

# Pertemuan 2

# Sistem Version Control & Kanban Board

Tim Ajar Dasar Pemrograman 2024

# Tujuan

- Mahasiswa memahami tentang apa itu version control, mengapa penting, dan bagaimana bekerja.
- Mahasiswa akan belajar prinsip-prinsip dasar Kanban, seperti visualisasi alur kerja, pembatasan pekerjaan yang sedang berlangsung, dan peningkatan berkelanjutan.

# Pendahuluan



Flashdisk

VS

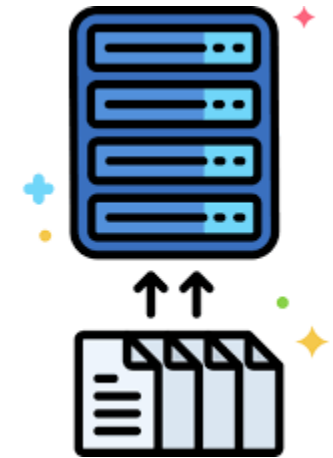


Repository

Repository memungkinkan tim pengembang untuk **bekerja bersama secara terstruktur, melacak perubahan kode, dan mengelola versi** berbagai komponen proyek.

# Apa itu Repositori?

Sebuah repositori (repository) dalam pengembangan perangkat lunak adalah tempat penyimpanan digital yang digunakan untuk **mengelola kode sumber, berkas konfigurasi, dokumen, dan sumber daya lainnya** yang terkait dengan suatu proyek.



# Komponen utama dari repositori

- **Kode Sumber:** berkas-berkas yang berisi **instruksi-instruksi yang akan dieksekusi** oleh komputer untuk menjalankan perangkat lunak.
- **Dokumen dan Berkas Konfigurasi:** **berkas-berkas dokumentasi, petunjuk instalasi, dan berkas konfigurasi** yang diperlukan untuk menjalankan perangkat lunak.
- **Isu dan Tugas (Issues):** menyediakan fitur untuk **melaporkan isu atau masalah** yang ditemukan dalam perangkat lunak.
- **Branch dan Merge:** memungkinkan pengembang untuk bekerja pada fitur atau perbaikan tertentu **tanpa mempengaruhi kode di cabang utama**. Setelah selesai, cabang tersebut dapat digabungkan (merge) kembali ke cabang utama.
- **Riwayat Perubahan (Commit History):** mencatat setiap perubahan yang dilakukan pada kode sumber



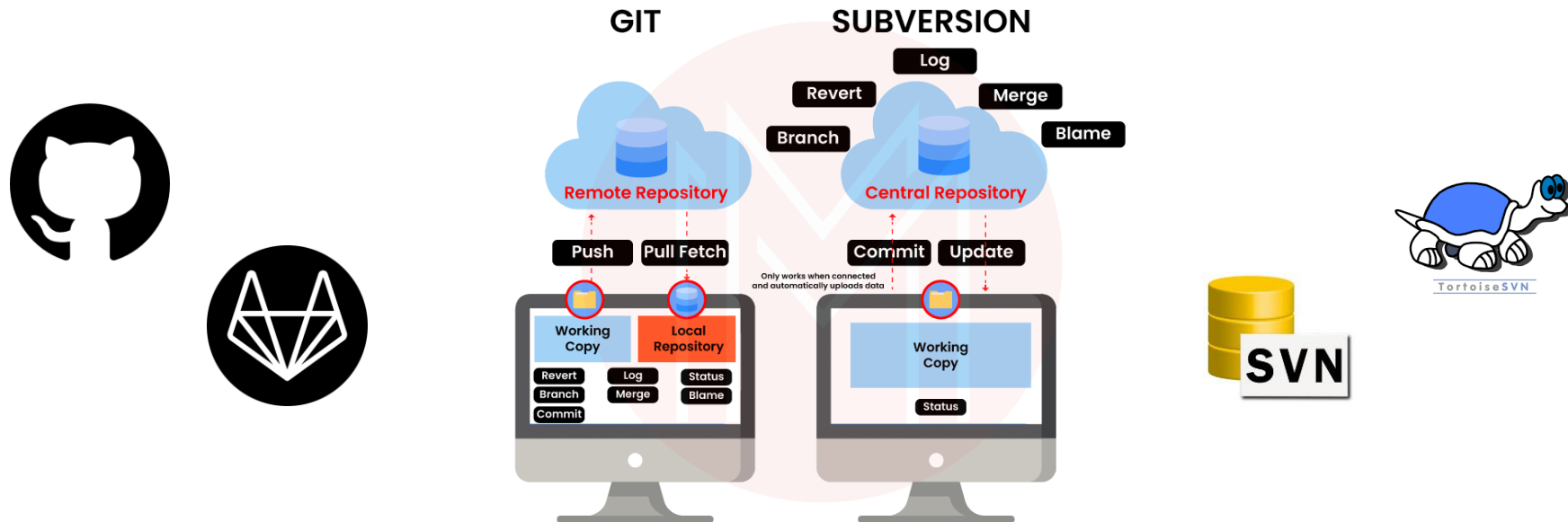
# Sistem Version Control (VCS)

Sistem Version Control (VCS) adalah sistem yang digunakan untuk **melacak perubahan pada kode sumber dan berkas-berkas lain** dalam suatu proyek perangkat lunak.



# Tipe-tipe VCS

- Sistem Version Control Tersebar (Distributed VCS): setiap **anggota tim memiliki salinan penuh** dari repositori. Contoh: Git.
- Sistem Version Control Sentral (Centralized VCS): repositori **pusat menyimpan semua versi kode sumber** dan berkas-berkas lainnya. Contoh: Subversion.



# Manfaat Sistem Version Control

- **Pelacakan Perubahan:** mencatat setiap perubahan yang dilakukan pada kode, memungkinkan pengembang untuk **melihat sejarah perubahan dan kembali ke versi sebelumnya** jika diperlukan.
- **Kolaborasi Tim:** memungkinkan beberapa pengembang untuk **bekerja pada kode yang sama secara bersamaan**, menggabungkan perubahan dengan aman.
- **Manajemen Konflik:** ketika dua atau lebih pengembang melakukan perubahan pada bagian yang sama dari kode, VCS **membantu mendeteksi dan menyelesaikan konflik** yang mungkin timbul.
- **Pemulihan dan Pemecahan Masalah:** Jika ada masalah dengan perubahan terbaru, memungkinkan untuk dengan **mudah kembali ke versi sebelumnya** yang berfungsi.
- **Revisi dan Versi:** memberi **nomor unik pada setiap versi kode**, memungkinkan pengembang untuk mengacu pada versi tertentu.
- **Pengujian dan Pengujian Perangkat Lunak:** memungkinkan untuk mengembangkan dalam lingkungan yang terisolasi, VCS mendukung **pengujian dan eksperimen tanpa mempengaruhi kode inti**.



