

BAB II

LANDASAN TEORI

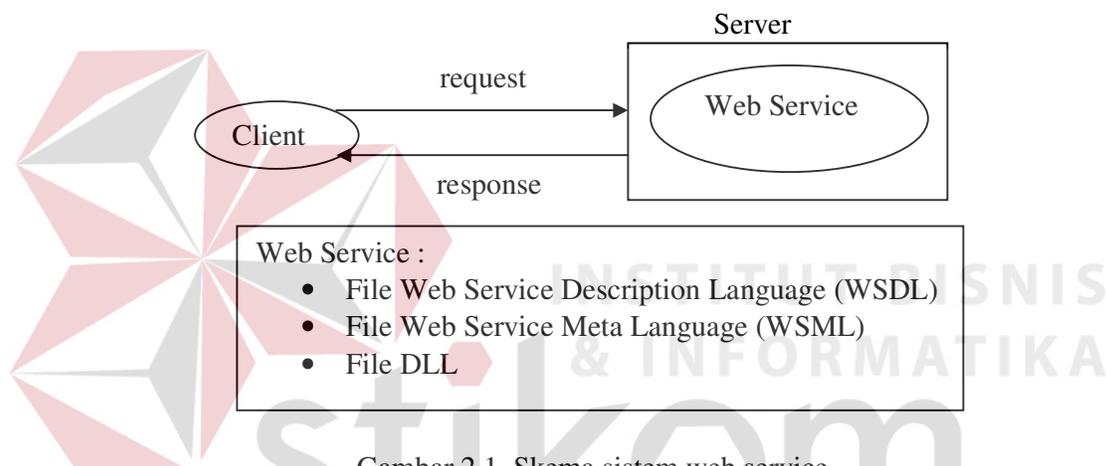
2.1 Web Service

Web Service merupakan kumpulan aplikasi logika yang menyediakan data dan service bagi aplikasi-aplikasi yang lain (Danny Ryan dan Tommy Ryan, 2002). Adapun aplikasi terdistribusi tersebut dapat diakses oleh aplikasi-aplikasi client tanpa memperhatikan sistem operasi maupun bahasa pemrograman.

Sebelum adanya web service terdapat teknologi CORBA dari OMG yang menggunakan bahasa Java dan DCOM dari Microsoft. Kekurangan yang dimiliki oleh kedua teknologi ini adalah program yang akan dipakai untuk mengakses komponen tersebut harus dibuat dengan bahasa yang sama dengan bahasa yang dipakai untuk membuat komponen tersebut untuk CORBA dan untuk DCOM, kecuali jika bisa di platform Microsoft.

Service yang disediakan oleh komponen Web Service umumnya berupa operasi-operasi logika maupun operasi query yang dimanfaatkan oleh banyak client (orang/program lain). Service tersebut dapat dimanfaatkan secara langsung dan juga dapat dimanfaatkan oleh web service lain. Sebagai contoh web service yang menangani operasi perkalian dapat dimanfaatkan secara langsung yaitu program client dapat langsung memanggil web service tersebut, begitu juga dengan web service pembagian/penjumlahan/pengurangan dan web service – web service tersebut dapat pula dimanfaatkan oleh web service yang lain misalnya web service kalkulator dimana program client memberikan inputan (berisi operasi matematika seperti perkalian dan lainnya) kepada web service kalkulator dan web

service ini akan memanggil web service yang menangani operasi-operasi matematika (web service perkalian, web service pembagian, web service penjumlahan serta web service pengurangan) yang sesuai dengan inputan yang diberikan oleh program client. Program client yang memanfaatkan layanan tersebut dapat dibuat menggunakan bahasa pemrograman yang berbeda selama mempunyai standard komunikasi yang sama dengan komponen web service tersebut.



