

Mata Kuliah : EL-695 Keamanan Jaringan Informasi

Dosen : Ir. Budi Rahardjo PhD.

Jur. / Prog. : Sistem dan Teknologi Informasi / Magister Teknik Elektro

---

**Tinjauan Keamanan Dokumen XML**  
**(eXtensible Markup Language)**



**TUGAS AKHIR SEMESTER**

Disusun oleh:

Simeon Wicaksana Nugraha – 23202053

**PROGRAM MAGISTER TEKNIK ELEKTRO**  
**INSTITUT TEKNOLOGI BANDUNG**

**2003**

## DAFTAR ISI

<b>Abstrak</b>	<b>3</b>
<b>1 Pendahuluan</b>	<b>4</b>
<b>2 XML</b>	<b>6</b>
<b>3 Keamanan XML</b>	<b>8</b>
3.1 XML Digital Signature.....	9
3.2 XML Encryption (XML Enc) .....	14
3.3 XML Key Management Specification (XKMS) .....	18
3.4 Security Assertion Markup Language (SAML) .....	23
3.5 XML Access Control Markup Language (XACML).....	26
<b>4 Kesimpulan</b>	<b>28</b>
<b>Referensi</b>	<b>29</b>

## *Abstrak*

Saat ini representasi data menggunakan XML dalam aplikasi berbasis web di internet berkembang pesat. XML mampu menampilkan data dalam format yang mudah dipahami oleh mesin maupun manusia (*machine-usable and human readable*). Karena itu data yang ditulis dalam dokumen XML lebih mudah digunakan untuk beberapa tujuan sekaligus.

Internet memungkinkan setiap komputer yang terhubung di dalamnya saling bertukar informasi. Karena itu diperlukan suatu mekanisme keamanan yang menjamin data hanya dapat diakses orang dengan otorisasi tertentu.

Tugas akhir ini akan membahas standar-standar yang digunakan untuk keamanan dokumen XML yaitu menggunakan XML *Signature* dan XML *Encryption*.

**Kata kunci:** XML, XML *Signature*, XML *Encryption*.

# 1 Pendahuluan

Aspek keamanan mempunyai peran yang sangat penting di dunia bisnis untuk memberikan kepastian materi dan transaksi dalam bisnis, memberi perlindungan kerahasiaan dan menjamin informasi digunakan secara benar. Model bisnis berbasis internet terutama penggunaan website membuat sistem keamanan juga turut berubah. Usaha untuk membuat infrastruktur sistem keamanan sangat bergantung dengan macam perangkat keras, perangkat lunak, aplikasi dan tingkat kebutuhan keamanan. Untuk itu diperlukan suatu standar yang mampu berkembang untuk dapat menyesuaikan diri dengan teknologi-teknologi baru di bidang sistem keamanan. Standar ini dapat digunakan secara modular ataupun bekerja bersamaan.

XML digunakan pada beragam aplikasi untuk bermacam materi. XML mampu menampilkan data dalam format yang mudah dipahami oleh mesin maupun manusia (*machine-usable and human readable*). XML dirancang untuk dapat diolah atau dikombinasikan. XML menyediakan integritas, kerahasiaan dan manfaat keamanan yang lain untuk dokumen XML. Sistem keamanan XML telah terintegrasi dengan XML itu sendiri sebagai suatu fitur dalam XML itu sendiri.

Teknologi keamanan yang telah ada menyediakan kumpulan algoritma dan teknologi keamanan yang dapat digunakan di dalam sistem keamanan XML. Akan tetapi format terbaru dalam teknologi keamanan juga digunakan untuk kebutuhan implementasi keamanan untuk aplikasi sistem keamanan XML. Salah satu alasannya adalah penggunaan format binary yang membutuhkan *software* khusus meski hanya untuk mengolah informasi bagian keamanannya saja. Alasan lain adalah standar ini tidak

dirancang untuk digunakan dengan XML dan tidak mendukung pendekatan teknis XML untuk pengaturan materi seperti menspesifikasikan materi dengan *uniform resource identifier* (URIs) atau menggunakan definisi standar yang lain dari XML untuk menentukan bagian dari materi XML.

XML menangani permasalahan ini dengan menentukan suatu kerangka kerja yang umum dan aturan pemrosesan yang dapat digunakan aplikasi secara bersama dengan menggunakan perangkat-perangkat umum, menghindari kebutuhan penyesuaian aplikasi yang berlebihan untuk menambahkan keamanan. XML menggunakan kembali konsep, algoritma, dan teknologi inti dari sistem keamanan yang telah ada dengan juga mengenalkan perubahan-perubahan yang diperlukan untuk mendukung integrasi yang dapat diperluas dengan XML. Hal ini memungkinkan interoperabilitas dengan beragam infrastruktur yang telah ada dan untuk penggunaan lebih lanjut.

Tulisan ini membahas dokumen XML dan konsep-konsep keamanan yang digunakan dalam sistem keamanan XML. Bagian awal membahas secara umum tentang XML kemudian diikuti dengan pembahasan tentang konsep keamanan dalam XML yang terinci dalam standar sistem keamanan XML.

