

BAB 2 LANDASAN TEORI

2.1 XML dan HTML

XML singkatan dari *eXtensible Markup Language* yang sudah menjadi bagian penting bagi programmer yang akan mengembangkan web services. Hal ini karena XML dibangun dengan kemampuan melakukan transfer data antarplatform. XML juga memiliki kemampuan integrasi data disamping pertukaran data antar platform. Untuk web statis yang sederhana, programmer seringkali hanya menggunakan HTML untuk membangunnya. Sedangkan jika web tersebut kompleks ataupun membutuhkan konten yang dinamis, bahasa HTML akan dikombinasikan dengan bahasa pemrograman internet *server-side* seperti PHP, ASP, dan lain-lain. Namun setelah diparser oleh web server, program akan diubah menjadi HTML untuk dikirim ke browser client.

Bahasa XML tidak sama dengan HTML. HTML didesain untuk menampilkan data dan berfokus bagaimana data tersebut ditampilkan, sedangkan XML didesain untuk membawa data, mendeskripsikan data dan berfokus pada apakah data itu. Secara sederhana, XML yang mendefinisikan data dan HTML yang menampilkan data. Kesamaan dari kedua bahasa tersebut yakni sama-sama menggunakan tag.

Menurut Siswoutomo (2004), kelebihan XML antara lain :

1. Intelligence

XML dapat menangani berbagai level kompleksitas markup data yang bertingkat-tingkat.

2. **Adaptation**

Programmer dapat mengadaptasi untuk membuat bahasa FickaXML atau LindaXML. Tag dapat diberi nama sesuai dengan keinginan dan kebutuhan.

3. **Maintenance**

Mudah dalam pemeliharaan karena hanya berisi data dan markup.

4. **Linking**

XML dapat me-link satu atau lebih poin link didalam maupun diluar data.

5. **Simplicity**

XML lebih sederhana dibandingkan bahasa pemrograman lain.

6. **Portability**

XML mempunyai portabilitas yang bagus karena memisahkan antara data dan pengaturan tampilan.

2.2 XSD (XML Schema Definition)

Skema XML mendeskripsikan struktur dari dokumen XML dan sering disebut *XML Schema definition (XSD)*. Menurut Siswoutomo (2004), tujuan utama dari skema XML adalah mendefinisikan *legal building blocks* dari sebuah dokumen XML. Jadi, sebuah skema XML :

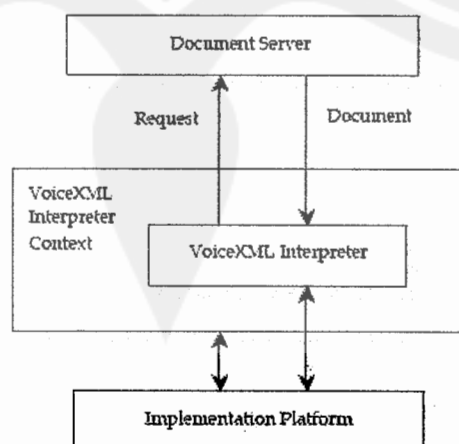
- Mendefinisikan elemen yang dapat muncul di dokumen.
- Mendefinisikan atribut yang dapat muncul di dokumen.
- Mendefinisikan elemen mana yang merupakan anak elemen.

- Mendefinisikan apakah elemen dapat berisi teks atau kosong.
- Mendefinisikan nilai default dan nilai tetap untuk elemen dan atribut.
- Mendefinisikan tipe data dari elemen maupun atribut.

2.3 VoiceXML

VoiceXML dikelola oleh vendor-vendor yang berkecimpung lama dalam dunia komputer antara lain Motorola, IBM, AT&T, Lucent dan Voxeo. Setiap perusahaan mempunyai solusi yang berbeda-beda pada pengembangan aplikasi berbasis suara menggunakan teknologi *voiceXML*.

VoiceXML dirancang untuk menciptakan dialog suara, pengenalan suara dari percakapan dan masukan DTMF tone. Tujuan dari *VoiceXML* adalah memberi peranan yang besar bagi pengembangan layanan web dan perantara layanan berbasis suara yang interaktif. Arsitektur model *VoiceXML* dapat dilihat pada gambar 2.1.



Gambar 2.1 Arsitektur Model VoiceXML

Arsitektur dari VoiceXML masih menggunakan standart aplikasi web yang sering dipakai. Seperti yang diketahui, dalam kasus aplikasi web, dokumen HTML berada pada dokumen server. Ketika user meminta suatu dokumen dari server, user meminta dengan menggunakan browser. Server akan me -response permintaan dari client dan mengirimkan ke client berdasarkan permintaan.

Seperti pada gambar 2.1, pada aplikasi VoiceXML, dokumen server disimpan pada web server. Terdapat pula voice server yang disebut *voice XML Interpreter server* yang didalamnya terdapat *voiceXML Interpreter Context* dan *VoiceXML Interpreter*. Voice server menangani proses inputan permintaan dari user berupa suara dan dan menyaluran permintaan dari user ke web server. Voice server juga menyediakan *voice-based interface* untuk user dengan membangkitkan pembicaraan menggunakan *text to speech (TTS)*. *VoiceXML documents* dihasilkan oleh web server sebagai *response* dari user yang akan diproses oleh *VoiceXML Interpreter Context*.

