

# Aplikasi Iklan Baris *Online* menggunakan Arsitektur *REST Web Service*

Yuli Fauziah

Jurusan Teknik Informatika, UPN "Veteran" Yogyakarta

Jln. Babarsari No. 2 Tambakbayan Yogyakarta

e-mail: [yuli.if@gmail.com](mailto:yuli.if@gmail.com)

## Abstract

*An organization requires an integration if composed of members who live in different places. Organization requires the automation of business processes by exchanging business documents from applications and system platforms are different, both inside or outside the area of the organization. In this study, an application will be prepared by applying web service technology utilizing REST Web Service Architecture, where the server and client can interact with a unified interface, server and client will mengghost resource to consume resources provided by the server. Research is organized so that the business processes that occur when a consumer (client) do a query or search items, integrated with web applications. Server provides a client API which is then utilized. Client in this application is a web application. After receiving data from the client, the server and then disseminate the information needs of the goods in question to all members.*

**Keywords :** *Web Services, REST, Arsitektur, Client, Server*

Sebuah organisasi membutuhkan sebuah integrasi apabila terdiri dari para anggota yang berdomisili di tempat yang berbeda. Organisasi membutuhkan otomatisasi bisnis proses dengan melakukan pertukaran dokumen bisnis dari aplikasi dan platform sistem yang berbeda, baik di dalam atau pun di luar area organisasi. Pada penelitian ini akan disusun sebuah aplikasi dengan menerapkan teknologi *web service* memanfaatkan *REST Web Service Arsitektur*, dimana server dan client dapat berinteraksi dengan interface yang seragam, server akan mengghost resource dan client mengkonsumsikan resource yang disediakan oleh server. Penelitian ini disusun agar proses bisnis yang terjadi pada saat seorang konsumen (client) melakukan permintaan atau pencarian barang, terintegrasi dengan aplikasi web. Server menyediakan API yang kemudian dimanfaatkan client. Client pada aplikasi ini adalah aplikasi web. Setelah menerima data dari client, server kemudian menyebarkan informasi kebutuhan barang yang bersangkutan kepada seluruh member.

**Kata Kunci:** *Web service, REST, Arsitektur, client, server.*

## 1. PENDAHULUAN

Konsep teknologi *Web Service* muncul untuk mendukung sistem terdistribusi yang berjalan pada infrastruktur yang berbeda. *XML* merupakan suatu format dokumen dengan berbasis teks. Dengan menggunakan format dokumen *XML*, *Web service* memungkinkan menjanjikan banyak kemudahan dan perbaikkan dalam mendukung integrasi berbagai platform sistem dan aplikasi, baik melalui infrastruktur intranet maupun internet.

Iklan baris merupakan konsumsi yang ekonomis bagi para perusahaan atau UKM. Iklan jenis ini hanya memiliki pesan berupa teks sehingga tidak cocok untuk jenis iklan yang menawarkan atau mencari sebuah barang, maka kegiatan yang mungkin diiklankan hanya berupa perdagangan jasa informasi. Media iklan baris biasanya terpasang di media Koran atau surat kabar. Menurut beberapa orang atau perusahaan yang pernah melakukan iklan produknya baik barang/jasa melalui iklan baris, ada suatu jenis manfaat tertentu yang didapat dari jenis iklan ini, diantaranya mengenai biaya yang relative lebih efisien dibandingkan beriklan dengan bentuk yang lain. Dari sisi pembaca iklan, membaca iklan baris adalah suatu hal yang efektif dari segi waktu akan kebutuhan akan mencari barang/jasa. Dan harga koranpun saat ini sudah sangat terjangkau dengan Rp.1000,- pencarian iklan baris akan bias dilakukan dengan mudah. Akan tetapi sekarang ini sudah menjadi kebiasaan banyak orang untuk melakukan pencarian berita termasuk iklan tadi melalui *online*. Sudah barang tentu keberadaan iklan baris

dalam media surat kabar cetak kertas menjadi suatu hal yang kurang diminati, sehingga untuk mengkomunikasikan antara pengiklan dan yang mencari iklan perlu ada media yang dapat dijadikan wadah. Dan setiap biro iklan memiliki aplikasi pengolahan data dengan berbagai platform.

Bertitik tolak dari hal tersebut, pada penelitian ini akan merancang suatu aplikasi dengan menggunakan arsitektur *REST web service*. Penelitian ini diterapkan pada sistem iklan baris *online* pada beberapa biro jasa yang memiliki beberapa cabang dan klien yang berada di tempat yang saling terpisah.

