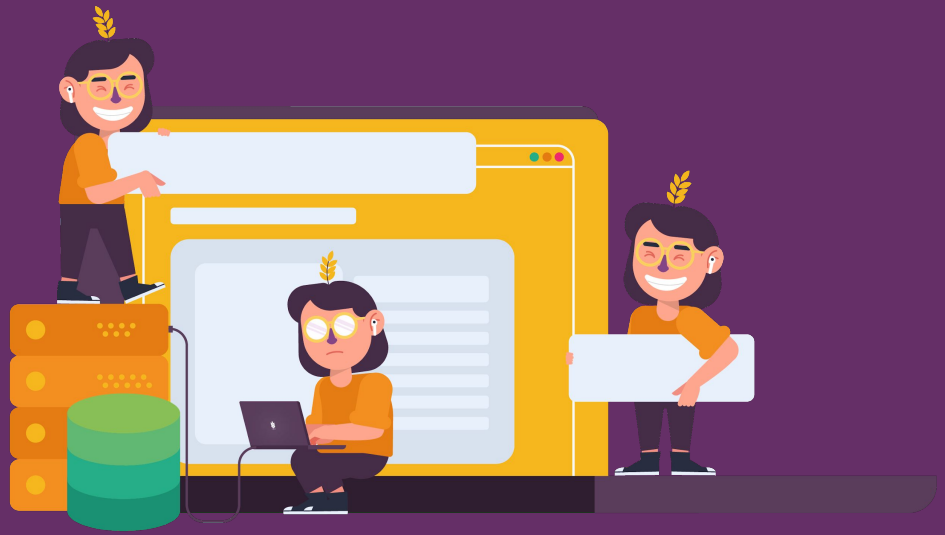
Fullstack Web (Pengantar)

Gold - Chapter 3 - Topic 1

Selamat datang di Chapter 3 *online course* Full-Stack Web dari Binar Academy!



Di **Chapter 3**, ada enam topik yang akan kamu pelajari untuk mengerti **teknis pemrograman *Full-stack Web***.

Di materi pertama, kita akan mengulas lebih jauh tentang ***Full-stack Engineering*** beserta cakupannya.

Dari sesi ini, harapannya kamu bisa mendapatkan beberapa hal, antara lain:

a. Menemukan **definisi** Full-stack Web
b. Tahu **cakupan pekerjaan** Full-stack Web
c. Ngerti soal **Arsitektur Monolith** dan **Microservices**
d. Paham ***job desc*** dari *Full-stack Engineer*
e. Paham ***career path*** untuk *Full-stack Engineer*

BINAR

Apa sih sebenernya **Fullstack Engineering itu?**

Sebelum memahami Fullstack, kita bahas dulu pengembangan website itu kaya gimana. Coba pakai analogi, ya!

Sebuah restoran punya dua unsur utama dalam pengoperasiannya: bagian pelanggan (kasir, pramusaji dan dekorasi) serta dapur (koki dan staf masak).

Hal tersebut juga berlaku untuk website, yang sama-sama memiliki dua bagian pengoperasian, yaitu: ***Frontend*** (Tampilan yang dilihat *user*) dan ***Backend*** (*Service* yang mengurusi segala sesuatu tentang data).

Kalau kita korelasikan lagi dengan analogi restoran, maka bagian pelanggan adalah *Frontend*-nya Restoran. Sedangkan, bagian dapur adalah *Backend* dari Restoran.

