



Pengantar Fullstack Web Developer

Silver - Chapter 1 - Topic 1

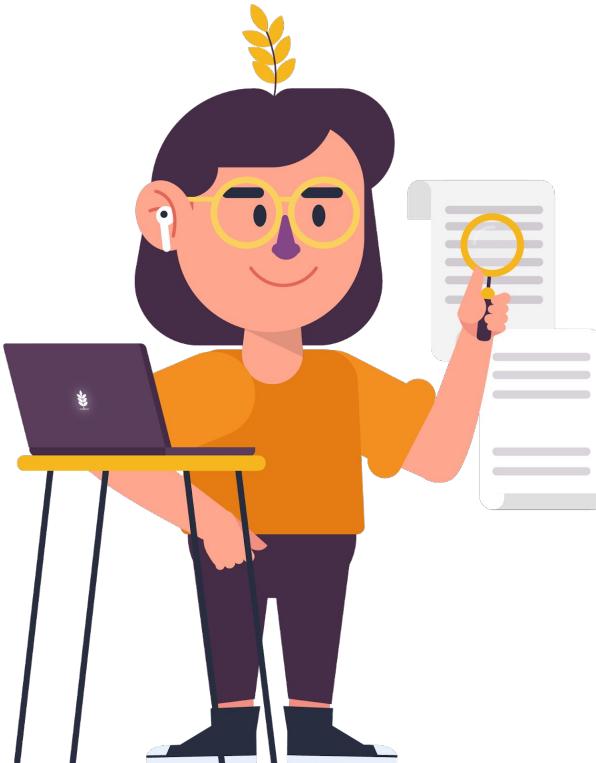
**Selamat datang di Chapter 1 Topik 1 online
course Fullstack Web dari Binar Academy!**





Sebagai pembuka course, **Chapter 1 ini bakalan ngajak kamu buat menguasai kemampuan dasar untuk membuat halaman web.** Mulai dari penjelasan terkait role fullstack web developer, penjelasan HTML, CSS sampai penggunaannya dengan menggunakan CSS framework.

Nah, pada topik pertama, kita bakal cari tahu **siapa itu fullstack web developer? Terus apa saja tugas dan skill yang dibutuhkan, hingga pengenalan tools yang biasa digunakan mereka.** Cus langsung aja~





Detailnya, kita bakal bahas hal-hal berikut ini :

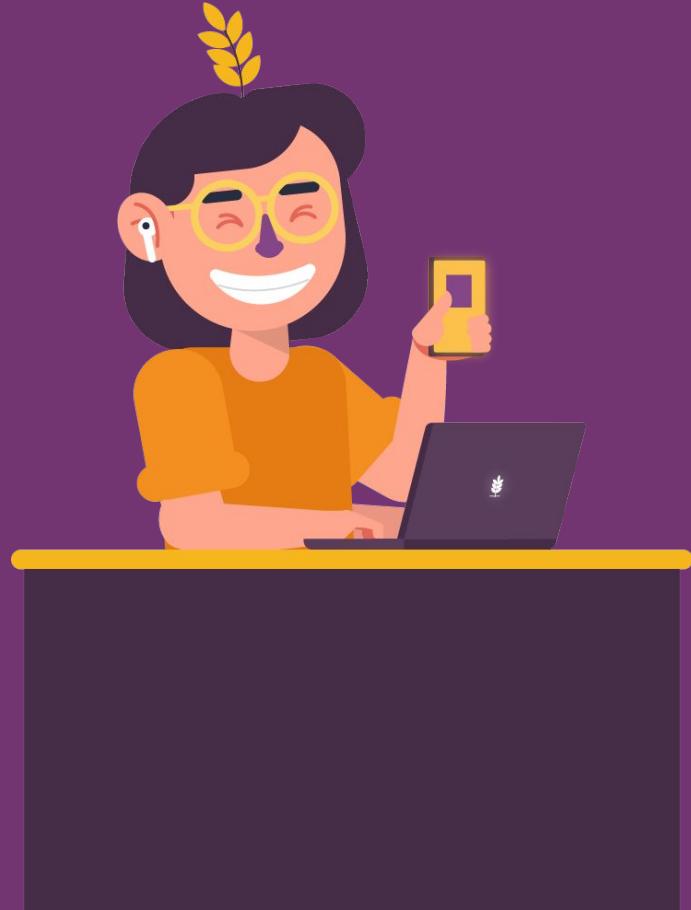
- Fullstack web developer introduction
- Tugas fullstack web developer
- Skill set fullstack web developer
- Tech Stack dan Tools yang digunakan



Gengs.. coba kamu inget-inget aktivitas terakhir yang kamu lakuin sebelum belajar topik ini.

Ngobrol sama temen lewat Whatsapp? Kuliah pakai Zoom? Atau bahkan nonton anime di Netflix?

Naahh bisa dibilang kegiatan-kegiatan tadi itu tersedia dalam situs website yang bisa kita akses dimanapun, kapanpun oleh siapapun.





Kita cukup sering ya mendengar kata website, tapi apa sih website itu?





Situs web (website)

Merupakan tempat menampung informasi yang bisa diakses menggunakan web browser (penjelajah web).

Kita bisa membuka website di mana pun, selama situs web tersebut dibuat dalam domain publik. Contoh web browser itu adalah Mozilla Firefox, Safari, Google Chrome, Internet Explorer, dll.



Ini adalah salah satu contoh dari sebuah situs web (website) binaracademy.com



Informasi yang akan diterima oleh user situs web berupa halaman (Page). Kita coba lihat deh gambar disamping~

Kalau kamu buka web **binaracademy.com**, kamu akan dapet informasi caranya belajar di dalam Binar Academy, atau cara untuk menjadi partner di Binar Academy.



Ini adalah salah satu contoh dari sebuah situs web (website) binaracademy.com



Eitssts, tapi situs web nggak cuma menampung informasi dan menampilkannya ke user aja lho. Situs web juga bisa digunakan untuk **mengolah informasi** yang diberikan user, misalnya :

- Membayar tagihan listrik
- Membuat akun
- Mengunggah foto KTP untuk melakukan verifikasi data
- dan sebagainya

Pokoknya situs web yang kamu akses saat ini udah lengkap banget deh, bisa melakukan banyak aktivitas~





Usut punya usut, situs web ini punya dua ragam. Ada situs **web statis (static website)** dan **situs web dinamis (dynamic website)**.

bedanya apa? Kalau kata Om Ebiet G Ade, kita tanya dulu ke rumput yang bergoyang~



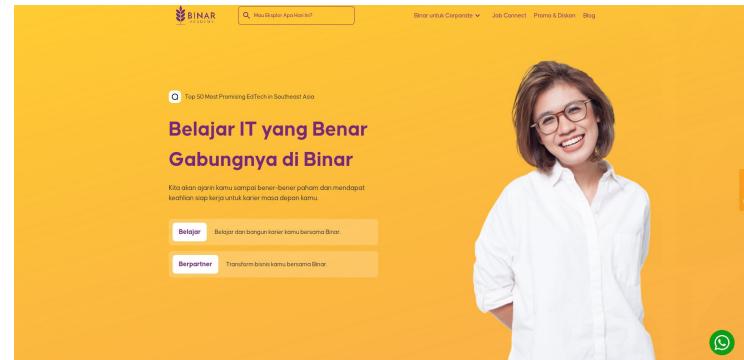


Web Statis

Kata rumput, **situs web statis** adalah web yang menampilkan page pada sebuah server dengan **format sama** kemudian mengirimnya ke web browser user.

Situs web statis ini biasanya dibuat dengan menggunakan HTML, CSS dan Javascript aja.

Nah, karena situs web ini bersifat statis, artinya **siapapun yang mengakses situs tersebut akan melihat informasi yang sama**.



Contoh situs web statis



Kita pake contoh deh. Misalnya kamu punya temen namanya Jungkook. Kalian berdua buka website **binaracademy.com** pada waktu yang sama.

