

# Bangun aplikasi seluler dan web lebih cepat

Tiga kiat terbukti untuk mempercepat pengembangan aplikasi modern



Organisasi ingin berinovasi lebih cepat dengan aplikasi seluler dan web yang memuaskan pelanggan dan mendukung karyawan. Tim pengembangan harus memenuhi standar yang makin meningkat untuk pengalaman pengguna dan berbagai fitur **cloud** yang kompleks—dengan tetap menghadirkan aplikasi berperforma tinggi yang aman, dapat diskalakan, dan rendah biaya—ini bukanlah tugas yang mudah.

Saat membangun aplikasi seluler dan web modern, pastikan Anda membekali tim dengan kemampuan agar bisa dengan cepat menghadirkan aplikasi yang dapat diskalakan. Cari alat developer yang mampu menutup kesenjangan antara developer baru dan berpengalaman agar dapat meluncurkan aplikasi dengan cepat dan mengimbangi hal ini dengan berbagai opsi yang dapat diperluas serta bisa berintegrasi dengan proses internal.

Untuk makin mempercepat pengembangan, cari solusi untuk merampingkan akses ke data yang tersebar dan terdistribusi di seluruh tim agar tim **front-end** bisa membangun fitur-fitur yang didorong data dengan lebih cepat dan efisien. Tentukan pilihan arsitektural untuk mendukung aplikasi Anda saat ini dan ketika permintaan memuncak atau melandai ke depannya. Apa pun keputusan Anda, tugas utama Anda adalah memastikan performa dan keamanan aplikasi.

Dalam panduan ini, kami membagikan tiga kiat untuk mempercepat pengembangan aplikasi seluler dan web modern:

1. Alat dan Kerangka Kerja Modern – berdayakan tim **front-end** Anda dengan kerangka kerja dan alat yang fleksibel dan dibuat khusus
2. API Modern – sederhanakan akses ke sumber data yang terdistribusi dengan API GraphQL terpadu
3. Komputasi Modern – kurangi pengoperasian dengan layanan terkelola dan nirserver

## Kiat #1

# Berdayakan tim *front-end* Anda dengan kerangka kerja dan alat yang fleksibel dan memiliki tujuan khusus

### Situasi

Developer *front-end* menciptakan pengalaman pengguna yang berbeda untuk aplikasinya menggunakan bahasa pemrograman *front-end* seperti JavaScript, Swift, atau Kotlin, kerangka kerja populer seperti React, React Native, Flutter, atau Angular, dan berbagai alat seperti lingkungan pengembangan terintegrasi (IDE). Mereka membuat kode lapisan presentasi aplikasi, mengakses dan mengintegrasikan data melalui API, dan membangun fitur yang didukung *cloud* seperti video atau obrolan dengan kemampuan waktu nyata dan *offline* untuk aplikasi web atau seluler.

Developer *front-end* mungkin ahli dalam membangun antarmuka pengguna, tetapi umumnya mereka bukan ahli *cloud*. **Membangun dan menghubungkan fungsionalitas *cloud* ke dalam aplikasi seluler atau web bisa menjadi hal yang kompleks dan menyita waktu**, mengurangi waktu tim *front-end* yang seharusnya difokuskan untuk mengembangkan pengalaman pengguna.