Gambar 2.1 Skema sistem web service

Keterangan skema :

1. Program client me-request service dengan cara memanggil fungsi/method yang terdapat dalam web service dan dalam melakukan request dapat hanya dengan memanggil nama service ataupun menambahkan parameter yang dibutuhkan oleh web service tersebut. Program client dapat dibuat dengan bahasa pemrograman apa saja (tidak harus sama dengan bahasa pemrograman untuk web service) dan juga dapat berupa aplikasi berbasis web maupun aplikasi yang tidak berbasis web. Program client hanya berupa aplikasi yang

berisi user interface untuk menerima input dan output serta proses logika untuk koneksi ke web service.

2. Web Service akan memberikan response sesuai dengan method yang dipanggil oleh program client.

2.2 WSDL (Web Service Description Language)

WSDL adalah bahasa berbasis XML untuk mendiskripsikan service-service yang disediakan oleh server serta mengatur operasi dari tiap-tiap service yang disediakan (Microsoft, 2002). Untuk tiap-tiap operasi, WSDL juga mendiskripsikan format pesan yang harus digunakan client dalam melakukan request sebuah operasi yang ada pada komponen web service. Adapun elemen-elemen yang ada pada WSDL antara lain :

- Elemen Message

Elemen <message> dalam sebuah file WSDL mendeskripsikan isi logika dari pesan-pesan yang saling berhubungan. Tiap-tiap elemen <message> terdiri dari bagian-bagian logika yang dideskripsikan dalam elemen anak (child element) <part>. Tiap-tiap element <part> mempunyai atribut name dan type yang menspesifikasikan nama dari elemen <part> itu sendiri dan tipe datanya. Jika sebuah elemen <message> mempunyai banyak bagian logika maka perlu dispesifikasikan banyak elemen <part> untuk masing-masing bagian logika.

Berikut adalah contoh elemen <message> untuk operasi penjumlahan :

```
<message name='AddNumbers'>
  <part name='NumberOne' type='double' />
  <part name='NumberTwo' type='double' />
</message>
<message name='AddNumbersResponse'>
  <part name='Result' type='double' />
```

```
</message>
```

- Elemen Port Type

Elemen <portType> mengidentifikasi kumpulan operasi, dimana masing-masing operasi dipisahkan elemen anak <operation>. Contoh elemen <portType> :

```
<portType name='CalcPortType'>
  <operation name='AddNumbers'
    parameterOrder='NumberOne NumberTwo'>
    <input message='AddNumbers' />
    <output message='AddNumbersResponse' />
  </operation>
</portType>
```

Untuk tiap-tiap operasi yang ada melakukan proses request dan response. Spesifikasi format pesan untuk melakukan request menggunakan elemen <input>, sedangkan elemen <output> menspesifikasikan format pesan untuk response.

- Elemen Binding

Untuk setiap elemen <portType> yang mengidentifikasi sebuah operasi dihubungkan dengan elemen <binding> yang menyediakan detail protokol operasi tersebut. Sebagai contoh elemen <binding> :

```
<binding name='CalcBinding' type='CalcPortType'>
  <stk:binding preferredEncoding='UTF-8' />
  <soap:binding style='rpc'
    transport='http://schemas.xmlsoap.org/soap/http' />
  <operation name='AddNumbers'>
    <soap:operation soapAction='http://tempuri.org/Doc1/action/
      Doc.AddNumbers' />
    <input>
      <soap:body use='encoded' namespace='http://tempuri.org/Doc1/' message/'
        encodingStyle='http://schemas.xmlsoap.org/soap/' encoding/'
        parts='NumberOne NumberTwo' />
    </input>
  <output>
```

```

<soap:body use='encoded' namespace='http://tempuri.org/Doc1/
message/' encodingStyle='http://schemas.xmlsoap.org/soap/ encoding/'
parts='Result' />
</output>
</operation>
</binding>

```

Setiap elemen <binding> mempunyai atribut name dan type. Atribut name menyediakan nama khusus untuk binding sedangkan atribut type menyediakan nama dari <portType>. Elemen <binding> mempunyai tiga elemen anak, yaitu :

- Elemen <stk:binding>

Atribut preferredEncoding menentukan pengkodean karakter yang digunakan.