# Peran GitHub, GitLab, dan Platform Serupa

Peran utamanya adalah menyediakan infrastruktur untuk **mengelola repositori Git dan memfasilitasi kolaborasi tim** dalam pengembangan perangkat lunak.

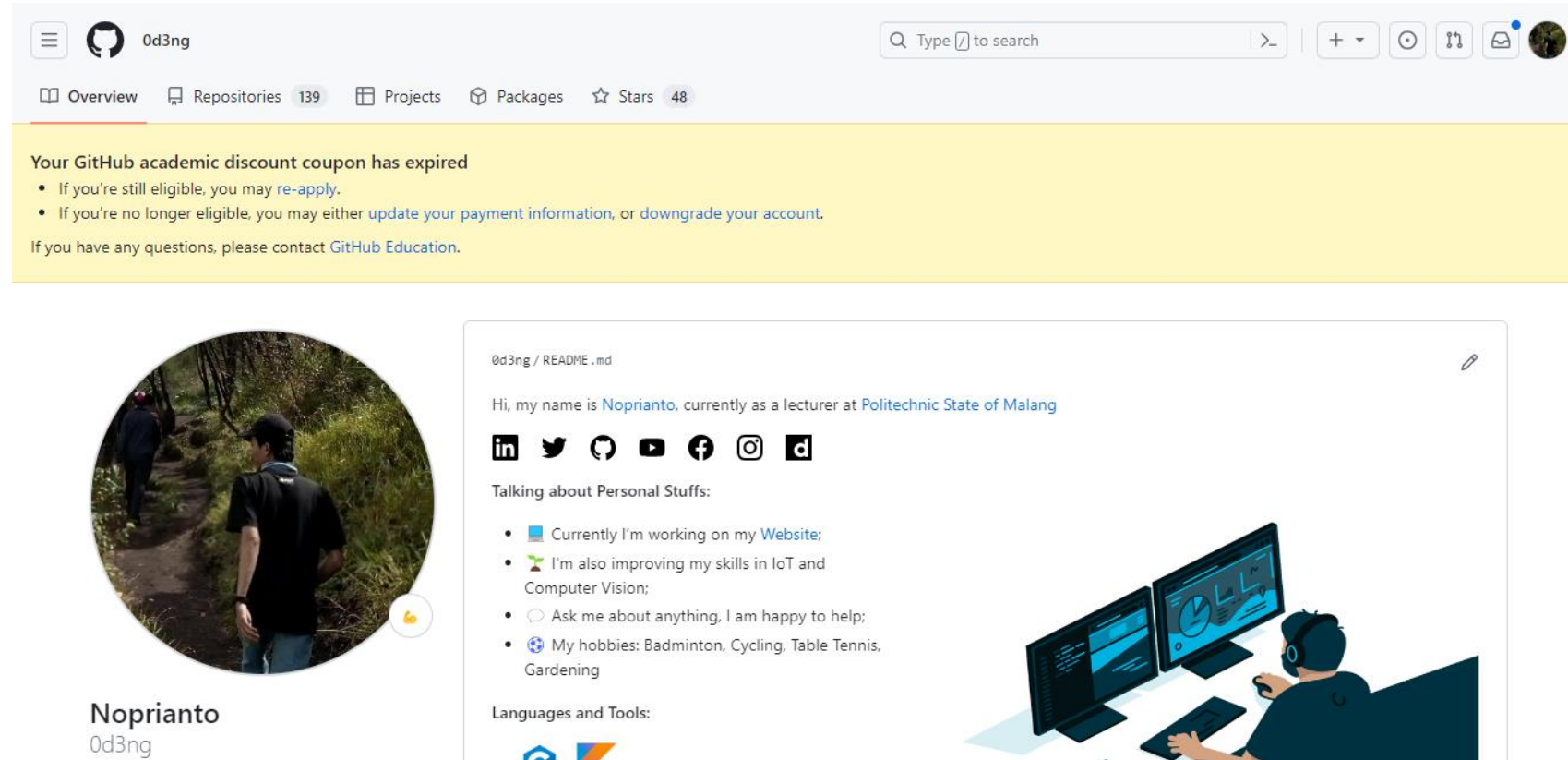
- **Hosting Repositori**; tempat untuk **menyimpan repositori** Git secara online.
- **Kolaborasi Tim**; mengizinkan tim pengembang untuk **bekerja bersama** dalam satu repositori.
- **Manajemen Isu (Issues) dan Tugas**; melacak dan mengelola isu, permintaan fitur, dan tugas dalam proyek.
- **Pull Request (Permintaan Tarik)**; mengajukan perubahan kode untuk ditinjau oleh pengembang lain sebelum digabungkan ke repositori utama.
- **Integrasi CI/CD**; mendukung **otomatisasi pengujian, penggabungan otomatis, dan penyebaran perangkat lunak**.
- **Pengaturan Akses**; mengelola **hak akses dan izin berdasarkan peran** untuk memastikan kontrol yang tepat terhadap repositori.

# Peran GitHub, GitLab, dan Platform Serupa (2)

- **Dokumentasi Proyek**; menyediakan **dokumentasi proyek dan panduan** bagi pengembang dan pengguna.
- **Peninjauan Kode (Code Review)**; mendukung proses **peninjauan kode dengan komentar, saran, dan tanggapan** lainnya.
- **Integrasi Pihak Ketiga**; memiliki **integrasi dengan alat-alat pengembangan lain** seperti alat manajemen proyek, sistem pelacakan isu, dan alat perencanaan.
- **Publikasi Open Source**; pengembangan perangkat lunak sumber terbuka dengan **memudahkan berbagi kode dengan komunitas**.
- **Versi Berbayar dan Gratis**; **berbayar dengan fitur tambahan**, sementara yang lain memiliki opsi **gratis untuk proyek open source**.

# GitHub

- GitHub adalah platform pengelolaan repositori berbasis web yang memanfaatkan sistem kontrol versi Git.
- Salah satu platform paling populer dan digunakan secara luas oleh pengembang perangkat lunak, baik dalam proyek open source maupun proyek swasta.



