

## Pengujian Penggunaan *Simple Object Access Protocol (SOAP)* pada *Remote Procedure Call (RPC)*

**Djoni Haryadi Setiabudi, Irwan Kristanto Julistiono**

Fakultas Teknologi Industri, Jurusan Teknik Informatika - Universitas Kristen Petra  
e-mail : djonihs@petra.ac.id, irwankj@petra.ac.id

**Teguh Wijaya**

Alumni Fakultas Teknologi Industri Jurusan Teknik Informatika – Universitas Kristen Petra

### Abstrak

Umumnya protokol RPC yang digunakan pada saat ini adalah DCOM. Saat ini ada alternatif protokol baru, yakni SOAP, yang berdasarkan pada teknologi XML. Untuk mengetahui sejauh mana keunggulan pemanfaatan SOAP pada RPC, dan kelebihanannya dibandingkan dengan teknologi DCOM, pada penelitian ini dilakukan pengumpulan informasi serta uji coba pada suatu aplikasi. Aplikasi yang dibuat terdiri dari 2 macam. Aplikasi pertama adalah aplikasi terdistribusi pemesanan barang dengan menggunakan SOAP. Aplikasi kedua adalah penghitungan aritmatika yang menggunakan 2 versi yaitu RPC dengan SOAP dan RPC dengan DCOM sebagai pembandingnya. Aplikasi untuk RPC dengan menggunakan protokol SOAP dengan Visual Basic 6.0 untuk penulisan kode dan kompilasi, Microsoft SOAP toolkits sebagai pembuatan file WSDL dan WSML. Pada aplikasi pemesanan barang yang menggunakan RPC dengan SOAP sebagai aplikasi terdistribusinya didapatkan hasil aplikasi dapat berjalan dengan efisien dan praktis, karena dengan aplikasi terdistribusinya beban tugas kerja *server* dapat terbagi-bagi antar *server* lain. Pada aplikasi penghitungan, yang dibuat dengan menggunakan 2 versi yaitu RPC dengan SOAP dan RPC dengan DCOM. DCOM memiliki kelebihan yaitu waktu koneksi 1,99 kali lebih cepat dan proses perhitungan 9,92 kali lebih cepat dibandingkan SOAP.

**Kata Kunci:** SOAP (Simple Object Access Protocol), RPC (Remote Procedure Call), DCOM (Distributed Component Object Model).

### Abstract

*Mostly RPC protocol, which is commonly used now is DCOM. Nowadays another protocol is created, that is SOAP, which is based on XML technology. In this research, it is investigated the use of SOAP on RPC and the advantage compared to DCOM. To do that, the collection of information and application benchmarking are necessary to be performed. Application, which was did, consist two kinds. First, distributed application in ordering products is used to know the use of SOAP. Next, arithmetic calculation is used to compare SOAP to DCOM on RPC. The application using SOAP protocol needs Visual Basic 6 software to write and to compile the code of the application, Microsoft SOAP toolkits to create WSDL and WSML files. The result of the distributed application in ordering products using RPC by SOAP is efficient and practical, because server tasks can be shared to each server. Then, the comparison between the application arithmetic calculation using SOAP and DCOM show different results. The application using DCOM has the advantages on connection time faster 1.99 times than SOAP-WSDL and on performance by doing calculation operation faster 9.92 times than SOAP-WSDL.*

*Keywords: SOAP (Simple Object Access Protocol), RPC (Remote Procedure Call), DCOM (Distributed Component Object Model)*

### Pendahuluan

Pengolahan terdistribusi antar Web dikenal dengan nama *Web services*. *Web services* berupa suatu objek di Web yang diakses dengan HTTP, menerima masukan dalam bentuk XML dan memberikan keluaran dalam bentuk XML. *Web services* dapat diakses oleh suatu Web atau oleh aplikasi yang mampu mengirim request dan menerima response

melalui HTTP. *Web services* menggunakan konsep RPC (remote procedure call) dengan format data yang dikirim adalah XML. Format data XML yang dipertukarkan antara *web services* dengan Web atau aplikasi pengaksesnya adalah SOAP (Simple Object Access Protocol).

Pada penelitian ini dilakukan :

- Perbandingan penggunaan aplikasi RPC dengan teknologi DCOM (Distributed Component Object Model)
- Menemukan kelebihan serta kekurangan dari SOAP.

