

# **STANDARISASI CODE REPOSITORY MANAGEMENT**

**Versi 1.1**



Tanggal 17 Agustus 2018

**PT TELEKOMUNIKASI INDONESIA  
DIVISI DIGITAL SERVICE  
Jl Kebon Sirih 12  
Jakarta  
10110**

## A. PENDAHULUAN

Inisiatif pengembangan aplikasi digital yang dilakukan di Divisi Digital Service dilakukan oleh banyak inisiator dalam waktu yang hampir bersamaan. Hal ini menyebabkan pertumbuhan *source code* yang ada terus meningkat dan terjadi secara sporadis. Tidak tertutup kemungkinan bahwa pada tingkat tertentu, antar inisiatif tersebut memiliki kesamaan sifat dan kebutuhan asset digital sehingga banyak terjadi duplikasi aktivitas. Dengan adanya standarisasi ini diharapkan setiap aktivitas agile development dapat menjadi produsen ataupun konsumen *source code* yang dibutuhkan pada suatu aktivitas *agile development*.

## B. MAKSUD DAN TUJUAN

Maksud dan tujuan dibuat dokumen standarisasi ini adalah untuk memastikan bahwa seluruh proses pengembangan aplikasi digital yang dilakukan di lingkungan Divisi Digital Service terdokumentasi dengan baik dan mendukung prinsip-prinsip:

1. **reusability** sehingga proses pengembangan aplikasi digital dapat dilakukan dengan lebih cepat, efisien dan berkualitas.
2. **Devops** sehingga konsep CI/CD dapat berjalan dengan baik dan mendukung konsep *incremental product*.

## C. STANDAR

### 1. Penggunaan Repository

Setiap *project* inisiatif digital baik yang dikerjakan *in-house*, *vendor*, *freelancer*, dan lain-lain harus membuat dan menggunakan repository sebelum dan sampai selesainya proses *development*.

### 2. Version Control System

*Version Control System* yang digunakan adalah *Git* dan menggunakan <https://gitlab.playcourt.id>, kecuali untuk kasus-kasus tertentu yang telah disepakati.

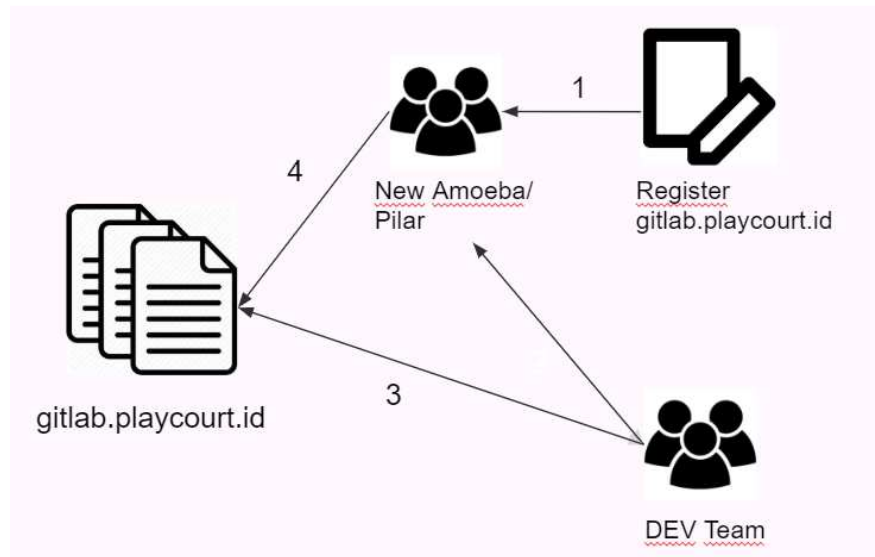
### 3. Project Naming

- a. Pemberian nama untuk sebuah *project / repository* harus menggunakan huruf kecil semua (lowercase).
- b. Apabila nama *project* terdiri dari beberapa kata, maka pemisah antar kata menggunakan *hyphens* (-) bukan *underscore* (\_), *space* ( ), atau simbol-simbol yang lain.

