

**PENERAPAN XML *WEB SERVICE* MENGGUNAKAN SOAP
(*SIMPLE OBJECT ACCESS PROTOCOL*) PADA SITUS WEB
PERIKLANAN**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh:

RISKI JUMADI
10551001502



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

PENERAPAN XML *WEB SERVICE* MENGGUNAKAN SOAP (*SIMPLE OBJECT ACCESS PROTOCOL*) PADA SITUS WEB PERIKLANAN

RISKI JUMADI
10551001502

Tanggal Sidang : 31 Oktober 2011
Periode Wisuda : Februari 2012

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas No. 155 Pekanbaru

ABSTRAK

Memasang iklan di *internet* merupakan salah satu cara yang cukup efektif untuk mempromosikan sebuah produk maupun jasa. Banyak ditemui situs web penyedia iklan mulai dari yang gratis hingga berbayar. Model sistem yang digunakan kebanyakan adalah ketika pengiklan memasang sebuah iklan maka penayangannya akan ditampilkan pada situs web yang sama, sehingga iklan hanya akan dibaca oleh pengunjung dari situs web tersebut. Pada tugas akhir ini dirancang bangun sebuah aplikasi periklanan berbasis web dengan XML *Web Service* menggunakan SOAP (*Simple Object Access Protocol*) dimana iklan ditayangkan pada sebuah jaringan *web/blog* milik *publisher*, sehingga iklan akan ditampilkan lebih dari satu situs web.

Hasil implementasi dari rancang bangun aplikasi ini adalah iklan yang dipasang oleh *advertiser* berhasil ditayangkan pada jaringan *web/blog* milik *publisher* sesuai dengan kategori iklan yang ditentukan. Proses pertukaran data melalui SOAP *Message* berhasil dilakukan antara aplikasi dengan *web /blog publisher*.

Kata Kunci: *Advertiser*, Iklan, *Publisher*, SOAP, *Web Service*, XML

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xvii
DAFTAR SIMBOL	xviii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	I-1
1.2 Rumusan Masalah	I-2
1.3 Batasan Masalah	I-2
1.4 Tujuan Penelitian	I-2
1.5 Sistematika Penulisan	I-3
BAB II LANDASAN TEORI	
2.1 <i>Web Service</i>	II-1
2.2 <i>Arsitektur Web Service</i>	II-3
2.3 <i>SOAP (Simple Object Access Protocol)</i>	II-5
2.4 <i>WSDL (Web Services Description Language)</i>	II-8
2.5 <i>UDDI (Universal Description, Discovery, and Integration)</i> ..	II-9
2.6 <i>eXtensible Markup Language</i>	II-10
2.7 Iklan	II-12
2.7.1 Iklan Baris	II-12

2.7.2 Iklan <i>Banner</i>	II-13
2.8 <i>FlowChart</i>	II-13
2.9 PHP.....	II-14
2.10 NuSOAP.....	II-15
BAB III METODOLOGI PENELITIAN	
3.1 Tahapan Penelitian	III-1
3.1.1 Pengumpulan Data.....	III-2
3.1.2 Analisa Sistem.....	III-2
3.1.2.1 Analisa Sistem Lama.....	III-2
3.1.2.2 Analisa Sistem Baru	III-2
3.1.2.3 Deskripsi Umum Perangkat Lunak	III-2
3.1.2.4 Analisis Data Sistem.....	III-3
3.1.2.5 Analisis Fungsional Sistem	III-3
3.1.3 Perancangan.....	III-3
3.1.4 Implementasi	III-3
3.1.5 Pengujian	III-3
BAB IV ANALISA DAN PERANCANGAN	
4.1 Analisa Sistem	IV-1
4.1.1 Sistem yang Berjalan.....	IV-1
4.1.2 Sistem yang akan dibangun.....	IV-2
4.1.2.1 Analisa Pada <i>Advertiser</i>	IV-2
4.1.2.2 Analisa Pada <i>Publisher</i>	IV-3
4.1.2.3 Analisa Kebutuhan	IV-4
4.1.2.4 Analisa Data	IV-5
4.1.3 Identifikasi <i>Web Service</i> pada sistem	IV-6
4.1.4 <i>Flowchart</i> Sistem.....	IV-7
4.1.4.1 <i>Flowchart</i> Pada <i>Advertiser</i>	IV-7
4.1.4.2 <i>Flowchart</i> Pada <i>Publisher</i>	IV-8
4.1.5 Analisa Fungsional Sistem	IV-9
4.1.5.1 <i>Context Diagram</i>	IV-9
4.1.5.2 Data Flow Diagram (DFD).....	IV-10

4.1.5.4 DFD Level 1 <i>Web Service</i>	IV-10
4.1.5.5 DFD Level 2 Proses <i>Member</i>	IV-12
4.1.5.6 DFD Level 2 Proses Pengelolaan iklan dan <i>Channel</i>	IV-13
4.1.5.5 DFD Level 2 Proses Pengelolaan <i>Admin</i>	IV-14
4.1.6 <i>Entity Relationship Diagram (ER-Diagram)</i>	IV-15
4.2 Perancangan.....	IV-17
4.2.1 Perancangan Menu	IV-17
4.2.2 Perancangan Basis Data	IV-18
4.2.3 Perancangan Antar Muka (<i>Interface</i>)	IV-20
BAB V IMPLEMENTASI DAN PENGUJIAN	
5.1 Implementasi Sistem	V-1
5.1.1 Batasan Implementasi.....	V-1
5.1.2 Lingkungan Implementasi	V-1
5.1.3 Hasil Implementasi	V-2
5.1.3.1 Implementasi <i>SOAP Service</i> dan dokumen <i>WSDL (Web Service Definition Language)</i>	V-2
5.1.3.2 Tampilan Antarmuka (<i>Interface</i>).....	V-8
5.1.3.2.1 Tampilan Halaman Utama.....	V-8
5.1.3.2.2 Tampilan Halaman <i>Advertiser</i>	V-8
5.1.3.2.3 Tampilan Halaman <i>Publisher</i>	V-9
5.2 Pengujian Sistem	V-9
5.2.1 Pengujian Aplikasi <i>Server</i>	V-10
5.2.1.1 Identifikasi dan Pengujian Menu <i>Login</i>	V-10
5.2.1.2 Identifikasi dan Pengujian Menu <i>Profil</i>	V-10
5.2.2 Pengujian <i>Client / Channel</i>	V-11
5.2.2.1 Skenario Pengujian.....	V-11
5.2.2.2 Hasil Pengujian.....	V-12
5.3 Kesimpulan Pengujian.....	V-15

BAB VI PENUTUP

6.1 Kesimpulan..... VI-1

6.2 Saran VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR TABEL

Tabel	Halaman
2.1 <i>Tag-tag</i> pada dokumen WSDL	II-9
4.1 Jenis dan Ukuran iklan	IV-5
4.2 Identifikasi <i>Web Service</i>	IV-6
4.3 DFD Level 1 <i>Web service</i>	IV-11
4.4 Keterangan Aliran Data DFD Level 1 <i>Web service</i>	IV-11
4.5 DFD Level 2 Proses Registrasi	IV-12
4.6 Keterangan Aliran Data DFD Level 2 Proses Registrasi	IV-12
4.7 DFD Level 2 Proses Pengelolaan iklan & <i>channel</i>	IV-13
4.8 Keterangan Aliran Data DFD Level 2 Proses iklan & <i>channel</i>	IV-13
4.9 DFD Level 2 Proses Pengelolaan <i>Admin</i>	IV-14
4.10 Keterangan Aliran Data DFD Level 2 Proses <i>Admin</i>	IV-14
4.11 Keterangan entitas pada ERD	IV-15
4.12 Perancangan Tabel Member.....	IV-18
4.13 Perancangan Tabel <i>Channel</i>	IV-18
4.14 Perancangan Tabel Iklan	IV-19
4.15 Perancangan Tabel Detail_klik	IV-19
4.16 Perancangan Tabel Kategori_iklan	IV-19
5.1 Identifikasi <i>service</i>	V-3
5.2 Hasil Pengujian Menu Login	V-10
5.3 Hasil Pengujian Menu Profil.....	V-10
5.4 Daftar <i>Channel</i>	V-11

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi informasi yang semakin berkembang dan didukung oleh struktur dan infrastruktur yang semakin baik membuat kebutuhan akan informasi dengan mudah didapatkan terutama dengan adanya *internet*. Hal tersebut memberikan peluang kepada pelaku bisnis untuk mulai melirik *internet* sebagai media yang sangat potensial untuk mendapatkan konsumen atas produk atau jasa yang ditawarkan, sehingga diharapkan mampu untuk meningkatkan penjualan. Berbagai cara dilakukan untuk meningkatkan jumlah pengunjung yang datang ke situs web mereka. Sebab, secara tidak langsung semakin banyak pengunjung yang datang, maka semakin besar peluang untuk mendapatkan lebih banyak konsumen.

Salah satu cara yang dapat dilakukan untuk mendapatkan banyak pengunjung adalah dengan memasang iklan di situs web penyedia jasa periklanan yang banyak ditemui di *internet*, mulai dari yang gratis hingga berbayar. Salah satu contoh adalah www.iklanbaris.co.id, dimana pengiklan (*advertiser*) melakukan pendaftaran pada situs tersebut lalu memasang iklannya, kemudian iklan akan ditayangkan di halaman web www.iklanbaris.co.id. Namun, beriklan dengan cara tersebut masih kurang efektif, sebab iklan masih harus bersaing untuk tampil dengan iklan yang dipasang oleh pengiklan lain. Sehingga, semakin kecil kemungkinan iklan akan dibaca.

Berdasarkan hal tersebut maka digunakan teknologi *web service* yang dapat membantu memberikan solusi terhadap permasalahan yang terjadi. Peranan *web service* diperlukan sebagai media untuk *sharing* data maupun fungsionalitas antar proses bisnis agar suatu aplikasi dapat berinteraksi dengan aplikasi lain tanpa tergantung dengan bahasa pemrograman maupun sistem operasi.

Jika sebelumnya iklan hanya dipasang dan ditayangkan pada satu situs web namun dengan menggunakan *web service* akan mengubah konsep tersebut, pemasang iklan cukup memasang iklannya pada satu situs (penyedia *web service*

periklanan), kemudian mesin *web service* yang akan bekerja untuk menayangkan iklan di halaman *web/blog* orang lain (*publisher*) yang mau menyisakan sedikit ruangan di halaman *web/blog* mereka, artinya terjadi interaksi antara dua aplikasi yang berbeda dimana *web/blog publisher* akan mengirimkan *request* ke *web service* dan kemudian *web service* akan merespon dengan mengirimkan data iklan yang dipasang oleh pengiklan. Disinilah peranan *web service* dalam melakukan proses pertukaran data. Dalam penelitian ini teknologi yang akan digunakan adalah SOAP (*Simple Object Acces Protocol*) karena SOAP merupakan protokol yang sudah distandarisasi oleh W3C (*World Wide Web Consortium*) dan didukung luas dari berbagai *vendor*.

Berdasarkan permasalahan diatas, dalam penelitian ini akan dirancang bangun sebuah situs web periklanan dengan memanfaatkan teknologi *web service* menggunakan SOAP.

1.2. Rumusan Masalah

Dari latar belakang yang telah dijelaskan sebelumnya dapat diambil suatu perumusan masalah yaitu bagaimana menerapkan teknologi *web service* menggunakan SOAP pada situs web periklanan.

1.3. Batasan Penelitian

Agar penulisan tugas akhir ini lebih terarah maka penulis membuat batasan permasalahan, yaitu :

1. Situs web hanya menampilkan iklan teks dan gambar.
2. Tidak membahas mengenai sistem pembayaran.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai adalah rancang bangun situs web periklanan dengan XML *Web Service* menggunakan SOAP dalam memfasilitasi penayangan iklan dari *advertiser* pada jaringan *web/blog publisher*.

1.5. Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan tugas akhir yang akan dibuat :

Bab I Pendahuluan

Bagian ini berisi tentang deskripsi umum tugas akhir yang meliputi latar belakang masalah, perumusan masalah, tujuan penelitian, batasan penelitian, serta sistematika penulisan.

Bab II Landasan Teori

Bagian ini menjelaskan tentang teori-teori umum, teori-teori khusus yang berhubungan dengan tugas akhir ini.

Bab III Metodologi Penelitian

Bagian ini menjelaskan tentang metode pengembangan modul aplikasi, pengumpulan data dan objek, tempat serta waktu penelitian.

Bab IV Analisa Dan Perancangan

Bagian ini berisi tentang analisis dan perancangan aplikasi. Pada tahap perancangan akan dibahas tentang perancangan aplikasi.

Bab V Implementasi Dan Pengujian

Pada bagian implementasi dan pengujian berisi pembahasan mengenai implementasi modul disertai dengan *preview* tampilan *interface* serta pengujian aplikasi.

Bab VI Kesimpulan Dan Saran

Bagian ini berisi kesimpulan hasil penelitian beserta saran-saran yang berkaitan dengan penelitian ini.

BAB II

LANDASAN TEORI

2.1. *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML (*eXtensible Markup Language*), sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *web service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di *upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di *port* 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

Web Service dapat diimplementasikan pada lingkungan internal (*Intranet*) untuk kebutuhan integrasi antar sistem aplikasi (EAI=*Enterprise Application Integration*) ataupun pada lingkungan eksternal (*Internet*) untuk mendukung aplikasi *business-to-business* (*e-business*).

Keuntungan penggunaan *Web Service* : (Wikipedia.org, 2010)

1. Format penggunaan terbuka untuk semua *platform*.
2. Mudah di mengerti dan mudah men-*debug*.
3. Menggunakan standar-standar “membuka *service* sekali” dan mempunyai pemakai banyak.
4. Mudah untuk menengahi pesan-pesan proses dan menambahkan nilai.
5. *Routing* and pengiriman.
6. *Security*.
7. *Management* and *monitoring*.
8. *Schema* and *service design*.
9. Akselerasi.
10. Mudah untuk mengembangkan dengan *semantic transport* tambahan.
11. Terbuka, standar-standar berbasis teks.
12. Pencapaian modular.
13. Tidak mahal untuk diimplementasikan (relatif).
14. Mengurangi biaya integrasi aplikasi *enterprise*.
15. Implementasi yang *incremental*.

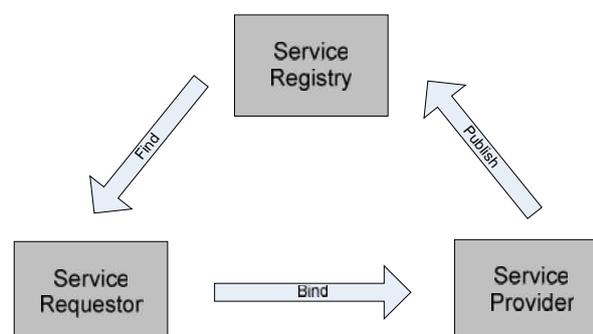
Web service ternyata sangat berbeda dengan *website*. Perbedaan yang paling terlihat adalah *website* dibuat untuk memiliki tampilan atau *user interface* yang bagus sedangkan *web service* tidak memiliki tampilan. Itu karena *web service* tidak dibuat untuk berinteraksi langsung dengan *user*. Sesuai dengan kata *service* yang ada pada namanya, *web service* hanya menyediakan *service* atau layanan. Layanan tersebutlah yang kemudian akan digunakan atau dipanggil oleh aplikasi lainnya. Dengan demikian yang akan menjadi *interface* adalah aplikasi yang memanggilnya bukan *web service* itu sendiri.

2.2. Arsitektur Web Service

Konsep arsitektur yang mendasari teknologi *Web service* adalah *Service Oriented Architecture* (SOA). Dalam arsitektur ini, suatu aplikasi dimodelkan sebagai komposisi dari sekumpulan *service* yang disediakan oleh suatu komponen. Lokasi keberadaan komponen tersebut dapat ditemukan oleh *client* secara dinamis, dalam arti tidak dinyatakan secara statis tetapi menggunakan mekanisme *discovery* untuk mencari keberadaan komponen tersebut. Demikian pula, *client* dapat meminta (*invoke*) *service* tersebut secara dinamis pula. Dalam arsitektur ini, SOA mendefinisikan 3 peran berbeda yang menunjukkan peran dari masing-masing komponen dalam sistem, yaitu :

1. *Service provider*, yaitu suatu entitas yang menyediakan *interface* terhadap sistem yang menjalankan suatu sekumpulan tugas tertentu. *Service provider* dapat merepresentasikan suatu entitas bisnis ataupun suatu komponen *software* yang *reusable*.
2. *Service requestor*, yaitu suatu entitas yang meminta/memperoleh (dan menemukan) *software service* dalam rangka menyelesaikan suatu tugas tertentu atau menyediakan solusi bisnis tertentu.
3. *Service registry*, yaitu entitas yang bertindak sebagai penyimpan (*repository*) suatu *software service* yang dipublikasikan oleh *Service provider*.

Peran dan interaksi antar komponen tersebut ditunjukkan pada gambar dibawah ini :



Gambar 2.1 - Konsep *Service Oriented Architecture*

Secara umum, *web service* memiliki tiga operasi yang terlibat di dalamnya, yaitu:

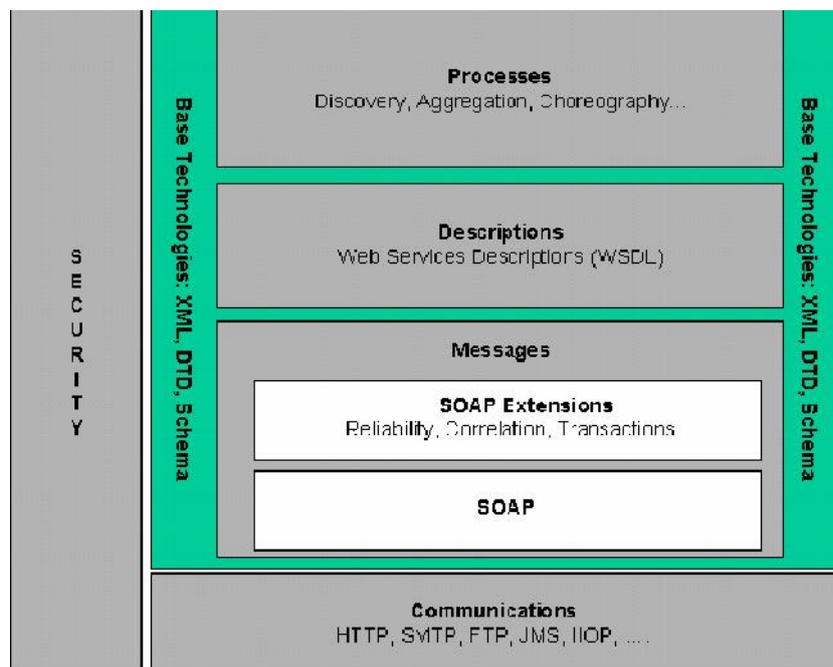
1. *Publish/Unpublish*: Menerbitkan/menghapus layanan ke dalam atau dari *registry*.
2. *Find*: *Service requestor* mencari dan menemukan layanan yang dibutuhkan.
3. *Bind*: *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke *service provider* untuk melakukan interaksi dan mengakses layanan/*service* yang disediakan oleh *service provider*.

Dalam arsitektur ini, *service* dapat didefinisikan sebagai komponen (*software component*) yang memiliki karakteristik :

1. Dapat dideskripsikan dalam suatu bahasa formal
2. Dapat dipublikasikan pada suatu *registry of service*
3. Dapat ditemukan (*discover*) menggunakan mekanisme standar
4. Dapat diminta/diperoleh (*invoke*) melalui jaringan
5. Dapat dibangun bersama *service* lain

Model Arsitektur *Web Service* menggunakan konsep SOA tersebut ,yaitu :

1. *Web Service* menggunakan standar terbuka, yaitu : XML (*eXtensible Markup Language*) sebagai bahasa untuk mendeskripsikan data, SOAP (*Simple Object Access Protocol*) sebagai protokol komunikasi antar komponen , WSDL (*Web Service Description Language*) untuk mendeskripsikan *service*, UDDI (*Universal Description, Discovery and Integration*) untuk *service discovery*, dan HTTP sebagai *transport layer*.
2. Standar *Web Service* merupakan *message-based*, bukan *API-based*. Dalam hal ini, komunikasi antara *Web Service*, *Service Requester* dan *Service Registry* menggunakan pesan SOAP berbasis XML. W3C mengembangkan *draft* arsitektur *Web Service* yang terdiri dari beberapa teknologi yang saling berhubungan dan berlapis seperti digambarkan sebagai berikut :



Gambar 2.2 Arsitektur Web Service

2.3. SOAP (*Simple Object Access Protocol*)

SOAP merupakan protokol komunikasi yang ringan berbasis *eXtensible Markup Language* (XML). Pesan SOAP berbentuk sekumpulan *XML Schema* yang mendefinisikan format untuk mentransmisikan pesan XML melalui jaringan, termasuk tipe data dan cara menstrukturkan pesan secara tepat sehingga dapat mudah dipahami oleh *server* atau *end-point* lainnya.

Pesan SOAP terdiri dari 3 bagian, yaitu :

1. *Envelope*, suatu kerangka yang mendefinisikan apa yang ada dalam pesan dan bagaimana pesan harus diproses serta menunjukkan resipien dari pesan tersebut.
2. *Header*, berisi informasi yang berhubungan dengan keamanan dan *routing*. Keberadaan *Header* dalam SOAP bersifat *optional*.
3. *Body*, berisi data yang berhubungan dengan aplikasi tertentu yang sedang dipertukarkan. Data di *mark-up* sebagai XML dan dimasukkan dalam format yang spesifik yang didefinisikan dalam *XML Schema*.

SOAP (*Simple Object Access Protocol*) adalah sebuah *XML based markup language* untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data *object* didalam *network*. SOAP mendefinisikan empat aspek didalam komunikasi: *Message envelope*, *Encoding*, *RPC call convention*, dan bagaimana menyatukan sebuah *message* didalam protokol *transport*.

Sebuah SOAP *message* terdiri dari SOAP *Envelope* dan bisa terdiri dari *attachments* atau tidak memiliki *attachment*. SOAP *envelope* tersusun dari SOAP *header* dan SOAP *body*, sedangkan SOAP *attachment* membolehkan non-XML data untuk dimasukkan kedalam SOAP *message*, di-*encoded*, dan diletakkan kedalam SOAP *message* dengan menggunakan *MIME-multipart*.

SOAP menspesifikan secara jelas bagaimana cara untuk meng-*encode header* HTTP dan *file* XML sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan.

SOAP adalah protokol ringan yang ditujukan untuk pertukaran informasi struktur pada lingkup desentralisasi, dan terdistribusi. SOAP menggunakan teknologi XML untuk mendefinisikan rangka kerja pemesanan terekstrensi di mana menyediakan konstruksi pesan yang dapat dipertukarkan pada protokol berbeda. Rangka kerja dirancang bebas dari model pemrograman dan spesifikasi implementasi semantik.

SOAP tidak berpasangan erat dengan protokol *transport* tertentu. Spesifikasi SOAP menerangkan bagaimana pesan-pesan SOAP seharusnya diikat ke HTTP. Namun, sebuah pesan SOAP tidak lebih daripada sebuah dokumen XML. Jadi, ia dapat diangkut melalui protokol apa pun yang mampu mengirimkan teks. (Scott Short, 2003).

Peran SOAP di dalam teknologi *web service* adalah sebagai protokol pemaketan untuk pesan-pesan (*messages*) yang digunakan secara bersama oleh aplikasi-aplikasi penggunanya. Spesifikasi yang digunakan tidak lebih seperti sebuah amplop biasa berbasis XML untuk informasi yang ditransfer, serta sekumpulan aturan bagi translasi aplikasi dan tipe-tipe data platform yang spesifik

menjadi bentuk XML. Desain bentuk dari SOAP membuatnya cocok untuk berbagai pertukaran pesan pada aplikasi. Berikut adalah salah satu contoh bentuk sederhana pesan SOAP :

Contoh 1 : SOAP Message Embedded in HTTP Request

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:
encodingStyle="http://schemas.xmlsoap.org/soap/encoding
/">
<SOAP-ENV:Body>
<m:GetLastTradePrice xmlns:m="Some-URI">
<symbol>DIS</symbol>
</m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Contoh 2 : SOAP Message Embedded in HTTP Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/enco
ding/">
<SOAP-ENV:Body>
<m:GetLastTradePriceResponse xmlns:m="Some-URI">
<Price>34.5</Price>
</m:GetLastTradePriceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.4. WSDL (*Web Services Description Language*)

WSDL (*Web Services Description Language*) adalah dialek berbasis XML yang lapisannya ada diatas skema yang menjelaskan layanan web. Dokumen WSDL menyediakan informasi yang dibutuhkan *client* untuk berinteraksi dengan layanan web. WSDL dapat diperluas dan dipakai untuk menjelaskan hampir semua layanan jaringan, termasuk SOAP lewat HTTP. Bahkan protokol yang bukan berbasis XML, seperti DCOM lewat UDP. (Scott Short,2003)

WSDL (*Web Services Description Language*) adalah format XML yang diterbitkan untuk menerangkan *web service*. WSDL mendefinisikan: (Wikipedia.org, 2010)

- Pesan-pesan (baik yang abstrak dan kongkrit) yang dikirim ke dan menuju *web service*
- Koleksi-koleksi digital dari pesan-pesan (*port type*, antarmuka)
- Bagaimana *port type* yang ditentukan dijadikan *wire* protokol
- Dimana *service* ditempatkan

WSDL digunakan di mana skema XML tidak digunakan lagi dengan menyediakan jalur pesan-pesan grup menjadi operasi-operasi dan operasi-operasi menjadi antarmuka. Ini juga menyediakan jalur untuk medefinisikan *binding-binding* untuk setiap antamuka dan kombinasi protokol sepanjang alamat titik akhir untuk setiap kalinya. Definisi WSDL yang lengkap terdiri dari seluruh informasi yang dibutuhkan untuk meminta *web service*. Pengembang yang mau mempermudah yang lain untuk mengakses *service-service* nya harus menyediakan definisi-definisi WSDL.

Sebuah dokumen WSDL dapat dibagi menjadi 6 bagian besar. Masing-masing bagian tersebut ditandai tag-tag tertentu. (Lucky, 2008)

Tabel 2.1 *Tag-tag* pada dokumen WSDL

Tag	Keterangan
<i>Definitions</i>	Tag ini adalah tag tertinggi dalam dokumen WSDL atau biasa disebut dengan tag <i>root</i> . Tag-tag lainnya harus berada didalam tag ini. Salah satu informasi yang ada didalam tag ini adalah <i>namespace</i> .
<i>Types</i>	Didalam tag ini didefinisikan tipe-tipe data yang digunakan oleh <i>web service</i> . Akan tetapi yang didefinisikan hanyalah tipe-tipe data kompleks seperti <i>array</i> atau <i>struct</i> . Sedangkan tipe-tipe data sederhana yang sudah <i>built-in</i> seperti <i>integer</i> , <i>double</i> , <i>string</i> , dan lain-lain tidak didefinisikan lagi.
<i>Messages</i>	Informasi yang ada didalam tag ini adalah <i>message</i> atau parameter <i>input</i> atau <i>output</i> apa saja yang akan digunakan
<i>PortType</i>	Didalam tag ini didefinisikan <i>method</i> atau fungsi yang disediakan oleh <i>web service</i>
<i>Binding</i>	Informasi yang ada didalam tag ini bagaimana sebuah operasi atau fungsi diimplementasikan. Apakah fungsi tersebut dapat dipanggil menggunakan protokol HTTP atau SMTP. Informasi seperti <i>soap action</i> juga didefinisikan didalam tag ini
<i>Service</i>	Didalam tag ini didefinisikan lokasi dari <i>web service</i>

2.5. UDDI (*Universal Description, Discovery, and Integration*)

UDDI adalah spesifikasi yang telah diakui secara luas untuk pendaftaran suatu *web service*. UDDI mendefinisikan metadata dan protokol yang digunakan untuk mengetahui dan mengubah informasi dari suatu *web service*. Langkah pertama dalam menemukan *web service* adalah dengan meminta alamat tempat penyimpanan dari *web service* yang akan dipakai yang biasa disebut *directory*.

Setelah menemukan *directory*, peminta *web service* dapat mengirimkan permintaan lagi untuk mendapatkan informasi detail tentang *web service* (misalnya penyedia *web service* dan dimana diletakkan). Selanjutnya perangkat lunak menggunakan informasi yang didapat untuk secara dinamik mengakses *web service* yang diinginkan.

2.6. *eXtensible Markup Language*

XML kependekan dari *eXtensible Markup Language*, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan *markup language* yang tidak kalah hebatnya dengan SGML.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup (diawali dengan '</' diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal `<form name="isidata">`). Hanya bedanya, HTML mendefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak.

XML adalah *meta-language*. *Meta-language* adalah bahasa yang dapat digunakan untuk mendefinisikan bahasa-bahasa lainnya. Contohnya adalah menggunakan XML untuk mendefinisikan bahasa seperti WML.

Dengan XML dapat dilakukan:

1. Mendefinisikan struktur data
2. Membuat struktur tersebut *platform independent*
3. Memproses data terdefinisi oleh XML secara otomatis
4. Mendefinisikan *tag* sendiri

Beberapa alasan mengapa menggunakan XML :

1. Pertukaran data.
XML digunakan dalam pertukaran data. XML memungkinkan untuk mendefinisikan data dengan jelas. Pengirim dan penerima data akan menggunakan XML untuk memahami data yang telah dikirimkan. Dengan menggunakan XML, pengirim dan penerima memiliki interpretasi sama tentang data yang digunakan.
2. Pengganti *Electronic Data Interchange* (EDI). EDI telah digunakan dalam pertukaran data. EDI mahal, karena menggunakan *dedicated communication infrastructure*. Definisi yang digunakan jauh dari fleksibel. XML adalah pengganti yang baik terhadap EDI. XML menggunakan *Internet* dalam pertukaran data dan fleksibel.
3. Kemungkinan lainnya. XML dapat digunakan untuk mendefinisikan bahasa lainnya. Salah satu contohnya adalah WML (Wireless Markup Language), bahasa yang digunakan dalam komunikasi WAP. Dengan kata lain, WML adalah salah satu dialek XML.

Pada dasarnya, kemampuan XML memiliki potensi untuk dapat digunakan pada kondisi dimana data dimasukkan/dikeluarkan, disimpan, atau ditransmisikan dari suatu tempat ke tempat lain. XML dapat digunakan sebagai pengganti (atau pelengkap) database tradisional, atau mentransfer informasi antar bisnis. Organisasi berita, termasuk individu, juga dapat menggunakan XML untuk mendistribusikan *syndicated news stories*.

Bagian-Bagian dari Dokumen XML

Sebuah dokumen XML terdiri dari bagian bagian yang disebut dengan *node*. *Node-node* itu adalah:

- *Root node* yaitu *node* yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu *root node*. *Node-node* yang lainnya berada di dalam *root node*.

- *Element node* yaitu bagian dari dokumen XML yang ditandai dengan *tag* pembuka dan *tag* penutup, atau bisa juga sebuah *tag* tunggal elemen kosong seperti `<anggota nama="budi" />`. *Root node* biasa juga disebut *root element*
- *Attribute note* termasuk nama dan nilai atribut ditulis pada *tag* awal sebuah elemen atau pada *tag* tunggal.
- *Text node*, adalah *text* yang merupakan isi dari sebuah elemen, ditulis diantara *tag* pembuka dan *tag* penutup
- *Comment node* adalah baris yang tidak dieksekusi oleh *parser*
- *Processing Instruction node*, adalah perintah pengolahan dalam dokumen XML. Node ini ditandai awali dengan karakter `<?` Dan diakhiri dengan `?>`. Tapi perlu diingat bahwa header standard XML `<?xml version="1.0" encoding="iso-8859-1"?>` bukanlah *processing instruction node*. *Header standard* bukanlah bagian dari hirarki pohon dokumen XML.
- *NameSpace Node*, node ini mewakili deklarasi *namespace*

2.7. Iklan

Iklan ialah promosi barang, jasa, perusahaan dan ide yang harus dibayar oleh sebuah sponsor. Pemasaran melihat iklan sebagai bagian dari strategi promosi secara keseluruhan. Komponen lainnya dari promosi termasuk publisitas, relasi publik, penjualan, dan promosi penjualan.

2.7.1. Iklan Baris

Iklan baris (bahasa Inggris: *Classified advertising*) adalah salah satu cara promosi barang dan jasa yang umumnya ditemukan di koran. Cara ini merupakan pengembangan dari promosi iklan yang mengutamakan daya tarik dengan gambar dan dengan informasi yang lebih lengkap dan terinci. (Wikipedia.org, 2010)

Iklan baris mengutamakan informasi yang paling inti yang perlu diketahui oleh peminatnya. Karena itu biasanya iklan baris hanya memuat informasi seperlunya dan hanya membutuhkan beberapa baris saja. Biasanya koran-koran mensyaratkan iklan baris minimal 2-3 baris.

Karena tidak menggunakan gambar sebagai daya tariknya, iklan baris dimuat secara berkelompok sesuai dengan isinya. Misalnya, kelompok barang yang dijual dipasang dalam baris yang berbeda dengan kelompok barang yang dicari. Selanjutnya, barang-barang yang dijual pun diklasifikasikan lebih jauh: tanah, rumah, mobil, motor, dan lain-lain.

2.7.2. Iklan Banner

Banner adalah papan iklan elektronik. Memasang iklan *banner* merupakan bentuk paling umum digunakan didalam periklanan *internet*. Biasanya sebuah *banner* berisi teks singkat atau pesan bergambar untuk mempromosikan suatu produk atau *vendor*. *Banner* juga mungkin berisi video klip atau suara. Pada saat pelanggan mengklik suatu banner, ia akan ditransfer ke *home page* atau *storefront* pengiklannya.

2.8. FlowChart

Flowchart adalah bagan yang menunjukkan urutan dan hubungan antar proses beserta instruksinya dan gambaran itu dinyatakan dengan simbol. Adapun simbol-simbol yang digunakan tertera digambar berikut ini.

Simbol	Nama	Keterangan
	<i>Process</i>	Untuk menunjukkan proses pengolahan yang dilakukan oleh computer.
	<i>Decision</i>	Untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.
	<i>Terminal</i>	Untuk permulaan atau akhir dari suatu program.
	<i>Input-output</i>	Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	<i>Menu</i>	Untuk menunjukkan menu pilihan di dalam program.
	<i>Connector</i>	Untuk keluar masuk/masuk prosedur atau proses dalam lembar/halaman yang sama.
	<i>Off-line connector</i>	Untuk keluar masuk/masuk prosedur atau proses dalam lembar/halaman yang lain.

Gambar 2.3 Simbol *Flowchart*

2.9. PHP

PHP dikenal sebagai sebuah bahasa *scripting* yang menyatu dengan *tag-tag* HTML, dieksekusi di *server* dan digunakan untuk membuat halaman *web* dinamis. Versi pertama PHP dibuat oleh Rasmus Lerdorf pada tahun 1995, berupa sekumpulan *script* PERL yang digunakan oleh Rasmus Lerdorf untuk membuat halaman *web* yang dinamis pada *homepage* pribadinya.

Berdasarkan hasil *survey* Netcraft pada bulan Desember 1999, lebih dari satu juta *site* yang menggunakan PHP, termasuk perusahaan besar seperti Mitsubishi, Redhat, NASA, Ericson dan banyak lagi. Berdasarkan *survey* Esoft pada bulan November 1999, 23% pengguna *Apache Server* menggunakan PHP (55% *webserver* dunia menggunakan *Apache*).

Dasar pertimbangan untuk mengembangkan kemampuan *object oriented* dari PHP adalah perkembangan aplikasi web sebagai sebuah *platform* yang terus meluas dengan cepat karena ruang lingkup aplikasi web semakin luas, sehingga aplikasi web yang dibangun juga menjadi semakin besar, rumit, dan kompleks.

Dengan kemampuan untuk mengimplementasikan konsep *object oriented*, PHP akan lebih mudah dapat berhubungan dan mengakses komponen-komponen yang dibuat dengan bahasa *object oriented* lain seperti Java atau C++ dan bahkan Visual Basic.

Berikut ini akan dijelaskan beberapa keunggulan PHP (Farid Azis, M, 2001):

- a. *Life cycle* yang singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
- b. *Cross platform*, PHP dapat dipakai di hampir semua *web server* yang ada di pasaran (*Apache*, *fhhttpd*, *phhttpd*, *Microsoft IIS* dan lain-lain) yang dijalankan pada berbagai sistem operasi. Dengan demikian proses *developing* dapat dilakukan menggunakan sistem operasi yang digunakan setelah *publish* (misalnya, pengembangan di *Windows* kemudian dipasang di *web server* yang menggunakan sistem operasi *Linux*).

- c. PHP mendukung banyak paket *database* baik yang komersil maupun yang non komersil.

2.10. NuSOAP

NuSOAP adalah sebuah kumpulan *class-class* PHP yang memungkinkan *user* untuk mengirim dan menerima pesan SOAP melalui protokol HTTP. NuSOAP didistribusikan oleh *NuSphere Corporation* (<http://www.nusphere.com>) sebagai *open source toolkit* dibawah lisensi GNU/GPL.

NuSOAP merupakan *toolkit web service* berbasis komponen. NuSOAP memiliki sebuah *class* dasar yang menyediakan method seperti serialisasi variabel dan pemaketan SOAP-Envelope. Interaksi *web service* dilakukan dengan *class client* yang disebut dengan *class "soapclient"* dan *class server* yang disebut dengan *class "soap-server"*. *Class-class* ini mengizinkan *user* untuk melakukan proses pengiriman dan penerimaan pesan-pesan SOAP dengan bantuan beberapa *class-class* pendukung lainnya untuk melengkapi proses tersebut.

BAB III

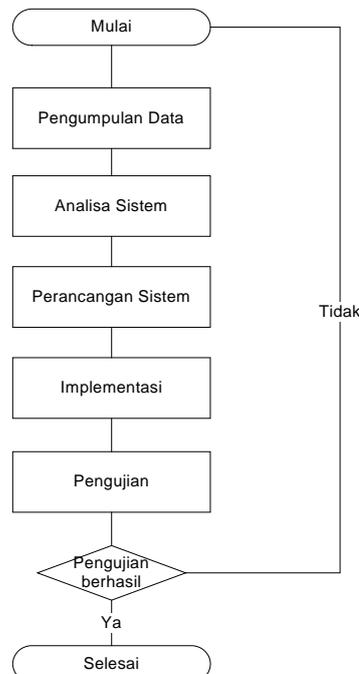
METODOLOGI PENELITIAN

Metodologi penelitian merupakan cara yang digunakan dalam memperoleh berbagai data untuk kemudian diproses menjadi informasi yang lebih akurat sesuai permasalahan yang akan diteliti. Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya.

3.1. Tahapan Penelitian

Penjelasan tentang metodologi penelitian didasarkan pada tujuan dari Tugas Akhir. Metodologi penelitian yang digunakan dalam penyusunan Tugas Akhir ini akan melalui beberapa tahapan yang membentuk sebuah alur yang sistematis. Tahapan-tahapan itu meliputi: Pengumpulan Data, Analisa dan Perancangan Sistem, Implementasi dan Pengujian.

Tahapan penelitian yang akan dilaksanakan pada perancangan aplikasi ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Tahapan Penelitian

3.1.1. Pengumpulan Data

Pengumpulan materi merupakan tahapan persiapan yang harus dilaksanakan terlebih dahulu sebelum dilakukan penelitian. Berikut merupakan aktivitas yang dilaksanakan dalam pengumpulan sumber teori :

- Studi Pustaka

Studi pustaka bermanfaat untuk mendukung penelitian yang akan dilaksanakan. Studi pustaka dilakukan dengan tujuan untuk menyelesaikan permasalahan yang diteliti, serta mendapatkan dasar-dasar yang kuat dalam menerapkan suatu metode yang nantinya dapat digunakan dalam tugas akhir ini.

Pengumpulan teori-teori yang mendukung dalam penelitian ini merupakan kegiatan dalam studi pustaka. Teori-teori bersumber dari buku, jurnal dan penelitian yang terkait dengan *eXtensible Markup Language* (XML), *Web Service*, *Simple Object Access Protocol* (SOAP), Bahasa Pemrograman PHP dan NuSOAP.

3.1.2. Analisa Sistem

Tahapan ini mengenali seluruh permasalahan yang muncul dalam mengenali komponen-komponen sistem dan mempelajari prosedur sistem yang akan dibuat, serta menganalisis solusi maksimal atas kebutuhan sistem.

3.1.2.1. Analisa Sistem Lama

Pada tahap ini dilakukan analisa terhadap sistem lama atau metode pengerjaan yang sedang berlangsung, termasuk untuk mengetahui kelemahan yang dimiliki oleh sistem lama tersebut.

3.1.2.2. Analisa Sistem Baru

Setelah menganalisa sistem lama, maka tahapan dapat dilanjutkan dengan menganalisa sistem baru. Dalam tahapan ini, akan diidentifikasi cara kerja dari sistem baru yang akan dibangun.

3.1.2.3. Deskripsi Umum Perangkat Lunak

Deskripsi umum perangkat lunak dibuat sebagai suatu gambaran tentang rancangan sistem yang akan dibuat. Dalam deskripsi Umum

Perangkat Lunak ini dilakukan dua tahapan yakni Deskripsi Umum Sistem dan Karakteristik Pengguna.

3.1.2.4. Analisis Data Sistem

Analisis ini dilakukan dengan menggunakan beberapa tahapan yakni Analisis Masukan (*input*), Analisis Proses dan Analisis *Output*.

3.1.2.5. Analisis Fungsional Sistem

Analisis yang digunakan pada sistem adalah pemodelan fungsional. Pemodelan fungsional merupakan pemodelan yang menggambarkan suatu masukan yang diproses pada sistem menjadi keluaran yang dibutuhkan bagi pengguna sistem.