- Elemen <soap:binding>

Elemen anak ini untuk mendefinisikan detail protokol menggunakan atribut style dan atribut transport .

- Elemen <operation>

Elemen ini mempunyai atribut name yang merupakan nama dari operasi yang ada di elemen <portType> yang diidentifikasi dalam atribut type pada elemen <binding>. Di bawah berikut merupakan elemen anak dari elemen ini :

- Elemen <soap:operation>

Elemen ini mempunyai atribut soapAction yang mendefinisikan operasi yang dipakai.

- Elemen <input> dan <output>

Kedua elemen ini mengidentifikasi proses input maupun output dari sebuah operasi.

- Service

Ketika client melakukan request sebuah operasi dari server, maka harus mengidentifikasi service, port dari service dan operasi yang akan dieksekusi. Client menggunakan elemen <service> untuk menyediakan informasi ini. Setiap elemen <service> berisikan elemen anak <port> dan elemen <port> mempunyai dua atribut dan satu elemen anak. Atribut name menyediakan nama khusus dari port yang telah terdefinisi, sedangkan atribut binding menunjuk pada binding yang telah dispesifikasikan sebelumnya pada elemen <binding>. Contoh elemen service :

```
<service name='Calc'>
  <port name='CalcPortType' binding='Calcbinding'>
    <soap:address location='http://localhost/Doc1/Sample1.WSDL' />
  </port>
</service>
```

- Types

Elemen <types> ini digunakan sebagai penampung untuk mendefinisikan berbagai tipe data yang akan digunakan dalam pertukaran pesan. Contoh :

```
<types>
<schema targetNamespace='http://tempuri.org/type'
  xmlns='http://www.w3.org/2001/XMLSchema'
  xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
  xmlns:wSDL='http://schemas.xmlsoap.org/wSDL/'>
</schema>
</types>
```

Operasi-operasi pada file WSDL dapat berorientasi “dokumen” atau berorientasi “remote procedure call(RPC)”, yang didefinisikan oleh atribut style pada elemen <soap:binding> file tersebut.

- Operasi berorientasi document/dokumen

Jika sebuah operasi pada file WSDL berorientasi dokumen maka pesan masukan (request) dan keluaran (response) dispesifikasikan pada operasi yang berisi dokumen XML. Sebagai contoh, sebuah file WSDL mendefinisikan sebuah operasi SubmitPO dimana operasi ini mengharapkan dokumen purchase order (dokumen XML) sebagai bagian dari pesan inputan dan mengembalikan dokumen XML yang lain sebagai response.

- Operasi berorientasi RPC

Operasi berorientasi RPC mempunyai pesan masukan yang berisi operasi parameter dan pesan keluaran yang berisi operasi hasil. Sebagai contoh, pesan masukan untuk operasi AddNumbers didefinisikan pada file WSDL merupakan parameter yang berisi metode AddNumber pada server, dan pesan keluaran mengembalikan hasil dari operasi penjumlahan.

2.3 WSML (Web Service Meta Language)

WSML merupakan bahasa untuk menyediakan informasi yang memetakan operasi pada sebuah service yang telah dideskripsikan pada file WSDL pada metode sebuah obyek yang ada di server (Microsoft, 2002). Berikut contoh sebuah document WSML :

```
<servicemapping name='xCalc'>
  <service name='Calc'>
    <using PROGID='Sample1' cachable='0' ID='Sample1Object' />
  <port name='CalcPortType'>
    <operation name='AddNumbers'>
      <execute uses='CalcObject' method='AddNumbers' dispID
        = '1610809345'>
        <parameter callIndex='-1' name='retval' elementName='Result' />
        <parameter callIndex='1' name='NumberOne' elementName
          = 'NumberOne' />
      </execute>
    </operation>
  </port>
</service>
</servicemapping>
```

```

        <parameter callIndex='2' name='NumberTwo' elementName
        = 'NumberTwo' />
    </execute>
</operation>
</port>
</service>
</servicemapping>

