

Article 1 (Perbedaan FE dengan BE)

1.1 Apa Perbedaan Antara Frontend dan Backend dalam Pengembangan Aplikasi?

Frontend dan *backend* adalah dua aspek penting dari aplikasi apa pun. *Frontend* adalah apa yang pengguna Anda lihat dan termasuk elemen visual, seperti tombol, kotak centang, grafik, dan pesan teks. Hal ini memungkinkan pengguna Anda untuk berinteraksi dengan aplikasi Anda. *Backend* adalah data dan infrastruktur yang membuat aplikasi Anda berfungsi. *Backend* menyimpan dan memproses data aplikasi untuk pengguna Anda.

1.1.1 Bagaimana cara kerja frontend aplikasi?

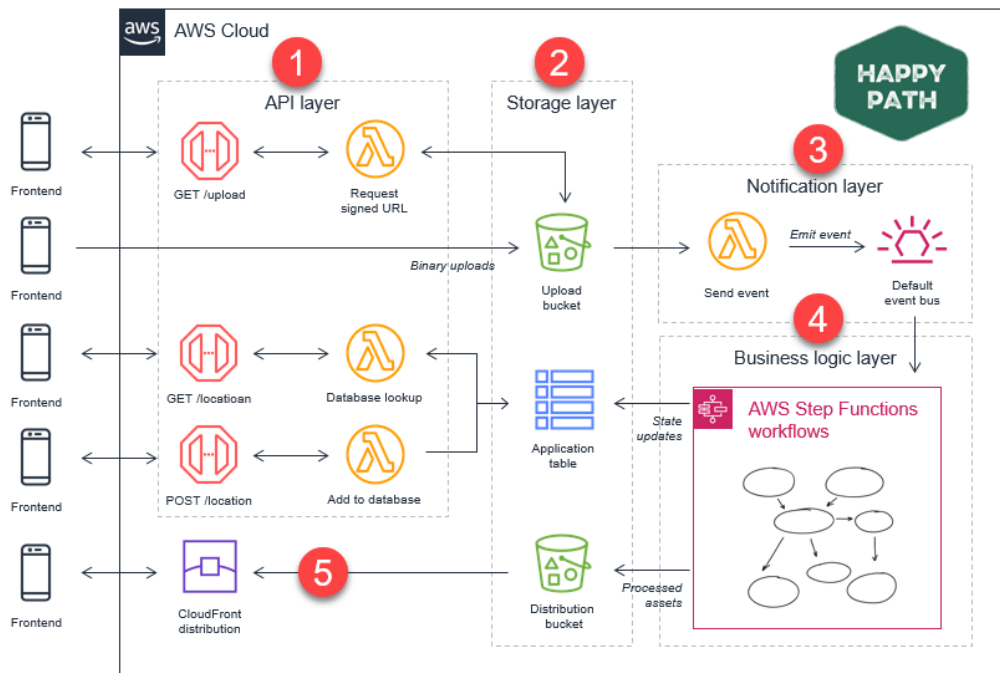
Istilah *frontend* mengacu pada antarmuka pengguna grafis (GUI) yang dapat berinteraksi langsung dengan pengguna Anda, seperti menu navigasi, elemen desain, tombol, gambar, dan grafik. Secara teknis, halaman atau layar yang dilihat pengguna Anda dengan beberapa komponen UI disebut *model objek dokumen* (*DOM*). Ada Tiga bahasa komputer utama memengaruhi cara pengguna Anda berinteraksi dengan *frontend* yaitu *HTML*, *CSS*, *JavaScript*.

JavaScript dapat memicu perubahan pada halaman dan menampilkan informasi baru. Artinya, *frontend* dapat menangani interaksi (atau permintaan) pengguna mendasar, seperti menampilkan kalender atau memeriksa apakah pengguna Anda telah memasukkan alamat email yang valid. *Frontend* meneruskan permintaan yang lebih kompleks ke *backend*.

1.1.2 Bagaimana cara kerja backend aplikasi?

Terkadang disebut *sisi server*, *backend* aplikasi Anda mengelola fungsionalitas aplikasi web Anda secara keseluruhan. Saat pengguna Anda berinteraksi dengan *frontend*, interaksi mengirimkan permintaan ke *backend* dalam format HTTP. *Backend* memproses permintaan dan mengembalikan respons. Saat *backend* Anda memproses permintaan, biasanya *backend* berinteraksi dengan *Server database*, *microservices*, *API*.

