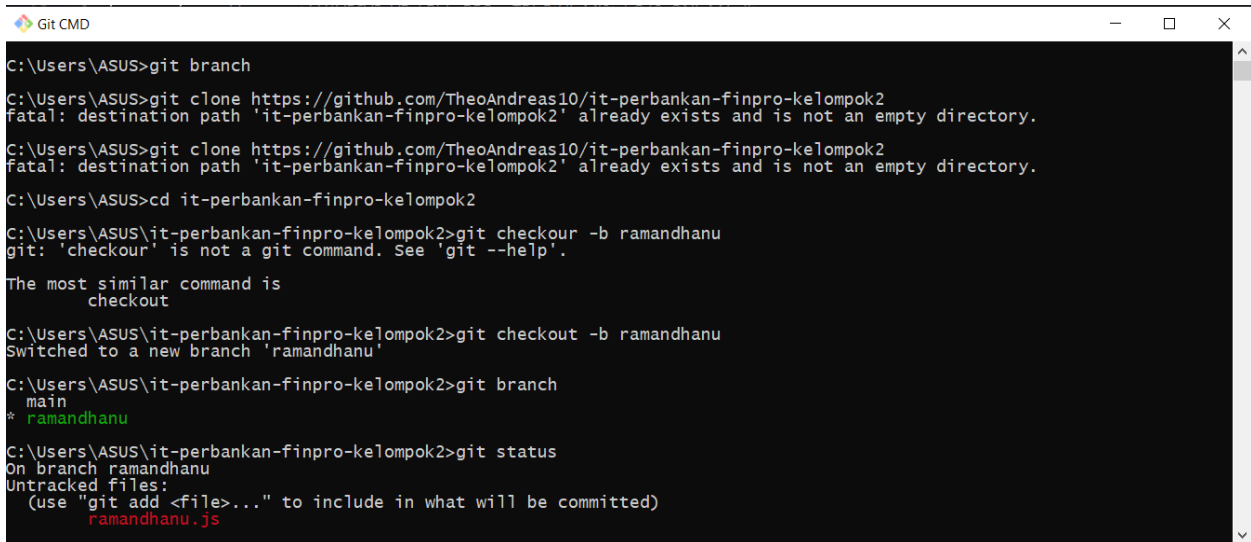


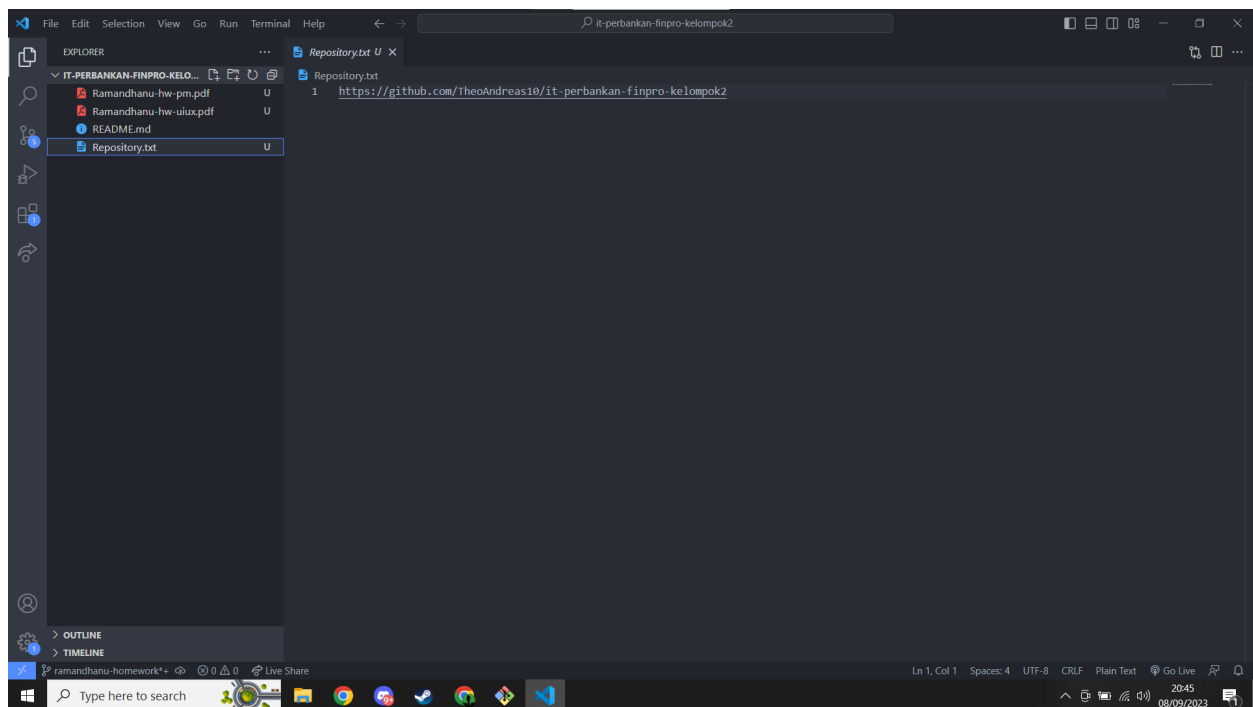
Screenshot Git



```
C:\Users\ASUS>git branch
C:\Users\ASUS>git clone https://github.com/TheoAndreas10/it-perbankan-finpro-kelompok2
fatal: destination path 'it-perbankan-finpro-kelompok2' already exists and is not an empty directory.
C:\Users\ASUS>git clone https://github.com/TheoAndreas10/it-perbankan-finpro-kelompok2
fatal: destination path 'it-perbankan-finpro-kelompok2' already exists and is not an empty directory.
C:\Users\ASUS>cd it-perbankan-finpro-kelompok2
C:\Users\ASUS\it-perbankan-finpro-kelompok2>git checkour -b ramandhanu
git: 'checkour' is not a git command. See 'git --help'.

The most similar command is
    checkout
C:\Users\ASUS\it-perbankan-finpro-kelompok2>git checkout -b ramandhanu
Switched to a new branch 'ramandhanu'
C:\Users\ASUS\it-perbankan-finpro-kelompok2>git branch
  main
* ramandhanu
C:\Users\ASUS\it-perbankan-finpro-kelompok2>git status
On branch ramandhanu
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ramandhanu.js
```

Screenshot Visual Studio Code



Soal 1 Summary

Full Stack Developer adalah seorang engineer yang bertanggung jawab dalam mengembangkan suatu aplikasi secara end-to-end, yaitu 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.

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.

Siklus SDLC

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

SDLC juga memiliki banyak model, setiap model SDLC memiliki kelebihan dan kelemahan tergantung pada jenis proyek dan kebutuhan organisasi. Pemilihan model SDLC yang tepat sangat penting untuk mencapai keberhasilan proyek pengembangan perangkat lunak.

Dengan mengintegrasikan Design Thinking dalam SDLC, tim pengembangan perangkat lunak dapat menciptakan produk yang lebih berorientasi pada pengguna, intuitif, dan sukses dalam memenuhi kebutuhan pengguna serta tujuan bisnis. Sifat iteratif dari Design Thinking memastikan perangkat lunak terus berkembang dan beradaptasi dengan perubahan kebutuhan pengguna dan dinamika pasar.

Langkah Desain Thinking

1. Empathize: Memahami kebutuhan pengguna
2. Define: Mendefinisikan masalah
3. Ideate: Membuat ide
4. Prototype: Membuat solusi
5. Test: Gather Umpan balik pengguna
6. Implement: Pengembangan software

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.