```

Elemen <servicemapping> digunakan untuk ‘membungkus’ isi dari WSMML. Elemen ini mempunyai elemen anak <service> untuk spesifikasi pemetaan. Elemen <service> ini sama dengan yang ada pada file WSDL yang terhubung. Atribut PROGID mengidentifikasi class dari object yang berisikan kumpulan method. Atribut cachable yang bertipe boolean menspesifikasi apakah class object tersimpan dimemory, untuk value “0” berarti object tidak tersimpan. Value yang dispesifikasikan oleh atribut ID menunjuk pada object.

Elemen <port> menspesifikasikan elemen <portType> yang didefinisikan pada file WSDL. Untuk setiap operasi yang diberikan elemen <portType> yang ada pada dokumen WSDL, terdapat satu elemen <operation> pada file WSMML. Elemen <execute> memetakan operasi pada object, atribut uses mempunyai ID sama dengan ID pada object, atribut method mengidentifikasi method pada object. Elemen anak <parameter> dari elemen <execute> mendeskripsikan parameter dari method. Elemen ini mempunyai tiga atribut yaitu :

- Atribut callIndex yang menyediakan nomer parameter (untuk parameter pertama adalah “1”, parameter kedua adalah “2” dan seterusnya), untuk value “-1” mengidentifikasi parameter sebagai parameter ‘return’.
- Atribut name menyediakan nama khusus untuk parameter.
- Atribut elementName menyediakan nama dari elemen dalam section <types> pada WSDL yang menangani value dari parameter.

2.4 ASP.NET

ASP.NET merupakan generasi berikutnya dari Active Server Page (ASP) milik Microsoft, yang merupakan fasilitas dari Internet Information Server (IIS). ASP dan ASP.NET dapat dimanfaatkan oleh web builder untuk membuat web site yang dinamis dengan menentukan query untuk menghubungkan database pada suatu web site. ASP.NET berbeda dengan pendahulunya dalam dua hal penting yaitu ASP.NET mendukung penulisan kode pada bahasa terkompilasi seperti Visual Basic, C++, C# dan Perl.

Meskipun teknologi Microsoft's Active Server Pages (ASP) terasa seperti teknologi yang lama tetapi sebenarnya merupakan teknologi yang realtif baru, diperkenalkan pada tahun 1996. Sebelum menggunakan ASP, para developer bisa membuat web site aktif dengan platform Microsoft menggunakan Common Gateway Interface (CGI) dan Internet Server Application Programming Interface (ISAPI), dimana masing-masing berperan dalam evolusi ASP.

CGI merupakan teknik yang diterima secara luas sebagai pengantar pembuatan web yang dinamis. CGI adalah metode yang efektif untuk mengembangkan fungsi-fungsi dari web server sehingga mampu untuk menggenerate response HTTP secara dinamis menggunakan program yang ditulis dengan bahasa C atau bahasa script seperti Perl. Meskipun bisa dikatakan baik tetapi CGI mempunyai beberapa kelemahan. Untuk setiap request HTTP yang diterima oleh sebuah aplikasi CGI, proses baru diciptakan. Setelah request ditangani, maka proses akan di'bunuh'. Pengulangan menciptakan dan mem'bunuh' proses terbukti memberatkan bahkan bagi web server yang handal.

Pada tahun 1996, Microsoft mengeluarkan Active Server Pages. ASP membuat para developer dapat mengeksekusi kode menjadi satu di dalam sebuah halaman web. Meskipun teknologi ASP masih relatif baru untuk membuat suatu web site yang dinamis, tetapi teknologi ini dirancang untuk menjadi satu dari produk pengembangan web site dinamis yang terkemuka. Hal ini karena ada kemudahan dalam pembuatan halaman atau aplikasi yang rumit.