### Solusinya

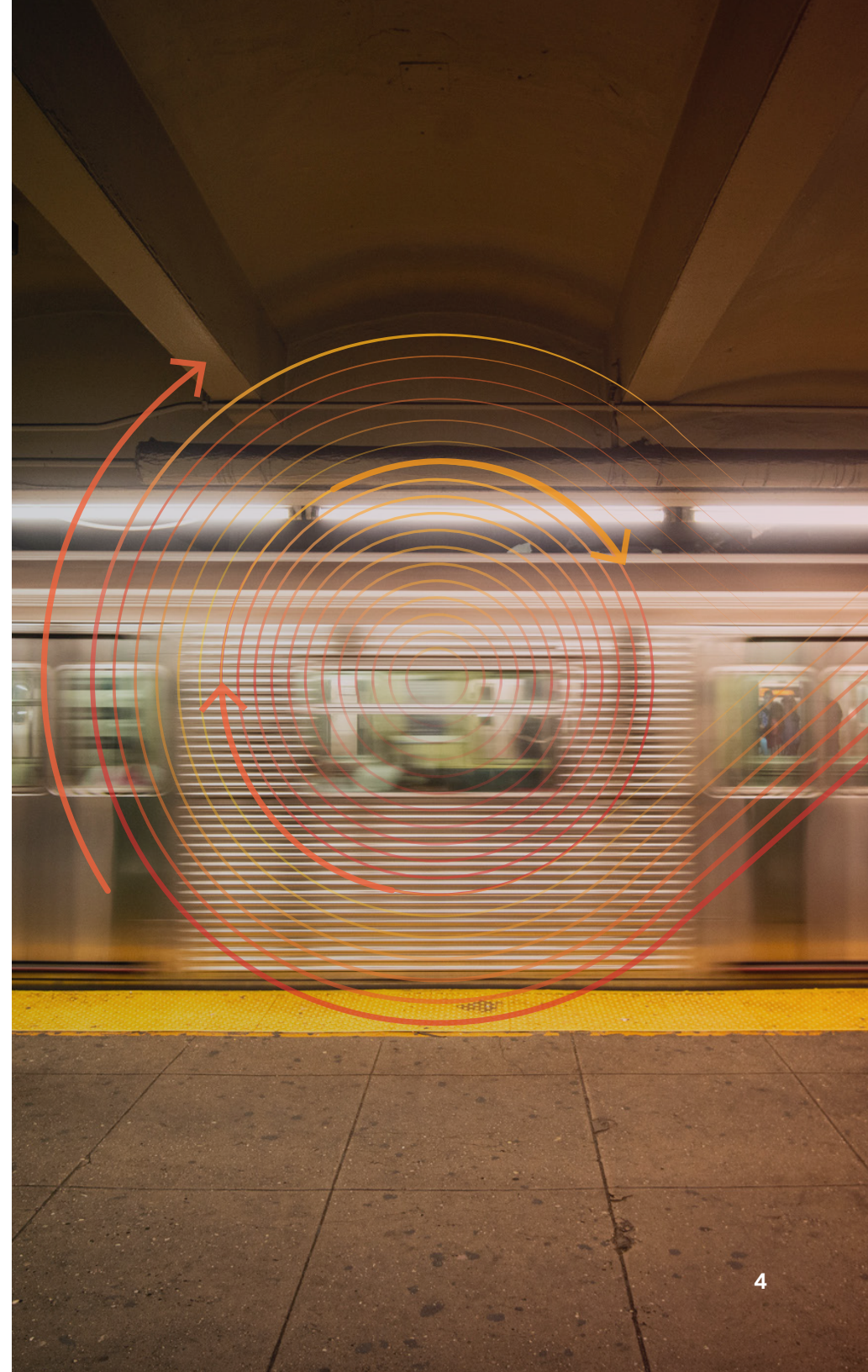
Bantu agar tim *front-end* Anda bisa menjadi lebih produktif dengan **alat pengembangan dan kerangka kerja yang mengurangi upaya penulisan kode yang diperlukan untuk menyelesaikan kasus penggunaan penting**. Manfaatkan kerangka kerja dengan kemampuan mendalam untuk tugas paling rumit seperti autentikasi dan akses data, dan carilah fitur-fitur tingkat lanjut untuk menangani *caching*, koneksi ulang, sinkronisasi data, dan resolusi konflik. Pilih alat yang menyederhanakan pengembangan—tanpa mengorbankan fleksibilitas untuk menyesuaikan atau memperluas fitur-fitur aplikasi dengan kode. Dengan memilih alat yang mengurangi kurva pembelajaran *cloud*, persiapan bisa menjadi lebih ringkas, alur kerja menjadi lebih cepat, dan keamanan yang otomatis—sehingga tim *front-end* bisa lebih fokus dalam mengatasi masalah yang dialami pelanggan Anda.

## Bagaimana AWS dapat membantu

**AWS Amplify** menawarkan berbagai alat dan layanan untuk membantu developer *front-end* membangun aplikasi web dan seluler *native* dengan *backend* bersama yang didukung oleh AWS. Pustaka Amplify menyederhanakan integrasi kasus penggunaan umum, seperti menyimpan gambar dan video di *cloud*, ke dalam aplikasi Anda dan mengintegrasikan secara mendalam dengan kerangka kerja, bahasa, dan platform seluler *front-end* populer. Selain itu, Amplify CLI dan Admin UI dapat membantu developer membuat *backend* AWS baru dengan cepat apabila diperlukan.