## 2. DASAR TEORI

Pada bagian ini akan dijelaskan landasan teori mengenai hal yang berkaitan dengan teknologi *Web Service*, *XML*, dan *REST*.

### 2.1 Web Service

*Web Service* merupakan suatu komponen software yang merupakan *self-containing*, aplikasi modular *self-describing* yang dapat dipublikasikan, dialokasikan, dan dilaksanakan pada *web*. (Wahli dkk, 2006) *Web Service* adalah teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan *transactional web*, yaitu kemampuan *web* untuk saling komunikasi dengan pola *program to program* (P2P). Fokus *web* selama ini didominasi oleh komunikasi *program to user* dengan interaksi *business to customer* (B2C), sedangkan *stransactional web* akan didominasi oleh P2P dengan interaksi *business to business*. (Gottschalk, 2002)

*Web service* dapat didefinisikan sebagai sekumpulan fungsionalitas yang dapat diakses melalui *Internet Protocol* (IP) standar. *Web service* banyak digunakan untuk aplikasi sistem yang terdistribusi secara heterogen baik dari sistem operasi sampai model objek. *Web service* merupakan entitas yang dapat diprogram dan menyediakan beberapa macam fungsi seperti informasi, aplikasi logik dan diakses banyak *platform* melalui beberapa media yaitu *Hypertext Transfer Protocol* (HTTP) dan *XML*.

Kekuatan dari *web service* adalah menggunakan model *messaging* berbasis teks untuk komunikasi yang memungkinkan dapat beroperasi secara efektif pada *platform* yang berbeda. Oleh karena itu, *web service* banyak digunakan dan diakui secara luas. Pada dasarnya teknik dari *web service* adalah membuat sebuah pelayanan (*service*) terhadap segala masukan dari *client*. Dalam hal ini *web service* dikembangkan dalam area ruang kerja berbasis *web*.

Konsep teknologi *web service* muncul untuk mendukung sistem terdistribusi yang berjalan pada infrastruktur yang berbeda. Dengan adanya kombinasi dari *eXtensible Markup Language* (XML) dan *HyperText Transport Protocol* (HTTP), *web service* yang berbasis XML sangat mungkin untuk diimplementasikan. Bahkan teknologi *web service* ini mampu menggantikan fungsi dari beberapa teknologi yang serupa yang telah berkembang yaitu *Common Object Request Broker Architecture* (CORBA), *Dynamic Component Object Model* (DCOM), dan *Java Remote Method Invocation* (RMI).

### 2.2 XML

*XML* merupakan dasar terbentuknya *web service* yang digunakan untuk mendeskripsikan data. Pada level paling detail *web service* secara keseluruhan dibenuk di atas *XML*. Fungsi utama dari *XML* adalah komunikasi antar aplikasi, integrasi data dan komunikasi aplikasi eksternal dengan partner luaran. Dengan standarisasi *XML*, aplikasi-aplikasi yang berbeda dapat mudah berkomunikasi antar satu dengan yang lain. (Wahli, dkk, 2006)

Saat ini *XML* dapat dikatakan telah menggantikan *HTML*. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau *HTML* digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, *XML* mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. *XML* terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standar layaknya *heading*, *paragraph*, *table* dan lain sebagainya.

### 2.3 Representational State Transfer (REST)

*REST* adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. (Tidwell, D., 2001) Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula *REST* bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah

HTTP yang bisa digunakan adalah fungsi GET, POST, PUT atau DELETE. Balasan yang dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi *client*.

*REST* merupakan salah satu teknologi *web service* yang terbilang cukup populer di masa sekarang ini. Teknologi ini bekerja berdasarkan *resource* untuk membuat sistem terdistribusi. *REST* (disebut juga *RESTful services*) adalah perangkat lunak yang didesain dengan penekanan pada kesederhanaan, skalabilitas, serta kegunaan.

Dalam pengaplikasiannya, *REST* lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktifitas yang mengolah *resource* itu. Alasan mengapa *REST* tidak digunakan dalam penelitian ini karena orientasi pada *resource*nya itu, sedangkan aplikasi *event calendar* membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan *resource event*. Selain itu, karena standarnya yang kurang sehingga tidak begitu cocok diterapkan dalam aplikasi yang membutuhkan kerjasama antar aplikasi lain, dimana standar yang baik akan sangat berguna karena berbicara dalam satu bahasa yang sama. Beberapa contoh *web service* yang menggunakan *REST* adalah: Flickr API (*Application Program Interface*), YouTube API, Amazon API.