## 2 XML

Spesifikasi XML mendefinisikan sintaks dan aturan-aturan penggunaan tag untuk membentuk struktur dari informasi. Sebuah kata dapat digunakan untuk mendefinisikan elemen tag dan atribut untuk membentuk struktur suatu informasi. Aturan-aturan yang telah dibuat dalam spesifikasi XML akan menjadikan dokumen XML menjadi bentuk yang *well-formed*, yaitu dokumen XML yang dapat diproses oleh perangkat XML. XML juga mendefinisikan struktur dokumen secara eksplisit dari suatu dokumen yang strukturnya telah didefinisikan dengan membuat *XML Schema* atau *Document Type Definition* (DTD) sehingga dokumen tersebut dapat divalidasi sebagai dokumen XML yang benar.

Bahasa XML yang dibuat oleh orang yang berbeda dapat digabungkan. Jika misalnya didefinisikan sintaks untuk alamat dan orang lain mendefinisikan juga untuk keperluan yang lain, Orang tersebut dapat menggunakan sintaks yang didefinisikan sebelumnya ke dalam sintaks yang sesuai dengan keperluan orang tersebut. XML *namespace* digunakan untuk menghubungkan elemen dengan *schema* yang tepat dan untuk menghindari konflik antar elemen. XML *namespace* menghubungkan tag-tag dengan unique identifiers (URI) dan digunakan untuk menghindari ambiguitas. Dokumen XML yang *well-formed* dapat diproses oleh perangkat yang mengerti XML secara baik, termasuk parser yang mengerti aturan dan pemrosesan dokumen XML secara umum. Salah satu keuntungannya adalah penggunaan XML *namespace* ini tidak bergantung dengan pembendaharan kata didefinisikan pada dokumen yang tertentu. Hal ini membuat setiap perangkat XML yang dibuat dapat digunakan untuk semua aplikasi XML.

Banyak bahasa XML yang sudah didefinisikan yaitu XHTML untuk membuat halaman web, DocBook untuk membuat dokumen teknik, RSS untuk distribusi materi, RDF untuk menampilkan informasi, MathML untuk informasi matematis, BRML untuk laporan bisnis, dan lain-lain.

Contoh berikut ini menampilkan dokumen XML untuk manajemen rumah sakit, termasuk didalamnya adalah elemen XML <PatientRecord>, <Name> and <Diagnosis> dan juga penggunaan XML *namespace* yang berhubungan dengan *lab* untuk <lab:Diagnosis> sebagai elemen yang tidak akan mengakibatkan konflik dengan elemen <Diagnosis>.

```
<PatientRecord xmlns="http://www.medical.org/" xmlns:lab="http://www.lab.org/">
  <Name>John Doe</Name>
  <Account>123456</Account>
  <Visit date="10pm March 10, 2002">
    <Diagnosis>Broken second metacarpal</Diagnosis>
    <lab:Diagnosis>
      <lab:Xray>encoded xray image</lab:Xray>
    </lab:Diagnosis>
  </Visit>
</PatientRecord>
```

### 3 Keamanan XML

Standar keamanan XML menyediakan kumpulan standar teknis untuk memenuhi kebutuhan keamanan. Standar-standar tersebut dirancang agar sejalan dengan konsep-konsep umum tentang XML dan memberikan perbaikan terutama dalam hal-hal sbb.:

Standar keamanan XML mendefinisikan perbendaharaan kata-kata untuk menunjukkan informasi keamanan dengan menggunakan teknologi XML, seperti *XML Schema*, sebagai definisinya.. Contohnya elemen `<KeyInfo>` yang didefinisikan di rekomendasi *XML Digital Signature* untuk membawa informasi tanda keamanan atau enkripsi.

Standar keamanan XML dapat menggunakan standar XML yang telah ada jika dimungkinkan untuk mengambil manfaat dari standar tersebut. Contohnya, *XML Digital Signature* dapat menggunakan ekspresi *XPath* untuk mengambil sebagian dari dokumen XML untuk diproses.

Standar keamanan XML dirancang untuk memberikan keleluasaan dan kemungkinan pengembangan aspek-aspek XML. Sistem keamanan dapat diaplikasikan dalam dokumen XML, dalam elemen XML dan materi elemen, sebaik yang terdapat dalam dokumen *binary*. Perluasan perbendaharaan kata-kata dalam XML dilakukan dengan menggunakan *XML namespace* dan kemudahan definisi *XML Schema*.

Teknologi keamanan XML dapat diaplikasikan dalam sistem keamanan *end-to-end* , yang sangat penting saat pesan XML dikirimkan melalui beberapa perantara. Sistem keamanan XML lebih ke arah keamanan materi, sehingga saling melengkapi bila dihubungkan dengan sistem keamanan yang borientasi ke keamanan transportasi misalnya SSL.

Teknologi keamanan XML menggunakan teknologi keamanan dan kriptografi yang telah ada. Contohnya X.509 V3 dapat digunakan tanpa memberi definisi ulang. Secara sederhana dikodekan kedalam format teks. Algoritma yang telah ada, seperti SHA1, juga masuk ke dalam standar keamanan XML menggunakan URI yang unik dan telah didefinisikan juga cara menggunakannya dalam XML.