3.1.3. Perancangan

Tahapan penganalisaan dan perancangan Alat bantu yang digunakan adalah Model Sistem, *Flowchart*, *Context Diagram*, *Data flow Diagram*, *ER-Diagram* dan *interface* rancangan sistem. Dan perangkat lunak yang digunakan untuk membantu proses perancangan sistem yakni: Microsoft Word 2007, dan Microsoft Visio 2003.

3.1.4. Implementasi

Implementasi akan dilakukan pada tahap ini, yaitu setelah fase perancangan selesai. Tahap implementasi merupakan tahap dimana hasil dari perancangan dituliskan kedalam bahasa pemrograman dan database sehingga mencapai hasil akhir berupa aplikasi xml *web service* untuk periklanan *online*. Perangkat lunak yang digunakan adalah PHP, MySQL, serta *library* NuSOAP.

3.1.5. Pengujian

Ini bertujuan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan. Apabila didalam *testing* tersebut terdapat kesalahan (*error*), maka harus dilakukan perbaikan kembali pada sistem tersebut.

BAB IV

ANALISA DAN PERANCANGAN

Bab ini akan membahas analisa dan perancangan yang merupakan dua tahapan yang saling berhubungan erat. Analisa bertujuan untuk mengidentifikasi permasalahan dan kebutuhan sistem, sementara perancangan bertujuan untuk mengimplementasikan permasalahan yang telah diidentifikasi kedalam sebuah aplikasi.

4.1. Analisa Sistem

Analisa merupakan suatu aktivitas kreatif dalam proses mentransformasikan suatu definisi permasalahan dari sebuah ketidakjelasan, meliputi poses memahami permasalahan, mengetahui keterbatasan dan menggagas metode yang tepat untuk mengatasi permasalahan serta keterbatasan tersebut

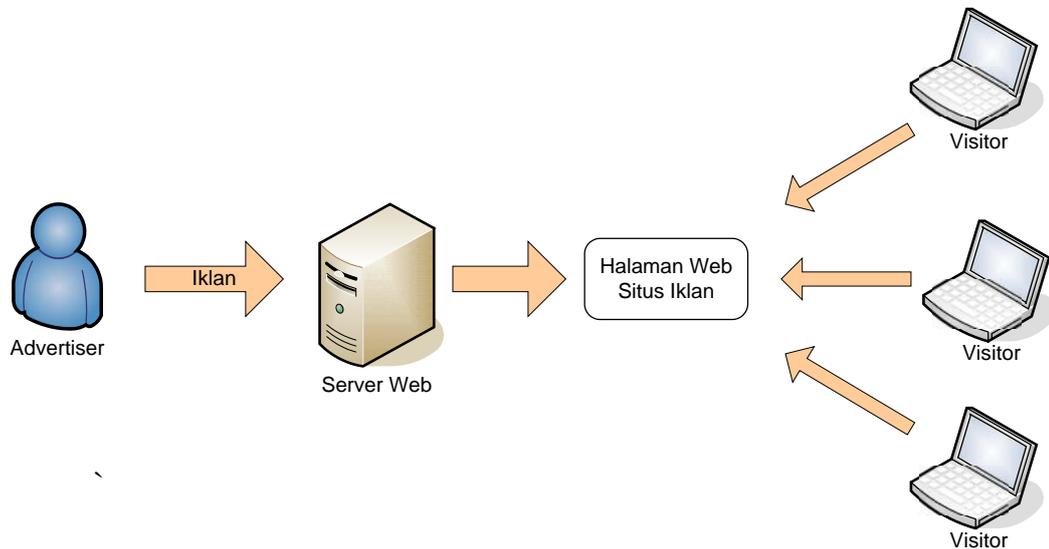
Analisa sistem dibagi dalam dua bagian pembahasan yaitu analisa pada sistem yang berjalan dan analisa pada sistem yang akan dibangun.

4.1.1. Sistem yang Berjalan

Secara umum sistem yang berjalan memiliki alur proses kerja sebagai berikut :

1. *Advertiser* melakukan registrasi untuk pendaftaran sebagai *member* ke situs web penyedia layanan periklanan online. Salah satu contoh penyedia layanan tersebut adalah www.iklanbaris.co.id. *Advertiser* diminta untuk memasukkan data-data seperti nama, *password*, *email*, *website* dan sebagainya. Setelah proses registrasi berhasil dilakukan, maka *advertiser* dapat memasang iklan pada website tersebut.
2. *Advertiser* membuat iklan serta memilih kategori yang sesuai. Setelah semua data lengkap kemudian iklan *diposting* dan di simpan ke *database server*.
3. Setelah iklan *terposting* maka secara otomatis iklan akan langsung ditayangkan pada halaman web situs tersebut agar dapat dilihat oleh pengunjung (*visitor*).

Untuk lebih jelasnya, alur kerja sistem yang sedang berjalan saat ini akan diilustrasikan melalui gambar 4.1 berikut.



Gambar 4.1 Ilustrasi kerja sistem yang berjalan

4.1.2. Sistem yang Akan Dibangun

Sistem baru yang akan dibangun merupakan suatu pengembangan dari cara kerja sistem yang berjalan. Jika pada sistem yang sedang berjalan hanya memiliki satu entitas yaitu *advertiser* yang berperan sebagai sumber data (iklan), maka pada sistem yang akan dibangun ini menggunakan dua entitas utama yaitu *Advertiser* dan *Publisher* yang memiliki keterkaitan satu sama lain dalam penerapan *web service* ini. Oleh sebab itu, analisa akan dibagi kedalam dua bagian sebagai berikut :

4.1.2.1. Analisa pada *Advertiser*

Model kerja yang dilakukan di sisi *advertiser* dapat dijelaskan sebagai berikut :

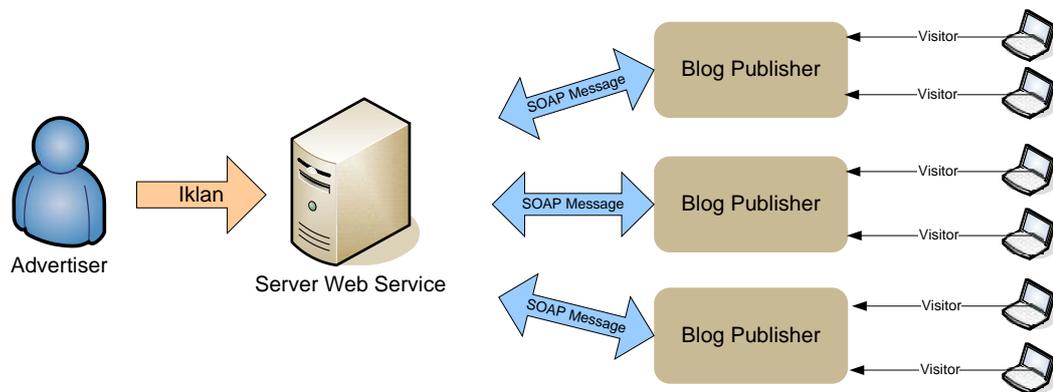
1. *Advertiser* harus membuat *account* sebagai “*advertiser*” dengan memasukkan data diri terlebih dahulu sebelum dapat memasang iklannya ke sistem.

2. *Advertiser* membuat iklan serta memilih kategori yang sesuai untuk iklannya dan memposting ke sistem, kemudian sistem akan menyimpan ke *database*. Pada proses ini, iklan tidak akan ditayangkan di *webpage* sistem, karena iklan akan tayang pada website milik *publisher* yang sebelumnya sudah terdaftar pada sistem.
3. Saat iklan sudah berhasil ditayangkan pada website *publisher* maka setiap iklan ditayangkan atau terdapat *visitor* dari web tersebut yang melakukan klik, semua *record* akan tercatat di *database* dan *advertiser* dapat melihat laporan tersebut pada *account-nya*.

4.1.2.2. Analisa pada *Publisher*

Model kerja yang dilakukan di sisi *publisher* dapat dijelaskan sebagai berikut :

1. *Publisher* harus membuat *account* sebagai “*publisher*” terlebih dahulu dengan memasukkan data diri.
2. *Publisher* harus membuat *channel* / mendaftarkan website yang akan dipasang iklan. *Publisher* dapat memasang lebih dari satu *channel* dalam satu buah *account*.
3. *Publisher* akan mengambil kode/*script* yang digenerate oleh *server*, kemudian kode tersebut akan ditempatkan pada web/*blog* milik *publisher* yang berfungsi untuk menampilkan iklan-iklan dari *advertiser*. Kode tersebut nantinya akan melakukan *request* ke *server*, kemudian *server* akan mengirimkan respon berisi data iklan yang akan ditayangkan. Proses tersebut menggunakan *SOAP Message*.



Gambar 4.2 Sistem yang akan dibangun

4.1.2.3. Analisa kebutuhan

Analisa kebutuhan dari aplikasi yang akan dibangun adalah sebagai berikut :

1. Sistem dapat menayangkan iklan yang tersebar secara *random* ke *website Publisher* sesuai pemilihan kategori yang sama.
2. Sistem dapat mencatat seluruh *history* klik yang terjadi pada iklan, *history* tersebut antara lain :
 - Jumlah klik iklan
 - Jumlah *impression* iklan/ berapa kali iklan tersebut tampil.
 - *Internet Protocol (IP)* yang tercatat setiap terjadi klik.
 - Jenis *browser*
 - Waktu

Berdasarkan data tersebut *advertiser* juga dapat melihat rasio perbandingan antara jumlah klik dengan jumlah *impression (Clickthrough)*

3. Sistem membatasi setiap klik pada iklan berdasarkan IP. Jika satu IP pernah melakukan klik, maka klik selanjutnya dari IP tersebut akan dikatakan *valid* jika memiliki jarak satu hari dari klik yang terekam sebelumnya.

4. Sistem dapat menampilkan dua jenis iklan yaitu iklan teks dan iklan gambar. Bentuk tampilan iklan terdiri atas tiga jenis yaitu Horizontal, Vertikal dan Persegi yang memiliki ukuran berbeda. Detail ukuran dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4.1 Jenis dan Ukuran iklan

No	Nama	Ukuran (dalam <i>pixel</i>)
1	Horizontal	728 x 90
		450 x 90
		234 x 100
2	Vertikal	120 x 600
		160 x 600
		120 x 250
3	Persegi	250 x 250
		200 x 200
		180 x 150
		125 x 125

5. Sistem dapat menampilkan laporan dari hasil klik kepada *advertiser* maupun *publisher*.
6. *Web service* yang disediakan aplikasi bisa menyediakan fungsi-fungsi yang dibutuhkan seperti pada proses *query* untuk menampilkan data iklan yang relevan.

4.1.2.4. Analisa Data

Beberapa data yang dibutuhkan untuk memulai pembuatan sistem adalah sebagai berikut :

1. *Data Member*

Data member berisi informasi *member* yang terbagi kedalam 2 kategori, yaitu *Advertiser* dan *Publisher*.

2. *Data Iklan*

Data iklan berisi tentang informasi iklan yang dipasang oleh *advertiser* ke sistem.

3. Data Channel

Data *channel* menjelaskan URL atau alamat web yang akan digunakan oleh *publisher* untuk menampilkan iklan.

4.1.3. Identifikasi Web Service pada sistem

Berdasarkan hasil analisis terhadap sistem yang akan dibangun maka selanjutnya perlu mengidentifikasi *service-service* yang akan digunakan oleh sistem. Hasil identifikasi *web service* yang akan diimplementasikan dapat dilihat pada tabel 4.2.

Tabel 4.2 Identifikasi Web Service

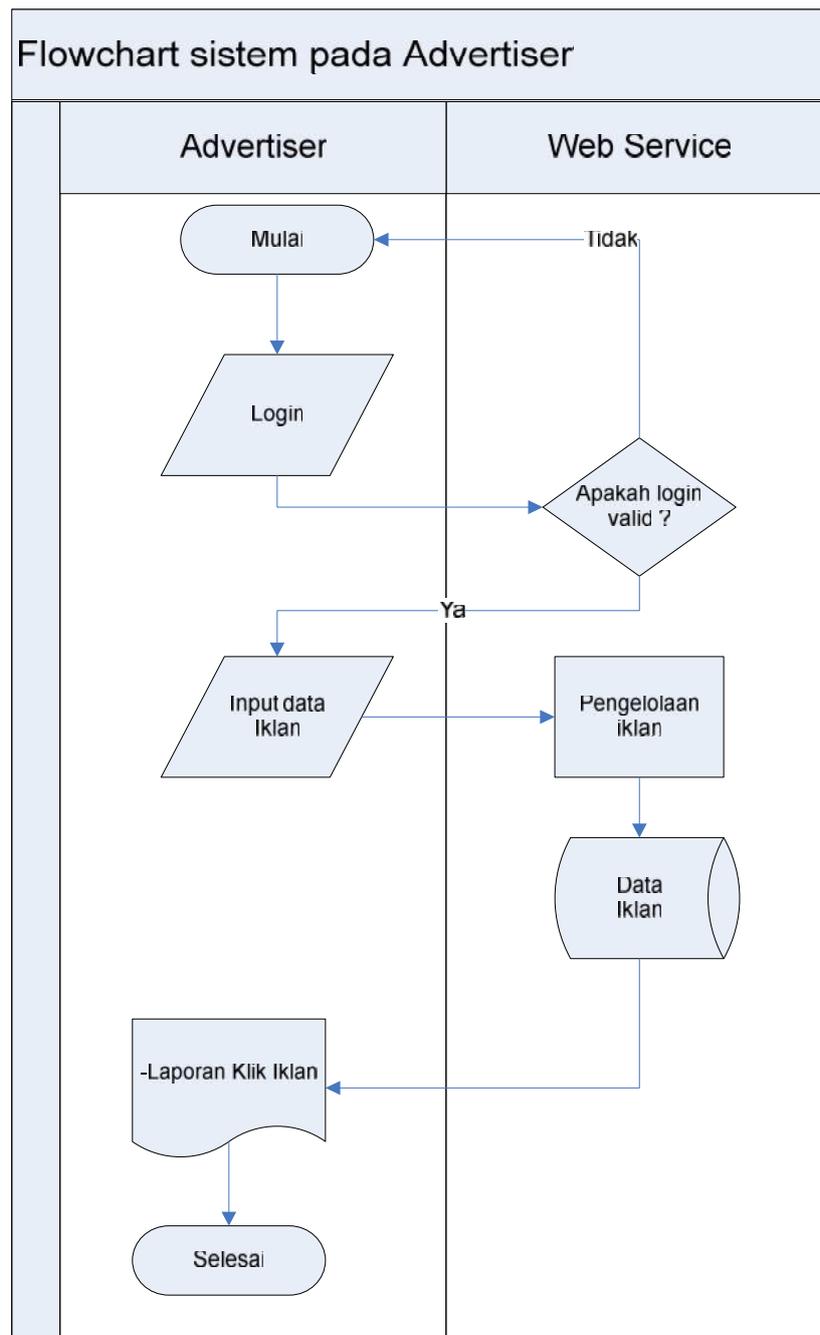
No	Nama Service	Deskripsi
1.	<i>Validate Pub-ID</i>	<i>Service</i> ini berfungsi untuk melakukan validasi terhadap ID milik <i>publisher</i> saat akan mengakses <i>web service</i> . <i>Client</i> akan mengirimkan inputan berupa <i>Publisher ID</i> ke <i>server</i> kemudian <i>output</i> yang dihasilkan berupa nilai <i>Boolean</i> untuk mengetahui apakah ID tersebut terdaftar atau tidak.
2.	<i>Validate Channel</i>	<i>Service</i> ini berfungsi untuk melakukan validasi terhadap <i>channel</i> yang telah di- <i>input</i> oleh <i>publisher</i> pada sistem. Parameter input berupa <i>Publisher ID</i> . <i>Output</i> yang dihasilkan adalah nilai <i>Boolean</i> untuk mengetahui apakah <i>channel</i> sudah terdaftar.
4.	<i>Get Ads</i>	<i>Service</i> ini berfungsi untuk mendapatkan dan mengirimkan data iklan yang dipasang oleh <i>advertiser</i> . Parameter <i>input</i> berupa <i>publisher ID</i> . <i>Output</i> yang dihasilkan adalah iklan.

4.1.4. Flowchart Sistem

Flowchart Sistem merupakan suatu cara yang digunakan untuk menggambarkan langkah-langkah kerja dari sistem yang akan dibangun.

4.1.4.1. Flowchart Pada Advertiser

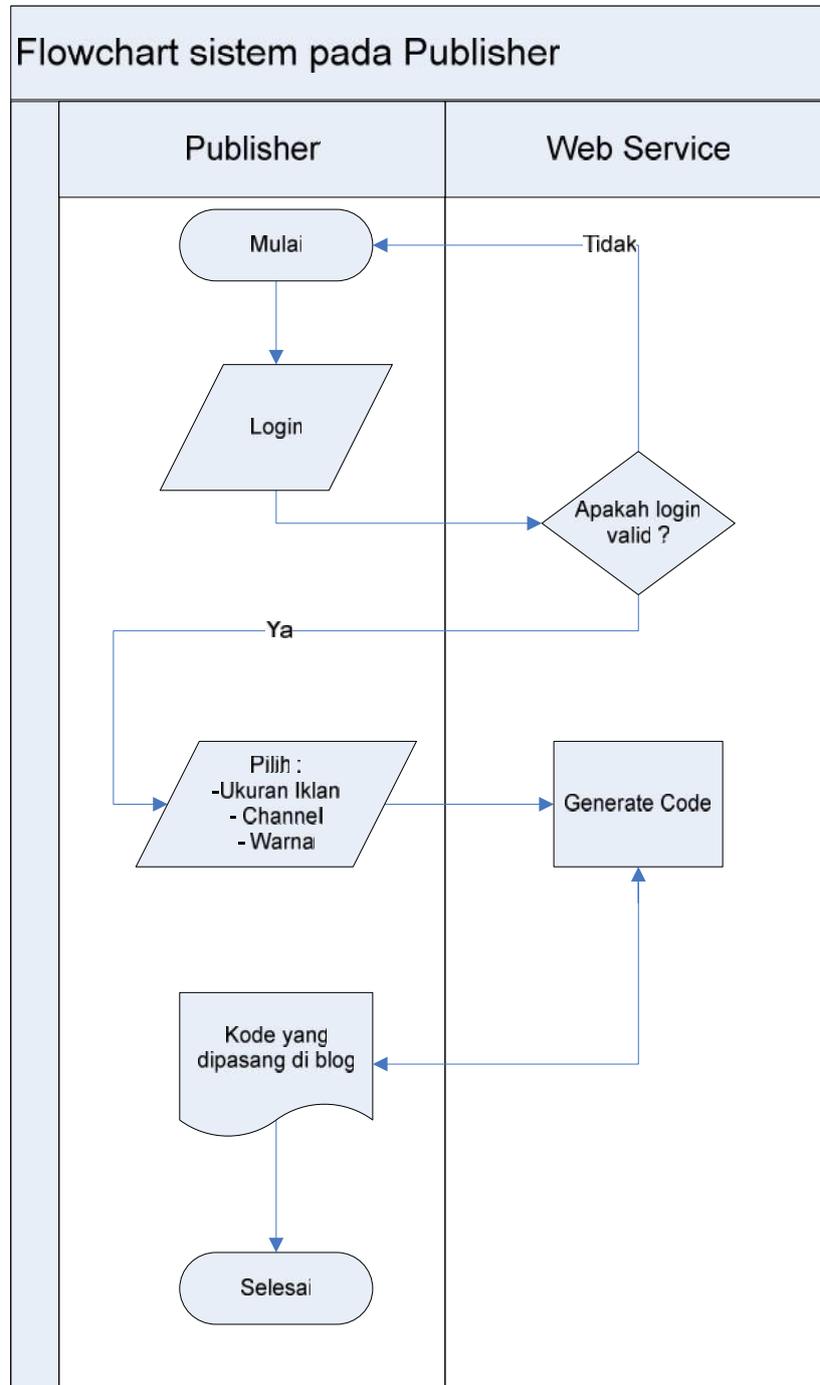
Berikut ini adalah *flowchart* sistem pada sisi *advertiser*.



Gambar 4.3 *Flowchart* pada Advertiser

4.1.4.2. Flowchart Pada Publisher

Berikut ini adalah *flowchart* sistem pada sisi *publisher*



Gambar 4.4 Flowchart pada Publisher

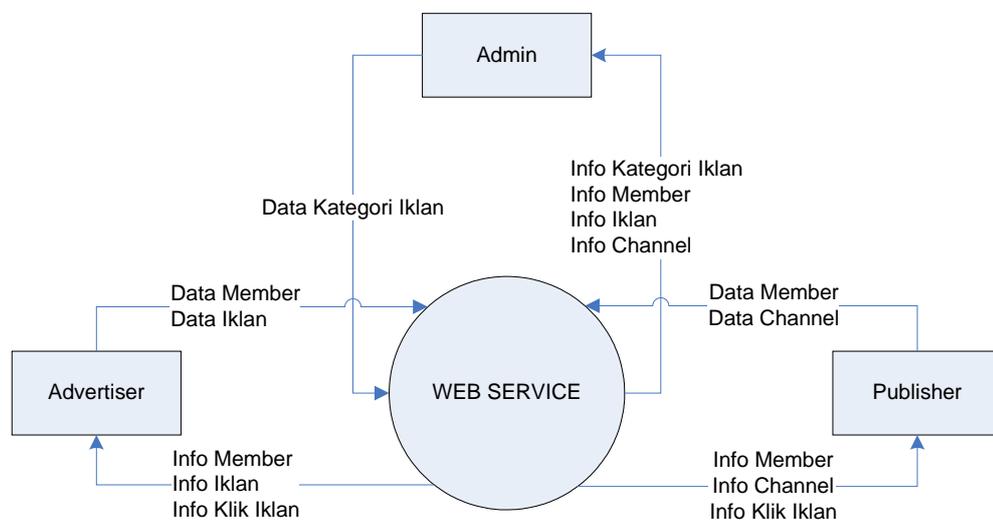
4.1.5. Analisa Fungsional Sistem

Analisa fungsional digunakan untuk menggambarkan bagaimana masukan diproses oleh aplikasi menjadi keluaran yang diharapkan oleh pengguna sistem. Analisa fungsional memuat beberapa diagram alir data yang memperlihatkan aliran data dari luar sistem yang diproses oleh sistem.(kadang melibatkan penyimpanan sistem) kemudian menghasilkan keluaran yang berguna.

Secara garis besar fungsi utama dari perangkat lunak yang akan dikembangkan dapat dilihat di Diagram Konteks (*Context Diagram*) dan diagram aliran data (*Data Flow Diagram*) yang akan diberikan. Seperti berikut ini:

4.1.5.1. Context Diagram

Context Diagram digunakan untuk menggambarkan proses kerja sistem secara umum. *Context Diagram* adalah *Data Flow Diagram* (DFD) yang menggambarkan garis besar operasional sistem.



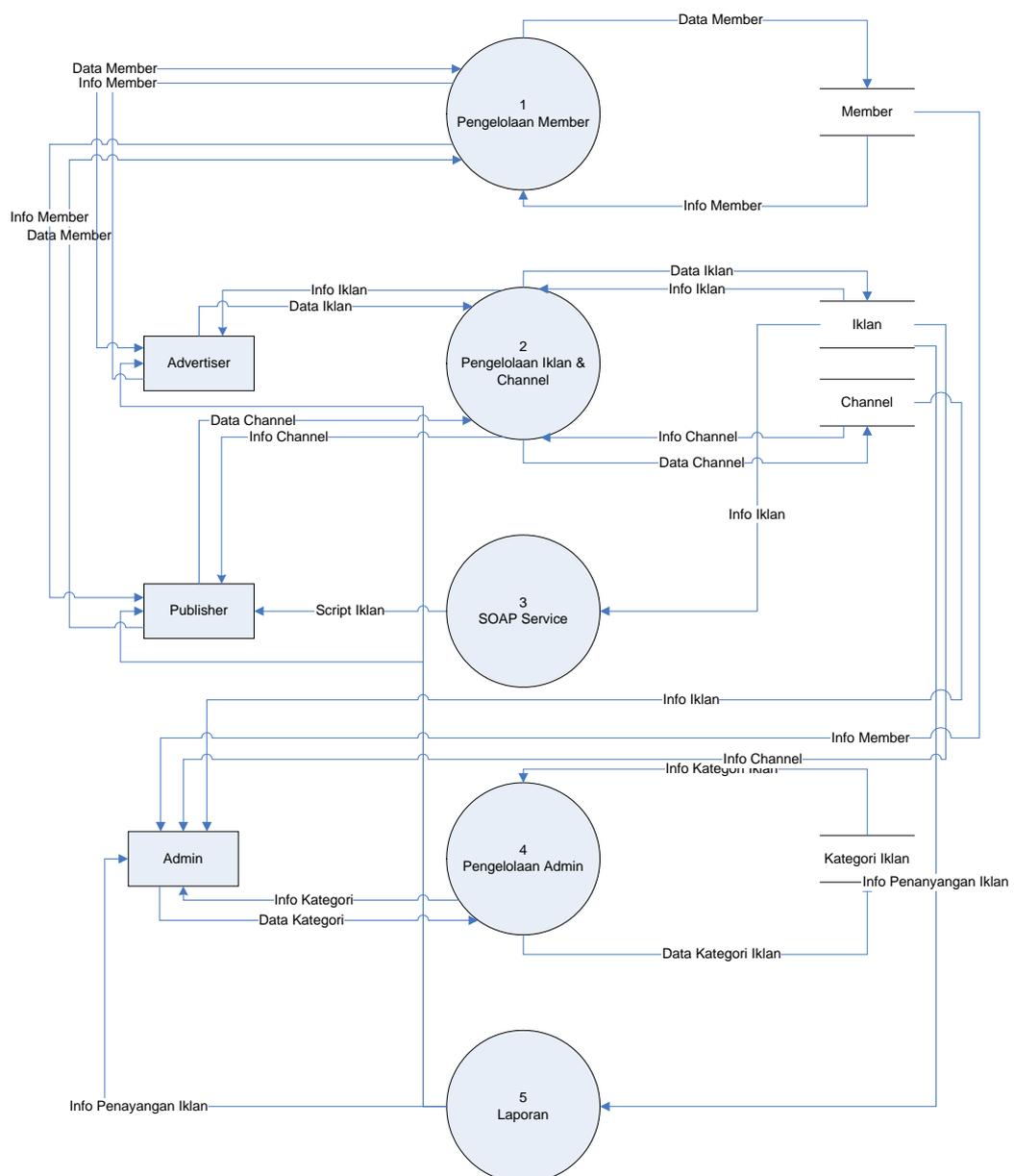
Gambar 4.5 *Context Diagram*

Advertiser dalam sistem berfungsi sebagai orang yang memiliki iklan dan akan memasang iklannya pada sistem, sedangkan *publisher* disini berfungsi sebagai media tempat menampilkan iklan dari *advertiser* di blog/website mereka.

4.1.5.2. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) sering digunakan untuk menggunakan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir, atau lingkungan fisik dimana data tersebut tersimpan.

4.1.5.3. DFD Level 1 WEB SERVICE



Gambar 4.6 DFD Level 1 *Web service*

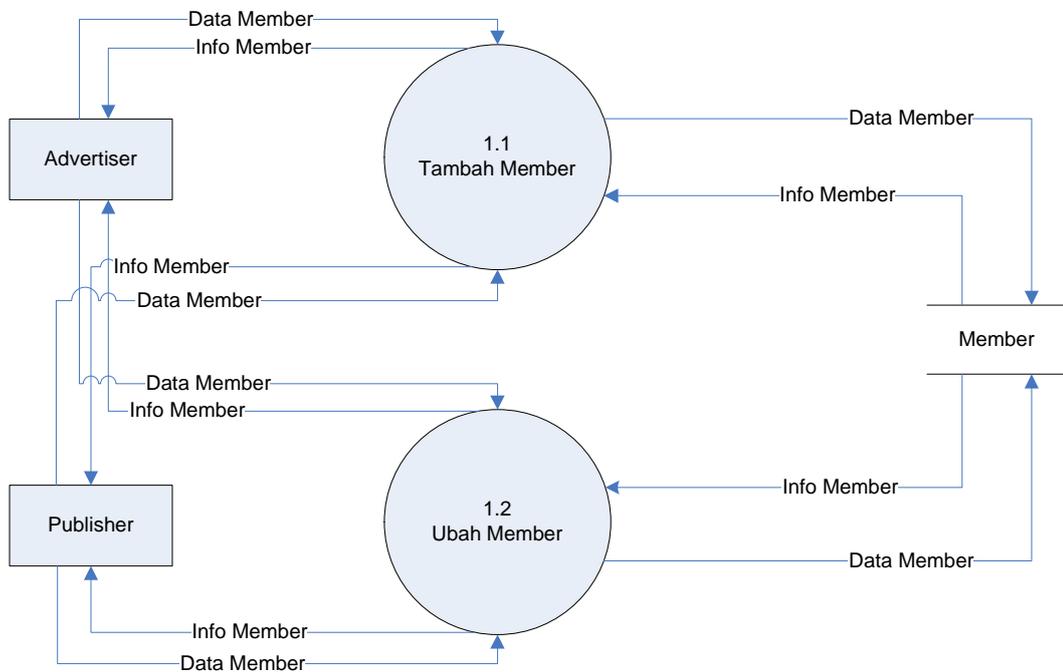
Tabel 4.3 DFD Level 1 *Web service*

No.	Nama Proses	Deskripsi
1.	<i>Member</i>	Proses registrasi yang dilakukan oleh <i>member</i> yaitu <i>advertiser</i> dan <i>publisher</i>
2.	Pengelolaan iklan & <i>channel</i>	Proses dalam pengelolaan iklan oleh <i>member</i> seperti menambah iklan, menghapus iklan
3.	SOAP <i>service</i>	Proses generate code oleh <i>publisher</i> untuk di pasang di situs web mereka.
4.	Pengelolaan <i>Admin</i>	Proses yang terjadi pada halaman administrator
5.	Laporan	Proses untuk melihat laporan hasil penayangan iklan

Tabel 4.4 Keterangan Aliran Data DFD Level 1 *Web service*

Nama Aliran Data	Deskripsi
Data_ <i>Member</i>	Data dari <i>advertiser</i> dan <i>publisher</i>
Info_ <i>Member</i>	Informasi data dari <i>member</i>
Data_ <i>Iklan</i>	Data iklan dari <i>advertiser</i>
Info_ <i>iklan</i>	Informasi iklan dari datastore iklan
Data_ <i>channel</i>	Data <i>channel</i> dari <i>publisher</i>
Info_ <i>channel</i>	Informasi <i>channel</i> dari datastore <i>channel</i>
Data_ <i>kategori</i>	Data kategori dari <i>admin</i>
Info_ <i>kategori</i>	Informasi kategori dari datastore kategori iklan
Info_ <i>penayangan_iklan</i>	Informasi hasil klik iklan secara detail

4.1.5.4. DFD Level 2 Proses *Member*



Gambar 4.7 DFD Level 2 Proses *Member*

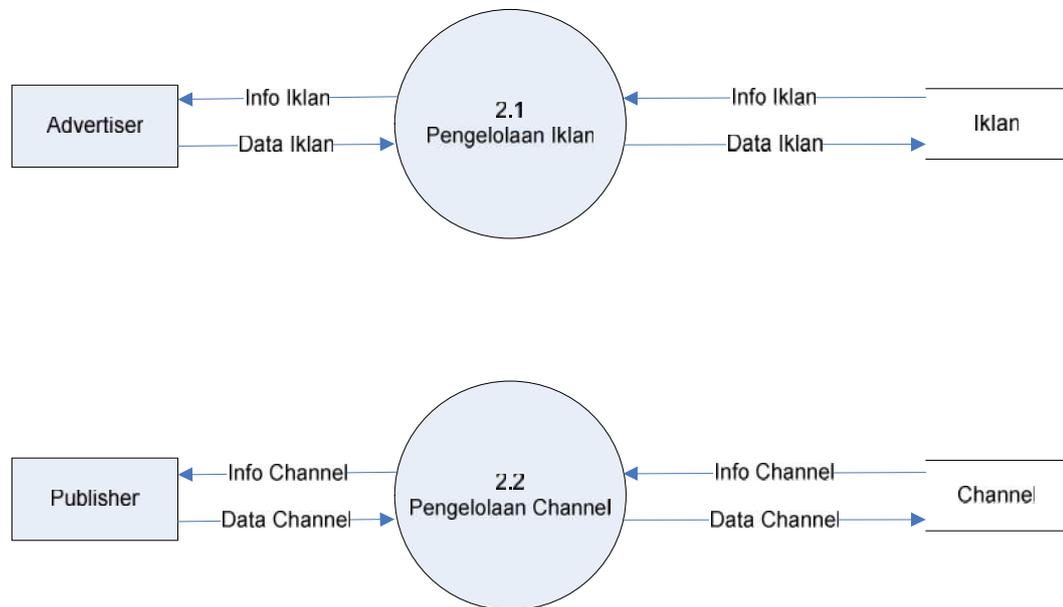
Tabel 4.5 DFD Level 2 Proses Registrasi

No.	Nama Proses	Deskripsi
1.	Tambah <i>member</i>	Proses penambahan <i>member</i> baru
2.	Ubah <i>member</i>	Proses untuk mengubah data <i>member</i>

Tabel 4.6 Keterangan Aliran Data DFD Level 2 Proses Registrasi

Nama Aliran Data	Deskripsi
Data_ <i>Member</i>	Data <i>member</i> dari <i>advertiser</i> dan <i>publisher</i>
Info_ <i>Member</i>	Informasi data dari <i>member</i>

4.1.5.5. DFD Level 2 Proses Pengelolaan iklan & Channel



Gambar 4.8 DFD Level 2 Proses Pengelolaan iklan & channel

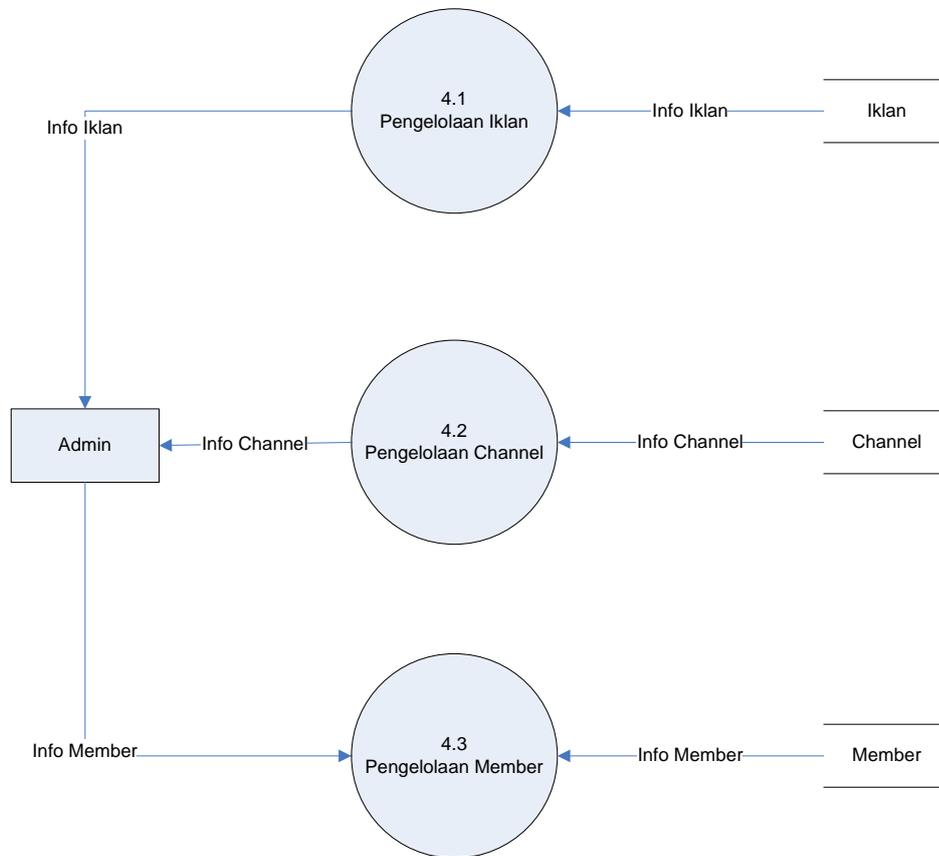
Tabel 4.7 DFD Level 2 Proses Pengelolaan iklan & channel

No.	Nama Proses	Deskripsi
1.	Pengelolaan Iklan	Proses untuk Pengelolaan iklan oleh <i>advertiser</i>
2.	Pengelolaan <i>channel</i>	Proses untuk Pengelolaan <i>channel</i> oleh <i>publisher</i>

Tabel 4.8 Keterangan Aliran Data DFD Level 2 Proses iklan & channel

Nama Aliran Data	Deskripsi
Data_Iklan	Data iklan dari <i>advertiser</i>
Info_Iklan	Informasi data iklan
Data_Channel	Data <i>channel</i> dari <i>publisher</i>
Info_Channel	Informasi data <i>channel</i>

4.1.5.6. DFD Level 2 Proses Pengelolaan *admin*



Gambar 4.9 DFD Level 2 Proses Pengelolaan *admin*

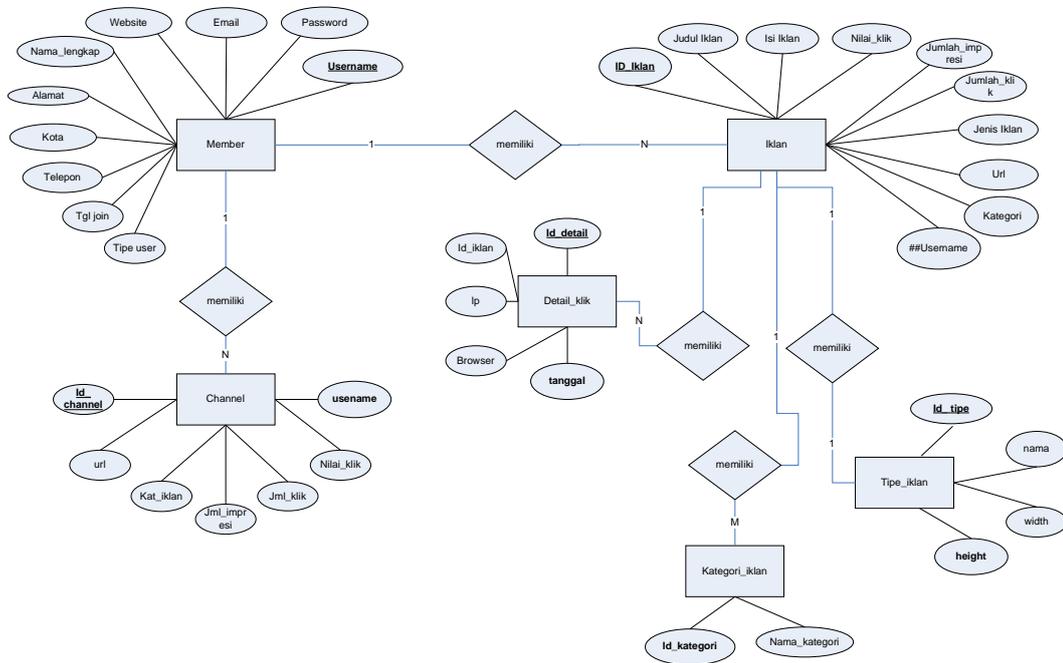
Tabel 4.9 DFD Level 2 Proses Pengelolaan *Admin*

No.	Nama Proses	Deskripsi
1.	Pengelolaan Iklan	Proses untuk Pengelolaan iklan oleh <i>advertiser</i>
2.	Pengelolaan <i>channel</i>	Proses untuk Pengelolaan <i>channel</i> oleh <i>publisher</i>
3.	Pengelolaan <i>member</i>	Proses untuk pengelolaan <i>member</i>

Tabel 4.10 Keterangan Aliran Data DFD Level 2 Proses *Admin*

Nama Aliran Data	Deskripsi
Info_Iklan	Informasi data iklan
Info_Channel	Informasi data <i>channel</i>
Info_Member	Informasi data <i>member</i>

4.1.6. Entity Relationship Diagram (ER-Diagram)



Gambar 4.10 ER-Diagram

Tabel 4.11 Keterangan entitas pada ERD

Nama	Deskripsi	Atribut	Primary Key
<i>Member</i>	Menyimpan data <i>member</i>	✓ Username ✓ Password ✓ Email ✓ Website ✓ Nama_lengkap ✓ Alamat ✓ Kota ✓ Telepon ✓ Tgl_join ✓ Tipe_user	Username
<i>Channel</i>	Menyimpan data <i>channel</i>	✓ ID_channel ✓ Url	ID_Channel

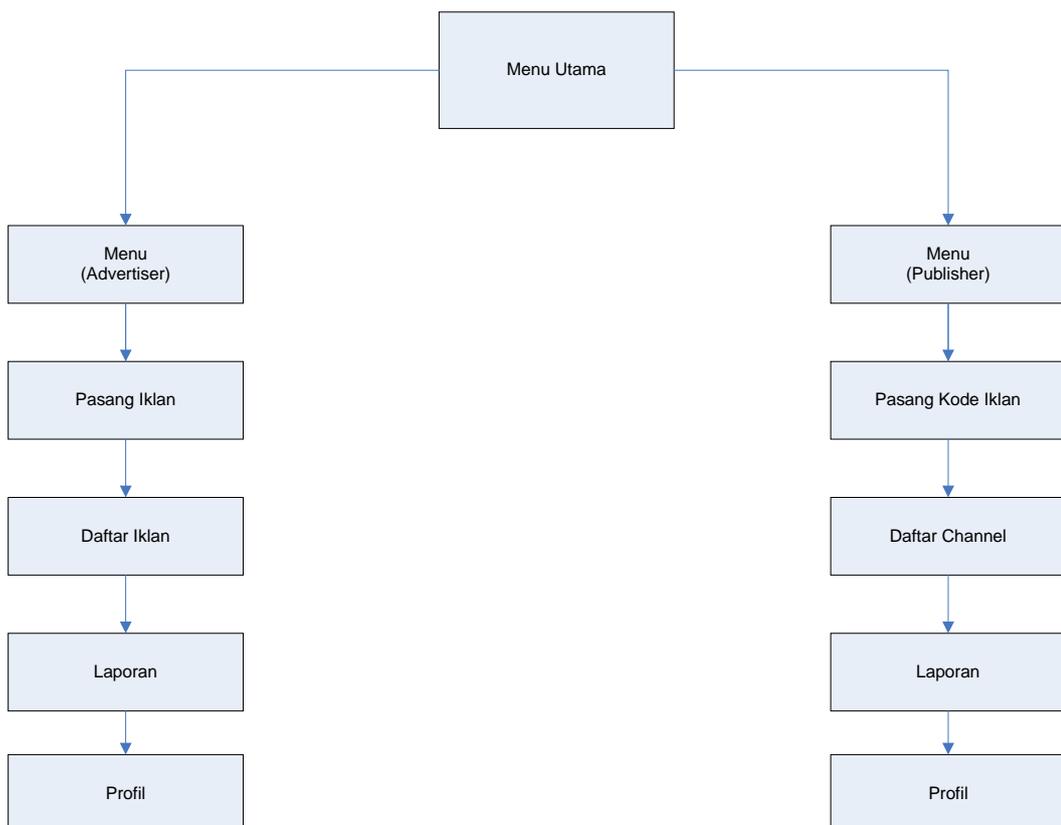
		<ul style="list-style-type: none"> ✓ kat_iklan ✓ jml_impresi ✓ jml_klik ✓ nilai_klik ✓ username 	
Iklan	Menyimpan data iklan	<ul style="list-style-type: none"> ✓ ID_iklan ✓ Judul ✓ Isi_iklan ✓ url ✓ idtipe ✓ lokasi_image ✓ kat_iklan ✓ deposit ✓ nilai_klik ✓ jml_impresi ✓ jenis_iklan ✓ username 	ID_iklan
Detail klik	Menyimpan data detail klik	<ul style="list-style-type: none"> ✓ id_detail ✓ id_iklan ✓ IP ✓ Browser ✓ tanggal 	id_detailklik
Tipe_iklan	Menyimpan data tipe iklan	<ul style="list-style-type: none"> ✓ id_tipe ✓ nama ✓ width ✓ height 	id_tipe
Kategori_iklan	Menyimpan data kategori iklan	<ul style="list-style-type: none"> ✓ id_kategori ✓ nama_kategori 	Id_kategori

4.2. Perancangan

Dalam tahap perancangan akan terbagi menjadi beberapa bagian pembahasan yaitu perancangan struktur menu, basis data serta perancangan antar muka yang akan dijelaskan sebagai berikut.

