ISBN 978-602-98563-0-9

PROSIDING

KONFERENSI NASIONAL ILMU KOMPUTER 2012

Peningkatan Daya Saing Bangsa Dalam Bidang Penelitian dan Pengajaran Berbasis IT

Aula STMIK Dipanegara, 14 Januari 2012

APTIKOM

BOOKS RESELLERDISTRIBUTION SYSTEM USINGSIMPLE OBJECT ACCESS PROTOCOL

MUNAWIR
FakultasTeknikElektro
UniversitasHasanuddin
Tamalanrea SULSEL
HP. 085255747018
Sangadam59@ymail.com

ISMINARTI
FakultasTeknikElektro
UniversitasHasanuddin
TamalanreaSULSEL
HP. 081355080221
ismi_lucky@yahoo.com

Abstrak **- SOAP (*Simple Object Access Protocol)* adalah protokol untuk pertukaran informasi dengan desentralisasi pada lingkungan terdistribusi. SOAP dibangun dengan menggunakan protokol komunikasi HTTP, karena HTTP didukung oleh semua browser dan server maka SOAP dapat berkomunikasi dengan berbagai aplikasi maka meskipun terdapat perbedaan sistem operasi, teknologi, dan bahasa pemrogramannya, hal ini menjadi dasar interoperabilitas dari sistem terdistribusi books reselleryang akan diangkat sebagai standar untuk bertukar pesan-pesan berbasis XML melalui jaringan komputer atau sebuah jalan untuk program yang berjalan pada suatu sistem operasi (OS) untuk berkomunikasi dengan program pada OS yang sama maupun berbeda dengan menggunakan HTTP dan XML sebagai mekanisme untuk menyampaikan pertukaran data pada *client* dan *server***

Kata Kunci :SOAP,*client, server*

1. PENDAHULUAN

1.1 Latar Belakang

Tidaksemua system client – server mampumengaplikasikanberagamkegiatandalamsatuaksitetapisali ngterintegrasisatusama lain sekalipundalamberagam platform. Sistemterdistribusimenjadisalahsatupilihan yangtepatuntukaplikasi client server menggunakan protocol SOAP yang mampumengintegrasikanpemintaan client ke server sehinggaapa yang diminta client dapatdiresponoleh server dalampelayanan yang sesuaidenganpermintaan client. Dalampendistribusianbukucontohnya, sebuahtoko reseller (book store = tokobuku)yangbertindaksebagai distributor buku (client) yang mengambilbarangdaripemasok (server) dimana server akansangatsulitmendistribusikanbuku – bukunyatanpamengetahuiberapabanyakstokbuku, bukuapasaja yang larisdanbukuapasaja yang kurangdiminatikonsumenpadatokobuku client. Denganmemanfaatkan web service yang menggunakan protocol SOAP diharapkanpemasokdapatmemonitorkebutuhantokobuku client secaraefektifdenganmenggunakan system packet SOAP yang mampusecaracepatmeresponpermintaan client.

1.2 Tujuan Dan Manfaat

Denganmenggunakan SOAP sebagai protocol, pesanserver ke client akanterjagadengansangatamankarenasetiappesan*request* yang diterimaoleh server harussesuaidengan format yang bias dimengertioleh server. Karena client mengirimpesan*request* denganmemformatpesansesuaikeinginan server. Format disinitermasuktipe databuku yang terlibatdalampesan*request*, namaoperasiataumetodekelas yang dipanggil.Dengandemikiantujuan yang ingindicapaipemasok/server dalamrangkapendistribusianbukuke client sesuaidengan yang diinginkan.

1.3. Batasan Masalah

Masalahpendistribusianbuku kami batasihanyapadamenampilkanpermintaan client ke server danrespon server ke client denganbantuan SOAP sebagaikuriratau mediator penyampaipesandanbagaimanaduabuah program yang berbedadisatukandalamsatu platform jikadalamsatukasus computer client memilikibahasapemrograman yang berbedadengan server atauadadua client yang memilikibahasapemrograman yang berbeda. Penulisanpadajurnalinitidaksampaipadaaplikasi inventory yang membangunsatu system books reseller yang terdistribusi.

