Mobile Programming (MP)

Pertemuan 13 : Dokumentasi

Alauddin Maulana Hirzan

Fakultas Teknologi Informasi dan Komunikasi
Universitas Semarang

Outline

Dokumentasi

Dokumentasi adalah Kumpulan informasi tertulis yang menjelaskan:

1. Tujuan sistem
2. Arsitektur
3. Implementasi
4. Cara penggunaan
5. Cara pemeliharaan

Dokumentasi

Mengapa Dokumentasi Penting?

Fakta Industri

1. Developer dapat berganti
2. Tim berkembang
3. Aplikasi terus berubah
4. Maintenance berlangsung bertahun-tahun

Dokumentasi

Dampak Tidak Ada Dokumentasi

Risiko

- Sulit maintenance
- Sulit debugging
- Sulit onboarding anggota baru
- Knowledge hanya dimiliki individu tertentu

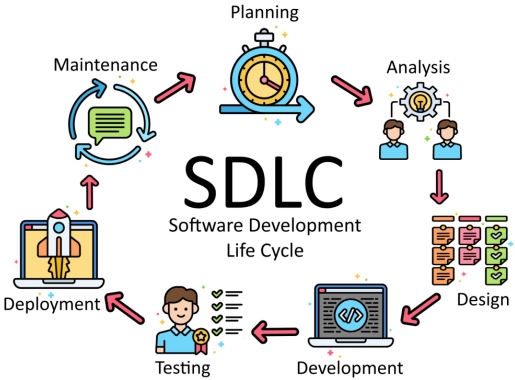
Dokumentasi

Dokumentasi dalam SDLC

Posisi Dokumentasi

- Requirement
- Design
- Development
- Testing
- Deployment
- Maintenance

Dokumentasi



Dokumentasi

Jenis Dokumentasi

Kategori Utama

1. Dokumentasi Kebutuhan
2. Dokumentasi Desain
3. Dokumentasi Kode
4. Dokumentasi API
5. Dokumentasi Pengguna
6. Dokumentasi Pengujian

Dokumentasi

Dokumentasi Kebutuhan

Requirement Documentation

Berisi:

- Tujuan aplikasi
- Fitur
- Batasan sistem
- Kebutuhan pengguna

Dokumentasi

Functional Requirement

Kebutuhan Fungsional

Contoh:

- Login
- Registrasi
- Upload foto
- Notifikasi

Dokumentasi

Non Functional Requirement

Kebutuhan Non-Fungsional

Contoh:

- Performa
- Keamanan
- Availability
- Scalability

Dokumentasi

Use Case Documentation

Tujuan : Menggambarkan interaksi pengguna dengan sistem.

Komponen:

- Actor
- Use Case
- System

Dokumentasi

User Story

Format

- Sebagai [pengguna]
- Saya ingin [fitur]
- Agar [manfaat]

Dokumentasi

Dokumentasi Desain Sistem

Isi

- Arsitektur
- Komponen
- Database
- Alur data

Dokumentasi Desain

Berisikan

1. Wireframe : Sketsa tampilan aplikasi.
2. Mockup dan Prototype:
   - Wireframe -> Struktur
   - Mockup -> Tampilan visual
   - Prototype -> Simulasi interaksi
3. Activity Diagram : Menggambarkan alur proses aplikasi.
4. Sequence Diagram : Menjelaskan komunikasi antar objek.

