdasarkan spesifikasi skema XML milik W3C. Semua nilai, jenis isi, dan format yang digunakan pada dokumen XML disandikan sebagai elemen data. Pada spesifikasi XKMS tipe data yang digunakan sangat terbatas akan tetapi dapat dilakukan penggunaan tipe data lain asalkan tetap bersesuaian dengan skema XML yang telah disepakati. Elemen-elemen data yang umum digunakan pada pengiriman pesan adalah sebagai berikut:

a. **`ds:KeyInfo`** merupakan elemen yang berisi informasi kunci yang digunakan pada pesan. Elemen **`ds:KeyInfo`** beserta format dan penggunaannya dijelaskan pada skema XML *Signature Specification*. Elemen **`ds:KeyInfo`** melakukan komunikasi data dengan menggunakan atribut dan elemen yang digunakan.
b. **`ResultCode`** merupakan tipe data dalam bentuk enumerasi. Elemen data **`ResultCode`** digunakan untuk mengembalikan nilai dari operasi antarmuka yang telah dilakukan (*result*). **`ResultCode`** memiliki nilai-nilai sebagai berikut:
   - **`Success`** apabila operasi berhasil dilakukan.
   - **`NoMatch`** apabila tidak ditemukan operasi yang akan dilakukan.
   - **`Incomplete`** apabila operasi yang ditemukan hanya tersedia sebagian.
   - **`Failure`** apabila terjadi kegagalan operasi tanpa adanya alasan yang spesifik.
   - **`Refused`** apabila operasi yang akan dilakukan ditolak.
   - **`Pending`** apabila operasi berada dalam antrian sehingga harus ditunda untuk pemrosesan di kemudian waktu.
c. **`AssertionStatus`** merupakan tipe data dalam bentuk enumerasi. Elemen data **`AssertionStatus`** digunakan sebagai laporan dari status permintaan (*assertion*) seperti pengkaitan kunci (*key bindings*). Nilai-nilai yang terdapat pada elemen **`AssertionStatus`** adalah:
   - **`Valid`** apabila proses pengkaitan dinyatakan valid.
   - **`Invalid`** apabila proses pengkaitan dinyatakan tidak valid.
   - **`Indeterminate`** apabila status tidak dapat ditentukan.

d. **Reason** merupakan elemen data yang berhubungan dengan elemen **AssertionStatus**. Elemen **Reason** berupa *string* yang menjelaskan alasan/informasi tertentu terkait dengan status yang didapat dari operasi yang dilakukan. Jika didapatkan **AssertionStatus Valid**, maka elemen **Reason** akan berisi aspek-aspek untuk status yang dinyatakan **Valid**. Begitu pula jika **AssertionStatus Invalid**, maka elemen **Reason** akan berisi aspek-aspek untuk status yang dinyatakan **Invalid** atau **Indeterminate** sedangkan apabila **AssertionStatus** mengembalikan nilai **Indeterminate**, maka elemen **Reason** akan berisi aspek-aspek yang digunakan untuk menyatakan status **Indeterminate**. Aspek-aspek yang terkait dengan status yang diberikan pada proses permintaan (*assertion*) adalah sebagai berikut:
   - **IssuerTrust** digunakan untuk menyatakan apakah proses permintaan disetujui oleh layanan *Trust* (*trust service*).
   - **Status** digunakan untuk menyatakan apakah layanan *Trust* berhasil melakukan verifikasi terhadap proses permintaan.
   - **ValidityInterval** digunakan untuk menyatakan apakah permintaan dilakukan dalam jangka waktu proses yang sah/valid.
   - **Signature** digunakan untuk menyatakan apakah tanda tangan digital yang disisipkan dalam data (elemen **ds:KeyInfo**) berhasil diverifikasi.

e. **Respond** merupakan elemen data berupa *string* yang diikutsertakan pada pesan (*response*) yang dikirimkan. Pada elemen **Respond** berisi permintaan penjelasan spesifik dari elemen data **ds:KeyInfo**. Layanan (*service*) haruslah dapat memberikan elemen data yang diminta apabila elemen data tersebut tersedia. Layanan akan memberikan elemen data yang diminta bersamaan dengan pesan (*response*). Jenis-jenis element **ds:KeyInfo** yang dapat diminta adalah sebagai berikut:
   - **KeyName** : nama kunci.
   - **KeyValue** : parameter kunci-publik.
   - **X509Cert** : sertifikat X.509 v3 yang digunakan pada kunci.
   - **X509Chain** : rangkaian sertifikat X.509 v3 yang digunakan pada kunci.
   - **X509CRL** : X.509 Certificate Revocation List v2; daftar pembatalan sertifikat.
   - **OCSP** : token OCSP PKIX yang digunakan untuk validasi sertifikat X.509 v3.
   - **RetrievalMethod** : data yang digunakan untuk Retrieval Method.
   - **MgmtData** : Management Data.
   - **PGP** : data penandaan kunci dengan PGP.
   - **PGPWeb** : kumpulan data penandaan kunci dengan PGP.
   - **SPKI** : penandaan kunci dengan SPKI.
   - **Private** : permintaan untuk memberikan kunci-privat bersamaan dengan pesan (response).

