

### **A. Full Stack Developer Career Path**

Full Stack Development adalah pengembangan aplikasi yang mencakup semua aspek, mulai dari antarmuka pengguna (front-end) hingga logika bisnis di belakang layar (back-end), juga termasuk di sisi pengguna/klien (client-side).

1. *Frontend Web Development* : Pengembangan web front-end adalah praktik pembuatan HTML, CSS, dan JavaScript untuk situs web atau Aplikasi Web.
2. *Backend Development* : Bagian backend bertanggung jawab untuk memproses permintaan pengguna, data di database, dan respons ke klien (front-end).
3. *Database Management* : Serangkaian konsep dan teknik yang digunakan untuk mengelola data dalam sebuah aplikasi atau sistem.
4. *Mobile Development* : serangkaian konsep dan teknologi yang digunakan untuk membangun aplikasi yang dapat dijalankan di perangkat mobile.

#### Kolaborasi Efektif

*Version Control* : Version control (pengendalian versi) adalah sistem yang memungkinkan pengembang perangkat lunak untuk melacak perubahan pada kode sumber aplikasi selama pengembangan.

### **B. SDLC & Design Thinking Implementation**

SDLC (Siklus Hidup Pengembangan Perangkat Lunak) adalah rangkaian proses yang terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak dari awal hingga selesai.

1. *Perencanaan dan Analisis* : Melibatkan identifikasi masalah atau kebutuhan bisnis yang perlu diselesaikan. Rencana keseluruhan untuk proyek perangkat lunak dibuat.
2. *Desain Produk* : Perangkat lunak dirancang secara rinci berdasarkan persyaratan yang telah dikumpulkan, mencakup arsitektur sistem, antarmuka pengguna, dan desain database.
3. *Pengembangan Produk* : Implementasi rancangan yang telah disetujui sebelumnya. Para pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.
4. *Pengujian Produk* : Memastikan bahwa perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditentukan.
5. *Penerapan Produk* : Implementasi rancangan yang telah disetujui sebelumnya. Para pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.
6. *Pemeliharaan Produk* : Pemeliharaan dilakukan untuk memperbaiki bug, meningkatkan fitur, dan menjaga perangkat lunak agar tetap sesuai dengan perubahan kebutuhan bisnis.

### Model Model SDLC

1. *Waterfall Model* : Waterfall model adalah yang linier dan berurutan. Setiap tahap dalam model ini harus selesai sebelum memulai tahap berikutnya.
2. *V-Shaped Model* : Tahapan pengujian diwakili oleh garis miring "V", yang berarti bahwa setiap tahap pengembangan memiliki tahapan pengujian yang sesuai.
3. *Prototype Model* : Model pengembangan perangkat lunak yang bertujuan untuk menciptakan prototipe atau contoh awal sebelum mengembangkan versi akhirnya.
4. *Spiral Model* : Setiap siklus spiral membangun pada inkremental sebelumnya, menghasilkan perangkat lunak yang semakin berkembang dengan fitur yang lebih banyak.
5. *Iterative Incremental Model* : Setiap iterasi menambahkan lebih banyak fitur hingga produk akhir mencapai tingkat kesempurnaan yang diinginkan.
6. *Big Bang Model* : Model Big Bang adalah model yang kurang terstruktur, di mana semua tahapan pengembangan dilakukan tanpa perencanaan yang detail.
7. *Agile Model* : Model Agile adalah pendekatan kolaboratif dan iteratif yang berfokus pada pengiriman perangkat lunak secara berkala dan inkremental.

### Design Thinking Implementation

- |              |              |
|--------------|--------------|
| 1. Empathize | 4. Prototype |
| 2. Define    | 5. Test      |
| 3. Ideate    | 6. Implement |

### **C. Basic Git & Collaborating Using Git**

#### Version Control Git

Kontrol versi adalah metode yang digunakan untuk melacak dan mengelola perubahan dalam kode sumber atau berkas proyek. Git merupakan salah satu sistem kontrol versi terdistribusi yang paling populer dan kuat

1. *Sistem Kontrol Versi Terpusat* : 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.
2. *Sistem Kontrol Versi Terdistribusi* : setiap anggota tim memiliki salinan lengkap dari seluruh repositori. Ini berarti setiap pengembang memiliki salinan lengkap sejarah perubahan, tidak hanya salinan terbaru.

#### Git

Git adalah 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.