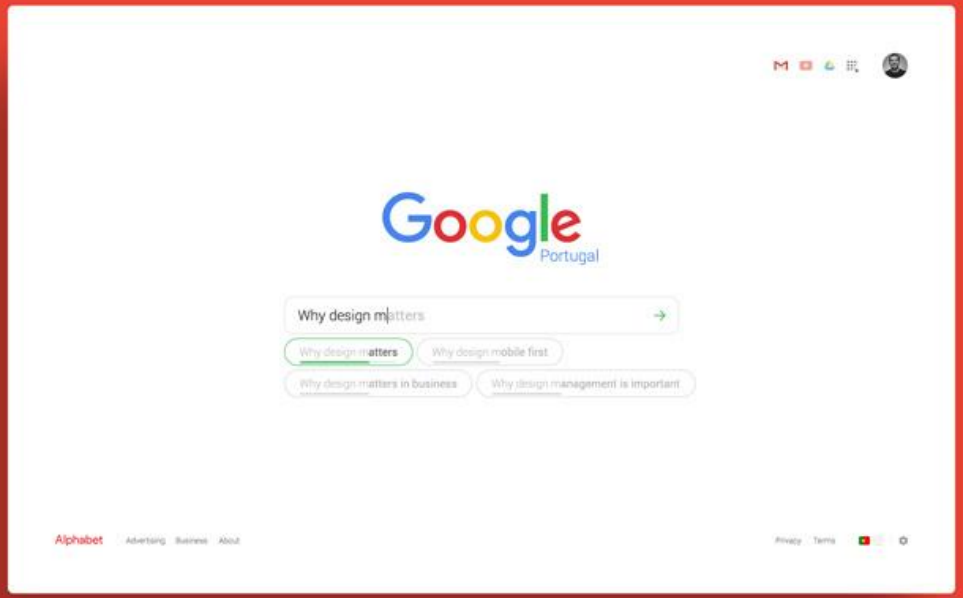
Oke, kita sudah tahu kalau sebuah website itu terdiri dari *Frontend* dan *Backend*.

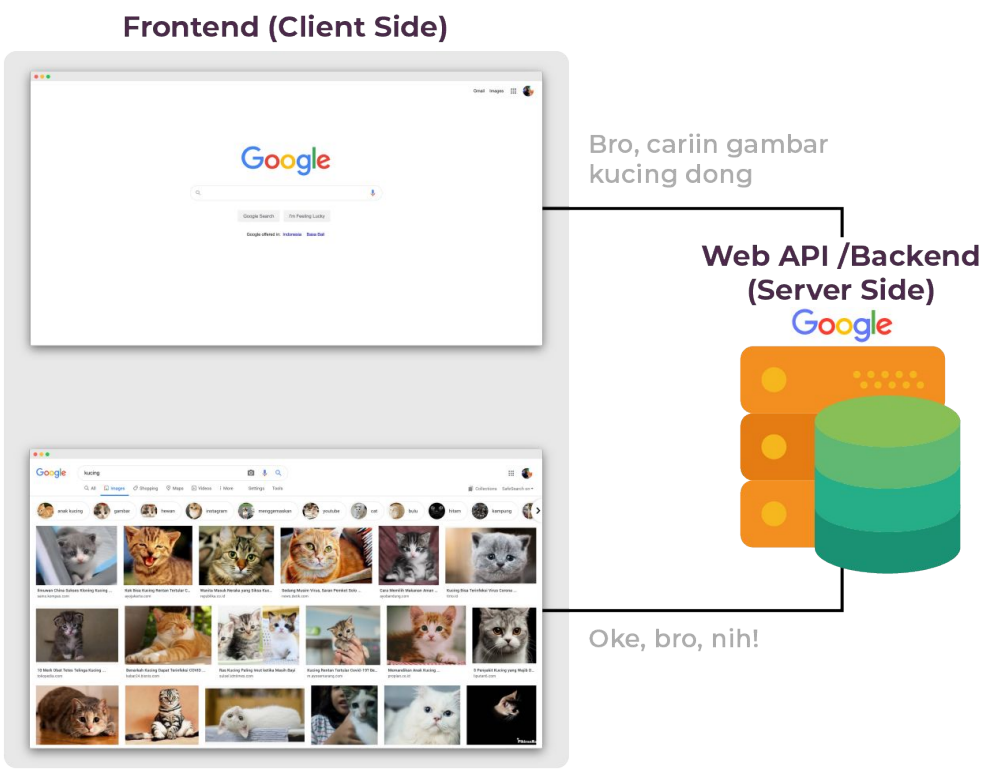
Sekarang, mari kita bahas lebih dalam tentang apa itu ***Frontend*** dan ***Backend***!

Pada umumnya, sebuah website itu punya yang ***webpage*/tampilan** *(Frontend)* dan **Web API** *(Backend)*.

Apa yang kamu lihat di sini adalah sebuah *webpage* dari *google.com.* Di halaman ini, kamu bisa melakukan pencarian berdasarkan apa yang kamu masukkan di dalam *search bar*.

Terus, apa yang bakal terjadi kalau kita tekan *enter* sesudah mengisi *keyword* di dalam *search bar*?





