UNIVERSITAS
ATMA JAYA YOGYAKARTA
serviens in lumine veritatis

Chapter 4

MVVM Concept, Data Binding, View Binding, and Shared Preferenced

PBP Android Kotlin

Design Pattern

Saat Banyak Software Engineer dalam memilih pengembangan, harus memikirkan bagaimana design arsitektur pada software yang dikembang. Ada 3 macam Design Pattern yang sampai sekarang digunakan pada Developer, yaitu MVC, MVP, dan MVVM.

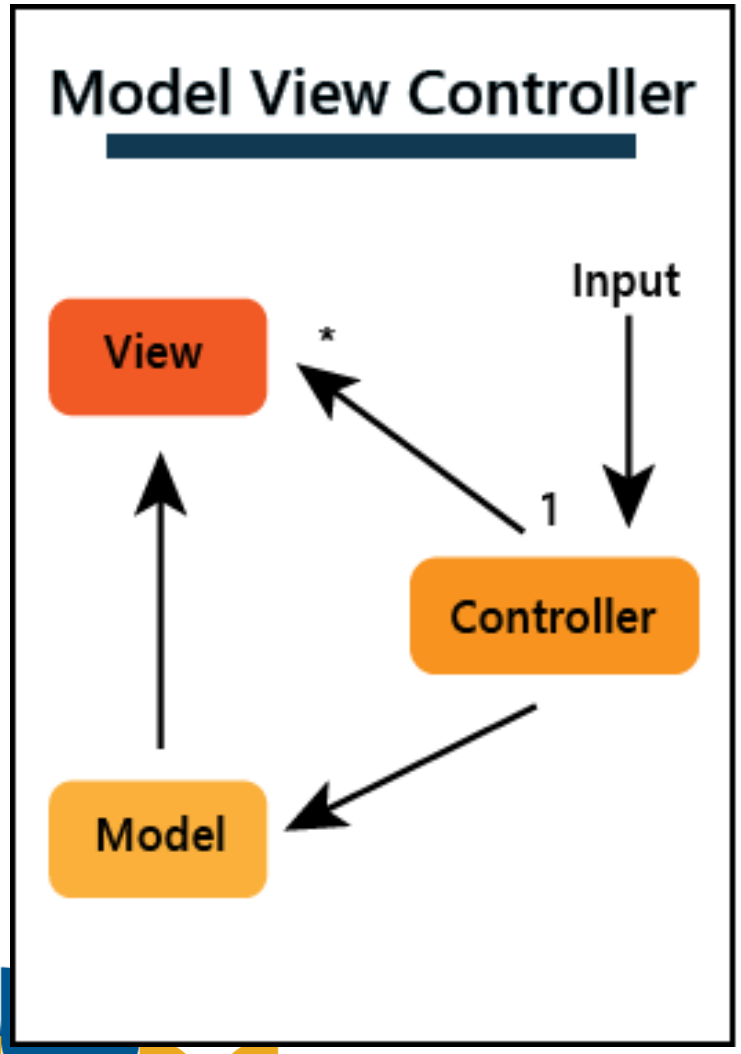
MVC

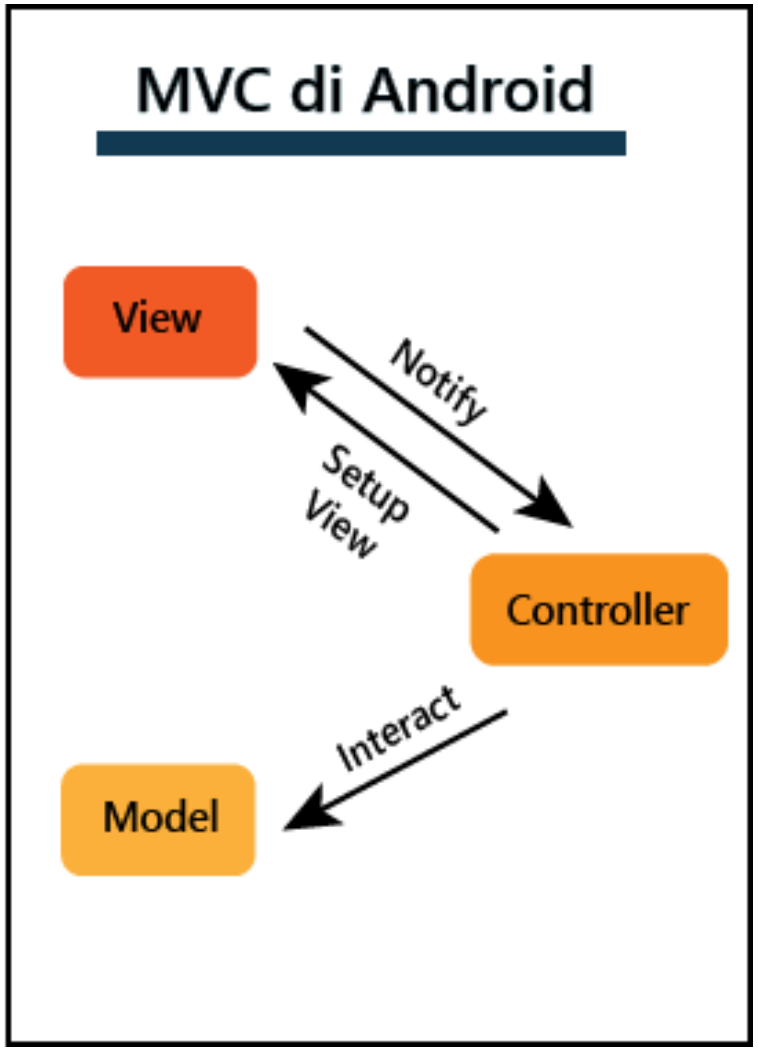
(Model View Controller)

Model : Logika bisnis dan status data yang ada pada aplikasi. Tugas utamanya untuk CRUDS data, berkomunikasi dengan controller dan database, update tampilan app

View : Antarmuka seperti HTML/CSS/XML. View berkomunikasi dengan model yang bertugas untuk tampilan *(front-end)* dan menyajikan data.

Controller : jembatan antara view dengan model. Komponen ini membutuhkan layanan inputan dari view lalu mengirimkan ke model dengan metode REST API. (Get dan Post)





Kekurangan : QA sulit karena memastikan REST API bekerja dengan baik, selain itu, terhalang dengan tergantungnya controller. Selain itu, kompleks dan besar. *Not suitable with Cross Platform required*

Software Developer Web, such as Laravel, ExpressJS, NodeJS still using MVC Concept cause have controlling biggest size database with 1 Brigde Comunication API

