

Version Control

Raih Mimpi #TanpaBatas



Basic Git & Collaborating Using Git

- ☐ Terminal and IDE
- ☐ Installing, initializing and committing GIT
- ☐ Collaborating Using Git

Basic Git & Collaborating Using Git



Terminal and IDE



Installing, initializing and committing GIT

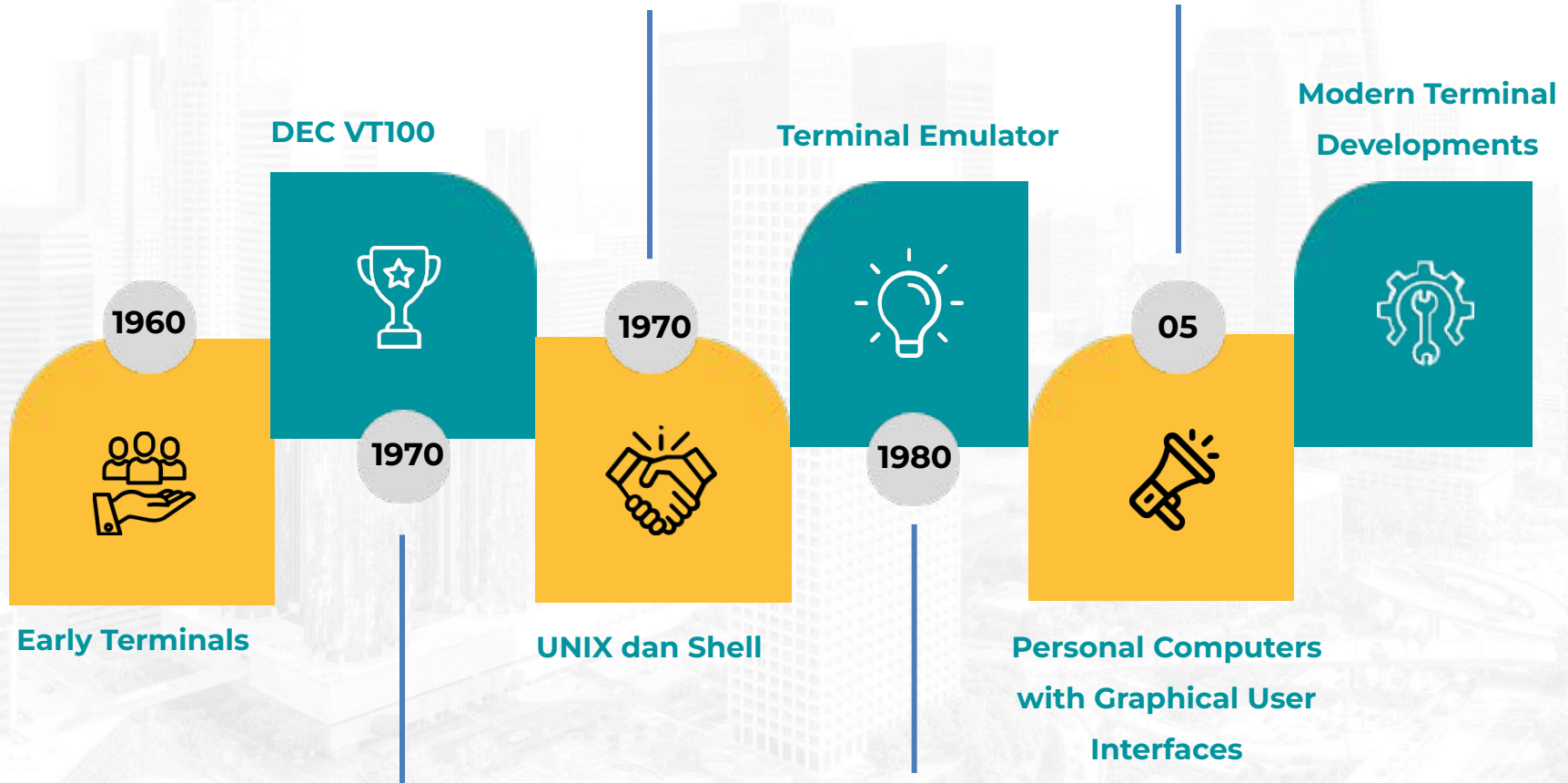


Collaborating Using Git

Objektif sesi

1. Memahami apa itu Terminal
2. Memahami dasar Terminal Command

Sejarah Singkat Terminal





Sejarah Singkat Terminal

Dengan perkembangan teknologi dan perangkat lunak, terminal tetap menjadi alat penting bagi para pengembang perangkat lunak, administrator sistem, dan pengguna teknis lainnya. Meskipun antarmuka grafis semakin canggih dan populer, terminal tetap memberikan fleksibilitas dan kekuatan untuk melakukan tugas-tugas khusus dan otomatisasi dalam lingkungan komputer modern.



