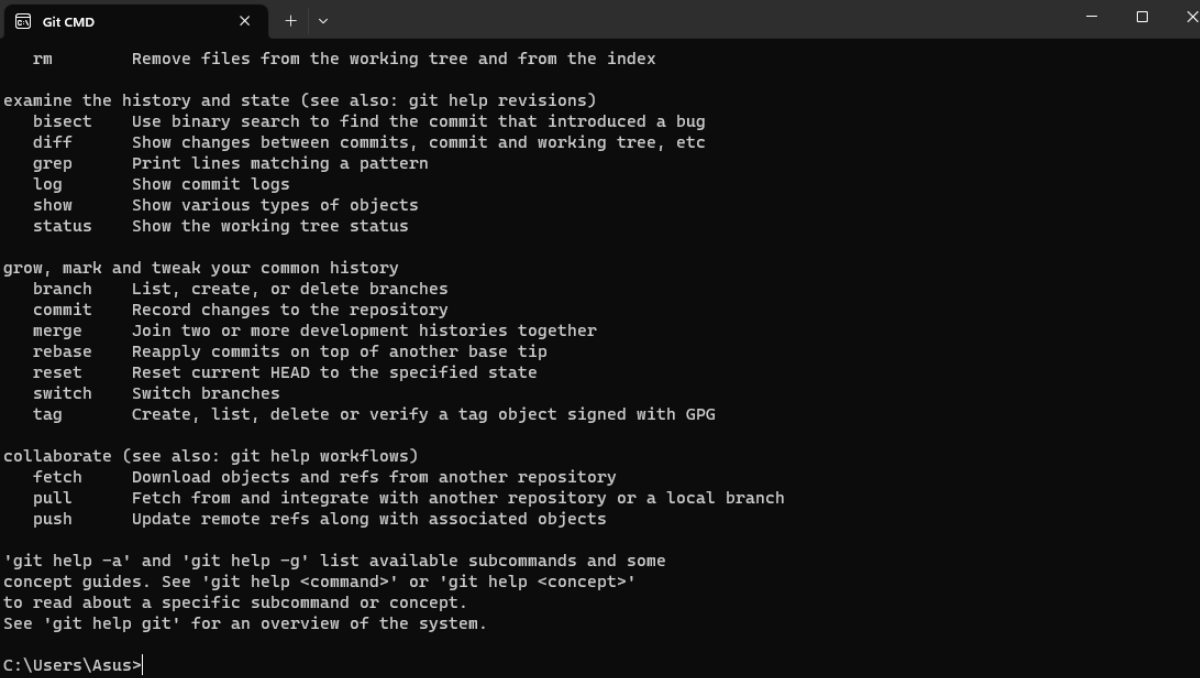


Nama : Alpin Apriliansyah Mohsa

Screenshoot GIT



```
Git CMD
rm      Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
bisect  Use binary search to find the commit that introduced a bug
diff    Show changes between commits, commit and working tree, etc
grep    Print lines matching a pattern
log     Show commit logs
show    Show various types of objects
status  Show the working tree status

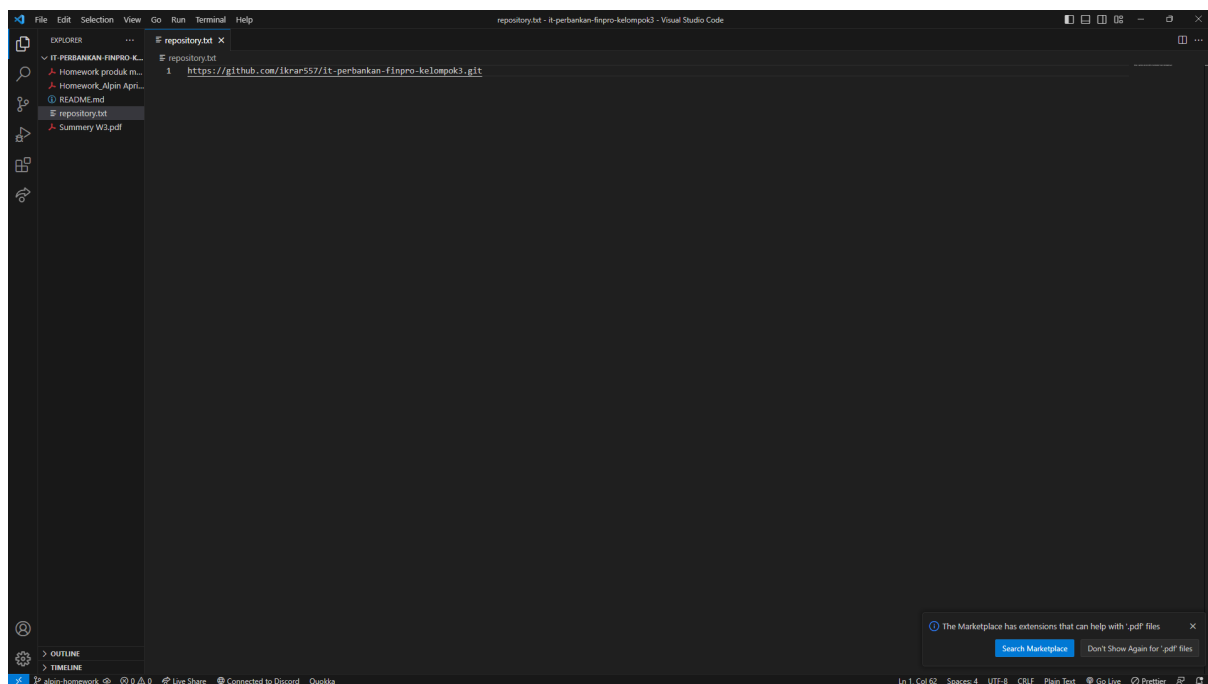
grow, mark and tweak your common history
branch  List, create, or delete branches
commit  Record changes to the repository
merge   Join two or more development histories together
rebase  Reapply commits on top of another base tip
reset   Reset current HEAD to the specified state
switch  Switch branches
tag     Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
fetch   Download objects and refs from another repository
pull    Fetch from and integrate with another repository or a local branch
push    Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

C:\Users\Asus>
```

