

## PEMROGRAMAN

### Version Control Menggunakan Git

**Kamarudin, M.Kom**

<http://coding4ever.net/>

<https://github.com/rudi-krsoftware/open-retail>

# Apa itu Version Control?



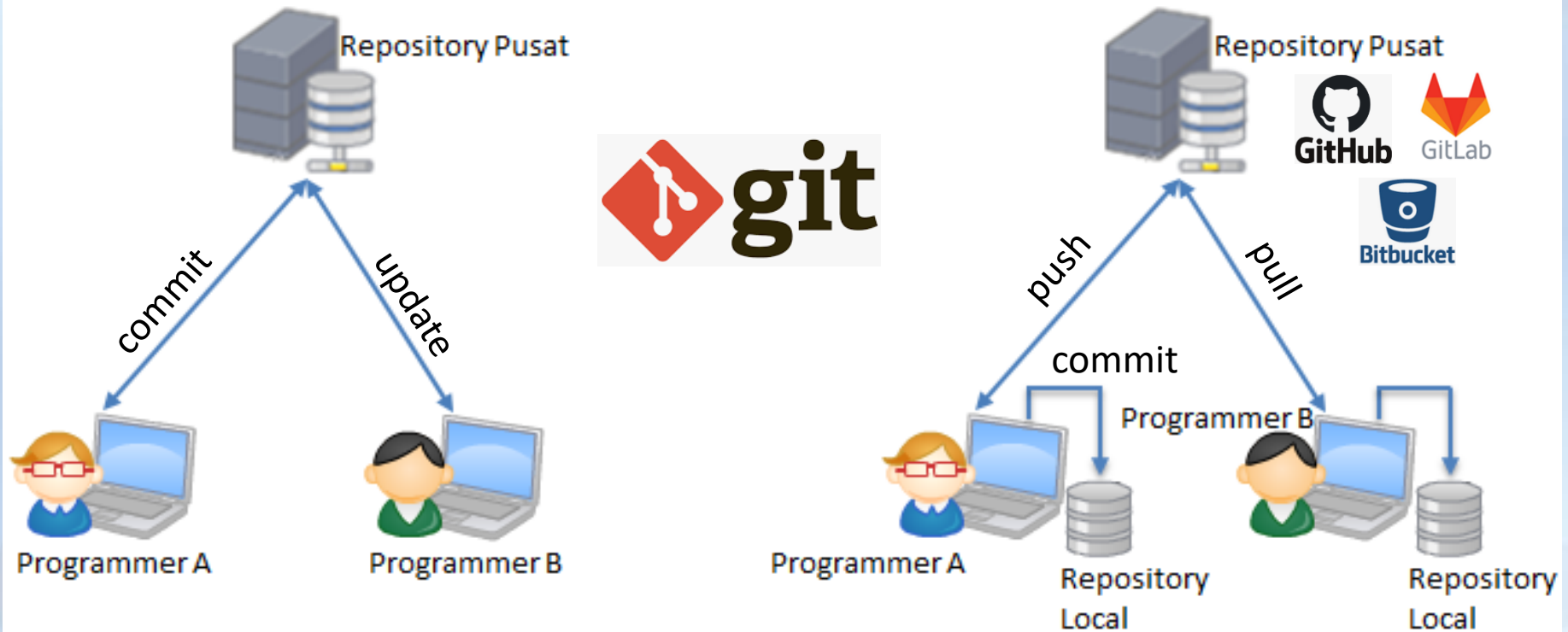
Version control adalah sebuah software yang dapat mencatat semua riwayat perubahan yang terjadi pada file atau folder.

# Manfaat Version Control

- ✓ Mencatat isi perubahan dan pelaku perubahan.
- ✓ Melihat riwayat perubahan kode, dari pertama dibuat sampai kondisi terakhir saat ini.
- ✓ Menyediakan fungsi undo, yaitu dapat mengembalikan kondisi kode program ke periode yang diinginkan.
- ✓ Memungkinkan pengembangan kode secara paralel (kolaborasi)
- ✓ Bisa juga difungsikan sebagai media backup untuk kode program.