Tidak seperti penawaran *backend-as-a-service*, AWS Amplify memungkinkan developer untuk memanfaatkan keragaman alat dan layanan AWS seiring berkembangnya aplikasi mereka. Saat Anda membangun *backend* aplikasi Anda menggunakan Amplify CLI atau Admin UI, AWS Amplify secara otomatis membuat kode *deployment* dan logika bisnis dasar agar developer dapat memperluas aplikasi mereka dengan kode logika bisnis kustom dalam bahasa pemrograman pilihan mereka. Developer yang membangun aplikasi web juga dapat menggunakan layanan *hosting* web statis terkelola penuh di AWS Amplify untuk *deploy* aplikasi dalam hitungan menit, cukup dengan menghubungkan repositori kode mereka.

AWS Amplify membantu developer *front-end* membangun aplikasi seluler dan web lebih cepat. Dengan mengodekan praktik terbaik untuk skenario umum dan mengirimkannya dalam cara yang dogmatis, AWS Amplify mengurangi keseluruhan langkah dan baris kode yang diperlukan untuk membangun aplikasi tumpukan penuh di AWS—dan tetap memberikan fleksibilitas bagi developer untuk menyesuaikan aplikasi seiring berkembangnya persyaratan bisnis.



The logo for Neiman Marcus, featuring the brand name in a white, elegant script font against a dark, blurred background of what appears to be a person's hands or a similar texture.

*“Menggunakan arsitektur nirserver dengan AWS Amplify dan AWS AppSync, kami meningkatkan kecepatan masuk pasar sebesar setidaknya 50% dan kami mampu mempercepat peluncuran Connect.”*

– Sriram Vaidyanathan, Direktur Senior (*Senior Director*), Omni Engineering, Neiman Marcus

## Neiman Marcus meningkatkan kecepatan masuk pasar aplikasi sebesar 50 persen di AWS

### Tantangan:

Neiman Marcus adalah perusahaan mewah kenamaan yang dikenal menghadirkan pengalaman layanan pelanggan kelas satu dan personal. Untuk memodernisasi dan menyempurnakan pengalaman tersebut, perusahaan ini ingin mengembangkan Connect—sebuah aplikasi penjualan digital multialuran yang akan memberdayakan karyawan agar dapat melihat informasi pelanggan yang lebih beragam dan dipersonalisasi. Sasaran: Menciptakan interaksi yang membekas bagi setiap pelanggan.

Saat pandemi COVID-19 memaksa tutupnya toko-toko retail, penjualan perusahaan beralih ke **online** secara eksklusif. Neiman Marcus perlu meluncurkan Connect dengan cepat agar dapat memberikan pengalaman berkualitas tinggi yang selalu disukai pelanggan dan memungkinkan karyawan mereka untuk melakukan penjualan secara digital.

### Alat dan layanan yang digunakan:

- AWS Amplify
- AWS AppSync
- AWS Lambda

### Hasil:

- Kecepatan masuk pasar meningkat 50 persen untuk Connect
- Penggunaan arsitektur nirserver menghemat 90 persen biaya dibandingkan metode tradisional

[Baca studi kasus lengkapnya »](#)

## Kiat #2

# Sederhanakan akses ke sumber data yang terdistribusi dengan API GraphQL terpadu

### Situasi

Banyak organisasi yang melanjutkan peralihan dari aplikasi monolitik ke arsitektur yang lebih modular di *cloud*. Apabila aplikasi dibangun dengan komponen independen modular, yang disebut dengan layanan mikro, kecepatan rilis dapat meningkat karena developer dapat membuat perubahan ke setiap komponen dengan mudah.

Layanan mikro membuat aplikasi lebih mudah diskalakan dan lebih cepat dikembangkan, memungkinkan inovasi serta mempercepat waktu masuk pasar. Namun, ini juga berarti bahwa **tim front-end mungkin perlu terhubung ke ratusan sumber data dan layanan mikro terdistribusi**, yang makin mempersulit mereka untuk mengakses data yang diperlukan untuk aplikasi *user-facing* mereka. Dengan merampingkan akses data dengan cara yang memudahkan tim *front-end* untuk memakainya di aplikasi, tim *backend* memainkan peran penting dalam mempercepat pengembangan web dan seluler.