**Catatan:** Diskusi untuk makalah ini diterima sebelum tanggal 1 Juni 2005. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 5, nomor 2, September 2005.

- Pembuatan aplikasi dengan menitikberatkan pada penggunaan SOAP di aplikasi RPC (Remote Procedure Call)

Bentuk RPC disini berupa COM (Component Object Model), dimana COM tersebut merupakan versi Object-Oriented dari DCE RPC(Distributed Computing Environment Remote Procedure Call).

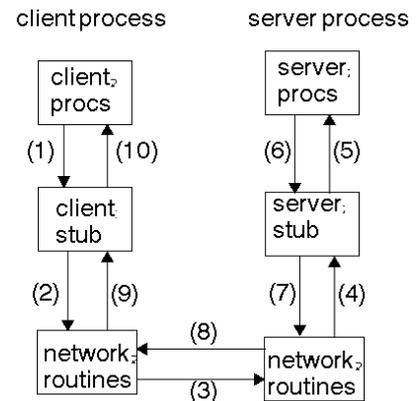
Aplikasi pertama yang dibuat adalah mengenai pemesanan barang elektronik. Client menginputkan data pesannya kemudian data tersebut akan dikirimkan ke Sales server dan Manufacturing server. Kemudian, client akan menerima konfirmasi bahwa data telah diterima pada kedua server tersebut. Aplikasi kedua adalah perbandingan program kalkulasi dengan menggunakan DCOM dan SOAP-WSDL.

## Teori

### Remote Procedure Call (RPC) [4]

*Remote procedure call* bertindak seperti pemanggilan pada prosedur biasa, tetapi bekerja melewati jaringan. Soket/penghubung penggunaan metode (*method*) pada jaringan adalah sistem pesan, dimana suatu proses menulis pesan kepada yang lain. Suatu proses menjalankan *remote procedure call* dengan memasukkan parameternya dan alamat pemanggil (*return address*) pada stack, dan mulai menjalankan awal prosedur. Prosedur tersebut bertanggung jawab pada pengaksesan dan penggunaan jaringan. Setelah eksekusi, prosedur akan kembali ke alamat pemanggil. Proses pemanggil akan dilanjutkan.

Ketika proses pemanggil (*calling process*) memanggil sebuah prosedur, aksi yang dikerjakan oleh prosedur tersebut tidak sama dengan kode yang ditulis, tetapi kode yang memulai komunikasi jaringan. Proses pemanggil (*calling process*) harus menyambung pada mesin yang dituju, mengirim semua parameternya, dan menunggu balasan, mengerjakan sesuatu pada stack dan hasil. Inilah yang terjadi pada *client side stub*. Sedangkan pada *server side stub* harus menunggu pesan untuk menjalankan suatu prosedur. *server side stub* harus membaca parameter, dan mengubahnya ke bentuk yang cocok untuk mengeksekusi suatu prosedur. Setelah eksekusi, *server side stub* harus mengirim hasilnya ke proses pemanggil. Lihat Gambar 1.



Gambar 1. Skema proses menggunakan RPC [4]

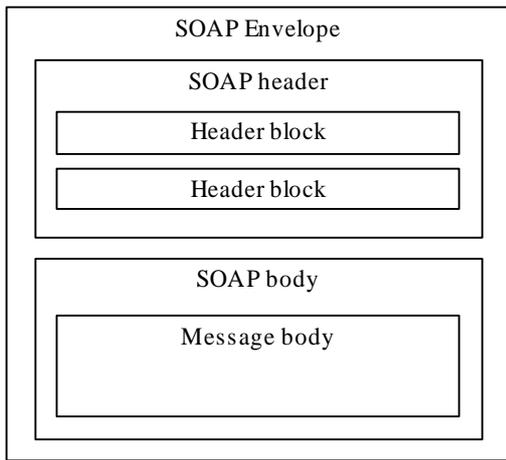
### SOAP (Simple Object Access Protocol) [5]

SOAP adalah protokol yang berisi sekumpulan aturan yang mengatur komunikasi antar komputer. Ide dasarnya adalah dua aplikasi, dengan tidak memperdulikan operating system, bahasa pemrograman, atau detail implementasi teknis yang lain, dapat menggunakan informasi secara bersama-sama dengan menggunakan pesan yang dikodekan dalam suatu cara dimana kedua aplikasi tersebut dapat mengerti. Pesan tersebut berupa data XML. Disini, SOAP berperan menyediakan standar cara untuk menyusun pesan XML tersebut. SOAP adalah XML, dengan kata lain, SOAP adalah aplikasi dari spesifikasi XML, dapat menggunakan standard XML seperti XML Schema dan XML Namespaces untuk pendefinisian dan fungsi/penggunaan.