Pada *Implementation Platforms* akan dikendalikan oleh *VoiceXML Interpreter context* dan *VoiceXML Interpreter*. Implementasi nyata dari bagian *Implementation Platforms* membutuhkan software atau hardware yang mendukung *VoiceXML Interpreter*.

Menurut McGraw-Hill/Osborne (2002), tujuan dari pengembangan teknologi VoiceXML adalah menciptakan suatu aplikasi berbasis suara yang handal dan memanfaatkan perkembangan aplikasi web untuk menghantarkan layanan suara kepada user. Hal ini mengijinkan terjadinya integrasi antara layanan suara

dan layanan web. *Document server (web server)* mengelola servis yang disediakan, mengakses database dan mengelola dialog. *VoiceXML Document* menspesifikasikan setiap interaksi dialog yang terjadi dan dikendalikan oleh *VoiceXML Interpreter*.

2.4 Text To Speech (TTS)

Menurut Arman (Peneliti dan dosen di departemen Teknik Elektro ITB), sistem *Text to Speech* pada prinsipnya terdiri dari dua sub sistem, yaitu :

1. Bagian Konverter Teks ke Fonem (*Text to Phoneme*)

Berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi serta *pitch*-nya. Bagian ini bersifat sangat *language dependant*. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut.

2. Bagian Konverter Fonem to Ucapan (*Phoneme to Speech*)

Bagian ini akan menerima masukan berupa kode-kode fonem serta *pitch* dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke Ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan.

Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian konverter fonem ke ucapan. Dua teknik yang banyak digunakan adalah :

1. *formant synthesizer*

Bekerja berdasarkan suatu model matematis yang akan melakukan komputasi untuk menghasilkan sinyal ucapan yang diinginkan. Synthesizer jenis ini telah lama digunakan pada berbagai aplikasi. Walaupun dapat menghasilkan ucapan dengan tingkat kemudahan interpretasi yang baik, synthesizer ini tidak dapat menghasilkan ucapan dengan tingkat kealamian yang tinggi.

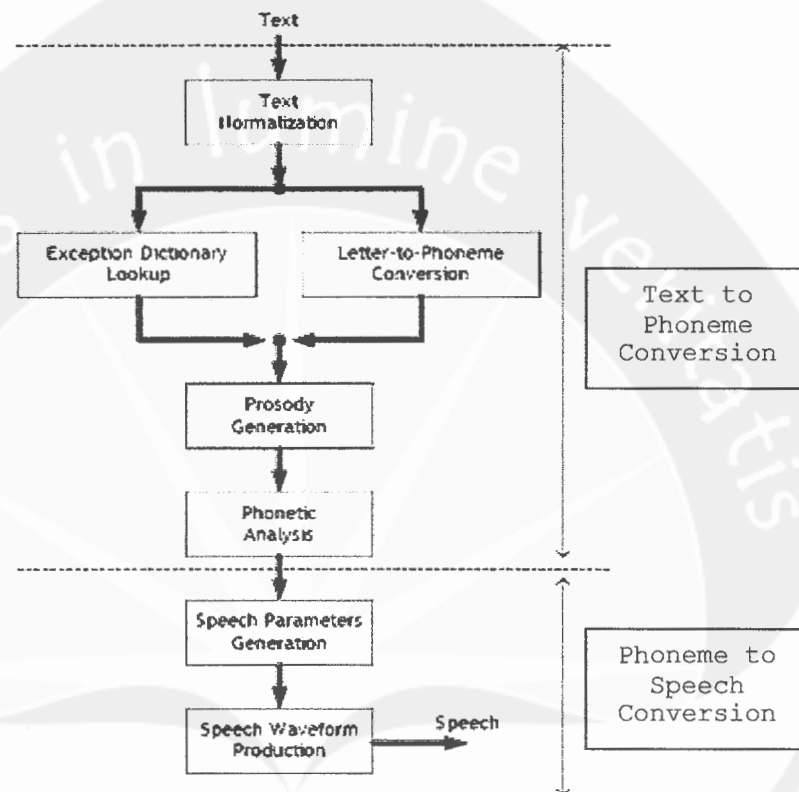
2. *diphone concatenation*

Synthesizer yang menggunakan teknik *diphone concatenation* bekerja dengan cara menggabungkan-gabungkan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa *diphone* (gabungan dua buah fonem). Synthesizer jenis ini dapat menghasilkan bunyi ucapan dengan tingkat kealamian (*naturalness*) yang tinggi.

Pada sistem yang menggunakan teknik *diphone concatenation*, sistem harus didukung oleh suatu *diphone database* yang berisi rekaman segmen-segmen ucapan yang berupa *diphone*. Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan *diphone database* yang berbeda.

Konversi dari teks ke fonem sangat dipengaruhi oleh aturan-aturan yang berlaku dalam suatu bahasa. Pada prinsipnya proses ini melakukan konversi dari simbol-simbol tekstual menjadi simbol-simbol fonetik yang merepresentasikan unit bunyi terkecil dalam suatu bahasa. Setiap bahasa memiliki aturan cara pembacaan dan cara pengucapan teks yang sangat spesifik. Hal ini

menyebabkan implementasi unit konverter teks ke fonem menjadi sangat spesifik terhadap suatu bahasa. Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dinyatakan pada gambar 2.2.