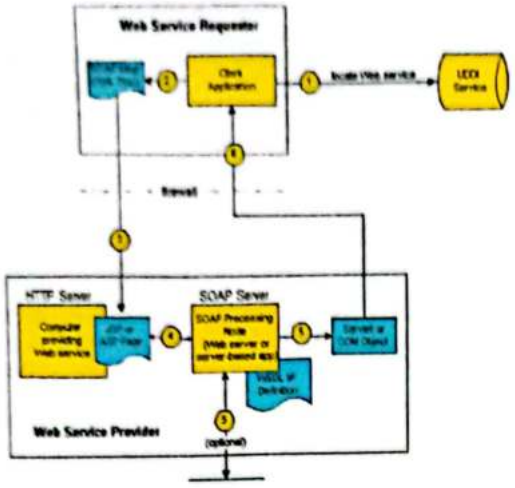
2. TINJAUAN PUSTAKA

2.1. Definisi SOAP

SOAP (*Simple Object Access Protocol)* adalah protocol untukpertukaraninformasidengandesentralisasipadalingkungante rdistribusi. SOAP dibangundenganmenggunakan protocol komunikasi HTTP, karena HTTP didukungolehsemua browser dan server sehingga SOAP dapatberkomunikasidenganberbagaiaplikasimeskipunterdapatper bedaan system operasi, teknologi, danbahasapemrograman.SOAPadalahstandaruntukbertukarpesan -pesanberbasis XML melaluijaringan computer atausebuahjalanuntuk program yang berjalanpadasuatu system operasi (OS) untukberkomunikasidengan program pada OS yang samamaupunberbedadenganmenggunakan HTTP dan XML sebagaimekanismeuntukpertukaran data.

2.2. Usecase SOAP

1. Client SOAP menggunakan registry UDDI untuk menemukan Web service. Daripada memanipulasi WSDL secara langsung, dalam kebanyakan kasus aplikasi SOAP akan didesain untuk menggunakan jenis tertentu dari port dan gaya mengikat, dan secara dinamis akan mengkonfigurasi alamat layanan yang akan dipanggil untuk mencocokkan apa yang ditemukan melalui UDDI. UDDI atau *Universal Description, Discovery and Integration* adalah sebuah directory service dimana perusahaan-perusahaan dapat mendaftar dan mencari layanan pada suatu web service. UDDI hanya menjadi tempat untuk menyimpan informasi tentang sebuah layanan yang disediakan oleh web service.



Gambar 1.

2. Aplikasi client membangun sebuah pesan SOAP, yang merupakan dokumen XML yang mampu melakukan operasi request / respon yang diinginkan.

3. Client mengirimkan pesan SOAP ke halaman JSP atau ASP pada Web server, kemudian web server mendengarkan request SOAP.

4. SOAP server Mem-parsing paket SOAP dan memanggil metode yang tepat dari objek dalam domainnya, melewati parameter yang mencakupdokumen SOAPdanmelakukan fungsi khusus seperti ditunjukkan oleh SOAP header sebelum penerimaan pesan oleh server SOAP.

5. Objek request melakukan fungsi mengembalikan data ke server SOAP, yang mempaketkansetiap respon dalam amplop SOAP. Server membungkus amplop SOAP dengan objek respon, seperti sebuah servlet atau objek COM, yang dikirim kembali ke mesin request.

6. Client menerima objek, menyegelamplop SOAP dan mengirim dokumen keprogram aslinya, menyelesaikan request/ respon.

2.3. Struktur SOAP Message

SOAP dalam system menanganiinformasitentangpermintaan service, balasandarisisi server. Proses tersebutmelibatkankomunikasi via jaringan computer. Untukkeperluantersebutpenggunaankarakter byte stream menjadipenopang SOAP.Hal tersebutmenjadikanindentifikasi format karakterpenyusun SOAP menjadihalpentinguntukmenyusunkembali byte stream danmemahamimaksudpesantersebut.Selainitupenandaanuntukid entifikasi protocol network yang digunakansemisal HTTP, FTP, menjadipentingdisertakandalampesan SOAP.