Tampilan page yang muncul di kamu bakal **sama** kayak yang muncul di layarnya Jungkook. Walaupun kalian mengakses website itu pada device yang berbeda.

Nah, kapan lagi coba kamu bisa kembaran gitu sama jungkook~



Contoh situs web statis



Analogi lebih spesifiknya gini.

Ditengah waktu kamu belajar, kamu pingin beli tiga permen milkita. Kamu pergi ke Indomarch. Aktivitas yang kamu lakukan adalah dateng, ambil permen, bayar dan pulang.

Indomarch adalah situs web statis sedangkan permen milkita adalah **halaman** dari situs web statis itu tadi.

Kamu nggak perlu nungguin si Permen milkita ini diproses dulu di pabrik. Kamu cuma nerima jadi aja.





Situs web statis cenderung lebih mudah untuk dikembangkan karena **sudah banyak tools yang tersedia** untuk membantu pengembangan situs web nya.

Selain itu, informasi yang ditampilkan di dalam situs web statis biasanya **bukanlah sebuah informasi yang kompleks**. Sehingga lebih mudah untuk diimplementasikan.

Contoh-contoh tools yang dapat digunakan untuk membuat situs web statis adalah **Wordpress, Webflow, Squarespace** dan sebagainya.



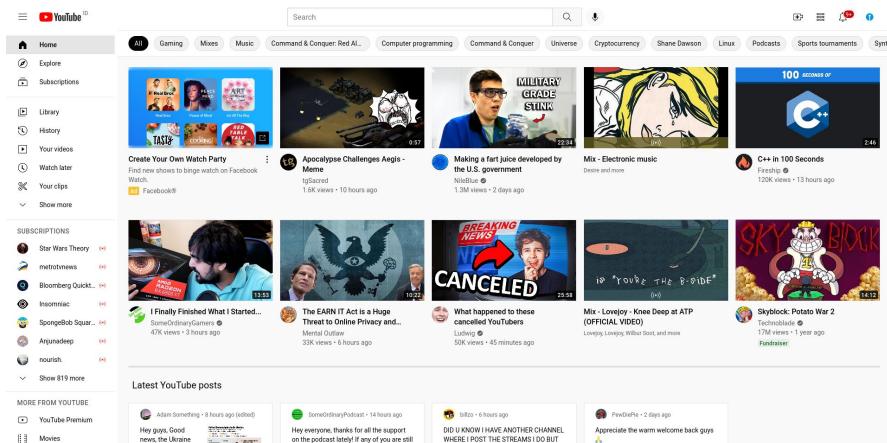


Web Dinamis

Kalau tadi kita bahas Situs web statis, sekarang kita menuju ke Situs web dinamis.

Rumput oh rumput apa sih web dinamis itu? Kamu bertanya, rumput menjawab.

Web dinamis merupakan situs web yang halaman-halamannya akan **dibuat dan diproses** tampilannya **ketika user sedang mengakses** situs tersebut.

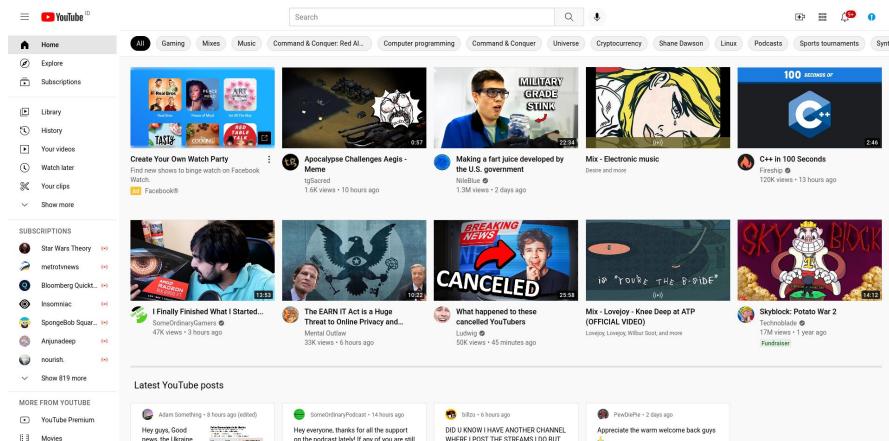




Situs web dinamis ini **akan melakukan render dari tiap halaman** on-demand berdasarkan permintaan user. Contohnya adalah web Youtube.

Ketika kamu buka halaman utama dari Youtube, web ini akan membuat halaman indeks yang disajikan khusus mengikuti tontonan kesukaan kamu.

Maka dari itu biasanya **homepage yang dilihat user A akan berbeda dengan user B** meskipun halaman yang dibuka sama.





Analaoginya gini, situs web dinamis ini mirip kayak nasi goreng gerobakan.

Kamu cari mba nasi gorengnya, terus kamu pesen tuh nasi goreng. Apa nasi goreng yang kamu pesen langsung jadi? Tentu enggak dong, perlu digoreng dulu.

Ternyata kamu pingin nasinya pedes banget, kamu request supaya abangnya masaknya pake cinta. Pas udah mateng baru deh kamu bawa pulang nasi goreng pedes karet dua sesuai permintaan kamu.

Abang masak
pake cinta loh, neng!



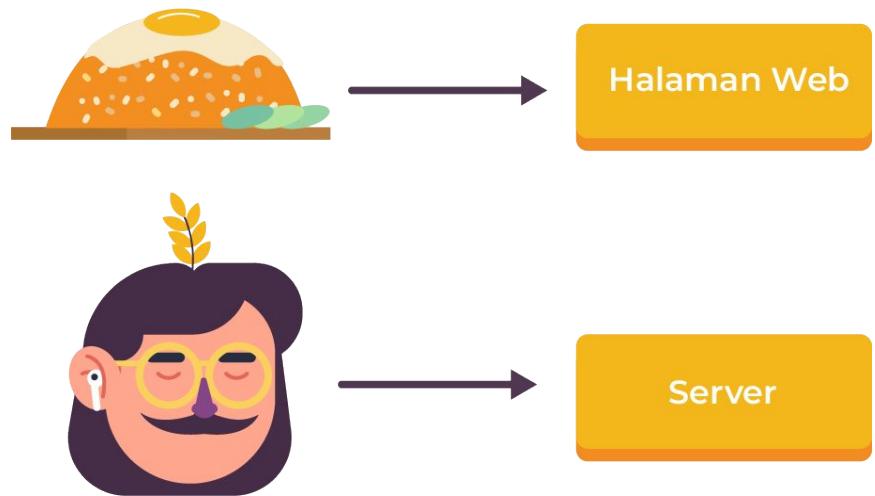
Kalo kalo kamu suka pedes, akan pesen nasi goreng pedes. Tapi kalau Jungkook suka asin ya dia akan pesen asin.

Permintaan ini adalah proses request ke server

Artinya, nanti si mba nasi goreng akan masak nasi goreng dengan perpaduan yang beda untuk setiap pelanggannya.

