

## BAB II

### LANDASAN TEORI

Dalam bab ini akan dijelaskan beberapa pengetahuan dasar yang menjadi landasan perancangan sistem, yaitu uraian tentang layanan TELKOMNet@Home, termasuk RADIUS. Berikutnya, penjelasan dari sisi pendukung dalam pembuatan program aplikasi yaitu XML dan XSLT.

#### 2.1 TELKOMNet@Home

TELKOMNet@Home adalah layanan akses Internet dial-up berlangganan dengan sistem paket jam akses per bulan, yang diperkenalkan oleh PT. TELKOM Divisi Regional V Jawa Timur. Karena menggunakan sistem berlangganan maka untuk bisa menggunakan layanan ini pelanggan harus mendaftarkan diri terlebih dahulu di tempat pelayanan TELKOM. Ada 3 (tiga) macam paket yang ditawarkan, yaitu : Paket 15 jam, Paket 50 jam, dan Paket 250 jam.

Biaya paket berlangganan dan kelebihan jam pemakaian dalam satu bulan dikenakan tarif seperti Tabel 2.1.

**Tabel 2.1** Tarif Paket TELKOMNet@Home.

No	Paket	Biaya per bulan ( termasuk pulsa )	Biaya Kelebihan per menit ( termasuk pulsa )
1.	15 Jam	Rp. 125.000,-	Rp. 125,-
2.	50 Jam	Rp. 350.000,-	Rp. 115,-

3.	250 Jam	Rp. 1.500.000,-	Rp. 100,-
----	---------	-----------------	-----------

Untuk bulan pertama pendaftaran, pelanggan hanya dikenakan biaya kelebihan per menit yang dikalikan dengan jumlah jam penggunaan. Untuk bulan berikutnya pelanggan akan membayar biaya dengan normal sebagai abonemen tiap bulannya sesuai dengan paket yang diambil. Jika pemakaian melebihi dari jam paket yang diambil maka pelanggan juga akan dikenakan biaya kelebihan per menit sesuai dengan paket yang diambil.

Adapun ketentuan-ketentuan yang harus digunakan dalam proses dial-up oleh pelanggan di antaranya yaitu :

Nomer Akses : 080989988

Username : kode area + notelp@home (misal : 0315611158@home)

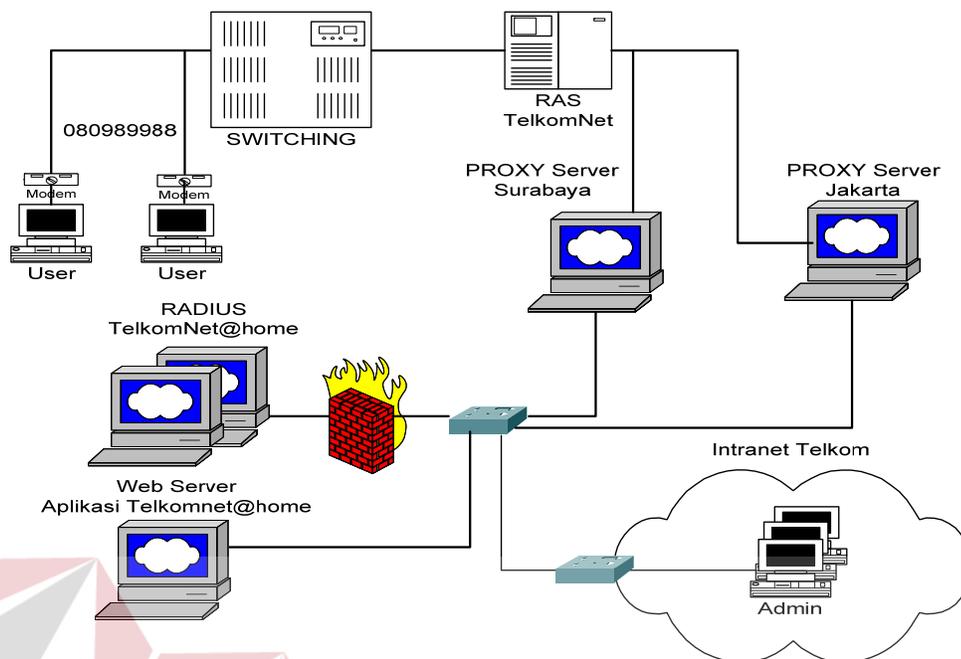
( akses hanya bisa dari nomor telepon yang didaftarkan)