### 4. Repository Account

- a. Setiap proyek harus memiliki **Master Account** yang terdaftar di [gitlab.playcourt.id](https://gitlab.playcourt.id)
- b. **Product Owner (PO)** yaitu **Amoeba/Pilar Bisnis/dll** mendaftarkan setiap proyek yg akan didevelop utk dibuatkan master repository oleh Tim CoDex chapter DeV melalui <http://bit.ly/codex-standardization>

- c. Tim codex dari chapter dev akan menginvite account master yg d daftarkan oleh PO di <http://bit.ly/codex-standardization> dan selanjutnya account Master tsb dpt dipakai oleh PO utk akses ke repository yang dibuat tim CoDex Chapter DEV.
- d. **Masing-masing developer** pada proyek tersebut harus memiliki **account tersendiri** untuk mengakses ke repository tersebut.
- e. *Account username* menggunakan format **<unit name> - <nickname>**.
- f. **Developer** diwajibkan melakukan **push** atas code yang dibuat ke repository tersebut.
- g. Setiap developer diharapkan mengakses repository dengan **SSH** bukan HTTP/HTTPS.



Gambar 1. Repository Management Flow

## 5. Description

- a. Deskripsi adalah penjelasan tentang cara menjalankan aplikasi dan tools pendukungnya yang ditulis oleh Developer yang berkontribusi ketika pembuatan aplikasi. Deskripsi ditulis pada README file (README.md)
- b. Deskripsi bertujuan untuk membantu para developer pengguna repo tsb dalam mempelajari dan menyempurnakan aplikasi.
- c. Deskripsi sedikitnya berisi informasi tentang nama project, kegunaan/fungsi aplikasi, dan langkah-langkah untuk menjalankan aplikasi.
- d. Tata cara penulisan deskripsi adalah sbb:

### 1) Project Title

Menginformasikan nama project dan kegunaan/fungsi aplikasi pada repository.

### 2) Prerequisites

Menginformasikan software/tools pendukung yang dibutuhkan dan mendiskripsikan bagaimana cara untuk menginstallnya.

### 3) Installing

Langkah - langkah untuk menjalankan aplikasi, menginstall dan mengatur konfigurasi yang dibutuhkan.

**4) Running the tests**

Mendiskripsikan langkah-langkah untuk menjalankan unit test dan mengatur konfigurasinya apabila ada.

**5) Deployment**

Menjelaskan tentang cara deploy aplikasi khususnya disisi server.

**6) Built With**

Menjelaskan tool yang digunakan untuk mem-built aplikasi

**7) Author**

Menginformasikan semua developer yang berkontribusi dalam pembuatan aplikasi.

**8) License**

Mendeskripsikan tentang lisensi, claim kepemilikan dan penggunaan repository.

Contoh:

####Project Title

This Project is used to create mockup API

####Prerequisites

What things you need to install the software and how to install them

####Installing

A step by step series of examples that tell you have to get a development env running

Say what the step will be

And repeat

until finished

End with an example of getting some data out of the system or using it for a little demo

##Running the tests

Explain how to run the automated tests for this system

##Deployment

Add additional notes about how to deploy this on a live system

##Built With

\*[Dropwizard](http://www.dropwizard.io/1.0.2/docs/) - The web framework used

\*[Maven](https://maven.apache.org/) - Dependency Management

\*[ROME](https://rometools.github.io/rome/) - Used to generate RSS Feeds

##Authors

\*\*Billie Thompson\* - Initial work - [PurpleBooth](https://github.com/PurpleBooth)

See also the list of [contributors](https://github.com/your/project/contributors) who participated in this project.

##License

This project is licensed under the MT License - see the [LICENSE.md](LICENSE.md) file for details

## 6. Versioning Schemes

*Versioning Schemes* yang digunakan adalah *Semantic Versioning* dengan format **<major>** . **<minor>** . **<patch>**. Perubahan *major number* dilakukan ketika terjadi perubahan yang besar, perubahan *compatibility*, dan lain-lain. Perubahan *minor number* ketika penambahan atau perubahan beberapa features. Perubahan *patch number* ketika terjadinya *bugfix* atau *hotfix*.