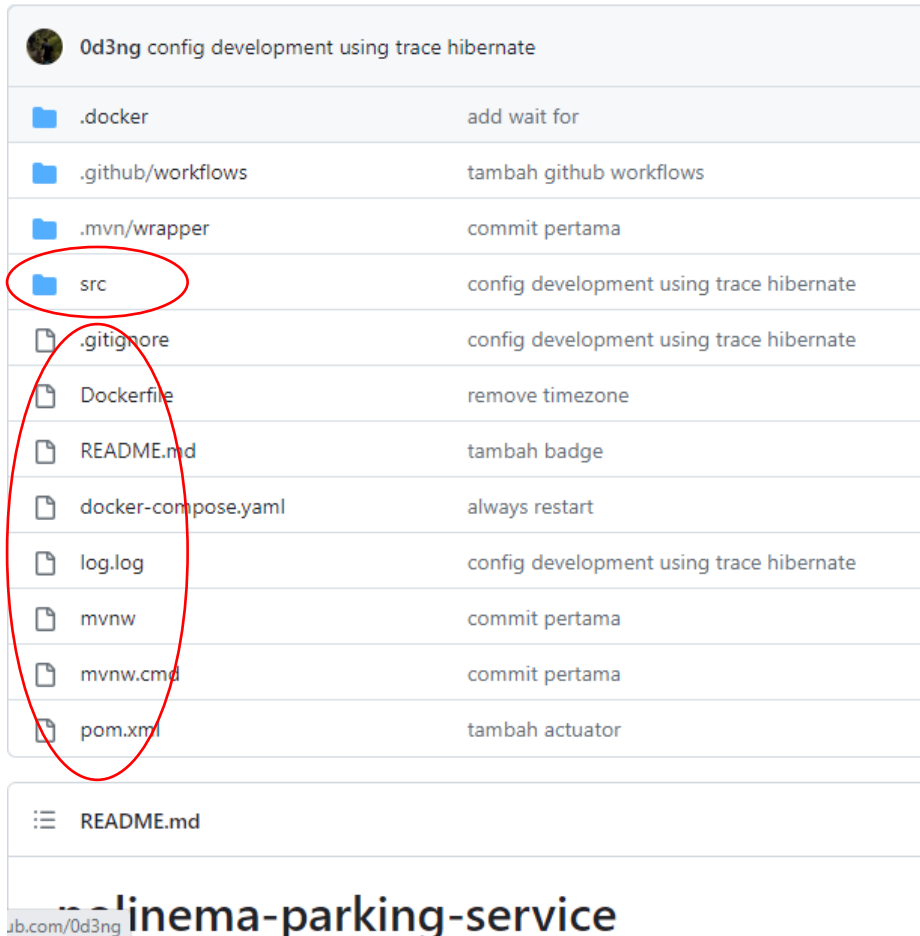
The screenshot shows the GitHub profile of user '0d3ng'. At the top, there's a navigation bar with a search bar and icons for repositories (139), projects, packages, stars (48), and a notification bell. Below the navigation bar, a yellow banner states: "Your GitHub academic discount coupon has expired". It lists two options: "If you're still eligible, you may [re-apply](#)." and "If you're no longer eligible, you may either [update your payment information](#), or [downgrade your account](#)." It also provides a link to "GitHub Education" for questions. The profile section features a circular profile picture of a person walking on a path, the name "Noprianto", and the username "0d3ng". To the right of the profile picture is a list of social media links (LinkedIn, Twitter, GitHub, YouTube, Facebook, Instagram, and Dribbble). Below the links, there's a section titled "Talking about Personal Stuff:" with a bulleted list: "Currently I'm working on my [Website](#);", "I'm also improving my skills in IoT and Computer Vision;", "Ask me about anything, I am happy to help;", and "My hobbies: Badminton, Cycling, Table Tennis, Gardening". At the bottom, there's a section titled "Languages and Tools:" with icons for JavaScript and Python. On the far right, there's an illustration of a person wearing headphones and working on two computer monitors.

# Fitur Utama GitHub

- **Repository**; membuat repository publik atau pribadi untuk **menyimpan kode sumber, berkas konfigurasi, dokumen, dan sumber daya lainnya**
- **Git Integration**; melakukan operasi standar Git seperti **commit, branch, merge, dan rebase** melalui antarmuka web atau perintah baris
- **Kolaborasi**; dapat **bekerja bersama** dalam repository yang sama
- **Integrasi CI/CD**; GitHub Actions adalah **alat integrasi Continuous Integration (CI) dan Continuous Deployment (CD)** yang terintegrasi langsung dengan repository GitHub
- **GitHub Pages**; memungkinkan pengguna untuk **meng-host situs web statis** langsung dari repository GitHub
- **GitHub Classroom**; **platform pendidikan** yang memfasilitasi pengajaran dan pembelajaran Git dan GitHub dalam lingkungan kelas



# GitHub - Repositori



0d3ng	config development using trace hibernate
.docker	add wait for
.github/workflows	tambah github workflows
.mvn/wrapper	commit pertama
src	config development using trace hibernate
.gitignore	config development using trace hibernate
Dockerfile	remove timezone
README.md	tambah badge
docker-compose.yaml	always restart
log.log	config development using trace hibernate
mvnw	commit pertama
mvnw.cmd	commit pertama
pom.xml	tambah actuator

cinema-parking-service

- **src**; folder yang digunakan untuk menyimpan kode program.
- File – file yang lain adalah berisi tentang konfigurasi sebuah proyek. Misalkan file **pom.xml** adalah file konfigurasi untuk menyimpan seluruh depedensi/Pustaka ketika menggunakan Maven (build tool).

# GitHub - Git Integration

## Commits

master

Commits on Dec 11, 2020

config development using trace hibernate  
0d3ng committed on Dec 11, 2020

Commits on Dec 4, 2020

Merge remote-tracking branch 'origin/master'  
0d3ng committed on Dec 4, 2020

tambahan untuk time received data  
0d3ng committed on Dec 4, 2020

Commits on May 4, 2020

always restart  
0d3ng committed on May 4, 2020

tambahan untuk time received data

master

0d3ng committed on Dec 4, 2020

1 parent c5518cf commit 4af4283

Showing 1 changed file with 8 additions and 0 deletions.

Split Unified

src/main/java/com/sinaungoding/parking/controller/RekapController.java

```
30
31 import javax.validation.Valid;
32 import java.sql.SQLIntegrityConstraintViolationException;
33 + import java.time.LocalDateTime;
34 + import java.time.ZoneId;
35 + import java.time.format.DateTimeFormatter;
36 import java.util.ArrayList;
37 import java.util.List;
38

33 import java.util.ArrayList;
34 import java.util.List;
35

44 @Autowired
45 private ModelMapper modelMapper;
46

47 @Autowired
48 private ModelMapper modelMapper;
49
50 + private static final DateTimeFormatter FORMATTER =
51 +     DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
```

Commit adalah memastikan dan mencatat setiap penambahan kode program, setiap proses commit akan dicatat informasinya kapan, siapa, yang mana.

