PEMANFAATAN JSON UNTUK MENAMPILKAN DATA *REALTIME* COVID-19 DENGAN *MODEL VIEW PRESENTER*

Rysa Sahrial[1)], Deri Fikri Fauzi [2)], Eva Susilawati [3)]
[1.3]*Fakultas Teknik,* [2]*Fakultas Sastra, Universitas Putra Indonesia*
[1.2.3]*Jl. Dr. Muwardi 66, Cianjur*
Email: [1]*risasyahrial@unpi-cianjur.ac.id,* [2]*deri.fikri@unpi-cianjur.ac.id,* [3]*evasusi@unpi-cianjur.ac.id*

Abstract

Corona virus pandemic 19 (codiv-19) has become a global problem that cannot be solved yet. Covid-19 is an infectious disease caused by the SARS-CoV-2 corona virus. The role of information technology continues to try to find solutions in co-19 prevention. The Indonesian government has created an information system (SI) in the form of a website and a mobile application to disseminate information about covid-19, one of the features contained in the SI is to display data on the spread of the area and the number of covid-19 infections that are reatime-infected. The purpose of this study is to utilize JSON data using Model View Presenter (MVP) which has the essence of the interface code and the view class and takes it to a different presenter class, so that the presenter can walk and do the testing process without affecting the processing of view code. The system development method used is Hierarchy plus Input-Process-Output (HIPO) with three level functions (diagrams), namely: Visual Table of Content (VTC), Overview Diagrams (OD) and Detail Diagrams. The usage pf JSON for data rest API, which uses the Model View Presenter method, is very easy to implement even though it uses three different JSON data calling methods.

Keyword: *JavaScript Object Notation, Model View Presenter, Hierarchy plus Input-Process-Output, Android Studio, Java Programming*

Abstrak

Pandemik *corona virus disease 19* (codiv-19) menjadi masalah global yang belum bisa teratasi. Covid-19 adalah penyakit menular yang disebabkan oleh virus korona SARS-CoV-2. Peran teknologi informasi terus berusaha mencari solusi dalam pencegahan covid-19. Pemerintah Indonesia telah membuat sistem informasi (SI) berupa *website* dan aplikasi *mobile* untuk menyebarkan informasi tentang covid-19, salah satu fitur yang terdapat pada SI tersebut adalah menampilkan data penyebaran wilayah dan jumlah yang terinfeksi covid-19 secara *reatime*. Tujuan dari penelitian ini adalah memanfaatkan data JSON menggunakan *Model View Presenter* (MVP) yang memiliki esensi antara kode antarmuka dengan *class view* dan membawanya ke *class presenter* berbeda sehingga *presenter* bisa berjalan dan melakukan proses pengujian tanpa mempengaruhi pengolahan kode *view*. Metode pengembangan sistem yang digunakan adalah *Hierarchy plus Input-Process-Output* (HIPO) dengan tiga fungsi tingkatan (diagram), yaitu: *Visual Table of Content* (VTC), *Overview Diagram* (OD) dan *Detail Diagram.* Penggunaan JSON untuk pertukaran data API *rest* yang menggunakan metode *Model View Presenter* sangat mudah diimplementasikan meskipun menggunakan tiga metode pemanggilan data JSON berbeda.

Kata Kunci: *JavaScript Object Notation, Model View Presenter, Hierarchy plus Input-Process-Output*, Android Studio, Java *Programming*

1. Pendahuluan

Pandemik *corona virus disease 19* (codiv-19) pada pertengahan tahun 2020 masih menjadi masalah global yang belum bisa teratasi. Covid-19 adalah penyakit menular yang disebabkan oleh virus korona SARS-CoV-2[1]. Dengan jumlah korban meninggal 3.710 (4,7%) dari total pasien 78.572 teridentifikasi covid-19 di Indonesia [1].

Selain bidang medis, peran teknologi informasi terus berusaha mencari solusi dalam pencegahan covid-19. Pemerintah Indonesia telah membuat sistem informasi (SI) berupa *website* dan aplikasi *mobile* untuk menyebarkan informasi tentang covid-19, agar masyarakat mendapatkan informasi secara valid dan terhindar dari informasi *hoax*.