Secara konseptual SOAP dapat dianggap sebagai DCOM versi XML. SOAP merupakan mekanisme lain yang memungkinkan penggunaan *remote procedure call*. SOAP bersifat netral platform, netral bahasa dan tidak bergantung pada suatu objek model. Sehingga SOAP-enabled distributed application dapat menjangkau beragam operating sistem, dimana terdiri dari objek yang berasal dari vendor yang berbeda, ditulis pada bahasa yang berbeda, dan didasarkan pada objek model yang berbeda.

### SOAP Messages

Pesan SOAP berbentuk seperti sebuah *envelope* yang berisi *header (optional)* dan *body (required)*, seperti ditunjukkan pada Gambar 2. *Header* berisi blok informasi yang berhubungan dengan bagaimana pesan tersebut diproses. Hal ini meliputi *pe-routing-an* dan *delivery setting, authentication* atau *authorization assertions, and transaction contexts*. *Body* berisi pesan sebenarnya yang dikirim dan diproses. Semua yang dapat ditampilkan dengan sintaks XML dapat dimasukkan dalam pesan *body*.



Gambar 2. Struktur pesan SOAP [5]

Setiap elemen *Envelope* harus berisi tepat satu elemen *Body*. Elemen *Body* dapat berisi sebanyak mungkin *child nodes* yang diperlukan. Isi dari elemen *Body* adalah pesan. Elemen *Body* ditentukan dalam suatu cara dimana dapat berisi valid dan well-formed XML yang telah dibatasi oleh suatu *namespace (qualified)*.

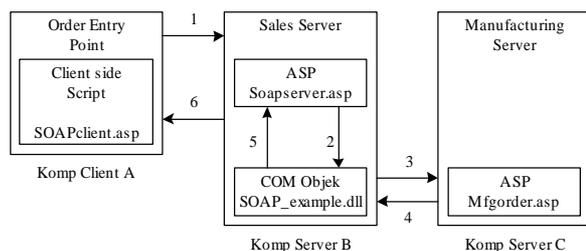
Jika sebuah *Envelope* berisi elemen *Header*, harus berisi tidak lebih dari satu, dan harus tampak pada *first child* dari *Envelope*, sebelum elemen *Body*. *Header* dapat berisi *valid, well-formed*, dan dibatasi dengan *namespace XML* dimana hendak dimasukkan oleh pencipta pesan SOAP.

Setiap elemen yang berada dalam *Header* disebut blok *header*. Tujuan dari blok *header* adalah untuk memberitahukan informasi yang berhubungan dengan pemrosesan pesan SOAP.

### Desain aplikasi

#### Desain Aplikasi Terdistribusi Pemesanan Barang Elektronik

Gambar 3 menunjukkan desain arsitektur dan aliran pesan aplikasi menggunakan SOAP mengenai aplikasi pemesanan barang elektronik. Pada aplikasi tersebut *client* akan menerima konfirmasi dari *Sales server* dan *Manufacturing Server* bahwa data telah diterima.



Gambar 3. Arsitektur dan aliran pesan aplikasi

Penjelasan kode Skenario yang terjadi:

- 1) pesanan dimasukkan dan di-submit (SOAP request),
- 2) sales server menerima pesanan,
- 3) sales server mengirim pesan ke manufacturing,
- 4) manufacturing server menerima pesan dan mengembalikan konfirmasi,
- 5) sales server menerima konfirmasi dan mengirimkan konfirmasinya ke client,
- 6) Client menampilkan detail konfirmasi. (SOAP responses)

#### Desain Aplikasi Kalkulasi SOAP-WSDL dan DCOM

Program kalkulasi ini ditujukan khusus untuk membandingkan program kalkulasi yang dibuat dengan menggunakan DCOM dan SOAP-WSDL. Desain yang digunakan kedua program sama, yang berbeda hanyalah cara pengimplementasian programnya. Dimana program menerima dua inputan, selanjutnya dilakukan proses penambahan, pengurangan, perkalian, pembagian. Ada dua bagian program, yaitu bagian server dan bagian client.