Standar-standar dalam keamanan XML adalah sbb.:

1. *XML Digital Signature* untuk integritas dan keaslian,
2. *XML Encryption* untuk kerahasiaan,
3. XML Key Management (XKMS),
4. *Security Assertion Markup Language (SAML)* berkenaan dengan autentifikasi,
5. *XML Access Control Markup Language (XACML)* berkenaan dengan aturan mengenai otorisasi.

### 3.1 XML Digital Signature

*XML Digital signature* digunakan untuk menyediakan kepastian integritas materi dalam dokumen dan untuk membuat serta menguji tandatangan elektronik tersebut. Dengan kepastian integritas materi, pengguna materi dapat mendeteksi perubahan isi materi yang tidak diinginkan, baik karena kesengajaan maupun karena kecelakaan. Tidak seperti mekanisme *checksum* yang sederhana, tanda digital menghubungkan inisiasi materi dengan penanda isi materi menggunakan teknik kriptografi. Tanda digital tersebut adalah sebuah angka pendek yang nilainya tetap, khas terhadap isi materi, dan tidak berguna untuk diketahui jika tanpa isi materi itu sendiri. Teknik kriptografi membuat materi dengan tanda digital menjadi lebih susah untuk diubah isinya oleh seseorang yang bukan memberi tanda digital itu sendiri tanpa terdeteksi. Kepastian integritas materi tidak

hanya memberikan perlindungan dalam transportasi tapi juga dalam penyimpanan dan dalam suatu proses.

Tanda tangan elektronik menyediakan kesamaan dengan tanda tangan konvensional sehingga dapat digunakan untuk berbagai tujuan, misalnya persetujuan, konfirmasi, kontrak. Dengan menggunakan tanda tangan digital maka dimungkinkan perubahan menuju konsep bisnis *online* tanpa membutuhkan proses persetujuan secara manual. Hal ini mengurangi tenggang waktu, biaya, dan ketidaknyamanan karena lokasi geografis yang berjauhan atau perbedaan zona waktu. Tanda tangan digital menggunakan kriptografi untuk membangun tanda tangan yang lebih kuat dibanding tanda tangan yang dihasilkan oleh teknik yang lain.

Rekomendasi *XML Digital signature* mendefinisikan mekanisme untuk mendukung seluruh kreasi dan pengujian dari tanda tangan digital, termasuk kemampuan untuk membuat dan menguji:

1. Seluruh dokumen XML sebaik elemen dan bagian materi elemen dari dokumen XML,
2. Dokumen yang berubah-ubah, termasuk dokumen binary,
3. Gabungan dokumen termasuk dokumen ganda dengan elemen XML dan materi elemen,
4. Hal-hal yang dimasukkan kedalam tanda tangan,
5. *Counter-signatures* (tanda tangan yang mempunyai tanda tangan didalamnya)

Rekomendasi *XML Signature* juga mendukung aplikasi dari dokumen XML dengan banyak *XML Signature* atau dari bagian lain dari dokumen. Spesifikasi *XML Digital Signature* dan spesifikasi yang berkaitan dengannya (*XML Canonicalization*) juga mendefinisikan teknik yang membuat tanda tangan digital tersebut mantap meskipun

banyak variasi yang dibolehkan di XML, misalnya spasi. Kanonikalisasi digunakan untuk mengurangi variasi sehingga antar aplikasi keamanan XML dapat saling berinteroperasi.

Elemen XML <Signature> dapat dituliskan dalam beberapa cara bergantung dari aplikasi yang diinginkan. Elemen tersebut dapat diletakkan terpisah dengan dokumen yang ditanda tangani. Tanda tangan ini disebut "detached" dan digunakan untuk materi bukan XML. Jika materi XML ditandatangani, maka elemen <Signature> dtambahkan di dokumen XML. Jika elemen <Signature> terletak di dokumen XML, maka elemen tersebut diletakkan dibawah elemen dokumen ("*enveloped*" *signature*). Pada beberapa kasus , elemen dokumen dapat diletakkan dalam elemen <Signature> ("*enveloping*" *signature*).

Dalam contoh berikut ini, tanda tangan diletakkan didalam elemen <PatientRecord> sehingga elemen <Signature> akan berada didalam elemen <PatientRecord>:

```
<PatientRecord xmlns="http://www.medical.org/">
  <Name>John Doe</Name>
  <Account> 123456 </Account>
  <Visit date="10pm March 10, 2002">
    <Diagnosis> Broken second metacarpal </Diagnosis>
  </Visit>
  <Signature xmlns='http://www.w3.org/2000/09/xmldsig#'> ... </Signature>
</PatientRecord>
```

Saat tanda tangan ditambahkan kedalam dokumen sebagai bagian dokumen, akan mengubah dokumen tersebut. Tanda tangan tersebut diperiksa dengan membandingkannya dengan dokumen asli tanpa tanda tangan. Rekomendasi *XML Digital Tinjauan Keamanan Dokumen XML*

*Signature* mendefinisikan penghilangan <Signature> sebaga bagian dalam proses verifikasi.