# GitHub - Kolaborasi

## JTI Dasar Pemrograman Menggunakan Binder

Tool online untuk membantu pembelajaran mata kuliah dasar pemrograman Java menggunakan jupyter notebook

Klik link di bawah ini untuk menjalankan

- - mode classic
- - mode lab

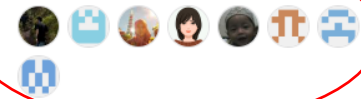
### Dosen pengampu matakuliah

- Noprianto, S.Kom., M.Eng
- Adevian Fairuz Pratama, S.ST, M.Eng
- Ika Kusumaning Putri, S.Kom., M.T.
- Mamluatul Hani'ah, S.Kom., M.Kom.
- Mungki Astiningrum, ST., M.Kom.
- Mustika Mentari, S.Kom., M.Kom
- Pramana Yoga Saputra, S.Kom., MMT.
- Vivin Ayu Lestari, S.Pd., M.Kom
- Vivi Nur Wijyaningrum, S.Kom, M.Kom
- Moch. Zawaruddin Abdullah, S.ST., M.Kom
- Imam Fahrur Rozi, ST., MT.

### Packages

No packages published  
[Publish your first package](#)

### Contributors 8



### Languages

Jupyter Notebook 99.7%  
Dockerfile 0.3%

### Suggested Workflows

Based on your tech stack



Actions Importer

[Set up](#)

Automatically convert CI/CD files to YAML for GitHub Actions

### Commits on Nov 7, 2021

update file

vivinurw committed on Nov 7, 2021

kecilkan gambar

vivinurw committed on Nov 7, 2021

Update gambar

vivinurw committed on Nov 7, 2021

Update array 2

vivinurw committed on Nov 7, 2021

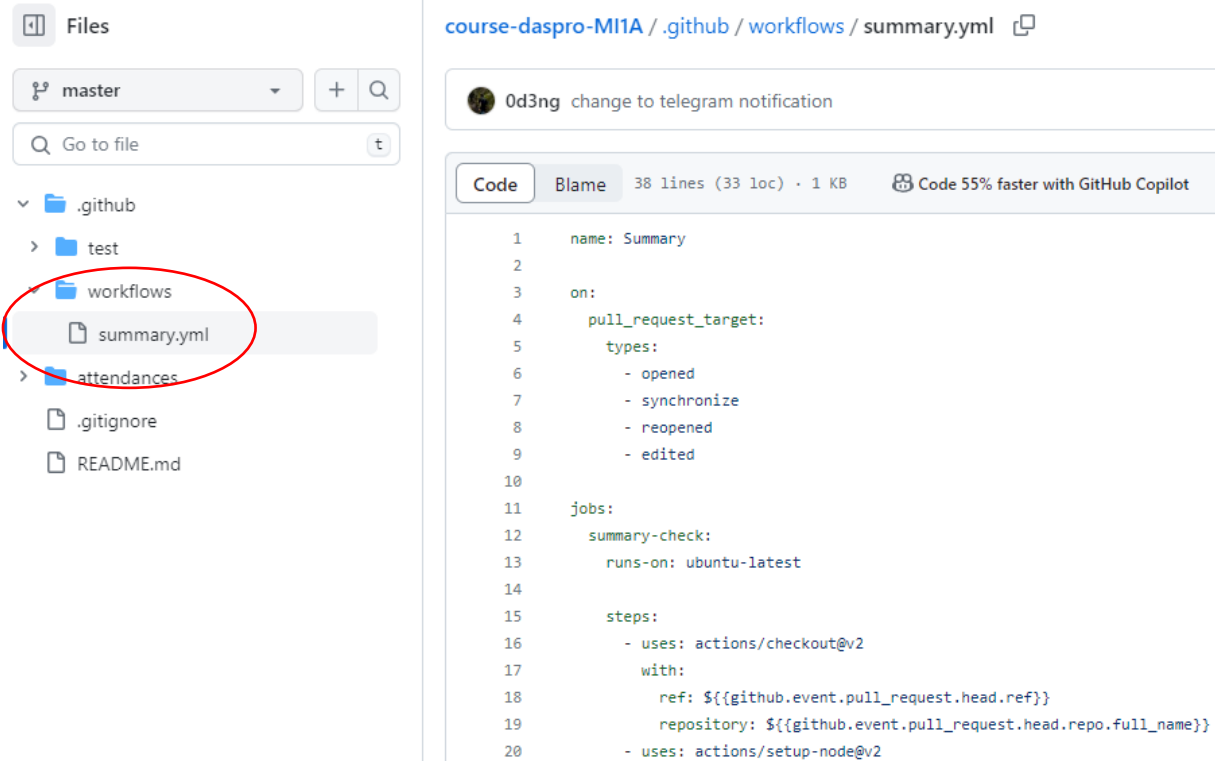
Array 2

vivinurw committed on Nov 7, 2021

Banyak pengguna bisa memberikan kontribusi ke dalam proyek yang sama.

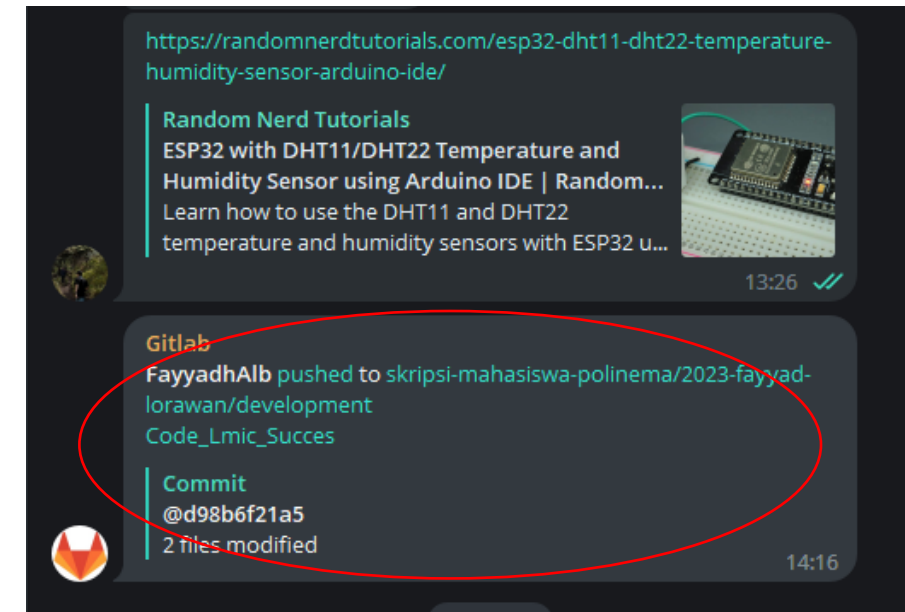


# GitHub - Integrasi CI/CD



The screenshot shows the GitHub repository interface for `course-daspro-MI1A`. The left sidebar displays the file structure, with the `.github/workflows` directory highlighted. The main area shows the content of `summary.yml`, which is a GitHub Actions workflow for sending Telegram notifications.