#### Program server

Program server memiliki source code yang sama, yang berbeda hanyalah ekstensinya, yakni untuk DCOM menggunakan file "kal\_dcom.exe" sedang untuk SOAP-WSDL menggunakan file "kalkulasi.dll". Pembuatan file ini dilakukan dengan menggunakan program Visual Basic 6, dimana pada saat "new project", dipilih "ActiveX Exe" atau "ActiveX DLL".

#### Program Client

Perbedaan kode baru tampak pada program kalkulasi client-nya. Berikut penjelasan program untuk kedua program kalkulasi dengan DCOM dan SOAP-WSDL.

##### □ Program Client DCOM

Pada saat program kalkulasi DCOM dijalankan, apabila tombol "Connect" ditekan, kan terjadi koneksi dengan server. Program melakukan penciptaan *instance object* dengan menggunakan fungsi "createobject", dengan inputan nama aplikasi dan tipe atau *class* objeknya. Setelah *instance object* terbentuk, *properties* dan *method* yang terdapat pada *instance object* tersebut dapat diakses. Saat *loading* program itu pula, dihitung waktu koneksi (penciptaan *instance object*) antara server dan client.

```
Private Sub koneksi_Click()
Dim start, finish
start = Timer
```

```
Set client =  
CreateObject("kal_dcom.c_kal_dcom")  
finish = Timer  
waktu.Text = CStr(finish - start)  
End Sub
```

Apabila tombol "Add" ditekan, akan dilakukan proses penambahan. Dalam prosedur ini, dilakukan pula penghitungan lama proses dengan menggunakan fungsi "timer". Fungsi "timer" ini akan mengembalikan nilai detik. Pada prosedur ini, apabila checkbox "cek\_repetisi" dicek, akan dilakukan pengulangan sebanyak seribu kali. Hal yang sama dilakukan pula pada prosedur pengurangan, perkalian, dan pembagian.

```
Private Sub cmdAdd_Click()  
    Dim I As Integer  
    Dim start, finish  
    If Cek_repetisi = 1 Then  
        start = Timer  
        For I = 0 To 1000  
            txtEquals.Text =  
CStr(client.Add(CDbl(txtA.Text),  
CDbl(txtB.Text)))  
        Next  
        finish = Timer  
        proses.Text = CStr(finish - start)  
    Else  
        start = Timer  
        txtEquals.Text =  
CStr(client.Add(CDbl(txtA.Text),  
CDbl(txtB.Text)))  
        finish = Timer  
        proses.Text = CStr(finish -  
start)  
    End If  
End Sub
```

#### □ Program Client SOAP-WSDL

Pada saat, program kalkulasi SOAP-WSDL dijalankan, apabila tombol "connect" ditekan, akan terjadi koneksi dengan server. Program menunggu konfirmasi apakah alamat url file "kalkulasi.wsdl" pada server telah benar. Ketika tombol "connect" ditekan, akan dilakukan *instancing* objek "mssoap" dengan class "soapclient", objek ini diimplementasikan pada file "mssoap1.dll", dimana merupakan salah satu file dari SOAP toolkit yang telah di-*install* sebelumnya pada komputer *client*. Setelah instance objek tercipta, instance akan memanggil *method* "mssoapinit", dimana menerima inputan alamat url dari file "kalkulasi.wsdl".

Saat *client* hendak mengirim SOAP *request* ke *server*, pertama-tama, diperlukan salinan file wsdl tersebut yang dimuat oleh method *mssoapinit* tersebut. Setelah melakukan inisialisasi, akan terbentuk suatu ikatan pada file wsdl tersebut, sehingga semua method yang dideskripsikan pada file "kalkulasi.wsdl" tersebut dapat dilakukan. Program akan menggunakan informasi dari

file wsdl ini untuk membentuk SOAP *request*, kemudian dikirimkan ke *server*. Server mengeksekusi operasi penambahan dan mengembalikan hasilnya ke *client* sebagai SOAP *response* yang kemudian hasilnya diolah oleh "soapclient" dan ditampilkan ke program.

```
Private Sub Form_Load()  
    sConnectedWSDL = "  
cbWSDL.Text =  
"http://hwe/kalkulasi/kalkulasi.wsdl"  
End Sub  
Private Sub connect()  
    If sConnectedWSDL <> cbWSDL.Text Then  
        Set Client =  
CreateObject("mssoap.SoapClient")  
        Client.mssoapinit cbWSDL.Text  
        sConnectedWSDL = cbWSDL.Text  
    End If  
End Sub  
Private Sub koneksi_Click()  
    Dim start, finish  
    start = Timer  
    connect  
    finish = Timer  
  
    waktu.Text = CStr(finish - start)  
End Sub
```

