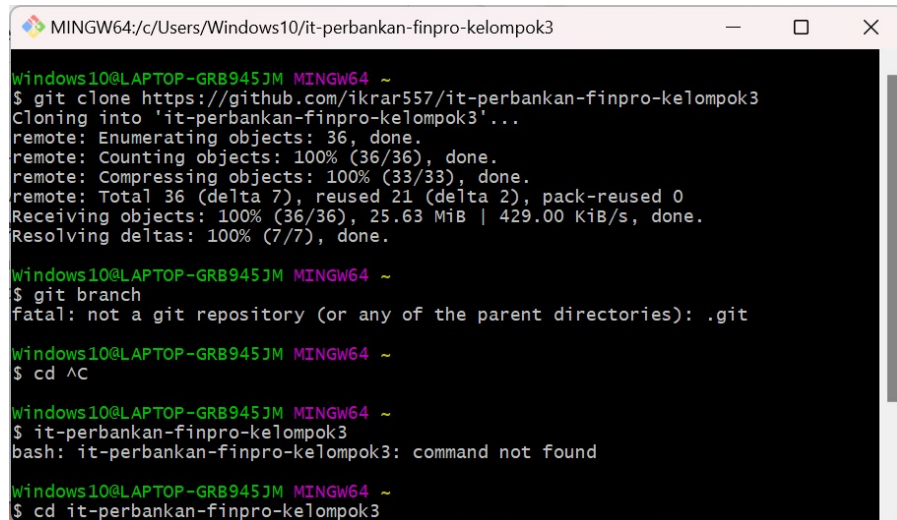


# Homework-Introduction to Software Engineering

## Soal 1

### Screenshot Git



```
MINGW64:/c/Users/Windows10/it-perbankan-finpro-kelompok3

Windows10@LAPTOP-GRB945JM MINGW64 ~
$ git clone https://github.com/ikrar557/it-perbankan-finpro-kelompok3
Cloning into 'it-perbankan-finpro-kelompok3'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 36 (delta 7), reused 21 (delta 2), pack-reused 0
Receiving objects: 100% (36/36), 25.63 MiB | 429.00 KiB/s, done.
Resolving deltas: 100% (7/7), done.

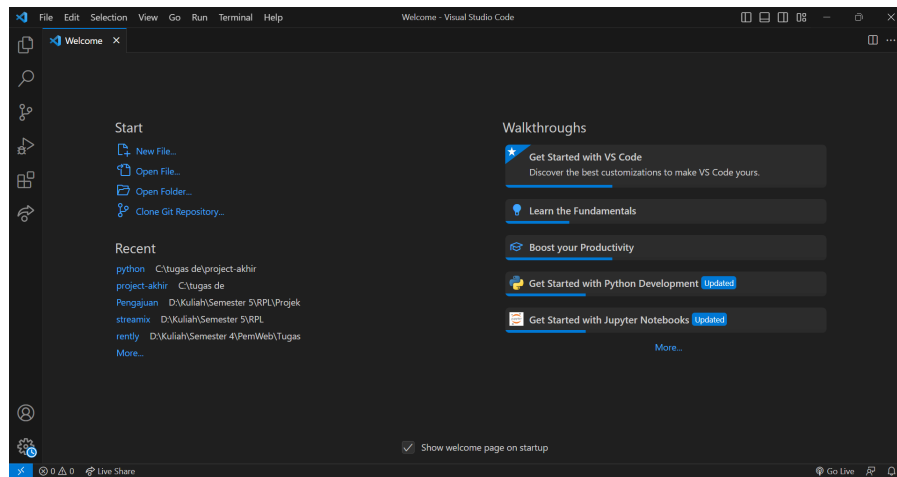
Windows10@LAPTOP-GRB945JM MINGW64 ~
$ git branch
fatal: not a git repository (or any of the parent directories): .git

Windows10@LAPTOP-GRB945JM MINGW64 ~
$ cd ^C

Windows10@LAPTOP-GRB945JM MINGW64 ~
$ it-perbankan-finpro-kelompok3
bash: it-perbankan-finpro-kelompok3: command not found

Windows10@LAPTOP-GRB945JM MINGW64 ~
$ cd it-perbankan-finpro-kelompok3
```

### Screenshot Visual Studio Code



## Summary

### Software Development Lifecycle (SDLC)

SDLC (Siklus Hidup Pengembangan Perangkat Lunak) adalah metodologi terstruktur untuk mengembangkan perangkat lunak dari awal hingga selesai yang terdiri dari beberapa tahap yang saling terkait:

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

Penerapan SDLC yang efektif dapat meningkatkan keberhasilan dan efisiensi dalam pengembangan aplikasi, memastikan pengiriman produk berkualitas tepat waktu, serta memberikan nilai tambah kepada pelanggan dan stakeholder. Terdapat beberapa varian model SDLC, sebagai berikut.

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

Git adalah sistem kontrol versi terdistribusi yang memungkinkan pengembang perangkat lunak untuk melacak perubahan kode, berkolaborasi dengan tim, dan mengelola revisi kode dengan efektif. Berikut beberapa perintah dasar Git yang sering digunakan:

1. `git init`: Untuk memulai repositori Git baru dalam direktori saat ini.
2. `git clone [URL]`: Untuk mengklon repositori Git yang ada dari URL ke direktori lokal.
3. `git status`: Untuk melihat status perubahan dalam repositori Anda.
4. `git add [nama_file]`: Untuk menambahkan perubahan atau file ke area staging.
5. `git commit -m "pesan commit"`: Untuk membuat commit dengan pesan yang menjelaskan perubahan yang telah Anda buat.
6. `git pull`: Untuk mengambil perubahan terbaru dari repositori remote ke repositori lokal Anda.
7. `git push`: Untuk mengirim perubahan lokal Anda ke repositori remote.
8. `git branch`: Untuk melihat daftar cabang (branch) yang ada dalam repositori.
9. `git checkout [nama_cabang]`: Untuk beralih ke cabang (branch) tertentu.
10. `git merge [nama_cabang]`: Untuk menggabungkan perubahan dari cabang lain ke cabang saat ini.

