

Ringkasan Materi Pengenalan Software Engineer

Definisi Full Stack (Full Stack Development)

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

Scope Pada Full Stack Development

1. **Front-End Development:** Tahap Membangun antarmuka pengguna (UI) yang menarik dan interaktif dibangun menggunakan HTML, CSS, dan JavaScript.
Contoh framework: *React, Vue.js, AngularJs*.
2. **Back-End Development:** Tahap yang berfokus untuk Membangun server dan aplikasi yang berfungsi sebagai "otak" dari aplikasi. Ini menerima permintaan dari Front-End, memproses data, dan memberikan respons yang sesuai.
Contoh framework: *ExpressJs, Spring, Laravel, Rails*.
3. **Database Management:** Pada tahap ini, basis data dirancang dan dikelola untuk menyimpan, mengambil, dan memanipulasi data aplikasi.
Jenis database yang umum digunakan adalah SQL (*MySQL, PostgreSQL, SQL Server*) dan NoSQL (*MongoDB, Firebase*).
4. **Integration of Front-End and Back-End:** Tahap ini melibatkan menghubungkan komponen front-end dengan layanan back-end melalui API (Application Programming Interface) agar dapat berkomunikasi dengan server dan database.
5. **Version Control and Collaboration:** Penting untuk menggunakan sistem pengendalian versi seperti Git untuk mengelola perubahan kode dan memungkinkan kolaborasi dalam tim pengembang.
6. **Mobile Development:** Pengembangan aplikasi mobile melibatkan beberapa aspek penting untuk memastikan aplikasi berjalan dengan baik dan memberikan pengalaman pengguna yang optimal.
Contoh framework: *React Native, Flutter*.

Tahap-tahap Pengembangan Aplikasi End-To-End

1. Perencanaan dan Analisis
2. Desain
3. Pengembangan Front-End
4. Pengembangan Back-End
5. Integrasi dan Pengujian
6. Pemeliharaan dan Peningkatan

Implementasi Design Thinking:

- Menggabungkan pendekatan Design Thinking dalam SDLC untuk menciptakan produk yang lebih berfokus pada kebutuhan pengguna dan tujuan bisnis.
- Metode ini bersifat iteratif untuk memastikan perangkat lunak tetap relevan dengan perubahan kebutuhan pengguna dan pasar.

Version Control Git (Kontrol Versi Git):

- Git adalah sistem kontrol versi terdistribusi yang populer dan kuat.
- Memungkinkan pengembang melacak perubahan dalam kode, berkolaborasi dengan tim, dan mengelola revisi kode secara efisien.
- Proses instalasi Git disebutkan untuk Windows, Linux, dan MacOS.

Kolaborasi Menggunakan Git:

- Melibatkan pengaturan dasar repository, pembuatan cabang, pembuatan perubahan, penanganan konflik, permintaan tarik, dan penggabungan menggunakan platform seperti GitHub.