

## **SUMMARY**

### **“Introduction to Software Engineering”**

**by Adinda Cahya Kamilla**

Pertemuan yang diadakan minggu lalu bertujuan untuk memberikan pemahaman dasar mengenai pengembangan full stack web dan mobile, memperkenalkan teknologi front-end dan back-end, serta kolaborasi menggunakan Git. Berikut adalah ringkasan dari topik yang dibahas sesuai dengan objektif sesi:

#### **1. Dasar-Dasar Pengembangan Full Stack Web/Mobile**

Dalam dunia pengembangan perangkat lunak, seorang Full Stack Web/Mobile Developer merupakan profesional yang memiliki keahlian di kedua sisi pengembangan, yakni front-end dan back-end. Mereka bertanggung jawab untuk menciptakan aplikasi web dan mobile yang fungsional, terstruktur, serta memiliki tampilan yang baik. Dengan kata lain seorang Full Stack Developer bertanggung jawab untuk membangun aplikasi dari antarmuka pengguna hingga pengelolaan basis data dan logika server. Dalam hal ini seorang Full Stack Developer harus mengetahui dan memahami skillset dan tools dalam pengembangan Full Stack.

#### **2. Pengembangan Front-End: Dasar-Dasar HTML, CSS, dan JavaScript**

Pada pengembangan front-end, materi fokus pada teknologi inti seperti HTML, CSS, dan JavaScript.

- HTML (HyperText Markup Language) digunakan untuk membangun struktur dasar dari halaman web, seperti paragraf, judul, tabel, dan gambar.
- CSS (Cascading Style Sheets) bertujuan untuk memberikan gaya visual pada elemen HTML, seperti warna, tata letak, dan animasi.
- JavaScript adalah bahasa pemrograman yang digunakan untuk menambah interaktivitas pada halaman web, seperti manipulasi DOM, validasi formulir, dan animasi.

Pengenalan ketiga teknologi ini memberikan dasar yang kuat untuk memahami pengembangan front-end. Mereka berfungsi sebagai blok bangunan untuk membuat antarmuka pengguna yang dinamis dan responsif.

### **3. Menjelajahi Framework dan Library Front-End**

Selain memahami teknologi inti, sesi ini juga membahas tentang framework dan library yang banyak digunakan dalam pengembangan front-end modern. Framework seperti React, Angular, dan Vue.js memainkan peran penting dalam membangun antarmuka pengguna yang kompleks dengan kode yang lebih terstruktur. Framework ini menawarkan cara yang lebih efisien untuk menangani logika aplikasi dan state management.

Dengan menggunakan library dan framework ini, pengembang dapat membangun aplikasi yang lebih interaktif, ramah pengguna, dan mudah untuk dipelihara.

### **4. Pengembangan Back-End: Pemrograman Sisi Server dan Database**

Selain pengembangan front-end, pengembangan back-end juga sangat penting dalam membangun aplikasi fullstack. Sesi ini mencakup konsep-konsep dasar dalam pemrograman sisi server, seperti menangani permintaan HTTP, mengelola autentikasi dan otorisasi pengguna, serta interaksi dengan database.

Teknologi back-end yang diperkenalkan meliputi:

- Node.js dan Express.js sebagai teknologi JavaScript untuk membangun server dan mengelola rute.
- Python (Django/Flask) untuk pengembangan aplikasi web berbasis Python yang kuat.
- Database SQL dan NoSQL, seperti Oracle, PostgreSQL, MySQL, serta MongoDB dan Redis untuk menyimpan dan mengelola data aplikasi.

### **5. Konsep dan Framework Pengembangan Aplikasi Mobile**

Selain pengembangan web, pengembangan aplikasi mobile juga menjadi bagian penting dalam jalur karir Full Stack Developer. Framework seperti React Native dan Flutter diperkenalkan sebagai solusi untuk

mengembangkan aplikasi mobile lintas platform (cross-platform) dengan menggunakan satu basis kode.

Kedua framework ini memungkinkan pengembang untuk membangun aplikasi iOS dan Android menggunakan bahasa pemrograman yang sudah dikenal (JavaScript untuk React Native dan Dart untuk Flutter). Ini memberikan keuntungan besar dalam hal penghematan waktu dan sumber daya pengembangan.

## 6. Basic Git & Collaboration Using Git

Git adalah sistem kontrol versi terdistribusi yang memungkinkan pengembang melacak perubahan dalam kode, berkolaborasi, dan mengelola revisi secara efektif. Git digunakan bersama dengan platform berbasis cloud seperti GitHub, GitLab, atau Bitbucket untuk kolaborasi proyek. Ada juga version control git tools seperti Sourcetree dan Gitlens. Perintah Dasar Git:

- git init: Menginisialisasi repository baru.
- git clone: Menduplikasi repository dari remote ke lokal.
- git add: Menambahkan perubahan ke staging area.
- git commit: Membuat commit dari perubahan yang di-staging.
- git push: Mengirim commit ke repository remote.
- git pull: Mengambil perubahan dari remote repository ke lokal.
- git branch: Menampilkan daftar branch.
- git checkout: Beralih antar branch.
- git merge: Menggabungkan perubahan dari satu branch ke branch lain.

Kolaborasi menggunakan Git biasanya melibatkan simulasi dengan beberapa peserta. Setiap peserta bekerja pada branch yang berbeda, melakukan perubahan, dan mengirim perubahan tersebut ke repository remote. Ketika terjadi konflik, mereka harus menyelesaikannya secara manual dengan editor teks sebelum menggabungkannya ke branch utama. Setelah konflik selesai, mereka dapat membuat *pull request* untuk menggabungkan perubahan mereka ke branch utama melalui platform seperti GitHub.