IMPLEMENTASI CLOUD COMPUTING DENGAN SOA(SERVICE ORIENTED ARCHITECTURE) PADA SISTEM INFORMASI AKADEMIK

Muhammad Zulficar

Teknik Informatika, Ilmu Komputer, Universitas Dian Nuswantoro, Jl. Nakula I No. 5-11, Semarang, 50131

muhammadzulficar@gmail.com

ABSTRAK: *Perangkat atau bahasa pemrograman dalam sebuah aplikasi adalah salah satu masalah yang dialami oleh para developer ketika mereka ingin mengembangkan aplikasi yang sudah dibuat sebelumnya, karena keterbatasan itu developer akhirnya tidak dapat mengembangkan aplikasi yang seharusnya dapat dikembangkan menjadi lebih optimal dan fungsional. Oleh sebab itu, penulis berinisiatif membuat suatu teknik webservice untuk perangkat lunak yang dimana dengan menggunakan API (Application Programming Interface) agar developer dapat mengembangkan suatu perangkat lunak dengan mudah tanpa berkendala dengan bahasa pemrograman yang bersangkutan dengan perangkat lunak yang ingin dikembangkan tersebut. Metode yang digunakan dalam pengembangan perangkat lunak ini adalah agile methodology, dan kerangka kerja web menggunakan Code Igniter.*

KATA KUNCI: *Cloud Computing, SOA, Web service, API, JSON*

ABSTRACT: *Device or in an application programming language is one of the problems experienced by developers when they want to develop an application that was made before, because of the limitations that developers finally can not develop applications that should be developed into a more optimal and functional. Therefore, the author took the initiative to create a webservice for software engineering which is which by using API (Application Programming Interface) so that developers can develop a software easily without any programming language is constrained by the software concerned with those who wish to develop. The method used int this software development is agile methodology and web frameworks using Code Igniter.*

KEYWORDS: *Cloud Computing, SOA, Web Services, APIs, JSON*

I. PENDAHULUAN

Dengan kemajuan teknologi informasi yang signifikan selama setengah abad terakhir ini, semakin dirasakannya bahwa komputas suatu hari nanti akan menjadi sangat berguna. Kegunaan komputasi ini akan memberikan tingkat dasar layanan komputasi yang dianggap penting untuk memenuhi kebutuhan sehari-hari masyarakat umum seperti, akan kebutuhannya informasi yang *up to date*, dokumentasi akan suatu aplikasi yang dikembangkan dan dibagikan ke orang lain atau ke dalam suatu grup dan bisa saja untuk pengolahan data local. Untuk menyampaikan visi ini, sejumlah paradigma komputasi telah diusulkan. Dimana yang terakhir ini dikenal sebagai komputasi awan[1](*Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, 2008)*.

SOA (*Service Oriented Architecture*) adalah Teknologi Informasi (TI) gaya arsitektur yang mendukung transformasi bisnis menjadi satu set layanan terkait atau tugas-tugas bisnis berulang yang bisa diakses ketika dibutuhkan melalui jaringan. Mungkin bias sebuah jaringan local, internet atau mungkin geografis dan teknologi yang beragam seolah-olah semua terinstal pada desktop local[2](*Iulia Surugiu, Manole Velicanu, 2008*).

Agar SOA (*Service Oriented Architecture*) dapat tergabung dengan computer, maka computer harus diberikan pengetahuan dasar tentang pola dasar *web service*. Sebenarnya sudah banyak peneliti yang melakukan penelitian tentang *Service Oriented Architecture* dan *web service*, namun kebanyakan dari penelitian tersebut menggunakan bahasa Inggris. Sebagai contoh, penelitian yang dilakukan oleh Iulia Surugiu dan Manole Velicanu pada tahun 2008. Pada penelitian tersebut, *Service Oriented Architecture* dan *web service* direpresentasikan pada *Business Process Management* dengan menggunakan *Enterprise Application Integration*.

