

MODUL 2

GIT

A. MAKSUD DAN TUJUAN

1. Memahami konsep dasar version control dan peran Git dalam pengembangan perangkat lunak.
2. Membuat Akun GitHub
3. Menginstal dan mengonfigurasi Git di lingkungan pengembangan.
4. Menggunakan perintah dasar Git untuk mengelola proyek, termasuk commit, branch, merge, dan bekerja dengan remote repository seperti GitHub.
5. Mengelola proyek secara kolaboratif menggunakan Git dan GitHub.

B. ALAT, BAHAN DAN PERANGKAT

Alat, bahan, perangkat lunak, perangkat keras yang digunakan adalah:

1. Perangkat Keras: Komputer atau laptop dengan sistem operasi Windows.
2. Perangkat Lunak:
 - a. Git (dapat diunduh dari git-scm.com).
 - b. Teks editor seperti Visual Studio Code.
 - c. Terminal atau Command Prompt.
 - d. Browser web untuk mengakses GitHub.
3. Akun GitHub: Untuk mengelola repository remote.

C. DASAR TEORI

Git adalah teknologi yang digunakan untuk melacak perubahan dalam suatu proyek dan membantu beberapa orang berkolaborasi dalam proyek tersebut. Secara sederhana, proyek yang dikontrol versinya oleh Git terdiri dari sebuah folder dengan file-file di dalamnya, dan Git melacak perubahan yang dilakukan pada file-file tersebut. Ini memungkinkan Anda untuk menyimpan berbagai versi dari pekerjaan yang sedang anda lakukan, itulah sebabnya Git disebut sebagai *version control*.

Git diciptakan oleh Linus Torvalds untuk mengontrol versi pekerjaan yang dilakukan pada proyek pengembangan perangkat lunak besar yang disebut kernel Linux. Namun, karena Git dapat melacak perubahan pada berbagai jenis file, Git dapat digunakan untuk berbagai macam proyek.

Secara ringkas, Git adalah sistem kontrol versi yang dapat diunduh ke komputer Anda, memungkinkan Anda melacak riwayat suatu proyek dan berkolaborasi dengan orang lain.

Perintah Dasar Git

- `git init`: Menginisialisasi repository Git di dalam direktori proyek.
- `git clone`: Menggandakan repository yang ada ke komputer lokal.
- `git status`: Menampilkan status perubahan dalam working directory.
- `git add`: Menambahkan file atau perubahan ke staging area.
- `git commit`: Merekam perubahan dari staging area ke repository.
- `git branch`: Membuat dan mengelola cabang (branch) dalam proyek.
- `git merge`: Menggabungkan perubahan dari satu branch ke branch lain.
- `git pull`: Mengambil perubahan terbaru dari remote repository dan menggabungkannya dengan local repository.
- `git push`: Mengirim perubahan dari local repository ke remote repository.

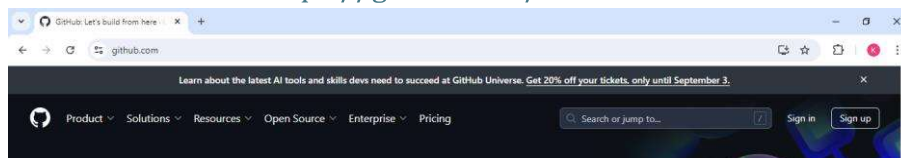
D. PRE-TEST

1. Apa tujuan dari menggunakan version control dalam pengembangan perangkat lunak?
2. Jelaskan perbedaan antara `git add` dan `git commit`!
3. Bagaimana cara membuat branch baru di Git?

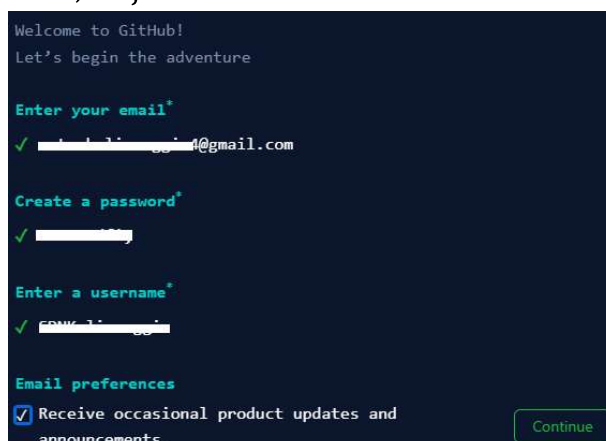
E. PRAKTIKUM

Membuat Akun GitHub.