Password : Diisi sesuai dengan keinginan pelanggan

(minimal 4 karakter, terdiri atas huruf ( a-z kecil semua) dan atau angka)

#### *Proses dial-up TELKOMNet@Home*

Untuk mempermudah penjelasan tentang proses dial-up TELKOMNet@Home, perlu digambarkan terlebih dahulu konfigurasi jaringan TELKOMNet@Home seperti ditunjukkan pada Gambar 2.1.



**Gambar 2.1** Konfigurasi Jaringan TELKOMNet@Home.

Pada Gambar 2.1 dapat dilihat bahwa untuk program aplikasi pendukung layanan TELKOMNet@Home akan mencakup penggunaan di sisi pelanggan dan internal TELKOM, seperti akan diuraikan dibawah.

- a. Pelanggan dengan komputer dan modem yang terhubung dengan line telepon akan melakukan dial-up ke nomer telepon RAS (Remote Access Server) TELKOMNet yaitu 080989988.
- b. Switching/ Sentral akan meroutingkan panggilan ke RAS, dan akan diterima oleh RAS.

- c. RAS akan melakukan proses Authentikasi dan Authorisasi ke RADIUS Server dengan melalui Proxy Server di Jakarta dan Surabaya. RADIUS secara detil ada di Sub Bab 2.4.
- d. Proses permintaan Authentikasi dan Authorisasi, sebelum diterima RADIUS Server akan dicek kebenarannya oleh Firewall bahwa proses permintaan tersebut dari RAS TELKOMNet dan bukan dari RAS lain.
- e. RADIUS Server akan mengecek kebenaran data dari user menyangkut username, password dan nomor pemanggil (Authentikasi), apabila data sesuai maka RADIUS Server akan memberikan hak akses, alokasi IP, DNS, Protokol dan Kompresi. Data tersebut akan dikirimkan kembali oleh RADIUS Server ke RAS TELKOMNet.
- f. Setelah melewati Proxy Server Surabaya dan Jakarta, data reply dari RADIUS akan diterima oleh RAS TELKOMNet.
- g. RAS TELKOMNet akan mengirimkan hasil dari Authentikasi dan Authorisasi ke dial-up user, kemudian akan mengirimkan permintaan Accounting Start Time ke RADIUS Server untuk user tersebut menyangkut waktu start terkoneksi, alokasi IP, protokol dan attribute-attribute yang lainnya. Di RADIUS Server data tersebut akan disimpan di database MySQL.
- h. Apabila dial-up user melakukan proses pemutusan koneksi maka RAS TELKOMNet akan meminta Accounting Stop Time ke RADIUS Server untuk

user tersebut. RADIUS Server akan melakukan proses updating di database untuk field Accounting Stop Time di session record user tersebut.

## 2.2 RADIUS (Remote Authentication Dial-In User Service)

RADIUS adalah sebuah protokol pengaman untuk client/server yang digunakan untuk mengelola dial-in user yang menyangkut proses AAA (Authentication, Authorization, Accounting), pertama kali diciptakan oleh Livingstone Enterprises. Informasi-informasi yang rahasia akan disimpan dalam sebuah lokasi tersentral yang disebut sebagai RADIUS server. RADIUS client (dalam hal ini adalah RAS TELKOMNet) berhubungan dengan RADIUS server untuk proses pengecekan kebenaran user (Authentication).

Tiga fungsi utama RADIUS Server yaitu AAA (Authentication, Authorization, Accounting). Proses AAA bisa dilakukan/ diintegrasikan dengan user sistem, lokal user file, atau menggunakan database (internal ataupun eksternal).

### *Authentication*

Proses Authentication digunakan untuk mengecek kebenaran user yang melakukan dial-in remote access. Terdapat dua macam hasil proses Authentication yaitu Access-Accept (Akses diterima) dan Access-Reject (Akses ditolak).