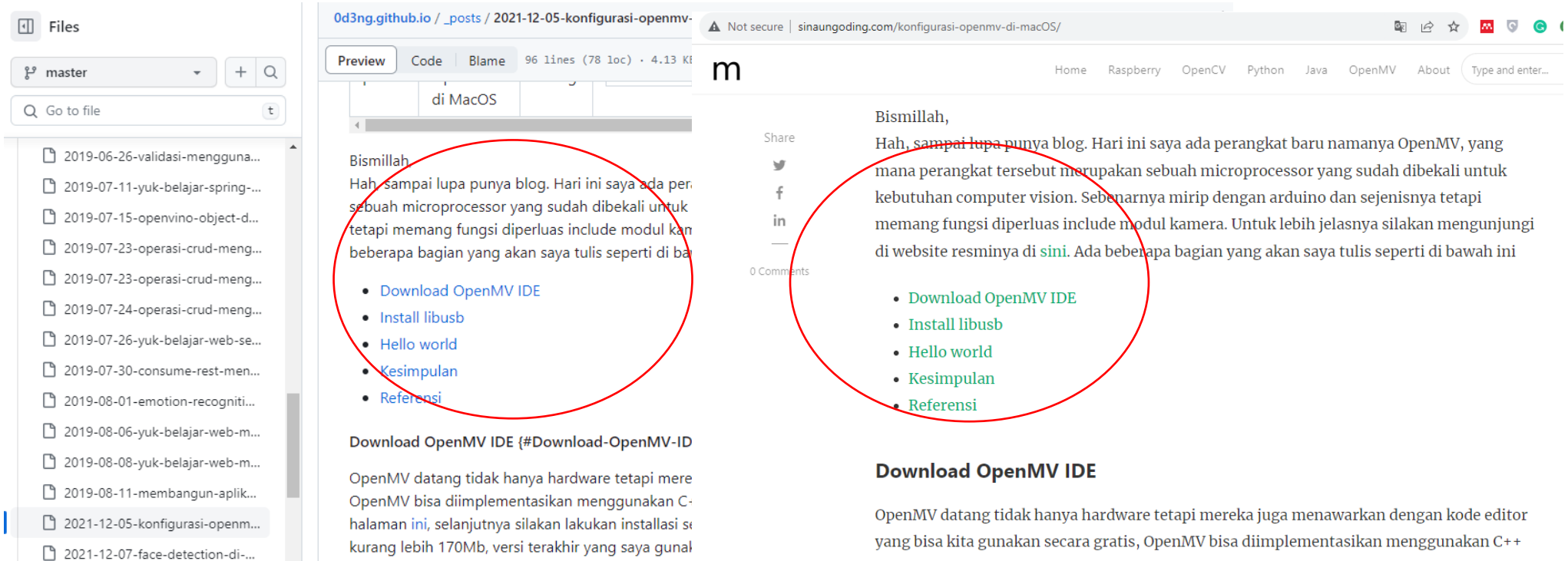
```
1 name: Summary
2
3 on:
4   pull_request_target:
5     types:
6       - opened
7       - synchronize
8       - reopened
9       - edited
10
11 jobs:
12   summary-check:
13     runs-on: ubuntu-latest
14
15   steps:
16     - uses: actions/checkout@v2
17     with:
18       ref: ${github.event.pull_request.head.ref}
19       repository: ${github.event.pull_request.head.repo.full_name}
20     - uses: actions/setup-node@v2
```



Contoh penggunaan **continue integration** yaitu ketika ada aksi pull request atau yang lain, disesuaikan kebutuhan kemudian akan melakukan merger ke master, selanjutnya memberikan notifikasi ke Telegram.



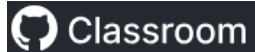
# GitHub - GitHub Pages



The image displays a GitHub repository and its corresponding GitHub Pages website. On the left, the GitHub repository interface shows a file named "2021-12-05-konfigurasi-openmv-di-MacOS" selected. The file content is visible, starting with "Bismillah," followed by a paragraph in Indonesian and a list of links: "Download OpenMV IDE", "Install libusb", "Hello world", "Kesimpulan", and "Referensi". On the right, the GitHub Pages website is shown, displaying the same content. The website has a navigation bar with links to "Home", "Raspberry", "OpenCV", "Python", "Java", "OpenMV", and "About". The main content area includes the same text and links as the repository. Red circles highlight the content on both the repository and the website.

Kita bisa membuat sebuah **website statis** untuk kebutuhan **dokumentasi proyek, portofolio, atau halaman pribadi** di **GitHub** menggunakan **GitHub Pages**.

# GitHub - GitHub Classroom



[Classrooms](#) / [daspro-1f-classroom-0aa819](#) / New assignment

Let's set up the basics for your assignment.

## Assignment creation steps

⚙️ Assignment basics

💻 Starter code and environment

💬 Grading and feedback

Assignment title \*

Intro to Data Structures

Student assignment repositories will have the prefix: `e.g. intro-to-data-structures-anakin-skywalker` ✎

Deadline

dd/mm/yyyy --:--

(Optional) If left blank, there will be no deadline. Date format: YYYY-MM-DD HH:MM a

Individual or group assignment

Individual assignment

GitHub Classroom memiliki fungsi seperti **Learning Management System (LMS)** untuk **mengelola kelas** seperti memberikan tugas, memberikan materi, melakukan penilaian, dan fungsi yang lain untuk pembelajaran.

# Memulai GitHub

- **Membuat Akun GitHub:**

- Kunjungi [github.com](https://github.com) dan klik "Sign up" untuk membuat akun baru. Ikuti langkah-langkahnya untuk mendaftar dengan alamat email dan username.

- **Menginstal Git:**

- Git adalah alat yang digunakan untuk melacak perubahan pada kode di komputer Anda. Anda bisa mengunduh dan menginstalnya dari [git-scm.com](https://git-scm.com).

- **Mengkonfigurasi Git:**

- Setelah Git terinstal, Anda perlu mengatur nama dan email Anda agar setiap perubahan yang Anda buat dapat dilacak:
  - Atur nama Anda: `git config --global user.name "Nama Anda"`
  - Atur email Anda: `git config --global user.email "email@contoh.com"`

# Membuat dan Mengelola Repository

## •Membuat Repository Baru di GitHub:

- Setelah login ke GitHub, klik tombol "**New**" di halaman utama untuk membuat repository baru.
- Isi nama repository, deskripsi, dan centang "**Initialize this repository with a README**" jika Anda ingin memulai dengan file README.

## •Mengkloning Repository:

- Jika Anda ingin bekerja pada proyek yang sudah ada di GitHub, Anda bisa mengkloningnya ke komputer Anda dengan perintah: `git clone https://github.com/username/repo.git`

## •Struktur Repository:

- Repository biasanya berisi berbagai file dan folder, dan semua perubahan yang dilakukan disimpan sebagai "**commit**".

