

SOAL:

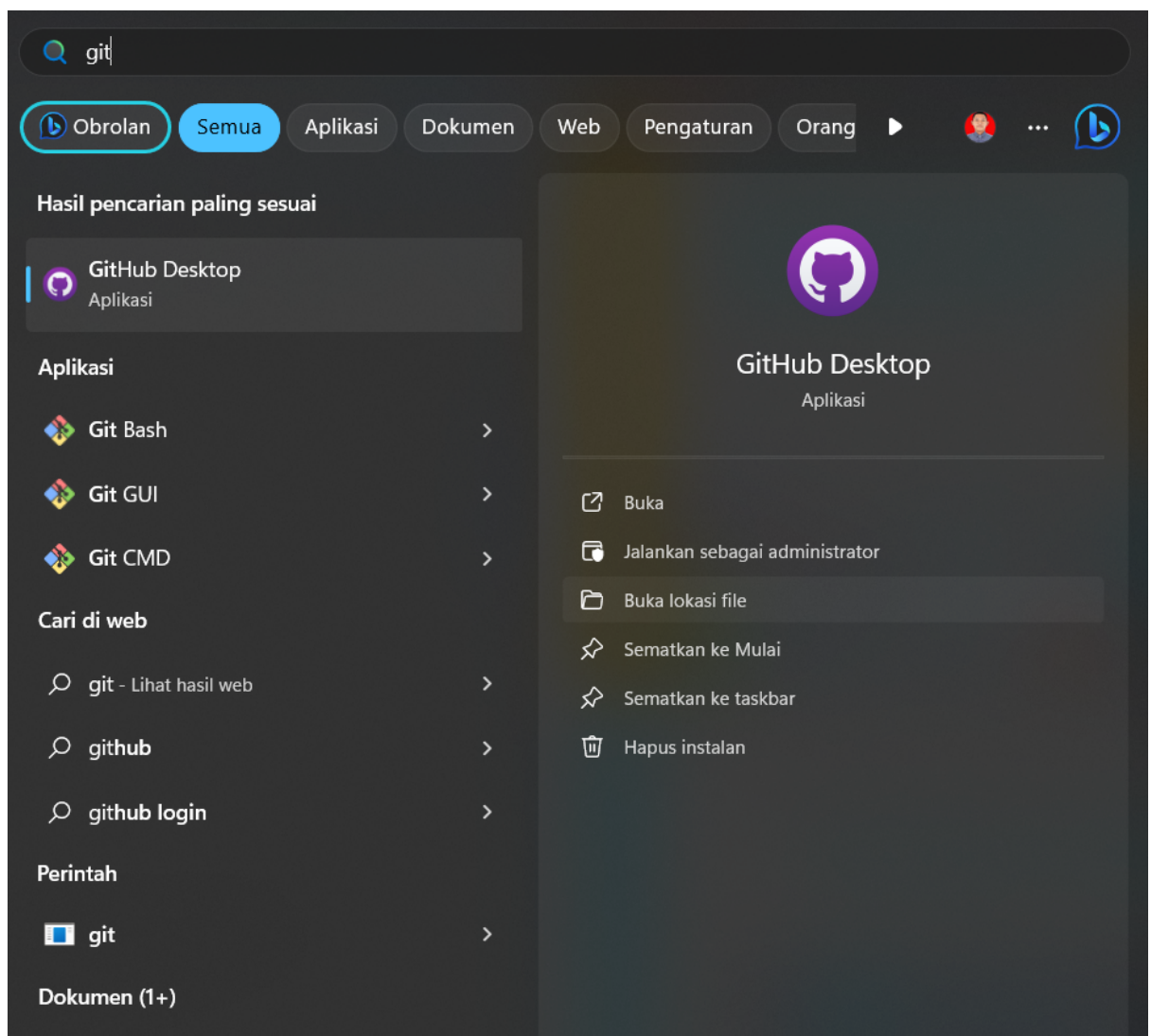
Silahkan teman teman melakukan proses instalasi beberapa tools dibawah ini :