Yang terjadi adalah

Frontend akan menerima pesananmu, yaitu **"Gambar Kucing".** Selanjutnya, ia akan menyampaikannya kepada koki di dapur agar bisa disiapkan.

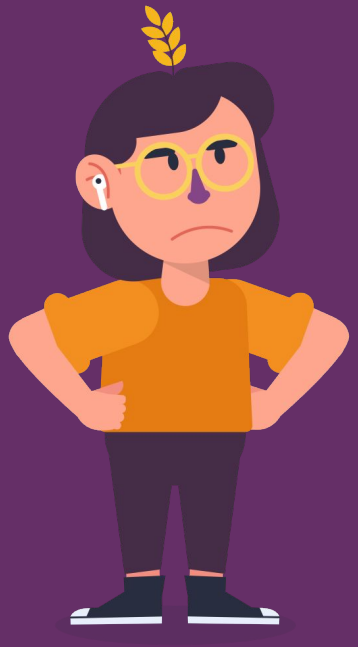
Nah, pihak ***backend*** akan mengolah pesananmu tadi. Setelah selesai diolah, pesanan akan diambil pihak *frontend*, untuk selanjutnya disajikan kepada kamu.

Secara Teknis...

Frontend di website akan melakukan ***request*** ke ***backend***, biasanya menggunakan protokol **HTTP**. Setelah *request* diterima, *backend* akan melakukan apa yang diminta dan memberi jawaban. Jawaban ini disebut dengan ***response***.

Dari sini, bisa disimpulkan, bahwa *frontend* adalah bagian dari website yang mempermudah kita dalam menyampaikan *request* ke *backend*.

Dan, *backend* sendiri merupakan bagian dari website yang berfokus kepada penyajian data untuk dikonsumsi oleh *frontend*. Nah, daftar permintaan ini akan dicatat di tempat yang bernama ***server backend***.

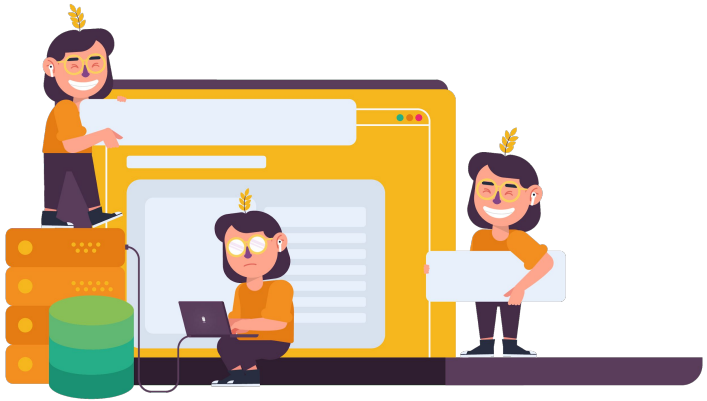


”Terus, Fullstack Sendiri itu apa?”

Fullstack itu ibarat kita buka usaha *street food*.

Kita merangkap menjadi lebih dari satu bagian. Masak sendiri, cuci sendiri, penyajiannya juga sendiri. Tapi, yang makan tetap orang lain~

Nah, ***Full stack Engineering*** dituntut untuk mengurusi baik *Frontend* maupun *Backend* dari suatu website di waktu yang bersamaan.



Seperti yang kita bicarakan tadi, cakupan kerja *full stack* adalah implementasi desain dan merubahnya menjadi tampilan yang bisa dipahami *user (frontend)* sekaligus membuat API yang dipakai untuk melakukan hal-hal yang berhubungan dengan *database (backend)*.

Kalau dibuat detail, jadinya seperti ini:

- Membuat **Web API** *(Backend)* untuk mengurusi segala sesuatu tentang penyimpanan data.
- Membuat ***User Interface*** *(Frontend)* yang dapat dilihat melalui *browser* dan bekerja sesuai fungsinya.
- Membuat desain database yang reliable dan bisa diuji ketahanannya

Sekadar info, di dalam dunia pemrograman, kamu harus bisa menjadi seorang *Generalist* atau *Jack of All Trades*, yakni bisa semua hal. Mengapa? Karena dunia IT itu sangat dinamis; sering berubah-ubah dalam waktu yang singkat.