11. git log: Untuk melihat sejarah commit dalam repositori.
12. git remote -v: Untuk melihat daftar remote repositories yang terhubung dengan repositori lokal Anda.
13. git fetch: Untuk mengambil informasi tentang perubahan terbaru dari repositori remote tanpa menggabungkannya.
14. git diff: Untuk melihat perbedaan antara kode yang ada dan kode yang ada di area staging.
15. git reset [nama\_file]: Untuk menghapus file dari area staging.

## **Full Stack Web Database Management**

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). Database Management melakukan desain dan mengelola basis data untuk menyimpan, mengambil, dan memanipulasi data aplikasi. Jadi kesimpulannya adalah Full Stack Web Database Management adalah kemampuan untuk mengelola database dalam konteks pengembangan web yang melibatkan semua lapisan atau komponen dari tumpukan teknologi yang digunakan dalam proyek pengembangan web. Terdapat dua tipe database utama yang umum digunakan aplikasi: SQL (Structured Query Language) atau database relasional dan NoSQL (Not Only SQL) atau database non-relasional. Bahasa Querynya menggunakan SQL yang memungkinkan pengguna untuk melakukan operasi seperti SELECT , INSERT , UPDATE , dan DELETE.

## **Front End**

Frontend Web Development merupakan Pengembangan web front-end, juga dikenal sebagai pengembangan sisi klien untuk situs web atau Aplikasi Web sehingga pengguna dapat melihat dan berinteraksi dengannya secara langsung yang mempraktikkan secara langsung bahasa berikut:

1. **HTML** : blok bangunan paling dasar dari Web. Ini mendefinisikan arti dan struktur konten web.
2. **CSS** : Bahasa yang kami gunakan untuk menata halaman Web.
3. **Javascript** : bahasa pemrograman Web. JS biasanya digunakan untuk membuat web kita punya lebih interaktif.

## **Back End**

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. Berikut merupakan penjelasan apa saja yang diimplementasikan:

1. Bahasa Pemrograman Server-Side  
Bahasa pemrograman seperti Node.js (JavaScript), Python, Ruby, Java, PHP, C#, dan lain-lain, digunakan untuk menulis kode di sisi server.
2. Server Framework

Framework seperti Express.js untuk Node.js, Flask untuk Python, Ruby on Rails untuk Ruby, Spring untuk Java, dan Laravel untuk PHP

### 3. Database Management

Jenis database yang umum digunakan adalah SQL (MySQL, PostgreSQL, SQL Server) dan NoSQL (MongoDB, Firebase).

## Mobile End

Pengembangan aplikasi mobile adalah proses menggunakan konsep dan teknologi untuk membangun aplikasi yang berfungsi di perangkat mobile seperti smartphone dan tablet. Ini melibatkan beberapa aspek penting untuk memastikan kinerja optimal dan pengalaman pengguna yang baik. Aplikasi mobile dapat dibangun untuk berbagai platform seperti Android, iOS, dan Windows Phone, masing-masing dengan bahasa pemrograman dan lingkungan pengembangan khusus. Contohnya, aplikasi Android menggunakan Java atau Kotlin, sementara aplikasi iOS menggunakan Swift atau Objective-C. Pengembangan aplikasi mobile menggunakan IDE (Integrated Development Environment). IDE (Integrated Development Environment) adalah perangkat lunak yang digunakan dalam pengembangan aplikasi mobile. IDE menyediakan alat, penyunting kode, pengelola proyek, simulator perangkat, dan fasilitas debugging untuk mempermudah pengembangan aplikasi.

## Collaboration

Kolaborasi dalam Git merujuk pada berbagi dan bekerja sama dalam pengembangan perangkat lunak menggunakan Git sebagai sistem kontrol versi. Berikut yang dapat dilakukan pada GIT ketika kolaborasi.

1. Repo Bersama: Tim pengembang berbagi repositori Git yang sama untuk menyimpan kode sumber proyek.
2. Forking: Pengembang dapat melakukan fork (gandakan) repositori proyek utama ke repositori pribadi mereka untuk bekerja pada perubahan tanpa mengubah repositori utama.
3. Branching: Setiap pengembang dapat membuat cabang (branch) yang berbeda dalam repositori untuk mengembangkan fitur atau memperbaiki bug secara terisolasi.
4. Commit dan Push: Pengembang membuat perubahan dalam cabang mereka, membuat commit, dan mengirimkan (push) perubahan tersebut ke repositori bersama.
5. Pull Request: Pengembang yang melakukan fork dapat mengajukan pull request untuk menggabungkan perubahan mereka ke repositori utama. Tim pengembang utama dapat memeriksa dan mengulas perubahan tersebut sebelum menggabungkannya.
6. Konflik: Ketika ada konflik (perubahan yang bertentangan) antara perubahan dalam cabang yang berbeda, pengembang harus menyelesaikannya sebelum penggabungan (merge) bisa dilakukan.

Kolaborasi dalam Git memungkinkan tim pengembang untuk bekerja secara efisien dan terstruktur dalam pengembangan perangkat lunak, dengan memungkinkan berbagai kontribusi

dari anggota tim yang berbeda dan memastikan pengelolaan perubahan yang baik dalam proyek.