- A. Git
- B. Visual Studio Code
- C. Browser

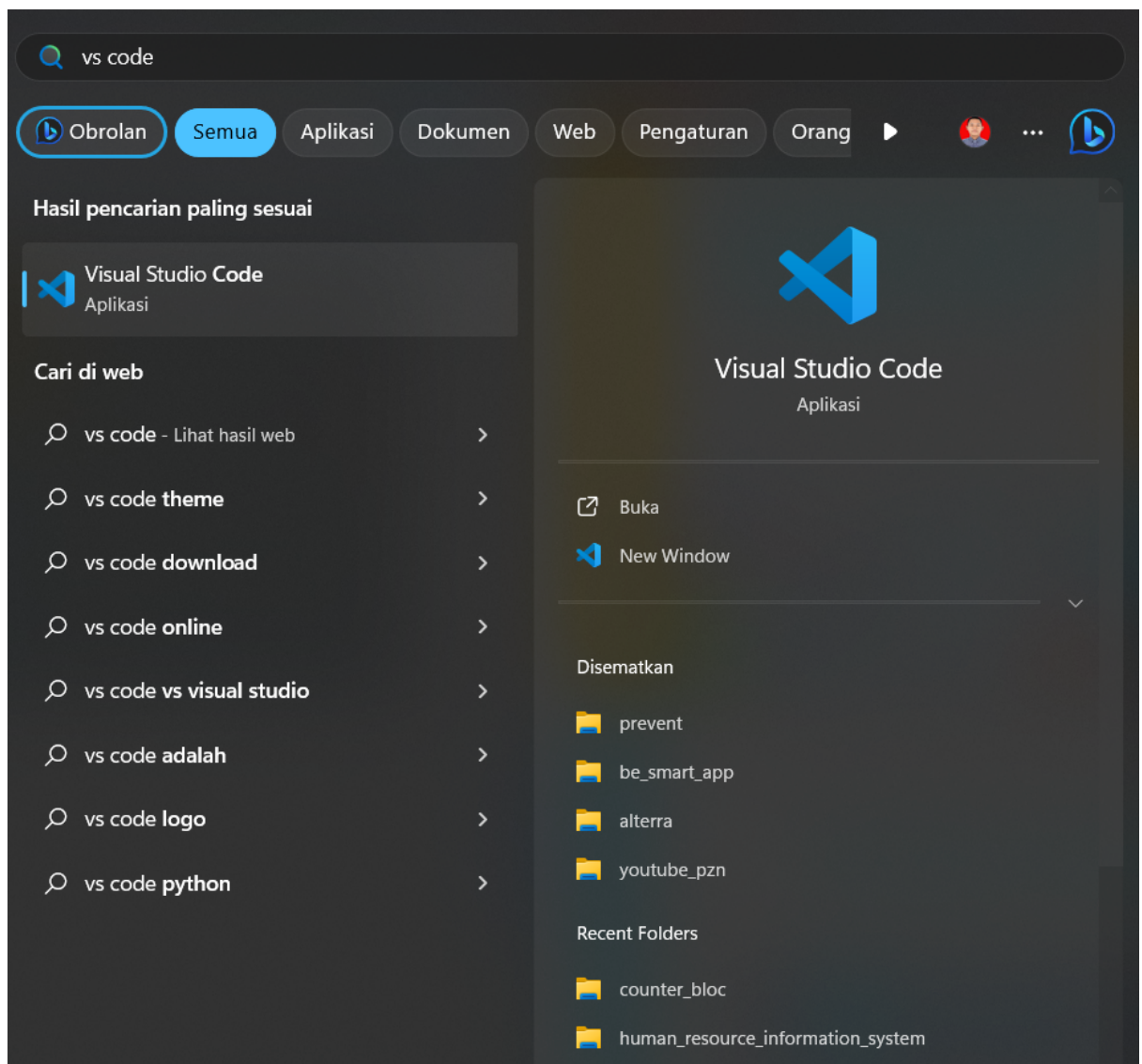
Setelah melakukan instalasi, buat sebuah summary untuk apa yang sudah dipelajari dalam materi pengenalan software engineer sebelumnya pada google docs.