Mba nasi goreng ini bisa dikatakan sebagai server di web dinamis

Nasi goreng yang dibawa pulang sesuai permintaan itu adalah halaman server kamu. Mirip kaya homepage youtube kamu kan~

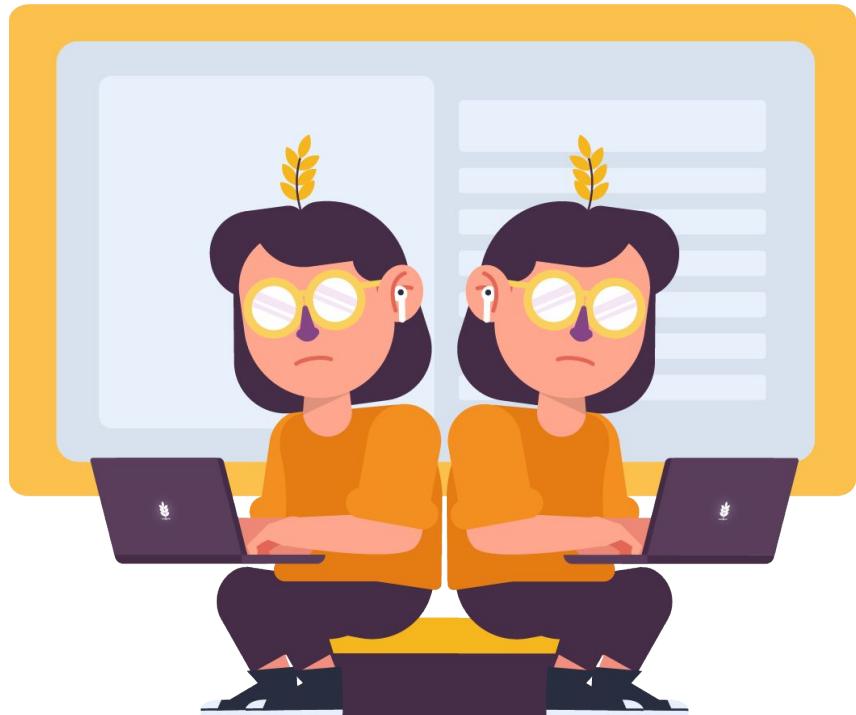




Setelah analogi itu, kita bisa dapetin bahwa ada 2 fungsi yang krusial dalam web dinamis, yaitu :

- peran server dalam melayani permintaan di situs web dinamis, dan
- peran untuk menampilkan informasi yang diminta.

maka dari itu biasanya situs web dinamis ini dipecah menjadi dua bagian, yaitu **client side (Frontend)**, dan **server side (Backend)**.





Karena ada dua bagian dari situs web dinamis, maka dari itu pengembangan situs web dinamis ini relatif lebih sulit dibandingkan dengan situs web statis.

Situs web dinamis ini biasanya digunakan untuk mengolah informasi yang sudah kompleks seperti manajemen keuangan, debit kredit, preferensi user dalam melihat video youtube, dan sebagainya.

Yang mana informasi tersebut perlu diolah terlebih dahulu sebelum disajikan ke user.





Nah setelah kita bahas keduanya~ jadi inilah perbedaan untuk web statis dan web dinamis.

	Web statis	Web dinamis
Interaksi antara pengguna dan pemilik web	Tidak memungkinkan terjadi interaksi	Terjadi interaksi melalui komentar, atau yang lainnya
Bahasa pemrograman	HTML, CSS dan Javascript	HTML, CSS, Javascript + Bahasa Pemrograman Native (PHP, Perl, NodeJS)
Penggunaan database	Tidak menggunakan database	Menggunakan database seperti MySQL
Konten	Jarang diupdate	Lebih sering diupdate



Kamu pasti pernah nonton Squidward nulis pesenan Krabby Patty di Krusty Krab kan?

Kalau kamu penasaran sama frontend tadi, Krusty Krab adalah frontend. Alias **tampilan** yang bisa diliat user.



Ketika kamu membuka sebuah situs web, **semua hal yang dapat berinteraksi dengan kamu itulah yang dinamakan frontend.**

Analoginya kayak gini, kamu sedang berkunjung ke sebuah restoran bernama Mahdi.

Nah di dalam restoran ini kan ada tempat makan, ada kasir, ada wastafel dan sebagainya, yang mana tiap benda-benda itu memiliki fungsinya masing-masing.





Karena kamu berkunjung ke Mahdi untuk makan, otomatis kamu akan langsung pergi ke kasirnya buat pesen makanan.

Kamu kepingin beli Mahdi paket BTS Meal, dan abis itu pelayan Mahdi bakal ngasih kamu nampan yang diatasnya ada minumanmu, nugget, dan kentangmu.





Kamu bawa nampan ke meja yang kamu duduki dan makan disitu.

Sebelum kamu makan pastinya kamu bakal cuci tangan dong, maka kamu pergi ke wastafel dan mencuci tanganmu disitu. Kemudian kembali ke meja untuk makan BTS Meal kamu.

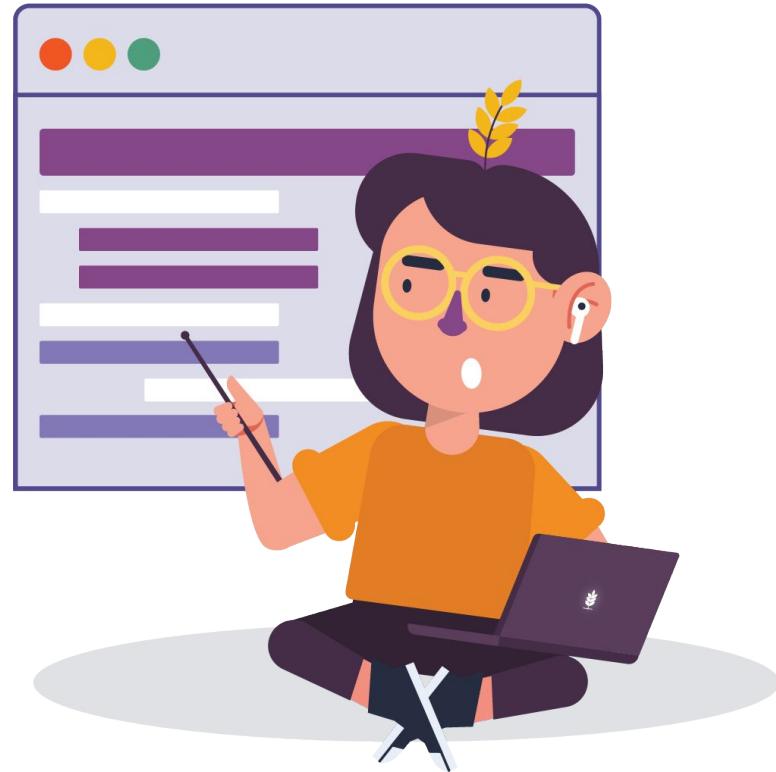




Dari analogi tadi, mulai dari meja, wastafel, dan kasir, itu semua merupakan user interface. Dimana user interface inilah yang juga kita sebut sebagai **frontend**.

Kalau kita kembali lagi ke konsep situs web, apa yang kamu lihat melalui web browser itulah yang disebut sebagai frontend.

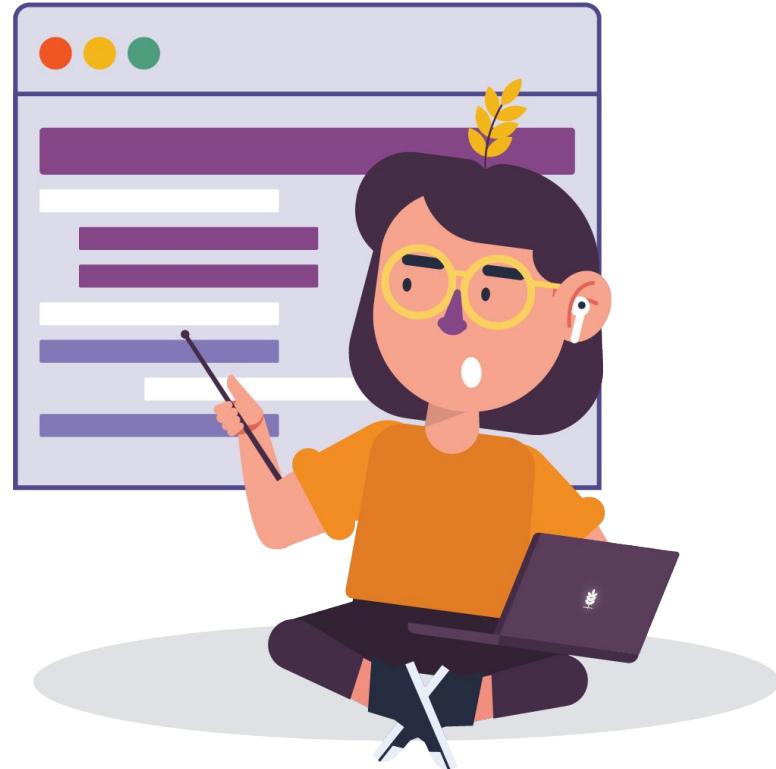
Lalu, BTS Meal dan air yang dipakai untuk mencuci tanganmu itu semua adalah data atau informasi yang **disajikan** melalui frontend.