4.2.1. Perancangan Menu

Dalam pemakaian sistem ini diperlukan susunan daftar pilihan/menu sehingga pengguna yang belum terbiasa dengan sistem juga dapat menggunakan sistem ini. Pengguna akan dihadapkan pada berbagai alternatif menu yang ada. Dalam menentukan pilihannya, pengguna dapat menggunakan tombol tertentu dan setiap pilihan akan menghasilkan tampilan/halaman tertentu. Sistem yang akan dibangun memiliki menu dan sub-sub menu yang digambarkan pada bagan di bawah ini:



Gambar 4.11 Rancangan Menu

4.2.2. Perancangan Basis Data

Tahapan perancangan basis data digunakan untuk membuat detail data yang akan dipersiapkan pada tahap implementasi selanjutnya. Berikut adalah hasil perancangan basis data :

Tabel 4.12 Perancangan Tabel *Member*

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>
1	Username	Varchar(10)	NOT NULL
2	Password	Varchar(20)	NOT NULL
3	Email	Varchar(50)	NOT NULL
4	Website	Varchar(50)	NOT NULL
5	Nama Lengkap	Varchar(30)	NOT NULL
6	Alamat	Varchar(100)	NOT NULL
7	Kode Pos	Varchar(5)	NULL
8	Telp/hp	Varchar(15)	NULL
9	Tgl_join	Datetime	NOT NULL
10	Tipeuser	char(1)	NOT NULL

Tabel 4.13 Perancangan Tabel *Channel*

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>
1	ID_channel	Integer (5)	NOT NULL
2	Url	Varchar(20)	NOT NULL
3	kat_iklan	Varchar(50)	NOT NULL
4	jml_impresi	Integer (5)	NOT NULL
5	jml_klik	Integer (5)	NOT NULL
6	nilai_klik	Integer (5)	NOT NULL
7	username	Varchar(10)	NOT NULL

Tabel 4.14 Perancangan Tabel Iklan

No	Nama Field	Tipe Data	Null
1	ID_iklan	Integer (10)	NOT NULL
2	Judul	Varchar(25)	NOT NULL
3	Isi_iklan	Varchar(60)	NOT NULL
4	url	Varchar(50)	NOT NULL
5	Id_tipe	Integer(5)	NOT NULL
6	lokasi_image	Varchar(50)	NOT NULL
7	kat_iklan	Varchar(50)	NOT NULL
8	deposit	Integer(5)	NOT NULL
9	nilai_klik	Integer(5)	NOT NULL
10	jml_impresi	Integer(5)	NOT NULL
11	jenis_iklan	Char(1)	NOT NULL
12	username	Varchar(10)	NOT NULL

Tabel 4.15 Perancangan Tabel detail_klik

No	Nama Field	Tipe Data	Null
1	id_detail	Integer (10)	NOT NULL
2	id_iklan	Integer (10)	NOT NULL
3	IP	Varchar(20)	NOT NULL
4	Browser	Varchar(50)	NOT NULL
5	Tanggal	Datetime	NOT NULL

Tabel 4.16 Perancangan Tabel kategori_iklan

No	Nama Field	Tipe Data	Null
1	id_kategori	Integer (10)	NOT NULL
2	nama_kategori	varchar (30)	NOT NULL

4.2.3. Perancangan Antar Muka (*Interface*)

Interface sistem merupakan suatu sarana pengembangan sistem yang ditujukan untuk mempermudah pemakai berkomunikasi dengan sistem yang ada dan konsisten data juga ditunjukkan dalam *interface* tersebut. Penekanan *interface* meliputi tampilan yang baik, mudah dipahami dan tombol-tombol yang familiar.

The wireframe illustrates the layout of the main page. At the top is a large rectangular area labeled "Banner". Below the banner is a horizontal navigation menu with five buttons: "Home", "Advertiser", "Publisher", "Layanan", and "Kontak Kami". The main content area is divided into three sections. On the left, there is a box labeled "Advertiser" containing a "Login" button. In the center, there is a box labeled "Publisher" containing a "Login" button. On the right, there is a login form with two input fields labeled "Username" and "Password", and a "Login" button below them.

Gambar 4.12 Rancangan Halaman Utama

Penjelasan lebih rinci mengenai perancangan antarmuka lainnya dapat dilihat pada lampiran A.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas tentang implementasi dan pengujian sistem berdasarkan hasil analisis yang telah dilakukan sebelumnya.

5.1 Implementasi Sistem

Implementasi sistem meliputi tahap pengkodean program (*coding*), uji sub sistem (*testing*) dan penggabungan sub sistem. Pada tahap ini, sistem yang telah selesai dirancang bangun akan dioperasikan dan selanjutnya dilakukan pengujian untuk melihat sejauh mana kebersesuaian hasil dengan tujuan yang dicapai dari pembuatan sistem ini.

Rancangan aplikasi *web service* periklanan online ini dibuat dengan menggunakan perangkat lunak PHP versi 5 dan *Database* yang digunakan adalah MySQL versi 5.0.

5.1.1 Batasan Implementasi

Agar pelaksanaan implementasi tidak keluar dari fokus pembahasan, maka perlu dijelaskan batasan implementasi sebagai berikut:

1. Menggunakan bahasa pemrograman PHP versi 5 dan *database* MySQL versi 5.
2. *Web service* hanya menampilkan iklan berjenis teks dan gambar.
3. Setiap klik iklan yang berasal dari IP (*Internet Protocol*) yang sama pada hari yang sama akan dihitung satu klik.

5.1.2 Lingkungan Implementasi

Lingkungan implementasi adalah lingkungan dimana aplikasi ini dikembangkan. Lingkungan implementasi sistem ada dua yaitu lingkungan perangkat keras dan lingkungan perangkat lunak, dengan spesifikasi sebagai berikut:

1. Perangkat Keras

Perangkat keras yang digunakan dalam rancang bangun sistem mempunyai spesifikasi sebagai berikut:

- a. *Processor* : Intel Atom 1.5GHz
- b. *Memory* : 1 GHz
- c. *Hardisk* : 320 GB

2. Perangkat Lunak

Perangkat lunak yang digunakan adalah sebagai berikut:

- a. Sistem Operasi : Windows XP Professional SP 2
- b. Bahasa Pemrograman : PHP 5.2.6
- c. DBMS : *Database* MySQL 5.0.51b
- d. *Browser* : Mozilla Firefox 3.0
Google Chrome v14.0.794.0

3. *Web Server*

Aplikasi ini dijalankan pada *server* hosting serta menggunakan domain yang beralamat di :

<http://www.xmladservice.info>

5.1.3 Hasil Implementasi

Berikut ini adalah hasil implementasi dari sistem yang telah di rancang. Pembahasan akan terbagi menjadi dua bagian, yaitu implementasi SOAP *Service* dan WSDL (*Web Service Definition Language*) serta implementasi tampilan antar muka sistem.

5.1.3.1 Implementasi SOAP *Service* dan dokumen WSDL (*Web Service Definition Language*)

Seperti yang telah dijelaskan pada sebelumnya untuk membangun sebuah *web service* maka dibutuhkan pendefinisian terhadap *service* itu sendiri. Implementasi terhadap *service* yang akan digunakan dapat dilihat pada tabel 5.1 berikut ini.

Tabel 5.1 Identifikasi *Service*

No	Nama <i>Service</i>	Deskripsi
1.	<i>Validate PubID</i>	'input' => 'xsd:Integer' - Parameter input 'output' => 'xsd:Boolean' - Parameter output
2.	<i>Validate Channel</i>	'input' => 'xsd:Integer' - Parameter input 'output' => 'xsd:Boolean' - Parameter output
3.	<i>Get Ads</i>	'id_channel' => 'xsd:Integer' - Parameter input 'adstype' => 'xsd:Integer' - Parameter input 'output' => 'tns:Array' - Parameter output

Detail masing-masing operasi *service* adalah sebagai berikut :

1. *Validate_PubID*

```
Name: Validate_PubID
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Validate_PubID
Style: rpc
Input:
  use: encoded
  namespace: urn:WSDLiklan
  encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
  message: Validate_PubIDRequest
  parts:
    input: xsd:String
Output:
  use: encoded
  namespace: urn:WSDLiklan
  encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
  message: Validate_PubIDResponse
  parts:
    output: xsd:Boolean
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Validasi Publisher
```

2. *Validate_Channel*

Name: Validate_Channel
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Validate_Channel
Style: rpc
Input:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Validate_ChannelRequest
 parts:
 input: xsd:Integer
Output:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Validate_ChannelResponse
 parts:
 output: xsd:Boolean
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Validasi Channel

3. *Get_Ads*

Name: Get_Ads
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Get_Ads
Style: rpc
Input:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Get_AdsRequest
 parts:
 id_channel: xsd:Int
 adstype: xsd:Int
Output:

```

    use: encoded
    namespace: urn:WSDLiklan
    encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
    message: Get_AdsResponse
    parts:
        output: tns:ListArray
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Mengambil data dari iklan dari database

```

WSDL untuk *service* tersebut adalah sebagai berikut (WSDL dapat dilihat di <http://xmladservice.info/soapcode/soapservice.php?wsdl>) :

```

<definitions xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="h
ttp://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="ur
n:WSDLiklan" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soa
p/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http
://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:WSDLiklan
">
<types>
<xsd:schema targetNamespace="urn:WSDLiklan">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encod
ing/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
<xsd:complexType name="ListArray">
<xsd:complexContent>
<xsd:restriction base="SOAP-ENC:Array">
<xsd:element name="0" type=""/>
<xsd:attribute ref="SOAP-
ENC:arrayType" wsdl:arrayType="xsd:string[]" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
</types>
<message name="Validate_PubIDRequest">
<part name="input" type="xsd:String" />
</message>
<message name="Validate_PubIDResponse">

```

```

<part name="output" type="xsd:Boolean"/>
</message>
<message name="Validate_ChannelRequest">
<part name="input" type="xsd:Integer"/>
</message>
<message name="Validate_ChannelResponse">
<part name="output" type="xsd:Boolean"/>
</message>
<message name="Get_AdsRequest">
<part name="id_channel" type="xsd:Int"/>
<part name="adstype" type="xsd:Int"/>
</message>
<message name="Get_AdsResponse">
<part name="output" type="tns:ListArray"/>
</message>
<portType name="Web service with SOAPPortType">
<operation name="Validate_PubID">
<documentation>Validasi Publisher</documentation>
<input message="tns:Validate_PubIDRequest"/>
<output message="tns:Validate_PubIDResponse"/>
</operation>
<operation name="Validate_Channel">
<documentation>Validasi Channel</documentation>
<input message="tns:Validate_ChannelRequest"/>
<output message="tns:Validate_ChannelResponse"/>
</operation>
<operation name="Get_Ads">
<documentation>Mengambil data dari iklan dari
database</documentation>
<input message="tns:Get_AdsRequest"/>
<output message="tns:Get_AdsResponse"/>
</operation>
</portType>
<binding name="Web service with SOAPBinding" type="tns:Web
service with SOAPPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.
org/soap/http"/>
<operation name="Validate_PubID">
<soap:operation soapAction="urn:WSDLiklan#Validate_PubID" st
yle="rpc"/>
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
<output>

```

```
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="Validate_Channel">
<soap:operation soapAction="urn:WSDLiklan#Validate_Channel"
style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="Get_Ads">
<soap:operation soapAction="urn:WSDLiklan#Get_Ads" style="rp
c" />
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
</binding>
<service name="Web service with SOAP">
<port name="Web service with SOAPPort" binding="tns:Web
service with SOAPBinding">
<soap:address location="http://xmladservice.info/soapcode/so
apserver.php" />
</port>
</service>
</definitions>
```

5.1.3.2 Tampilan Antarmuka (*Interface*)

Tampilan antarmuka terbagi menjadi beberapa sisi yaitu sisi *publisher*, *advertiser*.

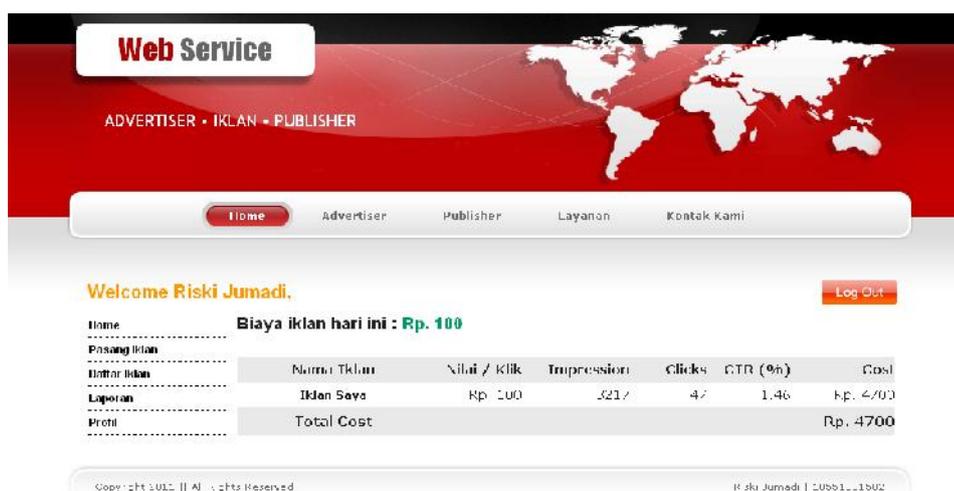
5.1.3.2.1 Tampilan Halaman Utama



Gambar 5.1 Tampilan Halaman Utama

User akan langsung dihadapkan pada pilihan untuk mendaftar sebagai *Advertiser* atau *Publisher*.

5.1.3.2.2 Tampilan Halaman *Advertiser*



Gambar 5.2 Tampilan Halaman *Advertiser*

Gambar 5.2 adalah tampilan pada akun *advertiser* setelah melakukan proses *login*.

5.1.3.2.3 Tampilan Halaman *Publisher*

The screenshot shows the 'Web Service' interface for a publisher. At the top, there is a navigation bar with 'Home', 'Advertiser', 'Publisher', 'Layanan', and 'Kontak Kami'. Below this, a welcome message reads 'Welcome Riski Jumadi,' with a 'Log Out' button. A green notification states 'Klik Iklan hari ini : Rp. 100'. A table displays advertising performance data:

Channel	Nama Iklan	Nilai / Klik	Impression	Clicks	CTR (%)	Earning
Operan	Ad Content	Rp. 100	2641	32	1.21	Rp. 3200
Profil	Total Pendapatan					Rp. 3200

At the bottom, there is a copyright notice: 'Copyright ©2011. All rights reserved. Riski Jumadi | 10511001802'.

Gambar 5.3 Tampilan Halaman *Publisher*

Gambar 5.3 adalah tampilan pada akun *publisher* setelah melakukan proses *login*.

Penjelasan lebih rinci mengenai hasil implementasi antarmuka lainnya dapat dilihat pada lampiran B.

5.2 Pengujian Sistem

Pengujian dilakukan dengan tujuan untuk menjamin sistem yang dibuat sesuai dengan hasil analisis dan perancangan dan menghasilkan satu kesimpulan. Sebelum sistem diimplementasikan terlebih dahulu harus dipastikan program bebas dari kesalahan-kesalahan yang mungkin terjadi.

Dalam hal ini pengujian akan terbagi menjadi dua bagian yaitu pengujian aplikasi pada *server* dan pengujian pada *client*.

5.2.1 Pengujian Aplikasi Server

Pengujian ini dilakukan pada aplikasi *server*. Pengujian dilakukan pada halaman *advertiser* dan *publisher* karena kedua user memiliki tampilan menu yang berbeda. Cara pengujian yang dilakukan menggunakan metode *Black Box*. Hasil dari pengujian adalah sebagai berikut.

5.2.1.1 Identifikasi dan Pengujian Menu Login

Prekondisi : Halaman Utama Aplikasi

Tabel 5.2 Hasil Pengujian Menu Login

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu <i>login</i>	Masukkan <i>username</i> dan <i>password</i> dan klik tombol <i>Login</i>	- <i>Username</i> atau <i>password</i> salah - <i>Username</i> dan <i>password</i> benar	- Pesan error “ <i>Login Invalid !</i> ” - Tampil ke menu halaman utama <i>member</i>	Berhasil

5.2.1.2 Identifikasi dan Pengujian Menu Profil

Prekondisi : Halaman Utama *Member*

Tabel 5.3 Hasil Pengujian Menu Profil

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu profil	- Masukkan Data pribadi dan klik tombol <i>edit</i> -	- Data pribadi <i>member</i>	- Tampilan pesan “Data berhasil di <i>edit</i> ”	Berhasil

Tabel 5.3 Hasil Pengujian Menu Profil

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu profil	- Input <i>password</i> lama dan baru Klik Submit.	- <i>Password</i> lama dan <i>password</i> baru	- Kembali ke menu profil	Berhasil

Penjelasan lebih rinci mengenai hasil pengujian lainnya dapat dilihat pada lampiran C.

Dari hasil pengujian pada aplikasi tersebut dapat disimpulkan bahwa seluruh menu dapat bekerja dengan baik dan memperoleh hasil seperti yang diinginkan.

5.2.2 Pengujian *Client / Channel*

Pengujian ini ditujukan untuk dapat melihat bahwa iklan yang di *request* oleh *client* dan dapat diproses oleh *server* dengan mengirimkan *response* serta menampilkan data iklan pada *client (Channel Publisher)*.

Berikut adalah daftar *channel* yang akan diuji.

Tabel 5.4 Daftar *Channel*

No.	<i>URL</i>	Ukuran Iklan
1	http://anaisine.blogspot.com	125 x 125
2	http://whitenews.blogspot.com/	728 x 90

5.2.2.1 Skenario Pengujian

Skenario pengujian yang akan dilakukan adalah sebagai berikut :

1. *Publisher login* ke aplikasi kemudian mengambil *script* yang disediakan oleh sistem.
2. *Publisher* meng-*copy script* tersebut kedalam blog/*website* miliknya.

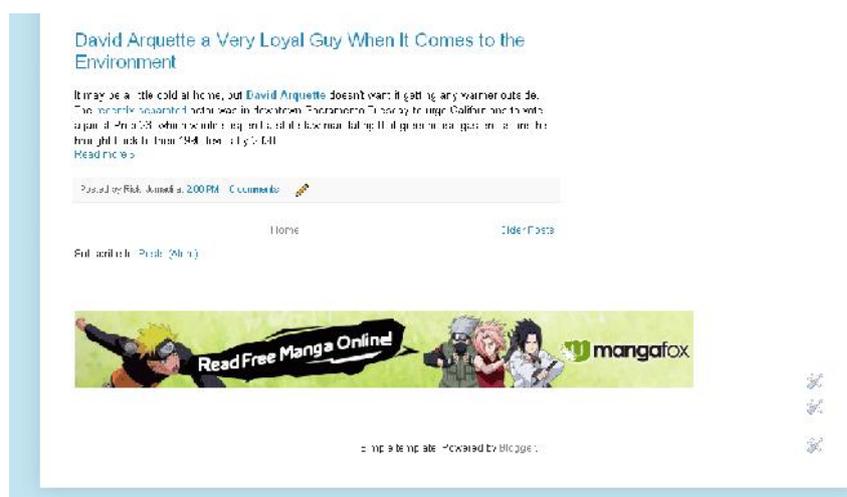
3. *Script* tersebut akan melakukan *request* ke *server* untuk mendapatkan iklan.

5.2.2.2 Hasil Pengujian

Setelah dilakukan pengujian maka iklan akan tayang pada *website* tersebut seperti gambar berikut ini.



Gambar 5.4 Hasil pengujian pada <http://anaisine.blogspot.com>



Gambar 5.5 Hasil pengujian pada <http://whiteneews.blogspot.com>

SOAP *Message* yang diperoleh dari hasil pengujian tersebut adalah sebagai berikut :

1. <http://anaisine.blogspot.com>

SOAP Request :

```
POST /soapcode/soapserver.php HTTP/1.0
Host: xmladservice.info
User-Agent: NuSOAP/0.9.5 (1.123)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "urn:WSDLiklan#Get_Ads"
Content-Length: 562
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:WSDLiklan"><SOAP-ENV:Body><tns:Get_Ads
xmlns:tns="urn:WSDLiklan"><id_channel
xsi:type="xsd:Int">12</id_channel><adstype
xsi:type="xsd:Int">10</adstype></tns:Get_Ads></SOAP-
ENV:Body></SOAP-ENV:Envelope>
```

SOAP Response :

```
HTTP/1.1 200 OK
Server: nginx admin
Date: Fri, 14 Oct 2011 19:28:33 GMT
Content-Type: text/xml; charset=ISO-8859-1
Connection: close
X-Powered-By: PHP/5.2.14
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Content-Length: 581
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:Get_AdsResponse
xmlns:ns1="urn:WSDLiklan"><output xsi:type="SOAP-ENC:Array"
SOAP-ENC:arrayType="xsd:string[3]"><item
xsi:type="xsd:string">Hosting Indonesia</item><item
xsi:type="xsd:string">Menyediakan hosting murah untuk server
indonesia</item><item
xsi:type="xsd:string">http://www.iixhosting.net</item></outp
```

```
ut></ns1:Get_AdsResponse></SOAP-ENV:Body></SOAP-
ENV:Envelope>
```

2. <http://whiteneews.blogspot.com>

SOAP Request :

```
POST /soapcode/soapserver.php HTTP/1.0
Host: xmladservice.info
User-Agent: NuSOAP/0.9.5 (1.123)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "urn:WSDLiklan#Get_Ads"
Content-Length: 561
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:WSDLiklan"><SOAP-ENV:Body><tns:Get_Ads
xmlns:tns="urn:WSDLiklan"><id_channel
xsi:type="xsd:Int">12</id_channel><adstype
xsi:type="xsd:Int">1</adstype></tns:Get_Ads></SOAP-
ENV:Body></SOAP-ENV:Envelope>
```