Informasi akan perkembangan teknologi informasi yang ada, para pengembang perangkat lunak (*software*) harus selalu mengikuti perkembangan dan melakukan uji coba maupun pengembangan suatu perangkat lunak. Informasi teknologi tidak hanya bergantung dengan bahasa pemrograman data saja, namun data juga penting untuk kebutuhan informasi yang diperlukan oleh *client/user/*masyarakat umum yang membutuhkan informasi tersebut. Karena setiap pengembang perangkat lunak berbeda dalam keahlian dan kemampuannya dan terbatasnya data yang dimana data yang dibutuhkan adalah data yang sangat privasi, maka akan menjadi kendala bagi para pengembang perangkat lunak untuk membuat maupun mengembangkan perangkat lunak untuk memberikan suatu informasi kepada *user*.

Berdasarkan uraian tersebut, maka penulis melakukan penelitian untuk Tugas Akhir yang diberi judul "**Implementasi Cloud Computing Dengan SOA(Service Oriented Architecture) Pada Sistem Informasi Akademik".**

II. LANDASAN TEORI

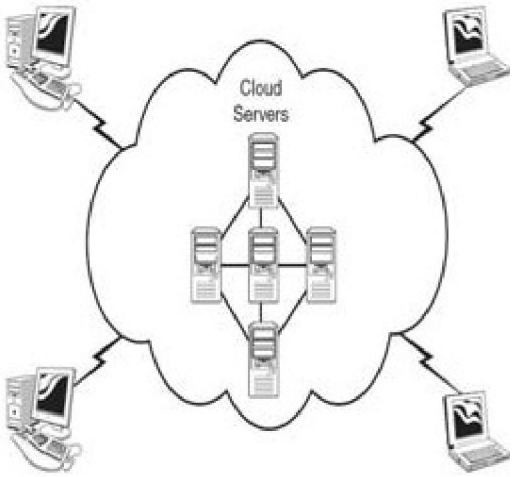
2.1 Cloud Computing

Cloud computing adalah gabungan pemanfaatan teknologi computer (komputasi) dan pengembangan berbasis internet (awan). *Awan (cloud)* adalah metafora dari internet, sebagaimana awan dalam diagram jaringan computer tersebut, *awan(cloud)* dalam cloud computing juga merupakan abstraksi dari infrastruktur kompleks yang disembunyikannya.[1](*Mariana Carroll, Paula Kotzé, Alta van der Merwe, 2012*)

Cloud computing adalah metode komputasi dimana kapabilitas terkait teknologi informasi disajikan sebagai suatu layanan(service), sehingga pengguna dapat mengkasesnya lewat internet tanpa mengetahui apa yang ada di dalamnya atau infrastruktur teknologi yang membantunya.

Menurut sebuah makalah tahun 2008 yang dipublikasi *IEEE Internet Computing* "Cloud Computing adalah suatu paradigm dimana informasi secara permanen tersimpan di server di internet dan tersimpan secara sementara di computer pengguna(*client*)" termasuk di dalamnya adalah desktop, computer table, notebook, handheld, monitor dan lain-lain.

Komputasi awan adalah suatu konsep umum yang mencakup SaaS, Web 2.0, dan teknologi terbaru lainnya yang sudah dikenal luas, dengan tema umum berupa ketergantungan terhadap internet untuk memberikan kebutuhan komputasi pengguna. Sebagai contoh, Google APPS menyediakan aplikasi bisnis umum secara daring yang diakses melalui suatu penjelajah web dengan dengan perangkat lunak dan data yang tersimpan di server. Komputasi awan saat ini merupakan teknologi terbaru dan contoh bentuk pengembangan dari teknologi Cloud Computing ini adalah iCloud.



