IMPLEMENTASI FRAGMENTASI HORIZONTAL DENGAN MENGGUNAKAN SOAP *WEB SERVICE* DAN WSDL

Indra Budiantho
Dosen Universitas INDONUSA Esa Unggul
budiantho_indra@yahoo.com

Abstrak

Pengelolaan data secara terpusat dengan berbagai kerugiannya menggiring kepada pendistribusian data ke beberapa *site*. Salah satu keuntungan daripada sistem tersebar ini adalah memungkinan pembagian beban kerja di mana kebijakannya dapat dilakukan dengan memfragmentasikan data yang ada, baik secara vertikal ataupun horizontal. Pertanyaan berikutnya yang timbul adalah bagaimana mekanisme pendistribusian tersebut diimplementasikan? CORBA adalah salah satu jawaban, sedangkan SOAP *web service* adalah jawaban lainnya. WSDL memberikan sumbangan di dalam kemudahan implementasi *web service* tersebut. Artikel ini membahas mekanisme terakhir dengan kebijakan pendistribusian fragmentasi horizontal. Pengodean *web service* menggunakan modul PHP 5 SOAP Extension, sedangkan sebagai DBMS dipilih MySQL.

Kata Kunci: fragmentasi horizontal, *web service*, WSDL, PHP 5 SOAP Extension.

Pendahuluan

Fragmentasi horizontal mengiris tabel data secara horizontal menjadi beberapa buah tabel sedemikian hingga tabel-tabel yang dihasilkan memiliki kolom yang sama dengan tabel asal, namun dengan jumlah *record* yang normalnya lebih sedikit.

SOAP web service pada dasarnya adalah sebuah RPC (*Remote Procedure Call*) – fungsi didefinisikan pada sebuah *host*/*site* dan dipanggil oleh kode yang terletak pada *host*/*site* yang lain – di mana pesan, baik berupa permintaan (*request*) ataupun tanggapan (*respond*), dikirimkan dalam bentuk XML yang mana sintaks dan semantiknya mengikuti aturan (protokol) yang disebut dengan SOAP.

WSDL (*Web Service Description Language*) adalah XML yang menggambarkan suatu *web service*. Penggunaan WSDL akan memudahkan pemrograman yang akan sangat berguna untuk memfokuskan perhatian kita pada kebijakan pendistribusian data, bukan pada kerumitan pemrograman.

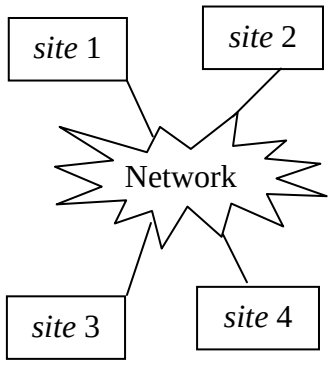
Tinjauan Teori

Data Terdistribusi

Sistem basisdata terdistribusi adalah sistem basisdata yang difragmentasikan atau direplikasikan pada berbagai konfigurasi perangkat keras dan perangkat lunak, yang mana masing-masing fragmen atau replika biasanya berada pada lokasi yang secara geografis berbeda. Fragmen data merupakan himpunan bagian

atau irisan dari basisdata aslinya. Replika data adalah *copy* keseluruhan dari basisdata aslinya.

Ide basis data terdistribusi dapat digambarkan sebagai berikut:



Site 3 mengakses fragmen/replika data yang disimpan di dalam sistem basis data pada *site* 1, *site* 2, dan *site* 4 lewat suatu jaringan. Di dalam artikel ini hanya fragmentasi horizontal yang diimplementasikan.

Fragmentasi dan Transparansi

Terdapat dua macam fragmentasi, yaitu fragmentasi horizontal dan fragmentasi vertikal. Yang pertama mengiris tabel asal secara horizontal sedemikian hingga memiliki baris *record* yang merupakan himpunan bagian dari tabel asal. Fragmen-fragmen tipe ini dapat digabungkan dengan melakukan operasi union. Sedangkan fragmentasi vertikal membelah tabel asal secara vertikal sedemikian hingga memiliki kolom yang adalah himpunan bagian dari kolom asal. Fragmen-fragmen vertikal dapat digabungkan dengan menggunakan operator join.