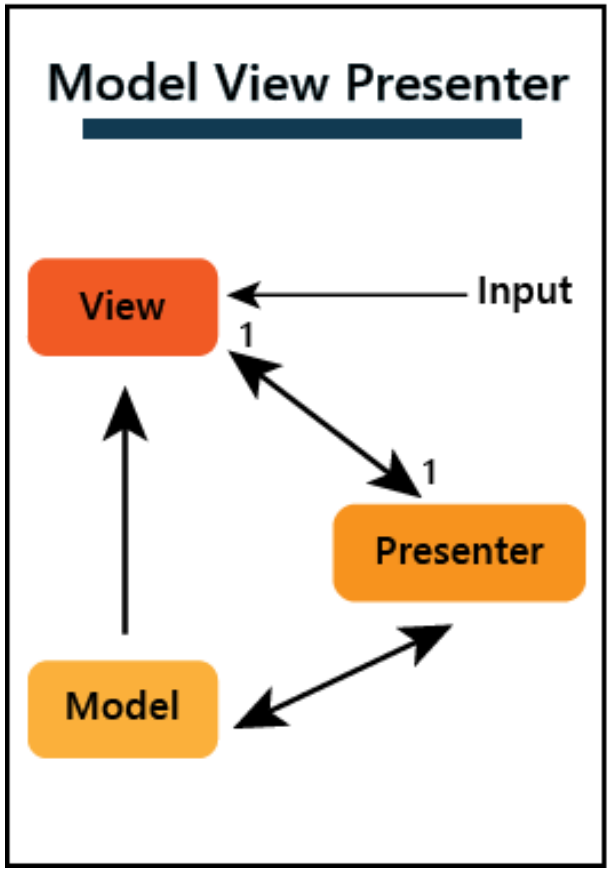
MVP

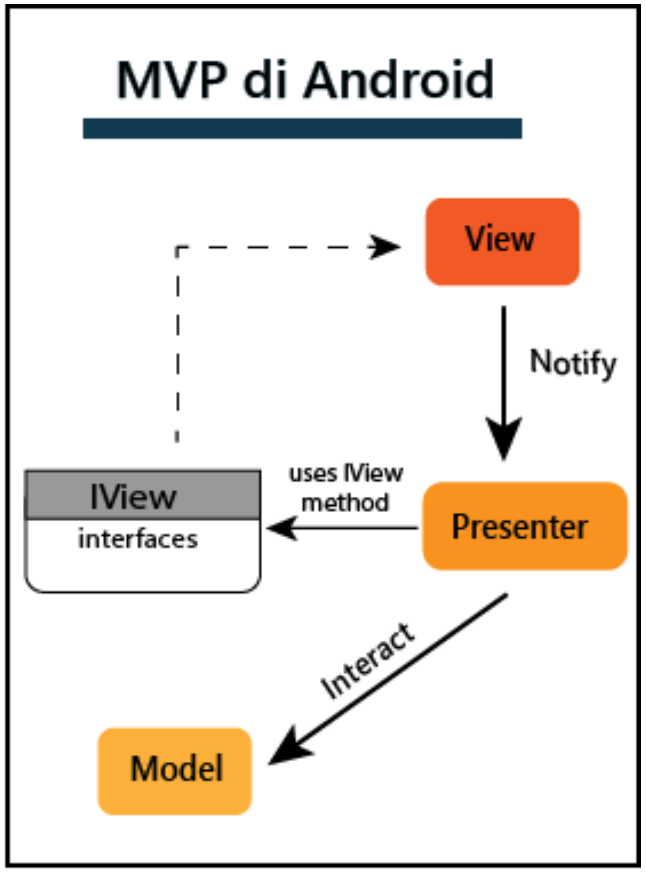
(Model View Presenter)

Model : sekumpulan kelas yang menjelaskan logika bisnis dan data. Tugasnya adalah untuk CRUDS data, namun model bisa berkomunikasi dengan presenter dan model tidak bisa langsung berkomunikasi dengan View

View : interaksi dengan XML (UI), aktivitas, dan fragment dengan Presenter. Hal-hal ini tidak berhubungan dengan logika implementasi dalam proses controlling.

Presenter : Presenter menyajikan data dari model dan mengontrol semua perilaku yang ingin ditampilkan dari aplikasi yang mendorong dan memberitahu View apa yang harus dilakukan. Interaksi yang dilakukan antara model dan view ditangani oleh presenter. Tidak hanya itu presenter juga bertugas untuk menyimpan data kembali ke Model.





Permasalahan yang sering terjadi jika MVP diimplementasikan adalah

1. Size untuk kode terlalu besar
2. Banyak menggunakan interface pada Presenter
3. Karena sangat bergantung dengan View dengan presenter jadi jika ada bug, harus dimulai dengan interface dulu.