### *Authorization*

Proses ini melakukan kontrol terhadap pelayanan yang disediakan dalam jaringan. Setelah user akses diterima dalam Authentication, maka RADIUS akan

memberitahukan bahwa user pengakses tersebut diterima untuk mengakses beserta dengan atribut-atribut yang diberikan seperti IP address, protokol PPP (Point to Point Protocol).

### *Accounting*

Proses Accounting digunakan untuk melihat dan mengamati seberapa banyak user yang telah melakukan dial-in, biasanya digunakan untuk proses tagihan .

Dalam aplikasi yang akan dibuat di Tugas Akhir ini, RADIUS yang digunakan yaitu FreeRADIUS versi 1.0 berbasis unix, dan proses AAA diintegrasikan dengan database eksternal MySQL. FreeRADIUS adalah *open source project* yang mempunyai banyak kemampuan dan kemudahan sehingga bisa diaplikasikan sesuai keperluan.

### **2.3 eXtensible Markup Language (XML)**

Extensible Markup Language (XML) adalah sebuah bahasa meta-markup yang menyediakan format untuk mendeskripsikan struktur data. Fasilitas ini lebih menekankan pada pendeklarasian content dan lebih memberi arti terhadap hasil pencarian antar berbagai platform. Dengan catatan, XML merupakan generasi yang baru dari penampilan data yang berbasis web dan pemanipulasian aplikasi. Pada XML kita dapat mendefinisikan set-set tag yang tidak terbatas. Tag-tag HTML dapat digunakan untuk menampilkan kata-kata dalam bentuk bold atau italic, sedangkan XML dapat mendeklarasikan data terasosiasinya menjadi harga retail, pajak

penjualan, judul buku, atau data yang lainnya. Secara singkat perbandingan HTML vs XML :

1. HTML menyatakan bagaimana teks ditampilkan pada jendela browser sedangkan XML menyatakan apa arti dari tiap kata.
2. HTML memiliki sintaks yang lebih leluasa sedang XML lebih kaku.
3. Tag HTML dibatasi, sedangkan XML tidak.
4. Pencarian HTML mengembalikan sejumlah besar data dan dokumen, sedangkan XML ditujukan untuk mengembalikan data yang tepat yang diinginkan user.

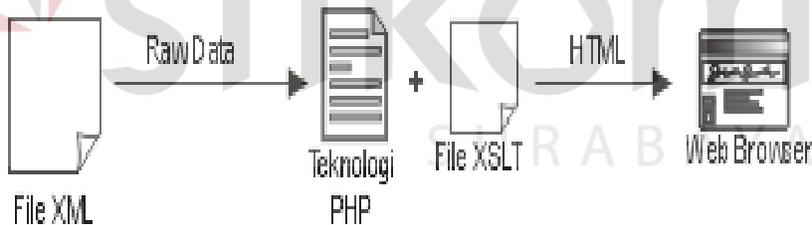
Tag-tag XML diadopsikan melalui organisasi Intranet dan juga melalui Internet, dan didalamnya akan ada kemampuan korespondensi untuk pencarian dan kemampuan manipulasi data aplikasi yang ditemukan. Suatu data telah dilokasikan, kemudian dapat dikirimkan melalui kabel dan direpresentasikan dalam browser seperti Internet Explorer (IE) versi 5 ke atas (yang sudah support XML) pada sistem operasi Microsoft Windows 9x/ME/NT/XP/2000. Dapat juga Anda pergunakan browser lainnya dengan lebih dulu memeriksa spesifikasi dukungannya terhadap XML. Sedangkan untuk tool editor dasar sebagai awalan, Anda cukup mempergunakan semacam Notepad atau bisa juga EditPlus, di samping editor-editor lain yang lebih baik.

Beberapa keuntungan dalam membuat dokumen menggunakan XML, antara lain :

1. XML bisa menyimpan dan mengatur semua jenis informasi sesuai dengan kebutuhan kita.

2. Sebagai sebuah standar yang terbuka, XML tidak terikat kepada perusahaan tertentu.
3. XML mendukung beragam jenis sistem penulisan dan simbol.
4. XML memberikan banyak cara untuk melakukan pemeriksaan kualitas dokumen.
5. Dengan sintaks yang jelas dan sederhana serta struktur yang tidak membingungkan, XML mudah dibaca dan diolah (parse) oleh manusia maupun program.
6. Dengan satu sumber dokumen, kita bisa menampilkan di mana saja sesuai dengan kebutuhan.