1. Buka situs GitHub <https://github.com/>



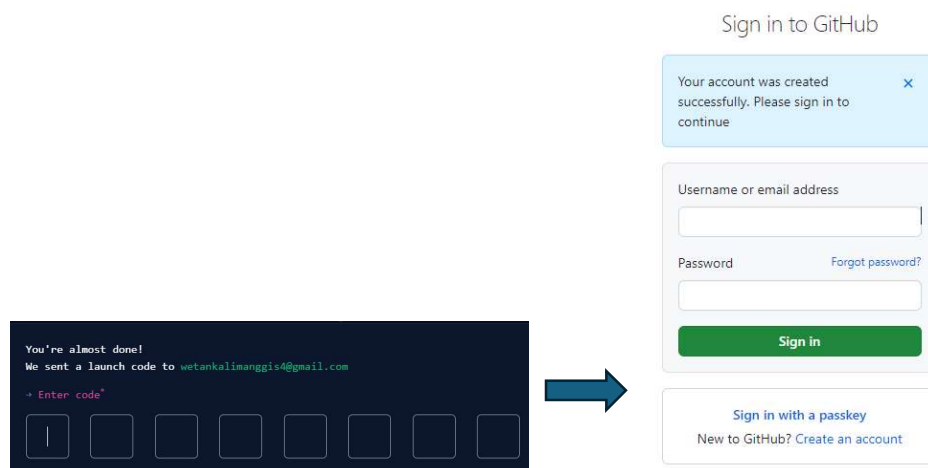
2. Klik tombol "Sign up" di pojok kanan atas.
3. Isi informasi yang diminta (email, password, username) sampai benar-benar valid, lanjut klik tombol **Continue**.



4. Ikuti arahan selanjutnya untuk mendapatkan kode verifikasi yang akan dikirim ke email anda.



5. Buka email pastikan anda mendapatkan email kode verifikasi dari GitHub.
6. Isikan kode verifikasi kedalam kotak isian yang tersedia, jika benar maka halaman selanjutnya anda akan diminta login. Artinya sampai disini anda sudah memiliki akun di GitHub.



7. Setelah berhasil login selanjutnya selesaikan pengisian profile termasuk status pengguna dimana pada modul praktikum ini akan menggunakan yang free.

Instalasi Git.

1. Unduh Git dari <https://git-scm.com/>
2. Ikuti petunjuk instalasi untuk sistem operasi anda (Windows, macOS, atau Linux).
3. Setelah instalasi, buka terminal atau command prompt, dan ketik **git --version** untuk memastikan Git terinstal dengan benar.
4. Tekan Ctrl + ` (backtick) atau buka **View > Terminal** untuk membuka terminal di Visual Studio Code.
5. Konfigurasi nama pengguna dan email dengan perintah berikut:

```
bash Copy code

git config --global user.name "Nama Anda"
git config --global user.email "email@example.com"
```

Mengelola Proyek dengan Git.

1. Inisialisasi repository Git baru di dalam direktori proyek anda dengan perintah **git init** pada bagian terminal Visual Studio Code.
2. Tambahkan file Latihan_01_1.html yang ada pada folder Modul_01 ke staging area dengan perintah **git add Modul_01/Latihan01_01.html**.
3. Periksa status repository anda dengan perintah **git status**, yang akan menunjukkan bahwa file Latihan01_01.html siap untuk di-commit.
4. Commit perubahan yang telah ditambahkan ke staging area dengan perintah **git commit -m "Initial commit: menambahkan file Latihan01_01.html"**.

Menghubungkan ke GitHub dan Push Perubahan

1. Buat repository baru di GitHub, misalnya dengan nama **pweb1**.
2. Kembali ke terminal, hubungkan repository lokal anda ke remote repository di GitHub dengan perintah:
git remote add origin https://github.com/SDNKalimanggis/pweb1.git
3. Ketikkan perintah **git branch -M main**
4. Akhiri dengan mengetikkan perintah **git push -u origin main**. Perintah ini akan mengirim semua commit di branch utama ke repository remote di GitHub. Sekarang, proyek anda sudah tersimpan di cloud dan dapat diakses dari mana saja
5. Amati hasilnya.

F. POST-TEST

1. Apa fungsi dari git merge dalam workflow Git?
2. Jelaskan cara mengatasi konflik saat melakukan merge!
3. Tulis langkah-langkah untuk menambahkan file baru ke repository Git dan melakukan commit.

G. LATIHAN/TUGAS

1. Buatlah repository Git untuk proyek web dan dokumentasikan proses pembuatan, commit, dan push ke GitHub.
2. Rancanglah workflow Git yang melibatkan branch untuk fitur baru dan integrasi dengan branch utama.
3. Simpan seluruh praktikum pada modul sebelumnya ke dalam repository Git yang sudah anda buat