**REST, REST API dan Client-Server**

**Definisi REST**

REST (REpresentational State Transfer) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protocol untuk komunikasi data. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.

Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

Dari beberapa sumber bahwa terdapat kelemahan dan kelebihan dari penggunaan REST ini :

**Kelemahan REST:**

1. 1.Mengasumsi model point-to-point komunikasi — tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara.
2. Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, dll, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan (“dipecahkan sendiri”).
3. Berkaitan dengan model transport HTTP. (codepolitan.com)

**Kelebihan REST:**

1. Bahasa dan platform agnostic.
2. Lebih sederhana/simpel untuk dikembangkan ketimbang SOAP.
3. Mudah dipelajari, tidak bergantung pada tools.
4. Ringkas, tidak membutuhkan layer pertukaran pesan (messaging) tambahan.
5. Secara desain dan filosofi lebih dekat dengan web.(codepolitan.com)

**Perbedaan REST dengan RESTfull API**

Dalam tulisan diatas telah dibahas sedikit tentang REST. Dalam REST terdapat juga RESTful API itu merupakan salah satu model implementasi dari web service. RESTful API merupakan implementasi dari API. RESTful itu protokol/aturan untuk melakukan REST. Jadi RESTful itu udah pasti REST, namun REST belum tentu bisa disebut RESTful. Walaupun terlihat sama tetapi mereka berbeda.

**Apa itu API**

API adalah singkatan dari Application Programming Interface. Merupakan suatu “penghubung” yang memungkinkan suatu aplikasi untuk berinteraksi dengan aplikasi lainnya dan berbagi data. Output dari RESTful ada XML dan JSON.

**JSON (JavaScript Object Notation), beberapa ciri-ciri JSON.**

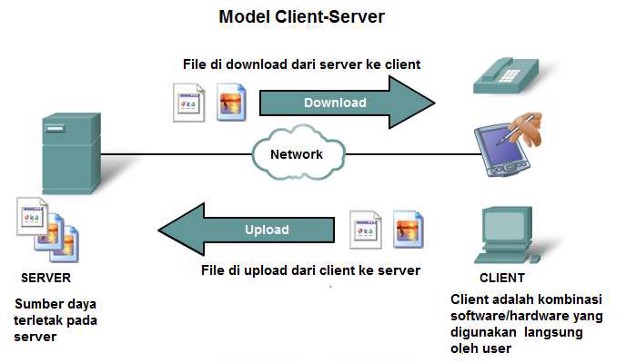
* Mudah di urai, bahkan oleh javascript pada sisi client, hanya object biasa
* Data transport lebih kecil
* Proses urai lebih cepat

**XML (Extensible Markup Language), beberapa ciri-ciri XML**

* Butuh XML Document untuk mengurai seperti Xpath
* Data transport lebih besar
* Proses urai lebih lambat

**Client-Server**

Seperti yang telah dibahas sebelumnya ketika software/perangkat lunak dijalankan terjasi sebuah pertukaran data atau informasi. Proses tersebut terjadi antara dua pihak yaitu antara client dan server.



Dalam model klien/server, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, tetapi masih merupakan sebuah kesatuan yakni komponen klien dan komponen server. Komponen klien juga sering disebut sebagai front-end, sementara komponen server disebut sebagai back-end.

Komponen klien dari aplikasi tersebut dijalankan dalam sebuah workstation dan menerima masukan data dari pengguna. Komponen klien tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan teknologi pemrosesan tertentu dan mengirimkannya kepada komponen server yang dijalankan di atas mesin server, umumnya dalam bentuk requestterhadap beberapa layanan yang dimiliki oleh server.

Komponen server akan menerima request dari klien, dan langsung memprosesnya dan mengembalikan hasil pemrosesan tersebut kepada klien. Klien pun menerima informasi hasil pemrosesan data yang dilakukan server dan menampilkannya kepada pengguna, dengan menggunakan aplikasi yang berinteraksi dengan pengguna.

