

## Perbandingan Kinerja Pola Perancangan MVC, MVP, dan MVVM Pada Aplikasi Berbasis Android (Studi kasus : Aplikasi Laporan Hasil Belajar Siswa SMA BSS)

Bahrur Rizki Putra Surya<sup>1</sup>, Agi Putra Kharisma<sup>2</sup>, Novanto Yudistira<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>bahrurrizki@student.ub.ac.id, <sup>2</sup>agi@ub.ac.id, <sup>3</sup>yudistira@ub.ac.id

### Abstrak

Sistem pelaporan hasil belajar merupakan sebuah sistem yang mengatur pelaporan hasil belajar siswa di SMAS Brawijaya Smart School. Sistem ini dapat menampilkan nilai dari siswa dalam bentuk aplikasi android. Kebutuhan akan keefesienan pada sebuah aplikasi android sangat diperlukan untuk mencapai kepuasan dari pengguna, dan kebutuhan akan pengetahuan tentang pola perancangan yang dibutuhkan untuk membangun atau mengembangkan sebuah aplikasi android. Ada beberapa pola perancangan yang digunakan untuk membangun atau mengembangkan sebuah aplikasi, diantaranya adalah Model View Controller, Model View Presenter, dan Model View ViewModel. Pola perancangan tersebut akan diterapkan pada aplikasi laporan hasil belajar untuk membandingkan pola perancangan mana paling efisien untuk aplikasi laporan hasil belajar tersebut. Tahapan awal untuk penelitian ini adalah rekayasa kebutuhan, pada tahap ini memberikan hasil 4 (empat) kebutuhan fungsional dan 2 (dua) kebutuhan non-fungsional, kebutuhan tersebut digunakan sebagai dasar dalam melakukan perancangan dan implementasi. Implementasi dilakukan dengan menerapkan pola perancangan Model View Controller, Model View Presenter, dan Model View ViewModel dengan bahasa pemrograman java. Selanjutnya setelah dilakukan tahap implementasi akan dilakukan tahap pengujian pada masing-masing aplikasi yang telah menerapkan pola perancangan Model View Controller, Model View Presenter, dan Model View ViewModel sebanyak 5 (lima) kali dengan hasil yang sama pada penggunaan energi yaitu medium dan penggunaan memori rata-rata sebesar 59,7 MB pada MVC, 59 MB pada MVP dan 73,2 MB untuk MVVM. Tahap pengujian fungsional dengan metode *blackbox testing* memberikan hasil validitas sebesar 100% di semua fungsi.

**Kata kunci:** laporan hasil belajar, pola perancangan, Model View Controller, Model View Presenter, Model View ViewModel, Android

### Abstract

*Laporan hasil belajar is a system that regulates student learning outcomes reporting at Brawijaya Smart School. This system can display grades from students in android application form. The need for efficiency in an android application is very necessary to achieve user satisfaction, and the need for knowledge of the design patterns needed to build or develop an android application. There are several design patterns that are used to build or develop an application, including Model View Controller, Model View Presenter, and Model View View Model. The design pattern will be applied to the laporan hasil belajar application to compare which design patterns are most efficient for laporan hasil belajar application. The initial stage for this research is engineering needs, at this stage gives the results of 4 (four) functional needs and 2 (two) non-functional needs, these needs are used as a basis for designing and implementing. The implementation is carried out by applying the design pattern of Model View Controller, Model View Presenter, and Model View View Model with the java programming language. Furthermore, after the implementation phase, the testing phase will be carried out on each application that has applied the design pattern of the Model View Controller, Model View Presenter, and Model View View Model 5 (five) times with the same results on energy use medium and memory usage*

average of 59.7MB for MVC, 59MB for MVP and 73.2MB for MVVM. The functional testing by using the blackbox testing method gives 100% validity in all functions.

**Keywords:** laporan hasil belajar, design pattern, Model View Controller, Model View Presenter, Model View ViewModel, Android

## 1. PENDAHULUAN

SMAS BSS atau Brawijaya Smart School merupakan Sekolah Menengah Atas Swasta dibawah nanungan dari Universitas Brawijaya Malang. Sekolah ini sudah banyak menerapkan penggunaan teknologi guna mengejar target sebagai Sekolah Bertaraf Internasional (SBI).

