



Homework - Introduction to Software Engineering

Muhamad Rifqi Fadhilah,-

1 - Definisi dari Full Stack Development 📖

Full Stack Development itu semacam kemampuan seseorang untuk handle semua lini dalam pengembangan web, mulai dari tampilan depan (frontend) sampai sistem di belakang layar (backend) dan manajemen database.

2 - Scope pada Full Stack Development 🌐

- **Frontend:** Bagian yang dilihat dan diakses user. Tampilannya, interaksinya, dan sebagainya.
- **Backend:** Mesin di belakang layar yang menjalankan semua fungsi.
- **Database Management:** Penyimpanan data dan informasi yang bisa diakses ketika dibutuhkan.

3- Pengertian Frontend Web Development 🎨

- **HTML:** Struktur dasar web.
- **CSS:** Penyempurna tampilan web, bikin lebih menarik.
- **Javascript:** Bikin web jadi interaktif.
- **Framework:** Toolkit buat mempermudah pengembangan web. Contoh:
 - **React:** Bikin interface user jadi responsif.
 - **Vue:** Ringan, mudah, dan cepat.
 - **Angular:** Kerangka kerja lengkap buat aplikasi web.

4- Pengertian Backend Web Development

- **Bahasa pemrograman server-side:** Bahasa yang dijalankan di server, seperti Python, Java, PHP.
- **Server Framework:** Tool yang membantu dalam pengembangan backend.
 - **ExpressJs:** Framework ringan untuk Node.js.
 - **Spring:** Framework Java untuk aplikasi enterprise.
 - **Laravel:** Framework PHP yang populer dan fleksibel.
- **Database untuk backend:** Tempat penyimpanan data, seperti MySQL, PostgreSQL.

5- Database Management untuk Development Software

- **Database Management System (DBMS):** Software buat mengelola database.
- **Tipe-tipe database:** Relasional (seperti MySQL), NoSQL (seperti MongoDB).
- **Bahasa Query:** Bahasa untuk berinteraksi dengan database, misalnya SQL.
- Contoh database populer: MySQL, PostgreSQL, MongoDB.

6- Pengertian Version Control

Version control itu tool buat track perubahan dalam kode. Contoh paling populer: **Git**.

- Inisialisasi proyek, pengembang paralel, branching (bikin cabang kode), merge (gabungin kode), pull request (minta gabung kode).

7- Tool Set untuk Develop Software/Web Fullstack 🛠️

Selain HTML, CSS, JavaScript, dan sebagainya, ada tools lain seperti:

- **Postman:** Buat test API.
- **Swagger:** Dokumentasi dan test API.

8- Model Software Development Life Cycle (SDLC) ↺

SDLC itu langkah-langkah dalam pengembangan software, dari ide sampai launching. Model model SDLC itu ada :

1. Waterfall Model 🌊

- Proses linear dan sekuensial.
- Setiap fase harus diselesaikan sebelum fase berikutnya dimulai.
- Cocok untuk proyek dengan spesifikasi yang jelas dan tetap.

2. Iterative Model ↻

- Proses pengembangan yang berulang-ulang.
- Setiap iterasi menghasilkan versi produk yang lebih baik.
- Cocok untuk proyek dengan ruang untuk evolusi dan perbaikan.

3. Spiral Model 🌀

- Kombinasi dari pendekatan Waterfall dan Iterative.
- Fokus pada risiko dan evaluasi risiko di setiap tahap spiral.
- Cocok untuk proyek kompleks dan besar.

4. V-Model (Validation and Verification) ✓

- Peningkatan dari model Waterfall.
- Setiap tahap pengembangan memiliki tahap pengujian yang sesuai.
- Pengujian dilakukan sejalan dengan tahap pengembangan.

5. Agile Model 🏃

- Fokus pada iterasi singkat dan feedback cepat dari klien atau pengguna.

- Fleksibel dengan perubahan.
- Cocok untuk lingkungan yang berubah-ubah dan tim yang lebih kecil.

6. Incremental Model +

- Produk dibangun dan dikembangkan bertahap.
- Setiap increment menambahkan fitur baru ke sistem.
- Cocok untuk proyek dengan milestones yang jelas.

7. Big Bang Model 🌟

- Dimulai dengan sedikit perencanaan dan fokus pada pengkodean.
- Pengembangan berdasarkan intuisi developer.
- Cocok untuk proyek kecil atau sebagai prototipe awal.

8. RAD (Rapid Application Development) 🚀

- Fokus pada prototipe cepat dan feedback dari pengguna.
- Menggunakan pendekatan terstruktur untuk memaksimalkan kecepatan pengembangan.
- Cocok untuk proyek yang memerlukan solusi cepat.

9- Design Thinking pada Software Development 💡

Proses kreatif buat menyelesaikan masalah dengan fokus pada kebutuhan user. Pake pendekatan empati, kolaborasi, dan eksperimen buat dapetin solusi yang optimal. Berikut cara penerapan Design Thinking dalam Software Development dengan contohnya:

1. Empathy (Empati) - Mengerti Pengguna 🤝

- Melakukan wawancara dengan pengguna untuk mengetahui kebutuhan, harapan, dan frustrasi mereka.
- **Contoh:** Tim pengembangan ingin membuat aplikasi belanja online. Mereka melakukan wawancara dengan pengguna aplikasi serupa untuk mengetahui apa yang mereka sukai dan apa yang mereka rasa kurang.

2. Define (Definisi) - Mengartikulasikan Masalah 🔍

- Mengidentifikasi masalah yang perlu dipecahkan berdasarkan informasi dari fase Empati.
- **Contoh:** Tim menemukan bahwa pengguna sering kesulitan mencari produk spesifik dalam aplikasi belanja, jadi mereka menentukan masalah yang harus dipecahkan adalah "Bagaimana kita dapat meningkatkan fitur pencarian?"

3. Ideate (Ideasi) - Menghasilkan Solusi 💡

- Brainstorming ide-ide solusi untuk masalah yang sudah didefinisikan.
- **Contoh:** Tim memikirkan ide-ide seperti: menambahkan filter pencarian, fitur pencarian suara, dan rekomendasi produk.

4. Prototype (Prototipe) - Membuat Model Awal 🛠️

- Mengembangkan versi awal dari solusi untuk menguji konsep.
- **Contoh:** Tim membuat mockup dari interface aplikasi dengan fitur pencarian yang ditingkatkan.

5. Test (Pengujian) - Mendapatkan Feedback ↺

- Menguji prototipe dengan pengguna sebenarnya dan mendapatkan feedback.
- **Contoh:** Pengguna menguji mockup dan memberikan feedback bahwa fitur pencarian suara sangat membantu tetapi filter pencarian perlu lebih spesifik.