Walaupun ASP mudah digunakan, tetapi dalam beberapa hal ASP juga mempunyai kekurangan, diantaranya :

- Kode ASP dapat menjadi rumit dengan sangat cepat.

Kode ASP cenderung tidak terstruktur dan morat-marit. Banyak kode server-side digabung dengan kode script client-side dan HTML, yang kemudian menjadi sulit untuk memahaminya.

- Keharusan untuk menulis kode apabila ingin melakukan sesuatu pada ASP.

ASP tidak mempunyai komponen model yang aktual. Developer cenderung untuk memulai penulisan kode dari awal halaman hingga akhir meliputi eksekusi query-query database, menjalankan logika dan menggenerate HTML.

- Kode digabung dengan presentasi.

Hal ini menimbulkan masalah ketika developer dan desainer bekerja bersama, dimana sulit untuk mendukung internasionalisasi dan banyak tipe client.

Microsoft menyadari bahwa ASP mempunyai kekurangan yang signifikan, yang kemudian mengembangkan ASP.NET. ASP.NET merupakan satu set komponen yang menyediakan developer dengan framework yang digunakan

untuk mengimplementasikan fungsi-fungsi yang kompleks (Jason Butler and Tony Caudill, 2002). Dua hal utama kelebihan ASP.NET dibanding dengan ASP adalah skalabilitas dan kelengkapannya. ASP.NET berskala luas dengan menyediakan service yang bisa digunakan untuk mengatur variabel session banyak web server dalam kumpulan server. ASP.NET juga memiliki model proses dengan kecepatan tinggi yang dapat mendeteksi kegagalan aplikasi dan memperbaikinya.

Selain kemajuannya dalam skalabilitas dan segi kelengkapan, ASP.NET juga mempunyai keuntungan tambahan, antara lain :

- Kemudahan dalam pengembangan

ASP.NET menawarkan banyak model obyek dimana developer dapat mengurangi jumlah penulisan kode.

- Kebebasan menggunakan bahasa pemrograman.

Halaman ASP harus ditulis dengan menggunakan bahasa script. Dengan kata lain, ASP harus ditulis dengan bahasa yang terinterpretasi daripada terkompilasi. ASP.NET membolehkan bahasa terkompilasi untuk digunakan, dimana menyediakan kinerja yang lebih baik dan kesesuaian antar bahasa.

- Kemudahan untuk menyebarkannya.

Dengan adanya komponen-komponen yang dimiliki ASP.NET, penyebarannya semudah meng-copy komponen ke lokasi yang diinginkan.

- Kemampuan antar client.

Salah satu masalah utama yang dihadapi oleh developer saat ini adalah penulisan kode yang dapat diterima oleh banyak tipe client. Sebagai contoh

menulis sebuah script yang berjalan baik pada Internet Explorer 5.5 dan Netscape Navigator 4.7, serta pada PDA dan mobile phone sangat sulit. ASP.NET menyediakan banyak komponen server-side yang dapat secara otomatis menghasilkan output secara khusus ke masing-masing tipe client.

- Web service

ASP.NET menyediakan kemampuan bagi developer ASP.NET untuk membuat Web service dengan mudah yang bisa digunakan oleh banyak client yang memahami HTTP dan XML, bahasa untuk komunikasi antar device.

- Kinerja.

Halaman ASP.NET disusun dimana halaman ASP diinterpretasi. Ketika sebuah halaman ASP.NET pertama kali di-request, maka akan terkompilasi dan disimpan di memory oleh .NET Common Language Runtime (CLR). Copy cache-nya dapat digunakan ulang untuk request berikutnya. Kinerjanya dijadikan lebih cepat karena setelah request pertama, kode dapat berjalan dari versi kompilasi yang lebih cepat.