Memasuki era globalisasi seperti saat ini, membuat banyak orang untuk menggunakan sesuatu dengan memakai media elektronik, dengan alasan utamanya adalah dapat membantu dan meningkatkan kinerja. Dengan kemajuan teknologi, perangkat mobile menjadi sering digunakan secara luas sebagai kenyamanan perangkat komputasi. Untuk mengembangkan aplikasi berkualitas tinggi dalam waktu singkat, pengetahuan yang kuat dan pemahaman tentang pola perancangan adalah faktor utama dalam mengembangkan aplikasi yang dirancang dengan baik. Desain yang baik memastikan bahwa crash pada aplikasi dapat dicegah juga sebagai peningkatan kepuasan pengguna. Selain itu penggunaan pola perancangan akan mempermudah pengembangan aplikasi untuk ke depannya (Fadilah Ezlina Shahbudin dan Fang Fang Chua (2013)).

Tujuh dari sepuluh developer menemui masalah dalam mengembangkan atau membangun aplikasi, dan permasalahan tersebut dapat diselesaikan dengan menerapkan *design pattern* atau pola perancangan. Penggunaan *design pattern* ditentukan dengan permasalahan yang ingin diatasi ataupun dihindari (Tyagita Larasati (2018)), ada beberapa pola perancangan yang dapat diterapkan dalam menerapkan aplikasi, yaitu dengan menerapkan pola perancangan *Model-View-Controller (MVC)*, *Model-View-Presenter (MVP)* dan *Model-View-ViewModel (MVVM)*. Dimana setiap pola perancangan memiliki kelebihan dan kekurangan masing-masing.

*MVC* memiliki kelebihan yaitu : (1) *Bussines logic* terpisah dengan model, (2) Modifikasi yang dilakukan tidak mempengaruhi komponen lainyang tidak diubah, (3) Proses pengembangan yang lebih cepat, sedangkan kekurangannya adalah controller bersi proses yang kelewat besar sehingga sulit dikelola dan kompleksitas aplikasi akan terus meningkat. *MVP* memiliki kelebihan yaitu : (1) Komponen View dan presenter bisa direuse, (2) Kode lebih mudah dipahami dan dikelola, sedangkan kekurangannya adalah ukuran kode bisa menjadi lebih besar dan antara view dan presenter memiliki hubungan terlalu erat. *MVVM* memiliki kelebihan yaitu : (1) Tidak ada interface antara viewa dan model, (2) Tidak ada hubungan erat dantar view dan viewmodel, sedangkan kekurangannya adalah kode yang ditulis bisa banyak (Axella Adryahmarthanino, 2018).

Dengan memanfaatkan keunggulan dan mempertimbangkan kekurangannya yang kemudian diterapkan pada aplikasi laporan hasil belajar yang sudah dikembangkan pada saat praktik kerja lapangan dengan judul “Pengembangan Aplikasi Raport Berbasis Android DI SMA Brawijaya Smart School” diharapkan dapat menemukan pola perancangan mana yang lebih efisien diteraokan pada aplikasi tersebut sehingga dapat meningkatkan efisiensi pada aplikasi tersebut. Sehingga untuk menyelesaikan masalah tersebut, penulis melakukan penelitian dengan judul “Perbandingan Kinerja Pola Perancangan MVC, MVP, dan MVVM Pada Aplikasi Berbasis Android.

## 2. KAJIAN PUSTAKA

Pada penelitan sebelumnya, Fadilah Ezlina Shahbudin dan Fang Fang Chua dalam tugas papernya yang berjudul “Design Pattern For Developing High Efficiency Mobile Aplication”. Pada hasil penelitian tersebut menjelaskan bagaimana penerapan design pattern dapat menigkatkan kinerja aplikasi, dan

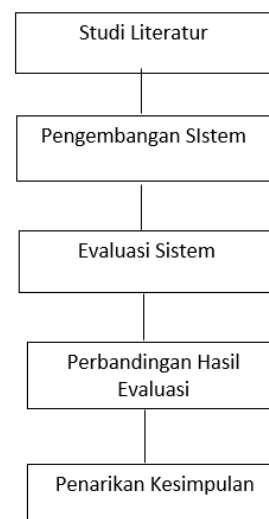
desing pattern yang digunakan dalam penelitiina tersebut adalah MVC.

Pola perancangan merupakan sebuah template kodingan yang digunakan oleh developer saat menemui permasalahan dalam membangun sistem yang dibangunnya (Erich Gamma, 1995). Pola perancangan , ibarat sebuah konsep atau template , yang dapat kita terapkan dan digunakan ke dalam bahasa pemrograman.

