

## PEMROGRAMAN

### Version Control Menggunakan Git

**Kamarudin, M.Kom**

[kamarudin@amikom.ac.id](mailto:kamarudin@amikom.ac.id)

<http://coding4ever.net/>

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

# Apa itu Version Control?

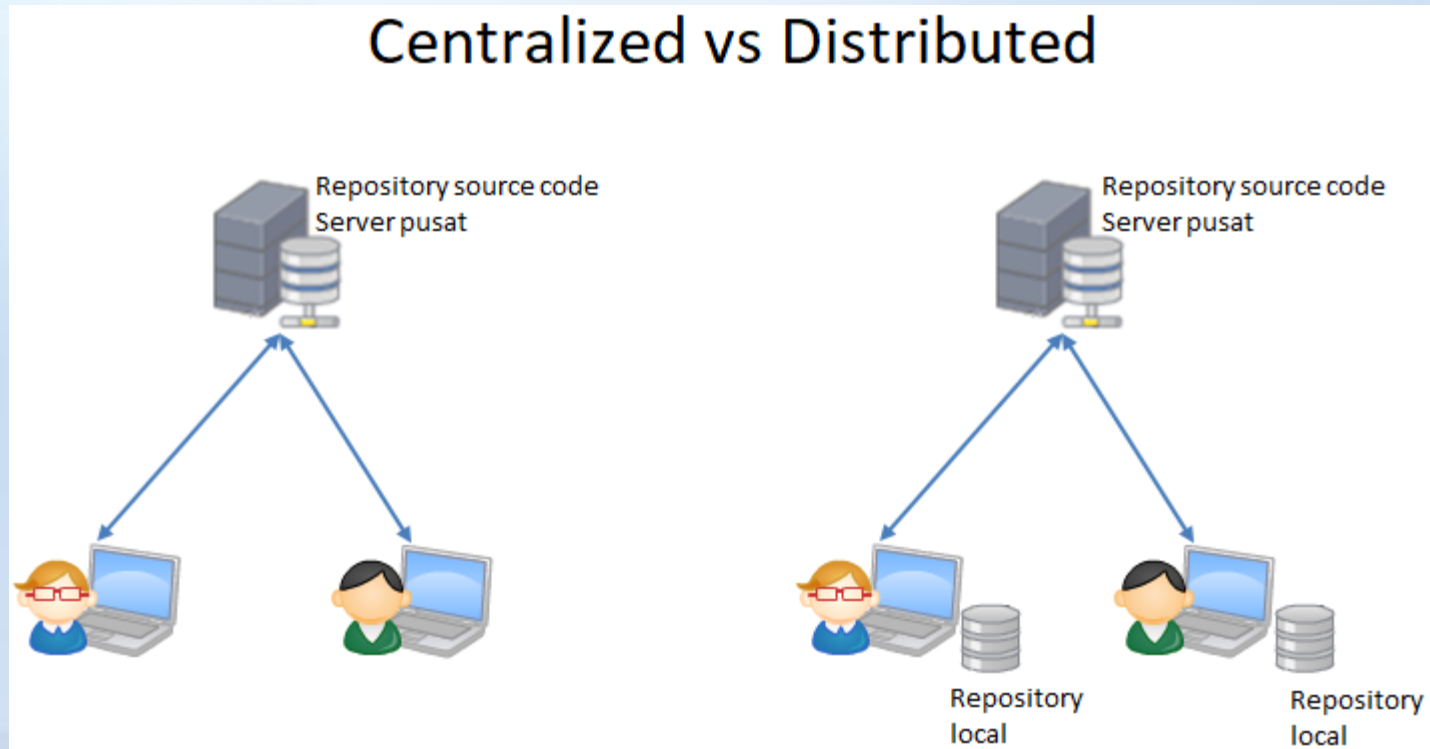
Version control adalah sebuah sistem yang dapat mencatat semua histori/riwayat perubahan yang terjadi pada file atau folder.

# 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.
- ✓ Bisa juga difungsikan sebagai media backup untuk kode program.