SOAP Response :

```
HTTP/1.1 200 OK
Server: nginx admin
Date: Fri, 14 Oct 2011 19:33:23 GMT
Content-Type: text/xml; charset=ISO-8859-1
Connection: close
X-Powered-By: PHP/5.2.14
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Content-Length: 655
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:Get_AdsResponse
xmlns:ns1="urn:WSDLiklan"><output xsi:type="SOAP-ENC:Array"
SOAP-ENC:arrayType="xsd:string[3]"><item
xsi:type="xsd:string">aaa</item><item
xsi:type="xsd:string"></item><item
xsi:type="xsd:string">http://mangafox.com</item></output></n
s1:Get_AdsResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Dari hasil pengujian yang dijelaskan diatas dapat disimpulkan proses SOAP *Request* dan SOAP *Response* dapat berjalan dengan baik seperti hasil yang diinginkan.

5.3 Kesimpulan Pengujian

Berdasarkan hasil pengujian aplikasi yang telah dilakukan diperoleh kesimpulan :

1. Seluruh menu pada aplikasi dapat berjalan dengan baik sesuai dengan fungsi masing-masing.
2. Proses pertukaran data dengan SOAP dapat bekerja dengan baik sehingga menghasilkan keluaran yang sesuai.

BAB VI

P E N U T U P

6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil penelitian tugas akhir ini adalah sebagai berikut :

1. Penerapan XML *Web Service* dengan SOAP (*Simple Object Access Protocol*) dalam situs periklanan *online* telah berhasil dirancang dan diimplementasikan dalam bentuk aplikasi berbasis web yang dapat memudahkan *advertiser* dalam mengoptimalkan penayangan iklannya.
2. *SOAP Message* berhasil mengirimkan pesan *SOAP Request* dan *SOAP Response* dalam melakukan proses pertukaran data antara aplikasi server dengan *website/blog publisher*.
3. Informasi yang diberikan oleh aplikasi *server* kepada *client* adalah berupa deskripsi iklan (isi dan judul) serta *url* tujuan dari iklan tersebut.

6.2 Saran

Beberapa hal yang disarankan untuk pengembangan sistem selanjutnya adalah sebagai berikut :

1. Menambahkan fitur dimana *advertiser* dapat memilih menayangkan iklannya hanya pada *visitor* yang berasal dari daerah tertentu.
2. Aplikasi *web service* ini dapat diimplementasikan menggunakan metode XML lainnya seperti XML-RPC dan REST yang masing-masing memiliki struktur dan spesifikasi yang berbeda.
3. Mengembangkan aplikasi pada *server* yang memungkinkan iklan ditayangkan sesuai dengan *keyword* atau tema yang berhubungan pada situs web *publisher*.

DAFTAR PUSTAKA

Azis, M. Farid. *Belajar Sendiri Pemrograman PHP 4 Bagi Web Programmer*. PT.Elex Media Komputindo, Jakarta, 2001.

Daniel, Roy. *Pengenalan Konsep XML Web Service*. Ilmukomputer.com. 2003

Junaedi, Moh.. *Pengantar XML*. Ilmukomputer.com. 2003

Lucky. *XML Web Service*. Jasakom. 2008

Muller, John Paul. *Special Edition Using SOAP*. Que, Indianapolis USA, 2002.

Short, Scott. *Building XML Web Services For The Microsoft .NET Platform*. PT.Elex Media Komputindo, Jakarta, 2003.

Simpson John E. *Just XML*. ANDI, Yogyakarta, 1999.

Turban, Efraim, dkk. *Decision Support Systems and Intelligent System*. ANDI, Yogyakarta, 2005.

“AdSense”. [Online] Available <http://id.wikipedia.org/wiki/adsense>, diakses 28 April 2010.

“SOAP”. [Online] Available <http://id.wikipedia.org/wiki/soap>, diakses 26 Maret 2010.

“WSDL”. [Online] Available <http://id.wikipedia.org/wiki/wsdl>, diakses 23 Maret 2010.

**PENERAPAN XML *WEB SERVICE* MENGGUNAKAN SOAP
(*SIMPLE OBJECT ACCESS PROTOCOL*) PADA SITUS WEB
PERIKLANAN**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh:

RISKI JUMADI
10551001502



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

**PENERAPAN XML *WEB SERVICE* MENGGUNAKAN SOAP
(*SIMPLE OBJECT ACCESS PROTOCOL*) PADA SITUS WEB
PERIKLANAN**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh:

RISKI JUMADI
10551001502



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

PENERAPAN XML *WEB SERVICE* MENGGUNAKAN SOAP (*SIMPLE OBJECT ACCESS PROTOCOL*) PADA SITUS WEB PERIKLANAN

RISKI JUMADI
10551001502

Tanggal Sidang : 31 Oktober 2011
Periode Wisuda : Februari 2012

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas No. 155 Pekanbaru

ABSTRAK

Memasang iklan di *internet* merupakan salah satu cara yang cukup efektif untuk mempromosikan sebuah produk maupun jasa. Banyak ditemui situs web penyedia iklan mulai dari yang gratis hingga berbayar. Model sistem yang digunakan kebanyakan adalah ketika pengiklan memasang sebuah iklan maka penayangannya akan ditampilkan pada situs web yang sama, sehingga iklan hanya akan dibaca oleh pengunjung dari situs web tersebut. Pada tugas akhir ini dirancang bangun sebuah aplikasi periklanan berbasis web dengan XML *Web Service* menggunakan SOAP (*Simple Object Access Protocol*) dimana iklan ditayangkan pada sebuah jaringan *web/blog* milik *publisher*, sehingga iklan akan ditampilkan lebih dari satu situs web.

Hasil implementasi dari rancang bangun aplikasi ini adalah iklan yang dipasang oleh *advertiser* berhasil ditayangkan pada jaringan *web/blog* milik *publisher* sesuai dengan kategori iklan yang ditentukan. Proses pertukaran data melalui SOAP *Message* berhasil dilakukan antara aplikasi dengan *web /blog publisher*.

Kata Kunci: *Advertiser*, Iklan, *Publisher*, SOAP, *Web Service*, XML

PENERAPAN XML *WEB SERVICE* MENGGUNAKAN SOAP (*SIMPLE OBJECT ACCESS PROTOCOL*) PADA SITUS WEB PERIKLANAN

RISKI JUMADI
10551001502

Tanggal Sidang : 31 Oktober 2011
Periode Wisuda : Februari 2012

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas No. 155 Pekanbaru

ABSTRAK

Memasang iklan di *internet* merupakan salah satu cara yang cukup efektif untuk mempromosikan sebuah produk maupun jasa. Banyak ditemui situs web penyedia iklan mulai dari yang gratis hingga berbayar. Model sistem yang digunakan kebanyakan adalah ketika pengiklan memasang sebuah iklan maka penayangannya akan ditampilkan pada situs web yang sama, sehingga iklan hanya akan dibaca oleh pengunjung dari situs web tersebut. Pada tugas akhir ini dirancang bangun sebuah aplikasi periklanan berbasis web dengan XML *Web Service* menggunakan SOAP (*Simple Object Access Protocol*) dimana iklan ditayangkan pada sebuah jaringan *web/blog* milik *publisher*, sehingga iklan akan ditampilkan lebih dari satu situs web.

Hasil implementasi dari rancang bangun aplikasi ini adalah iklan yang dipasang oleh *advertiser* berhasil ditayangkan pada jaringan *web/blog* milik *publisher* sesuai dengan kategori iklan yang ditentukan. Proses pertukaran data melalui SOAP *Message* berhasil dilakukan antara aplikasi dengan *web /blog publisher*.

Kata Kunci: *Advertiser*, Iklan, *Publisher*, SOAP, *Web Service*, XML

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xvii
DAFTAR SIMBOL	xviii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	I-1
1.2 Rumusan Masalah	I-2
1.3 Batasan Masalah	I-2
1.4 Tujuan Penelitian	I-2
1.5 Sistematika Penulisan	I-3
BAB II LANDASAN TEORI	
2.1 <i>Web Service</i>	II-1
2.2 <i>Arsitektur Web Service</i>	II-3
2.3 <i>SOAP (Simple Object Access Protocol)</i>	II-5
2.4 <i>WSDL (Web Services Description Language)</i>	II-8
2.5 <i>UDDI (Universal Description, Discovery, and Integration)</i> ..	II-9
2.6 <i>eXtensible Markup Language</i>	II-10
2.7 Iklan	II-12
2.7.1 Iklan Baris	II-12

2.7.2 Iklan <i>Banner</i>	II-13
2.8 <i>FlowChart</i>	II-13
2.9 PHP.....	II-14
2.10 NuSOAP.....	II-15
BAB III METODOLOGI PENELITIAN	
3.1 Tahapan Penelitian	III-1
3.1.1 Pengumpulan Data.....	III-2
3.1.2 Analisa Sistem.....	III-2
3.1.2.1 Analisa Sistem Lama.....	III-2
3.1.2.2 Analisa Sistem Baru	III-2
3.1.2.3 Deskripsi Umum Perangkat Lunak	III-2
3.1.2.4 Analisis Data Sistem.....	III-3
3.1.2.5 Analisis Fungsional Sistem	III-3
3.1.3 Perancangan.....	III-3
3.1.4 Implementasi	III-3
3.1.5 Pengujian	III-3
BAB IV ANALISA DAN PERANCANGAN	
4.1 Analisa Sistem	IV-1
4.1.1 Sistem yang Berjalan.....	IV-1
4.1.2 Sistem yang akan dibangun.....	IV-2
4.1.2.1 Analisa Pada <i>Advertiser</i>	IV-2
4.1.2.2 Analisa Pada <i>Publisher</i>	IV-3
4.1.2.3 Analisa Kebutuhan	IV-4
4.1.2.4 Analisa Data	IV-5
4.1.3 Identifikasi <i>Web Service</i> pada sistem	IV-6
4.1.4 <i>Flowchart</i> Sistem.....	IV-7
4.1.4.1 <i>Flowchart</i> Pada <i>Advertiser</i>	IV-7
4.1.4.2 <i>Flowchart</i> Pada <i>Publisher</i>	IV-8
4.1.5 Analisa Fungsional Sistem	IV-9
4.1.5.1 <i>Context Diagram</i>	IV-9
4.1.5.2 Data Flow Diagram (DFD).....	IV-10

4.1.5.4 DFD Level 1 <i>Web Service</i>	IV-10
4.1.5.5 DFD Level 2 Proses <i>Member</i>	IV-12
4.1.5.6 DFD Level 2 Proses Pengelolaan iklan dan <i>Channel</i>	IV-13
4.1.5.5 DFD Level 2 Proses Pengelolaan <i>Admin</i>	IV-14
4.1.6 <i>Entity Relationship Diagram (ER-Diagram)</i>	IV-15
4.2 Perancangan.....	IV-17
4.2.1 Perancangan Menu	IV-17
4.2.2 Perancangan Basis Data	IV-18
4.2.3 Perancangan Antar Muka (<i>Interface</i>)	IV-20
BAB V IMPLEMENTASI DAN PENGUJIAN	
5.1 Implementasi Sistem	V-1
5.1.1 Batasan Implementasi.....	V-1
5.1.2 Lingkungan Implementasi	V-1
5.1.3 Hasil Implementasi	V-2
5.1.3.1 Implementasi <i>SOAP Service</i> dan dokumen <i>WSDL (Web Service Definition Language)</i>	V-2
5.1.3.2 Tampilan Antarmuka (<i>Interface</i>).....	V-8
5.1.3.2.1 Tampilan Halaman Utama.....	V-8
5.1.3.2.2 Tampilan Halaman <i>Advertiser</i>	V-8
5.1.3.2.3 Tampilan Halaman <i>Publisher</i>	V-9
5.2 Pengujian Sistem	V-9
5.2.1 Pengujian Aplikasi <i>Server</i>	V-10
5.2.1.1 Identifikasi dan Pengujian Menu <i>Login</i>	V-10
5.2.1.2 Identifikasi dan Pengujian Menu <i>Profil</i>	V-10
5.2.2 Pengujian <i>Client / Channel</i>	V-11
5.2.2.1 Skenario Pengujian	V-11
5.2.2.2 Hasil Pengujian.....	V-12
5.3 Kesimpulan Pengujian.....	V-15

BAB VI PENUTUP

6.1 Kesimpulan..... VI-1

6.2 Saran VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xvii
DAFTAR SIMBOL	xviii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	I-1
1.2 Rumusan Masalah	I-2
1.3 Batasan Masalah	I-2
1.4 Tujuan Penelitian	I-2
1.5 Sistematika Penulisan	I-3
BAB II LANDASAN TEORI	
2.1 <i>Web Service</i>	II-1
2.2 <i>Arsitektur Web Service</i>	II-3
2.3 <i>SOAP (Simple Object Access Protocol)</i>	II-5
2.4 <i>WSDL (Web Services Description Language)</i>	II-8
2.5 <i>UDDI (Universal Description, Discovery, and Integration)</i> ..	II-9
2.6 <i>eXtensible Markup Language</i>	II-10
2.7 Iklan	II-12
2.7.1 Iklan Baris	II-12

2.7.2 Iklan <i>Banner</i>	II-13
2.8 <i>FlowChart</i>	II-13
2.9 PHP.....	II-14
2.10 NuSOAP.....	II-15
BAB III METODOLOGI PENELITIAN	
3.1 Tahapan Penelitian	III-1
3.1.1 Pengumpulan Data.....	III-2
3.1.2 Analisa Sistem.....	III-2
3.1.2.1 Analisa Sistem Lama.....	III-2
3.1.2.2 Analisa Sistem Baru	III-2
3.1.2.3 Deskripsi Umum Perangkat Lunak	III-2
3.1.2.4 Analisis Data Sistem.....	III-3
3.1.2.5 Analisis Fungsional Sistem	III-3
3.1.3 Perancangan.....	III-3
3.1.4 Implementasi	III-3
3.1.5 Pengujian	III-3
BAB IV ANALISA DAN PERANCANGAN	
4.1 Analisa Sistem	IV-1
4.1.1 Sistem yang Berjalan.....	IV-1
4.1.2 Sistem yang akan dibangun.....	IV-2
4.1.2.1 Analisa Pada <i>Advertiser</i>	IV-2
4.1.2.2 Analisa Pada <i>Publisher</i>	IV-3
4.1.2.3 Analisa Kebutuhan	IV-4
4.1.2.4 Analisa Data	IV-5
4.1.3 Identifikasi <i>Web Service</i> pada sistem	IV-6
4.1.4 <i>Flowchart</i> Sistem.....	IV-7
4.1.4.1 <i>Flowchart</i> Pada <i>Advertiser</i>	IV-7
4.1.4.2 <i>Flowchart</i> Pada <i>Publisher</i>	IV-8
4.1.5 Analisa Fungsional Sistem	IV-9
4.1.5.1 <i>Context Diagram</i>	IV-9
4.1.5.2 Data Flow Diagram (DFD).....	IV-10

4.1.5.4 DFD Level 1 <i>Web Service</i>	IV-10
4.1.5.5 DFD Level 2 Proses <i>Member</i>	IV-12
4.1.5.6 DFD Level 2 Proses Pengelolaan iklan dan <i>Channel</i>	IV-13
4.1.5.5 DFD Level 2 Proses Pengelolaan <i>Admin</i>	IV-14
4.1.6 <i>Entity Relationship Diagram (ER-Diagram)</i>	IV-15
4.2 Perancangan.....	IV-17
4.2.1 Perancangan Menu	IV-17
4.2.2 Perancangan Basis Data	IV-18
4.2.3 Perancangan Antar Muka (<i>Interface</i>)	IV-20
 BAB V IMPLEMENTASI DAN PENGUJIAN	
5.1 Implementasi Sistem	V-1
5.1.1 Batasan Implementasi.....	V-1
5.1.2 Lingkungan Implementasi	V-1
5.1.3 Hasil Implementasi	V-2
5.1.3.1 Implementasi <i>SOAP Service</i> dan dokumen <i>WSDL (Web Service Definition Language)</i>	V-2
5.1.3.2 Tampilan Antarmuka (<i>Interface</i>).....	V-8
5.1.3.2.1 Tampilan Halaman Utama.....	V-8
5.1.3.2.2 Tampilan Halaman <i>Advertiser</i>	V-8
5.1.3.2.3 Tampilan Halaman <i>Publisher</i>	V-9
5.2 Pengujian Sistem	V-9
5.2.1 Pengujian Aplikasi <i>Server</i>	V-10
5.2.1.1 Identifikasi dan Pengujian Menu <i>Login</i>	V-10
5.2.1.2 Identifikasi dan Pengujian Menu <i>Profil</i>	V-10
5.2.2 Pengujian <i>Client / Channel</i>	V-11
5.2.2.1 Skenario Pengujian	V-11
5.2.2.2 Hasil Pengujian.....	V-12
5.3 Kesimpulan Pengujian.....	V-15

BAB VI PENUTUP

6.1 Kesimpulan..... VI-1

6.2 Saran VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR TABEL

Tabel	Halaman
2.1 <i>Tag-tag</i> pada dokumen WSDL	II-9
4.1 Jenis dan Ukuran iklan	IV-5
4.2 Identifikasi <i>Web Service</i>	IV-6
4.3 DFD Level 1 <i>Web service</i>	IV-11
4.4 Keterangan Aliran Data DFD Level 1 <i>Web service</i>	IV-11
4.5 DFD Level 2 Proses Registrasi	IV-12
4.6 Keterangan Aliran Data DFD Level 2 Proses Registrasi	IV-12
4.7 DFD Level 2 Proses Pengelolaan iklan & <i>channel</i>	IV-13
4.8 Keterangan Aliran Data DFD Level 2 Proses iklan & <i>channel</i>	IV-13
4.9 DFD Level 2 Proses Pengelolaan <i>Admin</i>	IV-14
4.10 Keterangan Aliran Data DFD Level 2 Proses <i>Admin</i>	IV-14
4.11 Keterangan entitas pada ERD	IV-15
4.12 Perancangan Tabel Member.....	IV-18
4.13 Perancangan Tabel <i>Channel</i>	IV-18
4.14 Perancangan Tabel Iklan	IV-19
4.15 Perancangan Tabel Detail_klik	IV-19
4.16 Perancangan Tabel Kategori_iklan	IV-19
5.1 Identifikasi <i>service</i>	V-3
5.2 Hasil Pengujian Menu Login	V-10
5.3 Hasil Pengujian Menu Profil.....	V-10
5.4 Daftar <i>Channel</i>	V-11

DAFTAR TABEL

Tabel	Halaman
2.1 <i>Tag-tag</i> pada dokumen WSDL	II-9
4.1 Jenis dan Ukuran iklan	IV-5
4.2 Identifikasi <i>Web Service</i>	IV-6
4.3 DFD Level 1 <i>Web service</i>	IV-11
4.4 Keterangan Aliran Data DFD Level 1 <i>Web service</i>	IV-11
4.5 DFD Level 2 Proses Registrasi	IV-12
4.6 Keterangan Aliran Data DFD Level 2 Proses Registrasi	IV-12
4.7 DFD Level 2 Proses Pengelolaan iklan & <i>channel</i>	IV-13
4.8 Keterangan Aliran Data DFD Level 2 Proses iklan & <i>channel</i>	IV-13
4.9 DFD Level 2 Proses Pengelolaan <i>Admin</i>	IV-14
4.10 Keterangan Aliran Data DFD Level 2 Proses <i>Admin</i>	IV-14
4.11 Keterangan entitas pada ERD	IV-15
4.12 Perancangan Tabel Member.....	IV-18
4.13 Perancangan Tabel <i>Channel</i>	IV-18
4.14 Perancangan Tabel Iklan	IV-19
4.15 Perancangan Tabel Detail_klik	IV-19
4.16 Perancangan Tabel Kategori_iklan	IV-19
5.1 Identifikasi <i>service</i>	V-3
5.2 Hasil Pengujian Menu Login	V-10
5.3 Hasil Pengujian Menu Profil.....	V-10
5.4 Daftar <i>Channel</i>	V-11

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi informasi yang semakin berkembang dan didukung oleh struktur dan infrastruktur yang semakin baik membuat kebutuhan akan informasi dengan mudah didapatkan terutama dengan adanya *internet*. Hal tersebut memberikan peluang kepada pelaku bisnis untuk mulai melirik *internet* sebagai media yang sangat potensial untuk mendapatkan konsumen atas produk atau jasa yang ditawarkan, sehingga diharapkan mampu untuk meningkatkan penjualan. Berbagai cara dilakukan untuk meningkatkan jumlah pengunjung yang datang ke situs web mereka. Sebab, secara tidak langsung semakin banyak pengunjung yang datang, maka semakin besar peluang untuk mendapatkan lebih banyak konsumen.

Salah satu cara yang dapat dilakukan untuk mendapatkan banyak pengunjung adalah dengan memasang iklan di situs web penyedia jasa periklanan yang banyak ditemui di *internet*, mulai dari yang gratis hingga berbayar. Salah satu contoh adalah www.iklanbaris.co.id, dimana pengiklan (*advertiser*) melakukan pendaftaran pada situs tersebut lalu memasang iklannya, kemudian iklan akan ditayangkan di halaman web www.iklanbaris.co.id. Namun, beriklan dengan cara tersebut masih kurang efektif, sebab iklan masih harus bersaing untuk tampil dengan iklan yang dipasang oleh pengiklan lain. Sehingga, semakin kecil kemungkinan iklan akan dibaca.

Berdasarkan hal tersebut maka digunakan teknologi *web service* yang dapat membantu memberikan solusi terhadap permasalahan yang terjadi. Peranan *web service* diperlukan sebagai media untuk *sharing* data maupun fungsionalitas antar proses bisnis agar suatu aplikasi dapat berinteraksi dengan aplikasi lain tanpa tergantung dengan bahasa pemrograman maupun sistem operasi.

Jika sebelumnya iklan hanya dipasang dan ditayangkan pada satu situs web namun dengan menggunakan *web service* akan mengubah konsep tersebut, pemasang iklan cukup memasang iklannya pada satu situs (penyedia *web service*

periklanan), kemudian mesin *web service* yang akan bekerja untuk menayangkan iklan di halaman *web/blog* orang lain (*publisher*) yang mau menyisakan sedikit ruangan di halaman *web/blog* mereka, artinya terjadi interaksi antara dua aplikasi yang berbeda dimana *web/blog publisher* akan mengirimkan *request* ke *web service* dan kemudian *web service* akan merespon dengan mengirimkan data iklan yang dipasang oleh pengiklan. Disinilah peranan *web service* dalam melakukan proses pertukaran data. Dalam penelitian ini teknologi yang akan digunakan adalah SOAP (*Simple Object Acces Protocol*) karena SOAP merupakan protokol yang sudah distandarisasi oleh W3C (*World Wide Web Consortium*) dan didukung luas dari berbagai *vendor*.

Berdasarkan permasalahan diatas, dalam penelitian ini akan dirancang bangun sebuah situs web periklanan dengan memanfaatkan teknologi *web service* menggunakan SOAP.

1.2. Rumusan Masalah

Dari latar belakang yang telah dijelaskan sebelumnya dapat diambil suatu perumusan masalah yaitu bagaimana menerapkan teknologi *web service* menggunakan SOAP pada situs web periklanan.

1.3. Batasan Penelitian

Agar penulisan tugas akhir ini lebih terarah maka penulis membuat batasan permasalahan, yaitu :

1. Situs web hanya menampilkan iklan teks dan gambar.
2. Tidak membahas mengenai sistem pembayaran.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai adalah rancang bangun situs web periklanan dengan XML *Web Service* menggunakan SOAP dalam memfasilitasi penayangan iklan dari *advertiser* pada jaringan *web/blog publisher*.

1.5. Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan tugas akhir yang akan dibuat :

Bab I Pendahuluan

Bagian ini berisi tentang deskripsi umum tugas akhir yang meliputi latar belakang masalah, perumusan masalah, tujuan penelitian, batasan penelitian, serta sistematika penulisan.

Bab II Landasan Teori

Bagian ini menjelaskan tentang teori-teori umum, teori-teori khusus yang berhubungan dengan tugas akhir ini.

Bab III Metodologi Penelitian

Bagian ini menjelaskan tentang metode pengembangan modul aplikasi, pengumpulan data dan objek, tempat serta waktu penelitian.

Bab IV Analisa Dan Perancangan

Bagian ini berisi tentang analisis dan perancangan aplikasi. Pada tahap perancangan akan dibahas tentang perancangan aplikasi.

Bab V Implementasi Dan Pengujian

Pada bagian implementasi dan pengujian berisi pembahasan mengenai implementasi modul disertai dengan *preview* tampilan *interface* serta pengujian aplikasi.

Bab VI Kesimpulan Dan Saran

Bagian ini berisi kesimpulan hasil penelitian beserta saran-saran yang berkaitan dengan penelitian ini.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi informasi yang semakin berkembang dan didukung oleh struktur dan infrastruktur yang semakin baik membuat kebutuhan akan informasi dengan mudah didapatkan terutama dengan adanya *internet*. Hal tersebut memberikan peluang kepada pelaku bisnis untuk mulai melirik *internet* sebagai media yang sangat potensial untuk mendapatkan konsumen atas produk atau jasa yang ditawarkan, sehingga diharapkan mampu untuk meningkatkan penjualan. Berbagai cara dilakukan untuk meningkatkan jumlah pengunjung yang datang ke situs web mereka. Sebab, secara tidak langsung semakin banyak pengunjung yang datang, maka semakin besar peluang untuk mendapatkan lebih banyak konsumen.

Salah satu cara yang dapat dilakukan untuk mendapatkan banyak pengunjung adalah dengan memasang iklan di situs web penyedia jasa periklanan yang banyak ditemui di *internet*, mulai dari yang gratis hingga berbayar. Salah satu contoh adalah www.iklanbaris.co.id, dimana pengiklan (*advertiser*) melakukan pendaftaran pada situs tersebut lalu memasang iklannya, kemudian iklan akan ditayangkan di halaman web www.iklanbaris.co.id. Namun, beriklan dengan cara tersebut masih kurang efektif, sebab iklan masih harus bersaing untuk tampil dengan iklan yang dipasang oleh pengiklan lain. Sehingga, semakin kecil kemungkinan iklan akan dibaca.

Berdasarkan hal tersebut maka digunakan teknologi *web service* yang dapat membantu memberikan solusi terhadap permasalahan yang terjadi. Peranan *web service* diperlukan sebagai media untuk *sharing* data maupun fungsionalitas antar proses bisnis agar suatu aplikasi dapat berinteraksi dengan aplikasi lain tanpa tergantung dengan bahasa pemrograman maupun sistem operasi.

Jika sebelumnya iklan hanya dipasang dan ditayangkan pada satu situs web namun dengan menggunakan *web service* akan mengubah konsep tersebut, pemasang iklan cukup memasang iklannya pada satu situs (penyedia *web service*

periklanan), kemudian mesin *web service* yang akan bekerja untuk menayangkan iklan di halaman *web/blog* orang lain (*publisher*) yang mau menyisakan sedikit ruangan di halaman *web/blog* mereka, artinya terjadi interaksi antara dua aplikasi yang berbeda dimana *web/blog publisher* akan mengirimkan *request* ke *web service* dan kemudian *web service* akan merespon dengan mengirimkan data iklan yang dipasang oleh pengiklan. Disinilah peranan *web service* dalam melakukan proses pertukaran data. Dalam penelitian ini teknologi yang akan digunakan adalah SOAP (*Simple Object Acces Protocol*) karena SOAP merupakan protokol yang sudah distandarisasi oleh W3C (*World Wide Web Consortium*) dan didukung luas dari berbagai *vendor*.

Berdasarkan permasalahan diatas, dalam penelitian ini akan dirancang bangun sebuah situs web periklanan dengan memanfaatkan teknologi *web service* menggunakan SOAP.

1.2. Rumusan Masalah

Dari latar belakang yang telah dijelaskan sebelumnya dapat diambil suatu perumusan masalah yaitu bagaimana menerapkan teknologi *web service* menggunakan SOAP pada situs web periklanan.

1.3. Batasan Penelitian

Agar penulisan tugas akhir ini lebih terarah maka penulis membuat batasan permasalahan, yaitu :

1. Situs web hanya menampilkan iklan teks dan gambar.
2. Tidak membahas mengenai sistem pembayaran.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai adalah rancang bangun situs web periklanan dengan XML *Web Service* menggunakan SOAP dalam memfasilitasi penayangan iklan dari *advertiser* pada jaringan *web/blog publisher*.

1.5. Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan tugas akhir yang akan dibuat :

Bab I Pendahuluan

Bagian ini berisi tentang deskripsi umum tugas akhir yang meliputi latar belakang masalah, perumusan masalah, tujuan penelitian, batasan penelitian, serta sistematika penulisan.

Bab II Landasan Teori

Bagian ini menjelaskan tentang teori-teori umum, teori-teori khusus yang berhubungan dengan tugas akhir ini.

Bab III Metodologi Penelitian

Bagian ini menjelaskan tentang metode pengembangan modul aplikasi, pengumpulan data dan objek, tempat serta waktu penelitian.

Bab IV Analisa Dan Perancangan

Bagian ini berisi tentang analisis dan perancangan aplikasi. Pada tahap perancangan akan dibahas tentang perancangan aplikasi.

Bab V Implementasi Dan Pengujian

Pada bagian implementasi dan pengujian berisi pembahasan mengenai implementasi modul disertai dengan *preview* tampilan *interface* serta pengujian aplikasi.

Bab VI Kesimpulan Dan Saran

Bagian ini berisi kesimpulan hasil penelitian beserta saran-saran yang berkaitan dengan penelitian ini.

BAB II

LANDASAN TEORI

2.1. *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML (*eXtensible Markup Language*), sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *web service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di *upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di *port* 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

Web Service dapat diimplementasikan pada lingkungan internal (*Intranet*) untuk kebutuhan integrasi antar sistem aplikasi (*EAI=Enterprise Application Integration*) ataupun pada lingkungan eksternal (*Internet*) untuk mendukung aplikasi *business-to-business (e-business)*.

Keuntungan penggunaan *Web Service* : (Wikipedia.org, 2010)

1. Format penggunaan terbuka untuk semua *platform*.
2. Mudah di mengerti dan mudah *men-debug*.
3. Menggunakan standar-standar “membuka *service* sekali” dan mempunyai pemakai banyak.
4. Mudah untuk menengahi pesan-pesan proses dan menambahkan nilai.
5. *Routing* and pengiriman.
6. *Security*.
7. *Management* and *monitoring*.
8. *Schema* and *service design*.
9. Akselerasi.
10. Mudah untuk mengembangkan dengan *semantic transport* tambahan.
11. Terbuka, standar-standar berbasis teks.
12. Pencapaian modular.
13. Tidak mahal untuk diimplementasikan (relatif).
14. Mengurangi biaya integrasi aplikasi *enterprise*.
15. Implementasi yang *incremental*.

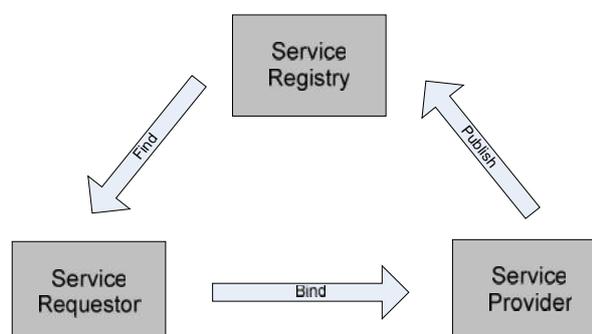
Web service ternyata sangat berbeda dengan *website*. Perbedaan yang paling terlihat adalah *website* dibuat untuk memiliki tampilan atau *user interface* yang bagus sedangkan *web service* tidak memiliki tampilan. Itu karena *web service* tidak dibuat untuk berinteraksi langsung dengan *user*. Sesuai dengan kata *service* yang ada pada namanya, *web service* hanya menyediakan *service* atau layanan. Layanan tersebutlah yang kemudian akan digunakan atau dipanggil oleh aplikasi lainnya. Dengan demikian yang akan menjadi *interface* adalah aplikasi yang memanggilnya bukan *web service* itu sendiri.

2.2. Arsitektur Web Service

Konsep arsitektur yang mendasari teknologi *Web service* adalah *Service Oriented Architecture* (SOA). Dalam arsitektur ini, suatu aplikasi dimodelkan sebagai komposisi dari sekumpulan *service* yang disediakan oleh suatu komponen. Lokasi keberadaan komponen tersebut dapat ditemukan oleh *client* secara dinamis, dalam arti tidak dinyatakan secara statis tetapi menggunakan mekanisme *discovery* untuk mencari keberadaan komponen tersebut. Demikian pula, *client* dapat meminta (*invoke*) *service* tersebut secara dinamis pula. Dalam arsitektur ini, SOA mendefinisikan 3 peran berbeda yang menunjukkan peran dari masing-masing komponen dalam sistem, yaitu :

1. *Service provider*, yaitu suatu entitas yang menyediakan *interface* terhadap sistem yang menjalankan suatu sekumpulan tugas tertentu. *Service provider* dapat merepresentasikan suatu entitas bisnis ataupun suatu komponen *software* yang *reusable*.
2. *Service requestor*, yaitu suatu entitas yang meminta/memperoleh (dan menemukan) *software service* dalam rangka menyelesaikan suatu tugas tertentu atau menyediakan solusi bisnis tertentu.
3. *Service registry*, yaitu entitas yang bertindak sebagai penyimpan (*repository*) suatu *software service* yang dipublikasikan oleh *Service provider*.

Peran dan interaksi antar komponen tersebut ditunjukkan pada gambar dibawah ini :



Gambar 2.1 - Konsep *Service Oriented Architecture*

Secara umum, *web service* memiliki tiga operasi yang terlibat di dalamnya, yaitu:

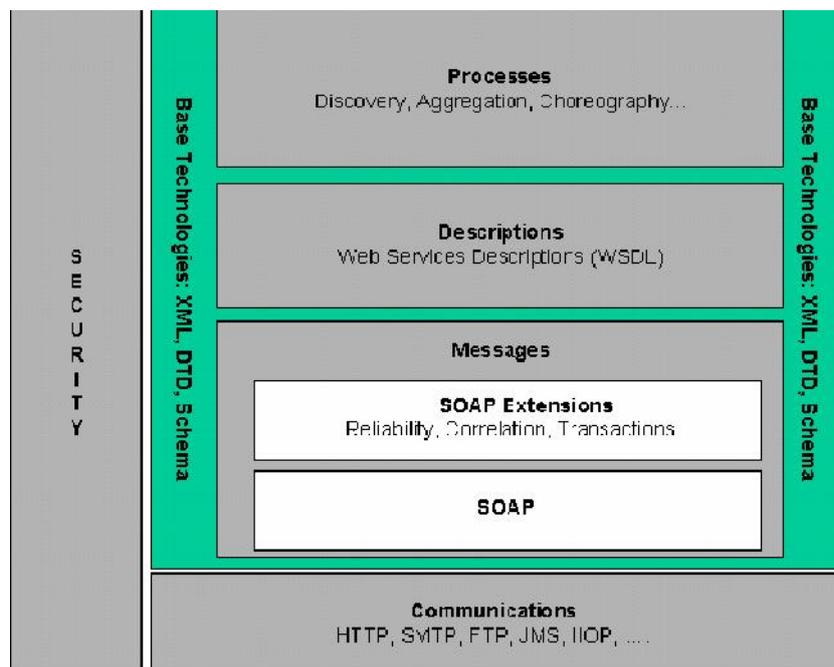
1. *Publish/Unpublish*: Menerbitkan/menghapus layanan ke dalam atau dari *registry*.
2. *Find*: *Service requestor* mencari dan menemukan layanan yang dibutuhkan.
3. *Bind*: *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke *service provider* untuk melakukan interaksi dan mengakses layanan/*service* yang disediakan oleh *service provider*.

Dalam arsitektur ini, *service* dapat didefinisikan sebagai komponen (*software component*) yang memiliki karakteristik :