# Jenis-jenis Version Control

## Centralized vs Distributed



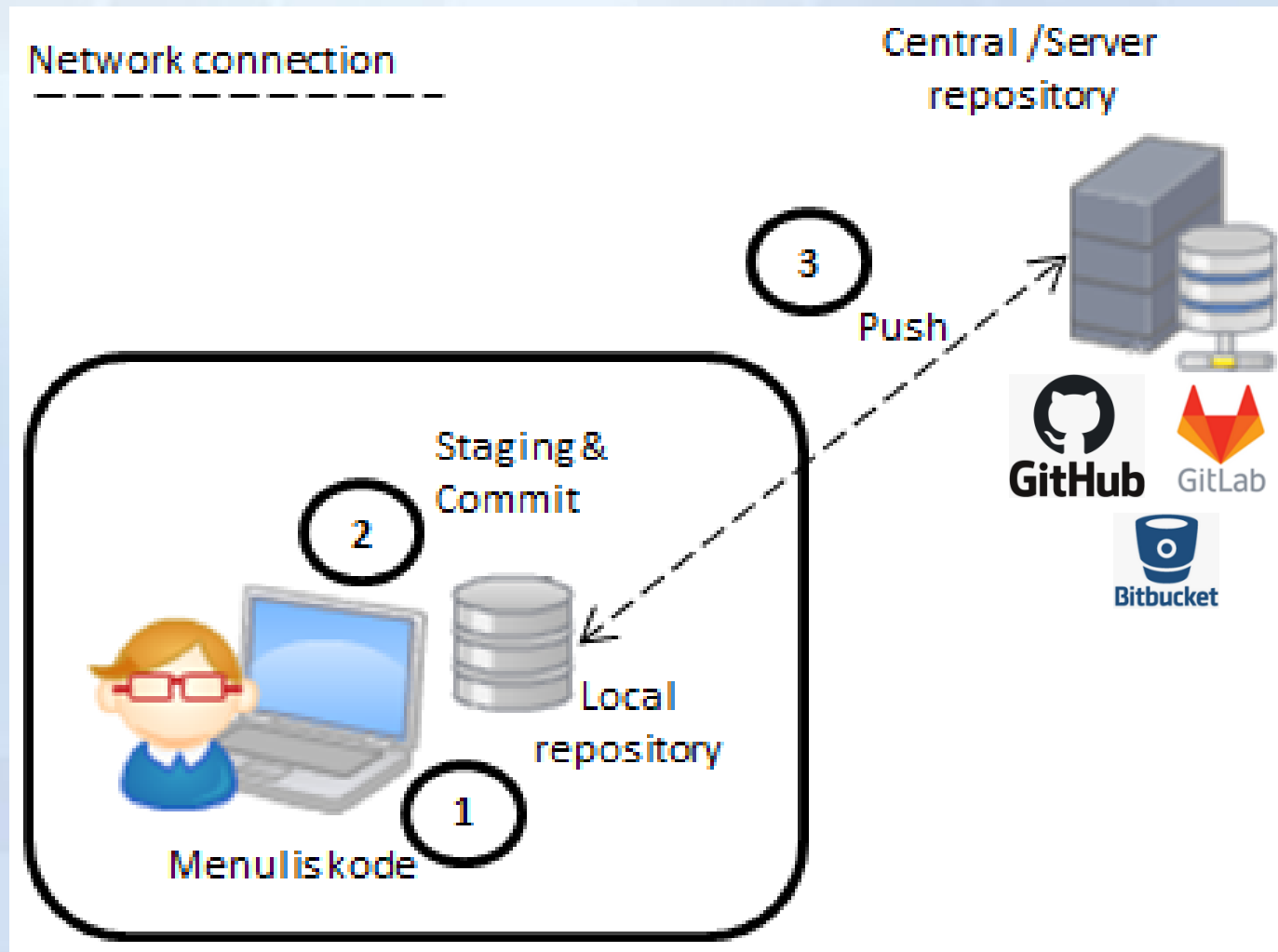
- ✓ Repository – Tempat penyimpanan yang menyimpan riwayat perubahan file atau folder

