



Petunjuk Pro: Mastering Career Preparation for Website Developer 2025

Purpose

Petunjuk berupa ebook ditulis untuk membantu website developer tingkat pemula dan menengah dalam mempersiapkan karir masa depan mereka untuk mendapatkan hasil yang lebih baik. Beberapa poin pentingnya sebagai berikut:

- Mempersiapkan **portfolio yang lebih menjual** (di atas standard)
- Mempersiapkan resume (CV) **yang mudah dipahami** dan professional

- Memberikan gambaran bekerja di sebuah perusahaan agar lebih berpengalaman
- Mempelajari tata cara **menghadapi proses interview** dengan professional
- Mempelajari beberapa contoh dari **take home coding test** agar lebih terbiasa
- Mempelajari tata cara yang tepat untuk **nego gaji** di sebuah perusahaan
- Membangun **sumber passive income** lainnya sebagai website developer
- Memanfaatkan **AI dalam pekerjaan** sebagai web developer

Who is it for?

- Fresh graduate yang belum punya strategy terbaru untuk melamar kerja
- Seseorang yang sedang switch career menjadi website developer
- Seorang pemula website developer yang butuh arahan persiapan kerja
- Seorang website developer yang ingin belajar memperbanyak sumber income

- Seseorang yang sedang mencari kerja remote di bidang web development

It will improve your career

Sebelum membaca petunjuk pro	Setelah menerapkan petunjuk pro
Bikin portfolio asal jadi dan fokus kepada kuantitas dibandingkan kualitas	Memiliki portfolio sesuai kebutuhan market dan berkualitas di atas rata-rata
Sering apply loker tapi jarang mendapatkan balasan dari perusahaan	Melamar pekerjaan secara professional dan berkesempatan besar mendapat balasan
Kesulitan adaptasi pada lingkungan kerja sehingga performace menurun	Memahami alur bekerja website developer yang professional sehingga performace tinggi
Cepat menyerah mengerjakan coding test karena belum paham strateginya	Dapat mengerjakan take home coding test dengan percaya diri dan hasilnya lebih baik
Tidak percaya diri ketika mempresentasikan diri pada tahap awal interview	Percaya diri memperkenalkan diri, skills, dan pengalaman kepada recruiter
Hanya memiliki satu sumber income sebagai pekerja kantoran web developer	Memiliki lebih banyak sumber income sebagai website developer (jual digital product)

Kesulitan negosiasi gaji demi
mendapatkan bayaran di atas rata-rata

Lebih paham tata cara yang berhasil
untuk mendapatkan gaji lebih besar

Purpose

Who is it for?

It will improve your career

Perkenalan

Tujuan petunjuk pro dibangun

Warming Up

Chapter 1: Mempersiapkan diri sebagai Website Developer

Apa itu website developer?

Apa saja yang dilakukan oleh seorang web developer?

Apa saja tugasnya?

Saya harus pilih yang mana?

Dimana saya bisa belajar?

Langkah selanjutnya

Chapter 2: Portfolio hebat pertamamu

Portfolio itu bisa kita dapat dari mana aja sih mas?

Portfolio yang bagus itu biasanya seperti apa?

Tugas kamu selanjutnya

Membuat portfolio website developer pertama kita

Meningkatkan value pada portfolio pertama kita

Portfolio kita bisa dijadikan sumber income juga

Memperlihatkan hasil karya portfolio yang kamu miliki

Summary

Glossarium

Chapter 3: Mengenal workflow pada dunia kerja

Memahami flow projek

Adaptasi dengan workflow perusahaan

Memahami dokumentasi projek

Glossarium

Chapter 4: Membangun reputasi dan pengalaman

Membangun produk yang bermanfaat

Mengikuti magang (paid or non-paid)

Memilih program magang yang tepat

Soft skills penting ketika magang

Mengikuti program volunteer (sukarela)

Glossarium

Chapter 5: Menyusun professional curriculum vitae

Chapter 6: Mencari Kerja Remote dan Melamar Pekerjaan Pertama

Menyusun Strategi untuk Meningkatkan Peluang Diterima

Dimana Bisa Mencari Pekerjaan Remote?

Apakah Perlu Menggunakan Bahasa Inggris?

Negara yang Direkomendasikan untuk Pekerjaan Remote

Cara Melamar Pekerjaan Melalui LinkedIn, Kalibrr, dan Situs Lainnya

Menghindari Penipuan dalam Lowongan Pekerjaan

Kesimpulan

Chapter 7: Menguasai strategy untuk interview

Pemanasan diri dengan cara latihan coding

Etika yang baik ketika menjalani interview

Chapter 8: Mengerjakan take home coding test

Mempelajari instruksi dari coding test yang diberikan

Mempelajari solution dari case yang sudah pernah dibuat oleh orang lain

Memulai untuk mengerjakan coding test dengan waktu yang diberikan

Menulis code dengan mengikuti best practice (clean code, DRY, dll)

Memanfaatkan AI untuk berkonsultasi dan memvalidasi pekerjaan yang dibuat

Cara saya mengerjakan dan menyelesaikan coding test

Glossarium

Chapter 8 Bonus: Contoh Soal Coding Test dan Solusi Jawabannya

Grow Up Number:

Character Mapping

Anagram

Character Count

Get Days

Chapter 9: Mempresentasikan hasil coding test

Mempersiapkan solusi yang sudah dibuat sebelumnya

Buat penjelasan yang singkat, padat, dan jelas dalam bentuk catatan singkat

Memastikan bahwa coding test yang dikerjakan memenuhi test case yang sudah di tentukan sebelumnya

Memberikan alternatif solusi yang dibuat

Chapter 10: Menerima pekerjaan sebagai website developer

Beradaptasi dengan tim yang baru

Luangkan waktu untuk interaksi

Mempelajari dokumentasi resmi perusahaan

Responsibilities semakin besar

Chapter 11: Beradaptasi dan Persiapan Naik Jabatan

Mengapa Kenaikan Jabatan Itu Penting?

Langkah-langkah Mempersiapkan Diri untuk Kenaikan Jabatan

1. Evaluasi Diri Sendiri

2. Kumpulkan Bukti Kontribusi dan Prestasi

3. Diskusikan dengan HR atau Atasan

4. Bersikap Sabar dan Terbuka terhadap Evaluasi

5. Siapkan Alternatif Jika Permintaan Ditolak

Kesimpulan

Chapter 12: Strategi untuk Nego Ulang Gaji Kita

Mengapa Kita Layak Mendapatkan Kenaikan Gaji?

Strategi Efektif untuk Menegosiasikan Kenaikan Gaji

1. Kumpulkan Data dan Bukti Kontribusi

2. Pilih Waktu yang Tepat untuk Bernegosiasi
3. Ajukan Permintaan dengan Pendekatan Profesional
4. Tunjukkan Kesiapan untuk Terus Berkembang
5. Siapkan Alternatif Jika Permintaan Ditolak

Kesimpulan

Chapter 13: Membangun Sumber Passive Income sebagai Website Developer

1. Menjual Produk Digital
2. Membangun Layanan SaaS (Software as a Service)
3. Menyediakan Layanan Konsultasi dan Edukasi
4. Monetisasi Blog atau Website

Chapter 14: Giving Back to the Community

Mengapa Harus Berbagi dengan Komunitas?

Membangun Project Open-Source

Berbagi Tips Melalui Artikel

Chapter 15: Memanfaatkan AI dalam Pekerjaan sebagai Web Developer

AI Tools yang Bisa Bantu Web Developer

1. ChatGPT
2. GitHub Copilot
3. Claude

Tren terbaru di dunia AI Coding, Vibe Coding

Cursor (cursor.com)

Lovable (lovable.dev)

Kesimpulan

Bonus Chapter 15: Contoh Prompt Untuk Web Developer

Tips Memakai Prompt

Pecah jadi beberapa langkah

Berikan contoh

Jelaskan dengan spesifik

Minta AI untuk merefleksi jawabannya

Dokumentasi / Penjelasan

Buat Dokumentasi

Menulis Syarat dan Ketentuan

Membuat Cheat Sheet

Membuat File README

Menulis Blog Mendetail

Menjelaskan Kode

Diagram Arsitektur (Mermaid)

Entity Relationship Diagram (Mermaid)

Refactor Kode

Refactor Kode

Modernisasi Kode Lama

Membagi Kode ke dalam Beberapa Metode

Meningkatkan Performance

Menambahkan Parameter ke Function

Mengikuti Panduan Code Style / Standar Code

Cek Bug dan Error

Debug Coding di Component

Membuat Unit Test

Menulis RegEx

What's next?

Penutup dari mentor

Perkenalan

Halo people with the spirit of learning ✨, perkenalkan, nama Saya **Galih Pratama**, mentor dari Petunjuk Pro ini.

Saat ini, Saya bekerja secara remote **sebagai Front-end Developer** di perusahaan internasional di **Germany**, yaitu **2Viz**, yang merupakan perusahaan agency yang bekerja sama dengan beberapa perusahaan besar seperti **Disney**, **Royal Caribbean** dan perusahaan-perusahaan besar lainnya.

Saya juga sebelumnya merupakan **COO, Co-Founder di BuildWith Angga** dari tahun 2019 s/d 2023. Selain itu, Saya berkontribusi sebagai Full-stack Coding mentor sampai saat ini di BuildWith Angga.

Selain itu, Saya juga aktif mengerjakan project web development klien Indonesia, seperti pada Event The 90s Festival, Everblast Festival, Arctic Monkeys in Jakarta, Avenged Sevenfold in Jakarta dan project-project lainnya yang cukup terkenal dengan ratusan ribu pengguna.

Keseharian Saya sebagai software engineer tentunya tidak terlepas dari belajar dan improving skill setiap harinya, baik di pekerjaan maupun di luar pekerjaan. Sebagai software engineer, kita harus selalu mengasah skill kita, mulai dari membaca, menonton video, bahkan bereksperimen (ngulik).

Kegiatan Saya lainnya diluar bekerja, biasanya saya bermain game dengan genre simulation dan rhythm game, seperti Cities: Skylines,

Ayodance, osu! dan Pump It Up. Selain itu, Saya biasanya menghabiskan waktu quality time bersama keluarga dan anak di waktu senggang Saya.

Beberapa pengalaman terbesar yang Saya alami sejauh ini sebagai website developer adalah sebagai berikut:

- Membangun website yang punya security baik
- Membangun website yang selalu stabil dari jutaan visitor
- Membangun website yang SEO friendly dan performance baik
- Membangun website yang sesuai dibutuhkan oleh perusahaan

Dari beberapa pengalaman tersebut maka Saya bisa tumbuh lebih cepat karena hal-hal itu hanya dapat kita dapatkan apabila bekerja mengerjakan berbagai macam projek website, dan kamu akan mencapai fase itu juga nantinya.

Pada petunjuk pro ini Saya berharap bisa membagikan banyak ilmu terbaru di dalam dunia kerja terutama bidang website development yang cepat berubah-ubah karena teknologi dan update terbaru.

Saya ingin memberikan arahan kepada kamu tentang apa saja hal penting yang wajib dipelajari, selanjutnya kamu perlu membuat apa dan latihan apa saja untuk menghadapi interview agar hasilnya lebih maksimal.

Tujuan petunjuk pro dibangun

Setelah saya bersama dengan tim telah melakukan interview kepada fresh graduate atau kandidat lainnya, masih **banyak dari mereka yang belum memenuhi kriteria** seperti apa yang dibutuhkan oleh perusahaan teknologi saat ini.

Sebagian dari mereka sudah memenuhi kriteria tersebut hanya saja ketika menjalani tahap proses interview bersama dengan HRD, **mereka tidak begitu percaya diri** dalam menjelaskan keahliannya masing-masing sebagai seorang website developer.

Selain itu juga masih banyak sekali website developer pemula yang **bingung harus melakukan apa saja dalam mempersiapkan karirnya** sehingga ketika ada kesempatan bekerja mereka lebih siap dibandingkan dengan kandidat lainnya dan kesempatan diterima kerja oleh perusahaan tersebut menjadi lebih besar.

Oleh karena itu, Saya Galih Pratama bersama dengan co-author Saya yaitu Angga (founder BuildWithAngga) akan membagikan ilmu penting apa saja melalui petunjuk pro kali ini sehingga kamu bisa mempersiapkan beberapa hal berikut:

- portfolio web developer terbaik di atas rata-rata

- memiliki value sebagai seorang professional (siap bekerja)
- jago dalam menghadapi interview dan nego gaji
- mendapatkan pekerjaan remote (work from home)
- dan hal baik lainnya sebagai web developer



Saya sarankan untuk tidak melongkap-longkap sehingga hasil yang didapatkan menjadi lebih maksimal, setelah membaca satu chapter baiknya dipraktikkan terlebih dahulu sebelum lanjut kepada chapter selanjutnya.

Warming Up

Saya percaya bahwa saat ini kamu sudah mempelajari dan memperkuat fundamentals utama apabila ingin bekerja sebagai website developer. Kamu setidaknya sudah mempelajari beberapa hal berikut terkait technical:

- HTML, CSS, Vanilla JavaScript, dan PHP
- Framework JavaScript populer seperti React/Vue
- Framework PHP seperti Laravel
- Deploy project ke shared-hosting atau VPS

Mungkin sebagian besar dari kamu juga sudah membuat dan memiliki sebuah proyek kecil-kecilan untuk mengasah beberapa ilmu yang dibutuhkan oleh seorang website developer seperti di atas contohnya.



Saya sarankan kamu untuk mempelajari beberapa hal tersebut di atas untuk mendapatkan hasil lebih maksimal dari Petunjuk Pro ini.

Chapter 1: Mempersiapkan diri sebagai Website Developer

Apa itu website developer?

Pada sejatinya, website developer adalah seseorang yang membuat atau membangun sebuah website (atau dalam bahasa indonesianya adalah situs), yang dapat diakses melalui internet. Website sendiri terbagi ke dalam banyak kategori, mulai dari website apps, landing page, company profile, dan lain - lain.

Website Developer pada umumnya dapat dibagi kepada beberapa bagian, antara lain adalah:

1. Front-end web Developer

2. Back-end web Developer
3. Full-stack web Developer

Apa saja yang dilakukan oleh seorang web developer?

Web developer terbagi ke dalam beberapa fokus. Namun secara umum, ada beberapa hal yang dilakukan oleh developer, antara lain:

1. Membuat dan Mengembangkan Website

Sebagai seorang web developer, tugas paling utamanya adalah menulis code untuk membangun website. Penggunaan berbagai tools dan bahasa pemrograman diperlukan untuk menciptakan website yang baik. Selain itu, saat mendvelop website, web developer perlu code yang dapat digunakan ulang (reusable) dan mudah dipahami serta dimodifikasi

2. Melakukan Pengujian Website

Sebagai seorang web developer, rutin memeriksa website untuk mendeteksi kesalahan merupakan bagian dari tugasnya. Pemeriksaan ini penting sebelum website dilaunch dan dipakai. Web developer perlu mereview kode untuk memastikan tidak adanya bug yang dapat menyebabkan masalah di kemudian hari

3. Memelihara dan Menjamin Fungsionalitas Website

Web developer bertanggung jawab untuk memastikan semua fungsi website beroperasi dengan cepat. Waktu loading website yang lama

dapat menyebabkan pengunjung beralih ke website lain. Jika terjadi masalah dalam fungsionalitas website, web developer harus segera mendeteksi, memperbaiki, dan memperbarui.

Apa saja tugasnya?

Secara umum, tugas dari web developer dibagi ke dalam (namun tidak terbatas pada) dua bagian, yaitu Front-end Developer dan Back-end Developer. Berikut penjelasannya:

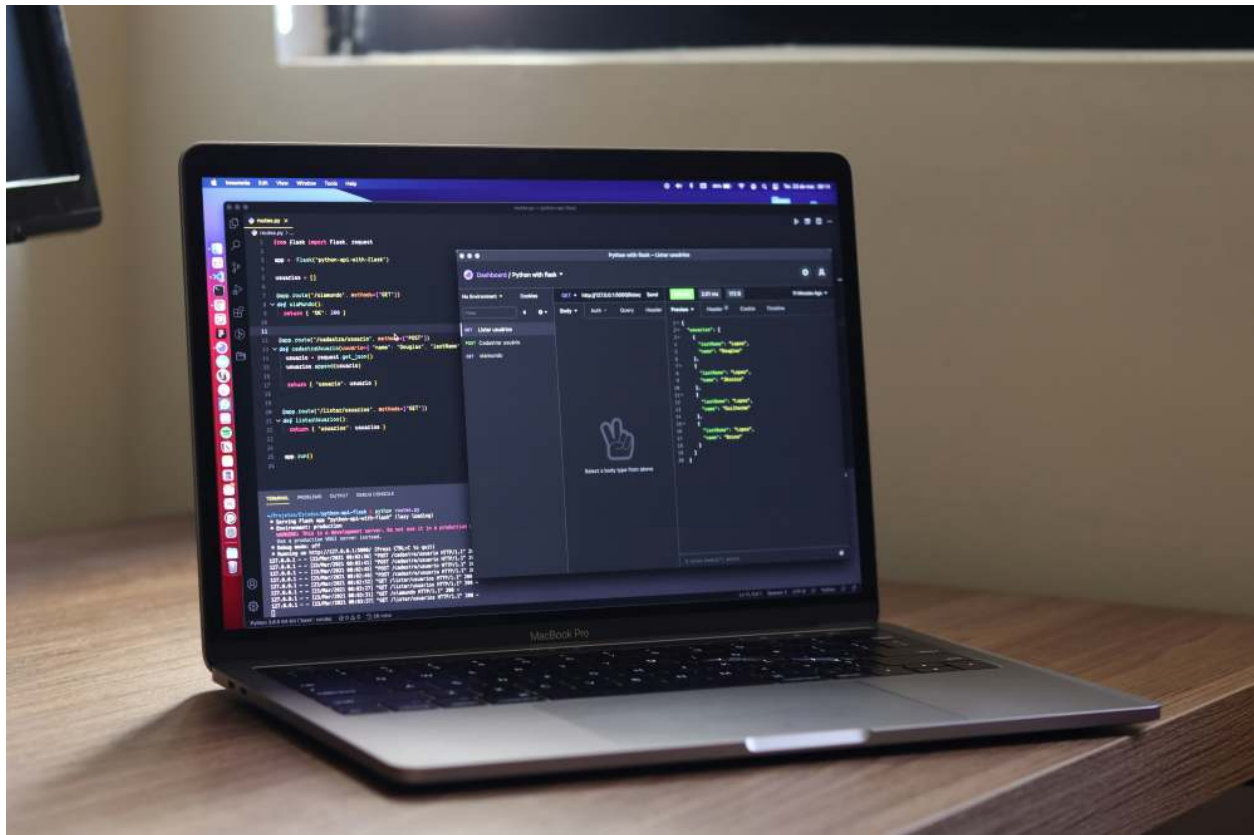
Front-end Developer

Front-end Web Developer sendiri merupakan seseorang yang mengerjakan code layout sebuah website yang sebelumnya layout dan gambaran websitenya sudah dikerjakan oleh UI/UX Designer.



Seorang Front-end Developer bertanggung jawab atas aspek-aspek yang berkaitan dengan tampilan dan fungsionalitas situs web yang diakses oleh pengguna (client-side). Seorang Front-end Web Developer minimal harus memiliki keahlian dalam HTML, CSS, dan Javascript, yang merupakan dasar penting dalam peran ini.

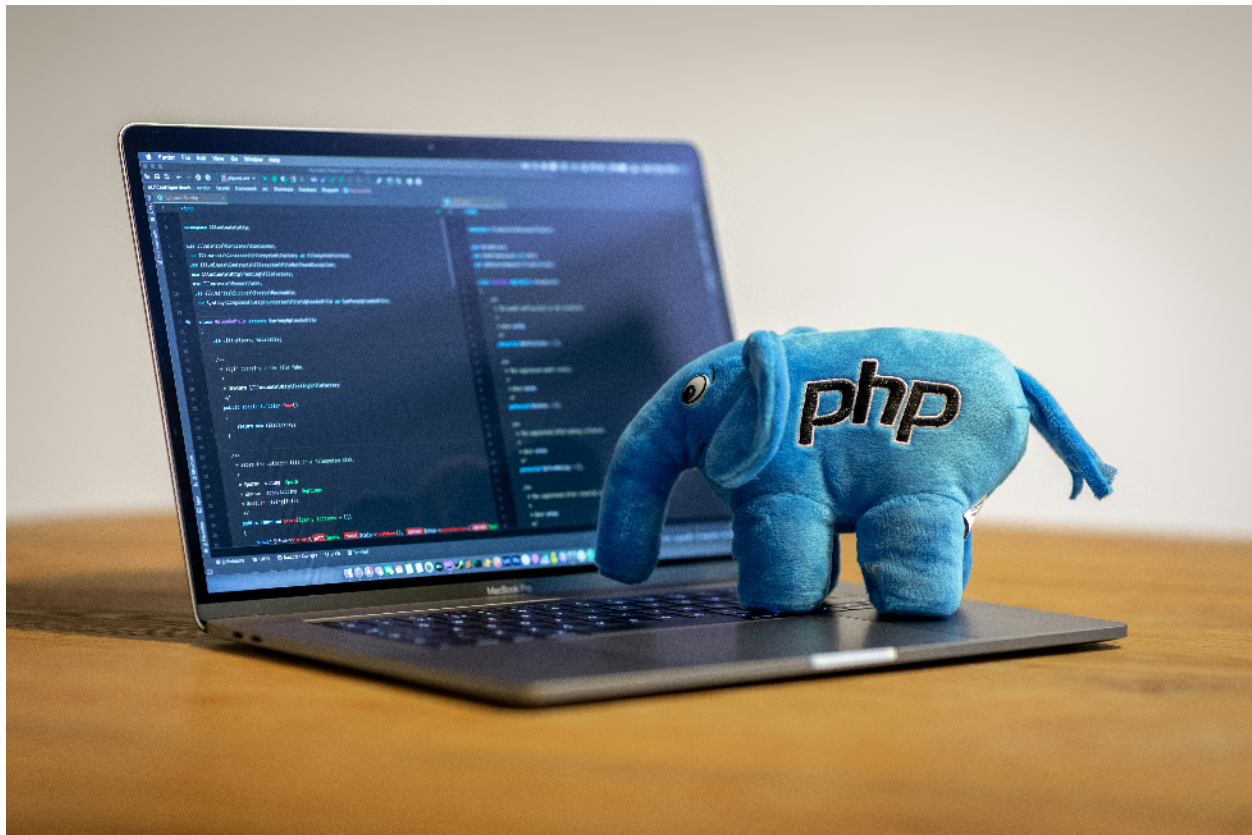
Kemampuan menggunakan Framework seperti (namun tidak terbatas pada) Tailwind CSS, React JS, atau VueJS akan menjadi nilai tambah. Terkadang, seorang Front-end Web Developer juga melibatkan diri dalam peran sebagai Web Designer.