JAWAB:

- A. Git



B. Visual Studio Code



C. Browser



Summary Full Stack Developer Career Path:

1. Introduction Full Stack Web/Mobile Developer

A. Scope Penting Full Stack Development

- Front End Development

Membangun antarmuka pengguna yang menarik dan interaktif menggunakan HTML, CSS, dan JavaScript. Menggunakan framework dan pustaka front-end, seperti React, Angular, Vue.js, atau jQuery, untuk mempercepat pengembangan dan meningkatkan efisiensi.

Menghubungkan komponen front-end dengan layanan back-end melalui API (Application Programming Interface) untuk berkomunikasi dengan server dan database. Menyelaraskan data dan tampilan antara sisi depan dan sisi belakang aplikasi.

- 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. Menggunakan bahasa pemrograman server-side seperti Node.js, Python, Ruby, Java, PHP, atau C#.

- Database Management

Mendesain dan mengelola basis data untuk menyimpan, mengambil, dan memanipulasi data aplikasi. Menggunakan teknologi database seperti MySQL, PostgreSQL, MongoDB, atau Firebase.

- 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. Menyelaraskan data dan tampilan antara sisi depan dan sisi belakang aplikasi.

- Version Control and Collaboration

Menggunakan sistem pengendalian versi, seperti Git, untuk mengelola perubahan kode dan kolaborasi dalam tim pengembang. Memastikan kode terus berkembang dengan aman dan sesuai dengan tujuan proyek.

- Mobile Development

Beberapa Pengembang Full Stack juga memiliki kemampuan untuk mengembangkan aplikasi mobile menggunakan framework seperti React Native, Flutter.

B. Dasar Dasar Front End Web Development

- HTML
- CSS
- Javascript

C. Dasar Dasar Back End Development

- Bahasa Pemrograman Server Side
- Server Framework
- Database Management

D. Dasar Dasar Database Management System

- Database Management System
- Tipe Database
- Bahasa Query

E. Dasar Dasar Mobile Development

- Platform Mobile
- Integrated Development Environment (IDE)

2. Skillset Full Stack Web/Mobile Developer

A. Pengembangan Aplikasi End to End

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.

B. Tahap - Tahap Pengembangan Aplikasi End to End

- Perencanaan dan Analisis

Tahap awal ini melibatkan pengumpulan kebutuhan dan pemahaman mendalam tentang tujuan aplikasi, sasaran pengguna, dan lingkungan operasional. Analisis kebutuhan dan riset pasar dilakukan untuk mengidentifikasi fitur utama yang harus dimasukkan dalam aplikasi.

- Desain

Proses desain melibatkan merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX) yang intuitif dan menarik. Pengembang juga merencanakan arsitektur aplikasi, termasuk pemilihan teknologi, database, dan framework yang sesuai.

- Pengembangan Front-End

Pada tahap ini, tim pengembang bekerja pada bagian depan aplikasi, menggunakan HTML, CSS, dan JavaScript untuk membuat tampilan dan interaksi yang menarik bagi pengguna. Pengembang mungkin juga menggunakan framework front-end seperti React, Angular, atau Vue.js untuk mempercepat pengembangan.

- Pengembangan Back-End

Tahap ini melibatkan pengembangan sisi server dan logika bisnis aplikasi. Pengembang menggunakan bahasa pemrograman server-side seperti Node.js, Python, Ruby, atau Java, dan menggunakan framework back-end seperti Express.js, Flask, atau Ruby on Rails.