Salah satu fitur yang terdapat pada SI tersebut adalah menampilkan data penyebaran wilayah dan jumlah yang terinfeksi covid-19 secara *reatime*. Selain pemerintah Indonesia terdapat juga pengambang SI yang memberikan akses informasi jumlah orang terinfeksi covid-19 berupa data JavaScript Object Notation (JSON). Data JSON yang bisa di akses oleh semua pengembang SI seperti *website* kawal corona yang di buat oleh Ethical Hacker Indonesia.

JSON sebagai media pertukaran data yang bersifat ringan [2] dan menggunakan memori sangat kecil. Data berupa text yang bisa dibaca oleh manusia, yang digunakan untuk mempresentasikan struktur data sederhana dan objek. Selain itu JSON bisa diakses oleh beberapa bahasa pemrograman seperti java.

Dalam penelitian ini, penulis merancang aplikasi informasi penyebaran covid-19 dengan memanfaatkan

data JSON menggunakan *Model View Presenter* (MVP). Perancangan sistem MVP memiliki esensi antara kode antarmuka dengan *class view* dan membawanya ke *class presenter* berbeda. Sehingga *presenter* bisa berjalan dan proses pengetesan tanpa mempengaruhi pengolahan kode *view* [3].

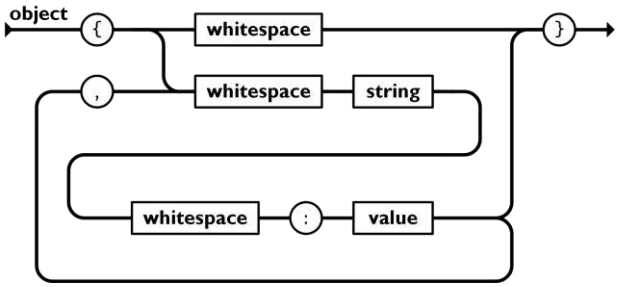
2. Landasan Teori

JavaScript Object Notation (JSON)

JSON merupakan format pertukaran data ringan yang mudah untuk dibaca dan ditulis oleh manusia, serta mudah di terjemahkan (*parse*) dan dibuat (*generate*) oleh komputer[4]. Penulisan format JSON tidak bergantung pada salah satu bahasa pemrograman, sehingga JSON bisa digunakan sebagai bahasa pertukaran data antar bahasa pemrograman.

Struktur data *universal* digunakan pada JSON, meliputi kumpulan pasangan nilai atau objek (*object*) dan daftar nilai terurutkan atau larik (*array*).

Objek merupakan nilai yang tidak terurutkan. Penulisan objek dimulai dengan simbol { dan diakhiri dengan simbol }. Setiap nama diikuti dengan titik dua (:) dan koma (,) untuk memisahkan setiap pasangan nilai .



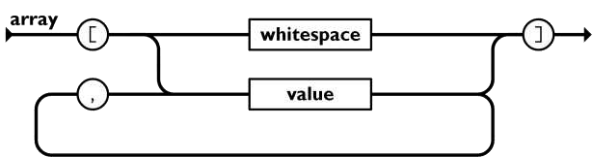
Gambar 1. Struktur penulisan objek pada JSON[4]

Contoh Penulisan struktur objek tersebut dalam kode sebagai berikut:

```
{
    "id" : 62 ,
    "name" : "Indonesia",
}
```

Gambar 2. Struktur penulisan kode objek

Larik merupakan data yang terurutkan dimulai dengan [dan diakhiri dengan]. Serta koma (,) untuk memisahkan setiap nilai.



Gambar 3. Struktur penulisan larik pada JSON[4]

Contoh Penulisan struktur larik tersebut dalam kode sebagai berikut:

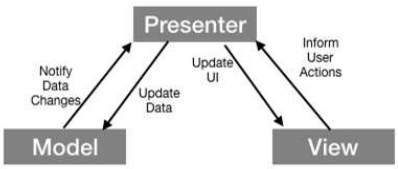
```
[
    {
        "name": "Indonesia",
        "positif": "514",
        "sembuh": "29",
        "meninggal": "48"
    }
]
```

Gambar 4. Struktur penulisan kode larik

Arsitektur *Model View Presenter*(MVP)

MVP merupakan teknik pemrograman yang secara khusus mengarah ke *page event model* [5], seperti bahasa pemrograman java di Android Studio. Pola arsitektur MVP dibuat menjadi 3 (tiga) bagian atau lapisan dasar, yaitu *model*, *view*, dan *presenter*[6].