Secara garis besar, **frontend adalah tampilan yang kamu lihat di dalam peramban web beserta interaksi-interaksi yang dapat kamu lakukan dalam tampilan tersebut.**

Contoh hal yang dapat berinteraksi dengan user di dalam situs web adalah tombol yang bisa ditekan, formulir yang dapat diisi, dan video yang bisa di-pause dan sebagainya.

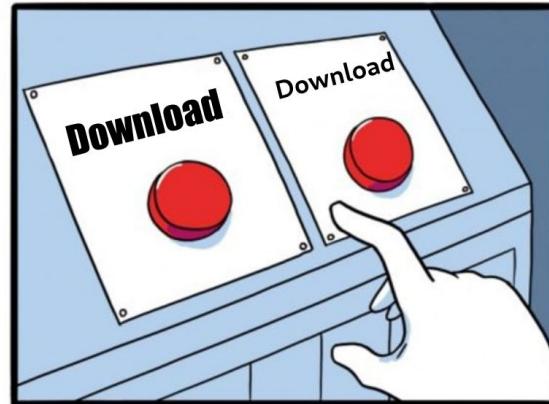




Sebenarnya istilah frontend ini sangat luas, tapi bisa kita sederhanakan dengan istilah **client**.

Client adalah semua aplikasi atau semua hal yang dilihat oleh pengguna situs kita (frontend), baik itu aplikasi desktop, aplikasi mobile, situs web dan sebagainya.

Aktor yang **bertanggung jawab** terhadap pembuatan **frontend** dari sebuah situs web adalah **frontend web developer**.





Balik lagi nih ke analogi restoran tadi, kamu mungkin bertanya-tanya :

- Lha terus yang masakin BTS Meal tadi siapa?

Terus wastafelnya bisa muncul air, airnya tuh dari mana?

Bagian yang mengurus itu semua disebut sebagai **backend**.





Apa itu Backend?

Kalau frontend itu yang ada di bagian depan, kalau backend adalah yang mempersiapkan data dibelakangnya.

Alias spongebob yang masakin Krabby Patty.





Backend adalah bagian dari situs web yang berfungsi untuk **mengolah informasi yang datang dan menyajikan informasi** tersebut ke frontend untuk diolah lagi agar menjadi sebuah tampilan.

Backend inilah yang bertugas untuk membuat informasi dan menentukan informasi mana yang cocok buat ditampilkan ke user A dan mana yang nggak cocok buat user B.



Backend juga bertanggung jawab dalam mengurus segala kepentingan bisnis yang terjadi di dalam sebuah situs web, seperti :

- Menghubungi bank untuk menagih tagihan dari user A.
- Memberikan notifikasi ke user A supaya segera membayar tagihannya.
- Memberi tahu user bahwa email yang ingin mereka daftarkan sudah terdaftar.
- Dan sebagainya.





Backend ini sangat abstrak, dimana user nggak bisa dapet hasil Backend tanpa adanya Frontend.

Coba pake analogi dulu~ kembali ke Mahdi, pas kamu pesen makanan, Kasir bakal catet pesanan kamu dan kasih ke bagian dapur.

Catatan tersebut pastinya punya format sendiri yang udah ditentuin antara kasir dan bagian dapur supaya mereka nggak salah tangkep pesanan kamu.





Setelah bagian dapur menerima pesanan dari kasir, mereka mulai siapin bahan mentah buat bikin BTS Meal.

Di dapur, koki bakal menggoreng Nugget dan kentang goreng sebagai bagian dari paket BTS Meal. Sampai akhirnya nambahin dua macam saus yang bakal kamu nikmati bersama paketmu.



Setelah semua itu selesai dilakukan, bagian dapur akan memberikan pesanan tadi ke kasir supaya bisa menaruhnya di nampan dan kasih ke kamu.

Kamu nggak perlu terlibat di dalam pembuatan BTS Meal, yang perlu kamu lakukan hanyalah **menunggu sampai pesananmu siap disajikan**, dan kamu tinggal makan.





Sama juga ketika kamu pake wastafel buat cuci tangan.

Wastafel tersebut udah tersambung ke pengairan yang ada di sekitar restoran. Kamu hanya perlu memutar tuas keran supaya air keluar dan bisa dipake buat cuci tangan.

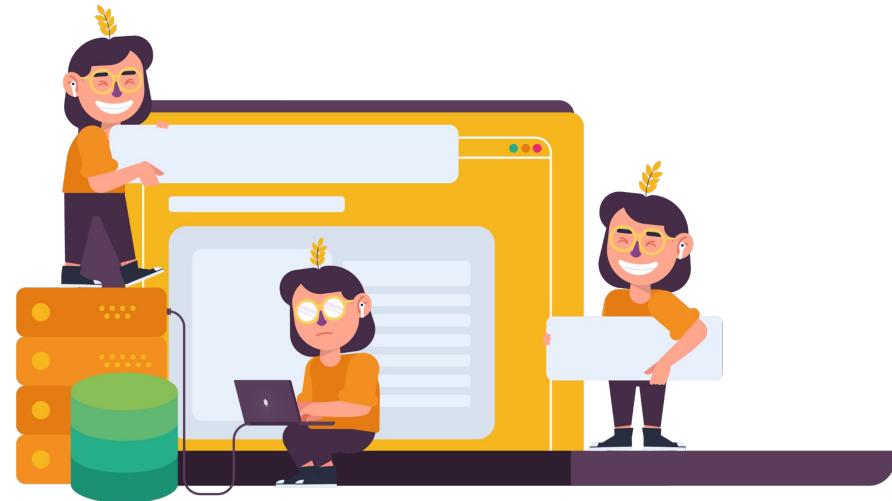
Nggak perlu ngikutin tukang pipa bangun saluran air sampai air bisa keluar di wastafel.



Berbeda dengan frontend yang bisa muncul pada smartphone user, backend dijalankan pada sebuah server yang melayani kebutuhan situs web tersebut.

Maka dari itu, user nggak akan tahu apa yang terjadi di backend. Mereka hanya perlu tahu tampilan yang ditampilkan di dalam peramban web aja.

Aktor yang bertanggung jawab terhadap **pembuatan backend** dari sebuah situs web adalah **backend web developer**.





**Kamu tahu nggak sih? backend dan
frontend merupakan cakupan
pekerjaan dari seorang Fullstack
Developer?**

**Bisa dibilang, Fullstack Developer itu
akan beragam banget tugasnya.**

**Setelah ini kita mulai masuk yaa ke
Fullstack Web Developer.**

Fullstack web developer itu kayak mba nasi goreng yang jualan nasi goreng gerobakan. Iyaa sama kaya analogi di awal tadi~

Mba-mba itu menerima pesanan dari kamu, terus dia juga yang masak pesanannya, dia juga yang menyajikan pesanannya dan dia juga yang nganter pesanannya ke kamu.



Si mba nasi goreng juga yang nyiapin tempat duduk buat kamu yang makan di tempat.

Mba nasi goreng juga yang nyuci piring setelah kamu selesai makan. Pokoknya semuanya yang terkait dagangan dia, dia yang ngurusin.

Nah kurang lebih fullstack web developer itu kayak gitu. Banyak hal yang perlu dilakukan dan disiapin supaya permintaan user terpenuhi.





Tugas Fullstack Web Developer di bagian Backend.

Terus apa sih tugas fullstack developer? Kita akan bagi tugasnya menjadi dua kategori, Backend dan Frontend.