Gambar 2.2 Konversi Text ke Ucapan

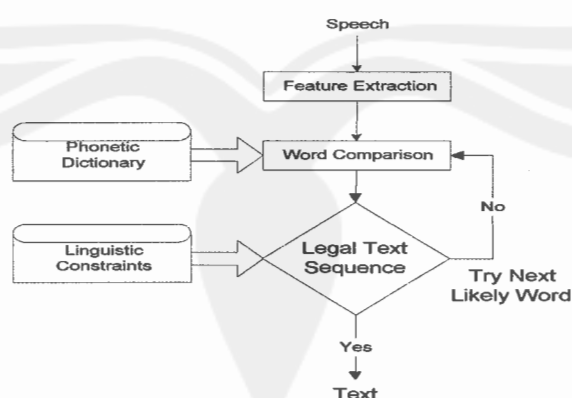
Dalam blok diagram pada gambar 2.2, kondisi yang masih dapat ditangani oleh aturan diimplementasikan dengan blok *Letter to Phoneme Conversion*. Konversi yang tidak teratur ditangani oleh bagian *Exception Dictionary Lookup*. Hasil dari tahap tersebut adalah rangkaian fonem yang merepresentasikan bunyi kalimat yang ingin diucapkan. Bagian *prosody generator* akan melengkapi setiap unit fonem yang dihasilkan dengan data durasi pengucapannya serta pitchnya. Data durasi serta *pitch* diperoleh berdasarkan kombinasi antara tabel atau database serta model prosodi. Secara

simbolik, hasil dari bagian ini sudah menghasilkan informasi yang cukup untuk menghasilkan ucapan yang diinginkan. Satu tahap berikutnya yang masih sering dilakukan adalah *Phonetic Analysis*. Tahap ini dapat dikatakan sebagai tahap penyempurnaan, yaitu melakukan perbaikan di tingkat bunyi.

2.5 ASR (*Automatic Speech Recognition*)

ASR (*Automatic Speech Recognition*) merupakan aplikasi *speech processing* yang menitikberatkan pada informasi dari suara dan bukan pada reproduksi suara. ASR menggunakan suatu "matching routine" untuk membandingkan informasi suara.

Teknologi *Speech Recognition* telah menjadi sangat populer saat ini. Dengan menggunakan fungsi inverse dari *speech synthesis*, ASR menggunakan suara sebagai inputnya dan keluarannya berupa text sebagai padanannya. Diagram blok dari ASR ditunjukkan seperti pada gambar 2.3.



Gambar 2.3 Blok diagram ASR (*Automatic Speech Recognition*)

Pada bagian *feature extraction*, pembicaraan dikenal oleh sistem dan dilanjutkan ke proses *word comparison* untuk perbandingan kata yang diucapkan oleh user.

Phonetic Dictionary digunakan untuk perbaikan bunyi yang diucapkan. Selanjutnya akan dihasilkan teks sesuai ucapan user.

2.6 Web Services

Web Services merupakan salah satu bentuk implementasi dari arsitektur model aplikasi N-Tier. Dalam implementasinya *web services* tidak mempunyai tampilan, karena *web services* memang termasuk dalam tier business services. Artinya di dalam *web services* hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh suatu aplikasi. Fungsi-fungsi yang mendukung business logic dari suatu sistem yang dibuat nantinya akan dapat digunakan dalam platform manapun, melalui bahasa pemrograman apapun. Menurut Siswoutomo (2004), berikut adalah definisi yang dikeluarkan oleh konsorsium W3C :

" A web services is a software system designed to supported interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web Services in a manner prescribed by its description using SOAP messages, typically confeyed using HTTP with an XML serialization in conjunction with other Web-related standards."

Kata kunci yang tidak kalah penting dalam definisi diatas adalah interaksi yang dapat dilakukan berbagai software dari berbagai mesin maupun paltform yang berbeda. Untuk dapat menjalankan fungsi tersebut, sebuah *web services* memerlukan agen. Agen adalah

potongan software atau hardware yang mengirim dan menerima pesan (*messages*). Agen tersebut dapat ditulis dengan berbagai bahasa pemrograman. Dalam web services dikenal pula istilah *providers* dan *requesters*. *Provider entity* adalah seseorang yang menyediakan agen untuk layanan fungsi tertentu. Sedangkan *Request entity* adalah seseorang yang menggunakan layanan fungsi tersebut. Agar dapat berkomunikasi, kedua entitas ini harus sepakat menggunakan mekanisme yang sama dalam pertukaran pesan. Mekanisme pertukaran pesan didokumentasikan di *Web Service Description (WSD)*. Platform yang digunakan dalam *web services* adalah XML dan HTTP.

Menurut Hadiwinata (2003), keuntungan menggunakan web services :

1. Platform Independent

Salah satu keunggulan web services adalah platform independent sehingga dapat digunakan oleh aplikasi yang berada dalam platform lainnya.

2. Reusable Function

Untuk implementasi dari aplikasi N-Tier ini, fungsi-fungsi yang terdapat didalam web services dapat digunakan berulang kali dan mudah untuk memeliharanya.

3. Simple Interaction with Database Platform

Web services dapat dijadikan interface antara aplikasi client dengan database server. Sehingga di komputer client, tidak perlu dilakukan proses instalasi dan konfigurasi ODBC Driver atau OLEDB Provider.

