

Titled Artisans Prince

A Legacy by Gilang Teja Krishna

Daftar Isi

Preface	I
1 The Chronicles of Tech	2
2 The Chronicles of Tech	3
2.1 Era Fondasi (1940 – 1959)	3
2.2 Era Abstraksi (1960 – 1979)	4
2.3 Era Konsumerisme & GUI (1980 – 1999)	4
2.4 Era Konektivitas & AI (2000 – 2026)	5
3 The Artisan’s Choice	6
4 The Artisan’s Choice	7
4.1 Filosofi Pemilihan Stack	7
4.2 The Artisan Framework	8
4.3 Studi Kasus: Memilih Stack untuk 2026	8
5 Living the Tech	9
6 Living the Tech	10
6.1 Otomatisasi sebagai Gaya Hidup	10
6.2 The Artisan’s Cheatsheet	11
6.2.1 Development Tools	11

6.2.2	Productivity Commands	II
6.2.3	Mindset Ritual	II
6.3	Penutup: Meninggalkan Jejak	II

Preface

Buku ini bukan sekadar catatan teknis. Ini adalah jejak yang ingin saya tinggalkan—sebuah warisan bagi mereka yang ingin menempuh jalan sebagai seorang *Artisan*.

Saya menulis ini di tahun 2026, setelah tiga tahun menyelami dunia teknologi dengan cara saya sendiri. Meracik *stack*, membangun bahasa sendiri, dan mencari esensi dari setiap baris kode.

Gilang Teja Krishna
2026

Bab I

The Chronicles of Tech

Bab 2

The Chronicles of Tech

Sejarah teknologi bukan sekadar daftar penemuan. Ini adalah evolusi pemikiran manusia yang dituangkan ke dalam silikon dan listrik. Sebagai seorang *artisan*, saya melihat setiap era sebagai pergeseran cara kita berinteraksi dengan kemungkinan.

2.1 Era Fondasi (1940 – 1959)

Di sinilah segalanya dimulai. Ketika komputer masih sebesar ruangan dan diprogram menggunakan kabel fisik.

1941 Z3: Komputer digital pertama yang dapat diprogram sepenuhnya, lahir di Jerman oleh Konrad Zuse.

1945 ENIAC: Menjadi sorotan dunia sebagai mesin kalkulasi raksasa. Di tahun yang sama, John von Neumann menulis laporan tentang EDVAC, mendefinisikan arsitektur komputer yang kita gunakan hingga hari ini.

1947 Transistor: Revolusi sejati. Bell Labs mengganti tabung vakum yang panas dan rapuh dengan silikon yang efisien.

1953 COBOL & Grace Hopper: Munculnya bahasa yang lebih dekat dengan bahasa manusia, membawa teknologi keluar dari laboratorium murni.

2.2 Era Abstraksi (1960 – 1979)

Teknologi mulai mengecil dan menyebar. Ini adalah era di mana *Artisan* mulai mendapatkan alat yang lebih presisi.

1969 UNIX: Ken Thompson dan Dennis Ritchie melahirkan sistem operasi yang menjadi fondasi hampir semua hal modern hari ini.

1972 C Language: Sebuah alat musik yang sangat kuat. Fleksibel, cepat, dan menjadi bahasa ibu bagi banyak sistem operasi.

1975 Microsoft: Bill Gates dan Paul Allen mulai meracik masa depan perangkat lunak.

1976 Apple I: Wozniak merancang papan yang akan mengubah cara individu memiliki komputer.

2.3 Era Konsumerisme & GUI (1980 – 1999)

Komputer masuk ke rumah-rumah. Grafis menggantikan teks mentah.

1984 Macintosh: Memperkenalkan GUI kepada massa. Mouse menjadi tangan digital kita.

1991 Linux: Linus Torvalds merilis kernel yang akan menjalankan dunia, dari server hingga *embedded systems*.

1995 Java & JavaScript: Web mulai bernafas. Interaktivitas bukan lagi mimpi.

1998 Google: Cara kita mencari pengetahuan berubah selamanya.

2.4 Era Konektivitas & AI (2000 – 2026)

Kita sampai di masa sekarang. Masa di mana teknologi menjadi ekstensi dari diri kita sendiri.

2007 iPhone: Dunia di telapak tangan. Aplikasi menjadi ekosistem baru.

2015 TensorFlow: Google membuka gerbang kecerdasan buatan untuk semua orang.

2022 ChatGPT: Ledakan LLM. Komputer mulai "berbicara" dan "memahami".

2025-2026 The Agentic Era: AI bukan lagi sekadar *chatbox*. Ia adalah *multi-agent systems* yang bekerja secara otonom. Di tahun 2026 ini, saat saya menulis buku ini, kita melihat *coding* telah berevolusi dari sekadar mengetik sintaks menjadi kolaborasi tingkat tinggi dengan agen cerdas.

