

IMPLEMENTASI SERVICE-ORIENTED ARCHITECTURE MENGUNAKAN TEKNOLOGI WEB SERVICE UNTUK APLIKASI PENDATAAN PDTA SE-KOTA PEKANBARU

Ridho Hilmawan¹, Iwan Iskandar²

Jurusan Teknik Informatika, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. H.R. Soebrantas No.155 KM. 15 Simpang Baru Panam Pekanbaru
PO. Box 1004
Telp. (0761) 8359937, Faks. (0761) 859428
E-mail: ridho.tif@gmail.com¹, iwaniskandar01@gmail.com²

ABSTRAK

Pada saat ini Kementerian Agama kota Pekanbaru masih menerapkan proses manual terhadap pendataan PDTA (Pendidikan Diniyah Ta'miliah Awaliyah) se-kota Pekanbaru. Proses pengumpulan data menggunakan dokumen excel yang disebarakan ke setiap PDTA dan mengumpulkan kembali dokumen excel tersebut dalam batas waktu yang ditentukan. Proses penghimpunan kembali dokumen tersebut membutuhkan waktu yang lama. Data PDTA yang terkumpul akan diolah menjadi data statistik, namun data tersebut tidak dapat ditampilkan secara *realtime*. Pada penelitian ini diimplementasikan metode *Service Oriented Architecture* (SOA) menggunakan teknologi *web service* untuk menerapkan suatu sistem yang berbasiskan integrasi data. Penelitian ini menggunakan metodologi *Design Science Research* (DSR) yang memfokuskan pada pengembangan dan peningkatan kinerja model/desain dari sebuah sistem. Penerapan SOA menggunakan teknologi *web service* dilakukan dengan menghubungkan aplikasi Pendataan Kemenag dan beberapa aplikasi PDTA. Dari uji coba aplikasi berbasis SOA, beberapa instansi memperoleh keuntungan diantaranya proses penghimpunan data PDTA dapat dilakukan setiap saat tanpa terikat pada waktu tertentu. Selain itu, PDTA dapat memanfaatkan proses bisnis yang dimiliki Kemenag kedalam proses bisnis yang dimiliki PDTA seperti proses penghimpunan data PNS Kemenag.

Kata kunci : Integrasi data, Kemenag, Layanan, SOA (*Service Oriented Architecture*), *Web Service*.

1. PENDAHULUAN

Pengolahan data dan informasi merupakan salah satu bagian penting dalam suatu organisasi termasuk pendidikan. Dengan semakin kompleks dan berkembangnya suatu organisasi pendidikan, pengolahan dan integrasi data atau informasi menjadi isu yang mendapat perhatian serius. Proses-

proses bisnis terus mengalami perubahan mengikuti kebutuhan organisasi, mengakibatkan diperlukan pengembangan sistem dan aplikasi yang ada. Untuk kasus yang lebih kompleks diperlukan integrasi sistem dan aplikasi lainnya untuk menghadirkan proses dan aktifitas pendidikan yang cepat, berkualitas dan berdaya saing. Dalam perkembangannya organisasi dihadapkan dengan tantangan yang muncul diantaranya bagaimana menggunakan kembali data atau informasi lama pada aplikasi baru, membangun suatu antarmuka aplikasi independen yang bisa berjalan antar platform serta antar sistem aplikasi, dan membangun aplikasi yang bisa digunakan dimasa akan datang [1].

PDTA merupakan lembaga pendidikan Islam untuk anak usia dini yang berada dibawah naungan Kementerian Agama. Pada saat ini Kementerian Agama kota Pekanbaru mengalami kesulitan dalam melakukan pendataan terhadap seluruh PDTA yang ada di kota Pekanbaru. Sulitnya menghimpun data dari masing-masing PDTA mengakibatkan kurang optimalnya pihak Kementerian Agama dalam menyajikan data statistik PDTA tersebut. Selain itu, dibutuhkan juga berbagai layanan yang dapat menunjang proses bisnis antara pihak kementerian agama kota Pekanbaru dengan masing-masing PDTA. Seperti sebuah layanan yang dapat membantu administrator masing-masing PDTA dalam melakukan *entry* data pegawai PNS Kementerian Agama yang memiliki tugas kerja pada PDTA bersangkutan, dimana data pegawai PNS Kementerian Agama tersebut dapat digunakan kembali dari database Kemenag ke database PDTA bersangkutan.