Nanti, bakal dijelaskan kok kiat sukses memikul tuntutan yang berat itu :D

Tapi, sebelum ke sana, kita pahami dulu arsitektur dari sebuah website, ya!

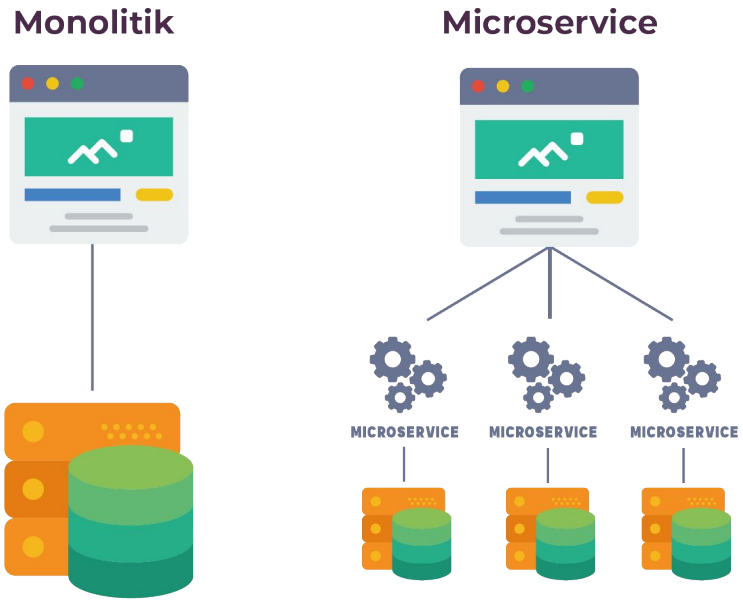
Ada dua tipe arsitektur website yang banyak digunakan, yaitu:

1. **Monolitik**

Di arsitektur ini, *backend* dan *frontend* mengerjakan aplikasi tersebut di satu *project* yang sama. Jadi, keduanya dikerjakan dalam satu server yang sama juga.

2. **Microservice**

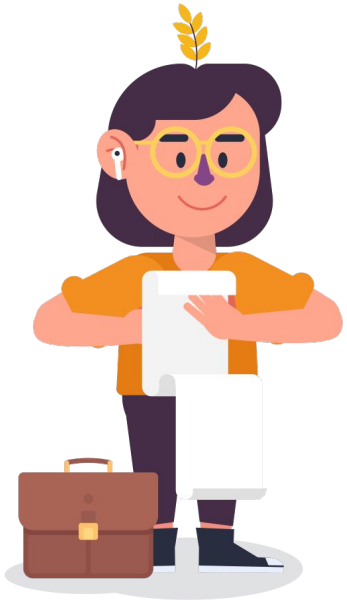
Di sini, *backend* dan *frontend* memiliki dua *repository/project* yang berbeda. Otomatis, keduanya berada di server yang berbeda juga.



Arsitektur yang dipakai dalam pengembangan website itu akan sangat bergantung kepada ***use case***. Apabila web sudah terlalu besar maka biasanya *engineer* akan **memecahnya menjadi aplikasi-aplikasi kecil** agar kode yang dibuat lebih terstruktur dan terfokus. Nah, inilah yang kita sebut dengan **arsitektur *Microservice*.**

Namun, apabila web kita masih tergolong baru menerapkan **arsitektur Monolitik** saja sudah cukup. Mengapa? Karena pengembangannya lebih cepat dibanding *microservice*.

Tenang! Kalau perfoma yang menjadi kendala, kita bisa mengubah arsiteknur monolitik ke arsitektur *microservice* saat dibutuhkan. Karena, sejatinya *microservice* itu berasal dari monolitik juga~



Oke, karena kalian sudah tahu arsitektur sebuah web itu kayak gimana, sekarang kita bisa omongin cakupan pekerjaan seorang full stack developer.

Cakupannya meliputi:

- Membuat Web API (Backend) yang berfungsi untuk mengurusi segala sesuatu tentang penyimpanan data.
- Membuat User Interface (Frontend) yang dapat dilihat melalui browser dan berfungsi sesuai fungsinya.
- Membuat desain database yang reliable dan bisa diuji ketahanannya

Oh, iya! Tuntutan ***Full-stack Engineer*** memang tampak banyak.

Untuk mempermudah, kamu harus punya ***skill set*** yang tepat!