2.5 SOAP (Simple Object Access Protocol)

Simple Object Acces Protocol adalah sebuah cara agar suatu program yang berjalan di suatu sistem operasi dapat berkomunikasi dengan sebuah program pada sistem operasi yang sama atau lain dengan menggunakan protokol HTTP dan XML sebagai mekanisme pertukaran informasi (Jason Butler and Tony Caudill, 2002).

2.6 XML (eXtensible Markup Language)

XML adalah kependekan dari eXtensible Markup Language dan telah diterima sebagai standard oleh World Wide Web Consortium (W3C) pada bulan februari 1998. XML adalah suatu dasar baru untuk berkomunikasi melalui internet maupun media yang lainnya. Dengan xml suatu data dapat ditampilkan dalam format yang berbeda-beda (Mario Hadiwinata, 2003).

XML sebenarnya tidak dapat dikatakan sebuah bahasa yang baru, XML adalah meta-language, bahasa untuk mendefinisikan bahasa yang lain. XML digunakan untuk menciptakan suatu dokumen yang terstruktur sehingga dokumen tersebut menjadi “pintar”. XML berkenaan dengan memasukan struktur dan klasifikasi sistem informasi kedalam dokumen itu sendiri, sehingga suatu dokumen dapat mendeskripsikan sendiri isi dari informasi yang terdapat dalam dokumen tersebut. XML diperkirakan akan menjadi standart untuk pertukaran data.

Pada HTML digunakan tags untuk mendefinisikan data. Tags ini secara keseluruhan disebut “markup”. Tidak seperti HTML, XML memberikan informasi bagi kita apa arti dari sebuah data. XML berfungsi sebagai nama colum dalam program. XML memberi label pada data (contoh : <name>Ronny</name>). Kita diberi kebebasan dalam menentukan XML tags untuk suatu aplikasi, sama seperti halnya kita bebas dalam menentukan nama suatu colum pada struktur data.

Berikut ini contoh sebuah data XML :

```
<mahasiswa>
  <nim>96410104058</nim>
  <nama>Ronny Setyawan</nama>
  <program>S1</program>
  <jurusan>Manajemen Informatika</jurusan>
</mahasiswa>
```

Perbedaan antara HTML dan XML adalah suatu dokumen XML harus well formed (terbentuk dengan baik). Suatu dokumen dikatakan well formed jika setiap tag memiliki tag penutup dan tidak terdapat nested tag yang saling overlap. Perbedaan lainnya adalah XML bersifat case-sensitive. Setiap dokumen XML harus dimulai dengan prolog, yang mengidentifikasi dokumen tersebut sebagai dokumen XML, prolog tersebut berbentuk sebagai berikut : `<?xml version="1.0"?>`.

2.7 JavaScript

JavaScript adalah suatu script yang berjalan di client. Script ini ditulis dengan menggunakan bahasa Java. JavaScript digunakan untuk melakukan suatu procedure programming pada sisi client berdasarkan environment dari client tersebut. Hadirnya JavaScript telah memberikan kemudahan, karena segala sesuatu yang dapat dilakukan pada sisi client tidak perlu lagi dilakukan pada sisi server.

Salah satu contoh kegunaan JavaScript adalah untuk melakukan validasi data. Pada saat sebelum ada JavaScript validasi apakah suatu data pada suatu form web diisi atau tidak dilakukan setelah data tersebut dikirim ke web server dengan adanya JavaScript maka pengecekan apakah user mengisi atau tidak suatu form dapat dilakukan sebelum data tersebut dikirim. Hal ini memberikan keuntungan dimana web server hanya akan melakukan proses dimana data yang dikirim sudah valid dan user tidak perlu lagi menunggu apakah proses dilakukannya sukses atau tidak karena user telah di-“paksa” untuk mengisi form tersebut dengan lengkap.

JavaScript tidak dapat dikatakan sama persis dengan Java. JavaScript hanya merupakan bagian kecil dari Java yang diambil dan diletakkan pada web browser sehingga programmer dapat melakukan suatu proses – proses dasar / proses – proses yang tidak rumit pada sisi client, seperti validasi data, perhitungan matematis sederhana dan sebagainya.