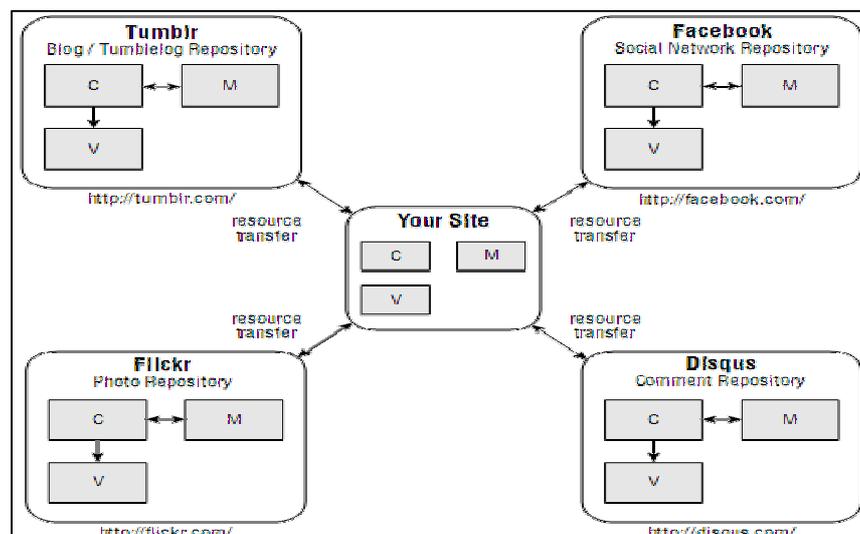
Menurut Pautasso 2008, metode *REST* didasari oleh empat prinsip utama teknologi yaitu :

- 1) *Resource identifier* melalui *Uniform Resource Identifier (URI)*, *REST Web service* mencari sekumpulan sumber daya yang mengidentifikasi interaksi antar klien.
- 2) *Uniform interface*, sumber daya yang dimanipulasi *CRUD (Create, Read, Update, Delete)* menggunakan operasi *PUT, GET, POST, dan DELETE*.
- 3) *Self-descriptive messages*, sumberdaya informasi tidak terikat, sehingga dapat mengakses berbagai format konten (*HTML, XML, PDF, JPEG, Plain Text dan lainnya*). *Metadata* pun dapat digunakan.
- 4) *Stateful interactions* melalui *hyperlinks*, setiap interaksi dengan suatu sumber daya bersifat *stateless*, yaitu *request messages* tergantung jenis kontennya.

### 2.3.1 Arsitektur REST

*REST* merupakan penyederhanaan dari *HTTP*. Dengan pertumbuhan web yang semakin populer, banyak keputusan desain asli yang memandu *HTTP* diabaikan. (Tidwell, D., 2001) Para pengembang aplikasi web cenderung melihat hal-hal seperti *verb HTTP* dan kode status respon sebagai sesuatu yang *incidental* untuk aplikasi, atau sebagai suatu hal tidak penting yang akan ditangani jika waktu masih mengijinkan. Penggunaan *HTTP* sebagaimana yang diharapkan, sering terlihat sebagai sesuatu yang tidak diperlukan atau menyulitkan. Namun, dalam beberapa tahun belakangan ini, dengan hadirnya kembali prinsip-prinsip *REST* telah mengindikasikan bahwasannya *HTTP* telah lebih dari cukup baik di atas segalanya.

*REST* merupakan cara baru berpikir tentang arsitektur jaringan berdasarkan pengamatan atas bagaimana jaringan bekerja.



Gambar 1. Web dengan Gaya Arsitektur REST (Benson, 2008)

### 2.3.2 API Flexibility & Simplicity

Kunci metodologi *REST* adalah untuk menulis *web service* menggunakan antarmuka yang sudah tersedia dan banyak digunakan : URI. Sebagai contoh, service/layanan untuk mengkonversi mata uang, yang mana seorang user memasukkan simbol mata uang untuk mengembalikan harga mata uang secara *real-time*, dapat dilakukan semudah membuat *script* yang dapat diakses melalui web server seperti

URI: [http://www.ExampleCurrencyBrokerage.com/convert?=usdollar&value=100&target=pound](http://www.ExampleCurrencyBrokerage.com/convert?=<u>usdollar&value=100&target=pound</u>).

Aplikasi *client* atau *server* dengan dukungan *HTTP* dapat dengan mudah memanggil *service* tersebut dengan *command HTTP GET*. Berdasar pada bagaimana cara penyedia *service* menulis *script*, hasil *respons HTTP* kan menjadi lebih simpel seperti beberapa *header standar* dan *string teks* yang mengandung harga terkini untuk symbol yang diberikan. Atau, dapat berupa dokumen *XML*.