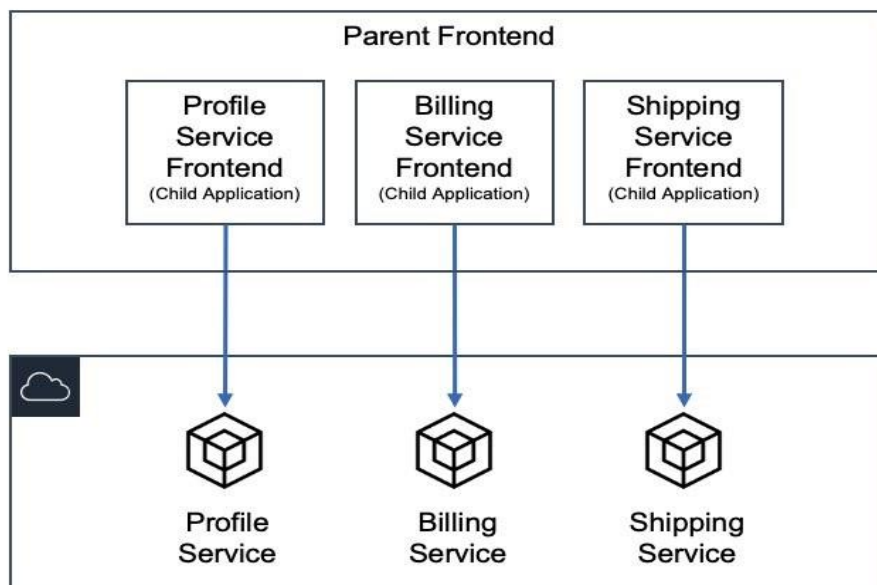
Backend menggunakan beberapa protokol komunikasi dan teknologi untuk menyelesaikan permintaan. Selain itu, *backend* menangani ribuan permintaan berbeda secara bersamaan. *Backend* menggabungkan teknik konkurensi dan paralelisme, seperti mendistribusikan permintaan di banyak server, *caching*, dan duplikasi data.



Gambar 1. Cara Kerja Frontend dan Backend

1.2 Perbedaan utama: frontend vs. backend

Frontend dan *backend* adalah istilah umum yang mengelompokkan berbagai teknologi dan lapisan *software* aplikasi apa pun. *Frontend* berfokus pada aspek-aspek yang dapat dilihat pengguna Anda. Sebaliknya, *backend* adalah segala sesuatu yang membuat aplikasi berfungsi.



Gambar 1.2 Perbedaan Utama Lainnya Antara *Frontend* Dan *Backend*.

1.2.1 Tujuan pengembangan

Pengembangan *full-stack* bertujuan untuk membuat aplikasi yang responsif, efisien, dan fungsional. Developer *frontend* dan *backend* memiliki tujuan yang berbeda untuk membuat aplikasi akhir.

Developer *frontend* bertujuan untuk mengembangkan pengalaman pengguna yang positif, mengoptimalkan aksesibilitas dan performa aplikasi, dan membuat desain responsif. Sedangkan developer *backend* membangun lalu mempertahankan operasi sisi server aplikasi. Tujuan pengembangan utama mereka adalah membuat arsitektur yang andal yang melakukan fungsi aplikasi secara akurat dan efisien.

1.2.2 Teknologi

Pengembangan *frontend* berhubungan dengan sisi situs web yang menghadap pengguna. Teknologi ini mencakup bahasa komputer, seperti JavaScript, CSS, dan HTML. Pengembangan perangkat lunak *frontend* juga menggunakan kerangka kerja *frontend* untuk mempercepat efisiensi produksi.

Pengembangan *backend* menggunakan bahasa pemrograman, seperti Ruby, Java, dan Python untuk menulis logika sisi server. Developer *backend* juga menggunakan *database*, teknologi penyimpanan, dan teknologi API sehingga aplikasi dan sistem dapat berkomunikasi satu sama lain.

1.2.3 Konkurensi

Konkurensi adalah kemampuan aplikasi untuk menjalankan beberapa tugas secara bersamaan. Di *frontend*, setiap pengguna memiliki salinan aplikasi mereka sendiri di peramban atau aplikasi seluler mereka. Di sisi lain, *backend* mungkin harus menangani ribuan permintaan secara bersamaan.

Dalam pendekatan komputasi terdistribusi, developer dapat membagi tugas *backend* lebih dari beberapa simpul sehingga *backend* mengelola beban kerja intensif data secara bersamaan.

