

Full Stack Developer Career Path

➤ Full Stack Web / Mobile Developer

Merupakan pengembangan seluruh aplikasi secara end-to-end, dari sisi depan (front-end) hingga sisi belakang (back-end) dan dalam beberapa kasus terdapat dari sisi klien (client-side).

- ✓ Front-End Development
Membangun User Interface menggunakan HTML, CSS dan JavaScript. Menggunakan framework seperti React, Angular, Vue.js atau JQuery.
- ✓ Back-End Development
Membangun server dan aplikasi yang berfungsi sebagai otak dari aplikasi. Bertanggung jawab untuk mengelola data, berinteraksi dengan database, dan menjalankan logika bisnis. Bahasa pemrograman server-side yang biasa dipakai seperti Node.js, Python, Ruby, Java, PHP, atau C#.
- ✓ Database Management
Desain dan Kelola basis data untuk menyimpan, mengambil dan memanipulasi data aplikasi. Teknologi database yang digunakan seperti MySQL, PostgreSQL, MongoDB, atau Firebase.

Cakupan penting dalam Full Stack Development yaitu :

- ✓ Integrasi antara Front-End dan Back-End
Untuk berkomunikasi dengan server dan database, komponen front-end dan layanan back-end dihubungkan melalui API (Application Programming Interface).
- ✓ Version Control and Collaboration
Untuk mengelola perubahan kode dan kolaborasi dalam tim pengembang, dan memastikan mampu bekerja sama yaitu menggunakan sistem pengendalian versi seperti Git.
- ✓ Mobile Development
Beberapa pengembang full stack juga memiliki kemampuan untuk mengembangkan aplikasi berbasis mobile, menggunakan framework seperti React Native, atau Flutter.

Dasar – dasar Front-End Web Development

Pengembangan ini juga dikenal sebagai pengembangan sisi klien, dengan HTML, CSS dan JS dapat membuat situs web atau aplikasi web sehingga user dapat melihat dan berinteraksi secara langsung.

- ✓ HTML (Hypertext Markup Language) → elemen dasar untuk membangun situs web, digunakan untuk membuat dan mengatur struktur konten pada halaman web.
- ✓ CSS → digunakan untuk menata halaman web
- ✓ JavaScript → digunakan untuk membuat web lebih interaktif.

Framework yang populer digunakan untuk front-end yaitu React, Vue.js, dan Angularjs.

Dasar – dasar Back-End Development

- ✓ Bahasa pemrograman server-side → Node.js (JS), Python, Ruby, Java, C#, dan lain lain.
- ✓ Server framework → Express.js untuk Node.js, Flask untuk Python, Ruby on Rails untuk Ruby, Spring untuk Java, dan Laravel untuk PHP.

- ✓ Database Management → SQL (MySQL, PostgreSQL, SQL Server), dan NoSQL (MongoDB, Firebase)

Dasar – dasar Database Management

Database merupakan bagian kritis dari aplikasi karena menyimpan dan mengorganisir informasi yang diperlukan.

- ✓ DBMS yaitu perangkat lunak untuk mengelola dan mengakses data dalam database
- ✓ Tipe Database ada 2 yaitu SQL (relasional) dan No SQL (non-relasional). Bahasa query yang digunakan SQL, contohnya seperti SELECT, INSERT, UPDATE DELETE, dan sebagainya.

DBMS yang populer digunakan SQL yaitu Oracle, PostgreSQL, MySQL. Sedangkan untuk NoSQL ada MongoDB dan redis.

Dasar – dasar Mobile Development

Aplikasi mobile dapat dikembangkan untuk berbagai platform yaitu Android, iOS dan Windows Phone. Setiap platform memiliki Bahasa pemrograman dan lingkungan yang khas. Terdapat perangkat lunak untuk mengembangkan aplikasi mobile disebut IDE (Integrated Development Environment), setiap platform memiliki IDE yang berbeda. Android menggunakan Android studio sedangkan iOS menggunakan Xcode.