Command Line Dasar

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



Command Line Dasar

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



Command Line Dasar

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

Basic Git & Collaborating Using Git

- ☒ **Terminal and IDE**
- ☐ Installing, initializing and committing GIT
- ☐ Collaborating Using Git

Basic Git & Collaborating Using Git



Terminal and IDE



Installing, initializing and committing GIT



Collaborating Using Git

Objektif sesi

1. Memahami pengertian dan penggunaan GIT
2. Mampu menginstall GIT
3. Memahami perintah-perintah GIT sederhana



Memahami Version Control 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-langkah untuk memahami kontrol versi dan Git

• Sistem Kontrol Versi Terpusat (Centralized Version Control System)

Dalam sistem kontrol versi terpusat, ada satu repositori sentral yang berfungsi sebagai "**master**" untuk menyimpan seluruh sejarah proyek. Setiap pengembang melakukan perubahan pada salinan lokal, kemudian mengirimkan perubahan tersebut ke repositori sentral. Contoh sistem kontrol versi terpusat adalah **Subversion** (SVN).

• Sistem Kontrol Versi Terdistribusi (Distributed Version Control System)

Dalam sistem kontrol versi terdistribusi, setiap anggota tim memiliki salinan lengkap dari seluruh repositori. Ini berarti setiap pengembang memiliki salinan lengkap sejarah perubahan, tidak hanya salinan terbaru. Contoh sistem kontrol versi terdistribusi adalah **Git**, **Mercurial**, dan **Bazaar**.

Apa itu GIT?



Pengantar

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.

Topic ini akan mempelajari menginstal Git pada berbagai sistem operasi, menginisialisasi repositori Git baru, dan melakukan commit pertama.





Installing GIT

Sumber: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Install Git pada Windows

- Panduan langkah demi langkah untuk menginstal Git di Windows.
- Download dan install GIT pada link berikut
<https://git-scm.com/download/win>
- Memverifikasi instalasi dan mengatur PATH Environment Variables

Note that this is a project called Git for Windows, which is separate from Git itself;



Installing GIT

Sumber: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Install Git pada Linux

- Menginstal Git melalui manajer paket seperti APT, YUM, atau DNF.
- Menginstal dari sumber pada distribusi Linux atau jalankan perintah ini pada terminal **`sudo apt install git-all`**
- Memverifikasi instalasi dan mengatur PATH environment variable.



Installing GIT

Sumber: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Install Git pada MacOS

- Instruksi rinci untuk menginstal Git di macOS.
- Memanfaatkan manajer paket macOS atau menginstal dari sumber atau dapat menginstall terlebih dahulu **Xcode Command Line Tools**
- Memverifikasi instalasi dan mengatur variabel lingkungan PATH..

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

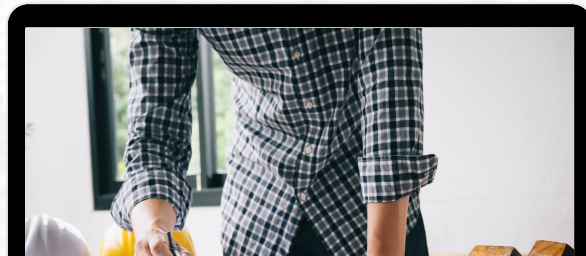
Menampilkan status perubahan yang belum
dikomit di repositori lokal.

```
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).

```
git push origin main
```



07 git pull

Mengambil commit terbaru dari repositori jarak jauh dan menggabungkannya ke 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 beserta 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
```



Continue...

14 git diff

Menampilkan perbedaan antara versi yang sudah di-staging dengan versi sebelumnya.

```
git diff
```

15 git reset

Mengembalikan file yang sudah di-staging ke direktori kerja sebelumnya.

```
git reset file.txt
```

Basic Git & Collaborating Using Git

- ☒ Terminal and IDE
- ☒ **Installing, initializing and committing GIT**
- ☐ Collaborating Using Git

Basic Git & Collaborating Using Git



Terminal and IDE



Installing, initializing and committing GIT



Collaborating Using Git

Objektif sesi

1. Memahami penggunaan command GIT untuk berkolaborasi

Simulasi Kolaborasi Menggunakan GIT

Simulasi kolaborasi Git dengan **tiga** orang menggunakan GitHub melibatkan **pengaturan dasar repository, pembuatan cabang, membuat perubahan, mengelola konflik**, dan menyelesaikannya dengan **permintaan tarik (pull request)** dan **penggabungan (merge)**.



