**DAY 1 : Pengenalan Web**

**1. Sejarah Website**

Penemu website adalah [Sir Timothy John "Tim" Berners-Lee](https://id.wikipedia.org/wiki/Tim_Berners-Lee), sedangkan situs web yang tersambung dengan jaringan pertama kali muncul pada tahun 1991. Web awalnya dibuat dan dikembangkan untuk memenuhi permintaan dalam berbagi informasi secara otomatis antar ilmuwan di universitas dan lembaga di seluruh dunia. Sebelum ada Website, banyak hal yang dilakukan dalam internet, tetapi tidak ada yang dapat dilakukan dengan mudah. Solusi Berners-Lee adalah teknologi hypertext untuk membentuk sebuah dokumen Website. Pada tanggal 30 April 1993, [CERN](https://id.wikipedia.org/wiki/CERN) (tempat di mana Tim bekerja) mengumumkan bahwa [WWW](https://id.wikipedia.org/wiki/WWW) dapat digunakan secara gratis oleh publik.

**2 Pengenalan Website**

Website adalah sebuah halaman yang menampilkan informasi melalui teks atau gambar. Website dapat diakses melalui internet menggunakan browser. Website ini unik karena tiap halaman dapat saling terhubung dengan menggunakan hypertext links. Umumnya sebuah website memuat informasi yang disediakan secara perorangan, kelompok, atau organisasi. Semua informasi yang tersedia pada halaman website disimpan pada server.

Website sudah berkembang sangat pesat. Sekarang, website digunakan tidak hanya untuk media berbagi informasi atau berita. Layaknya sebuah aplikasi pada komputer dan *handphone* kita, website dapat digunakan untuk komunikasi secara *real-time,* mendengarkan sebuah lagu, bahkan *photo editing* sekarang dapat dilakukan pada sebuah website. Seluruh kemampuan web terkini dapat kita lihat melalui situs <https://whatwebcando.today/>.

**3 Perkembangan Website**

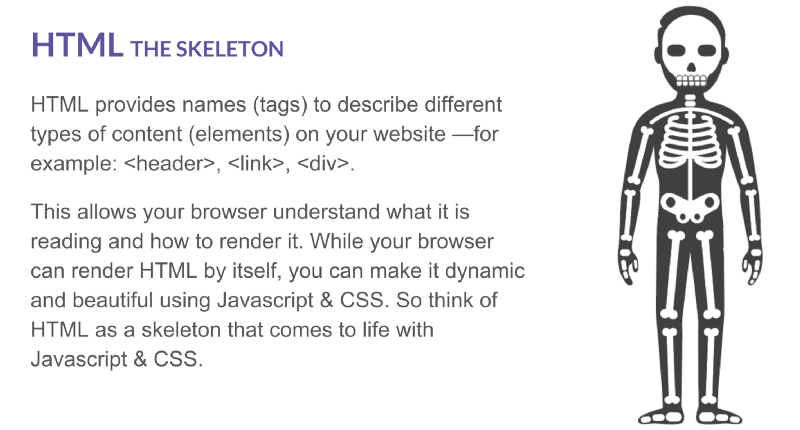
Perkembangan Website yang pertama adalah Standar Web 1.0. Web 1.0 merupakan bentuk website yang paling awal. Hal yang disajikan dalam website ini masih bersifat statis dan cenderung hanya bersifat informatif. Layanan yang internet kala itu masih berkisar antara static website yang saling dihubungkan dengan hyperlink. Umumnya website berformat “brosur online” (website yang menyampaikan informasi satu arah) umumnya berbentuk profile, portal berita, toko online, layanan email, dll. Website kala itu dihuni oleh website-website yang di desain menggunakan table dan flash.

Kemudian Era pengembangan website kedua (Web 2.0) di mana pengunjung mulai dapat melakukan interaksi dengan diatur oleh sistem yang ada pada website. Web 2.0 sendiri merupakan sebuah istilah yang pertama kali dicetuskan pada tahun 2003 oleh O’Reilly Media, dan dipopulerkan pada konferensi web 2.0 pertama di tahun 2004. Dapat disimpulkan Dalam standar web 2.0, website sudah merupakan ajang interaksi antar sesama pengguna. Bentuk yang menjadi khas pada generasi ini adalah website bukannya hanya merupakan sumber bacaan dan mencari informasi namun juga sebagai bagian dari interaksi sosial.