➤ Skillset Full Stack Web / Mobile Developer

Pengembangan aplikasi End-to-end → pendekatan pengembangan PL yang mencakup keseluruhan siklus pembuatan aplikasi, dari tahap perencanaan, pengujian, hingga implementasi. Bertujuan untuk menghasilkan apk yang lengkap, fungsional dan siap digunakan end user.

Tahap pengembangan aplikasi end-to-end

1. Perencanaan dan Analisis → Mengumpulkan dan menganalisis informasi kebutuhan sasaran pengguna, lingkungan operasional dan tujuan aplikasi
2. Desain → Merancang UI/UX dan arsitektur aplikasi yang sesuai
3. Pengembangan Front-end → Pengembang mengimplementasikan rancangan ke dalam code menggunakan HTML, CSS, dan JS untuk membuat tampilan. Framework digunakan untuk mempercepat pengembangan.
4. Pengembangan Back-end → Pengembangan sisi server dan logika bisnis aplikasi. Dapat menggunakan framework juga untuk mempercepat pengembangan.
5. Integrasi dan Pengujian → API digunakan untuk mengintegrasikan sehingga sisi depan dan belakang dapat berkomunikasi dan berbagi data. Pengujian dilakukan untuk memastikan fitur berfungsi dan mengidentifikasi bug.
6. Pemeliharaan dan Peningkatan → Memperbaiki bug, dan meningkatkan fitur untuk memastikan aplikasi tetap relevan dalam waktu yang berlanjut.

Version Control (pengendalian versi) → sistem yang memungkinkan pengembang PL untuk melacak perubahan pada kode sumber aplikasi selama pengembangan. Contoh software yang populer yaitu Git dan Mercurial.

Manfaat : Rekam perubahan, pencatatan Riwayat, pemecahan konflik, dan pemulihan mudah.

Penggunaan Version Control untuk kolaborasi :

Inisialisasi proyek → membuat repositori yang akan menyimpan semua kode sumber, file dan perubahan.	Pengembangan Paralel → anggota tim menyalin repositori pada komputernya sendiri, kemudian membuat perubahan.
Branching → Membuat cabang yang terpisah dari kode utama.	Merge → setelah selesai perubahan, cabang digabung ke cabang utama.
Pull request → pengembang mengajukan perubahan yang dikerjakan untuk ditinjau oleh anggota tim lain sebelum digabung ke cabang utama.	

Tools Full Stack Developer

- IDE → Visual Studio Code
- Version Control → Github, Gitlab, Bitbucket, Sourcetree, Gitlens
- DMS → PostgreSQL, MySQL, Oracle, mongoDB, redis
- API → Postman, swagger
- Test dan Debugging → Jest, Mochachai, Junit5
- Mobile development → ReactNative, Flutter
- Cloud → aws, Gcloud, Azure
- CI/CD → Jenkins, circleci
- Desain UI/UX → Figma, Sketch

SDLC & Design Thinking Implementation

- SDLC (Software Development Life Cycle) → Siklus hidup perangkat lunak, rangkaian terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak. Dilakukan secara berurutan, untuk memastikan pengembangan berjalan dengan baik.

Fase SDLC

1. Perencanaan dan Analisis → identifikasi dan analisis masalah atau kebutuhan bisnis. Mengumpulkan persyaratan dan menentukan ruang lingkup mencakup alokasi sumber daya, jadwal waktu, definisi tugas dan tanggung jawab anggota.
2. Desain → arsitektur sistem, ui, rancangan database. Dirancang sesuai SRS atau persyaratan.
3. Pengembangan → implementasi rancangan perangkat lunak. Pengkodean
4. Pengujian → untuk memastikan berfungsi dengan baik, pengujian mencakup verifikasi fungsionalitas, kinerja, keamanan dan kualitas keseluruhan perangkat lunak.
5. Penerapan → menerapkan ke client apakah sudah cocok atau belum, jika belum bisa dilakukan pengkodean lagi.
6. Pemeliharaan → memperbaiki bug, meningkatkan fitur, menjaga perangkat lunak agar tetap sesuai dengan kebutuhan bisnis.