### Solusinya

Teknologi seperti GraphQL dapat meningkatkan kecepatan dan efisiensi akses data serta integrasi di lebih dari satu sumber data *backend*. GraphQL adalah bahasa kueri untuk API dan waktu aktif untuk memenuhi kueri tersebut dengan data Anda yang sudah ada. GraphQL menyediakan deskripsi yang lengkap dan mudah dipahami dari data di API Anda, yang memberikan kemampuan kepada developer *front-end* untuk secara tepat melakukan kueri data yang diperlukan pada tiap panggilan API dari sejumlah basis data dan layanan mikro *backend*.

**Tim pengembangan backend dapat menggunakan GraphQL untuk mengabstraksi sumber data yang ada ke satu titik akhir dan memberikan akses data yang lebih mudah bagi tim front-end.** Untuk sebuah kumpulan API yang sudah ada, abstraksi sejumlah titik akhir ke dalam satu API GraphQL memberikan keuntungan GraphQL—tidak perlu memigrasikan API atau menulis ulang aplikasi dari awal. Membuat lapisan API terpadu dengan GraphQL dapat melengkapi strategi REST API yang ada untuk membantu developer *front-end* bergerak lebih cepat dengan menyederhanakan interaksi mereka dengan arsitektur layanan terdistribusi.

## Bagaimana AWS dapat membantu

**AWS AppSync** adalah layanan terkelola penuh yang memudahkan pengembangan API GraphQL dengan menangani tugas berat, yaitu menghubungkan secara aman ke sumber data seperti Amazon DynamoDB, AWS Lambda, dan lainnya. Menambahkan **cache** untuk meningkatkan performa, berlangganan untuk mendukung pembaruan waktu nyata, dan penyimpanan data di sisi klien agar klien **offline** bisa mengetahui perkembangan terbaru. Setelah di-**deploy**, AWS AppSync secara otomatis menaikkan atau menurunkan skala pada mesin yang menjalankan API GraphQL Anda untuk memenuhi volume permintaan API.

Dengan API GraphQL yang didukung oleh AWS AppSync, organisasi bisa mengembangkan aplikasi lebih cepat dengan memberikan kemampuan kepada developer **front-end** untuk mengkueri lebih dari satu basis data, layanan mikro, dan API dari satu titik akhir GraphQL. AWS AppSync juga memudahkan pengembangan dan penskalaan API GraphQL. Dan AWS AppSync secara otomatis menyesuaikan kapasitas **throughput** sebagai respons terhadap pola lalu lintas yang sebenarnya.

AWS AppSync menawarkan integrasi dengan layanan AWS lainnya, yang memudahkan pengamanan, pemantauan, dan pemecahan masalah API. Layanan-layanan ini mencakup:

1. AWS WAF untuk melindungi API dari eksploitasi web umum
2. Amazon CloudWatch untuk metrik dan log
3. AWS X-Ray untuk pelacakan
4. AWS CloudTrail untuk log audit





PUBLIC GOOD

*“Dari konsep hingga produksi, kami hanya memerlukan waktu kurang dari satu bulan dengan satu orang rekayasawan dengan layanan AWS AppSync.”*

– Michael Manley, Direktur Teknologi (*Chief Technology Officer*),  
Public Good Software

## Public Good beralih dari konsep ke produksi dalam waktu kurang dari satu bulan, hanya dengan satu rekayasawan

### Tantangan:

Public Good Software adalah platform kenamaan dalam pemasaran berbasis prakarsa yang bertujuan untuk membuat berita lebih bisa ditindaklanjuti dengan mencocokkan artikel jurnalistik ke prakarsa amal yang relevan. AWS AppSync memungkinkan organisasi untuk beralih dari konsep ke produksi dalam waktu kurang dari satu bulan, dengan satu orang rekayasawan yang mengerjakan *backend* API. Karena lebih sedikit panggilan yang diperlukan dari klien ke *backend*, penggunaan AWS AppSync menurunkan latensi dalam pengiriman konten yang dapat ditindaklanjuti. Kombinasi AWS AppSync dan GraphQL menandakan bahwa data dapat diambil dan diintegrasikan di berbagai layanan mikro dalam sebuah permintaan jaringan.

