

**Khrisna Widhi D**

**155150201111275**

1. SOAP (Simple Object Access Protocol) merupakan sebuah protokol yang dipergunakan untuk bertukar pesan berbasis XML. Melewati sebuah 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 pertukaran data, maka SOAP dapat berkomunikasi dengan berbagai aplikasi meskipun terdapat perbedaan sistem operasi, teknologi, dan bahasa pemrogramannya.

REST adalah sebuah software yang memiliki arsitektur untuk sebuah sistem terdistribusi salah satu contohnya adalah sebuah web. REST sendiri memiliki perkembangan sebagai model desain dari web service yang mendominasi pada saat ini. Melibatkan proses pembacaan laman web tertentu yang memuat sebuah file XML atau JSON. File inilah yang menguraikan dan memuat konten yang hendak disajikan. Setelah melalui sebuah proses definisi tertentu, konsumen akan bisa mengakses antarmuka aplikasi yang dimaksudkan.

Uraian (1)	SOAP (2)	REST (3)
Protokol komunikasi	HTTP, HTTPS, SMTP, FTP	HTTP, HTTPS
Penggunaan bandwidth	Dalam jumlah request yang banyak, relatif boros bandwidth. Hal ini karena banyaknya markup dalam penulisan format XML	Relatif hemat bandwidth, karena markup-markup ekstra seperti pada XML tidak dipakai
Tren penggunaan	Banyak mulai beralih ke REST, meski masih tetap ada yang mempertahankan, misalnya untuk integrasi aplikasi ke sistem legasi pada sebuah perusahaan.	Mulai populer, banyak dipakai oleh penyedia web servis terkemuka, seperti twitter, yahoo!, flickr, bloglines, technorati, google, amazon, eBay, dsb
Aturan penulisan	Ketat, mengikuti spesifikasi XML (SOAP v1.2)	Tidak ada spesifikasi khusus
Format respon	XML dengan spesifikasi SOAP. Agak sulit bagi kita untuk membaca langsung dan memahaminya.	XML, JSON, atau format plain teks lainnya. Hal ini memudahkan penerima respon membaca dan memahaminya.
Attachment file	Bisa (karena dapat mengembalikan respon dalam format binary)	Tidak bisa
Sifat web servis pada umumnya	Tertutup, lebih ditujukan untuk vendor atau perusahaan tertentu	Terbuka, bisa diakses siapa saja
Caching web	Relatif sulit	Mudah, karena menggunakan URI
Penggunaan standar	Standar lama (XML, HTTP) dan baru (SOAP) digunakan bersamaan	Standar yang sudah ada, seperti XML dan HTTP
Tool pengembangan	Banyak, baik komersial maupun opensource	Beberapa, karena tidak begitu dibutuhkan
Tool manajemen	Perlu, bahkan kadang harganya mahal	Menggunakan tool yang sudah ada pada sistem jaringan

Persamaan dari keduanya adalah

- Keduanya memiliki fungsi yang sama yaitu bisa digunakan untuk berkomunikasi dengan cara bertukar pesan dengan web service.
- Dapat menggunakan beberapa Protokol web yang serupa seperti (HTTP, HTTPS)

2. Code On Demand merupakan salah satu permasalahan opsional yang terdapat di dalam REST. Yang memungkinkan klien untuk meningkatkan sebuah fleksibilitas karena seharusnya serverlah yang dapat menentukan itu semua. Contohnya dengan Code-On-Demand klien dapat mengunduh javascript, java applet atau bahkan aplikasi flash untuk mengenkripsi komunikasi sehingga server tidak menyadari setiap rutin / kunci enkripsi yang digunakan dalam proses ini. Namun, menggunakan COD mengurangi visibilitas, itulah sebabnya mengapa konstrain ini opsional. Juga, tidak setiap API membutuhkan fleksibilitas semacam ini.

Semua interaksi sebaiknya dibangun dengan interface yang seragam. RESTful service menampilkan semua resource dan interaksinya dengan interface yang seragam, tidak seperti RPC yang menampilkan fungsi yang ada melalui method yang bisa dipanggil secara remote. Interface yang memiliki keseragaman merupakan sebuah konstrain yang sangat dasar pada design REST. Berikut adalah beberapa jenis konstrain:

- **Resource identification in request**  
Semua informasi yang mempunyai nama adalah resource diidentifikasi oleh sebuah URL
- **Resources manipulation through representations**  
Ketika pengguna mengetahui sebuah representasi dari suatu resource maka bisa mengubah dan menghapus sebuah resources.
- **Self descriptive messages**  
Setiap pesan yang memiliki sebuah informasi yang memenuhi untuk mendiskripsikan bagaimana sebuah pesan berproses agar servernya mengetahui bagaimana cara pesan tersebut terproses.
- **Hypermedia as the engine of application state**  
Dalam mengakses API harus sama seperti ketika mengakses sebuah web pages, dimana interaksi dengan server harus dimonitor oleh hypermedia seluruhnya