- Integrasi dan Pengujian

Bagian depan dan belakang aplikasi harus diintegrasikan melalui API (Application Programming Interface) sehingga mereka dapat berkomunikasi dan berbagi data. Pengujian aplikasi dilakukan untuk memastikan semua fitur berfungsi dengan benar dan mengidentifikasi dan memperbaiki bug yang mungkin ada.

- Pemeliharaan dan Peningkatan

Setelah diluncurkan, aplikasi harus dipelihara dengan memperbaiki bug dan menangani perubahan lingkungan atau kebutuhan bisnis. Peningkatan terus menerus dilakukan untuk memperbarui fitur, meningkatkan kinerja, dan memastikan aplikasi tetap relevan dalam waktu yang berlanjut.

C. Kolaborasi Efektif Dengan Version Control

- 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 2 Version Control yang cukup populer yaitu:

- Git
- Mercurial

- Beberapa Manfaat Version Control Untuk Berkolaborasi

- Rekam Perubahan

Setiap kali pengembang membuat perubahan pada kode, sistem version control merekam detail perubahan tersebut.

- Pencatatan Riwayat

Version control memungkinkan tim untuk melihat riwayat lengkap dari semua perubahan yang terjadi pada proyek dari awal hingga saat ini.

- Pemecahan Konflik

Ketika dua atau lebih pengembang melakukan perubahan pada area kode yang sama, version control membantu mengidentifikasi dan menyelesaikan konflik.

- Pemulihan dengan mudah

Version control memungkinkan pengembang untuk memulihkan kode ke versi sebelumnya jika ada masalah atau bug yang terjadi, sehingga mengurangi resiko kehilangan pekerjaan.

- Penggunaan Version Control untuk Berkolaborasi

- Inisialisasi Proyek

Tim memulai proyek dengan membuat repository version control. Repository ini akan menyimpan semua kode sumber, file, dan perubahan yang dilakukan selama pengembangan.

- Pengembangan Paralel

Setiap anggota tim akan memiliki salinan repository pada komputernya sendiri. Mereka dapat bekerja secara paralel, membuat perubahan.

- Branching

Version control memungkinkan pembuatan cabang (branch) yang terpisah dari kode utama. Ini memungkinkan tim untuk mengisolasi perubahan dan fitur yang sedang dikembangkan.

- Merge

Setelah fitur atau perubahan selesai, cabang dapat digabungkan kembali ke cabang utama (biasanya disebut sebagai "merge").

- Pull Request

Di beberapa platform version control seperti GitHub, GitLab, dan Bitbucket, pull request adalah mekanisme yang memungkinkan pengembang untuk mengajukan perubahan mereka untuk ditinjau oleh anggota tim lain sebelum digabungkan ke cabang utama.

3. Tools Set Full Stack Web/Mobile Developer

Sebagai Pengembang Full Stack Web dan Mobile, Anda akan memerlukan kombinasi dari berbagai alat dan teknologi untuk secara efisien

membangun dan mengelola aplikasi. Berikut adalah beberapa set alat yang penting yang mungkin digunakan oleh Pengembang Full Stack:

- IDE - Code Editor
 - Visual Studio Code

- Version Control - Repository
 - GitHub
 - GitLab
 - Bitbucket

- Version Control - Git Tools
 - Sourcetree
 - GitLens

- Database Management System (DBMS)
 - PostgreSQL
 - MySQL
 - Oracle
 - MongoDB
 - Redis

- Application Programming Interface (API)
 - Postman
 - Swagger

- Tests dan Debugging
 - Jest
 - Mocha
 - JUnit

- Mobile Development
 - Framework React Native
 - Framework Flutter

- Layanan Cloud
 - AWS
 - Google Cloud
 - Azure
- CI/CD
 - Jenkins
 - Circleci
- Desain UI/UX
 - Adobe XD
 - Figma
 - Sketch