# Istilah-istilah Git

- Repository – tempat penyimpanan yang menyimpan riwayat perubahan file atau folder.
- Local Repository - repository yang ada di pc/laptop developer.
- Remote Repository - repository yang ada di server. Ex: github
- Staging area – Sebuah tempat di memory yang berisi informasi file atau folder dari working folder yang sudah siap untuk di commit.
- Commit - menyimpan perubahan file atau folder ke local repository.
- Push - mengupload perubahan dari local ke remote repository
- Clone - mengcopy remote repository.
- Pull - mengambil perubahan terbaru di remote repository, kemudian di merge ke repository local.
- Branch – Atau cabang, mirip dengan sebuah folder yang ada di dalam repository. Branch default adalah *master*.

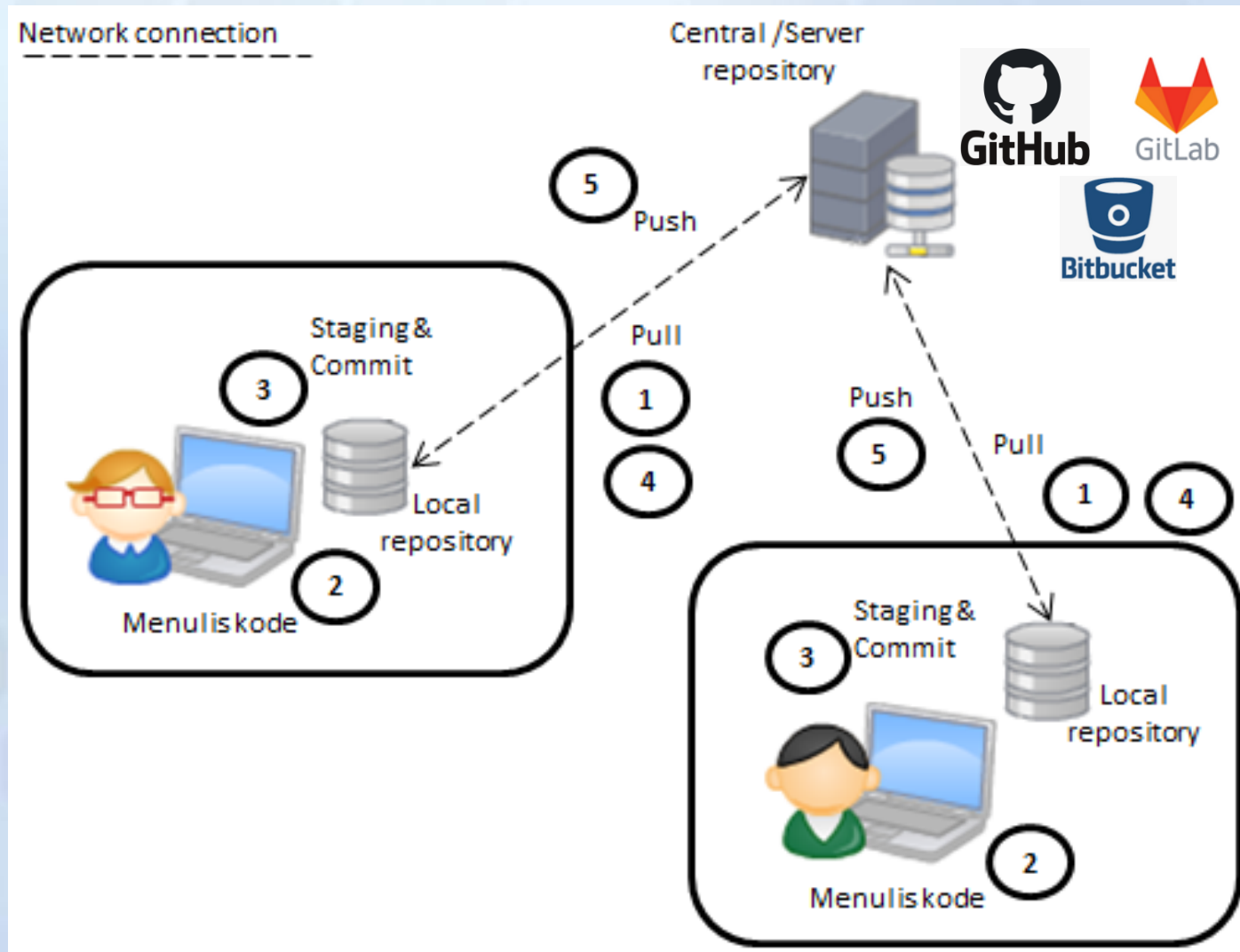
# Git Workflow

## Git Workflow jika bekerja sendiri



# Git Workflow (Lanjutan)

Git Workflow jika bekerja di dalam tim



# Contoh Kasus

```
class Calculator
{