Menurut Hadiwinata (2003), kelemahan menggunakan web services :

1. Performance

Web services membutuhkan suatu infrastruktur jaringan yang cukup baik, dalam hal ini kita tidak dapat begitu saja mengimplementasikan web services dalam sistem yang tidak didukung dengan infrastruktur jaringan yang memadai.

2. Mekanisme callback

Web services tidak dapat digunakan untuk melakukan call back terhadap client. Artinya web services tidak dapat mengaktifkan method-method yang berada di client.

Menurut Siswoutomo (2004), secara umum hal-hal yang mendukung kualitas layanan web services adalah :

1. **Availability** adalah apakah web services tersebut selalu tersedia atau tidak. hal ini merepresentasikan probabilitas ketersediaan layanan.
2. **Accessibility** adalah aspek kualitas yang menunjukkan kemampuan web services melayani permintaan. Misalkan jika web services diakses oleh banyak peminta kinerjanya tetap stabil dan bagus atau tidak.
3. **Integrity** adalah aspek kualitas yang menunjukkan kemampuan web services melakukan penjagaan terhadap interaksi antarsumber.
4. **Performance** adalah aspek kecepatan pengiriman dan permintaan suatu layanan.

5. Security adalah aspek kualitas dari web services berkaitan dengan otentifikasi, enkripsi, dan akses kontrol.

2.7 SOAP (Simple Object Access Protocol)

Menurut Siswoutomo(2004), *SOAP (Simple Object Access Protocol)* adalah protokol untuk pertukaran informasi. Karena di dalam suatu web services mengandung fungsi-fungsi yang dapat digunakan oleh aplikasi lain, XML saja tidak cukup untuk mengimplementasikan web services. SOAP merupakan suatu dokumen XML yang mengatur bagaimana request dan response dari suatu web services akan bekerja. Agar mudah dipahami, dapat dikatakan bahwa :

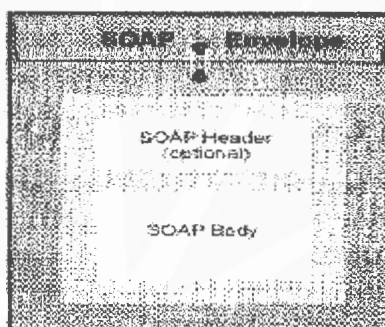
- SOAP untuk berkomunikasi antar aplikasi dan berbasis XML.
- SOAP adalah sebuah format untuk mengirim pesan.
- SOAP didesain untuk berkomunikasi melalui internet.
- SOAP adalah platform dan bahasa yang independen.

Sebuah pesan SOAP adalah dokumen XML yang berisi elemen-elemen berikut (lihat pada gambar 2.4) :

- *Envelope element* yang mengidentifikasi dokumen XML sebagai sebuah pesan SOAP. Elemen ini bersifat wajib karena elemen ini adalah elemen root dari pesan.
- *SOAP Header* yang berisi informasi header. Element ini bersifat opsional. Berisi informasi yang tidak ditampilkan di aplikasi.

Informasi tersebut dapat berupa informasi kontekstual yang berhubungan dengan pemrosesan pesan, waktu pemrosesan, otentifikasi, dan sebagainya.

- *SOAP Body* yang berisi panggilan dan merespon informasi. Elemen ini wajib karena berisi informasi yang dibawa oleh SOAP.
- *Fault Element* yang berisi pesan kesalahan yang terjadi pada waktu proses. Elemen ini opsional.

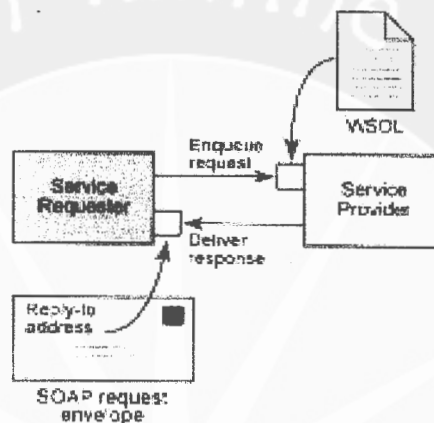


Gambar 2.4 Struktur dokumen SOAP

2.8 WSDL (Web Service Describe Language)

WSDL (Web Service Describe language) adalah format XML untuk menjelaskan web services. Menurut Siswoutomo (2004), WSDL merupakan suatu dokumen XML yang menjelaskan method-method apa saja yang tersedia dalam suatu web services, parameter apa saja yang diperlukan untuk memanggil suatu method, dan apa hasil dari method yang dipanggil. WSDL dapat dikatakan representasi kontrak antara *requestor* dan *provider-nya*. Dengan menggunakan WSDL, klien dapat memanfaatkan fungsi-fungsi publik yang disediakan server. Pada gambar 2.5 digambarkan skema pola pertukaran pesan (*message*

exchange) yang disebabkan oleh interaksi antara peminta (*requestor*) dan penyedia (*provider*) dan bersifat *request-response*. peminta (*requestor*) akan meminta dokumen WSDL kepada penyedia (*provider*), dan penyedia (*provider*) akan memberi nilai balikan atas response dari peminta (*requestor*) dengan memanfaatkan SOAP untuk pertukaran data.



Gambar 2.5 Mekanisme request-response WSDL

Struktur utama dari WSDL adalah :

- **Elemen definisi (*definitions element*)**

Sebagai root elemen. Mendefinisikan nama dari web service, mendeklarasikan namespace, dan semua deskripsi elemen.