Di dalam artikel ini, data difragmentasikan ke dalam tabel-tabel dengan DDL sebagai berikut:

```
CREATE TABLE fragment1(
    id BIGINT NOT NULL AUTO_INCREMENT,
  name VARCHAR(25) NOT NULL,
  PRIMARY KEY (id)
);

CREATE TABLE fragment2(
    id BIGINT NOT NULL AUTO_INCREMENT,
  name VARCHAR(25) NOT NULL,
  PRIMARY KEY (id)
);
```

fragment1 menampung *record* yang nilai *field* name-nya diawali oleh karakter A s/d M, sedangkan fragment2 untuk *record* yang nilai *field* name-nya diawali oleh karakter N s/d Z.

Sistem basisdata terdistribusi yang baik mensyaratkan transparansi fragmentasi di mana *user* tidak perlu mengetahui bagaimana data difragmentasikan, transparansi replikasi di mana *user* tidak perlu tahu bagaimana replikasi data dilakukan, dan yang terakhir adalah replikasi lokasi di mana *user* tidak perlu tahu pada *site* mana data diletakkan. Hal terakhir ini dapat dicapai dengan menggunakan WSDL yang adalah sebuah dokumen XML.

Keuntungan Data Terdistribusi

Dibandingkan dengan sistem basisdata terpusat, sistem basisdata terdistribusi memiliki beberapa keuntungan, yaitu:

- Mengemulasikan struktur organisasi. Di era *e-commerce*, organisasi/perusahaan dapat memiliki cabang tersebar di seluruh dunia secara geografis; demikian pula hendaknya data didistribusikan.

- Kendali yang lebih besar. Fragmen data yang didistribusikan ke suatu *site* hanya dapat di-*update* oleh pekerja pada *site* tersebut.
- Keterpercayaan yang meningkat. Tidak berfungsinya sistem pada sebuah *site* dapat digantikan oleh *site* lain yang memiliki replika data yang bersangkutan.
- Unjuk kerja yang lebih besar. Akses data tertentu di sebuah *site* dapat dilakukan secara lokal.
- Pertumbuhan yang lebih mudah. Data yang jumlahnya terus meningkat akan membutuhkan tempat penyimpanan yang besar. Dengan distribusi data maka kebutuhan ini dapat disebarkan ke *site-site* yang lain.

Tipe Sistem Basisdata Terdistribusi

Ada tiga buah tipe sistem basisdata terdistribusi, yaitu sistem basisdata terdistribusi homogen, sistem basis data terdistribusi heterogen, dan sistem basisdata terdistribusi federasi (*multi-database*). Yang pertama terdiri atas *site-site* yang menyimpan fragmen data di dalam DBMS (*Database Management System*) yang semuanya sama (misal ORACLE) dan umumnya juga berjalan pada *platform* (sistem operasi dan perangkat keras) yang sama. Sebaliknya, yang kedua menjalankan berbagai DBMS dengan berbagai *platform* (misal *site* 1 menggunakan Informix pada UNIX, sedangkan yang lain menjalankan DBMS Ingres pada Windows NT); antarmuka DBMS biasanya dilakukan dengan menggunakan *gateway* (seperti ODBC). Yang ketiga, sistem federasi, memperlakukan setiap DBMS yang memiliki otonomi tersendiri, di mana komunikasi dilakukan secara universal dengan menggunakan standar XML.

