

Homework

Introduction to Software Engineering

Qurrota A'yun Liulinuha  
[qurrotaayunliulinuha@gmail.com](mailto:qurrotaayunliulinuha@gmail.com)

## 1. Soal & Instruksi - Instalasi Tools

Silahkan teman teman melakukan proses instalasi beberapa tools dibawah ini :

- Git
- Visual Studio Code
- Browser

Setelah melakukan instalasi, buat sebuah summary untuk apa yang sudah dipelajari dalam materi pengenalan software engineer sebelumnya pada google docs.

### 1. Fullstack Web Developer

**Fullstack Web Developer** adalah seorang pengembang perangkat lunak yang memiliki keahlian menyeluruh dalam membangun aplikasi web, baik dari sisi tampilan (frontend) maupun logika di belakangnya (backend). Mereka seperti seorang arsitek yang merancang dan membangun sebuah bangunan secara keseluruhan, mulai dari fondasi hingga dekorasi interior.

- **Frontend Development:** Berfokus pada tampilan dan interaksi pengguna dengan aplikasi. Teknologi yang umum digunakan meliputi HTML, CSS, dan JavaScript.
- **Backend Development:** Berfokus pada logika bisnis, database, dan server. Teknologi yang umum digunakan meliputi bahasa pemrograman seperti Python, Ruby, Node.js, dan database seperti MySQL, PostgreSQL.

#### a. Front-end Development

**Front-end Development** berfokus pada pembangunan antarmuka pengguna (User Interface) dan interaksi pengguna dengan aplikasi. Tugas seorang Front-end Developer adalah memastikan pengguna memiliki pengalaman yang baik saat mengakses website atau aplikasi web.

Alat dan teknologi yang sering digunakan oleh Front-end Developer antara lain:

- **HTML** (Hypertext Markup Language) untuk struktur halaman.
- **CSS** (Cascading Style Sheets) untuk styling atau tampilan.
- **JavaScript** untuk interaktivitas.
- **Framework/Libraries:** React.js, Vue.js, Angular, jQuery untuk pengembangan yang lebih cepat dan efisien.

Fokus utama dari front-end adalah user experience (UX) dan user interface (UI) yang responsif dan cepat.

## b. Back-end Development

**Back-end Development** berhubungan dengan pengelolaan server, database, dan logika bisnis aplikasi web. Back-end bekerja di balik layar dan menangani pengolahan data yang diminta oleh front-end.

Teknologi dan alat yang sering digunakan dalam back-end development termasuk:

- **Bahasa Pemrograman:** Node.js, Python, Ruby, PHP, Java, C#.
- **Framework:** Express.js, Django, Flask, Laravel.
- **Database:** MySQL, MongoDB, PostgreSQL.
- **API:** Pengembang back-end juga harus mampu membuat dan mengelola **API** (Application Programming Interface) agar front-end dan back-end dapat berkomunikasi.

Fokus back-end adalah performa aplikasi, keamanan, dan pengelolaan data yang efisien.

## 2. SDLC (Software Development Life Cycle): Kerangka Kerja Pengembangan Perangkat Lunak

**SDLC** adalah metodologi standar yang digunakan untuk merencanakan, mengembangkan, menguji, dan merilis perangkat lunak. Tahapan-tahapan dalam SDLC meliputi:

- **Planning:** Identifikasi kebutuhan bisnis dan analisis risiko.
- **Requirement Analysis:** Menganalisis kebutuhan sistem dan perangkat lunak.
- **Design:** Membuat blueprint atau arsitektur sistem.
- **Development:** Pengembangan kode dan implementasi desain.
- **Testing:** Pengujian untuk menemukan bug dan memastikan sistem berjalan sesuai spesifikasi.
- **Deployment:** Merilis perangkat lunak ke pengguna.
- **Maintenance:** Memperbaiki masalah yang ditemukan setelah deployment.

SDLC membantu memastikan perangkat lunak dikembangkan dengan cara yang terorganisir, konsisten, dan berkualitas tinggi.

## 3. Metodologi Agile

**Agile** adalah metodologi pengembangan perangkat lunak yang mengutamakan fleksibilitas, iterasi cepat, dan kolaborasi antar tim. Agile memecah proyek menjadi **sprint** (periode waktu yang pendek, biasanya 1-4 minggu), di mana tim pengembang merilis bagian kecil dari produk yang dapat diuji dan dievaluasi. Tujuan utamanya adalah untuk memberikan nilai kepada pengguna secara terus-menerus dan beradaptasi dengan perubahan kebutuhan proyek.

Prinsip utama Agile meliputi:

- **Iterasi singkat:** Pengembangan dilakukan dalam siklus pendek untuk memungkinkan adaptasi cepat terhadap perubahan.
- **Kolaborasi antar tim:** Tim yang terdiri dari pengembang, desainer, dan pemangku kepentingan berkolaborasi untuk memastikan semua kebutuhan terpenuhi.
- **Feedback terus-menerus:** Setiap sprint berakhir dengan pengujian dan evaluasi, diikuti oleh peningkatan pada sprint berikutnya.

Agile menggunakan kerangka kerja seperti **Scrum** dan **Kanban** untuk manajemen tim dan tugas.

#### 4. Basic Git

**Git** adalah alat version control system (VCS) yang paling populer dan digunakan untuk melacak perubahan pada kode dan memungkinkan kolaborasi antara pengembang. Git memastikan bahwa semua perubahan dapat dilacak dan dikembalikan ke versi sebelumnya jika diperlukan, serta memfasilitasi kerja tim secara efektif.

Beberapa perintah dasar Git yang wajib dikuasai:

- **git init:** Membuat repositori baru.
- **git clone <URL>:** Meng-clone repositori dari GitHub atau sumber lain ke mesin lokal.
- **git add <nama\_file>:** Menambahkan perubahan pada staging area.
- **git commit -m "pesan commit":** Menyimpan perubahan dengan deskripsi yang jelas.
- **git push:** Mengirimkan perubahan dari repositori lokal ke repositori remote.
- **git pull:** Mengambil perubahan terbaru dari repositori remote ke lokal.
- **git branch:** Melihat branch yang ada atau membuat branch baru.

**Git** sangat penting dalam pengembangan perangkat lunak karena memungkinkan kolaborasi dan memastikan semua perubahan tercatat dengan rapi.

## 2. Soal & Instruksi - Collaborating Using Git

Buatlah repositori GitHub baru dengan nama "finpro-msib-7-kelompok X"

Contoh : finpro msib 7 kelompok 1

*\*1 kelompok hanya membuat 1 repositori saja*

- Undang teman-teman kelompokmu sebagai kolaborator pada repositori tersebut
- Setiap peserta meng-*clone* repositori ke mesin lokal masing-masing
- Setiap peserta harus membuat cabang untuk pekerjaannya. Misal : doni-homework, nita-homework, etc
- Masukkan Homework dari minggu sebelumnya ke branch lokalmu.
- Setelah melakukan perubahan, teman-teman harus melakukan commit dengan pesan commit yang deskriptif
- Lakukan push and pull ke branch homework dari branch lokalmu.
- Pastikan konflik sudah terselesaikan sebelum melakukan pull request

Setelah selesai, bisa teman teman zip hasil soal pertama dan kedua, lalu upload ke dalam LMS Rakamin.

Link Repository: <https://github.com/yulia30359/finpro-msib-7-kelompok-2.git>