Layanan *Locate* (*Locate Service*)

Layanan *Locate* menerima masukan sebuah elemen **ds:KeyInfo** pada kunci-publik dengan mengembalikan satu atau lebih elemen **ds:KeyInfo** yang terkait dengan kunci-publik yang sama. Elemen **ds:KeyInfo** yang dikembalikan terspesifikasi pada elemen **Respond** di dalam pesan (*request*).

i. ***Request Message*** berisi elemen **Locate** yang memiliki upaelemen **Query** dan **Respond**. Elemen **Query** berisi permintaan terhadap data tambahan kunci-publik sedangkan elemen **Respond** berisi jenis-jenis elemen

ds:KeyInfo yang akan dikembalikan bersamaan dengan pesan (*response*).

ii. ***Response Message*** berisi elemen **LocateResult** yang memiliki upaelemen **Answer**. Elemen **Answer** merupakan rangkaian *string* yang berisi elemen **ds:KeyInfo** berdasarkan permintaan informasi tambahan yang dituliskan pada elemen **Query** dan **Respond** (*request message*). Pada *response message* terdapat **ResultCode** yang berisi nilai untuk menyatakan hasil dari proses *locate* yang telah dilakukan. Nilai-nilai tersebut terdiri atas:

- **Success** : proses *locate* berhasil dengan semua informasi yang dibutuhkan tersedia.
- **NoMatch** : proses *locate* berhasil tetapi tidak ada informasi yang sesuai.
- **Incomplete** : proses *locate* berhasil tetapi hanya ada sebagian informasi yang tersedia.
- **Failure** : proses *locate* gagal.

Layanan Validasi (*Validate Service*)

Layanan *Validate* memungkinkan *client* untuk melakukan *query* terhadap keterikatan (*binding*) antara elemen **ds:KeyInfo** dengan data lainnya. Pihak *client* akan memiliki acuan untuk melakukan permintaan **KeyBinding**. Acuan tersebut memberikan penjelasan mengenai **KeyId** ataupun elemen **ds:KeyInfo**. Pihak *server* akan memberikan satu atau lebih permintaan **KeyBinding** yang memenuhi kriteria pada pesan (*request*).

i. **ValidityInterval** merupakan struktur pada layanan *validate* yang menjelaskan batasan berlakunya suatu permintaan

ii. **KeyId** merupakan elemen yang digunakan untuk menjelaskan *Universal Resource Identifier* (URI) milik kunci – dapat berupa nama, lokasi, atau lainnya yang sesuai dengan spesifikasi URI.

iii. **KeyUsage** merupakan elemen yang digunakan untuk menentukan penggunaan dari kunci yang tersedia. Penggunaan kunci dapat ditujukan untuk keperluan enkripsi (**Encryption**), penandatanganan (**Signature**), ataupun pertukaran kunci (**Exchange**).

iv. **KeyBinding** merupakan elemen yang digunakan untuk melakukan permintaan (*assertion*) keterikatan (*binding*) elemen-elemen data yang berkaitan dengan kunci-publik meliputi **KeyName**, **KeyID**, **KeyValue**, dan **X509Data**.

v. *Request Message* berisi elemen **Validate** dengan upaelemen **Query** dan **Respond**. **Query** pada layanan *validate* berisi struktur **KeyBinding** yang perlu untuk dilengkapi ataupun dilakukan proses validasi sedangkan elemen **Respond** berisi jenis-jenis elemen **ds:KeyInfo** yang akan dikembalikan bersamaan dengan pesan (*response*).

vi. *Response Message* berisi elemen **ValidateResult** dengan upaelemen **Answer**. Elemen **Answer** merupakan rangkaian struktur **KeyBinding** yang berisi hasil dari proses validasi. Pada *response message* terdapat **ResultCode** yang berisi nilai untuk menyatakan hasil dari proses *validate* yang telah dilakukan. Nilai-nilai yang digunakan untuk menyatakan hasil proses terdiri atas:

- **Success** : proses *validate* berhasil dengan semua informasi yang dibutuhkan tersedia.
- **NoMatch** : proses *validate* berhasil tetapi tidak ada informasi yang sesuai.
- **Incomplete** : proses *validate* berhasil tetapi hanya ada sebagian informasi yang tersedia.
- **Failure** : proses *validate* gagal.