# Apa itu Git?

- ✓ Git adalah version control yang bersifat terdistribusi.

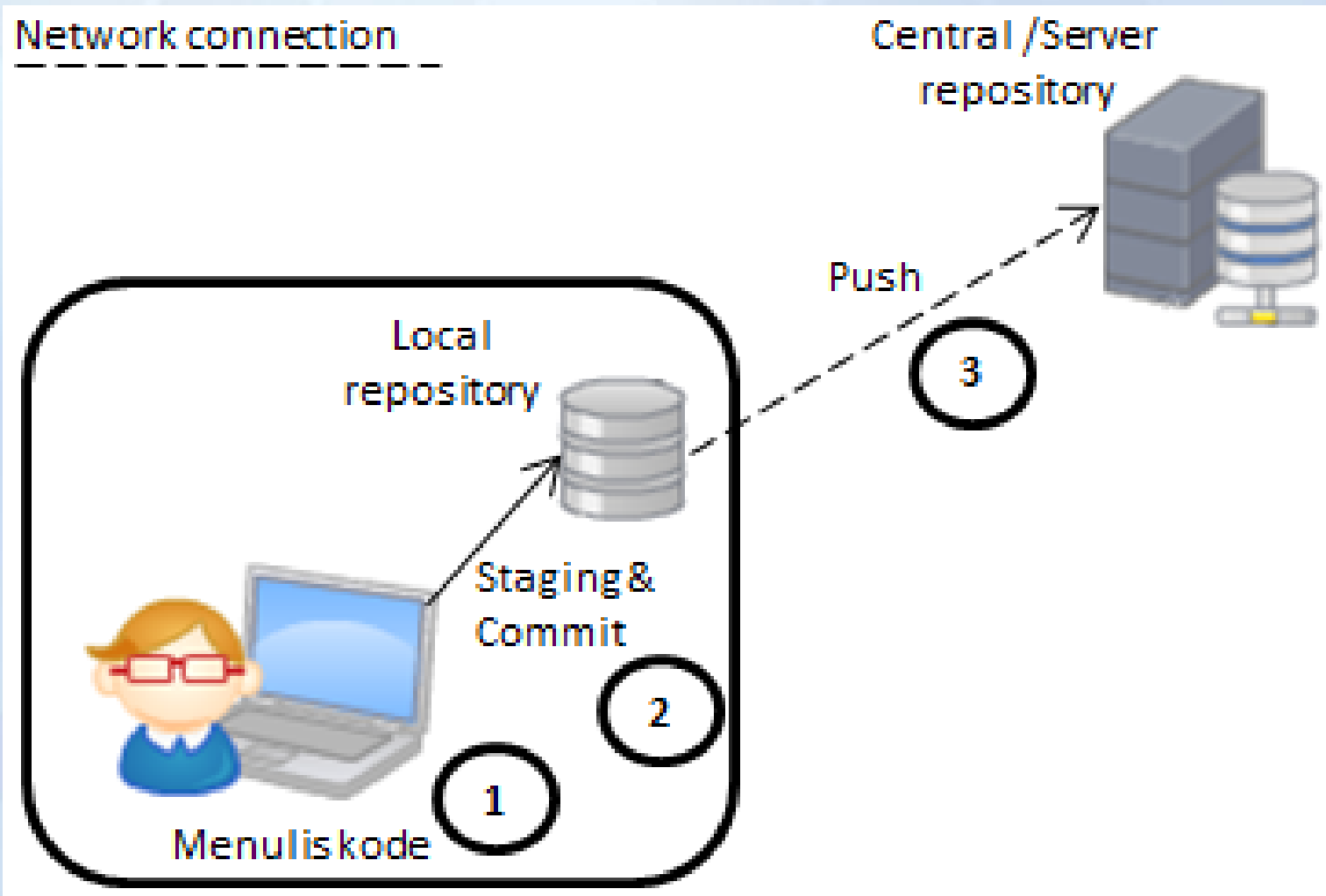


# Istilah-istilah Git

- Repository - database yang menyimpan history / riwayat perubahan file atau folder.
- Local - repository yang ada di pc/laptop developer.
- Remote - repository yang ada di server.
- 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 repository local.
- Master - nama branch default/utama yang diberikan git pada waktu membuat repository.
- Push - mengupload perubahan di local ke remote.
- Clone - mengcopy repository lain (remote).
- Pull - mengambil perubahan terbaru di remote, kemudian di merge ke local.