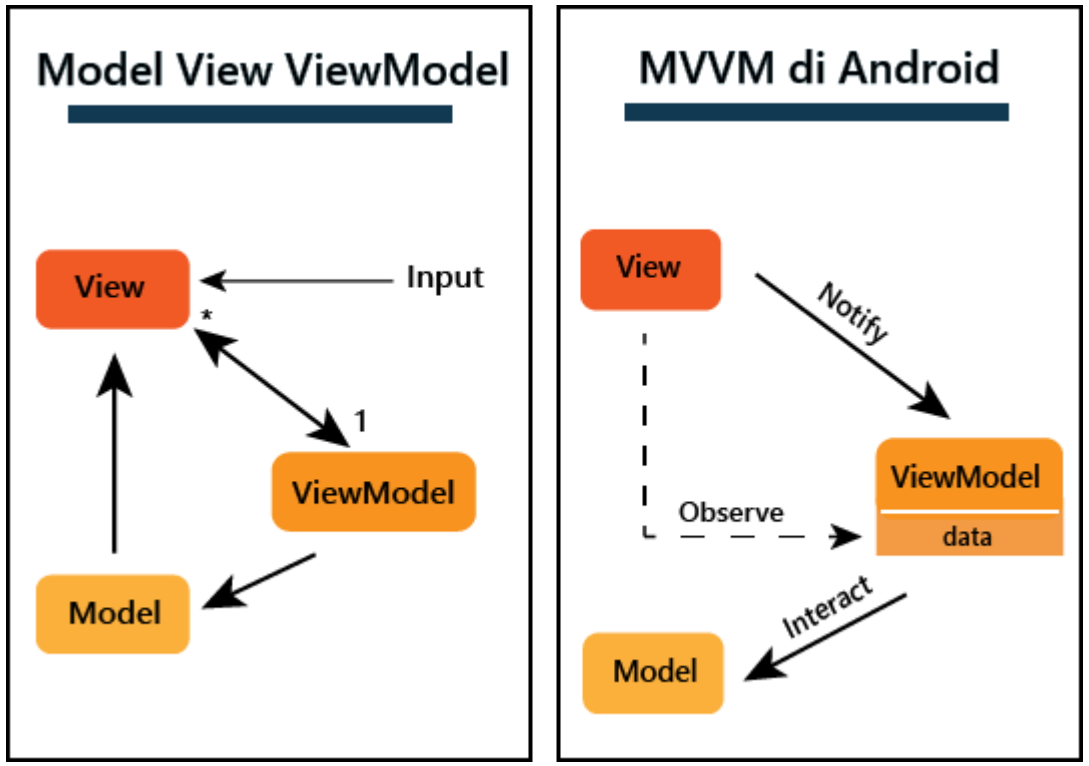
Contoh Bahasa pemrograman yang memakai MVP adalah :

Python (Django) dan Rust a Rails

MVVM

(Model-View-ViewModel)

- Model : Model yang digunakan untuk MVVM mirip dengan model yang digunakan MVC, dimana model tersebut terdiri dari data dasar yang digunakan untuk menjalankan perangkat lunak.
- View : View digunakan sebagai antarmuka grafis antara pengguna dan pola desain, serta menampilkan output dari data yang telah diproses. View yang digunakan MVVM mirip dengan View yang digunakan dalam MVC.
- ViewModel : ViewModel di satu sisi adalah abstraksi dari View, lalu di sisi yang lain, sebagai penyedia pembungkus data model untuk ditautkan. ViewModel terdiri dair Model yang diubah menjadi View, dan berisi perintah yang dapat digunakan oleh View untuk mempengaruhi Model.



Hanya 1 masalah yang terjadi jika memakai MVVM yaitu :

Pengembang harus membuat suatu kuantitas yang dapat diukur di setiap komponen UI. (Observable)

Karena keuntungan dari itu banyak Mobile developer menggunakan **MVVM Design Pattern** dalam pengerjaan

Komparasi MVC vs MVP vs MVVM

Nama Design Pattern	Peforma	Kesesuaian	Easy Modification	Referensi	Titik Point	Developer Experience
MVC						
MVP						
MVVM						

Ket :

Mudah

Cukup

Sulit

Terima kasih

uajy Universitas Atma Jaya Yogyakarta www.uajy.ac.id