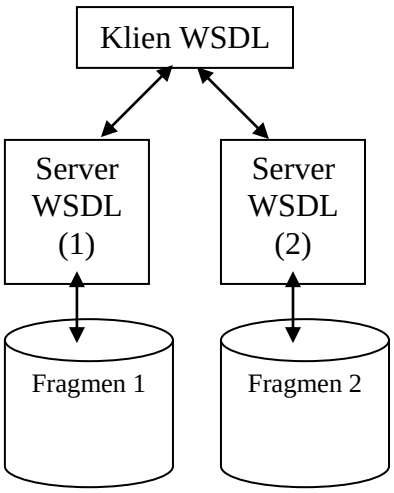
Artikel ini menerapkan sistem federasi dengan menggunakan SOAP dan WSDL sebagai dokumen XML serta *gateway* berupa skrip PHP yang adalah sebuah *client* WSDL. Sedangkan DBMS yang digunakan adalah homogen, yaitu MYSQL, di mana data disimulasikan terdistribusi dengan menfragmentasikannya ke dalam dua buah tabel yang berbeda. Walaupun dilakukan secara lokal, namun hal ini dapat diterapkan juga di dalam jaringan sesuai dengan konsep dasar Internet.

Pembahasan

SOAP *Web Service*

Web service adalah suatu *remote procedure call* (RPC) yang dengan bantuan protokol SOAP dan atau WSDL mengaktualisasikan konsep sistem terdistribusi yang dapat dimanfaatkan untuk mengimplementasikan sistem basisdata terdistribusi. Jadi, sebenarnya sistem terdistribusi ini dapat digunakan sebagai perantara untuk menerapkan prinsip transparansi lokasi daripada sistem basisdata terdistribusi. Hal ini dapat dicapai karena RPC memanggil *remote* fungsi seolah-olah fungsi tersebut berada pada mesin lokal yang memanggil fungsi tersebut.

Di dalam artikel ini, *web service* yang diimplementasikan akan memiliki struktur berikut:



di mana *client* WSDL akan memutuskan ke *server* mana suatu *record* akan dilemparkan. *Server* WSDL(1) akan menangani fragmen 1, sedangkan *server* WSDL(2) akan menangani fragmen 2.

WSDL

WSDL adalah XML yang digunakan untuk menggambarkan sebuah *web service*, yaitu: bagaimana *web service* tersebut diakses, operasi-operasi (fungsi) apa yang disediakan, bagaimana pesan dikirimkan, dan bagaimana struktur dari pesan tersebut. Adapun *client* dan *server* dari WSDL adalah sebagai berikut:

fragment. wsdl

```
<?xml version="1.0"?>
<definitions name="MyDefinition"

targetNamespace="urn:myTargetNamespace"
        xmlns:tns="urn:myTns"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns="http://schemas.xmlsoap.org/wsdl/">
  <message name="myRequest">
    <part name="reqParam" type="xsd:string"/>
  </message>
  <message name="myResponse">
    <part name="resParam" type="xsd:int"/>
  </message>
  <portType name="MyPortType">
    <operation name="doInsert">
      <input message="tns:myRequest"/>
      <output message="tns:myResponse"/>
    </operation>
  </portType>
  <binding name="MyBinding" type="tns:MyPortType">
    <soap:binding style="rpc"

transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="doInsert">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="encoded"

namespace="urn:myInputNamespace"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use="encoded"

namespace="urn:myOutputNamespace"
```

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>
  </binding>
  <service name="MyService">
    <documentation>Returns a greeting string.</documentation>
    <port name="MyPort" binding="tns:MyBinding">
      <soap:address location="http://localhost/tempMe/FragmentServerWsdl.php"/>
      <soap:address location="http://localhost/tempMe/FragmentServerWsdl2.php"/>
    </port>
  </service>
</definitions>
```

Sebuah *service* berupa fungsi dengan nama doInsert() dideskripsikan di dalam XML ini yang melakukan operasi penyisipan sebuah *record* ke dalam sebuah tabel dan mengembalikan nilai bertipe int yang menyatakan *host* di mana fragmen tabel tersebut didefinisikan.

Server yang dapat dituju ada dua buah, yaitu: http://localhost/tempMe/FragmentServerWsdl.php dan http://localhost/tempMe/FragmentServerWsdl2.php

FragmentServerWsdl1.php

