# Conventional Commits

git commit -m "feat: consistent commit"

#BukaJalan

## The Story of Git Commits

It's all rainbow and sunshine, we have one dev working on repo, he do all the commits, all his commits messages are consistent.

```
295bb9c Add an module #3 for awesome feature
07ab2d4 Fix bug in module #1
ca36bfe Remove unnecessary code in module #2
```

Then second dev is coming and he just have his own commit style.

```
72c370d some big changes in module #3
62f1436 fix bug
87cfb6f add foobar feature
```

## Notice what's wrong?

- Inconsistent commit style hurt readability
- Some commits don't have context, `fix bug`? What? Where?
- No way to categorize/group commits by it's types
- No way to quickly identify commits to Project Tracker (Jira?!)
- Would be hard to make an automated changelog

## Solution?

Please welcome **conventional commits**

## What?

The Conventional Commits specification is a lightweight convention on top of commit messages. It provides an easy set of rules for creating an explicit commit history; which makes it easier to write automated tools on top of. This convention dovetails with SemVer, by describing the features, fixes, and breaking changes made in commit messages.

## Format

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

## Example

```
chore: add xls parser package
feat(employee): add create personal info
fix(BUG-91): fix attendance calculation of night shift employee
refactor(attendance): DEV-1730 remove unused overtime calculation
```

## Common type of commits

- feat
- fix
- chore (adding package, app configuration)
- ci (continous integration related)
- build (build system, ex: npm, webpack, docker)
- test
- docs (README.md, CHANGELOG.md)
- refactor
- perf (code optimization)

## How To Setup

```
# install commitizen as global util
$ npm install commitizen -g

# commitizen initilization
$ commitizen init cz-conventional-changelog --save-dev --save-exact

# or if you use yarn
$ commitizen init cz-conventional-changelog --yarn --dev --exact

# install commitlint
$ npm install -g @commitlint/cli @commitlint/config-conventional
$ echo "module.exports = {extends: ['@commitlint/config-conventional']}" > commitlint.config.js

# use cz to commit
$ git add .
$ cz
```

## Benefits

- Enforce cross developer commit message consistency
- Encourage for atomic commit message
- Ability to link commit with Project Tracker (JIRA?!)
- Easier to work with semver
- Automated tool such as commitizen, commitlint & standard version

## References

- https://www.conventionalcommits.org/en/v1.0.0/
- http://commitizen.github.io/cz-cli/
- https://commitlint.js.org/

## Easter Egg

If you have physical calculator, do this calculation

"From about **2000** developer about **270** developer are not handsome, how many were the handsome?"

After you done with the calculation, reverse your calculator

#BikinBangga

## Sharing Next Session Vote

Pilih topik apa yang mau kita share

**tinyurl.com/next-topic**