

EDISI DIGITAL · BAHASA INDONESIA

Yang baru *kreator* teknis.

CUPLIKAN

Pendahuluan · Daftar isi lengkap · Bab 2

PENULIS

Hernán Capucci

Edisi Independen · 2026

Cuplikan ini berisi pendahuluan, daftar isi lengkap, dan satu bab representatif dari buku ini. Dengan begitu Anda bisa menilai nada, pendekatan, dan cara belajarnya sebelum mengambil edisi lengkap.

Apa yang ada dalam cuplikan ini

- Pendahuluan lengkap (Bab 0).
- Daftar isi lengkap buku ini.
- Bab 2 secara utuh: *Cara kerja internet*.
- Catatan tentang kesinambungan.

Daftar isi lengkap

Pendahuluan

- Bab 0 — Mengapa buku ini ada

Bagian I — Fondasi ekosistem digital

- Bab 1 — Semua berkarya, sedikit yang memahami
- Bab 2 — Cara kerja internet
- Bab 3 — Klien dan server
- Bab 4 — Apa itu basis data
- Bab 5 — Apa itu kode

Bagian II — Arsitektur sebuah aplikasi

- Bab 6 — Frontend: yang Anda lihat
- Bab 7 — Backend: yang tidak Anda lihat
- Bab 8 — API: bahasa yang dipakai aplikasi untuk saling bicara
- Bab 9 — Basis data: jenis, skema, dan keputusan
- Bab 10 — Autentikasi dan sesi

Bagian III — Perkakas dan proses

- Bab 11 — Git dan GitHub: kontrol versi
- Bab 12 — Terminal perintah

- Bab 13 — Lingkungan: pengembangan, staging, dan produksi
- Bab 14 — Deploy: mempublikasikan apa yang Anda bangun
- Bab 15 — Keamanan untuk orang nonteknis

Bagian IV — Bekerja dengan AI secara mempertimbangan

- Bab 16 — Apa yang bisa dan tidak bisa dilakukan AI
- Bab 17 — Cara meminta dengan baik kepada AI
- Bab 18 — Meninjau apa yang dihasilkan AI
- Bab 19 — AI dan alur kerja nyata

Bagian V — Membangun sesuatu yang nyata

- Bab 20 — Sebelum memulai: pertanyaan yang menentukan
- Bab 21 — Berbicara dengan developer tanpa tersesat
- Bab 22 — Siklus lengkap: dari gagasan ke produk

Bagian VI — Lampiran praktis

- Lampiran A — Checklist: cara meminta dengan baik kepada AI
- Lampiran B — Checklist: meninjau sebuah hasil kerja teknis
- Lampiran C — Perintah dasar untuk referensi
- Lampiran D — Template prompt yang bisa dipakai ulang
- Lampiran E — Sumber yang direkomendasikan

Glosarium teknis

- Glosarium praktis ekosistem digital, dijelaskan dengan bahasa yang jernih.

Penutup

- Epilog — Apa yang menanti

Bab 0 — Mengapa buku ini ada

Suatu hari Anda meminta AI menghasilkan sesuatu — kode untuk sebuah fitur, automasi yang menghubungkan dua layanan, sepenggal kode yang memecahkan masalah konkret. Hasil yang Anda terima tampak berfungsi. Anda memakainya. Dan di suatu titik dalam proses itu muncul sebuah pertanyaan yang mungkin tidak berani Anda ucapkan dengan lantang: bagaimana saya tahu kalau ini benar?

Pertanyaan itu bukan hal sepele. Itulah perbedaan antara bekerja dengan pertimbangan sendiri dan bekerja sambil berharap tidak ada yang menemukan kesalahannya.

Buku ini ada supaya Anda bisa menjawabnya.

Akses tanpa pemahaman

Beberapa tahun lalu, membangun **perangkat lunak** menuntut pelatihan teknis tertentu. Mempelajari bahasa pemrograman, memahami struktur data, menyiapkan lingkungan pengembangan, membaca dokumentasi yang mengasumsikan Anda sudah tahu banyak. Itu adalah penghalang yang tinggi dan nyata.

Lalu datang perkakas **kecerdasan buatan (AI) generatif**. Tiba-tiba, siapa pun bisa menulis instruksi dalam bahasa sehari-hari dan menerima kode yang berfungsi, automasi yang sudah terkonfigurasi, sistem yang terhubung ke layanan eksternal. Penghalang teknis turun secara signifikan.

Tetapi yang turun adalah penghalang teknis, bukan penghalang konseptual.

Sebuah perkakas yang menghasilkan kode untuk Anda tidak berarti Anda memahami apa yang dihasilkannya. Sebuah platform yang men-deploy aplikasi Anda dengan satu klik tidak berarti Anda memahami apa itu deploy, apa yang bisa gagal, atau cara mengembalikannya saat ada yang salah. AI yang menjelaskan sebuah error kepada Anda tidak berarti Anda tahu mengapa error itu terjadi atau bagaimana mencegahnya terulang.

Akses membaik. Pemahaman atas **sistem** yang ada di baliknya, dalam banyak kasus, tidak mengikuti irama yang sama.

Berkarya tidak sama dengan memahami

Hari ini lebih banyak orang membangun produk digital tanpa pelatihan teknis dibanding kapan pun sebelumnya. Aplikasi, automasi, integrasi antarlayanan, platform konten, proses internal yang didigitalkan.

Demokratisasi itu nyata dan berharga. Masalah muncul ketika ada yang tidak berjalan seperti yang diharapkan, atau ketika ada keputusan yang harus diambil tentang sistem yang sudah dibangun.

Seorang pelaku usaha yang memakai AI untuk membangun toko onlinenya tidak tahu mengapa “basis datanya tumbang”. Ia tidak tahu apakah data pelanggannya terlindungi. Ia tidak tahu apakah hasil kerja developer yang ia rekrut sudah dikerjakan dengan baik atau belum. Ia tidak tahu pertanyaan apa yang harus diajukan untuk mengetahuinya.