Penggunaan teknologi XML pada aplikasi TELKOMNet@Home, secara aliran data dapat digambarkan seperti ditunjukkan Gambar 2.2.



**Gambar 2.2** Aliran data XML.

File XML pada Gambar 2.2 berfungsi sebagai media penyimpanan data mentah yang memuat beberapa data transaksional, sehingga tidak semua data sistem akan ditransformasikan ke dalam file XML. Adapun jenis dan struktur data yang

ditransformasikan ke dalam file XML dapat diuraikan pada bagian yang menjelaskan ERD.

Karena sifatnya sebagai *backup* media penyimpanan data, keberadaan file XML tidak ditujukan untuk menggantikan fungsionalitas dari sistem database. Namun lebih ditujukan untuk backup penyimpanan data serta mendukung proses pengurangan *load traffic* sistem database yang sudah ada sebagai dampak dari adanya penambahan sistem baru pada sistem yang sudah ada.

**a. Struktur representasi data**

XML menyediakan struktur representasi data yang dapat diimplementasikan dan relatif lebih mudah untuk pengembangannya. Implementasi secara industri dalam komunitas Standard Generalized Markup Language (SGML) dan demonstrasi kualitas secara intrinsik dimanapun dan kemampuan secara industri dari struktur pohon format data yang dimiliki oleh XML.

XML merupakan subset dari SGML yang dioptimalkan untuk pengiriman melalui web. Didefinisikan oleh World Wide web Consortium (W3C), untuk memastikan struktur data mempunyai keseragaman dan aplikasi atau vendor yang independen. Hasil interoperabilitasnya adalah memulai generasi baru dari bisnis dan aplikasi e-commerce pada halaman web.

XML menyediakan standar data yang dapat mengkodekan content, semantik, dan skema untuk permasalahan variasi range yang luas, dari sederhana menjadi kompleks. XML dapat digunakan untuk menangani hal-hal berikut ini :

1. Document ordinary.
2. Struktur record, seperti record persetujuan atau purchase order.

3. Object dengan data dan metode, seperti form persistent dari object Java atau ActiveX control.
4. Record data, seperti hasil set dari query.
5. Meta-content tentang web site, seperti Channel definition format (CDF).
6. Skema entity standard dan tipenya.
7. Semua link antara informasi dan orang dalam web.

Saat data dalam desktop client, data dapat dimanipulasi, diedit, dan dipresentasikan dalam multiple views, tanpa mengembalikan nilai ke server. Kemudian server dapat lebih bermanfaat, menjadikannya sebuah komputer yang bersifat mendasar dan penggunaan bandwidth menjadi lebih sedikit. Data yang diubah menjadi format XML, dapat lebih mudah dimerged dari sumber-sumber yang berbeda.

XML bernilai tinggi dalam Internet, sebaik lingkungan Intranet perusahaan, karena XML menyediakan interoperabilitas penggunaan yang fleksibel, terbuka, format berbasis standard, dengan cara yang baru dari pengaksesan database yang legal dan pengiriman data ke web client. Aplikasi dapat dibangun lebih cepat, lebih mudah dimaintenance, dan dapat menyediakan multiple view dengan lebih mudah dalam struktur data.

#### **b. Separasi data presentasi XML**

XML sangat berguna karena memaintenance separasi user interface dari struktur datanya. *Hypertext Markup Language* (HTML) menspesifikasikan bagaimana menampilkan data dalam browser, sedangkan XML mendefinisikan contentnya. Sebagai contoh : dalam HTML kita menggunakan tag-tag untuk

memberitahukan browser untuk menampilkan data sebagai bold, italic, sedangkan dalam XML kita menggunakan tag-tag hanya untuk mendeskripsikan data, seperti kota, nama, temperature, dan tekanan barometrik.