Cara lain untuk memberi tanda tangan adalah dengan membuat dokumen XML baru kemudian memasukkan informasi-informasi yang berkaitan dengan tanda tangan ke dalam elemen <Signature>. Contohnya sbb.:

```
<Signature xmlns='http://www.w3.org/2000/09/xmldsig#'>
  <SignedInfo> ... </SignedInfo>
  <SignatureValue> ... </SignatureValue>
  <Object>
    <SignatureProperties>
      <p:Purpose xmlns:p="http://www.myexample.com/schemas">Approval </p:Purpose>
    </SignatureProperties>
  </Object>
</Signature>
```

Konsep-konsep penting dalam *XML digital signatures*:

1. Suatu tanda tangan hanya valid jika materi yang ditandatangani tidak berubah. Tanda tangan ini dibuat menggunakan angka yang pendek, dengan panjang yang tetap, dan dirancang agar jika materi yang ditandatangani berubah maka juga dapat ikut berubah. Jadi tanda tangan hanya akan valid jika angka yang digunakan untuk membuat tanda tangan sama dengan yang digunakan untuk memeriksanya.
2. Elemen <Signature> adalah struktur XML yang berisi nilai kriptografi dalam elemen <SignatureValue> seperti dalam struktur XML dokumen yang

- ditandatangani, yaitu struktur <SignedInfo>. Hal ini berarti isi dari struktur <SignedInfo> tidak diubah agar tanda tangan tersebut valid.
3. Penandatanganan membuat <Reference> untuk setiap pokok materi yang dimasukkan dalam tanda tangan. Setiap <Reference> memasukkan angka setiap pokok materi dan sebuah URI untuk pokok materi tersebut. <Reference> juga digunakan untuk membuat isi tanda tangan, menentukan jenis algoritma, dan informasi-informasi penting lainnya yang merupakan bagian dari struktur <SignedInfo>.
  4. URI digunakan untuk menunjukkan semua isi materi, termasuk materi yang bukan XML seperti data gambar dan file teks. Tidak selalu menggunakan URI akan tetapi untuk beberapa kasus, penggunaan URI sangat berguna.. Bentuk khusus URI dapat digunakan untuk menunjukkan elemen XML dalam dokumen yang sama dengan tanda tangannya, sehingga tanda tangan dapat dipindahkan dalam materi XML yang akan ditandatangani.
  5. <Reference> dapat memilih satu atau lebih perubahan yang dapat dilakukan sebelum digunakan untuk membuat isi tanda tangan. Salah satu penggunaannya adalah untuk menandai bagian dari dokumen XML yang sudah diketahui tidak akan diubah isinya. Hal ini dilakukan dengan mendefinisikan bagian dokumen yang akan ditandai, misalnya menggunakan ekspresi Xpath.
  6. Algoritma untuk menghasilkan isi tanda tangan yang sama antara yang dibuat dengan yang digunakan untuk memeriksa membutuhkan bentuk XML yang kanonik.

## 3.2 XML Encryption (XML Enc)

Rekomendasi *XML Encryption* mendefinisikan perbendaharaan kata dan aturan pemrosesan dalam XML agar tercipta kerahasiaan dalam berbagai jenis materi. *XML Encryption* menjaga kerahasiaan informasi baik dalam transportasi maupun ketika disimpan dalam media penyimpanan data.. Teknologi yang menyediakan kerahasiaan yang lain seperti *secure sockets layer* (SSL) / transport layer security (TLS) atau virtual private networks (VPNs) hanya menyediakan kerahasiaan selama informasi ditransit, dan bukan selama disimpan di server.

Pemilik materi informasi dapat melakukan enkripsi sehingga datanya terjamin kerahasiaannya. Proses enkripsi tersebut akan membuat data menjadi tak terbaca hingga dilakukan proses dekripsi. Umumnya, enkripsi dilakukan dengan menggunakan enkripsi kunci simetris, teknik ini efisien meskipun untuk dokumen yang besar. Enkripsi kunci simetris menggunakan kunci yang sama untuk enkripsi dan dekripsinya. Untuk mengirimkan informasi rahasia menuju penerima, pengirim harus memberikan juga kunci simetrisnya kepada penerima.

Masalah tersebut dapat dihindari dengan menggunakan kriptografi asimetris atau konsep *public-key*. Kriptografi ini menggunakan sepasang kunci, satu untuk enkripsi, satu lagi untuk dekripsinya. Pengirim dapat menggunakan *public-key* penerima yang tersebar luas. Dekripsi dilakukan dengan menggunakan *private-key* penerima. Pada kenyataannya konsep *public-key* dan kriptografi simetris digunakan bersamaan. Kunci simetris dienkripsi dengan konsep *public-key* hal ini dilakukan karena kriptografi *public-key* tidak efisien untuk data yang besar. Data kunci simetris dan data materi yang dienkripsi selanjutnya dikirimkan ke penerima.

