

## 1. Introduction to Web Development

Lain dengan desktop application (stand-alone application) dan mobile application, web application tidak membutuhkan proses instalasi di device untuk digunakan, cukup dengan menggunakan web-browser yang seluruh device miliki. Device-device ini disebut juga sebagai client-side/ user-side/ local-side, sedangkan segala aktivitas yang bekerja di server di sebut juga server-side.

Pada desktop application dan mobile application, seluruh aplikasi di implement pada client-side, tetapi tidak kasusnya dengan web-application. Web application seluruh aplikasinya di implement pada server-side, tetapi system harus selalu menemukan caranya, bagaimana seluruh aplikasi yang di implement pada server-side bisa dipakai dan diakses oleh user di client-side dengan menggunakan web-browser seperti Google Chrome, Safari, Microsoft Edge, Fire Fox dan Opera.

Pada desktop application dan mobile application, Java, C#, C++, Visual Basic dan lain-lain berjalan dan di eksekusi pada client-side. Di sana pula UI(User-Interface) yang di jalankan oleh aplikasi-aplikasi tersebut langsung ditampilkan untuk user. Tetapi pada web-application, web-browser tidak akan mengenali code atau run-time language dari Java, C# dan lain sebagainya. Web-browser hanya akan mengenali 3 macam jenis code: Mark up Language (HTML & XML), Styling script (CSS) dan ECMA (javascript & JSON).

Bisa disimpulkan:

Client-Side: Semua yang bekerja dan berjalan di dalam computer user atau dalam dunia web bisa dibilang semua yang bekerja di dalam web-browser si user.

Server-Side: Semua yang bekerja dan berjalan di dalam computer server atau bekerja di virtual local server.

Dan pada dunia web application, semua programming language yang sering kamu dengar seperti C#, C++, VB, Java, PHP, Python, Ruby adalah bahasa pemrograman yang bekerja di dalam Server-Side. Begitu juga dengan query language (SQL) dalam DBMS (Database Management System) seperti Microsoft SQL Server, Oracle, MySQL, PostgreSQL juga bekerja pada Server-Side.

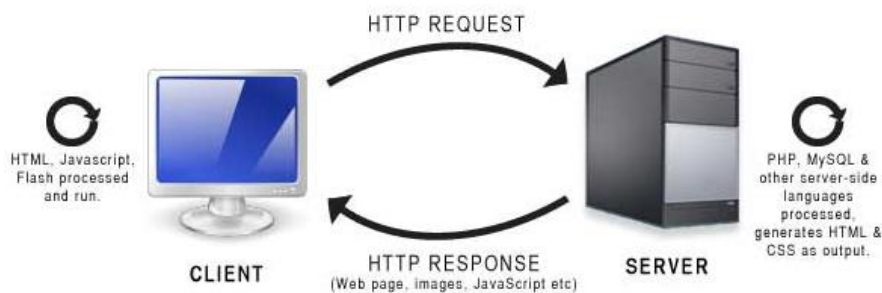
## 2. Client-Side vs Server-Side

Web application menggunakan Internet dan HTTP Protocol untuk meng-komunikasikan Client-side dengan Server-side. Komunikasi ini akan berjalan secara bergantian, kirim-mengirim/balas-membalas sama seperti 2 orang yang sedang chat, sms atau berkomunikasi dengan walkie-talkie.

Saat kita berbicara dengan teman kita satu sama lain, kita bisa mengerti yang mereka bicarakan, itu disebabkan adanya 2 hal, media dan protocol. Media yang kita gunakan untuk berbicara dengan teman kita adalah udara, karena udara bisa menghantarkan getaran, maka telinga kita bisa mendengar. Tapi kita tidak bisa mengerti apa yang teman kita bicarakan dalam bahasa asing yang kita belum pahami, walaupun kita bisa mendengarnya. Bahasa dalam hal ini adalah protocol, yaitu suatu tata-cara untuk memahami code yang di sampaikan. Mulut digunakan sebagai alat untuk meluncurkan suatu pesan yang didistribusi media dan dibungkus dalam protocol, dan telinga digunakan untuk menerima komunikasi yang dihantarkan media, lalu dengan otak dan pikiran, kita bisa menerjemahkan protocol komunikasinya.

Sama seperti pada radio, telfon, sandi-morse, seluruh bentuk komunikasi harus menggunakan media dan protocol agar bisa dipahami. Microphone dan Earphone pada telfon, sama seperti mulut dan telinga pada manusia. Lalu gelombang transversal atau longitudinal dalam ketukan tertentu akan di translate menjadi kata-kata yang bisa dipahami.

Sama seperti pada web application, internet adalah media yang digunakan untuk client-side dan server-side berkomunikasi, dan http adalah protocol yang digunakan agar server-side dan client-side bisa memahami satu sama lain. HTTP sendiri kepanjangan dari Hypertext Transfer Protocol, dimana protocol sendiri artinya adalah bahasa/aturan/atau tata cara suatu informasi di ucapkan dan didengar dalam bahasa Hypertext.