Gambar 2.1 Cloud Computing Architecture (*Shivaji P Mirashe, Dr. N.V. Kalyankar, 2010*)

2.1.1 **Sifat Cloud Computing**

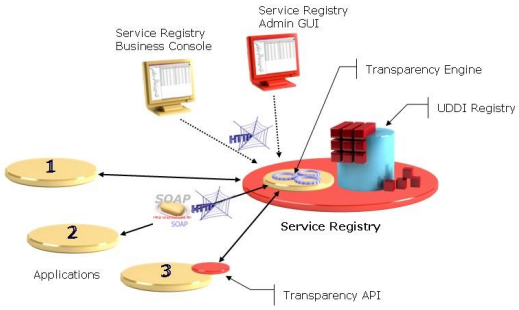
Dari sudut pandang Google, ada enam sifat utama yang dimiliki Cloud Computing yaitu:

1. Cloud Computing adalah user-centric Ketika anda menjadi sebagai pengguna cloud computing, apapun yang disimpan di dalam *awan* dalam bentuk dokumen, pesan, gambar, aplikasi apapun menjadi milik anda. Selain itu tidak hanya data anda saja, tetapi anda juga dapat membagikannya ke orang lain. Namun perangkat apapun yang anda akses yang ada di *awan* juga menjadi milik anda.
2. Cloud Computing adalah task-centric Bukan hanya focus pada aplikasi dan apa yang dapat dilakukan, fokusnya kepada apa yang anda butuh lakukan dan bagaimana aplikasi dapat melakukannya untuk anda. Aplikasi tradisional – pengolahan kata, lembar kerja, email dan sebagainya menjadi kurang penting dari dokumen yang mereka buat.
3. Cloud Computing adalah powerful Dengan terhubungnya ratusan atau ribuan computer secara bersama-sama di *awan* membuat kekayaan daya komputasi tidak mungkin hanya dengan satu desktop PC. Cloud Computing dapat diakses karena data disimpan di *awan*, pengguna dapat langsung mengambil informasi dari beberapa repository. Anda tidak akan terbatas untuk mengakses ke sumber data, seperti halnya anda dengan desktop PC anda.
4. Cloud Computing adalah Intelegent Dengan semua berbagai data yang tersimpan di komputasi awan, pengolahan dan analisis data diperlukan untuk mengakses informasi secara cerdas (*intelegent*).
5. Cloud Computing adalah Programmable Banyak tugas yang perlu di-otomatisasi oleh komputasi awan, misalnya untuk melindungi integritas data. Informasi disimpan pada satu computer di *awan* dan harus

diduplikat di computer di *awan*. Jika satu computer offline (tidak aktif), maka pemrograman *awan* secara otomatis mendistribusikan data computer ke computer baru di *awan*.[2](*Shivaji P. Mirashe, Dr. N.V. Kalyankar, 2010*)

2.2 SOA (Service Oriented Architecture)

SOA (*Service Oriented Architecture*) adalah suatu metode arsitektur system yang membuat dan menggunakan proses bisnis dalam bentuk kumpulan-kumpulan layanan. SOA juga mendefinisikan dan menentukan arsitektur yang dapat menunjang berbagai aplikasi untuk saling bertukar data dan dapat mengembangkan aplikasi dengan proses bisnis yang lain. Fungsi-fungsi yang ada tidak terikat dengan system operasi dan bahasa pemrograman yang mendasari aplikasi-aplikasi tersebut.[3](*Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, 2010*)



Gambar 2.2 Virtualization engine (*Iulia Surugiu, Manole Velicanu, 2008*)

2.2.1 Karakteristik SOA(Service Oriented Architecture)

Suatu konsep biasanya dikenal dari karakteristiknya, dan karakteristik dari SOA adalah :

I. *Logical view*, service yang dilihat dari level operasi bisnis yang diidentifikasi sebagai *interface* yang independen.
II. *Message orientation*, sebuah service yang berhubungan dengan client yang bertuka message.
III. *Description orientation*, service dituntut untuk dipakai di dalam jaringan. Hal ini menekankan pada kebutuhan service yang secara otomatis dan mudah ditemukan.
IV. *Platform neutrality*, pesan disampaikan melalui interface yang menggunakan platform netral (*multi platform*) dan format data yang standart seperti XML.
    I. *Service interface*, menyatakan bagaimana service tersebut dapat dipanggil seperti parameter input / output.
    II. *Service implementation*, sangat terkati dengan teknologi pemrograman yang digunakan. SOA tidak perlu memperdulikan bagaimana sebuah service diimplementasikan.

