**Dokumentasi download Git dan Vscode**

**Materi 1 Fullstack Web Developer Career Path**

Materi oleh: Rizky Fadilah

1. **Introduction Full Stack Web/Mobile Developer**
    - Pengertian dari pengembangan full stack (full stack developer) merujuk pada pengembangan seluruh aplikasi end to end dari sisi depan (front-end) hingga sisi belakang (back end) dan dalam beberapa kasus, hingga sisi klien (client side). Scope penting pada full stack developer adalah
        1. Front end developer
        2. Backend developer
        3. Database management
        4. Integrasi of front end and backend developer
        5. Versi control and collaboration, mobile developer.
    - Dasar dasar frontend web developer
        1. HTML, digunakan untuk membuat struktur konten web
        2. CSS, digunakan untuk menata atau mempercantik halaman web
        3. Javascript, membuat website yang di kembangkan lebih interaktif
    - Framework yang populer untuk full stack adalah react, vue.js, angularjs.
    - Dasar backend development
    - Backend bertanggung jawab untuk memproses permintaan pengguna.
    - Bahasa pemrograman sever side: Node.js, python, ruby, php, c#, dll
    - Server framework: express.js untuk Node js, Flask untuk python, rubyon rails untuk ruby, spring untuk java, dan Laravel untuk php
    - Database Management: SQL (MySQL, PostgreSQL, SQL Server) dan noSQL (MongoDB, Firebase)
    - Mobile Developer: Flutter, React Native

2. **Skillset Full Stack Web/Mobile Developer**
    - Pengembangaan aplikasi end to end adalah pengembangan perangkat lunak yang mencakup keseluruhan siklus pembuatan aplikasi dari tahap perancangan hingga tahap pengujian dan implementasi. Tujuannya adalah untuk menghasilkan aplikasi yang lengkap, fungsional, dan siap digunakan oleh pengguna.
    - Tahap tahap pengembangan aplikasi end to end adalah
        1. Perancangan dan analisis
        2. Desain
        3. Pengembangan font end
        4. Pengembangan backend
        5. Integrasi dan pengujian
        6. Pemeliharaan dan peningkatan
    - Kolaborasi efektif dalam pengembangan perangkat lunak yang populer adalah Git dan Mercurial
    - Manfaat dari version control untuk kolaborasi adalah
        1. Rekam perubahan
        2. Pencataan Riwayat
        3. Pemecahan konfilk
        4. Pemulihan mudah.
    - Tahapan penggunaan version control
        1. Inisialisasi proyek
        2. Pengembangan pararel
        3. Branching
        4. Marge
        5. Pull request

3. **Tools Full Stack Web/Mobile Developer**
    - Tools yang digunakan untuk full stack developer adalah
        1. Text editor : Vscode
        2. Version control repository : GitHub, GitLab, Bitbucket
        3. Version control – git tools : Sourcertree dan GitLens
        4. DBMS : MySQL, Oracle, redis, MongoDB, PostgreSQL
        5. API : Postman, swagger
        6. Test dan Debugging : Ject, mocha, chai, Junit5
        7. Mobile Developer : Flutter, React Native
        8. Layanan clode : Google Cloud, Azure, AWS
        9. CI/CD : jenskins, circleci
        10. Desain UI/UX : Figma, Sketch

**Materi 2 SDLC & Design Thinking Implementation**

Materi oleh: Rizky Fadilah

1. **What is SDLC**
    - SDLC merupakan rangkaian proses yang terstruktur dan metodologi yang digunakan untuk mengembangkan perangkat lunak dari awal hingg akhir. SDLC terdiri dari serangkaian tahapan yang saling terkait dan dilakukan secara berurutan untuk memastikan bahwa pengembangan perangkat lunak berjalan dengan baik sesuai dengan tujuan yang ditentukan.
    - Siklus SDLC adalah
        1 Perencanaan dan analisis
        2 Desain
        3 Pengembangan
        4 Pengujian
        5 Penerapan
        6 Pemeliharaan
    - Manfaat penggunaan SDLC
        1. Prediktabilitas dan pengendalian
        2. Peningkatan kualitas perangkat lunak
        3. Pengelolaan risiko yang lebih baik
        4. Efisiensi tim dan kolaborasi
        5. Memenuhi kebutuhan pengguna
        6. Penghematan biaya dan waktu
        7. Meningkatkan pengawasan dan evaluasi
        8. Peningkatan dokumenasi

2. **Model-model Software Development Life Cycle (SDLC)**
    - Model-model SDLC adalah sebagai berikut:
        1. Waterfall Model
        2. V-shaped Model
        3. Prototype Model
        4. Spiral Model
        5. Interative Incremental Model
        6. Big Bang Model
        7. Agile Model

3. **Design Thinking Implementation**
    - Steps Design Thinking
        1. Empathize: Understand User Needs
        2. Define: Define the problem
        3. Ideate: Generate Ideas
        4. Prorotype: Build Quick and Iterative Solutions
        5. Test: gather User Feedback
        6. Implement: Develop Software