1. Dapat dideskripsikan dalam suatu bahasa formal
2. Dapat dipublikasikan pada suatu *registry of service*
3. Dapat ditemukan (*discover*) menggunakan mekanisme standar
4. Dapat diminta/diperoleh (*invoke*) melalui jaringan
5. Dapat dibangun bersama *service* lain

Model Arsitektur *Web Service* menggunakan konsep SOA tersebut ,yaitu :

1. *Web Service* menggunakan standar terbuka, yaitu : XML (*eXtensible Markup Language*) sebagai bahasa untuk mendeskripsikan data, SOAP (*Simple Object Access Protocol*) sebagai protokol komunikasi antar komponen , WSDL (*Web Service Description Language*) untuk mendeskripsikan *service*, UDDI (*Universal Description, Discovery and Integration*) untuk *service discovery*, dan HTTP sebagai *transport layer*.
2. Standar *Web Service* merupakan *message-based*, bukan *API-based*. Dalam hal ini, komunikasi antara *Web Service*, *Service Requester* dan *Service Registry* menggunakan pesan SOAP berbasis XML. W3C mengembangkan *draft* arsitektur *Web Service* yang terdiri dari beberapa teknologi yang saling berhubungan dan berlapis seperti digambarkan sebagai berikut :



Gambar 2.2 Arsitektur Web Service

2.3. SOAP (*Simple Object Access Protocol*)

SOAP merupakan protokol komunikasi yang ringan berbasis *eXtensible Markup Language* (XML). Pesan SOAP berbentuk sekumpulan *XML Schema* yang mendefinisikan format untuk mentransmisikan pesan XML melalui jaringan, termasuk tipe data dan cara menstrukturkan pesan secara tepat sehingga dapat mudah dipahami oleh *server* atau *end-point* lainnya.

Pesan SOAP terdiri dari 3 bagian, yaitu :

1. *Envelope*, suatu kerangka yang mendefinisikan apa yang ada dalam pesan dan bagaimana pesan harus diproses serta menunjukkan resipien dari pesan tersebut.
2. *Header*, berisi informasi yang berhubungan dengan keamanan dan *routing*. Keberadaan *Header* dalam SOAP bersifat *optional*.
3. *Body*, berisi data yang berhubungan dengan aplikasi tertentu yang sedang dipertukarkan. Data di *mark-up* sebagai XML dan dimasukkan dalam format yang spesifik yang didefinisikan dalam *XML Schema*.

SOAP (*Simple Object Access Protocol*) adalah sebuah *XML based markup language* untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data *object* didalam *network*. SOAP mendefinisikan empat aspek didalam komunikasi: *Message envelope*, *Encoding*, *RPC call convention*, dan bagaimana menyatukan sebuah *message* didalam protokol *transport*.

Sebuah SOAP *message* terdiri dari SOAP *Envelope* dan bisa terdiri dari *attachments* atau tidak memiliki *attachment*. SOAP *envelope* tersusun dari SOAP *header* dan SOAP *body*, sedangkan SOAP *attachment* membolehkan non-XML data untuk dimasukkan kedalam SOAP *message*, di-*encoded*, dan diletakkan kedalam SOAP *message* dengan menggunakan *MIME-multipart*.

SOAP menspesifikan secara jelas bagaimana cara untuk meng-*encode header* HTTP dan *file* XML sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan.

SOAP adalah protokol ringan yang ditujukan untuk pertukaran informasi struktur pada lingkup desentralisasi, dan terdistribusi. SOAP menggunakan teknologi XML untuk mendefinisikan rangka kerja pemesanan terekstrensi di mana menyediakan konstruksi pesan yang dapat dipertukarkan pada protokol berbeda. Rangka kerja dirancang bebas dari model pemrograman dan spesifikasi implementasi semantik.

SOAP tidak berpasangan erat dengan protokol *transport* tertentu. Spesifikasi SOAP menerangkan bagaimana pesan-pesan SOAP seharusnya diikat ke HTTP. Namun, sebuah pesan SOAP tidak lebih daripada sebuah dokumen XML. Jadi, ia dapat diangkut melalui protokol apa pun yang mampu mengirimkan teks. (Scott Short, 2003).

Peran SOAP di dalam teknologi *web service* adalah sebagai protokol pemaketan untuk pesan-pesan (*messages*) yang digunakan secara bersama oleh aplikasi-aplikasi penggunanya. Spesifikasi yang digunakan tidak lebih seperti sebuah amplop biasa berbasis XML untuk informasi yang ditransfer, serta sekumpulan aturan bagi translasi aplikasi dan tipe-tipe data platform yang spesifik

menjadi bentuk XML. Desain bentuk dari SOAP membuatnya cocok untuk berbagai pertukaran pesan pada aplikasi. Berikut adalah salah satu contoh bentuk sederhana pesan SOAP :

Contoh 1 : SOAP Message Embedded in HTTP Request

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:
encodingStyle="http://schemas.xmlsoap.org/soap/encoding
/">
<SOAP-ENV:Body>
<m:GetLastTradePrice xmlns:m="Some-URI">
<symbol>DIS</symbol>
</m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Contoh 2 : SOAP Message Embedded in HTTP Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/enco
ding/">
<SOAP-ENV:Body>
<m:GetLastTradePriceResponse xmlns:m="Some-URI">
<Price>34.5</Price>
</m:GetLastTradePriceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.4. WSDL (*Web Services Description Language*)

WSDL (*Web Services Description Language*) adalah dialek berbasis XML yang lapisannya ada diatas skema yang menjelaskan layanan web. Dokumen WSDL menyediakan informasi yang dibutuhkan *client* untuk berinteraksi dengan layanan web. WSDL dapat diperluas dan dipakai untuk menjelaskan hampir semua layanan jaringan, termasuk SOAP lewat HTTP. Bahkan protokol yang bukan berbasis XML, seperti DCOM lewat UDP. (Scott Short,2003)

WSDL (*Web Services Description Language*) adalah format XML yang diterbitkan untuk menerangkan *web service*. WSDL mendefinisikan: (Wikipedia.org, 2010)

- Pesan-pesan (baik yang abstrak dan kongkrit) yang dikirim ke dan menuju *web service*
- Koleksi-koleksi digital dari pesan-pesan (*port type*, antarmuka)
- Bagaimana *port type* yang ditentukan dijadikan *wire* protokol
- Dimana *service* ditempatkan

WSDL digunakan di mana skema XML tidak digunakan lagi dengan menyediakan jalur pesan-pesan grup menjadi operasi-operasi dan operasi-operasi menjadi antarmuka. Ini juga menyediakan jalur untuk medefinisikan *binding-binding* untuk setiap antamuka dan kombinasi protokol sepanjang alamat titik akhir untuk setiap kalinya. Definisi WSDL yang lengkap terdiri dari seluruh informasi yang dibutuhkan untuk meminta *web service*. Pengembang yang mau mempermudah yang lain untuk mengakses *service-service* nya harus menyediakan definisi-definisi WSDL.

Sebuah dokumen WSDL dapat dibagi menjadi 6 bagian besar. Masing-masing bagian tersebut ditandai tag-tag tertentu. (Lucky, 2008)

Tabel 2.1 *Tag-tag* pada dokumen WSDL

Tag	Keterangan
<i>Definitions</i>	Tag ini adalah tag tertinggi dalam dokumen WSDL atau biasa disebut dengan tag <i>root</i> . Tag-tag lainnya harus berada didalam tag ini. Salah satu informasi yang ada didalam tag ini adalah <i>namespace</i> .
<i>Types</i>	Didalam tag ini didefinisikan tipe-tipe data yang digunakan oleh <i>web service</i> . Akan tetapi yang didefinisikan hanyalah tipe-tipe data kompleks seperti <i>array</i> atau <i>struct</i> . Sedangkan tipe-tipe data sederhana yang sudah <i>built-in</i> seperti <i>integer</i> , <i>double</i> , <i>string</i> , dan lain-lain tidak didefinisikan lagi.
<i>Messages</i>	Informasi yang ada didalam tag ini adalah <i>message</i> atau parameter <i>input</i> atau <i>output</i> apa saja yang akan digunakan
<i>PortType</i>	Didalam tag ini didefinisikan <i>method</i> atau fungsi yang disediakan oleh <i>web service</i>
<i>Binding</i>	Informasi yang ada didalam tag ini bagaimana sebuah operasi atau fungsi diimplementasikan. Apakah fungsi tersebut dapat dipanggil menggunakan protokol HTTP atau SMTP. Informasi seperti <i>soap action</i> juga didefinisikan didalam tag ini
<i>Service</i>	Didalam tag ini didefinisikan lokasi dari <i>web service</i>

2.5. UDDI (*Universal Description, Discovery, and Integration*)

UDDI adalah spesifikasi yang telah diakui secara luas untuk pendaftaran suatu *web service*. UDDI mendefinisikan metadata dan protokol yang digunakan untuk mengetahui dan mengubah informasi dari suatu *web service*. Langkah pertama dalam menemukan *web service* adalah dengan meminta alamat tempat penyimpanan dari *web service* yang akan dipakai yang biasa disebut *directory*.

Setelah menemukan *directory*, peminta *web service* dapat mengirimkan permintaan lagi untuk mendapatkan informasi detail tentang *web service* (misalnya penyedia *web service* dan dimana diletakkan). Selanjutnya perangkat lunak menggunakan informasi yang didapat untuk secara dinamik mengakses *web service* yang diinginkan.

2.6. *eXtensible Markup Language*

XML kependekan dari *eXtensible Markup Language*, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan *markup language* yang tidak kalah hebatnya dengan SGML.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup (diawali dengan '</' diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal `<form name="isidata">`). Hanya bedanya, HTML mendefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak.

XML adalah *meta-language*. *Meta-language* adalah bahasa yang dapat digunakan untuk mendefinisikan bahasa-bahasa lainnya. Contohnya adalah menggunakan XML untuk mendefinisikan bahasa seperti WML.

Dengan XML dapat dilakukan:

1. Mendefinisikan struktur data
2. Membuat struktur tersebut *platform independent*
3. Memproses data terdefinisi oleh XML secara otomatis
4. Mendefinisikan *tag* sendiri

Beberapa alasan mengapa menggunakan XML :

1. Pertukaran data.
XML digunakan dalam pertukaran data. XML memungkinkan untuk mendefinisikan data dengan jelas. Pengirim dan penerima data akan menggunakan XML untuk memahami data yang telah dikirimkan. Dengan menggunakan XML, pengirim dan penerima memiliki interpretasi sama tentang data yang digunakan.
2. Pengganti *Electronic Data Interchange* (EDI). EDI telah digunakan dalam pertukaran data. EDI mahal, karena menggunakan *dedicated communication infrastructure*. Definisi yang digunakan jauh dari fleksibel. XML adalah pengganti yang baik terhadap EDI. XML menggunakan *Internet* dalam pertukaran data dan fleksibel.
3. Kemungkinan lainnya. XML dapat digunakan untuk mendefinisikan bahasa lainnya. Salah satu contohnya adalah WML (Wireless Markup Language), bahasa yang digunakan dalam komunikasi WAP. Dengan kata lain, WML adalah salah satu dialek XML.

Pada dasarnya, kemampuan XML memiliki potensi untuk dapat digunakan pada kondisi dimana data dimasukkan/dikeluarkan, disimpan, atau ditransmisikan dari suatu tempat ke tempat lain. XML dapat digunakan sebagai pengganti (atau pelengkap) database tradisional, atau mentransfer informasi antar bisnis. Organisasi berita, termasuk individu, juga dapat menggunakan XML untuk mendistribusikan *syndicated news stories*.

Bagian-Bagian dari Dokumen XML

Sebuah dokumen XML terdiri dari bagian bagian yang disebut dengan *node*. *Node-node* itu adalah:

- *Root node* yaitu *node* yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu *root node*. *Node-node* yang lainnya berada di dalam *root node*.

- *Element node* yaitu bagian dari dokumen XML yang ditandai dengan *tag* pembuka dan *tag* penutup, atau bisa juga sebuah *tag* tunggal elemen kosong seperti `<anggota nama="budi" />`. *Root node* biasa juga disebut *root element*
- *Attribute note* termasuk nama dan nilai atribut ditulis pada *tag* awal sebuah elemen atau pada *tag* tunggal.
- *Text node*, adalah *text* yang merupakan isi dari sebuah elemen, ditulis diantara *tag* pembuka dan *tag* penutup
- *Comment node* adalah baris yang tidak dieksekusi oleh *parser*
- *Processing Instruction node*, adalah perintah pengolahan dalam dokumen XML. Node ini ditandai awali dengan karakter `<?` Dan diakhiri dengan `?>`. Tapi perlu diingat bahwa header standard XML `<?xml version="1.0" encoding="iso-8859-1"?>` bukanlah *processing instruction node*. *Header standard* bukanlah bagian dari hirarki pohon dokumen XML.
- *NameSpace Node*, node ini mewakili deklarasi *namespace*

2.7. Iklan

Iklan ialah promosi barang, jasa, perusahaan dan ide yang harus dibayar oleh sebuah sponsor. Pemasaran melihat iklan sebagai bagian dari strategi promosi secara keseluruhan. Komponen lainnya dari promosi termasuk publisitas, relasi publik, penjualan, dan promosi penjualan.

2.7.1. Iklan Baris

Iklan baris (bahasa Inggris: *Classified advertising*) adalah salah satu cara promosi barang dan jasa yang umumnya ditemukan di koran. Cara ini merupakan pengembangan dari promosi iklan yang mengutamakan daya tarik dengan gambar dan dengan informasi yang lebih lengkap dan terinci. (Wikipedia.org, 2010)

Iklan baris mengutamakan informasi yang paling inti yang perlu diketahui oleh peminatnya. Karena itu biasanya iklan baris hanya memuat informasi seperlunya dan hanya membutuhkan beberapa baris saja. Biasanya koran-koran mensyaratkan iklan baris minimal 2-3 baris.

Karena tidak menggunakan gambar sebagai daya tariknya, iklan baris dimuat secara berkelompok sesuai dengan isinya. Misalnya, kelompok barang yang dijual dipasang dalam baris yang berbeda dengan kelompok barang yang dicari. Selanjutnya, barang-barang yang dijual pun diklasifikasikan lebih jauh: tanah, rumah, mobil, motor, dan lain-lain.

2.7.2. Iklan Banner

Banner adalah papan iklan elektronik. Memasang iklan *banner* merupakan bentuk paling umum digunakan didalam periklanan *internet*. Biasanya sebuah *banner* berisi teks singkat atau pesan bergambar untuk mempromosikan suatu produk atau *vendor*. *Banner* juga mungkin berisi video klip atau suara. Pada saat pelanggan mengklik suatu banner, ia akan ditransfer ke *home page* atau *storefront* pengiklannya.

2.8. FlowChart

Flowchart adalah bagan yang menunjukkan urutan dan hubungan antar proses beserta instruksinya dan gambaran itu dinyatakan dengan simbol. Adapun simbol-simbol yang digunakan tertera digambar berikut ini.

Simbol	Nama	Keterangan
	<i>Process</i>	Untuk menunjukkan proses pengolahan yang dilakukan oleh computer.
	<i>Decision</i>	Untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.
	<i>Terminal</i>	Untuk permulaan atau akhir dari suatu program.
	<i>Input-output</i>	Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	<i>Menu</i>	Untuk menunjukkan menu pilihan di dalam program.
	<i>Connector</i>	Untuk keluar masuk/masuk prosedur atau proses dalam lembar/halaman yang sama.
	<i>Off-line connector</i>	Untuk keluar masuk/masuk prosedur atau proses dalam lembar/halaman yang lain.

Gambar 2.3 Simbol *Flowchart*

2.9. PHP

PHP dikenal sebagai sebuah bahasa *scripting* yang menyatu dengan *tag-tag* HTML, dieksekusi di *server* dan digunakan untuk membuat halaman *web* dinamis. Versi pertama PHP dibuat oleh Rasmus Lerdorf pada tahun 1995, berupa sekumpulan *script* PERL yang digunakan oleh Rasmus Lerdorf untuk membuat halaman *web* yang dinamis pada *homepage* pribadinya.

Berdasarkan hasil *survey* Netcrat pada bulan Desember 1999, lebih dari satu juta *site* yang menggunakan PHP, termasuk perusahaan besar seperti Mitsubishi, Redhat, NASA, Ericson dan banyak lagi. Berdasarkan *survey* Esoft pada bulan November 1999, 23% pengguna *Apache Server* menggunakan PHP (55% *webserver* dunia menggunakan *Apache*).

Dasar pertimbangan untuk mengembangkan kemampuan *object oriented* dari PHP adalah perkembangan aplikasi web sebagai sebuah *platform* yang terus meluas dengan cepat karena ruang lingkup aplikasi web semakin luas, sehingga aplikasi web yang dibangun juga menjadi semakin besar, rumit, dan kompleks.

Dengan kemampuan untuk mengimplementasikan konsep *object oriented*, PHP akan lebih mudah dapat berhubungan dan mengakses komponen-komponen yang dibuat dengan bahasa *object oriented* lain seperti Java atau C++ dan bahkan Visual Basic.

Berikut ini akan dijelaskan beberapa keunggulan PHP (Farid Azis, M, 2001):

- a. *Life cycle* yang singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
- b. *Cross platform*, PHP dapat dipakai di hampir semua *web server* yang ada di pasaran (*Apache*, *fhhttpd*, *phhttpd*, *Microsoft IIS* dan lain-lain) yang dijalankan pada berbagai sistem operasi. Dengan demikian proses *developing* dapat dilakukan menggunakan sistem operasi yang digunakan setelah *publish* (misalnya, pengembangan di *Windows* kemudian dipasang di *web server* yang menggunakan sistem operasi *Linux*).

- c. PHP mendukung banyak paket *database* baik yang komersil maupun yang non komersil.

2.10. NuSOAP

NuSOAP adalah sebuah kumpulan *class-class* PHP yang memungkinkan *user* untuk mengirim dan menerima pesan SOAP melalui protokol HTTP. NuSOAP didistribusikan oleh *NuSphere Corporation* (<http://www.nusphere.com>) sebagai *open source toolkit* dibawah lisensi GNU/GPL.

NuSOAP merupakan *toolkit web service* berbasis komponen. NuSOAP memiliki sebuah *class* dasar yang menyediakan method seperti serialisasi variabel dan pemaketan SOAP-Envelope. Interaksi *web service* dilakukan dengan *class client* yang disebut dengan *class "soapclient"* dan *class server* yang disebut dengan *class "soap-server"*. *Class-class* ini mengizinkan *user* untuk melakukan proses pengiriman dan penerimaan pesan-pesan SOAP dengan bantuan beberapa *class-class* pendukung lainnya untuk melengkapi proses tersebut.

BAB II

LANDASAN TEORI

2.1. *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML (*eXtensible Markup Language*), sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *web service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di *upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di *port* 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

Web Service dapat diimplementasikan pada lingkungan internal (*Intranet*) untuk kebutuhan integrasi antar sistem aplikasi (EAI=*Enterprise Application Integration*) ataupun pada lingkungan eksternal (*Internet*) untuk mendukung aplikasi *business-to-business* (*e-business*).

Keuntungan penggunaan *Web Service* : (Wikipedia.org, 2010)

1. Format penggunaan terbuka untuk semua *platform*.
2. Mudah di mengerti dan mudah men-*debug*.
3. Menggunakan standar-standar “membuka *service* sekali” dan mempunyai pemakai banyak.
4. Mudah untuk menengahi pesan-pesan proses dan menambahkan nilai.
5. *Routing* and pengiriman.
6. *Security*.
7. *Management* and *monitoring*.
8. *Schema* and *service design*.
9. Akselerasi.
10. Mudah untuk mengembangkan dengan *semantic transport* tambahan.
11. Terbuka, standar-standar berbasis teks.
12. Pencapaian modular.
13. Tidak mahal untuk diimplementasikan (relatif).
14. Mengurangi biaya integrasi aplikasi *enterprise*.
15. Implementasi yang *incremental*.

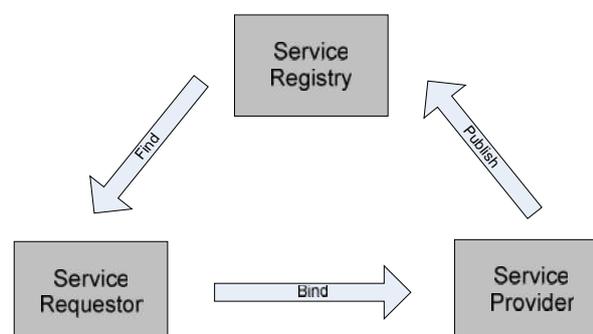
Web service ternyata sangat berbeda dengan *website*. Perbedaan yang paling terlihat adalah *website* dibuat untuk memiliki tampilan atau *user interface* yang bagus sedangkan *web service* tidak memiliki tampilan. Itu karena *web service* tidak dibuat untuk berinteraksi langsung dengan *user*. Sesuai dengan kata *service* yang ada pada namanya, *web service* hanya menyediakan *service* atau layanan. Layanan tersebutlah yang kemudian akan digunakan atau dipanggil oleh aplikasi lainnya. Dengan demikian yang akan menjadi *interface* adalah aplikasi yang memanggilnya bukan *web service* itu sendiri.

2.2. Arsitektur Web Service

Konsep arsitektur yang mendasari teknologi *Web service* adalah *Service Oriented Architecture* (SOA). Dalam arsitektur ini, suatu aplikasi dimodelkan sebagai komposisi dari sekumpulan *service* yang disediakan oleh suatu komponen. Lokasi keberadaan komponen tersebut dapat ditemukan oleh *client* secara dinamis, dalam arti tidak dinyatakan secara statis tetapi menggunakan mekanisme *discovery* untuk mencari keberadaan komponen tersebut. Demikian pula, *client* dapat meminta (*invoke*) *service* tersebut secara dinamis pula. Dalam arsitektur ini, SOA mendefinisikan 3 peran berbeda yang menunjukkan peran dari masing-masing komponen dalam sistem, yaitu :

1. *Service provider*, yaitu suatu entitas yang menyediakan *interface* terhadap sistem yang menjalankan suatu sekumpulan tugas tertentu. *Service provider* dapat merepresentasikan suatu entitas bisnis ataupun suatu komponen *software* yang *reusable*.
2. *Service requestor*, yaitu suatu entitas yang meminta/memperoleh (dan menemukan) *software service* dalam rangka menyelesaikan suatu tugas tertentu atau menyediakan solusi bisnis tertentu.
3. *Service registry*, yaitu entitas yang bertindak sebagai penyimpan (*repository*) suatu *software service* yang dipublikasikan oleh *Service provider*.

Peran dan interaksi antar komponen tersebut ditunjukkan pada gambar dibawah ini :



Gambar 2.1 - Konsep *Service Oriented Architecture*

Secara umum, *web service* memiliki tiga operasi yang terlibat di dalamnya, yaitu:

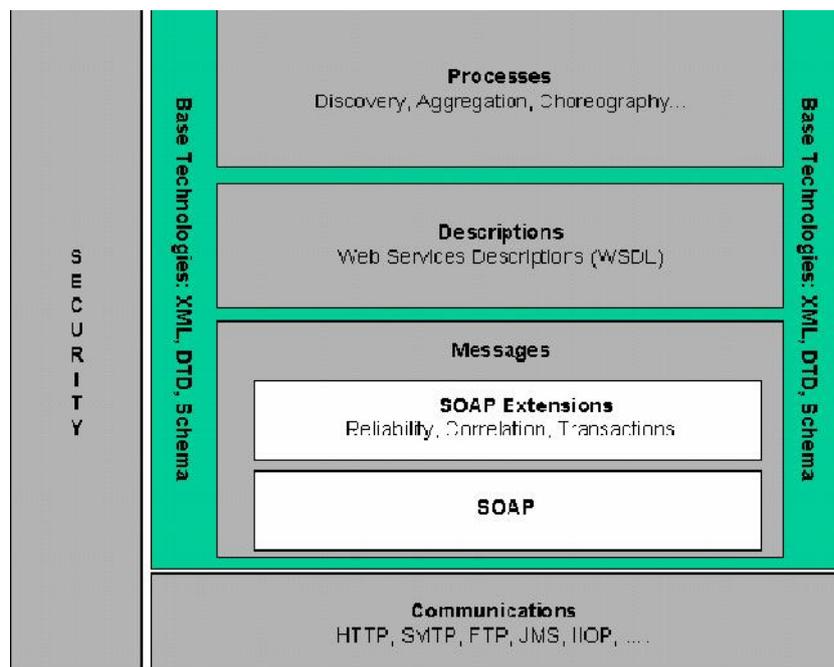
1. *Publish/Unpublish*: Menerbitkan/menghapus layanan ke dalam atau dari *registry*.
2. *Find*: *Service requestor* mencari dan menemukan layanan yang dibutuhkan.
3. *Bind*: *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke *service provider* untuk melakukan interaksi dan mengakses layanan/*service* yang disediakan oleh *service provider*.

Dalam arsitektur ini, *service* dapat didefinisikan sebagai komponen (*software component*) yang memiliki karakteristik :

1. Dapat dideskripsikan dalam suatu bahasa formal
2. Dapat dipublikasikan pada suatu *registry of service*
3. Dapat ditemukan (*discover*) menggunakan mekanisme standar
4. Dapat diminta/diperoleh (*invoke*) melalui jaringan
5. Dapat dibangun bersama *service* lain

Model Arsitektur *Web Service* menggunakan konsep SOA tersebut ,yaitu :

1. *Web Service* menggunakan standar terbuka, yaitu : XML (*eXtensible Markup Language*) sebagai bahasa untuk mendeskripsikan data, SOAP (*Simple Object Access Protocol*) sebagai protokol komunikasi antar komponen , WSDL (*Web Service Description Language*) untuk mendeskripsikan *service*, UDDI (*Universal Description, Discovery and Integration*) untuk *service discovery*, dan HTTP sebagai *transport layer*.
2. Standar *Web Service* merupakan *message-based*, bukan *API-based*. Dalam hal ini, komunikasi antara *Web Service*, *Service Requester* dan *Service Registry* menggunakan pesan SOAP berbasis XML. W3C mengembangkan *draft* arsitektur *Web Service* yang terdiri dari beberapa teknologi yang saling berhubungan dan berlapis seperti digambarkan sebagai berikut :



Gambar 2.2 Arsitektur Web Service

2.3. SOAP (*Simple Object Access Protocol*)

SOAP merupakan protokol komunikasi yang ringan berbasis *eXtensible Markup Language* (XML). Pesan SOAP berbentuk sekumpulan *XML Schema* yang mendefinisikan format untuk mentransmisikan pesan XML melalui jaringan, termasuk tipe data dan cara menstrukturkan pesan secara tepat sehingga dapat mudah dipahami oleh *server* atau *end-point* lainnya.

Pesan SOAP terdiri dari 3 bagian, yaitu :

1. *Envelope*, suatu kerangka yang mendefinisikan apa yang ada dalam pesan dan bagaimana pesan harus diproses serta menunjukkan resipien dari pesan tersebut.
2. *Header*, berisi informasi yang berhubungan dengan keamanan dan *routing*. Keberadaan *Header* dalam SOAP bersifat *optional*.
3. *Body*, berisi data yang berhubungan dengan aplikasi tertentu yang sedang dipertukarkan. Data di *mark-up* sebagai XML dan dimasukkan dalam format yang spesifik yang didefinisikan dalam *XML Schema*.

SOAP (*Simple Object Access Protocol*) adalah sebuah *XML based markup language* untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data *object* didalam *network*. SOAP mendefinisikan empat aspek didalam komunikasi: *Message envelope*, *Encoding*, *RPC call convention*, dan bagaimana menyatukan sebuah *message* didalam protokol *transport*.

Sebuah SOAP *message* terdiri dari SOAP *Envelope* dan bisa terdiri dari *attachments* atau tidak memiliki *attachment*. SOAP *envelope* tersusun dari SOAP *header* dan SOAP *body*, sedangkan SOAP *attachment* membolehkan non-XML data untuk dimasukkan kedalam SOAP *message*, di-*encoded*, dan diletakkan kedalam SOAP *message* dengan menggunakan *MIME-multipart*.

SOAP menspesifikan secara jelas bagaimana cara untuk meng-*encode header* HTTP dan *file* XML sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan.

SOAP adalah protokol ringan yang ditujukan untuk pertukaran informasi struktur pada lingkup desentralisasi, dan terdistribusi. SOAP menggunakan teknologi XML untuk mendefinisikan rangka kerja pemesanan terekstrensi di mana menyediakan konstruksi pesan yang dapat dipertukarkan pada protokol berbeda. Rangka kerja dirancang bebas dari model pemrograman dan spesifikasi implementasi semantik.

SOAP tidak berpasangan erat dengan protokol *transport* tertentu. Spesifikasi SOAP menerangkan bagaimana pesan-pesan SOAP seharusnya diikat ke HTTP. Namun, sebuah pesan SOAP tidak lebih daripada sebuah dokumen XML. Jadi, ia dapat diangkut melalui protokol apa pun yang mampu mengirimkan teks. (Scott Short, 2003).

Peran SOAP di dalam teknologi *web service* adalah sebagai protokol pemaketan untuk pesan-pesan (*messages*) yang digunakan secara bersama oleh aplikasi-aplikasi penggunanya. Spesifikasi yang digunakan tidak lebih seperti sebuah amplop biasa berbasis XML untuk informasi yang ditransfer, serta sekumpulan aturan bagi translasi aplikasi dan tipe-tipe data platform yang spesifik

menjadi bentuk XML. Desain bentuk dari SOAP membuatnya cocok untuk berbagai pertukaran pesan pada aplikasi. Berikut adalah salah satu contoh bentuk sederhana pesan SOAP :

Contoh 1 : SOAP Message Embedded in HTTP Request

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:
encodingStyle="http://schemas.xmlsoap.org/soap/encoding
/">
<SOAP-ENV:Body>
<m:GetLastTradePrice xmlns:m="Some-URI">
<symbol>DIS</symbol>
</m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Contoh 2 : SOAP Message Embedded in HTTP Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/enco
ding/">
<SOAP-ENV:Body>
<m:GetLastTradePriceResponse xmlns:m="Some-URI">
<Price>34.5</Price>
</m:GetLastTradePriceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.4. WSDL (*Web Services Description Language*)

WSDL (*Web Services Description Language*) adalah dialek berbasis XML yang lapisannya ada diatas skema yang menjelaskan layanan web. Dokumen WSDL menyediakan informasi yang dibutuhkan *client* untuk berinteraksi dengan layanan web. WSDL dapat diperluas dan dipakai untuk menjelaskan hampir semua layanan jaringan, termasuk SOAP lewat HTTP. Bahkan protokol yang bukan berbasis XML, seperti DCOM lewat UDP. (Scott Short,2003)

WSDL (*Web Services Description Language*) adalah format XML yang diterbitkan untuk menerangkan *web service*. WSDL mendefinisikan: (Wikipedia.org, 2010)

- Pesan-pesan (baik yang abstrak dan kongkrit) yang dikirim ke dan menuju *web service*
- Koleksi-koleksi digital dari pesan-pesan (*port type*, antarmuka)
- Bagaimana *port type* yang ditentukan dijadikan *wire* protokol
- Dimana *service* ditempatkan

WSDL digunakan di mana skema XML tidak digunakan lagi dengan menyediakan jalur pesan-pesan grup menjadi operasi-operasi dan operasi-operasi menjadi antarmuka. Ini juga menyediakan jalur untuk medefinisikan *binding-binding* untuk setiap antamuka dan kombinasi protokol sepanjang alamat titik akhir untuk setiap kalinya. Definisi WSDL yang lengkap terdiri dari seluruh informasi yang dibutuhkan untuk meminta *web service*. Pengembang yang mau mempermudah yang lain untuk mengakses *service-service* nya harus menyediakan definisi-definisi WSDL.

Sebuah dokumen WSDL dapat dibagi menjadi 6 bagian besar. Masing-masing bagian tersebut ditandai tag-tag tertentu. (Lucky, 2008)

Tabel 2.1 *Tag-tag* pada dokumen WSDL

Tag	Keterangan
<i>Definitions</i>	Tag ini adalah tag tertinggi dalam dokumen WSDL atau biasa disebut dengan tag <i>root</i> . Tag-tag lainnya harus berada didalam tag ini. Salah satu informasi yang ada didalam tag ini adalah <i>namespace</i> .