Oleh karena itu, dibutuhkan suatu sistem aplikasi yang saling terintegrasi satu sama lainnya, dimana pihak Kementerian Agama kota Pekanbaru dapat menghimpun berbagai macam informasi dan data statistik dari masing-masing PDTA kota Pekanbaru secara *realtime* dari aplikasi yang dimiliki masing-masing PDTA. Selain itu juga, masing-masing PDTA dapat memperoleh layanan dari proses bisnis yang ada di Kementerian Agama, seperti sebuah

layanan untuk melakukan *entry* data pegawai PNS Kemenag yang dapat diperoleh oleh masing-masing PDTA. Oleh sebab itu, pada aplikasi tersebut diterapkan teknologi *web service* yang dapat mengurangi atau mengatasi permasalahan tersebut. Pada penelitian ini dibuat sebuah teknologi *web service* menggunakan metode SOA (*Service Oriented Architecture*).

SOA merupakan arsitektur sistem informasi yang mengemas beberapa bagian arsitektur aplikasi sebagai *service* [2]. Selain itu, metode ini dapat diterjemahkan berdasarkan beberapa elemen dimana setiap elemennya dapat saling berhubungan dengan berbagai rancangan dan gaya arsitektur yang berbeda. SOA secara umum terbangun berdasarkan beberapa karakteristik, yaitu berdasarkan layanan yang siap untuk diintegrasikan [3].

Web services merupakan kumpulan layanan yang disediakan melalui jaringan berbasis web dengan standar yang telah ditetapkan mampu menunjang interoperabilitas. Interoperabilitas dalam *web services* berarti layanan yang disediakan dapat dioperasikan oleh berbagai *standard* dan *platform* yang berbeda [4]. Dalam perkembangannya mengimplementasikan SOA menggunakan *Web Services* merupakan hal yang tepat, karena teknologi ini dapat mewakili sebuah layanan/service dalam SOA dan dengan penggunaan *web services* ini, potensi-potensi SOA baru dapat dimunculkan. Sehingga *web services* cocok untuk menerapkan konsep dan sifat-sifat SOA.

Pada penelitian terkait sebelumnya yang dilakukan oleh Hendro Joko Prasetyo yang berjudul “*Implementasi Service Oriented Architecture (SOA) Menggunakan Web Service*” menyatakan bahwa *Service Oriented Architecture (SOA)* merupakan konsep pembangunan perangkat lunak yang melakukan partisi sistemnya menjadi beberapa *service* yang dapat berdiri secara independent, sedangkan *web service* merupakan suatu aplikasi *web service* yang berkomunikasi dengan aplikasi *web service* lainnya dalam rangka pertukaran data.

Oleh karena itu implementasi *Service Oriented Architecture (SOA)* dengan menggunakan teknologi *web services* layak diterapkan dalam mengatasi permasalahan yang ada pada saat ini, serta dalam menghadapi tantangan global yang menuntut kecepatan akses informasi dan data yang *realtime* bagi pendataan dan pengolahan data statistik PDTA se-kota Pekanbaru.

2. SERVICE ORIENTED ARCHITECTURE

Service Oriented Architecture (SOA) merupakan sebuah representasi model baru untuk membangun aplikasi yang terdistribusi [5]. SOA adalah sebuah gaya arsitektural yang memodularisasi sistem

informasi menjadi *services* [6]. SOA merupakan sebuah framework yang mengintegrasikan proses bisnis dan mendukung infrastruktur IT yang aman, berkomponen terstandarisasi (*services*) yang dapat digunakan kembali dan disertakan dalam prioritas bisnis yang berubah [7].

Ciri utama SOA yakni mempromosikan melalui penggabungan komponen-komponen perangkat lunak. Diantara komponen perangkat lunak tersebut saling berinteraksi dalam satu kesatuan. Keuntungan dari penggunaan arsitektur SOA adalah *flexible*, *scalable*, *replacability*, dan *fault tolerance* [1]. Terdapat beberapa aspek kunci pada prinsip SOA [8], yaitu :

