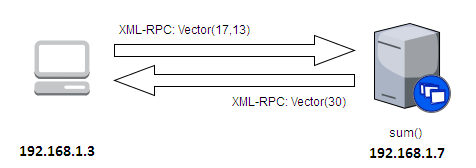
**Sistem Paralel dan Terdistribusi (CSH3J3)**

**Tugas 2: Remote Procedure Call**

1. Remote Procedure Call

Pada tugas ini, kita akan mengimplementasikan RPC (Remote Procedure Call) menggunakan java. Secara ringkas RPC adalah memanggil prosedur/fungsi yang berada di komputer lain.

Contoh: terdapat sebuah klien yang akan memanggil prosedur (bernama: *sum*) di server. Server kemudian mengembalikan hasil penjumlahan.



Dengan menggunakan RPC, seolah-olah klien memanggil prosedur lokal padahal klien memanggil prosedur (sum) yang berada di server. Klien cukup mengetahui nama prosedur yang akan dipanggil dan memberikan parameter yang sesuai.

Terdapat banyak cara untuk implementasi RPC. Pada tugas ini kita akan mengimplementasikan RPC akan menggunakan XML-RPC (https://en.wikipedia.org/wiki/XML-RPC; www.**xmlrpc**.com ) dan Java.

*“XML-RPC is among the simplest and most foolproof web service approaches that makes it easy for computers to call procedures on other computers.”*

XML-RPC: dapat memanggil fungsi/procedure dari komputer lain melalui jaringan, menggunakan protokol HTTP. XML-RPC dapat di download pada https://archive.apache.org/dist/ws/xmlrpc/binaries/xmlrpc-2.0.zip

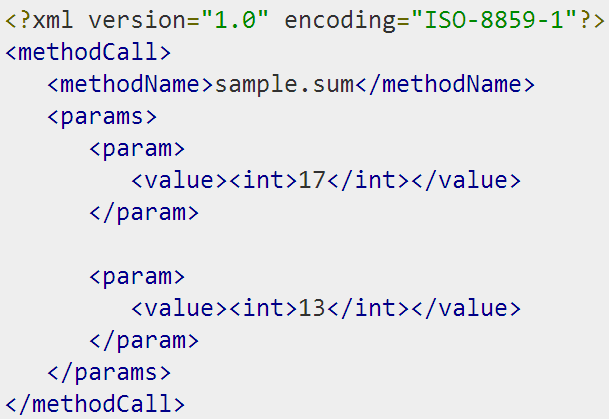
1. XML-RPC Client

Kita akan membuat klien bernama *JavaClient* yang akan memanggil fungsi *sum* yang berada di server.



Penjelasan;

* Baris 1: kita harus import java.util karena kita akan menggunakan vector dalam java. Vector adalah array dinamis. Pada array biasa kita tidak bisa mengubah ukuran array jika sudah dideklarasikan. Dengan vector, ukuran array dapat diubah sesuai kebutuhan.
* Baris 2: gunakan library XML-RPC buatan apache versi 2.0. Versi xmlrpc-3.0 tidak mempunyai webserver.
* Baris 9: [http://192.168.1.7:8888/RPC2](http://localhost/RPC2) adalah alamat server/komputer yang menyediakan remote prosedur/fungsi. Kita juga bisa mengubah port default (80) menjadi port lain sesuai keinginan (8888). Pada beberapa OS, memerlukan hak akses admin untuk membuka port <1024
* Baris 10-13: Inisiasi vector kemudian menambahkan parameter 17 dan 13 ke dalam vector. Vector inilah yang akan dikirimkan ke server dalam bentuk xml menggunakan bantuan *commons-codec-1.10.jar* untuk encode/decode. Berikut adalah bentuk xml yang dikirim oleh 192.168.1.3. Karena digunakan XML-RPC maka kita tidak perlu susah-susah membuat format XMLnya.



* Baris 15-17: Kita memanggil prosedur “sum” yang berada di server. “sample” adalah handler yang berada di server. Klien harus mengetahui nama prosedur yang akan dipanggil dan memberikan jumlah paramater yang sesuai. Karena yang diterima klien masih dalam bentuk vector, kita harus mengubahnya menjadi integer.

1. XML-RPC Server

Berikut adalah source code untuk server dengan prosedur “sum”.



Penjelasan:

* Baris 1: server juga harus mengimport XML-RPC
* Baris 5-7: fungsi untuk menambahkan 2 buah integer
* Baris 15: kita membuat webserver. Webserver yang kita gunakan adalah webserver minimal yang berasal dari xmlrpc-2.0.jar. Webserser ini selalu beralamatkan http://IP:port/RPC2
* Baris 16-17: server akan membuat handler bernama “sample” dan mencoba menjalankan webservice pada port 8888
* Server akan terus berjalan hingga kita mematikan server secara manual.

1. Running
   1. Compile JavaClient pada komputer B. Gunakan classpath xmlrpc-2.0.jar dan commons-codec-1.10.jarsaat mengcompile. commons-codec-1.10.jar digunakan untuk proses decoding dan encoding. Ubah source code JavaClient sehingga merujuk pada alamat IP server A.

*javac -cp .:xmlrpc-2.0.jar:commons-codec-1.10.jar JavaClient.java*

* 1. Compile JavaServer pada komputer A. Gunakan classpath xmlrpc.jar saat mengcompile

*javac -cp .:xmlrpc-2.0.jar:commons-codec-1.10.jar JavaServer.java*

* 1. Jalankan JavaServer pada komputer A

*java -cp .:xmlrpc-2.0.jar:commons-codec-1.10.jar JavaServer*

* 1. Jalankan JavaClient pada komputer B

*java -cp .:xmlrpc-2.0.jar:commons-codec-1.10.jar JavaClient*

***Kode telah berhasil dijalankan pada Sistem operasi Linux Mint 17 Rosa.***

**Tugas**

1. Buatlah 3 klien dengan parameter yang berbeda-beda. Jalankan masing-masing klien dari komputer yang berbeda. Apakah setiap klien memberikan output yang benar?
2. Buatlah program RPC yang dapat menambah, mengurangi, mengali dan membagi.
3. Buatlah program RPC yang dapat menjumlahkan n bilangan.