Seorang front-end umumnya tidak terlibat secara langsung dalam manajemen Database atau bahasa pemrograman PHP (alias Back-end Development). Adapun jika memang terlibat, biasanya hanya untuk menampilkan data yang telah disiapkan oleh Back-end Web Developer dalam bentuk API, yang memiliki fokus pada bagian tersebut. Proses ini biasanya dinamakan dengan Integrasi API.

Back-end Developer

Seorang Back-end Web Developer memiliki peran dalam mengembangkan sistem yang mengatur cara sebuah website beroperasi. Sistem ini berjalan di dalam server (alias server-side).



Tugas utama dari seorang Back-end Web Developer sendiri salah satunya adalah melibatkan penggunaan bahasa pemrograman server-side seperti PHP, Javascript (NodeJS), Golang, Python, ASP, atau yang lainnya. Seorang Back-end web Developer jugamenghandle bagian Database, seperti MySQL, PostgreSQL, MongoDB, dan sejenisnya. Di perusahaan besar atau di corporate, pekerjaan ini biasanya ditangani oleh Database Analyst.

Seorang Backend Developer sendiri perlu memiliki pemahaman yang kuat tentang konsep algoritma, yang biasanya dipelajari pada awal semester kuliah, serta beberapa konsep pemrograman seperti OOP (Object Oriented Programming). Selain itu, back-end developer juga

harus mampu memastikan keamanan aplikasi atau website yang dibuat dari potensi serangan (hacking) oleh pihak yang tidak bertanggung jawab.

Full-stack Developer

Seorang Full-stack Developer adalah gabungan dari kedua bagian tersebut, atau dalam kata lain adalah menjadi "Superman" atau "Single Fighter". Seorang Full-stack Developer biasanya mempunyai tanggung jawab yang melibatkan semua aspek pekerjaan yang biasanya dilakukan oleh Front-end dan Back-end Developer.

Seorang Full-stack Developer tidak hanya menguasai tampilan dan interaksi antarmuka pengguna, tetapi juga memiliki pemahaman mendalam tentang struktur server, database, dan business logic di balik layar.

Dalam banyak kasus, terutama di perusahaan skala menengah kecil (UKM) dan perusahaan non-startup, peran seorang web developer sering kali melibatkan tanggung jawab sebagai Full-stack Web Developer. Hal ini biasanya disebabkan oleh keterbatasan SDM, termasuk kendala pada gaji. Di lingkungan kerja yang serba terbatas, para web developer diharapkan untuk dapat menangani seluruh spektrum pekerjaan mulai dari desain antarmuka pengguna, pembuatan tampilan depan (Front-end), hingga development dan database di bagian Back-end.

Saya harus pilih yang mana?

Tentunya, ini dikembalikan ke kalian, tergantung keinginan dan kemampuan kalian. Untuk saya pribadi, saya bisa bisa dibilang sebagai Full-stack Web Developer. Tapi, secara pribadi, saya lebih memilih menjadi jadi Front-End Web Developer, alasannya karena saya lebih senang dengan visual dibandingkan logic.

Kalau kalian punya skill di logic, hitung-hitungan, atau lebih memilih untuk mengerjakan business process dalam suatu sistem, mungkin cocok jadi Back-end Web Developer. Tapi, kalau kalian lebih enjoy ngurusin tampilan website yang dilihat sama pengguna, lebih baik untuk menjadi Front-end Web Developer.

Intinya, pilih sesuai passion kalian. Do what you love and love what you do.

Dimana saya bisa belajar?

Banyak sumber yang dapat kamu pelajari untuk menjadi seorang web developer, antara lain adalah:

- 1. Mencari referensi di Google dan situs-situs pemrograman**

Google merupakan alat yang sangat efektif, jika kalian memaksimalkannya dengan baik. Situs-situs pemrograman seperti Stack Overflow, Quora, dan lainnya menyediakan banyak informasi dan solusi untuk masalah pemrograman yang umum dihadapi.

2. Belajar dari kelas online

Kelas online dari platform seperti BuildWithAngga, dan yang lainnya menawarkan kursus-kursus berkualitas dalam bidang web development. Ini memberikan fleksibilitas waktu dan tempat untuk belajar sesuai kebutuhan kalian.

3. Membaca dokumentasi

Dokumentasi resmi untuk bahasa pemrograman, framework, atau tools yang sedang kamu pelajari adalah sumber informasi yang sangat berharga. Meskipun kadang terlihat kompleks, membaca dokumentasi membantu kamu memahami secara mendalam konsep dan fungsionalitas yang ada. Website dokumentasi seperti MDN Web Docs bisa menjadi sumber yang bagus untuk kalian belajar.

4. Menggunakan situs roadmap web developer

Roadmap.sh adalah platform yang menyediakan roadmap atau peta jalan untuk berbagai bidang, termasuk development web. Di sini, kamu bisa menemukan panduan langkah demi langkah untuk menjadi seorang web developer yang sukses. Roadmap ini mencakup berbagai topik mulai dari dasar-dasar pemrograman,

pengembangan Front-end, Back-end, hingga spesialisasi dalam teknologi tertentu. Mengikuti roadmap dari sumber ini dapat membantu kamu merencanakan belajar dengan lebih terstruktur.

5. Belajar langsung dari ahli-nya (mentor)

Mengambil kesempatan untuk belajar langsung dari seseorang yang sudah berpengalaman dapat memberikan wawasan yang berharga. Bergabung dengan komunitas developer web atau forum online bisa membuka peluang untuk menemukan mentor atau mendapatkan feedback dari para ahli.

Langkah selanjutnya

Selanjutnya apa? selanjutnya adalah memperdalam keahlian tambahan untuk menjual jasa kita dan mendapatkan bayaran yang layak atau bahkan lebih tinggi dari biasanya. Adapun beberapa cara yang bisa kita lakukan untuk mendapatkan bayaran adalah sebagai berikut:

- Membuat template HTML & CSS lalu menjualnya di Envato
- Mengerjakan projek freelance dibayar per jam
- Mengerjakan projek freelance dibayar per project
- Bekerja sebagai full/part-time di dalam sebuah perusahaan, pada case ini kita bisa bekerja secara on site (di tempat) atau remote (dari rumah/kafe).

Well, agar kita bisa melakukan beberapa hal di atas yang dapat memberikan penghasilan maka kita membutuhkan sebuah portfolio berkualitas tinggi, apakah kamu sudah punya? jika belum maka silahkan simak pada chapter selanjutnya.

Chapter 2: Portfolio hebat pertamamu

Ijazah dan sertifikat keahlian kita sebagai professional website developer sangat penting untuk dimiliki sebagai modal bekerja. Namun sebuah portfolio nilainya jauh lebih penting dibandingkan kedua hal tersebut.

Portfolio adalah bukti utama untuk membuktikan sebegus apa skills dan pengalaman yang kita miliki saat ini sebagai website developer.

Portfolio itu bisa kita dapat dari mana aja sih mas?

Kita biasanya dapat memiliki sebuah portfolio dari berbagai sumber berikut:

- Portfolio dari hasil latihan menguasai bahasa pemrograman
- Portfolio dari hasil volunteer di sekolah atau kampus kita
- Portfolio dari hasil bekerja dari perusahaan kita sebelumnya
- Portfolio dari hasil mengerjakan projek freelance bersama klien
- Portfolio dari hasil eksplorasi untuk meningkatkan website yang sudah tersedia

Demi mendapatkan hasil yang maksimal maka kita tidak bisa memilih sebuah projek sebagai **portfolio kita secara asal-asalan** atau yang penting punya portfolio. Karena nanti khawatirnya kita hanya membuang waktu saja yang harusnya bisa kita pake untuk fokus membangun portfolio berkualitas tinggi dan sesuai kebutuhan industri.

Portfolio yang bagus itu biasanya seperti apa?

Adapun beberapa point penting yang perlu kita ingat apabila ingin membangun sebuah portfolio pertama kita yang menarik, bermanfaat, dan berdampak besar sebagai berikut:

- Projek yang kita bangun wajib menggunakan **bahasa pemrograman sesuai kekuatan kita** misalnya PHP atau JavaScript dan projek tersebut wajib memiliki fitur yang cukup banyak sehingga kita bisa menerapkan kedua bahasa pemrograman tersebut dengan baik di dalam sebuah projek.
- Projek untuk kita jadikan portfolio wajib **memberikan dampak yang baik** untuk banyak orang misalnya kita bikin website yang membantu orang lain menemukan tutor yang tepat untuk kesuksesan karir mereka.
- Projek tersebut wajib menerapkan konsep pemrograman yang sesuai standard misalnya **clean-code architecture** agar mudah di-maintain bersama dengan developer lainnya

- Membuat dokumentasi yang details terkait proyek tersebut sehingga portfolio kita menjadi lebih terstruktur rapih dan membuat recruiter senang
- Proyek bukan asal jadi namun **perlu kita perhatikan dari segi security** karena apabila codingan kita terlihat asal-asalan maka recruiter akan meragukan keahlian kita
- Desain dan estetika dari proyek yang kita bangun perlu diperhatikan agar portfolio yang kita buat mempunyai user experience yang baik, dan nantinya akan menjadi nilai plus.



Portfolio adalah bukti nyata bahwa kita telah memiliki skills dan pengalaman yang dibutuhkan oleh dunia dan berdampak besar kepada manusia lainnya.

Tugas kamu selanjutnya

▼ ☒ Action items

- ☐ Review beberapa contoh portfolio menarik

Rekomendasi website portfolio

Aa Name	≡ URL
<u>Website Awards</u>	<u>https://www.awwwards.com/</u>
<u>Landing Page Inspirations</u>	<u>https://www.landingfolio.com/</u>

Aa Name	≡ URL
<u>Dribbble Shots</u>	<u>https://dribbble.com/</u>

☐ Tulis beberapa poin penting tentang apa yang kamu sukai dari berbagai portfolio yang tersedia pada website referensi di atas



Silahkan selesaikan action items di atas terlebih dahulu sebelum lanjut kepada materi lainnya sehingga proses belajar menjadi lebih maksimal — your mentor 🙌

Membuat portfolio website developer pertama kita

Saya akan memberikan beberapa langkah penting ketika ingin membangun portfolio pertama berkualitas di atas standard yang biasanya perusahaan pilih. Saya berharap langkah berikut dapat memberikan gambaran yang jelas agar kamu punya portfolio bagus dan berkesempatan besar diundang interview bahkan sampai diterima kerja.

Apabila kamu sudah selesai menyiapkan sebuah portfolio sesuai dengan kriteria yang telah saya bagikan di atas pada bagian portfolio yang bagus, maka selanjutnya ikuti langkah berikut:

1. Deploy portfolio tersebut kepada server pribadi, jika ingin yang gratisan maka bisa coba layanan dari vercel app atau github pages



Jika project yang kamu miliki membutuhkan proses installation agar bisa digunakan oleh pengamat portfoliomu maka sediakan juga file README pada proyek tersebut.

2. Apabila sudah berhasil terupload, selanjutnya pastikan kembali bahwa URL portfolio tersebut **dapat kita akses secara online** dari berbagai device seperti desktop, mobile, dan tablet (dan pastikan juga performace-nya cepat).
3. Bagikan URL portfolio tersebut ke orang lain untuk **mendapatkan feedback** yang bagus, sebenarnya hal ini sekaligus memperkenalkan diri dan keahlian kita.

Kita bisa membagikan URL portfolio tersebut **sesuai dengan persona dari proyek** yang kita bangun, apabila proyek tersebut kita bangun untuk membantu orang-orang mencari mentor yang tepat maka kita bisa bagikan portfolio tersebut kepada beberapa grup Facebook kumpulan mahasiswa yang sedang membutuhkan mentor.

Meningkatkan value pada portfolio pertama kita

Mungkin saat ini kamu bingung tentang ‘kok portfolio kita bagikan juga ke kumpulan mahasiswa?’ atau ‘ini sebenarnya portfolio atau proyek sampingan ya?’. Well, seperti yang sudah saya jelaskan sebelumnya bahwa memang sebaiknya portfolio yang kita bangun adalah berupa proyek yang memiliki dampak bagus juga untuk banyak orang lain.

Mengapa begitu mas? agar ketika perusahaan, client, atau recruiter melihat portfolio yang kita kirim maka mereka akan segera menyadari bahwa sebenarnya proyek tersebut sudah bernilai tinggi di mata masyarakat (pengguna dari website kita).

Singkatnya sih kita tidak perlu menjelaskan kembali skills kita dari A sampai Z, cukup tampilkan sebuah portfolio tersebut yang mewakili skills dan pengalaman kita sebagai seorang professional website developer.

Salah satu portfolio saya dari hasil membantu orang lain belajar online adalah website BuildWithAngga.com. Berkat portfolio tersebut saya dipercaya mengerjakan berbagai proyek freelance klien lokal dan pekerjaan full-time dari luar negeri yaitu Jerman.



Portfolio yang berupa projek website yang sudah digunakan dan dipercaya banyak orang akan memiliki nilai yang jauh lebih tinggi dari pada sekedar projek latihan saja.

Portfolio kita bisa dijadikan sumber income juga

Saya dulu berpikir bahwa portfolio hanya sekedar projek yang kita bangun lalu kita diamkan, well, sebenarnya projek tersebut bisa kita jadikan sumber income tambahan sembari menunggu pekerjaan atau projek freelance lainnya.

Di sini saya bagikan beberapa contoh yang bisa kita tiru sebagai berikut:

- Jual template HTML CSS menggunakan framework Tailwind atau Bootstrap misalnya contoh pada website <https://tailwindui.com/>
- Menjual API yang bakalan dibutuhkan oleh banyak developer lain
- Menjual layanan SaaS untuk bantu team marketing mempromosikan produk melalui email, misalnya pada website <https://convertkit.com/>
- Menjual projek website yang memiliki beberapa fitur siap pakai lalu dijual kepada website <https://codecanyon.net/>

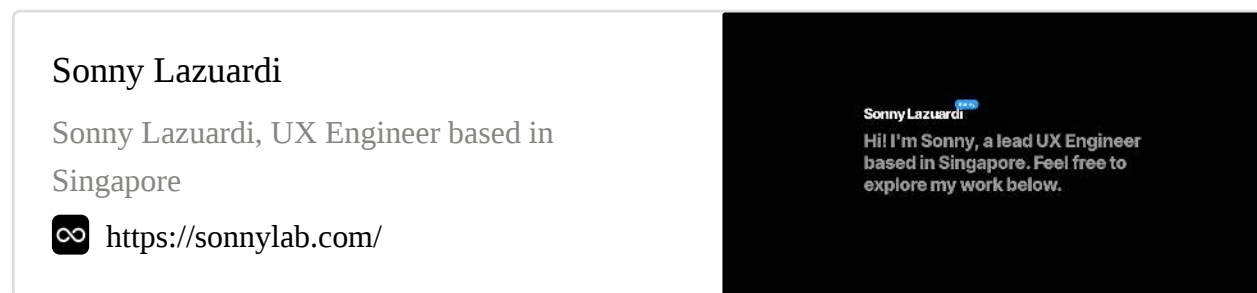
Memperlihatkan hasil karya portfolio yang kamu miliki

Untuk dapat bisa membangun portfolio yang baik, tentunya akan lebih mudah jika kita mencontoh dari hasil portfolio dari contoh yang baik dan sesuai best-practice. Dari situ, kamu dapat meng-explore hal-hal apa saja yang dapat kamu tambahkan ke portfolio yang kamu miliki.

Tentu saja, portfolio yang kalian miliki ini, perlu dimunculkan dan ditampilkan, agar orang lain dapat mengetahui apa yang kalian lakukan dan kerjakan, salah satunya dengan membuat landing page showcase portfolio yang memperlihatkan kemampuan dan hasil karya kita.

Oleh karena itu, kita akan coba untuk explore beberapa website showcase portfolio milik orang lain, lalu kita akan coba bahas dari masing-masing website.

Portfolio 1: Sonny Lazuardi



Situs milik Sonny Lazuardi merupakan contoh showcase portfolio yang dapat kamu contoh dan ikuti. Beliau membuat satu landing page yang

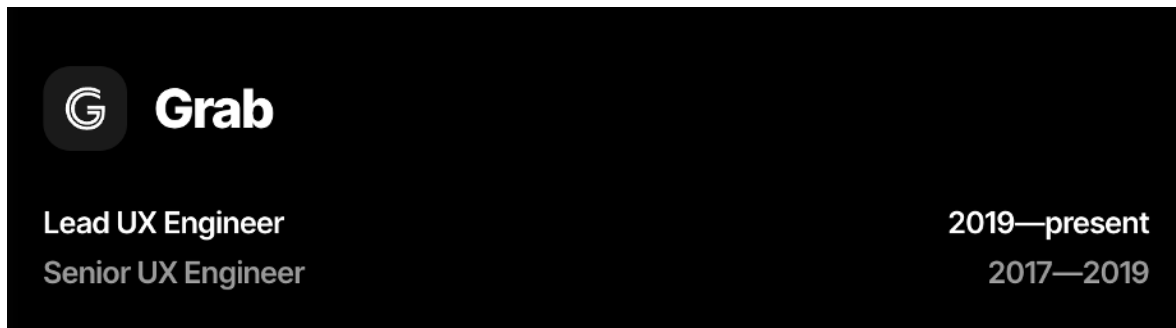
mewakilkkan pekerjaan apa saja yang dilakukan oleh beliau.

Ada beberapa hal yang membuat mengapa portfolio dari Sonny patut untuk dicontoh:

1. **Informasi yang jelas** tentang job-description atau fokus pekerjaan yang diambil pada halaman utama.



2. **Pekerjaan utama saat ini yang sedang aktif**, dan berapa lama beliau bekerja. Hal ini penting untuk mengetahui apakah seseorang sedang bekerja di perusahaan tertentu atau tidak.

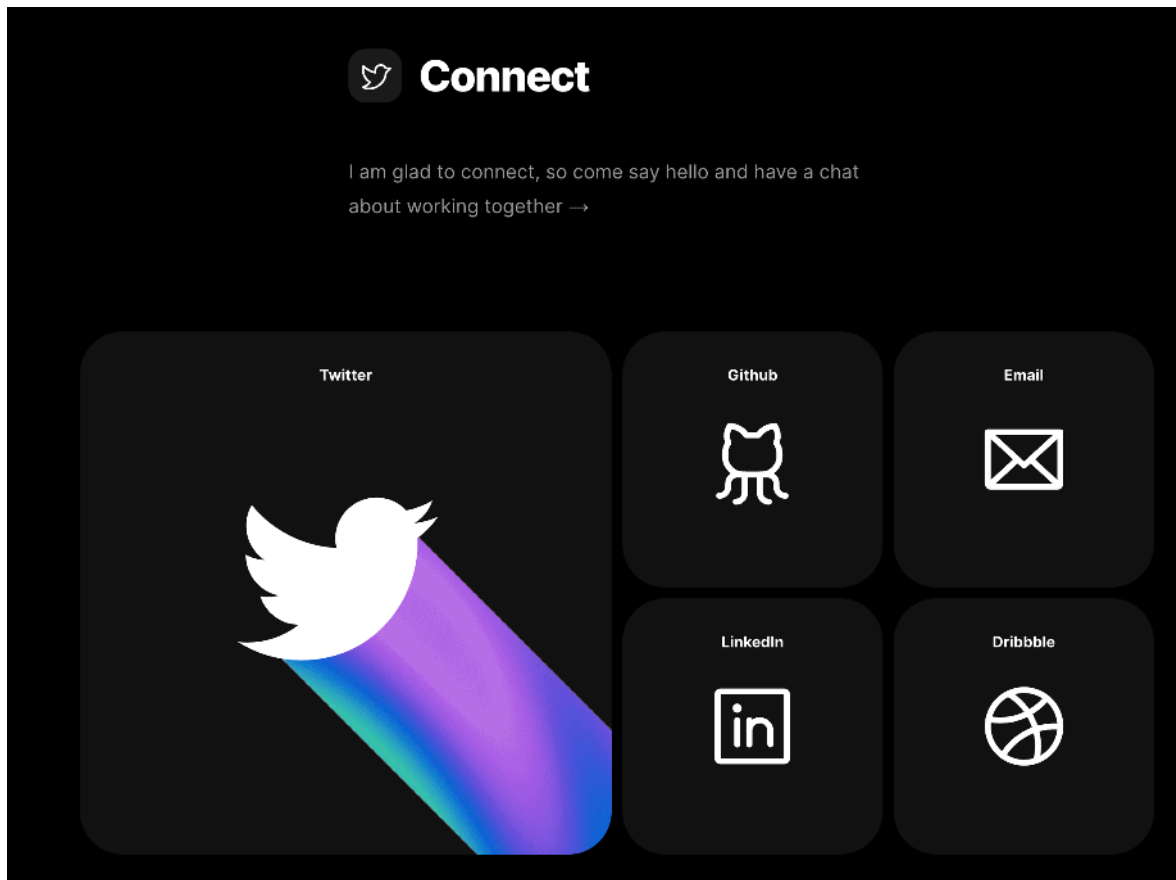


3. **Penjelasan tentang pekerjaan yang sedang dikerjakan / aktivitas yang sedang dikerjakan.** Hal ini penting sekali untuk mengetahui detail kegiatan apa saja yang dilakukan secara professional, baik itu menjadi Designer, Developer, atau hal yang lain.