Pertama, tugas Fullstack developer bagian **Backend** akan seputar :

- Merancang dan membuat database
- Membuat web API
- Merancang logika bisnis





Tugas Fullstack Web Developer di bagian Frontend.

Selanjutnya, tugas Fullstack developer bagian **Frontend**, akan seperti :

- Slicing (Merubah desain halaman product menjadi halaman web yang sebenarnya dengan menggunakan HTML, CSS, dan Javascript)
- Mengolah data/ informasi dari backend agar menjadi sebuah tampilan.
- Mengolah input dari user untuk dikirimkan ke backend agar dapat diolah oleh backend.

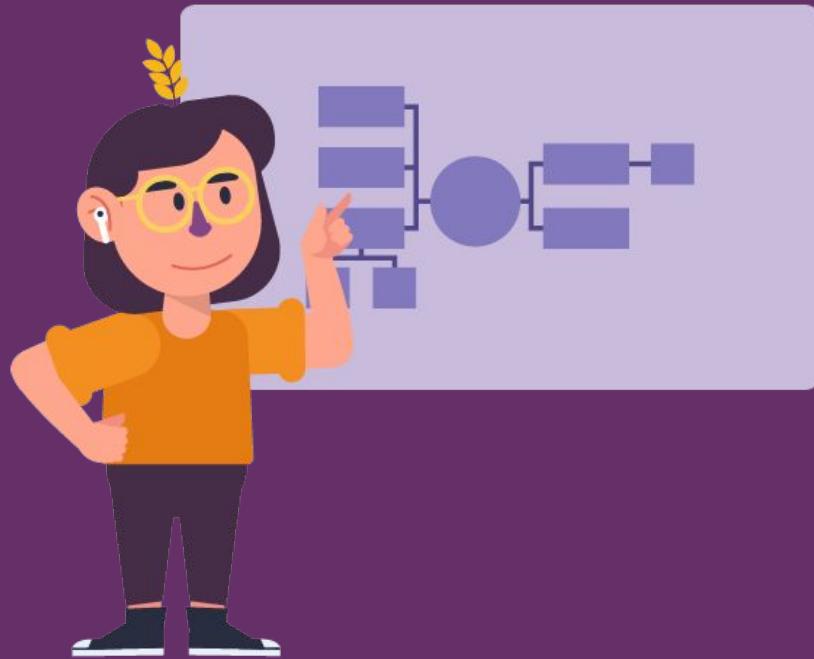




Tadi kita udah ngomongin tugas fullstack web developer, sekarang waktunya skill set~

Kita akan bedakan skill set Fullstack web developer menjadi 3 bagian :

- **Backend**
- **Frontend**
- **General**





Skill Set untuk bagian Frontend

Untuk membuat tampilan interface pada website maka skill yang dibutuhkan adalah :

- Menguasai penggunaan HTML.
- Menguasai penggunaan CSS.
- Paham akan bahasa pemrograman Javascript
- Paham akan salah satu framework yang digunakan untuk membangun frontend, sebagai contoh: ReactJS, AngularJS, dan VueJS.





Kemudian untuk bisa mengintegrasikan database pada backend dengan tampilan yang dilihat oleh user, maka perlu untuk memiliki **Pengetahuan protokol HTTP** bekerja agar dapat berkomunikasi dengan backend



Skill Set untuk bagian Backend

Untuk membuat tampilan interface pada website maka skill yang dibutuhkan adalah

- Menguasai salah satu bahasa pemrograman native yang dapat berjalan di dalam komputer, sebagai contoh: Node.js
- Menguasai salah satu framework untuk membuat aplikasi backend, sebagai contoh: Express.js (cont.)





- Menguasai penggunaan sistem pengelolaan database (DBMS), sebagai contoh: Postgresql
- Mengerti bagaimana protokol HTTP bekerja agar dapat merespon dengan baik apa yang dikomunikasikan oleh aplikasi frontend.





Skill Set umum Fullstack Web

Dalam mengembangkan sebuah situs web, diperlukan **SOP** yang akan membantu kita lebih tertata dalam melakukan tugas-tugas kita dan memastikan proses delivery nya tepat.





Nah, SOP di dalam pengembangan situs web ini biasa disebut sebagai **Software Development Lifecycle**.

Jadi sebagai developer secara umum, kita perlu memahami salah satu jenis framework software development lifecycle. Satu contoh yang umum digunakan saat ini adalah **SCRUM**.





**Sebagai Fullstack web Developer
tentu kita bakal dibantu sama
Tools-nya dong.**

**Abis ini kita akan jelasin Tools
sekaligus Techstack yang umum
dipakai saat ini.**

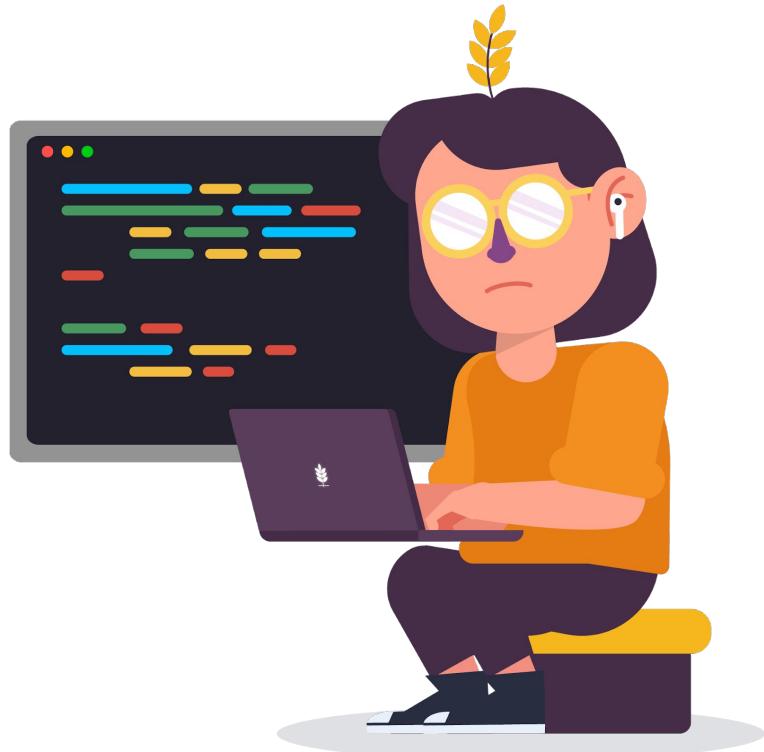




Apa itu Techstack?

Merupakan kumpulan teknologi yang bisa mempermudah kamu dalam kategorisasi keahlian seorang fullstack web developer.

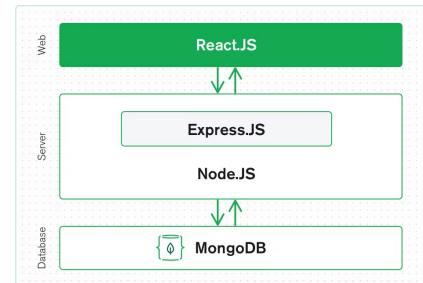
Bersamaan dengan hal tersebut, muncul istilah-istilah yang merepresentasikan seperangkat keahlian dari seorang fullstack web developer, seperti frontend dan backend.