Saat Client-side berbicara kepada server-side, proses itu disebut juga dengan HTTP Request. HTTP Request adalah URL atau Link yang kalian masukan di browser atau sebuah link yang anda click di tampilan web. Disebut sebagai HTTP Request karena bentuknya adalah permintaan dari client-side kepada server-side untuk menunjukan halaman yang baru dengan data yang baru. Saat server-side memberikan data atau halaman yang user minta kepada client-side, maka proses itu disebut juga sebagai HTTP Response.

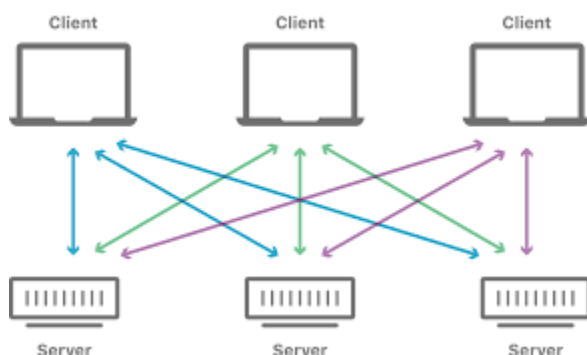
Kesimpulan:

HTTP Request adalah saat Client berbicara dan Server mendengar.

HTTP Response adalah saat Server berbicara dan Client mendengar.

Karena HTTP Request dan HTTP Response komunikasinya satu persatu saling balas membalas, pada saat koneksi internet putus tampilan pada web-browser anda tidak langsung menghilang dengan kabar Page Not Found, terkecuali anda refresh atau click link lain.

Dengan menggunakan teknologi web application seperti ini, artinya satu aplikasi pada server-side bisa melayani lebih dari satu computer sekaligus tanpa setiap komputer harus meng-install aplikasi tersebut, dan juga itu artinya satu computer bisa mengakses lebih dari 1 server dimana semuanya memiliki aplikasi yang berbeda-beda. Ini lah yang membuat web application begitu powerful ketimbang dengan desktop application atau mobile application.



Semua Server-Side bersifat secure dan tersembunyi dari pandangan user, hanya akses ke dalam server yang bisa melihat semua aktifitas Server-Side, lain dengan Client-side. Seluruh program pada client-side akan terbuka jelas dan bisa diakses user lewat browser. Sebenarnya browser anda tidak akan tahu persis atau peduli apakah aplikasi web tersebut dibuat dengan php, .net, java, ruby, python atau yang lainnya, yang browser anda perdulikan adalah hanya 3 hal yang saya sudah sebelumnya, yaitu: Markup language, Styling Script dan ECMA script (web scripting).

Hal yang kita akan pelajari pada pelajaran ini adalah 2 dari ketiga hal tersebut:

**HTML: bertugas menampilkan content/isi/informasi pada tampilan.**

**CSS: bertugas untuk memberikan presentasi/tampilan/design pada UI.**

**JS: bertugas untuk memberikan behaviour/aksi dan reaksi pada setiap logic yang berjalan pada client-side.**

Trivia:

Code dari Client-Side selalu bisa dilihat dari web-browser sehingga sifatnya terbuka dan tidak secure. Tetapi Client-Side bersifat offline setelah mereka selesai di download oleh web-browser, sehingga performancenya tidak akan dihambat oleh lemah dan lambatnya jaringan.

Developer yang hanya men-develop server-side disebut back-end developer, Developer yang hanya men-develop client-side disebut front-end developer, developer yang men-develop client dan server side disebut full-stack developer.

### 3. Non-programming language

Seluruh bahasa yang digunakan oleh client-side adalah non-programming language. Untuk HTML disebut juga dengan markup language, dan untuk javascript disebut juga sebagai scripting language.

Scripting language berbeda dengan programming language, contoh dari scripting language adalah Javascript, dan programming language adalah Java dan C#. Javascript tidak memerlukan process Compile atau Build untuk dijalankan. Karena Compile atau Build adalah satu proses dimana programming language diterjemahkan kedalam bahasa mesin yang komputer mengerti, sedangkan scripting language akan langsung di interpretasi/dibaca langsung oleh komputer. Bahasa mesin dari C# dan C++ adalah CLR (Common Language Runtime) dan bahasa mesin dari Java adalah JRE (Java Runtime Environment).

Markup language adalah bahasa yang membawa informasi dalam "markup tags", contohnya seperti ini <></>. Sehingga seluruh informasi harus berada di dalam markup tags tersebut.

Ada 2 Markup Language yang sering kamu dengar HTML dan XML:

1. HTML (Hypertext Markup Language): Karena HTTP adalah protocol yang memang didesign untuk mengirim informasi dalam hypertext, maka HTML menerima informasi dalam bentuk hypertext dan menamilkannya ke user dalam Markup. Seperti yang sudah pernah diceritakan kalau HTML bekerja pada Client-Side.
2. XML (eXtensible Markup Language): Solusi memanfaatkan sistem bahasa markup untuk mengirim data pada server-side dari satu system ke system lainnya. XML akan kita bahas lebih dalam pada saat belajar Javascript.

Kesimpulan:

Pada saat kalian belajar menjadi seorang Front-end developer, kalian tidak akan pernah menemui compile error. Segala kesalahan akan bisa dilihat langsung hasilnya pada saat di jalankan.