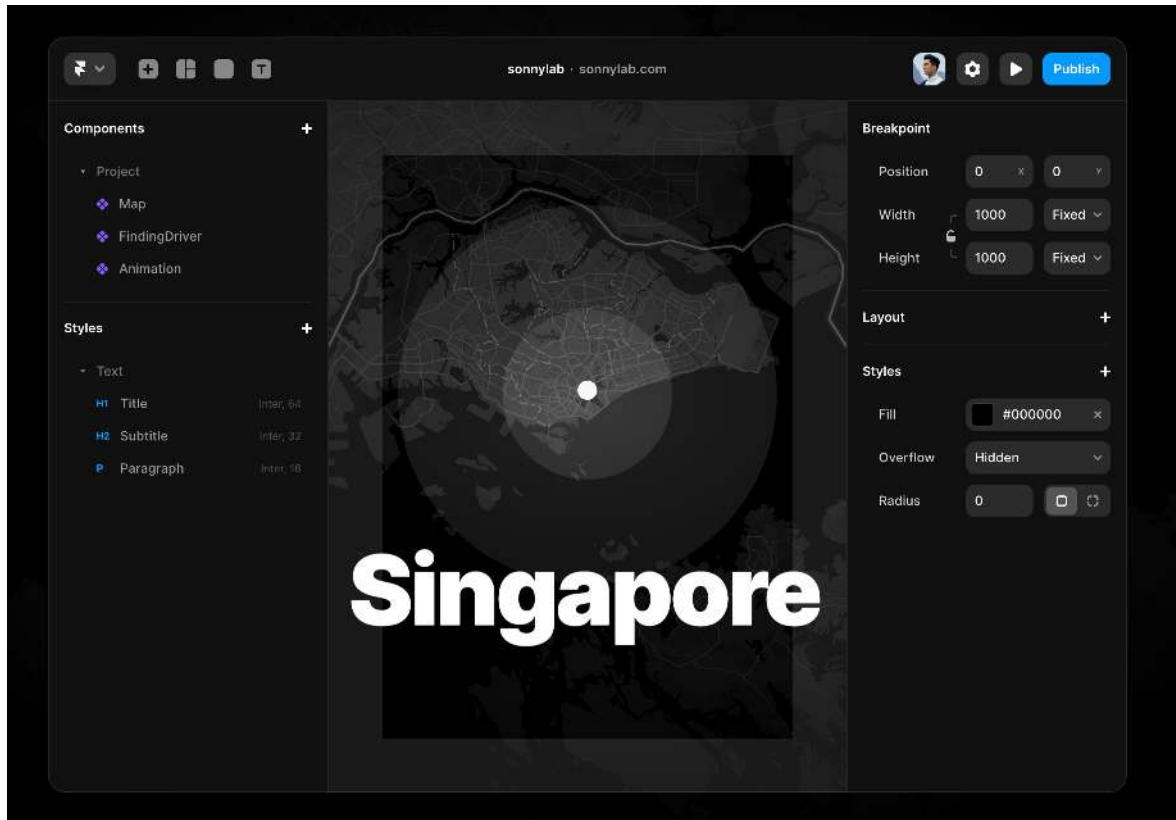
At Grab, I co-created design platform website that is the source of truth for designers and engineers. We created design system as a product. This product includes internal tools and figma plugins. Before that, I shipped high-fidelity prototypes and involved in user research to create the best experience for users.

As a lead UXE, I'm proud in building cross-team projects that embrace a "show don't tell" & "can lah" mentality—for me, engineering is an iterative process, what you ship, creating momentum forward, experiment with the ideas by putting them out there and learning from them, are the ones that can really drive innovation.


4. **Informasi media sosial untuk dapat dihubungi.** Ini sangat penting agar calon client / recruiter dapat menghubungi kita. Media sosial yang dibuat menyesuaikan dengan jenis pekerjaan yang diambil.




5. **Menggunakan tools yang sama untuk membuat website portfolio,** dengan pekerjaan yang dilakukan. Pada case kali ini, beliau memperlihatkan kemampuannya dalam menggunakan Framer. Dan website-nya sendiri buat menggunakan framer juga.




Portfolio 2: Irfan Maulana

 About // mazipan.space



About the Author of mazipan.space

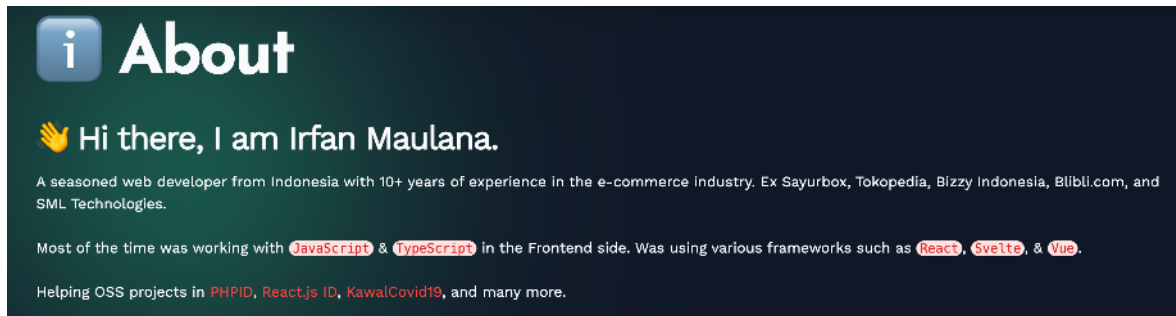
 <https://mazipan.space/about>

Situs milik Irfan Maulana merupakan contoh showcase portfolio yang dapat kamu contoh dan ikuti. Sama seperti Sony, beliau membuat satu

landing page yang mewakilkan pekerjaan apa saja yang dilakukan oleh beliau, namun dengan bentuk tampilan yang lebih compact.

Ada beberapa hal yang membuat mengapa portfolio dari Irfan patut untuk dicontoh:

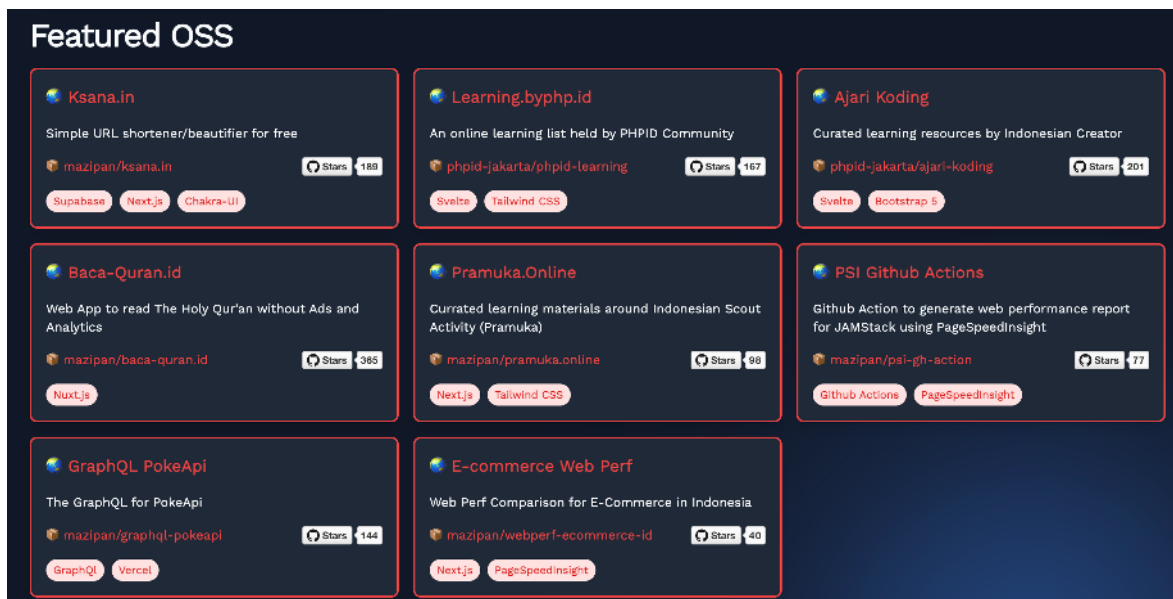
1. **Informasi yang jelas tentang apa saja yang dilakukan oleh beliau.** Menuliskan pekerjaan yang sudah dilakukan dan perusahaan mana saja dia bekerja sebelumnya.



2. **Career path yang sesuai dengan pekerjaan yang diambil.** Kita bisa melihat pengalaman seseorang dengan melihat role pekerjaan dan perusahaan mana sebelumnya dia bekerja. Hal ini sangat penting, karena seseorang dengan fokus yang baik maka akan menghasilkan hasil yang baik pula



3. **Aktif pada komunitas open-source.** Ini adalah salah satu cara untuk meningkatkan skill dan juga recognition dari seseorang. Ketika seseorang berkontribusi pada komunitas open source, banyak orang yang akan menggunakan library kalian.



4. **Menulis tulisan blog.** Selain untuk media penyimpanan dan pencatatan jikalau kita lupa dengan hal yang kita pelajari, ini jadi salah satu hal untuk memperlihatkan kemampuan kalian dalam skillset yang kalian buat

Portfolio 3: Galih Pratama

galpratama - Overview

Front-end Developer @2Viz & Full-stack

Coding Mentor - galpratama

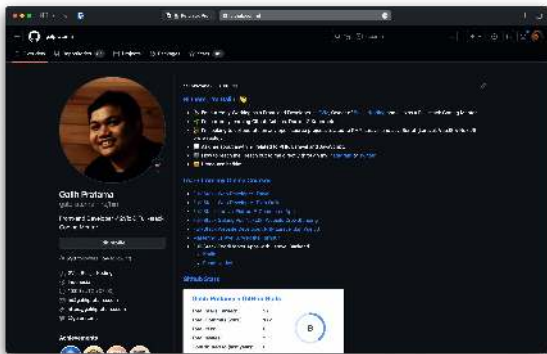


<https://github.com/galpratama>



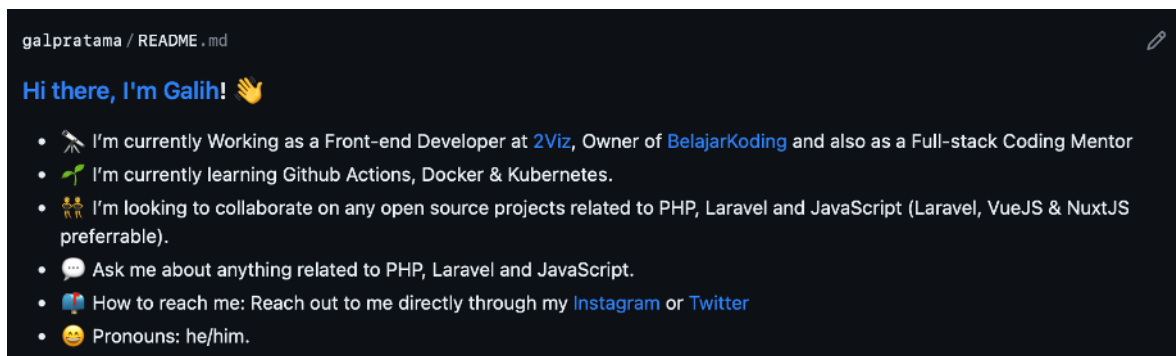
<https://www.linkedin.com/in/galihpratama/>

Portfolio milik saya pribadi, Galih Pratama, menggunakan **GitHub** sebagai platform untuk memperlihatkan hasil karya dan portfolio yang dimiliki. Jika kamu adalah seorang developer, Github bisa menjadi salah satu platform untuk dapat showcase kemampuan kamu sebagai developer. **Linkedin** juga dapat menjadi alternatif untuk kamu yang ingin memperlihatkan kemampuan dan network yang luas untuk mencari pekerjaan



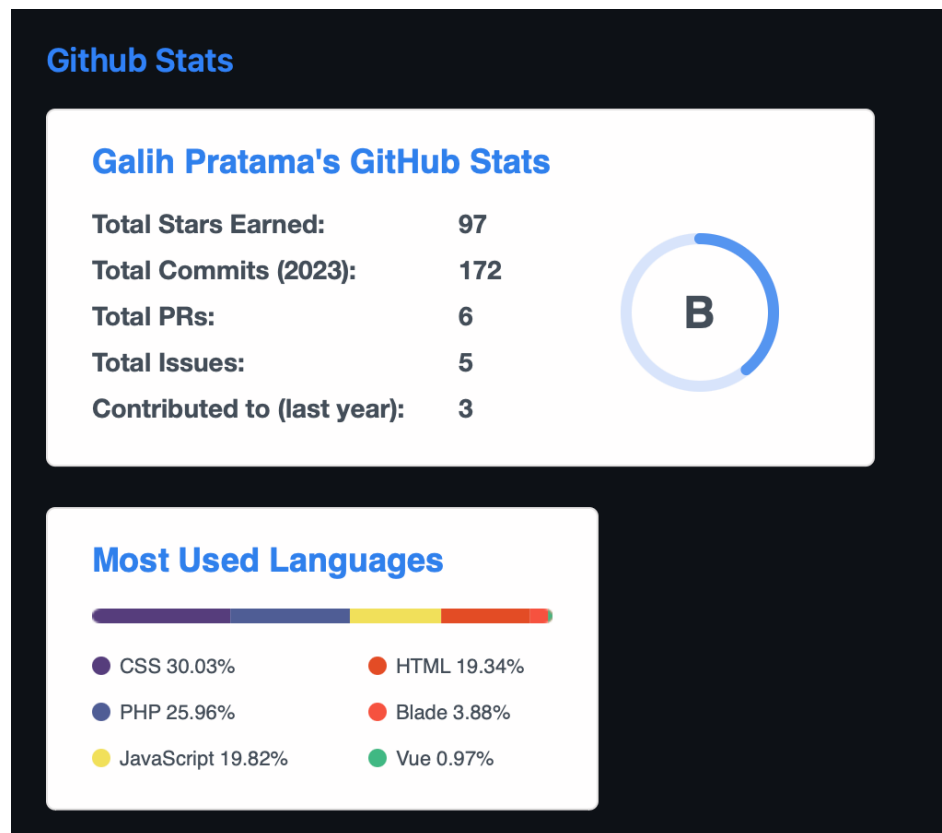
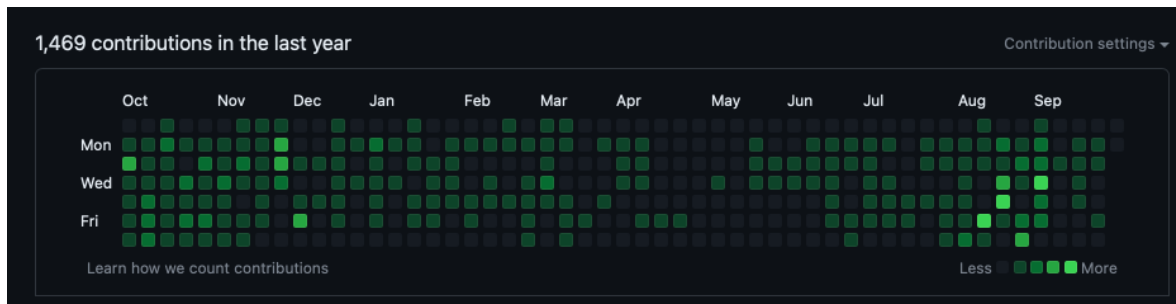
Ada beberapa hal yang membuat mengapa portfolio dari Galih patut untuk dicontoh:

1. **Informasi yang jelas tentang apa saja yang dilakukan oleh beliau.** Menuliskan pekerjaan yang sudah dilakukan dan perusahaan mana saja dia bekerja sebelumnya. Di sini juga, kita bisa menulis kegiatan apa saja, lalu informasi lainnya seperti peluang untuk open collaboration, dan hal-hal lainnya.



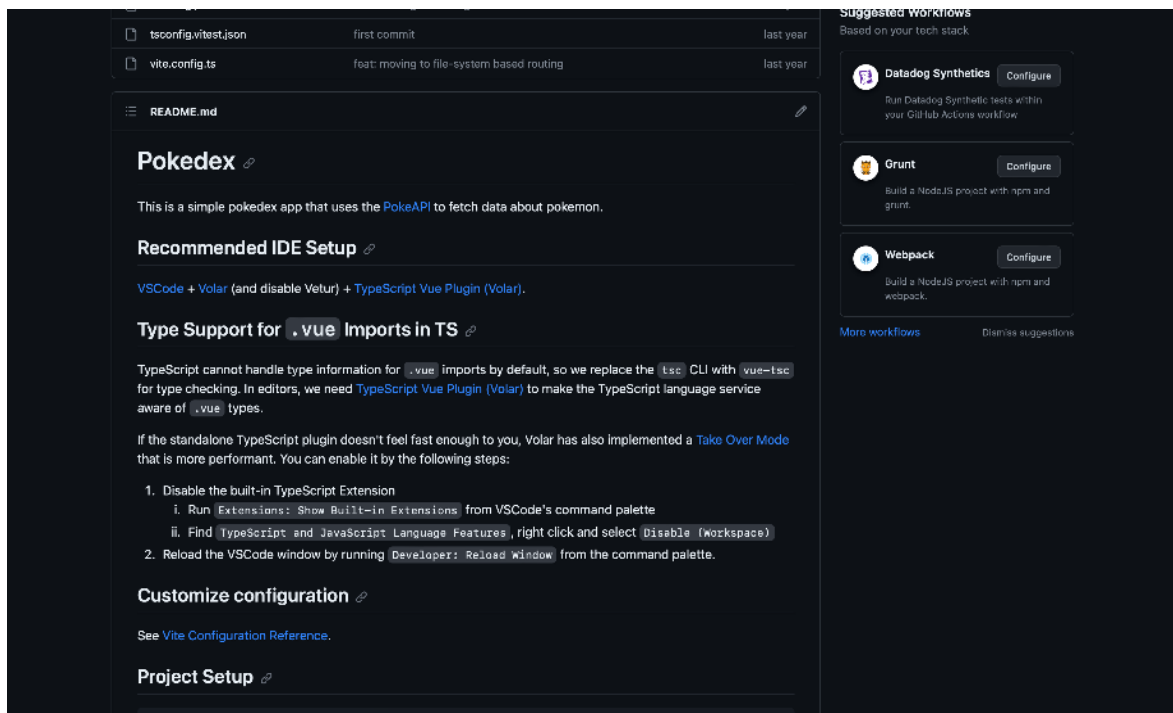
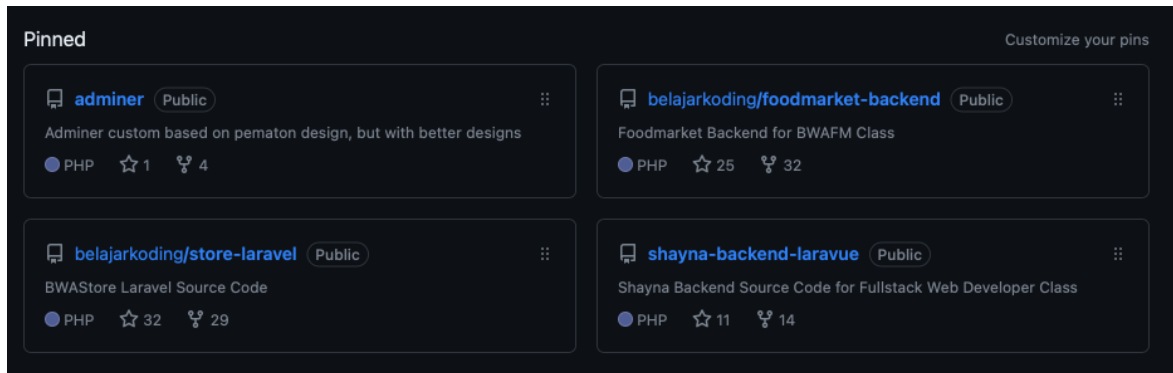
2. **Memperlihatkan keaktifan di Github.** Ini bisa menjadi salah satu bukti nyata bahwa seseorang aktif sebagai Developer. Karena hasil

karya yang dibuat, dapat dilihat langsung. Seorang developer pun dapat memperlihatkan kemampuannya dalam bahasa tertentu, dengan melihat bahasa yang sering dipakai pada project-nya.



3. **Berbagi dan berkontribusi pada open-source community.** Hal ini pun bisa dilakukan untuk meningkatkan exposure seorang developer, karena orang lain akan melihat hasil karya nyata kita


langsung, dan secara tidak langsung, akan membantu mereka pula saat mereka menggunakan project kita.