Konsep Web 3.0 pertama kali diperkenalkan pada tahun 2001, saat Tim Berners-Lee, penemu World Wide Web, menulis sebuah artikel ilmiah yang menggambarkan Web 3.0 sebagai sebuah sarana bagi mesin untuk membaca halaman-halaman Website. Hal ini berarti bahwa mesin akan memiliki kemampuan membaca Website sama seperti yang manusia dapat lakukan sekarang ini. Web 3.0 berhubungan dengan konsep Website Semantik, yang memungkinkan isi website dinikmati tidak hanya dalam bahasa asli pengguna, tapi juga dalam bentuk format yang bisa diakses oleh agen-agen software. Beberapa ahli bahkan menamai Website 3.0 sebagai Website Semantik itu sendiri. Keunikan dari Web 3.0 adalah konsep dimana manusia dapat berkomunikasi dengan mesin pencari. Kita bisa meminta Website untuk mencari suatu data spesifik tanpa bersusah-susah mencari satu per satu dalam situs-situs Website. Web 3.0 juga mampu menyediakan keterangan keterangan yang relevan.

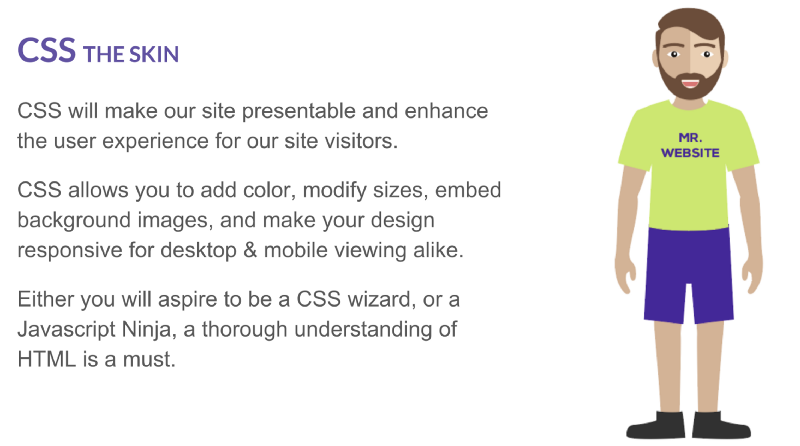
**4 Anatomi Website**

Terdapat 3 (tiga) pondasi penting dalam membuat website. Pertama adalah *Hypertext Markup Language (HTML),* salah satu [*markup language*](https://en.wikipedia.org/wiki/Markup_language) yang digunakan untuk membuat struktur dan menampilkan konten pada *World Wide Web* (Website). Bila dianalogikan sebagai tubuh manusia, HTML merupakan tulang-tulang yang menopang tubuh manusia.



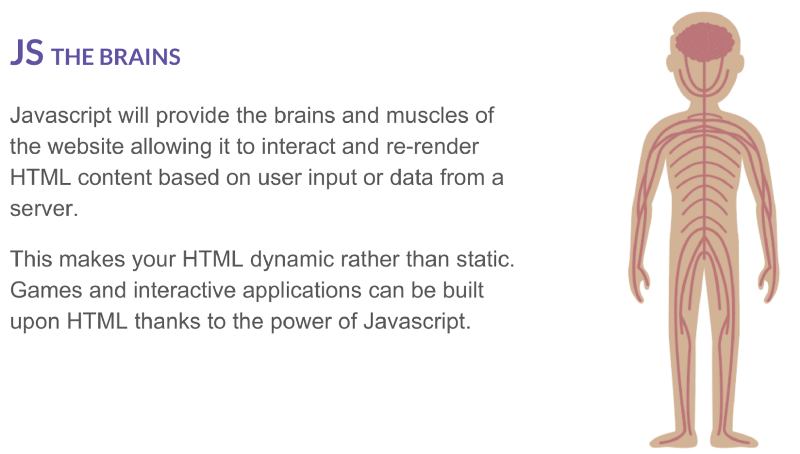
*Gambar 1. Analogi HTML pada Manusia*

Selanjutnya adalah *Cascading Style Sheets (*CSS)yang berfungsi untuk memberikan styling sehingga website dapat memiliki tampilan yang menarik. Bila dianalogikan kembali pada tubuh manusia, CSS memberikan tampilan sehingga manusia satu dengan yang lainnya dapat dikenali.