1.2.4 Caching

Caching menyimpan sementara salinan *file* aplikasi sehingga membuatnya lebih mudah untuk diambil kembali saat diperlukan. Di *frontend* aplikasi klien menyimpan data, seperti gambar *header* saat pertama kali pengguna mengaksesnya. *Frontend* memuat *file cache* untuk meningkatkan performa saat kali berikutnya mereka mengakses konten yang sama.

Pengembangan *backend* menggunakan *caching* untuk mengurangi beban pada server aplikasi. Konten yang di-*cache* mencakup halaman statis, hasil kueri *database*, respons API, data sesi, gambar, dan video.

Salah satu strategi adalah menyimpan *file* pada jaringan pengiriman konten (CDN) yang bertindak sebagai perantara antara *frontend* dan *backend*. Setiap kali pengguna membuat permintaan di *frontend*, CDN memeriksa apakah data tersedia dan merespons secara langsung.

1.2.5 Keamanan

Keamanan *frontend* berfokus pada komponen yang menghadap pengguna. Hal ini termasuk formulir input, skrip sisi klien, dan pengalaman pengguna dalam alur kerja keamanan, seperti autentikasi.

Strategi keamanan pengembangan *frontend* yang umum mencakup validasi input, pengaturan penonaktifan yang memungkinkan pengguna memasukkan kode di kotak teks, dan alur kerja autentikasi multifaktor.

Keamanan *backend* berfokus pada keamanan data dalam penyimpanan dan saat bergerak. Keamanan *backend* mengelola semua aspek autentikasi, kontrol akses, dan keamanan sesi. Keamanan *backend* mencakup semua layanan *backend*, termasuk *database* yang terhubung, API, dan bahasa pemrograman sisi server. Praktik keamanan *backend* inti mencakup pengodean yang aman, enkripsi data sensitif sebelum dan sesudah transmisi, serta sistem otorisasi dan autentikasi yang aman.

1.2.6 Keterampilan developer

Developer *frontend* memahami pengalaman pelanggan dan mengakomodasi kebutuhan pengguna. Keterampilan utama mereka melibatkan kemahiran dalam JavaScript, CSS & HTML, Framework dan pengetahuan desain dalam membuat alur pengguna yang menarik secara visual.

Developer *backend* mengodekan fungsi aplikasi dan membuat aplikasi lebih aman, bebas dari kesalahan, dan cepat. Developer *backend* mahir dalam beberapa bahasa pemrograman, seperti Python, Ruby, Java, dan PHP. Mereka paham mengenai kerangka kerja pengembangan aplikasi, seperti Django, Ruby on Rails, dan Laravel yang mengintegrasikan *frontend* dan *backend*. Developer *backend* juga tahu bagaimana mengelola dan merancang *database* relasional dan nonrelasional.

1.3 Ringkasan perbedaan: frontend vs. backend

	<i>Frontend</i>	<i>Backend</i>
Teknologi	<i>Frontend</i> menggunakan kerangka kerja HTML, CSS, JavaScript, dan <i>frontend</i> .	<i>Backend</i> menggunakan bahasa pemrograman, seperti Java, Python, Ruby, API, dan sistem manajemen database.
Konkurensi	Setiap pengguna memiliki salinan aplikasi mereka sendiri sehingga <i>frontend</i> tidak harus menangani masalah konkurensi.	<i>Backend</i> menggunakan beberapa strategi untuk menangani ribuan permintaan pengguna secara bersamaan.
Caching	Peramban atau aplikasi klien menyimpan <i>file cache</i> dari aplikasi dan menggunakannya untuk meningkatkan performa.	Sistem <i>backend</i> melakukan <i>cache file</i> di server yang berbeda atau di CDN.
Keamanan	Sebagian besar keamanan dalam pengembangan <i>frontend</i> merupakan tanggung jawab pengguna. Developer <i>frontend</i> memiliki fokus utama pada validasi input pengguna dan alur kerja autentikasi pengguna.	Keamanan dalam pengembangan <i>backend</i> lebih komprehensif untuk melindungi <i>database</i> , layanan <i>backend</i> , dan aplikasi itu sendiri. Hal ini dicapai menggunakan enkripsi, sistem autentikasi aman, dan praktik pengodean yang aman.
Tujuan Pengembangan	Pengembangan <i>frontend</i> berfokus pada pembuatan antarmuka pengguna yang berfungsi penuh, responsif, dan dirancang dengan baik.	Pengembangan <i>backend</i> melibatkan pembuatan arsitektur yang andal yang mendukung pengembangan <i>frontend</i> .
Keterampilan Pengembangan	Developer <i>frontend</i> paham tentang HTML, CSS, dan JavaScript. Mereka dapat menggunakan kerangka kerja <i>frontend</i> dan membuat halaman yang menarik secara visual. Mereka mengungkap masalah pengguna saat berinteraksi dengan aplikasi.	Developer <i>backend</i> memiliki keterampilan pengodean dan manajemen <i>database</i> . Mereka juga memahami keamanan kode dan cara menggunakan alat, platform, dan kerangka kerja saat pengembangan aplikasi.