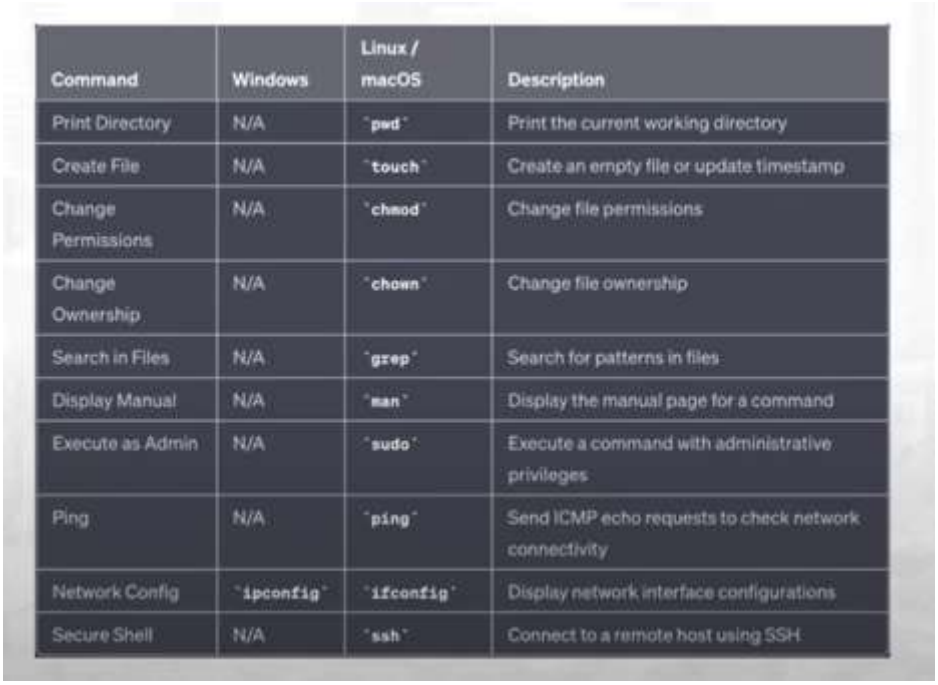
**Materi 3 Basic Git & Collaborating Using Git**

Materi oleh: Rizky Fadilah

1. **Terminal and IDE**
    - Sejarah singkat Terminal
        1. Pada tahun 1960 adalah Early Terimals, terminal hanya berupa layar dan keyboard. User memberikan perintah ke computer kemudian perintah akan di proses menggunakan layar utama
        2. Pada 1970 adalah DEC VT100, menawarkan fitur yang cangkih pada zamannya dengan menyediakan fitur layar dan kursor yang dapat di pindah pindah. Kemudian UNIX and Shell pengguna dapat memberikan peritah dengan beberapa garis perintah.
        3. Pada tahun 1980 adalah Terminal Emulator mengefaluasikan terminal fisik dengan memungkinkan pengguna berinteraksi dengan sistem.
        4. Personal computers with graphical user interface
        5. Modern Terminals Development adalah
    - Command line dasar

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

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



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

**2. Installing, initializing and committing GIT**

- Kontrol versi adalah metode yang digunakan untuk melacak dan mengelola perubahan dalam kode sumber atau berkas proyek.
- Git merupakan salah satu sistem kontrol versi terdistribusi yang paling populer dan kuat. Berikut adalah Langkah untuk memahami kontrol versi dan Git
    1. Sistem Kontrol Versi Terpusat (Centralized Version Control System)
    2. Sistem Kontrol Versi Terdistribusi (Distributed Version Control System)
- 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.
- Berikut Langkah install Git pada windows
    1. Download dan install git pada link: https://git- scm.com/download/win
- Install GIT pada Linux
    1. Menginstal Git melalui manajer paket seperti APT, YUM, atau DNF.
    2. Menginstal dari sumber pada distribusi Linux atau jalankan perintah ini pada terminal sudo apt install git-all
    3. Memverifikasi instalasi dan mengatur PATH environment variable
- Install GIT pada macOS
    1. Instruksi rinci untuk menginstal Git di macOS.
    2. Memanfaatkan manajer paket macOS atau menginstal dari sumber
    3. atau dapat menginstall terlebih dahulu Xcode Command Line Tools
    4. Memverifikasi instalasi dan mengatur variabel lingkungan PATH
- Dasar-dasar command GIT


Dasar Dasar Command GIT
01 git init
Menginisialisasi direktori sebagai repositori Git kosong
git init
02 git clone
Menduplikasi repositori Git yang sudah ada ke direktori lokal.
git clone https://github.com/username/repo.git
03 git status
Menampilkan status perubahan dikomit di repositori lokal.
git status
04 git add
Menambahkan perubahan ke area persiapan (staging area) untuk disiapkan menjadi commit.
git add file1.txt
git add .


Continue...
05 git commit
Membuat commit dari perubahan yang sudah di-staging dan menambahkan pesan commit.
git commit -m "Menambahkan fitur login"
06 git push
Mengirimkan commit ke repositori jarak jauh (remote repository).
07 git pull
Mengambil commit terbaru dari jarak jauh dan menggabungkan repositori lokal.
git pull origin main
08 git branch
Menampilkan daftar cabang (branch) yang ada di repositori dan menunjukkan cabang aktif.
git branch


Continue...
09 git checkout
Beralih ke cabang lain atau ke commit tertentu.
git checkout feature-branch
git checkout abc1234 (commit hash)
10 git merge
Menggabungkan perubahan dari satu cabang ke cabang aktif.
git merge feature-branch
11 git log
Menampilkan daftar commit be riwayatnya dalam repositori.
git log
12 git remote
Menampilkan daftar repositori jarak jauh yang terhubung dengan repositori lokal.
git remote -v
13 git fetch
Mengambil informasi terbaru dari repositori jarak jauh tanpa menggabungkan perubahan.
git fetch origin
21


### 3. Collaborating Using Git