- **Elemen tipe (*types element*)**

Mendeskrripsikan semua tipe data yang digunakan.

- **Elemen pesan (*message element*)**

Mendefinisikan nama dari pesan yang dapat merujuk ke parameter pesan atau nilai pesan kembali.

- **WSDL Port**

Menjelaskan interface yang diekspos oleh web service dan mendefinisikan web service, operasi yang dapat dijalankan dan pesan yang dapat terlibat.

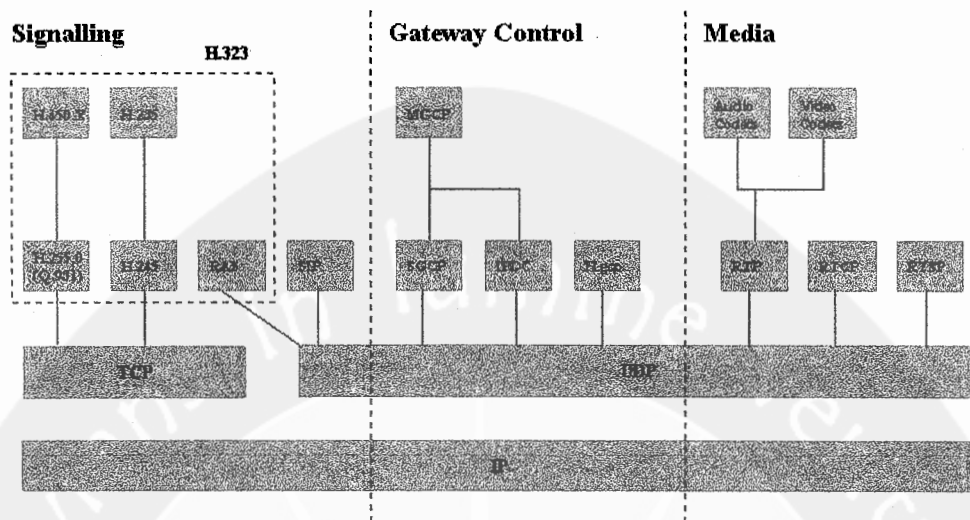
- **Elemen binding (*binding element*)**

Menjelaskan spesifikasi konkret bagaimana layanan akan diimplementasikan. WSDL menyertakan ekstensi yang built-in untuk mendefinisikan layanan SOAP. Informasi spesifik SOAP terdapat pada bagian ini.

- **Elemen service (*service element*)**

Mendefinisikan alamat untuk membangkitkan layanan. Pada umumnya menyertakan url untuk membangkitkan layanan SOAP.

2.9 Protokol Architecture



Gambar 2.6 Layer protocols

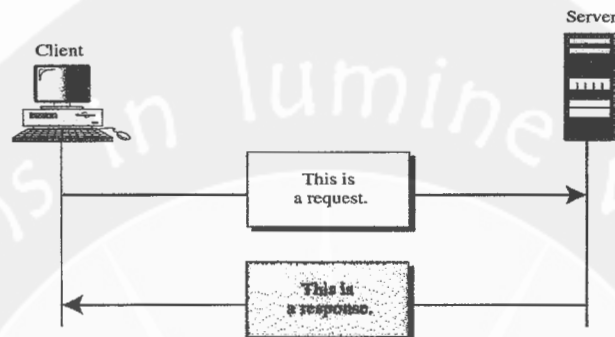
Protokol yang digunakan pada aplikasi VoiceXML (seperti pada gambar 2.6) sebagian besar bekerja pada *application layer protocol* dan *transport layer protocols*.

2.9.1 HTTP (Hypertext Transfer Protocol)

HTTP adalah protokol dasar *World Wide Web (WWW)* dan bisa digunakan dalam aplikasi *client/server* yang melibatkan *hypertext* (William Stallings, 2002). Arti nama ini sebagaimana salah satu kegunaan dalam HTTP tersebut bukanlah protokol untuk mentransfer *hypertext*, melainkan protokol untuk mentransmisikan informasi dengan tingkat efisiensi yang diperlukan untuk membuat lompatan *hypertext*. Data-data yang ditransfer oleh protokol ini bisa berupa teks asli, *hypertext*, audio, gambar, atau informasi-informasi lainnya yang bisa diakses melalui *internet*.

HTTP adalah protokol *client/server* yang berorientasi transaksi. Untuk menyediakan kehandalan,

HTTP menggunakan TCP dengan nomor port standart 80. HTTP merupakan protokol level aplikasi dengan kemudahan dan kecepatan penyaluran data dengan menggunakan konsep *request-response* pada client dan server seperti yang ditunjukkan pada gambar 2.8.



Gambar 2.7 HTTP

2.9.2 SIP (Session Initiation Protocol)

SIP (session initiation protocol) diterbitkan sebagai standar oleh IETF (RFC 3261 dan RFC 2543) setelah adanya VoIP. SIP disiapkan sebagai protokol dalam suite IP untuk membentuk dan melakukan pengendalian atas sesi multimedia over IP. SIP menggunakan protokol UDP dengan port 5060. Bentuknya berupa teks. Pengalamatan SIP dapat dilakukan mirip nomor telepon atau mirip alamat web. Jika pengalamatan dilakukan mirip web, digunakan juga URL seperti web, yang lebih lanjut akan diterjemahkan menjadi alamat IP oleh suatu DNS.