**

*Gambar 2. Analogi CSS pada Manusia*

Pengetahuan akan HTML dan Javascript sebenarnya sudah dapat menjadi modal dalam membuat sebuah situs web yang baik. Namun tidak lengkap rasanya apabila situs web tersebut tidak memiliki kemampuan yang dinamis dan interaktif. Oleh sebab itu hadirlah *JavaScript (JS)*  yang memberikan kemampuan dinamis dan interaktif pada website. Pada analogi manusia, JS merupakan otak dan otot dari website yang mampu berinteraksi dengan dan menampilkan konten website berdasarkan input dari penggunanya.



*Gambar 3. Analogi JS pada Manusia*

**5. Unsur-Unsur Website**

Berikut adalah unsur-unsur dari website :

**a. Nama Domain**

Alamat unik di dunia maya (internet) yang berguna untuk menemukan sebuah website. Umumnya URL ini di perjualbelikan dengan sistem sewa tahunan. Dan biasanya di belakang URL ini mempunyai akhiran sesuai dengan lokasi dan kepentingan atas di buatnya website tersebut.

**b. Hosting**

Ruangan yang terdapat dalam harddisk sebagai tempat penyimpanan data, file, video, email, database, dan lain-lain yang nantinya akan ditampilkan di dalam website tersebut.

**c. Scripting Language**

Menerjemahkan setiap perintah pada saat website tersebut sedang dijalankan. Contoh dari scripting language, yakni Javascript, XML, JSP, HTML, PHP, dan lain-lain

**d. Desain Website**

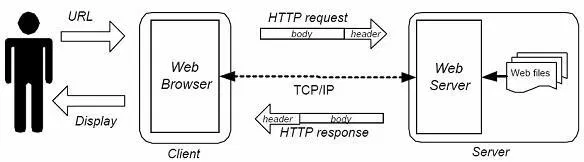
Desain website merupakan hal yang penting. Faktor kenyamanan pengunjung harus diterapkan jika membuat website. Buatlah website yang menarik agar pengunjung mudah dalam penggunaannya sehingga akan terus mengunjungi website tersebut. Untuk melakukannya kita dapat menggunakan CSS, SCSS, SASS, bahkan framework seperti Bootstrap, Tailwind, Materialize, dll.

**e. Program Transfer Data ke Pusat**

Data FTP (File Transfer Protocol) merupakan akses yang diberikan pada saat kita memesan website hosting, FTP berguna untuk memindahkan file-file website yang ada pada komputer ke pusat website hosting agar dapat terakses ke seluruh dunia.

**6. Cara Kerja Website**

Website dapat dibuka menggunakan web browser seperti Firefox, Edge atau Chrome. Kita cukup mengetikkan URL di kolom browser untuk dapat membuka website. Sebagai contoh, silakan ketikkan URL https://www.axarschool.com di kolom browser maka akan tampil halaman landing page dari website Axar School. Bagaimana hal ini dapat terjadi? Mengapa kita bisa memperoleh hasil yang sesuai dan bukan website lain? Mari kita perhatikan gambar alur di bawah ini.



*Gambar 4. Ilustrasi Cara Kerja Website*

Berikut adalah penjelasan tahapan-tahapan pada alur tersebut :

1. User/pengguna yang akan mengakses suatu website berupa URL melalui *web browser*
2. Website browser tersebut mengirimkan permintaan (request) berupa HTTP request kepada web server melalui layer-layer TCP/IP
3. Website Server memberikan website files yang di-request jika ada
4. Website files yang telah diberikan tadi tidak langsung ditampilkan di-display begitu saja, namun website Server memberikan respon kembali ke website browser melalui HTTP Response yang juga melalui layer-layer TCP/IP
5. Response diterima oleh *web browser*, dan kemudian dikirimkan kepada user berupa display

Untuk menampilkan informasi pada jendela, browser menggunakan HTML dan CSS yang dikirimkan dari server. Dengan begitu, informasi dalam bentuk HTML dan CSS-lah yang dikirimkan server untuk client (browser). Beberapa halaman website juga membutuhkan informasi ekstra seperti berkas gambar, suara atau video, tetapi berkas tersebut sebenarnya hanya ditanamkan (*embed*) pada HTML. Dan beberapa halaman website sebenarnya juga membutuhkan JavaScript untuk me-*render* HTML atau CSS agar dapat menampilkan informasi secara dinamis.