Berikut ini adalah langkah-langkah yang harus dilakukan pengirim:

1. Mengenkripsi materi dengan menggunakan kunci simetris
2. Mengenkripsi kunci simetris dengan menggunakan *public-key* penerima
3. Menyatukan kedua data terenkripsi menjadi satu paket beserta keterangan algoritma yang diperlukan
4. Mengirimkan paket tersebut menuju penerima

Penerima membaca informasi yang dikirimkan setelah melalui langkah-langkah berikut:

1. Membuka paket data yang dikirimkan
2. Mendekripsi kunci simetris menggunakan *private key*
3. Mendekripsi materi menggunakan kunci simetris

Rekomendasi *XML Encryption* mendefinisikan kerangka kerja dan aturan pemrosesan untuk enkripsi dan dekripsi dokumen XML. *XML Encryption* mendefinisikan perbendaharaan kata dalam XML untuk membentuk paket yang berisi semua informasi yang dibutuhkan untuk proses enkripsi. Isi paket tersebut misalnya algoritma enkripsi dan parameter-parameternya, informasi mengenai kunci enkripsi, dan materi terenkripsi.

Rekomendasi *XML Encryption* mendukung hal-hal berikut:

1. materi XML dan bukan materi XML dapat dienkripsikan.
2. Kerahasiaan dapat dicapai dalam tingkatan yang cukup bagus dalam materi XML. Enkripsi dapat dilakukan pada tingkatan elemen XML sebaik pada XML dokumen. Kelebihan ini membuat kerahasiaan dapat dicapai untuk bagian yang ingin dirahasiakan saja.

3. Proses enkripsi menghasilkan dokumen XML yang teratur dari dokumen XML yang juga teratur. Sebagian elemen dari dokumen XML dapat dienkripsi secara subsekuensial oleh perangkat XML.
4. *XML Encryption* kompatibel dengan XML Digital Signatures dan dapat digunakan bersamaan.
5. *XML Encryption* dapat mengenkripsi kunci simetri yang terpaket bersama materi terenkripsi.
6. *XML Encryption* mendukung banyak algoritma dan teknik enkripsi.

Konsep-konsep penting dalam *XML Encryption*:

1. Elemen XML atau elemen materi yang terenkripsi digantikan dengan elemen `<EncryptedData>`.
2. Jika yang dienkripsi materi yang bukan XML, maka akan dihasilkan dokumen XML yang baru yang mengandung elemen `<EncryptedData>`.
3. Elemen `<EncryptedData>` dapat memasukkan atribut `Type` untuk keperluan dekripsi oleh penerima. `Type` ini yang memberi indikasi suatu elemen XML atau materi elemen telah dienkripsi, atau memberi jenis informasi lain.
4. Elemen `<EncryptedData>` mendefinisikan algoritma untuk enkripsi, menyediakan materi terenkripsi, dan mungkin memasukkan informasi yang penting untuk menentukan kunci yang dibutuhkan untuk dekripsi.
5. Kunci simetris yang digunakan untuk mengenkripsi elemen dapat disertakan dalam elemen `<EncryptedKey>`.
6. *XML Encryption* mendukung pemilihan algoritma enkripsi yang tepat.
7. Definisi untuk mengidentifikasi informasi mengenai kunci didasarkan pada definisi *XML Digital Signature* dan diperluas.

8. informasi yang didefinisikan pemakai dapat dihubungkan dengan elemen terenkripsi, misalnya penunjuk waktu atau log.

Bagian penting dalam sistem keamanan XML adalah interaksi antara *XML Digital Signatures* dan *XML Encryption*. Misalnya dokumen XML yang terenkripsi dan tertandatangani sbb:

```
<PatientRecord xmlns="http://www.medical.org/" xmlns:lab="http://www.lab.org/">
  <Name> John Doe </Name>
  <Account> 123456 </Account>
  <EncryptedData Type='element' ... </EncryptedData>
  <Signature>
    <SignedInfo>
      <Reference URI=""> ... </Reference>
    </SignedInfo>
  </Signature>
</PatientRecord>
```

Dalam kasus ini, tanda tangan menandai seluruh dokumen <PatientRecord>, karena URI <Reference> "". Tanda tangan hanya dapat diuji jika materi tidak diubah. Ketika memberi tanda tangan, penandatanganan harus mengidentifikasi dimana elemen <EncryptedData> berada sebagai bagian dari tandatangan. Hal tersebut berguna untuk memberi tahu bahwa elemen <EncryptedData> harus didekripsi sebelum memeriksa tanda tangan.