## •Menambahkan dan Meng-commit File:

- Tambahkan file ke dalam repository: `git add`.
- Commit perubahan: `git commit -m "Pesan commit pertama"`

## •Mendorong Perubahan ke GitHub:

- Setelah commit, Anda bisa mengirim (push) perubahan tersebut ke GitHub: `git push origin main`

# Memahami Branching

## •Apa itu Branch?

- Branch adalah cabang dari repository Anda yang memungkinkan Anda untuk mengerjakan fitur atau perbaikan baru tanpa mempengaruhi kode utama. Ini seperti membuat salinan dari proyek Anda di mana Anda bisa bereksperimen.

## •Membuat dan Berpindah Branch:

- Membuat branch baru: `git branch nama-branch`
- Berpindah ke branch baru: `git checkout nama-branch` atau `git switch nama-branch`

## •Menggabungkan Branch (Merge):

- Setelah selesai bekerja di branch, Anda bisa menggabungkannya kembali ke branch utama: `git merge nama-branch`
- Sering kali, ini memerlukan penyelesaian konflik jika ada perubahan yang bertentangan.

## •Menghapus Branch:

- Setelah menggabungkan, Anda bisa menghapus branch yang sudah tidak diperlukan: `git branch -d nama-branch`

# Menyelesaikan Konflik

## •Apa yang Menyebabkan Konflik?

- Konflik terjadi ketika dua orang atau lebih membuat perubahan yang bertentangan pada bagian kode yang sama. Git tidak bisa secara otomatis memutuskan perubahan mana yang harus disimpan.

## •Mengidentifikasi dan Menyelesaikan Konflik:

- Gunakan `git status` untuk melihat file mana yang mengalami konflik.
- Buka file tersebut, dan Anda akan melihat bagian yang mengalami konflik dengan tanda seperti <<<<<<< HEAD. Anda harus memilih perubahan mana yang akan disimpan.
- Setelah menyelesaikan konflik, tambahkan file tersebut: `git add nama-file` dan lakukan commit untuk menyelesaikan merge.

## •Tips Menghindari Konflik:

- Selalu pull (menarik) perubahan terbaru dari branch utama sebelum mulai bekerja.
- Komunikasikan dengan tim tentang siapa yang mengerjakan bagian apa untuk menghindari tumpang tindih.

# Menyelesaikan Konflik (Contoh)

- Bayangkan Anda memiliki sebuah repository sederhana dengan sebuah file **index.html**. Dua pengembang, **A** dan **B**, bekerja di **branch yang berbeda** dan membuat perubahan pada file **index.html**.
- Branch **main**: Pengembang **A** menambahkan judul "Welcome to My Website" di **index.html**
- Branch **feature-update**: Pengembang **B** menambahkan judul yang berbeda "Hello World!" di **index.html**.

```
<!DOCTYPE html>
<html>
<head>
    <title>My Website</title>
</head>
<body>
    <h1>Welcome to My Website</h1>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>My Website</title>
</head>
<body>
    <h1>Hello World!</h1>
</body>
</html>
```

# Menyelesaikan Konflik (Contoh)

- Pengembang A bekerja di branch main dan membuat perubahan. Dia kemudian melakukan commit dan push perubahan ke repository GitHub.

```
git add index.html  
git commit -m "Added Welcome message"  
git push origin main
```

- Pengembang B bekerja di branch feature-update, membuat perubahan yang berbeda pada file index.html, dan juga melakukan commit.

```
git add index.html  
git commit -m "Changed message to Hello World"
```

- Pengembang B sekarang ingin menggabungkan (merge) perubahan ke branch main. Dia beralih ke branch main dan mencoba melakukan merge.

```
git checkout main  
git merge feature-update
```

**Konflik Terjadi!** Git mendeteksi bahwa ada dua perubahan yang berbeda di bagian yang sama pada file `index.html`. Git tidak bisa memutuskan perubahan mana yang harus dipilih, jadi Git meminta Anda untuk menyelesaikan konflik.





# Menyelesaikan Konflik (Contoh)

- Periksa status

`git status`

dan Git akan menampilkan file mana yang terdapat konflik

- Buka file yang konflik tersebut menggunakan text editor.

Bagian antara <<<<<< HEAD dan ===== adalah perubahan yang ada di branch main. Bagian antara ===== dan >>>>>> feature-update adalah perubahan yang dibuat di branch feature-update.

- Anda perlu memutuskan perubahan mana yang ingin disimpan atau bahkan menggabungkan keduanya. Misalkan Anda memilih untuk menyimpan pesan dari feature-update.
- Setelah selesai mengedit dan konflik terselesaikan, Anda perlu menambahkan perubahan tersebut dan melakukan commit kembali dan merge ke branch main