### Alat dan layanan yang digunakan:

- AWS AppSync
- Amazon SageMaker

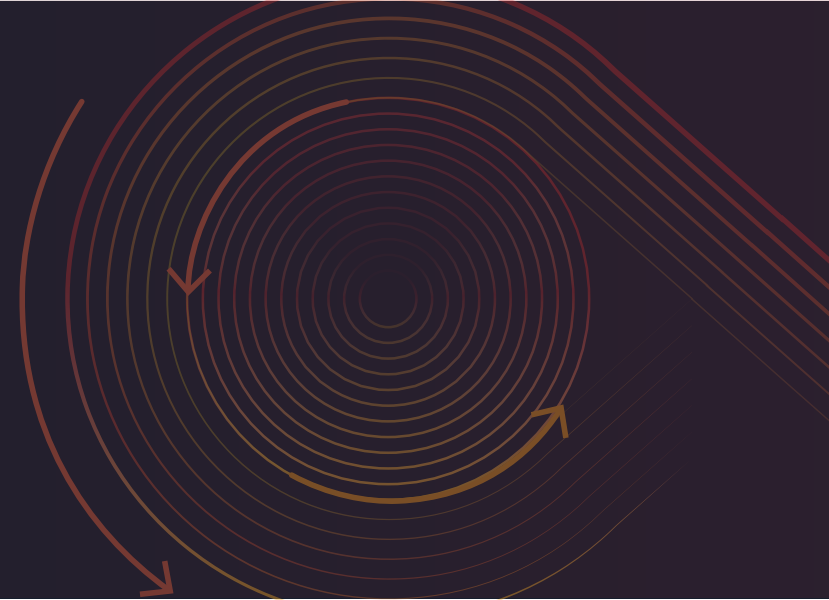
### Hasil:

- Menjalankan peningkatan API dan beralih ke produksi dalam waktu kurang dari satu bulan
- API yang diperbarui telah meningkatkan pengalaman pelanggan dan mempercepat waktu pemuatan halaman
- Penskalaan aplikasi dapat dilakukan dengan mudah untuk disesuaikan dengan naik turunnya lalu lintas situs

[Baca studi kasus lengkapnya »](#)

# Kiat #3

## Kurangi pengoperasian dengan layanan terkelola dan nirserver



### Situasi

Deloitte melaporkan bahwa 80 persen sumber daya IT fokus dalam menjalankan bisnis, sementara 20 persen fokus pada inovasi.<sup>1</sup> Cara tercepat untuk meningkatkan rasio tersebut dan lebih menitikberatkan pada inovasi adalah dengan mengurangi waktu yang dihabiskan pada aktivitas yang bukan merupakan hal inti dalam bisnis Anda—seperti mengelola infrastruktur.

**Mengelola server, penyimpanan, jaringan, dan sistem operasi adalah hal yang menyita waktu, mahal, dan menghambat ketangkasan karena adanya siklus pengadaan.** Mengelola infrastruktur akan menghabiskan lebih banyak waktu tim Anda, dan waktu tersebut sebenarnya bisa menghadirkan peluang besar untuk mengoptimalkan dan berinvestasi dalam inovasi.