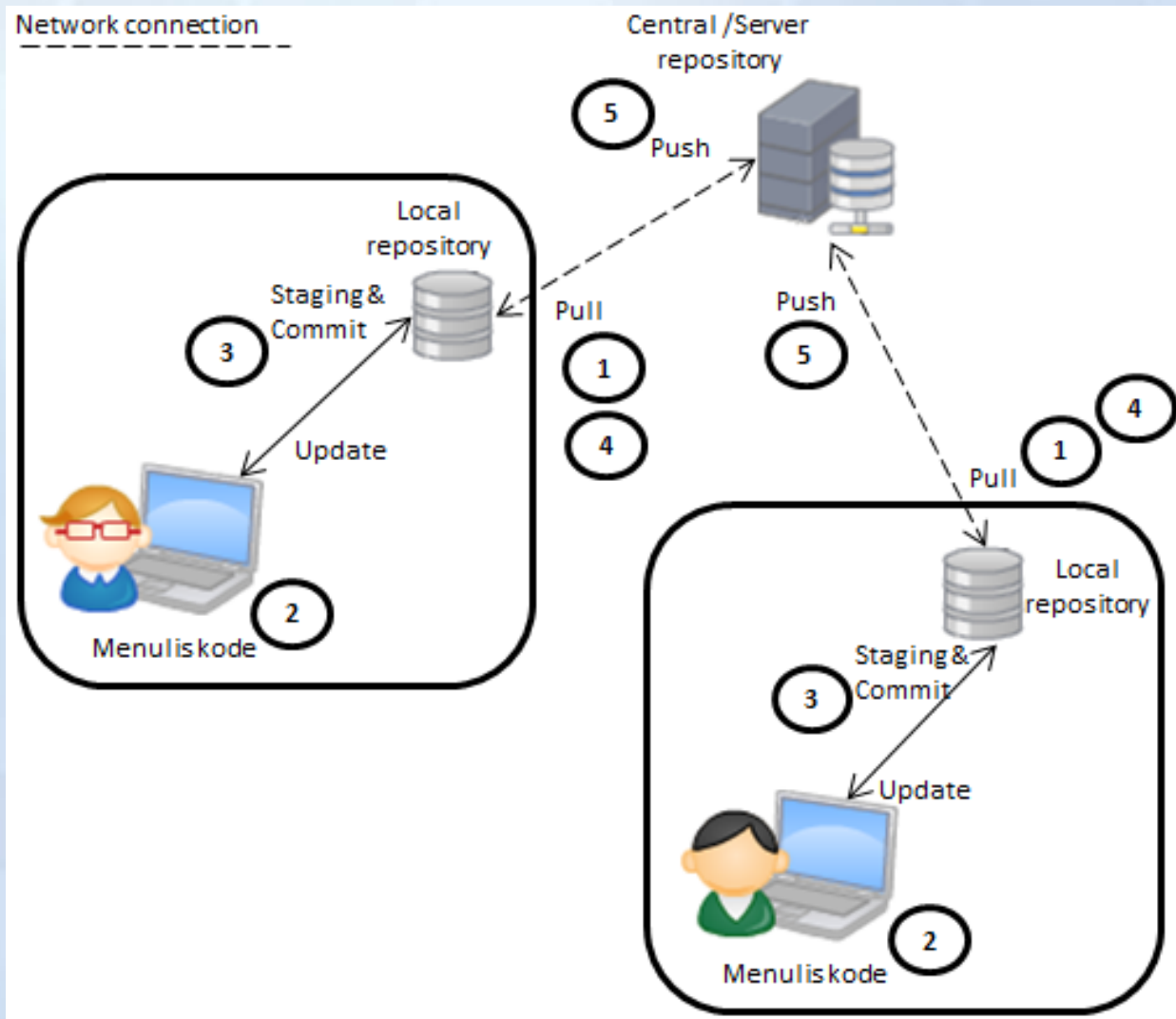
# Git Workflow

Git Workflow jika bekerja sendiri



# Git Workflow (Lanjutan)

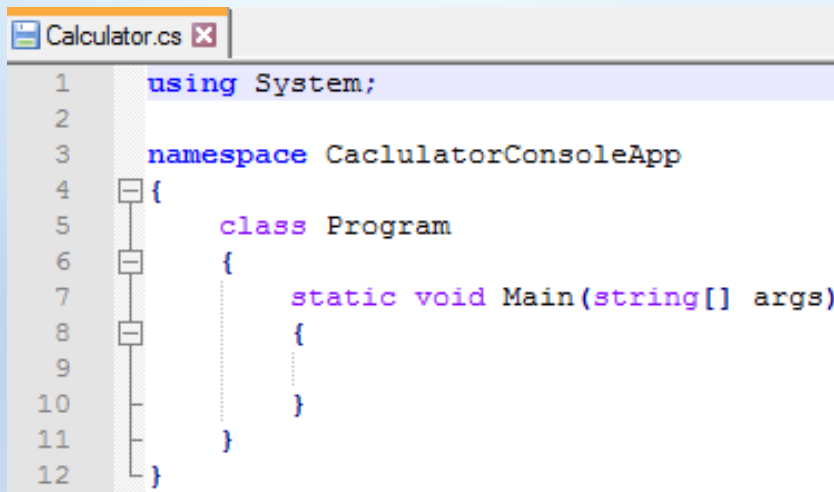
Git Workflow jika bekerja di dalam tim



# Persiapan Latihan Git

- Mempersiapkan direktori kerja (working folder)

Buat file baru dengan nama *Calculator.cs*, kemudian lengkapi kodenya seperti berikut:

A screenshot of a code editor window titled 'Calculator.cs'. The code is written in C# and includes a using statement for 'System', a namespace 'CaclulatorConsoleApp' (note the typo), and a class 'Program' with a 'Main' method. The code is as follows:

```
1 using System;
2
3 namespace CaclulatorConsoleApp
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             //
10        }
11    }
12 }
```

- Mempersiapkan software git
  - ✓ Ekstrak file PortableGit.exe
  - ✓ Jalankan file git-cmd.exe
- Membuat repository public (github.com)



# 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"
```

- Membuat repository baru

Untuk bisa mulai bekerja, kita harus memiliki repository dulu. Repository adalah database yang menyimpan history / riwayat perubahan file atau folder. Caranya aktifkan direktori kerja dengan menggunakan perintah *CD*. Kemudian ketikkan perintah Git berikut:

```
git init  
git init <project baru>
```

# 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
```

- Membuat ignore file

Dalam mengerjakan project, ada file-file yang ada di folder kerja, tapi tidak kita masukkan ke repository. Misalnya file hasil kompilasi, setting IDE, dan sebagainya. File dan folder hasil generate ini biasanya kita daftarkan di ignore list, supaya tidak ikut dicommit ke repository.

Untuk nama filenya adalah **.gitignore**

# 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 "keterangan commit"  
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>  
git remote add calculator https://github.com/belajar-  
ng0ding/calculator.git
```

- Mendapatkan informasi remote repository

```
git remote -v
```

- Mengupload repository local ke remote

```
# git push <nama remote> <nama branch>  
git push calculator master
```

# Dasar-dasar Perintah Git (Lanjutan)

- Mengcopy repository remote (clone)

```
# git clone <url remote repository> [folder]  
git clone https://github.com/belajar-ng0ding/calculator.git  
git clone https://github.com/belajar-ng0ding/calculator.git  
my-calculator
```

- Mengambil perubahan terbaru di repository remote

```
# git pull <nama remote> <nama branch>  
git pull calculator master
```

Selesai 😊