`git add index.html`

`git commit -m "Resolved merge conflict in index.html"`

`git push origin main`

```
<!DOCTYPE html>
<html>
<head>
  <title>My Website</title>
</head>
<body>
<<<<<< HEAD
  <h1>Welcome to My Website</h1>
=====
  <h1>Hello World!</h1>
>>>>>> feature-update
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <title>My Website</title>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

# Menggunakan .gitignore

- Apa itu .gitignore?
  - .gitignore adalah file yang memberi tahu Git untuk mengabaikan file atau direktori tertentu. Ini berguna untuk mengecualikan file yang tidak perlu, seperti file sementara, log, atau file yang dihasilkan oleh sistem operasi atau IDE.
- Contoh Penggunaan Umum:
  - Mengabaikan file seperti **\*.log**, **\*.tmp**, atau direktori seperti **node\_modules/**.
- Membuat dan Mengkonfigurasi .gitignore:
  - Anda bisa membuat file .gitignore di root repository Anda dan menambahkan pola file atau direktori yang ingin Anda abaikan.
- Best Practice:
  - Perbarui .gitignore secara berkala sesuai dengan kebutuhan proyek Anda.
  - Gunakan alat online seperti [gitignore.io](https://gitignore.io) untuk membuat .gitignore khusus untuk lingkungan pengembangan Anda.

laravel / laravel

<> Code Pull requests Actions Security Insights

Files

11.x + Q

Go to file t

- > .github
- > app
- > bootstrap
- > config
- > database
- > public
- > resources
- > routes
- > storage
- > tests
- .editorconfig
- .env.example
- .gitattributes
- .gitignore**
- .styleci.yml
- CHANGELOG.md
- README.md
- artisan
- composer.json

laravel / .gitignore

princejohnsantillan Add .phpactor.json to .gitignore (#6400) ✓

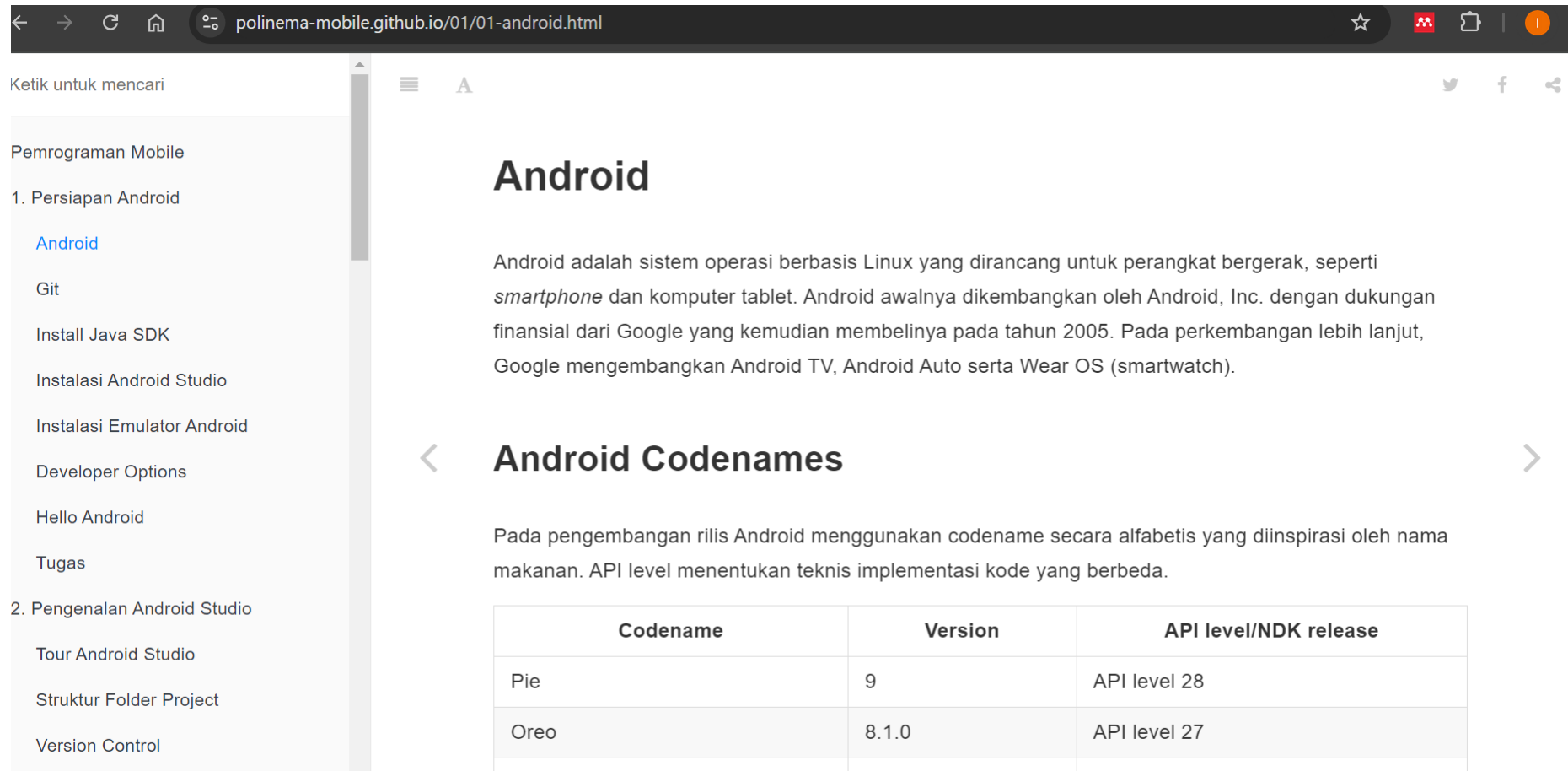
Code Blame 20 lines (20 loc) · 258 Bytes