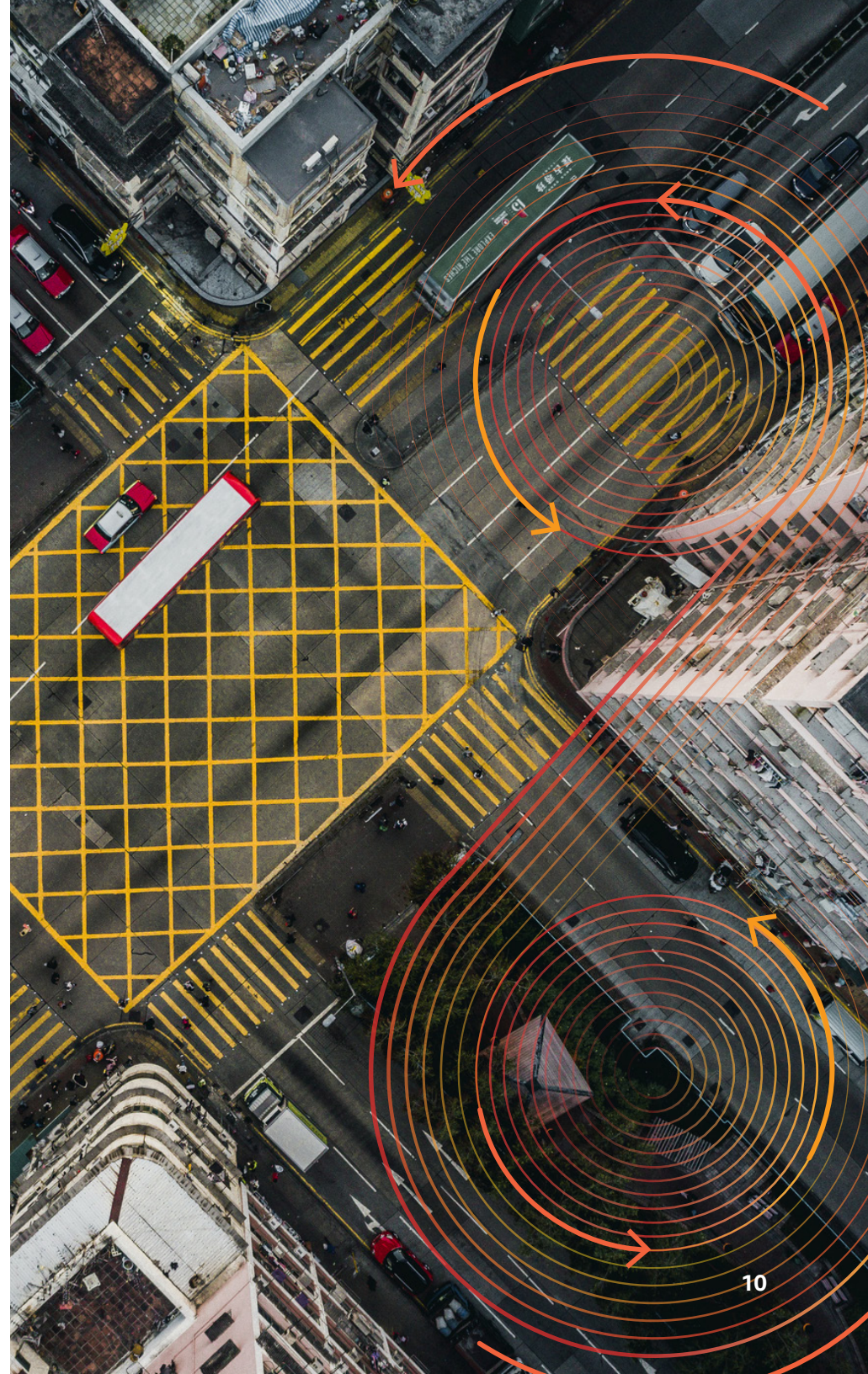
### Solusinya

Salah satu cara termudah untuk mengurangi pengoperasian adalah dengan mengalihkan beban operasional kepada penyedia **cloud** dengan mengadopsi layanan terkelola atau nirserver untuk infrastruktur Anda. Dengan mengurangi waktu untuk penyediaan, pengelolaan, dan pelaksanaan **patching** pada server, developer Anda dapat meluangkan lebih banyak waktu untuk berinovasi—menulis dan mengiterasi kode untuk produk yang memuaskan pelanggan Anda. **Dengan komputasi nirserver, Anda tidak perlu menyediakan atau mengelola infrastruktur.** Infrastruktur Anda diskalakan secara otomatis berdasarkan unit konsumsi—bukan unit server. Bayar komputasi yang Anda pakai saja; ketersediaan dan toleransi kesalahan sudah tertanam di dalamnya. Komputasi nirserver membuat Anda tidak terbebani pengoperasian server, sehingga Anda bisa fokus membangun aplikasi, bukan mengelola dan menskalakan infrastruktur untuk mendukung aplikasi tersebut.

## Bagaimana AWS dapat membantu

Anda bisa mengawalinya dengan [AWS Lambda](#), sebuah layanan komputasi nirserver yang didorong peristiwa.

Dengan AWS Lambda, Anda hanya perlu menulis kode untuk logika bisnis. AWS Lambda berorientasi pada peristiwa, yang artinya layanan ini akan tetap diam hingga ada peristiwa yang terpicu, dan sistem pun akan langsung bertindak. Bahkan, Anda dapat memicu fungsi AWS Lambda secara otomatis lewat lebih dari 200 layanan AWS dan aplikasi SaaS—tanpa perlu menulis kode integrasi. Setelah peristiwa terpicu, sumber daya pun aktif memenuhi kebutuhan tugas—dalam kecepatan milidetik. Anda tidak perlu merisaukan penyediaan atau konfigurasi infrastruktur, dan Anda hanya membayar sesuai penggunaan. Ini artinya produksi bisa dicapai dalam waktu lebih singkat dengan total biaya kepemilikan (TCO) serendah mungkin. AWS Lambda menawarkan ketangkasan maksimum dengan pengoperasian minimum, tetapi terkadang Anda mungkin mendapati bahwa penyesuaian pada infrastruktur bisa menambahkan nilai bagi bisnis Anda—di AWS, Anda selalu dapat melengkapi AWS Lambda dengan layanan kontainer di AWS, seperti Amazon ECS atau Amazon EKS, untuk memenuhi kebutuhan tersebut.