III. *Service business oriented*, setiap service yang didefinisikan harus melakukan suatu aktifitas bisnis tertentu.

2.1.2 Keunggulan Service Oriented Architecture

Ada beberapa kelebihan yang ditawarkan oleh SOA :

I. Dapat menyatukan berbagai platform yang berbeda.
II. Tahan terhadap perubahan teknologi.

2.3 Web Service

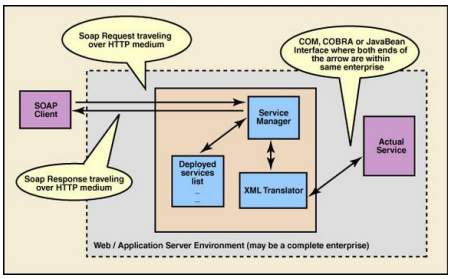
Web service adalah sebuah teknik yang dirancang untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan.[4] (*David Booth, Hugo Haas, Francis McCabe, Michael Champion, Chris Ferris, David Orchard; 2004*)

Web service secara teknis memiliki mekanisme interkasi antar system sebagai penunjang interoperabilitas, baik berupa agregasi (pengumpulan) maupun sindikasi

(penyatuan). *Web service* memiliki layanan terbuka untuk kepentingan integrasi data dan kolaborasi informasi yang bias diakses melalui internet oleh berbagai pihak menggunakan teknologi yang dimiliki oleh masing-masing pengguna.

Untuk dapat menggunakan layanan *web service*, maka *web service* dapat dipanggil menggunakan beberapa model messaging:

1. **SOAP**
   SOAP (*Simple Object Access Protocol*) adalah standar untuk bertukar document atau pesan berbasis XML melalui jaringan computer atau sebuah jalan untuk program yang berjalan pada suatu system operasi untuk berkomunikasi dengan program pada OS yang sama maupun berbeda dengan menggunakan HTTP dan XML sebagai mekanisme untuk pertukaran data.[5](*Martin Gudgin, Marc Hadley and team, 2007*)



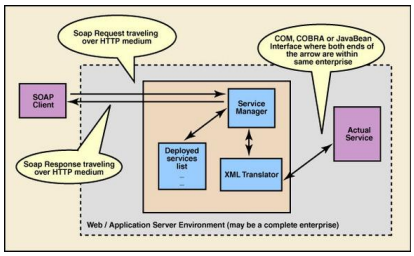
Gambar 2.4 Arsitektur SOAP (*Bilal Siddiqui, 2002*)

2.3.1 SOAP (*Simple Object Access Protocol*)

SOAP (*Simple Object Access Protocol*) adalah standar untuk bertukar document atau pesan berbasis XML melalui jaringan computer atau sebuah jalan untuk program yang berjalan pada suatu system operasi untuk berkomunikasi dengan program pada OS yang sama maupun berbeda dengan menggunakan HTTP dan XML sebagai mekanisme untuk pertukaran data.

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

SOAP adalah protocol ringan untuk pertukaran informasi struktur pada lingkup desentralisasi, dan terdistribusi. SOAP menggunakan teknologi XML untuk mendefinisikan kerangka kerja dimana yang menyediakan konstruksi pesan yang dapat dipertukarkan dengan protocol yang berbeda.