```
<?php
class DBService{
  function doInsert($someone){
    $host = "localhost";
    $user = "root";
    $pwd = "mentalimage";
    $database = "mydb";

    $mysqli = new mysqli($host,$user,$pwd,$database);

    $q = "INSERT INTO fragment1 VALUES(NULL,'$someone')";
    mysqli_query($mysqli,$q);

    return 1;
  }
};

ini_set("soap.wsdl_cache_enabled","0");
$server = new SoapServer("http://localhost/tempMe/fragment.wsdl",

array('soap_version'=>SOAP_1_2));
$server->setClass("DBService");
$server->handle();
?>
```

Klas DBService mengandung fungsi *service* untuk menyisipkan sebuah *record*.

Objek $server di dalam skrip ini menangani *record* untuk fragmen pertama; di dalam contoh ini, fragmen tersebut berada pada *localhost* dan diimplementasikan dalam tabel bernama fragment1.

FragmentServerWsdl2.php

```
<?php
class DBService{
  function doInsert($someone){
    $host = "localhost";
    $user = "root";
    $pwd = "mentalimage";
    $database = "mydb";

    $mysqli = new mysqli($host,$user,$pwd,$database);

    $q = "INSERT INTO fragment2 VALUES(NULL,'$someone')";
    mysqli_query($mysqli,$q);
```

```
   return 2;
  }
};

ini_set("soap.wsdl_cache_enabled","

0");


$server = new
SoapServer("http://localhost/tempMe
/fragment.wsdl",

array('soap_version'=>SOAP_1_2));
$server->setClass("DBService");
$server->handle();
?>
```

Objek $server di dalam skrip ini menangani *record* untuk fragmen kedua; di dalam contoh ini, fragmen tersebut berada pada localhost dan diimplementasikan dalam tabel bernama fragment2.

FragmentClientWsdl.php

```
<?php
$client = new
SoapClient("http://localhost/tempMe
/fragment.wsdl",

array('soap_version'=>SOAP_1_2,
'trace'=>1));
$name = 'indra';
if(strtoupper($name{0})<'N'){
 $location = $client-
>__setLocation("http://localhost/tem
pMe/FragmentServerWsdl1.php");
}
else{
 $location = $client-
>__setLocation("http://localhost/tem
pMe/FragmentServerWsdl2.php");
}
echo "Record inserted into host
".$client->doInsert($name)."<br>";
?>
```

Client WSDL menangani kebijakan ke mana sebuah *record* mesti dilemparkan. Di dalam artikel ini masalah disederhanakan dengan hanya mengecek huruf pertama dari suatu masukan. Hal yang menarik adalah penggunaan API daripada PHP 5 SOAP Extension, yaitu: __setLocation() untuk mendistribusikan data *record*. Parameter berupa URI skrip server yang dituju.

Kesimpulan

Fragmentasi horizontal dapat diimplementasikan dengan menggunakan *web service* dan WSDL.

Penggunaan WSDL mewujudkan transparansi lokasi sesuai dengan karakteristik sistem basisdata terdistribusi.

Pemanggilan *web service* dengan wsdl yang mudah pada sisi *client* membuat kita dapat berfokus pada kebijakan pendistribusian data yang akan diterapkan.

Daftar Pustaka

Andy Gutmans, Stig Saether Bakken, Derick Rethans, "*PHP 5 Power Programming",* Prentice Hall, 2005.

Bach, "*The Unix Operating System",* Prentice Hall,1987.

Kevin Kaichuan He, "*Why and How to Use Netlink Socket",* Linux Journal, 2005.

Michael Kofler, "*The Definitive Guide to MySQL 5", 3rd Edition,* APress, 2005.

Paul Beynon, Davies, "*Database Systems", 3rd Edition,* Palgrave Macmillan, 2004.

Robert Richards, "*Pro PHP XML and Web Services",* APress, 2006.