Seorang profesional yang mengotomasi proses kerjanya dengan AI tidak memahami persis apa yang dilakukan setiap bagian dari yang ia rangkai. Ia menerima hasilnya karena kemarin berfungsi. Ketika berhenti berfungsi, ia tidak punya perkakas untuk memahami harus mencari ke mana.

Seorang founder yang bekerja dengan tim teknis eksternal mengangguk ketika mereka berbicara tentang **API**, tentang **repositori**, atau tentang arsitektur sistem. Tetapi ia tidak paham. Dan ia tidak bertanya, karena ia tidak tahu apa yang harus ditanyakan.

Dalam semua kasus ini, masalahnya bukan kurangnya kecerdasan atau kemampuan. Masalahnya adalah kurangnya kosakata dan kerangka konseptual yang memungkinkan seseorang bekerja di ekosistem digital dengan pertimbangan sendiri.

AI sebagai titik berangkat

Kecerdasan buatanlah yang membuat buku ini mendesak. Itulah alarmnya.

Tetapi tujuan buku ini bukan membahas kecerdasan buatan. Tujuannya adalah membahas sistem yang ada di balik proyek digital apa pun, dengan AI atau tanpanya.

Sebuah API bukan konsep AI. Ia sudah ada sejak puluhan tahun lalu. Sebuah **repositori** kode juga bukan hal baru. Perbedaan antara lingkungan pengembangan dan lingkungan produksi bukan ciptaan model bahasa mana pun. Autentikasi pengguna, basis data relasional,

siklus hidup sebuah **deploy** — semua ini sudah ada sebelum AI generatif dan akan tetap ada sesudahnya.

Yang berubah dengan AI adalah kini hal-hal ini berada dalam jangkauan orang-orang yang dulu tidak menyentuhnya. Dan aksesibilitas tanpa pemahaman itu menciptakan ketergantungan baru.

Ketergantungan pada perangkat: kalau ia berubah, kalau ia gagal, kalau ia memberi hasil yang tak terduga, Anda tidak punya cara untuk menilainya. Ketergantungan pada developer yang memang paham: kalau ia berubah, kalau ia menaikkan tarifnya, kalau ia menghilang, Anda tidak bisa melanjutkan tanpanya. Ketergantungan pada orang yang mengaku tahu lebih banyak: tanpa kerangka sendiri, Anda tidak bisa membedakan pengetahuan nyata dari janji kosong.

AI membuka sebuah pintu. Buku ini memberi kosakata untuk memahami apa yang ada di dalamnya.

Kosakata yang hilang

Ada kekeliruan yang kerap terjadi dan layak diluruskan sejak awal.

Masalah yang dibahas buku ini bukanlah bahwa Anda tidak bisa membuat program. Tidak bisa membuat program bukanlah masalahnya. Sangat banyak orang yang sangat efektif di dunia digital tidak bisa membuat program dan tidak perlu mempelajarinya.

Yang penting bukan kemampuan menulis kode. Yang penting adalah kemampuan membacanya dengan cukup pertimbangan untuk memahami apa fungsinya. Yang penting adalah memahami apa itu **basis data** supaya bisa mengambil keputusan tentang cara menyimpan informasi. Tahu apa itu repositori supaya bisa meninjau hasil kerja yang diserahkan developer kepada Anda. Tahu apa itu **endpoint** supaya bisa meminta AI membangunnya dengan tepat. Tahu apa itu lingkungan produksi supaya tidak merusak sesuatu yang sedang live karena keliru.

Tak satu pun dari itu menuntut Anda menulis satu baris kode pun.

Bayangkan situasi konkret ini: developer berkata kepada Anda “migrasi di lingkungan staging gagal”. Tanpa konteks, kalimat itu hanya kebisingan. Dengan kosakata buku ini, Anda paham bahwa “migrasi” adalah perubahan pada struktur basis data, bahwa “staging” adalah lingkungan uji sebelum produksi, dan bahwa masalahnya kemungkinan belum berdampak

pada pengguna — tetapi harus diselesaikan sebelum melanjutkan. Perbedaan itu tidak menuntut kemampuan membuat program. Ia menuntut pemahaman konsep.

Yang memang dibutuhkan adalah kosakata teknis. Kosakata yang memungkinkan Anda mengajukan pertanyaan yang tepat, menafsirkan jawaban, menilai usulan, mendeteksi tanda peringatan, dan mengambil keputusan dengan pertimbangan. Kosakata itu tidak diperoleh dengan memakai perkakas. Ia diperoleh dengan memahami konsep di baliknya.

Itulah yang dibangun buku ini. Konsep demi konsep, dari nol, dengan contoh universal dan tanpa mengasumsikan pengetahuan awal apa pun.

Untuk siapa buku ini

Buku ini ditulis untuk orang-orang yang membangun hal-hal digital tanpa pelatihan teknis dan ingin melakukannya dengan pertimbangan yang lebih baik.

Itu mencakup mereka yang sedang merangkai produk digital pertamanya dengan bantuan AI dan tidak sepenuhnya memahami apa yang sedang ia bangun. Mereka yang bekerja dengan tim teknis eksternal dan ingin bisa berbicara dengan mereka tanpa setiap percakapan menjadi kerja penerjemahan. Mereka yang memakai perkakas automasi dalam pekerjaan sehari-hari dan ingin memahami apa yang sebenarnya sedang dilakukannya. Mereka yang punya ide dan perlu menilainya secara teknis sebelum mengorbankan waktu dan uang untuk mengembangkannya.

Industri atau bidang kerjanya tidak penting. Usia atau tingkat pendidikan formal di bidang teknologi juga tidak penting.