### 3.3 XML Key Management Specification (XKMS)

*XML Key Management Specification (XKMS)* mendefinisikan protocol untuk pelayanan manajemen *Public Key*. Manajemen *public key* terdiri dari pembuatan pasangan *public key* dan *private key*, menggabungkan pasangan kunci tersebut dengan identitas dan atribut yang lain, dan representasi pasangan kunci ini dalam bentuk yang berbeda, seperti nama kunci, sertifikat digital atau parameter kunci. Teknologi *public key* merupakan bagian penting dari *XML Digital Signatures*, *XML Encryption* dan aplikasi keamanan yang lain. Ketika diberi tanda tangan, *private key* digunakan untuk menandatangani dan *public key* digunakan untuk verifikasi. Untuk enkripsi, *public key* digunakan untuk enkripsi dan *private key* digunakan untuk dekripsinya. Dalam hal tersebut, *private key* dijaga oleh pemilik kunci sedangkan *public key* dapat diketahui secara umum. XKMS dirancang untuk membantu mengatur *public key* untuk memungkinkan verifikasi tanda tangan dan enkripsi materi untuk penerima

Manajemen *public key* biasanya membutuhkan proses registrasi pasangan kunci yang akan dibuat dan beberapa keterangan singkat berhubungan dengan *public key* untuk identitas dan keterangan dari pemiliknya. Proses registrasi menggabungkan beberapa ketentuan untuk mengurangi akibat penggunaan *public key* yang tidak tepat. Jika terjadi perubahan dalam informasi kepemilikan kunci, manajemen *public key* dapat menghapus informasi mengenai pasangan kunci tersebut sehingga tidak disalahgunakan. Demikian juga apabila terdapat tambahan informasi pasangan kunci, misalnya saat jumlah penggunaannya bertambah. XKMS mendefinisikan format XML untuk mendukung permintaan dan jawaban untuk manajemen *public key* yaitu untuk proses registrasi, pembatalan dan penambahan.

Pasangan *public key* dapat digunakan untuk menandai dan verifikasi atau untuk enkripsi dan dekripsi setelah proses registrasi selesai. Untuk memberi informasi kepada penerima dokumen yang bertanda tangan atau terenkripsi, rekomendasi *XML Digital Signature* mendefinisikan elemen <KeyInfo>. Informasi ini dapat terdiri dari nama kunci, sertifikat digital yang mengandung *public key*, kumpulan parameter kunci, atau URI yang menunjukkan dimana *public key* diperoleh. XKMS menyediakan suatu format XML agar aplikasi dapat memproses atau menemukan kunci yang digunakan. Format XML tersebut menjamin proses dikerjakan secara berurutan dan menyediakan informasi tentang kunci dalam format yang benar untuk penerima.

XKMS mendefinisikan 3 spesifikasi:

- *XML Key Registration Service Specification (XKRSS)*,
- *XML Key Information Service Specification (XKISS)*
- *Protocol Bindings*

Spesifikasi tersebut mendefinisikan pesan-pesan yang dibutuhkan untuk proses registrasi dan mengatur informasi yang berhubungan dengan *public key* dan memastikan keamanan.

Layanan informasi mengenai kunci diberikan dengan menggunakan elemen <KeyInfo>. Layanan ini termasuk mencari dan memutuskan elemen <KeyInfo> yang dipakai setelah mendapat permintaan informasi mengenai kunci. Layanan yang lain memberikan validasi pasangan kunci yang digunakan.

Konsep-konsep penting dalam XKMS:

1. Spesifikasi dalam XKMS mendefinisikan protokol XML yang membawa informasi registrasi kunci dan permintaan informasi mengenai kunci menuju ke *trust server* dan membawa jawaban informasi yang dibutuhkan dari server.
2. Elemen <ds:KeyInfo> memberikan suatu proses yang dapat dipercaya oleh pemakai dan mengurangi kompleksitas di sisi pemakai. Implementasi layanan ini tergantung dari jenis layanan yang digunakan dan juga dimungkinkan untuk berperan dalam *public key infrastructure* (PKI).
3. Spesifikasi XKMS mendukung penggunaan *XML Digital Signatures* untuk integritas dan autentifikasi. Spesifikasi XKMS juga mendefinisikan sistem autentifikasi yang lain yang mendukung prosedur keamanan yang lain.

Contoh penggunaan XKMS:

#### 1. XKMS Register Request:

```
<RegisterRequest xmlns:ds=http://www.w3.org/2000/09/xmldsig#
  Service=http://test.xmltrustcenter.org/XKMS
  RequestId="hZMRGyATbUL4H7rYOanR6Q=="
  xmlns="http://www.w3.org/2002/03/xkms#">
  <RespondWith>X509Cert</RespondWith>
  <Prototype Id="tX4Y83grmj/eIVoeYNuTNg==">
    <KeyInfo>
      <ds:KeyValue>
        <ds:RSAKeyValue>
          <ds:Modulus> zvbTdKsTprGAKJdg
            i7ulDR0eQBptL...
          </ds:Modulus>
          <ds:Exponent> AQAB </ds:Exponent>
        </ds:RSAKeyValue>
      </ds:KeyValue>
    </Prototype>
  </RegisterRequest>
```

```

        </ds:KeyValue>
    </KeyInfo>
    <KeyUsage>Signature</KeyUsage>
    <UseKeyWith Application="urn:ietf:rfc:2633"
        Identifier="alice@alicecorp.test" />
</Prototype>
<Authentication>
    <ProofOfPossession>
    <ds:Signature> signing with the pr
        ivate key demonstrates possession of it</ds:Signature>
    </ProofOfPossession>
</Authentication>
</RegisterRequest>