Menurut Axella Adryahmarthanino (2018) *Model View Controller* (MVC) adalah sebuah pola perancangan untuk membuat sebuah aplikasi dengan memisahkan antara data, view dan cara proses. Model yang merupakan data yang digunakan oleh aplikasi sebagaimana proses bisnis yang digunakan terhadapnya. *Model* mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain. *View*, Kelas yang berguna untuk menampilkan *user interface*. *MVP* atau *Model View Presenter* adalah sebuah konsep dimana kita memisahkan antara data, proses dan tampilan, jadi tugas utama dari *presenter* adalah sebagai penyedia data dari *model*, selain sebagai penyedia data, *presenter* juga bisa menyimpan method yang dgunakan dalam aplikasi yang biasa kita simpan di *view* dan menerima action yang dikirim dari layer *view* (Axella Adryahmarthanino, 2018). *Model View ViewModel* merupakan sebuah pola pengembangan aplikasi yang berasal dari presentasi *Model* (Martin Fowler, 2005). MVVM biasa digunakan untuk membangun user interface pada *class xml* dan bisa memilih bagian mana yang harus ditampilkan.

### 3. METODOLOGI

Tahapan dalam penelitian ini dimulai dilakukannya studi literatur, pengembangan sistem, evaluasi sistem, perandingan hasil evaluasi, dan penarikan kesimpulan. Gambar 1 adalah gambaran diagram alir dalam penelitian “Perbandingan Kinerja Pola Perancangan MVC, MVP, dan MVVM Pada Aplikasi Berbasis Android”.



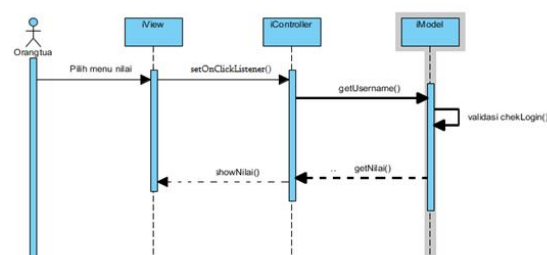
Gambar 1 Metodologi Penelitian

### 4. PERANCANGAN

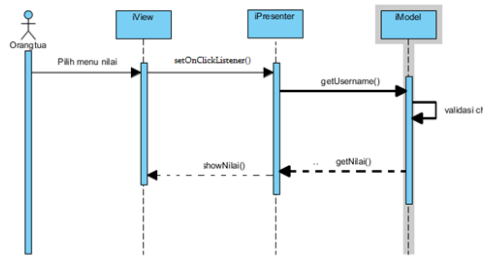
Perancangan terdiri dari beberapa tahapan, yaitu perancangan sistem yang terdiri dari perancangan sequence diagram, class diagram, dan perancangan user interface dalam bentuk mookup.

#### 4.1 Perancangan Arsitektur Sistem

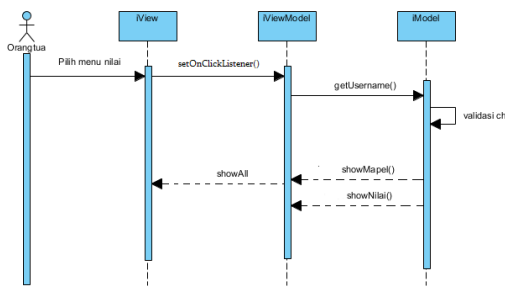
Perancangan arsitektur dengan menggambarkan sequence diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan output tertentu, Gambar 2 sampai dengan Gambar 4 merupakan sequence diagram dengan menerapkan pola perancangan MVC, MVP, dan MVVM



Gambar 2 Arsitektur Dengan Menerapkan MVC



Gambar 3 Arsitektur Dengan Menerapkan MVP



Gambar 4 Arsitektur Dengan Menerapkan MVVM

#### 4.2 Perancangan Antarmuka

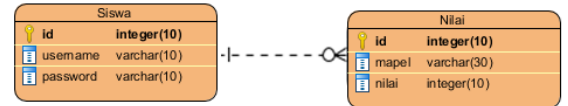
Antarmuka merupakan sebuah halaman yang menghubungkan sistem dengan pengguna. Antarmuka akan dimodelkan dalam bentuk mockup.



Gambar 5 Mockup Halaman Laporan

#### 4.3 Perancangan Database

Perancangan ini akan menghasilkan struktur basisdata yang nantinya akan digunakan menyimpan data sistem.



Gambar 6 Conceptual Data Model

#### 5. IMPLEMENTASI

Penerapan atau implementasi merupakan tahapan proses pembuatan perangkat lunak dari hasil perancangan yang telah dilakukan. Penerapannya harus sesuai dengan rancangan yang telah dilakukan sebelumnya. Penerapan terdiri dari beberapa tahap, yaitu, ipenerapan kode program, penerapan antarmuka, dan penerapan database.