Keseluruhanpersyaratantersebuttelahdipenuhiberdasarkanstandar spesifikasi SOAP.

Spesifikasi SOAP Protocol terdiriatas 4 bagian, yaitu :

a. SOAP Envelope, yang menjelaskan format SOAP message.
b. SOAP Encoding, menggambarkanrepresentasi data dalamsebuahpengirimanpesan.
c. SOAP RPC, mendefinisikanbagaimana SOAP message dapatmengakses Remote Procedure Calls (RPCs)
d. SOAP Binding Framework, yang mendefinisikan protocol yang digunakanoleh SOAP message yang dikirimolehaplikasi.

2.4.Web-services Description Language (WSDL)

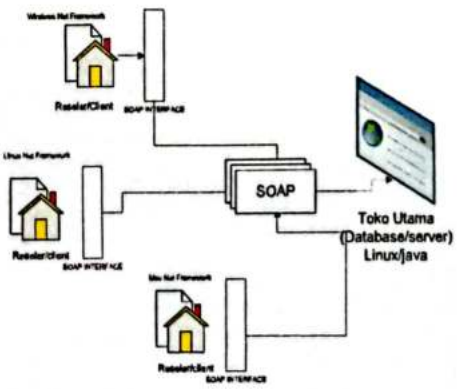
Menurut Shohoud (2001) WSDL merupakan sebuah bahasa berbasis XML yang digunakan untuk mendefinisikan web-service dan menggambarkan bagaimana cara untuk mengakseswеb-service tersebut.

Deskripsi WSDL mendefinisikan sebuahservice sebagai kumpulan dari port dimana tiap-tiapport didefinisikan secara abstrak sebagai portTypeyang mendukung sekumpulan operasi-operasi.Tiap-tiap operasi memproses sekumpulan pesantertentu.

Dalam Manes (2001) disebutkan bahwa ada5 elemen utama dalam sebuah dokumen WSDLyaitu:

a. Elemen <type>, berfungsi untuk mendefinisikantipe data-tipe data yang digunakan dalam pesan.
b. Elemen <message>, berfungsi untukmendefinisikan format dari sebuah pesan. Pesandigunakan sebagai struktur masukan (input) ataukeluaran (output) bagi operasi.
c. Elemen <portType>, berfungsi untukmendefinisikan sekumpulan operasi-operasi.Tiap-tiap elemen <operation> mendefinisikansebuah operasi dan pesan masukan atau keluaranyang berkaitan dengan operasi tersebut.
d. Elemen <binding>, berfungsi untuk memetakanoperasi-operasi dan pesan yang terdefinisikanpada port type ke protokol tertentu.
e. Elemen <service>, berfungsi untukmendefinisikan sekumpulan port-port yang salingberhubungan. Elemen <port> memetakanbinding ke lokasi dari sebuah web-service.

2.5. Arsitektur System Books Reseller



Gambar 2 Arsitektur Sistem Books Reseller

Dalamaplikasinya, Tokoutamadigambarkansebagaipemasok/server yang berhubungandengan reseller /client dimanabeberapa reseller/client menggunakanOS yang berbeda (beragam

platform) yang berfungsimemberikaninformasiberupa update hargabukuterbaru, barang yang tersedia, barang yang laris, barang yang kurangpeminatnya, sisastokdanberbagai info lainnyayang
kemudiandapatmendistribusikaninformasitersebutketokoutamas ebagaiserver, jadiapabilaada reseller/client yang inginmencari data bukuterbaru, clienttinggalmengakses data yang terdapat di tokobukuutamakemudiandari data base
tokoutamaakanmengiriminformasi yang dimintamelaluiSOAP yang terdapatpadaaplikasi client.

Padabagianiniakanditunjukan proses komunikasi yang terjadiantaraaplikasimelaluipertukaranpesan plain-SOAP (pesan SOAP murni)
Padabagianinisengajadiperlihatkantanpamelakukan parsing terhadappesan XML semata-matainginmenunjukan proses yang sesungguhnyaterjadiantaraaplikasi yang
berkomunikasidenganberbasis protocol SOAP.