1. *Model* merupakan *class* yang menunjukan objek dan data pada aplikasi android.
2. *View* merupakan class yang menampilkan data dan interaksi langsung dengan *user*. Pada aplikasi android view berupa *class* activity, *fragment* atau dialog.
3. *Presenter* merupakan *class* yang menghubungkan antara *model* dan *view*. *Prenseter* bertugas memproses dan mengakses *mode*l yang responnya dikembalikan kepada *view*.

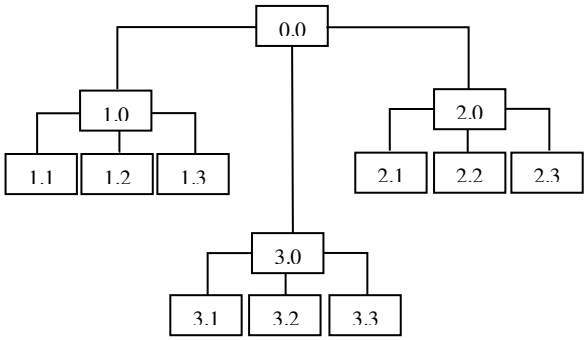


Gambar 5. Arsitektur MVP pada Android[7]

3. *Hierarchy plus Input-Process-Output*

Hierarchy plus Input-Process-Output (HIPO) merupakan alat desain dan teknik dokumentasi dalam siklus pengembangan sistem[8]. HIPO memiliki tiga fungsi tingkatan (diagram) untuk menggambarkan suatu sistem, yaitu:

1. *Visual Table of Content* (VTC) adalah diagram yang menggambarkan hubungan dari setiap fungsi secara berjenjang yang terperinci dan terstruktur. Gambar 6 merupakan struktur VTC penelitian yang dibuat secara hirarki.



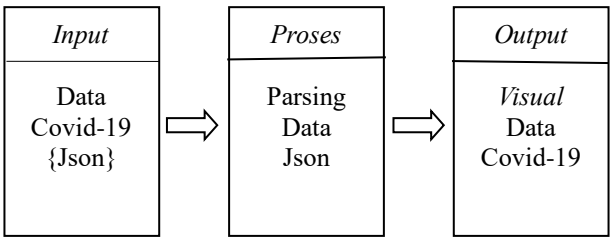
Gambar 6. VTC fungsi penelitian menampilkan JSON covid-19[9]

Dari gambar 6, penulis membagi 9 fungsi untuk menampilkan JSON yang dibagi menjadi 3 metode pengambilan data covid-19. Dua diantaranya menggunakan liblary Http (*volley* dan *retrofit*).

Tabel 1. Deskripsi VTC gambar 6

Fungsi		Keterangan
0.0		Keseluruhan fungsi sistem pengambilan data covid-19 menggunakan JSON
1.0	1.1	Proses pengambilan data JSON negara Indonesia menggunakan HttpHandler
	1.2	Proses pengambilan data JSON negara Indonesia menggunakan *librari volley*
	1.3	Proses pengambilan data JSON negara Indonesia menggunakan *librari retrofit*
2.0	2.1	Proses pengambilan data JSON berdasarkan provinsi dengan format *list item* menggunakan HttpHandler
	2.2	Proses pengambilan data JSON berdasarkan provinsi dengan format *list item* menggunakan *librari volley*
	2.3	Proses pengambilan data JSON berdasarkan provinsi dengan format *list item* menggunakan *librari retrofit*
3.0	3.1	Proses pengambilan data JSON jumlah global menggunakan HttpHandler
	3.2	Proses pengambilan data JSON jumlah global menggunakan *librari volley*
	3.3	Proses pengambilan data JSON jumlah global menggunakan *librari retrofit*

2. *Overview Diagram* (OD) menghubungkan masing-masing diagram dengan salah satu fungsi sistem. Keseluruhan fungsi dimulai dari *input* data yang dipakai oleh *process*. Sedangkan process mengurutkan dan menjelaskan fungsi untuk menghasilkan *output*.



Gambar 7. *Overview Diagram* penelitian menampilkan data JSON covid-19

3. *Detail Diagram* merupakan diagram rinci yang berisi unsur-unsur paket yang menjelaskan semua fungsi, menunjukan item-item *output* dan *input*.