Berikut **daftar tech stack** yang populer untuk fullstack web developer :

- LAMP (Linux, Apache, MySQL, PHP/Perl/Python)
- MERN (MongoDB, ExpressJS, ReactJS, NodeJS)
- MEAN (MongoDB, ExpressJS, AngularJS, NodeJS)
- MEVN (MongoDB, ExpressJS, VueJS, NodeJS)
- Ruby on Rails
- Django
- Java Spring
- ASP.NET



MERN Stack



Pada topik ini kita bakal belajar teknologi-teknologi yang paling up to date dengan industri. Supaya mempermudah belajar, berikut Techstack sesuai kategorinya:

1. Untuk Database

- PostgreSQL
- MongoDB

2. Untuk Server

- NodeJS
- ExpressJS
- Firebase

3. Untuk Web

- NodeJS
- ReactJS
- NextJS
- Firebase

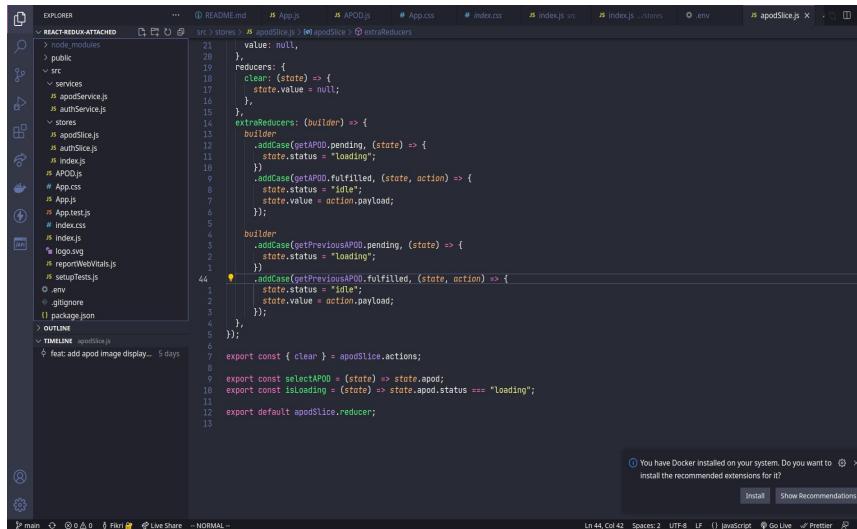


Ketika kamu menjadi seorang developer, kamu bakal menjadi seorang tukang web. Pak tukang kalo lagi nukang kan butuh paku, palu, semen, dan lain-lain. Nggak beda sama kita tukang web, kita juga butuh peralatan.

Terus apa aja sih Tools-nya?

Cuma 2 hal ini kok yang sangat wajib kalian punya :

- Text Editor
- Terminal



The screenshot shows a code editor interface with a dark theme. On the left is a file tree for a 'REACT-REDUX-ATTACHED' project. The tree includes 'node_modules', 'src' (containing 'services' with 'apodService.js' and 'authService.js'), 'stores' (with 'apodSlice.js' and 'extraReducers.js'), and various configuration files like 'index.js', 'App.css', 'App.js', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js', and environment files ('env', 'gitignore', 'package.json'). The right pane displays the content of 'apodSlice.js'. The code defines a slice reducer for the 'apod' store. It starts with an initial state where 'value' is null. The 'reducers' object contains a 'clear' case that sets 'state.value' to null. An 'extraReducers' object uses a builder pattern to handle actions related to fetching an APOD. It adds cases for 'getAPOD.pending', which sets 'state.status' to 'loading'; 'getAPOD.fulfilled', which sets 'state.status' to 'idle' and 'state.value' to the action's payload; and 'getPreviousAPOD.pending', which also sets 'state.status' to 'loading'. Finally, it adds a case for 'getPreviousAPOD.fulfilled' that sets 'state.status' to 'idle' and 'state.value' to the action's payload. The code concludes with an export statement for the 'apodSlice.reducer'.

```
value: null,
},
reducers: {
  clear: (state) => {
    state.value = null;
  },
},
extraReducers: (builder) => {
  builder
    .addCase(getAPOD.pending, (state) => {
      state.status = "loading";
    })
    .addCase(getAPOD.fulfilled, (state, action) => {
      state.status = "idle";
      state.value = action.payload;
    })
    .addCase(getPreviousAPOD.pending, (state) => {
      state.status = "loading";
    })
    .addCase(getPreviousAPOD.fulfilled, (state, action) => {
      state.status = "idle";
      state.value = action.payload;
    });
},
};

export const { clear } = apodSlice.actions;
export const selectAPOD = (state) => state.apod;
export const isLoading = (state) => state.apod.status === "loading";
export default apodSlice.reducer;
```

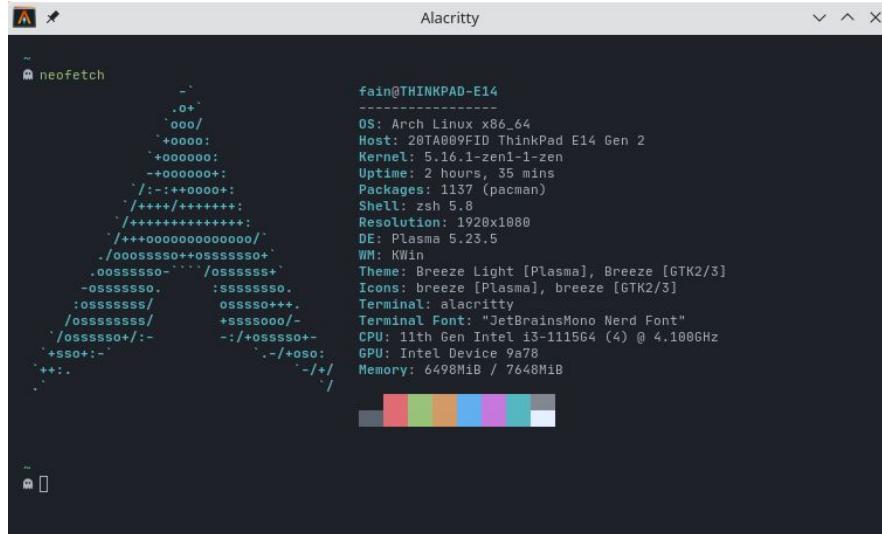
You have Docker installed on your system. Do you want to install the recommended extensions for it?

Install Show Recommendations

Ln 44, Col 42 Spacing: 2 UTF-8 (JavaScript Go Live Prettier)

Text Editor

Text editor adalah aplikasi yang membantu kita dalam menulis kode. Misalnya seperti koding dan bobok.



```
fain@THINKPAD-E14
-----
OS: Arch Linux x86_64
Host: 20TA009FID ThinkPad E14 Gen 2
Kernel: 5.16.1-zen1-1-zen
Uptime: 2 hours, 35 mins
Packages: 1137 (pacman)
Shell: zsh 5.8
Resolution: 1920x1080
DE: Plasma 5.23.5
WM: KWin
Theme: Breeze Light [Plasma], Breeze [GTK2/3]
Icons: breeze [Plasma], breeze [GTK2/3]
Terminal: alacritty
Terminal Font: "JetBrainsMono Nerd Font"
CPU: 11th Gen Intel i3-1115G4 (4) @ 4.100GHz
GPU: Intel Device 9s78
Memory: 6498MiB / 7648MiB
```

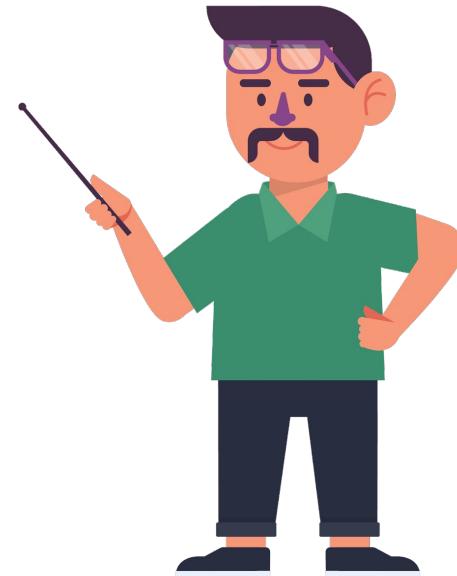
Terminal

Terminal berfungsi untuk **ngejalanin website kita** dan ngetes apakah website kita udah berjalan dengan baik atau belum. Terminal ini nggak perlu di install karena udah ada di device kalian masing-masing.