4. **Menuslkan pengalaman berkarir secara lengkap di LinkedIn.**
- Pada dunia professional, portfolio dan hasil karya akan lebih diapresiasi. Ketika kita menuliskan secara jelas pengalaman kita pada suatu perusahaan, dan menuliskan role apa saja yang dilakukan

pada perusahaan tersebut, ini akan meningkatkan kepercayaan seseorang atas kemampuan yang dimilikinya.


Experience




Frontend Developer
2Viz · Full-time
Jul 2023 – Present · 4 mos
Frankfurt, Hesse, Germany · Remote
Skills: Website Development · PHP · Front-End Development · HTML · CSS · Javascript




BuildWith Angga
Freelance · 4 yrs 5 mos
Bandung, West Java, Indonesia



Full-Stack Coding Mentor
Jun 2019 – Present · 4 yrs 5 mos
Remote
I'm an experienced educator specializing in both front-end and back-end development, with a strong focus on PHP, JavaScript, Laravel, and VueJS. I absolutely love sharing knowledge and empowering inc ...see more
Skills: Website Development · Back-End Web Development · PHP · MySQL · Laravel · Payment Gateways · Front-End Development · Tailwind CSS · Vue.js



CTO, Co-Founder
Aug 2019 – Mar 2023 · 3 yrs 8 mos
Hybrid
Having had the opportunity to collaborate with various companies and startups, I've come to realize something profound – their genuine enthusiasm for making a positive impact and enhancing peo ...see more
Skills: Management · Product Management



Ticketing Partner (Dinotix)
Akselerasi Group · Freelance
Jan 2017 – Present · 6 yrs 10 mos
Bandung, West Java, Indonesia · Remote
I have created an event management system with a range of useful features, including ticketing, crew management, and wristband functionalities. Moreover, I have designed eye-catching landing pag ...see more

Summary

Do's	Don'ts
Setidaknya 80% dari proyek yang dijadikan portfolio wajib menggunakan bahasa pemrograman sesuai keahlian kita	Melampirkan portfolio yang tidak menggunakan bahasa pemrograman sesuai keahlian kita
Perbanyak portfolio dengan real-world case study sehingga berdampak baik kepada orang lain	Hanya mengandalkan sertifikat kelulusan kelas atau Bootcamp saja
Membuat proyek untuk portfolio dengan case study yang beraneka ragam agar dapat kita gunakan untuk apply pekerjaan di berbagai sektor	Hanya memilih beberapa portfolio dengan case study yang ditujukan pada sektor yang sama
Menggunakan cinta ketika membangun portfolio layaknya membesarkan anak sendiri sehingga menjadi lebih details dan mudah dimaintain	Membangun portfolio asal jadi yang penting kodingan dan proyek jalan sesuai ilmu yang telah dipelajari sebelumnya sebagai web developer

Glossarium

API	API adalah kepanjangan dari application programming interface yang digunakan oleh developer untuk mendapatkan informasi dari database lain, sebagai contoh biasanya kita butuh API untuk cek jadwal penerbangan pesawat Garuda Indonesia.
------------	---

Portfolio	Sebuah projek yang dibangun oleh kita dan sesuai dengan keahlian kita saat ini. Portfolio digunakan untuk melamar pekerjaan karena perusahaan ingin melihat bukti nyata bahwa kita memenuhi standard.
SaaS	Kepanjangan dari Software as a Service, yaitu model bisnis dimana software dapat diakses lewat internet, tanpa harus diunduh. Biasanya layanan SaaS berbayar, ada pula yang gratis.
Deployment	Proses dimana sistem atau aplikasi yang kita buat dipublish / diunggah pada server agar dapat diakses oleh pengguna
Framework	Kerangka kerja untuk membuat aplikasi agar pembuatan aplikasi menjadi lebih mudah, cepat dan efisien

Chapter 3: Mengenal workflow pada dunia kerja

Sebagai seorang website developer yang bekerja freelance atau full-time di dalam sebuah perusahaan maka wajib bagi kita memahami alur bekerja yang tepat sehingga kita bisa memberikan hasil pekerjaan yang terbaik untuk mengejar career ladder.

Memahami flow proyek

Apabila hari ini adalah hari pertama kita bekerja pada proyek di dalam perusahaan tersebut maka hal pertama yang perlu dilakukan adalah memahami cara bekerja, goal, dan target pengguna dari proyek tersebut.

Mengapa kita perlu melakukan hal tersebut?

Agar proses adaptasi menjadi lebih baik karena kita sudah sepenuhnya memahami proyek yang ingin kita kerjakan. Kita tidak harus langsung memahami barisan koding dari proyek tersebut, cukup pelajari dulu saja bagaimana proyek tersebut dapat membantu penggunannya.

Setelah kita mengetahui cara proyek tersebut bekerja maka selanjutnya adalah mencatat poin-poin penting terkait apa saja yang dapat kita improve dari proyek tersebut misalnya meningkatkan performance ketika menarik data dari database kepada frontend.

Adapun yang dapat kamu lakukan di hari pertama sebagai website developer antara lain:

1. Banyak bertanya kepada rekan kantor
2. Mengetahui harus bertanya ke siapa
3. Ingat nama dan job desc tiap rekan kerja

4. Pelajari tools yang digunakan dari awal
5. Catat dan tulis yang sudah dipelajari

Adaptasi dengan workflow perusahaan

Setiap perusahaan memiliki alur bekerja yang berbeda-beda dalam menyelesaikan sebuah proyek, kita perlu memahami framework yang mereka gunakan apakah **kanban, scrum, atau agile** . Seorang website developer perlu mempelajari proyek manajemen sehingga dapat diandalkan ketika bekerja sama dengan tim pada perusahaan tersebut.

Memahami dokumentasi proyek

Dokumentasi tersedia untuk membantu kita para website developer yang baru bekerja pada proyek terkait, tanpa adanya dokumentasi maka akan sulit untuk kita beradaptasi dan mulai membangun fitur-fitur terbaru yang dibutuhkan oleh proyek tersebut.

Pada sebuah dokumentasi biasanya terdiri dari beberapa informasi berikut:

- Tentang keseluruhan proyek beserta goal utamanya
- Tata **cara melakukan proses instalasi** kepada laptop kita
- Susunan file pada proyek tersebut beserta tujuan utamanya

- Informasi berbagai **framework dan software yang dibutuhkan**
- Berbagai fitur yang telah dibangun pada proyek tersebut sesuai versinya
- **Endpoint API** yang digunakan pada proyek tersebut
- Dan informasi penting lainnya

Sejujurnya tidak seluruh perusahaan memiliki dokumentasi tersebut, biasanya perusahaan yang sudah besar dan stabil saja yang menyediakannya karena apabila bekerja di Startup maka mereka masih terfokus untuk menumbuhkan penggunaanya terlebih dahulu sebelum merapihkan hal penting seperti membuat dokumentasi untuk internal.

Glossarium

Coding documentation	Berisi panduan utama dari proyek tersebut seperti cara install, cara proyek bekerja, penjelasan setiap fitur yang dimiliki, dan lain-lainnya.
Startup	Sebuah organisasi yang memiliki ide terbaru untuk memberikan manfaat baik kepada orang-orang dan membuat hidup menjadi lebih mudah.
API	API adalah kepanjangan dari application programming interface yang digunakan oleh developer untuk mendapatkan informasi dari database lain, sebagai contoh biasanya kita butuh

	API untuk cek jadwal penerbangan pesawat Garuda Indonesia.
Frontend	Salah satu bagian dari website yang berupa tampilan depan dari website tersebut dan digunakan oleh pengguna untuk berinteraksi seperti mendaftar akun baru atau menambahkan barang kepada keranjang.
Kanban	Sebuah kerangka kerja yang dapat membantu kita menjadi lebih disiplin dalam menyelesaikan proyek, pada dokumen Kanban akan terbagi beberapa kategori seperti To-Do, In Progress, dan Done.

Chapter 4: Membangun reputasi dan pengalaman

Apabila kita sudah mempelajari fundamental dari website development seperti HTML, CSS, JavaScript. Selanjutnya juga kita sudah mulai bermain dengan framework populer seperti Laravel atau Vue/React JS, maka kita bisa coba membangun sebuah proyek yang dapat kita jadikan portfolio modal bekerja nantinya.

Sebagai seorang pemula website developer maka wajib bagi kita untuk memulai membangun reputasi, karena kita akan menggunakan reputasi tersebut untuk mencari pekerjaan pertama atau proyek freelance. Tanpa

memiliki reputasi dan track record yang baik maka kesempatan bisa diterima oleh klien/perusahaan sangat kecil, well, jagalah reputasi kita sebagai website developer.

Ada beberapa cara yang bisa kita lakukan untuk memulai bangun reputasi dan pengalaman kita sebagai website developer diantaranya sebagai berikut:

Membangun produk yang bermanfaat

Kita gunakan keahlian coding untuk membangun sebuah proyek yang memiliki fitur-fitur bermanfaat untuk banyak orang, misalnya bikin website ShaynaKit BuildWithAngga untuk membantu developer agar bisa bekerja lebih cepat demi hasil yang maksimal.

ShaynaKit BuildWithAngga adalah sebuah website yang dapat digunakan website developer untuk mendapatkan beberapa design website menarik untuk kebutuhan proyek mereka, dari segi user-experience juga cukup bagus sehingga proyek yang dibangun dapat memberikan impact besar kepada penggunanya.

Produk tersebut saya bangun bersama dengan tim dan telah memberikan dampak bagus kepada designer dan developer untuk menyelesaikan proyek mereka dengan cepat dan hasilnya maksimal sehingga memberikan reputasi yang baik untuk karir saya sampai saat ini.

Produk apa yang ingin kamu bangun dengan keahlian coding kamu saat ini? pastikan memiliki tujuan baik dan bermanfaat untuk pengguna ya. Jangan nunggu perfect baru dirilis, fokus bikin MVP terlebih dahulu saja untuk mendapatkan feedback dari pengguna.

Mengikuti magang (paid or non-paid)

Program magang masih menjadi paling banyak diminati oleh orang-orang untuk mendapatkan pengalaman kerja pertama mereka yang nantinya bisa meningkatkan kesempatan diterima kerja ketika interview di salah satu perusahaan.

Saya juga menyarankan kepada ribuan students di BuildWithAngga apabila mereka telah menyelesaikan kelas Premium maka selanjutnya proyek kelas tersebut dimodifikasi kembali untuk digunakan melamar magang yang tersedia sebagai website developer.

Setelah mereka magang nanti mereka dapat pengalaman yang bagus sekali untuk meningkatkan skills yang mereka miliki saat itu menjadi

lebih matang lagi. Biasanya setiap perusahaan punya cara bekerja dan projek yang berbeda-beda, hal tersebut dapat memberikan kita motivasi untuk selalu belajar hal baru, dan yang terpenting adalah keahlian beradaptasi.

Memilih program magang yang tepat

Ketika kita sudah siap untuk mengikuti program magang maka sebaiknya **tidak boleh asal apply pada seluruh lowongan tersedia**, kita perlu memilih dengan baik sehingga hasil akhir dari program magang tersebut dapat memberikan benefit dan impact yang baik kepada karir kita.

Sebelum apply magang maka sebaiknya kita siapkan terlebih dahulu CV dan **portfolio terbaik kita sesuai dengan keahlian kita** saat ini apakah PHP atau JavaScript atau bahkan keduanya. Portfolio yang kita berikan kepada pihak perusahaan juga harus sesuai dengan lowongan yang mereka buka saat itu, misalnya perusahaan mencari pemegang yang punya pengalaman di bidang Fintech, maka kita perlu kirimkan portfolio yang berkategori finance technology.



Portfolio memiliki nilai lebih tinggi dibandingkan selebar sertifikat kelulusan mengikuti Bootcamp, oleh karena itu selalu tingkatkan kembali proyek yang ingin kamu jadikan portfolio dari segi fitur dan design.

Sebagai tambahan di sini saya mau informasikan beberapa poin penting yang perlu kita pertimbangkan ketika ingin apply magang sebagai website developer:

- Apakah perusahaan tersebut memiliki reputasi yang baik ?
- Apakah keahlian saya sudah sesuai dengan kriterianya ?
- Apakah saya sudah siap berkomitmen untuk magang selama 3 bulan?
- Apa saja yang perlu saya pelajari agar proses interview lancar?

Soft skills penting ketika magang

Ketika kita mengikuti program magang maka ada beberapa soft skills yang perlu kita siapkan terlebih dahulu, namun selanjutnya nanti kita juga bisa sekaligus meningkatkan soft skills tersebut setelah kita diterima magang oleh perusahaan tersebut.

Berikut beberapa soft skills yang perlu kita persiapkan terlebih dahulu agar kesempatan diterima magang oleh perusahaan tersebut menjadi lebih besar dibandingkan kandidat lainnya:

- Communication skills
- Adaptability
- Project & time management
- Critical thinking
- Positive attitude
- Creativity

Beberapa soft skills di atas menjadi syarat utama agar kita bisa diterima sebagai pemagang, tanpa adanya soft skills di atas maka sulit bagi kita untuk adaptasi dan mengikuti bagaimana tim pada perusahaan tersebut seperti developer, marketing, dan designer untuk bekerja sama.

Kebanyakan website developer jarang bisa berkomunikasi dengan baik karena tidak terlalu sering menghabiskan waktu berkomunikasi dengan manusia lainnya, website developer menghabiskan banyak waktu di depan laptop untuk membangun sebuah website.

Saya selalu memberikan saran kepada mereka untuk coba habiskan waktu 1-2 jam kepada pengguna dari website yang mereka bangun untuk

mendapatkan feedback, well, sebenarnya hal tersebut untuk meningkatkan skills empathy dan communication kita sebagai developer.



Seorang website developer yang pandai **berkomunikasi dan memiliki empati tinggi** akan lebih mudah beradaptasi dan bekerja sama di dalam sebuah proyek kecil maupun besar demi memberikan impact yang baik kepada banyak orang.

Mengikuti program volunteer (sukarela)

Apabila kita sedang kesulitan dalam mencari lowongan magang maka bisa cobain juga alternatifnya yaitu adalah menjadi **sukarelawan sebagai website developer** dalam membangun proyek-proyek tersedia.

Kita bisa memanfaatkan grup Facebook atau Telegram tersedia yang telah dibangun komunitas untuk **mempromosikan skills dan portfolio yang kita miliki** untuk membantu menyelesaikan proyek dari anggota komunitas pada group tersebut.

Adapun beberapa perbedaan utama yang perlu kita pahami terlebih dahulu antara magang dan volunteer adalah sebagai berikut:

- kebanyakan program magang akan membayar kita sedangkan volunteer tidak.
- magang di perusahaan bisa kenal **banyak karyawan untuk belajar dari mereka** sedangkan volunteer belum tentu.
- kita dapat menganggap **projek volunteer sebagai portfolio** sedangkan magang belum tentu bisa (tergantung perusahaanya).
- kebanyakan program magang harus on-site di kantor sedangkan volunteer bisa kita ikuti dari mana saja tidak harus pergi ke kantor.
- magang wajib 3 sampai 12 bulan sedangkan volunteer boleh hanya satu hari saja.



Mengikuti volunteer untuk beberapa hari akan **memberikan wawasan terbaru** untuk pertumbuhan karir kita sebagai website developer, dan biasanya banyak hal baru yang belum kita pelajari sebelumnya.

Program volunteer dapat kita ikuti kapan saja meskipun sudah menjadi seorang professional website developer, **zaman cepat berubah dan banyak sekali teknologi yang perlu kita pelajari** dalam membangun sebuah website. Dengan menjadi volunteer maka kita akan mengerjakan berbagai projek dengan kategori yang berbeda-beda sehingga memberikan kesempatan

terbaik kepada kita untuk belajar lagi dari awal untuk **memperkaya pengalaman dan skills.**

Glossarium

Volunteer	Seorang sukarelawan yang ikut andil dalam membantu menyelesaikan sebuah proyek tanpa mengharapkan bayaran. Sukarelawan ini hanya berharap dapat mendapatkan banyak pengalaman dan skills terbaru.
Remote & On-site	Remote adalah jenis pekerjaan yang dapat kita lakukan tanpa harus datang ke lokasi sedangkan on-site adalah pekerjaan yang mewajibkan kita untuk datang ke kantor/lokasi.
Adaptability	Adaptability adalah sebuah keahlian yang wajib kita asah untuk beradaptasi dengan cepat terutama ketika berkarir sebagai website developer karena akan ada tools baru pada setiap bulannya.
Project management	Project management adalah hal penting yang dapat membantu kita mengatur proyek-proyek mulai dari timeline, deadline, handover sehingga proyek tersebut bisa selesai dengan baik.
CV/Resume	CV adalah kepanjangan dari curriculum vitae yang berisi informasi diri kita seperti keahlian, pengalaman, portfolio. Dokumen tersebut kita gunakan untuk melamar pekerjaan.

Chapter 5: Menyusun professional curriculum vitae

Portfolio memang penting untuk kita miliki dalam melamar pekerjaan, akan tetapi ada satu dokumen penting juga yang perlu kita tulis dengan baik yaitu adalah CV/Resume. Pada CV kita akan menulis informasi penting terkait diri kita, keahlian, pengalaman, dan sebagainya.

Saya akan memberikan contoh utama tentang bagaimana menulis bagian-bagian penting pada CV kita sebagai berikut:

1. Informasi pribadi

Pada bagian ini maka kita perlu menampilkan nama lengkap, no. handphone, e-mail yang aktif, dan jika memungkinkan lampirkan juga profile LinkedIn kita karena itu dapat meningkatkan value (terutama apabila kita cukup aktif di LinkedIn).

2. Keahlian

Kita punya banyak sekali keahlian namun perlu diingat bahwa kita hanya perlu menuliskan beberapa yang paling penting saja, misalnya dapat membangun website yang stabil dikunjungi oleh jutaan pengguna atau dapat menjaga performance website tetap cepat.

3. Pengalaman

Pada bagian ini kita bisa lampirkan beberapa portfolio terbaik dari yang paling terbaik dan kita jelaskan juga apa saja yang sudah kita kontribusikan pada projek tersebut, misalnya kita yang mengatur payment gateway agar tidak ada celah hacker.

Saya sarankan dalam **menulis CV sebaiknya tidak lebih dari 1 lembar** agar pembaca CV kita juga tidak merasa bosan karena terlalu banyak informasi yang padahal masih bisa kita persingkat kembali (seperti menampilkan yang penting saja).

Charles Bloomberg

📍 New York City, United States ✉ charlesbloomberg@wisc.edu ☎ (621) 799-5548 🌐 in/bloomberg

EXPERIENCE

Investment Banking Intern Citigroup

June 2014 - Present, Palo Alto, CA

- Invests Partnerships Proprietary Capital, as well as capital raised from outside investors and institutions, in real estate assets across Korea.
- Able to articulate and train the fundamental concepts, practices, and procedures associated with commercial real estate finance, including but not limited to market analysis, investment analysis and appraisal.
- Provides financial evaluation, modeling, structuring and analysis for income producing properties such as office, retail, hotel, with an emphasis on multi-family investments.
- Assists team in day-to-day responsibilities, which include the preparation of offering materials and transactional support and negotiation.
- Prepares forecasts and analyzes trends in commercial real estate finance, general business and industry conditions. Interfaces with clients, equity investors and lenders to execute financings and document negotiations.

Banking & Financial Analyst Intern Oracle

December 2011 - March 2012, Palo Alto, CA

- Executed background checks on current clients and cold called potential clients interested in buying municipal or junk bonds.
- Trained on the responsibilities required to be a trader and the different types of bonds, maturity, and rates under the Bloomberg system.
- Researched and assessed the performance of different bonds under the Bloomberg system and recommended buy/sell strategy based upon the bonds current and predicted strength.
- Learned about the different roles in a financial services company, which includes bond analyst and bond broker.
- Supported the company with all generic administration tasks from organizing conference calls and meetings as well as formatting Excel files and presentations.

Financial Analyst Intern Amazon

May 2010 - September 2010, Palo Alto, CA

- Analyzed database sales and transactions to ensure proper numbers through auditing and transformation into new SPRING database.
- Created revenue reporting and insights while developing dashboards with performance indicators and tools to drive incremental revenue.
- Communicated through financial presentations of data findings of financial reports, cost analysis, and revenue. Learned about audit reporting as well as analyzing invoices and management.
- Through analysis, created recommendations with finance team to improve operations through business meetings and the use of Microsoft PowerPoint.

RELEVANT PROJECTS

Predicting Market Behavior

- Completed a self-directed project that developed and maintained complex financial models, and performed valuation analysis by using Discounted Cash Flow, comparable company trading multiples, and precedent transactions.

EDUCATION

Bachelors of Science in Finance

Minor in Economics • University of Wisconsin - Madison • Madison, WI • 2014 • 3.69

CERTIFICATIONS

Chartered Financial Analyst Level 1 | CFA Institute | 2019

CAMPUS INVOLVEMENT

Capital Finance Club

University of Wisconsin - Madison • Vice President of Operations • September 2013 - June 2014

- Developed, planned, and implemented short and long-term goals, to include organizing and scheduling work activities for development.

SKILLS

Technical Skills: Tableau, R, STATA, MicroSoft Excel, Bloomberg Terminal, Python, Financial Analysis

Soft Skills: Interpersonal Communications, Research, Data Entry

Fields of Interest: Analysis, Capital Markets, Cryptocurrency

Kita juga bisa menambahkan beberapa link portfolio pada CV agar recruiter dapat melihatnya lebih details, kita bisa menggunakan link proyek aslinya dari klien atau perusahaan tempat kita bekerja/magang sebelumnya atau bisa menggunakan link dari website pribadi saja.



Perlu diingat bahwa CV akan menjadi sebuah dokumen first impression kita kepada recruiter sebelum masuk ke tahap interview. Pastikan informasi yang kamu tulis dapat menggambarkan seberapa professional-nya diri kamu.

