

20 Prompt AI Praktis

Untuk Kerja, Coding, Debugging, dan Belajar - Build With Rio

Cara pakai: ganti bagian dalam tanda [KURUNG] dengan konteksmu. Jangan masukkan password, API key, data pribadi sensitif, atau data klien rahasia ke layanan AI publik.

1. Ringkas Dokumen untuk Keputusan

Baca teks berikut. Jangan hanya merangkum. Pisahkan output menjadi: inti masalah, fakta penting, keputusan yang perlu dibuat, risiko, action item, penanggung jawab bila disebut, dan pertanyaan yang masih belum terjawab. Teks: [TEMPEL DOKUMEN].

2. Ubah Catatan Kasar Menjadi Email

Ubah catatan berikut menjadi email profesional. Audiens: [SIAPA]. Tujuan: [TUJUAN]. Tone: jelas, hangat, tidak bertele-tele. Pertahankan semua fakta, jangan menambah klaim baru. Catatan: [TEMPEL CATATAN].

3. Brainstorming dengan Kriteria

Berikan 10 ide untuk [TUJUAN]. Nilai masing-masing berdasarkan dampak, biaya, waktu implementasi, dan risiko. Pilih 3 terbaik dan jelaskan trade-off-nya.

4. Rencana Project dari Requirement

Ubah requirement berikut menjadi: tujuan, user flow, modul, data yang dibutuhkan, aturan bisnis, edge case, acceptance criteria, dan urutan implementasi. Requirement: [TEMPEL].

5. Prompt Coding yang Aman

Saya menggunakan [STACK + VERSI]. Buat perubahan hanya pada [FILE/FUNGSI]. Tujuan: [TUJUAN]. Jangan ubah layout/API lain. Gunakan prepared statement, validasi server-side, error handling, dan jelaskan asumsi sebelum memberi kode.

6. Debugging Berbasis Bukti

Bantu debug masalah ini. Tujuan: [TUJUAN]. Error lengkap: [ERROR]. Expected: [EXPECTED]. Actual: [ACTUAL]. Langkah reproduksi: [LANGKAH]. Environment: [VERSI]. Susun hipotesis dari paling mungkin, cara membuktikan masing-masing, lalu berikan perubahan sekecil mungkin.

7. Code Review

Review kode berikut untuk bug, security, performa, maintainability, dan edge case. Kelompokkan temuan: Critical, High, Medium, Improvement. Jangan mengubah kode sebelum menjelaskan masalah dan dampaknya. Kode: [TEMPEL].

8. Security Review

Audit endpoint/form berikut terhadap SQL Injection, XSS, CSRF, IDOR, upload abuse, auth/authz, session, data leakage, dan error handling. Untuk tiap temuan beri bukti, dampak, dan mitigasi.

9. Desain Database

Dari kebutuhan berikut, usulkan entitas, field penting, primary key, foreign key, unique constraint, index, dan alasan desain. Hindari over-normalization. Requirement: [TEMPEL].

10. Optimasi Query

Analisis query SQL berikut. Jelaskan bottleneck, index yang mungkin dibutuhkan, risiko full scan, dan cara mengukur perbaikan dengan EXPLAIN. Query: [TEMPEL].

11. Dokumentasi Fitur

Buat dokumentasi fitur ini dengan struktur: tujuan, prasyarat, konfigurasi, alur, endpoint/file terkait, cara test, error umum, troubleshooting, dan rollback. Fakta fitur: [TEMPEL].

12. SOP Deployment

Buat checklist deployment production untuk project [STACK]. Sertakan backup, database, config, permission, SSL, cache, cron, email, log, smoke test, dan rollback.

13. Belajar Konsep Sulit

Jelaskan [TOPIK] dalam tiga level: untuk orang awam, untuk junior developer, lalu untuk developer berpengalaman. Gunakan analogi hanya jika membantu dan tunjukkan miskonsepsi umum.

14. Tutor Interaktif

Jadilah tutor untuk [TOPIK]. Jangan langsung memberi semua jawaban. Ajarkan satu konsep kecil, beri satu latihan, tunggu jawaban saya, lalu koreksi dan lanjutkan.

15. Membandingkan Pilihan Teknologi

Bandingkan [A] vs [B] untuk konteks [KONTEKS]. Nilai dari biaya, learning curve, security, scalability, maintenance, ecosystem, dan risiko migrasi. Akhiri dengan rekomendasi bersyarat, bukan jawaban mutlak.

16. Membuat Test Case

Buat test case untuk fitur [FITUR]. Sertakan happy path, invalid input, boundary, unauthorized, forbidden, duplicate, concurrency bila relevan, dan failure dependency.

17. Refactor Tanpa Mengubah Perilaku

Refactor kode berikut untuk keterbacaan dan maintainability tanpa mengubah output/API. Jelaskan perubahan satu per satu dan berikan checklist regression test. Kode: [TEMPEL].

18. Analisis Error Log

Analisis log berikut. Kelompokkan symptom, kemungkinan root cause, bukti yang mendukung/menolak, pengecekan berikutnya, dan fix sementara vs fix permanen. Log: [TEMPEL].

19. Review Proposal Teknis

Kritik proposal teknis berikut sebagai reviewer senior. Cari requirement ambigu, risiko security, bottleneck, dependency, biaya tersembunyi, dan pertanyaan yang harus dijawab sebelum development.

20. Workflow AI Pribadi

Bantu saya membuat SOP penggunaan AI untuk pekerjaan [PERAN]. Petakan tugas berulang, mana yang aman dibantu AI, input yang dibutuhkan, template prompt, langkah verifikasi, dan data yang tidak boleh diberikan ke AI.