Dalam XML, kita menggunakan stylesheet seperti *Exstensible Stylesheet Language (XSL)* atau *Cascading Style Sheets (CSS)* untuk mempresentasikan data dalam browser. XML menseparasikan data dari presentasi dan proses, mengijinkan kita untuk menampilkan dan memproses data seperti yang kita inginkan dengan menggunakan style sheet yang berbeda dan aplikasi yang berbeda pula.

Separasi data dari pengijinan presentasi integrasi data yang lebih sedikit dari berbagai sumber. Informasi Customer, order purchase, hasil penelitian, pembayaran tagihan, record medik, data catalog, dan informasi yang lainnya dapat dikonversikan kedalam XML dalam middle tier, pengijinan data untuk diubah secara online semudah HTML dalam menampilkan halaman-halaman web. Data dicodekan dalam XML kemudian dapat dikirimkan melalui web ke desktop. Tidak perlu retrofiting untuk penyimpanan informasi secara legal dalam database-database atau dokumen-dokumen mainframe, dan karena HTTP digunakan untuk mengirimkan XML melalui kabel, tidak ada perubahan yang diperlukan untuk fungsi tersebut.

Dokumen XML mudah untuk ditulis, terlebih lagi jika kita sudah biasa menggunakan HTML. Seperti contoh berikut ini, XML digunakan untuk mendeskripsikan laporan cuaca :

```
<weather-report>
```

```
<date>March 25, 1998</date>
```

```
<time>08:00</time>
```

```
<area>
  <city>Seattle</city>
  <state>WA</state>
  <region>West Coast</region>
  <country>USA</country>
</area>
<measurements>
  <skies>partly cloudy</skies>
  <temperature>46</temperature>
  <wind>
    <direction>SW</direction>
    <windspeed>6</windspeed>
  </wind>
  <h-index>51</h-index>
  <humidity>87</humidity>
  <visibility>10</visibility>
  <uv-index>1</uv-index>
</measurements>
</weather-report>
```

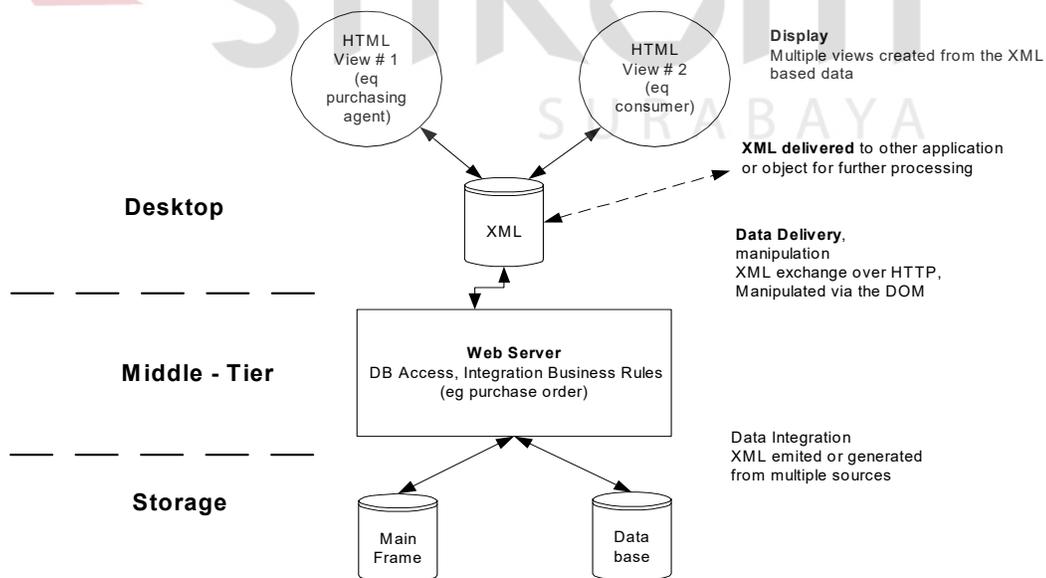
Kita dapat menampilkan data ini melalui berbagai macam cara, atau mengirimkan ke aplikasi yang lainnya untuk proses selanjutnya. Kita dapat juga menggunakan style sheet untuk menampilkannya dalam browser.