4. Pengujian Sistem

Dalam pengujian sistem, penulis membandingkan alokasi *memory* dan *heap* java dengan *memory profiler* dan *CPU profiler*. Dengan menggunakan tiga metode pemanggilan data JSON, maka diharapkan menjadikan acuan metode mana yang paling ringan dan *realtime* untuk digunakan. Tiga metode tersebut:

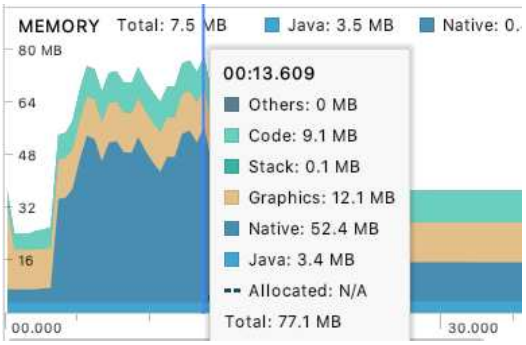
1. HttpURLConnection (HttpHandler), adalah metode yang digunakan untuk menghubungkan / memanggil data dari *Uniform Resource Locator* (URL).
2. *Volley* adalah *library* HTTP yang mempermudah dan mempercepat *networking*[10] pemanggilan data URL. *Volley* menyediakan kelebihan fitur *debug* dan penelusuran dengan dukungan untuk memprioritaskan permintaan data.
3. *Retrofit* adalah *client* HTTP *type-safe* untuk memudahkan menterjemahkan data dari *Web Rest* ke pemrograman Java.

Tabel 2. Pengujian fungsi 1.0 Data Covid19 Indonesia

Fungsi	Metode	JB xml	JB Java
1.1	HttpHandler	219	214
1.2	Volley	219	137
1.3	Retrofit	219	163

Keterangan: JB xml (jumlah baris *code* xml), JB Java (jumlah baris *code* java)

Gambar 8. Tampilan (fungsi 1.0) data covid-19 Negara Indonesia



Gambar 9. *Memory Profile* (fungsi 1.1) pengujian ke 1

data covid-19 negara Indonesia menggunakan HttpHandler.

Pengujian Fungsi 1.0

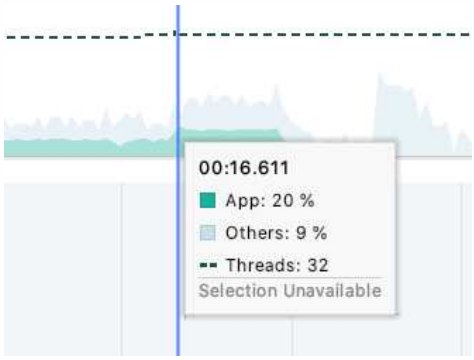
Pengujian fungsi 1.0 adalah pemanggilan JSON covid-19 negara Indonesia menggunakan antar muka yang sama dengan *run* aplikasi sebanyak tiga kali setiap fungsinya. Yang dibedakan dengan tiga metode, didapatkan grafik perbedaan penggunaan *memory* (dalam hitungan *magabyte*) dan jumlah persentase pemrosesan *Central Processing Unit* (CPU) tertinggi pada setiap pengujian.

Rumus penghitungan *memory* pada penelitian:

Memory : *Code* + *Stack* + *Graphics* + *Native* + Java

Keterangan:

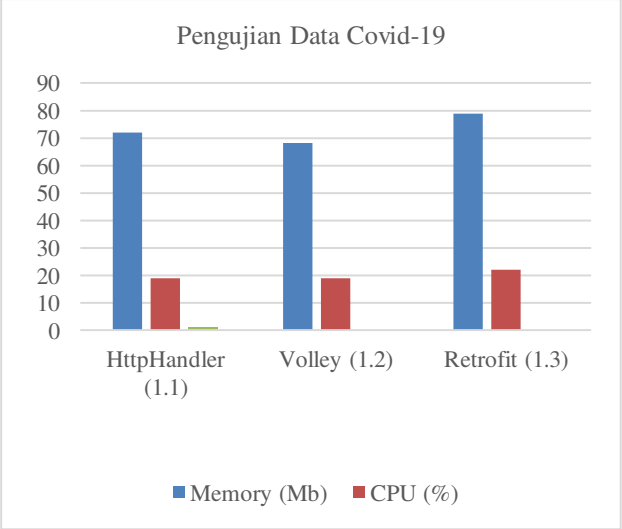
1. *Code* adalah *memory* yang digunakan oleh aplikasi untuk *resource, code dex, bytcode dex, library* dan *font*.
2. *Stack* adalah *memory* yang digunakan oleh *stack* dan *native* yang berkaitan dengan jumlah *thread*.
3. *Graphics* merupakan *memory* yang digunakan bersama CPU untuk menampilkan pixel ke layar.
4. *Native* merupakan *memory* yang digunakan untuk menangani asset gambar dan grafik lainnya yang ditulis menggunakan bahasa Java
5. Java merupakan m*emory* yang dialokasikan dari *code* java.