X-KRSS (XML *Key Registration Service Specification*)

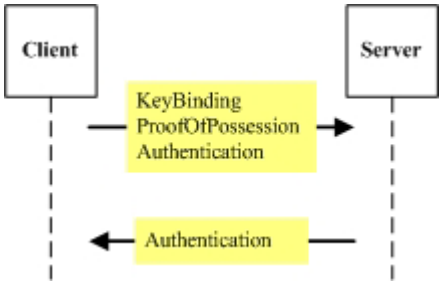
XML *Key Registration Service Specification* (X-KRSS) memungkinkan proses manajemen/pengelolaan informasi yang terikat dengan pasangan kunci-publik (kunci-publik dan kunci-privat). Spesifikasi layanan pada X-KRSS mendukung operasi **Register**. Informasi yang dibutuhkan akan diikatkan pada pasangan kunci-publik melalui elemen **`KeyBinding`** XKMS. Dengan demikian proses pembangkitan pasangan kunci-publik akan terdukung untuk dapat dilakukan oleh pihak *client* ataupun pihak *server*. Protokol manajemen sertifikat yang ada saat ini – contohnya pada PKIX – hanyalah terfokus pada salah satu bagian dari daur-hidup (*lifecycle*) sertifikat (umumnya berupa penerbitan sertifikat) ataupun terlalu luas (kompleks) untuk ditangani aplikasi yang telah digunakan saat ini. Oleh karenanya, spesifikasi X-KRSS ditujukan untuk dapat memenuhi tantangan kebutuhan adanya suatu protokol XML untuk manajemen kunci ataupun sertifikat yang berfokus pada *client*. Spesifikasi pada X-KRSS mendukung keseluruhan daur-hidup sertifikat yang meliputi:

- *Key Registration* (pendaftaran pasangan kunci publik)
- *Key Revocation* (pembatalan pasangan kunci publik)
- *Key Recovery* (pemulihan pasangan kunci publik)

Registration

Untuk tahap *registration* (registrasi), melalui aplikasi XML pemilik pasangan kunci (*key pair holder*) dilakukan pendaftaran melalui sebuah *registration server* [gambar 3]. Kunci-publik akan dikirimkan kepada registration server menggunakan pesan (*request*) yang sudah ditandatangani – menggunakan operasi *register*. Untuk menjamin keamanan, pada pesan tersebut dapat disisipkan:

- Informasi nama dan atribut. Pemilik kunci mengirimkan *request* pada infrastrukur untuk melakukan penyimpanan nama dan atribut yang kemudian dikirimkan dengan menggunakan kunci-publik. Infra-struktur tersebut memungkinkan untuk mendaftarkan, merubah, ataupun menggantikan informasi tersebut sesuai dengan kebijakan yang digunakan.
- Informasi otentikasi. Informasi yang dibutuhkan pada infrastruktur untuk melakukan proses otentikasi terhadap pemilik kunci.
- *Proof-of-possession* (POP) kunci-privat. Apabila proses pem-bangkitan pasangan kunci-publik dilakukan pada pihak *client* maka dibutuhkan adanya *proof-of-possession* (POP) kunci-privat.



Gambar 4 Proses pendaftaran **`KeyBinding`**

Setelah pemilik kunci melakukan pengiriman pesan (*request*) kepada *registration server*, maka *registration server* akan memberikan pesan (*response*) konfirmasi yang berisi status proses pendaftaran yang telah dilakukan (diterima, ditolak, atau ditunda) dan juga konfirmasi informasi nama dan atribut yang didaftarkan bersamaan dengan kunci-publik. Pada proses pendaftaran kunci-publik, layanan registrasi (*registration service*) dimungkinkan untuk dapat memberikan semua ataupun sebagian permintaan dari pihak *client* [gambar 4].

Pada proses pendaftaran, *request message* berisi penjelasan mengenai elemen **`KeyBinding`** yang akan didaftarkan. Elemen **`KeyBinding`** dapat hanya memiliki informasi yang kurang lengkap. Untuk kasus semacam ini, pihak *client* dapat meminta *registration service* untuk menyediakan informasi tambahan yang dibutuhkan untuk pengkaitan kunci (*key binding*) pada proses pendaftaran tersebut. Sebagai contoh, pihak *client* tidak dapat menentukan parameter pada kunci-publik karena pasangan kunci-publik (kunci-

publik dan kunci-privat) dibangkitkan oleh *registration service*.