**7. Jenis-Jenis Website**

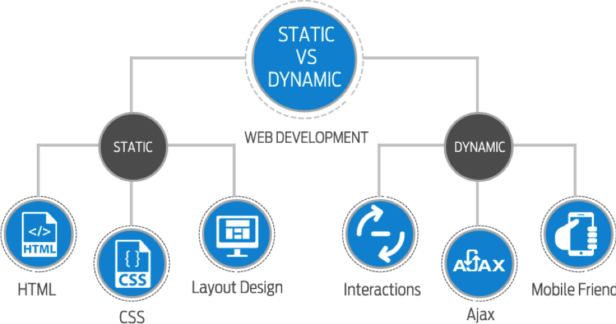
Secara garis besar, website dapat dibagi menjadi 2 berdasarkan jenisnya, yaitu :

**1. Website Dinamis**

Website yang halamannya selalu update, biasanya terdapat halaman backend yang digunakan untuk menambah atau mengubah konten. Website dinamis membutuhkan database untuk menyimpan. Website dinamis mempunyai arus informasi dua arah, yakni berasal dari pengguna dan pemilik, sehingga mengupdate dapat dilakukan oleh pengguna dan juga pemilik website.

**2. Website Statis**

Website yang halamannya tidak berubah, biasanya untuk melakukan perubahan dilakukan secara manual dengan mengubah kode. Website Statis informasinya merupakan informasi satu arah, yakni hanya berasal dari pemilik softwarenya saja, hanya bisa diupdate oleh pemiliknya saja. Contoh dari pengertian website statis ini, yaitu profil perusahaan.



*Gambar 5. Static VS Dynamic Website*

**8. Server & Client**

Seperti yang sudah disebutkan sebelumnya, informasi pada sebuah website disimpan pada *server*. Lantas apa itu *server*? Banyak yang mengira bahwa *server* adalah sebuah komputer dengan performa tinggi dan berukuran besar. Hal tersebut tidak salah karena biasanya komputer yang dijadikan *server* memang harus memiliki performa yang tinggi. Selain untuk menyimpan data (HTML, CSS, dan JS disimpan di sini), *server* juga harus melakukan pekerjaan yang cukup berat, yaitu menanggapi, menyediakan, dan mengelola data yang diminta oleh *client*. Tidak hanya satu *client*, namun bisa ratusan bahkan ribuan.

Pengertian *server*sebenarnya lebih merujuk pada sebuah *software* yang dapat menghubungkan sebuah komputer dengan komputer lain. Jadi penekanan sebuah server bukan pada sebuah komputernya, melainkan pada *rule* atau fungsi dari sebuah komputer tersebut.

Di sisi lain, perangkat yang meminta (*request*) suatu layanan tertentu ke suatu *server* disebut dengan *client*. Browser merupakan salah satu *client* yang memanfaatkan HTTP Server dalam transaksi datanya. Jelas, tujuannya adalah untuk mengolah, menampilkan, dan melakukan interaksi dengan dokumen yang disediakan oleh web server. Layaknya sebuah pelukis, browser mampu menyajikan informasi dalam bentuk visual yang indah bersumber dari data mentah yang diperoleh dari server.

**9. Pengenalan Karir Web Developer**

Web developer adalah seorang programmer yang membuat program dan aplikasi untuk world wide web. Web developer sendiri dapat dikategorikan kembali berdasarkan bagian yang ditanganinya. Apabila kita membuka lowongan situs kerja, maka kita sering mendengar istilah *frontend developer, backend developer, dan fullstack developer*. Apa maksudnya? Mari kita bahas satu-satu.