Yang penting adalah Anda memakai komputer dengan lancar dalam pekerjaan atau proyek Anda. Bahwa Anda pernah mencoba setidaknya satu perkakas AI. Dan bahwa Anda ingin lebih memahami **ekosistem digital** tempat Anda bekerja.

Titik berangkat Anda boleh jadi adalah belum pernah membuka satu pun repositori kode. Belum pernah menyiapkan lingkungan pengembangan. Belum jelas apa beda server dan cloud. Tak satu pun dari titik berangkat itu menjadi penghalang. Buku ini dimulai dari sana.

Apa yang tidak akan Anda temukan

Sebelum melanjutkan, ada baiknya berterus terang tentang apa yang tidak dilakukan buku ini.

Buku ini tidak mengajari Anda membuat program. Ini bukan manual bahasa pemrograman mana pun atau perkakas tertentu mana pun. Kalau itu yang Anda cari, ada sumber yang lebih baik dan lebih khusus untuk itu.

Buku ini tidak menyatakan bahwa AI bisa melakukan segalanya. AI tidak begitu. Ia punya kemampuan nyata dan batas yang nyata pula. Buku ini menunjukkan keduanya dengan jujur.

Buku ini tidak menjanjikan bahwa dalam tenggat tertentu proyek Anda siap. Proses teknis punya kerumitannya sendiri dan waktunya sendiri. Meremehkannya sama saja dengan berbohong.

Buku ini tidak bergantung pada perkakas, platform, atau layanan tertentu mana pun. Ketika sebuah perkakas konkret disebut, itu sekadar sebagai contoh. Konsep yang diajarkan berlaku sama dengan stack atau lingkungan apa pun yang Anda pakai.

Buku ini tidak membahas kisah sukses tertentu atau proyek nyata tertentu sebagai model untuk ditiru. Contoh-contohnya bersifat generik dan universal secara sengaja.

Buku ini tidak berkata “ini mudah” ketika kenyataannya tidak. Sebagian konsep dalam buku ini sederhana. Sebagian lain menuntut perhatian dan pembacaan ulang. Ketika sesuatu memang rumit, buku ini menandainya.

Apa yang akan Anda temukan

Setelah menyelesaikan buku ini, Anda akan punya pemahaman konkret atas konsep-konsep yang hari ini mungkin hanya Anda kenal namanya.

Anda akan memahami isi sebuah aplikasi dari dalam: bagian mana yang mengurus apa yang Anda lihat di layar, bagian mana yang memproses logika dan data, dan bagaimana kedua bagian itu saling berkomunikasi. Itu tidak akan membuat Anda bisa membangunnya sendiri, tetapi membuat Anda paham apa yang mereka bicarakan ketika menjelaskannya kepada Anda atau ketika ada yang gagal.

Anda akan bisa membuka sebuah repositori kode dan membaca strukturnya: folder apa saja yang ada, apa fungsi tiap bagian proyek, apa isi riwayat perubahannya. Informasi itu tersedia di repositori mana pun, dan hari ini mungkin Anda belum tahu bahwa Anda bisa memakainya untuk memahami apa yang diserahkan seseorang kepada Anda.

Anda akan bisa menulis **prompt** dengan konteks yang cukup supaya AI menghasilkan sesuatu yang berguna. Perbedaan antara instruksi yang kabur dan instruksi yang tepat bukanlah bakat: itu soal tahu informasi apa yang dibutuhkan perangkat agar bekerja dengan baik.

Anda akan bisa meninjau apa yang dihasilkan AI atau developer dengan pertanyaan yang konkret: apa persisnya fungsi bagian ini? Apa yang terjadi kalau kolom ini datang dalam keadaan kosong? Adakah bagian yang mengakses informasi sensitif? Pertanyaan-pertanyaan itu tidak menuntut kemampuan membuat program. Mereka menuntut pemahaman atas konsep yang terlibat.

Anda akan bisa memakai glosarium buku ini sebagai perangkat kerja: membuka sebuah definisi ketika muncul istilah yang asing, memahaminya, lalu kembali ke percakapan atau dokumen dengan kejernihan yang lebih besar dari sebelumnya.

Dan Anda akan bisa mengenali kapan sebuah jawaban — dari seseorang atau dari sebuah perangkat — tidak bisa dinilai dengan pengetahuan Anda saat ini. Itu tidak selalu berarti tahu jawaban yang benar. Itu berarti tahu kapan pertanyaan yang harus Anda ajukan lebih baik daripada yang sudah Anda ajukan.

Bagaimana buku ini disusun

Buku ini terdiri atas enam bagian, ditambah bab pendahuluan ini, lampiran praktis, dan glosarium teknis.

Bagian I mencakup fondasi: apa itu perangkat lunak, cara kerja internet, apa itu model klien-server, apa itu basis data, dan apa itu kode. Inilah konsep paling dasar dari ekosistem digital. Setiap bagian berikutnya menganggap konsep ini sudah dipahami.

Bagian II masuk ke arsitektur sebuah aplikasi: **frontend**, **backend**, API, basis data secara mendalam, autentikasi, dan sesi. Inilah struktur internal dari produk digital modern mana pun.

Bagian III mencakup perangkat dan proses yang membuat sebuah proyek berjalan di dunianya: kontrol versi dengan **Git**, terminal perintah, lingkungan pengembangan, deploy, dan keamanan dasar.

Bagian IV dikhususkan untuk bekerja dengan AI secara cerdas: apa yang bisa dilakukannya, di mana ia gagal secara sistematis, cara merumuskan prompt yang efektif, cara meninjau apa

yang dihasilkannya, dan cara memakai model, **agen**, serta automasi tanpa kehilangan kendali atas proses.