Untuk proses otentikasi pesan (*request*) digunakan data-data elemen **`Authentication`**. Bentuk dan isi dari elemen **`Authentication`** bergantung pada media yang digunakan, algoritma kunci-publik yang digunakan, dan pihak yang membangkitkan kunci (*client* atau *server*).

i. *Request Message* berisi elemen **`Register`** dengan upaelemen **`Template`**, **`Authentication`**, dan **`Respond`**. Pada **`Template`** berisi struktur **`KeyBinding`** yang berisi elemen-elemen dari permintaan (*request*) *client* untuk didaftarkan. **`Authentication`** merupakan elemen yang berisi informasi yang digunakan untuk proses otentikasi. **`Respond`** berisi kumpulan data elemen yang akan dikembalikan bersamaan dengan pesan (*response*).

ii. *Response Message* berisi elemen **`RegisterResult`** dengan upaelemen **`KeyBinding`** dan **`Private`**. Elemen **`KeyBinding`** berisi informasi pengkaitan kunci yang telah dilakukan oleh layanan (*service*) sedangkan **`Private`** merupakan informasi tambahan disediakan oleh *server* dan berisi paramater kunci-privat yang dibangkitkan oleh *registration service*.

Revocation

Pada *registration service* memungkinkan bagi client untuk membatalkan/menarik kembali (*revoke*) permintaan (*assertion*) yang telah diterbitkan sebelumnya. Pesan (*request*) untuk melakukan pembatalan (*revocation*) dilakukan dengan metode yang kurang lebih sama dengan proses pendaftaran kunci hanya saja untuk proses pembatalan status pada **`KeyBinding`** atau **`KeyAssertion`** adalah **`Invalid`** dan jika *registration service* tidak memiliki data permintaan maka nilai pada **`ResultCode`** adalah **`NotFound`**.

Proses pembatalan dilakukan dengan pihak *client* mengirim pesan (*request*) pembatalan kunci-publik kemudian pihak *client* melakukan proses otentikasi terhadap pesan tersebut dengan menggunakan kunci-privat. Selanjutnya *registration service* akan melakukan pemrosesan terhadap pesan (*request*) dari *client* dan mengirimkan pesan (*response*) kepada *client* bahwa proses pembatalan telah dilakukan.

Recovery

Pada *registration service* memungkinkan bagi *client* untuk meminta pemulihan (*recovery*) kunci. Pemulihan kunci dilakukan apabila *client* lupa kunci-privat yang telah didaftarkan sebelumnya. Pesan (*request*) untuk melakukan pemulihan kunci dilakukan dengan metode yang kurang lebih sama dengan proses pendaftaran kunci hanya saja untuk proses pemulihan kunci, layanan *key recovery* membutuhkan waktu dalam memberikan respon terhadap permintaan pemulihan kunci (*key recovery request*) sehingga **`ResultCode`** akan bernilai **`Pending`** dan jika *registration service* tidak memiliki data permintaan maka nilai pada **`ResultCode`** adalah **`NotFound`**.

Proses yang dilakukan untuk melakukan pemulihan kunci diawali dengan pihak *client* menghubungi bagian administrator layanan pemulihan kunci (*key recovery service*) dengan menggunakan prosedur otentikasi berdasarkan kebijakan yang telah ditetapkan. Kemudian bagian administrator akan mengirimkan kode otorisasi kepada client untuk pemrosesan pemulihan kunci.

KESIMPULAN

Teknologi *web service* telah menjadi teknologi yang banyak digunakan untuk proses pertukaran data ataupun informasi. Penggunaan teknik kriptografi kunci-publik dan sertifikat digital menjadi bentuk pengamanan pada web service. Untuk pengelolaan kunci-publik dan sertifikat digital digunakan *public-key infrastructure* (PKI). Pengembangan XML – sebagai teknologi yang mendasari *web services* –

dengan PKI menghasilkan infrastruktur pengamanan dan pengelolaan kunci-publik berbasis XML, XML *Key Management Specification* (XKMS). XKMS menjadi bentuk PKI generasi berikutnya dengan mengurangi tingkat kerumitan pada PKI yang ada saat ini. Walaupun demikian, XKMS saat ini masih bersifat draf dan rekomendasi sehingga masih perlu dilakukan pengembangan lebih lanjut sebagai langkah untuk pengamanan *web services*.

DAFTAR PUSTAKA

[1] Gutiérrez, Carlos, Fernández-Medina, Eduardo, dan Piattini, Mario. *Web Services Security: Is The Problem Solved?*. 2004.

[2] Hardjono, Thomas. *XKMS Overview*. VeriSign. 2002.

[3] Kermaier, Ari. *Securing Web Services: XML Security Standards in Practice*. Phaos Technology Corp. 2003.

[4] Lambeth, Charlie. White Paper *Understanding Web Services*. Software Spectrum. 2003.

[5] Munir, Rinaldi. Diktat Kuliah IF5054 – Kriptografi. Bandung: ITB. 2005.

[6] Nguyen, Michael. *XML and PKI*. Infocomm Development Authority of Singapore.

[7] Schäfer, Alexandra dan Rechert, Werner. *Security and Web Services*. Jerman: Universitas Teknologi Darmstadt.

[8] W3C (*World Wide Web Consortium*) – http://www.w3c.org; terakhir diakses pada tanggal 15 Desember 2006.