

	SEKOLAH TINGGI TEKNOLOGI BONTANG		
	PROGRAM STUDI TEKNIK INFORMATIKA		
	Praktikum	:	Pemrograman Web
	Modul 7	:	Git & GitHub
	Tgl Update	:	07-07-2025
			Revisi 01

A. Tujuan Praktikum

1. Memahami konsep dasar dan perbedaan antara Git sebagai *Version Control System* dan GitHub sebagai platform *hosting*.
2. Melakukan instalasi Git dan konfigurasi awal yang diperlukan pada sistem operasi Windows dan macOS.
3. Mengintegrasikan Git dengan Visual Studio Code (VS Code) dan menggunakan terminal terintegrasinya.
4. Menguasai alur kerja dasar Git: init, add, commit.
5. Memahami dan menerapkan penggunaan *branch* (cabang) untuk mengisolasi pekerjaan.
6. Menghubungkan repositori lokal ke repositori *remote* di GitHub.
7. Melakukan push (mendorong) perubahan dari repositori lokal ke GitHub.

B. Teori Dasar

1. Git vs. GitHub: Apa Bedanya?

Seringkali pemula bingung antara Git dan GitHub. Penting untuk memahami bahwa keduanya adalah dua hal yang berbeda namun saling melengkapi.

- **Git:** Anggap Git sebagai sebuah "mesin waktu" untuk kode Anda. Git adalah *Distributed Version Control System* (DVCS), sebuah perangkat lunak yang berjalan di komputer lokal Anda. Fungsinya adalah untuk melacak setiap perubahan yang terjadi pada file-file proyek Anda. Anda dapat "menyimpan" sebuah versi (disebut *commit*), dan jika terjadi kesalahan, Anda bisa kembali ke versi sebelumnya dengan mudah. Git tidak memerlukan koneksi internet untuk bekerja.
- **GitHub:** Anggap GitHub sebagai sebuah "garasi online" sekaligus "arena kolaborasi" untuk proyek-proyek Git Anda. GitHub adalah platform berbasis web yang menyediakan layanan *hosting* untuk repositori Git. Dengan GitHub, Anda bisa menyimpan salinan proyek Anda secara online (sebagai *backup*), berbagi kode dengan orang lain, dan bekerja sama dalam sebuah tim pada proyek yang sama.

Secara analogi: Jika proyek Anda adalah mobil, Git adalah mesin dan sistem mekanik yang membuatnya berjalan dan bisa diperbaiki. GitHub adalah garasi tempat Anda memarkir mobil, memamerkannya, dan tempat teman-teman Anda bisa ikut membantu memperbaikinya.

 STITEK SEKOLAH TINGGI TEKNOLOGI BONTANG	SEKOLAH TINGGI TEKNOLOGI BONTANG			
	PROGRAM STUDI TEKNIK INFORMATIKA			
	Praktikum	:	Pemrograman Web	
	Modul 7	:	Git & GitHub	
	Tgl Update	:	07-07-2025	Revisi 01

2. Alur Kerja Dasar Git

Setiap proyek Git di komputer Anda memiliki tiga "area" utama:

- **Working Directory:** Folder proyek tempat Anda membuat dan mengedit file secara langsung.
- **Staging Area (Index):** Sebuah "ruang tunggu" atau area persiapan. Di sini Anda mendaftarkan perubahan-perubahan dari *Working Directory* yang siap untuk "disimpan" secara permanen dalam riwayat proyek.
- **Local Repository (.git directory):** "Buku catatan sejarah" proyek Anda. Ketika Anda melakukan *commit*, perubahan dari *Staging Area* akan disimpan secara permanen di sini. Semua riwayat *commit* tersimpan di dalam folder tersembunyi *.git*.

Alur kerjanya adalah sebagai berikut:

Working Directory → git add → Staging Area → git commit → Local Repository

3. Konsep Branching (Percabangan)

Branch adalah salah satu fitur terkuat di Git. Bayangkan riwayat proyek Anda adalah sebuah batang pohon utama (biasanya disebut *main* atau *master*). Ketika Anda ingin mengerjakan fitur baru atau mencoba sesuatu yang eksperimental, Anda bisa membuat branch (cabang) baru.

Anda bisa bekerja di cabang baru ini tanpa mengganggu kode yang stabil di cabang *main*. Setelah pekerjaan Anda di cabang baru selesai dan diuji, Anda bisa menggabungkannya (*merge*) kembali ke cabang *main*. Ini adalah praktik standar dalam pengembangan perangkat lunak untuk menghindari konflik dan menjaga stabilitas kode utama.