### 2.3.3 Keuntungan Arsitektur REST

Beberapa keuntungan penggunaan arsitektur REST adalah (Tidwell, D., 2001) :

#### a. Penyederhanaan Konsep

Dasar dari REST adalah kesederhaan. Keputusan untuk menggunakan seperangkat *verb* standar (apakah menggunakan *verb HTTP* ataupun seperangkat *verb* lainnya) secara *virtual* mengeleminasi seluruh daerah diskusi. Registrasi yang seragam dan sistem penamaan *MIME type* mungkin tidak memenuhi perdebatan, tapi secara pasti menyederhanakan.

Dengan demikian sudut dari segitiga tersebut mendapat perhatian, secara potensial area abu-abu terbesar adalah mengidentifikasi dan menamai *resource*. Penamaan adalah salah satu area dimana kesederhanaan benar-benar dibayar mahal, karena sangat mudah menjadikannya keliru. Akan tetapi, jika kita benar-benar mentaati pemakaian seperangkat *verb* standard dan tipe konten, maka hal itu akan membantu konstrain pemilihan *noun*.

#### b. Ketahanan dari perubahan

Keuntungan lainnya yang didapat dari desain RESTful adalah desain cenderung lebih tahan terhadap perubahan daripada antarmuka bergaya Remote Procedure Call (RPC). Desain RESTful membawa keputusan arsitektural ke dalam *noun*. Pemilihan *noun* cenderung bersifat *domain-driven*, sementara antarmuka RPC cenderung lebih *implementation-driven*, karena prosedurnya (detail implementasi) diperlihatkan sebagai bagian dari antarmuka. Manakala antarmuka RPC sering membutuhkan lapisan tambahan untuk memisahkan antarmuka dengan implementasi, REST mengusahakan pemisahan antarmuka dengan implementasi dengan pengajuran antarmuka yang lebih abstrak.

Juga, dikarenakan REST membedakan ide "*resource*" dari "*representasi*", adalah mudah untuk menambah tipe konten baru yang mempresentasikan *resource* yang sama sebagai format baru yang diperlukan. Tidak ada perubahan arsitektural yang diperlukan, karena REST didasarkan pada pemisahan antara *resource* abstrak dan representasinya.

#### c. Keseragaman

Salah satu keuntungan yang diberikan oleh REST adalah antarmuka yang seragam. *Verb* (metode HTTP) secara universal seragam, di seluruh domain aplikasi. Tipe konten, walaupun tidak universal (akan berbeda di sepanjang domain), sudah distandarisasi dan relatif terkenal.

Fakta bahwa pengembang dibatasi pada seperangkat kecil metode mungkin terlihat memaksa, tetapi dalam prakteknya bukanlah hal yang perlu menjadi perhatian. Apa saja yang ingin dimodelkan dapat dengan mudah disusun dalam bentuk operasi CRUD. Dengan cara berpikir seperti ini, membantu untuk mendorong kompleksitas yang esensi ke dalam salah satu bagian arsitektur dimana mudah memperlakukannya. Biasanya, ketika nampak diperlukan lebih dari metode dasar, akan ada entri *resource* lain yang tersembunyi di dalam model yang menunggu untuk diekstrak.

### 2.3.4 Representasi JSON (JavaScript Object Notation)

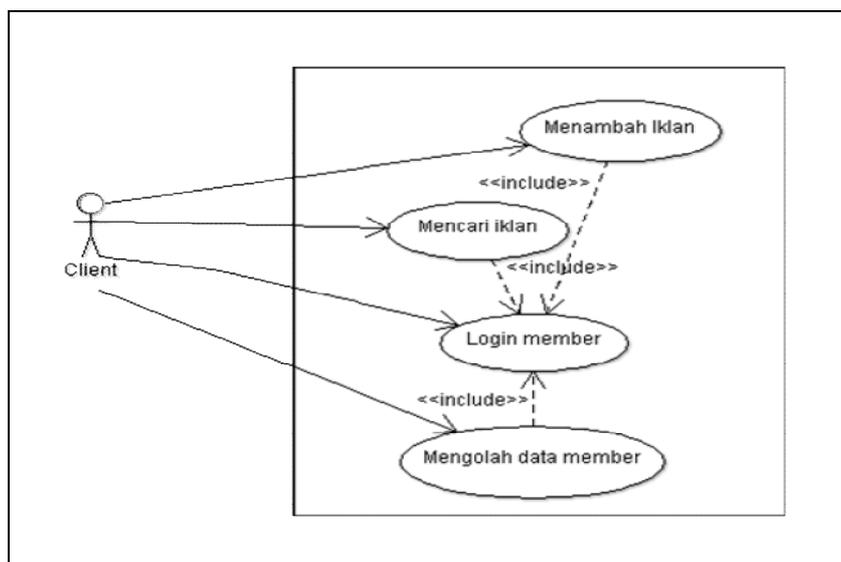
JSON (JavaScript Object Notation) adalah format pertukaran data yang kecil. Format ini sangat mudah untuk dimengerti dan juga bagi sebuah mesin dapat dengan mudah untuk membaca dan menghasilkan. Representasi JSON digunakan nama kunci yang sama yaitu dua tipe sumber: *users* dan *messages*. Sekali lagi, struktur-struktur ini adalah spesifikasi dari apa yang dibutuhkan untuk pengembalian tiap permintaan.