Screenshoot VsCode



Introduction to Software Engineering Full Stack Developer Career Path

Definisi

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 pada Full Stack

- **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.
- **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.

Dasar-Dasar Frontend Web Development

Pengembangan web front-end, juga dikenal sebagai pengembangan sisi klien adalah praktik pembuatan HTML, CSS, dan JavaScript untuk situs web atau Aplikasi Web sehingga pengguna dapat melihat dan berinteraksi dengannya secara langsung.

- **HTML (HyperText Markup Language)** adalah blokbangunan paling dasar dari Web. Ini mendefinisikan arti dan struktur konten web.
- **CSS** adalah bahasa yang kami gunakan untuk menata halaman Web. Dan biasanya CSS digunakan untuk kosmetik dari sebuah web yang kita punya.
- **JavaScript** adalah bahasa pemrograman paling populer di dunia. JavaScript adalah bahasa pemrograman Web. JS biasanya digunakan untuk membuat web kita punya lebih interaktif.

Selain itu pada Frontend memiliki beberapa framework populer yaitu :

- React adalah library JavaScript yang digunakan untuk membangun user interface yang interaktif berbasis component.
- Vue.js adalah kerangka kerja JavaScript yang bersifat progresif, bersumber terbuka untuk membangun antarmuka pengguna.
- AngularJS adalah kerangka kerja web berbasis JavaScript sumber terbuka dan gratis yang dihentikan untuk mengembangkan aplikasi satu halaman.

Dasar-Dasar Backend Development

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, selain itu ada beberapa framework pada Backend Dev .

- 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.
- Server Framework
Framework seperti Express.js untuk Node.js, Flask untuk Python, Ruby on Rails untuk Ruby, Spring untuk Java, dan Laravel untuk PHP
- Database Management
Jenis database yang umum digunakan adalah SQL (MySQL, PostgreSQL, SQL Server) dan NoSQL (MongoDB, Firebase).

Dasar-Dasar Database Management

Serangkaian konsep dan teknik yang digunakan untuk mengelola data dalam sebuah aplikasi atau sistem. Database merupakan bagian kritis dari aplikasi karena menyimpan dan mengorganisir informasi yang diperlukan untuk menjalankan aplikasi dengan benar.

- Database Management System
Perangkat lunak yang memungkinkan pengguna untuk mengelola dan mengakses data dalam database. DBMS menyediakan antarmuka untuk berinteraksi dengan database
- Tipe Database
Ada dua tipe database utama yang umum digunakan dalam pengembangan aplikasi: SQL (Structured Query Language) atau database relasional dan NoSQL (Not Only SQL) atau database non-relasional.
- Bahasa Query
SQL adalah bahasa query yang digunakan untuk berinteraksi dengan database SQL. Bahasa query memungkinkan pengguna untuk melakukan operasi seperti SELECT , INSERT , UPDATE DELETE

Ada beberapa populer Database Management Full Stack Web Development, selain itu :

Sql : Oracle, Postgre, MySql

No Sql : Mongo DB dan Redis

Dasar-Dasar Mobile Dev

Mobile Dev adalah serangkaian konsep dan teknologi yang digunakan untuk membangun aplikasi yang dapat dijalankan di perangkat mobile, seperti smartphone dan tablet. Pengembangan aplikasi mobile mencakup beberapa aspek penting untuk memastikan aplikasi dapat berjalan dengan baik dan memberikan pengalaman pengguna yang optimal.