Bagian V merangkai semuanya dalam konteks membangun sesuatu yang konkret: cara bersiap sebelum mulai, cara berkomunikasi dengan tim teknis, cara berpindah dari sebuah ide ke produk yang berfungsi.

Bagian VI adalah lampiran praktis: checklist rujukan cepat, perintah dasar terminal dan Git, templat prompt yang bisa dipakai ulang, dan sumber untuk mendalami tiap bidang.

Di akhir buku ada **glosarium teknis** berisi lebih dari 200 istilah yang dijelaskan dalam bahasa yang jernih, disertai contoh dan tanpa jargon. Ini bukan lampiran hiasan. Ia adalah bagian integral dari buku ini.

Urutan bagiannya tidak sembarangan. Masing-masing dibangun di atas yang sebelumnya. Bagian tentang sistem dan perkakas memerlukan fondasi dari Bagian I. Bagian tentang AI dan pembangunan memerlukan arsitektur dan perkakas. Kalau Anda memilih membaca berurutan, itulah alasannya.

Cara memakainya

Ada dua cara membaca buku ini dan keduanya sah.

Cara pertama adalah membaca menerus, dari awal sampai akhir. Inilah yang disarankan untuk mereka yang punya sedikit atau belum punya kontak sama sekali dengan ekosistem digital. Perjalanannya membangun pemahaman secara bertumpuk: tiap bab menyiapkan lahan untuk bab berikutnya.

Cara kedua adalah seperti buku rujukan. Seseorang menyebut API kepada Anda dan Anda tidak paham apa itu: buka Bab 8. Anda akan melakukan deploy pertama dan tidak tahu apa konsekuensinya: buka Bab 14. Anda perlu memahami apa itu **otentikasi** sebelum rapat tentang keamanan: buka Bab 10. Tiap bab bisa dibaca secara mandiri, meski rujukan-silang menunjukkan apa yang sebaiknya sudah dibaca lebih dulu.

Glosarium bekerja dengan cara yang sama. Ketika sebuah istilah teknis muncul untuk pertama kali dalam satu bab, ia ditulis dengan **huruf tebal** disertai rujukan ke glosarium. Pada versi digital buku ini, rujukan itu adalah hipertaut aktif: Anda bisa langsung menuju definisinya dan kembali ke bab dengan satu klik. Pada versi cetak, rujukan itu mencantumkan nomor halaman.

Satu catatan tentang contoh-contoh: sepanjang buku Anda akan menemukan jenis konteks yang sama — sebuah toko online, sebuah aplikasi pengelolaan jadwal, sebuah platform kursus. Itu adalah situasi generik yang dipilih karena tidak menuntut pengetahuan awal tentang industri apa pun. Konsepnya berlaku persis sama pada jenis proyek apa pun yang lain.

Satu catatan tentang perkakas: ketika sebuah perkakas disebut sebagai contoh, ia memang hanya itu, sebuah contoh. Konsep yang diajarkan buku ini tidak bergantung pada platform tertentu mana pun. Kalau perkakas yang Anda pakai hari ini berganti besok, konsepnya tetap berlaku.

Berkarya dengan pertimbangan sendiri

Ada perbedaan antara orang yang memakai perkakas digital tanpa memahaminya dan orang yang memakainya sambil tahu apa yang sedang ia lakukan.

Perbedaan itu tidak terletak pada kemampuan membuat program. Ia terletak pada punya kosakata untuk mengajukan pertanyaan yang tepat. Kerangka konseptual untuk menilai jawaban. Kemampuan mendeteksi ketika ada yang salah meski Anda tidak tahu persis mengapa. Otonomi untuk mengambil keputusan tanpa sepenuhnya bergantung pada orang lain demi memahami apa yang sedang terjadi.

Itulah sosok yang dibangun buku ini: kreator teknis baru. Seseorang yang memakai AI, bekerja dengan tim teknis, membangun hal-hal digital — dan melakukannya dengan pertimbangan sendiri.

Ia bukan programmer. Ia tidak berpura-pura menjadi programmer. Tetapi ia juga tidak bekerja dengan mata tertutup, tidak menerima hasil tanpa bisa menilainya, dan tidak bergantung pada orang lain untuk menjelaskan segalanya kepadanya.

Ia memahami sistem tempat ia bekerja. Ia tahu apa yang harus ditanyakan, cara menilai apa yang ia terima, dan kapan harus mengakui bahwa ada yang tidak beres. Pemahaman itu tidak menuntut kemampuan menulis kode. Ia menuntut pemahaman tentang cara kerja apa yang Anda pakai.

Apa yang Anda pelajari di bab ini

- AI mendemokratisasi akses ke eksekusi teknis, tetapi bukan pemahaman atas ekosistem digital.

- Tidak bisa membuat program bukanlah masalahnya. Kurangnya kosakata teknis minimal lah masalahnya.
- Buku ini tidak mengajari Anda membuat program. Ia mengajarkan konsep yang memungkinkan Anda berkarya dengan pertimbangan, berbicara dengan tim teknis, dan meninjau apa yang dihasilkan AI.
- Buku ini terdiri atas enam bagian, lampiran praktis, dan glosarium berisi lebih dari 200 istilah. Bisa dibaca menerus atau dipakai sebagai rujukan.
- Pada versi digital, istilah-istilah glosarium adalah hipertaut aktif di tiap bab.

Apa yang berikutnya

Bab 1 dimulai dari awal: apa itu perangkat lunak dan mengapa penting memahaminya meski Anda tidak akan pernah menulisnya. Inilah blok pertama Bagian I dan fondasi dari semua yang datang sesudahnya.

Bab 2 — Cara kerja internet

Anda mengetik sebuah alamat di browser lalu menekan Enter. Dalam waktu kurang dari sedetik, sebuah halaman muncul: teks, gambar, data terbaru.

Dari mana semua itu datang? Bagaimana ia sampai ke layar Anda? Mengapa kadang ia tidak sampai?