Saat penekanan tombol "Add" dilakukan, akan dijalankan prosedur berikut, dimana dilakukan proses penambahan. Dalam prosedur ini, dilakukan pula penghitungan lama proses dengan menggunakan fungsi "timer". Fungsi "timer" ini akan mengembalikan nilai bertipe *single* yang merepresentasikan nilai detik. Pada prosedur ini, apabila *checkbox* "cek\_repetisi" dicek, akan dilakukan pengulangan sebanyak seribu kali. Hal yang sama dilakukan pula pada prosedur pengurangan, perkalian, dan pembagian.

```
Private Sub cmdAdd_Click()  
    Dim I As Integer  
    Dim start, finish  
    On Error GoTo ErrorHandler  
    Me.MousePointer = vbHourglass  
  
    connect  
  
    If Cek_repetisi = 1 Then  
        start = Timer  
        For I = 0 To 1000  
            txtEquals.Text =  
CStr(Client.Add(CDbl(txtA.Text),  
CDbl(txtB.Text)))  
        Next  
        finish = Timer  
        proses.Text = CStr(finish -  
start)  
    Else  
        start = Timer  
        txtEquals.Text =  
CStr(Client.Add(CDbl(txtA.Text),  
CDbl(txtB.Text)))  
        finish = Timer  
        proses.Text = CStr(finish -  
start)  
    End If  
End Sub
```

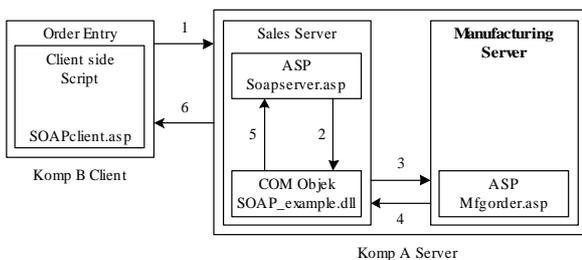
**Pengujian**

Secara prinsip, pengujian didasarkan pada desain jaringan komputer seperti pada Gambar 3, tetapi untuk penyederhanaan, pada pelaksanaan pengujian hanya digunakan dua komputer yaitu sebagai client dan server dengan spesifikasi pada Tabel 1.

Tabel 1. Konfigurasi Komputer

|                  | Komputer A Server                               | Komputer B Client  |
|------------------|---|--|
| Prosesor         | Pentium IV 1,8 Ghz                              | Pentium 150 Mhz  |
| Memori           | DDR333 256 Mb                                   | SDR 48 Mb  |
| Operating System | Microsoft Windows 2000 Server                   | Microsoft Windows 98   |
| Ethernet Adapter | On Board Realtek 100/10 Mbps                    | Surecom 100/10 Mbps  |
| IP Address       | 10.10.1.1                                       | 10.10.1.2  |
| Subnet Mask      | 255.0.0.0                                       | 255.0.0.0  |
| Program          | Microsoft SOAP Toolkit<br>Microsoft IE 5 keatas | Microsoft SOAP Toolkit, dcom98,<br>Dcm95cnfg,<br>Microsoft IE 5 keatas |

Pada konfigurasi ini, “soap\_client.asp” terletak pada komputer B, sedang “soap\_server.asp”, “soap\_example.dll”, “mfg\_order.asp” terletak pada komputer A. Hal ini dilakukan khusus ditujukan untuk menunjukkan bahwa “soap\_client.asp” dapat mengirimkan inputan yang dibutuhkan oleh “soap\_example.dll” serta pembuatan file XML oleh file mfg\_order.asp berhasil pula dilakukan. Lihat Gambar 4.