Coca-Cola

*“Yang tadinya merupakan arsitektur yang kompleks—dengan persyaratan keamanan, presisi, dan latensi yang tinggi—kini disederhanakan lewat layanan seperti AWS Lambda untuk menciptakan pengalaman yang sangat memuaskan pengguna.”*

– Michael Connor, Kepala Arsitek (*Chief Architect*), Coca-Cola Freestyle

## Coca-Cola membangun aplikasi untuk dispenser minuman tanpa sentuhan, hanya dalam waktu 100 hari

### Tantangan:

Ketika pandemi COVID-19 menerpa, Coca-Cola melihat adanya peluang untuk berinovasi. Perusahaan ini ingin menghadirkan pengalaman nirsentuh pada dispenser minuman Freestyle dengan membangun aplikasi ponsel cerdas yang akan memungkinkan pelanggan untuk memesan dan membayar minuman tanpa sentuhan fisik pada mesin jual otomatis.

Coca-Cola memilih untuk memanfaatkan fitur-fitur keamanan, latensi, dan skalabilitas bawaan dari AWS Lambda, memungkinkan developer mereka untuk fokus pada aplikasi itu sendiri. Hasilnya, Coca-Cola menyelesaikan pembangunan aplikasi tersebut hanya dalam 100 hari. Kini, lebih dari 30.000 mesin sudah dilengkapi dengan kemampuan nirsentuh ini.

### Alat dan layanan yang digunakan:

- AWS Lambda
- Amazon API Gateway
- Amazon CloudFront

### Hasil:

- Meluncurkan prototipe aplikasi seluler dalam satu minggu
- Menskalakan ke 10.000 mesin dalam 150 hari
- Memungkinkan perangkat seluler untuk tersambung ke dispenser dalam hitungan milidetik

[Baca studi kasus lengkapnya »](#)

## Memulai

Setelah Anda mengetahui tiga kiat untuk mempercepat pengembangan aplikasi seluler dan web modern, Anda mungkin ingin segera memulai. Kami menyarankan agar tim memulai dengan satu kiat ini atau lebih guna melengkapi pendekatan yang saat ini mereka gunakan untuk membangun aplikasi modern.

Berdayakan tim *front-end* Anda dengan AWS Amplify—mengurangi langkah secara keseluruhan dan baris kode yang diperlukan untuk membangun aplikasi tumpukan lengkap dapat membantu developer meluncurkan dengan lebih cepat. Sederhanakan akses ke sumber data yang terdistribusi dengan AWS AppSync—mengabstraksi sumber data yang ada ke dalam sebuah titik akhir API GraphQL terpadu akan memudahkan akses data bagi tim *front-end* untuk membangun fitur yang didorong data. Dan operasikan lebih sedikit infrastruktur menggunakan layanan seperti AWS Lambda—mengadopsi layanan terkelola atau nirserver untuk infrastruktur modern Anda memungkinkan tim Anda fokus membangun aplikasi.

Apa pun jalur yang Anda pilih, AWS akan membantu Anda memenuhi standar yang terus meningkat untuk pengalaman pengguna dan layanan *cloud*—memungkinkan tim Anda menghadirkan aplikasi berperforma tinggi dan dapat diskalakan, secara hemat biaya. Ketahui bagaimana Anda bisa membangun [aplikasi seluler dan web](#) yang kaya fitur, dengan cepat dan mudah. Atau langsung saja mulai dengan satu atau beberapa alat atau layanan kami sekarang.

[Mulai perjalanan modernisasi aplikasi Anda »](#)

[Terhubung dengan ahli AWS »](#)

# Sumber Daya Lainnya

[Mulai membangun aplikasi seluler dan web dengan AWS Amplify »](#)

[Ketahui lebih dalam apa itu modernisasi API dengan GraphQL »](#)

[Baca laporan 2021 Deloitte TCO tentang nirserver »](#)

[Dapatkan kiat praktis di Blog AWS AppSync GraphQL »](#)

[Telusuri berbagai artikel di Blog Front-End Web & Mobile »](#)