### c. Arsitektur XML

XML language, XML namespaces, dan DOM adalah rekomendasi dari W3C, merupakan situasi akhir dalam pengembangan dan proses persetujuan dalam W3C. Karena spesifikasi yang stabil ini, pengembang dapat memulai tag dan perubahan data mereka dalam format XML. XML menawarkan solusi yang bagus seperti arsitektur data dalam arsitektur three-tier.

XML dapat digenerate dari database yang sudah ada dengan menggunakan model three-tier yang scalable. Dengan XML, struktur data dimaintenance secara separasi dari aturan bisnis (atau proses) dan juga dari tampilannya. Seperti yang terlihat pada diagram berikut ini, proses yang ditekankan dari XML mencakup 4 bagian yaitu : Integrasi data, Pengiriman data, Manipulasi data, dan Menampilkan data.

Untuk lebih jelasnya arsitektur XML ditunjukkan pada Gambar 2.3.



**Gambar 2.3** Arsitektur XML

#### d. Integrasi data

XML namespaces memungkinkan pengembang untuk mengkualifikasikan nama elemen dalam sebuah tata cara yang dapat diakui untuk menghindari konflik antara elemen yang mempunyai kesamaan nama. Elemen-elemen yang direferensikan dalam dokumen tunggal, seperti purchase order, dapat didefinisikan dalam skema yang berbeda dalam web. Namespaces memastikan nama elemen tersebut tidak konflik dan mengklarifikasikan dengan yang aslinya, tapi tidak mendeterminasikan bagaimana proses elemen tersebut. Parser harus mengetahui arti elemen dan bagaimana memprosesnya.

Tag-tag multiple namespaces dapat digabungkan, tag yang sangat diperlukan dengan data yang datang dari sumber yang multiple melalui web. Dengan namespaces, semua elemen harus exist dalam dokumen berbasis XML yang sama tapi dalam hal ini dapat kembali pada dua (2) skema yang berbeda, kualifikasi semantik yang unik. Dalam hal ini, seperti terlihat pada bookstore purchase order, satu elemen title tidak dapat mencakup title buku, dan elemen title yang lainnya dapat mencakup title authornya.

W3C telah merilis XML namespaces sebagai rekomendasi, memungkinkan elemen menjadi subordinat dari URL. Kepastian ini merupakan nama yang unambigus sekalipun dipilih oleh multiple author. Hanya oleh seseorang yang dapat mempublikasikan halaman web mereka atau memperlihatkannya pada yang lainnya, fasilitas namespaces memungkinkan user untuk mendefinisikan kondisi kosa-kata yang bersifat private, atau menggunakan namespaces yang bersifat public dari kondisi yang biasa. Kode berikut ini memberitahukan kita bahwa jika sebuah elemen nama

dimulai dengan "dsig:" itu artinya didefinisikan oleh orang yang memiliki namespaces dari "http://www.dsig.org". Untuk lebih jelasnya perhatikan tag-tag berikut ini :

```
<orders xmlns:person=http://www.schemas.org/people
xmlns:dsig="http://dsig.org">
<order>
  <sold-to>
    <person:name>
      <person:first-name>Layanan</person:last-name>
      <person:first-name>Andrew</person:last-name>
    </person:name>
  </sold-to>
  <sold-on>1997-03-17</sold-on>
  <dsig:digital-signature>1234567890</dsig:digital-signature>
</order>
</orders>
```

Namespaces memastikan bahwa nama elemen tidak konflik, dan mengklarifikasikan yang mendefinisikannya. Mereka tidak memberikan instruksi tentang cara memproses elemen tersebut. Kita masih memerlukan untuk mengetahui arti dari elemen dan menentukan cara untuk memprosesnya.

Author dapat menspesifikasikan type data dari elemen (number, date, dan sebagainya) dan juga format dari content stringnya. Satu dapat menggunakan atribut

dari type data namespaces pada `''urn:schema-microsoft-com:datatypes''` untuk tujuan tersebut.

```
<sold-on dt:dt=''date''xmlns:dt=''urn:schemas-microsoft-com:datatypes''>1997—3-17</sold-on>
```

Dalam hal ini, `''date''` menspesifikasikan bahwa content elemen *sold-on* adalah date dalam format standar yang dispesifikasikan oleh namespaces `''data types''`. Seperti nama elemen, author juga dapat mendesign type datanya sendiri, dan juga menggunakan type yang dishare secara public. Microsoft bekerja bersama dengan W3C untuk mendefinisikan set standar dari type, dan telah menyediakan daftar inisial sebagai bagian dari XML schema yang didukung dalam Internet Explorer 5.

