

**Nama : Mulyana Nurcahyani**

**Kelompok : 5**

**Homework – Software Developer**

## **SUMMARY**

### **Pengertian Fullstack Development**

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 Full Stack Development :

- **Front-End Development**  
Membangun antarmuka pengguna yang menarik dan interaktif menggunakan HTML, CSS, dan JavaScript.
- **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.
- **Database Management**  
Mendesain dan mengelola basis data untuk menyimpan, mengambil, dan memanipulasi data aplikasi.
- **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.
- **Version Control and Collaboration**  
Menggunakan sistem pengendalian versi, seperti Git, untuk mengelola perubahan kode dan kolaborasi dalam tim pengembang.
- **Mobile Development**
- Beberapa Pengembang Full Stack juga memiliki kemampuan untuk mengembangkan aplikasi mobile menggunakan framework seperti React Native, Flutter,

### **Pengembangan aplikasi end to end**

Merupakan pendekatan pengembangan perangkat lunak yang mencakup keseluruhan siklus pembuatan aplikasi, dari tahap perencanaan hingga tahap pengujian dan implementasi. Tujuannya adalah untuk menghasilkan aplikasi yang lengkap, fungsional, dan siap digunakan oleh pengguna akhir. Berikut adalah tahapan-tahapan dalam pengembangan aplikasi end-to-end.

1. Perencanaan dan Analisis
2. Desain
3. Pengembangan Front-End

4. Pengembangan Back-End
5. Integrasi dan Pengujian
6. Pemeliharaan dan Peningkatan

## **Version Control**

Version control (pengendalian versi) adalah sistem yang memungkinkan pengembang perangkat lunak untuk melacak perubahan pada kode sumber aplikasi selama pengembangan. Ini memungkinkan kolaborasi yang efisien di antara anggota tim, terutama ketika banyak orang bekerja pada proyek yang sama. Berikut manfaat version control :

1. Rekam Perubahan
2. Pencatatan Riwayat
3. Pemecahan Konflik
4. Pemulihan Mudah

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. Terdapat beberapa dasar dasar dalam penggunaan GIT, yaitu

- Commit  
Commit berfungsi untuk membuat komitmen dan deskripsi terhadap perubahan sebelum terjadinya push.
- Push  
Push berfungsi untuk mengupload perubahan yang telah dicommit pada repositori cloud
- Pull  
Pull berfungsi untuk mengambil atau mendownload commit terbaru dari repositori cloud ke repositori lokal
- Fetch  
Fetch berfungsi untuk mererefresh repositori jika telah terjadi perubahan
- Branch  
Branch merupakan cabang yang ada pada repositori
- Merge  
Merge merupakan penggabungan antara cabang satu dengan cabang lainnya.

## **SDLC (System Development Life Cycle)**

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. SDLC terdiri dari 6 fase, yaitu :

1. Perencanaan dan Analisis
2. Desain
3. Pengembangan
4. Pengujian
5. Penerapan
6. Pemeliharaan

Dengan menggunakan SDLC secara efektif, organisasi dapat meningkatkan keberhasilan dan efisiensi dalam mengembangkan aplikasi, memastikan pengiriman produk berkualitas tepat waktu, dan memberikan nilai yang lebih besar bagi pelanggan dan stakeholder.

Terdapat 7 model SLDC, yaitu :

- Waterfall Model
- V-Shaped Model
- Prototype Model
- Spiral Model
- Iterative Incremental Model
- Big Bang Model
- Agile Model

# COLLABORATING USING GIT

Link Github : <https://github.com/laqueta/it-perbankan-finpro-kelompok5.git> dengan branch mulyana-homework