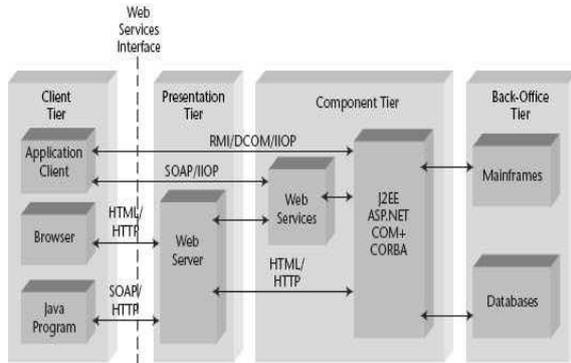
- a. *Loose Coupling*, yaitu bahwa *service* tersebut mempertahankan sebuah hubungan yang meminimalisasi ketergantungan dan mereka hanya perlu menjaga kesadaran antar satu sama lain.
- b. *Service Contract*, *service* melekat dan taat pada sebuah kesepakatan komunikasi, yang didefinisikan secara kolektif oleh atau lebih deskripsi *service* dan dokumen yang berhubungan.
- c. *Autonomy*, bahwa *service* mempunyai kendali berdasarkan logika yang di enkapsulasi.
- d. *Abstraction*, diluar apa yang dideskripsikan pada *service contract*, *services* menyembunyikan logika dari dunia luar.
- e. *Reusability*, logika terbagi menjadi *services* dengan tujuan untuk digunakan kembali.
- f. *Composability*, kumpulan dari *services* dapat dikoordinasikan dan dihimpun untuk membentuk *services* yang berbeda.
- g. *Statelessness*, *services* meminimalisasi sifat berpegang teguh pada informasi tertentu untuk sebuah aktivitas.

2.1. WEB SERVICE

Web Service adalah sebuah sistem perangkat lunak yang didesain untuk mendukung interaksi *machine-to-machine* yang dapat beroperasi melalui suatu jaringan. *Web service* memiliki sebuah antar muka yang digambarkan dalam sebuah format yang dapat diproses mesin (khususnya WSDL). Sistem lain berinteraksi dengan *web service* dengan cara yang ditentukan oleh deskripsi *web service*. Deskripsi ini menggunakan pesan SOAP yang disampaikan menggunakan HTTP dengan sebuah serialisasi XML yang berhubungan standar dengan web yang lain.

Web service dapat diimplementasikan pada lingkungan internal (intranet) untuk kebutuhan

integrasi antar sistem aplikasi ataupun pada lingkungan eksternal (internet) untuk mendukung aplikasi B2B dan B2C (e-business). Jika dilihat lingkungan kerja *web service* dapat ditunjukkan pada Gambar 1 berikut :



Gambar 1. Lingkungan Kerja *Web Service*

2.2. STANDAR KEAMANAN WEB SERVICE

Standar Keamanan *Web services* yang ada saat ini mengacu kepada standar yang dikembangkan para pemakainya sendiri. Standar keamanan ini dikembangkan oleh W3C dan OASIS. Berikut standar keamanan yang dikembangkan oleh W3C dan OASIS.

A. XML Digital Signature (XDSIG)

Protokol ini memberikan aspek integritas data untuk memverifikasi data, dan nonrepudiation untuk memverifikasi pengirim dalam menandai isi dokumen digital. *XML Digital Signature* merupakan implementasi *digital signature* dalam XML. Prinsip yang digunakan pun secara umum sama dengan *digital signature*. Seluruh data digunakan untuk menghasilkan nilai hash. Dengan menggunakan algoritma dan kunci *public-privat* yang dioperasikan pada nilai hash tersebut, maka akan menghasilkan *signature*. *Signature* ini akan dikirim bersamaan dengan dikirimnya data.

B. XML Encryption (XENC)

Protokol ini berfungsi untuk melakukan enkripsi dan dekripsi isi dokumen digital. *XML Encryption* akan membungkus elemen yang dienkripsi menggunakan tag XML penanda. Dalam tag XML penanda tersebut terdapat kunci enkripsi, petunjuk metode yang digunakan untuk melakukan enkripsi, *chiphertext*, dan properti tambahan lainnya.

C. XML Key Management Specification (XKMS)

XML Key Management Specification (XKMS) terdiri dari dua bagian yaitu *XML Key Information Services Specification (X-KISS)* dan *XML Key Registration Services Specification (X-KRSS)*. *XML Key Information Services Specification* adalah sebuah protokol yang mendukung aplikasi dalam memproses *Key* yang berhubungan dengan *XML Signature*, *XML Encryption* dan *public key* lainnya.