## 7. Repository Structure

Struktur Repositoi adalah sebagai berikut:

- accountid
  - a. telkomdds
    - group
      - a. amoeba
        - sub-group
          - <amoeba name>
            - a. project list
              - <project name>
  - b. non-amoeba
    - sub-group
      - <non-amoeba name>
        - a. project list
          - <project name>
- b. non-dds

## 8. Git Workflow

### a. Master Branches

*Master Branch* digunakan untuk mengelola *history release* yang disertai dengan *tag* dan *version number* untuk setiap *commit* yang dilakukan. Branch ini pun yang bakal digunakan sebagai acuan utama atau *single source of truth* dari semua branch.

### b. Development Branches

*Develop Branch* digunakan sebagai *integration branch* antara *feature branch* dan *master branch* sehingga *history* menjadi lebih bersih sebelum benar-benar *release*. *Develop branch* diambil dari *master branch* langsung.

### c. Feature Branches

*Feature Branch* digunakan untuk fokus pada *features* yang dibuat. *Feature branch* tidak boleh berinteraksi langsung dengan *master branch* dan *develop branch* adalah *parent* dari *feature branch* ini.

#### d. Release Branches

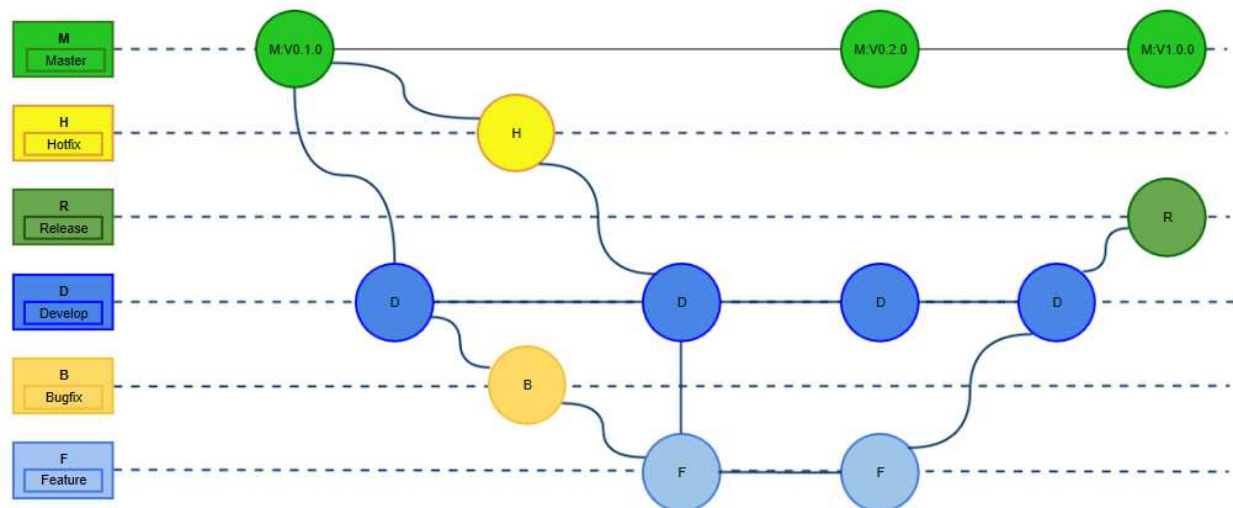
*Release Branch* digunakan sebagai *merged point* setelah *features* selesai dan siap. *Release branch* dibuat dari *develop branch* dan disertai dengan *tag* dan *version number* yang baru sebelum *merge* dilakukan ke *master branch*.

#### e. Hotfix Branches

*Hotfix Branch* digunakan ketika terjadinya situasi dimana harus dilakukan *patching* pada *master branch* segera mungkin. *Hotfix branch* merupakan salah satu *maintenance branch* yang diharapkan tidak mengganggu *workflow* yang lain. *Hotfix branch* diambil langsung dari *master branch* dan ketika selesai harus dilakukan *merge* kembali ke *master branch* yang disertai dengan *tag* dan *version number* yang baru yang berubah di bagian *patch version numbers* saja.

#### f. Bugfix Branches

*Bugfix Branch* merupakan salah satu *maintenance branch* juga seperti *hotfix branch*, tapi *bugfix branch* fokus pada *develop branch* dan *feature branch*. Fungsinya sama yaitu untuk *patching bugs*, tapi ketika masih berada di *development phase*.



Gambar 2. Git Workflow