##### 1. Implementasi Kode Program

Implementasi method tiap pola perancangan memiliki fungsi yang sama saat method tersebut dipanggil namun penempatan yang berbeda tiap class sesuai dengan arsitektur yang telah dirancang sebelumnya.

##### 2. Implementasi Antarmuka

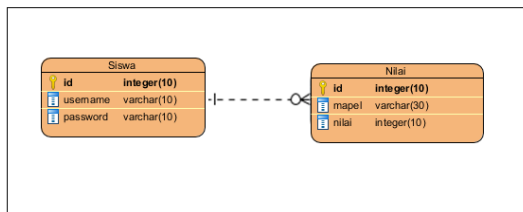
Mengimplementasikan antarmuka yang telah dirancang dalam bentuk mockup pada tahap perancangan.



Gambar 7 Implementasi Antarmuka Halaman Laporan

### 3. Implementasi Database

Mengimplementasikan database yang telah dirancang pada tahap sebelumnya.



Gambar 8 Implementasi Database

### 6. PENGUJIAN DAN EVALUASI

Pengujian adalah melakukan uji terhadap aplikasi yang telah dikembangkan. Validasi dilakukan pada kebutuhan fungsional dan non-fungsional dengan menggunakan metode *black box testing*. Dan evaluasi dilakukan dengan membandingkan hasil uji pada setiap aplikasi tanpa pola perancangan dan yang telah menerapkan pola perancangan MVC, MVP, dan MVVM.

Tabel 1 Hasil Pengujian Fungsional Sistem

No	Test Name	Test Case	Expected Result	Result	Status
1	Pengujian login aplikasi	Memasukkan username dan password pada textfield yang tersedia kemudian menekan tombol login	Masuk ke dalam sistem aplikasi dan menampilkan menu about	Masuk ke dalam sistem aplikasi dan menampilkan menu about	Valid
2	Pengujian lihat nilai	Pada saat user sudah login user menekan tombol menu nilai	Menampilkan nilai user	Menampilkan nilai user	Valid
3	Pengujian halaman about	Pada saat user sudah login user	Menampilkan halaman aplikasi	Menampilkan halaman	Valid

		menekan tombol menu about	about aplikasi	
4	Pengujian logout	Pada saat user sudah login user menekan tombol logout	User logout dari aplikasi user login	User logout dari aplikasi Valid

Tabel 2 merupakan hasil dari pengujian fungsional sistem dengan 2 jenis pengujian, yaitu pengujian performance yang akan menguji waktu eksekusi aplikasi dan uji compatibility yang akan menguji sistem berjalan pada android oreo.

Tabel 2 Pengujian Non Fungsional

No	Jenis uji	Expectet Result	Test Result
1	Uji Performance	Sistem akan menyelesaikan proses eksekusi dalam waktu 5 detik	Sistem akan menyelesaikan proses eksekusi dalam waktu kurang dari 5 detik
2	Uji Compatibility	Sistem dapat dijalankan pada sistem Operasi Android Oreo	Sistem dapat dijalankan pada sistem Operasi Android Oreo.

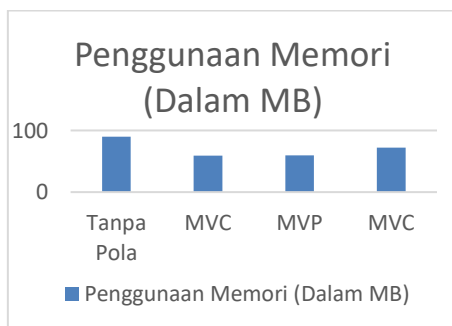
Tabel 3 merupakan hasil perbandingan penggunaan energi pada aplikasi tanpa pola perancangan dan aplikasi yang menerapkan pola perancangan

Tabel 3 Perbandingan Penggunaan Energi dan Memori

Pola Perancangan	Penggunaan Energi	Penggunaan Memori
Tanpa pola	Heavy	92 MB
MVC	Medium	59,7 MB
MVP	Medium	59 MB
MVVM	Medium	72,9

Gambar 9 merupakan grafik perbandingan penggunaan memori yang digunakan aplikasi (dalam megabyte) terlihat penggunaan memori yang digunakan oleh smartphone pada aplikasi tanpa menerapkan pola perancangan lebih

besar dibandingkan dengan aplikasi yang menerapkan pola perancangan, hal tersebut dikarenakan banyaknya jumlah *source code* aplikasi tanpa menerapkan pola perancangan sehingga hal tersebut dapat memperbesar jumlah penggunaan memori yang digunakan, sedangkan jika menggunakan pola perancangan jumlah penggunaan memori lebih sedikit karena jumlah *source code* lebih sedikit setelah memisahkan antar *business logic* dengan *model*.