Pertanyaan-pertanyaan itu tidak menuntut kemampuan memprogram untuk dijawab. Yang dituntut adalah memahami infrastruktur yang menghubungkan perangkat apa pun dengan server mana pun di dunia. Pemahaman itu adalah dasar untuk bisa mendiagnosis mengapa sesuatu berfungsi — dan mengapa kadang tidak.

Perjalanan sebuah permintaan

Ketika Anda mengetik sebuah alamat di browser, yang Anda tulis itu disebut **URL** — Uniform Resource Locator, atau pelokasi sumber daya seragam. Ini adalah alamat yang mengidentifikasi sebuah sumber daya spesifik di internet: sebuah halaman, sebuah gambar, sebuah berkas, sebuah data.

Browser perlu menemukan server yang menyimpan sumber daya itu. Namun server tidak diidentifikasi lewat nama yang terbaca manusia: ia diidentifikasi lewat angka. Setiap server

punya **alamat IP** — rangkaian angka yang menempatkannya secara unik di dalam jaringan, mirip cara sebuah alamat pos menempatkan sebuah gedung di dalam kota.

Masalahnya, Anda tidak mengetik angka. Anda mengetik `www.ensiklopedia.org` atau `berita.contoh.com`. Agar browser bisa menemukan server yang sesuai, ia perlu menerjemahkan nama itu menjadi sebuah alamat IP. Proses itu disebut resolusi **DNS** — Domain Name System, atau sistem nama domain.

DNS bekerja seperti buku telepon internet. Browser bertanya ke sebuah server DNS: “alamat IP apa yang dimiliki `berita.contoh.com`?” Server DNS menjawab dengan angkanya. Kini browser punya alamat yang sebenarnya.

Dengan alamat itu, browser mengirim sebuah permintaan ke server. Permintaan itu menempuh jaringan — lewat kabel, serat optik, dan koneksi nirkabel — sampai tiba di server yang menunggunya. Server menerimanya, memproses apa yang perlu diproses, lalu mengembalikan sebuah respons. Respons itu menempuh perjalanan balik sampai ke browser. Browser menafsirkan isinya dan menampilkannya di layar.

Seluruh proses itu — permintaan DNS, perjalanan pergi, pemrosesan, respons balik — terjadi dalam hitungan pecahan detik pada kondisi normal.

Ada satu optimalisasi yang layak disebut: **cache**. Browser tidak bertanya ke DNS setiap kali Anda mengunjungi situs yang sama. Ia menyimpan jawabannya — terjemahan nama ke IP — untuk sementara waktu. Sistem operasi melakukan hal serupa. Itu membuat kunjungan berulang jadi lebih cepat. Tetapi itu juga berarti jika server baru-baru ini berganti alamat IP, perangkat Anda mungkin masih memakai alamat lama. Mengosongkan cache browser atau menunggu sampai ia kedaluwarsa akan mengatasi masalah itu.

Ketika Anda mengetik sebuah alamat, browser tidak menuju ke awan ajaib: ia memulai sebuah rantai penerjemahan, permintaan, dan respons.

Apa itu domain

Sebuah **domain** adalah nama terbaca manusia yang mengidentifikasi sebuah situs atau layanan di internet. `ensiklopedia.org`, `berita.contoh.com`, `aplikasiku.io` adalah domain.

Domain tidak ada dengan sendirinya: seseorang harus mendaftarkannya. Ada perusahaan yang berspesialisasi mendaftarkan domain — disebut registrar — dan domain disewa untuk



GAMBAR 1. Ketika Anda mengetik sebuah alamat, browser tidak menuju ke awan ajaib: ia memulai sebuah rantai penerjemahan, permintaan, dan respons.

jangka waktu tertentu, umumnya satu tahun. Jika tidak diperpanjang, domain kedaluwarsa dan tersedia bagi orang lain untuk mendaftarkannya.

Bagian akhir dari domain — .com, .org, .io, .id — disebut domain tingkat atas atau TLD (Top Level Domain). Ia menunjukkan jenis organisasi atau negara asal. Perbedaan itu punya nilai historis dan posisi, tetapi secara teknis tidak mengubah cara kerja sistem.

Bagian yang bisa membingungkan adalah hubungan antara domain dan server. Sebuah domain bukanlah server: ia adalah label yang menunjuk ke server. Satu server yang sama bisa punya beberapa domain yang menunjuk kepadanya. Satu domain yang sama bisa mengarahkan ke server berbeda tergantung waktu dalam sehari atau wilayah dunia tempat ia diakses. DNS adalah sistem yang menjaga hubungan antara nama dan alamat itu tetap mutakhir.

Ketika sebuah domain tidak dikonfigurasi dengan benar — atau ketika ia kedaluwarsa — DNS tidak bisa meresolusi alamatnya. Browser tidak tahu harus menuju ke server yang mana. Halaman tidak termuat, walaupun server yang menyimpannya bekerja dengan sempurna.

Apa itu server — dan apa itu hosting

Sebuah **server** adalah komputer yang dirancang untuk menerima permintaan dan mengembalikan respons. Pada dasarnya, ia tidak berbeda dari komputer mana pun: ia punya prosesor, memori, penyimpanan, dan koneksi ke jaringan.

Perbedaan praktisnya adalah server selalu menyala, terhubung ke internet dengan ketersediaan tinggi, dan dikonfigurasi untuk melayani banyak permintaan secara bersamaan. Ia tidak punya layar maupun papan ketik. Ia tidak ada di meja siapa pun. Ia tinggal di pusat data —

fasilitas yang dirancang untuk menjaga banyak server tetap berjalan secara terus-menerus dan aman.