**Karakteristik Client-Server**

Karakteristik dari Arsitektur Client Server yaitu:

1. Servis (layanan)Hubungan antara proses yang berjalan pada mesin yang berbeda. Pemisahan fungsi berdasarkan ide layanannya. Server sebagai provider, client sebagai konsumen.
2. Sharing resources (sumber daya), Server bisa melayani beberapa client pada waktu yang sama, dan meregulasi akses bersama untuk share sumber daya dalam menjamin konsistensinya.
3. Asymmetrical protocol (protokol yang tidak simetris), Many-to-one relationship antara client dan server. Client selalu menginisiasikan dialog melalui layanan permintaan, dan server menunggu secara pasif request dari client.
4. Transparansi lokasi, Proses yang dilakukan server boleh terletak pada mesin yang sama atau pada mesin yang berbeda melalui jaringan.Lokasi server harus mudah diakses dari client.
5. Mix-and-Match, Perbedaan server client platforms
6. Pesan berbasiskan komunikasi, Interaksi server dan client melalui pengiriman pesan yang menyertakan permintaan dan jawaban.
7. Pemisahan interface dan implementasi, Server bisa diupgrade tanpa mempengaruhi client selama interface pesan yang diterbitkan tidak berubah.

**Model Arsitektur Client-Server**

Arsitektur client server dapat dibedakan menjadi 4 model, yaitu single-tier (satu lapis), two-tier (dua lapis), three-tier (3 lapis) dan multi-tier. Berikut penjelasan dari ketiga model arsitektur client server tersebut di atas:

* Arsitektur Single-tier (Satu Lapis)

Semua komponen produksi dari sistem dijalankan pada komputer yang sama pada arsitektur single tier ini. Model single tier adalah model yang sederhana, mudah digunakan pengguna (user) dan paling sedikit memiliki alternatif. Kelemahan dari arsitektur ini adalah kurang aman dan kurang memiliki skalabilitas.

* Arsitektur Two-tier (Dua Lapis)

Pengolahan informasi pada arsitektur ini dapat dibagi menjadi dua, yaitu sistem user interface (antarmuka pengguna) lingkungan dan lingkungan server manajemen database. Arsitektur two tier memiliki tingkat kemanan yang lebih tinggi dan terukur daripada arsitektur single-tier. Arsitektur ini memiliki database pada komputer yang terpisah dan hal tersebut menyebabkan arsitektur ini dapat meningkatkan kinerja keseluruhan situs. Arsitektur two- tier memiliki kelemahan, yaitu biayanya yang mahal, arsitekturnya yang kompleks, tidak adanya pembaruan kode, skalabilitasnya kurang dan tingkat kemanannya kurang. Di samping itu, kelebihan dari arsitektur two tier adalah mudah digunakan oleh pengguna, dapat menangani database server secara khusus dan bisnis lingkup kecil sangat cocok menggunakan arsitektur ini.

* Arsitektur Three-tier (Tiga Lapis)

Karena arsitektur sebelumnya memiliki cukup banyak kelemahan, maka dikembangkanlah arsitektur three tier ini yang akan membantu mengatasi kelemahan dari arsitektur two-tier. Arsitektur three-tier memiliki 3 lapisan. Kelebihan dari arsitektur ini adalah memiliki skala yang besar, transfer informasi antara web server dan server database optimal, tidak akan menyebabkan lapisan lain terkontaminasi salah jika salah satu lapisan terdapat keslahan. Dan kekurangannya, arsitektur ini lebih sulit untuk merancang, lebih sulit untuk mengatur dan lebih mahal.

* Arsitektur Multi-Tier

adalah suatu metode yang sangat mirip dengan Three Tier. Bedanya, pada Multi-Tier akan diperjelas bagian UI (User Interface) dan Data Processing. Yang membedakan arsitektur ini adalah dengan adanya Business Logic Server. Database Server dan Bussines Logic Server merupakan bagian dari Data Processing, sedangkan Application Server dan Client/Terminal merupakan bagian dari UI. Business Logic Server biasanya masih menggunakan bahasa pemrograman terdahulu, seperti COBOL. Karena sampai saat ini, bahasa pemrograman tersebut masih sangat mumpuni sebagai business process.

**Kelebihan dan Kekurangan Client-Server Secara Umum:**

**Kelebihan atau Keunggulan dari sistem jaringan ini adalah:**

1. Kecepatan akses lebih tinggi karena penyediaan fasilitas jaringan dan pengelolaannya dilakukan secara khusus oleh satu komputer (server) yang tidak dibebani dengan tugas lain seperti sebagai workstation.
2. Sistem keamanan dan administrasi jaringan lebih baik, karena terdapat sebuah komputer yang bertugas sebagai administrator jaringan, yang mengelola administrasi dan sistem keamanan jaringan.
3. Sistem backup data lebih baik, karena pada jaringan client-server backup dilakukan terpusat di server, yang akan membackup seluruh data yang digunakan di dalam jaringan.
4. Lebih mudah pengaturannya bila networknya besar karena administrasinya disentralkan.
5. Semua data dapat dibackup pada satu lokasi sentral.