3. IMPLEMENTASI

3.1 Pengujian Komunikasi Antar-Aplikasi Berbasis SOAP

Dalam konsep protokol SOAP, dua aplikasi atau lebih yang berkomunikasi satu sama lain pada dasarnya proses yang terjadi hanya dengan saling bertukar pesan plain-teks (teks murni) yang terformat sesuai standar SOAP. Suatu aplikasi yang membutuhkan proses aplikasi lain, dalam hal ini meng-invoke layanan yang disediakan oleh web service, maka aplikasi tersebut akan mengirim suatu pesan plain-text dalam format SOAP. Pesan ini disebut sebagai *request*.

Setiap pesan *request* yang diterima oleh server haruslah sesuai dengan format yang bisa dimengerti oleh server tersebut.Oleh karenanya client yang mengirim pesan *request* harus memformat pesan tersebut sesuai keinginan server. Format disini termasuk tipe data yang terlibat dalam pesan *request*, nama operasi atau metode kelas yang dipanggil dan lain sebagainya. Demikian pula pesan *response* yang menjadi balasan server nantinya harus dimengerti pula oleh client. Penting dalam hal ini server memberi tanda dengan tag yang menjadi konten hasil proses.

Oleh karena ini semua hal standar tersebut termasuk format komunikasi dideskripsikan secara rinci oleh server melalui suatu bahasa deskripsi layanan yaitu *Web Service Description Language* (WSDL).WSDL ini menjadi kunci informasi tentang format pesan SOAP dalam komunikasi.Pada bagian ini akan ditunjukan proses komunikasi yang terjadi antara aplikasi melalui pertukaran pesan plain-SOAP (pesan SOAP murni). Pada bagian ini sengaja diperlihatkan tanpa melakukan parsing terhadap pesan XML semata-mata ingin menunjukan proses yang sesungguhnya terjadi antara aplikasi yang berkomunikasi dengan berbasis protokolSOAP.Disini akan diangkat aplikasi yang berbeda platform. Sebuah platform diposisikan sebagai penyedia layanan, sementara yang lainnya sebagai *service requester*. Berikut model pengujian dan langkah-langkah yang ditempuh dalam melakukan pengujian

3.1.1 Coding dan deploy Server

Berikutini file pada server dengan mengunakan aplikasi netbeans

```
package bookStore;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
*/
```

```
@WebService(serviceName = "ResellerApp")
public class ResellerApp {
    private Book book1,book2, book3, book4;
  private String tes;
   /** This is a sample web service operation */
   @WebMethod(operationName = "makeConnection")
   public boolean makeConnection() {
      book1 = new Book(1,"Biru Langit", "Novel", "Muna" ,
120000,25);
      book2 = new Book(2,"Darker Than Dark", "Filosofi",
"AarM" , 175000,30);
      book3 = new Book(3,"In The End", "Fiksi", "Mona
Awhie" , 98000,25);
      book4 = new Book(4,"Programming Game", "Komputer",
"Joy Kanre" , 45000,12);

      return true;
   }

   @WebMethod(operationName = "getID")
   public int getID(@WebParam(name = "item") int item) {

      if (item==1){
        return 1;
      }else if (item==2){
        return 2;
      }else if (item==3){
        return 3;
      }else {
        return 4;
      }
   }

   private void booking(){
      book1 = new Book(1,"Negeri Ratu", "Novel", "Muna" ,
120000,25);
      book2 = new Book(2,"Darker Than Dark", "Filosofi",
"AarM" , 175000,30);
      book3 = new Book(3,"In The End", "Fiksi", "Mona
Awhie" , 98000,25);
      book4 = new Book(4,"Programming Game", "Komputer",
"Joy Kanre" , 45000,12);
   }

   @WebMethod(operationName = "getTitle")
   public String getTitle(@WebParam(name = "item") int
item) {
      booking();
      if (item==1){
        return book1.getTitle();
      }else if (item==2){
        return book2.getTitle();
      }else if (item==3){
        return book3.getTitle();
      }else {
        return book4.getTitle();
      }
   }
   @WebMethod(operationName = "getCategory")
   public String getCategory(@WebParam(name = "item") int
item) {
  booking();
      if (item==1){
        return book1.category;
      }else if (item==2){
        return book2.category;
      }else if (item==3){
```

```
        return book3.category;
      }else {
        return book4.category;
      }
    }

    @WebMethod(operationName = "getAuthor")
    public String getAuthor(@WebParam(name = "item") int
item) {
      booking();
      if (item==1){
        return book1.author;
      }else if (item==2){
        return book2.author;
      }else if (item==3){
        return book3.author;
      }else {
        return book4.author;
      }
    }

    @WebMethod(operationName = "getPrice")
    public double getPrice(@WebParam(name = "item") int
item) {
      booking();
          if (item==1){
        return book1.price;
      }else if (item==2){
        return book2.price;
      }else if (item==3){
        return book3.price;
      }else {
        return book4.price;
      }
    }
    @WebMethod(operationName = "getPercent")
    public int getPercent(@WebParam(name = "item") int item)
{
      booking();
      if (item==1){
        return book1.percent_reseller;
      }else if (item==2){
        return book2.percent_reseller;
      }else if (item==3){
        return book3.percent_reseller;
      }else {
        return book4.percent_reseller;
      }
    }
}
```