Hosting adalah layanan yang menyewakan ruang dan sumber daya pada server-server itu kepada Anda. Ketika Anda “memublikasikan” sesuatu di internet — sebuah halaman, sebuah aplikasi, sebuah berkas — yang Anda lakukan adalah menempatkan isi itu di server sebuah perusahaan hosting, mengonfigurasi sebuah domain agar menunjuk ke sana, lalu membiarkan server mengirimkannya kepada siapa pun yang memintanya.

Ada berbagai bentuk hosting. Pada bentuk paling dasar, Anda berbagi server dengan situs lain — lebih murah tetapi sumber dayanya terbatas. Pada bentuk yang lebih lanjut, Anda punya server virtual atau fisik yang eksklusif — lebih mahal tetapi dengan kontrol dan kapasitas yang lebih besar. Pada model cloud, sumber daya dialokasikan secara dinamis sesuai permintaan.

Bagi siapa pun yang membangun sesuatu yang digital, memahami apa itu hosting berarti memahami bahwa sistem yang Anda buat tinggal di suatu tempat fisik, bahwa tempat itu punya batas kapasitas, dan bahwa jika tempat itu gagal — atau jika domain tidak menunjuk dengan benar kepadanya — sistem itu tidak bisa diakses meski dibangun dengan sempurna.

Sebuah pembedaan yang berguna: server web adalah komponen yang menerima permintaan HTTP dan mengembalikan isi. Server aplikasi adalah yang menjalankan logika sistem — hal yang akan dibahas lebih detail di Bab 3. Pada banyak sistem sederhana, kedua fungsi itu berjalan di mesin yang sama. Pada sistem yang lebih besar, keduanya terpisah. Yang penting untuk saat ini adalah bahwa “server” bukanlah satu benda monolitik: ia adalah kumpulan bagian yang bekerja sama untuk menjawab permintaan.

HTTP dan HTTPS: bahasa web

Ketika browser mengirim sebuah permintaan dan server mengembalikan sebuah respons, keduanya perlu berbicara dalam bahasa yang sama. Bahasa itu disebut **protokol**. Sebuah protokol adalah seperangkat aturan yang mendefinisikan cara pesan diformat, dikirim, dan ditafsirkan.

Protokol web adalah **HTTP** — HyperText Transfer Protocol. Ia mendefinisikan struktur permintaan dan respons: informasi apa yang datang lebih dulu, cara jenis isi ditunjukkan, cara hasil operasi dikomunikasikan.

Sebuah permintaan HTTP dasar mencakup: metode (apa yang ingin Anda lakukan — mengambil informasi, mengirim data, menghapus sesuatu), alamat sumber daya, dan header yang berisi informasi tambahan tentang siapa yang bertanya dan jenis respons apa yang ia terima. Sebuah respons HTTP mencakup: kode status (berhasil atau tidak), header dengan informasi tentang isinya, dan badan — isi itu sendiri.

HTTPS adalah versi aman dari HTTP. Huruf “S” berasal dari “Secure”. Perbedaan teknisnya adalah pada HTTPS seluruh komunikasi menempuh perjalanan dalam keadaan terenkripsi: data yang dikirim browser ke server dan yang dikembalikan server tidak bisa dibaca oleh siapa pun yang menyadap koneksi di tengah jalan.

Enkripsi itu dimungkinkan oleh sebuah **sertifikat SSL/TLS** — sebuah berkas digital yang ditunjukkan server kepada browser untuk membuktikan bahwa ia memang seperti yang ia klaim, dan yang mengaktifkan enkripsi komunikasi. Browser modern menampilkan ikon gembok di bilah alamat ketika koneksi memakai HTTPS.

Mengapa ini penting bagi siapa pun yang membangun: sistem apa pun yang menangani data pengguna — kata sandi, informasi pribadi, pembayaran — harus memakai HTTPS tanpa kecuali. Tanpa HTTPS, data itu menempuh jaringan dalam bentuk teks polos dan bisa disadap. Dengan HTTP, browser bisa menampilkan peringatan “situs tidak aman” yang menjauhkan pengguna dan menurunkan kepercayaan terhadap sistem.

***Konsep kunci: protokol** Seperangkat aturan yang mendefinisikan cara dua pihak berkomunikasi. HTTP adalah protokol yang dipakai browser dan server untuk bertukar informasi di web. HTTPS adalah versi terenkripsinya.*

Apa yang dijawab server

Setiap kali sebuah server menerima sebuah permintaan, ia mengembalikan sebuah respons yang mencakup sebuah angka: **kode status**. Angka itu menunjukkan apa yang terjadi dengan permintaan tersebut.

Kode-kode itu dikelompokkan menurut kategori. Yang berawalan 2 berarti berhasil. Yang berawalan 3 adalah pengalihan. Yang berawalan 4 menunjukkan kesalahan di sisi klien — ada yang salah pada permintaan. Yang berawalan 5 menunjukkan kesalahan di sisi server — server menerima permintaan tetapi tidak bisa memprosesnya dengan benar.

Tiga yang paling penting bagi siapa pun yang bekerja dengan sistem:

200 — OK. Permintaan diproses dengan benar dan server mengembalikan isi yang diharapkan. Inilah hasil normalnya. Jika Anda melihat sebuah halaman, server menjawab 200.

404 — Not Found. Server menerima permintaan, tetapi sumber daya yang Anda minta tidak ada di server itu. Alamat yang Anda pakai tidak sesuai dengan berkas, halaman, atau data apa pun yang tersedia. Ini bukan kesalahan server: yang Anda cari memang tidak ada di sana.

500 — Internal Server Error. Server menerima permintaan, mencoba memprosesnya, lalu ada yang gagal dalam kerjanya sendiri. Masalahnya bukan pada apa yang Anda minta: melainkan pada cara server menangani permintaan itu. Ini adalah kesalahan pada kode atau pada konfigurasi server.