- Platform Mobile

Aplikasi mobile dapat dikembangkan untuk berbagai platform, termasuk Android, iOS, dan Windows Phone. Setiap platform memiliki bahasa pemrograman dan lingkungan pengembangan yang khas. Misalnya, aplikasi Android dapat ditulis dalam Java atau Kotlin, sedangkan aplikasi iOS menggunakan Swift atau Objective-C.

- IDE (Integrated Development Environment)

IDE adalah perangkat lunak yang digunakan untuk mengembangkan aplikasi mobile. IDE menyediakan alat bantu, penyunting kode, pengelola proyek, simulator perangkat, dan fasilitas debugging untuk mempermudah proses pengembangan.

Ada beberapa framework yang populer yang bisa kita gunakan, antara lain adalah : ReactNative dan 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.

Tahap tahap 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 (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.

Selain itu ada beberapa Popular Version Control Full Stack Web Development, antara lain : git dan mercurial

Manfaat Version Control untuk Berkolaborasi

- Rekam Perubahan
- Pencatatan Riwayat
- Pemecahan Konflik
- Pemulihan Mudah

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:

Tools sets Sebagai Full Stack Developer

IDE - Code Editor

- VsCode

Version Control - Repository

- Git
- Github
- BitBucket

Version Control - Git Tools.

- Sourcetree
- GitLens

DBMS

- PostgreSQL
- Mysql
- Oracle
- mongoDB
- Redis

API

- Postman
- Swagger

Test dan Debugging

- jest
- Mocha chai
- JUnit5

Mobile Development

- React native
- Flutter

Layanan Cloud

- aws
- google cloud
- azure

CI/CD

- jenkins
- circleci

Desain UI/UX

- Figma
- Sketch

Introduction to Software Engineering SDLC & Design Thinking Implementation

Apa itu SDLC?

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 Fase fase pada SDLC

1. 1. Perencanaan dan Analisis

Tahap pertama ini melibatkan identifikasi masalah atau kebutuhan bisnis yang perlu diselesaikan oleh perangkat lunak. Para pemangku kepentingan berinteraksi untuk mengumpulkan persyaratan dan menentukan ruang lingkup proyek.

2. Desain

Di tahap ini, perangkat lunak dirancang secara rinci berdasarkan persyaratan yang telah dikumpulkan. Desain mencakup arsitektur sistem, antarmuka pengguna, dan desain database.

3. Pengembangan

Tahap ini melibatkan implementasi rancangan perangkat lunak yang telah disetujui sebelumnya. Para pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.

4. Pengujian

Setelah perangkat lunak dikembangkan, tahap pengujian dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditentukan. Pengujian mencakup verifikasi fungsionalitas, kinerja, keamanan, dan kualitas keseluruhan perangkat lunak.

5. Penerapan

Tahap ini melibatkan implementasi rancangan perangkat lunak yang telah disetujui sebelumnya. Para pengembang menulis kode untuk menghasilkan produk perangkat lunak yang berfungsi.

6. Pemeliharaan

Setelah perangkat lunak diimplementasikan, pemeliharaan dilakukan untuk memperbaiki bug, meningkatkan fitur, dan menjaga perangkat lunak agar tetap sesuai dengan perubahan kebutuhan bisnis.

Model Model SDLC

- Waterfall
Waterfall model adalah model SDLC yang linier dan berurutan. Setiap tahap dalam model ini harus selesai sebelum memulai tahap berikutnya.
- Model V-Shaped
Model V-Shaped adalah model yang terkait erat dengan model waterfall, tetapi menekankan pada pengujian.
- Model Prototype
Model Prototype adalah model pengembangan perangkat lunak yang bertujuan untuk menciptakan prototipe atau contoh awal sebelum mengembangkan versi finalnya.
- Model Spiral
Model ini menggabungkan elemen model spiral dengan pendekatan inkremental.
- Model Iterative Incremental
Model ini melibatkan pengulangan siklus pembangunan dan peningkatan perangkat lunak dalam tahapan-tahapan kecil.
- Model Big Bang
Model Big Bang adalah model yang kurang terstruktur, di mana semua tahapan pengembangan dilakukan tanpa perencanaan yang detail.
- Model Agile Model
Model Agile adalah pendekatan kolaboratif dan iteratif yang berfokus pada pengiriman perangkat lunak secara berkala dan inkremental.

Steps Design Thinking

- Empathize: Understand User Needs
Pada tahap ini, fokus pada memahami secara mendalam pengguna akhir dan kebutuhan mereka, keinginan, serta masalah yang dihadapi.
- Define: Define the Problem
Pada tahap ini, informasi yang dikumpulkan selama fase empati dianalisis untuk menentukan masalah dan menetapkan tujuan yang jelas untuk proyek.