XML Key Registration Services Specification adalah sebuah protokol untuk mendukung registrasi *Key* yang digunakan dalam *XML Key Information Services Specification* atau tingkat yang lebih tinggi seperti *XML Trust Assertion Services Specification*.

D. Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML) adalah sebuah *framework* berbasis XML untuk pertukaran informasi seperti *user authentication*, *entitlement*, dan informasi lainnya. Beberapa hal yang merupakan standar SAML adalah *Single Sign-On* dan *Federated identity*. *Single Sign-On* mengurus status informasi dari *user authentication*, sedangkan *Federated identity* berguna ketika membangun aplikasi yang digunakan oleh banyak *user*.

E. Extensible Access Control Markup Language (XACML)

XACML adalah spesifikasi XML untuk mengekspresikan *policy* tentang akses informasi melalui internet. *Policy* disini lebih menekankan pada *security policy* yang akan menangani mengenai *privilege*. Hal yang dapat dilakukan adalah menulis, meninjau, menguji, menyetujui, mengkombinasikan, menganalisis, memodifikasi, membatalkan, mendapatkan kembali dan menjalankan *policy*.

F. WS-Security

WS-Security atau juga dikenal sebagai *Web Services Security Core Language (WSSCore)* merupakan spesifikasi keamanan *web services* yang mendefinisikan mekanisme pengamanan pada level pesan SOAP untuk menjamin *message integrity & confidentiality*. Secara umum standard *WS-Security* mendefinisikan spesifikasi mengenai bagaimana mengamankan pesan, dengan cara menyertakan *digital signature*, *enkripsi* dan *security token* pada bagian header dari pesan. Spesifikasi WSS-Core menyediakan 3 mekanisme untuk memproteksi pesan dari ancaman terhadap upaya gangguan keamanan pesan, yaitu kemampuan untuk mengirim *security token* sebagai bagian dari pesan, *message integrity* dan *message confidentiality*.

2.3. ALGORITMA BASE64 ENCODE

Transformasi base64 merupakan salah satu algoritma untuk *Encoding* dan *Decoding* suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan *encoding* (penyandian) terhadap data binary. Karakter yang dihasilkan pada transformasi Base64 ini terdiri dari A..Z, a..z dan 0..9, serta ditambah dengan dua karakter ter- akhir yang bersimbol yaitu + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data binary atau mengisi pada

Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan.

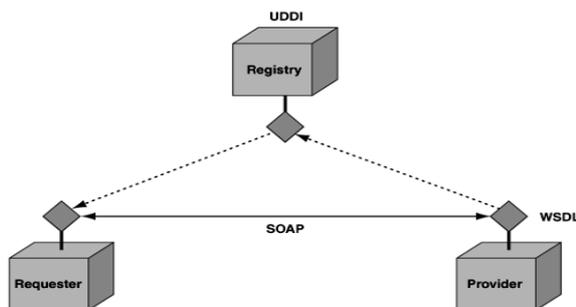
Pada *Base64 encode* satu karakter diwakili 8 bits kemudian kumpulan 8 bits tersebut di ekstrak menjadi kumpulan per 6 bits yang mewakili satu karakter yang disusun membentuk 4 bagian per blok (jadi tersusun beberapa blok dimana satu blok terdapat 24 bit data). Untuk kemungkinan jika terdapat data bit yang tidak mencapai 6 bits setelah proses ekstrak tadi, solusinya adalah dengan menambahkan bit 0 pada bit 6 hingga mencapai 6 bit dan bit 1 sisanya hingga mencapai 24 bit pada blok yang tersisa. Setelah kumpulan bit tersusun menjadi 24 bits dimana setiap blok nya dibagi menjadi 4 bagian, yang terdiri dari susunan-susunan 6 bits, barulah dari setiap bagian [9].

2.4. KOMPONEN WEB SERVICE

Komponen utama arsitektur *web service* adalah *service provider*, *service registry* dan *service requester*. Sebuah *service* adalah sebuah aplikasi yang tersedia untuk digunakan oleh requester yang sesuai prasyarat awal yang telah ditetapkan *service provider*. *Web service* dapat disusun dengan berbagai *service* lain menjadi *service* atau aplikasi baru. Berbagai *service* disebar pada suatu tempat pada web oleh *service provider*. Sebuah *service* tertentu yang disebut *registry* menyediakan dukungan untuk mempublikasikan dan menemukan *service*.