Gambar 4. Konfigurasi komputer untuk pengujian

**Pengujian Program Aplikasi Terdistribusi Pemesanan Barang Elektronik**

Pengujian ini dimulai dengan membuka *web browser*, kemudian mengisikan alamat URL yang menunjuk ke *file* “soap\_client.asp”, misalnya: “http://hwe/distapp\_single/soap\_client.asp”. Pada layar *browser* tampak terbagi dua bagian “Order Form” dan “Order Confirmation Info”. Bagian “Order Form” meminta inputan jumlah (*quantity*) dan jenis item, yakni: MP3 Player, Portables CD Player, Wireless PIM. Sedang bagian “Order Confirmation Info” akan menampilkan data kon-

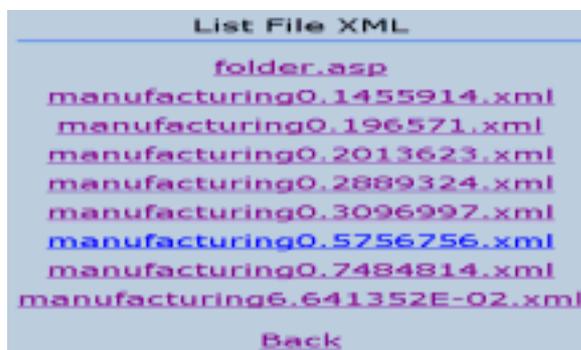
fimasi yang berasal dari komputer *client*, komputer *server sales*, komputer *server manufacturing*, juga konfirmasi *order* yang telah dilakukan

Proses dimulai dengan memasukkan jumlah barang yang diinginkan dan memilih jenis itemnya (MP3 Player, Portables CD Player, Wireless PIM). Setelah keduanya ditentukan, langkah berikut adalah menekan tombol “Place Order”. Beberapa saat kemudian, data-data akan tampak pada bagian kanan form, dimana data tersebut berasal dari komputer client, komputer server sales, komputer server manufacturing, juga akan muncul konfirmasi order yang telah dilakukan.



Gambar 5. Program aplikasi terdistribusi

Penghubung (*link*) “List File XML” pada sebelah kiri layar menunjuk ke file “folder.asp” yang terletak pada server manufacturing. File “folder.asp” digunakan untuk menampilkan file-file xml, yang dihasilkan oleh mfg\_order.asp pada server manufacturing. Lihat Gambar 6.



Gambar 6. List File XML

Apabila salah satu nama file tersebut ditekan, akan menunjuk kepada file XML tersebut, seperti pada Gambar 7.

```
<?xml version="1.0" ?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/" xmlns:m="http://www.muhimbi.com/Services/ASPEngineering/Manufacturing" >
  <SOAP:Body>
    <m:order>
      <quantity>111</quantity>
      <item>7854</item>
      <description>MP3 Player</description>
      <originator>
        <computer>16.16.1.1</computer>
        <user>(not determined)</user>
      </originator>
      <manufacturer>
        <engine>78711</engine>
        <status>Accepted</status>
        <updated>13/12/2002 0:25:58</updated>
        <updatedBy>ASP running on Computer: iwe</updatedBy>
      </manufacturer>
    </m:order>
  </SOAP:Body>
</SOAP:Envelope>
```

Gambar 7. File XML pada Manufacturing server



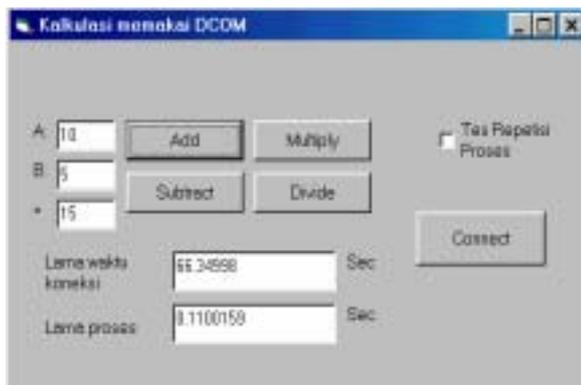
Gambar 9. Operasi kalkulasi dengan SOAP-WSDL

### Pengujian Program Kalkulasi

Pengujian dilakukan menggunakan DCOM dan SOAP-WSDL, dengan menguji waktu loading dan kecepatan pengerjaan proses di client dengan melakukan looping seribu kali dengan inputan yang sama.