C. Pra-Syarat: Work-Environment Setup

Sebelum memulai praktikum, pastikan Anda telah menyiapkan semua yang diperlukan.

1. Membuat Akun GitHub

Jika Anda belum memiliki akun, daftar terlebih dahulu.

- **Langkah 1:** Buka browser dan kunjungi <https://github.com/>. Klik tombol **Sign up**.
- **Langkah 2:** Ikuti semua proses pendaftaran: masukkan email, buat *password*, dan pilih *username*. Verifikasi akun Anda melalui email yang dikirimkan oleh GitHub.

	SEKOLAH TINGGI TEKNOLOGI BONTANG		
	PROGRAM STUDI TEKNIK INFORMATIKA		
	Praktikum	:	Pemrograman Web
	Modul 7	:	Git & GitHub
	Tgl Update	:	07-07-2025
			Revisi 01

2. Instalasi Git

- **Untuk Pengguna Windows:**

- **Langkah 1:** Unduh installer Git dari <https://git-scm.com/download/win>.
- **Langkah 2:** Jalankan installer. Pada sebagian besar langkah, Anda bisa menggunakan pengaturan *default* (cukup klik "Next"). Pastikan pada bagian "Choosing the default editor used by Git", Anda memilih **Visual Studio Code** jika sudah terinstal.
- **Langkah 3:** Selesaikan instalasi.

- **Untuk Pengguna macOS:**

- **Langkah 1:** Buka aplikasi **Terminal**.
- **Langkah 2:** Ketik `git --version` dan tekan Enter. Jika Git belum terinstal, macOS akan secara otomatis menawarkan untuk menginstal *Xcode Command Line Tools*. Klik **Install** dan ikuti prosesnya.
- **Langkah 3 (Alternatif):** Anda juga bisa menginstal Git menggunakan [Homebrew](https://brew.sh/) (jika sudah terinstal) dengan perintah `brew install git`.

3. Instalasi dan Konfigurasi Visual Studio Code (VS Code)

- **Langkah 1: Instal VS Code**

- Unduh VS Code dari situs resminya: <https://code.visualstudio.com/>.
- Jalankan installer dan ikuti petunjuknya.

- **Langkah 2: Instal Ekstensi yang Direkomendasikan**

- Buka VS Code. Klik ikon Ekstensi di bilah samping kiri (terlihat seperti 4 kotak).
- Cari dan instal ekstensi berikut:
 - 1) **GitLens — Git supercharged:** Ekstensi ini sangat membantu untuk melihat riwayat file, siapa yang mengubah baris kode tertentu (blame), dan banyak lagi, langsung di dalam editor.
 - 2) **GitHub Pull Requests and Issues:** Memudahkan interaksi dengan GitHub langsung dari VS Code.

 STITEK SEKOLAH TINGGI TEKNOLOGI BONTANG	SEKOLAH TINGGI TEKNOLOGI BONTANG			
	PROGRAM STUDI TEKNIK INFORMATIKA			
	Praktikum	:	Pemrograman Web	
	Modul 7	:	Git & GitHub	
	Tgl Update	:	07-07-2025	Revisi 01

• Langkah 3: Konfigurasi Awal Git (Sangat Penting!)

- Buka terminal terintegrasi di VS Code dengan menekan Ctrl + `` (tombol backtick, di sebelah kiri angka 1) atau melalui menu View > Terminal`.
- Di terminal, Anda perlu mengatur nama dan email Anda. **Ini hanya perlu dilakukan sekali.** Git akan menggunakan informasi ini untuk menandai setiap *commit* yang Anda buat. Ganti Nama Anda dan emailanda@contoh.com dengan data Anda sendiri. Lakukan satu per satu.

```
1. git config --global user.name "Nama Anda"
2. git config --global user.email "emailanda@contoh.com"
```

D. Langkah-langkah Praktikum

Skenario: Anda telah menyelesaikan tugas dari **Modul 1 (HTML Dasar)** dan **Modul 2 (Pengenalan CSS)**. Sekarang kita akan menyimpan dan mengelola kedua proyek ini menggunakan Git dan GitHub.