```

## 2. XKMS Register Response:

```

<RegisterResult xmlns:ds=http://www.w3.org/2000/09/xmldsig#
    Service="http://test.xmltrustcenter.org/XKMS"
    ResultMajor="Success"
    RequestId="hZMRGyATbUL4H7rYOanR6Q=="
    ResponseId="k9gyjDdhLLV1vbF7RzJjIw=="
    xmlns="http://www.w3.org/2002/03/xkms#">
    <KeyBinding Id="LVrJqd26QzO9GWJD0usQwg==">
        <KeyInfo>
            <KeyName>Sally Smith key</KeyName>
        </KeyInfo>
        <KeyUsage>Signature</KeyUsage>
        <UseKeyWith Application="urn:ietf:rfc:2633"
            Identifier="alice@alicecorp.test" />
    </KeyBinding>
</RegisterResult>

```

### 3. XKMS Validate Request:

```
<ValidateRequest xmlns:ds=http://www.w3.org/2000/09/xmldsig#
    xmlns:xenc=http://www.w3.org/2001/04/xmlenc#
    Service=http://test.xmltrustcenter.org/XKMS
    RequestId="zzjmNi9YL+dnkRXzDoqPoQ=="
    xmlns="http://www.w3.org/2002/03/xkms#">
  <RespondWith>KeyName</RespondWith>
  <RespondWith>KeyValue</RespondWith>
  <RespondWith>Multiple</RespondWith>
  <KeyBindingQuery Id="T/QMi7gGuKCCNWPi120A/w==">
    <KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate> certificate encoded as text </ds:X509Certificate>
      </ds:X509Data>
    </KeyInfo>
    <KeyUsage>Signature</KeyUsage>
    <UseKeyWith Application="urn:ietf:rfc:2633"
      Identifier="alice@alicecorp.test" />
  </KeyBindingQuery>
</ValidateRequest>
```

### 4. XKMS Validate Response:

```
<ValidateResult xmlns:ds=http://www.w3.org/2000/09/xmldsig#
    xmlns:xenc=http://www.w3.org/2001/04/xmlenc#
    Service=http://test.xmltrustcenter.org/XKMS
    ResultMajor="Success"
    RequestId="zzjmNi9YL+dnkRXzDoqPoQ=="
    ResponseId="0WeinJVdbyBKruXhiqTscg=="
    xmlns="http://www.w3.org/2002/03/xkms#">
  <KeyBinding Id="m0/p5bekjemI4tV+FPBkig==">
```

```

    <KeyInfo>
      <ds:KeyValue>
        <ds:RSAKeyValue>
          <ds:Modulus>...</ds:Modulus>
          <ds:Exponent>AQAB</ds:Exponent>
        </ds:RSAKeyValue>
      </ds:KeyValue>
    </KeyInfo>
    <KeyUsage>Signature</KeyUsage>
    <UseKeyWith Application="urn:ietf:rfc:2633"
      Identifier="alice@alicecorp.test" />
    <Reason>IssuerTrust</Reason>
    <Reason>RevocationStatus</Reason>
    <Reason>ValidityInterval</Reason>
  </KeyBinding>
</ValidateResult>

```

### 3.4 Security Assertion Markup Language (SAML)

Autentifikasi adalah suatu proses pengenalan identitas. Proses ini diperlukan untuk membatasi akses terhadap materi yang ada, mengidentifikasi pelaku transaksi, dan memberikan informasi yang sesuai dengan identitas yang dipunyai. Proses autentifikasi ini harus mampu dilakukan dalam proses "single sign-on" sebaik jika harus melakukan autentifikasi berkali-kali.

Untuk seorang pelanggan yang menyukai kenyamanan, langkah yang berbelit dalam proses autentifikasi mempunyai nilai yang merugikan dalam sebuah website. Jadi, meskipun dalam sebuah sistem yang kompleks, sistem "single sign-on" sangat berguna untuk kepraktisan dan transparansi sistem secara keseluruhan. Pelanggan hanya perlu

melakukan sekali proses autentifikasi di awal saja untuk selanjutnya dapat masuk ke dalam sistem dengan kewenangan yang dimilikinya. Otorisasi adalah proses yang dilakukan setelah proses autentifikasi dilakukan. Proses ini yang menentukan seseorang yang telah diautentifikasi dapat mengakses materi yang mana saja sesuai dengan kewenangan yang dimilikinya. Aturan kontrol akses bergantung dengan identitas, materi, proses yang akan dilakukan, ataupun informasi-informasi yang lain.

*XML Security Assertion Markup Language (SAML)* mendefinisikan perbendaharaan kata dalam XML untuk menangani autentifikasi dan otorisasi, memungkinkan proses "single sign-on" fungsi-fungsi diluar SAML yang mempunyai fungsi yang sama. Spesifikasi SAML juga mendefinisikan kerangka kerja umum yang memungkinkan kepastian diberikan dengan validasi waktu, dan kepastian diberikan kepada pengguna yang spesifik.