#### Pengujian Program Kalkulasi Menggunakan DCOM

Dengan menggunakan file “client\_kal.exe” versi DCOM, apabila program kalkulasi DCOM dijalankan, program akan melakukan koneksi dengan server apabila tombol “Connect” ditekan. Sekaligus juga dilakukan perhitungan waktu koneksi dalam satuan detik. Pengisian inputan angka dilakukan pada kotak teks A dan B, dimana akan dilakukan operasi penambahan, pengurangan, perkalian, pembagian. Tes repetisi dilakukan dengan memberikan tanda cek pada check box yang disediakan. Lihat Gambar 8.



Gambar 8. Operasi kalkulasi dengan DCOM

#### Pengujian Program Kalkulasi Menggunakan SOAP-WSDL

Dengan menggunakan file “client\_kal.exe” versi SOAP-WSDL, program client kalkulasi SOAP-WSDL ini dijalankan. Penggunaan program ini sama dengan pada program untuk DCOM. Lihat Gambar 9.

### Hasil Pengujian

Bagian ini berisi hasil pengujian dari program kalkulasi SOAP-WSDL dan DCOM. Untuk masalah waktu koneksi, dilakukan pengujian pada saat komputer client baru pertama kali melakukan koneksi. Bila diambil data rata-rata program kalkulasi DCOM 1.99 kali lebih cepat daripada program kalkulasi SOAP-WSDL. Lihat Tabel 2.

Tabel 2. Waktu Koneksi Program Kalkulasi

| Koneksi         | DCOM (sec) | SOAP-WSDL (sec) | SOAP-WSDL lebih lambat (kali) |
|-----------------|------------|-----------------|-------------------------------|
| 1               | 1.49       | 3.57            | 2.40                          |
| 2               | 0.93       | 1.16            | 1.25                          |
| 3               | 0.61       | 1.1             | 1.80                          |
| 4               | 0.6        | 1.27            | 2.12                          |
| 5               | 0.61       | 1.32            | 2.16                          |
| Waktu rata-rata | 0.85       | 1.68            | 1.99                          |

Untuk masalah waktu pengerjaan proses repetisi, program kalkulasi SOAP-WSDL rata-rata 9.92 kali lebih lambat daripada program kalkulasi DCOM dapat dilihat pada Tabel 3.

Tabel 3. Waktu rata-rata Program Kalkulasi

| Waktu rata-rata | SOAP-WSDL | DCOM     | SOAP-WSDL lebih lambat |
|-----------------|-----------|----------|------------------------|
| Add             | 35.29 sec | 3.57 sec | 9.89                   |
| Substract       | 35.44 sec | 3.55 sec | 9.98                   |
| Multiply        | 35.34 sec | 3.57 sec | 9.90                   |
| Devide          | 35.53 sec | 3.58 sec | 9.92                   |
| Total rata-rata | 35.4 sec  | 3.57 sec | 9.92                   |

Penyebab lambatnya proses di SOAP-WSDL bukan pada proses perhitungan di programnya, tetapi disebabkan karena terjadi proses pada SOAP request dan SOAP response antara client dan server, seperti dijelaskan pada sub bab 3.2 bagian Program Client.

## Kesimpulan

1. Dari hasil pengujian mengenai “Aplikasi Terdistribusi Pemesanan Barang Elektronik”, dapat disimpulkan aplikasi tersebut berhasil dijalankan pada dua komputer client dan server, sehingga diharapkan dapat dikembangkan untuk dapat mengatasi masalah yang lebih kompleks.
2. Dari hasil pengujian program kalkulasi, dapat dilihat bahwa waktu rata-rata pengkoneksian dengan server (*instancing object*) antar *client-server* pada program yang menggunakan SOAP-WSDL, lebih lambat 1.99 kali dari program DCOM, juga ketika melakukan proses yang banyak dengan repetisi seribu kali *performance* program SOAP-WSDL lebih lambat 9.92 kali dari program DCOM.

## Daftar Pustaka

- [1] Baartse, Mark., et.al. *Professional ASP XML*. Birmingham, UK: Wrox Press, 2000.
- [2] Balena, Francesco. *Programming Microsoft Visual Basic 6.0*. Washington: Microsoft Press, 1999.
- [3] Kirtland, Mary. *Designing Component-Based Applications*. Washington : Microsoft Press, 1998.
- [4] Pattison, Ted. *Programming Distributed Applications with COM and Microsoft Visual Basic 6.0.*, Washington : Microsoft Press, 1998.
- [5] Tidwell, Doug., Snell, James., and Kulchenko, Pavel., *Programming Web Services with SOAP.*: O’Reilly, 2001.