Asumsi Struktur Folder:

Untuk latihan ini, mari kita asumsikan Anda memiliki struktur folder proyek seperti ini di suatu tempat di komputer Anda:

Praktikum-PemrogramanWeb/

```
├── Modul-1/
│   └── tugas1.htm
├── Modul-2/
│   ├── tugas_1_modul2.htm
│   └── style.css
```

1. Membuka Proyek dan Menginisialisasi Git

- **Langkah 1:** Buka VS Code, lalu klik File > Open Folder... dan pilih folder induk proyek Anda, yaitu Praktikum-PemrogramanWeb.
- **Langkah 2:** Buka terminal terintegrasi (Ctrl + ``). Pastikan Anda berada di direktori yang benar (Praktikum-PemrogramanWeb`).
- **Langkah 3:** Ketik perintah berikut di terminal untuk menginisialisasi Git di folder ini.

```
git init
```

- Perintah ini membuat sebuah subfolder tersembunyi `.git` yang akan melacak semua perubahan.

	SEKOLAH TINGGI TEKNOLOGI BONTANG			
	PROGRAM STUDI TEKNIK INFORMATIKA			
	Praktikum	:	Pemrograman Web	
	Modul 7	:	Git & GitHub	
	Tgl Update	:	07-07-2025	Revisi 01

2. Perintah Dasar Bash dan Alur Kerja Commit Pertama

• Langkah 1: Memeriksa Status (`git status`)

- Perintah ini adalah teman terbaik Anda. Gunakan untuk melihat status proyek Anda saat ini.

```
git status
```

- *Outputnya akan menunjukkan bahwa folder Modul-1/ dan Modul-2/ adalah untracked files (file yang belum dilacak).*

• Langkah 2: Menambahkan File ke *Staging Area* (`git add`)

- Kita akan menambahkan semua file dan folder dalam proyek ini ke *staging area*.

```
git add .
```

(Tanda titik `.` adalah wildcard yang berarti "semua yang ada di direktori ini").

- Jalankan `git status` lagi. Anda akan melihat file-file tersebut sekarang berada di bawah "Changes to be committed".

• Langkah 3: Melakukan *Commit* (`git commit`)

- Sekarang, simpan perubahan dari *staging area* ke riwayat repositori lokal Anda dengan pesan yang jelas.

```
git commit -m "Initial commit: Menambahkan proyek Modul 1 dan Modul 2"
```

- *Pesan commit (-m) harus singkat namun deskriptif.*
- *Ini akan menandakan perubahan yang telah berhasil disimpan dalam riwayat lokal.*

3. Membuat Repositori di GitHub dan Menghubungkannya

- **Langkah 1:** Buka GitHub di browser. Klik ikon + di pojok kanan atas, lalu pilih **New repository**.
- **Langkah 2:** Isi detailnya:
- **Langkah 3:** Klik **Create repository**.
- **Langkah 4:** GitHub akan menampilkan halaman dengan beberapa perintah. Salin URL HTTPS dari repositori Anda. - *URL ini adalah alamat dari repositori remote Anda.*
- **Langkah 5:** Kembali ke terminal VS Code dan hubungkan repositori lokal Anda ke *remote* di GitHub dengan perintah berikut (ganti URL dengan milik Anda):

```
git remote add origin https://github.com/usernameAnda/portofolio-pemweb.git
```

 STITEK SEKOLAH TINGGI TEKNOLOGI BONTANG	SEKOLAH TINGGI TEKNOLOGI BONTANG			
	PROGRAM STUDI TEKNIK INFORMATIKA			
	Praktikum	:	Pemrograman Web	
	Modul 7	:	Git & GitHub	
	Tgl Update	:	07-07-2025	Revisi 01

4. Praktik Terbaik: Bekerja dengan Branch

Sebelum kita mendorong kode kita, mari kita praktikkan alur kerja yang benar dengan membuat branch baru. Jangan bekerja langsung di branch main.

- **Langkah 1: Membuat *Branch* Baru**

- Buat *branch* baru dengan nama yang deskriptif. Misalnya, kita akan menamainya tugas/modul-1-dan-2.

```
git branch tugas/modul-1-dan-2
```

- **Langkah 2: Pindah ke *Branch* Baru**

- Sekarang, pindah dari main ke *branch* baru tersebut.

```
git checkout tugas/modul-1-dan-2
```

Anda juga bisa melakukan kedua langkah di atas dengan satu perintah:

```
git checkout -b tugas/modul-1-dan-2
```

- **Langkah 3: Mendorong (*Push*) *Branch* Baru ke GitHub**

- Sekarang, unggah *branch* baru ini beserta semua *commit*-nya ke GitHub.

```
git push -u origin tugas/modul-1-dan-2
```

- Opsi *-u* akan mengatur agar *branch* lokal Anda terhubung dengan *branch* remote dengan nama yang sama, sehingga lain kali Anda cukup mengetik *git push*.