<i>Types</i>	Didalam tag ini didefinisikan tipe-tipe data yang digunakan oleh <i>web service</i> . Akan tetapi yang didefinisikan hanyalah tipe-tipe data kompleks seperti <i>array</i> atau <i>struct</i> . Sedangkan tipe-tipe data sederhana yang sudah <i>built-in</i> seperti <i>integer</i> , <i>double</i> , <i>string</i> , dan lain-lain tidak didefinisikan lagi.
<i>Messages</i>	Informasi yang ada didalam tag ini adalah <i>message</i> atau parameter <i>input</i> atau <i>output</i> apa saja yang akan digunakan
<i>PortType</i>	Didalam tag ini didefinisikan <i>method</i> atau fungsi yang disediakan oleh <i>web service</i>
<i>Binding</i>	Informasi yang ada didalam tag ini bagaimana sebuah operasi atau fungsi diimplementasikan. Apakah fungsi tersebut dapat dipanggil menggunakan protokol HTTP atau SMTP. Informasi seperti <i>soap action</i> juga didefinisikan didalam tag ini
<i>Service</i>	Didalam tag ini didefinisikan lokasi dari <i>web service</i>

2.5. UDDI (*Universal Description, Discovery, and Integration*)

UDDI adalah spesifikasi yang telah diakui secara luas untuk pendaftaran suatu *web service*. UDDI mendefinisikan metadata dan protokol yang digunakan untuk mengetahui dan mengubah informasi dari suatu *web service*. Langkah pertama dalam menemukan *web service* adalah dengan meminta alamat tempat penyimpanan dari *web service* yang akan dipakai yang biasa disebut *directory*.

Setelah menemukan *directory*, peminta *web service* dapat mengirimkan permintaan lagi untuk mendapatkan informasi detail tentang *web service* (misalnya penyedia *web service* dan dimana diletakkan). Selanjutnya perangkat lunak menggunakan informasi yang didapat untuk secara dinamik mengakses *web service* yang diinginkan.

2.6. *eXtensible Markup Language*

XML kependekan dari *eXtensible Markup Language*, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan *markup language* yang tidak kalah hebatnya dengan SGML.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup (diawali dengan '</' dan diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal `<form name="isidata">`). Hanya bedanya, HTML mendefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak.

XML adalah *meta-language*. *Meta-language* adalah bahasa yang dapat digunakan untuk mendefinisikan bahasa-bahasa lainnya. Contohnya adalah menggunakan XML untuk mendefinisikan bahasa seperti WML.

Dengan XML dapat dilakukan:

1. Mendefinisikan struktur data
2. Membuat struktur tersebut *platform independent*
3. Memproses data terdefinisi oleh XML secara otomatis
4. Mendefinisikan *tag* sendiri

Beberapa alasan mengapa menggunakan XML :

1. Pertukaran data.
XML digunakan dalam pertukaran data. XML memungkinkan untuk mendefinisikan data dengan jelas. Pengirim dan penerima data akan menggunakan XML untuk memahami data yang telah dikirimkan. Dengan menggunakan XML, pengirim dan penerima memiliki interpretasi sama tentang data yang digunakan.
2. Pengganti *Electronic Data Interchange* (EDI). EDI telah digunakan dalam pertukaran data. EDI mahal, karena menggunakan *dedicated communication infrastructure*. Definisi yang digunakan jauh dari fleksibel. XML adalah pengganti yang baik terhadap EDI. XML menggunakan *Internet* dalam pertukaran data dan fleksibel.
3. Kemungkinan lainnya. XML dapat digunakan untuk mendefinisikan bahasa lainnya. Salah satu contohnya adalah WML (Wireless Markup Language), bahasa yang digunakan dalam komunikasi WAP. Dengan kata lain, WML adalah salah satu dialek XML.

Pada dasarnya, kemampuan XML memiliki potensi untuk dapat digunakan pada kondisi dimana data dimasukkan/dikeluarkan, disimpan, atau ditransmisikan dari suatu tempat ke tempat lain. XML dapat digunakan sebagai pengganti (atau pelengkap) database tradisional, atau mentransfer informasi antar bisnis. Organisasi berita, termasuk individu, juga dapat menggunakan XML untuk mendistribusikan *syndicated news stories*.

Bagian-Bagian dari Dokumen XML

Sebuah dokumen XML terdiri dari bagian bagian yang disebut dengan *node*. *Node-node* itu adalah:

- *Root node* yaitu *node* yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu *root node*. *Node-node* yang lainnya berada di dalam *root node*.

- *Element node* yaitu bagian dari dokumen XML yang ditandai dengan *tag* pembuka dan *tag* penutup, atau bisa juga sebuah *tag* tunggal elemen kosong seperti `<anggota nama="budi" />`. *Root node* biasa juga disebut *root element*
- *Attribute note* termasuk nama dan nilai atribut ditulis pada *tag* awal sebuah elemen atau pada *tag* tunggal.
- *Text node*, adalah *text* yang merupakan isi dari sebuah elemen, ditulis diantara *tag* pembuka dan *tag* penutup
- *Comment node* adalah baris yang tidak dieksekusi oleh *parser*
- *Processing Instruction node*, adalah perintah pengolahan dalam dokumen XML. Node ini ditandai awali dengan karakter `<?` Dan diakhiri dengan `?>`. Tapi perlu diingat bahwa header standard XML `<?xml version="1.0" encoding="iso-8859-1"?>` bukanlah *processing instruction node*. *Header standard* bukanlah bagian dari hirarki pohon dokumen XML.
- *NameSpace Node*, node ini mewakili deklarasi *namespace*

2.7. Iklan

Iklan ialah promosi barang, jasa, perusahaan dan ide yang harus dibayar oleh sebuah sponsor. Pemasaran melihat iklan sebagai bagian dari strategi promosi secara keseluruhan. Komponen lainnya dari promosi termasuk publisitas, relasi publik, penjualan, dan promosi penjualan.

2.7.1. Iklan Baris

Iklan baris (bahasa Inggris: *Classified advertising*) adalah salah satu cara promosi barang dan jasa yang umumnya ditemukan di koran. Cara ini merupakan pengembangan dari promosi iklan yang mengutamakan daya tarik dengan gambar dan dengan informasi yang lebih lengkap dan terinci. (Wikipedia.org, 2010)

Iklan baris mengutamakan informasi yang paling inti yang perlu diketahui oleh peminatnya. Karena itu biasanya iklan baris hanya memuat informasi seperlunya dan hanya membutuhkan beberapa baris saja. Biasanya koran-koran mensyaratkan iklan baris minimal 2-3 baris.

Karena tidak menggunakan gambar sebagai daya tariknya, iklan baris dimuat secara berkelompok sesuai dengan isinya. Misalnya, kelompok barang yang dijual dipasang dalam baris yang berbeda dengan kelompok barang yang dicari. Selanjutnya, barang-barang yang dijual pun diklasifikasikan lebih jauh: tanah, rumah, mobil, motor, dan lain-lain.

2.7.2. Iklan Banner

Banner adalah papan iklan elektronik. Memasang iklan *banner* merupakan bentuk paling umum digunakan didalam periklanan *internet*. Biasanya sebuah *banner* berisi teks singkat atau pesan bergambar untuk mempromosikan suatu produk atau *vendor*. *Banner* juga mungkin berisi video klip atau suara. Pada saat pelanggan mengklik suatu banner, ia akan ditransfer ke *home page* atau *storefront* pengiklannya.

2.8. FlowChart

Flowchart adalah bagan yang menunjukkan urutan dan hubungan antar proses beserta instruksinya dan gambaran itu dinyatakan dengan simbol. Adapun simbol-simbol yang digunakan tertera digambar berikut ini.

Simbol	Nama	Keterangan
	<i>Process</i>	Untuk menunjukkan proses pengolahan yang dilakukan oleh computer.
	<i>Decision</i>	Untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.
	<i>Terminal</i>	Untuk permulaan atau akhir dari suatu program.
	<i>Input-output</i>	Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	<i>Menu</i>	Untuk menunjukkan menu pilihan di dalam program.
	<i>Connector</i>	Untuk keluar masuk/masuk prosedur atau proses dalam lembar/halaman yang sama.
	<i>Off-line connector</i>	Untuk keluar masuk/masuk prosedur atau proses dalam lembar/halaman yang lain.

Gambar 2.3 Simbol *Flowchart*

2.9. PHP

PHP dikenal sebagai sebuah bahasa *scripting* yang menyatu dengan *tag-tag* HTML, dieksekusi di *server* dan digunakan untuk membuat halaman *web* dinamis. Versi pertama PHP dibuat oleh Rasmus Lerdorf pada tahun 1995, berupa sekumpulan *script* PERL yang digunakan oleh Rasmus Lerdorf untuk membuat halaman *web* yang dinamis pada *homepage* pribadinya.

Berdasarkan hasil *survey* Netcraft pada bulan Desember 1999, lebih dari satu juta *site* yang menggunakan PHP, termasuk perusahaan besar seperti Mitsubishi, Redhat, NASA, Ericson dan banyak lagi. Berdasarkan *survey* Esoft pada bulan November 1999, 23% pengguna *Apache Server* menggunakan PHP (55% *webserver* dunia menggunakan *Apache*).

Dasar pertimbangan untuk mengembangkan kemampuan *object oriented* dari PHP adalah perkembangan aplikasi web sebagai sebuah *platform* yang terus meluas dengan cepat karena ruang lingkup aplikasi web semakin luas, sehingga aplikasi web yang dibangun juga menjadi semakin besar, rumit, dan kompleks.

Dengan kemampuan untuk mengimplementasikan konsep *object oriented*, PHP akan lebih mudah dapat berhubungan dan mengakses komponen-komponen yang dibuat dengan bahasa *object oriented* lain seperti Java atau C++ dan bahkan Visual Basic.

Berikut ini akan dijelaskan beberapa keunggulan PHP (Farid Azis, M, 2001):

- a. *Life cycle* yang singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
- b. *Cross platform*, PHP dapat dipakai di hampir semua *web server* yang ada di pasaran (*Apache*, *fhhttpd*, *phhttpd*, *Microsoft IIS* dan lain-lain) yang dijalankan pada berbagai sistem operasi. Dengan demikian proses *developing* dapat dilakukan menggunakan sistem operasi yang digunakan setelah *publish* (misalnya, pengembangan di *Windows* kemudian dipasang di *web server* yang menggunakan sistem operasi *Linux*).

- c. PHP mendukung banyak paket *database* baik yang komersil maupun yang non komersil.

2.10. NuSOAP

NuSOAP adalah sebuah kumpulan *class-class* PHP yang memungkinkan *user* untuk mengirim dan menerima pesan SOAP melalui protokol HTTP. NuSOAP didistribusikan oleh *NuSphere Corporation* (<http://www.nusphere.com>) sebagai *open source toolkit* dibawah lisensi GNU/GPL.

NuSOAP merupakan *toolkit web service* berbasis komponen. NuSOAP memiliki sebuah *class* dasar yang menyediakan method seperti serialisasi variabel dan pemaketan SOAP-Envelope. Interaksi *web service* dilakukan dengan *class client* yang disebut dengan *class "soapclient"* dan *class server* yang disebut dengan *class "soap-server"*. *Class-class* ini mengizinkan *user* untuk melakukan proses pengiriman dan penerimaan pesan-pesan SOAP dengan bantuan beberapa *class-class* pendukung lainnya untuk melengkapi proses tersebut.

BAB III

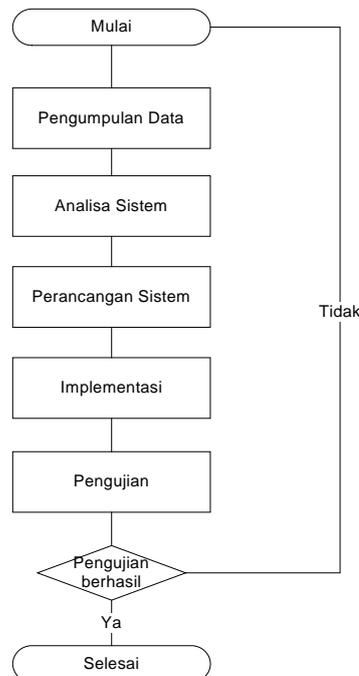
METODOLOGI PENELITIAN

Metodologi penelitian merupakan cara yang digunakan dalam memperoleh berbagai data untuk kemudian diproses menjadi informasi yang lebih akurat sesuai permasalahan yang akan diteliti. Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya.

3.1. Tahapan Penelitian

Penjelasan tentang metodologi penelitian didasarkan pada tujuan dari Tugas Akhir. Metodologi penelitian yang digunakan dalam penyusunan Tugas Akhir ini akan melalui beberapa tahapan yang membentuk sebuah alur yang sistematis. Tahapan-tahapan itu meliputi: Pengumpulan Data, Analisa dan Perancangan Sistem, Implementasi dan Pengujian.

Tahapan penelitian yang akan dilaksanakan pada perancangan aplikasi ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Tahapan Penelitian

3.1.1. Pengumpulan Data

Pengumpulan materi merupakan tahapan persiapan yang harus dilaksanakan terlebih dahulu sebelum dilakukan penelitian. Berikut merupakan aktivitas yang dilaksanakan dalam pengumpulan sumber teori :

- Studi Pustaka

Studi pustaka bermanfaat untuk mendukung penelitian yang akan dilaksanakan. Studi pustaka dilakukan dengan tujuan untuk menyelesaikan permasalahan yang diteliti, serta mendapatkan dasar-dasar yang kuat dalam menerapkan suatu metode yang nantinya dapat digunakan dalam tugas akhir ini.

Pengumpulan teori-teori yang mendukung dalam penelitian ini merupakan kegiatan dalam studi pustaka. Teori-teori bersumber dari buku, jurnal dan penelitian yang terkait dengan *eXtensible Markup Language* (XML), *Web Service*, *Simple Object Access Protocol* (SOAP), Bahasa Pemrograman PHP dan NuSOAP.

3.1.2. Analisa Sistem

Tahapan ini mengenali seluruh permasalahan yang muncul dalam mengenali komponen-komponen sistem dan mempelajari prosedur sistem yang akan dibuat, serta menganalisis solusi maksimal atas kebutuhan sistem.

3.1.2.1. Analisa Sistem Lama

Pada tahap ini dilakukan analisa terhadap sistem lama atau metode pengerjaan yang sedang berlangsung, termasuk untuk mengetahui kelemahan yang dimiliki oleh sistem lama tersebut.

3.1.2.2. Analisa Sistem Baru

Setelah menganalisa sistem lama, maka tahapan dapat dilanjutkan dengan menganalisa sistem baru. Dalam tahapan ini, akan diidentifikasi cara kerja dari sistem baru yang akan dibangun.

3.1.2.3. Deskripsi Umum Perangkat Lunak

Deskripsi umum perangkat lunak dibuat sebagai suatu gambaran tentang rancangan sistem yang akan dibuat. Dalam deskripsi Umum

Perangkat Lunak ini dilakukan dua tahapan yakni Deskripsi Umum Sistem dan Karakteristik Pengguna.

3.1.2.4. Analisis Data Sistem

Analisis ini dilakukan dengan menggunakan beberapa tahapan yakni Analisis Masukan (*input*), Analisis Proses dan Analisis *Output*.

3.1.2.5. Analisis Fungsional Sistem

Analisis yang digunakan pada sistem adalah pemodelan fungsional. Pemodelan fungsional merupakan pemodelan yang menggambarkan suatu masukan yang diproses pada sistem menjadi keluaran yang dibutuhkan bagi pengguna sistem.

3.1.3. Perancangan

Tahapan penganalisaan dan perancangan Alat bantu yang digunakan adalah Model Sistem, *Flowchart*, *Context Diagram*, *Data flow Diagram*, *ER-Diagram* dan *interface* rancangan sistem. Dan perangkat lunak yang digunakan untuk membantu proses perancangan sistem yakni: Microsoft Word 2007, dan Microsoft Visio 2003.

3.1.4. Implementasi

Implementasi akan dilakukan pada tahap ini, yaitu setelah fase perancangan selesai. Tahap implementasi merupakan tahap dimana hasil dari perancangan dituliskan kedalam bahasa pemrograman dan database sehingga mencapai hasil akhir berupa aplikasi xml *web service* untuk periklanan *online*. Perangkat lunak yang digunakan adalah PHP, MySQL, serta *library* NuSOAP.

3.1.5. Pengujian

Ini bertujuan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan. Apabila didalam *testing* tersebut terdapat kesalahan (*error*), maka harus dilakukan perbaikan kembali pada sistem tersebut.

BAB III

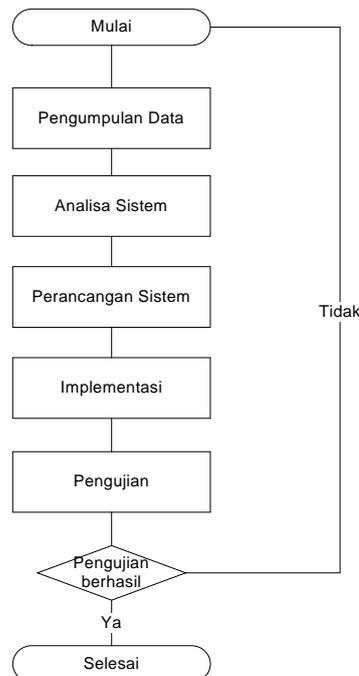
METODOLOGI PENELITIAN

Metodologi penelitian merupakan cara yang digunakan dalam memperoleh berbagai data untuk kemudian diproses menjadi informasi yang lebih akurat sesuai permasalahan yang akan diteliti. Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya.

3.1. Tahapan Penelitian

Penjelasan tentang metodologi penelitian didasarkan pada tujuan dari Tugas Akhir. Metodologi penelitian yang digunakan dalam penyusunan Tugas Akhir ini akan melalui beberapa tahapan yang membentuk sebuah alur yang sistematis. Tahapan-tahapan itu meliputi: Pengumpulan Data, Analisa dan Perancangan Sistem, Implementasi dan Pengujian.

Tahapan penelitian yang akan dilaksanakan pada perancangan aplikasi ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Tahapan Penelitian

3.1.1. Pengumpulan Data

Pengumpulan materi merupakan tahapan persiapan yang harus dilaksanakan terlebih dahulu sebelum dilakukan penelitian. Berikut merupakan aktivitas yang dilaksanakan dalam pengumpulan sumber teori :

- Studi Pustaka

Studi pustaka bermanfaat untuk mendukung penelitian yang akan dilaksanakan. Studi pustaka dilakukan dengan tujuan untuk menyelesaikan permasalahan yang diteliti, serta mendapatkan dasar-dasar yang kuat dalam menerapkan suatu metode yang nantinya dapat digunakan dalam tugas akhir ini.

Pengumpulan teori-teori yang mendukung dalam penelitian ini merupakan kegiatan dalam studi pustaka. Teori-teori bersumber dari buku, jurnal dan penelitian yang terkait dengan *eXtensible Markup Language* (XML), *Web Service*, *Simple Object Access Protocol* (SOAP), Bahasa Pemrograman PHP dan NuSOAP.

3.1.2. Analisa Sistem

Tahapan ini mengenali seluruh permasalahan yang muncul dalam mengenali komponen-komponen sistem dan mempelajari prosedur sistem yang akan dibuat, serta menganalisis solusi maksimal atas kebutuhan sistem.

3.1.2.1. Analisa Sistem Lama

Pada tahap ini dilakukan analisa terhadap sistem lama atau metode pengerjaan yang sedang berlangsung, termasuk untuk mengetahui kelemahan yang dimiliki oleh sistem lama tersebut.

3.1.2.2. Analisa Sistem Baru

Setelah menganalisa sistem lama, maka tahapan dapat dilanjutkan dengan menganalisa sistem baru. Dalam tahapan ini, akan diidentifikasi cara kerja dari sistem baru yang akan dibangun.

3.1.2.3. Deskripsi Umum Perangkat Lunak

Deskripsi umum perangkat lunak dibuat sebagai suatu gambaran tentang rancangan sistem yang akan dibuat. Dalam deskripsi Umum

Perangkat Lunak ini dilakukan dua tahapan yakni Deskripsi Umum Sistem dan Karakteristik Pengguna.

3.1.2.4. Analisis Data Sistem

Analisis ini dilakukan dengan menggunakan beberapa tahapan yakni Analisis Masukan (*input*), Analisis Proses dan Analisis *Output*.

3.1.2.5. Analisis Fungsional Sistem

Analisis yang digunakan pada sistem adalah pemodelan fungsional. Pemodelan fungsional merupakan pemodelan yang menggambarkan suatu masukan yang diproses pada sistem menjadi keluaran yang dibutuhkan bagi pengguna sistem.

3.1.3. Perancangan

Tahapan penganalisaan dan perancangan Alat bantu yang digunakan adalah Model Sistem, *Flowchart*, *Context Diagram*, *Data flow Diagram*, *ER-Diagram* dan *interface* rancangan sistem. Dan perangkat lunak yang digunakan untuk membantu proses perancangan sistem yakni: Microsoft Word 2007, dan Microsoft Visio 2003.

3.1.4. Implementasi

Implementasi akan dilakukan pada tahap ini, yaitu setelah fase perancangan selesai. Tahap implementasi merupakan tahap dimana hasil dari perancangan dituliskan kedalam bahasa pemrograman dan database sehingga mencapai hasil akhir berupa aplikasi xml *web service* untuk periklanan *online*. Perangkat lunak yang digunakan adalah PHP, MySQL, serta *library* NuSOAP.

3.1.5. Pengujian

Ini bertujuan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan. Apabila didalam *testing* tersebut terdapat kesalahan (*error*), maka harus dilakukan perbaikan kembali pada sistem tersebut.

BAB IV

ANALISA DAN PERANCANGAN

Bab ini akan membahas analisa dan perancangan yang merupakan dua tahapan yang saling berhubungan erat. Analisa bertujuan untuk mengidentifikasi permasalahan dan kebutuhan sistem, sementara perancangan bertujuan untuk mengimplementasikan permasalahan yang telah diidentifikasi kedalam sebuah aplikasi.

4.1. Analisa Sistem

Analisa merupakan suatu aktivitas kreatif dalam proses mentransformasikan suatu definisi permasalahan dari sebuah ketidakjelasan, meliputi poses memahami permasalahan, mengetahui keterbatasan dan menggagas metode yang tepat untuk mengatasi permasalahan serta keterbatasan tersebut

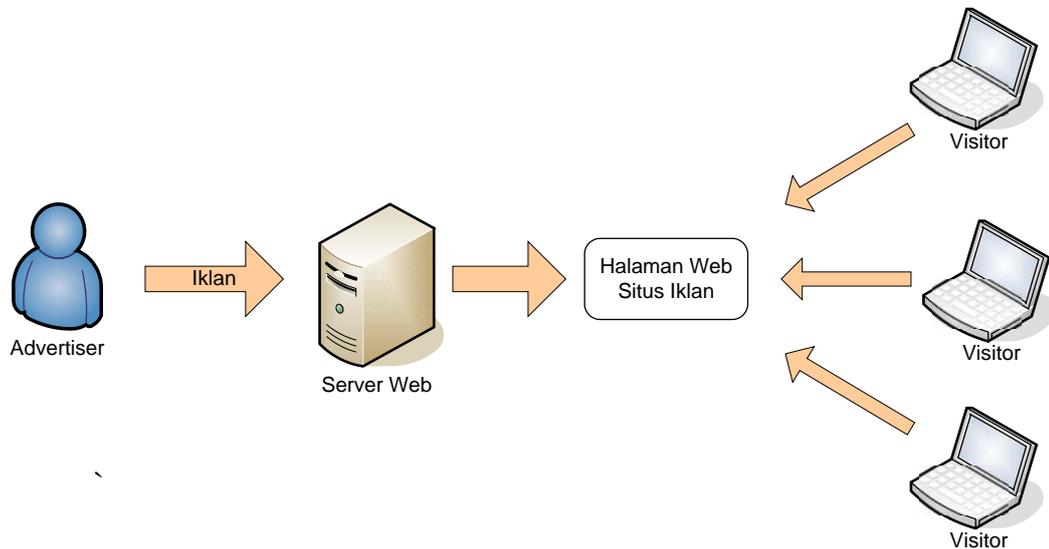
Analisa sistem dibagi dalam dua bagian pembahasan yaitu analisa pada sistem yang berjalan dan analisa pada sistem yang akan dibangun.

4.1.1. Sistem yang Berjalan

Secara umum sistem yang berjalan memiliki alur proses kerja sebagai berikut :

1. *Advertiser* melakukan registrasi untuk pendaftaran sebagai *member* ke situs web penyedia layanan periklanan online. Salah satu contoh penyedia layanan tersebut adalah www.iklanbaris.co.id. *Advertiser* diminta untuk memasukkan data-data seperti nama, *password*, *email*, *website* dan sebagainya. Setelah proses registrasi berhasil dilakukan, maka *advertiser* dapat memasang iklan pada website tersebut.
2. *Advertiser* membuat iklan serta memilih kategori yang sesuai. Setelah semua data lengkap kemudian iklan *diposting* dan di simpan ke *database server*.
3. Setelah iklan *terposting* maka secara otomatis iklan akan langsung ditayangkan pada halaman web situs tersebut agar dapat dilihat oleh pengunjung (*visitor*).

Untuk lebih jelasnya, alur kerja sistem yang sedang berjalan saat ini akan diilustrasikan melalui gambar 4.1 berikut.



Gambar 4.1 Ilustrasi kerja sistem yang berjalan

4.1.2. Sistem yang Akan Dibangun

Sistem baru yang akan dibangun merupakan suatu pengembangan dari cara kerja sistem yang berjalan. Jika pada sistem yang sedang berjalan hanya memiliki satu entitas yaitu *advertiser* yang berperan sebagai sumber data (iklan), maka pada sistem yang akan dibangun ini menggunakan dua entitas utama yaitu *Advertiser* dan *Publisher* yang memiliki keterkaitan satu sama lain dalam penerapan *web service* ini. Oleh sebab itu, analisa akan dibagi kedalam dua bagian sebagai berikut :

4.1.2.1. Analisa pada *Advertiser*

Model kerja yang dilakukan di sisi *advertiser* dapat dijelaskan sebagai berikut :

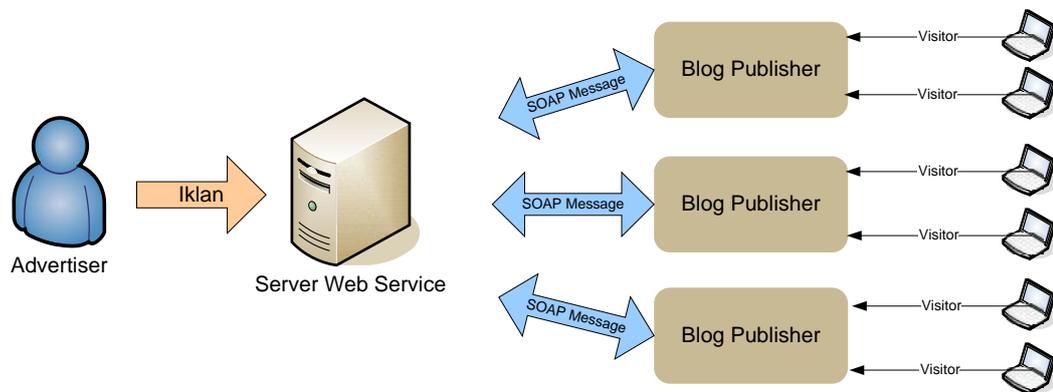
1. *Advertiser* harus membuat *account* sebagai “*advertiser*” dengan memasukkan data diri terlebih dahulu sebelum dapat memasang iklannya ke sistem.

2. *Advertiser* membuat iklan serta memilih kategori yang sesuai untuk iklannya dan memposting ke sistem, kemudian sistem akan menyimpan ke *database*. Pada proses ini, iklan tidak akan ditayangkan di *webpage* sistem, karena iklan akan tayang pada website milik *publisher* yang sebelumnya sudah terdaftar pada sistem.
3. Saat iklan sudah berhasil ditayangkan pada website *publisher* maka setiap iklan ditayangkan atau terdapat *visitor* dari web tersebut yang melakukan klik, semua *record* akan tercatat di *database* dan *advertiser* dapat melihat laporan tersebut pada *account-nya*.

4.1.2.2. Analisa pada *Publisher*

Model kerja yang dilakukan di sisi *publisher* dapat dijelaskan sebagai berikut :

1. *Publisher* harus membuat *account* sebagai “*publisher*” terlebih dahulu dengan memasukkan data diri.
2. *Publisher* harus membuat *channel* / mendaftarkan website yang akan dipasang iklan. *Publisher* dapat memasang lebih dari satu *channel* dalam satu buah *account*.
3. *Publisher* akan mengambil kode/*script* yang digenerate oleh *server*, kemudian kode tersebut akan ditempatkan pada web/*blog* milik *publisher* yang berfungsi untuk menampilkan iklan-iklan dari *advertiser*. Kode tersebut nantinya akan melakukan *request* ke *server*, kemudian *server* akan mengirimkan respon berisi data iklan yang akan ditayangkan. Proses tersebut menggunakan *SOAP Message*.



Gambar 4.2 Sistem yang akan dibangun

4.1.2.3. Analisa kebutuhan

Analisa kebutuhan dari aplikasi yang akan dibangun adalah sebagai berikut :

1. Sistem dapat menayangkan iklan yang tersebar secara *random* ke *website Publisher* sesuai pemilihan kategori yang sama.
2. Sistem dapat mencatat seluruh *history* klik yang terjadi pada iklan, *history* tersebut antara lain :
 - Jumlah klik iklan
 - Jumlah *impression* iklan/ berapa kali iklan tersebut tampil.
 - *Internet Protocol* (IP) yang tercatat setiap terjadi klik.
 - Jenis *browser*
 - Waktu

Berdasarkan data tersebut *advertiser* juga dapat melihat rasio perbandingan antara jumlah klik dengan jumlah *impression* (*Clickthrough*)

3. Sistem membatasi setiap klik pada iklan berdasarkan IP. Jika satu IP pernah melakukan klik, maka klik selanjutnya dari IP tersebut akan dikatakan *valid* jika memiliki jarak satu hari dari klik yang terekam sebelumnya.

4. Sistem dapat menampilkan dua jenis iklan yaitu iklan teks dan iklan gambar. Bentuk tampilan iklan terdiri atas tiga jenis yaitu Horizontal, Vertikal dan Persegi yang memiliki ukuran berbeda. Detail ukuran dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4.1 Jenis dan Ukuran iklan

No	Nama	Ukuran (dalam <i>pixel</i>)
1	Horizontal	728 x 90
		450 x 90
		234 x 100
2	Vertikal	120 x 600
		160 x 600
		120 x 250
3	Persegi	250 x 250
		200 x 200
		180 x 150
		125 x 125

5. Sistem dapat menampilkan laporan dari hasil klik kepada *advertiser* maupun *publisher*.
6. *Web service* yang disediakan aplikasi bisa menyediakan fungsi-fungsi yang dibutuhkan seperti pada proses *query* untuk menampilkan data iklan yang relevan.

4.1.2.4. Analisa Data

Beberapa data yang dibutuhkan untuk memulai pembuatan sistem adalah sebagai berikut :

1. *Data Member*

Data member berisi informasi *member* yang terbagi kedalam 2 kategori, yaitu *Advertiser* dan *Publisher*.

2. *Data Iklan*

Data iklan berisi tentang informasi iklan yang dipasang oleh *advertiser* ke sistem.

3. Data Channel

Data *channel* menjelaskan URL atau alamat web yang akan digunakan oleh *publisher* untuk menampilkan iklan.

4.1.3. Identifikasi Web Service pada sistem

Berdasarkan hasil analisis terhadap sistem yang akan dibangun maka selanjutnya perlu mengidentifikasi *service-service* yang akan digunakan oleh sistem. Hasil identifikasi *web service* yang akan diimplementasikan dapat dilihat pada tabel 4.2.

Tabel 4.2 Identifikasi Web Service

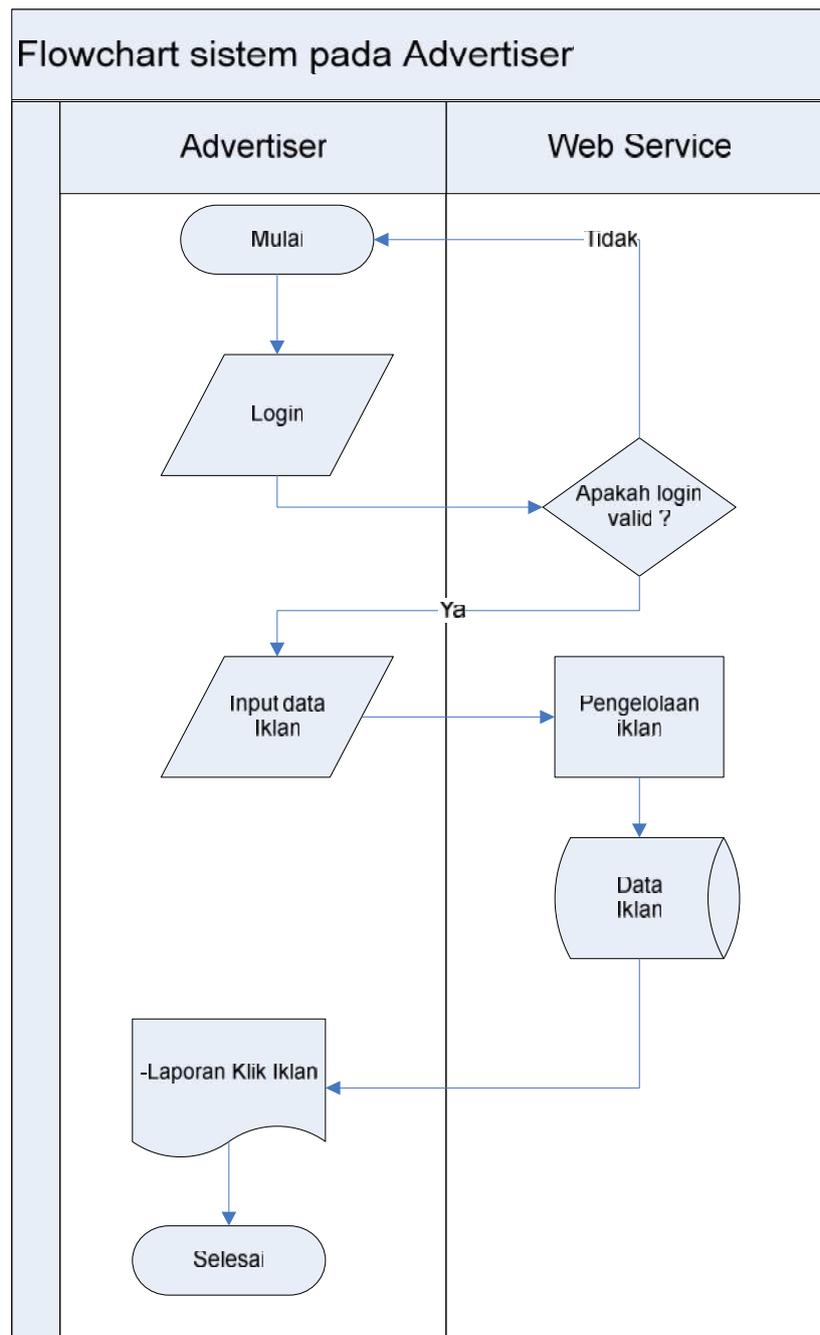
No	Nama Service	Deskripsi
1.	<i>Validate Pub-ID</i>	<i>Service</i> ini berfungsi untuk melakukan validasi terhadap ID milik <i>publisher</i> saat akan mengakses <i>web service</i> . <i>Client</i> akan mengirimkan inputan berupa <i>Publisher ID</i> ke <i>server</i> kemudian <i>output</i> yang dihasilkan berupa nilai <i>Boolean</i> untuk mengetahui apakah ID tersebut terdaftar atau tidak.
2.	<i>Validate Channel</i>	<i>Service</i> ini berfungsi untuk melakukan validasi terhadap <i>channel</i> yang telah di- <i>input</i> oleh <i>publisher</i> pada sistem. Parameter input berupa <i>Publisher ID</i> . <i>Output</i> yang dihasilkan adalah nilai <i>Boolean</i> untuk mengetahui apakah <i>channel</i> sudah terdaftar.
4.	<i>Get Ads</i>	<i>Service</i> ini berfungsi untuk mendapatkan dan mengirimkan data iklan yang dipasang oleh <i>advertiser</i> . Parameter <i>input</i> berupa <i>publisher ID</i> . <i>Output</i> yang dihasilkan adalah iklan.

4.1.4. Flowchart Sistem

Flowchart Sistem merupakan suatu cara yang digunakan untuk menggambarkan langkah-langkah kerja dari sistem yang akan dibangun.

4.1.4.1. Flowchart Pada Advertiser

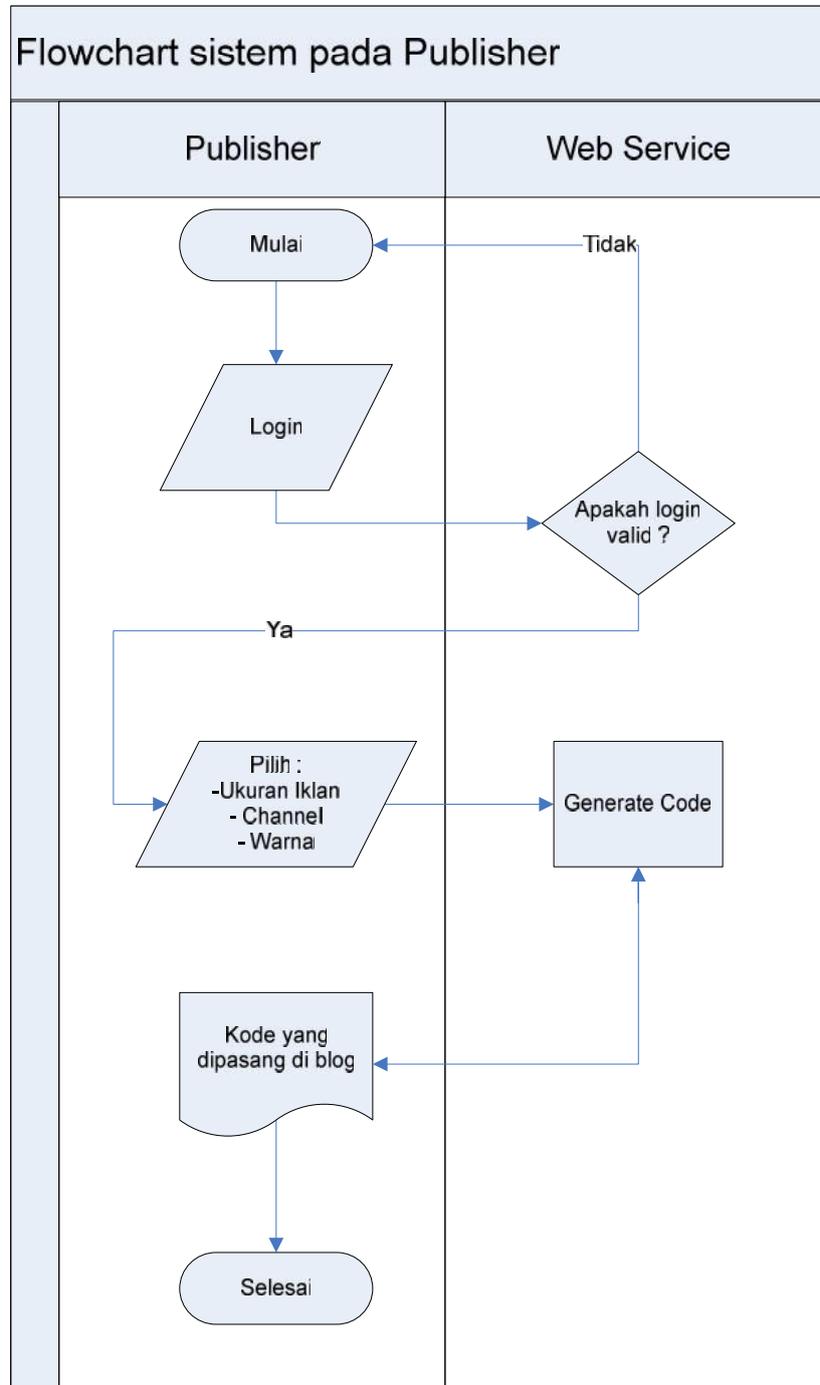
Berikut ini adalah *flowchart* sistem pada sisi *advertiser*.



Gambar 4.3 *Flowchart* pada Advertiser

4.1.4.2. Flowchart Pada Publisher

Berikut ini adalah *flowchart* sistem pada sisi *publisher*



Gambar 4.4 *Flowchart* pada *Publisher*

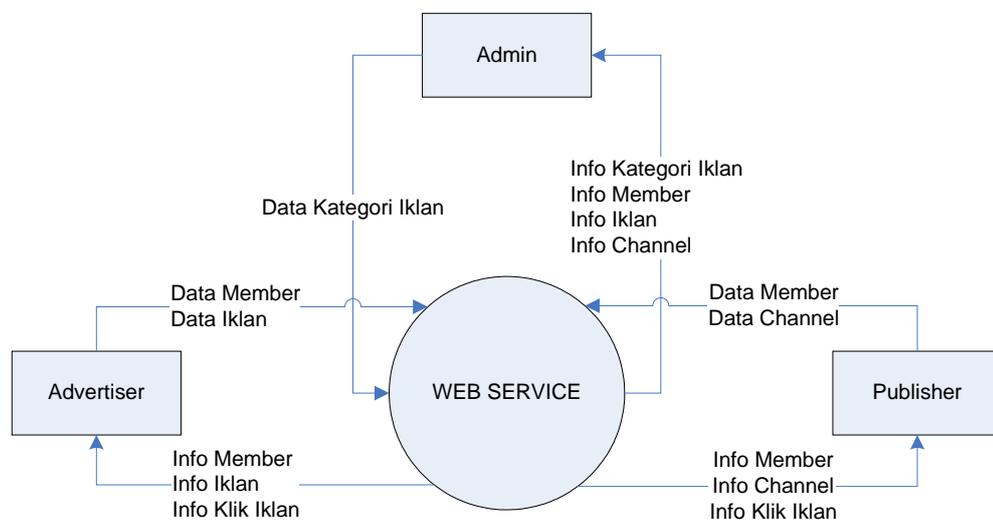
4.1.5. Analisa Fungsional Sistem

Analisa fungsional digunakan untuk menggambarkan bagaimana masukan diproses oleh aplikasi menjadi keluaran yang diharapkan oleh pengguna sistem. Analisa fungsional memuat beberapa diagram alir data yang memperlihatkan aliran data dari luar sistem yang diproses oleh sistem.(kadang melibatkan penyimpanan sistem) kemudian menghasilkan keluaran yang berguna.

Secara garis besar fungsi utama dari perangkat lunak yang akan dikembangkan dapat dilihat di Diagram Konteks (*Context Diagram*) dan diagram aliran data (*Data Flow Diagram*) yang akan diberikan. Seperti berikut ini:

4.1.5.1. Context Diagram

Context Diagram digunakan untuk menggambarkan proses kerja sistem secara umum. *Context Diagram* adalah *Data Flow Diagram* (DFD) yang menggambarkan garis besar operasional sistem.



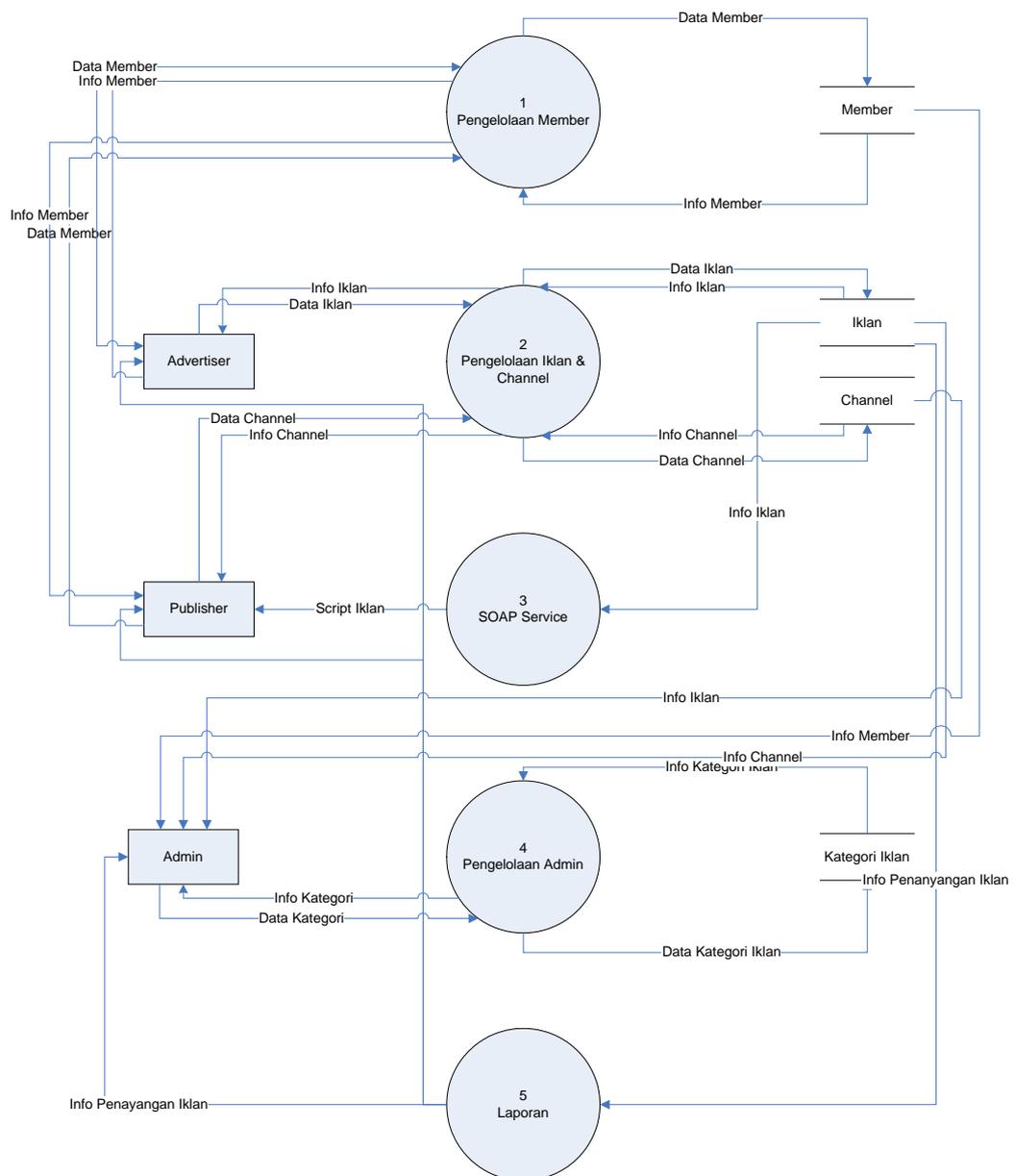
Gambar 4.5 Context Diagram

Advertiser dalam sistem berfungsi sebagai orang yang memiliki iklan dan akan memasang iklannya pada sistem, sedangkan *publisher* disini berfungsi sebagai media tempat menampilkan iklan dari *advertiser* di blog/website mereka.

4.1.5.2. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) sering digunakan untuk menggunakan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir, atau lingkungan fisik dimana data tersebut tersimpan.

4.1.5.3. DFD Level 1 WEB SERVICE



Gambar 4.6 DFD Level 1 *Web service*

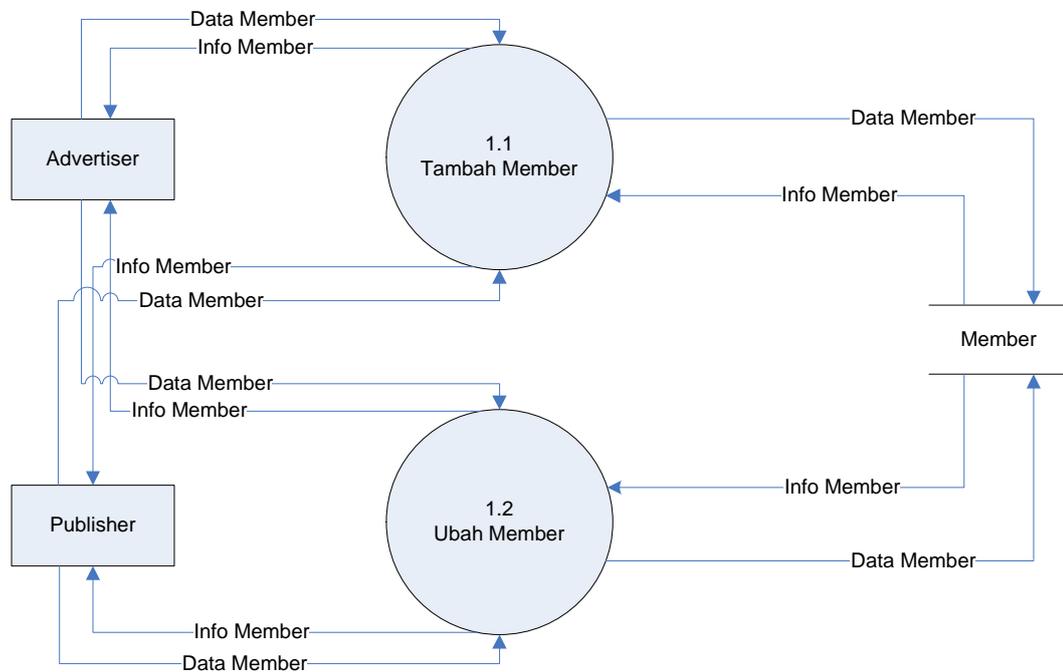
Tabel 4.3 DFD Level 1 *Web service*

No.	Nama Proses	Deskripsi
1.	<i>Member</i>	Proses registrasi yang dilakukan oleh <i>member</i> yaitu <i>advertiser</i> dan <i>publisher</i>