Chapter 6: Mencari Kerja Remote dan Melamar Pekerjaan Pertama

Setelah kita memiliki pengalaman magang dan keterampilan yang sesuai dengan standar seorang website developer, sudah saatnya mencoba melamar pekerjaan pertama kita. Tujuan utama dalam tahap ini bukanlah langsung mendapatkan pekerjaan, tetapi setidaknya bisa lolos seleksi awal dan mendapatkan kesempatan wawancara. Dengan begitu, kita bisa mulai memahami bagaimana proses rekrutmen bekerja dan mendapatkan pengalaman berharga.

Namun, banyak fresh graduate yang melakukan kesalahan dengan **melamar pekerjaan secara asal-asalan** tanpa strategi yang jelas. Mereka mengirimkan aplikasi ke banyak perusahaan tanpa mempertimbangkan kecocokan dengan keterampilan yang dimiliki. Akibatnya, mereka sering kali tidak mendapatkan panggilan wawancara dan cepat merasa frustrasi.

Menyusun Strategi untuk Meningkatkan Peluang Diterima

Alih-alih melamar pekerjaan secara acak, kita bisa menggunakan strategi yang lebih efektif untuk meningkatkan peluang mendapatkan wawancara:

1. **Pahami Keterampilan yang Dimiliki** – Identifikasi keahlian utama yang kita miliki, misalnya pengalaman dalam framework JavaScript seperti Vue atau React.
2. **Cari Lowongan yang Sesuai** – Gunakan filter pencarian pekerjaan untuk menemukan posisi yang benar-benar sesuai dengan keterampilan yang kita miliki.
3. **Sesuaikan Portofolio dengan Lowongan** – Jika melamar sebagai Frontend Developer dengan Vue, pastikan ada proyek terkait di dalam portofolio.
4. **Persiapkan Jawaban yang Meyakinkan** – Saat wawancara, pastikan kita bisa menjelaskan mengapa memilih teknologi tertentu, bagaimana menggunakannya secara optimal, serta kasus penggunaan yang sesuai.



Perusahaan lebih menghargai kandidat yang memahami alasan teknis di balik pemilihan teknologi, bukan sekadar menggunakannya karena tren.

Dimana Bisa Mencari Pekerjaan Remote?

Jika tertarik untuk bekerja secara remote, ada beberapa platform yang bisa digunakan untuk mencari lowongan pekerjaan:

1. **LinkedIn** – Salah satu platform paling populer yang digunakan perusahaan untuk merekrut kandidat dari seluruh dunia.
2. **We Work Remotely** – Situs khusus yang berfokus pada pekerjaan remote di berbagai bidang, termasuk pengembangan web.
3. **Remote OK** – Menyediakan daftar pekerjaan remote dengan berbagai kategori, termasuk untuk developer.
4. **Upwork & Fiverr** – Platform freelance yang memungkinkan kita mendapatkan proyek jangka pendek hingga jangka panjang.
5. **Kalibrr** – Platform pencarian kerja yang populer di Asia Tenggara, dengan banyak lowongan dari perusahaan lokal maupun internasional.

Apakah Perlu Menggunakan Bahasa Inggris?

Bekerja di perusahaan luar negeri sering kali mengharuskan kita untuk berkomunikasi dalam bahasa Inggris. Namun, tingkat kebutuhan bahasa Inggris bisa bervariasi tergantung pada perusahaan dan posisi yang kita lamar:

- **Perusahaan Global** (AS, Eropa, Kanada) biasanya menggunakan bahasa Inggris dalam komunikasi sehari-hari.
- **Perusahaan Asia (Jepang, Korea, China)** mungkin masih menggunakan bahasa lokal untuk komunikasi internal, tetapi bahasa Inggris tetap dibutuhkan untuk laporan atau pertemuan dengan tim global.
- **Startup Internasional** sering kali menggunakan bahasa Inggris sebagai bahasa utama, bahkan jika timnya tersebar di berbagai negara.

Jika kemampuan bahasa Inggris masih terbatas, kita bisa mulai dengan meningkatkan keterampilan menulis dan berbicara dengan mengikuti kursus atau latihan percakapan.

Negara yang Direkomendasikan untuk Pekerjaan Remote

Beberapa negara lebih terbuka terhadap pekerja remote dibandingkan yang lain. Berikut adalah beberapa negara yang sering merekrut developer remote:

1. **Amerika Serikat** – Banyak perusahaan teknologi besar dan startup yang mencari developer berbakat dari seluruh dunia.
2. **Kanada** – Memiliki budaya kerja yang fleksibel dan banyak perusahaan teknologi yang menawarkan pekerjaan remote.
3. **Jerman** – Banyak perusahaan yang menerima pekerja remote, terutama dalam industri teknologi.

4. **Singapura** – Salah satu pusat teknologi di Asia yang menawarkan banyak pekerjaan remote dengan gaji kompetitif.
5. **Australia** – Memiliki kebijakan fleksibel untuk pekerja teknologi, termasuk untuk pekerja remote dari luar negeri.

Negara-negara ini biasanya menawarkan kompensasi yang lebih tinggi dibandingkan pekerjaan lokal, sehingga bisa menjadi pilihan menarik bagi developer yang ingin bekerja remote.

Cara Melamar Pekerjaan Melalui LinkedIn, Kalibrr, dan Situs Lainnya

Melamar pekerjaan secara online membutuhkan strategi yang tepat agar aplikasi kita dilirik oleh recruiter. Berikut beberapa langkah yang bisa dilakukan:

1. **Optimalkan Profil LinkedIn** – Pastikan profil terisi lengkap dengan pengalaman kerja, portofolio, serta kata kunci yang relevan.
2. **Buat Resume yang Menarik** – Gunakan format yang profesional, sesuaikan dengan posisi yang dilamar, dan soroti keterampilan utama.
3. **Tulis Cover Letter yang Personal** – Jangan hanya mengirimkan surat lamaran standar, tetapi sesuaikan dengan perusahaan yang dituju.
4. **Gunakan Fitur Job Alerts** – Banyak platform seperti LinkedIn dan Kalibrr menyediakan notifikasi untuk lowongan pekerjaan yang sesuai.

5. **Terhubung dengan Recruiter** – Jangan ragu untuk menghubungi recruiter melalui pesan langsung dan menunjukkan ketertarikan kita terhadap posisi tertentu.
-

Menghindari Penipuan dalam Lowongan Pekerjaan

Dalam mencari pekerjaan remote, kita juga perlu berhati-hati agar tidak menjadi korban penipuan. Berikut beberapa cara untuk menghindari scam:

1. **Periksa Kredibilitas Perusahaan** – Cari ulasan tentang perusahaan di situs seperti Glassdoor atau Trustpilot.
 2. **Hati-hati dengan Tawaran yang Terlalu Bagus** – Jika gaji yang ditawarkan jauh lebih tinggi dari standar pasar, itu bisa menjadi tanda peringatan.
 3. **Jangan Bayar untuk Melamar** – Perusahaan yang sah tidak akan meminta pembayaran untuk proses rekrutmen.
 4. **Gunakan Email Resmi** – Pastikan komunikasi dilakukan melalui email resmi perusahaan, bukan alamat email gratis seperti Gmail atau Yahoo.
 5. **Waspada Permintaan Informasi Pribadi** – Jangan pernah memberikan informasi sensitif seperti nomor rekening bank atau data pribadi lainnya sebelum ada kontrak resmi.
-

Kesimpulan

Mencari pekerjaan remote dan melamar pekerjaan pertama membutuhkan strategi yang tepat agar peluang diterima lebih besar. Dengan memahami cara menyusun aplikasi yang menarik, mencari lowongan di tempat yang tepat, serta menghindari scam, kita bisa meningkatkan peluang untuk mendapatkan pekerjaan impian.



Kemampuan komunikasi, terutama dalam bahasa Inggris, juga memainkan peran penting dalam bekerja dengan perusahaan luar negeri. Oleh karena itu, teruslah belajar dan tingkatkan keterampilan agar lebih siap menghadapi tantangan di dunia kerja remote.

Chapter 7: Menguasai strategy untuk interview

Jika saat ini kamu sedang mendapatkan undangan interview pertamamu, well, selamat ya! kamu membutuhkan persiapan yang matang agar kesempatan diterima bekerja pada perusahaan tersebut menjadi lebih besar dibandingkan kandidat lainnya.

Saya biasanya melakukan riset terlebih dahulu terkait perusahaan tersebut tentang visi dan misi mereka karena itu adalah hal penting.

Perusahaan hanya akan memperkerjakan orang-orang yang bukan hanya bekerja untuk uang tapi juga memiliki ketertarikan pada visi misi tersebut dalam mencapai tujuan utama perusahaan.



Memahami visi dan misi perusahaan dapat memberikan keuntungan kepada kita untuk menunjukkan bahwa keahlian dan pengalaman yang kita miliki dapat membantu perusahaan mereka mencapai tujuan utamanya.

Pemanasan diri dengan cara latihan coding

Beberapa hari sebelum interview maka biasanya saya habiskan waktu untuk memanaskan skills coding saya terlebih dahulu dengan cara menjawab beberapa tantangan yang tersedia pada website leetcode, codewars, dan sebagainya.

Perusahaan tentunya akan memberikan kita coding test sederhana yang perlu kita kerjakan ketika interview atau juga ada yang dibawa pulang. Latihan coding test pada website berikut dapat meningkatkan kepercayaan diri kita apabila nantinya menghadapi live coding test.

Ketika live coding test, kita bukan cuma ngoding berdasarkan challenge begitu saja tapi kita juga perlu menjelaskan mengapa memilih solusi tersebut dan impactnya kepada keseluruhan proyek bagaimana (apakah performancenya semakin bagus?).

Jadi memang keahlian berkomunikasi dan menjelaskan sesuatu itu perlu kita latih juga karena setiap tindakan yang kita lakukan akan dipertanyakan oleh senior dari perusahaan tersebut, well, perbanyak berlatih public speaking ya.

Etika yang baik ketika menjalani interview

Sejujurnya perusahaan bukan hanya mencari website developer yang kompeten, akan tetapi dari segi attitude juga akan sangat diperhatikan. Perusahaan akan memilih kandidat yang berperilaku sopan dalam menjawab dan menjelaskan sesuatu.

Selain itu juga ada beberapa mindset yang pasti dapat membantu meningkatkan value kita sebagai website developer adalah sebagai berikut:

- Bersedia untuk selalu mempelajari hal baru
- Memiliki keterampilan baik dalam menyelesaikan masalah
- Teliti pada bagian-bagian kecil di dalam sebuah proyek

- Sangat perhatian pada security di dalam sebuah projek
- Dapat mengatur waktu dan tugas dengan baik
- Dapat berkolaborasi dengan baik antar tim
- Suka membaca dokumentasi yang telah diberikan

Saya kira awalnya sebagai website developer itu hanya ngoding saja tapi ternyata setelah bekerja jadi mengerti banyak sekali hal lainnya yang perlu dipelajari dan ditingkatkan pada setiap harinya jika ingin terus bersaing dengan developer lain.

Chapter 8: Mengerjakan take home coding test

Mempelajari instruksi dari coding test yang diberikan

Sebelum memulai, ada baiknya kita mengatur waktu yang cukup dan kondisi yang nyaman sebelum mempelajari instruksi dari coding test, agar kita bisa fokus sepenuhnya kepada instruksi yang diberikan. Setelah itu, pastikan tidak ada detail kecil yang terlewatkan. Setiap detail kecil tersebut biasanya menjadi penilaian dalam coding test.

Mempelajari solution dari case yang sudah pernah dibuat oleh orang lain

Jangan ragu untuk mengintip solusi dari kasus-kasus sebelumnya, yang mungkin bisa kamu cari solusinya di Google, atau di Stack Overflow. Seringkali, ada lebih dari satu cara untuk menyelesaikan solution dari case tersebut, dan kita bisa belajar banyak dari pengalaman orang lain, yang seharusnya ini akan mempermudah kamu.

Memulai untuk mengerjakan coding test dengan waktu yang diberikan

Siapkan mental, siapkan waktu, dan siapkan hal-hal yang dapat membuat kamu fokus. Pastikan hindari semua distraksi yang mungkin terjadi, seperti notifikasi chat, telepon, dan hal-hal lainnya.

Setelah itu, pastikan kita sudah membuat rencana dengan waktu yang disediakan, dan kita bagi-bagi ke dalam beberapa bagian. Pastikan kita sudah memikirkan cara alternatif, jika cara yang kita lakukan tidak sesuai dengan harapan kita.

Menulis code dengan mengikuti best practice (clean code, DRY, dll)

Mulai menulis code dengan best practice yang tersedia, seperti Clean Code, DRY, dan yang lainnya. Clean code akan membuat coding kita mudah dibaca, dan prinsip DRY akan memastikan kita tidak membuang tenaga ketika membuat coding, karena mengulang-ulang code yang seharusnya bisa diperingkas.

Setiap bahasa pemrograman mempunyai best practicenya masing-masing, misalnya untuk PHP, ada standar yang namanya adalah PSR (PHP Standards Recommendations). Atau pada Javascript yang bernama Standard JS (JavaScript Standard Style).

Memanfaatkan AI untuk berkonsultasi dan memvalidasi pekerjaan yang dibuat

AI dibuat untuk dapat membantu kita, bukan untuk menggantikan kita. Jadi tidak ada salahnya untuk meminta bantuan pada teknologi, ketika kita merasa stuck. Namun yang perlu diingat, kita tidak bisa 100% mengandalkan AI, tapi lebih ke sebagai *copilot* atau asisten yang membantu kita. Kita bisa berkonsultasi kepada ChatGPT atau Claude dengan prompt yang sesuai, dan biasanya, kita akan mendapatkan insight yang bagus dari sana.



How to make proxy pass for nextjs dev server (via 3000 port) on nginx. I am using code-server to code my app

Edit

I'll help you set up an Nginx proxy for your Next.js dev server running on port 3000. This is a common setup when you're using code-server and want to access your Next.js application through Nginx.

Here's how to configure the Nginx proxy pass:

Nginx configuration for Next.js proxy

Code

```
server {
    listen 80;
    # Replace with your
    domain or IP
    server_name your-
    domain.com;

    location / {
        proxy_pass
        http://localhost:3000;
    }
}
```

To implement this proxy, follow these steps:

1. Save the configuration:

- Save the above configuration to a file in `/etc/nginx/sites-available/`, for example: `/etc/nginx/sites-available/nextjs`
- Replace `your-domain.com` with your actual domain name or server IP address

2. Enable the site:

bash

Copy

```
sudo ln -s /etc/nginx/sites-available/nextjs /etc/nginx/sites-enabled/
```

3. Test the configuration:

bash

Copy

```
sudo nginx -t
```

4. Restart Nginx:

Cara saya mengerjakan dan menyelesaikan coding test

Berikut ini dibawah salah satu contoh coding test yang saya kerjakan pada saat melakukan apply ke salah satu startup travel di Indonesia.

Hi Your Name,

Thank you **for** your interest **in** a career at Company Name!

As the next step **in** our recruitment process, we would like to learn more about your coding skills by sharing a coding challenge powered by coding platform xxxx.

At **this** stage, we want to see how you approach and solve problems.

This challenge was carefully chosen by our engineers.

Please use the link below to access your challenge.

Once you start the challenge, there is no turning back.

Therefore, we advise setting some time aside when you are relaxed and rested, and more importantly, **get** ready to show us your skills.

[[**TEST LINK**]]

Please note:

1. You can choose your most preferred programming **language** (Ace it **with** confidence!)
2. There will be **1** section - Tic Tac Toe Game Recreate using **JavaScript**
3. The test will take **1** hour **5** minutes.
4. Your score will be reviewed by our talent acquisition team

We kindly ask you to complete it maximum on xxxx, xx xxxx xxxx at xx.**xx** (However, sooner is better)

In the name **of** code, may the force be **with** you!

Kind regards,
Company Name Talent Acquisition Team

Saya diberikan waktu sekitar 2 jam untuk mengerjakan coding test diatas dengan menggunakan ReactJS. Pada saat itu, langkah pertama yang saya lakukan adalah mem-breakdown terlebih dahulu yang perlu saya kerjakan, dan kemampuan saya dalam bahasa yang digunakan. Langkah yang saya lakukan adalah sebagai berikut:



Klik tombol panah > untuk melihat lebih detail

▼ **Memahami kemampuan saya dalam framework React.**

Pada saat saya melakukan interview test, kemampuan saya pada bahasa React bisa dikatakan menengah, namun dikarenakan saya sudah lama tidak memegang code React, dan biasa menggunakan VueJS, maka ada learning curve yang perlu saya pelajari sebelum saya mengerjakan coding test. Saya mempersiapkan sekitar 1-3 hari untuk mempelajari dan catch-up tentang materi React yang perlu saya pelajari untuk membuat aplikasi tersebut

▼ **Memahami alur pembuatan game Tic Tac Toe secara umum**

Seperti yang diketahui, game tic tac toe adalah sebuah game sederhana yang mempunyai 2 player, dimana pemainnya menaruh karakter x dan o. Beberapa logika yang perlu ditambahkan pada aplikasi tersebut yaitu:

1. Kondisi untuk membuat 9 kotak box untuk permainan
2. Kondisi untuk mengganti karakter antar x dan o.
3. Kondisi saat player x dan o membuat pola sejajar
4. Kondisi saat permainan selesai
5. Kondisi yang menentukan pemain menang atau kalah.

Dari situ, kita dapat memahami alur untuk membuat game tersebut.

▼ Mencari referensi aplikasi yang serupa pada Github

Untuk kasus pembuatan game tic tac toe, sebenarnya sudah banyak referensi yang bisa kamu dapatkan di Github. Tidak hanya itu, kasus pembuatan coding test sebenarnya banyak tersebar di situs-situs pemrograman. Yang perlu kalian pahami, bahwa coding test mengajarkan kita untuk mencari solusi dari masalah. Solusinya tersebut, bisa didapatkan dari mana saja, termasuk research dari situs atau internet. Seorang programmer, pada kesehariannya akan mencari (research) dari manapun, termasuk Google, dan itu tidak menjadi masalah.

Pada kasus saya, saya mencari implementasi Tic Tac Toe menggunakan Javascript, dan membaca algoritma dibalik aplikasi tersebut. Baru, saya implementasikan pada code yang saya buat sendiri

▼ Bertanya kepada AI (ChatGPT) untuk mendapatkan referensi untuk algoritma pembuatan tic tac toe

Salah satu cara untuk memahami dari algoritma, salah satunya dapat kita tanyakan dengan AI. Di masa sekarang, AI sudah dapat menjelaskan secara detail, maksud dari algoritma tertentu. Tentunya, ini akan sangat memudahkan kita untuk memahami dari aplikasi yang kita buat, tidak asal copy paste saja.

▼ **Mengerjakan coding test**

Setelah kita siap dan matang untuk mengerjakan coding test, saatnya penentuan dari hasil jerih payah yang kita lakukan. Siapkan kopi, teh, atau apapun yang dapat meningkatkan produktivitasmu, karena pada waktu yang terbatas, yang menentukan apakah kedepannya akan diterima atau tidak

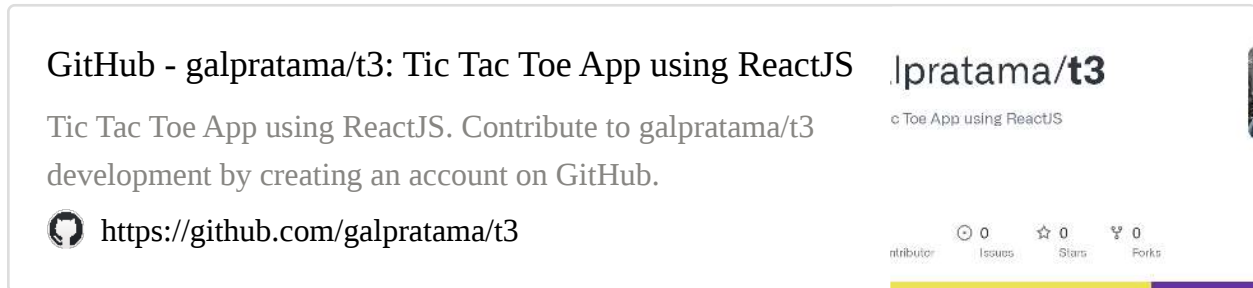
▼ **Testing hasil coding test**

Setelah kita membuat coding, tentunya kita harus mencoba dan testing berbagai kemungkinan pada aplikasi kita. Walaupun pada kenyataannya, aplikasi kita pasti akan ada saja kekurangan, namun kita harus fokus kepada test-case yang diberikan pada coding test tersebut. Biasanya, ini yang menentukan apakah seseorang lulus atau tidak pada sebuah coding test dengan melihat test case yang telah diberikan

▼ **Mempersiapkan hasil coding test kepada recruiter**

Tentunya, ini akan menjadi nilai penentu juga, saat kita dapat menjelaskan apa saja yang kita buat di coding test ini. Poin ini akan saya jelaskan di sub bab selanjutnya.

Hasil coding test dari atas dapat kamu cek pada repository Github saya yang ada di tautan dibawah ini:



Dari hasil coding test tersebut, ada beberapa poin yang sudah saya bahas diatas sebelumnya, yang saya implementasikan pada code tersebut.

Glossarium

Best Practice	Metode atau standar yang digunakan dalam penulisan kode agar lebih efisien, mudah dibaca, dan dikelola, seperti Clean Code dan DRY (Don't Repeat Yourself).
PSR (PHP Standards Recommendations)	Standar pengkodean yang direkomendasikan dalam bahasa pemrograman PHP untuk memastikan konsistensi dan keterbacaan kode.
Standard JS (JavaScript Standard Style)	Panduan standar dalam penulisan kode JavaScript untuk memastikan kualitas dan konsistensi dalam pengembangan perangkat lunak.