Pengalamatan SIP mengikuti format :

[SIP:]<username@host>

SIP bukan media transfer protocol, sehingga SIP tidak membawa paket suara atau video. SIP memanfaatkan RTP (Real Time Protocol) untuk media transfer.

Menurut Gunawan dan Haryadi (Departemen Teknik Elektro - ITB), pada jaringan SIP ada 2 komponen yang selalu ada untuk melakukan komunikasi yaitu *user agent client* dan *server*.

User agent adalah komponen SIP yang memulai, menerima dan menutup sesi komunikasi. *User agent* dapat berupa software (*softphone*) ataupun hardware (*hardphone*). Contoh dari *softphone* SIP adalah *SIP SoftPhone* yang digunakan dalam SIPERMAYA.

User Agent terdiri dari 2 komponen utama yaitu:

a. User Agent Client (UAC)

Komponen yang memulai sesi komunikasi.

b. User Agent Server (UAS)

Komponen yang menerima atau menanggapi sesi komunikasi.

Pada SIP terdiri dari 3 komponen penting yaitu

a. *Proxy server*

Proxy server adalah *server* yang menerima *request*, mengolahnya, serta meneruskan *request* yang diterimanya ke *next hop server* atau ke *end user* setelah mengubah beberapa *header* pada pesan *request*.

b. *Redirect server*

Komponen ini merupakan *server* yang menerima pesan *request* serta memberikan respon terhadap *request* tersebut yang berisi alamat dari *next hop server*.

c. *Registrar*

Komponen yang menerima *request message Register*. Komponen ini menambahkan fungsi otentifikasi user untuk validasi. Registrar menyimpan database user untuk otentifikasi dan lokasi sebenarnya (berupa

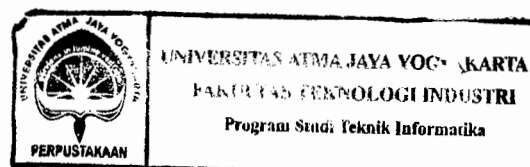
IP dan port) agar user yang terdaftar dapat dihubungi oleh komponen lainnya. Komponen ini biasanya disandingkan dengan Proxy Server.

2.9.3 RTP (Real Time Protocol)

RTP diterbitkan sebagai standar oleh IETF (RFC 1889) menyediakan fungsi pengangkutan jaringan untuk aplikasi yang akan menyalurkan data secara realtime, seperti audio, video atau data simulasi. Menurut Sugiantoro, RTP tidak menunjuk reservasi sumber daya yang digunakan dan tidak menjamin quality-of-service untuk data realtime. Pengangkutan data yang terjadi dikendalikan oleh suatu protokol kendali (RTCP) untuk mengijinkan monitoring data dan untuk menyediakan kemampuan identifikasi dan kendali minimal.

RTP terdiri dari dua bagian, header dan data. Bagian data merupakan satu protokol tersendiri yang digunakan untuk mendukung fungsi dari aplikasi yang digunakan. Bagian header RTP berukuran cukup besar, sebagai contoh apabila digabungkan dengan header IP dan UDP, ukuran header RTP dapat berukuran 40 byte. Untuk meningkatkan efisiensi penggunaan bandwidth pada transmisi data menggunakan RTP, maka dikembangkan Compressed Real Time Protocol (CRTP). Protokol ini mampu melakukan kompresi header RTP/UDP/IP dari 40 byte hingga berukuran 2 sampai dengan 5 byte saja.

RTP dapat menggunakan berbagai *port* sembarang mulai dari 1024 sampai 65,5534.



2.9.4 RTCP (Real Time Control Protocol)

RTCP (Real Time Control Protocol) merupakan protokol kendali RTP untuk bertukar informasi kendali dan status antara pengirim dan penerima. RTCP dienkapsulasi oleh UDP. RTCP merupakan protokol QoS(Quality of Services) untuk menjamin kualitas streaming data. Dan bertugas memberikan feedback atas kualitas distribusi data audio dan video.

2.9.5 RTSP (Real Time Streaming Protocol)

RTSP (Real Time Streaming Protocol) adalah protokol level aplikasi untuk kendali pengiriman data dengan menggunakan perangkat *real-time* seperti audio dan video. Protokol ini menyediakan mekanisme yang akan digunakan pada RTP (Real Time Protocol).

2.9.6 MRCP (Media Resource Control Protocol)

MRCP (Media Resource Control Protocol) merupakan media sumber layanan yang akan mengelola layanan suara seperti *speech synthesizers (mesin ASR)* dan *speech recognition (mesin TTS)* pada jaringan yang didesain untuk bekerja dengan streaming data pada SIP (Session Initiation Protocol). MRCP akan menjaga konektivitas ke media pengantaran pada RTP (Real Time Protocol). Protokol MRCP mendefinisikan request, response, dan event yang digunakan untuk mengatur media sumber layanan yang ada. MRCP mengelola implementasi tanggapan suara yang interaktif pada *VoiceXML interpreter*.

Client dari aplikasi VoiceXML memerlukan media suara yang telah diproses dan dihasilkan oleh *VoiceXML*

Interpreter dan diatur dengan protokol kendali RTSP maupun RTP. Pesan MRCP digunakan untuk mengendalikan sumber daya yang dibutuhkan dalam pemrosesan *streaming* pada audio dan mengacu pada alamat URL layanan suara.