Contoh penggunaan SAML:

#### 1. Autentifikasi menggunakan SAML:

```
<Assertion>
  <!-- Conditions may include optional XML attributes
           defining a time period for validity -->
  <Conditions NotBefore="dateTime" NotOnOrAfter="dateTime">
    <!-- limit who can rely on this assertion -->
    <AudienceRestrictionCondition>
      <Audience>http://www.example.com/Members</Audience>
    </AudienceRestrictionCondition>
  </Conditions>
  <!-- Optional Advice used to include supporting evidence,
           proofs, assertions, pointers to updates etc.
           One or more of the possible sub-elements. -->
```

```

<Advice>
  <AssertionIDReference>id</AssertionIDReference>
  <!-- refer to other supporting assertion -->
  <Assertion>...</Assertion>
  <!-- provide inline information -->
</Advice>
<!-- Authentication - example: SSL client certificate authentication --> <AuthenticationStatement
AuthenticationMethod="urn:ietf:rfc:2246"
      AuthenticationInstant="dateTime">
  <Subject>
    <NameIdentifier Format="urn:oasis:names:tc:
      SAML:1.0:assertion#emailAddress">
      john_doe@example.com </NameIdentifier>
    </Subject>
  </AuthenticationStatement>
  <ds:Signature> XML Digital Signature for assertion </ds:Signature>
</Assertion>

```

## 2. Otorisasi menggunakan SAML

```

<Assertion>
  <!-- Conditions with optional XML attributes
      defining a time period for validity -->
  <Conditions NotBefore="dateTime" NotOnOrAfter="dateTime">
    <!-- limit who can rely on this assertion -->
    <AudienceRestrictionCondition>
      <Audience>http://www.example.com/Members</Audience>
    </AudienceRestrictionCondition>
  </Conditions>
  <!-- Decision choices: "Permit", "Deny", "Indeterminate"-->
  <AuthorizationDecisionStatement Resource="http://www.fjhirsch.com/info" Decision="Permit">
    <Subject>

```

```

        <NameIdentifier
            Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
            john_doe@example.com </NameIdentifier>
        </Subject>
        <Action Namespace="urn:oasis:names:tc:SAML:1.0:action:rwe"> Read </Action>
        <Evidence>
            <AssertionIDReference>LOG-Abc12Z</AssertionIDReference>
        </Evidence>
    </AuthorizationDecisionStatement>
    <ds:Signature> XML Digital Signature for assertion </ds:Signature>
</Assertion>

```

### 3.5 XML Access Control Markup Language (XACML)

XACML mendefinisikan hal-hal sbb.:

1. Perbendaharaan kata XML untuk ekspresi aturan otorisasi,
2. Perbendaharaan kata XML untuk ekspresi berbagai kondisi yang digunakan untuk membuat aturan,
3. Bagaimana aturan-aturan dikombinasikan dan dievaluasi
4. Pernyataan-pernyataan yang dibuat sebagai kumpulan aturan.

Contoh aturan akses menggunakan XACML:

```

<Rule RuleId="//medico.com/rules/rule3" Effect="Permit">
    <Target>
        <Subjects>
            <saml:Attribute AttributeName="RFC822Name"
                AttributeNamespace="//medico.com">
                <saml:AttributeValue>*</saml:AttributeValue>
            </saml:Attribute>
        </Subjects>
    </Target>
</Rule>

```

```

        </saml:Attribute>
    </Subjects>
    <Resources>
        <saml:Attribute AttributeName="documentURI"
            AttributeNamespace="//medico.com">
            <saml:AttributeValue>//medico.com/records.*</saml:AttributeValue>
        </saml:Attribute>
    </Resources>
    <Actions>
        <saml:Action>read</saml:Action> </Actions>
</Target>
<Condition>
    <Equal>
        <AttributeDesignator AttributeName="urn:oasis:names:tc:xacml:
            identifiers:AccessSubject" /> <AttributeDesignator
            AttributeName="patientName" />
    </Equal>
</Condition>
</Rule>

```

## 4 Kesimpulan

Standar keamanan XML mendefinisikan bahasa XML dan aturan pemrosesan untuk memenuhi kebutuhan keamanan. Sebagian besar standarnya merupakan penggabungan dari standar yang lainnya, terutama dari *XML Digital Signature* dan *XML Encryption*. Contoh yang lain adalah pernyataan dalam penentuan aturan di SAML dan XACML. Kombinasi dalam standar keamanan XML ini berkembang cepat seiring dengan perkembangan dalam hal-hal praktis atau karena teknologinya sendiri yang juga berkembang.

Standar keamanan XML sangat penting untuk bisnis model online yang menggunakan teknologi XML yang diadaptasi untuk layanan web, manajemen hak cipta digital, dan aplikasi lain. Standar keamanan XML menangani autentifikasi, otorisasi, kerahasiaan, integritas, dan privasi yang dibangun menggunakan gabungan standar-standar yang sudah ada.

# Referensi

- [1] Damiani, E., et.al, *Securing XML Documents*
- [2] Hirsch, F., <http://www.fjhirsch.com/xml/xmlsec/starting-xml-security.html>, 2002.
- [3] W3, <http://www.w3.org/TR/xmlenc-decrypt>, 2002
- [4] XMLcom, <http://www.xml.com>, 2002
- [5] XMLsec, <http://www.xmlsec.com/WhyXMLSecurity.html>, 2002