2.	Pengelolaan iklan & <i>channel</i>	Proses dalam pengelolaan iklan oleh <i>member</i> seperti menambah iklan, menghapus iklan
3.	SOAP <i>service</i>	Proses generate code oleh <i>publisher</i> untuk di pasang di situs web mereka.
4.	Pengelolaan <i>Admin</i>	Proses yang terjadi pada halaman administrator
5.	Laporan	Proses untuk melihat laporan hasil penayangan iklan

Tabel 4.4 Keterangan Aliran Data DFD Level 1 *Web service*

Nama Aliran Data	Deskripsi
Data_ <i>Member</i>	Data dari <i>advertiser</i> dan <i>publisher</i>
Info_ <i>Member</i>	Informasi data dari <i>member</i>
Data_ <i>Iklan</i>	Data iklan dari <i>advertiser</i>
Info_ <i>iklan</i>	Informasi iklan dari datastore iklan
Data_ <i>channel</i>	Data <i>channel</i> dari <i>publisher</i>
Info_ <i>channel</i>	Informasi <i>channel</i> dari datastore <i>channel</i>
Data_ <i>kategori</i>	Data kategori dari <i>admin</i>
Info_ <i>kategori</i>	Informasi kategori dari datastore kategori iklan
Info_ <i>penayangan_iklan</i>	Informasi hasil klik iklan secara detail

4.1.5.4. DFD Level 2 Proses *Member*



Gambar 4.7 DFD Level 2 Proses *Member*

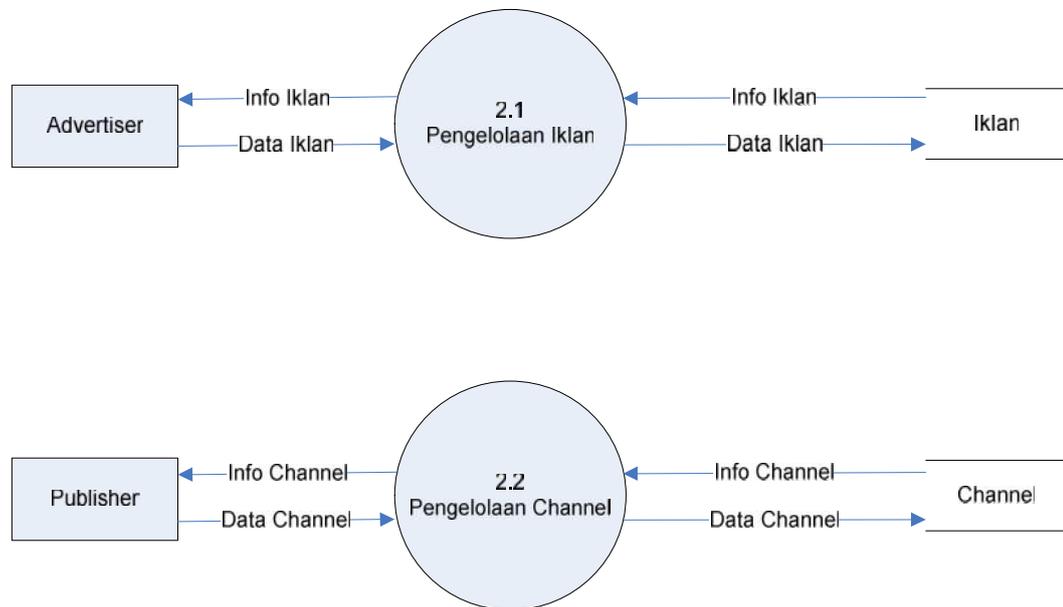
Tabel 4.5 DFD Level 2 Proses Registrasi

No.	Nama Proses	Deskripsi
1.	Tambah <i>member</i>	Proses penambahan <i>member</i> baru
2.	Ubah <i>member</i>	Proses untuk mengubah data <i>member</i>

Tabel 4.6 Keterangan Aliran Data DFD Level 2 Proses Registrasi

Nama Aliran Data	Deskripsi
<i>Data_Member</i>	Data <i>member</i> dari <i>advertiser</i> dan <i>publisher</i>
<i>Info_Member</i>	Informasi data dari <i>member</i>

4.1.5.5. DFD Level 2 Proses Pengelolaan iklan & Channel



Gambar 4.8 DFD Level 2 Proses Pengelolaan iklan & channel

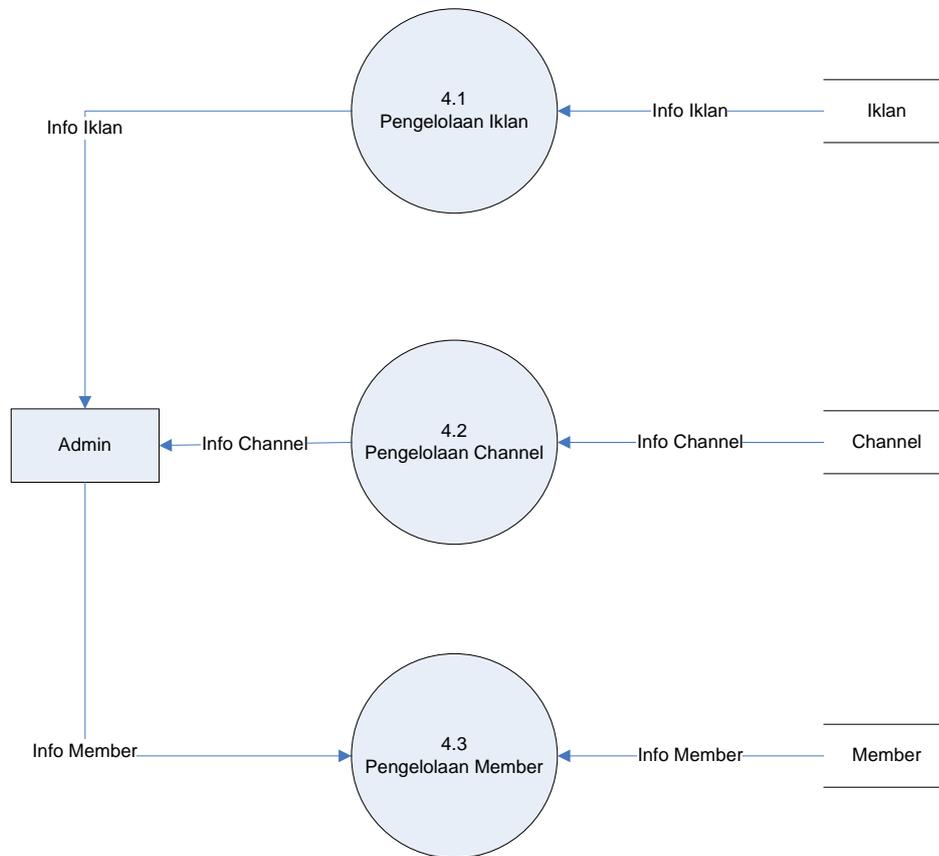
Tabel 4.7 DFD Level 2 Proses Pengelolaan iklan & channel

No.	Nama Proses	Deskripsi
1.	Pengelolaan Iklan	Proses untuk Pengelolaan iklan oleh <i>advertiser</i>
2.	Pengelolaan <i>channel</i>	Proses untuk Pengelolaan <i>channel</i> oleh <i>publisher</i>

Tabel 4.8 Keterangan Aliran Data DFD Level 2 Proses iklan & channel

Nama Aliran Data	Deskripsi
Data_Iklan	Data iklan dari <i>advertiser</i>
Info_Iklan	Informasi data iklan
Data_Channel	Data <i>channel</i> dari <i>publisher</i>
Info_Channel	Informasi data <i>channel</i>

4.1.5.6. DFD Level 2 Proses Pengelolaan *admin*



Gambar 4.9 DFD Level 2 Proses Pengelolaan *admin*

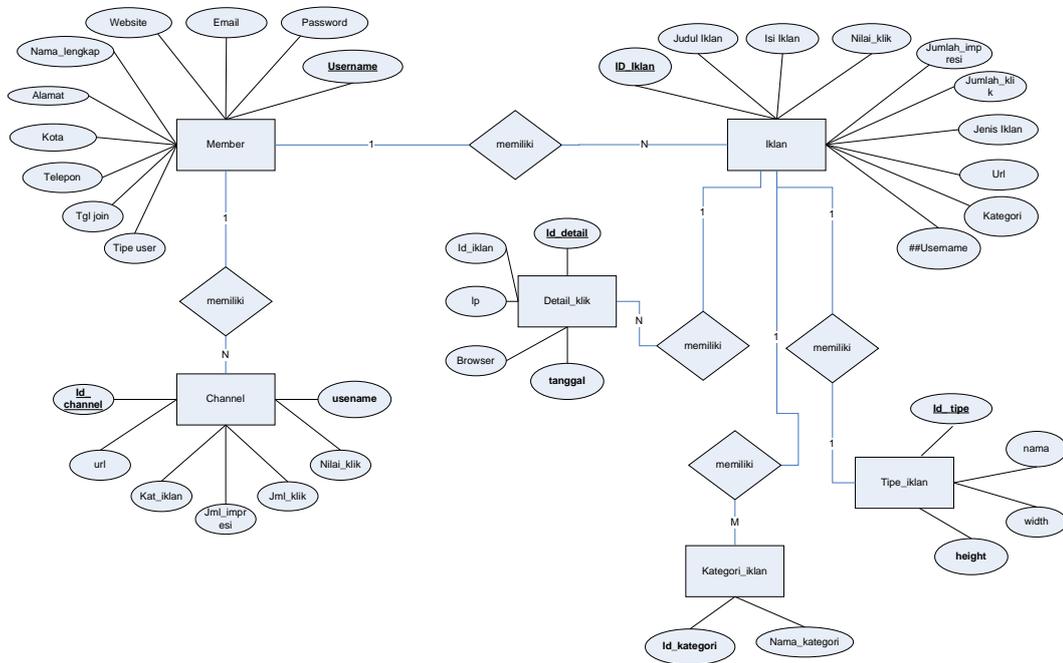
Tabel 4.9 DFD Level 2 Proses Pengelolaan *Admin*

No.	Nama Proses	Deskripsi
1.	Pengelolaan Iklan	Proses untuk Pengelolaan iklan oleh <i>advertiser</i>
2.	Pengelolaan <i>channel</i>	Proses untuk Pengelolaan <i>channel</i> oleh <i>publisher</i>
3.	Pengelolaan <i>member</i>	Proses untuk pengelolaan <i>member</i>

Tabel 4.10 Keterangan Aliran Data DFD Level 2 Proses *Admin*

Nama Aliran Data	Deskripsi
Info_Iklan	Informasi data iklan
Info_Channel	Informasi data <i>channel</i>
Info_Member	Informasi data <i>member</i>

4.1.6. Entity Relationship Diagram (ER-Diagram)



Gambar 4.10 ER-Diagram

Tabel 4.11 Keterangan entitas pada ERD

Nama	Deskripsi	Atribut	Primary Key
<i>Member</i>	Menyimpan data <i>member</i>	✓ Username ✓ Password ✓ Email ✓ Website ✓ Nama_lengkap ✓ Alamat ✓ Kota ✓ Telepon ✓ Tgl_join ✓ Tipe_user	Username
<i>Channel</i>	Menyimpan data <i>channel</i>	✓ ID_channel ✓ Url	ID_Channel

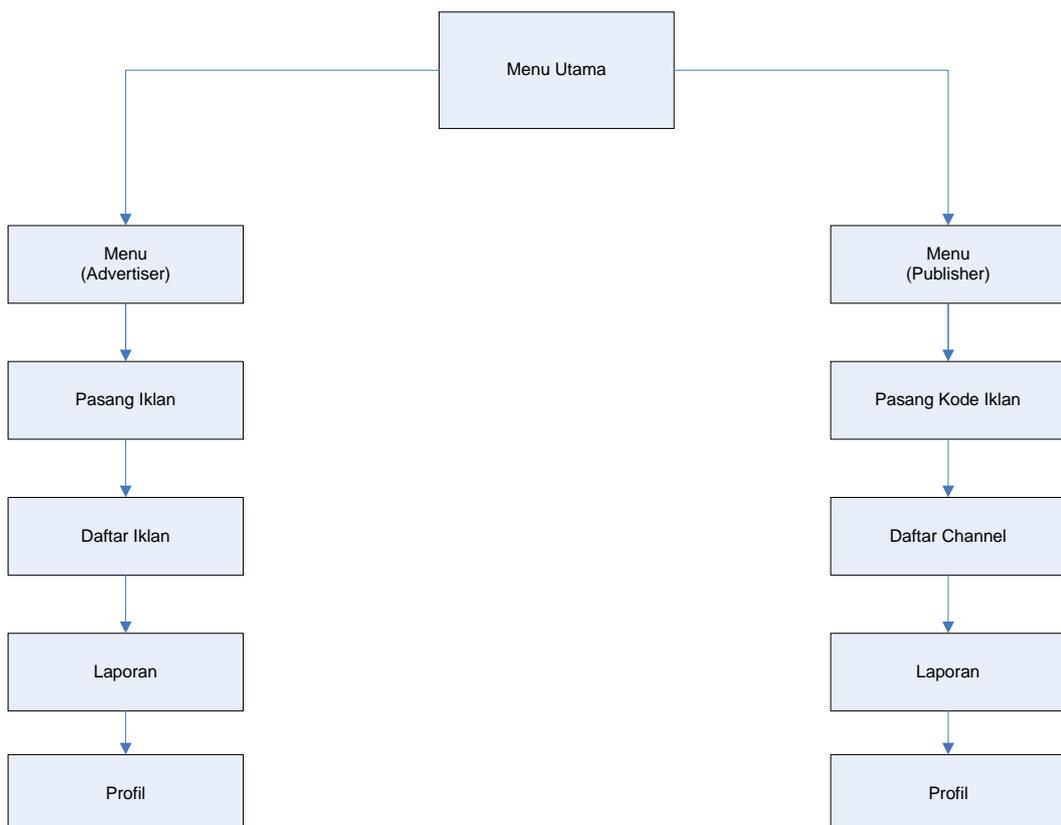
		<ul style="list-style-type: none"> ✓ kat_iklan ✓ jml_impresi ✓ jml_klik ✓ nilai_klik ✓ username 	
Iklan	Menyimpan data iklan	<ul style="list-style-type: none"> ✓ ID_iklan ✓ Judul ✓ Isi_iklan ✓ url ✓ idtipe ✓ lokasi_image ✓ kat_iklan ✓ deposit ✓ nilai_klik ✓ jml_impresi ✓ jenis_iklan ✓ username 	ID_iklan
Detail klik	Menyimpan data detail klik	<ul style="list-style-type: none"> ✓ id_detail ✓ id_iklan ✓ IP ✓ Browser ✓ tanggal 	id_detailklik
Tipe_iklan	Menyimpan data tipe iklan	<ul style="list-style-type: none"> ✓ id_tipe ✓ nama ✓ width ✓ height 	id_tipe
Kategori_iklan	Menyimpan data kategori iklan	<ul style="list-style-type: none"> ✓ id_kategori ✓ nama_kategori 	Id_kategori

4.2. Perancangan

Dalam tahap perancangan akan terbagi menjadi beberapa bagian pembahasan yaitu perancangan struktur menu, basis data serta perancangan antar muka yang akan dijelaskan sebagai berikut.

4.2.1. Perancangan Menu

Dalam pemakaian sistem ini diperlukan susunan daftar pilihan/menu sehingga pengguna yang belum terbiasa dengan sistem juga dapat menggunakan sistem ini. Pengguna akan dihadapkan pada berbagai alternatif menu yang ada. Dalam menentukan pilihannya, pengguna dapat menggunakan tombol tertentu dan setiap pilihan akan menghasilkan tampilan/halaman tertentu. Sistem yang akan dibangun memiliki menu dan sub-sub menu yang digambarkan pada bagan di bawah ini:



Gambar 4.11 Rancangan Menu

4.2.2. Perancangan Basis Data

Tahapan perancangan basis data digunakan untuk membuat detail data yang akan dipersiapkan pada tahap implementasi selanjutnya. Berikut adalah hasil perancangan basis data :

Tabel 4.12 Perancangan Tabel *Member*

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>
1	Username	Varchar(10)	NOT NULL
2	Password	Varchar(20)	NOT NULL
3	Email	Varchar(50)	NOT NULL
4	Website	Varchar(50)	NOT NULL
5	Nama Lengkap	Varchar(30)	NOT NULL
6	Alamat	Varchar(100)	NOT NULL
7	Kode Pos	Varchar(5)	NULL
8	Telp/hp	Varchar(15)	NULL
9	Tgl_join	Datetime	NOT NULL
10	Tipeuser	char(1)	NOT NULL

Tabel 4.13 Perancangan Tabel *Channel*

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>
1	ID_channel	Integer (5)	NOT NULL
2	Url	Varchar(20)	NOT NULL
3	kat_iklan	Varchar(50)	NOT NULL
4	jml_impresi	Integer (5)	NOT NULL
5	jml_klik	Integer (5)	NOT NULL
6	nilai_klik	Integer (5)	NOT NULL
7	username	Varchar(10)	NOT NULL

Tabel 4.14 Perancangan Tabel Iklan

No	Nama Field	Tipe Data	Null
1	ID_iklan	Integer (10)	NOT NULL
2	Judul	Varchar(25)	NOT NULL
3	Isi_iklan	Varchar(60)	NOT NULL
4	url	Varchar(50)	NOT NULL
5	Id_tipe	Integer(5)	NOT NULL
6	lokasi_image	Varchar(50)	NOT NULL
7	kat_iklan	Varchar(50)	NOT NULL
8	deposit	Integer(5)	NOT NULL
9	nilai_klik	Integer(5)	NOT NULL
10	jml_impresi	Integer(5)	NOT NULL
11	jenis_iklan	Char(1)	NOT NULL
12	username	Varchar(10)	NOT NULL

Tabel 4.15 Perancangan Tabel detail_klik

No	Nama Field	Tipe Data	Null
1	id_detail	Integer (10)	NOT NULL
2	id_iklan	Integer (10)	NOT NULL
3	IP	Varchar(20)	NOT NULL
4	Browser	Varchar(50)	NOT NULL
5	Tanggal	Datetime	NOT NULL

Tabel 4.16 Perancangan Tabel kategori_iklan

No	Nama Field	Tipe Data	Null
1	id_kategori	Integer (10)	NOT NULL
2	nama_kategori	varchar (30)	NOT NULL

4.2.3. Perancangan Antar Muka (*Interface*)

Interface sistem merupakan suatu sarana pengembangan sistem yang ditujukan untuk mempermudah pemakai berkomunikasi dengan sistem yang ada dan konsisten data juga ditunjukkan dalam *interface* tersebut. Penekanan *interface* meliputi tampilan yang baik, mudah dipahami dan tombol-tombol yang familiar.

The wireframe illustrates the main page layout. At the top is a large rectangular area labeled "Banner". Below it is a horizontal navigation bar containing five buttons: "Home", "Advertiser", "Publisher", "Layanan", and "Kontak Kami". The main content area is divided into three sections. On the left, there is a box titled "Advertiser" with a "Login" button below it. In the center, there is a box titled "Publisher" with a "Login" button below it. On the right, there is a login form with two input fields labeled "Username" and "Password", and a "Login" button below them.

Gambar 4.12 Rancangan Halaman Utama

Penjelasan lebih rinci mengenai perancangan antarmuka lainnya dapat dilihat pada lampiran A.

BAB IV

ANALISA DAN PERANCANGAN

Bab ini akan membahas analisa dan perancangan yang merupakan dua tahapan yang saling berhubungan erat. Analisa bertujuan untuk mengidentifikasi permasalahan dan kebutuhan sistem, sementara perancangan bertujuan untuk mengimplementasikan permasalahan yang telah diidentifikasi kedalam sebuah aplikasi.

4.1. Analisa Sistem

Analisa merupakan suatu aktivitas kreatif dalam proses mentransformasikan suatu definisi permasalahan dari sebuah ketidakjelasan, meliputi poses memahami permasalahan, mengetahui keterbatasan dan menggagas metode yang tepat untuk mengatasi permasalahan serta keterbatasan tersebut

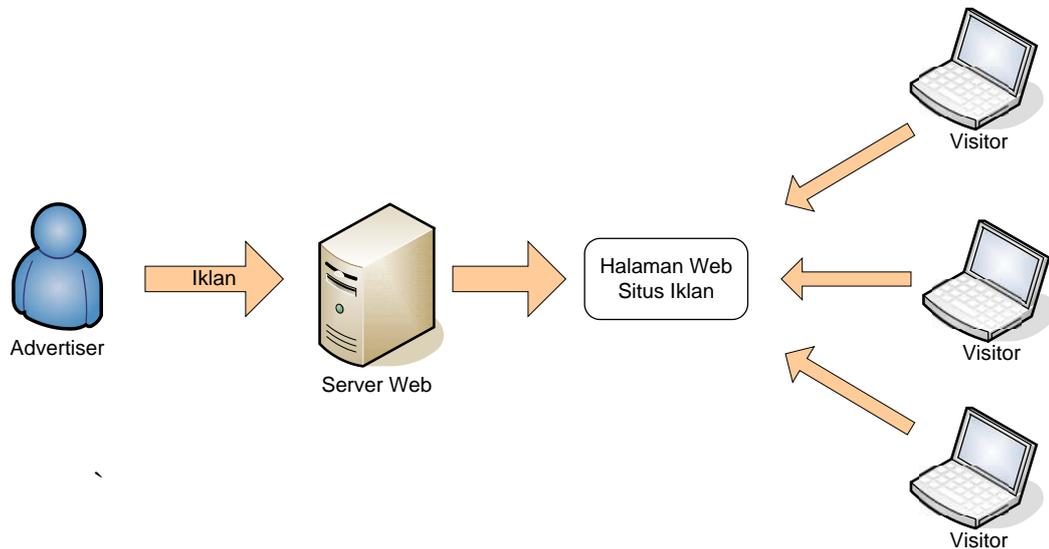
Analisa sistem dibagi dalam dua bagian pembahasan yaitu analisa pada sistem yang berjalan dan analisa pada sistem yang akan dibangun.

4.1.1. Sistem yang Berjalan

Secara umum sistem yang berjalan memiliki alur proses kerja sebagai berikut :

1. *Advertiser* melakukan registrasi untuk pendaftaran sebagai *member* ke situs web penyedia layanan periklanan online. Salah satu contoh penyedia layanan tersebut adalah www.iklanbaris.co.id. *Advertiser* diminta untuk memasukkan data-data seperti nama, *password*, *email*, *website* dan sebagainya. Setelah proses registrasi berhasil dilakukan, maka *advertiser* dapat memasang iklan pada website tersebut.
2. *Advertiser* membuat iklan serta memilih kategori yang sesuai. Setelah semua data lengkap kemudian iklan *diposting* dan di simpan ke *database server*.
3. Setelah iklan *terposting* maka secara otomatis iklan akan langsung ditayangkan pada halaman web situs tersebut agar dapat dilihat oleh pengunjung (*visitor*).

Untuk lebih jelasnya, alur kerja sistem yang sedang berjalan saat ini akan diilustrasikan melalui gambar 4.1 berikut.



Gambar 4.1 Ilustrasi kerja sistem yang berjalan

4.1.2. Sistem yang Akan Dibangun

Sistem baru yang akan dibangun merupakan suatu pengembangan dari cara kerja sistem yang berjalan. Jika pada sistem yang sedang berjalan hanya memiliki satu entitas yaitu *advertiser* yang berperan sebagai sumber data (iklan), maka pada sistem yang akan dibangun ini menggunakan dua entitas utama yaitu *Advertiser* dan *Publisher* yang memiliki keterkaitan satu sama lain dalam penerapan *web service* ini. Oleh sebab itu, analisa akan dibagi kedalam dua bagian sebagai berikut :

4.1.2.1. Analisa pada *Advertiser*

Model kerja yang dilakukan di sisi *advertiser* dapat dijelaskan sebagai berikut :

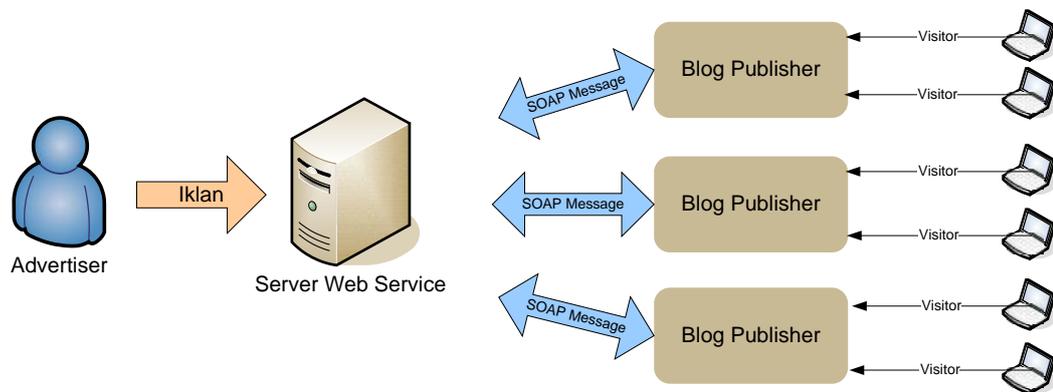
1. *Advertiser* harus membuat *account* sebagai “*advertiser*” dengan memasukkan data diri terlebih dahulu sebelum dapat memasang iklannya ke sistem.

2. *Advertiser* membuat iklan serta memilih kategori yang sesuai untuk iklannya dan memposting ke sistem, kemudian sistem akan menyimpan ke *database*. Pada proses ini, iklan tidak akan ditayangkan di *webpage* sistem, karena iklan akan tayang pada website milik *publisher* yang sebelumnya sudah terdaftar pada sistem.
3. Saat iklan sudah berhasil ditayangkan pada website *publisher* maka setiap iklan ditayangkan atau terdapat *visitor* dari web tersebut yang melakukan klik, semua *record* akan tercatat di *database* dan *advertiser* dapat melihat laporan tersebut pada *account-nya*.

4.1.2.2. Analisa pada *Publisher*

Model kerja yang dilakukan di sisi *publisher* dapat dijelaskan sebagai berikut :

1. *Publisher* harus membuat *account* sebagai “*publisher*” terlebih dahulu dengan memasukkan data diri.
2. *Publisher* harus membuat *channel* / mendaftarkan website yang akan dipasang iklan. *Publisher* dapat memasang lebih dari satu *channel* dalam satu buah *account*.
3. *Publisher* akan mengambil kode/*script* yang digenerate oleh *server*, kemudian kode tersebut akan ditempatkan pada web/*blog* milik *publisher* yang berfungsi untuk menampilkan iklan-iklan dari *advertiser*. Kode tersebut nantinya akan melakukan *request* ke *server*, kemudian *server* akan mengirimkan respon berisi data iklan yang akan ditayangkan. Proses tersebut menggunakan *SOAP Message*.



Gambar 4.2 Sistem yang akan dibangun

4.1.2.3. Analisa kebutuhan

Analisa kebutuhan dari aplikasi yang akan dibangun adalah sebagai berikut :

1. Sistem dapat menayangkan iklan yang tersebar secara *random* ke *website Publisher* sesuai pemilihan kategori yang sama.
2. Sistem dapat mencatat seluruh *history* klik yang terjadi pada iklan, *history* tersebut antara lain :
 - Jumlah klik iklan
 - Jumlah *impression* iklan/ berapa kali iklan tersebut tampil.
 - *Internet Protocol* (IP) yang tercatat setiap terjadi klik.
 - Jenis *browser*
 - Waktu

Berdasarkan data tersebut *advertiser* juga dapat melihat rasio perbandingan antara jumlah klik dengan jumlah *impression* (*Clickthrough*)

3. Sistem membatasi setiap klik pada iklan berdasarkan IP. Jika satu IP pernah melakukan klik, maka klik selanjutnya dari IP tersebut akan dikatakan *valid* jika memiliki jarak satu hari dari klik yang terekam sebelumnya.

4. Sistem dapat menampilkan dua jenis iklan yaitu iklan teks dan iklan gambar. Bentuk tampilan iklan terdiri atas tiga jenis yaitu Horizontal, Vertikal dan Persegi yang memiliki ukuran berbeda. Detail ukuran dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4.1 Jenis dan Ukuran iklan

No	Nama	Ukuran (dalam <i>pixel</i>)
1	Horizontal	728 x 90
		450 x 90
		234 x 100
2	Vertikal	120 x 600
		160 x 600
		120 x 250
3	Persegi	250 x 250
		200 x 200
		180 x 150
		125 x 125

5. Sistem dapat menampilkan laporan dari hasil klik kepada *advertiser* maupun *publisher*.
6. *Web service* yang disediakan aplikasi bisa menyediakan fungsi-fungsi yang dibutuhkan seperti pada proses *query* untuk menampilkan data iklan yang relevan.

4.1.2.4. Analisa Data

Beberapa data yang dibutuhkan untuk memulai pembuatan sistem adalah sebagai berikut :

1. *Data Member*

Data member berisi informasi *member* yang terbagi kedalam 2 kategori, yaitu *Advertiser* dan *Publisher*.

2. *Data Iklan*

Data iklan berisi tentang informasi iklan yang dipasang oleh *advertiser* ke sistem.

3. Data Channel

Data *channel* menjelaskan URL atau alamat web yang akan digunakan oleh *publisher* untuk menampilkan iklan.

4.1.3. Identifikasi Web Service pada sistem

Berdasarkan hasil analisis terhadap sistem yang akan dibangun maka selanjutnya perlu mengidentifikasi *service-service* yang akan digunakan oleh sistem. Hasil identifikasi *web service* yang akan diimplementasikan dapat dilihat pada tabel 4.2.

Tabel 4.2 Identifikasi Web Service

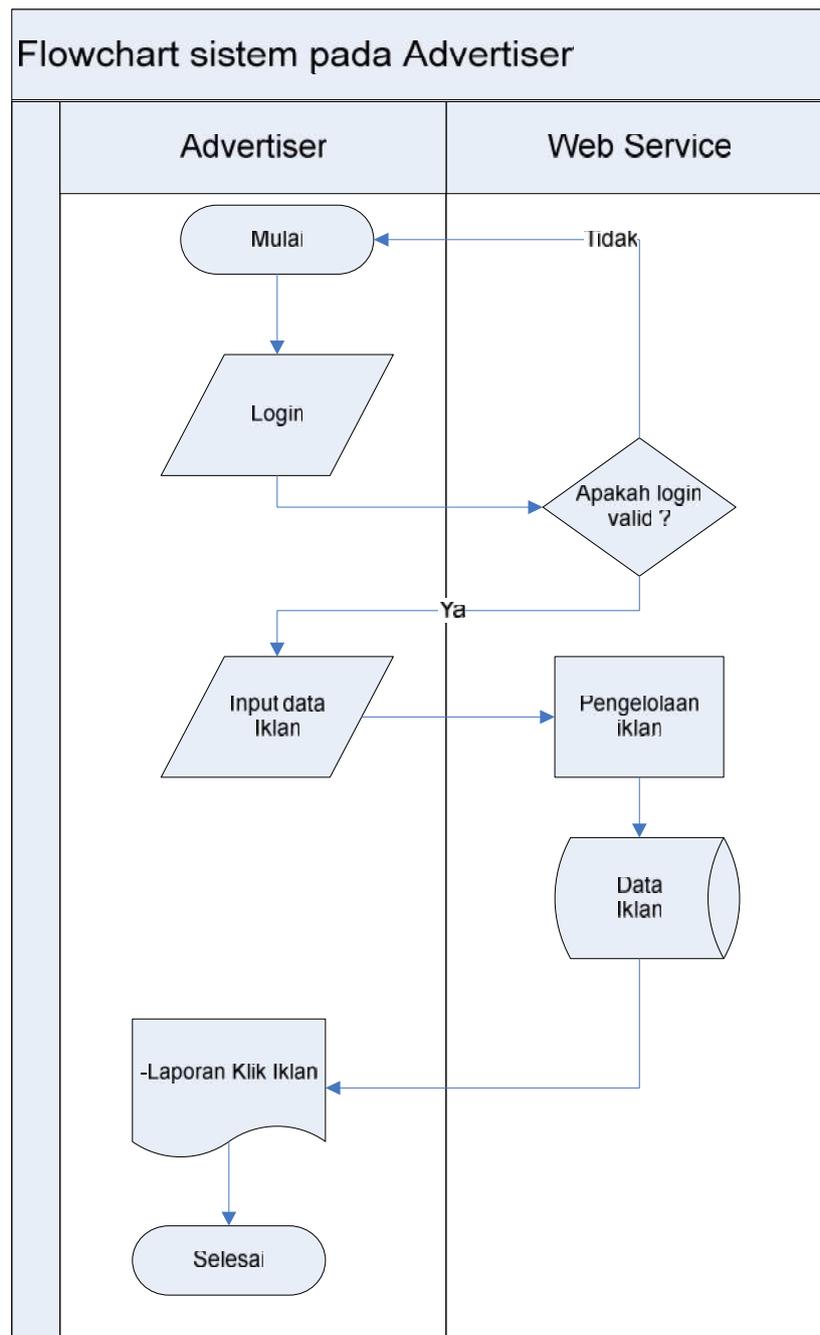
No	Nama Service	Deskripsi
1.	<i>Validate Pub-ID</i>	<i>Service</i> ini berfungsi untuk melakukan validasi terhadap ID milik <i>publisher</i> saat akan mengakses <i>web service</i> . <i>Client</i> akan mengirimkan inputan berupa <i>Publisher ID</i> ke <i>server</i> kemudian <i>output</i> yang dihasilkan berupa nilai <i>Boolean</i> untuk mengetahui apakah ID tersebut terdaftar atau tidak.
2.	<i>Validate Channel</i>	<i>Service</i> ini berfungsi untuk melakukan validasi terhadap <i>channel</i> yang telah di- <i>input</i> oleh <i>publisher</i> pada sistem. Parameter input berupa <i>Publisher ID</i> . <i>Output</i> yang dihasilkan adalah nilai <i>Boolean</i> untuk mengetahui apakah <i>channel</i> sudah terdaftar.
4.	<i>Get Ads</i>	<i>Service</i> ini berfungsi untuk mendapatkan dan mengirimkan data iklan yang dipasang oleh <i>advertiser</i> . Parameter <i>input</i> berupa <i>publisher ID</i> . <i>Output</i> yang dihasilkan adalah iklan.

4.1.4. Flowchart Sistem

Flowchart Sistem merupakan suatu cara yang digunakan untuk menggambarkan langkah-langkah kerja dari sistem yang akan dibangun.

4.1.4.1. Flowchart Pada Advertiser

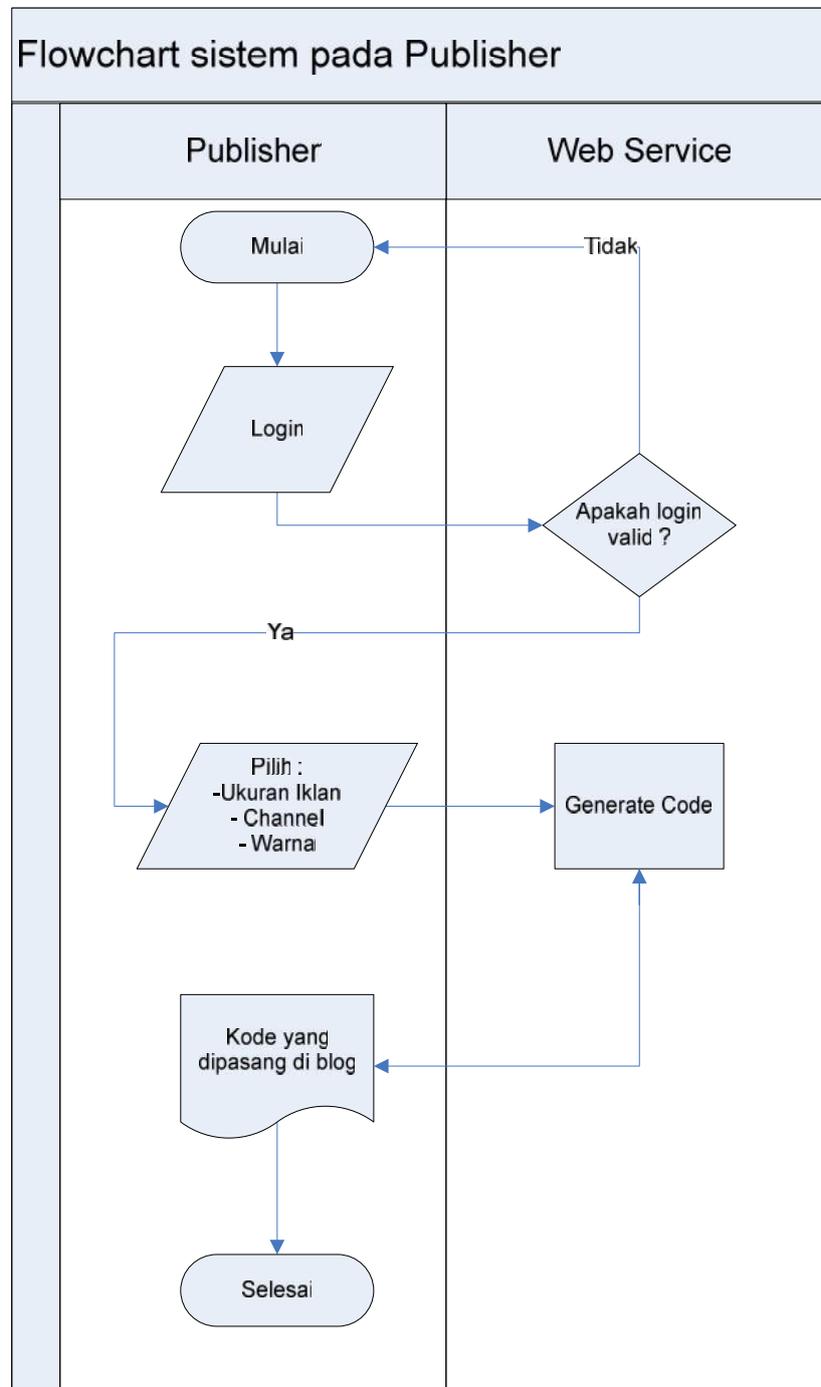
Berikut ini adalah *flowchart* sistem pada sisi *advertiser*.



Gambar 4.3 *Flowchart* pada Advertiser

4.1.4.2. Flowchart Pada Publisher

Berikut ini adalah *flowchart* sistem pada sisi *publisher*



Gambar 4.4 Flowchart pada Publisher

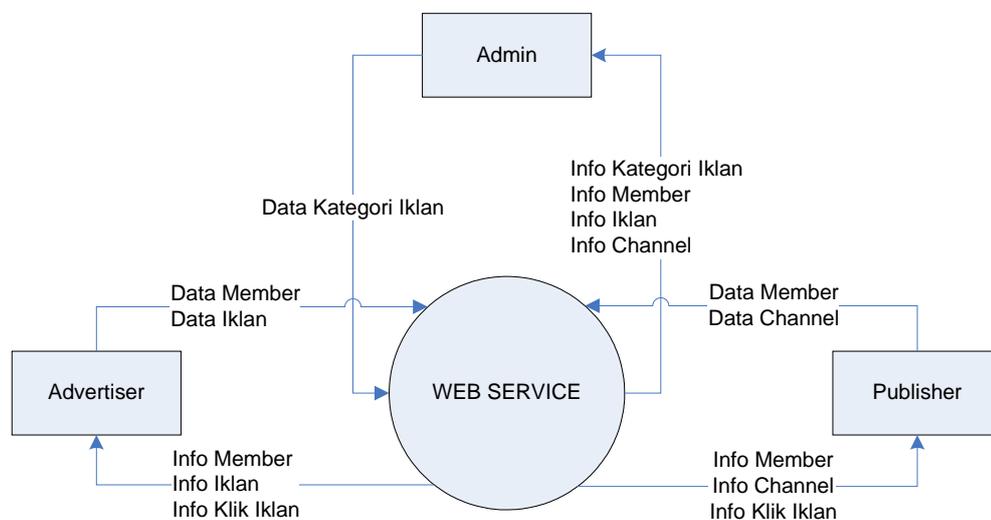
4.1.5. Analisa Fungsional Sistem

Analisa fungsional digunakan untuk menggambarkan bagaimana masukan diproses oleh aplikasi menjadi keluaran yang diharapkan oleh pengguna sistem. Analisa fungsional memuat beberapa diagram alir data yang memperlihatkan aliran data dari luar sistem yang diproses oleh sistem.(kadang melibatkan penyimpanan sistem) kemudian menghasilkan keluaran yang berguna.

Secara garis besar fungsi utama dari perangkat lunak yang akan dikembangkan dapat dilihat di Diagram Konteks (*Context Diagram*) dan diagram aliran data (*Data Flow Diagram*) yang akan diberikan. Seperti berikut ini:

4.1.5.1. Context Diagram

Context Diagram digunakan untuk menggambarkan proses kerja sistem secara umum. *Context Diagram* adalah *Data Flow Diagram* (DFD) yang menggambarkan garis besar operasional sistem.



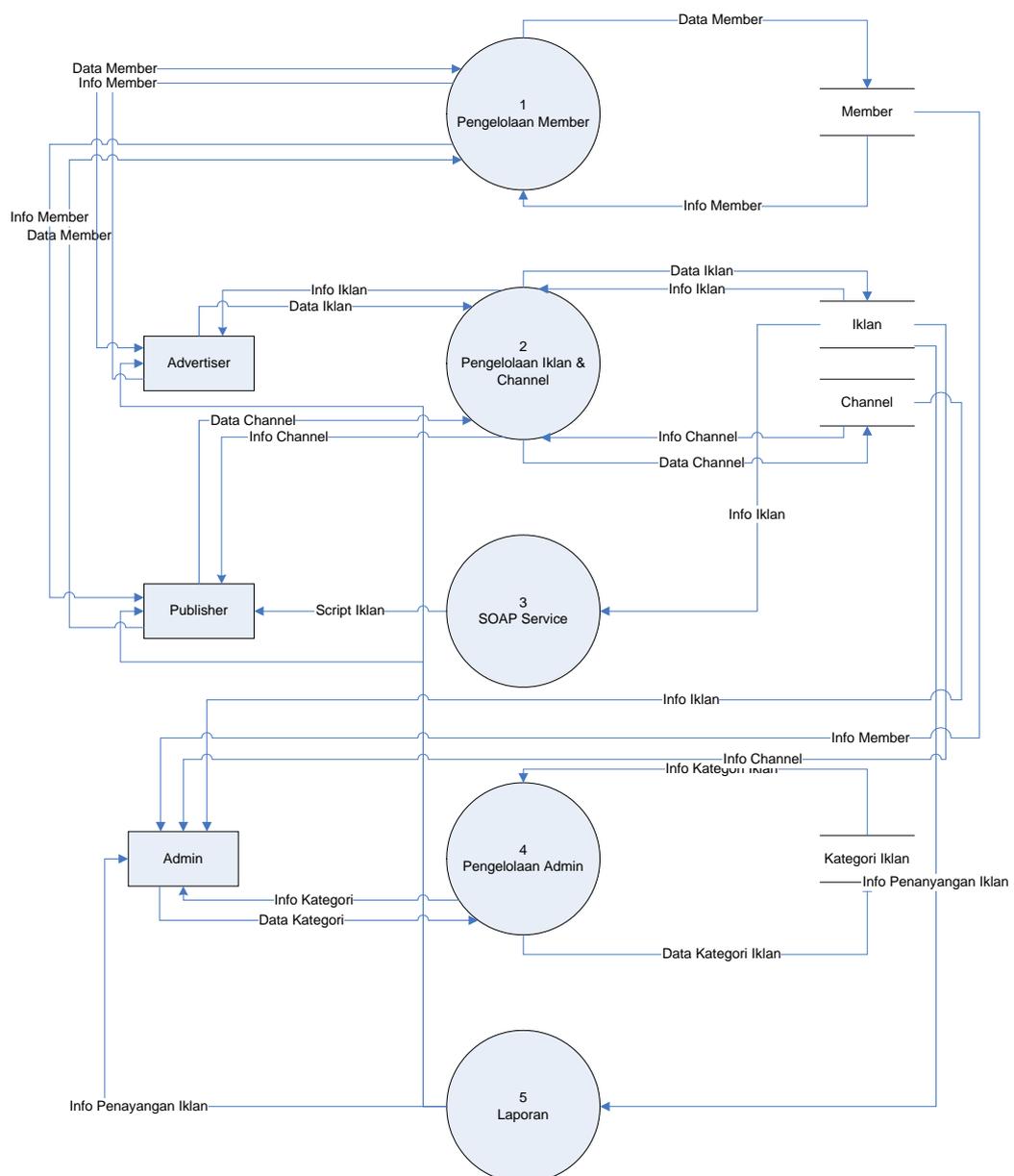
Gambar 4.5 *Context Diagram*

Advertiser dalam sistem berfungsi sebagai orang yang memiliki iklan dan akan memasang iklannya pada sistem, sedangkan *publisher* disini berfungsi sebagai media tempat menampilkan iklan dari *advertiser* di blog/website mereka.

4.1.5.2. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) sering digunakan untuk menggunakan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir, atau lingkungan fisik dimana data tersebut tersimpan.

4.1.5.3. DFD Level 1 WEB SERVICE



Gambar 4.6 DFD Level 1 *Web service*

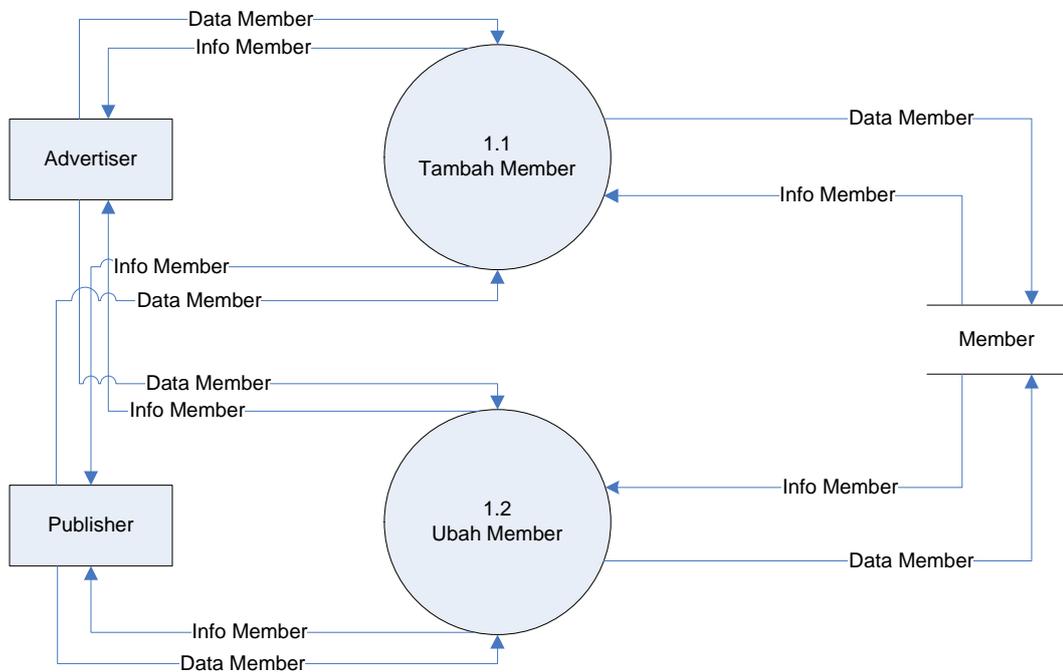
Tabel 4.3 DFD Level 1 *Web service*

No.	Nama Proses	Deskripsi
1.	<i>Member</i>	Proses registrasi yang dilakukan oleh <i>member</i> yaitu <i>advertiser</i> dan <i>publisher</i>
2.	Pengelolaan iklan & <i>channel</i>	Proses dalam pengelolaan iklan oleh <i>member</i> seperti menambah iklan, menghapus iklan
3.	SOAP <i>service</i>	Proses generate code oleh <i>publisher</i> untuk di pasang di situs web mereka.
4.	Pengelolaan <i>Admin</i>	Proses yang terjadi pada halaman administrator
5.	Laporan	Proses untuk melihat laporan hasil penayangan iklan

Tabel 4.4 Keterangan Aliran Data DFD Level 1 *Web service*

Nama Aliran Data	Deskripsi
Data_ <i>Member</i>	Data dari <i>advertiser</i> dan <i>publisher</i>
Info_ <i>Member</i>	Informasi data dari <i>member</i>
Data_ <i>Iklan</i>	Data iklan dari <i>advertiser</i>
Info_ <i>iklan</i>	Informasi iklan dari datastore iklan
Data_ <i>channel</i>	Data <i>channel</i> dari <i>publisher</i>
Info_ <i>channel</i>	Informasi <i>channel</i> dari datastore <i>channel</i>
Data_ <i>kategori</i>	Data kategori dari <i>admin</i>
Info_ <i>kategori</i>	Informasi kategori dari datastore kategori iklan
Info_ <i>penayangan_iklan</i>	Informasi hasil klik iklan secara detail

4.1.5.4. DFD Level 2 Proses *Member*



Gambar 4.7 DFD Level 2 Proses *Member*

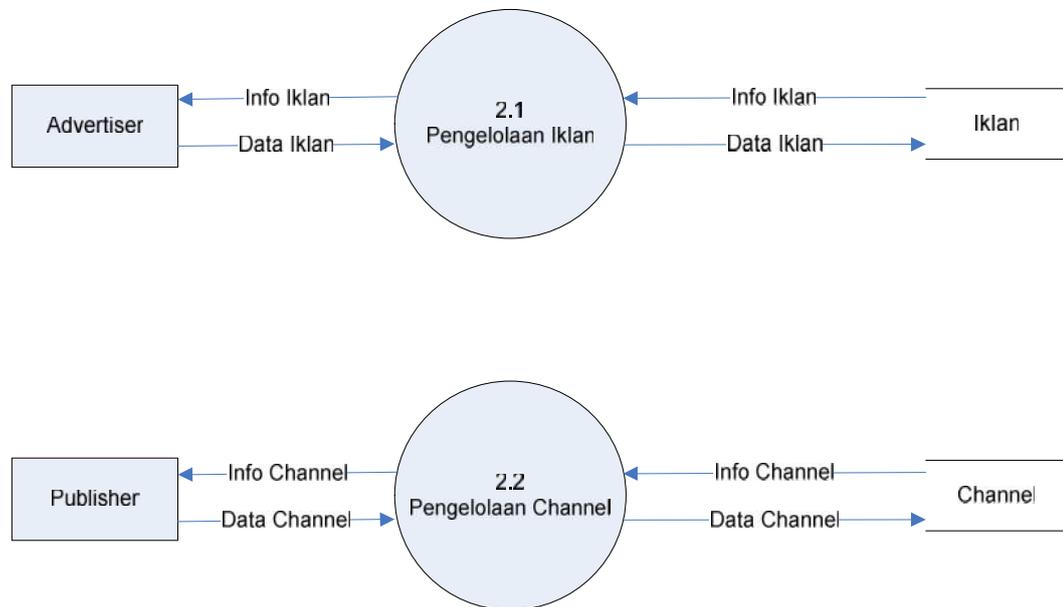
Tabel 4.5 DFD Level 2 Proses Registrasi

No.	Nama Proses	Deskripsi
1.	Tambah <i>member</i>	Proses penambahan <i>member</i> baru
2.	Ubah <i>member</i>	Proses untuk mengubah data <i>member</i>

Tabel 4.6 Keterangan Aliran Data DFD Level 2 Proses Registrasi

Nama Aliran Data	Deskripsi
Data_ <i>Member</i>	Data <i>member</i> dari <i>advertiser</i> dan <i>publisher</i>
Info_ <i>Member</i>	Informasi data dari <i>member</i>

4.1.5.5. DFD Level 2 Proses Pengelolaan iklan & Channel



Gambar 4.8 DFD Level 2 Proses Pengelolaan iklan & channel

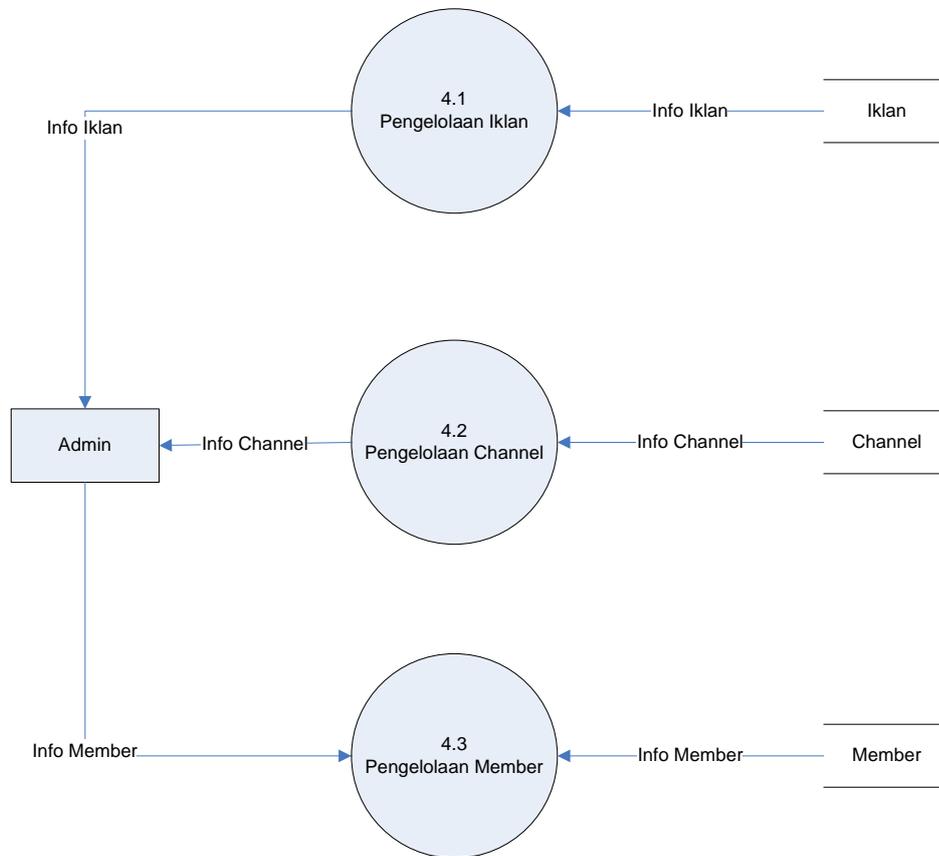
Tabel 4.7 DFD Level 2 Proses Pengelolaan iklan & channel

No.	Nama Proses	Deskripsi
1.	Pengelolaan Iklan	Proses untuk Pengelolaan iklan oleh <i>advertiser</i>
2.	Pengelolaan <i>channel</i>	Proses untuk Pengelolaan <i>channel</i> oleh <i>publisher</i>

Tabel 4.8 Keterangan Aliran Data DFD Level 2 Proses iklan & channel

Nama Aliran Data	Deskripsi
Data_Iklan	Data iklan dari <i>advertiser</i>
Info_Iklan	Informasi data iklan
Data_Channel	Data <i>channel</i> dari <i>publisher</i>
Info_Channel	Informasi data <i>channel</i>

4.1.5.6. DFD Level 2 Proses Pengelolaan *admin*



Gambar 4.9 DFD Level 2 Proses Pengelolaan *admin*

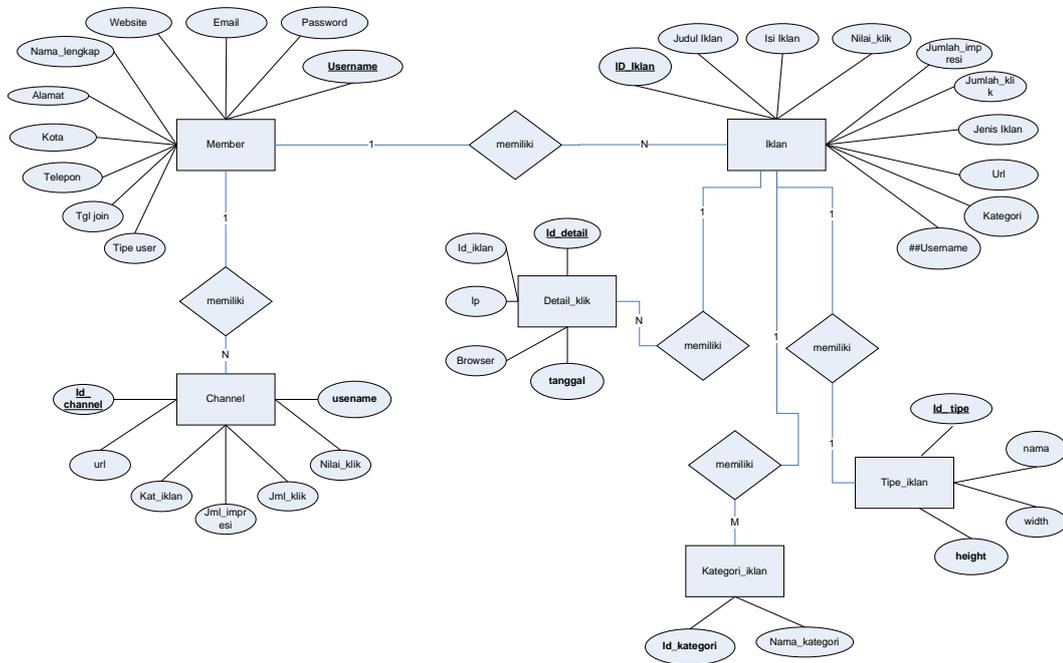
Tabel 4.9 DFD Level 2 Proses Pengelolaan *Admin*

No.	Nama Proses	Deskripsi
1.	Pengelolaan Iklan	Proses untuk Pengelolaan iklan oleh <i>advertiser</i>
2.	Pengelolaan <i>channel</i>	Proses untuk Pengelolaan <i>channel</i> oleh <i>publisher</i>
3.	Pengelolaan <i>member</i>	Proses untuk pengelolaan <i>member</i>

Tabel 4.10 Keterangan Aliran Data DFD Level 2 Proses *Admin*

Nama Aliran Data	Deskripsi
Info_Iklan	Informasi data iklan
Info_Channel	Informasi data <i>channel</i>
Info_Member	Informasi data <i>member</i>

4.1.6. Entity Relationship Diagram (ER-Diagram)



Gambar 4.10 ER-Diagram

Tabel 4.11 Keterangan entitas pada ERD

Nama	Deskripsi	Atribut	Primary Key
<i>Member</i>	Menyimpan data <i>member</i>	✓ Username ✓ Password ✓ Email ✓ Website ✓ Nama_lengkap ✓ Alamat ✓ Kota ✓ Telepon ✓ Tgl_join ✓ Tipe_user	Username
<i>Channel</i>	Menyimpan data <i>channel</i>	✓ ID_channel ✓ Url	ID_Channel

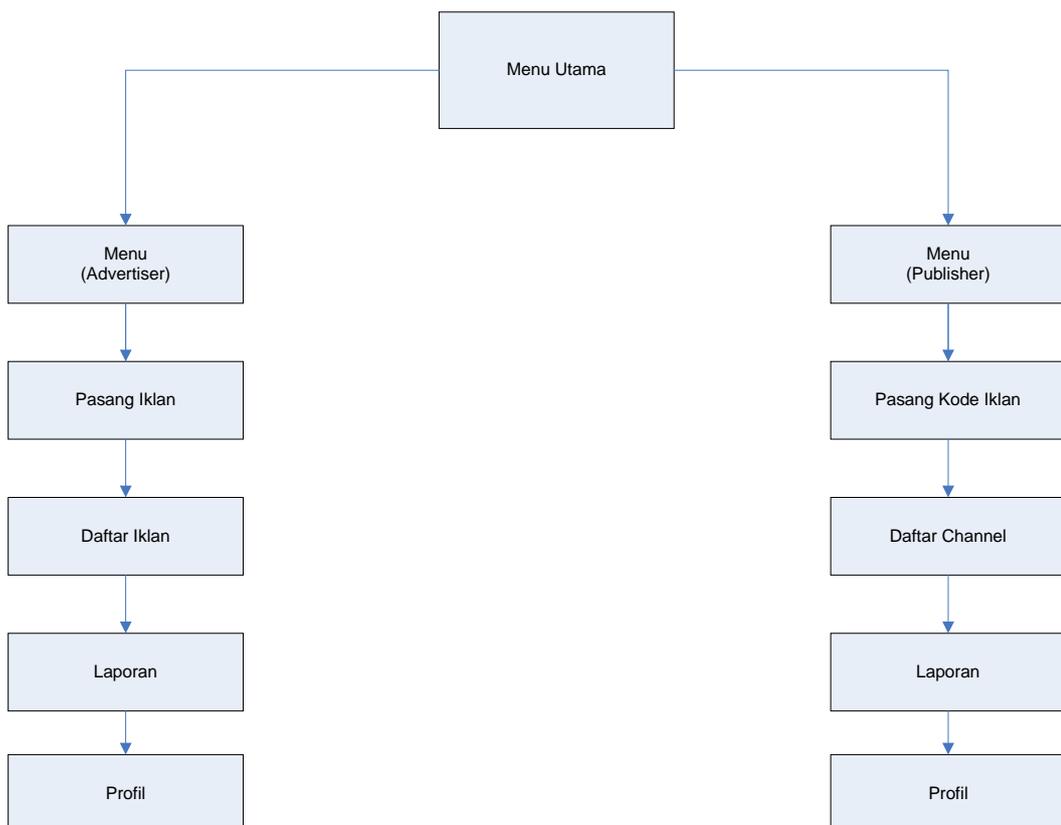
		<ul style="list-style-type: none"> ✓ kat_iklan ✓ jml_impresi ✓ jml_klik ✓ nilai_klik ✓ username 	
Iklan	Menyimpan data iklan	<ul style="list-style-type: none"> ✓ ID_iklan ✓ Judul ✓ Isi_iklan ✓ url ✓ idtipe ✓ lokasi_image ✓ kat_iklan ✓ deposit ✓ nilai_klik ✓ jml_impresi ✓ jenis_iklan ✓ username 	ID_iklan
Detail klik	Menyimpan data detail klik	<ul style="list-style-type: none"> ✓ id_detail ✓ id_iklan ✓ IP ✓ Browser ✓ tanggal 	id_detailklik
Tipe_iklan	Menyimpan data tipe iklan	<ul style="list-style-type: none"> ✓ id_tipe ✓ nama ✓ width ✓ height 	id_tipe
Kategori_iklan	Menyimpan data kategori iklan	<ul style="list-style-type: none"> ✓ id_kategori ✓ nama_kategori 	Id_kategori

4.2. Perancangan

Dalam tahap perancangan akan terbagi menjadi beberapa bagian pembahasan yaitu perancangan struktur menu, basis data serta perancangan antar muka yang akan dijelaskan sebagai berikut.

4.2.1. Perancangan Menu

Dalam pemakaian sistem ini diperlukan susunan daftar pilihan/menu sehingga pengguna yang belum terbiasa dengan sistem juga dapat menggunakan sistem ini. Pengguna akan dihadapkan pada berbagai alternatif menu yang ada. Dalam menentukan pilihannya, pengguna dapat menggunakan tombol tertentu dan setiap pilihan akan menghasilkan tampilan/halaman tertentu. Sistem yang akan dibangun memiliki menu dan sub-sub menu yang digambarkan pada bagan di bawah ini:



Gambar 4.11 Rancangan Menu

4.2.2. Perancangan Basis Data

Tahapan perancangan basis data digunakan untuk membuat detail data yang akan dipersiapkan pada tahap implementasi selanjutnya. Berikut adalah hasil perancangan basis data :

Tabel 4.12 Perancangan Tabel *Member*

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>
1	Username	Varchar(10)	NOT NULL
2	Password	Varchar(20)	NOT NULL
3	Email	Varchar(50)	NOT NULL
4	Website	Varchar(50)	NOT NULL
5	Nama Lengkap	Varchar(30)	NOT NULL
6	Alamat	Varchar(100)	NOT NULL
7	Kode Pos	Varchar(5)	NULL
8	Telp/hp	Varchar(15)	NULL
9	Tgl_join	Datetime	NOT NULL
10	Tipeuser	char(1)	NOT NULL

Tabel 4.13 Perancangan Tabel *Channel*

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>
1	ID_channel	Integer (5)	NOT NULL
2	Url	Varchar(20)	NOT NULL
3	kat_iklan	Varchar(50)	NOT NULL
4	jml_impresi	Integer (5)	NOT NULL
5	jml_klik	Integer (5)	NOT NULL
6	nilai_klik	Integer (5)	NOT NULL
7	username	Varchar(10)	NOT NULL

Tabel 4.14 Perancangan Tabel Iklan

No	Nama Field	Tipe Data	Null
1	ID_iklan	Integer (10)	NOT NULL
2	Judul	Varchar(25)	NOT NULL
3	Isi_iklan	Varchar(60)	NOT NULL
4	url	Varchar(50)	NOT NULL
5	Id_tipe	Integer(5)	NOT NULL
6	lokasi_image	Varchar(50)	NOT NULL
7	kat_iklan	Varchar(50)	NOT NULL
8	deposit	Integer(5)	NOT NULL
9	nilai_klik	Integer(5)	NOT NULL
10	jml_impresi	Integer(5)	NOT NULL
11	jenis_iklan	Char(1)	NOT NULL
12	username	Varchar(10)	NOT NULL

Tabel 4.15 Perancangan Tabel detail_klik

No	Nama Field	Tipe Data	Null
1	id_detail	Integer (10)	NOT NULL
2	id_iklan	Integer (10)	NOT NULL
3	IP	Varchar(20)	NOT NULL
4	Browser	Varchar(50)	NOT NULL
5	Tanggal	Datetime	NOT NULL

Tabel 4.16 Perancangan Tabel kategori_iklan

No	Nama Field	Tipe Data	Null
1	id_kategori	Integer (10)	NOT NULL
2	nama_kategori	varchar (30)	NOT NULL

4.2.3. Perancangan Antar Muka (*Interface*)

Interface sistem merupakan suatu sarana pengembangan sistem yang ditujukan untuk mempermudah pemakai berkomunikasi dengan sistem yang ada dan konsisten data juga ditunjukkan dalam *interface* tersebut. Penekanan *interface* meliputi tampilan yang baik, mudah dipahami dan tombol-tombol yang familiar.

The wireframe illustrates the main page layout. At the top is a large rectangular area labeled "Banner". Below it is a horizontal navigation bar containing five buttons: "Home", "Advertiser", "Publisher", "Layanan", and "Kontak Kami". The main content area is divided into three sections. On the left, there is a box for "Advertiser" with a "Login" button below it. In the center, there is a box for "Publisher" with a "Login" button below it. On the right, there is a shared login form with labels "Username" and "Password" next to input fields, and a "Login" button below them.

Gambar 4.12 Rancangan Halaman Utama

Penjelasan lebih rinci mengenai perancangan antarmuka lainnya dapat dilihat pada lampiran A.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas tentang implementasi dan pengujian sistem berdasarkan hasil analisis yang telah dilakukan sebelumnya.

5.1 Implementasi Sistem

Implementasi sistem meliputi tahap pengkodean program (*coding*), uji sub sistem (*testing*) dan penggabungan sub sistem. Pada tahap ini, sistem yang telah selesai dirancang akan dioperasikan dan selanjutnya dilakukan pengujian untuk melihat sejauh mana kebersesuaian hasil dengan tujuan yang dicapai dari pembuatan sistem ini.

Rancangan aplikasi *web service* periklanan online ini dibuat dengan menggunakan perangkat lunak PHP versi 5 dan *Database* yang digunakan adalah MySQL versi 5.0.

5.1.1 Batasan Implementasi

Agar pelaksanaan implementasi tidak keluar dari fokus pembahasan, maka perlu dijelaskan batasan implementasi sebagai berikut:

1. Menggunakan bahasa pemrograman PHP versi 5 dan *database* MySQL versi 5.
2. *Web service* hanya menampilkan iklan berjenis teks dan gambar.
3. Setiap klik iklan yang berasal dari IP (*Internet Protocol*) yang sama pada hari yang sama akan dihitung satu klik.

5.1.2 Lingkungan Implementasi

Lingkungan implementasi adalah lingkungan dimana aplikasi ini dikembangkan. Lingkungan implementasi sistem ada dua yaitu lingkungan perangkat keras dan lingkungan perangkat lunak, dengan spesifikasi sebagai berikut:

1. Perangkat Keras

Perangkat keras yang digunakan dalam rancang bangun sistem mempunyai spesifikasi sebagai berikut:

- a. *Processor* : Intel Atom 1.5GHz
- b. *Memory* : 1 GHz
- c. *Hardisk* : 320 GB

2. Perangkat Lunak

Perangkat lunak yang digunakan adalah sebagai berikut:

- a. Sistem Operasi : Windows XP Professional SP 2
- b. Bahasa Pemrograman : PHP 5.2.6
- c. DBMS : *Database* MySQL 5.0.51b
- d. *Browser* : Mozilla Firefox 3.0
Google Chrome v14.0.794.0

3. *Web Server*

Aplikasi ini dijalankan pada *server* hosting serta menggunakan domain yang beralamat di :

<http://www.xmladservice.info>

5.1.3 Hasil Implementasi

Berikut ini adalah hasil implementasi dari sistem yang telah di rancang. Pembahasan akan terbagi menjadi dua bagian, yaitu implementasi SOAP *Service* dan WSDL (*Web Service Definition Language*) serta implementasi tampilan antar muka sistem.

5.1.3.1 Implementasi SOAP *Service* dan dokumen WSDL (*Web Service Definition Language*)

Seperti yang telah dijelaskan pada sebelumnya untuk membangun sebuah *web service* maka dibutuhkan pendefinisian terhadap *service* itu sendiri. Implementasi terhadap *service* yang akan digunakan dapat dilihat pada tabel 5.1 berikut ini.

Tabel 5.1 Identifikasi *Service*

No	Nama <i>Service</i>	Deskripsi
1.	<i>Validate PubID</i>	'input' => 'xsd:Integer' - Parameter input 'output' => 'xsd:Boolean' - Parameter output
2.	<i>Validate Channel</i>	'input' => 'xsd:Integer' - Parameter input 'output' => 'xsd:Boolean' - Parameter output
3.	<i>Get Ads</i>	'id_channel' => 'xsd:Integer' - Parameter input 'adstype' => 'xsd:Integer' - Parameter input 'output' => 'tns:Array' - Parameter output

Detail masing-masing operasi *service* adalah sebagai berikut :

1. *Validate_PubID*