Full-stack engineer harus memiliki skill yang dimiliki oleh *backend* dan *frontend engineer*, antara lain:

Frontend

1. Menguasai Javascript
2. Menguasai CSS
3. Menguasai HTML
4. Menguasai setidaknya salah satu framework ini:
    - React.js
    - Vue.js
    - Angular.js
5. Dapat mengimplementasi desain dengan baik
6. Dan lain-lain

Backend

1. Menguasai setidaknya satu bahasa pemrograman yang bisa digunakan untuk membuat Web API, seperti **Ruby, Python, Node.js,** serta **PHP**
2. Menguasai DBMS, baik SQL maupun Non-SQL, sebagai contoh:
    - Postgresql
    - MongoDB
3. Menguasai Desain Database
4. Dan lain-lain

Banyak banget, ya...

Nah, ada beberapa ***Stack*** (kumpulan teknologi) yang bisa kita jadikan acuan untuk dipelajari supaya bebannya jadi lebih ringan.

- **MERN**
  - MongoDB
  - Express
  - React
  - Node JS
- **LAMP**
  - Linux
  - Apache
  - MySQL
  - PHP
- **Ruby on Rails**
  - Ruby
  - SQLite
  - Rails
- **Django**
  - Python
  - Django
  - MySQL

Tapi, di Binar Academy ini, kalian bakal belajar stack-stack ini aja:

- Node.js
- SQL (Backend)
- Express (Backend)
- React (Frontend)

Dan juga, nanti arsitektur yang kita gunakan di dalam aplikasi kita adalah arsitektur monolitik dan microservice. Karena nanti kita akan membuat dua aplikasi yang berbeda untuk backend dan frontend nya (Microservice). Dan juga, di kita akan membuat tampilan juga di server backend. (Monolitik)

BINAR

Bootstrap Layouting

Sebuah role dalam suatu website yang berfokus pada pembuatan tampilan yang dipakai oleh user.

Back-end Engineering

Sebuah role dalam suatu website yang berfokus pada database dan manajemen-nya menggunakan WebAPI.

Fullstack Engineering

Sebuah role dalam suatu website dimana role ini menguasai Frontend dan Backend Sekaligus

BINAR

Saatnya

Quiz

BINAR

1

Berikut ini yang merupakan scope pekerjaan Frontend

A. Membuat tampilan

B. Mengatur database

C. Membuat Web API

BINAR

2

Pembuatan desain database merupakan salah satu tanggung jawab dari

A. Frontend
B. Backend
C. UI/UX

BINAR

3

Sebuah arsitektur aplikasi dimana Frontend dan Backend dibuat menjadi satu aplikasi saja disebut dengan arsitektur

A. Monolitik
B. Microservice
C. MongoDB

BINAR

4

Berikut ini adalah teknologi yang harus dikuasai oleh seorang full stack engineer

A. HTML, CSS, Javascript, dan Ruby/Java/Node.js/Python

B. HTML, CSS, Javascript

C. UI/UX design

BINAR

5

Berikut ini adalah stack untuk mempelajari MERN

A. MongoDB, Express, Ruby, NodeJS

B. MySQL, Express, Ruby, NodeJS

C. MongoDB, Express, React, NodeJS

BINAR

Pembahasan

Quiz

BINAR

1

Berikut ini yang merupakan scope pekerjaan Frontend

A. Membuat tampilan website

B. Mengatur database

C. Membuat Web API

BINAR

2

Pembuatan desain database merupakan salah satu tanggung jawab dari

A. Frontend

B. Backend

C. UI/UX

BINAR

3

Sebuah arsitektur aplikasi dimana Frontend dan Backend dibuat menjadi satu aplikasi saja disebut dengan arsitektur

A. Monolitik

B. Microservice

C. MongoDB

BINAR

4

Berikut ini adalah teknologi yang harus dikuasai oleh seorang full stack engineer

A. **HTML, CSS, Javascript, dan Ruby/Java/Node.js/Python**
B. HTML, CSS, Javascript
C. UI/UX design

BINAR

5

Berikut ini adalah *stack* untuk mempelajari MERN

A. MongoDB, Express, Ruby, NodeJS

B. MySQL, Express, Ruby, NodeJS

C. MongoDB, Express, React, NodeJS

BINAR

Terima

Kasih