Registry merupakan sebuah tempat penyimpanan deskripsi *service* yang dapat dicari dimana *service provider* mempublikasikan *service*-nya. Sebuah bahasa deskripsi digunakan untuk mendiskripsikan *web service*. Fungsionalitas dan kebijakan akses dicatat dan diterbitkan dengan sebuah *registry*. Berbagai *service* digunakan melalui sebuah jaringan dengan menggunakan informasi yang disimpan dalam sebuah deskripsi *service*.

Web service bersandar pada pola SOA. Pemetaan pola SOA dalam *web service* beserta komponennya dapat dilihat pada Gambar 2 berikut :



Gambar 2. Pemetaan pola *web service* dan komponennya

2.5. PROTOKOL WEB SERVICE

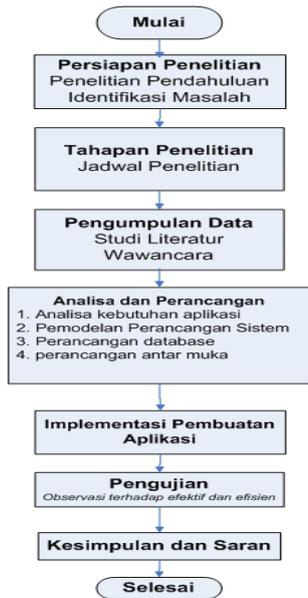
Web service berdasarkan pada sekumpulan protokol kunci. Protokol-protokol ini merupakan blok bangunan *web service platform*. protokol utama *web service* adalah sebagai berikut [1] :

- a. XML (*Extensible Markup Language*), membuat *web service* lebih mudah bertukar data antara aplikasi yang bervariasi dan untuk mengesahkan dan menerjemahkan data tersebut. Sebuah dokumen XML menggambarkan sebuah *web service* dan memasukkan detail informasi bagaimana *web service* dapat dijalankan.
- b. SOAP (*Simple Object Acces Protocol*) adalah seperangkat aturan yang memfasilitasi XML untuk melakukan pertukaran data antar aplikasi jaringan. SOAP mendefinisikan sebuah standar umum yang mengizinkan berbagai *web service* yang berbeda untuk saling beroperasi. SOAP merupakan spesifikasi *platform independent* yang mendefinisikan bagaimana pesan dapat dikirimkan diantara dua sistem perangkat lunak melalui penggunaan XML. Pesan tersebut khususnya mengikuti pola *request/response*.
- c. WSDL (*Web Service Description Language*), digunakan untuk menciptakan dokumen XML yang menggambarkan tugas yang dilakukan oleh *web service*. WSDL mendefinisikan antarmuka yang dari *web service*.
- d. UDDI (*Universal Description Discovery and Integration*), memungkinkan untuk pembuatan direktori *public* atau *privat*. Direktori *web service* ini dapat dicari. UDDI merupakan registry dari dekripsi *web service*.

Protokol Keamanan, Standar Keamanan yang masih dalam tahap pengembangan adalah *Security Assertion Markup Language (SAML)*. Standar ini digunakan untuk autentikasi dan otorisasi. Standar keamanan lainnya adalah XML Signature, XML Encryption, XKMS, dan XACML.

2.6. METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan tentang tahapan-tahapan yang dilakukan pada proses penelitian agar berjalan sesuai dengan tujuan yang telah ditentukan sehingga mencapai hasil yang baik. Penelitian ini menggunakan metodologi *Design Science Research (DSR)* yang memfokuskan pada pengembangan dan peningkatan kinerja model/desain dari sebuah sistem[10]. Tahapan penelitian dapat dilihat dari Gambar 3 berikut :



Gambar 3. Tahapan Metodologi Penelitian

2.7. IMPLEMENTASI MODEL PERSOALAN

Model persoalan pada aplikasi ini akan menghasilkan data yang saling terintegrasi antara Kementerian Agama kota Pekanbaru bidang Pendidikan Islam dan PDTA kota Pekanbaru. Kementerian Agama kota Pekanbaru dapat memperoleh data statistik masing-masing PDTA. Selanjutnya PDTA tersebut memperoleh service dari Kementerian Agama kota Pekanbaru, yaitu berupa service PNS Kemenag kota Pekanbaru dan service Nomor Statistik Madrasah (NISM) untuk proses penginputan data PDTA yang dilakukan masing-masing PDTA kota Pekanbaru.