**Kelemahan dalam jaringan Server — Client adalah sebagai berikut:**

1. Biaya operasional relatif lebih mahal.
2. Diperlukan adanya satu komputer khusus yang berkemampuan lebih untuk ditugaskan sebagai server.
3. Kelangsungan jaringan sangat tergantung pada server. Bila server mengalami gangguan maka secara keseluruhan jaringan akan terganggu.
4. Membutuhkan software NOS yang mahal contoh : NT atau server Windows 2000, XP,Novell, UNIX.
5. Membutuhkan hardware yang lebih tinggi dan mahal untuk mesin server.
6. Membutuhkan administrator yang professional.
7. Mempunyai satu titik lemah jika menggunakan satu server, data user menjadi tak ada jika server mati.

**Kelebihan dan Kekurangan Client-Server Berdasarkan Tipenya**

1. Standalone (one-tier)

Pemrosesan data dilakukan pada mainframe.

Kelebihan

* + Sangat mudah
  + Cepat dalam merancang dan mengaplikasikan

Kekurangan

* Skala kecil
* Susah diamankan
* Menyebabkan perubahan terhadap salah satu komponen diatas tidak mungkin dilakukan, karena akan mengubah semua bagian.
* Tidak memungkinkan adanya re-usable component dan code.
* Cepat dalam merancang dan mengaplikasikan

1. Client/Server (two tier)

Pemrosesan data pada sebuah aplikasi terjadi pada client dan server.

Kelebihan

* Mudah
* Menangani Database Server secara khusus
* Relatif lebih sederhana untuk di develop dan diimplementasikan.
* Lebih cocok diterapkan untuk bisnis kecil.

Kekurangan

* Kurangnya skalabilitas
* Koneksi database dijaga
* Tidak ada keterbaharuan kode
* Tidak ada tingkat menengah untuk menangani keamanan dan transaksi skala kecil.
* Susah di amankan.
* Lebih mahal.

1. Three Tier

Arsitektur ini pengembangan arsitektur Client Server. Arsitektur Three Tier ini terdapat Application Server yang berdiri di antara Client dan Database Server. Contoh dari Application server adalah IIS, WebSphere, dan sebagainya.

Kelebihan

* Segala sesuatu mengenai database terinstalasikan pada sisi server, begitu pula dengan pengkonfigurasiannya. Hal ini membuat harga yang harus dibayar lebih kecil.
* Apabila terjadi kesalahan pada salah satu lapisan tidak akan menyebabkan lapisan lain ikut salah
* Perubahan pada salah satu lapisan tidak perlu menginstalasi ulang pada lapisan yang lainnya dalam hal ini sisi server ataupun sisi client.
* Skala besar.
* Keamanan dibelakang firewall.
* Transfer informasi antara web server dan server database optimal.
* Komunikasi antara system-sistem tidak harus didasarkan pada standart internet, tetapi dapat menggunakan protocol komunikasi yang lebvih cepat dan berada pada tingkat yang lebih rendah.
* Penggunaan middleware mendukung efisiensi query database dalam SQL di pakai untuk menangani pengambilan informasi dari database.

Kekurangan

* Lebih susah untuk merancang
* Lebih susah untuk mengatur
* Lebih mahal

1. Multi-Tier

Arsitektur Multi-Tier mirip dengan Three Tier. Kelebihan arsitektur ini dilengkapi UI (User Interface) dan Data Processing. Khususnya dengan adanya Business Logic Server.

Kelebihan

* Dengan menggunakan aplikasi multi-tier database, maka logika aplikasi dapat dipusatkan pada middle-tier, sehingga memudahkan untuk melakukan control terhadap client-client yang mengakses middle server dengan mengatur seting pada dcomcnfg.
* Dengan menggunakan aplikasi multi-tier, maka database driver seperti BDE/ODBC untuk mengakses database hanya perlu diinstal sekali pada middle server, tidak perlu pada masing-masing client.
* Pada aplikasi multi-tier, logika bisnis pada middle-tier dapat digunakan lagi untuk mengembangkan aplikasi client lain,sehingga mengurangi besarnya program untuk mengembangkan aplikasi lain. Selain itu meringankan beban pada tiap-tiap mesin karena program terdistribusi pada beberapa mesin.
* Memerlukan adaptasi yang sangat luas ruang lingkupnya apabila terjadi perubahan sistem yang besar.

Kekurangan

* Program aplikasi tidak bisa mengquery langsung ke database server, tetapi harus memanggil prosedur-prosedur yang telah dibuat dan disimpan pada middle-tier.
* Lebih mahal