Gambar disamping itu contoh terminal yang ada di **GNU/Linux**. Bagi pengguna Windows, ada yang namanya **Powershell**, sedangkan kalau kalian pengguna MacOS kalian juga punya terminal namanya **iTerm**.

Alat sakti sebagai text editor

Seiring dengan makin canggihnya teknologi, maka makin canggih juga alat untuk ngoding. Banyak aplikasi-aplikasi **top-notch** yang bisa kamu pake. Nah aplikasi ini biasanya disebut dengan **IDE** (*Integrated Development Environment*).

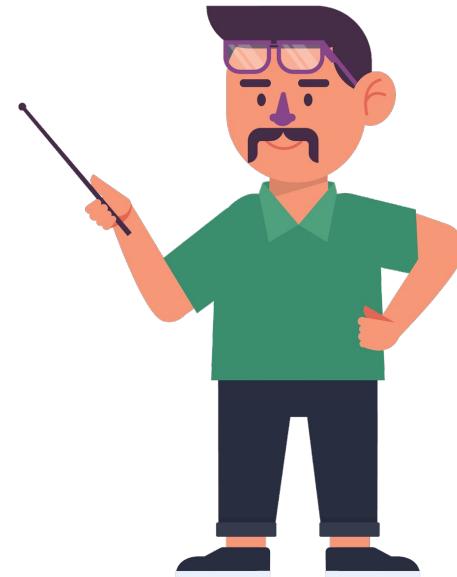




Nah, mas Gus nyaranin kamu pake IDE karena selain canggih IDE-IDE ini juga populer.

Yuk kita simak ada IDE apa aja yang *recommended* banget buat dicobain.

- **Visual Studio Code** (buatan Microsoft, 10/10 lah)
- **JetBrains IntelliJ Idea** (buatan JetBrains)
- **Atom** (bikinan Github, yang sekarang jadi bawahan Microsoft xixixi)





Selain itu, dengan menggunakan IDE, kamu dapet all in one peralatan buat ngoding.

IDE bisa dipake untuk ngoding dan juga bisa dipake buat jalankan aplikasi tanpa lewat terminal. Bahkan kalau kalian masih insist buat pake terminal, kalian juga bisa buka terminal di dalam IDE ini lho.

Nunggu apa lagi, cus langsung instal IDE abis ini.

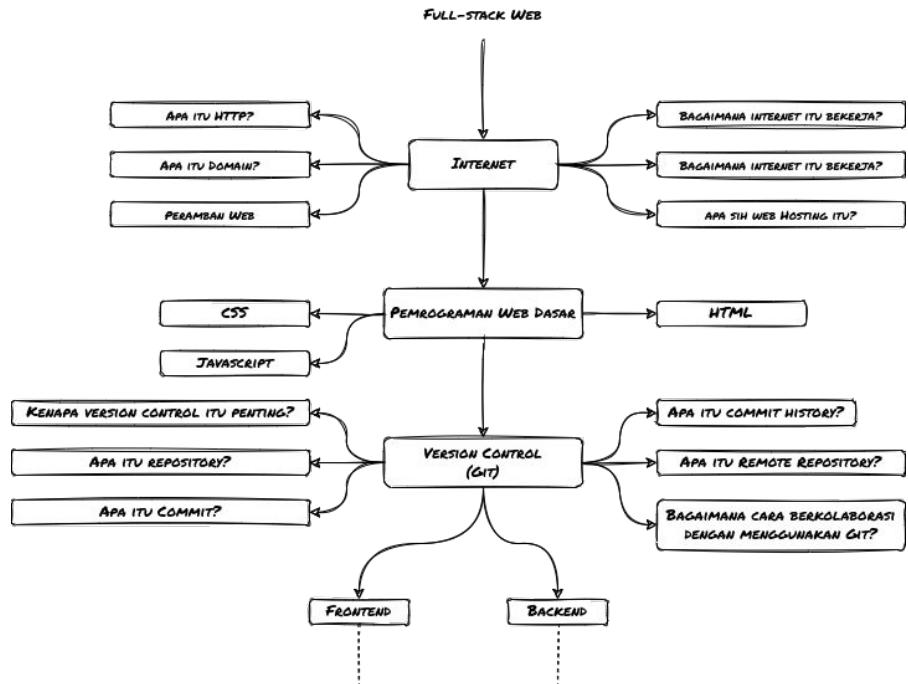
[Download Visual Studio Code](#)





Learning Path

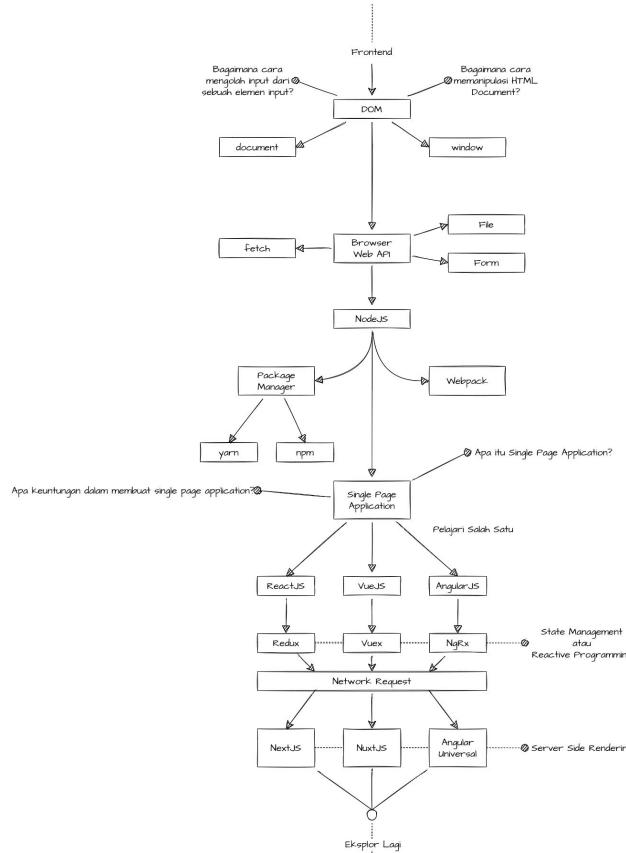
Dalam belajar untuk menjadi seorang full stack web developer, tentunya kita perlu tau dari mana kita harus memulai. Diagram berikut akan membantumu dalam mengerti harus mulai dari mana.





Learning Path - Frontend

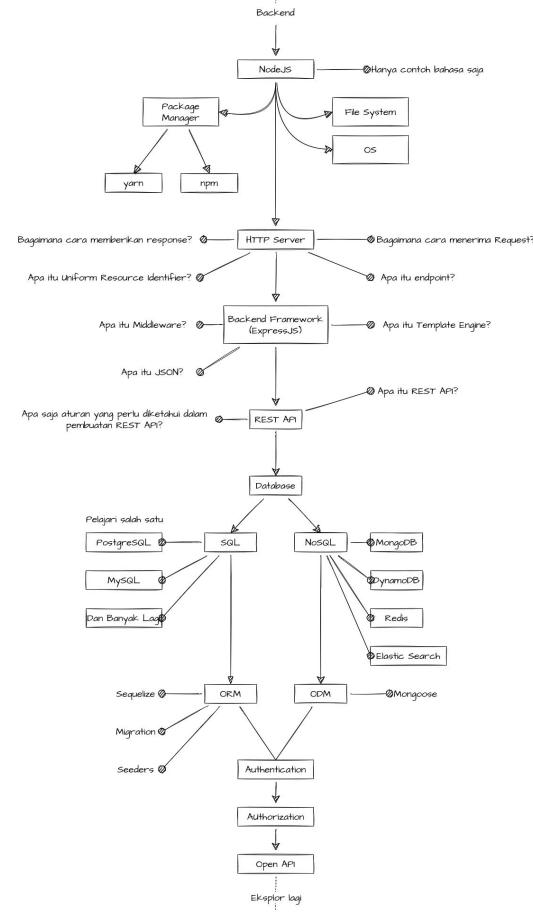
Peta konsep berikut akan membantumu dalam belajar full stack web developer apabila kalian ingin memulai dari frontend terlebih dahulu.