Manfaat SDLC

Prediktabilitas dan pengendalian proyek	Memenuhi kebutuhan pengguna
Peningkatan kualitas perangkat lunak	Penghematan biaya dan waktu
Pengelolaan risiko yang lebih baik	Meningkatkan pengawasan dan evaluasi
Efisiensi tim dan kolaborasi	Peningkatan dokumentasi

➤ Model – model SDLC

- Waterfall → linier dan berurutan. Setiap tahap harus selesai sebelum memulai tahap lain.
Cocok untuk proyek dengan persyaratan yang jelas dan stabil.
- V-Shaped → terkait erat dengan waterfall, menekankan pada pengujian. Setiap tahap pada pengujian. Cocok untuk proyek dengan yang focus pada kualitas tinggi.
(+) easy to understand, cocok untuk proyek kecil, sistematis yang disiplin, deliver high quality product
(-) tidak cocok dengan proyek yang sedang berjalan
- Prototype → menciptakan prototype atau contoh awal sebelum kembangkan versi final.
(+) reduce time and cost, mudah mendapatkan evaluasi user
(-) user banyak melakukan intervensi, menjadi kompleks
- Spiral → menggabungkan elemen model spiral dengan pendekatan incremental.
(+) risk analysis, request bisa dilakukan pada iterasi selanjutnya, mudah lakukan perubahan.
(-) cocok untuk proyek besar, mahal karena banyak iterasi dan memakan waktu yang lama.
- Iterative incremental → pengulangan siklus Pembangunan dan peningkatan perangkat lunak dalam tahapan kecil. Tiap iterasi menambahkan banyak fitur sampai produk akhir tercapai.
(+) cocok untuk proyek dengan waktu dan biaya terbatas, deteksi bug lebih awal karena iterasi kecil kecil.
(-) harus detail breakdown
- Bigbang → sederhana tetapi kurang terstruktur, setiap tahap dilakukan tanpa perencanaan detail.
(+) cocok untuk proyek kecil atau prototyping, fleksibel dan cepat
(-) kurang terstruktur, tidak ada tahapan pengujian khusus.
- Agile → kolaboratif, iterative, focus pada pengiriman perangkat lunak secara berkala, incremental, bekerja secara sprint (iterasi singkat)
(+) terbuka terhadap perubahan, cocok untuk proyek dengan lingkungan dinamis dan persyaratan yang berubah ubah
(-) kebutuhan kolaborasi yang kuat, tidak cocok untuk semua proyek.

➤ Design Thinking

- Empati : user research, empathy mapping (menggambarkan sikap, pemikiran, perasaan dan user problem)
- Define : problem statement, stakeholder alignment
- Ideate : brainstorming, idea consolidation
- Prototype : LowFi, HighFi
- Test : pengujian

Basic Git and Collaborating using Git

- Terminal and IDE → program yang memungkinkan pengguna untuk berinteraksi dengan sistem operasi atau computer melalui antarmuka teks atau baris perintah (cli).
- Version Control Git → sistem kontrol versi terdistribusi yang memungkinkan pengembang perangkat lunak untuk melacak perubahan dalam kode mereka, berkolaborasi dengan anggota tim, dan mengelola revisi kode secara efektif.

Sistem control terpusat : ada satu repositori sentral yang berfungsi sebagai "master" untuk menyimpan seluruh sejarah proyek. Setiap pengembang melakukan perubahan pada Salinan lokal, kemudian mengirimkan perubahan tersebut ke repositori sentral. Contoh sistem kontrol versi terpusat adalah Subversion (SVN).

Sistem control versi terdistribusi : terdistribusi, setiap anggota tim memiliki salinan lengkap dari seluruh repositori. Ini berarti setiap pengembang memiliki salinan lengkap Sejarah perubahan, tidak hanya salinan terbaru. Contoh sistem control versi terdistribusi adalah Git, Mercurial, dan Bazaar.