Gambar 2.7 Arsitektur SOAP(*Bilal Siddiqui, 2002*)

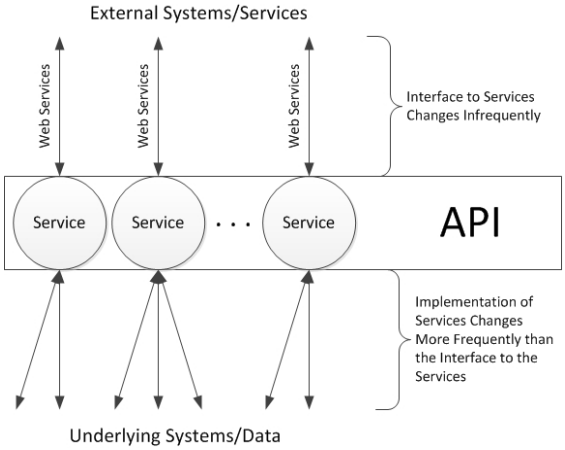
2.3.1.1 Keuntungan SOAP

1. SOAP cukup fleksibel untuk memungkinkan penggunaan protocol yang berbeda. HTTP menjadi sebagai standar protocol, tetapi protocol lain seperti SMTP juga dapat digunakan.
2. SOAP dapat dengan mudah melewati firewall dan proxy yang ada, tanpa modifikasi pada protocol SOAP dan dapat menggnakan infrastruktur yang ada.

2.3.1.2 **Kekurangan SOAP**

1. SOAP lebih lambat dibandingkan dari model-model messaging *web service* lainnya, karena panjang format XML yang telah mengikuti dan parsing yang menyelimuti yang diperlukan.

2. Ketika mengandalkan HTTP sebagai protocol transport dan tidak menggunakan WS-Addressing atau ESB, peran dari pihak interaksi akan tetap. Hanya satu kelompok *client* yang dapat menggunakan service lainnya.



Gambar 2.8 Struktur API (*Roger S. Pressman, 1982*)

2.4 API (*Application Programming Interface*)

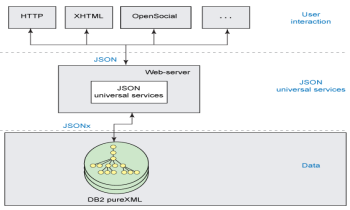
API (*Application Programming Interface*) adalah sekumpulan perintah, fungsi dan protocol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk system operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan system operasi.

API dapat menjelaskan cara sebuah tugas tertentu dilakukan. Dalam pemrograman procedural seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi yang disediakan.

Ketika digunakan dalam konteks pengembangan web, API biasanya didefinisikan sebagai satu pasangan HTTP request. API memiliki struktur format pesan, biasanya dalam bentuk XML atau JSON format.

2.4.1 JSON (*Javascript Object Notation*)

JSON (*Javascript Object Nation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh computer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman javascript, standar ECMA-262 Edisi ke-3-Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C. Oleh karena sifat-sifat tersebut menjadikan JSON ideal sebagai bahasa pertukaran data.[8] (*Atif Aziz, Scott Mitchell, 2007*)



Gambar 2.9 struktur JSON (*Nuno Job, Susan Malaika, Michael Schenker, 2010*)

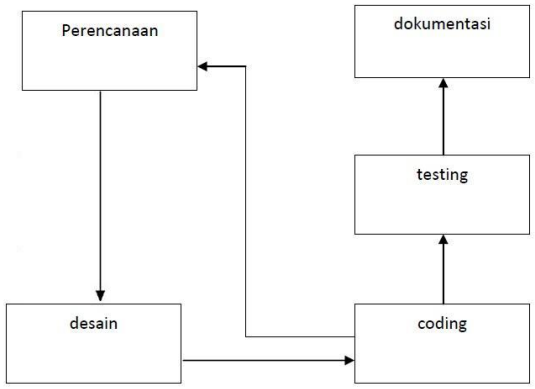
2.4.1.1 **Struktur JSON**

1. Kumpulan pasangan/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek(*object*), rekaman(*record*), struktur(*struct*), kamus(*dictionary*), tabel hash(*hash table*), daftar kunci(*keyed list*) atau *associative array*.

2.6 Agile Methodology

Agile Software development adalah salah satu metodologi dalam pengembangan sebuah perangkat lunak. Kata Agile berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata ini digunakan sebagai kata yang menggambarkan konsep model proses yang berbeda dari konsep model-model proses yang sudah ada. Konsep Agile software development dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa Agile Software Development adalah cara membangun software dengan melakukannya dan membantu orang lain membangunnya sekaligus.