#### 1. Pengiriman data

Karena XML bersifat terbuka, dengan format berbasis text, maka dapat dikirim melalui HTTP dengan cara yang sama seperti HTML. Agent juga akan mendukung kemampuan untuk mengenerate update dari XML, yang dapat dikirim dalam berbagai cara untuk menginformasikannya pada client tentang perubahan data yang dibuat pada middle tier atau server database. Konsekwensinya, agent akan menerima update dari client dan mengirimkannya ke media penyimpanan server.

#### 2. Manipulasi data

Data sekarang yang terdapat pada desktop dapat dimanipulasi menggunakan DOM. DOM merupakan sebuah application programming Interface (API) yang mendefinisikan cara standard dimana pengembang dapat berinteraksi dengan elemen dari struktur tree XML. Object model mengontrol bagaimana user berkomunikasi

dengan struktur tree sebagai objects, yang dapat diakses secara programatik tanpa nilai balik ke server.

### 3. Menampilkan data

Microsoft XML parser dalam Internet Explorer 5 dapat membaca string data XML, memprosesnya, mengenerate struktur treenya, dan mengekso semua elemen data sebagai objects dengan menggunakan DOM. Parser menampilkan data ini menggunakan CSS atau XSL stylesheet, membuat data tersedia untuk proses manipulasi selanjutnya dengan menggunakan script, atau menanganinya dengan aplikasi lain atau object lain untuk pemrosesan selanjutnya. Serta mendukung Namespaces, type data, queries dan transformasi XSL, dan juga metode yang telah dikembangkan tersedia dalam DOM.

### 4. Menampilkan data berbasis XML pada halaman HTML

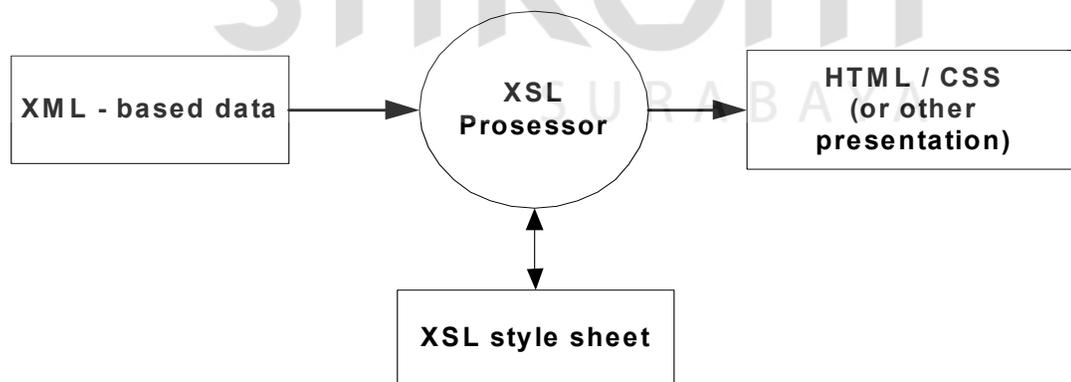
Dokumen XML tidak menspesifikasikan sendiri bagaimana informasinya ditampilkan. Data XML hanya mencakup kenyataannya (seperti siapa yang order buku pada harga tertentu). HTML merupakan bahasa yang ideal untuk mempresentasikan data kepada user. Sebagai contohnya, seorang karyawan dari toko buku online dapat mengunjungi halaman web untuk menemukan daftar entry order. Record data individual dikirimkan dalam XML, dipresentasikan kepada karyawan sebagai halaman HTML. Untuk mengkonstruksi halaman web ini, web server atau web browser harus mengkonversi record data XML kedalam presentasi HTML.

