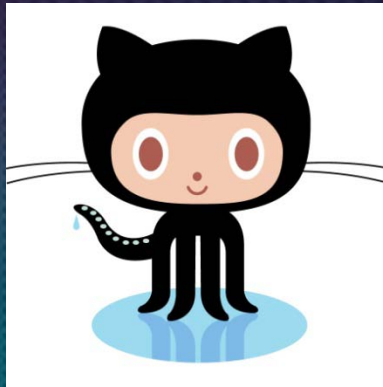


*Pengenalan kepada*

# *GitHub*

*Pengaturcaraan Sosial*



ANJURAN



PESIMA

# SILIBUS

## SESI 1

- Pengenalan Git
  - Sejarah
  - Konsep
- Pengenalan Github
- Istilah penting

## SESI 3

- Kolaborasi
  - Repo, Clone, fetch, pull, push

## SESI 2

- Pemasangan
  - Linux
  - Windows
- Perintah standart

## SESI 4

- Latihan Github
  - Akaun, fork, pull request

**@hariadi**

**diperakui@gmail.com**

**0x4d1**



# SESI 1

## PENGENALAN

# GIT

## Sejarah

- 2002 - Linux kernel menggunakan proprietary DVCS yang dipanggil BitKeeper
- 2005 – Konflik BitKeeper
- 2005 - Linus Torvalds membangun Git



# GIT

## Pengenalan



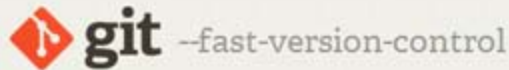
Git adalah perisian sumber terbuka dan merupakan DVCS yang direkabentuk untuk prestasi dan kecekapan.





## Pengenalan

Git adalah perisian **sumber terbuka** dan merupakan *distribute version control system* yang direkabentuk untuk prestasi dan kecekapan



Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



Learn Git in your browser for free with **Try Git**.



## About

The advantages of Git compared to other source control systems.



## Documentation

Command reference pages, Pro Git book content, videos and other material.



## Downloads

GUI clients and binary releases for all major platforms.



## Community

Get Involved! Mailing list, chat, development and more.



**Pro Git** by Scott Chacon is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



Windows GUIs



Tarballs



Mac Build



Source Code



# GIT

## Pengenalan

Git adalah perisian sumber terbuka dan merupakan **distribute** *version control system* yang direkabentuk untuk prestasi dan kecekapan

## ***Fully Distribute***

**(hampir) semuanya dalam local**

**Bermaksud:**

- Laju
- Klon adalah backup
- Kerja secara offline

# **GIT**

## **Konsep**

- Decentralized
  - Semua boleh jadi server..
- Offline
  - Salinan local adalah repositori.
  - Commit/merge changes



## **Perbezaan Git dengan {cvs|svn|p4}**

***Git is not better than CVS/Subversion/ Perforce/Bazaar  
But is also not worse. It's different.***

- Staging Area



# GIT

**Konsep (err.. keburukan?)**

- Kompleksiti

**Checkout vs Clone**

**Commit vs Push**

Kena tahu arahan itu sama ada bertindak pada local atau pada server



# GITHUB

## Pengenalan



**github**  
SOCIAL CODING

- Web-based hosting untuk pembangunan projek yang menggunakan Git sebagai kawalan versi.
- GitHub, Inc (dahulu dikenali sebagai Logical Awesome) ditubuhkan pada April 2008.

# Github, Inc

San Francisco, USA

548 4th St

San Francisco, CA, 94107

USA

- Tom Preston-Werner – Co-Founder / CEO
- PJ Hyett - Founder / COO.
- Chris Wanstrath - Founder

