



- 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.

2. Desain

Desain mencakup arsitektur sistem, antarmuka pengguna, dan desain database.

3. Pengembangan

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

4. Pengujian

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.

5. Penerapan

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

6. Pemeliharaan

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

Manfaat Penggunaan SDLC

1. Prediktabilitas dan Pengendalian Proyek
2. Peningkatan Kualitas Perangkat Lunak
3. Efisiensi Tim dan Kolaborasi
4. Peningkatan Dokumentasi
5. Memenuhi Kebutuhan Pengguna
6. Penghematan Biaya dan Waktu
7. Pengelolaan Risiko yang Lebih Baik

Model-Model SDLC

1. Waterfall Model
2. V-Shaped Model
3. Prototype Model
4. Spiral Model
5. Iterative Incremental Model
6. Big Bang Model
7. Agile Model

- GIT

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.

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. Topic ini akan mempelajari menginstal Git pada berbagai sistem operasi, menginisialisasi repositori Git baru, dan melakukan commit pertama.

- Fullstack Web/Mobile 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) dan, dalam beberapa kasus, hingga sisi klien (client-side)

Scope Penting Fullstack Development

1. Front-End Development
2. Back-End Development
3. Database Management
4. Integration of FE and BE
5. Version control and collaboration
6. Mobile development

- Front-End Development

Pengembangan web front-end, juga dikenal sebagai pengembangan sisi klien adalah praktik pembuatan HTML, CSS, dan JavaScript untuk situs web atau Aplikasi Web sehingga pengguna dapat melihat dan berinteraksi dengannya secara langsung

- **Back-End Development**

Bagian backend bertanggung jawab untuk memproses permintaan dari pengguna, mengelola dan menyimpan data di database, serta memberikan respons kepada klien (front-end) berdasarkan permintaan yang diterima.

- **Mobile Development**

Serangkaian konsep dan teknologi yang digunakan untuk membangun aplikasi yang dapat dijalankan di perangkat mobile, seperti smartphone dan tablet. Pengembangan aplikasi mobile mencakup beberapa aspek penting untuk memastikan aplikasi dapat berjalan dengan baik dan memberikan pengalaman pengguna yang optimal.