AI sebagai Copilot	Pemanfaatan AI sebagai asisten dalam proses coding, membantu validasi dan memberikan saran solusi, tetapi tidak menggantikan sepenuhnya peran developer.
Tic Tac Toe	Permainan sederhana berbasis grid 3x3 yang sering digunakan sebagai tantangan dalam coding test untuk menguji logika dan pemahaman algoritma.

Chapter 8 Bonus: Contoh Soal Coding Test dan Solusi Jawabannya

Untuk kamu yang menginginkan untuk latihan coding test, sebelum benar-benar melakukan coding test yang sebenarnya pada saat proses interview, saya telah merangkum contoh soal coding test beserta solusi dan jawabannya yang dapat kamu lihat dibawah ini:

Grow Up Number:

Buatlah sebuah fungsi untuk perhitungan mundur dari sebuah angka yang diambil dari parameternya, outputnya harus berupa angka yang menghitung mundur sesuai dengan digit angka yang diberikan sampai dengan index ke 0, jadi buatlah sebuah fungsi untuk soal berikut ini dan outputnya harus mengembalikan sebuah Array seperti dibawah ini.

Output:

```
growUpNumber(5) → [5, 4, 3, 2, 1, 0]
```

```
growUpNumber(3) → [3, 2, 1, 0]
```

growUpNumber(0) → []

Jawaban:

Fungsi ini menggunakan loop **for** untuk membuat array yang berisi angka dari nilai yang diberikan sampai dengan 0, dan kemudian mengembalikan array tersebut sebagai output. Contoh penggunaan menunjukkan hasil sesuai dengan contoh yang diberikan.

```
<?php
```

```
function growUpNumber($number) {  
    $result = [];  
    for ($i = $number; $i >= 0; $i--) {  
        $result[] = $i;  
    }  
    return $result;  
}
```

```
// Contoh penggunaan
```

```
$output1 = growUpNumber(5);  
$output2 = growUpNumber(3);  
$output3 = growUpNumber(0);
```

```
echo "[" . implode(', ', $output1) . "] \n"; // Output: [5, 4, 3, 2  
echo "[" . implode(', ', $output2) . "] \n"; // Output: [3, 2, 1, 0
```

```
echo "[" . implode(', ', $output3) . "]" \n"; // Output: []  
?>
```

```
function growUpNumber(number) {  
  const result = [];  
  for (let i = number; i >= 0; i--) {  
    result.push(i);  
  }  
  return result;  
}
```

// Contoh penggunaan

```
const output1 = growUpNumber(5);  
const output2 = growUpNumber(3);  
const output3 = growUpNumber(0);
```

```
console.log(output1); // Output: [5, 4, 3, 2, 1, 0]  
console.log(output2); // Output: [3, 2, 1, 0]  
console.log(output3); // Output: []
```

Character Mapping

Buatlah sebuah fungsi untuk menentukan pemetaan karakter unik pada sebuah String, dimana yang nantinya akan mengembalikan pemetaan karakter dari masing-masing setiap karakter tersebut, jadi buatlah sebuah

fungsi untuk soal berikut ini dan outputnya harus mengembalikan sebuah Array seperti dibawah ini.

Output:

```
characterMapping("babbbcb") → [0, 1, 0, 0, 2, 0]
characterMapping("aaabb") → [0, 0, 0, 1, 1]
characterMapping("fluffy") → [0, 1, 2, 0, 0, 3]
characterMapping("buttery") → [0, 1, 2, 2, 3, 4, 5]
characterMapping("canine") → [0, 1, 2, 3, 2, 4]
```

Jawaban:

Fungsi ini menggunakan associative array (untuk PHP) dan objek (untuk JavaScript) untuk melacak pemetaan karakter ke nilai numerik yang unik. Setiap karakter yang baru muncul dalam string akan diberikan pemetaan baru.

```
<?php
```

```
function characterMapping($str) {
    $charMap = [];
    $result = [];
    $currentMapping = 0;

    for ($i = 0; $i < strlen($str); $i++) {
        $currentChar = strtolower($str[$i]);
```



```

        if (!isset($charMap[$currentChar])) {
            $charMap[$currentChar] = $currentMapping;
            $currentMapping++;
        }

        $result[] = $charMap[$currentChar];
    }

    return $result;
}

// Contoh penggunaan
$output1 = characterMapping("babbbcb");
$output2 = characterMapping("aaabb");
$output3 = characterMapping("fluffy");
$output4 = characterMapping("buttery");
$output5 = characterMapping("canine");

echo "[" . implode(', ', $output1) . "]" \n"; // Output: [0, 1, 0, 0
echo "[" . implode(', ', $output2) . "]" \n"; // Output: [0, 0, 0, 1
echo "[" . implode(', ', $output3) . "]" \n"; // Output: [0, 1, 2, 0
echo "[" . implode(', ', $output4) . "]" \n"; // Output: [0, 1, 2, 2
echo "[" . implode(', ', $output5) . "]" \n"; // Output: [0, 1, 2, 3

?>

```

```
function characterMapping(str) {  
  const charMap = {};  
  const result = [];  
  let currentMapping = 0;  
  
  for (let i = 0; i < str.length; i++) {  
    const currentChar = str[i].toLowerCase();  
  
    if (!charMap[currentChar]) {  
      charMap[currentChar] = currentMapping;  
      currentMapping++;  
    }  
  
    result.push(charMap[currentChar]);  
  }  
  
  return result;  
}
```

```
// Contoh penggunaan  
const output1 = characterMapping("babbcb");  
const output2 = characterMapping("aaabb");  
const output3 = characterMapping("fluffy");  
const output4 = characterMapping("buttery");  
const output5 = characterMapping("canine");
```

```
console.log(output1); // Output: [0, 1, 0, 0, 2, 0]
console.log(output2); // Output: [0, 0, 0, 1, 1]
console.log(output3); // Output: [0, 1, 2, 0, 0, 3]
console.log(output4); // Output: [0, 1, 2, 2, 3, 4, 5]
console.log(output5); // Output: [0, 1, 2, 3, 2, 4]
```

Anagram

Buatlah sebuah fungsi untuk menghasilkan kata - kata dari sebuah nama yang telah disediakan, kembalikan nilai true jika nama yang diberikan dapat menghasilkan sebuah kata - kata seperti yang berada didalam sebuah Array, jika nama nya tidak sama dengan kata - kata yang berada didalam sebuah Array maka kembalikan nilai false, jadi buatlah sebuah fungsi untuk soal berikut ini dan outputnya harus mengembalikan sebuah Boolean seperti dibawah ini.

Output:

```
isAnagram("Justin Bieber", ["injures", "ebb", "it"]) → true
isAnagram("Natalie Portman", ["ornamental", "pita"]) → true
isAnagram("Emma Watson", ["mows", "meant", "a"]) → true
isAnagram("Daniel Radcliffe", ["clarified", "elf", "and"]) → true
isAnagram("Chris Pratt", ["chirps", "rat"]) → false
isAnagram("Jeff Goldblum", ["jog", "meld", "bluffs"]) → false
isAnagram("Taylor Swift", ["stratify", "ow"]), → false
isAnagram("Adam Levine", ["medieval", "man"]), → false
```

Jawaban:

Kode di bawah ini menggunakan algoritma sederhana untuk mengatasi permasalahan ini dengan mengurutkan karakter-karakter dalam string dan membandingkan hasil pengurutan tersebut.

```
<?php
```

```
function isAnagram($name, $words) {  
    // Menghapus spasi dan mengonversi huruf menjadi huruf  
    $name = str_replace(' ', '', strtolower($name));  
  
    // Menggabungkan semua kata dalam array menjadi satu  
    $joinedWords = implode('', $words);  
  
    // Mengurutkan karakter-karakter dalam string  
    $sortedName = str_split($name);  
    sort($sortedName);  
    $sortedName = implode('', $sortedName);  
  
    // Mengurutkan karakter-karakter dalam string gabungan  
    $sortedWords = str_split($joinedWords);  
    sort($sortedWords);  
    $sortedWords = implode('', $sortedWords);  
  
    // Membandingkan string yang telah diurutkan  
    return $sortedName === $sortedWords;
```

```
}
```

// Contoh penggunaan

```
echo isAnagram("Justin Bieber", ["injures", "ebb", "it"]); // true
echo isAnagram("Natalie Portman", ["ornamental", "pita"]); // true
echo isAnagram("Emma Watson", ["mows", "meant", "a"]); // true
echo isAnagram("Daniel Radcliffe", ["clarified", "elf", "and"]); // true
echo isAnagram("Chris Pratt", ["chirps", "rat"]); // false
echo isAnagram("Jeff Goldblum", ["jog", "meld", "bluffs"]); // true
echo isAnagram("Taylor Swift", ["stratify", "ow"]); // false
echo isAnagram("Adam Levine", ["medieval", "man"]); // false
```

```
function isAnagram(name, words) {
    // Menghapus spasi dan mengonversi huruf menjadi huruf kecil
    name = name.replace(/\s/g, '').toLowerCase();

    // Menggabungkan semua kata dalam array menjadi satu string
    const joinedWords = words.join('');

    // Mengurutkan karakter-karakter dalam string
    const sortedName = name.split('').sort().join('');
    const sortedWords = joinedWords.split('').sort().join('');

    // Membandingkan string yang telah diurutkan
    return sortedName === sortedWords;
}
```

```
// Contoh penggunaan
console.log(isAnagram("Justin Bieber", ["injures", "ebb", "it"]);
console.log(isAnagram("Natalie Portman", ["ornamental", "p"]);
console.log(isAnagram("Emma Watson", ["mows", "meant", ""]);
console.log(isAnagram("Daniel Radcliffe", ["clarified", "elf", ""]);
console.log(isAnagram("Chris Pratt", ["chirps", "rat"])); // false
console.log(isAnagram("Jeff Goldblum", ["jog", "meld", "bluff"]);
console.log(isAnagram("Taylor Swift", ["stratify", "ow"])); // false
console.log(isAnagram("Adam Levine", ["medieval", "man"]));
```

Character Count

Buatlah sebuah fungsi untuk menghitung instans dari setiap karakter dalam sebuah String, yang dimana argumentnya memiliki dua buah String, yang harus anda lakukan hanyalah menghitung berapa banyak String pertama itu muncul, didalam karakter String yang kedua, jadi buatlah sebuah fungsi untuk soal berikut ini dan outputnya harus mengembalikan sebuah Number seperti dibawah ini.

Output:

```
charCount('b', 'big fat bubble') → 4
charCount('c', 'Chamber of secrets') → 1
charCount('f', 'frank and his friends have offered five foxes and')
charCount('x', 'edabit') → 0
```

```
charCount('a', 'Adam and Eve bit the apple and found a snake') → 2  
charCount('7', '10795426697') → 2
```

Jawaban:

Kode di bawah menggunakan fungsi `substr_count` pada PHP dan menggunakan ekspresi reguler pada JavaScript untuk menghitung kemunculan karakter dalam string. Fungsi ini mengembalikan jumlah kemunculan karakter yang dicari dalam string yang diberikan.

```
<?php  
  
function charCount($char, $string) {  
    // Menghitung jumlah kemunculan karakter dalam string  
    $count = substr_count($string, $char);  
    return $count;  
}  
  
// Contoh penggunaan  
echo charCount('b', 'big fat bubble'); // 4  
echo charCount('c', 'Chamber of secrets'); // 1  
echo charCount('f', 'frank and his friends have offered five friends'); // 5  
echo charCount('x', 'edabit'); // 0  
echo charCount('a', 'Adam and Eve bit the apple and found a snake'); // 2  
echo charCount('7', '10795426697'); // 2
```

```
function charCount(char, string) {
  // Menghitung jumlah kemunculan karakter dalam string
  const count = (string.match(new RegExp(char, 'g')) || []).length;
  return count;
}

// Contoh penggunaan
console.log(charCount('b', 'big fat bubble')); // 4
console.log(charCount('c', 'Chamber of secrets')); // 1
console.log(charCount('f', 'frank and his friends have offered')); // 1
console.log(charCount('x', 'edabit')); // 0
console.log(charCount('a', 'Adam and Eve bit the apple and ate')); // 1
console.log(charCount('7', '10795426697')); // 2
```

Get Days

Buatlah sebuah fungsi untuk menentukan selisih berapa hari antara dua tanggal berikut ini, jadi buatlah sebuah fungsi untuk soal berikut ini dan outputnya harus mengembalikan sebuah Number seperti dibawah ini.

Output:

```
getDays(new Date("June 14, 2019"), new Date("June 20, 2019")); // 6
getDays(new Date("December 29, 2018"), new Date("January 5, 2019")); // 7
getDays(new Date("July 20, 2019"), new Date("July 30, 2019")); // 10
```

Answer:

Fungsi ini menggunakan objek `DateTime` pada PHP dan objek `Date` pada JavaScript untuk menghitung selisih waktu antara dua tanggal. Selisih waktu tersebut kemudian diubah ke dalam satuan hari untuk mendapatkan hasil yang diinginkan.

```
<?php
```

```
function getDays($date1, $date2) {  
    $diff = $date2->diff($date1);  
    return $diff->days;  
}
```

```
// Contoh penggunaan
```

```
$date1 = new DateTime("June 14, 2019");  
$date2 = new DateTime("June 20, 2019");  
echo getDays($date1, $date2); // 6
```

```
$date1 = new DateTime("December 29, 2018");  
$date2 = new DateTime("January 1, 2019");  
echo getDays($date1, $date2); // 3
```

```
$date1 = new DateTime("July 20, 2019");  
$date2 = new DateTime("July 30, 2019");  
echo getDays($date1, $date2); // 10
```

```
function getDays(date1, date2) {  
    const diffInMilliseconds = Math.abs(date2 - date1);  
    const diffInDays = Math.ceil(diffInMilliseconds / (1000 * 6  
    return diffInDays;  
}
```

// Contoh penggunaan

```
const date1 = new Date("June 14, 2019");  
const date2 = new Date("June 20, 2019");  
console.log(getDays(date1, date2)); // 6
```

```
const date3 = new Date("December 29, 2018");  
const date4 = new Date("January 1, 2019");  
console.log(getDays(date3, date4)); // 3
```

```
const date5 = new Date("July 20, 2019");  
const date6 = new Date("July 30, 2019");  
console.log(getDays(date5, date6)); // 10
```

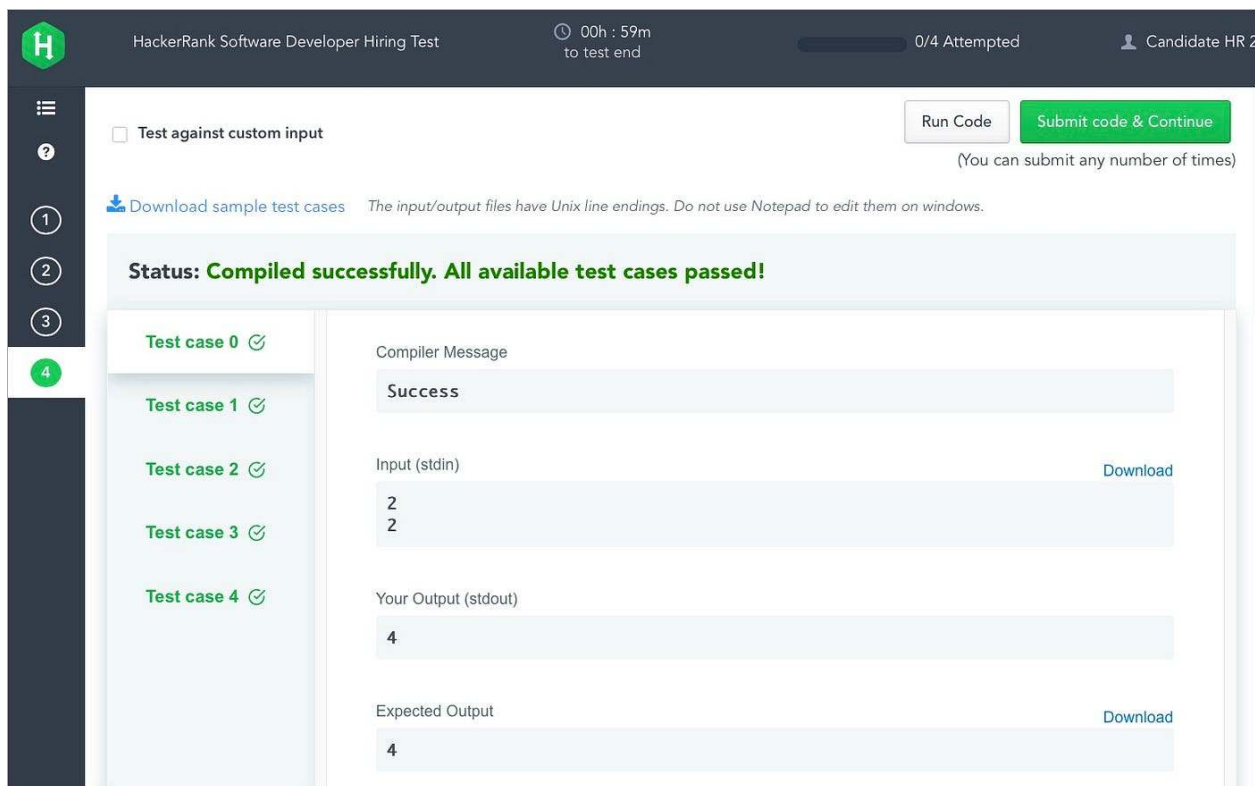
Chapter 9: Mempresentasikan hasil coding test

Sebuah hasil coding test perlu dipresentasikan dengan baik dan jelas, agar maksud dan tujuan dari solusi yang kamu buat, bisa tersampaikan

dengan baik kepada tester coding test yang kamu lakukan. Berikut ini merupakan poin-poin yang perlu kalian perhatikan

Mempersiapkan solusi yang sudah dibuat sebelumnya

Sebelum mempresentasikan hasil dari coding test, penting untuk memastikan bahwa solusi yang telah disiapkan telah melalui tahap uji dan validasi. Hal ini termasuk memastikan bahwa kode telah diuji terhadap berbagai kasus uji yang telah ditetapkan sebelumnya.



Buat penjelasan yang singkat, padat, dan jelas dalam bentuk catatan singkat

Dalam mempresentasikan solusi, penting untuk mempresentasikannya dengan cara yang mudah dipahami. Gunakan catatan singkat yang

mengandung keypoints dari solusi yang diusulkan. Pastikan catatan tersebut memberikan gambaran yang jelas tentang algoritma dan approach yang digunakan oleh kamu.

Contoh penjelasan jawaban coding test:

Langkah-langkah dalam algoritma bubble sort adalah:

1. Melakukan pengulangan melalui seluruh array dimulai dari
2. Jika angka dalam array pada indeks $i-1$ lebih besar dari i , t
3. Setelah mencapai akhir array, mulai pengulangan kembali
4. Setelah tidak ada elemen lagi yang dapat ditukar, penguru

```
function swap(arr, i1, i2) {  
  var temp = arr[i1];  
  arr[i1] = arr[i2];  
  arr[i2] = temp;  
}
```

```
function bubblesort(arr) {
```

```
  var swapped = true;
```

```
  // keep going unless no elements can be swapped anymore
```

```
  while (swapped) {
```

```
    // set swapped to false so that the loop stops
```

```

// unless two element are actually swapped
swapped = false;

// loop through the whole array swapping adjacent elements
for (var i = 1; i < arr.length; i++) {
  if (arr[i-1] > arr[i]) {
    swap(arr, i-1, i);
    swapped = true;
  }
}

return arr;
}

bubblesort([103, 45, 2, 1, 97, -4, 67]);

```

Memastikan bahwa coding test yang dikerjakan memenuhi test case yang sudah ditentukan sebelumnya

Sebelum presentasi, pastikan bahwa solusi yang dihasilkan berhasil memenuhi semua test case yang telah ditetapkan sebelumnya.

Periksa apakah hasil dari solusi sesuai dengan harapan dan tidak ada kegagalan atau error yang terjadi.

Memberikan alternatif solusi yang dibuat

Terkadang, ada lebih dari satu cara untuk menyelesaikan suatu masalah. Dalam presentasi coding test, selain solusi utama, berikan juga beberapa alternatif solusi yang mungkin relevan. Jelaskan keuntungan dan kelemahan dari masing-masing alternatif. Hal ini akan menunjukkan kemampuan untuk berpikir kreatif dan mempertimbangkan berbagai pendekatan untuk menyelesaikan masalah.

Chapter 10: Menerima pekerjaan sebagai website developer

Akhirnya kita masuk kepada chapter menerima pekerjaan pertama sebagai website developer, selamat ya karena kamu telah berjuang dengan keras untuk bisa mencapai tahap ini. Hal pertama yang perlu kita lakukan adalah sebagai berikut:

- Memahami cara perusahaan tersebut bekerja, kita bisa pelajari employee handbook yang telah mereka buat untuk setiap peran karyawan
- Memahami secara menyeluruh proyek pada perusahaan tersebut terkait cara bekerja, penggunaanya siapa saja, dan tersedia fitur penting apa saja

- Mempelajari framework project management yang mereka gunakan misalnya apakah kanban, scrum, atau agile

Beradaptasi dengan tim yang baru

Apabila kita pernah magang atau bekerja sebelumnya lalu saat ini diterima pada perusahaan baru maka kita wajib rendah hati untuk mengenal dan mempelajari kembali bagaimana setiap karyawan pada perusahaan tersebut bekerja.