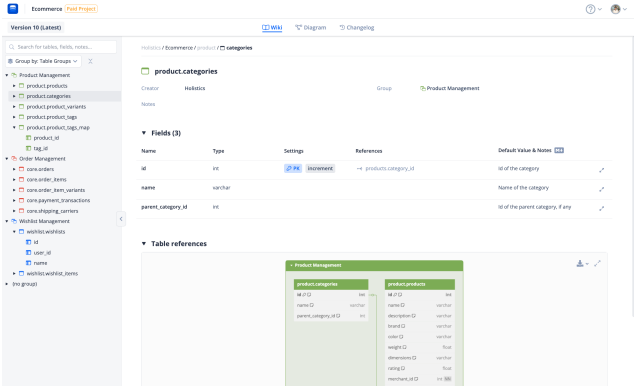
Dokumentasi

Dokumentasi Database

Berisi

- Entity
- Relasi
- Tipe data
- Constraint

Dokumentasi Database



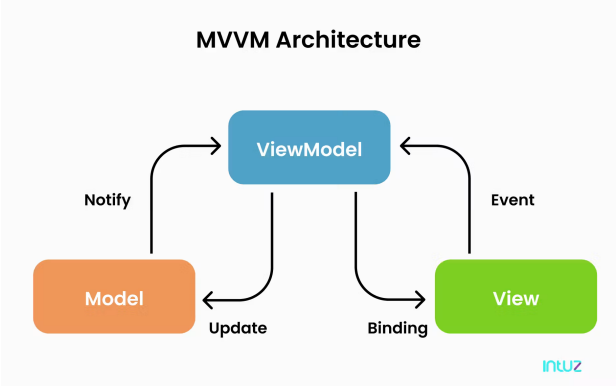
Dokumentasi

Dokumentasi Arsitektur Android

Contoh Modern Android

- UI Layer
- ViewModel
- Repository
- Data Source

Dokumentasi Arsitektur



Dokumentasi

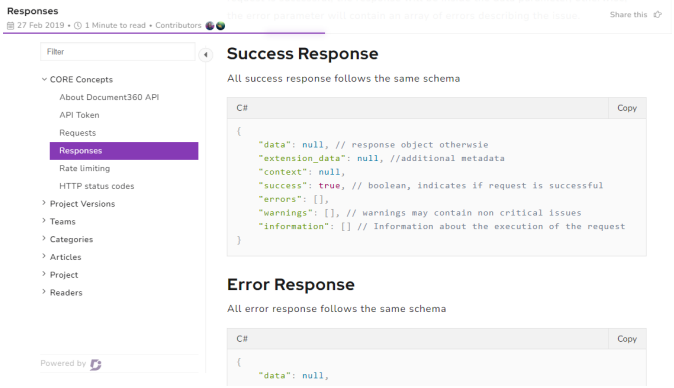
Dokumentasi Source Code

Tujuan : Membantu memahami kode.

Berisi:

- Komentar
- Penjelasan fungsi
- Struktur proyek

Dokumentasi Source Code



README Project

Dokumen pertama yang dibaca developer.

Isi:

- Nama proyek
- Deskripsi
- Instalasi
- Dependensi
- Cara menjalankan

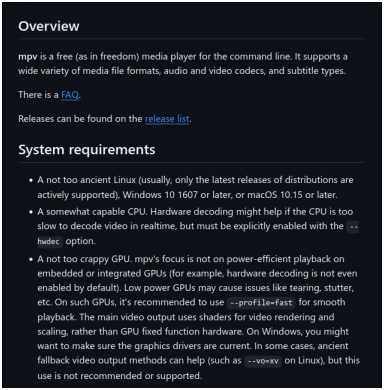
Contoh Readme


```
# Mobile Attendance

## Features
- Login
- QR Attendance

## Installation
1. Clone repository
2. Open Android Studio
3. Sync Gradle
```

Ilustrasi



Overview

mpv is a free (as in freedom) media player for the command line. It supports a wide variety of media file formats, audio and video codecs, and subtitle types.

There is a FAQ.

Releases can be found on the release list.

System requirements

- A not too ancient Linux (usually, only the latest releases of distributions are actively supported), Windows 10 1607 or later, or macOS 10.15 or later.
- A somewhat capable CPU. Hardware decoding might help if the CPU is too slow to decode video in realtime, but must be explicitly enabled with the `--hwdec` option.
- A not too crappy GPU. mpv's focus is not on power-efficient playback on embedded or integrated GPUs (for example, hardware decoding is not even enabled by default). Low power GPUs may cause issues like tearing, stutter, etc. On such GPUs, it's recommended to use `--profile=fast` for smooth playback. The main video output uses shaders for video rendering and scaling, rather than GPU fixed function hardware. On Windows, you might want to make sure the graphics drivers are current. In some cases, ancient fallback video output methods can help (such as `--vo=xv` on Linux), but this use is not recommended or supported.

Lastly

Terima Kasih