Alice



Bob



Charlie

Step 1 : Menyiapkan Repository



Buat repositori GitHub baru dengan nama ***"git-collaboration-simulation."***



Undang Alice, Bob, dan Charlie sebagai **kolaborator** ke repositori tersebut.




Buat repository baru pada gthub <https://github.com/>

Step 2 : Persiapan Lokal



Setiap peserta **mengklon repositori ke mesin lokal** mereka menggunakan perintah berikut:

```
bash
```

 Copy code

```
git clone <repository_url>
```

Step 3 : Pembuatan Branch dan Alur Kerja



Setiap peserta harus membuat cabang untuk pekerjaannya.

Misalkan Alice bekerja di branch "alice-feature,"

Bob di branch "bob-feature," dan

Charlie di branch "charlie-feature."

Mereka dapat membuat branch dengan perintah berikut:

bash

 Copy code

```
git checkout -b alice-feature  
git checkout -b bob-feature  
git checkout -b charlie-feature
```


Step 4 : Melakukan Perubahan



Peserta bekerja di cabang masing-masing, membuat perubahan pada berkas proyek.



Setelah melakukan perubahan, mereka harus melakukan **commit** dengan **pesan commit yang deskriptif**:

```
bash
```

[Copy code](#)


```
git add .  
git commit -m "Deskripsi perubahan"
```

Step 5 : Mengirim Perubahan ke Remote



Peserta mengirimkan branch mereka ke repositori remote

```
bash
```

 Copy code

```
git push origin <nama-cabang>
```

Step 6.0 : Mengelola Konflik



Konflik dapat terjadi ketika **dua atau lebih peserta** membuat **perubahan pada berkas yang sama** di **branch yang berbeda**.

Mari simulasi konflik:

- Alice dan Bob sama-sama melakukan perubahan pada file "index.html."
- Charlie melakukan perubahan pada file "styles.css."



Instruksikan **Alice dan Bob** untuk **mencoba menggabungkan branch masing-masing**

bash

 Copy code

```
# Alice mencoba menggabungkan perubahan Bob ke cabangnya  
git checkout alice-feature  
git merge bob-feature
```

Step 6.1 : Mengelola Konflik




Jika terjadi konflik, mereka **harus menyelesaikannya secara manual**. **Buka file yang konflik** dalam editor teks, cari penanda konflik, dan **ubah kode untuk menyelesaikan konflik**.



Setelah menyelesaikan konflik, mereka harus melakukan **commit perubahan dan push branch** yang digabungkan:

bash

 Copy code

```
git add .  
git commit -m "Perubahan yang digabungkan dari Bob"  
git push origin alice-feature
```

Step 7 : Pull Request dan Merge



Setelah peserta selesai bekerja di cabang masing-masing, mereka harus membuat **permintaan tarik (pull request)** dari **branch mereka ke branch utama (biasanya "main" atau "master")** melalui GitHub.



Peserta lain akan **meninjau pull request dan memberikan umpan balik** jika diperlukan.



Jika semuanya terlihat baik, **pemilik repository (atau orang lain yang ditunjuk) dapat merge pull request ke branch utama** melalui GitHub.

Step 8.0 : Memperbarui Repositori Lokal



Setelah pull request digabungkan, semua peserta perlu **memperbarui repositori lokal** mereka **untuk mencerminkan perubahan yang dilakukan di branch utama:**

bash

 Copy code

```
git checkout main  
git pull origin main
```


Step 8.1 : Memperbarui Repositori Lokal



Setiap peserta harus beralih kembali ke branch masing-masing dan menggabungkan branch utama ke cabang fitur mereka untuk tetap terbaru:

```
bash
```

[Copy code](#)

```
git checkout <nama-cabang>  
git merge main
```

Step 9 : Kesimpulan Simulasi



Simulasi selesai ketika **semua peserta berhasil** mengirimkan perubahan mereka, **menyelesaikan konflik melalui pull request**, dan **menggabungkan branch ke branch utama**.

Proyek sekarang mencerminkan kolaborasi dari ketiga peserta.



Thank you