Web Service yang disajikan oleh Kemenag kota Pekanbaru bidang Pendidikan Islam. Pada interface web service terdapat dua services yang disajikan oleh Kementerian Agama kota Pekanbaru bidang Pendidikan Islam yaitu Service list_pegawai dan Service NISM, seperti yang terlihat pada Gambar 4 berikut.



Gambar 4. Interface webservice kemenag kota Pekanbaru

Berdasarkan Web Service yang dibangun pada aplikasi pendataan PDTA tersebut terdapat pola script WSDL yang mewakili fungsi Service tersebut. Berikut keterangan potongan WSDL pada Service yang disediakan:

1. Message - pesan abstrak, mendefinisikan tipe data yang akan dikomunikasikan.

```

<message name="list_pegawaiRequest">
  <part name="p_key_search" type="xsd:string"/>
  <part name="p_page" type="xsd:integer"/>
  <part name="p_page_size" type="xsd:integer"/>
</message>
<message name="list_pegawaiResponse">
  <part name="return" type="xsd:string"/>
</message>
    
```

2. Operation - deskripsi abstrak dari suatu aksi yang didukung oleh service. Pada dasarnya menunjukkan nama operasi web service dan pesan input output.

```

<operation name="list_pegawai">
  <documentation>Daftar PNS Kemenag Kota Pekanbaru</documentation>
  <input message="tns:list_pegawaiRequest"/>
  <output message="tns:list_pegawaiResponse"/>
</operation>
<operation name="nism">
  <documentation>Nomor Statistik Madrasah</documentation>
  <input message="tns:nismRequest"/>
  <output message="tns:nismResponse"/>
</operation>
    
```

3. Port Type - mendeskripsikan sebuah web service, operasi-operasi yang dapat dijalankan, dan pesan-pesan yang dilibatkan pada Web Service.

```

<portType name="Web Service Kementerian Agama kota PekanbaruPortType">
  <operation name="list_pegawai">
    <documentation>Daftar PNS Kemenag Kota Pekanbaru</documentation>
    <input message="tns:list_pegawaiRequest"/>
    <output message="tns:list_pegawaiResponse"/>
  </operation>
  <operation name="nism">
    <documentation>Nomor Statistik Madrasah</documentation>
    <input message="tns:nismRequest"/>
    <output message="tns:nismResponse"/>
  </operation>
</portType>
    
```

4. Binding - protokol komunikasi yang digunakan oleh web service

```

<binding name="Web Service Kementerian Agama kota PekanbaruBinding" type="tns:Web Service Kementerian Agama kota PekanbaruPortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="list_pegawai">
    <soap:operation soapAction="urn:hellowsdl#list_pegawai" style="rpc"/>
    <input>
      <soap:body use="encoded" namespace="urn:hellowsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:hellowsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="nism">
    <soap:operation soapAction="urn:hellowsdl#nism" style="rpc"/>
    <input>
      <soap:body use="encoded" namespace="urn:hellowsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:hellowsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
</binding>
    
```

```
<input>
<soap:body use="encoded" namespace="urn:hellowsdl" encoding
Style="http://schemas.xmlsoap.org/soap/encoding/">
</input>
<output>
<soap:body use="encoded" namespace="urn:hellowsdl" encoding
Style="http://schemas.xmlsoap.org/soap/encoding/">
</output>
</operation>
</binding>
```

No	Nama PDTA	NISM	Kecamatan	Alamat	Proses Data
1	PDTA Nurul Salam	311214710003	Bukit Raya	Jl. Taman Sari No.50	<input type="button" value="Proses Data"/>

Gambar 6. Proses pemanggilan *Web Service* NISM

5. **Port** - Titik akhir tunggal (*single endpoint*) yang didefinisikan sebagai sebuah '*binding*' dan alamat jaringan (*network address*)

```
<port name="Web Service Kementerian Agama kota
PekanbaruPort" binding="tns:Web Service Kementerian Agama
kota PekanbaruBinding">
<soap:address location="http://kemenag.mastardjo.com/ws/index.
php"/>
</port>
```

Pada Gambar 6 dijelaskan proses *web service* memberikan informasi NISM PDTA setelah PDTA bersangkutan memasukkan NISM-nya. Jika PDTA tersebut valid maka proses selanjutnya dapat diteruskan. Berikut Gambar 7 proses input biodata PDTA yang NIMS-nya telah terdata di Kemenag.

