

PIC : DevOps

1. Muhammad Fakhruzaky Maulyandri

D1041211030

### **SOP/Tutorial: Menggunakan GitHub di Windows (Push dan Commit)**

1. Install Git: Pastikan Git sudah terinstal di sistem Windows.
  - Unduh Git di sini <https://git-scm.com/downloads/win>
  - Ikuti petunjuk instalasi dan pastikan memilih opsi Git Bash saat diminta.

### **Menyiapkan Repository GitHub**

1. Buka GitHub:
  - Login ke akun GitHub kamu di [GitHub.com](https://github.com).
2. Buat Repository Baru:
  - Klik tombol "New Repository".
  - Isi nama repository (misalnya: my-project).
  - Pilih opsi Public atau Private sesuai kebutuhan.
  - Klik Create Repository.
3. Salin URL Repository:
  - Di halaman repository yang baru saja dibuat, salin URL repository di bagian atas (misalnya: <https://github.com/username/my-project.git>).

### **Menggunakan Git di Windows dengan Git Bash**

1. Buka Git Bash:
  - Setelah Git terinstal, buka Git Bash dari menu Start.
2. Clone Repository ke Mesin Lokal:
  - Di Git Bash, jalankan perintah berikut untuk meng-clone repository yang telah kamu buat:  
git clone <https://github.com/username/my-project.git>

- Gantilah `https://github.com/username/my-project.git` dengan URL repository yang sudah kamu salin sebelumnya.
2. Masuk ke Direktori Proyek:
    - Setelah proses clone selesai, masuk ke direktori proyek dengan perintah:  
`cd my-project`

## Menambah dan Mengubah File

1. Buat *branch* baru:
  - Buat branch baru dengan perintah:  
`git checkout -b <nama branch>`
  - Dengan melakukan perubahan di branch baru, kita bisa mengerjakan fitur atau perbaikan tanpa mengganggu pekerjaan orang lain. Setelah selesai, baru kita gabungkan (merge) perubahan tersebut ke utama.
2. Lakukan Perubahan:
  - Di branch yang baru, lakukan perubahan atau penambahan
2. Menambahkan Perubahan ke Staging Area:
  - Setelah selesai melakukan perubahan atau penambahan file, sekarang adalah saatnya menambahkan file ke *staging area* dengan perintah:  
`git add <path ke file>`
  - Perintah ini akan menambahkan file ke *staging area*. Jika ingin menambahkan semua file, ganti perintah `git add <path ke file>` dengan `git add .`

## Commit Perubahan

1. Melakukan Commit:
  - Setelah file ditambahkan ke staging area, lakukan commit dengan perintah:  
`git commit -m "deskripsi"`
  - Gantilah "deskripsi" dengan pesan yang menggambarkan perubahan yang dilakukan, misalnya: "Menambah fitur login".

## Push Perubahan ke GitHub

### 1. Push ke Repository GitHub:

- Setelah commit selesai, push perubahan ke repository GitHub dengan perintah:  
git push origin <nama branch>
- Sesuaikan saja nama branchnya dengan nama branch yang tadi dibuat.

### 2. Masukkan Kredensial GitHub:

- Masukkan username dan password akun GitHub jika diminta

## Melakukan *Pull Request*

### 1. Cek Repository di GitHub:

- Kunjungi repository di GitHub dan pastikan perubahan yang kamu buat sudah muncul di halaman repository

### 2. Buat Pull Request Baru:

- Buka tab “Pull Request” dan klik tombol “New Pull Request”
- Pilih main sebagai *base branch*
- Pilih branch yang barusan dipush sebagai *compare branch*
- Klik “Create pull request”

### 3. Isi detail yang diperlukan:

- Berikan judul *pull request*
- Berikan deskripsi *pull request*. Deskripsi bersifat opsional, namun disarankan untuk memberikan deskripsi agar memperjelas tujuan dari *pull request*
- (Opsional) tambahkan juga *Reviewers*, *Assignees*
  - i. Reviewers biasanya itu QA
  - ii. Assignees itu yang mengerjakan fiturnya
- Klik “Create pull request”

### 4. Merge pull request

- Setelah membuat pull request, scroll kebawah dan klik tombol “Merge pull request”

### 5. Selesai

- Semua perubahan yang tadi dilakukan sudah masuk ke branch main

## **Tips dan Trik**

1. Siapkan Branch Protection Rule untuk branch main:
  - a. Branch Protection Rule digunakan untuk mengamankan branch supaya tidak kejadian “tidak sengaja”
  - b. Rule yang biasanya diterapkan antara lain:
    - Restrict Deletions
    - Require a pull request before merging
    - Block force pushes
2. Tambahkan Anggota Lain Sebagai Collaborator:
  - a. Agar anggota lain, terutama QA dan DevOps juga bisa melihat kode dari aplikasi yang dibuat.
  - b. Demi kenyamanan bersama