- **Ideate: Generate Ideas**
Fase ideasi mendorong pemikiran kreatif dan menghasilkan berbagai solusi potensial.
- **Prototype: Build Quick and Iterative Solutions**
Pada tahap ini, fokus pada menciptakan representasi nyata dari ide-ide yang dipilih. Prototype digunakan untuk mengumpulkan umpan balik dan memvalidasi asumsi.
- **Test: Gather User Feedback**
Tahap pengujian melibatkan pengumpulan umpan balik dari pengguna nyata untuk memvalidasi solusi-solusi tersebut.
- **Implement: Develop the Software**
Pada tahap ini, desain diterjemahkan ke dalam kode yang sebenarnya dan diimplementasikan.

Introduction to Software Engineering Basic Git & Collaborating Using Git

Dengan perkembangan teknologi dan perangkat lunak, terminal tetap menjadi alat penting bagi para pengembang perangkat lunak, administrator sistem, dan pengguna teknis lainnya. Meskipun antarmuka grafis semakin canggih dan populer, terminal tetap memberikan fleksibilitas dan kekuatan untuk melakukan tugas-tugas khusus dan otomatisasi dalam lingkungan komputer modern.

Ada beberapa Command Line Dasar antara lain pada gambar di bawah ini :

Command	Windows	Linux / macOS	Description
List Files	<code>`dir`</code>	<code>`ls`</code>	List files and directories in the current folder
Change Directory	<code>`cd`</code>	<code>`cd`</code>	Change current working directory
Make Directory	<code>`mkdir`</code>	<code>`mkdir`</code>	Create a new directory
Remove Directory	<code>`rmdir`</code>	<code>`rm -r`</code>	Remove a directory
Delete Files	<code>`del`</code>	<code>`rm`</code>	Delete files
Copy Files	<code>`copy`</code>	<code>`cp`</code>	Copy files
Move / Rename	<code>`move`</code>	<code>`mv`</code>	Move or rename files/directories

Command	Windows	Linux / macOS	Description
Display File	<code>`type`</code>	<code>`cat`</code>	Display file content
Display Text	<code>`echo`</code>	<code>`echo`</code>	Display text or enable/disable echoing of commands
List Processes	<code>`tasklist`</code>	<code>`ps`</code>	List running processes
Terminate Process	<code>`taskkill`</code>	<code>`kill`</code>	Terminate processes

Command	Windows	Linux / macOS	Description
Print Directory	N/A	<code>`pwd`</code>	Print the current working directory
Create File	N/A	<code>`touch`</code>	Create an empty file or update timestamp
Change Permissions	N/A	<code>`chmod`</code>	Change file permissions
Change Ownership	N/A	<code>`chown`</code>	Change file ownership
Search in Files	N/A	<code>`grep`</code>	Search for patterns in files
Display Manual	N/A	<code>`man`</code>	Display the manual page for a command
Execute as Admin	N/A	<code>`sudo`</code>	Execute a command with administrative privileges
Ping	N/A	<code>`ping`</code>	Send ICMP echo requests to check network connectivity
Network Config	<code>`ipconfig`</code>	<code>`ifconfig`</code>	Display network interface configurations
Secure Shell	N/A	<code>`ssh`</code>	Connect to a remote host using SSH

Apa itu GIT?

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.

Dasar Dasar Command GIT

- `git init`
Menginisialisasi direktori sebagai repositori Git kosong
- `git clone`

Menduplikasi repositori Git yang sudah ada ke direktori lokal.

- `git status`
Menampilkan status perubahan yang belum dikomit di repositori lokal.
- `git add`
Menambahkan perubahan ke area persiapan (staging area) untuk disiapkan menjadi commit.
- `git commit`
Membuat commit dari perubahan yang sudah di-staging dan menambahkan pesan commit.
- `git push`
Mengirimkan commit ke repositori jarak jauh (remote repository).
- `git pull`
Mengambil commit terbaru dari repositori jarak jauh dan menggabungkannya ke repositori lokal.
- `git branch`
Menampilkan daftar cabang (branch) yang ada di repositori dan menunjukkan cabang aktif.
- `git checkout`
Beralih ke cabang lain atau ke commit tertentu.
- `git merge`
Menggabungkan perubahan dari satu cabang ke cabang aktif.
- `git log`
Menampilkan daftar commit beserta riwayatnya dalam repositori.
- `git remote`
Menampilkan daftar repositori jarak jauh yang terhubung dengan repositori lokal.
- `git fetch`
Mengambil informasi terbaru dari repositori jarak jauh tanpa menggabungkan perubahan.
- `git diff`
Menampilkan perbedaan antara versi yang sudah di-staging dengan versi sebelumnya.
- `git reset`
Mengembalikan file yang sudah di-staging ke direktori kerja sebelumnya.