JavaScript digunakan dengan cara menyisipkannya pada bagian tertentu dari suatu halaman web. Berikut ini adalah suatu contoh JavaScript pada suatu halaman web yang digunakan untuk memeriksa apakah user telah mengisi Nim atau tidak, jika tidak maka user akan diminta untuk mengisi dan form tidak akan dikirim ke web server untuk diproses.

```
<html>
<head><title>Test JavaScript</title>
<script language="JavaScript">
function checkEntry {
if ( nim.value = "" ) {
alert("Anda harus memasukan NIM anda");
name.focus();
return false;
}
}
</script>
<body>
<form method=post onSubmit="return checkEntry()">
<input type=text name=nim maxlength=11 size=11>
<input type=submit value=" Lihat nilai ">
</form>
</body>
</html>
```

2.8 Microsoft Visual Basic

Aplikasi web service dapat dibuat dengan menggunakan sebagian besar bahasa pemrograman seperti Delphi, C, Java ataupun Microsoft Visual Basic. Penggunaan Visual Basic dalam tugas akhir ini dikarenakan web service

merupakan teknologi yang baru dan sejumlah tutorialnya dalam bentuk bahasa pemrograman Visual Basic. Visual Basic itu sendiri sebagai salah satu bahasa pemrograman berbasis windows yang sangat interaktif menyediakan kemudahan dalam berhubungan dengan database besar seperti MS SQL Server, penanganan interface sampai menghasilkan suatu laporan yang lengkap dan akurat. Terdapat juga fasilitas berupa komponen lengkap untuk penerapan client-server serta mengaktifkan dan menghubungkannya pada suatu jaringan komputer.

2.9 Database Ms SQL Server

Penggunaan database dalam aplikasi web service tidak terbatas pada Ms SQL Server saja, bisa juga menggunakan database lain seperti MySql, Oracle. Penggunaan database Ms SQL Server dalam tugas akhir ini dikarenakan lebih compatible terhadap produk Microsoft yang lain seperti Microsoft Visual Basic maupun ASP.Net. Database Ms SQL Server merupakan suatu sistem manajemen database relasional yang menyediakan berbagai fasilitas dan kemudahan dalam mengelola suatu database termasuk pemanfaatan query dalam manipulasi data.

2.10 Internet

Komunikasi dalam lingkungan berbasis web terjadi antara dua entity, yaitu Web client yang merupakan aplikasi yang melakukan request file dari Web server dan Web server yang merupakan aplikasi yang menangani request dari client. Pada lingkungan web, server melayani response HTTP. Server umumnya memiliki 'kekuatan' lebih dibanding PC dalam penanganan request yang

dilakukan client dalam jumlah besar secara simultan (Jason Butler and Tony Caudill, 2002).

Web browser dan server berkomunikasi dengan menggunakan protokol yang disebut Transmission Control Protocol /Internet Protocol (TCP/IP). Protokol merupakan rangkaian prosedur yang mendefinisikan bagaimana dua entity dapat berkomunikasi. TCP/IP terdiri dari dua bagian yaitu TCP dan IP. TCP atau sering juga disebut transport protocol, ‘membungkus’ data dalam sebuah amplop digital yang disebut packet dan memastikan bahwa data yang diterima berada ditempat yang sama dimana data sebelumnya dikirim. IP atau network protocol menangani rute packet dalam jaringan seperti Internet.

HyperText Transfer Protocol (HTTP) merupakan protokol tipe request-response yang menspesifikasikan client untuk koneksi ke server dan mengirim request dengan format spesifik. Server kemudian memberikan response dan menutup koneksi. HTTP mempunyai kemampuan untuk mentransfer web page, grafis dan tipe-tipe media lain yang digunakan oleh aplikasi web.