---

### 3. PERANCANGAN SISTEM

Sistem yang akan dikembangkan dalam penelitian ini adalah sistem berbasis *web* yang dirancang untuk berfungsi dalam menangani aplikasi iklan baris *online* yang menggunakan aritektur *REST web service*. Aplikasi pada sistem terbagi menjadi 2 jenis aplikasi yaitu Aplikasi pada server ditangani oleh administrator sistem dan aplikasi yang terpasang pada client, yang merupakan user yang telah mendaftar sebagai member aplikasi.

Metode yang digunakan untuk pengembangan sistem dalam penelitian ini adalah menggunakan metode berorientasi objek menggunakan kerangka GRAPPLE. Penelitian ini disusun agar proses bisnis yang terjadi pada saat seorang konsumen (client) melakukan permintaan atau pencarian barang, terintegrasi dengan aplikasi web. Server menyediakan API yang kemudian dimanfaatkan client. Client pada aplikasi ini adalah aplikasi web. Setelah menerima data dari client, server kemudian menyebarkan informasi kebutuhan barang yang bersangkutan kepada seluruh member.



Gambar 2. Diagram Usecase Aplikasi Iklan Baris Online

### 4. HASIL DAN PEMBAHASAN

Aplikasi Iklan Baris Online ini merupakan aplikasi yang mengimplementasikan arsitektur REST Web Service. Server dan client berinteraksi dengan interface yang seragam dimana server meng-host resources dan client mengkonsumsi resource yang disediakan oleh server.

Pada sisi server untuk aplikasi Iklan Baris *Online* ini terdapat *database* dengan 3 buah table, yaitu table Daftar Iklan, table Member, dan table user. Kemudian juga terdapat file PHP yang siap diakses untuk mengambil data dari database MySQL, diantaranya :

- Cariproses.php : digunakan saat user akan mencari iklan baris
- Tambahproses.php : digunakan saat user akan menambah iklan baris
- Daftar.php : digunakan saat user akan mendaftarkan diri sebagai member
- Datamember.php dan updateprofile.php : digunakan untuk menampilkan data member
- Login.php : digunakan oleh aplikasi saat login

*Script request* yang dikirim dari *client* ke *server* adalah berbentuk alamat *url* yang memanggil file PHP untuk mengakses data dari dataase server. Dalam proses pengambilan data, *client* akan memanfaatkan API yang terdapat di *server*. *Method-method* yang dipanggil diantaranya :

- Get* daftar iklan
- Search*
- Login
- Post* daftar iklan
- JSON encode*

## 5. KESIMPULAN

Aplikasi *web* dapat diakses oleh *client* dalam memasukkan data iklan permintaan atau pencarian barang/jasa dan menambah iklan serta dapat melakukan olah data *member*. Sedangkan *server* menyediakan API yang kemudian dimanfaatkan *client*, Setelah menerima data dari *client*, *server* kemudian menyebarkan informasi kebutuhan barang/jasa yang bersangkutan kepada seluruh *member*.

## Daftar Pustaka

- Gottschalk, K., 2002, Introduction to Web Service Architecture, <http://www.research.ibm.com/jurnal/sj/412/gottschalk.pdf> (diakses 21 Januari 2013)
- Tidwell, D., 2001, Web Service: The Web's next Revolution, <http://www6.software.ibm.com/developerworks/education/wsbasic/wsbasic-a4.pdf> (diakses 20 Januari 2013)
- Wahli, U., Burroughs, O., Cline, O., Tung, L., 2006, Service Handbook for WebSphere Application Server 6.1, <http://www.redbooks.ibm.com/redbooks/pdfs/> (diakses 25 Januari 2013)
- \_\_\_\_\_, <https://www.ibm.com/developerworks/webservices/library/ws-restful/> (diakses 5 Februari 2013)
-