Learning Path - Backend

Peta konsep berikut akan membantumu dalam belajar full stack web developer apabila kalian ingin memulai dari backend terlebih dahulu.



Saatnya kita Quiz!





1. Berikut ini adalah pernyataan yang benar tentang situs web, kecuali..

- A. Tempat menampung informasi yang dapat diakses melalui web browser
- B. Terdapat dua jenis situs web, yaitu situs web statis dan situs web dinamis
- C. Situs web hanya diakses oleh seorang full stack developer saja



1. Berikut ini adalah pernyataan yang benar tentang situs web, kecuali..

- A. Tempat menampung informasi yang dapat diakses melalui web browser
- B. Terdapat dua jenis situs web, yaitu situs web statis dan situs web dinamis
- C. Situs web hanya diakses oleh seorang full stack developer saja

Situs web dapat diakses oleh semua orang selama ia termasuk ke dalam domain publik dan dapat diakses melalui internet.

2. Pernyataan sebab akibat yang benar adalah..

- A. Karena aku seorang full stack developer maka aku harus mengembangkan aplikasi frontend dan backend sekaligus
- B. Karena binaracademy.com adalah sebuah situs web statis maka halaman-halamannya akan diproses oleh server agar tampilan halaman tersebut disesuaikan preferensi user
- C. Karena aku ingin membuat sebuah situs web dinamis maka aku hanya perlu belajar HTML, CSS, dan Javascript

2. Pernyataan sebab akibat yang benar adalah..

- A. Karena aku seorang full stack developer maka aku harus mengembangkan aplikasi frontend dan backend sekaligus
- B. Karena binaracademy.com adalah sebuah situs web statis maka halaman-halamannya akan diproses oleh server agar tampilan halaman tersebut disesuaikan preferensi user
- C. Karena aku ingin membuat sebuah situs web dinamis maka aku hanya perlu belajar HTML, CSS, dan Javascript

Frontend dan backend adalah cakupan dari pekerjaan full stack developer, karena tanpa kedua benda tersebut, sebuah situs web tidak dapat berjalan.



3. Manakah pernyataan-pernyataan yang benar di bawah ini?

- A. Backend adalah server side application; Frontend adalah client side application;
- B. Backend adalah client side application; Frontend adalah server side application;
- C. Backend dan frontend keduanya merupakan server side application;



3. Manakah pernyataan-pernyataan yang benar di bawah ini?

- A. Backend adalah server side application; Frontend adalah client side application;
- B. Backend adalah client side application; Frontend adalah server side application;
- C. Backend dan frontend keduanya merupakan server side application;

Backend adalah server side application; Frontend adalah client side application



4. Sabrina sedang mengembangkan sebuah situs bernama belajarbarengsabrina.com, situs tersebut hanya akan menyajikan konten yang akan selalu berubah karena kursus di situs tersebut akan selalu Sabrina perbarui. Untuk memperbarui konten di dalam situs tersebut Sabrina harus merubah HTML, CSS dan Javascript-nya saja. Kira-kira tipe situs yang sedang dikerjain Sabrina tuh situs web apa ya?

- A. Situs Web Dinamis (Dynamic Website)
- B. Situs Web Statis (Static Website)
- C. Client Side Application



4. Sabrina sedang mengembangkan sebuah situs bernama belajarbarengsabrina.com, situs tersebut hanya akan menyajikan konten yang akan selalu berubah karena kursus di situs tersebut akan selalu Sabrina perbarui. Untuk memperbarui konten di dalam situs tersebut Sabrina harus merubah HTML, CSS dan Javascript-nya saja. Kira-kira tipe situs yang sedang dikerjain Sabrina tuh situs web apa ya?

- A. Situs Web Dinamis (Dynamic Website)
- B. Situs Web Statis (Static Website)
- C. Client Side Application

Ketika perubahan konten hanya perlu dilakukan di HTML, CSS dan Javascript saja untuk menambahkan informasi baru, maka situs tersebut adalah situs web statis, karena TIDAK memerlukan backend.



5. Berikut ini adalah pernyataan yang salah tentang frontend,

- A. Bagian dari situs web yang bertugas untuk mengolah informasi dan logika bisnis
- B. Bagian dari situs web yang bertugas untuk mengolah informasi menjadi tampilan
- C. Bagian dari situs web yang berguna untuk menerima input dari penggunanya melalui tampilan



5. Berikut ini adalah pernyataan yang salah tentang frontend

- A. Bagian dari situs web yang bertugas untuk mengolah informasi dan logika bisnis
- B. Bagian dari situs web yang bertugas untuk mengolah informasi menjadi tampilan
- C. Bagian dari situs web yang berguna untuk menerima input dari penggunanya melalui tampilan

Mengolah informasi dan logika bisnis adalah tugas dari backend



6. Fitur yang mengimplementasi fungsi pengiriman email ke pengguna setelah pengguna melakukan registrasi di situs web ini di bagian situs web yang mana ya?

- A. Backend
- B. Frontend
- C. Full Stack



6. Fitur yang mengimplementasi fungsi pengiriman email ke pengguna setelah pengguna melakukan registrasi di situs web adalah bagian dari

- A. Backend
- B. Frontend
- C. Full Stack

Pengiriman email ke pengguna saat pengguna berhasil melakukan registrasi ini termasuk bagian dari logika bisnis, maka dari itu fitur ini akan diimplementasikan pada bagian backend suatu situs web



7. Bagian situs web yang bertugas untuk melayani penggunanya secara langsung adalah

- A. Server Side Application
- B. Backend
- C. Frontend



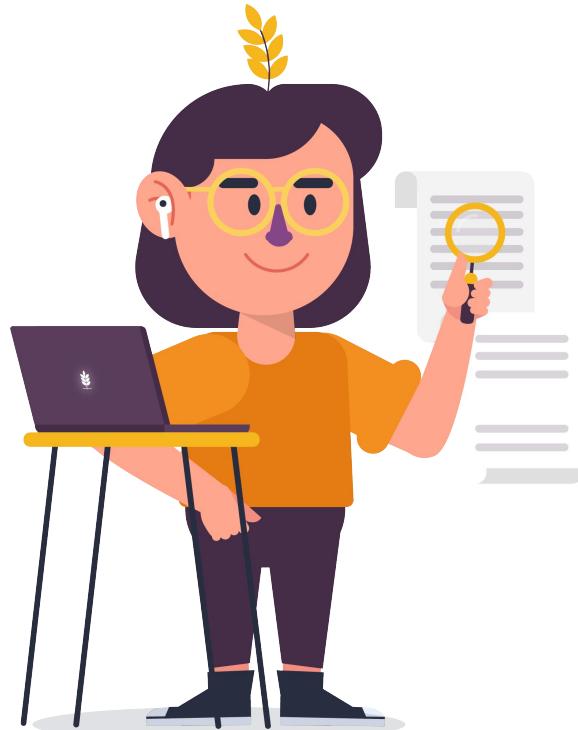
7. Bagian situs web yang bertugas untuk melayani penggunanya secara langsung adalah

- A. Server Side Application
- B. Backend
- C. Frontend

Backend hanya akan melayani request dari frontend saja, sedangkan frontend akan melayani segala aktifitas penggunanya.

Referensi dan bacaan lebih lanjut~

1. <https://scrum.org>
2. <https://en.wikipedia.org/wiki/Website>





Nah, selesai sudah pembahasan kita di Chapter 1 Topic 1 ini.

Selanjutnya, kita bakal bahas penggunaan HTML dalam membuat web apps

Penasaran kayak gimana? Cus langsung ke topik selanjutnya~



Terima Kasih!



Next Topic

loading...



Terima Kasih