5. Pentingnya Tidak Melakukan push Langsung ke main

Dalam proyek tim, mendorong kode langsung ke branch main sangat berbahaya. Ini bisa memasukkan bug atau kode yang belum selesai ke dalam basis kode utama yang digunakan semua orang. Alur kerja yang benar adalah:

1. Bekerja di *branch* terpisah (seperti yang kita lakukan).
2. Setelah selesai, buat *Pull Request* di GitHub.
3. *Pull Request* adalah permintaan untuk menggabungkan kode dari *branch* Anda ke *branch* main. Ini memberi kesempatan kepada anggota tim lain untuk meninjau (*review*) kode Anda sebelum digabungkan.

Untuk proyek pribadi seperti ini, Anda bisa saja langsung mendorong ke main, tetapi membiasakan diri dengan alur kerja *branching* adalah investasi yang sangat berharga.

	SEKOLAH TINGGI TEKNOLOGI BONTANG			
	PROGRAM STUDI TEKNIK INFORMATIKA			
	Praktikum	:	Pemrograman Web	
	Modul 7	:	Git & GitHub	
	Tgl Update	:	07-07-2025	Revisi 01

E. Tugas Praktikum

Tugas untuk modul ini adalah membangun sebuah fondasi untuk portofolio digital Anda. Anda akan membuat satu repositori pusat yang akan berisi semua hasil kerja praktikum Anda dari Modul 1 hingga Modul 7, dan seterusnya.

- Inisialisasi:** Buat sebuah repositori **publik** baru di GitHub dengan nama Portofolio-Pemrograman-Web-[NIM].
- Langkah Awal (Modul 1-3):**
 - *Clone* repositori tersebut ke komputer Anda.
 - Buat struktur folder untuk Modul 1, 2, dan 3.
 - Salin hasil kerja Anda ke folder-folder tersebut.
 - Buat *branch* tugas/modul-1-3, lalu add, commit, dan push pekerjaan Anda.
- Langkah Berkelanjutan (Modul 4-6):**
 - Setiap kali Anda menyelesaikan tugas dari Modul 4, 5, dan 6, ulangi proses berikut:
 - Pastikan Anda berada di *branch* main dan pull perubahan terbaru (git pull origin main).
 - Buat *branch* baru untuk tugas tersebut (misalnya, tugas/modul-4-javascript).
 - Tambahkan folder dan file tugas baru Anda.
 - Lakukan add, commit, dan push branch baru tersebut.
- Langkah Akhir (Modul 7):**
 - Buat sebuah file README.md di direktori utama repositori Anda.
 - Di dalam README.md, tuliskan daftar isi yang rapi, menjelaskan apa saja yang ada di setiap folder modul, dan berikan *link* ke setiap folder.
 - Lakukan commit dan push untuk file README.md ini di *branch* main.
- Pengumpulan:**
 - Sertakan URL ke repositori GitHub Anda di laporan praktikum.
 - Pastikan repositori diatur ke publik agar dapat diakses.
 - Lampirkan link video. Penjelasan mengikuti rubrik penilaian. (Rubrik penilaian ada di halaman selanjutnya)

	SEKOLAH TINGGI TEKNOLOGI BONTANG			
	PROGRAM STUDI TEKNIK INFORMATIKA			
	Praktikum	:	Pemrograman Web	
	Modul 7	:	Git & GitHub	
	Tgl Update	:	07-07-2025	Revisi 01

• **Rubrik Penilaian**

Aspek Penilaian	Keterangan	Bobot (%)
Struktur README	<p>Isi: Mencakup judul, deskripsi, dan daftar isi/tautan ke folder modul.</p> <p>Format: Penggunaan Markdown yang baik untuk keterbacaan.</p> <p>Tampilan: Kesan keseluruhan rapi dan profesional.</p>	50%
Git dan Github (Video)	<p>Pemahaman konseptual dan praktis yang ditunjukkan dalam video penjelasan.</p> <p>Poin Penilaian:</p> <ul style="list-style-type: none"> • Konsep: Mampu menjelaskan fungsi Git & GitHub dengan bahasa sendiri. • Praktik: Mendemonstrasikan alur kerja Git (add, commit, branch, push). • Penyampaian: Penjelasan jelas, lancar, dan menunjukkan pemahaman materi. 	50%