Dalam Agile Software Development proses saling mempengaruhi dalam hubungan timbal balik (interaksi) lebih penting dari pada proses yg memebutuhkan banyak alat, software yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klient lebih penting daripada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana. Agile Software Development memungkinkan model proses yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat cepat ditanggapi, namun di sisi lain menyebabkan produktivitas menurun. Software dikerjakan dengan menggunakan metode Agile, maka selama waktu pengerjaannya akan selalu dijumpai proses pengembangan yang dilakukan berulang. Setiap perulangan (iterasi) meliputi berbagai kegiatan yang wajib dilakukan dalam proyek pengembangan software itu sendiri yaitu :



Gambar 2.10 Agile Process Model

2.6.1 **Perencanaan**

Perencanaan merupakan struktur awal untuk melakukan pembuatan system yang akan dibuat, perencanaan perlu dilakukan agar pengembang dapat memahami apa yang akan dibuat sebelum melakukan penulisan script atau desain system.

2.6.2 **Analisa Sistem**

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pada tahahapan ini akan digali sebanyak-banyaknya informasi yang mungkin akan dibutuhkan sistem. Dokumen ini disebut *user requirement*, dari dokumen inilah yang akan menjadi acuan sistem analis untuk menterjemahkan ke dalam bahasa pemprogram.

2.6.3 **Desain**

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada : arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirment*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan aplikasi.

2.6.4 **Coding**

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan aksi yang diinginkan oleh user. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan. Dalam hal ini penulis menggunakan bahasa pemprograman PHP sebagai bahasa pemprograman yang digunakan pada studi kasus ini.

2.6.5 **Testing**

Testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

2.6.6 **Dokumentasi**

Merupakan bagian penting dari pengembangan perangkat lunak. Masing-masing tahapan dalam model biasanya menghasilkan sejumlah tulisan, diagram, gambar atau bentuk-bentuk lain yang harus didokumentasi dan merupakan bagian tak terpisahkan dari perangkat lunak yang dihasilkan.

III. METODOLOGI PENELITIAN

3.1 **Objek Penelitian**

Dalam tugas akhir ini penulis memilih SOA(*Service Oriented Architecture*) sebagai objek penelitian dimana penelitian yg dilakukan ini berfokus pada implementasi *web service* menggunakan bahasa PHP pada system informasi akademik.

3.2 **Fokus Penelitian**

Fokus penelitian adalah sesuatu yang menjadi hal utama untuk dijadikan bahan penelitian. Penelitian yang di kerjakan di sini adalah membuat aplikasi yang menerapkan *web service* dengan menggunakan bahasa PHP.

3.3 **Ruang Lingkup Penelitian**

Agar penelitian dapat terfokus dan terarah, maka perlu adanya ruang lingkup yang digunakan sebagai pedoman dalam melaksanakan penelitian. Ruang lingkup penelitian ini adalah *share* data pada aplikasi web dengan menggunakan *web service*.

3.4 Prosedur Pengambilan atau Pengumpulan Data

Kualitas data tidak hanya di tentukan oleh reliabilitas dan validitas dari alat ukuranya saja, tetapi juga di tentukan oleh bagaimana cara pengumpulannya, beberapa aspek dalam proses mengumpulkan data yang digunakan adalah sebagai berikut:

I. Data apa yg akan dikumpulkan(what)
II. Dengan apa data itu dikumpulkan(with)
III. Darimana data akan dikumpulkan(where)
IV. Kapan data tersebt akan dikumpulkan(when)
V. Bagaimana cara mengumpulkan(how)

Sedangkandalam pengumpulan data pada penelitian ini dengan menggunakan beberapa metode,yaitu studi pustaka.

Studi pustaka merupakan metode pengumpulan data dengan cara mencari informasi melalui buku-buku, internet, Koran, majalah dan literatur lainnya.