3.1.2 Coding client

Berikut ini source code pada sisi client dimana menggunaka Aplikasi VbNet

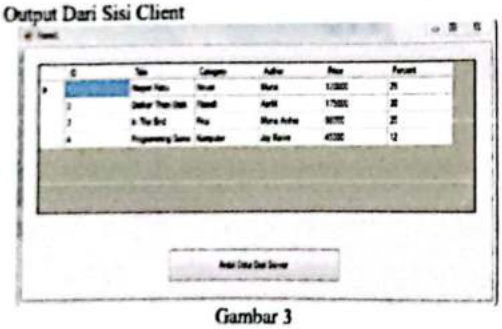
```
PublicClass Form1
Dim Books AsNew List(Of BookList)
Private id AsInteger
Private title AsString
Private category AsString
Private author AsString
Private price AsDouble
Private percent AsInteger
PrivateSub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) HandlesMyBase.Load
EndSub
PrivateSub getBookList()
Dim i AsInteger
Dim nRecord AsInteger = 4
Dim prog AsInteger
      prog = 100 / 4
      prog = prog / 5
      Books.Clear()
For i = 1 To nRecord
Dim client AsNew bookStore.ResellerAppClient
        id = client.getID(i)
        ProgressBar1.Value = ProgressBar1.Value + prog
        title = client.getTitle(i)
        ProgressBar1.Value = ProgressBar1.Value + prog
        category = client.getCategory(i)
        ProgressBar1.Value = ProgressBar1.Value + prog
        author = client.getAuthor(i)
        ProgressBar1.Value = ProgressBar1.Value + prog
        price = client.getPrice(i)
        ProgressBar1.Value = ProgressBar1.Value + prog
        percent = client.getPercent(i)
        Books.Add(New BookList(id, title, category, author,
price, percent))
Next
EndSub
PrivateSub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
      ProgressBar1.Value = 0
      Button1.Visible = False
      Label1.Visible = True
      ProgressBar1.Visible = True
      getBookList()
      DataGridView1.DataSource = Books
      ProgressBar1.Visible = False
      Label1.Visible = False
      Button1.Visible = True
EndSub
EndClass
```