    static void Main(string[] args)
    {
        Tugas 1: Membuat struktur
        kode aplikasi calculator
    }
}
```



# Contoh Kasus (Lanjutan)

```
class Calculator
{
    static void Main(string[] args)
    {
        int a = 10;
        int b = 6;

        Console.WriteLine("Hasil Penambahan: {0} + {1} = {2}", a, b, Penambahan(a, b));
        Console.WriteLine("Hasil Pengurangan: {0} - {1} = {2}", a, b, Pengurangan(a, b));

        Console.WriteLine("\nTekan sembarang key untuk keluar");
        Console.ReadKey();
    }

    static int Penambahan(int a, int b)
    {
        return a + b;
    }

    static int Pengurangan(int a, int b)
    {
        return a - b;
    }
}
```

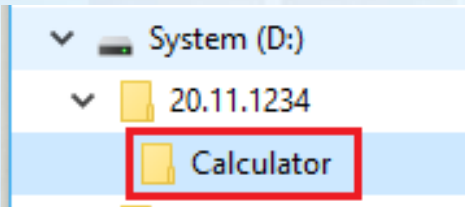
Tugas 4: Melengkapi fungsi Main

Tugas 2: Menambahkan fungsi Penambahan

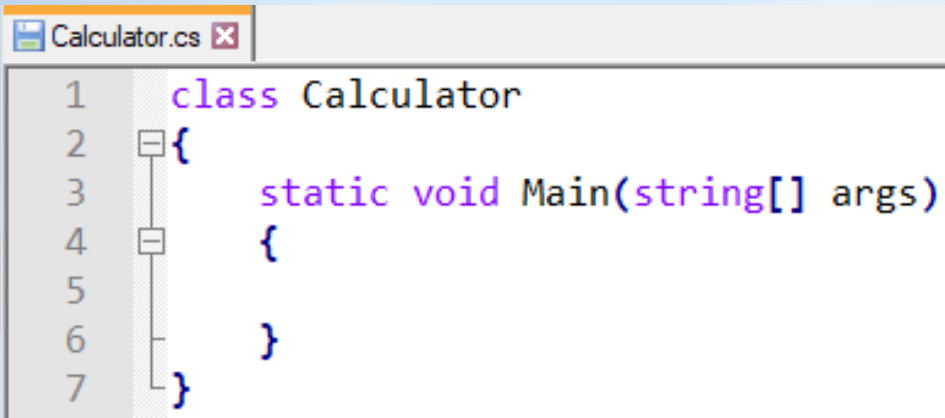
Tugas 3: Menambahkan fungsi Pengurangan

# Persiapan Latihan Git

- Mempersiapkan direktori kerja (working folder)