3.4.1. Jenis Data

Jenis data yang digunakan dalam penyusunan penelitian ini adalah kualitatif yaitu prosedur penelitian yang menghasilkan data tidak berbentuk angka yang diperoleh dari rekaman, pengamatan, wawancara, atau bahan tertulis.

3.4.2. Sumber Data

Sumber data yang digunakan dalam penyusunan laporan tugas akhir ini adalah data sekunder ,yaitu data yang diperoleh secara tidak langsung artinya sumber-sumber yang secara tidak langsung misalnya buku-buku, laporan-laporan tertulis, dokumen-dokumen internet dan makalah serta daftar pustaka atau literatur lainya mendukung penelitian.

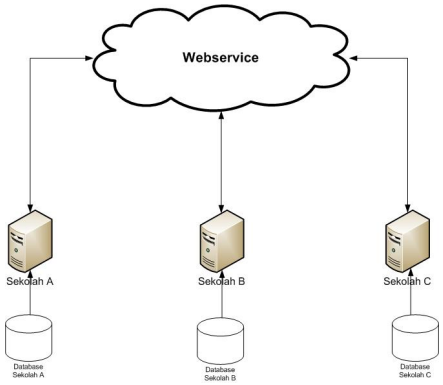
3.5 Analisa dan Perancangan Sistem

3.5.1. Analisa Kebutuhan Sistem

Tahap ini merupakan tahap awal yang merupakan visi dari software yang akan dibangun. Penulis merencanakan aplikasi yang akan dibuat, dalam hal ini aplikasi website dengan menerapkan *web service* dengan menggunakan PHP yang implementasinya akan dibatasi pada system informasi akademik. Aplikasi ini dibuat dengan memanfaatkan *web service* sebagai media perantara antara *client* dengan *server*, dimana *client* akan me-*request* data ke *server* namun sebenarnya *client* tidak langsung berinteraksi dengan *server* melainkan melalui *web service* dan akan dikembalikan lagi ke *client* melalui *web service*.

3.5.2. Arsitektur Sistem

Dari Analisa sistem tersebut diatas, penulis merancang arsitektur aplikasi sebagai berikut



Gambar 3.1 Arsitektur Webservice Aplikasi Sistem Akademik

Pada gambar 3.1 diatas, masing-masing aplikasi ini saling terhubung pada database tersendiri. Ketiga aplikasi server tersebut terintegrasi melalui media *web service* yang berjalan dalam koneksi jaringan internet.

.

3.5.3. Metode Pengembangan Sistem

Metode pengembangan system yang digunakan oleh penulis adalah dengan menggunakan model proses pengembangan perangkat lunak *agile* melalui paradigma / pendekatan berorientasi objek yang dimodelkan menggunakan *Unified Modeling Language* (UML).

Metode *agile* merupakan metode yang menyajikan gambaran yang lengkap tentang sistemnya, metode ini banyak digunakan karena pengembang mungkin tidak memiliki kepastian terhadap efisiensi algoritma, kemampuan penyesuaian dari sebuah sistem operasi, atau bentuk-bentuk yang harus dilakukan oleh interaksi manusia dengan mesin sehingga paradigma *agile* ini merupakan pendekatan terbaik yang ditawarkan.

Paradigma *agile* dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu untuk mendefinisikan obyektif kebutuhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area agris besar dimana definisi lebih jauh merupakan keharusan yang kemudian dilakukan perancangan kilat. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan terlihat bagi pelanggan. Perancangan kilat membentuk konstruksi sebuah *agile. Agile* tersebut dievaluasi oleh pelanggan dan digunakan untuk menyaring kebutuhan pengembangan perangkat lunak. Iterasi terjadi pada saat *agile* dirancang untuk memenuhi kebutuhan pelanggan dan pada saat yang sama memungkinkan pengembang untuk memahami apa yang akan dilakukan selanjutnya.