Kita tidak bisa menggunakan metode yang biasanya kita gunakan dan meminta mereka semua mengikuti diri kita, justru kita yang harus beradaptasi dengan baik dan cepat, maka dari itu sebelumnya saya selalu mengingatkan bahwa skills beradaptasi ini perlu kita latih.



Skills adaptasi bukan cuma dipake untuk mengikuti perkembangan website development terbaru tapi juga ketika bekerja sama di dalam tim yang kecil atau besar.

Luangkan waktu untuk interaksi

Sebagai seorang website developer memang pekerjaan kita lebih banyak di depan laptop, akan tetapi berinteraksi dengan setiap anggota tim juga wajib kita lakukan. Interaksi ini dapat kita lakukan secara bertemu atau juga jarak jauh (online).

Adapun beberapa rekomendasi aktivitas yang bisa kita lakukan bersama mereka:

- Konsultasi online 1-1 terkait pekerjaan yang sedang dikerjakan
- Bermain game online bareng dengan tim misalnya mobile legend
- Mengikuti webinar secara bersama-sama

Poin penting di atas dapat meningkatkan trust dan bonding di dalam sebuah tim sehingga kita bisa mengenal mereka lebih jauh dan apabila sudah cukup akrab maka akan mudah untuk kita beradaptasi dan juga bekerja sama dengan mereka di dalam sebuah proyek.

Mempelajari dokumentasi resmi perusahaan

Sebelumnya saya sudah bahas mengenai project documentation yang berisi details dari proyek yang akan kita kerjakan pada perusahaan tersebut seperti nama, target pengguna, fitur utama, fungsi setiap coding, dan hal penting lainnya.

Kita harus mengikuti seluruh tech stack yang mereka gunakan misalnya PHP dengan framework Laravel dan untuk Frontend menggunakan Vue JavaScript. Apabila kita belum memperdalam kedua stack tersebut maka kita sampaikan kepada mereka untuk mempelajari hal-hal yang belum dipelajari terlebih dahulu lalu mengerjakan task kecil.

Tujuan utamanya adalah agar kita bisa mulai terbiasa dengan cara ngoding mereka, cara mereka melempar task dari tim A kepada tim B, cara update task tersebut pada project management, dan juga sebagainya. Tidak masalah apabila hanya mengerjakan task kecil untuk awal bekerja karena kita perlu menyempurnakan terlebih dahulu workflow perusahaan tersebut.

Responsibilities semakin besar

Ketika tingkatan karir kita sudah tinggi misalnya Senior maka sebagai website developer akan jarang sekali mengerjakan pekerjaan coding. Kita akan lebih sering menghabiskan waktu dalam mengatur developer dan tim lainnya demi kesuksesan projek tersebut.

Pada tingkatan senior ini maka biasanya kita perlu mendvelop beberapa ketangkasan utama dan penting sebagai berikut:

- Belajar memutuskan tech stack apa saja yang perlu digunakan di dalam sebuah projek kecil, menengah, dan besar seperti enterprise.

- Lebih sering mengeksplor hal baru di dunia website development agar dapat memberikan solusi terbaik kepada stakeholder pada perusahaan tersebut.
- Meningkatkan komunikasi dengan developer dan project manager agar projek dapat diselesaikan dengan baik sesuai kriteria dan waktu yang ditentukan.

Kalau tidak siap melakukan hal berikut baiknya bagaimana? well, kita bisa tetap menjadi senior namun meminta untuk tetap mengerjakan kerjaan coding tanpa harus dilibatkan dalam melakukan people management dan timeline project bersama dengan project manager pada umumnya di sebuah perusahaan.

Chapter 11: Beradaptasi dan Persiapan Naik Jabatan

Setelah bekerja sebagai website developer selama lebih dari dua tahun, itu bisa menjadi waktu yang tepat untuk mempertimbangkan kenaikan jabatan. Dalam dunia teknologi, kemajuan karier sering kali didasarkan pada pengalaman, keterampilan yang berkembang, serta kontribusi nyata terhadap perusahaan. Jika kita merasa telah menunjukkan peningkatan dalam kemampuan teknis dan kepemimpinan, maka ini saatnya untuk mulai mempersiapkan diri naik ke tingkat berikutnya, misalnya dari junior ke intermediate atau bahkan ke senior.

Mengapa Kenaikan Jabatan Itu Penting?

Kenaikan jabatan bukan hanya tentang status atau gaji yang lebih tinggi, tetapi juga tentang pertumbuhan profesional dan kesempatan untuk mengambil tantangan yang lebih besar. Beberapa manfaat dari naik jabatan meliputi:

1. **Pengakuan atas Kinerja dan Kontribusi** – Kenaikan jabatan menunjukkan bahwa perusahaan mengakui dan menghargai usaha serta pencapaian kita.
2. **Peluang untuk Meningkatkan Keterampilan** – Dengan jabatan yang lebih tinggi, kita memiliki kesempatan untuk mengembangkan keterampilan baru, termasuk manajemen tim dan pengambilan keputusan strategis.
3. **Meningkatkan Kompensasi** – Posisi yang lebih tinggi biasanya diiringi dengan kenaikan gaji dan manfaat tambahan.
4. **Lebih Banyak Pengaruh dalam Tim dan Perusahaan** – Dengan jabatan yang lebih tinggi, kita bisa lebih berperan dalam menentukan arah teknis dan strategi pengembangan produk.

Langkah-langkah Mempersiapkan Diri untuk Kenaikan Jabatan

Jika kita merasa sudah siap untuk naik jabatan, ada beberapa hal yang bisa dilakukan untuk meningkatkan peluang tersebut:

1. Evaluasi Diri Sendiri

Sebelum mengajukan permintaan naik jabatan, lakukan refleksi diri terlebih dahulu:

- Apakah saya telah mengembangkan keterampilan teknis yang dibutuhkan untuk posisi berikutnya?
- Apakah saya telah menunjukkan kepemimpinan, inisiatif, dan kontribusi yang signifikan?
- Apakah saya sudah memahami tantangan dan tanggung jawab yang lebih besar dari jabatan yang saya incar?

Jika jawaban atas pertanyaan-pertanyaan ini adalah “ya”, maka itu bisa menjadi sinyal bahwa kita siap untuk naik jabatan.

2. Kumpulkan Bukti Kontribusi dan Prestasi

Saat mengajukan kenaikan jabatan, penting untuk memiliki data dan bukti konkret yang menunjukkan nilai kita bagi perusahaan:

- **Proyek-proyek besar yang telah diselesaikan** dan dampaknya terhadap bisnis.
- **Feedback positif dari atasan dan rekan kerja** yang menegaskan kontribusi kita.
- **Peningkatan keterampilan** yang telah diperoleh selama bekerja.
- **Inovasi atau solusi yang kita buat** yang membantu meningkatkan efisiensi atau revenue perusahaan.

3. Diskusikan dengan HR atau Atasan

Setelah mempersiapkan semua bukti dan argumen, kita bisa mulai berdiskusi dengan HR atau atasan langsung mengenai peluang kenaikan

jabatan. Kita bisa menghubungi mereka melalui email, WhatsApp, atau bahkan datang langsung ke ruang kerja mereka.

Berikut adalah contoh kalimat yang bisa digunakan:

“Pak/Bu, saya telah bekerja di perusahaan ini kurang lebih dua tahun dan saat ini posisi saya sebagai [posisi saat ini]. Saya ingin berdiskusi mengenai kemungkinan untuk naik jabatan dari [posisi A] menjadi [posisi B]. Saya ingin mengetahui apakah ada peluang bagi saya untuk mendapatkan peran yang lebih besar berdasarkan kontribusi yang telah saya berikan.”

Dengan menggunakan bahasa yang sopan dan profesional, kita menunjukkan niat baik serta kesiapan untuk mendiskusikan kenaikan jabatan secara konstruktif.

4. Bersikap Sabar dan Terbuka terhadap Evaluasi

Setelah mengajukan permintaan, kita perlu menunggu respons dari HR atau atasan dengan sikap profesional. Biasanya, mereka akan memberikan jawaban seperti:

“Baik, terima kasih atas pertanyaan ini. Kami akan mengevaluasi terlebih dahulu kinerja kamu dan akan memberikan tanggapan setelahnya.”

Jika perusahaan membutuhkan waktu untuk menilai performa kita, gunakan kesempatan ini untuk terus bekerja dengan baik dan membuktikan bahwa kita memang layak untuk promosi.

5. Siapkan Alternatif Jika Permintaan Ditolak

Tidak semua permintaan kenaikan jabatan akan langsung disetujui. Jika permintaan kita ditolak, kita bisa meminta masukan dari HR atau atasan mengenai hal-hal berikut:

- Apa yang perlu saya tingkatkan untuk memenuhi kriteria jabatan berikutnya?
- Kapan bisa dilakukan evaluasi ulang untuk mempertimbangkan kenaikan jabatan?
- Apakah ada kesempatan lain di dalam perusahaan yang lebih sesuai dengan keterampilan saya?

Dengan sikap terbuka terhadap masukan, kita bisa menjadikan ini sebagai peluang untuk terus berkembang dan meningkatkan kompetensi kita.

Kesimpulan

Naik jabatan adalah bagian dari pertumbuhan karier yang alami, tetapi perlu persiapan yang matang agar prosesnya berjalan lancar. Dengan mengevaluasi diri sendiri, mengumpulkan bukti kontribusi, berdiskusi secara profesional dengan HR atau atasan, serta bersikap sabar dalam proses evaluasi, kita bisa meningkatkan peluang untuk mendapatkan posisi yang lebih tinggi. Jika permintaan belum disetujui, gunakan kesempatan ini untuk terus belajar dan membuktikan bahwa kita layak mendapatkan peran yang lebih besar di masa depan.

Chapter 12: Strategi untuk Nego Ulang Gaji Kita

Ada saatnya dalam perjalanan karier di mana kita merasa pantas mendapatkan gaji yang lebih tinggi dibandingkan sebelumnya. Seiring dengan meningkatnya keterampilan dan kontribusi kita terhadap perusahaan, wajar jika kita menginginkan kompensasi yang lebih baik sebagai bentuk apresiasi atas kerja keras dan dedikasi yang telah diberikan.

Perusahaan biasanya melakukan evaluasi kinerja karyawan secara berkala, misalnya setiap tiga bulan atau setiap tahun. Evaluasi ini mencakup aspek seperti cara kita bekerja, performa yang kita tunjukkan, serta sikap dan kontribusi kita dalam tim. Dengan memahami bagaimana perusahaan menilai kita, kita bisa mempersiapkan strategi yang lebih baik untuk menegosiasikan kenaikan gaji.

Mengapa Kita Layak Mendapatkan Kenaikan Gaji?

Sebelum mengajukan permintaan kenaikan gaji, kita harus memiliki pemahaman yang jelas tentang alasan mengapa kita layak mendapatkannya. Beberapa pertanyaan yang perlu kita tanyakan pada diri sendiri:

1. **Sudah berapa lama saya bekerja di perusahaan ini?** – Jika kita sudah bekerja lebih dari satu tahun dan terus menunjukkan peningkatan, maka ini bisa menjadi alasan kuat untuk negosiasi.
2. **Dampak apa yang telah saya berikan kepada perusahaan?** – Apakah kita telah membantu meningkatkan efisiensi tim,

meningkatkan omset, atau menciptakan solusi yang berdampak besar bagi operasional perusahaan?

3. **Bagaimana perkembangan keterampilan saya?** – Jika kita telah mengembangkan keterampilan baru yang memberikan nilai tambah bagi perusahaan, ini bisa menjadi faktor penting dalam negosiasi gaji.

Strategi Efektif untuk Menegosiasikan Kenaikan Gaji

Jika kita yakin bahwa kita layak mendapatkan kenaikan gaji, berikut beberapa strategi yang dapat membantu dalam proses negosiasi:

1. Kumpulkan Data dan Bukti Kontribusi

Sebelum bertemu dengan atasan atau HRD, persiapkan semua data yang mendukung permintaan kita. Beberapa hal yang bisa dikumpulkan antara lain:

- Laporan kinerja (KPI) yang menunjukkan peningkatan produktivitas.
- Proyek-proyek penting yang telah diselesaikan dan dampaknya terhadap perusahaan.
- Feedback positif dari rekan kerja atau atasan yang menguatkan kontribusi kita.
- Perbandingan gaji dengan standar industri untuk posisi yang sama di perusahaan lain.

2. Pilih Waktu yang Tepat untuk Bernegosiasi

Memilih momen yang tepat sangat penting dalam negosiasi gaji.

Beberapa waktu yang ideal untuk mengajukan permintaan ini antara lain:

- Setelah menyelesaikan proyek besar yang berdampak signifikan.
- Saat perusahaan sedang dalam kondisi finansial yang stabil dan berkembang.
- Selama periode evaluasi kinerja tahunan atau kuartalan.
- Setelah mendapatkan tanggung jawab tambahan yang sebelumnya tidak termasuk dalam peran kita.

3. Ajukan Permintaan dengan Pendekatan Profesional

Setelah data siap dan waktu sudah tepat, kita bisa menghubungi HRD atau langsung ke pemilik bisnis jika perusahaan tidak memiliki divisi HRD. Saat melakukan negosiasi:

- Gunakan bahasa yang sopan dan profesional.
- Jelaskan alasan kenaikan gaji berdasarkan kontribusi dan data yang telah dikumpulkan.
- Hindari membandingkan diri dengan rekan kerja secara langsung.
- Berikan angka yang realistis berdasarkan riset pasar dan standar industri.

4. Tunjukkan Kesiapan untuk Terus Berkembang

Perusahaan cenderung lebih terbuka terhadap kenaikan gaji jika melihat bahwa karyawannya memiliki komitmen untuk terus berkembang. Kita bisa menegaskan bahwa kita:

- Bersedia mengambil tanggung jawab lebih besar.
- Terus belajar dan meningkatkan keterampilan untuk memberikan dampak lebih besar.
- Memiliki visi jangka panjang dalam perusahaan dan ingin terus berkontribusi.

5. Siapkan Alternatif Jika Permintaan Ditolak

Tidak semua negosiasi gaji langsung berhasil. Jika permintaan kita ditolak, jangan langsung kecewa. Sebaliknya, tanyakan:

- Apa yang perlu ditingkatkan agar bisa mendapatkan kenaikan gaji di masa depan?
- Apakah ada kompensasi lain yang bisa diberikan, seperti bonus atau tunjangan tambahan?
- Kapan bisa dilakukan evaluasi ulang untuk mempertimbangkan kenaikan gaji?

Dengan memiliki rencana cadangan, kita tetap bisa menjaga hubungan baik dengan perusahaan dan meningkatkan peluang mendapatkan kenaikan gaji di masa depan.

Kesimpulan

Negosiasi gaji adalah langkah penting dalam perjalanan karier kita. Dengan persiapan yang matang, bukti kontribusi yang jelas, dan pendekatan yang profesional, kita dapat meningkatkan peluang untuk mendapatkan kenaikan gaji yang pantas. Ingatlah bahwa negosiasi gaji bukan hanya tentang meminta lebih banyak uang, tetapi juga tentang menghargai diri sendiri dan memastikan bahwa kita mendapatkan kompensasi yang sesuai dengan nilai yang kita berikan kepada perusahaan.



Kita tidak bisa hanya asal-asalan minta kenaikan gaji tanpa membuktikan sesuatu yang berdampak besar kepada perusahaan tempat kita bekerja, terus tekankan poin tersebut sehingga perusahaan lebih sadar kepada keahlian diri kita.

Chapter 13: Membangun Sumber Passive Income sebagai Website Developer

Sebagai seorang website developer, kita memiliki kesempatan besar untuk membangun berbagai sumber penghasilan pasif dengan modal yang relatif kecil. Kemampuan kita dalam pemrograman dapat dimanfaatkan untuk menciptakan produk digital, layanan berbasis langganan, atau konsultasi yang dapat memberikan pemasukan secara

terus-menerus. Banyak bisnis yang sukses dibangun oleh para developer hanya dengan modal awal yang terjangkau, seperti:

- **Domain:** Rp 120.000 per tahun
- **Hosting awal:** Rp 80.000 per bulan
- **Skill ngoding:** Rp 0 (karena sudah dimiliki)

Dengan modal ini, ada beberapa cara efektif untuk menciptakan passive income sebagai website developer.

1. Menjual Produk Digital

Salah satu cara paling umum untuk menghasilkan passive income adalah dengan menjual produk digital. Sebagai seorang developer, kita bisa membuat berbagai produk seperti:

- **Template HTML, PHP, atau WordPress** – Produk ini bisa dijual di berbagai marketplace seperti **UI8**, **Envato Market** (**ThemeForest**, **CodeCanyon**), atau **Creative Market**. Setiap kali ada seseorang yang membeli template kita, kita mendapatkan penghasilan tanpa harus bekerja ulang.
- **Plugin atau script siap pakai** – Banyak orang membutuhkan fitur tambahan untuk website mereka. Kita bisa menjual plugin untuk WordPress, WooCommerce, atau script berbasis PHP dan JavaScript.
- **Aplikasi mobile atau web sederhana** – Aplikasi dengan fitur tertentu, seperti to-do list atau kalkulator keuangan, bisa dijual di platform seperti **Gumroad** atau **Itch.io**.

Keuntungan dari model bisnis ini adalah bahwa produk hanya perlu dibuat sekali, tetapi bisa dijual berulang kali kepada banyak pelanggan tanpa batas.

2. Membangun Layanan SaaS (Software as a Service)

Membangun layanan berbasis langganan (SaaS) adalah langkah berikutnya untuk menciptakan passive income. SaaS memungkinkan pelanggan menggunakan layanan berbasis web dengan membayar biaya langganan bulanan atau tahunan. Contohnya:

- **Layanan invoicing untuk freelancer** – Seperti **Hello Bonsai**, yang membantu pekerja lepas mengelola keuangan dan kontrak mereka.
- **Sistem reservasi online** – Layanan untuk restoran, salon, atau konsultasi online yang memudahkan pelanggan untuk melakukan pemesanan.
- **Monitoring dan analitik website** – Layanan seperti **UptimeRobot** yang memberi tahu pemilik website ketika situs mereka mengalami downtime.

Keunggulan model bisnis ini adalah penghasilan yang berulang (recurring revenue) dari pelanggan yang terus menggunakan layanan kita.

3. Menyediakan Layanan Konsultasi dan Edukasi

Jika kita sudah berpengalaman di dunia website development, kita juga bisa menghasilkan passive income dengan berbagi pengetahuan kita. Beberapa cara yang bisa dilakukan antara lain:

- **Membuat kursus online** – Kita bisa membuat kursus pemrograman dan menjualnya di platform seperti **Udemy, Skillshare, atau Teachable**.
- **Webinar dan kelas berbayar** – Mengadakan sesi pelatihan online dengan model berlangganan atau sekali bayar.
- **Menulis eBook** – Jika lebih suka menulis, kita bisa membuat eBook tentang topik tertentu dalam pemrograman dan menjualnya di **Amazon Kindle, Gumroad, atau Leanpub**.
- **Membangun komunitas berbayar** – Seperti Discord atau Telegram dengan sistem membership untuk memberikan akses eksklusif ke materi premium.

Model bisnis ini tidak hanya memberikan passive income tetapi juga meningkatkan kredibilitas kita sebagai developer.

4. Monetisasi Blog atau Website

Jika kita suka menulis, kita bisa membangun blog teknis dan memonetisasinya dengan berbagai cara seperti:

- **Iklan Google AdSense** – Memasang iklan di website kita dan mendapatkan bayaran setiap kali ada yang mengklik iklan tersebut.
- **Afiliasi produk digital** – Memasarkan tools atau layanan yang relevan dengan pengembangan web, seperti hosting atau software, dan mendapatkan komisi dari setiap pembelian.
- **Menjual membership premium** – Menyediakan konten eksklusif bagi anggota yang membayar langganan.

Chapter 14: Giving Back to the Community

Sebagai seorang profesional di bidang website development, saya selalu menyempatkan waktu untuk berbagi ilmu dan pengalaman dengan komunitas di Indonesia. Saya melakukannya melalui berbagai cara, seperti mengadakan kelas gratis, memberikan webinar, mengorganisir meetup offline, dan berkontribusi dalam berbagai diskusi teknis.

Mengapa Harus Berbagi dengan Komunitas?

Berbagi ilmu bukan hanya bermanfaat bagi orang lain, tetapi juga bagi diri kita sendiri. Beberapa alasan utama mengapa berbagi dengan komunitas sangat penting antara lain:

1. **Meningkatkan Personal Branding** – Dengan berbagi ilmu, kita akan lebih dikenal dalam industri dan membangun reputasi sebagai seorang ahli di bidang kita.
2. **Memperdalam Pemahaman** – Ketika kita mengajar atau menjelaskan sesuatu kepada orang lain, kita sebenarnya sedang mengulang kembali apa yang telah kita pelajari. Ini membantu memperkuat pemahaman kita sendiri.
3. **Mempelajari Hal Baru** – Saat berinteraksi dengan komunitas, kita sering menemukan perspektif baru dan belajar tentang teknologi atau pendekatan yang belum kita ketahui sebelumnya.

4. **Membangun Jaringan Profesional** – Dengan aktif dalam komunitas, kita akan bertemu dengan banyak orang dari berbagai latar belakang yang bisa menjadi rekan kerja, mentor, atau bahkan klien di masa depan.
5. **Mendapatkan Kepuasan Pribadi** – Berbagi ilmu memberikan kebahagiaan tersendiri karena kita tahu bahwa apa yang kita sampaikan bermanfaat bagi banyak orang.