```
1  /.phpunit.cache
2  /node_modules
3  /public/build
4  /public/hot
5  /public/storage
6  /storage/*.key
7  /vendor
8  .env
9  .env.backup
10 .env.production
11 .phpactor.json
12 .phpunit.result.cache
13 Homestead.json
14 Homestead.yaml
15 auth.json
16 npm-debug.log
17 yarn-error.log
18 /.fleet
19 /.idea
20 /.vscode
```

# Membuat Github Page

- Apa itu GitHub Pages?
  - GitHub Pages adalah layanan dari GitHub yang memungkinkan Anda untuk membuat dan meng-host situs web statis langsung dari repository Anda. Ini sering digunakan untuk mendemonstrasikan proyek atau membuat dokumentasi.
- Membuat GitHub Pages:
  - Buat file index.html di repository Anda.
  - Masuk ke pengaturan (Settings) repository di GitHub dan aktifkan GitHub Pages.
  - Pilih branch (biasanya main) dan direktori root untuk situs Anda.
- Menyesuaikan dengan Jekyll:
  - Anda bisa menggunakan Jekyll untuk menambahkan tema dan template pada situs Anda.
- Mengelola dan Memperbarui:
  - Setiap kali Anda membuat perubahan pada file di repository Anda, GitHub Pages akan otomatis diperbarui.

<https://polinema-mobile.github.io/01/01-android.html>



Ketik untuk mencari

Pemrograman Mobile

1. Persiapan Android

- Android
- Git
- Install Java SDK
- Instalasi Android Studio
- Instalasi Emulator Android
- Developer Options
- Hello Android
- Tugas

2. Pengenalan Android Studio

- Tour Android Studio
- Struktur Folder Project
- Version Control

## Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak, seperti *smartphone* dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc. dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Pada perkembangan lebih lanjut, Google mengembangkan Android TV, Android Auto serta Wear OS (smartwatch).

### Android Codenames

Pada pengembangan rilis Android menggunakan codename secara alfabetis yang diinspirasi oleh nama makanan. API level menentukan teknis implementasi kode yang berbeda.

Codename	Version	API level/NDK release
Pie	9	API level 28
Oreo	8.1.0	API level 27
-	-	-

# Menulis Readme

- Kenapa README Penting?
  - README adalah dokumen pertama yang dilihat oleh pengguna saat mereka mengunjungi repository Anda. Ini menjelaskan tujuan proyek, cara instalasi, cara menggunakan, dan informasi lain yang penting.
- Struktur Umum README:
  - Judul Proyek: Nama proyek dan deskripsi singkat.
  - Instalasi: Langkah-langkah untuk menginstal dan mengatur proyek.
  - Penggunaan: Petunjuk tentang cara menjalankan dan menggunakan proyek.
  - Kontribusi: Panduan bagi yang ingin berkontribusi pada proyek Anda.
  - Lisensi: Informasi tentang lisensi proyek.
- Tips Menulis README:
  - Gunakan bahasa yang jelas dan singkat.
  - Sertakan tangkapan layar atau GIF untuk menjelaskan fitur utama.
  - Perbarui README secara berkala untuk mencerminkan perubahan pada proyek.

> .github  
> app  
> bootstrap  
> config  
> database  
> public  
> resources  
> routes  
> storage  
> tests  
  .editorconfig  
  .env.example  
  .gitattributes  
  .gitignore  
  .styleci.yml  
  CHANGELOG.md  
  **README.md**  
  artisan  
  composer.json  
  package.json  
  phpunit.xml

Preview

Code

Blame

66 lines (45 loc) · 4.01 KB



tests passing downloads 354M packagist v11.21.0 license MIT

## About Laravel

Laravel is a web application framework with expressive, elegant syntax. We believe development must be a joy, and it should be truly fulfilling. Laravel takes the pain out of development by easing common tasks used in many web applications.

- [Simple, fast routing engine.](#)
- [Powerful dependency injection container.](#)
- Multiple back-ends for [session](#) and [cache](#) storage.
- Expressive, intuitive [database ORM](#).
- Database agnostic [schema migrations](#).
- [Robust background job processing](#).
- [Real-time event broadcasting](#).

Laravel is accessible, powerful, and provides tools required for large, robust applications.

## Learning Laravel

# Kanban Board - Pendahuluan



- Ada yang pernah melihat benda seperti apa?
- Tujuannya buat apa?



# Apa itu Kanban Board?

- **Kanban Board** sebagai alat visual untuk **melacak tugas dan pekerjaan**.
- Kanban board memberikan cara yang sangat efektif untuk **memvisualisasikan status tugas, mengatur alur kerja, dan meningkatkan produktivitas**.
- Kanban board terdiri dari **kolom-kolom yang mewakili tahapan-tahapan** berbeda dalam alur kerja, serta **kartu-kartu atau item yang mewakili tugas** atau pekerjaan yang harus dilakukan.
- Setiap kartu **dipindahkan secara visual dari satu kolom ke kolom berikutnya** seiring dengan kemajuan tugas.
- Memberikan gambaran yang jelas tentang **alur kerja, status tugas, dan mengidentifikasi hambatan** atau masalah dengan cepat.



# Manfaat utama dari Kanban board

- **Visualisasi alur kerja;** memberikan gambaran visual yang jelas tentang alur kerja dan status tugas.
- **Pengelolaan waktu dan prioritas;** memungkinkan pengaturan prioritas dan mengontrol aliran tugas.
- **Kolaborasi tim;** membantu tim berkomunikasi tentang pekerjaan, bertanggung jawab, dan menyelesaikan masalah.
- **Perbaikan berkelanjutan;** memfasilitasi evaluasi dan perbaikan terus-menerus dalam alur kerja.
- **Pengurangan pemborosan;** mengurangi waktu tunggu dan kelebihan pekerjaan dalam alur kerja.
- **Adaptasi terhadap perubahan;** memungkinkan fleksibilitas dalam menghadapi perubahan dan prioritas baru.
- **Transparansi;** semua anggota tim dapat dengan mudah melihat status tugas dan progres.



# Struktur Kanban Board

- **Backlog (To Do)**; berisi daftar tugas atau pekerjaan yang **belum dimulai**.
- **In Progress**; berisi tugas atau pekerjaan yang **sedang aktif** dikerjakan oleh tim.
- **Review**; tahap di mana tugas atau pekerjaan telah selesai dikerjakan dan **sedang ditinjau atau direview** oleh anggota tim lain atau pihak terkait.
- **Testing/QA**; berisi tugas atau pekerjaan yang telah melewati tahap review dan **siap untuk diuji atau diverifikasi**.
- **Done**; tugas atau pekerjaan yang **telah selesai** dikerjakan, diuji, dan disetujui dipindahkan ke tahap ini.

**Struktur Kanban Board dapat disesuaikan dengan kebutuhan dan alur kerja spesifik tim atau proyek.**



# Pengelolaan Waktu dan Prioritas – Kanban Board

- Tempatkan tugas-tugas yang memiliki **prioritas tertinggi** di **bagian atas** kolom "To Do" atau backlog.
- Tentukan **batasan berapa banyak tugas** yang boleh berada dalam tahap "In Progress" secara bersamaan.
- Dorong tim untuk **menyelesaikan satu tugas** sebelum memulai tugas baru.
- Prioritaskan ulang tugas berdasarkan **perubahan kebutuhan atau situasi**.
- Saat memilih tugas berikutnya untuk dikerjakan, **pertimbangkan kompleksitas, durasi, dan prioritasnya**.
- Gunakan **kode warna atau label** untuk menandai tugas berdasarkan prioritas atau jenis.



# Kolaborasi Tim – Kanban Board

- Gunakan Kanban Board sebagai titik pusat untuk **melihat status pekerjaan seluruh tim**.
- Ketika tugas ditambahkan ke papan Kanban, jelaskan **siapa yang akan mengerjakan tugas** tersebut.
- Gunakan fitur komentar pada kartu atau item Kanban untuk **berdiskusi tentang pekerjaan**
- Jika tim Anda terlibat dalam pengembangan perangkat lunak, gunakan tahap "Review" atau kolom terpisah untuk **peninjauan kode**.
- Jika ada hambatan atau masalah yang mempengaruhi pekerjaan, anggota tim dapat **menandai atau menambahkan isu** terkait pada tugas.
- **Pertemuan rutin tim** untuk dapat melihat papan Kanban bersama dan berbicara tentang pekerjaan yang sedang dikerjakan.



# Software – Kanban Board

- **Trello**; menawarkan antarmuka yang intuitif dengan kartu yang dapat dipindahkan di antara kolom-kolom yang mewakili tahap alur kerja.
- **Notion**: sebagai alat pembuatan dokumen, juga memungkinkan membuat papan Kanban yang dapat diakses tim.
- **Jira software**; digunakan untuk pengembangan perangkat lunak, memiliki fitur Kanban Board yang kuat.





# Kanban Board - Trello



Contoh di atas terdapat 3 pekerjaan atau tugas, di dalam sebuah pekerjaan kita diizinkan atau diperkenankan untuk membuat sebuah list.

# Kanban Board - Notion



Hide description

## Kampus dan Uborampe

Task untuk daily activity urusan akademik bro

Incomplete Complete All +

Aa Task	Urgency	Importance	Compl...	Due date
🧑‍🎓 Membuat RPP untuk MK Algoritma dan Pemrograman	Urgent	Important	<input type="checkbox"/>	August 18, 2023
🧑‍🎓 RPS MK Sistem Informasi Manajemen	Urgent	Important	<input type="checkbox"/>	August 18, 2023
🧑‍🎓 Cek BKD semester genap 2022/2023	Urgent	Important	<input type="checkbox"/>	August 31, 2023

Kita dapat membuat prioritas dan batas waktu dari setiap tugas.





# Tugas di Kelas

- Silakan mencari 5 repository **open source** dengan jumlah **contributor, Star, dan commit** paling banyak!
- Buatlah **sebuah aktifitas sehari-hari** menggunakan kanban board **selama 4 minggu** (2 minggu ke belakang dan 2 minggu ke depan), minimal struktur kanban terdiri dari **To Do, In Progress, dan Done!**