Keunggulan dari penggunaan paradigma *agile* adalah :

1. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan
2. Lebih menghemat waktu dalam pengembangan sistem

Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

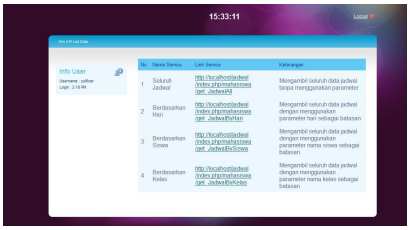
IV. PERANCANGAN DAN IMPLEMENTASI

4.1 Tabel Inputan Jadwal

Nama Sekolah	SMA BPI 1 Bandung
Nama Siswa	Zulficar
Nama Kelas	Kelas 3
Nama Wali Kelas	Budi R.H
Nama Matapelajaran	Matematika
Hari	Senin
Jam	09.00
Simpan	

Untuk desain insert_Jadwal(), dimana kegunaannya untuk memasukkan data jadwal dan dimana untuk data nama sekolah, nama siswa, nama kelas, nama wali kelas dan nama matapelajaran akan secara otomatis tampil di dalam form inputan yang ada. Hanya tinggal memasukkan text field hari dan jam yang telah tersedia di dalam form inputan

4.2 Gambar List API Jadwal



Gambar ini adalah halaman yang dimana setelah memilih menu "jadwal", maka akan menampilkan seluruh daftar list API yang berhubungan dengan jadwal. Dimana user dapat memilih link dengan sesuai yang dibutuhkan dan hanya *copy* lalu *paste* di dalam script aplikasi user.

Ketika user memilih link yang nama servicenya adalah "seluruh jadwal" maka akan menampilkan output data dengan format JSON seperti gambar di bawah ini.

V. KESIMPULAN

Kesimpulan yang dapat diberikan terhadap penyusunan laporan tugas akhir ini sebagai berikut:

1. Dengan menggunakan teknik *web service* data menjadi dapat diintegrasikan dengan aplikasi lain tanpa terkendala oleh bahasa pemrograman maupun *device* yang ada.
2. Pihak eksternal dapat mengambil dan melihat data yang ada di dalam database tanpa mengetahui struktur dari database tersebut

DAFTAR PUSTAKA

[1] Mariana Carroll, Paula Kotze, Alta Van Der Merwe (2012). "*Securing Virtual and Cloud Environments*". In I. Ivanov et al. *Cloud Computing and Services Science, Service Science: Research and Innovations in the Service Economy*. Springer Science+Business Media. DOI : 10.1007/978-1-4614-2326-3.

[2] Shivaji P. Mirashe, Dr. N.V. Kalyankar (March 2010). "*Cloud Computing*". *Journal of Computing*, Volume 2, Issue 3, ISSN: 2151-9617.

[3] Anthony T. Velte, Toby J. Velte, Ph.D, Robert Elsenpeter (2010). "*Cloud Computing*". ISBN: 978-0-07-162695-8, MHID: 0-07-162695-6.

[4] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, David Orchard (2004). "*Web Services Architecture*". W3C Working Group Note 11 February 2004. Diakses dari

http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/ pada 26 April 2013.

[5] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, Yves Lafon (27 April 2007). "*Messaging Framework (Second Edition)*", SOAP Version 1.2 Part 1. Diakses dari http://www.w3.org/TR/soap12-part1/#intro pada 27 April 2013.

[6] Roberto Chinnici, Jean-Jacques Moreau, Arthur Ryman, Sanjiva Weerawarana (26 June 2007). "*Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*". Diakses dari http://www.w3.org/TR/wsdl20/#intro pada 27 April 2013.

[7] Roy Thomas Fielding (2000). "*Architectural Styles and the Design of Network-based Software Architectures*". Disertasi yang diterbitkan. Irvine: University of California. Diakses dari http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm pada 27 April 2013.

[8] Atif Aziz, Scott Mitchel (2007). "*An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET*". Diakses dari http://msdn.microsoft.com/en-us/library/bb299886.aspx pada tanggal 28 April 2013.