ACTIVITY PERTEMUAN 7

NAMA **: Simon Praja H P Pasaribu**

NPM **: 51421422**

KELAS **: 4IA14**

MATERI **: Dokumentasi Pengembangan Perangkat Lunak Terpadu, Version Control System, Git dan Github**

MATA PRAKTIKUM **: Rekayasa Perangkat Lunak 2**

(Screenshoot langkah-langkah sesuai video pembelajaran dan jelaskan dengan ringkas)

1. Jelaskan apa itu Git dan bagaimana git dapat membantu dalam kerja sama tim

   Jawab:

   Git adalah sistem kontrol versi terdistribusi yang dirancang untuk melacak perubahan dalam file dan memfasilitasi kerja sama dalam pengembangan perangkat lunak. Git memungkinkan banyak pengembang untuk bekerja bersama secara efisien pada satu proyek dengan meminimalkan konflik dan memastikan bahwa setiap orang memiliki versi terbaru dari proyek tersebut.

Bagaimana Git Membantu Kerja Sama Tim

1. Pengelolaan Versi (Version Control)
   Git menyimpan riwayat perubahan dari setiap file dalam sebuah proyek, sehingga memudahkan tim untuk melacak siapa yang melakukan perubahan apa dan kapan. Jika ada kesalahan, tim dapat mengembalikan proyek ke versi sebelumnya.

2. Kolaborasi Paralel
   Git memungkinkan anggota tim bekerja pada fitur atau perbaikan bug secara independen di cabang mereka sendiri tanpa mengganggu pekerjaan orang lain. Setelah selesai, perubahan tersebut dapat digabungkan ke cabang utama (misalnya main atau master).

3. Deteksi Konflik
   Jika dua orang mengedit bagian yang sama dari sebuah file, Git mendeteksi konflik tersebut saat penggabungan (merge) dan meminta pengguna untuk menyelesaikannya sebelum melanjutkan.

4. Replikasi Lokal
   Setiap pengembang memiliki salinan lengkap dari repositori di komputer mereka. Ini berarti mereka dapat bekerja secara offline dan hanya perlu menyinkronkan perubahan saat mereka online.

5. Kolaborasi melalui Platform
   Dengan integrasi ke layanan seperti GitHub, GitLab, atau Bitbucket, tim dapat mengelola repositori mereka di cloud. Fitur tambahan seperti *pull requests* dan *issue tracking* mendukung kerja sama yang lebih terorganisir.

6. Pembuatan Cabang (Branching)
   Git memungkinkan pengembang membuat cabang (*branch*) untuk mengembangkan fitur baru atau menguji ide tanpa memengaruhi kode utama. Setelah selesai, cabang dapat digabungkan kembali.

7. Review Kode (Code Review)
   Dengan fitur seperti *pull request* di GitHub atau *merge request* di GitLab, anggota tim dapat meninjau dan memberikan masukan sebelum kode digabungkan ke cabang utama.

Contoh Alur Kerja Tim dengan Git

1. Clone Repositori: Setiap anggota tim menggandakan repositori utama ke komputer lokal mereka menggunakan git clone.
2. Membuat Cabang: Masing-masing anggota membuat cabang baru untuk fitur yang mereka kerjakan, misalnya, git checkout -b fitur-baru.
3. Mengembangkan Fitur: Setiap anggota bekerja pada cabang mereka sendiri.
4. Commit Perubahan: Setelah perubahan selesai, mereka menyimpan progres dengan git commit.
5. Push ke Repositori Remote: Perubahan diunggah ke repositori remote dengan git push.
6. Merge ke Cabang Utama: Setelah perubahan selesai dan direview, cabang digabungkan ke main.

Git adalah alat yang sangat penting untuk memastikan bahwa proyek tim tetap terorganisir, efisien, dan dapat dikelola dengan baik, bahkan ketika tim bekerja dari lokasi yang berbeda.

2. lakukan operasi-operasi yang dapat dilakukan pada github seperti membuat repository, clone repository, git push dan git pull

Jawaban :

Berikut adalah langkah-langkah untuk melakukan operasi-operasi dasar pada GitHub, mulai dari membuat repository hingga melakukan push dan pull:

---

1. Membuat Repository di GitHub

1. Masuk ke GitHub:
   - Buka GitHub dan login ke akun Anda.
2. Buat Repository Baru:
   - Klik tombol New atau pilih Repositories > New Repository.
   - Masukkan nama repository (misalnya my-project).
   - Pilih visibilitas: Public atau Private.
   - Pilih opsi Initialize this repository with a README (opsional).
   - Klik Create repository.

---

2. Clone Repository

Setelah repository dibuat, Anda dapat mengkloningnya ke komputer lokal Anda:

1. Salin URL Repository:
    - Pada halaman repository GitHub, klik tombol Code.
    - Salin URL HTTPS/SSH (misalnya: https://github.com/username/my-project.git).
2. Clone Repository ke Lokal:
    - Buka terminal/command prompt.
    - Jalankan perintah berikut:

      bash
      Copy code
      git clone https://github.com/username/my-project.git

    - Repository akan diunduh ke komputer Anda di folder dengan nama repository tersebut.

---

3. Menambah File dan Commit

1. Masuk ke Direktori Repository:

   bash
   Copy code
   cd my-project

2. Tambahkan File Baru:
    - Buat file atau folder baru dalam direktori ini (misalnya index.html).
3. Tambah File ke Staging Area:

   bash
   Copy code
   git add .

   Perintah ini akan menambahkan semua file yang diubah ke staging area.

4. Commit Perubahan:

   bash
   Copy code
   git commit -m "Initial commit"

---

4. Push ke GitHub

Jika repository lokal Anda ingin disinkronkan dengan yang di GitHub:

1. Hubungkan Repository Lokal ke Remote (jika belum):

   bash
   Copy code
   git remote add origin https://github.com/username/my-project.git

2. Push Perubahan ke Remote:

bash
Copy code
git push -u origin main

- -u origin main digunakan untuk menetapkan cabang utama sebagai default.
- Gunakan git push untuk *push* berikutnya tanpa argumen tambahan.

---

5. Pull dari GitHub

Jika ada perubahan baru pada repository remote yang ingin Anda unduh:

1. Perbarui Repository Lokal:

   bash
   Copy code
   git pull origin main

   Perintah ini akan mengambil perubahan terbaru dari cabang main pada remote repository.

---

6. Perubahan Berikutnya

1. Lakukan Perubahan:
   - Ubah atau tambahkan file dalam folder proyek lokal.
2. Git Workflow:

   bash
   Copy code
   git add .
   git commit -m "Deskripsi perubahan"
   git push

   Langkah ini akan memastikan semua perubahan Anda diperbarui di GitHub.

Ringkasan Perintah

Operasi	Perintah
Clone	git clone <URL>
Tambah File	git add .
Commit	git commit -m "Pesan commit"
Push	git push
Pull	git pull origin main
Remote Add	git remote add origin <URL>