

Summary – Software Engineering

- **Introduction Full Stack Web / Mobile Developer**

Pengembangan Full Stack merujuk pada pengembangan seluruh aplikasi secara end-to-end, dari sisi depan (front-end) hingga sisi belakang (back-end), hingga sisi klien (client-side). Scope penting dalam Full Stack Development yaitu, Front-End Development, Back-End Development, Database Management, Integration of Front-End and Back-End, Version Control and Collaboration, dan Mobile Development.

Front-End Web Development terdiri dari pembuatan HTML, CSS, dan JavaScript untuk website sehingga pengguna dapat melihat dan berinteraksi langsung. Framework yang populer untuk Front-End ada 3 yaitu, React, Vue.js, dan AngularJS by Google.

Backend Development bertanggung jawab memproses permintaan dari pengguna, terdiri dari bahasa pemrograman Server-Side seperti Node.js, Python, Java, PHP, C# yang digunakan untuk menulis kode di sisi server, selain itu ada juga Server Framework dan Database Management. Framework yang populer untuk Backend ada 4 yaitu, Express.js, Spring, Laravel, dan Rails.

Database Management digunakan untuk mengelola data dalam sebuah aplikasi atau sistem, terdiri dari Database Management System, Tipe Database, dan Bahasa Query. Framework yang populer untuk Database Management ada 2 yaitu SQL dan No SQL. SQL terdiri dari Oracle, PostgreSQL, dan MySQL. Sedangkan No SQL terdiri dari MongoDB dan Redis.

Mobile Development digunakan untuk membangun aplikasi yang dapat dijalankan di perangkat mobile, seperti smartphone dan tablet, terdiri dari Platform Mobile dan IDE. Framework yang populer untuk Mobile Development yaitu React Native dan Flutter.

- **Skillset Full Stack Web / Mobile Developer**

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

Kolaborasi Efektif yaitu dengan Version Control yang melacak perubahan pada kode sumber aplikasi selama pengembangan. Version Control yang populer adalah Git dan Mercurial. Manfaatnya dalam berkolaborasi untuk rekam perubahan, pencatatan riwayat, pemecahan konflik, dan pemulihan mudah. Penggunaan untuk kolaborasi yaitu inisialisasi proyek, pengembangan paralel, branching, merge, dan pull request.

- **Tools Full Stack Web / Mobile Developer**

IDE – Code Editor (Visual Studio Code)
Version Control – Repository (GitHub, GitLab, Bitbucket)
Version Control – Git Tools (Sourcetree, GitLens)
DBMS (PostgreSQL, MySQL, ORACLE, MongoDB, Redis)
API (POSTMAN, Swagger)
Test dan Debugging (Jest, Mocha, Chai, JUnit 5)

Mobile Development (React Native, Flutter)

Layanan Cloud (aws, Google Cloud, Azure)

CI/CD (Jenkins, circleci)

Desain UI/UX (Figma, Sketch)

- **SDLC**

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

Siklus SDLC :

- Perencanaan dan Analisis, identifikasi masalah atau kebutuhan bisnis yang perlu diselesaikan oleh perangkat lunak.
- Desain, perangkat lunak dirancang secara rinci berdasarkan persyaratan yang telah dikumpulkan.
- Pengembangan, implementasi rancangan perangkat lunak yang telah disetujui sebelumnya.
- Pengujian, memastikan bahwa perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditentukan.
- Penerapan, pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.
- Pemeliharaan, untuk memperbaiki bug, meningkatkan fitur, dan menjaga perangkat lunak agar tetap sesuai dengan perubahan kebutuhan bisnis.

Model-model SDLC terdiri dari Waterfall Model, V-Shaped Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model.

Design Thinking Process terdiri dari Empathize (Understand User Needs), Define the Problem, Ideate (Generate Ideas), Prototype (Build and Iterative Solutions), Test (Gather User Feedback), dan Implement (Develop the Software).

- **Basic Git & Collaborating Using Git**

Terminal memberikan fleksibilitas dan kekuatan untuk melakukan tugas-tugas khusus dan otomatisasi dalam lingkungan komputer modern.

Kontrol Versi metode yang digunakan untuk melacak dan mengelola perubahan dalam kode sumber atau berkas proyek, terdiri dari Sistem Kontrol Versi Terpusat dan Sistem Kontrol Versi Terdistribusi.

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.