Kode	Nama	Apa yang terjadi	Contoh sehari-hari
200	OK	Server menemukan dan mengirimkan yang Anda minta.	Anda membuka sebuah halaman dan semuanya termuat tanpa masalah.
301 / 302	Pengalihan	Sumber daya ada di alamat lain. Browser membawa Anda ke sana secara otomatis.	Sebuah URL lama mengarahkan Anda ke domain baru situs tersebut.
400	Bad Request	Permintaan tiba dalam keadaan cacat; server tidak bisa menafsirkannya.	Anda mengirim sebuah formulir dengan data wajib yang belum lengkap.
401	Unauthorized	Anda tidak menunjukkan kredensial yang sah. Server tidak tahu siapa Anda.	Anda mencoba melihat isi privat tanpa login.

Kode	Nama	Apa yang terjadi	Contoh sehari-hari
403	Forbidden	Kredensial Anda sah, tetapi Anda tidak punya izin untuk itu.	Anda masuk ke akun Anda, tetapi bagian itu tidak sesuai dengan peran Anda.
404	Not Found	Yang Anda minta tidak ada di server itu.	Sebuah URL salah ketik atau sudah tidak ada.
500	Internal Server Error	Server menerima permintaan Anda tetapi gagal memprosesnya.	Anda mencoba menyelesaikan pembelian dan muncul “kesalahan tak terduga”.
503	Service Unavailable	Server tidak bisa melayani saat ini; ia kelebihan beban atau sedang dalam perawatan.	Sebuah situs tumbang saat penjualan besar atau peluncuran.

Anda tidak perlu menghafal semua kode. Yang penting adalah mengenali polanya: kode 2xx umumnya menunjukkan keberhasilan, 3xx pengalihan, 4xx masalah di sisi permintaan, dan 5xx masalah di sisi server.

Mengapa aplikasi “tumbang”

“Halamannya tidak termuat”, “sistemnya tumbang”, “tidak berfungsi” — ketika sesuatu berhenti tersedia, biasanya ada penyebab teknis yang spesifik. Memahami titik-titik kegagalan yang mungkin memungkinkan Anda mempersempit masalah sebelum memanggil seseorang untuk menyelesaikannya.

Penyebab yang paling umum, diurutkan menurut tempat terjadinya:

Server tumbang. Server berhenti menjawab. Bisa karena layanan hosting bermasalah, karena server dimulai ulang akibat pembaruan, atau karena sistem kehabisan sumber daya (memori, CPU) lalu berhenti berfungsi. Dalam kasus ini, browser tidak menerima respons jenis apa pun.

Domain tidak meresolusi. DNS tidak bisa menerjemahkan nama domain menjadi sebuah alamat IP. Bisa karena domain kedaluwarsa, karena konfigurasi DNS berubah secara keliru, atau karena ada masalah dengan server DNS. Hasilnya mirip dengan yang sebelumnya: browser tidak tahu harus menuju ke mana.

Sertifikat kedaluwarsa. Jika sertifikat HTTPS server kedaluwarsa, browser memblokir koneksi dan menampilkan peringatan keamanan. Halaman secara teknis ada dan server berfungsi, tetapi browser melindungi pengguna dengan menolak koneksi. Yang terlihat: sebuah layar peringatan, bukan halamannya.

Error 500. Server menjawab tetapi dengan kesalahan internal. Halaman ada, server menyala, tetapi kode menemui masalah saat memproses permintaan. Pengguna melihat sebuah pesan kesalahan alih-alih isi yang diharapkan.

Masalah jaringan. Koneksi antara perangkat pengguna dan server terputus di suatu titik perantara. Bisa koneksi internet pengguna, penyedia jaringan, atau sebuah titik di infrastruktur di antara kedua sisi.

Membedakannya secara permukaan membantu Anda tahu dari mana harus memulai:

- Jika halaman tidak memuat apa pun dan tidak ada peringatan → kemungkinan server tumbang atau DNS tidak meresolusi.
- Jika browser menampilkan peringatan keamanan yang jelas → sertifikat kedaluwarsa atau HTTP tanpa enkripsi.
- Jika halaman termuat dengan sebuah pesan kesalahan pada isinya → kemungkinan error 500, masalah pada kode.
- Jika hanya Anda yang tidak bisa mengakses sementara orang lain bisa → kemungkinan masalah jaringan atau cache lokal.

Mengetahui ini bukan berarti bisa menyelesaikannya secara langsung. Artinya, Anda bisa mengomunikasikannya dengan tepat kepada orang yang memang bisa melakukannya.

Pertanyaan berguna ketika sebuah web tidak termuat

Ketika sistem tidak tersedia, pertanyaan-pertanyaan ini membantu mempersempit masalah sebelum memanggil orang yang bisa menyelesaikannya:

- *Apakah masalah ini terjadi pada semua orang atau hanya pada saya?*

- *Apakah domainnya meresolusi? Apakah namanya menunjuk ke suatu alamat?*
- *Apakah servernya menjawab? Apakah ada sesuatu yang sampai, sekalipun itu sebuah kesalahan?*
- *Kode kesalahan apa yang ada — 404, 500 — atau memang tidak ada respons jenis apa pun?*
- *Apakah yang gagal itu produksi atau hanya lingkungan uji?*
- *Apakah ada log dari kesalahan itu?*
- *Apakah ada perubahan baru-baru ini — deploy, perubahan konfigurasi, perpanjangan domain?*

Ada satu kasus yang layak disebut karena sering terjadi dan membingungkan: sistem berfungsi untuk sebagian orang dan tidak untuk yang lain. Itu hampir tidak pernah berarti servernya tumbang. Biasanya itu menunjukkan masalah propagasi DNS — ketika sebuah perubahan dilakukan pada konfigurasi domain, perubahan itu butuh waktu antara beberapa menit hingga beberapa jam untuk sampai ke seluruh server DNS di dunia. Selama waktu itu, pengguna berbeda bertanya ke server DNS berbeda dan memperoleh jawaban berbeda. Sebagian melihat situs baru; sebagian lain masih melihat yang lama atau tidak melihat apa pun. Ini adalah perilaku yang wajar, bukan kesalahan yang harus diselesaikan secara mendesak.