6. **Service** - Sekumpulan *endpoint* yang saling berhubungan, akan menunjukkan *file/path* mana yang akan ditempatkan pada file WSDL tersebut.

```
<service name="Web Service Kementerian Agama kota
Pekanbaru">
<port name="Web Service Kementerian Agama kota
PekanbaruPort" binding="tns:Web Service Kementerian Agama
kota PekanbaruBinding">
<soap:address location="http://kemenag.mastardjo.com/ws/index.
php"/>
</port>
</service>
```

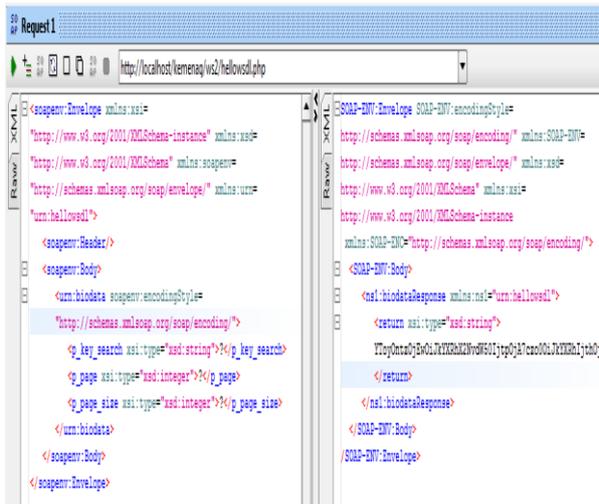
Gambar 7. *Form Input* Biodata PDTA

Pada studi kasus ini, aplikasi PDTA mendapat layanan dari aplikasi Kemenag untuk melakukan proses bisnis PDTA seperti penginputan biodata PDTA. Untuk menginputkan biodata maka PDTA bersangkutan terlebih dahulu harus mengetahui nomor statistik madrasah (NISM) yang dimilikinya. Selanjutnya PDTA bersangkutan harus menginputkan NISM tersebut pada kolom aplikasi yang disediakan. Hal ini bertujuan untuk memanggil *web service* atau layanan yang disediakan oleh Kementerian Agama kota Pekanbaru. Berikut Gambar 5 menjelaskan form input data NISM.

No	Nama PDTA	NISM	Kecamatan
----	-----------	------	-----------

Gambar 5. *Form Input* NISM PDTA

Hasil pengujian *request* dan *response* terhadap *service* menunjukkan bahwa pesan SOAP yang dibungkus kedalam WSDL di enkripsi, sehingga pesan akan tetap terjaga. Berikut Gambar 8 menjelaskan proses *request* dan *response*.



Gambar 8. Request-response data biodata PDTA

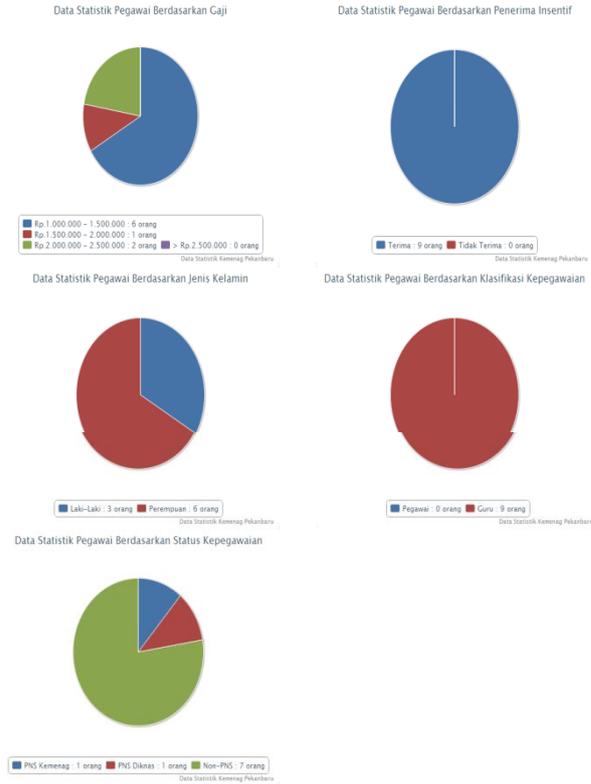
Pada pengujian kecepatan waktu *service (Load Time Service)* dengan menggunakan *tools soapUI* untuk mengukur tingkat keefektifan *web service* yang disediakan. Prosesnya dilakukan dengan mengukur jangka waktu yang dibutuhkan dalam proses pengiriman data *service* dari *server* kepada *client*. Hasil pengukuran diperoleh pada Tabel 1 berikut.