Gambar 9. CPU *Profile* (fungsi 1.3) pengujian ke 3 data covid-19 negara Indonesia menggunakan *library Reftrofit*

Tabel 3. Hasil tiga kali pengujian setiap fungsi 1.0 dengan tiga metode

Fungsi/ Metode	Cek Uji	Pengujian 1	2	3	Rerata
1.1	*Memory*	73,6	70,2	70,1	71,3
	Cpu	18	19	20	19
1.2	*Memory*	67,6	68	68,7	68,1
	Cpu	19	19	19	19
1.3	*Memory*	80,1	78,8	77,9	78,9
	Cpu	21	25	20	22



Gambar 10. Grafik rata-rata perbandingan fungsi 1.0 metode pemanggilan data JSON berdsarkan *memory* dan CPU.

Pengujian Fungsi 2.0

Pengujian fungsi 2.0 adalah pengujian data korban covid-19 berdasarkan provinsi di Indonesia. Fungsi 2.0 menampilkan data JSON kedalam *listview* untuk menampilkan seluruh data berdasarkan provinsi.

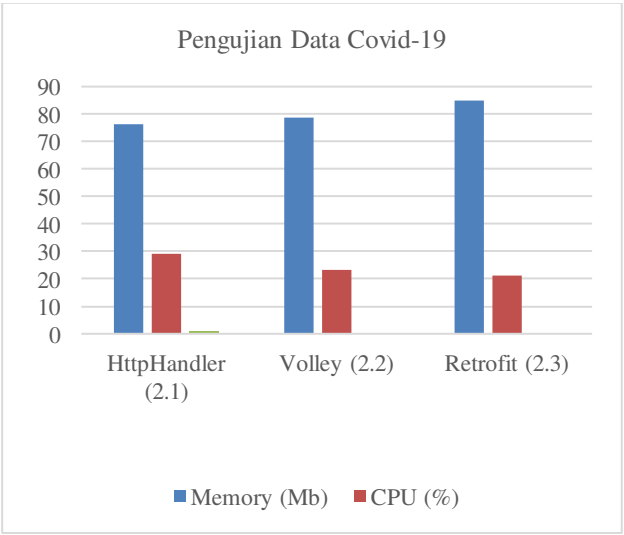
Tabel 4. Pengujian fungsi 2.0 data covid19 Indonesia

Fungsi	Metode	JB xml	JB Java
1.1	HttpHandler	172	286
1.2	Volley	172	204
1.3	Retrofit	172	234

Tabel 5. Hasil tiga kali pengujian setiap fungsi 2.0 dengan tiga metode

Fungsi/ Metode	Cek Uji	Pengujian 1	2	3	Rerata
2.1	*Memory*	78,8	73,3	76,5	76,2
	Cpu	33	20	34	29
2.2	*Memory*	78,3	79,4	78,7	78,8
	Cpu	32	20	24	25,3
2.3	*Memory*	85,2	84,9	84,5	84,8
	Cpu	21	19	24	21,3

Gambar 11. Tampilan (fungsi 2.0) data covid-19 berdasarkan provinsi negara Indonesia



Gambar 12. Grafik rata-rata perbandingan fungsi 2.0 metode pemanggilan data JSON berdasarkan *memory* dan CPU.

Pengujian Fungsi 3.0

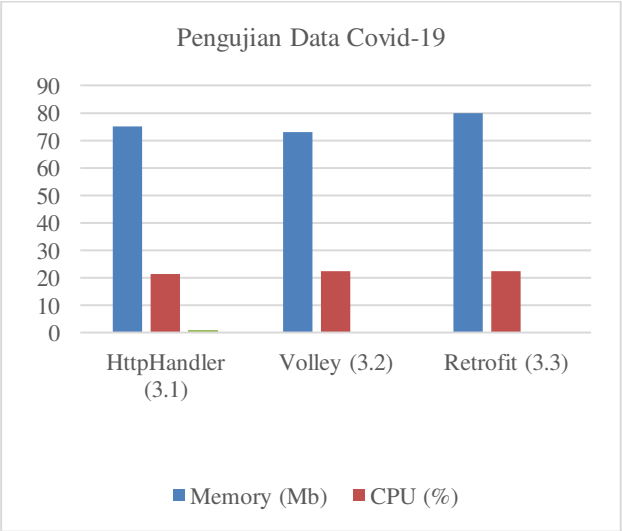
Pengujian fungsi 3.0 adalah pengujian data korban covid-19 seluruh dunia dan hanya menampilkan jumlah positif.