2.9.7 UDP (User Datagram Protocol)

UDP (User Datagram Protocol) merupakan salah satu protokol transport selain TCP yang umum digunakan sebagai bagian dari suite protokol IP yang ditetapkan dalam RFC 768. Pada dasarnya UDP adalah layanan yang tidak handal karena perlindungan duplikasi dan pengiriman tidak terjamin. Menurut William Stallings (2001), berikut adalah perbandingan antara TCP dan UDP antara lain :

| TCP | UDP |
|--|---|
| Connection-oriented | Connectionless |
| Reliable <ul style="list-style-type: none"> - Memastikan paket sampai tujuan - Pengurutan ulang paket di sisi penerima - Tidak terjadi duplikasi - Pengiriman ulang jika paket hilang/rusak | Unreliable <ul style="list-style-type: none"> - Tidak menjamin sampai tujuan - Tidak melakukan pengurutan ulang - Mungkin terjadi duplikasi - Paket mungkin hilang/rusak |
| Bandwidth overhead lebih besar | Bandwidth overhead lebih kecil |
| Melakukan flow control | Tidak melakukan flow control |
| Tidak mendukung broadcast | Mendukung broadcast |
| Stream | Datagram |
| Digunakan untuk transmisi yang membutuhkan kehandalan | Digunakan untuk transmisi yang berorientasi waktu atau pengiriman pesan secara periodik |

Header UDP mencakup sebuah port sumber dan port tujuan. Bidang panjang berisikan seluruh segmen UDP, termasuk header dan data. Sedangkan checksum adalah algoritma yang sama seperti algoritma seperti yang digunakan untuk TCP dan IP. Untuk UDP, checksum diterapkan untuk segmen UDP secara keseluruhan ditambah dengan pseudoheader yang dipersiapkan untuk header TCP pada saat penghitungan dan merupakan pseudoheader yang sama seperti yang digunakan pada TCP. Bila suatu kesalahan terdeteksi, segmen dibuang dan tidak ada tindakan lebih jauh yang diambil. Bidang checksum didalam UDP bersifat pilihan. Bila tidak digunakan, checksum diset ke nol. Jadi, bila tidak ada perhitungan checksum yang ditampilkan UDP, maka tidak ada pengecekan bagi data.

2.10 Circuit-Switching pada jaringan telepon

Circuit switching dikembangkan sedemikian rupa untuk mengendalikan lalu lintas suara, namun sekarang juga dikembangkan untuk lalu lintas data. Contoh yang paling dikenal dalam hal jaringan circuit switching adalah jaringan telepon umum. Ini sebenarnya merupakan kumpulan jaringan nasional yang saling dihubungkan untuk membentuk layanan internasional. Meskipun awalnya dirancang dan diimplementasikan untuk melayani pelanggan telepon analog, yang mengendalikan lalu lintas data secara substansial melalui modem dan secara bertahap dikonversikan menjadi sebuah jaringan digital. Aplikasi yang sudah dikenal lainnya dalam hal circuit switching adalah Private Branch Exchange (PBX), yang

digunakan untuk interkoneksi telepon didalam bangunan gedung atau kantor.

Menurut William Stallings (2001), dalam suatu circuit switching, terdapat sinyal-sinyal kontrol yang berfungsi mengatur jaringan dan menetapkan panggilan serta menghentikan panggilan. Pengaturan panggilan dan pengaturan jaringan secara keseluruhan memerlukan hal-hal dimana informasi dipindahkan diantara pesawat dan switch, diantara switch itu sendiri, serta diantara switch dan pusat manajemen jaringan.

2.11 Tools dan Teknologi yang Digunakan pada Web Based

2.11.1 Teknologi .NET

Microsoft Visual Studio .NET adalah sebuah platform untuk membangun, menjalankan dan meningkatkan generasi lanjut dari aplikasi terdistribusi. *Microsoft Visual Studio .NET* memperluas *client*, *server* dan *service-service* yang terdiri atas:

1. Sebuah model pemrograman yang memungkinkan *developer* membangun aplikasi dan layanan web XML.
2. Sekumpulan XML Web services seperti *Microsoft .NET My Services*, yang membantu *developer* menghasilkan aplikasi yang sederhana dan terpadu.
3. *Tool-tool* pengembang yang menyediakan IDE (*Integrated Development Environment*) untuk memaksimalkan produktivitas pengembangan menggunakan *.NET Framework*.

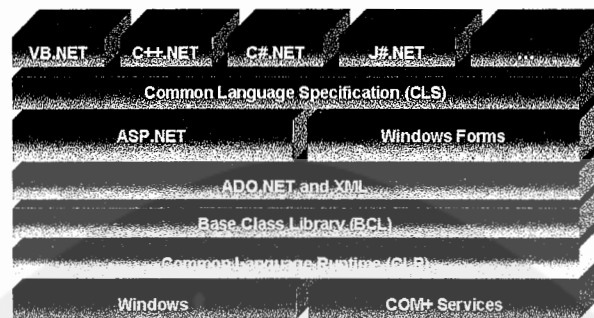
2.11.2 Framework .NET

Framework .NET adalah lingkungan untuk membangun, menyebarkan/*deploying*, dan menjalankan aplikasi dan layanan berbasis web. *.Net Framework* disusun oleh dua komponen utama, yaitu *Common Language Runtime (CLR)* dan *.NET Framework Class Library* termasuk aplikasi *Console*, *Windows GUI*, *ASP.NET*, layanan web *XML* dan layanan *Windows*.