Tabel 1. Loading Test Webservice (second)

Service	min	max	avg	last	cnt	tps	bytes	bps
Biodata_pdtA	36	305	54.3	44	370	6.15	207200	3446
Foto_PD TA	36	321	56.29	66	370	6.15	208680	3473
List_peg awai	36	183	53.33	40	375	6.23	213750	3556
List_sisw a	37	318	54.46	56	300	6.16	169800	3488
NISM	37	132	52.28	55	378	6.28	210160	3494
PNS_Ke menag	39	230	57.44	43	374	6.18	213928	3538

2.8. DATA STATISTIK

Data statistik PDTA kota Pekanbaru digunakan oleh Kementerian Agama kota Pekanbaru bidang Pendidikan Islam untuk mengetahui perkembangan setiap PDTA dalam waktu-waktu tertentu. Data statistik yang dihimpun berupa data statistik siswa, data statistik pegawai/guru, data statistik biodata, dan data statistik foto bangunan gedung PDTA. Data ini ditampilkan secara realtime dari setiap PDTA yang ada. Berikut Gambar 9 data statistik dari aplikasi *Web Service* yang telah dibuat pada penelitian ini.



Gambar 9. Data Statistik PDTA

3. PENUTUP

Setelah melalui tahap pengujian pada aplikasi pendataan PDTA se-kota Pekanbaru yang diimplementasikan dengan metode *Service-Oriented Architecture (SOA)* menggunakan teknologi *Web Service*, dapat diambil kesimpulan bahwa:

1. Layanan yang diberikan Kementerian Agama kota Pekanbaru melalui *Web Service* memberikan suatu proses yang lebih efisien terutama pada proses input data guru/pegawai PNS kemenag di PDTA bersangkutan.
2. Data statistik PDTA dapat digunakan setiap saat karena data diperoleh secara *realtime* yang langsung diambil dari aplikasi setiap PDTA kota Pekanbaru.
3. *Service* hanya dapat digunakan oleh *client* yang telah diberikan logika otentikasi untuk mengakses layanan.
4. *Server* atau pihak penyedia *service* memiliki kendali penuh pada logika *service* yang disediakan.
5. Data yang dijadikan sebagai *service* dapat digunakan kembali pada aplikasi yang berbeda.

UCAPAN TERIMA KASIH

Ucapan terimakasih kepada pihak Kementerian Agama kota Pekanbaru dan PDTA (Pendidikan Diniyah Ta'miliah Awaliyah) se-kota Pekanbaru yang telah membantu memberikan data untuk penelitian ini.

DAFTAR PUSTAKA

- [1] Prasetyo, J Hendro, *Implementasi Service Oriented Architecture (SOA) Menggunakan Teknologi Web Services*, Kalimantan Tengah : Universitas Widya Dharma, 2009.
- [2] Papazoglou, M.P., & Georgekopoulus, D., *Service Oriented Computing : Introduction, Communication of The ACM* , 46(10) pp.25-28, 2007.
- [3] Meredith, L.G., & Bjorg, S., *Contract and Types, Communication of The ACM*, 46(10) pp.41-47, 2007.
- [4] Josuttis, Nicholai M, *SOA in Practice*, O'Relly 2007, pp.210 , 2007.
- [5] Hassan, Jeffrey, *Expert Service-Oriented Architecture in C# Using The Web Services Enhancements*, New York : 2.0 Appress, 2008.
- [6] Brown, Paul C, *Implementing SOA : A Pratical Guide for The Service Oriented Architecture*, Indiana : IBM Press, 2008.
- [7] Whitten, J.L, & Bentley, L.D, *Systems Analysis and Design Methods 6th Edition*, New York : McGraw Hill, 2004.
- [8] Erl, Thomas, *Service Oriented Architecture, Concepts, Technology, and Design*, Indiana : Prentice Hall PTR, 2008.
- [9] Mukhlis, Y & Dwinurti, T, *Aplikasi Transformasi Base 64 pada Kriptografi*, Depok : Gunadarma Univerity, 2010.
- [10] Peffers, "A Design Science Research Methodology for Information System Research," *Jurnal University of Nevada, Las Vegas, College of Business Administration*, 2007.