Struktur class untyuk menerima databuku dari server

```
PublicClass BookList
Private _id AsInteger
Private _title AsString
Private _category AsString
Private _author AsString
Private _price AsDouble
Private _percent AsInteger
PublicSubNew(ByVal id AsInteger, ByVal title AsString,
ByVal category AsString, ByVal author AsString, ByVal price
AsDouble, ByVal percent AsInteger)
      _id = id
      _title = title
      _category = category
      _author = author
      _price = price
      _percent = percent
EndSub
PublicProperty ID() AsInteger
Get
Return _id
EndGet
Set(ByVal value AsInteger)
        _id = value
EndSet
EndProperty
PublicProperty Title() AsString
Get
Return _title
EndGet
Set(ByVal value AsString)
```

```
        _title = value
    EndSet
    EndProperty
    PublicProperty Category() AsString
    Get
    Return _category
    EndGet
    Set(ByVal value AsString)
        _category = value
    EndSet
    EndProperty
    PublicProperty Author() AsString
    Get
    Return _author
    EndGet
    Set(ByVal value AsString)
        _author = value
    EndSet
    EndProperty
    PublicProperty Price() AsDouble
    Get
    Return _price
    EndGet
    Set(ByVal value AsDouble)
        _price = value
    EndSet
    EndProperty
    PublicProperty Percent() AsInteger
    Get
    Return _percent
    EndGet
    Set(ByVal value AsInteger)
        _percent = value
    EndSet
    EndProperty
EndClass
```

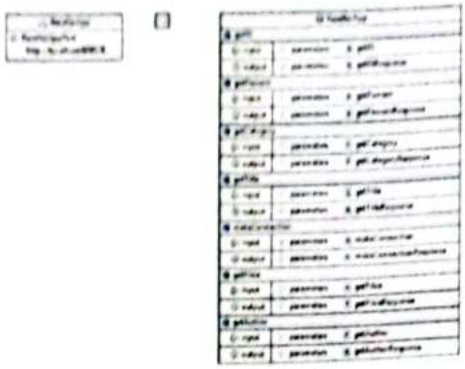
Output Dari Sisi Client



Gambar 3

3.2 WSDL Server

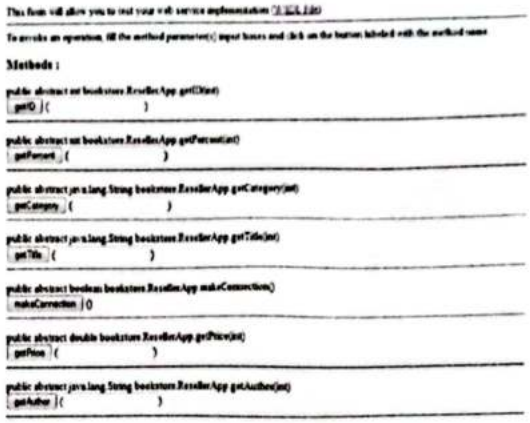
Berikut ini merupakan tampilan Visual WSDL



Gambar 4 Visual WSDL

Output pada Web browser pada sisi Server

ResellerApp Web Service Tester



3.2.1 Pengujianpesan response denganmengirim SOAP-request dengan SOAPUI

Untuk mengetahui format pesan SOAP (*request* dan *response*), maka dapat dilakukan dengan mengakses langsung operasi-operasi yang tercantum pada halaman diatas. Alamat setiap operasi dapat juga diakses melalui alamat url.

Request:

```
POST /SOAP/webservice2.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
<soap12:Body>
<Add xmlns="http://localhost/SOAP/">
<x>int</x>
<y>int</y>
</Add>
</soap12:Body>
</soap12:Envelope>
```

Response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml;
charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLS
chema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLS
chema"
xmlns:soap12="http://www.w3.org/2003/0
5/soap-envelope">
<soap12:Body>
<AddResponse
xmlns="http://localhost/SOAP/">
<AddResult>int</AddResult>
</AddResponse>
</soap12:Body>
</soap12:Envelope>
```

4. PENUTUP

4.1. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dapat ditarik kesimpulan :

1. Dengan menggunakan SOAP sebagai protocol, pesan server ke client akan terjaga dengan sangat aman karena setiap pesan *request* yang diterima oleh server harus sesuai dengan format yang bias dimengerti oleh server.
2. Tujuan yang ingin dicapai pemasok/server dalam rangka pendistribusian buku ke client sesuai dengan yang diinginkan

DaftarPustaka

1. http://www.sitepoint.com/really-good-introduction-xml/
2. http://www.w3schools.com
3. http://www.soatutorial.net/soa-technology-soap-wsdl-uddi/