- Buat file baru dengan nama *Calculator.cs* di folder *Calculator* kemudian lengkapi kodenya seperti berikut:

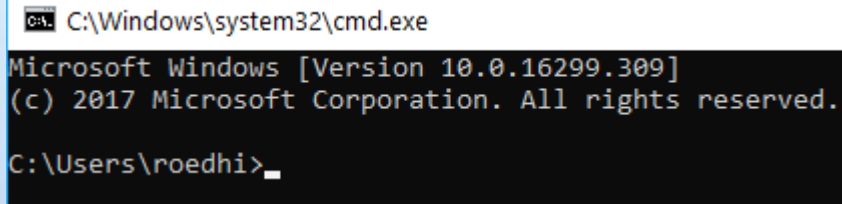
A screenshot of a code editor window titled 'Calculator.cs'. The code is as follows:

```
1 class Calculator
2 {
3     static void Main(string[] args)
4     {
5
6     }
7 }
```

Anda bisa menggunakan Notepad++ untuk membuat file tersebut.

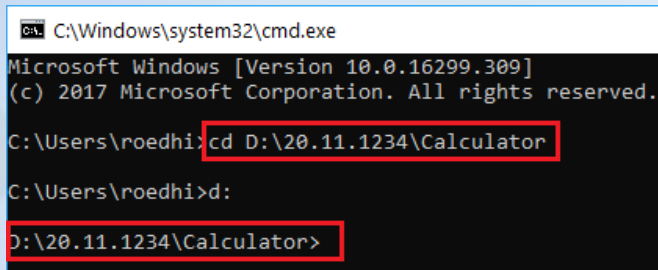
# Persiapan Latihan Git (Lanjutan)

- Mempersiapkan software git
  - ✓ Download di <https://git-scm.com/downloads>
  - ✓ Buka command prompt



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\roedhi>
```

- ✓ Kemudian pindah ke direktori kerja, dengan menggunakan perintah cd (change directory)



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\roedhi>cd D:\20.11.1234\Calculator
C:\Users\roedhi>d:
D:\20.11.1234\Calculator>
```


# Persiapan Latihan Git (Lanjutan)

## ■ Membuat repository public (github.com)

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner

 k4m4r82 ▾

/


Repository name \*


calculator-app ✓

Great repository names are short and memorable. Need inspiration? How about [laughing-lamp](#)?

Description (optional)

Contoh aplikasi calculator menggunakan bahasa C#


☒  **Public**  
Anyone can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository

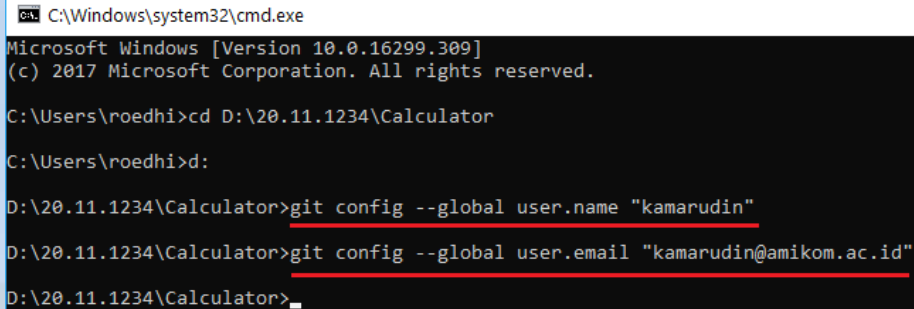
# Dasar-dasar Perintah Git

- Git Config

Konfigurasi awal sebelum menggunakan Git adalah mengeset informasi nama dan email.

```
git config --global user.name "kamarudin"  
git config --global user.email "kamarudin@amikom.ac.id"
```