Berbagi ilmu tidak akan merugikan diri sendiri atau menciptakan banyak saingan. Justru, semakin banyak kita berbagi, semakin banyak kita belajar dan semakin besar dampak positif yang bisa kita berikan.

Membangun Project Open-Source

Jika kita belum siap untuk berbicara di depan publik atau mengajar, salah satu cara lain untuk "giving back to the community" adalah dengan membangun proyek open-source. Proyek open-source memungkinkan banyak orang menggunakan dan berkontribusi pada software yang kita buat tanpa biaya.

Misalnya, kita bisa memilih komunitas tertentu, seperti **Web Developer Indonesia**, lalu membuat proyek yang membantu mereka bekerja lebih efisien. Sebagai contoh, kita bisa mengembangkan **framework CSS ringan** yang dapat digunakan dalam berbagai proyek.

Dengan berkontribusi dalam open-source, kita akan dikenal sebagai seseorang yang **profesional dan rendah hati**. Dalam jangka panjang, hal ini bisa menjadi keuntungan besar, karena recruiter atau perusahaan sering mencari kandidat dengan kontribusi nyata di dunia open-source.

Berikut beberapa ide proyek open-source yang bisa dibuat:

- **Template website untuk backend** – Membantu developer backend memulai proyek lebih cepat.
- **CMS website sederhana** – Alternatif ringan untuk WordPress atau Joomla.
- **API cek ongkir ekspedisi** – Berguna bagi pengembang yang membangun aplikasi e-commerce.
- **Plugin WordPress** – Menambah fitur baru yang berguna untuk pengguna WordPress.
- **Data Visualization Library** – Membantu developer menampilkan data dengan lebih menarik dan interaktif.

Mulailah meluangkan waktu dari pekerjaan utama atau freelance untuk membangun proyek kecil. Dengan begitu, kita bisa terus **kreatif dan inovatif**, sesuatu yang sangat diperlukan untuk tetap kompetitif di dunia teknologi.



Kamu tidak ingin digantikan oleh orang yang lebih hebat, bukan? Maka dari itu, teruslah belajar dan berbagi ilmu!

Berbagi Tips Melalui Artikel

Jika kamu lebih suka menulis daripada berbicara, maka berbagi ilmu melalui artikel adalah cara yang tepat untuk memberikan manfaat jangka panjang. Dengan menulis artikel tentang website development, kita bisa membantu banyak orang sambil membangun reputasi sebagai ahli di bidang ini.

Setiap artikel yang kita tulis bisa mencantumkan profil lengkap agar mudah dikenali oleh calon klien, rekan kerja, atau perusahaan besar yang mungkin membutuhkan jasa kita. Artikel yang baik dapat membuka banyak peluang karier dan bisnis.

Contoh topik artikel yang bisa ditulis:

- **Cara membuat desain responsif yang optimal di berbagai perangkat**
- **Teknik meningkatkan performa website agar tetap cepat meskipun dikunjungi jutaan pengguna**
- **Best practices dalam pengembangan API untuk aplikasi modern**
- **Strategi SEO terbaru untuk meningkatkan peringkat website di mesin pencari**



Terkadang, apa yang sudah kita anggap biasa bisa menjadi sesuatu yang sangat berharga bagi orang lain. Jadi, jangan ragu untuk mulai berbagi ilmu, sekecil apa pun itu!

Chapter 15: Memanfaatkan AI dalam Pekerjaan sebagai Web Developer

Di dunia website development, bikin website yang berkualitas itu nggak cuma soal nulis kode doang. Ada banyak aspek yang harus diperhatiin, mulai dari memahami kebutuhan user, bikin desain yang interaktif, mastiin responsive di berbagai perangkat, sampai optimasi SEO biar website bisa muncul di search engine.

Nah, AI ini bisa banget bantu kita buat ngejalanin semua itu dengan lebih gampang dan efisien. AI tuh bisa bantu kita coding, debugging, testing, bahkan kasih rekomendasi solusi buat masalah yang kita hadapi.

AI Tools yang Bisa Bantu Web Developer

Sekarang udah banyak AI tools yang bisa kita manfaatin buat ngeboost produktivitas dan efisiensi kerja. Berikut beberapa yang paling populer dan berguna buat web developer:

1. ChatGPT

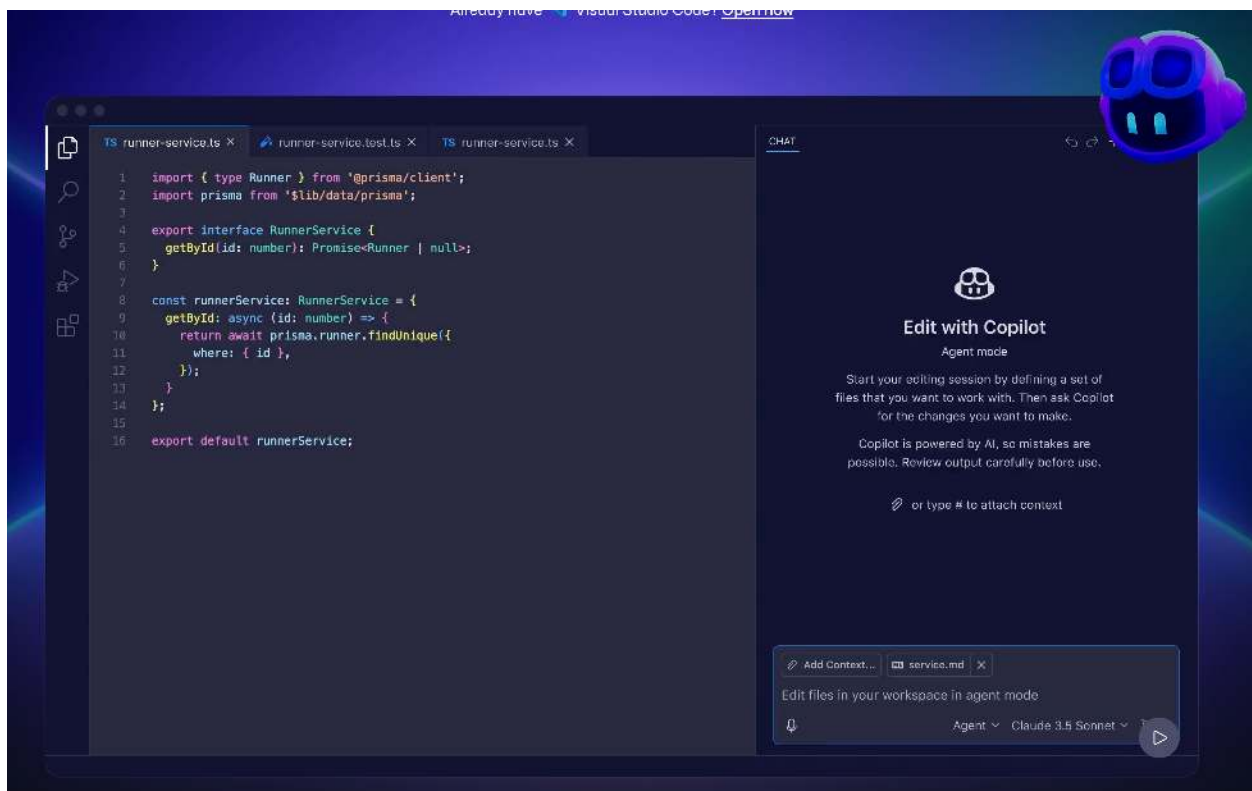
Kita semua udah familiar sama ChatGPT, kan? AI ini bisa bantu kita brainstorming ide, generate kode sederhana, ngejelasin konsep yang rumit, bahkan kasih saran buat debugging. Dengan ChatGPT, kita bisa

ngurangi human error, hemat waktu, dan tetap bisa eksplorasi kreativitas kita dalam coding.

Manfaat utama ChatGPT untuk web developer:

- Bantu generate snippet kode sesuai kebutuhan.
- Ngebantu nge-scan kode buat cari error dan kasih solusi.
- Bisa bantu otomatisasi unit testing buat quality assurance kode kita.

2. GitHub Copilot



Buat yang sering pakai GitHub, ada fitur keren bernama GitHub Copilot. Ini kayak asisten coding yang bisa kasih rekomendasi kode secara otomatis berdasarkan apa yang kita ketik.

Keunggulan GitHub Copilot:

- Bisa generate kode berdasarkan input yang kita kasih.
- Ngasih saran fungsi atau kode langsung di text editor.
- Bantu belajar library atau framework yang belum familiar.
- Bisa dipakai di berbagai text editor kayak VS Code.

3. Claude

Claude adalah AI dari Anthropic yang bisa kamu pakai buat bantu coding kamu, kemampuannya dalam memahami konteks kode yang lebih dalam dibandingkan AI lain bikin AI ini jadi favorit untuk dipake buat coding, terutama model barunya yaitu Claude 3.7 Sonnet (per 5 April 2025)

Fitur unggulan Claude:

- Bisa ngobrol dengan lebih natural dan kontekstual.
- Mampu memahami dan menganalisis kode dengan lebih akurat.
- Cocok buat debugging dan refactoring kode.
- Bisa bantu generate kode yang lebih optimal.

Tren terbaru di dunia AI Coding, Vibe Coding

Pernah denger istilah **vibe coding**? Ini tuh salah satu metode coding yang manfaatin AI buat nulis kode berdasarkan instruksi dalam bahasa

natural. Jadi, kita cukup bilang "Tambahin halaman login dengan autentikasi Google dan email," terus AI langsung generate kodenya.

Beberapa tools AI yang sudah menerapkan konsep vibe coding salah satunya adalah **Cursor dan Lovable**.

Cursor (cursor.com)

Cursor adalah editor yang sudah termasuk AI, yang didesain khusus buat ningkatkan produktivitas developer. Cursor ini memungkinkan kita untuk berinteraksi dengan AI langsung di dalam editor, sehingga kita bisa meminta saran kode, melakukan debugging, atau bahkan menulis ulang kode dengan lebih efisien.

Keunggulan Cursor:

- Bisa memahami kode yang sedang kita tulis dan memberikan saran yang relevan.
- Membantu refactoring kode agar lebih rapi dan efisien.
- Bisa dipakai untuk menulis dokumentasi kode secara otomatis.

Lovable (lovable.dev)

Lovable adalah AI coding assistant yang dirancang untuk membantu developer bekerja lebih cepat dengan menggunakan pendekatan "vibe coding." Tool ini memungkinkan kita untuk memberikan instruksi dalam

bahasa natural, lalu AI akan menerjemahkannya ke dalam kode yang siap digunakan.

Fitur unggulan Lovable:

- Mendukung berbagai bahasa pemrograman populer.
 - Bisa memahami instruksi kompleks dan menghasilkan kode berkualitas tinggi.
 - Memungkinkan developer untuk fokus pada ide dan logika tanpa terlalu banyak terjebak dalam syntax.
 - Membantu meningkatkan efisiensi dalam pengembangan produk dengan minim effort.
-

Vibe coding punya **beberapa kelebihan** yang bikin kerjaan kita jadi lebih gampang.

1. Pertama, aksesnya lebih luas—kita nggak perlu jago coding, cukup tahu konsepnya aja, dan AI bisa bantu generate kode yang kita butuhin.
2. Selain itu, vibe coding bisa ningkatin efisiensi kerja karena AI bakal ngerjain bagian-bagian repetitif, jadi kita bisa lebih fokus ke hal yang lebih penting.
3. Yang paling keren, metode ini bikin kita bisa lebih mikirin logika bisnis daripada sekadar ngurusin syntax terus-terusan.

Tapi, vibe coding juga punya **beberapa kelemahan** yang perlu diperhatiin.

1. Kadang, kode yang dihasilkan AI butuh perbaikan manual supaya bisa berjalan dengan optimal.
2. Selain itu, vibe coding bisa menyulitkan iterasi dan perubahan kompleks, karena AI belum tentu memahami konteks kode yang lebih dalam seperti developer manusia.
3. Terakhir, ada risiko kita jadi terlalu bergantung pada AI, yang bisa berdampak buruk kalau kita sendiri nggak memahami fundamental coding dengan baik.

Intinya, AI bisa bantu kita coding lebih cepat, tapi tetep harus ngerti fundamentalnya biar hasilnya tetap berkualitas.

Kesimpulan

AI udah jadi bagian penting dalam dunia development, termasuk buat kita para web developer. Dengan tools seperti ChatGPT, GitHub Copilot, dan Claude AI, kita bisa kerja lebih cepat, lebih efisien, dan lebih fokus ke inovasi. Tapi ingat, AI itu alat bantu, bukan pengganti skill coding kita. Jadi, tetap belajar, eksplor, dan manfaatin AI dengan bijak biar kerjaan makin lancar! 🚀

Bonus Chapter 15: Contoh Prompt Untuk Web Developer

Prompt di bawah ini dapat kalian gunakan untuk bertanya soal coding ke tools AI yang kamu gunakan seperti ChatGPT, Cursor, dan Claude.

Tips Memakai Prompt

Hasil yang kamu dapat dari AI bergantung pada seberapa baik kamu memberikan konteks, instruksi, dan panduan. Semakin jelas dan detail input yang kamu berikan, semakin baik juga output yang dihasilkan.

Berikut beberapa tips untuk mendapatkan hasil yang lebih optimal:

Pecah jadi beberapa langkah

Coba bagi prompt kamu ke dalam beberapa tahap. Misalnya, daripada langsung meminta AI untuk menulis dan merapikan kode dalam satu prompt, lebih baik minta review dulu, lalu minta refactor berdasarkan review tersebut. Cara ini biasanya menghasilkan output yang lebih baik dibandingkan dengan prompt yang menggabungkan banyak permintaan sekaligus.

Berikan contoh

Sertakan contoh input dan output yang kamu harapkan. Ini akan membantu AI memahami lebih baik apa yang kamu inginkan dan meningkatkan akurasi hasilnya.

Jelaskan dengan spesifik

Jangan ragu untuk menuliskan secara detail apa yang kamu butuhkan, apa yang sudah kamu ketahui, dan hal apa saja yang tidak perlu

dimasukkan. Semakin spesifik, semakin tepat hasilnya.

Minta AI untuk merefleksi jawabannya

Teknik bernama *reflexion* terbukti bisa meningkatkan akurasi AI.

Caranya? Minta AI untuk mengevaluasi jawabannya sendiri, misalnya dengan bertanya, "*Kenapa jawaban ini bisa salah?*" atau meminta revisi berdasarkan analisisnya sendiri.

Dengan pendekatan yang lebih terarah, kamu bisa memanfaatkan AI dengan lebih efektif dan mendapatkan hasil yang lebih akurat!

Dokumentasi / Penjelasan

Buat Dokumentasi

Prompt 1#:

I don't know how to code, but I want to understand how it works. Explain the following code to me in a way that a non-technical person can understand. Always use Markdown with nice formatting to make it easier to follow. Organize it by sections with headers. Include references to the code as markdown code blocks in each section. The code:

[insert code here]

Prompt 2#:

Please add comprehensive documentation for [nama modul atau file], including clear and concise explanations of its purpose, design, and implementation. Consider including examples of how to use the module, as well as any relevant diagrams or flow charts to help illustrate its workings. Ensure that the documentation is easily accessible to other developers and is updated as the module evolves. Consider using documentation tools such as inline comments, markdown files, or a documentation generator to simplify the process.

[masukan code disini]

Menulis Syarat dan Ketentuan

Prompt:

Create terms and services for my website about an [bla bla] called [nama].

Membuat Cheat Sheet

Prompt:

Write a cheat sheet for [markdown formatting].

Membuat File README

Prompt:

Generate documentation for the code below. You should include detailed instructions to allow a developer to run it on a local machine, explain what the code does, and list vulnerabilities that exist in this code.

[masukkan code]

Menulis Blog Mendetail

Prompt:

Write a detailed blog on How to build a [nama aplikasi] using [nama stack] with proper structuring of code.

Menjelaskan Kode

Prompt:

Context: I'm starting a new position as backend developer and I have to start to understand how some functions are

e working

Technologies: [tuliskan stacknya]

You have to: explain me the code line by line

[masukkan codenya disini]

Diagram Arsitektur (Mermaid)

Prompt:

Write the Mermaid code for an architecture diagram for this solution [jelaskan solusinya]

Contoh:

```
graph TD;
A[Client] -->|HTTP Request| B(API Gateway);
B -->|HTTP Request| C[Service 1];
B -->|HTTP Request| D[Service 2];
C -->|Database Query| E[Database];
D -->|Database Query| E;
```

Entity Relationship Diagram (Mermaid)

Prompt:

Write the Mermaid code for an entity relationship diagram for these classes [masukkan code classnya, bisa migration atau model laravel]

Refactor Kode

Refactor Kode

Prompt:

I have a piece of code and I need you do a refactor of it:

[masukkan codenya]

Modernisasi Kode Lama

Prompt:

Refactor the following code to modern es6 programming standards:

[masukkan codenya]

Membagi Kode ke dalam Beberapa Metode

Prompt:

Refactor the following code into multiple methods to improve readability and maintainability:

[masukkan codenya]

Meningkatkan Performance

Prompt:

Refactor the following code to improve performance:

[masukkan codenya]

Menambahkan Parameter ke Function

Prompt:

Add a parameter to this function to do [fungsionalitas]

[masukkan codenya]

Mengikuti Panduan Code Style / Standar Code

Prompt:

Review the following code and refactor it to make it more DRY and adopt the SOLID programming principles.

[masukkan codenya]

Cek Bug dan Error

Prompt 1#:

Review this code for errors and refactor to fix any issues:

[masukkan codenya]

Prompt 2#:

I'm developing software in [masukkan stacknya] and I need you help me to find and fix all the errors in my code, following the best practices. I'll provide you my code and you'll give me the code with all the corrections explained line by line

Prompt 3#:

I wrote this code
[kodenya]

I got this error
[errornya]

How can I fix it? or What does this error mean?

Debug Coding di Component

Prompt:

Please find and fix the bug in the [nama component] component that is causing [jelasin isunya].

[masukkan codenya]

Membuat Unit Test

Prompt 1#:

Please write unit tests for [nama modul atau filenya] to ensure its proper functioning

[masukkan codenya]

Prompt 2#:

Create 2 unit tests for the provided code. One for a successful condition and one for failure.

Menulis RegEx

Prompt:

Write a regular expression that matches / Write a RegEx pattern for:

[REQUEST]

What's next?

Tentunya, proses belajar tidak hanya berhenti sampai sini saja. Ada beberapa langkah yang dapat kamu lakukan secara berkala setelah membaca petunjuk pro ini.

▼ Mengasah skill coding secara berkala

Skill coding perlu diasah agar kemampuan kita dalam coding bisa lebih baik lagi. Banyak cara untuk mengasah skill coding, salah satunya adalah dengan mengerjakan project, baik itu real project, maupun bukan. Dengan mengetahui alur dari suatu project, maka kita akan bisa mengetahui pola dari setiap coding pada project, yang tentunya ini nantinya akan membuat pengerjaan codingmu menjadi lebih efisien

▼ Meningkatkan ilmu dengan belajar dari berbagai referensi

Ilmu coding adalah ilmu yang terus berkembang, dan tentunya hal ini menuntut kita harus meningkatkan skill kita secara berkala.

Tentunya, banyak cara yang dapat kita lakukan untuk meningkatkan ilmu kita, salah satunya adalah dengan mempelajari dokumentasi, tutorial, ebook, dan kelas online.

Kelas online sendiri, dapat menjadi opsi untuk kalian yang ingin mempelajari secara mendalam sebuah materi, baik itu framework, bahasa pemrograman, tools, dan lainnya.

Saya, sebagai mentor coding, memiliki beberapa kelas online yang dapat kalian pelajari, mulai dari front-end sampai back-end. Jika kamu ingin mempelajari ilmu dari kelas online yang saya miliki, kalian dapat melihat materi yang saya ajarkan pada tautan ini:

Profile mentor Galih Pratama di BuildWithAngga



Belajar design dan code di bidang IT langsung dari mentor Galih Pratama yang sangat berpengalaman pada bidangnya di BuildWithAngga.



<https://buildwithangga.com/mentor/galpratama>

Selain dari kelas online, tentunya kalian dapat mempelajari ilmu secara langsung dari dokumentasi. Biasanya, dokumentasi ini akan menjadi referensi kita secara berkala saat kita melakukan coding.

▼ Berdiskusi dengan rekanan satu profesi

Selain mempelajari ilmu, tentunya *networking* menjadi salah satu hal penting yang perlu dilakukan oleh seorang developer. Kita bisa mendapatkan insight lebih dari bersosialisasi dan berdiskusi bersama.

Student dari Petunjuk Pro ini pun, dapat bersosialisasi dengan sesama student, bahkan mentor sekalipun, dan hal ini dapat dimanfaatkan oleh student untuk dapat menambah wawasan lainnya.

Penutup dari mentor

Akhir kata dari saya, Galih Pratama selaku mentor **Petunjuk Pro: Mastering Career Preparation for Website Developer**, mengucapkan terima kasih yang sebesar-besarnya kepada student yang sudah mempercayai saya menjadi mentor kalian.

Petunjuk Pro ini tidak luput dari kekurangan. Maka itu, saya selaku mentor dan tim BuildWithAngga menerima kritik dan saran yang konstruktif dari para student, agar Petunjuk Pro ini bisa lebih baik lagi.

Mari kita support mentor dan creator dengan tidak membajak dan membeli konten secara illegal. Dengan membeli konten secara langsung di BuildWithAngga, kalian akan mendukung untuk kemajuan creator dan mendukung agar konten akan jauh lebih baik lagi.

Akhir kata, selamat berjumpa di kesempatan lain. Thank you! ❤️