**INTRODUCTION TO SOFTWARE ENGINEERING**

**Fullstack Developer**

**Definisi dan Scope Fullstack Developer**

Pengembangan Full Stack (Full Stack Development) merujuk pada pengembangan seluruh aplikasi secara end-to-end, dari sisi depan (front-end) hingga sisi belakang (back-end).

**Scope Penting Dalam Fullstack Development**

1. **Front End Development**

Membangun antarmuka pengguna yang menarik dan interaktif menggunakan HTML, CSS, dan JavaScript.

1. **Back End Development**

Membangun server dan aplikasi yang berfungsi sebagai "otak" dari aplikasi, menerima permintaan dari sisi depan, memproses data, dan memberikan respons yang sesuai.

1. **Database Management**

Mendesain dan mengelola basis data untuk menyimpan, mengambil, dan memanipulasi data aplikasi.

1. **Integration of Front-End and Back-End**

Menghubungkan komponen front-end dengan layanan back-end melalui API (Application Programming Interface) untuk berkomunikasi dengan server dan database.

1. **Version Control and Collaboration**

Menggunakan sistem pengendalian versi, seperti Git, untuk mengelola perubahan kode dan kolaborasi dalam tim pengembang.

1. **Mobile Development**

Beberapa Pengembang Full Stack juga memiliki kemampuan untuk mengembangkan aplikasi mobile menggunakan framework seperti React Native, Flutter

**SDLC & Desing Thinking Implementation**

**Apa itu SDLC**

SDLC (Siklus Hidup Pengembangan Perangkat Lunak) adalah rangkaian proses yang terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak dari awal hingga selesai. SDLC terdiri dari serangkaian tahap yang saling terkait dan dilakukan secara berurutan untuk memastikan bahwa pengembangan perangkat lunak berjalan dengan baik dan sesuai dengan kebutuhan dan tujuan yang ditentukan.

**Siklus SDLC**

1. **Perencanaan dan Analisis**

Tahap pertama ini melibatkan identifikasi masalah atau kebutuhan bisnis yang perlu diselesaikan oleh perangkat lunak. Para pemangku kepentingan berinteraksi untuk mengumpulkan persyaratan dan menentukan ruang lingkup proyek. Rencana keseluruhan untuk proyek perangkat lunak dibuat. Rencana ini mencakup alokasi sumber daya, jadwal waktu, dan definisi tugas dan tanggung jawab anggota tim.

1. **Desain Produk**

Di tahap ini, perangkat lunak dirancang secara rinci berdasarkan persyaratan yang telah dikumpulkan. Desain mencakup arsitektur sistem, antarmuka pengguna, dan desain database.

1. **Pengembangan Produk**

Tahap ini melibatkan implementasi rancangan perangkat lunak yang telah disetujui sebelumnya. Para pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.

1. **Pengujian Produk**

Setelah perangkat lunak dikembangkan, tahap pengujian dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditentukan. Pengujian mencakup verifikasi fungsionalitas, kinerja, keamanan, dan kualitas keseluruhan perangkat lunak.

1. **Penerapan Produk**

Tahap ini melibatkan implementasi rancangan perangkat lunak yang telah disetujui sebelumnya. Para pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.

1. **Pemeliharaan Produk**

Setelah perangkat lunak diimplementasikan, pemeliharaan dilakukan untuk memperbaiki bug, meningkatkan fitur, dan menjaga perangkat lunak agar tetap sesuai dengan perubahan kebutuhan bisnis.

**Design Thinking**

1. **Emphatize : Understand User Needs**

Pada tahap ini, fokus pada memahami secara mendalam pengguna akhir dan kebutuhan mereka, keinginan, serta masalah yang dihadapi. Anda dapat melakukan berbagai kegiatan, seperti user research, empathy mapping, user personas

1. **Define the Problem**

Pada tahap ini, informasi yang dikumpulkan selama fase empati dianalisis untuk menentukan masalah dan menetapkan tujuan yang jelas untuk proyek. Kegiatan kunci meliputi problem statement, stakeholder alignment.

1. **Ideate**

Fase ideasi mendorong pemikiran kreatif dan menghasilkan berbagai solusi potensial. Kegiatan kunci meliputi brainstorming sessions, idea consolidation

1. **Prototype : Build and Iterative Solutions**

Pada tahap ini, fokus pada menciptakan representasi nyata dari ide-ide yang dipilih. Prototype digunakan untuk mengumpulkan umpan balik dan memvalidasi asumsi. Kegiatan kunci meliputi low fidelity prototypes, high fidelity prototypes

1. **Test : Gather User Feedback**

Tahap pengujian melibatkan pengumpulan umpan balik dari pengguna nyata untuk memvalidasi solusi-solusi tersebut. Kegiatan kunci meliputi usability testing, iterative testing

1. **Implement : Develop the Software**

Pada tahap ini, desain diterjemahkan ke dalam kode yang sebenarnya dan diimplementasikan. Kegiatan kunci meliputi agile development, cross-functional collaboration.

**Basic Git & Collaborating Using Git**

**Sejarah Singkat Terminal**

Dengan perkembangan teknologi dan perangkat lunak, terminal tetap menjadi alat penting bagi para pengembang perangkat lunak, administrator sistem, dan pengguna teknis lainnya. Meskipun antarmuka grafis semakin canggih dan populer, terminal tetap memberikan fleksibilitas dan kekuatan untuk melakukan tugas-tugas khusus dan otomatisasi dalam lingkungan komputer modern.

**Apa Itu 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.