Visual Studio.NET dibangun menggunakan fondasi *.NET Framework*. *.NET Framework* menyediakan lingkungan yang cerdas, mudah dikembangkan untuk membangun, menyebarkan dan menjalankan aplikasi dan layanan web *XML* yang terdistribusi. Dalam istilah yang mudah, *.NET Framework* memisahkan *platform* sistem operasi menjadi dua lapisan, yaitu lapisan pemrograman dan lapisan eksekusi.

Tujuan dari *.NET Framework* adalah :

1. Menyediakan lingkungan pemrograman berorientasi objek, apakah kode objek disimpan dan dijalankan secara lokal, dijalankan secara lokal tetapi disebarluaskan melalui internet atau dijalankan secara *remote* (dijalankan dari suatu tempat).
2. Menyediakan lingkungan untuk menjalankan suatu kode yang menjamin keamanan saat kode dijalankan.
3. Menyatukan model-model pemrograman dengan didukung oleh banyak bahasa dan membuat berbagai tipe aplikasi.



Gambar 2.8 Arsitektur .NET Framework

2.11.3 Visual C#.NET

C# adalah bahasa pemrograman baru yang diciptakan oleh *Microsoft*, dikembangkan dibawah kepemimpinan Anders Hejlsberg. Latar belakang Anders kemudian disatukan dengan aspek teknologi C++ yang telah ada, sehingga hadirlah C# sebuah bahasa yang canggih.

Definisi C# dari *Microsoft* adalah sebagai berikut :

"C# is a simple, modern, object oriented and type-safe programming language derived from C and C++. C# is firmly planted in the C and C++ family tree of languages, and will immediately be familiar to C and C++ programmers. C# aim to combine the high productivity of Visual Basic and the raw power of C++"

Ada dua kata kunci pada definisi diatas yakni *high productivity* dan *raw power C++*. C# hadir bukan saja sebagai official language bagi .NET, dia juga hadir sebagai alternatif bagi para pengembang yang berasal dari C, C++ atau bahkan Java. Sehingga pengembang dapat dengan mudah mengembangkan suatu solusi sistem informasi dengan konsep dan paradigma yang modern dibawah naungan teknologi .NET.

2.11.4 ASP.NET

Menurut Ario Suryo Kusumo (2004), ASP.NET adalah kumpulan teknologi dalam .NET Framework untuk membangun aplikasi Web dinamik dan Layanan Web XML. Dengan menggunakan Active Server Pages yang sudah ada, kode server-side digabungkan dan diselingi dengan client-side HTML. Halaman ASP.NET dijalankan di server dan akan dibuat halaman *markup* seperti HTML atau XML yang akan dikirim ke browser pada sisi client. Terdapat 3 macam kontrol server yang terdapat pada ASP.NET, yaitu :

1. Kontrol Server HTML

Kontrol server HTML adalah elemen HTML biasa yang ditambah dengan atribut "*runat=sever*". Programmer dapat memanipulasi kontrol ini pada sisi server. Setelah menampilkan form ke client, mesin ASP.NET akan mengkonversinya menjadi halaman HTML. Kontrol server HTML mirip dengan elemen HTML ketika membuat aplikasi web.

2. Kontrol Server Web

Kontrol server web mirip dengan kontrol HTML tetapi dilengkapi dengan kontrol tingkat tinggi dan *programmability* (memiliki kemampuan untuk diprogram). Pada kontrol Web terdapat kontrol form tradisional seperti label dan textbox, juga kontrol yang kompleks seperti tables.

3. Kontrol Kustom

Kontrol kustom digunakan untuk menambah fungsionalitas dan memperluas kontrol bagi suatu aplikasi sesuai dengan kebutuhan programmer.

Menurut Joe Martin dan Brett Tomson (2004) dalam bukunya dikatakan bahwa ASP.NET mewakili evolusi terakhir dari pengembangan teknologi web. Perubahan terbesar dari ASP.NET adalah aplikasi kode yang sekarang merupakan kode yang telah di-*compile* dan dapat langsung dieksekusi. Karena ASP.NET didasarkan pada Common language Runtime, maka komponen .NET platform telah disediakan bagi pengembang aplikasi web. Sejumlah fitur telah tersedia dengan komponennya yang membantu pengembang web untuk menggabungkannya ke dalam aplikasi.

2.11.5 SQL Server 2000

Basis data adalah tempat penyimpanan data. Untuk bekerja dengan basis data, kita harus memakai sebuah bahasa. Bahasa basis data yang paling banyak dipakai adalah SQL (*Structured Query Language*).

Microsoft SQL Server 2000 adalah sistem manajemen basis data yang memakai perintah-perintah *Transact-SQL* untuk mengirim perintah dari komputer klien ke komputer server. *Transact-SQL* adalah bahasa SQL yang dikembangkan oleh Microsoft dengan menambahkan dialek tertentu. Microsoft SQL Server 2000 berisi basis data, mesin basis data, dan aplikasi yang diperlukan untuk mengelola data dan komponen-komponennya.