Article 2 (Definisi Pemrograman Web)

2.1 Apa yang Dimaksud dengan Pemrograman Web?

Pemrograman web adalah proses pengembangan aplikasi atau *website* yang diakses melalui internet atau intranet. Proses ini melibatkan berbagai Tugas antara lain penulisan kode, desain halaman web, pengelolaan *database*, dan penanganan aspek teknis server web. Tujuan dari pemrograman web adalah menciptakan aplikasi yang interaktif dan fungsional sesuai kebutuhan pengguna atau bisnis.

2.2 Apa Saja yang Dipelajari dalam Pemrograman Web?

Untuk mempelajari pemrograman web, ada beberapa konsep, teknologi, dan praktik yang perlu dipelajari. Beberapa hal yang wajib dipelajari dalam pemrograman web adalah:

2.2.1 HTML dan CSS

Hal pertama yang harus dipelajari dalam pemrograman web adalah HTML & CSS. HTML adalah bahasa *markup* standar untuk membuat dan menyusun konten pada web. HTML digunakan untuk mendefinisikan struktur halaman web, termasuk teks, gambar, dan elemen multimedia lainnya.

Sementara itu, CSS adalah bahasa *stylesheet* untuk mengatur tampilan dan *formatting* halaman web yang ditulis dalam HTML. CSS memungkinkan pengembang untuk mengontrol *layout*, warna, *font*, dan bahkan animasi dalam desain web. Dengan CSS, pengembang dapat membuat halaman web yang estetik dan responsif.

2.2.2 JavaScript dan Frameworks

JavaScript adalah bahasa pemrograman tingkat tinggi yang digunakan untuk membuat halaman web menjadi interaktif dan dinamis. Dengan JavaScript, pengembang dapat menambahkan efek, mengelola form, memuat konten secara dinamis, dan interaksi pengguna lainnya.

Sementara itu, Frameworks dan Libraries JavaScript menyediakan kumpulan *pre-written JavaScript code*. Hal ini mempermudah pembuatan aplikasi web yang kompleks dan interaktif. Contoh Frameworks dan Libraries JavaScript adalah React, Angular, dan Vue.js.

2.2.3 Backend Development

Pengembangan *backend* melibatkan semua proses yang terjadi di *server*. Ini termasuk pengelolaan *database*, autentikasi pengguna, dan logika aplikasi. Hal tersebut dilakukan menggunakan berbagai bahasa pemrograman seperti Python, Ruby, PHP, Node.js, dan Java. Pengembang *backend* bertugas untuk mengembangkan API yang memungkinkan aplikasi *frontend* berkomunikasi dengan *server* dan *database*.

2.2.4 Version Control

Hal lain yang harus dipelajari dalam pemrograman web adalah *version control*. Ini merupakan sistem yang merekam perubahan pada *file* atau *set file* dari waktu ke waktu. Dengan ini, kamu bisa kembali ke versi tertentu kapan saja.

Git adalah sistem kontrol versi yang paling banyak digunakan. Hal ini memungkinkan tim *developer* untuk bekerja secara efisien melalui *branching*, *merging*, dan *versioning* kode.

2.2.5 Responsive Design

Berikutnya adalah *responsive design*. Ini merupakan pendekatan dalam *web design* yang bertujuan membuat halaman web berfungsi dengan baik di berbagai perangkat dan ukuran layar. Penggunaan CSS media queries, *flexible grid layouts*, dan gambar responsif dilibatkan dalam proses pembuatannya.

2.2.6 Testing dan Debugging

Testing adalah proses memverifikasi bahwa kode bekerja seperti yang diharapkan dan memenuhi semua persyaratan. Jenis testing yang umum digunakan adalah pengujian unit, pengujian integrasi, dan pengujian *end-to-end*.