**a. Frontend Developer**

Front-end developer adalah orang yang bertanggung jawab atas tampilan depan website termasuk penyusunan serta pengelolaan website untuk membuat user interface dan user experience yang bagus bagi user website. Seorang front-end developer diharuskan untuk memahami tiga bahasa pemrograman utama yaitu HTML, CSS, dan JavaScript. Selain fasih dalam bahasa-bahasa ini, front-end developer juga harus familiar dengan framework seperti Bootstrap, Foundation, Backbone, AngularJS, dan Ember JS untuk memastikan bahwa konten akan selalu terlihat bagus di semua device atau mobile friendly.

Tidak hanya itu, front-end developer juga diharapkan tahu tentang library seperti jQuery dan LESS yang menjadikan satu kode agar bisa digunakan dengan lebih efisien. Kalau Anda melihat deskripsi pekerjaan front-end developer, banyak juga yang meminta pengalaman dengan Ajax, sebuah teknik yang banyak digunakan untuk JavaScript agar halaman website dimuat secara dinamis dengan mendownload data server di background. Front-end developers biasa bekerja sama dengan designer untuk membuat mockups dari development (pengembangan) sampai website itu jadi. Front-end developer yang ahli juga bisa mengidentifikasi masalah apa yang ada pada user experience dan memberikan rekomendasi selama proses desain agar mereka tidak memiliki masalah yang sama.

Tidak hanya itu, front-end developer juga perlu bekerja sama dengan tim lain untuk lebih mengerti tentang tujuan, kebutuhan, dan kesempatan apa yang ingin didapatkan bisnis Anda dari website tersebut. Ini dilakukan agar mereka tahu bagaimana mereka harus membuat websitenya supaya tujuan dan kebutuhan Anda tercapai dan terpenuhi. Front-end developers bisa dibilang bertanggung jawab atas desain sebuah rumah yang sudah dibangun oleh back-end developer. Jadi kebanyakan front-end developers adalah orang yang mungkin lebih teknis tetapi memiliki kreativitas yang tinggi.

**b. Backend Developer**

Back-end sebuah website terdiri dari sebuah server, aplikasi, dan database. Developer backend bertanggung jawab untuk membangun dan mengelola teknologi yang mendukung ketiga komponen tersebut. Ini dilakukan agar front-end sebuah website bisa dibuat. Untuk memastikan komunikasi antara server, aplikasi, dan database berjalan dengan lancar, back-end developer biasa menggunakan bahasa programming server-side seperti PHP, Ruby, Python, Java, dan .Net untuk membangun aplikasi dan tools seperti MySQL, Oracle, dan server SQL. Ini dibutuhkan untuk mencari, menyimpan, atau mengubah data dan menyajikannya ke coding front-end.

Biasanya job description untuk web developers yang berfokus pada back-end developers harus familiar dengan framework PHP seperti Zend, Symfony, dan CakePHP, pengalaman dengan version control software seperti SVN, CVS, atau Git, dan pengalaman dengan Linux sebagai sistem pengembangan dan deployment. Back-end developer biasa menggunakan tools ini untuk membuat atau berkontribusi ke aplikasi web dengan kode yang bersih, portable, dan terdokumentasi dengan baik.Tetapi sebelum Anda menulis kode tersebut, back-end developer perlu berkolaborasi dengan tim lain untuk mengerti apa yang ingin dicapai dengan website ini.

**c. Fullstack Developer**

Peran ini pertama kali diperkenalkan oleh *Facebook Engineering Department* tahun 2010 yang lalu. Full-stack developer adalah developer yang bisa bekerja secara cross-functional, dalam hal ini front-end dan back-end. Full stack developer kebanyakan bekerja seperti back-end developer yaitu di bagian server website. Tetapi mereka fasih dalam bahasa front-end yang mengatur tampilan sebuah website. Biasanya, mereka mengerti semua bahasa programming dari HTML sampai Python.

Meskipun tergantung dari proyeknya, full stack developer harus memiliki pemahaman yang dalam tentang cara kerja website. Misalnya mengatur dan konfigurasi server Linux, menulis API server-side, menggunakan JavaScript untuk keperluan front-end, dan menggunakan CSS untuk mendesain website. Dengan menggunakan berbagai bahasa pemograman ini, full stack developer harus bisa tahu solusi baik untuk front-end maupun back-end. Pada dasarnya, seorang full stack developer akan bertanggung jawab atas seluruh experience pengguna website, mulai dari tampilan website sampai dengan waktu loading website