Mekanisme data binding dan style sheets dapat digunakan untuk menyusun data XML kedalam presentasi visual dan menambahkan interaktivitasnya. Data binding merupakan aspek dari Dynamic HTML (DHTML) yang memindahkan item

data individual dari sumber informasi (seperti dokumen XML) kedalam tampilan HTML, HTML digunakan sebagai template untuk menampilkan data XML.

XSL (Ekstensible Stylesheet Language) dapat menambahkan kemampuan yang lebih bagus pada proses tersebut. XSL style sheet mencakup instruksi menampilkan dokumen XML dan mentransformasikannya kedalam format yang lain, seperti HTML. Transformasi XML kedalam format lain dilakukan dengan cara deklaratif, biasanya membuatnya lebih mudah dan lebih dapat diakses dari pada melalui scripting.

Cascading Style Sheets dapat tetap digunakan untuk struktur data XML yang sederhana dan pada situasi tertentu sangat bermanfaat. CSS tidak menyediakan struktur tampilan yang menyimpang dari struktur dari sumber data. Dengan XSL memungkinkan mengenerate struktur presentasi yang sangat berbeda dari struktur data XML yang asli seperti ditunjukkan Gambar 2.4.



**Gambar 2.4** Struktur presentasi Data XML.

XSL menyediakan kedua semantik dan struktur independen dari content dan presentasinya (<http://www.xml.com>, *XML in Programming*)

## 2.4 XSL Transformation (XSLT)

XSLT adalah bagian dari XSL yang fungsinya untuk mengubah sebuah dokumen XML ke dokumen XML lain. XSLT mirip dengan mail merge. Jadi kita punya data, kemudian kita punya template. Lalu data dan template itu kita gabungkan, jadilah sebuah dokumen baru. Sebuah *XSL processor* (program) akan membaca data (dokumen XML) dan template (dokumen XSLT). Berdasarkan instruksi yang ditemukan oleh program pada dokumen XSLT, maka program akan menghasilkan sebuah dokumen XML baru. Dokumen XML baru itu bisa bermacam-macam, bahkan bisa berupa HTML. XSL merupakan bahasa turunan XML, jadi file XSL harus *well formed*.

XSLT akan memenuhi beberapa kebutuhan XML, yaitu :

1. Menampilkan dokumen XML, sehingga lebih enak dilihat, misalnya diubah ke HTML.
2. Mengubah file XML sehingga bisa sesuai antara DTD yang satu dengan DTD yang lain.
3. Mengurutkan dan menyaring data pada dokumen XML.

Transformasi bisa terjadi di :

1. *Web Browser* atau di komputer pemakai. Kedua dokumen (XML dan XSL), dari server akan dikirim ke *clien*, dan oleh *web browser* akan diolah sesuai aturan pada XSLT, untuk kemudian ditampilkan ke pemakai. Web browser yang sudah bisa melakukan hal ini adalah IE5.

2. *Server*. Sebuah server akan mengolah dokumen XML dan XSL, dan hasilnya akan dikirim ke pemakai. Saat ini sudah banyak aplikasi server yang bisa melakukan hal ini. Misalnya *Cocoon* dari Apache, dan *AxKit*.
3. Sebuah program *XSL processor*. Program ini akan mentransformasikan XML, sebelum diletakkan di server. Contohnya program yang namanya Saxon. Dengan perintah DOS, ubah XML ke HTML. Kemudian hasilnya kita letakkan ke web server.

Pada Gambar 2.2, file XSLT pada sistem aplikasi pendukung TELKOMNet@Home berfungsi sebagai template yang akan membentuk data mentah file XML menjadi file XML lain, dalam hal ini adalah file HTML. Dengan demikian, file XSLT memegang peranan penting dalam proses pengolahan data XML yang dibutuhkan untuk kemudian disajikan menjadi sebuah halaman web pada sisi client atau web browser.

Setelah data mentah dan template tampilan tersedia, bagian terpenting dalam penyajian informasi berikutnya adalah proses transformasi. Proses transformasi mencakup mekanisme perubahan data XML menjadi sebuah halaman web sesuai dengan template yang dibuat pada file XSLT. Proses ini dilakukan di sisi server dengan memanfaatkan fungsi-fungsi standar yang telah tersedia pada teknologi PHP.