Gambar 9 Perbandingan Penggunaan Memori

## 7. KESIMPULAN

Perbedaan performa aplikasi tanpa pola perancangan dengan menerapkan pola perancangan berdasarkan penelitian ini terlihat dari penggunaan energi *smartphone* dan memori yang dipakai yang tercantum pada *android profiler* pada saat tahap pengujian, dimana status penggunaan energi pada *smartphone* tanpa pola perancangan adalah *heavy* sedangkan dengan menerapkan pola perancangan adalah *medium*. Untuk memori yang digunakan pada aplikasi tanpa pola perancangan sebesar 92 MB sedangkan dengan pola perancangan sebesar 59,7 MB pada MVC, 59 MB pada MVP dan 73,2 MB untuk MVVM.

Dalam penelitian ini menerapkan tiga pola perancangan yaitu MVC, MVP, dan MVVM. Untuk hasil pengujian aplikasi dengan menerapkan pola perancangan tersebut mendapatkan hasil yang sama pada penggunaan energi pada *smartphone* yaitu *medium*. Sedangkan memori yang digunakan mendapatkan hasil rata-rata sebesar 59,7

MB pada MVC, 59 MB pada MVP dan 73,2 MB untuk MVVM.

Setelah mendapatkan hasil pengujian dari setiap pola perancangan yang diterapkan pada aplikasi diperoleh hasil pola perancangan MVC dan MVP memiliki performa yang sama. Maka dari itu pola perancangan paling optimal untuk diterapkan pada aplikasi laporan hasil belajar ini adalah pola perancangan MVC dan MVP.

## 8. DAFTAR PUSTAKA

Shabudin, F dan Chua, F., 2013. *Design Pattern For Developing High Efficiency Application*. Tersedia di: <https://www.longdom.org/open-access/design-patterns-for-developing-high-efficiency-mobile-application-2165-7866.1000122.pdf>.

Fakhroutdinov, K., 2009. *The Unified Modeling Language*. Tersedia di: <https://www.uml-diagrams.org>.

Fakhroutdinov, K., 2009. *UML Sequence Diagrams*. Tersedia di: <https://www.uml-diagrams.org/sequence-diagrams.html>

Liu, Z., 2014. *Research on Software Security and Compatibility Test for Mobile Application*. Tersedia di: <https://ieeexplore.ieee.org/document/6927764>.

Ojeda-Guerra, C.N., 2015. *A Simple Software Development Methodology Based on MVP for Android Application*. Tersedia di: <https://ieeexplore.ieee.org/document/4363258>.

Luthfiansyah, Arif. 2019. "Pengembangan Aplikasi Pemantauan Alat Berat Pertambangan Menggunakan Teknologi Geofencing Dengan Arsitektur MVP".

- Skripsi. FILKOM, Teknik Informatika, Universitas Brawijaya Malang.
- Sommerville, Ian. 2011. *Software engineering*. USA: Addison-Wesley.
- Android Developer, 2019. *Measure Application Performance with Andoid Profiler*. Tersedia di : <https://developer.android.com/studio/profile/android-profiler>.
- Sommerville, I., 2011. *Software Engineering 9<sup>th</sup> Edition*. USA: Pearsin Educaion,Inc.
- Kurniawan, T.A., 2018. Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi dan Ilmu Komputer*. Tersedia di : <http://jtiik.ub.ac.id/index.php/jtiik/article/view/610>
- Fakhroutdinov, K., 2009. *UML Use Case Diagrams* . Tersedia di: <https://www.uml-diagrams.org/usecase-diagrams.html>.
- Software Testing Fundamentals, 2019. *Black Box Testing*. Tersedia di : <http://softwaretestingfundamentals.co/black-box-testing/>
- Harini, 2004. Sistem Informasi Nilai Berbasis Web (Studi Kasus : SMPN 1 Semarang).
- Puji, 2004. Pengembangan Sistem Informasi Laporan Prestasi Belajar Siswa Berbasis Web (Studi Kasus : SMAN 8 Bogor).
- Rasdiana, E., .2014. Analisa Pengembangan Sistem Informasi Student Information Services (SIS) Pada Perguruan Tinggi Raharja
- Neyfa, B.C., dan D. Tamara. 2016. "E-Canteen Android Based Application Design Using Object Oriented Analisis & Design Method." *Jurnal Penelitian Komunikasi dan Opini Publik* 20: 83-91.