Bagaimana Anda akan mendengarnya di sebuah rapat

Dalam rapat bersama tim teknis atau bersama vendor, dipakai ungkapan-ungkapan yang mengasumsikan semua orang paham apa yang sedang dibicarakan. Inilah yang paling umum dalam topik ini, apa artinya dalam praktik, dan apa yang bisa Anda tanyakan.

“Sepertinya ini masalah DNS.” Nama domain tidak meresolusi dengan benar — ia tidak bisa diterjemahkan ke alamat IP server. Bisa jadi konfigurasi yang keliru atau perubahan baru yang belum sepenuhnya terpropagasi.

Pertanyaan berguna: “Apakah ini terjadi pada semua pengguna atau hanya sebagian?” Jika pada semua, kemungkinan itu kesalahan konfigurasi. Jika hanya sebagian, kemungkinan besar itu propagasi yang sedang berjalan — sebuah proses yang butuh waktu berjam-jam dan selesai dengan sendirinya.

Yang sebaiknya tidak diasumsikan: bahwa servernya tumbang. DNS dan server adalah bagian yang berbeda. Server bisa berfungsi dengan sempurna sementara DNS gagal.

“Server menjawab 500.” Server menerima permintaan tetapi menemui kesalahan saat memprosesnya. Masalahnya ada pada kode atau pada konfigurasi sistem, bukan pada permintaan pengguna maupun pada jaringan.

Pertanyaan berguna: “Apakah ada log dari kesalahan itu?” Log adalah catatan tentang apa yang dilakukan sistem sebelum gagal. Sebuah 500 hampir selalu meninggalkan jejak yang mengidentifikasi penyebabnya. Tanpa log, diagnosis menjadi jauh lebih sulit.

Yang sebaiknya tidak diasumsikan: bahwa ini masalah internet atau perangkat pengguna. Sebuah 500 berarti server benar-benar menjawab — hanya saja dengan kesalahan internal.

“Sertifikatnya kedaluwarsa.” Sertifikat HTTPS server kedaluwarsa. Browser mendeteksi bahwa koneksi tidak lagi bisa dijamin aman lalu memblokir akses secara bawaan. Sistem bisa saja berfungsi sempurna di dalam, tetapi tidak ada pengguna yang bisa masuk.

Pertanyaan berguna: “Berapa lama perpanjangannya?” Pada kebanyakan kasus, ini proses yang cepat. Yang penting diketahui adalah perkiraan waktu penyelesaian dan apakah ada pengguna yang sudah terdampak yang perlu dikomunikasikan.

Yang sebaiknya tidak diasumsikan: bahwa ada masalah kode atau infrastruktur. Ini adalah kedaluwarsa administratif, mirip dengan kedaluwarsanya sebuah dokumen. Ia tidak menunjukkan apa pun tentang kualitas sistem.

“Produksi sedang tumbang.” Sistem yang dipakai pengguna sungguhan tidak tersedia. Ini bukan lingkungan uji maupun pengembangan: ini adalah yang memikul lalu lintas sungguhan dan data sungguhan. Ungkapan ini menunjukkan urgensi.

Pertanyaan berguna: “Apakah sudah ada yang menangani penyelesaiannya? Berapa perkiraan waktunya?” Dua pertanyaan itu memberi konteks yang dibutuhkan untuk tahu apakah perlu mengomunikasikan sesuatu kepada pengguna atau menunggu dalam diam.

Yang sebaiknya tidak diasumsikan: bahwa masalahnya berdampak sama pada semua orang. Kadang gangguannya parsial — hanya berdampak pada satu wilayah, satu perangkat, atau satu fungsi tertentu.

Yang Anda pelajari di bab ini

- Sebuah URL adalah alamat yang mengidentifikasi sebuah sumber daya di internet. Untuk sampai kepadanya, browser bertanya ke DNS, yang menerjemahkan nama domain menjadi sebuah alamat IP.
- Sebuah server adalah komputer yang selalu menyala yang menerima permintaan dan mengembalikan respons. Hosting adalah layanan yang menyewakan server itu.
- HTTP adalah protokol yang mendefinisikan cara browser dan server berkomunikasi. HTTPS adalah versi terenkripsinya, wajib bagi sistem apa pun yang menangani data sensitif.
- Kode status adalah bahasa yang dipakai server untuk menunjukkan apa yang terjadi: 200 adalah keberhasilan, 404 adalah sumber daya tidak ditemukan, 500 adalah kesalahan internal server.
- “Aplikasinya tumbang” punya penyebab teknis yang spesifik: server tumbang, DNS tidak meresolusi, sertifikat kedaluwarsa, kesalahan kode. Mengenalinya memungkinkan Anda mempersempit masalah.

Yang berikutnya

Bab 3 masuk ke dua pelaku dalam percakapan yang baru saja Anda pahami: klien dan server. Anda sudah tahu bagaimana informasi menempuh perjalanan di antara keduanya. Yang berikutnya adalah memahami apa yang dilakukan masing-masing dalam pertukaran itu — dan mengapa pembagian itu adalah struktur fundamental dari aplikasi apa pun.

Cuplikan ini berakhir di sini.

Buku lengkapnya berlanjut dengan peta utuhnya:
klien, server, basis data, kode,
API, keamanan, kecerdasan buatan, dan deployment.

Edisi lengkap mencakup
22 bab, lampiran praktis,
glosarium teknis, dan versi PDF + EPUB.

elnuevocreadortecnico.com