Tabel 6. Pengujian fungsi 3.0 data positif Covid19 seluruh dunia

Fungsi	Metode	JB xml	JB Java
3.1	HttpHandler	97	201
3.2	Volley	97	117
3.3	Retrofit	97	131

Tabel 7. Hasil tiga kali pengujian setiap fungsi 3.0 dengan tiga metode

Fungsi/ Metode	Cek Uji	Pengujian 1	2	3	Rerata
3.1	*Memory*	76,4	74,8	73,6	74,9
	Cpu	24	21	19	21,3
3.2	*Memory*	72,1	74,5	72,9	73
	Cpu	20	25	22	22.3
3.3	*Memory*	80,2	80,2	79,3	79,9
	Cpu	23	20	24	22,3



Gambar 13. Grafik rata-rata perbandingan fungsi 3.0 metode pemanggilan data JSON berdasarkan *memory* dan CPU.

5. Kesimpulan

Berdasarkan hasil penelitian pemenfaatan JSON covid-19 dengan menggunakan tiga metode dapat disimpulkan penggunaan JSON untuk pertukaran data API *rest* menggunakan metode *Model View Presenter,* sangat mudah diimplementasikan meskipun menggunakan tiga metode pemanggilan data JSON berbeda.

Pada metode HttpHandler membutuhkan baris *code* java yang lebih banyak untuk memanggil data JSON. Pemanggilan objek dan *array* menggunakan memori serta CPU yang hampir sama dengan metode *volley*. Akan tetapi metode HttpHandler memiliki kelemahan pada pemanggilan *array listview* dengan penggunaan CPU yang lebih besar. Pengujian metode *Volley* membutuhkan baris *code* java paling sedikit untuk memanggil data JSON. Selain itu penggunaan *memory* dan CPU paling kecil, meskipun pada pemanggilan *array listview* membutuhkan CPU yang besar. Pujian terakhir adalah metode *Retrofit* menggunakan *memory* dan CPU yang besar dan stabil, meskipun digunakan untuk pemanggilan data *array listview.*

Daftar Pustaka

[1] Gugus Tugas Percepatan Penanganan Covid-19, Peta sebaran, https://covid19.go.id/peta-sebaran, 2020. Diakses 15 Juli 2020.

[2] Chasseur, Craig., Li, Y., dan Patel, Jm. *Enabling JSON Document Stores in Relational Systems. Sixteenth International Workshop on the Web and Databases* (WebDB 2013). 2013.

[3] Agyun, C., & Kazan, *The Performance Analysis of Applications Written Using. International Journal of Scientific Research in Information Systems and Engineering* (IJSRISE), 1(2), 2 - 8. 2015.

[4] Crockford, Douglas. Introducing JSON, https://www.JSON.org/JSON-en.html, 2020. Diakses 16 Juli 2020.

[5] Christoforus, A., dkk. Implementasi Model View Presenter Dan Object Relutionul Mapping Nhibernate Pada Aplikasi Esnp Card Berbasis Leb (studi Kasus: Pt. Xyz Jakarta). Informatika, 8(2), 2012.

[6] Luthfiansyah, Arif., dkk., *Pengembangan Aplikasi Pemantauan Alat Berat Pertambangan menggunakan Teknologi Geofencing dengan Arsitektur MVP*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 3(8), 7616-7625. 2019.

[7] Chugh, Anupam. Android MVP Architecture, https://www.journaldev.com/14886/android-mvp, 2020. Diakses 19 Juli 2020.

[8] Jogiyanto, H.M., *Analisa dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*, ANDI, 2005.

[9] Sahrial, Rysa., JSON3metode, https://github.com/rysasahrial-99/JSON3metode. 2020. Diakses 27 Juli 2020.

[10] Android Developer, Ringkasan Volley, Training, https://developer.android.com/training/volley?hl=id. 2020. Diakses 20 Juli 2020.