Bab 3

The Artisan's Choice

Bab 4

The Artisan's Choice

Dunia IT hari ini adalah sebuah rimba raya pilihan. Setiap minggu ada *framework* baru, setiap bulan ada bahasa baru yang mengklaim lebih cepat. Sebagai seorang *artisan*, tugas kita bukan untuk mempelajari semuanya, tapi memilih yang paling tepat.

4.1 Filosofi Pemilihan Stack

Saya memiliki tiga pilar utama saat harus memutuskan teknologi apa yang akan saya gunakan dalam sebuah proyek:

1. **Purpose (Tujuan):** Apakah teknologi ini menyelesaikan masalah utamanya, atau ia justru menambah kompleksitas baru? Jangan gunakan *microservices* jika *monolith* sudah lebih dari cukup.
2. **Craftsmanship (Kualitas Rancang Bangun):** Seberapa indah kodenya? Seberapa konsisten dokumentasinya? Teknologi yang baik adalah teknologi yang membuat pengembangnya merasa bangga saat menulisnya.
3. **Longevity (Keberlanjutan):** Apakah teknologi ini akan tetap ada dalam 5 tahun ke depan? Sebagai artisan, kita membangun untuk

jangka panjang.

4.2 The Artisan Framework

Saat menghadapi kasus spesifik, saya menggunakan matriks sederhana:

Kebutuhan	Prioritas	Rekomendasi Pendekatan
Eksperimen Cepat	Kecepatan Iterasi	Gunakan <i>batteries-included framework</i> seperti Django atau Next.js.
Sistem Kritis	Keamanan & Performa	Kembalilah ke bahasa yang lebih rendah seperti Rust atau C++.
Skalabilitas Tinggi	Konkuransi	Cari teknologi yang menangani <i>async</i> dengan elegan seperti Go atau Elixir.

4.3 Studi Kasus: Memilih Stack untuk 2026

Di tahun 2026, kita tidak lagi hanya memilih bahasa pemrograman. Kita memilih *ecosystem synergy*. Integrasi antara AI-native tools dan *runtime* yang hemat energi menjadi prioritas utama saya saat ini.

Bab 5

Living the Tech

Bab 6

Living the Tech

Teknologi bukan hanya untuk diselesaikan di kantor. Ini adalah bagian dari kehidupan sehari-hari. Dari otomatisasi rumah hingga efisiensi kerja. Bagi seorang *artisan*, teknologi adalah perpanjangan dari efisiensi berpikir kita.

6.1 Otomatisasi sebagai Gaya Hidup

Saya tidak suka mengerjakan hal yang sama dua kali. Jika sebuah tugas memakan waktu lebih dari 5 menit dan akan berulang besok, saya akan menulis skrip untuknya.

- **Dotfiles:** Seluruh lingkungan kerja saya adalah sebuah kode. Dengan satu perintah *git clone*, saya bisa bekerja di mesin mana pun dengan kenyamanan yang sama.
- **Local AI Agents:** Di tahun 2026, saya menjalankan model AI lokal untuk membantu menyortir email, menjadwalkan pertemuan, dan bahkan memberikan *code review* awal sebelum saya melakukan komit.

6.2 The Artisan's Cheatsheet

Berikut adalah rangkuman cepat alat dan filosofi yang saya gunakan setiap hari:

6.2.1 Development Tools

- **Editor:** Neovim atau VS Code dengan *custom configuration*.
- **Terminal:** Alacritty + Tmux untuk manajemen sesi.
- **Shell:** Zsh dengan *prok* untuk informasi visual yang cepat.

6.2.2 Productivity Commands

- `git standup`: Skrip *custom* untuk melihat apa yang saya kerjakan kemarin.
- `deploy-all`: Otomatisasi CI/CD dari terminal.

6.2.3 Mindset Ritual

- *Read Code Every Day*: Membaca kode orang lain sama pentingnya dengan menulis kode sendiri.
- *Delete More, Build Less*: Menghapus 100 baris kode yang redundan lebih memuaskan daripada menulis 100 baris kode baru.

6.3 Penutup: Meninggalkan Jejak

Gelar *Artisan* bukan diberikan oleh institusi, tapi oleh dedikasi pada kualitas. Buku ini adalah bukti perjalanan saya, dan saya harap ini menjadi kompas bagi kamu yang ingin menempuh jalan yang sama.