## Contoh:



```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 10.0.16299.309]  
(c) 2017 Microsoft Corporation. All rights reserved.  
  
C:\Users\roedhi>cd D:\20.11.1234\Calculator  
  
C:\Users\roedhi>d:  
  
D:\20.11.1234\Calculator>git config --global user.name "kamarudin"  
D:\20.11.1234\Calculator>git config --global user.email "kamarudin@amikom.ac.id"  
D:\20.11.1234\Calculator>_
```

# Dasar-dasar Perintah Git (Lanjutan)

- Membuat repository local

Untuk mulai bekerja, kita harus memiliki repository dulu. Repository adalah database yang menyimpan riwayat perubahan file atau folder.

```
git init
```

## Contoh:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\roedhi>cd D:\20.11.1234\Calculator

C:\Users\roedhi>d:

D:\20.11.1234\Calculator>git config --global user.name "kamarudin"

D:\20.11.1234\Calculator>git config --global user.email "kamarudin@amikom.ac.id"

D:\20.11.1234\Calculator>git init
Initialized empty Git repository in D:/20.11.1234/Calculator/.git/

D:\20.11.1234\Calculator>_
```

# Dasar-dasar Perintah Git (Lanjutan)

- Melihat status working folder

Untuk melihat status perubahan file atau folder, mana yang masih di working folder dan mana yang sudah di staging.

```
git status
```

Contoh:

```
C:\Windows\system32\cmd.exe
D:\20.11.1234\Calculator>git init
Initialized empty Git repository in D:/20.11.1234/Calculator/.git/

D:\20.11.1234\Calculator>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        Calculator.cs

nothing added to commit but untracked files present (use "git add" to track)
D:\20.11.1234\Calculator>_
```

# Dasar-dasar Perintah Git (Lanjutan)

- Menambahkan perubahan ke staging area

```
git add .  
git add <nama file 1> <nama file 2> <nama file n>
```

- Menghapus perubahan dari staging area

```
git reset  
git reset <nama file>
```

- Menyimpan perubahan ke repository

```
git commit -m "diisi dengan keterangan commit"
```

Contoh:

```
git commit -m "inisialisasi project"
```

- Melihat history perubahan

```
git log  
git log -5 # log lima commit terakhir  
git log --oneline # log hanya menampilkan summary
```



# Dasar-dasar Perintah Git (Lanjutan)

- Mendaftarkan remote repository

```
git remote add <nama remote> <url remote repository>
```

Contoh:

```
git remote add calculator  
https://github.com/k4m4r82/calculator-app.git
```

*Perintah di atas, ditulis satu baris.*

- Mendapatkan informasi remote repository

```
git remote -v
```

- Mengupload perubahan di local repository ke remote.

```
git push <nama remote> <nama branch>
```

Contoh:

```
git push calculator master
```

*Kemudian cek repository github Anda.*

# Contoh Kasus (Lanjutan)

```
class Calculator
{
    static void Main(string[] args)
    {
        int a = 10;
        int b = 6;

        Console.WriteLine("Hasil Penambahan: {0} + {1} = {2}", a, b, Penambahan(a, b));
        Console.WriteLine("Hasil Pengurangan: {0} - {1} = {2}", a, b, Pengurangan(a, b));

        Console.WriteLine("\nTekan sembarang key untuk keluar");
        Console.ReadKey();
    }

    static int Penambahan(int a, int b)
    {
        return a + b;
    }

    static int Pengurangan(int a, int b)
    {
        return a - b;
    }
}
```

Tugas 4: Melengkapi fungsi Main

Tugas 2: Menambahkan fungsi Penambahan

Tugas 3: Menambahkan fungsi Pengurangan

# Dasar-dasar Perintah Git (Lanjutan)

- Mengcopy repository remote (clone)

```
git clone <url remote repository> [folder]
```

Contoh:

```
git clone https://github.com/k4m4r82/calculator.git  
git clone https://github.com/k4m4r82/calculator.git my-  
calculator
```

- Mengambil perubahan terbaru di repository remote

```
git pull <nama remote> <nama branch>
```

Contoh:

```
git pull calculator master
```

Selesai 😊