```

Name: Validate_PubID
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Validate_PubID
Style: rpc
Input:
  use: encoded
  namespace: urn:WSDLiklan
  encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
  message: Validate_PubIDRequest
  parts:
    input: xsd:String
Output:
  use: encoded
  namespace: urn:WSDLiklan
  encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
  message: Validate_PubIDResponse
  parts:
    output: xsd:Boolean
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Validasi Publisher

```

2. *Validate_Channel*

Name: Validate_Channel
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Validate_Channel
Style: rpc
Input:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Validate_ChannelRequest
 parts:
 input: xsd:Integer
Output:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Validate_ChannelResponse
 parts:
 output: xsd:Boolean
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Validasi Channel

3. *Get_Ads*

Name: Get_Ads
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Get_Ads
Style: rpc
Input:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Get_AdsRequest
 parts:
 id_channel: xsd:Int
 adstype: xsd:Int
Output:

```

    use: encoded
    namespace: urn:WSDLiklan
    encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
    message: Get_AdsResponse
    parts:
        output: tns:ListArray
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Mengambil data dari iklan dari database

```

WSDL untuk *service* tersebut adalah sebagai berikut (WSDL dapat dilihat di <http://xmladservice.info/soapcode/soapservers.php?wsdl>) :

```

<definitions xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="h
ttp://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="ur
n:WSDLiklan" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soa
p/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http
://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:WSDLiklan
">
<types>
<xsd:schema targetNamespace="urn:WSDLiklan">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encod
ing/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
<xsd:complexType name="ListArray">
<xsd:complexContent>
<xsd:restriction base="SOAP-ENC:Array">
<xsd:element name="0" type=""/>
<xsd:attribute ref="SOAP-
ENC:arrayType" wsdl:arrayType="xsd:string[]" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
</types>
<message name="Validate_PubIDRequest">
<part name="input" type="xsd:String" />
</message>
<message name="Validate_PubIDResponse">

```

```

<part name="output" type="xsd:Boolean"/>
</message>
<message name="Validate_ChannelRequest">
<part name="input" type="xsd:Integer"/>
</message>
<message name="Validate_ChannelResponse">
<part name="output" type="xsd:Boolean"/>
</message>
<message name="Get_AdsRequest">
<part name="id_channel" type="xsd:Int"/>
<part name="adstype" type="xsd:Int"/>
</message>
<message name="Get_AdsResponse">
<part name="output" type="tns:ListArray"/>
</message>
<portType name="Web service with SOAPPortType">
<operation name="Validate_PubID">
<documentation>Validasi Publisher</documentation>
<input message="tns:Validate_PubIDRequest"/>
<output message="tns:Validate_PubIDResponse"/>
</operation>
<operation name="Validate_Channel">
<documentation>Validasi Channel</documentation>
<input message="tns:Validate_ChannelRequest"/>
<output message="tns:Validate_ChannelResponse"/>
</operation>
<operation name="Get_Ads">
<documentation>Mengambil data dari iklan dari
database</documentation>
<input message="tns:Get_AdsRequest"/>
<output message="tns:Get_AdsResponse"/>
</operation>
</portType>
<binding name="Web service with SOAPBinding" type="tns:Web
service with SOAPPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.
org/soap/http"/>
<operation name="Validate_PubID">
<soap:operation soapAction="urn:WSDLiklan#Validate_PubID" st
yle="rpc"/>
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
<output>

```

```
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="Validate_Channel">
<soap:operation soapAction="urn:WSDLiklan#Validate_Channel"
style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="Get_Ads">
<soap:operation soapAction="urn:WSDLiklan#Get_Ads" style="rp
c" />
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
</binding>
<service name="Web service with SOAP">
<port name="Web service with SOAPPort" binding="tns:Web
service with SOAPBinding">
<soap:address location="http://xmladservice.info/soapcode/so
apserver.php" />
</port>
</service>
</definitions>
```

5.1.3.2 Tampilan Antarmuka (*Interface*)

Tampilan antarmuka terbagi menjadi beberapa sisi yaitu sisi *publisher*, *advertiser*.

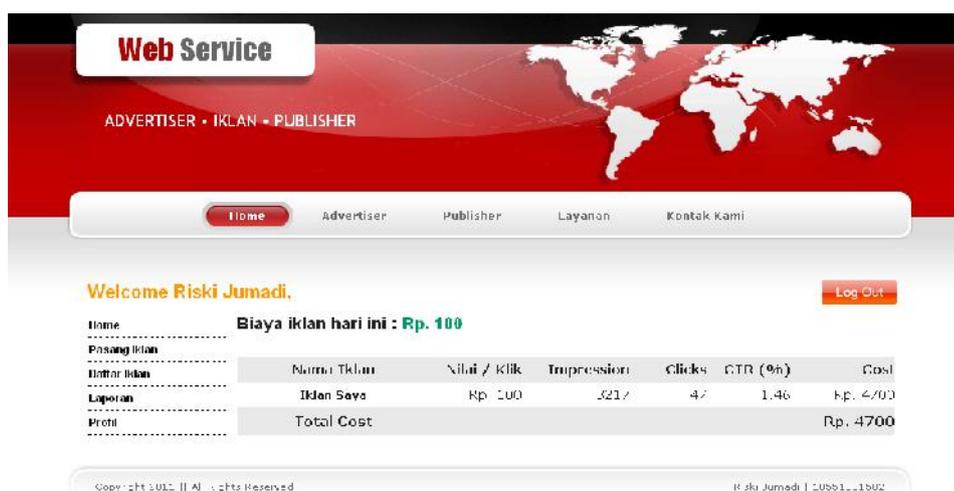
5.1.3.2.1 Tampilan Halaman Utama



Gambar 5.1 Tampilan Halaman Utama

User akan langsung dihadapkan pada pilihan untuk mendaftar sebagai *Advertiser* atau *Publisher*.

5.1.3.2.2 Tampilan Halaman *Advertiser*



Gambar 5.2 Tampilan Halaman *Advertiser*

Gambar 5.2 adalah tampilan pada akun *advertiser* setelah melakukan proses *login*.

5.1.3.2.3 Tampilan Halaman *Publisher*



The screenshot shows a web service interface for a publisher. At the top, there is a 'Web Service' logo and a navigation menu with options: Home, Advertiser, Publisher, Layanan, and Kontak Kami. Below the navigation, a welcome message reads 'Welcome Riski Jumadi' with a 'Log Out' button. A summary section displays 'Klik iklan hari ini : Rp. 100'. A table provides detailed performance metrics for an advertisement.

Channel	Nama Iklan	Nilai / Klik	Impression	Clicks	CTR (%)	Earning
Layanan	Ad Content	Rp. 100	2641	32	1.21	Rp. 3200
Profil	Total Pendapatan					Rp. 3200

At the bottom of the page, there is a copyright notice: 'Copyright ©2011 All rights reserved. Riski Jumadi | 10511001802'.

Gambar 5.3 Tampilan Halaman *Publisher*

Gambar 5.3 adalah tampilan pada akun *publisher* setelah melakukan proses *login*.

Penjelasan lebih rinci mengenai hasil implementasi antarmuka lainnya dapat dilihat pada lampiran B.

5.2 Pengujian Sistem

Pengujian dilakukan dengan tujuan untuk menjamin sistem yang dibuat sesuai dengan hasil analisis dan perancangan dan menghasilkan satu kesimpulan. Sebelum sistem diimplementasikan terlebih dahulu harus dipastikan program bebas dari kesalahan-kesalahan yang mungkin terjadi.

Dalam hal ini pengujian akan terbagi menjadi dua bagian yaitu pengujian aplikasi pada *server* dan pengujian pada *client*.

5.2.1 Pengujian Aplikasi Server

Pengujian ini dilakukan pada aplikasi *server*. Pengujian dilakukan pada halaman *advertiser* dan *publisher* karena kedua user memiliki tampilan menu yang berbeda. Cara pengujian yang dilakukan menggunakan metode *Black Box*. Hasil dari pengujian adalah sebagai berikut.

5.2.1.1 Identifikasi dan Pengujian Menu Login

Prekondisi : Halaman Utama Aplikasi

Tabel 5.2 Hasil Pengujian Menu Login

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu <i>login</i>	Masukkan <i>username</i> dan <i>password</i> dan klik tombol <i>Login</i>	- <i>Username</i> atau <i>password</i> salah - <i>Username</i> dan <i>password</i> benar	- Pesan error “ <i>Login Invalid !</i> ” - Tampil ke menu halaman utama <i>member</i>	Berhasil

5.2.1.2 Identifikasi dan Pengujian Menu Profil

Prekondisi : Halaman Utama *Member*

Tabel 5.3 Hasil Pengujian Menu Profil

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu profil	- Masukkan Data pribadi dan klik tombol <i>edit</i> -	- Data pribadi <i>member</i>	- Tampilan pesan “Data berhasil di <i>edit</i> ”	Berhasil

Tabel 5.3 Hasil Pengujian Menu Profil

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu profil	- Input <i>password</i> lama dan baru Klik Submit.	- <i>Password</i> lama dan <i>password</i> baru	- Kembali ke menu profil	Berhasil

Penjelasan lebih rinci mengenai hasil pengujian lainnya dapat dilihat pada lampiran C.

Dari hasil pengujian pada aplikasi tersebut dapat disimpulkan bahwa seluruh menu dapat bekerja dengan baik dan memperoleh hasil seperti yang diinginkan.

5.2.2 Pengujian *Client / Channel*

Pengujian ini ditujukan untuk dapat melihat bahwa iklan yang di *request* oleh *client* dan dapat diproses oleh *server* dengan mengirimkan *response* serta menampilkan data iklan pada *client (Channel Publisher)*.

Berikut adalah daftar *channel* yang akan diuji.

Tabel 5.4 Daftar *Channel*

No.	<i>URL</i>	Ukuran Iklan
1	http://anaisine.blogspot.com	125 x 125
2	http://whitenews.blogspot.com/	728 x 90

5.2.2.1 Skenario Pengujian

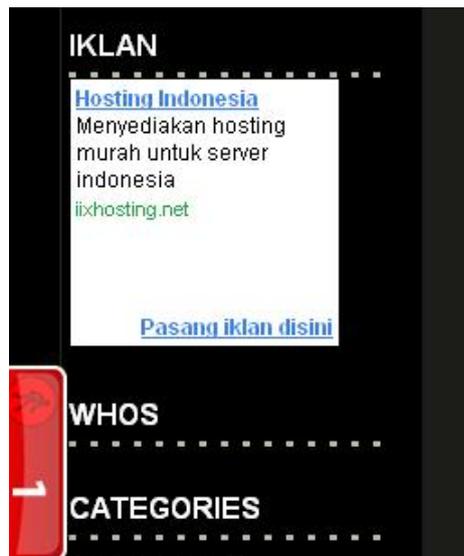
Skenario pengujian yang akan dilakukan adalah sebagai berikut :

1. *Publisher login* ke aplikasi kemudian mengambil *script* yang disediakan oleh sistem.
2. *Publisher* meng-*copy script* tersebut kedalam blog/*website* miliknya.

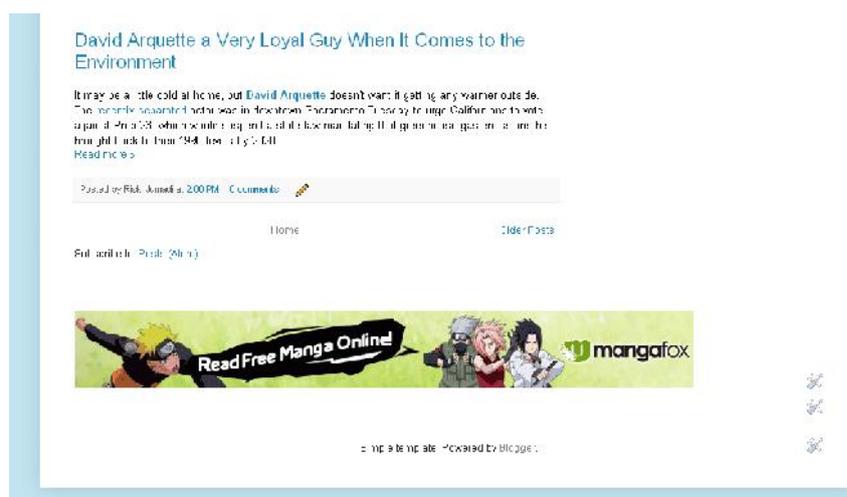
3. *Script* tersebut akan melakukan *request* ke *server* untuk mendapatkan iklan.

5.2.2.2 Hasil Pengujian

Setelah dilakukan pengujian maka iklan akan tayang pada *website* tersebut seperti gambar berikut ini.



Gambar 5.4 Hasil pengujian pada <http://anaisine.blogspot.com>



Gambar 5.5 Hasil pengujian pada <http://whiteneews.blogspot.com>

SOAP *Message* yang diperoleh dari hasil pengujian tersebut adalah sebagai berikut :

1. <http://anaisine.blogspot.com>

SOAP Request :

```
POST /soapcode/soapserver.php HTTP/1.0
Host: xmladservice.info
User-Agent: NuSOAP/0.9.5 (1.123)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "urn:WSDLiklan#Get_Ads"
Content-Length: 562
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:WSDLiklan"><SOAP-ENV:Body><tns:Get_Ads
xmlns:tns="urn:WSDLiklan"><id_channel
xsi:type="xsd:Int">12</id_channel><adstype
xsi:type="xsd:Int">10</adstype></tns:Get_Ads></SOAP-
ENV:Body></SOAP-ENV:Envelope>
```

SOAP Response :

```
HTTP/1.1 200 OK
Server: nginx admin
Date: Fri, 14 Oct 2011 19:28:33 GMT
Content-Type: text/xml; charset=ISO-8859-1
Connection: close
X-Powered-By: PHP/5.2.14
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Content-Length: 581
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:Get_AdsResponse
xmlns:ns1="urn:WSDLiklan"><output xsi:type="SOAP-ENC:Array"
SOAP-ENC:arrayType="xsd:string[3]"><item
xsi:type="xsd:string">Hosting Indonesia</item><item
xsi:type="xsd:string">Menyediakan hosting murah untuk server
indonesia</item><item
xsi:type="xsd:string">http://www.iixhosting.net</item></outp
```

```
ut></ns1:Get_AdsResponse></SOAP-ENV:Body></SOAP-
ENV:Envelope>
```

2. <http://whiteneeds.blogspot.com>

SOAP Request :

```
POST /soapcode/soapserver.php HTTP/1.0
Host: xmladservice.info
User-Agent: NuSOAP/0.9.5 (1.123)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "urn:WSDLiklan#Get_Ads"
Content-Length: 561
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:WSDLiklan"><SOAP-ENV:Body><tns:Get_Ads
xmlns:tns="urn:WSDLiklan"><id_channel
xsi:type="xsd:Int">12</id_channel><adstype
xsi:type="xsd:Int">1</adstype></tns:Get_Ads></SOAP-
ENV:Body></SOAP-ENV:Envelope>
```

SOAP Response :

```
HTTP/1.1 200 OK
Server: nginx admin
Date: Fri, 14 Oct 2011 19:33:23 GMT
Content-Type: text/xml; charset=ISO-8859-1
Connection: close
X-Powered-By: PHP/5.2.14
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Content-Length: 655
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:Get_AdsResponse
xmlns:ns1="urn:WSDLiklan"><output xsi:type="SOAP-ENC:Array"
SOAP-ENC:arrayType="xsd:string[3]"><item
xsi:type="xsd:string">aaa</item><item
xsi:type="xsd:string"></item><item
xsi:type="xsd:string">http://mangafox.com</item></output></n
s1:Get_AdsResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Dari hasil pengujian yang dijelaskan diatas dapat disimpulkan proses SOAP *Request* dan SOAP *Response* dapat berjalan dengan baik seperti hasil yang diinginkan.

5.3 Kesimpulan Pengujian

Berdasarkan hasil pengujian aplikasi yang telah dilakukan diperoleh kesimpulan :

1. Seluruh menu pada aplikasi dapat berjalan dengan baik sesuai dengan fungsi masing-masing.
2. Proses pertukaran data dengan SOAP dapat bekerja dengan baik sehingga menghasilkan keluaran yang sesuai.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas tentang implementasi dan pengujian sistem berdasarkan hasil analisis yang telah dilakukan sebelumnya.

5.1 Implementasi Sistem

Implementasi sistem meliputi tahap pengkodean program (*coding*), uji sub sistem (*testing*) dan penggabungan sub sistem. Pada tahap ini, sistem yang telah selesai dirancang bangun akan dioperasikan dan selanjutnya dilakukan pengujian untuk melihat sejauh mana kebersesuaian hasil dengan tujuan yang dicapai dari pembuatan sistem ini.

Rancangan aplikasi *web service* periklanan online ini dibuat dengan menggunakan perangkat lunak PHP versi 5 dan *Database* yang digunakan adalah MySQL versi 5.0.

5.1.1 Batasan Implementasi

Agar pelaksanaan implementasi tidak keluar dari fokus pembahasan, maka perlu dijelaskan batasan implementasi sebagai berikut:

1. Menggunakan bahasa pemrograman PHP versi 5 dan *database* MySQL versi 5.
2. *Web service* hanya menampilkan iklan berjenis teks dan gambar.
3. Setiap klik iklan yang berasal dari IP (*Internet Protocol*) yang sama pada hari yang sama akan dihitung satu klik.

5.1.2 Lingkungan Implementasi

Lingkungan implementasi adalah lingkungan dimana aplikasi ini dikembangkan. Lingkungan implementasi sistem ada dua yaitu lingkungan perangkat keras dan lingkungan perangkat lunak, dengan spesifikasi sebagai berikut:

1. Perangkat Keras

Perangkat keras yang digunakan dalam rancang bangun sistem mempunyai spesifikasi sebagai berikut:

- a. *Processor* : Intel Atom 1.5GHz
- b. *Memory* : 1 GHz
- c. *Hardisk* : 320 GB

2. Perangkat Lunak

Perangkat lunak yang digunakan adalah sebagai berikut:

- a. Sistem Operasi : Windows XP Professional SP 2
- b. Bahasa Pemrograman : PHP 5.2.6
- c. DBMS : *Database* MySQL 5.0.51b
- d. *Browser* : Mozilla Firefox 3.0
Google Chrome v14.0.794.0

3. *Web Server*

Aplikasi ini dijalankan pada *server* hosting serta menggunakan domain yang beralamat di :

<http://www.xmladservice.info>

5.1.3 Hasil Implementasi

Berikut ini adalah hasil implementasi dari sistem yang telah di rancang. Pembahasan akan terbagi menjadi dua bagian, yaitu implementasi SOAP *Service* dan WSDL (*Web Service Definition Language*) serta implementasi tampilan antar muka sistem.

5.1.3.1 Implementasi SOAP *Service* dan dokumen WSDL (*Web Service Definition Language*)

Seperti yang telah dijelaskan pada sebelumnya untuk membangun sebuah *web service* maka dibutuhkan pendefinisian terhadap *service* itu sendiri. Implementasi terhadap *service* yang akan digunakan dapat dilihat pada tabel 5.1 berikut ini.

Tabel 5.1 Identifikasi *Service*

No	Nama <i>Service</i>	Deskripsi
1.	<i>Validate PubID</i>	'input' => 'xsd:Integer' - Parameter input 'output' => 'xsd:Boolean' - Parameter output
2.	<i>Validate Channel</i>	'input' => 'xsd:Integer' - Parameter input 'output' => 'xsd:Boolean' - Parameter output
3.	<i>Get Ads</i>	'id_channel' => 'xsd:Integer' - Parameter input 'adstype' => 'xsd:Integer' - Parameter input 'output' => 'tns:Array' - Parameter output

Detail masing-masing operasi *service* adalah sebagai berikut :

1. *Validate_PubID*

```
Name: Validate_PubID
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Validate_PubID
Style: rpc
Input:
  use: encoded
  namespace: urn:WSDLiklan
  encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
  message: Validate_PubIDRequest
  parts:
    input: xsd:String
Output:
  use: encoded
  namespace: urn:WSDLiklan
  encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
  message: Validate_PubIDResponse
  parts:
    output: xsd:Boolean
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Validasi Publisher
```

2. *Validate_Channel*

Name: Validate_Channel
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Validate_Channel
Style: rpc
Input:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Validate_ChannelRequest
 parts:
 input: xsd:Integer
Output:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Validate_ChannelResponse
 parts:
 output: xsd:Boolean
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Validasi Channel

3. *Get_Ads*

Name: Get_Ads
Binding: Web service with SOAPBinding
Endpoint:
http://xmladservice.info/soapcode/soapserver.php
SoapAction: urn:WSDLiklan#Get_Ads
Style: rpc
Input:
 use: encoded
 namespace: urn:WSDLiklan
 encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
 message: Get_AdsRequest
 parts:
 id_channel: xsd:Int
 adstype: xsd:Int
Output:

```

    use: encoded
    namespace: urn:WSDLiklan
    encodingStyle:
http://schemas.xmlsoap.org/soap/encoding/
    message: Get_AdsResponse
    parts:
        output: tns:ListArray
Namespace: urn:WSDLiklan
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: Mengambil data dari iklan dari database

```

WSDL untuk *service* tersebut adalah sebagai berikut (WSDL dapat dilihat di <http://xmladservice.info/soapcode/soapserver.php?wsdl>) :

```

<definitions xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="h
ttp://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="ur
n:WSDLiklan" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soa
p/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http
://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:WSDLiklan
">
<types>
<xsd:schema targetNamespace="urn:WSDLiklan">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encod
ing/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
<xsd:complexType name="ListArray">
<xsd:complexContent>
<xsd:restriction base="SOAP-ENC:Array">
<xsd:element name="0" type=""/>
<xsd:attribute ref="SOAP-
ENC:arrayType" wsdl:arrayType="xsd:string[]" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
</types>
<message name="Validate_PubIDRequest">
<part name="input" type="xsd:String" />
</message>
<message name="Validate_PubIDResponse">

```

```

<part name="output" type="xsd:Boolean"/>
</message>
<message name="Validate_ChannelRequest">
<part name="input" type="xsd:Integer"/>
</message>
<message name="Validate_ChannelResponse">
<part name="output" type="xsd:Boolean"/>
</message>
<message name="Get_AdsRequest">
<part name="id_channel" type="xsd:Int"/>
<part name="adstype" type="xsd:Int"/>
</message>
<message name="Get_AdsResponse">
<part name="output" type="tns:ListArray"/>
</message>
<portType name="Web service with SOAPPortType">
<operation name="Validate_PubID">
<documentation>Validasi Publisher</documentation>
<input message="tns:Validate_PubIDRequest"/>
<output message="tns:Validate_PubIDResponse"/>
</operation>
<operation name="Validate_Channel">
<documentation>Validasi Channel</documentation>
<input message="tns:Validate_ChannelRequest"/>
<output message="tns:Validate_ChannelResponse"/>
</operation>
<operation name="Get_Ads">
<documentation>Mengambil data dari iklan dari
database</documentation>
<input message="tns:Get_AdsRequest"/>
<output message="tns:Get_AdsResponse"/>
</operation>
</portType>
<binding name="Web service with SOAPBinding" type="tns:Web
service with SOAPPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.
org/soap/http"/>
<operation name="Validate_PubID">
<soap:operation soapAction="urn:WSDLiklan#Validate_PubID" st
yle="rpc"/>
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
<output>

```

```
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="Validate_Channel">
<soap:operation soapAction="urn:WSDLiklan#Validate_Channel"
style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="Get_Ads">
<soap:operation soapAction="urn:WSDLiklan#Get_Ads" style="rp
c" />
<input>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:WSDLiklan" encodingS
tyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
</binding>
<service name="Web service with SOAP">
<port name="Web service with SOAPPort" binding="tns:Web
service with SOAPBinding">
<soap:address location="http://xmladservice.info/soapcode/so
apserver.php" />
</port>
</service>
</definitions>
```

5.1.3.2 Tampilan Antarmuka (*Interface*)

Tampilan antarmuka terbagi menjadi beberapa sisi yaitu sisi *publisher*, *advertiser*.

5.1.3.2.1 Tampilan Halaman Utama



Gambar 5.1 Tampilan Halaman Utama

User akan langsung dihadapkan pada pilihan untuk mendaftar sebagai *Advertiser* atau *Publisher*.

5.1.3.2.2 Tampilan Halaman *Advertiser*



Gambar 5.2 Tampilan Halaman *Advertiser*

Gambar 5.2 adalah tampilan pada akun *advertiser* setelah melakukan proses *login*.

5.1.3.2.3 Tampilan Halaman *Publisher*

Web Service Exit full screen (F11)

ADVERTISER • IKLAN • PUBLISHER

Home Advertiser Publisher Layanan Kontak Kami

Welcome Riski Jumadi, Log Out

Home **Klik Iklan hari ini : Rp. 100**

Channel	Nama Iklan	Nilai / Klik	Impression	Clicks	CTR (%)	Earning
Layanan	Ad Content	Rp. 100	2641	32	1.21	Rp. 3200
Profil	Total Pendapatan					Rp. 3200

Copyright ©2011 All rights reserved. Riski Jumadi | 10511001802

Gambar 5.3 Tampilan Halaman *Publisher*

Gambar 5.3 adalah tampilan pada akun *publisher* setelah melakukan proses *login*.

Penjelasan lebih rinci mengenai hasil implementasi antarmuka lainnya dapat dilihat pada lampiran B.

5.2 Pengujian Sistem

Pengujian dilakukan dengan tujuan untuk menjamin sistem yang dibuat sesuai dengan hasil analisis dan perancangan dan menghasilkan satu kesimpulan. Sebelum sistem diimplementasikan terlebih dahulu harus dipastikan program bebas dari kesalahan-kesalahan yang mungkin terjadi.

Dalam hal ini pengujian akan terbagi menjadi dua bagian yaitu pengujian aplikasi pada *server* dan pengujian pada *client*.

5.2.1 Pengujian Aplikasi Server

Pengujian ini dilakukan pada aplikasi *server*. Pengujian dilakukan pada halaman *advertiser* dan *publisher* karena kedua user memiliki tampilan menu yang berbeda. Cara pengujian yang dilakukan menggunakan metode *Black Box*. Hasil dari pengujian adalah sebagai berikut.

5.2.1.1 Identifikasi dan Pengujian Menu Login

Prekondisi : Halaman Utama Aplikasi

Tabel 5.2 Hasil Pengujian Menu Login

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu <i>login</i>	Masukkan <i>username</i> dan <i>password</i> dan klik tombol <i>Login</i>	- <i>Username</i> atau <i>password</i> salah - <i>Username</i> dan <i>password</i> benar	- Pesan error “ <i>Login Invalid !</i> ” - Tampil ke menu halaman utama <i>member</i>	Berhasil

5.2.1.2 Identifikasi dan Pengujian Menu Profil

Prekondisi : Halaman Utama *Member*

Tabel 5.3 Hasil Pengujian Menu Profil

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu profil	- Masukkan Data pribadi dan klik tombol <i>edit</i> -	- Data pribadi <i>member</i>	- Tampilan pesan “Data berhasil di <i>edit</i> ”	Berhasil

Tabel 5.3 Hasil Pengujian Menu Profil

Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Tes
Pengujian menu profil	- Input <i>password</i> lama dan baru Klik Submit.	- <i>Password</i> lama dan <i>password</i> baru	- Kembali ke menu profil	Berhasil

Penjelasan lebih rinci mengenai hasil pengujian lainnya dapat dilihat pada lampiran C.

Dari hasil pengujian pada aplikasi tersebut dapat disimpulkan bahwa seluruh menu dapat bekerja dengan baik dan memperoleh hasil seperti yang diinginkan.

5.2.2 Pengujian *Client / Channel*

Pengujian ini ditujukan untuk dapat melihat bahwa iklan yang di *request* oleh *client* dan dapat diproses oleh *server* dengan mengirimkan *response* serta menampilkan data iklan pada *client (Channel Publisher)*.

Berikut adalah daftar *channel* yang akan diuji.

Tabel 5.4 Daftar *Channel*

No.	<i>URL</i>	Ukuran Iklan
1	http://anaisine.blogspot.com	125 x 125
2	http://whitenews.blogspot.com/	728 x 90

5.2.2.1 Skenario Pengujian

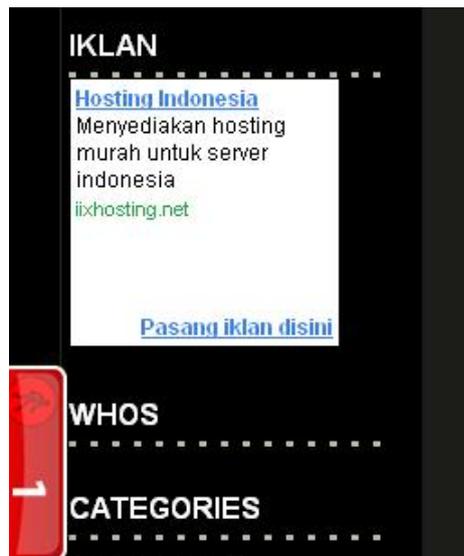
Skenario pengujian yang akan dilakukan adalah sebagai berikut :

1. *Publisher login* ke aplikasi kemudian mengambil *script* yang disediakan oleh sistem.
2. *Publisher* meng-*copy script* tersebut kedalam blog/*website* miliknya.

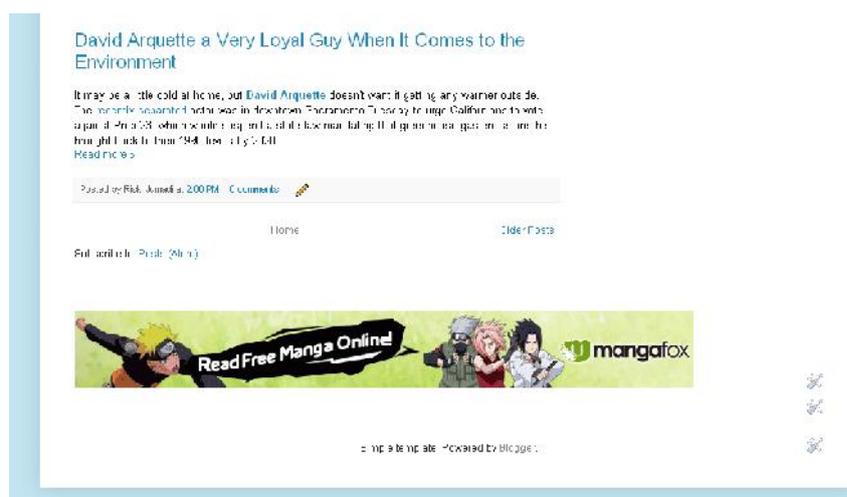
3. *Script* tersebut akan melakukan *request* ke *server* untuk mendapatkan iklan.

5.2.2.2 Hasil Pengujian

Setelah dilakukan pengujian maka iklan akan tayang pada *website* tersebut seperti gambar berikut ini.



Gambar 5.4 Hasil pengujian pada <http://anaisine.blogspot.com>



Gambar 5.5 Hasil pengujian pada <http://whiteneews.blogspot.com>

SOAP *Message* yang diperoleh dari hasil pengujian tersebut adalah sebagai berikut :

1. <http://anaisine.blogspot.com>

SOAP Request :

```
POST /soapcode/soapserver.php HTTP/1.0
Host: xmladservice.info
User-Agent: NuSOAP/0.9.5 (1.123)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "urn:WSDLiklan#Get_Ads"
Content-Length: 562
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:WSDLiklan"><SOAP-ENV:Body><tns:Get_Ads
xmlns:tns="urn:WSDLiklan"><id_channel
xsi:type="xsd:Int">12</id_channel><adstype
xsi:type="xsd:Int">10</adstype></tns:Get_Ads></SOAP-
ENV:Body></SOAP-ENV:Envelope>
```

SOAP Response :

```
HTTP/1.1 200 OK
Server: nginx admin
Date: Fri, 14 Oct 2011 19:28:33 GMT
Content-Type: text/xml; charset=ISO-8859-1
Connection: close
X-Powered-By: PHP/5.2.14
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Content-Length: 581
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:Get_AdsResponse
xmlns:ns1="urn:WSDLiklan"><output xsi:type="SOAP-ENC:Array"
SOAP-ENC:arrayType="xsd:string[3]"><item
xsi:type="xsd:string">Hosting Indonesia</item><item
xsi:type="xsd:string">Menyediakan hosting murah untuk server
indonesia</item><item
xsi:type="xsd:string">http://www.iixhosting.net</item></outp
```

```
ut></ns1:Get_AdsResponse></SOAP-ENV:Body></SOAP-
ENV:Envelope>
```

2. <http://whiteneews.blogspot.com>

SOAP Request :

```
POST /soapcode/soapserver.php HTTP/1.0
Host: xmladservice.info
User-Agent: NuSOAP/0.9.5 (1.123)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "urn:WSDLiklan#Get_Ads"
Content-Length: 561
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:WSDLiklan"><SOAP-ENV:Body><tns:Get_Ads
xmlns:tns="urn:WSDLiklan"><id_channel
xsi:type="xsd:Int">12</id_channel><adstype
xsi:type="xsd:Int">1</adstype></tns:Get_Ads></SOAP-
ENV:Body></SOAP-ENV:Envelope>
```

SOAP Response :

```
HTTP/1.1 200 OK
Server: nginx admin
Date: Fri, 14 Oct 2011 19:33:23 GMT
Content-Type: text/xml; charset=ISO-8859-1
Connection: close
X-Powered-By: PHP/5.2.14
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Content-Length: 655
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:Get_AdsResponse
xmlns:ns1="urn:WSDLiklan"><output xsi:type="SOAP-ENC:Array"
SOAP-ENC:arrayType="xsd:string[3]"><item
xsi:type="xsd:string">aaa</item><item
xsi:type="xsd:string"></item><item
xsi:type="xsd:string">http://mangafox.com</item></output></n
s1:Get_AdsResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Dari hasil pengujian yang dijelaskan diatas dapat disimpulkan proses SOAP *Request* dan SOAP *Response* dapat berjalan dengan baik seperti hasil yang diinginkan.

5.3 Kesimpulan Pengujian

Berdasarkan hasil pengujian aplikasi yang telah dilakukan diperoleh kesimpulan :

1. Seluruh menu pada aplikasi dapat berjalan dengan baik sesuai dengan fungsi masing-masing.
2. Proses pertukaran data dengan SOAP dapat bekerja dengan baik sehingga menghasilkan keluaran yang sesuai.

BAB VI

P E N U T U P

6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil penelitian tugas akhir ini adalah sebagai berikut :

1. Penerapan XML *Web Service* dengan SOAP (*Simple Object Access Protocol*) dalam situs periklanan *online* telah berhasil dirancang dan diimplementasikan dalam bentuk aplikasi berbasis web yang dapat memudahkan *advertiser* dalam mengoptimalkan penayangan iklannya.
2. *SOAP Message* berhasil mengirimkan pesan *SOAP Request* dan *SOAP Response* dalam melakukan proses pertukaran data antara aplikasi server dengan *website/blog publisher*.
3. Informasi yang diberikan oleh aplikasi *server* kepada *client* adalah berupa deskripsi iklan (isi dan judul) serta *url* tujuan dari iklan tersebut.

6.2 Saran

Beberapa hal yang disarankan untuk pengembangan sistem selanjutnya adalah sebagai berikut :

1. Menambahkan fitur dimana *advertiser* dapat memilih menayangkan iklannya hanya pada *visitor* yang berasal dari daerah tertentu.
2. Aplikasi *web service* ini dapat diimplementasikan menggunakan metode XML lainnya seperti XML-RPC dan REST yang masing-masing memiliki struktur dan spesifikasi yang berbeda.
3. Mengembangkan aplikasi pada *server* yang memungkinkan iklan ditayangkan sesuai dengan *keyword* atau tema yang berhubungan pada situs web *publisher*.

BAB VI

P E N U T U P

6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil penelitian tugas akhir ini adalah sebagai berikut :

1. Penerapan XML *Web Service* dengan SOAP (*Simple Object Access Protocol*) dalam situs periklanan *online* telah berhasil dirancang dan diimplementasikan dalam bentuk aplikasi berbasis web yang dapat memudahkan *advertiser* dalam mengoptimalkan penayangan iklannya.
2. *SOAP Message* berhasil mengirimkan pesan *SOAP Request* dan *SOAP Response* dalam melakukan proses pertukaran data antara aplikasi server dengan *website/blog publisher*.
3. Informasi yang diberikan oleh aplikasi *server* kepada *client* adalah berupa deskripsi iklan (isi dan judul) serta *url* tujuan dari iklan tersebut.

6.2 Saran

Beberapa hal yang disarankan untuk pengembangan sistem selanjutnya adalah sebagai berikut :

1. Menambahkan fitur dimana *advertiser* dapat memilih menayangkan iklannya hanya pada *visitor* yang berasal dari daerah tertentu.
2. Aplikasi *web service* ini dapat diimplementasikan menggunakan metode XML lainnya seperti XML-RPC dan REST yang masing-masing memiliki struktur dan spesifikasi yang berbeda.
3. Mengembangkan aplikasi pada *server* yang memungkinkan iklan ditayangkan sesuai dengan *keyword* atau tema yang berhubungan pada situs web *publisher*.

DAFTAR PUSTAKA

Azis, M. Farid. *Belajar Sendiri Pemrograman PHP 4 Bagi Web Programmer*. PT.Elex Media Komputindo, Jakarta, 2001.

Daniel, Roy. *Pengenalan Konsep XML Web Service*. Ilmukomputer.com. 2003

Junaedi, Moh.. *Pengantar XML*. Ilmukomputer.com. 2003

Lucky. *XML Web Service*. Jasakom. 2008

Muller, John Paul. *Special Edition Using SOAP*. Que, Indianapolis USA, 2002.

Short, Scott. *Building XML Web Services For The Microsoft .NET Platform*. PT.Elex Media Komputindo, Jakarta, 2003.

Simpson John E. *Just XML*. ANDI, Yogyakarta, 1999.

Turban, Efraim, dkk. *Decision Support Systems and Intelligent System*. ANDI, Yogyakarta, 2005.

“AdSense”. [Online] Available <http://id.wikipedia.org/wiki/adsense>, diakses 28 April 2010.

“SOAP”. [Online] Available <http://id.wikipedia.org/wiki/soap>, diakses 26 Maret 2010.

“WSDL”. [Online] Available <http://id.wikipedia.org/wiki/wsdl>, diakses 23 Maret 2010.

DAFTAR PUSTAKA

Azis, M. Farid. *Belajar Sendiri Pemrograman PHP 4 Bagi Web Programmer*. PT.Elex Media Komputindo, Jakarta, 2001.

Daniel, Roy. *Pengenalan Konsep XML Web Service*. Ilmukomputer.com. 2003

Junaedi, Moh.. *Pengantar XML*. Ilmukomputer.com. 2003

Lucky. *XML Web Service*. Jasakom. 2008

Muller, John Paul. *Special Edition Using SOAP*. Que, Indianapolis USA, 2002.

Short, Scott. *Building XML Web Services For The Microsoft .NET Platform*. PT.Elex Media Komputindo, Jakarta, 2003.

Simpson John E. *Just XML*. ANDI, Yogyakarta, 1999.

Turban, Efraim, dkk. *Decision Support Systems and Intelligent System*. ANDI, Yogyakarta, 2005.

“AdSense”. [Online] Available <http://id.wikipedia.org/wiki/adsense>, diakses 28 April 2010.

“SOAP”. [Online] Available <http://id.wikipedia.org/wiki/soap>, diakses 26 Maret 2010.

“WSDL”. [Online] Available <http://id.wikipedia.org/wiki/wsdl>, diakses 23 Maret 2010.