Sementara itu, *debugging* adalah proses mengidentifikasi dan memperbaiki bugs atau kesalahan dalam kode. *Tools* dan teknik *debugging* digunakan untuk menemukan *root cause* dari masalah dan memperbaikinya.

2.2.7 Web Security

Lalu, kamu juga harus mempelajari *web security*. Konsep ini merujuk pada langkah-langkah yang diambil untuk melindungi aplikasi web dari serangan yang bisa membahayakan pengguna atau mengompromikan data. Ini termasuk penerapan protokol HTTPS, sanitasi *input* untuk mencegah SQL injection dan *cross-site scripting* (XSS), serta strategi autentikasi dan otorisasi yang aman.

2.2.8 DevOps dan Deployment

Hal terakhir yang harus dipelajari dalam pemrograman web adalah *devops* dan *deployment*. *DevOps* adalah praktik yang mengintegrasikan pengembangan (Dev) dan operasi (Ops). Tujuannya adalah untuk mempercepat proses pengembangan dan *deployment* aplikasi.

Sementara itu, *deployment* adalah proses mempublikasikan aplikasi web ke *server* sehingga dapat diakses oleh pengguna melalui internet. Layanan *cloud* seperti AWS, Google Cloud, dan Heroku sering digunakan untuk *hosting* dan mengelola aplikasi web.

2.3 Jenis-Jenis Pemrograman Web

Pemrograman web dapat dibagi menjadi beberapa jenis berdasarkan fokus dan teknologi yang digunakan. Beberapa jenis pemrograman web adalah:

2.3.1 Pemrograman Front-End

Frontend berkaitan dengan pengembangan antarmuka pengguna (UI) yang interaktif dan responsif. Umumnya, *frontend* ini menggunakan teknologi seperti HTML, CSS, dan JavaScript. Lalu, Framework dan libraries JavaScript seperti React, Angular, dan Vue.js juga sering digunakan untuk mempercepat pengembangan.

2.3.2 Pemrograman Back-End

Backend merupakan pemrograman yang berfokus pada *server*, aplikasi, dan *database*. Umumnya, pekerjaannya melibatkan penanganan logika aplikasi, autentikasi pengguna, manipulasi database, dan pengelolaan server.

Bahasa pemrograman yang sering digunakan adalah Python, Ruby, Java, PHP, dan Node.js. Sementara itu, *database* yang digunakan termasuk SQL (seperti MySQL, PostgreSQL) dan NoSQL (seperti MongoDB).

2.3.3 Pemrograman Full Stack

Jenis ini menggabungkan kedua aspek *front-end* dan *back-end* dalam pengembangan web. Umumnya, pengembang *full stack* memiliki kemampuan untuk menangani semua aspek pengembangan web, dari antarmuka pengguna hingga database. Oleh sebab itu, ini memerlukan pengetahuan luas tentang teknologi dan kerangka kerja baik di *front-end* maupun *back-end*.

2.4 Apa Saja Bahasa Pemrograman Web?

Ada beberapa bahasa pemrograman web yang sering kali digunakan oleh *developer*. Beberapa bahasa pemrograman web yang sering digunakan adalah:

2.4.1 JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi, dinamis, dan interpreted yang sangat penting dalam pengembangan web. Ini seringkali digunakan untuk menambahkan interaktivitas, animasi, dan logika *klien-side* pada halaman web.

2.4.2 Python

Python adalah bahasa pemrograman yang dikenal dengan sintaksnya yang jelas dan mudah dibaca. Python populer dalam pengembangan web karena *framework* seperti Django dan Flask. Mereka mempermudah pembangunan aplikasi web yang cepat dan skalabel.

2.4.3 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa skrip *server-side* yang dirancang khusus untuk pengembangan web. PHP mudah ditanamkan dalam HTML. Ini membuatnya menjadi pilihan populer untuk banyak pengembang web guna membuat situs web dinamis.

2.4.4 Java

Java digunakan dalam berbagai aplikasi web dan *enterprise* karena portabilitasnya yang tinggi (tuliskan sekali, jalankan di mana saja). Selain itu, ini juga dikenal dengan *framework* yang kuat seperti Spring dan Hibernate.

2.4.5 Ruby

Ruby adalah bahasa pemrograman yang dikenal dengan sintaks yang mudah dibaca serta ditulis. Ruby on Rails (RoR) adalah *framework* yang paling terkenal dan banyak digunakan untuk pengembangan web.