# Akaun

## Github (contoh)

### Individu

```
https://github.com/hariadi
```

### Organisasi

```
https://github.com/pesima
```

### Avatar (optional)

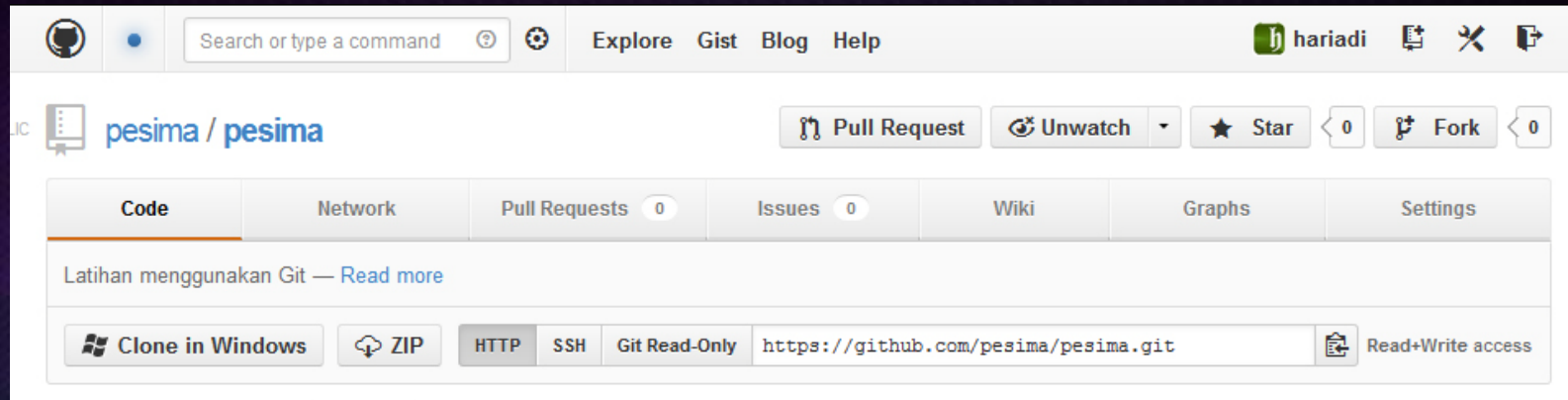
```
https://gravatar.com
```



- GitHub Flavored Markdown (GFM)
  - Isu
  - Mesej
  - komen

# Konsep

## Github Repositori



Setiap repositori mengandung:

- Kod sumber
- Rangkaian kolaborasi
- Pull request
- Isu
- Wiki
- Graf
- konfigurasi

# SESI 2

## PEMASANGAN



# Pemasangan Git

## Linux

### Debian/Ubuntu

```
apt-get install git
```

# Pemasangan Git

## Windows

Git

```
https://help.github.com/articles/set-up-git
```

# Konfigurasi

## Linux/Windows

### Identiti

```
git config --global user.name "Hariadi Hinta"  
git config --global user.email "diperakui@gmail.com"
```

### Global

```
git config user.name "Hariadi Hinta"  
git config user.email "hariadi@jpa.gov.my"
```

### Per projek



# SSH (Optional)

Linux/Windows

## Terminal/Putty

```
ssh-keygen -t rsa -C "your_email@example.com"
```

## Putty

```
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
```

# SESI 3

## KOLABORASI

# Initializing Git

## Init

```
$ git init
```

## Status

```
$ git status
```

## Add & Commit

```
$ touch README  
$ git status  
$ git add README  
$ git commit -m 'Initial commit'
```



# Cloning

## Git Protokol

### LOCAL (Basic)

```
$ git clone file:///main/code  
$ git clone file:///192.168.10.51/code  
$ git clone file:///<host>/<share>/<path>
```

### HTTPS

```
$ git clone file://github.com/pesima/Spoon-Knife
```

### SSH

```
$ git clone ssh://user@server/projek.git  
$ git clone ssh://hariadi@github.com/org.git
```

# Best Practice

## Contribution (Sync Fork )

### FORK

```
https://github.com/pesima/pesima.git
```

### CLONE YOUR OWN REPO

```
$ git clone https://github.com/<user>/pesima.git
```

### ADD REMOTE UPSTREAM

```
$ git remote -v  
$ git remote add upstream https://github.com/pesima/pesima.git
```

# Best Practice (samb.)

Contribution (Sync Fork )

## VERIFY REMOTE

```
$ git remote -v
```

## FETCHING

```
$ git pull upstream
```

## VERIFY

```
$ git branch -va
```



# Best Practice (samb.)

Contribution (Sync Fork )

## CHECKOUT MASTER (LOCAL)

```
$ git checkout master
```

## MERGING

```
$ git merge upstream/master
```

If your local branch didn't have any unique commits, git will  
instead perform a "fast-forward"

# Best Practice (samb.)

## Contribution (Sync Fork )

### CONTRIBUTE ON YOUR BRANCH

```
$ git checkout -b <your-fix-or-features-branch>
```

### PULL

```
$ git pull origin <your-fix-or-features-branch>
```

# Best Practice (samb.)

## SUMMARY

### SYNCING FORK

Fork <https://github.com/pesima/pesima.git>

```
$ mkdir pesima && cd pesima
$ git clone https://github.com/<username>/pesima.git
$ cd pesima
$ git remote add upstream
https://github.com/pesima/pesima.git
$ git checkout master
$ git pull upstream master
```



# SESI 4

## Latihan Github



# Account

```
https://github.com/<user>
```

# LATIHAN 1

**PESIMA: : Fork/Remote/Pull/Checkout/Branch**

## FORK REPOSITORY PESIMA

```
https://github.com/pesima/pesima.git
```

## CLONE

```
$ git clone https://github.com/<username>/pesima.git
```

## ADD REMOTE UPSTREAM

```
$ git remote add upstream  
https://github.com/pesima/pesima.git
```

## PULL UPSTREAM

```
$ git pull upstream master
```



# LATIHAN 1 (samb.)

PESIMA

## BRANCH

```
$ git checkout -b <topic-branch-name>
```

## MODIFY

```
$ vi kolaborasi.md
```

# LATIHAN 2: PESIMA Pages

PESIMA

## OBJECTIVE

`https://pesima.github.com/`

SOALAN

Q&A



ADD-ON

Node.js

## Pemasangan (Windows)

`http://nodejs.org`

```
$ npm install -g twitter-bootstrap-node
```

# Building Twitter Bootstrap

Nodejs, Jekyll

## Pemasangan (Windows)

```
$ git clone https://github.com/twitter/bootstrap.git
```

```
$ npm install
```



# Terima Kasih

[\*\*https://github.com/hariadi\*\*](https://github.com/hariadi)

[\*\*https://github.com/jpagov\*\*](https://github.com/jpagov)

[\*\*https://github.com/pesima\*\*](https://github.com/pesima)